



**PERANCANGAN APLIKASI SISTEM PEMESANAN RENTAL BUS DI
PANORAMA DESTINATION MENGGUNAKAN BAHASA
PEMOGRAMAN ANDROID STUDIO**

Disusun dan Diajukan Sebagai Salah Satu Syarat untuk Menempuh Ujian Akhir
Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi Medan

SKRIPSI

OLEH

NAMA : KINANTI NABELLA PUTRI
N.P.M : 1314370134
PROGRAM STUDI : SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI

2020

LEMBAR PENGESAHAN

**PERANCANGAN APLIKASI SISTEM PEMESANAN RENTAL BUS DI
PANORAMA DESTINATION MENGGUNAKAN BAHASA PEMROGRAMAN
ANDROID STUDIO**


Disusun Oleh :


Nama : Kinanti Nabella Putri
NPM : 1314370134
Program Studi : Sistem Komputer

**Skripsi telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal : 16 Juni 2020**

Dosen Pembimbing I,

Dosen Pembimbing II


Eko Hariyanto, S.Kom., M.Kom.
NIP: 7110041040


Nova Mayasari, S.Kom., M.Kom.
NIP: 1110040064

Mengetahui,

Dekan Fakultas Sains dan Teknologi

Ketua Program Studi Sistem Komputer


Hamdani, S.T., M.T.
NIP: 7110042006


Eko Hariyanto, S.Kom., M.Kom.
NIP: 7110041040

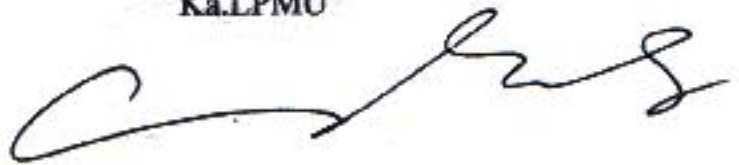
SURAT KETERANGAN PLAGIAT CHECKER

Surat ini saya Ka.LPMU UNPAB menerangkan bahwa surat ini adalah bukti pengesahan dari LPMU sebagai pengesah proses plagiat checker Tugas Akhir/ Skripsi/Tesis selama masa pandemi *Covid-19* sesuai dengan edaran rektor Nomor : 7594/13/R/2020 Tentang Pemberitahuan Perpanjangan PBM Online.

Demikian disampaikan.

Segala penyalahgunaan/pelanggaran atas surat ini akan di proses sesuai ketentuan yang berlaku UNPAB.

Ka.LPMU



Cahyo Pramono, SE.,MM

Plagiarism Detector v. 1460 - Originality Report 06/09/20 10:26:46

Uploaded document: KINANTI NABELLA PUTRI_1314370134_SYSTEM KOMPUTER.docx Universitas Pembangunan Panca Budi_License03
Comparison Preset: Rewrite. Detected language: Indonesian

Visualisasi



Distribusi grafik



Top sources of plagiarism

- 100% 100% <https://www.scribd.com/document/317274614/Prinsip-Prinsip-Perencanaan-5>
- 100% 100% <https://www.scribd.com/document/317274614/Prinsip-Prinsip-Perencanaan-5>
- 100% 100% <https://www.scribd.com/document/317274614/Prinsip-Prinsip-Perencanaan-5>

Perbandingan Simbol (0/0/0)

144 - Ok / 0 - Failed

[Show other Sources]

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Kinanti Nabella Putri
NPM : 1314370134
Program Studi : Sistem Komputer
Konsentrasi : Keamanan Jaringan Komputer
Judul Skripsi : Perancangan Aplikasi Sistem Pemesanan Rental Bus di
Panorama Destination Menggunakan Bahasa Pemrograman
Android Studio

Dengan ini menyatakan bahwa :

1. Tugas Akhir Skripsi saya bukan hasil plagiat
2. Saya tidak akan menuntut perbaikan nilai Indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau.
3. Skripsi saya dapat dipublikasikan oleh Pihak Lembaga dan saya tidak akan menuntut akibat publikasi tersebut.

Demikianlah surat pernyataan ini saya perbuat dengan sebenar – benarnya.

Terimakasih

Medan, 16 Juni 2020

Yang menyatakan



Kinanti Nabella Putri

NPM : 1314370134



YAYASAN PROF. DR. H. KADIRUN YAHYA

UNIVERSITAS PEMBANGUNAN PANCA BUDI

JL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514806
 MEDAN - INDONESIA
 Website : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id

LEMBAR BUKTI BIMBINGAN SKRIPSI

Nama Mahasiswa : KINANTI NABELLA PUTRI
 NPM : 1314370134
 Program Studi : Sistem Komputer
 Jenjang Pendidikan : Strata Satu
 Dosen Pembimbing : Eko Hariyanto, S.Kom., M.Kom
 Judul Skripsi : Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemrograman Android Studio

Tanggal	Pembahasan Materi	Status	Keterangan
29 April 2020	lanjutan penyelesaian projek kamu dan buat video penggunaanya untuk demo untuk tulisan laporan skripsi dibantu oleh doping 2	Revisi	
12 Mei 2020	Acc program, lanjutkan seminar hasil	Disetujui	
20 Mei 2020	Acc sidang meja hijau	Disetujui	
20 Agustus 2020	Acc jilid skripsi	Disetujui	

Medan, 05 November 2020
 Dosen Pembimbing,



Eko Hariyanto, S.Kom., M.Kom

YAYASAN PROF. DR. H. KADIRUN YAHYA

UNIVERSITAS PEMBANGUNAN PANCA BUDIJL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514808
MEDAN - INDONESIAWebsite : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id**LEMBAR BUKTI BIMBINGAN SKRIPSI**

Nama Mahasiswa : KINANTI NABELLA PUTRI
NPM : 1314370134
Program Studi : Sistem Komputer
Jenjang Pendidikan : Strata Satu
Dosen Pembimbing : Nova Mayasari, S. Kom., M.Kom
Judul Skripsi : Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemrograman Android Studio

Tanggal	Pembahasan Materi	Status	Keterangan
03 Mei 2020	1. Meringkan dalam penulisan bahasa asing 2. Perbaiki penulisan kutipan sesuai dengan panduan	Revisi	
13 Mei 2020	Acc Bab I dan Bab II	Disetujui	
13 Mei 2020	Acc Bab III, IV dan V	Disetujui	
13 Mei 2020	Acc Seminar Hasil	Disetujui	
06 Juni 2020	Acc Skripsi Meja Hijau	Disetujui	
19 Agustus 2020	Acc Jilid	Disetujui	

Medan, 05 November 2020
Dosen Pembimbing,

Nova Mayasari, S. Kom., M.Kom



YAYASAN PROF. DR. H. KADIRUN YAHYA
PERPUSTAKAAN UNIVERSITAS PEMBANGUNAN PANCA BUDI
Jl. Jend. Gatot Subroto KM. 4,5 Medan Sunggal, Kota Medan Kode Pos 20122

SURAT BEBAS PUSTAKA
NOMOR: 2051/PERP/BP/2020

Kepala Perpustakaan Universitas Pembangunan Panca Budi menerangkan bahwa berdasarkan data pengguna perpustakaan atas nama saudara/i:

Nama : KINANTI NABELLA PUTRI
N.P.M. : 1314370134
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Bahwasannya terhitung sejak tanggal 08 Juni 2020, dinyatakan tidak memiliki tanggungan dan atau pinjaman buku sekaligus tidak lagi terdaftar sebagai anggota Perpustakaan Universitas Pembangunan Panca Budi Medan.

Medan, 08 Juni 2020
Diketahui oleh,
Kepala Perpustakaan,



Muhammad Muttaqin, S. Kom., M.Kom.



KARTU BEBAS PRAKTIKUM
Nomor. 1168/BL/LAKO/2020

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : KINANTI NABELLA PUTRI
N.P.M. : 1314370134
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 18 Juni 2020
Ka. Laboratorium


Fachrudin Wadly, S. Kom., M. Kom.



PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di acu dalam skripsi ini dan sebutkan dalam daftar pustaka.

Medan, Juni 2020



Kinanti Nabella Putri

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau

Medan, 23 Juni 2020
 Kepada Yth : Bapak/Ibu Dekan
 Fakultas SAINS & TEKNOLOGI
 UNPA3 Medan
 Di -
 Tempat

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : KINANTI NABELLA PUTRI
 Tempat/Tgl. Lahir : BINJAI / 20 Maret 1995
 Nama Orang Tua : M. NAULANA ERWINSYAH
 N. P. M : 1314370134
 Fakultas : SAINS & TEKNOLOGI
 Program Studi : Sistem Komputer
 No. HP : 082273662278
 Alamat : Jl. Tengku Amir Hamzah, no. 568, Kel. Jati Makmur,
 Kec. Binjai Utara, Kota Binjai, Sumatera Utara

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemrograman Android Studio, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indeks prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 - 5 lembar dan 3x4 - 5 lembar Hitam Putih
6. Terlampir foto copy STTB SITA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah diprint lux 2 exemplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 exemplar untuk penguji (berikut dan warna penjiilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Sebelum menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya yang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan rincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	600,000
2. [170] Administrasi Wisuda	: Rp.	1,500,000
3. [202] Bebas Pustaka	: Rp.	100,000
4. [221] Bebas LAB	: Rp.	
Total Biaya	: Rp.	2,200,000

Periode Wisuda Ke : **65**

Ukuran Toga : **M**

Diketahui/Ditandatangani oleh :



Hamdani, ST., MT

Hormat saya



KINANTI NABELLA PUTRI



**UNIVERSITAS PEMBANGUNAN PANCABUDI
FAKULTAS ILMU KOMPUTER**

Jl. Jend. Gatot Subroto Km. 4,5 Telp. 061 – 8455571, Fax. 061 – 8458077, PO BOX 1099 MEDAN

PROGRAM STUDI SISTEM KOMPUTER (TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER (TERAKREDITASI)

PERMOHONAN MENGAJUKAN JUDUL SKRIPSI

Orang yang bertanda tangan di bawah ini :

Nama Lengkap : KINANTI NABELLA PUTRI
Tempat/Tgl. Lahir : BINJAI / 20 MARET 1995
Nomor Pokok Mahasiswa : 1314370134
Program Studi : Sistem Komputer
Konsentrasi :
Semua Kredit yang telah dicapai : 139 SKS,IPK.....
Dengan ini mengajukan judul skripsi sesuai dengan bidang ilmu, dengan judul :

Judul Skripsi	Persetujuan
Perancangan Aplikasi Pendataan Pelanggan dan Bookingan Transportasi pada Panorama Destinasi	<input type="checkbox"/>
Pembatasan Bandwith Internet Dengan Mikrotik RB951UI Pada Panorama Destinasi	<input checked="" type="checkbox"/>
Implementasi Metode Fuzzy Logic Dalam Merancang Aplikasi Sistem Pakar Penyakit Tanaman Kakao	<input type="checkbox"/>

Judul yang disertai oleh Kepala Program Studi diberikan tanda
Perancangan Aplikasi Keamanan File Data Metode File Data
Metode Reaktor Visual Studio 2010

Medan, 07 Juni 2017
Pemohon,

(Kinanti Nabella Putri)

(Irfan Ghakki Alamsyah, M.T., Ph.D.)

Nomor :
Tanggal :
Ditandatangani oleh :

M. Saifurrahman, S.Kom., M.Kom.
(M. Saifurrahman, S.Kom., M.Kom.)

Tanggal : 16 Juni 2017
Disetujui oleh:
Dosen Pembimbing I :

Andysah P. P. Sjaham, S.Kom., M.Kom.

Tanggal : 23/10 2017
Disetujui oleh :
Ka. Prodi Sistem Komputer

Eko Hariyanto, S.Kom., M.Kom.

Tanggal : 22 Agustus 2017
Disetujui oleh:
Dosen Pembimbing II :

Rizki Yandira Niska, S.Kom., M.Kom.

No. Dokumen : FM-LPPM-08-01 Revisi: 02 Tgl. Eff: 20 Des 2015



UNIVERSITAS PEMBANGUNAN PANCA BUDI

FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITAS)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITAS)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITAS)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITAS)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITAS)
PROGRAM STUDI PETERNAKAN	(TERAKREDITAS)

PERMOHONAN PRA PENGAJUAN TESIS / SKRIPSI / TUGAS AKHIR

Saya yang bertanda tangan di bawah ini :

Nama Lengkap : KINANTI NABELLA PUTRI
 Tempat/Tgl. Lahir : Binjai / 20 Maret 1995
 Nomor Pokok Mahasiswa : 1314370134
 Program Studi : Sistem Komputer
 Konsentrasi : Keamanan Jaringan Komputer
 Jumlah Kredit yang telah dicapai : 139 SKS, IPK 3.11

Dengan ini mengajukan judul skripsi sesuai dengan bidang ilmu, dengan judul:

No.	Judul	Persetujuan
1	Perancangan Aplikasi Booking Transportasi Pada Panorama Destinasi Berbasis Android	<input type="checkbox"/>
2	Pembatasan Bandwith Internet dengan Mikrotik RB551UI Pada Panorama Destinasi	<input type="checkbox"/>
3	Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemrograman Android Studio	<input checked="" type="checkbox"/> 2/3 - 20

Medan, 07 Maret 2020

Pemohon,


 (Kinanti Nabella Putri)

Tanggal : 07 Maret 2020
 Disetujui oleh :
 Dosen Pembimbing I :


 Eko Hariyanto, S.Kom., M.Kom.

Tanggal : 07 Maret 2020
 Disetujui oleh :
 Dosen Pembimbing II :


 Nova Maulana, S.Kom., M.Kom.

No. Dokumen: FM-UPBM-18-01

Revisi: 0

Tgl. Eff: 22 Oktober 2018

Sumber dokumen: <http://mahasiswa.poncabudi.ac.id>

Dicetak pada: Sabtu, 07 Maret 2020 10:33:02

ABSTRAK

KINANTI NABELLA PUTRI

Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemograman Android Studio

2020

Zaman sekarang ini *smartphone* Android mengalami perkembangan yang cepat didukung oleh kemampuan yang semakin baik serta harga yang semakin murah, maka *smartphone* dengan sistem operasi Android banyak dipilih masyarakat dalam menyelesaikan kebutuhan sehari-hari. Panorama *Destination* adalah perusahaan pelayanan peminjaman bus atau rental bus luar provinsi salah satu tujuan yang di layani Panoram *Destination* yaitu dari Medan ke Padang. Penumpang dari Panorama *Destination* memiliki kendala dalam hal pemesanan rental bus. Para penumpang harus datang ke Panorama *Destination* untuk melakukan pemesanan bus atau *booking* bus. Apabila bus sudah penuh di *booking* penumpang lainnya tentunya merasakan kerugian dan harus menunda keberangkatan, apalagi saat lebaran atau bertepatan dengan hari liburan sekolah. Maka pemesanan rental bus melonjak drastis dan banyak penumpang menjadi merasa kecewa. Demi memuaskan penumpang Panorama *Destination* ingin membuat sebuah sistem pemesanan rental bus yang praktis dan cepat dari sistem pemesanan yang ada, Untuk mengatasi masalah itu penulis ingin membuat sistem pemesanan rental bus menggunakan perangkat mobile dengan media telepon selular (hp) yang berbasis Android, dengan sistem ini penumpang Panorama *Destination* dapat langsung melakukan pemesanan dimanapun. Dengan memanfaatkan media *smartphone* Android ini diharapkan dapat menguntungkan kedua belah pihak yaitu Panorama *Destination* dan penumpang Panorama *Destination*.

Kata kunci : Panorama *Destination*, Android Studio, Android, Smartphone, Rental Bus.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul “Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemrograman Android Studio”.

Dalam kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan Skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Kepada kedua orang tua yang telah menjaga dan mengasahi saya dari kecil hingga dewasa.
2. Bapak Dr. H.M. Isa Indrawan, SE.,MM., selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Dekan Fakultas Sains dan Teknologi, Bapak Hamdani ST., MT.
4. Ketua Program Studi Sistem Komputer sekaligus Dosen Pembimbing I, Bapak Eko Hariyanto, S.Kom., M.Kom.
5. Dosen Pembimbing II, Ibu Nova Mayasari, S.Kom., M.Kom.
6. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
7. Manager Panorama Destination telah mengizinkan saya untuk melakukan penelitian di kantor tersebut.

8. Para karyawan Panorama Destination yang telah membantu saya dalam memberikan informasi.
9. Para sahabat dan teman yang telah memberikan semangat saya untuk dapat menyelesaikan skripsi ini.

Akhir kata penulis sampaikan rasa terima kasih bagi semua pihak yang secara langsung terlibat dalam penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu persatu. Semoga skripsi ini memberikan manfaat bagi penulis khususnya dan bagi kita semua umumnya.

Medan, 16 Juni 2020
Penulis,

KINANTI NABELLA PUTRI
1314370134

DAFTAR ISI

Halaman

HALAMAN JUDUL	
HALAMAN PENGESAHAN	
SURAT PERNYATAAN	
PERNYATAAN ORISINALITAS	
ABSTRAK	
KATA PENGANTAR	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan	3
1.5 Manfaat Penulisan	4
BAB II LANDASAN TEORI.....	5
2.1 Perancangan	5
2.2 Sistem	6
2.3 Aplikasi	8
2.4 Rental	9
2.5 Konsumen	10
2.6 Transportasi	10
2.7 Pengertian UML.....	11
2.7.1 <i>Use Case Diagram</i>	11
2.7.2 <i>Class Diagram</i>	12
2.7.3 <i>Sequence Diagram</i>	14
2.7.4 <i>Activity Diagram</i>	15
2.8 Android	16
2.9 Android SDK	17
2.10 SQLite	19
2.11 JDK (<i>Java Development Kit</i>)	20
2.12 Android Studio	21
2.13 <i>Database</i>	21
2.14 MySQL	24
BAB III METODE PENELITIAN.....	26
3.1 Tahapan Penelitian	26
3.2 Metode Pengumpulan Data	28

3.3 Analisis Sistem Sedang Berjalan	29
3.4 Rancangan Penelitian	32
3.4.1 <i>User Case Diagram</i>	32
3.4.2 <i>Activity Diagram</i>	32
3.4.3 <i>Sequence Diagram</i>	35
3.4.4 <i>Class Diagram</i>	36
3.5 Rancangan <i>Database</i>	37
1. Tabel Admin	37
2. Tabel Member	38
3. Tabel Paket	38
4. Tabel Pesanan	39
3.6 Rancangan <i>Interface</i>	39
3.6.1 Rancangan Halaman Home.....	39
3.6.2 Rancangan Halaman Paket	41
3.6.3 Rancangan Halaman Login.....	42
3.6.4 Rancangan Halaman Daftar	43
3.6.5 Rancangan Halaman Home Admin.....	44
3.6.6 Rancangan Halaman Data Member	45
3.6.7 Rancangan Halaman Data Paket	46
3.6.8 Rancangan Halaman Input Paket	47
3.6.9 Rancangan Halaman Pesanan	48
3.6.10 Rancangan Halaman Konfirmasi	49
3.6.11 Rancangan Halaman Pesan Paket	50
3.6.12 Rancangan Halaman Pengajuan	51
BAB IV HASIL DAN PEMBAHASAN.....	52
4.1 Pengujian Aplikasi	52
4.2 Tampilan Program	52
4.2.1 Tampilan Halaman Home	52
4.2.2 Tampilan Halaman Paket	54
4.2.3 Tampilan Halaman Login	55
4.2.4 Tampilan Halaman Daftar	56
4.2.5 Tampilan Halaman Home Admin	57
4.2.6 Tampilan Halaman Data Member	58
4.2.7 Tampilan Halaman Data Paket	59
4.2.8 Tampilan Halaman Input Paket	60
4.2.9 Tampilan Halaman Pesanan	61
4.2.10 Tampilan Halaman Konfirmasi	62
4.2.11 Tampilan Halaman Pesan Paket	63
4.2.12 Tampilan Halaman Pengajuan	64
BAB V PENUTUP	65

5.1 Kesimpulan	65
5.2 Saran	65

DAFTAR PUSTAKA
BIOGRAFI PENULIS
LAMPIRAN – LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Tahapan Metode Penelitian	26
Gambar 3.2 Flowmap Sistem yang Berjalan	31
Gambar 3.3 <i>Use Case Diagram</i>	32
Gambar 3.4 <i>Activity Diagram</i> Admin	33
Gambar 3.5 <i>Activity Diagram</i> Member.....	34
Gambar 3.6 <i>Sequence Diagram</i> Admin	35
Gambar 3.7 <i>Sequence Diagram</i> Member.....	36
Gambar 3.8 <i>Class Diagram</i>	37
Gambar 3.9 Rancangan Halaman Home	40
Gambar 3.10 Rancangan Halaman Paket	41
Gambar 3.11 Rancangan Halaman Login	42
Gambar 3.12 Rancangan Halaman Daftar	43
Gambar 3.13 Rancangan Halaman Home Admin.....	44
Gambar 3.14 Rancangan Halaman Data Member	45
Gambar 3.15 Rancangan Halaman Data Paket	46
Gambar 3.16 Rancangan Halaman Input Paket	47
Gambar 3.17 Rancangan Halaman Pesanan	48
Gambar 3.18 Rancangan Halaman Konfirmasi	49
Gambar 3.19 Rancangan Halaman Pesan Paket	50
Gambar 3.20 Rancangan Halaman Pengajuan.....	51
Gambar 4.1 Rancangan Halaman Home	53
Gambar 4.2 Rancangan Halaman Paket	54
Gambar 4.3 Rancangan Halaman Login	55
Gambar 4.4 Rancangan Halaman Daftar	56
Gambar 4.5 Rancangan Halaman Home Admin.....	57
Gambar 4.6 Rancangan Halaman Data Member	58
Gambar 4.7 Rancangan Halaman Data Paket	59
Gambar 4.8 Rancangan Halaman Input Paket	60
Gambar 4.9 Rancangan Halaman Pesanan	61
Gambar 4.10 Rancangan Halaman Konfirmasi	62
Gambar 4.11 Rancangan Halaman Pesan Paket	63
Gambar 4.12 Rancangan Halaman Pengajuan.....	64

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol <i>Use Case</i>	12
Tabel 2.2 Simbol <i>Class Diagram</i>	13
Tabel 2.3 Simbol <i>Sequence Diagram</i>	14
Tabel 2.4 Simbol Diagram Aktivitas	16
Tabel 3.1 Tabel Admin	38
Tabel 3.2 Tabel Member	38
Tabel 3.3 Tabel Paket.....	39
Tabel 3.4 Tabel Pesanan	39

BAB I

PENDAHULUAN

1.1 Latar Belakang

Zaman sekarang ini *smartphone* Android mengalami perkembangan yang cepat didukung oleh kemampuan yang semakin baik serta harga yang semakin murah, maka *smartphone* dengan sistem operasi Android banyak dipilih masyarakat dalam menyelesaikan kebutuhan sehari-hari. Mengikuti perkembangan teknologi dibidang *mobile device* yang diterapkan pada sarana transportasi sangat jarang ditemukan pada transportasi bus yang melalui perangkat *mobile*, dimana perangkat *mobile* sudah didukung fitur internet. Hal ini dapat dimanfaatkan oleh perusahaan jasa transportasi darat untuk mengatasi permasalahan penyajian informasi khususnya dalam ketersediaan bus dan proses pemesanan yang masih bersifat manual.

Panorama *Destination* adalah perusahaan biro perjalanan wisata, salah satunya menyediakan peminjaman bus atau rental bus dalam dan luar provinsi, salah satu tujuan yang di layani Panorama *Destination* yaitu dari Medan ke Padang. Pelanggan dari Panorama *Destination* memiliki kendala dalam hal pemesanan rental bus. Para pelanggan harus datang ke Panorama *Destination* untuk melakukan pemesanan bus atau *booking* bus. Apabila bus sudah penuh di *booking*, pelanggan lainnya tentunya merasakan kerugian dan harus menunda keberangkatan, apalagi saat *weekend* atau bertepatan dengan hari libur Nasional . Maka biasanya pemesanan rental bus melonjak drastis dan banyak pelanggan menjadi merasa kecewa.

Demi memuaskan pelanggan, Panorama *Destination* ingin membuat sebuah sistem pemesanan rental bus yang praktis dan cepat dari sistem pemesanan yang ada, Untuk mengatasi masalah itu penulis ingin membuat sistem pemesanan rental bus menggunakan perangkat *mobile* dengan media telepon selular (hp) yang berbasis Android, dengan sistem ini penumpang Panorama *Destination* dapat langsung melakukan pemesanan dimanapun. Dengan memanfaatkan media smartphone Android ini diharapkan dapat menguntungkan kedua belah pihak yaitu Panorama *Destination* dan pelanggan Panorama *Destination*

Maka dari latar belakang tersebut penulis mencoba untuk melakukan perancangan aplikasi sistem “**Perancangan Aplikasi Sistem Pemesanan Rental Bus di Panorama Destination Menggunakan Bahasa Pemograman Android Studio**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka penulis merumuskan masalah sebagai berikut :

1. Bagaimana membangun aplikasi sistem pemesan rental bus berbasis *mobile* android di Panorama *Destination* ?
2. Bagaimana melakukan pemesanan rental bus di Panorama *Destination* yang dapat digunakan oleh pelanggan?
3. Bagaimana mengimplementasikan aplikasi sistem pemesan rental bus pada Panorama *Destination* berbasis *mobile* ?

1.3 Batasan Masalah

Ruang lingkup batasan masalah pada perancangan aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman Android Studio ini adalah:

1. Aplikasi pemesanan rental bus di Panorama *Destination* hanya tersedia bagi pengguna android.
2. Aplikasi pemesanan rental bus hanya digunakan untuk *booking* bus yang akan dipesan.
3. Perangkat Lunak yang digunakan dalam membangun aplikasi pemesanan rental di Panorama *Destination* adalah *Android Studio* sebagai bahasa pemrograman, *MySQL* sebagai *Database Server* dan *SQLite* sebagai database internalnya.

1.4 Tujuan Penelitian

Tujuan perancangan aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman Android Studio adalah sebagai berikut:

1. Dapat mempermudah pegawai Panaroma *Destination* konsumen dan dalam melakukan pemesanan rental bus.

2. Dengan adanya aplikasi pemesanan rental bus dapat memberikan kemudahan dalam proses *booking* bus yang ingin di pesan oleh konsumen, sehingga konsumen tidak harus datang ke lokasi Panorama *Destination*.
3. Aplikasi pemesanan rental bus dapat diimplementasikan dengan mudah di Panorama *Destination* karena aplikasi yang dibuat mudah digunakan.

1.5 Manfaat Penelitian

Manfaat aplikasi sistem pemesan rental bus di Panorama *Destination* menggunakan bahasa pemograman Android Studio adalah sebagai berikut:

1. Mempermudah konsumen melakukan transaksi pemesanan rental bus untuk menghemat waktu, tenaga dan biaya.
2. Memanfaatkan teknologi *smartphone* yang kini semakin maju sehingga konsumen dapat memperoleh pelayanan *booking* bus secara cepat.
3. Meningkatkan efektivitas kerja pegawai Panorama *Destination*.

BAB II

LANDASAN TEORI

2.1 Perancangan

Perancangan menggambarkan rencana umum suatu kegiatan rancangan proyek atau aktivitas-aktivitas khusus yaitu teknik atau metode-metode dalam merancang sesuatu. Adapun tujuan dari strategi perancangan adalah memberikan kepastian apakah aktivitas-aktivitas tersebut benar-benar realistik dengan batasan waktu dan sumber-sumber yang telah ditetapkan. Berdasarkan tujuan inilah perancangan akan bekerja dalam melakukan perancangan. Strategi perancangan terbagi atas dua, yaitu:

1. Perancangan secara acak (*Random Search Strategy*)
2. Perancangan secara pasti atau berdasarkan urutan-urutan yang telah ditentukan (*Prefabricated*)

Dalam berbagai hal, strategi perancangan bertujuan untuk melakukan pendekatan dalam menyelesaikan suatu masalah yang sesuai dengan kepuasan rancangan. Taktik yang relevan terlihat dari penggunaan teknik yang umum dan metode-metode yang rasional.

Sebenarnya hakikat dasar dari proses perancangan dalam kaitannya dengan penghasilan bentuk, tidaklah berbeda antara seseorang dengan orang lain. Perbedaannya terletak pada prinsip-prinsip yang dianut dan metode-metode digunakan.

Perbedaan ini antara lain dipengaruhi oleh ideologi yang dianut oleh seseorang seperti: perbedaan kepribadian, perbedaan latar belakang budaya, perbedaan pelatihan profesional, dan juga perbedaan cara berpikir. Sehingga biasanya seseorang perancang terlibat dalam suatu aktivitas mental yang rumit dalam menghasilkan pemecahan perancangannya (**Arif, 2016**).

Pada evaluasi hasil perancangan produk terdiri dari beberapa langkah yaitu:

- a. Memonitor perubahan fungsi
- b. Evaluasi kinerja produk hasil rancangan melalui pembuatan model atau prototipe:
 - 1) Pengembangan model analitik
 - 2) Pengembangan model fisik
 - 3) Pengembangan model grafik
- c. Evaluasi biaya
- d. Evaluasi lain: yaitu dari segi:
 - 1) Kemudahan perakitan
 - 2) Analisa kegagalan
 - 3) Kemudahan perawatan
 - 4) Kemudahan pengerjaan

2.2 Sistem

Pengertian sistem menurut para ahli, konsep dasar sistem:

- a. Menurut Fat pengertian sistem adalah sebagai berikut: Sistem adalah suatu himpunan benda nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, ketergantungan, saling mendukung yang secara keseluruhan bersatu dalam satu kesatuan (*unity*) untuk mencapai tujuan tertentu secara efisien dan efektif.
- b. Pengertian sistem menurut Indrajit mengemukakan bahwa mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya.
- c. Pengertian sistem menurut Jogianto mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian yang nyata adalah suatu obyek, nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi.

Dengan demikian sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan di dalam sistem. Menurut *Richard F. Neuschel* suatu prosedur adalah suatu urutan operasi kelikual (tulis-menulis), yang melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi (**Henny et al, 2018**).

2.3 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user. Pengertian aplikasi menurut para ahli :

1. Pengertian aplikasi menurut **Jogiyanto (1999:12)** adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian sehingga komputer dapat memproses input menjadi output.
2. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang system untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

3. Menurut Wikipedia, aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna (**Adi dan Sri, 2017**).

2.4 Rental

Dalam hal ini tentu saja barang yang disewakan adalah mobil. Suatu perusahaan yang memiliki teknologi informasi yang bagus cenderung lebih cepat dalam perkembangan perusahaannya. Teknologi informasi merujuk pada seluruh bentuk teknologi yang digunakan untuk menciptakan, menyimpan, mengubah, dan untuk menggunakan informasi tersebut dalam segala bentuknya. Dan dalam hal ini komputer diharapkan tidak hanya berfungsi sebagai mesin ketik saja, tetapi juga sebagai alat pengolahan data yang mempunyai produktifitas tinggi.

Penyewaan adalah pemakaian sesuatu barang dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjamkan sesuatu, yang dibayarkan karena memakai atau meminjamkan sesuatu, yang boleh dipakai dengan membayar uang dengan uang. Sedangkan pengertian penyewaan adalah proses, cara pembuatan menyewa atau menyewakan. Rental merupakan suatu usaha dibidang jasa yang dalam kegiatan usahanya terdapat unsur sewa menyewa yang didalamnya terdapat suatu perjanjian atau kesepakatan dimana penyewa harus membayarkan atau memberikan imbalan kepada pemilik barang yang dipinjamkan (**Ranu et al, 2019**).

2.5 Konsumen

Dalam Pasal 1 Undang-undang No.8 tahun 1999 tentang Perlindungan Konsumen (UUPK), menyatakan mengenai pengertian konsumen yaitu bahwa konsumen adalah setiap orang pemakai barang dan/atau jasa yang tersedia dalam masyarakat, baik bagi kepentingan diri sendiri, keluarga, orang lain maupun makhluk hidup lain dan tidak untuk diperdagangkan. Sedangkan dalam kamus hukum, pengertian konsumen adalah pihak yang menggunakan atau memanfaatkan baik barang maupun jasa, untuk kepentingan diri sendiri maupun untuk kepentingan orang lain.

Konsumen juga dapat seorang individu maupun organisasi, yang masing-masing memiliki peran yang berbeda. mereka juga bisa berperan sebagai, *influencer*, *initiator*, *buyer*, *payer* ataupun *user*, semuanya bergantung pada produk apa yang akan dibeli oleh konsumen. Selain itu, konsumen merupakan bagian dari hukum atau aturan perlindungan konsumen yang memuat asas-asas, aturan-aturan atau kaidah-kaidah yang bersifat mengatur dan memiliki sifat yang memberikan perlindungan bagi konsumen (**Widi dan Mira, 2017**).

2.6 Transportasi

Transportasi adalah perpindahan orang atau barang dengan menggunakan alat atau kendaraan dari dan ke tempat - tempat yang terpisah secara geografis. Transportasi adalah perpindahan barang atau penumpang dari suatu tempat ke tempat lain, dimana produk dipindahkan ke tempat tujuan dibutuhkan. Dan secara umum

transportasi adalah suatu kegiatan memindahkan sesuatu (barang dan atau barang) dari suatu tempat ke tempat lain, baik dengan atau tanpa sarana. perpindahan manusia atau barang dari satu tempat ke tempat lainnya dengan menggunakan sebuah kendaraan yang digerakkan oleh manusia atau mesin. Transportasi digunakan untuk memudahkan manusia dalam melakukan aktivitas sehari-hari. Di negara maju, mereka biasanya menggunakan kereta bawah tanah (*subway*) dan taksi. Penduduk di sana jarang yang mempunyai kendaraan pribadi karena mereka sebagian besar menggunakan angkutan umum sebagai transportasi mereka. Transportasi sendiri dibagi 3 yaitu, transportasi darat, laut, dan udara. Transportasi udara merupakan transportasi yang membutuhkan banyak uang untuk memakainya. Selain karena memiliki teknologi yang lebih canggih, transportasi udara merupakan alat transportasi tercepat dibandingkan dengan alat transportasi lainnya (**Abdul dan Alvino, 2016**).

2.7 Pengertian UML

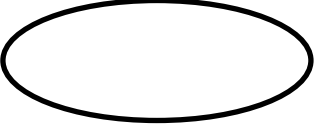
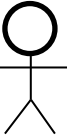
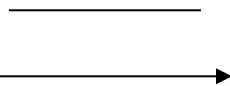
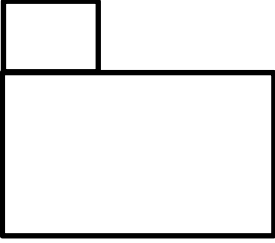
Unified Modeling Language (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak (**Abdul dan Alvino, 2016**).

2.7.1 Use case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Berikut ini adalah simbol-simbol yang ada pada diagram *use case* (Abdul dan Alvino, 2016).

Tabel 2.1. Simbol *Use case*

Simbol	Keterangan
	<p><i>Use case</i></p> <p>Menggambarkan fungsionalitas sistem.</p>
	<p>Aktor</p> <p>Menggambarkan sesuatu (entitas) yang berhubungan dengan sistem berpartisipasi dalam <i>use case</i></p>
	<p>Asosiasi</p> <p>Menggambarkan bagaimana hubungan antara <i>use case</i> dengan <i>use case</i>, dan <i>actor</i> dengan <i>actor</i> atau antara <i>actor</i> dengan <i>use case</i></p>
	<p><i>Package</i></p> <p>Digunakan ketika diagram mempunyai banyak <i>bubbles</i> dan tidak bisa dicetak dalam satu halaman</p>

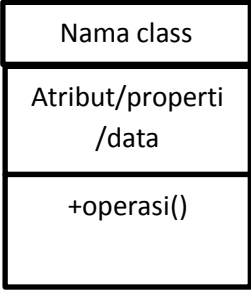
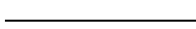
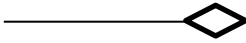
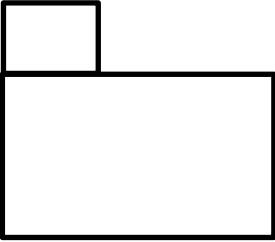
Sumber: (Abdul & Alvino. 2016)

2.7.2 *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan

di dalam kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron, Berikut adalah tabel simbol yang digunakan dalam pembuatan class diagram (Abdul dan Alvino, 2016).

Tabel 2.2. Simbol *Class Diagram*

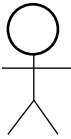

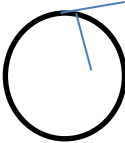
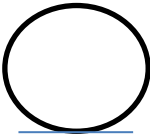
Simbol	Keterangan
	<p><i>Class</i></p> <p>Menggambarkan himpunan <i>object</i> sejenis yang mempunyai atribut dan metode yang sama.</p>
	<p><i>Asociation</i></p> <p>Menggambarkan hubungan yang terjadi antar <i>class</i>.</p>
	<p><i>Agregasi</i></p> <p>Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu kelas adalah bagian dari kelas lain</p>
	<p><i>Package</i></p> <p>Digunakan ketiak diagram mempunyai banyak <i>bubbles</i> dan tidak bisa dicetak dalam satu halaman</p>




Sumber: (Abdul & Alvino. 2016)

2.7.3 *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Berikut adalah simbol yang digunakan dalam pembuatan *sequence diagram* (Abdul dan Alvino, 2016).

Tabel 2.3. Simbol *Sequence Diagram*

Simbol	Keterangan
	<p><i>Actor Object</i></p> <p>Menggambarkan pihak yang melakukan interaksi atau yang memicu sistem untuk berfungsi</p>
	<p><i>Boundary Object</i></p> <p>Menggambarkan <i>object</i> yang menjadi <i>interface</i></p>
	<p><i>Control Object</i></p> <p>Menggambarkan <i>object</i> yang menjadi <i>control</i> sistem</p>
	<p><i>Entity Object</i></p> <p>Menggambarkan <i>object</i> yang berubah entitas</p>







	<p><i>Lifeline</i></p> <p>Menggambarkan eksekusi objek selama <i>Sequence</i></p>
	<p><i>Message</i></p> <p>Menggambarkan komunikasi yang terjadi antar objek</p>
	<p><i>Activation</i></p> <p>Menunjukkan periode selama suatu objek atau aktir sedang melakukan suatu tindakan</p>

Sumber: (Abdul & Alvino. 2016)

2.7.4 Activity Diagram

Diagram aktivitas atau Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem, Berikut adalah tabel simbol yang digunakan dalam pembuatan activity diagram (Abdul dan Alvino, 2016).

Tabel 2.4. Simbol Diagram Aktivitas

Simbol	Keterangan
	Titik Awal Titik awal aktivitas sistem, sebuah diagram aktivitas memiliki titik awal
	Aktivitas Aktivitas yang dilakuakn sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Titik akhir Titik akhir yang dilakuakn sistem sebuah diagram aktivitas memiliki sebuah titik akhir
	<i>Swilane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap jawab terhadap aktivitas yang terjadi

Sumber: (Abdul & Alvino. 2016)

2.8 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer *tablet*. Android awalnya

dikembangkan oleh Android, Inc dengan dukungan finansial Google, yang kemudian membelinya pada tahun 2005.

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, *Intel*, *Motorola*, *Qualcomm*, *TMobile*, dan *Nvidia*. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode - kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler (Harni dan Nicky, 2016).

2.9 Android SDK (*Software Deveploment Kit*)

Android SDK dalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemograman Java. Pada Android SDK ini terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi untuk platform Android SDK, Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan,

emulator dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis dengan bahasa pemrograman Java dan berjalan di Dalvik, mesin virtual yang dirancang khusus untuk penggunaan *embedded* yang berjalan diatas kernel Linux (Efmi, 2018).

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Beberapa fitur Android yang penting adalah sebagai berikut :

1. *Framework* aplikasi yang mendukung penggantian komponen dan reusable.
2. DVM dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *engine open source* WebKit.
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0.
5. SQLite untuk penyimpanan data.
6. Dukungan untuk audio, video dan gambar.
7. Bluetooth, Edge, 3G, Wifi.
8. Kamera, GPS, kompas dan *accelerometer*. Lingkungan *development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori serta *plugins* untuk IDE Eclipse (Harni dan Nicky, 2016).

2.10 SQLite

SQLite merupakan sebuah *system* management basis data *relasional* yang bersifat ACID - *compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikerjakan oleh *D. Richard Hipp*. SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92, termasuk transaksi yang bersifat *atomic*, konsistensi basis data, isolasi, dan durabilitas (dalam bahasa Inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks (**Harni dan Nicky, 2016**).

SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL, *interface*, dan penggunaan memori yang sangat sedikit dengan kecepatan yang sangat cepat. Dengan adanya SQLite *database* ini, banyak membantu dalam pembuatan berbagai versi android karena SQLite ini termasuk kedalam android *runtime*. Untuk pengembangannya, dalam membuat dan membuka *database* digunakan sebuah *libraries* yang harus kita *import*, yaitu `android.database.sqlite`. *SQLite Open Helper* yang menyediakan tiga *method*, adapun ketiga *method* tersebut adalah sebagai berikut :

1. *Constructor*

Menyediakan representasi versi dari *database* dan skema yang kita gunakan.

2. *onCreate()*

Menyediakan *SQLiteDatabase object* yang kita gunakan dalam definisi table dan inisialisasi data.

3. *onUpgrade()*

Menyediakan fasilitas konversi *database* dari *database* yang lama ke *database* versi yang baru atau sebaliknya. (Efmi, 2018)

2.11 JDK (*Java Development Kit*)

Java adalah sebuah teknologi yang diperkenalkan oleh *Sun Microsystems* pada pertengahan tahun 1990. Menurut definisi *Sun*, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada *computer standalone* ataupun pada lingkungan jaringan. Untuk membuat program Java dibutuhkan kompiler dan interpreter untuk program Java berbentuk *Java Development Kit* (JDK) yang diproduksi oleh *Sun Microsystems*. Sebelum memulai instalasi Android SDK, terlebih dahulu kita harus melakukan instalasi JDK di komputer. JDK yang kami gunakan untuk dapat mengompilasi aplikasi Android yang kami rancang ini adalah *Java SE Development Kit 8* (Harni dan Nicky, 2016).

JDK (*Java Development Kit*) adalah Paket fungsi API untuk bahasa pemrograman Java, meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM) (Efmi, 2018).

2.12 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) utama Google untuk mengembangkan pada *platform* Android. Karena Android Studio merupakan IDE dari Google, maka *software* ini dapat secara langsung terintegrasi dengan Google *Maps* menggunakan *API Key* yang dibuat di laman yang disediakan dari Google *Maps* API untuk mengintegrasikan peta dengan *software* sehingga peta akan secara otomatis ditampilkan di aplikasi yang dibuat. Selain terintegrasi dengan Google Maps, Android Studio juga dapat terintegrasi dengan *database* SQLite Manager, plugin untuk pengolahan dan penyimpanan informasi yang saling berkaitan untuk kemudian dibuat algoritma dari tiap data yang akan ditampilkan (Anisa et al, 2017).

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android (Saddam et al, 2018).

2.13 Database

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data atau arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data atau arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari

arsip menggunakan lemari sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti (*flasdisk, hardisk, MicroSD*), yang perlu diingat adalah bahwa tidak semua bentuk penyimpanan data secara elektronik bisa disebut basis data. Yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilihan, pengelompokan, pengorganisasian data yang akan kita simpan sesuai fungsi atau jenisnya. Pemilihan atau pengelompokan ini dapat berbentuk pendefinisian kolom-kolom atau *field-field* data dalam setiap file atau tabel.

Basis data digunakan untuk mengatasi permasalahan pengolahan data dengan metode pengarsipan berkas data sangat diperlukan dalam membangun sistem informasi pada sebuah perusahaan sehingga dengan basis data dapat meningkatkan kinerja perusahaan dan meningkatkan daya saing perusahaan. Karena basis data dibangun dengan tujuan sebagai berikut:

a. Kecepatan dan kemudahan

Dengan memanfaatkan basis data, memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan terhadap data atau menampilkan kembali data tersebut secara lebih efektif.

b. Efisiensi ruang penyimpanan

Karena keterkaitan yang erat antara kelompok data dalam sebuah basis data, maka redundansi pasti akan selalu ada, sehingga akan memperbesar ruang penyimpanan.

c. Keakuratan

Pengkodeaan atau pembentukan relasi antar data bersama dengan penerapan aturan (constraint), keunikan data, yang secara ketat dapat diterapkan dalam sebuah basis data sangat berguna untuk menekan ketidakakuratan penyimpanan data.

d. Ketersediaan

Dengan pemanfaatan jaringan komputer, maka data yang berada di suatu lokasi dapat diakses dari lokasi lain

e. Kelengkapan

Kelengkapan data yang disimpan dalam sebuah database bersifat relatif, bisa jadi saat ini dianggap sudah lengkap, tetapi belum tentu pada suatu saat dianggap lengkap. Untuk mengakomodasi kelengkapan data.

f. Keamanan

Aspek keamanan dapat diterapkan dengan ketat, dengan begitu dapat menentukan pemakaian basis data serta objek-objek didalamnya, serta jenis-jenis operasi apa saja yang boleh dilakukannya.

g. Kebersamaan

Basis data yang dikelola dengan aplikasi multiuser dapat memenuhi kebutuhan (Robi, 2016).

2.14 MySQL

MySQL adalah suatu perangkat lunak database relasi (Relational Database Management System atau DBMS), seperti halnya ORACLE, POSTGRESQL, MSSQL, dan sebagainya. SQL merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa program yang digunakan untuk mengelola suatu *database*. Jadi MySQL adalah software-nya dan SQL adalah bahasa perintahnya.

MySQL bersifat RDBMS (*Relational Database Management System*) RDBMS memungkinkan seorang admin dapat menyimpan banyak informasi kedalam table-table, dimana table-table tersebut saling berkaitan satu sama lain, Keuntungan RDBMS sendiri adalah kita dapat memecah database kedalam *table* table yang berbeda, setiap *table* memiliki informasi yang berkaitan dengan *table* yang lainnya.

Kelebihan MySQL banyak digunakan karna memiliki beberapa kelebihan, yaitu

1. Gratis, Sama dengan PHP, MySQL bersifat *opensource*, semua orang bebas menggunakannya tanpa harus membayar sepeser pun.
2. Cross Platform, MySQL dapat digunakan *under windows*, ataupun *under linux*.
3. Lengkap dan cepat, Pasangan yang cocok dengan PHP. Wajar jika banyak hosting saat ini mendukung adanya PHP dan MySQL karena kecepatan, gratis, dan dapat di jalankan di sistem operasi manapun.

Kekurangan MySQL Selain kelebihan MySQL, MySQL juga mempunyai kekurangan, diantaranya :

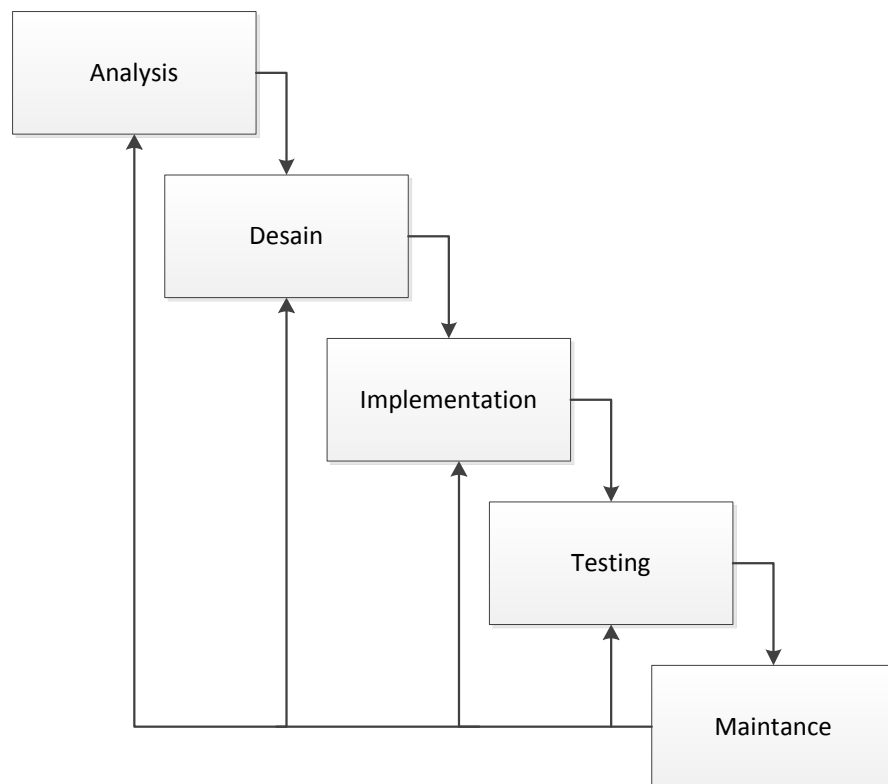
1. Untuk koneksi ke bahasa pemrograman visual seperti vb, delphi, dan foxpro, mysql kurang *support*, karena koneksi ini menyebabkan field yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan mysql jarang dipakai dalam program visual.
2. Data yang ditangani belum begitu besar
3. Dari sisi *security*, atau keamanan, yang agak terlalu sederhana bagi sebuah SQL Engine, meskipun tidak sesederhana SQLite yang juga datang dari dunia *Open Source* dan cukup digemari para *Web Developer* (**Abdul dan Alvino, 2016**).

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian untuk menyelesaikan skripsi mengenai perancangan aplikasi sistem pemesanan rental bus di Panorama Destination menggunakan bahasa pemrograman android studio ini, metode pembangunan perangkat lunak yang digunakan yaitu model *Waterfall*, meliputi beberapa proses:



Gambar 3.1. Tahapan Metode Penelitian

Berikut ini merupakan penjelasan dari gambar alur sistem *waterfall*:

1. *Analysis* (Analisa)

Pada tahapan penelitian *analysis* sistem sangat diperlukan untuk mencari data-data yang terkait atau yang berhubungan aplikasi sistem pemesanan rental bus di Panorama Destination menggunakan bahasa pemrograman android studio, seperti paket rental bus, cara pemesanan, dan harga dari setiap paket. Data-data tersebut akan memberikan kemudahan pada saat pembuatan program.

2. *Design* (Desain)

Pada tahapan penelitian *design* diperlukan sebuah alur program UML seperti *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram* sebelum melakukan perancangan *design* program, hal ini untuk mempermudah dalam pembuatan program aplikasi pemesanan rental bus agar tidak terjadi kesalahan dalam pembuatan sistem pendukung keputusan tersebut.

3. *Implementation* (Implementasi)

Pada tahapan penelitian selanjutnya adalah implementasi program atau mencoba program yang sudah jadi. Program tersebut di implementasikan di Panorama Destination.

4. *Testing*

Pada tahapan penelitian *testing*, program yang sudah di instal pada Panaroma Destination akan dilakukan uji coba, untuk melihat apakah aplikasi pemesanan bus rental tersebut dapat berjalan dengan baik atau tidak. Jika program berjalan dengan baik maka bisa dikatakan program sukses dibuat. Jika terjadi eror maka sistem informasi tersebut akan dilakukan perbaikan.

5. *Maintance*

Pada tahapan penelitian *maintance* diperlukan untuk menjaga dan memerlihara aplikasi sistem pemesanan rental bus agar tidak terjadi eror pada sistem tersebut. Dengan melakukan *backup* data sebulan sekali maka program akan terhindar dari eror.

3.2 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian aplikasi sistem pemesanan rental bus pada Panorama Destination adalah :

a. Studi Literatur

Pada tahap ini penulisan dimulai dengan proses pengumpulan bahan referensi baik dari buku, artikel, jurnal, makalah maupun melalui media internet mengenai sistem aplikasi android, sistem informasi pemesanan, serta beberapa referensi lainnya untuk menunjang pencapaian tujuan skripsi.

b. Studi Lapangan

Studi lapangan yang dilakukan penulis secara langsung di *Panorama Destination*, untuk menganalisa sistem manual yang berjalan. Merupakan sumber data yang diperoleh sesuai dengan kebutuhan yang digunakan. Teknik pengumpulan data yang digunakan adalah:

- **Obervasi**

Obervasi yaitu pengumpulan data yang dilakukan dengan melakukan pengamatan pada objek penelitian berupa pencarian data-data yang berkaitan dengan data pemesanan rental bus, paket-paket dan harga dari setiap yang akan dibuat.

- **Wawancara**

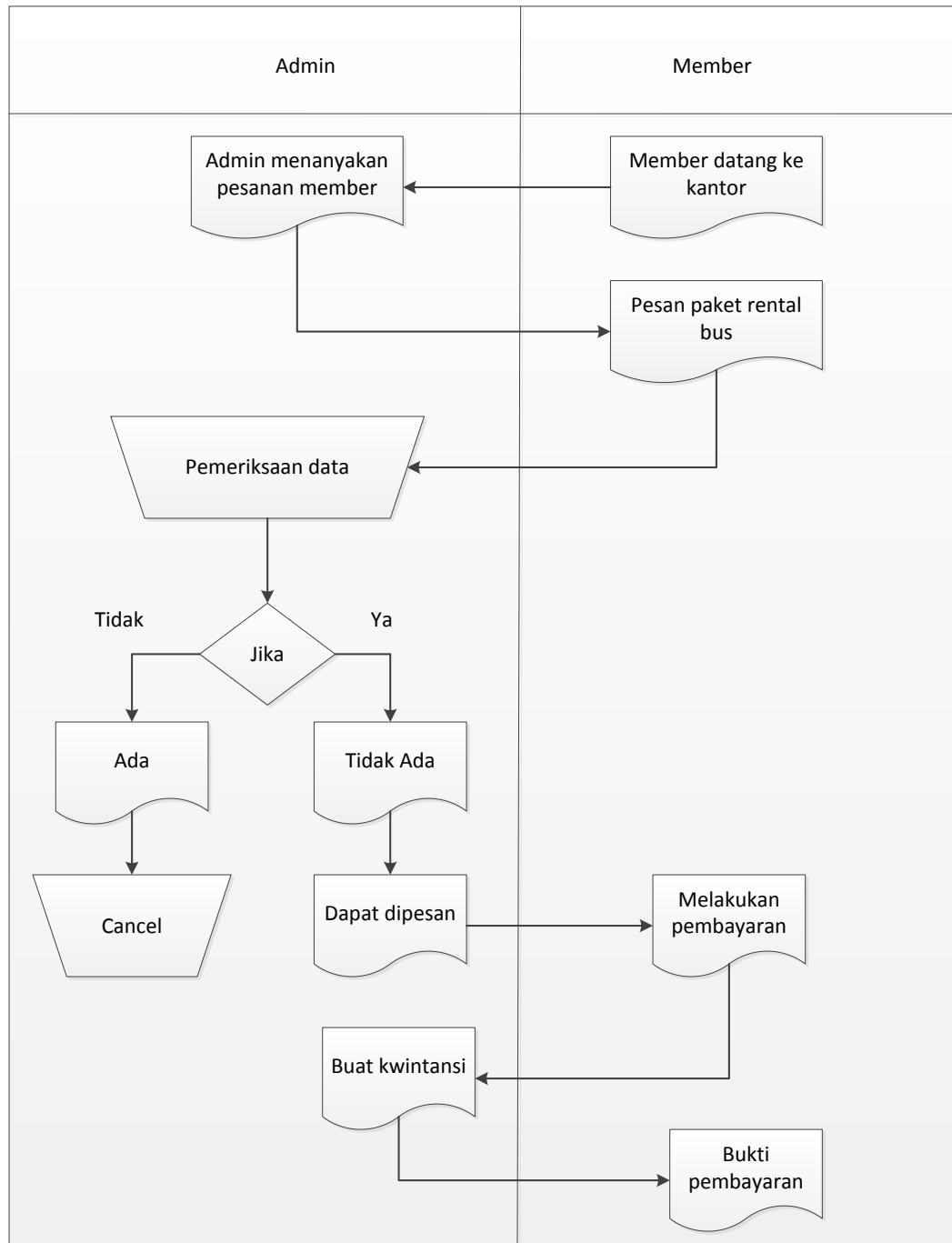
Penulis telah mewawancarai pegawai dan manager yang ada di *Panorama Destination*. Wawancara dilakukan secara langsung, dan hal tersebut didapatlah data atau keterangan yang diperlukan penulis sebagai bahan untuk menganalisa sistem yang berjalan pada *Panorama Destination*.

3.3 Analisis Sistem Sedang Berjalan

Analisa yang sedang berjalan dalam sistem pemesanan rental bus merupakan prosedur dimana akan melibatkan seorang admin pegawai dan member atau konsumen. Berdasarkan penelitian yang telah dilakukan di *Panorama Destination*,

maka dapat digambarkan prosedur pemesanan rental bus dengan tahapan-tahapan sebagai berikut:

- a. Member atau konsumen dapat ke kantor Panaroma Destinantion untuk melakukan pemesanan rental bus.
- b. Admin menanyakan pesanan paket apa yang di inginkan oleh member atau konsumen.
- c. Member melakukan pemilihan paket sesuai dengan keinginan.
- d. Admin memalukan pemeriksaan data apakah ada yang melakukan pemesanan rental di tanggal yang sama atau tidak.
- e. Jika tidak ada yang melakukan pemesanan pada tanggal yang sudah di tentukan maka paket tersebut dapat dipesan. Jika ada yang melalukan pemesanan di tanggal yang sama maka admin melalukan cancel pemesanan.
- f. Member melalukan pembayaran.
- g. Admin membuatkan kwintasi untuk bukti pembayaran.
- h. Bukti pembayaran diterima oleh member sebagai barang bukti pemesanan rental bus di Panorama Destinantion.



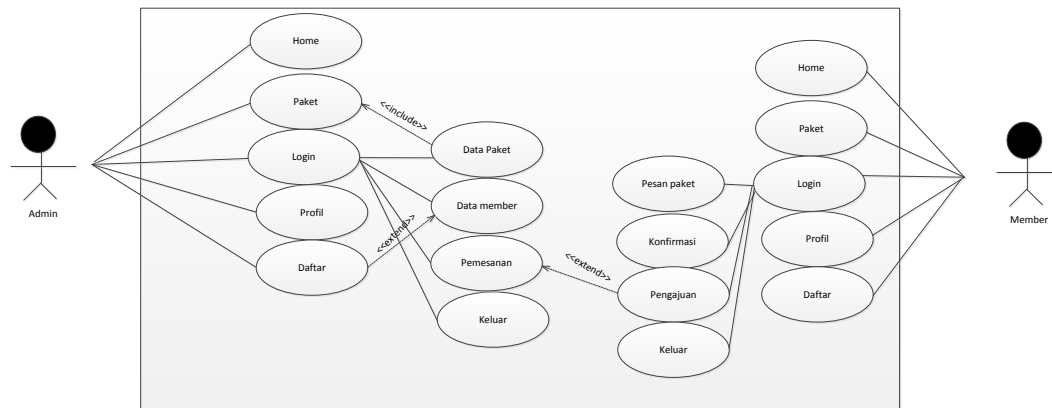
Gambar 3.2. Flowmap Sistem yang Berjalan

3.4 Rancangan Penelitian

Berikut ini adalah rancangan penelitian dari aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio. Rancangan penelitian ini menggunakan UML seperti *use case*, *activity diagram*, *sequence diagram*, dan *class diagram*. Berikut ini adalah rancanganya :

3.4.1 Use Case Diagram

Pada gambaran *use case diagram* yang dibawah ini menggambarkan alur program dari aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio. Berikut ini adalah gambarnya:



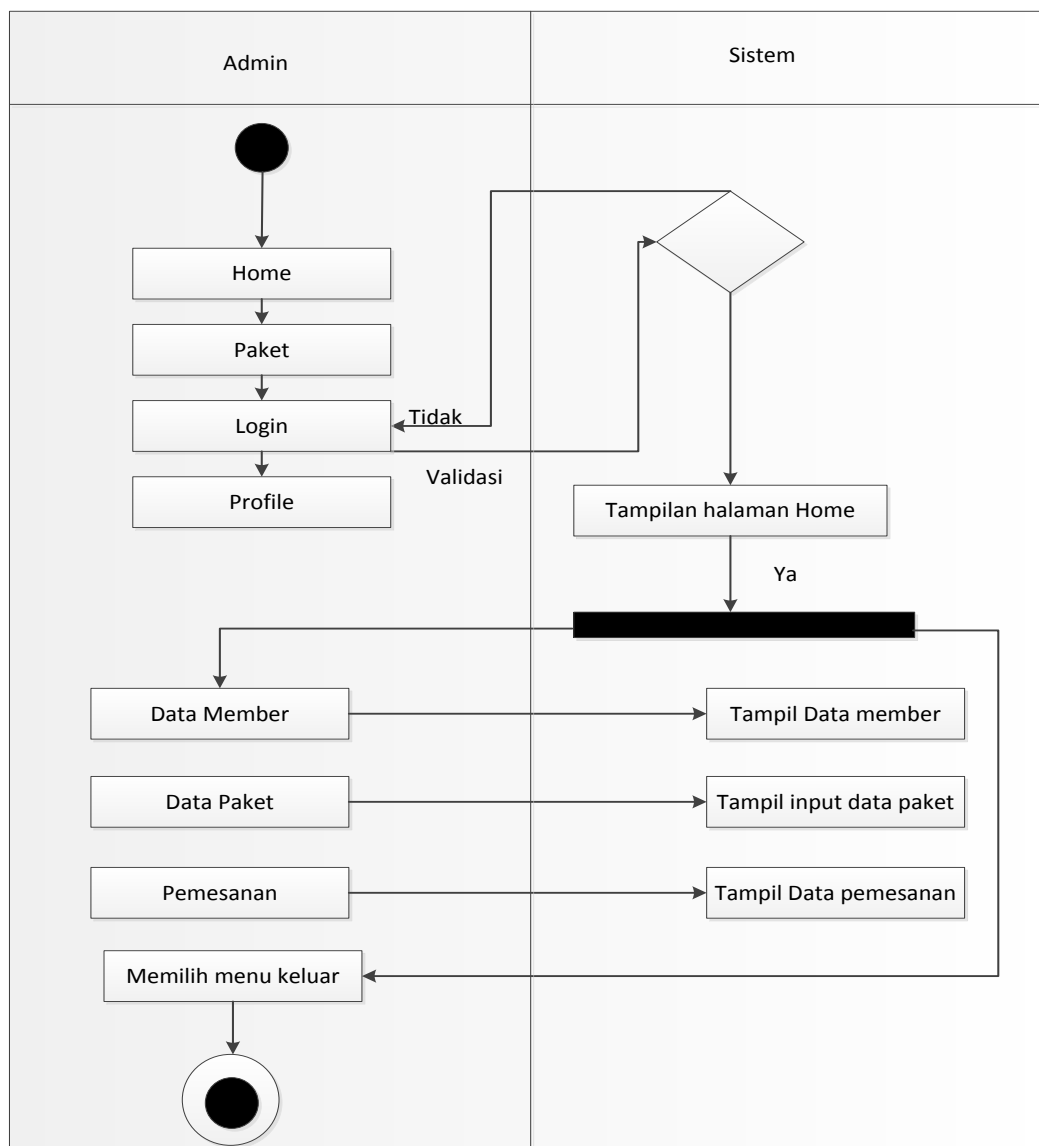
Gambar 3.3. *Use Case Diagram*

3.4.2 Activity Diagram

Pada gambaran *activity diagram* dibawah ini menggambarkan secara detail program dari setiap user yang menggunakan aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio yaitu admin, dan member. Berikut ini adalah gambarnya :

1. Activity Diagram Admin

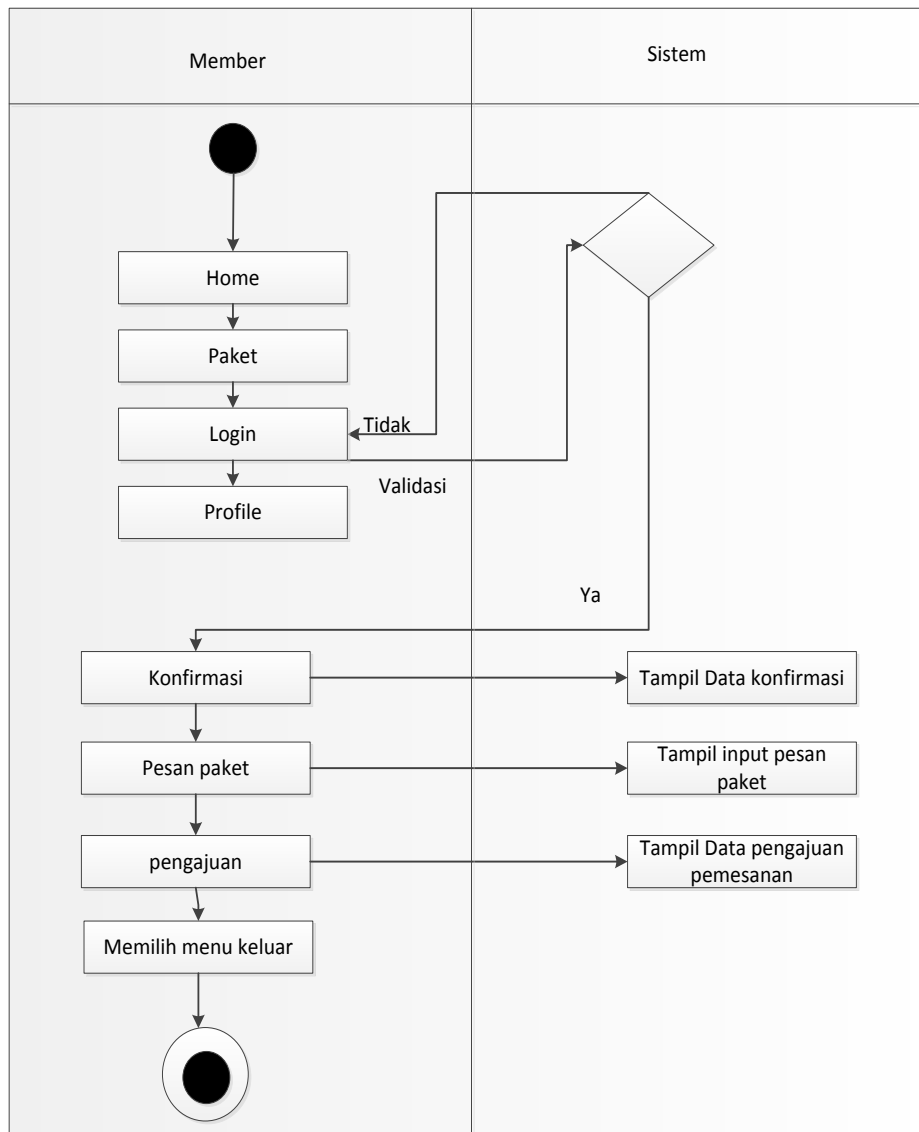
Berikut ini adalah gambaran dari *activity diagram* pada admin untuk aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio :



Gambar 3.4. Activity Diagram Admin

2. Activity Diagram Member

Berikut ini adalah gambaran dari *activity diagram* pada member untuk aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio:



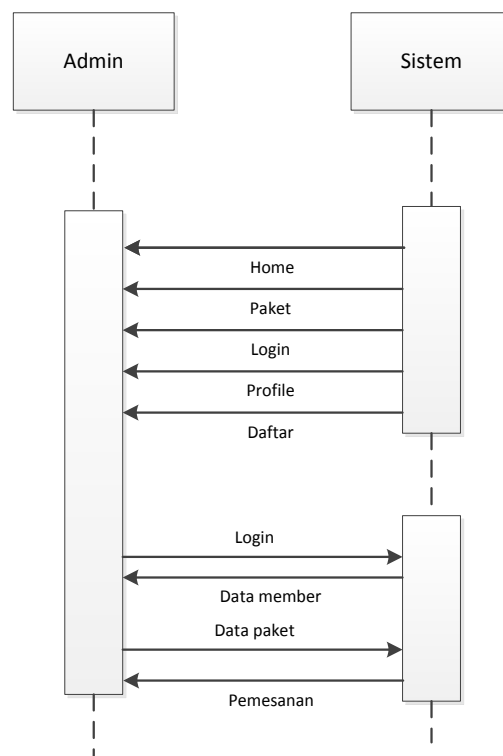
Gambar 3.5. Activity Diagram Member

3.4.3 Sequence Diagram

Pada gambaran *sequence diagram* dibawah ini menggambarkan secara detail program dari setiap user yang menggunakan aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio yaitu admin, dan member. Berikut ini adalah gambarnya :

1. Sequence Diagram Admin

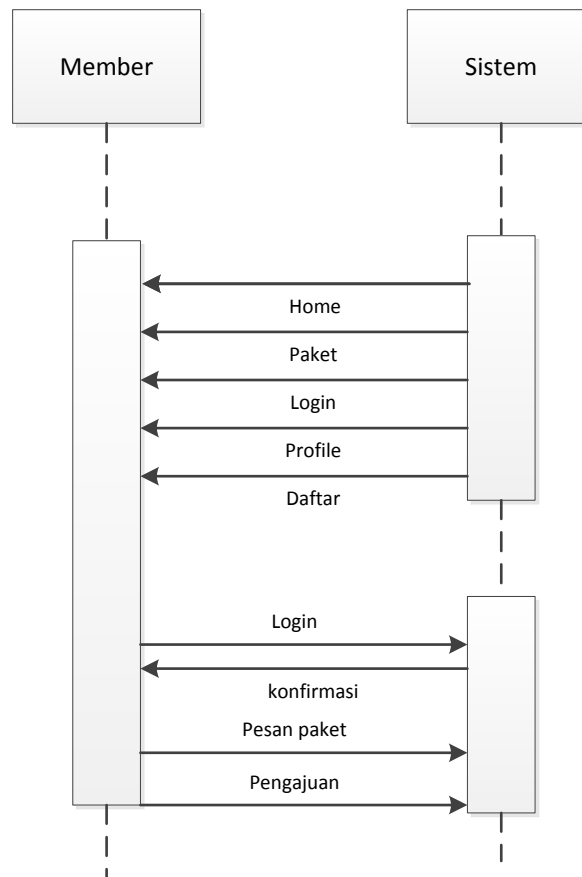
Berikut ini adalah gambaran dari *sequence diagram* pada admin untuk aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio:



Gambar 3.6. *Sequence Diagram Admin*

2. Activity Diagram Member

Berikut ini adalah gambaran dari *sequence diagram* pada member untuk aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio:

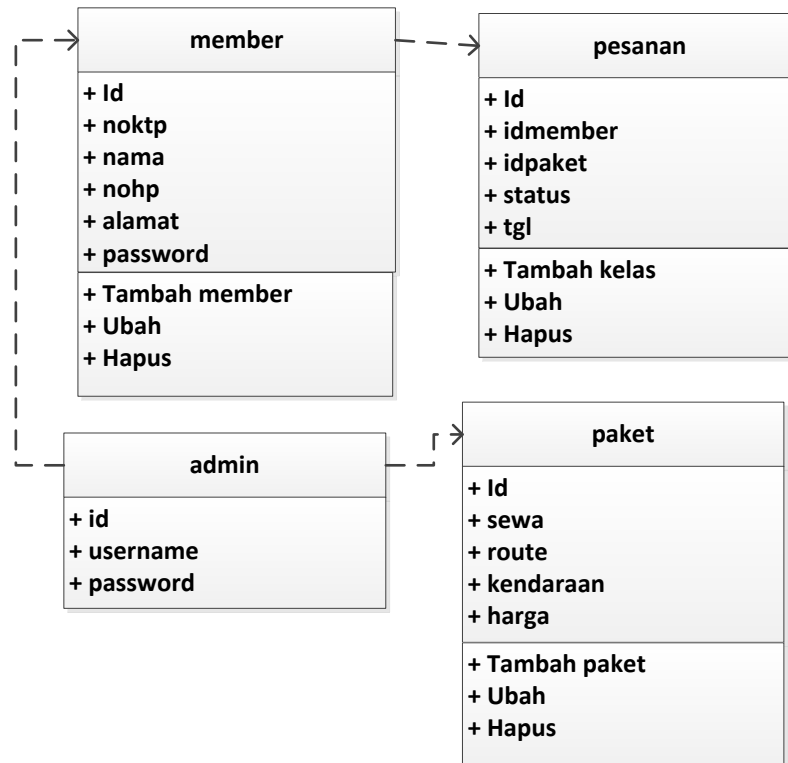


Gambar 3.7. *Sequence Diagram* Member

3.4.4 Class Diagram

Pada gambaran *class diagram* yang dibawah ini menggambarkan alur program dari aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan

bahasa pemrograman android studio. Berikut ini adalah gambarnya:



Gambar 3.8. *Class Diagram*

3.5 Rancangan Database

Database akan dinamakan “**id2746825**”, dengan jumlah tabel adalah 4 (empat), berikut struktur dari tabel-tabel tersebut pada database MySQL.

1. Tabel Admin

Berikut ini adalah tabel dari admin berisi data admin seperti *username* dan *password* admin.

Tabel 3.1. Tabel Admin

Nama Field	Tipe Data	Size	Keterangan
Id	Interger	11	<i>Primary key</i>
username	Varchar	50	Berisi data username admin
password	Varchar	25	Berisi data password admin

2. Tabel Member

Berikut ini adalah tabel dari guru berisi data atribut seperti nama member, no KTP, alamat, no HP.

Tabel 3.2. Tabel Member

Nama Field	Tipe Data	Size	Keterangan
Id	Interger	11	<i>Primary key</i>
Noktp	Varchar	100	Berisi data nomor identitas member
Nama	Varchar	100	Berisi data nama member
Nohp	Varchar	50	Berisi data nomor handphone
Alamat	Text		Berisi data alamat member
Password	Varchar	100	Berisi data password member

3. Tabel Paket

Berikut ini adalah tabel dari paket berisi data paket seperti sewa, router, kendaraan, dan harga.

Tabel 3.3. Tabel Paket

Nama Field	Tipe Data	Size	Keterangan
Id	Interger	11	<i>primary key</i>
Sewa	Varchar	100	Berisi data sewa member
Router	Varchar	100	Berisi data router
Kendaraan	Varchar	100	Berisi data jenis kendaraan
Harga	Varchar	100	Berisi data harga paket

4. Tabel Pesanan

Berikut ini adalah tabel dari pesanan berisi data pesanan seperti id member, id paket, status dan tanggal.

Tabel 3.4. Tabel Pesanan

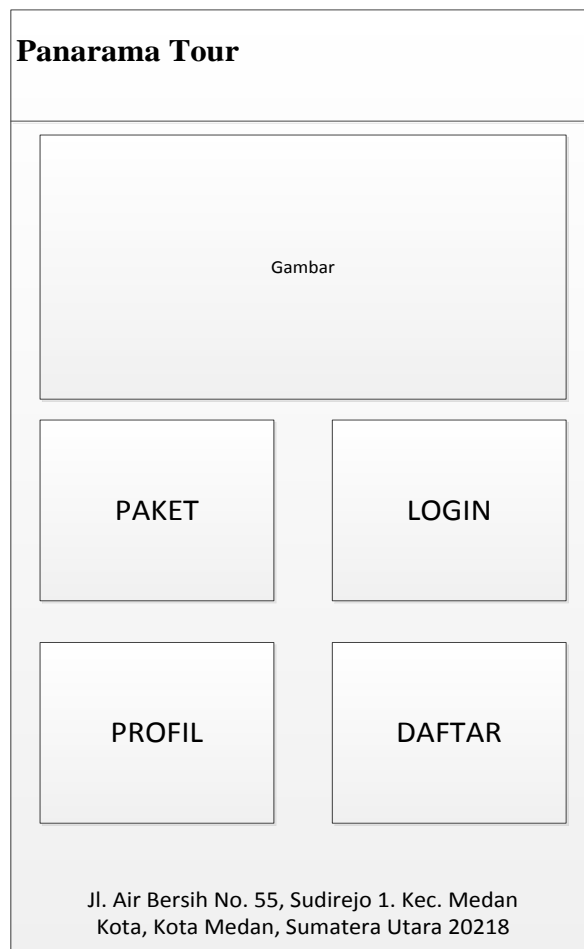
Nama Field	Tipe Data	Size	Keterangan
Id	Interger	11	<i>Primary key</i>
Idmember	Varchar	100	Berisi data id member
Idpaket	Varchar	100	Berisi data id pakte
Status	Varchar	100	Berisi status pesanan
Tgl	Varchar	100	Berisi data tanggal pesanan

3.6 Rancangan *Interface*

3.6.1 Rancangan Halaman *Home*

Rancangan halaman *home* pada awal tampilan dari aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio

dapat digunakan oleh admin dan member. Pada halaman *home* ini terdapat menu-menu yang dapat dipilih oleh admin atau member dan jika dipilih salah satu dari menu tersebut maka akan masuk ke rancangan halaman selanjutnya. Terdapat menu-menu seperti paket, login, profil, dan daftar. Dari setiap menu-menu mempunyai fungsinya masing-masing yang dapat memberikan kemudahan pada admin dan member. Berikut ini adalah rancangan dari halaman *home* :



Gambar 3.9. Rancangan Halaman *Home*

3.6.2 Rancangan Halaman Paket

Rancangan halaman paket pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member untuk melihat-lihat data paket-paket yang tersedia di *Panorama Destination* untuk pemesanan rental bus. Berikut ini adalah rancangan dari halaman paket:

Paket
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Transfer Carter</div> <p>Router : Kendaraan : Harga :</p>
<p>Router : Kendaraan : Harga :</p>
<p>Router : Kendaraan : Harga :</p>
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Sumatera Barat</div> <p>Router : Kendaraan : Harga :</p>
<p>Router : Kendaraan : Harga :</p>

Gambar 3.10. Rancangan Halaman Paket

3.6.3 Rancangan Halaman *Login*

Rancangan halaman *login* pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member. Sebelum masuk ke rancangan master data dari admin, terlebih dahulu harus *login* dengan memasukkan username dan password. Sedangkan untuk member sebelum melakukan pemesanan rental bus harus *login* dahulu juga. Berikut ini adalah rancangan dari halaman *login*:

The image shows a wireframe for a login page. At the top, there is a header section with the word "Login" in bold. Below the header is a large rectangular area containing a smaller box labeled "Gambar", which is a placeholder for a logo or image. Underneath the "Gambar" box are three input fields. The first is labeled "No KTP", the second is labeled "Password", and the third is a button labeled "LOGIN".

Gambar 3.11. Rancangan Halaman *Login*

3.6.4 Rancangan Halaman Daftar

Rancangan halaman daftar pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member. Jika member belum melakukan pendaftaran maka wajib untuk daftar ke dalam aplikasi sistem pemesanan rental bus. Berikut ini adalah rancangan dari halaman daftar:

The image shows a registration form titled "Daftar". It contains five input fields and a submit button. The fields are labeled "No KTP", "Nama", "No Hp", "Alamat", and "Password". The submit button is labeled "DAFTAR".

Daftar	
No KTP	<input type="text"/>
Nama	<input type="text"/>
No Hp	<input type="text"/>
Alamat	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="DAFTAR"/>	

Gambar 3.12. Rancangan Halaman Daftar

3.6.5 Rancangan Halaman *Home Admin*

Rancangan halaman *home admin* pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Setelah admin berhasil *login* kedalam aplikasi maka akan masuk ke halaman rancangan *home admin*. Pada halaman ini terdapat menu-menu seperti data member, data paket, dan pesanan. Berikut ini adalah rancangan dari halaman *home admin*:



Gambar 3.13. Rancangan Halaman *Home Admin*

3.6.6 Rancangan Halaman Data Member

Rancangan halaman data member pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman data member admin dapat melihat data-data member dan dapat melakukan pengeditan data serta penghapusan data. Berikut ini adalah rancangan dari halaman data member :

Data Member	
No KTP :	
Nama :	
No HP :	
Alamat :	
Password :	
<input type="button" value="UBAH"/>	<input type="button" value="HAPUS"/>
No KTP :	
Nama :	
No HP :	
Alamat :	
Password :	
<input type="button" value="UBAH"/>	<input type="button" value="HAPUS"/>

Gambar 3.14. Rancangan Halaman Data Member

3.6.7 Rancangan Halaman Data Paket

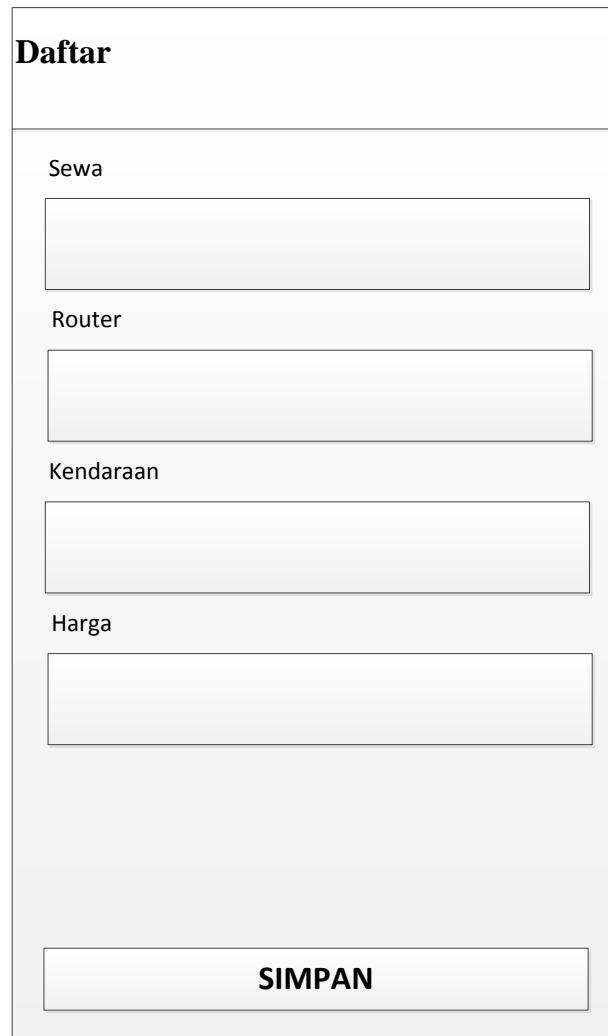
Rancangan halaman data paket pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman data member admin dapat melihat data-data paket dan dapat melakukan pengeditan data serta penghapusan data. Berikut ini adalah rancangan dari halaman data paket :

Data Paket	
Transfer Carter	
Router :	
Kendaraan :	
Harga :	
UBAH	HAPUS
Router :	
Kendaraan :	
Harga :	
UBAH	HAPUS
Router :	
Kendaraan :	
Harga :	
UBAH	HAPUS
Tambah Paket	

Gambar 3.15. Rancangan Halaman Data Paket

3.6.8 Rancangan Halaman *Input* Paket

Rancangan halaman *input* paket pada aplikasi sistem pemesanan rental bus di Panorama Destination menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman *input* paket admin dapat memasukkan data paket-paket yang ada di Panorama Destiantion. Berikut ini adalah rancangan dari halaman *input* paket:



The image shows a mobile application form titled "Daftar" (Register) for entering package information. The form is contained within a light gray border and has a white background. At the top left, the word "Daftar" is written in bold black text. Below the title, there are four input fields, each with a label above it: "Sewa" (Rent), "Router", "Kendaraan" (Vehicle), and "Harga" (Price). Each label is followed by a rectangular text input box. At the bottom of the form, there is a large, wide button with the text "SIMPAN" (Save) in bold black capital letters.

Gambar 3.16. Rancangan Halaman *Input* Paket

3.6.9 Rancangan Halaman Pesanan

Rancangan halaman pesanan pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman pesanan admin dapat melihat data pesanan dari member yang sudah melakuakn *booking* rental bus, jika sudah melakukan pembayaran maka admin dapat langsung menekan tombol sudah selesai. Berikut ini adalah rancangan dari halaman pesanan:

Pesanan Member	
Status :	
Tanggal Pesan :	
No KTP :	
Nama :	
Sewa :	
Router :	
Kendaraan :	
Harga :	
Tgl Berangkat :	
Tgl Bayar Terakhir:	
SUDAH SELESAI	
Status :	
Tanggal Pesan :	
No KTP :	
Nama :	
Sewa :	
Router :	
Kendaraan :	
Harga :	
Tgl Berangkat :	
Tgl Bayar Terakhir:	
SUDAH SELESAI	

Gambar 3.17. Rancangan Halaman Pesanan

3.6.10 Rancangan Halaman Konfirmasi

Rancangan halaman konfirmasi pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Jika sudah melakukan login kedalam aplikasi maka akan masuk ke halaman konfirmasi, disini member dapat melakukan konfirmasi pembayaran jika sudah melakukan pemesanan rental bus. Berikut ini adalah rancangan dari halaman konfirmasi:

The image shows a wireframe of a confirmation page for a bus rental application. It is divided into several sections:

- Halaman Member:** A header section containing the text "Halaman Member".
- User Information:** A section with two labels: "No KTP :" and "Nama :", each followed by a colon and a space, indicating where user details are displayed.
- KONFIRMASI:** A rectangular button with the text "KONFIRMASI" centered inside.
- Paket Tour:** A section header with a circular refresh icon to its right.
- Payment Details:** Three stacked rectangular boxes containing the text "Transfer Carter", "Sumatera Barat", and "Router Di Sumatera Utara" respectively.
- Kontak Admin:** A text label at the bottom of the page.

Gambar 3.18. Rancangan Halaman Konfirmasi

3.6.11 Rancangan Halaman Pesan Paket

Rancangan halaman pesan paket pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Pada halaman ini member dapat melakukan pemesanan paket dan router perjalanan serta dapat melihat cek harga. Berikut ini adalah rancangan dari halaman pesan paket:

Pesan Paket Tour	
Paket	Transfer Carter
Router	<input type="text"/>
Kendaraan	<input type="text"/>
Tanggal Berangkat	<input type="text"/>
<input type="button" value="CEK HARGA"/>	
Harga	Rp
<input type="button" value="PESAN"/>	

Gambar 3.19. Rancangan Halaman Pesan Paket

3.6.12 Rancangan Halaman Pengajuan

Rancangan halaman pengajuan pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Pada halaman pengajuan ini member dapat melakukan konfirmasi untuk kirim bukti pembayaran atau transfer pembayaran dan. Mengirimkan buktinya melalui WA. Berikut ini adalah rancangan dari halaman data pengajuan :

Pesan Paket Tor

Status Pesanan Pengajuan

Paket Sewa :
 Kendaraan :
 Router :
 Harga :
 Tanggal Berangkat :
 Tanggal Terakhir Konfirmasi :

Cara konfirmasi
 Transfer ke rekening 121323243 Atas Nama Bella. Kemudian kirim bukti transfer ke kontak WA di bawah ini sebelum 2 hari tanggal pemesanan.

Kontak WA

BATAL PESAN

Gambar 3.20. Rancangan Halaman Pengajuan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Aplikasi

Pada tahapan ini penulis akan menjelaskan bagaimana cara aplikasi ini bekerja. Pengguna dari aplikasi ini adalah admin dari Panorama Destiantion dan member dari Panorama Destiantion serta menginputkan pemesanan rental bus, dan seorang admin yang bertugas untuk mengawasi sistem aplikasi, menjaga aktivitas dalam aplikasi, melakukan upgrade aplikasi jika dibutuhkan, memajemen pengguna aplikasi, baik menambah atau menghapus, dan aktivitas lainnya yang mendukung penggunaan aplikasi sistem pemesanan rental bus. Dengan adanya aplikasi ini kegiatan aplikasi sistem pemesanan rental bus dapat dilakukan dengan lebih efektif.

4.2 Tampilan Program

4.2.1 Tampilan Halaman *Home*

Tampilan halaman *home* pada awal tampilan dari aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member. Pada halaman *home* ini terdapat menu-menu yang dapat dipilih oleh admin atau member dan jika dipilih salah satu dari menu tersebut maka akan masuk ke Tampilanhalaman selanjutnya. Terdapat menu-menu seperti paket, login, profil, dan daftar. Dari setiap menu-menu mempunyai

fungsinya masing-masing yang dapat memberikan kemudahan pada admin dan member. Berikut ini adalah tampilan dari halaman *home* :



Gambar 4.1. Tampilan Halaman *Home*

4.2.2 Tampilan Halaman Paket

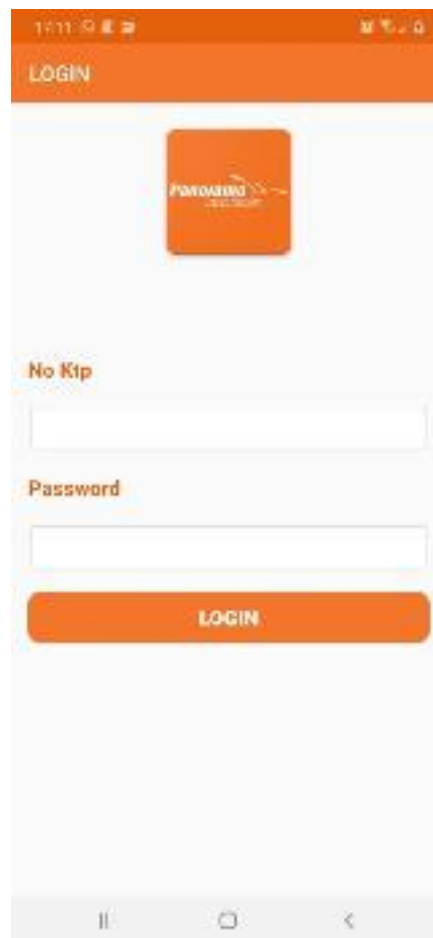
Tampilan halaman paket pada aplikasi sistem pemesanan rental bus di Panorama Destination menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member untuk melihat-lihat data paket-paket yang tersedia di Panorama Destination untuk pemesanan rental bus. Berikut ini adalah tampilan dari halaman paket:



Gambar 4.2. Tampilan Halaman Paket

4.2.3 Tampilan Halaman *Login*


Tampilan halaman *login* pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member. Sebelum masuk ketampilan master data dari admin, terlebih dahulu harus *login* dengan memasukkan username dan password. Sedangkan untuk member sebelum melakukan pemesanan rental bus harus *login* dahulu juga. Berikut ini adalah tampilan dari halaman *login*:



Gambar 4.3. Tampilan Halaman *Login*

4.2.4 Tampilan Halaman Daftar

Tampilan halaman daftar pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin dan member. Jika member belum melakukan pendaftaran maka wajib untuk daftar ke dalam aplikasi sistem pemesanan rental bus. Berikut ini adalah tampilan dari halaman daftar:



The image shows a mobile application registration screen. At the top, there is an orange header bar with the word "Daftar" in white text. Below the header, there are five white input fields with grey borders, each preceded by a label: "No KTP", "Nama", "No HP", "Alamat", and "Password". At the bottom of the form area, there is a wide, grey button with the word "DAFTAR" in black, uppercase letters. The entire form is set against a light grey background. At the very bottom of the screen, the standard Android navigation bar is visible, showing three icons: a square, a circle, and a triangle pointing left.

Gambar 4.4. Tampilan Halaman Daftar

4.2.5 Tampilan Halaman *Home Admin*

Tampilan halaman *home* admin pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Setelah admin berhasil *login* kedalam aplikasi maka akan masuk ke halaman Tampilan*home* admin. Pada halaman ini terdapat menu-menu seperti data member, data paket, dan pesanan. Berikut ini adalah tampilan dari halaman *home* admin:



Gambar 4.5. Tampilan Halaman *Home Admin*

4.2.6 Tampilan Halaman Data Member

Tampilan halaman data member pada aplikasi sistem pemesanan rental bus di *Panorama Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman data member admin dapat melihat data-data member dan dapat melakukan pengeditan data serta penghapusan data. Berikut ini adalah tampilan dari halaman data member :



Gambar 4.6. Tampilan Halaman Data Member

4.2.7 Tampilan Halaman Data Paket

Tampilan halaman data paket pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman data member admin dapat melihat data-data paket dan dapat melakukan pengeditan data serta penghapusan data. Berikut ini adalah tampilan dari halaman data paket :



Gambar 4.7. Tampilan Halaman Data Paket

4.2.8 Tampilan Halaman *Input* Paket

Tampilan halaman *input* paket pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman *input* paket admin dapat memasukkan data paket-paket yang ada di Panorama *Destiantion*. Berikut ini adalah tampilan dari halaman *input* paket:



The screenshot shows a mobile application interface for registering a bus rental package. The page has an orange header with the title "Daftar". Below the header, there are four input fields, each with a label above it: "Sewa", "Route", "Kendaraan", and "Harga". At the bottom of the form, there is a large orange button labeled "SIMPAN". The Android navigation bar is visible at the very bottom of the screen.

Gambar 4.8. TampilanHalaman *Input* Paket

4.2.9 Tampilan Halaman Pesanan

Tampilan halaman pesanan pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh admin. Pada halaman pesanan admin dapat melihat data pesanan dari member yang sudah melakuakn *booking* rental bus, jika sudah melakukan pembayaran maka admin dapat langsung menekan tombol sudah selesai. Berikut ini adalah tampilan dari halaman pesanan:



Gambar 4.9. Tampilan Halaman Pesanan

4.2.10 Tampilan Halaman Konfirmasi

Tampilan halaman konfirmasi pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Jika sudah melakukan login kedalam aplikasi maka akan masuk ke halaman konfirmasi, disini member dapat melakukan konfirmasi pembayaran jika sudah melakukan pemesanan rental bus. Berikut ini adalah tampilan dari halaman konfirmasi:



Gambar 4.10. Tampilan Halaman Konfirmasi

4.2.11 Tampilan Halaman Pesan Paket

Tampilan halaman *home* admin pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Pada halaman ini member dapat melakukan pemesanan paket dan router perjalanan serta dapat melihat cek harga. Berikut ini adalah tampilan dari halaman pesan paket:



Pesan Paket Tour

Paket Sumatera Barat

Route

Kendaraan

Tanggal Berangkat 2020-5-12

CEK HARGA

Harga

PESAN

Gambar 4.11. Tampilan Halaman Pesan Paket

4.2.12 Tampilan Halaman Pengajuan

Tampilan halaman pengajuan pada aplikasi sistem pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio dapat digunakan oleh member. Pada halaman pengajuan ini member dapat melakukan konfirmasi untuk kirim bukti pembayaran atau transfer pembayaran dan. Mengirimkan buktinya melalui WA. Berikut ini adalah tampilan dari halaman data pengajuan :



Gambar 4.12. Tampilan Halaman Pengajuan

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan mengenai menyelesaikan aplikasi sistem informasi pemesanan rental bus di Panorama *Destination* menggunakan bahasa pemrograman android studio, maka pada bagian penutup dari penelitian ini, penulis menarik kesimpulan sekaligus memberikan saran sebagai berikut.

1. Dengan adanya aplikasi sistem informasi pemesanan rental bus dapat memberikan kemudahan kepada admin sebagai media informasi kepada para konsumen atau member.
2. Aplikasi sistem informasi pemesanan rental bus dapat memberikan kemudahan kepada member atau konsumen dapat melihat informasi mengenai paket-paket rental bus yang ada di Panorama *Destination* dan dapat memberikan kemudahan dalam pemesanan rental bus.
3. Aplikasi sistem informasi pemesanan rental bus dapat memberikan kemudahan kepada pegawai atau admin dalam melakukan pendataan mengenai pemesanan rental bus.

5.2 Saran

Saran yang dapat diberikan untuk perbaikan dan pengembangan selanjutnya

menyelesaikan aplikasi sistem informasi pemesanan rental bus di Panorama

Destination menggunakan bahasa pemrograman android studio ini adalah :

1. Untuk pengembangan selanjutnya sebaiknya setiap dapat memberikan keamanan sehingga data-data member dapat terjamin keamanannya.
2. Untuk pengembangan aplikasi sistem informasi pemesanan rental bus di Panorama *Destination* dapat membuat program baru yang berbasis IOS sehingga bagi pengguna iPhone dapat menggunakannya juga.
3. Aplikasi sistem informasi pemesanan rental bus dapat menambahkan menu baru seperti pemesanan kamar hotel.

DAFTAR PUSTAKA

- Abdul Rohman Amat Ramat, Alvino Octaviano. (2016). Aplikasi Pemesanan Tiket Bus Berbasis Web (Studi Kasus Pada PO. Harapan Jaya). Vol. 1, No. 1. ISSN: 2541-1004.
- Anisa Rachmawati, Arief Laila Nugraha, Muhammad Awaludin. (2017). Desain Aplikasi Mobile Infomrasi Pemetaan Jalur Batik Solo Trans Berbasis Andorid Menggunakan Location Based Service. Vol. 6.No. 2. ISSN: 2337-845X.
- Arif Muhammad. (2016). Rancangan Teknik Industri. Yogyakarta: Deepublish.
- Adi Widarma, Sri Rahayu. (2017). Perancangan Aplikasi Gaji Karyawan Pada PT. PP London Sumatra Indonesia Tbk. Gunung Malayu Estate – Kabupaten Asahan. No. 1. No. 2. ISSN: 2615-2738.
- Efmi Maiyana. (2018). Pemanfaatan Andorid Dalam Perancangan Aplikasi Kumpulan Doa. Vol. 4. No. 2. ISSN: 2459-9549.
- Harni Kusniyati, Nicky Saputra Pangondian Sitanggang. (2016). Aplikasi Edukasi Budaya Toba Samosir Berbasis Android. Vol. 9. No. 1. ISSN: 1979-9160.
- O.K Saddam Hussein, Refni Wahyuni, Yuda Irawan, Harun Mukhtar. (2018). Sistem Informasi Deteksi Kehadiran Dan Media Penyimpanan Pengumuman Dosen Dengan Menggunakan Teknik Pengenalan QR CODE. Vo. 3. No. 2. ISSN: 2477-2062.
- Robi Yanto. (2016). Manajemen Basis Data Menggunakan MySql.Yogyakarta : Penerbit Deepublish.
- Ranu Triwibowo, Novita Br. Ginting, Fety Fatimah. (2019). Sistem Informasi Penyewaan Rental Mobil Berbasis Web Pada CV. Adelia Transport. ISSN: 978-6020-20-7.
- T.Henny Febriana Harumy, Indri Sulistriningsih, (2016). Sistem Penunjang Keputusan Penentuan Jabatan Manager Menggunakan Metode MFEP Pada CV. Sapo Durin. 6-7 Februari ISSN: 2302-3805.

- Widi Nugrahaningsih, Mira Erlinawati. (2017). Implementasi Undang-Undang Nomor 8 Tahun 1999 Tentang Perlindungan Konsumen Terhadap Bisnis Online. Vol. 11. No. 1. ISSN: 1693-0819.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Bahri, S. (2018). *Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS*. Penerbit Andi (Anggota Ikapi). Percetakan Andi Offset. Yogyakarta.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encrypted Text in End of File Hiding Steganography Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 214-219.
- Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). *Jurnal Teknik dan Informatika*, 6(1), 42-45.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.

- Muttaqin, Muhammad. "ANALISA PEMANFAATAN SISTEM INFORMASI E-OFFICE PADA UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN DENGAN MENGGUNAKAN METODE UTAUT." *Jurnal Teknik dan Informatika* 5.1 (2018): 40-43.
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *JURIKOM (Jurnal Riset Komputer)*, 5(2), 135-139.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.

Lampiran - lampiran

1. Login.java

```
public class Login extends AppCompatActivity {

    ActionBar bar ;
    Konek konek ;
    EditText edt_noktp , edt_pass ;
    Button btn_login ;
    Loading loading ;
    SQLite sqLite;
    SQLiteDatabase db ;
    Cursor c ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        bar = getSupportActionBar();
        bar.setTitle("LOGIN");

        konek = new Konek(this);

        edt_noktp = (EditText) findViewById(R.id.edt_noktp);
        edt_pass = (EditText) findViewById(R.id.edt_pass);
        btn_login = (Button) findViewById(R.id.btn_login);

        sqLite = new SQLite(this);
        db = sqLite.getReadableDatabase();
        db = sqLite.getWritableDatabase();
        sqLite.createTable(db);

        btn_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (edt_noktp.getText().toString().equals("") &&
                    edt_pass.getText().toString().equals("")){
                    Toast.makeText(getApplicationContext(),"Kolom tidak Boleh kosong
                    ",Toast.LENGTH_LONG).show();
```

```

    }else {
        if (edt_noktp.getText().toString().equals("admin")){
            login("admin","username");
        }else {
            login("member","noktp");
        }
    }
}
});
}

```

```

private void login(String AKSI,String TIPE){

```

```

    loading = new Loading(this);
    loading.panggil();

```

```

konek.get("login.php?aksi="+AKSI+"&"+TIPE+"="+edt_noktp.getText().toString
()
    +"&password="+edt_pass.getText().toString(),new
JsonHttpReponseHandler(){

```

```

    @Override
    public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
        super.onFailure(statusCode, headers, throwable, errorResponse);

```

```

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        //Log.e("Errro pengiriman ", errorResponse.toString());
    }

```

```

    @Override
    public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
        super.onFailure(statusCode, headers, responseString, throwable);

```

```

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        //Log.e("Errro pengiriman ", responseString.toString());

```

```

    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
        super.onSuccess(statusCode, headers, response);

        try{

            final int sukses    = Integer.valueOf(response.getString("success"));

            if (sukses == 1){

                JSONArray jsonArray = response.getJSONArray("member");

                for (int i = 0; i < jsonArray.length(); i++) {

                    JSONObject data = jsonArray.getJSONObject(i);

                    String ID = data.getString("id");
                    String NOKTP = data.getString("noktp");
                    String NAMA = data.getString("nama");
                    String NOHP = data.getString("nohp");
                    String ALAMAT = data.getString("alamat");
                    String PASSWORD = data.getString("password");

                    db.execSQL("INSERT INTO member
(id,noktp,nama,nohp,alamat,password) VALUES ("
+ID+"", ""+NOKTP+"", ""+NAMA+"", ""+NOHP+"", ""+ALAMAT+"", ""+PASSWORD
+"")");

                }

                finish();
                Intent i = new Intent(Login.this, HalamanMember.class);
                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(i);

            }else if (sukses == 3){

                finish();

```

```

        Intent ip = new Intent(Login.this, HalamanAdmin.class);
        ip.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(ip);

    }else {
        Toast.makeText(getApplicationContext(),"Gagal MASUK
",Toast.LENGTH_SHORT).show();
    }
    }catch (JSONException e){

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(),"Internet tidak tersambung
... ",Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

    }
    });
}
public boolean onKeyDown(int keyCode , KeyEvent event){

    if ((keyCode == KeyEvent.KEYCODE_BACK))
    {
        finish();
        Intent i = new Intent(Login.this, HalamanUtama.class);
        startActivity(i);
        return true; //I have tried here true also
    }
    return false;

}
}
}

```

2. Halamanutama.java

```

public class HalamanUtama extends AppCompatActivity {

    ViewPager viewPager;
    List<Integer> gambar;
    RelativeLayout rl_paket, rl_login, rl_daftar , rl_profil;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_halaman_utama);

    viewPager=(ViewPager)findViewById(R.id.viewPager);

    gambar      = new ArrayList<>();
    gambar.add(R.drawable.panorama);
    gambar.add(R.drawable.panoramaa);
    gambar.add(R.drawable.panoramaaa);

    viewPager.setAdapter(new SliderAdapter(this,gambar));

    Timer timer = new Timer();
    timer.scheduleAtFixedRate(new SliderTimer(), 4000, 5000);

    rl_paket    = (RelativeLayout) findViewById(R.id.rl_paket);
    rl_login    = (RelativeLayout) findViewById(R.id.rl_login);
    rl_daftar   = (RelativeLayout) findViewById(R.id.rl_daftar);
    rl_profil   = (RelativeLayout) findViewById(R.id.rl_profil);

    rl_daftar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(HalamanUtama.this, Daftar.class);
            i.putExtra("key","1");
            startActivity(i);
        }
    });

    rl_login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(HalamanUtama.this, Login.class);
            startActivity(i);
            finish();
        }
    });

    rl_paket.setOnClickListener(new View.OnClickListener() {
        @Override

```

```

        public void onClick(View view) {
            Intent i = new Intent(HalamanUtama.this, Paket.class);
            startActivity(i);
        }
    });

    rl_profil.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(HalamanUtama.this, Profil.class);
            startActivity(i);
        }
    });
}

private class SliderTimer extends TimerTask {

    @Override
    public void run() {
        HalamanUtama.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                if (viewPager.getCurrentItem() < gambar.size() - 1) {
                    viewPager.setCurrentItem(viewPager.getCurrentItem() + 1);
                } else {
                    viewPager.setCurrentItem(0);
                }
            }
        });
    }
}
}
}

```

3. Daftar.java

```

private void daftar(){

    loading = new Loading(this);
    loading.panggil();

    rp = new RequestParams();
}

```



```

rp.add("noktp", edt_noktp.getText().toString());
rp.add("nama", edt_nama.getText().toString());
rp.add("alamat", edt_alamat.getText().toString());
rp.add("nohp", edt_nohp.getText().toString());
rp.add("password", edt_pass.getText().toString());

konek.post("member.php?aksi=input",rp,new JsonHttpResponseHandler(){
    @Override
    public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
        super.onFailure(statusCode, headers, throwable, errorResponse);

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        //Log.e("Erro pengiriman ", errorResponse.toString());
    }

    @Override
    public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
        super.onFailure(statusCode, headers, responseString, throwable);

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        //Log.e("Erro pengiriman ", responseString.toString());
    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
        super.onSuccess(statusCode, headers, response);

        try{

            final int sukses    = Integer.valueOf(response.getString("success"));

            if (sukses == 1){

                finish();
            }
        }
    }
}

```

```

        Toast.makeText(getApplicationContext(),"Berhasil Daftar Silakan
Login !!",Toast.LENGTH_LONG).show();

        }else {
            Toast.makeText(getApplicationContext(),"Gagal Simpan
",Toast.LENGTH_SHORT).show();
        }
    }catch (JSONException e){

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(),"Internet tidak tersambung
... ",Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

}
});
}

private void lihatdata(){

    loading = new Loading(this);
    loading.panggil();

    konek.get("member.php?aksi=lihatdata&id="+b.getString("id"),new
JsonHttpResponseHandler(){

        @Override
        public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
            super.onFailure(statusCode, headers, throwable, errorResponse);

            loading.tutup();
            Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
            //Log.e("Errro pengiriman ", errorResponse.toString());

        }

        @Override

```

```

        public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
            super.onFailure(statusCode, headers, responseString, throwable);

            loading.tutup();
            Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
            //Log.e("Errro pengiriman ", responseString.toString());
        }

        @Override
        public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
            super.onSuccess(statusCode, headers, response);

            try {

                final int sukses = Integer.valueOf(response.getString("success"));

                if (sukses == 1) {

                    //item = new ArrayList<ItemSiswa>();

                    JSONArray jsonArray = response.getJSONArray("member");

                    for (int i = 0; i < jsonArray.length(); i++) {

                        JSONObject data = jsonArray.getJSONObject(i);

                        String ID = data.getString("id");
                        String NOKTP = data.getString("noktp");
                        String NAMA = data.getString("nama");
                        String NOHP = data.getString("nohp");
                        String ALAMAT = data.getString("alamat");
                        String PASSWORD = data.getString("password");

                        edt_noktp.setText(NOKTP);
                        edt_nama.setText(NAMA);
                        edt_nohp.setText(NOHP);
                        edt_alamat.setText(ALAMAT);
                        edt_pass.setText(PASSWORD);
                    }
                }
            }
        }
    }
}

```

```

        //item.add(new ItemSiswa(ID, NIS, KELAS, NAME,
HASILPRE, HASILPOST, HASILNILAI));

    }

    /*
    adapter = new AdapterAdmin(Admin.this, item);
    listView.setAdapter(adapter);
    */

    } else {
        Toast.makeText(getApplicationContext(), "Data tidak di temukan
", Toast.LENGTH_SHORT).show();
    }
    } catch (JSONException e) {

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(), "Internet tidak tersambung
... ", Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

    }
});
}

private void update(){
    loading = new Loading(this);
    loading.panggil();

    rp = new RequestParams();
    rp.add("id", b.getString("id"));
    rp.add("noktp", edt_noktp.getText().toString());
    rp.add("nama", edt_nama.getText().toString());
    rp.add("alamat", edt_alamat.getText().toString());
    rp.add("nohp", edt_nohp.getText().toString());
    rp.add("password", edt_pass.getText().toString());

    konek.post("member.php?aksi=update",rp,new JsonHttpResponseHandler(){

```

```

@Override
public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
    super.onFailure(statusCode, headers, throwable, errorResponse);

    loading.tutup();
    Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
    //Log.e("Erro pengiriman ", errorResponse.toString());
}

@Override
public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
    super.onFailure(statusCode, headers, responseString, throwable);

    loading.tutup();
    Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
    //Log.e("Erro pengiriman ", responseString.toString());
}

@Override
public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
    super.onSuccess(statusCode, headers, response);

    try{

        final int sukses    = Integer.valueOf(response.getString("success"));

        if (sukses == 1){

            finish();
            Intent i = new Intent(Daftar.this, DataMember.class);
            i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(i);
            Toast.makeText(getApplicationContext(),"Berhasil Ubah
!!",Toast.LENGTH_LONG).show();

```

```

        }else {
            Toast.makeText(getApplicationContext(),"Gagal Simpan
",Toast.LENGTH_SHORT).show();
        }
    }catch (JSONException e){

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(),"Internet tidak tersambung
... ",Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

    }
    });
}
}

```

4. Paket.java

```

public class Paket extends AppCompatActivity {

    ActionBar bar ;
    ListView listView ;
    Loading loading ;
    Konek konek ;
    AdapterPaket adapter;
    ArrayList<ItemPaket> item ;
    ArrayList<String> itemcek ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_paket);

        bar = getSupportActionBar();
        bar.setTitle("Paket");

        konek = new Konek(this);

        listView = (ListView) findViewById(R.id.listview);
    }
}

```

```

        ambildata();
    }

    private void ambildata(){
        loading = new Loading(this);
        loading.panggil();

        konek.get("paket.php?aksi=ambildata",new JsonHttpResponseHandler(){

            @Override
            public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
                super.onFailure(statusCode, headers, throwable, errorResponse);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", errorResponse.toString());
            }

            @Override
            public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
                super.onFailure(statusCode, headers, responseString, throwable);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", responseString.toString());
            }

            @Override
            public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
                super.onSuccess(statusCode, headers, response);

                try {

                    final int sukses = Integer.valueOf(response.getString("success"));

```

```

if (sukses == 1) {

    item = new ArrayList<ItemPaket>();
    itemcek = new ArrayList<String>();

    JSONArray jsonArray = response.getJSONArray("paket");

    for (int i = 0; i < jsonArray.length(); i++) {

        JSONObject data = jsonArray.getJSONObject(i);

        String ID = data.getString("id");
        String SEWA = data.getString("sewa");
        String ROUTE = data.getString("route");
        String KENDARAAN = data.getString("kendaraan");
        String HARGA = data.getString("harga");

        if (itemcek.contains(SEWA)){
            item.add(new ItemPaket(ID,
"",ROUTE,KENDARAAN,HARGA));
        }else {
            itemcek.add(SEWA);
            item.add(new ItemPaket(ID,
SEWA,ROUTE,KENDARAAN,HARGA));
        }

    }

}

adapter = new AdapterPaket(Paket.this, item,2);
listView.setAdapter(adapter);

} else {
    Toast.makeText(getApplicationContext(), "Data tidak di temukan
", Toast.LENGTH_SHORT).show();
}
} catch (JSONException e) {

```



```

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(), "Internet tidak tersambung
... ", Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

    }
    });
}
}

```

5. Profil.java

6.

```

public class Profil extends AppCompatActivity {

```

```

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profil);
    }
}

```

7. Datamember.java

```

public class DataMember extends AppCompatActivity {

```

```

    ActionBar bar ;
    ListView listView ;
    AdapterMember adapter ;
    ArrayList<ItemMember> item ;
    Konek konek ;
    Loading loading ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_member);

        bar = getSupportActionBar();
        bar.setTitle("Data Member");
    }
}

```

```

        listView    = (ListView) findViewById(R.id.listview);

        konek      = new Konek(this);

        ambildata();
    }

    private void ambildata(){
        loading    = new Loading(this);
        loading.panggil();

        konek.get("member.php?aksi=ambildata",new JsonHttpResponseHandler(){

            @Override
            public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
                super.onFailure(statusCode, headers, throwable, errorResponse);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", errorResponse.toString());
            }

            @Override
            public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
                super.onFailure(statusCode, headers, responseString, throwable);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", responseString.toString());
            }

            @Override
            public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
                super.onSuccess(statusCode, headers, response);
            }
        });
    }

```

```

try {

    final int sukses = Integer.valueOf(response.getString("success"));

    if (sukses == 1) {

        item = new ArrayList<ItemMember>();

        JSONArray jsonArray = response.getJSONArray("member");

        for (int i = 0; i < jsonArray.length(); i++) {

            JSONObject data = jsonArray.getJSONObject(i);

            String ID = data.getString("id");
            String NOKTP = data.getString("noktp");
            String NAMA = data.getString("nama");
            String NOHP = data.getString("nohp");
            String ALAMAT = data.getString("alamat");
            String PASSWORD = data.getString("password");

            item.add(new ItemMember(ID, NOKTP, NAMA, NOHP,
ALAMAT,PASSWORD));

        }

        adapter = new AdapterMember(DataMember.this, item,3);
        listView.setAdapter(adapter);

    } else {
        Toast.makeText(getApplicationContext(), "Data tidak di temukan
", Toast.LENGTH_SHORT).show();
    }
} catch (JSONException e) {

    Log.e("Error", e.getMessage());
    Toast.makeText(getApplicationContext(), "Internet tidak tersambung
... ", Toast.LENGTH_SHORT).show();

}

```

```

        loading.tutup();
    }
    });
}
}

```

8. Datapaket.java

```

public class DataPaket extends AppCompatActivity {

    ActionBar bar ;
    Konek konek ;
    AdapterPaket adapter ;
    ArrayList<ItemPaket> item ;
    ListView listView ;
    Loading loading ;
    Button tambah ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_paket);

        bar = getSupportActionBar() ;
        bar.setTitle("Data Paket");

        konek = new Konek(this);

        listView = (ListView) findViewById(R.id.listview);
        tambah = (Button) findViewById(R.id.btn_tambah);

        tambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent i = new Intent(DataPaket.this, TambahPaket.class);
                i.putExtra("key", "1");
                startActivity(i);

            }
        });

        ambildata();
    }
}

```

```

    }

    private void ambildata(){
        loading = new Loading(this);
        loading.panggil();

        konek.get("paket.php?aksi=ambildata",new JsonHttpResponseHandler){

            @Override
            public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
                super.onFailure(statusCode, headers, throwable, errorResponse);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", errorResponse.toString());
            }

            @Override
            public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
                super.onFailure(statusCode, headers, responseString, throwable);

                loading.tutup();
                Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
                //Log.e("Erro pengiriman ", responseString.toString());
            }

            @Override
            public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
                super.onSuccess(statusCode, headers, response);

                try {

                    final int sukses = Integer.valueOf(response.getString("success"));

                    if (sukses == 1) {

```

```

        item = new ArrayList<ItemPaket>();

        JSONArray jsonArray = response.getJSONArray("paket");

        for (int i = 0; i < jsonArray.length(); i++) {

            JSONObject data = jsonArray.getJSONObject(i);

            String ID = data.getString("id");
            String SEWA = data.getString("sewa");
            String ROUTE = data.getString("route");
            String KENDARAAN = data.getString("kendaraan");
            String HARGA = data.getString("harga");

            item.add(new ItemPaket(ID,
                SEWA,ROUTE,KENDARAAN,HARGA));

        }

        adapter = new AdapterPaket(DataPaket.this, item,1);
        listView.setAdapter(adapter);

    } else {
        Toast.makeText(getApplicationContext(), "Data tidak di temukan
", Toast.LENGTH_SHORT).show();
    }
    } catch (JSONException e) {

        Log.e("Error", e.getMessage());
        Toast.makeText(getApplicationContext(), "Internet tidak tersambung
... ", Toast.LENGTH_SHORT).show();

    }

    loading.tutup();

}
});

```

```
}  
}
```

9. Halamanadmin.java

```
public class HalamanAdmin extends AppCompatActivity {  
  
    ActionBar bar ;  
    LinearLayout ll_member , ll_datapaket , ll_pesanan , ll_profil ;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_halaman_admin);  
  
        bar = getSupportActionBar();  
        bar.setTitle("Halaman Admin");  
  
        ll_member = (LinearLayout) findViewById(R.id.ll_member);  
        ll_datapaket = (LinearLayout) findViewById(R.id.ll_datapaket);  
        ll_pesanan = (LinearLayout) findViewById(R.id.ll_pesanan);  
  
        ll_member.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
  
                Intent i = new Intent(HalamanAdmin.this, DataMember.class);  
                startActivity(i);  
  
            }  
        });  
  
        ll_datapaket.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
  
                Intent i = new Intent(HalamanAdmin.this, DataPaket.class);  
                startActivity(i);  
  
            }  
        });  
    }  
}
```

```

ll_pesanan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent i = new Intent(HalamanAdmin.this, Pesanan.class);
        startActivity(i);

    }
});
}
}

```

10. Pesanan.java

```

public class Pesanan extends AppCompatActivity {

    ActionBar bar ;
    Konek konek ;
    Loading loading ;
    ListView listView ;
    AdapterPesanan adapter ;
    ArrayList<ItemPesanan> item ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pesanan);

        bar = getSupportActionBar() ;
        bar.setTitle("Pesanan Member");
        konek = new Konek(this);

        listView = (ListView) findViewById(R.id.listview);

        ambildata();

    }

    private void ambildata(){

```



```

loading    = new Loading(this);
loading.panggil();

konek.get("pesanan.php?aksi=ambildata",new JsonHttpResponseHandler(){

    @Override
    public void onFailure(int statusCode, Header[] headers, Throwable
throwable, JSONObject errorResponse) {
        super.onFailure(statusCode, headers, throwable, errorResponse);

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        //Log.e("Erro pengiriman ", errorResponse.toString());

    }

    @Override
    public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
        super.onFailure(statusCode, headers, responseString, throwable);

        loading.tutup();
        Toast.makeText(getApplicationContext(),"Terjadi kesalahan silakan
ulang lagi ",Toast.LENGTH_SHORT).show();
        Log.e("Erro pengiriman ", responseString.toString());

    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, JSONObject
response) {
        super.onSuccess(statusCode, headers, response);

        try {

            final int sukses = Integer.valueOf(response.getString("success"));

            if (sukses == 1) {

                item = new ArrayList<ItemPesanan>();

```

```

JSONArray jsonArray = response.getJSONArray("pesanan");

for (int i = 0; i < jsonArray.length(); i++) {

    JSONObject data = jsonArray.getJSONObject(i);

    String ID = data.getString("id");
    String IDm = data.getString("idmember");
    String IDp = data.getString("idpaket");
    String STATUS = data.getString("status");
    String TGL = data.getString("tgl");
    String TGLBERANGKAT = data.getString("tglberangkat");
    String TGLTERAKHIR = data.getString("tglterakhir");
    String BATAS = data.getString("batas");
    String NOKTP = data.getString("noktp");
    String NAMA = data.getString("nama");
    String NOHP = data.getString("nohp");
    String ALAMAT = data.getString("alamat");
    String SEWA = data.getString("sewa");
    String ROUTE = data.getString("route");
    String KENDARAAN = data.getString("kendaraan");
    String HARGA = data.getString("harga");

    item.add(new
ItemPesanan(ID,IDm,IDp,STATUS,TGL,SEWA,ROUTE,KENDARAAN,HARG
A, NOKTP, NAMA, NOHP,
ALAMAT,TGLBERANGKAT,TGLTERAKHIR,BATAS));

}

adapter = new AdapterPesanan(Pesanan.this, item);
listView.setAdapter(adapter);

} else {
    Toast.makeText(getApplicationContext(), "Data tidak di temukan
", Toast.LENGTH_SHORT).show();
}
} catch (JSONException e) {

    Log.e("Error", e.getMessage());
}

```

```
        Toast.makeText(getApplicationContext(), "Internet tidak tersambung  
... ", Toast.LENGTH_SHORT).show();  
    }  
    loading.tutup();  
    }  
});  
}  
}
```