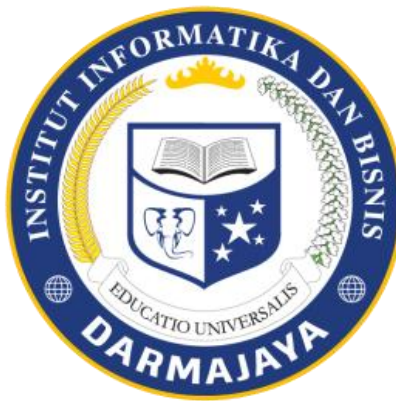


**RANCANG BANGUN SISTEM INFORMASI E-TICKETING BAGI PARA
PENGUNJUNG WISNUS DAN WISMAN PADA DESTINASI WISATA PESAWARAN
BERBASIS ANDROID**

SKRIPSI



Oleh :

RESTI RIANA PUTRI

1511050154

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2019**

PERNYATAAN ORISINILITAS PENELITIAN



PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa tugas akhir yang saya ajukan ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 19 Desember 2019



RESTI RIANA PUTRI
1511050154

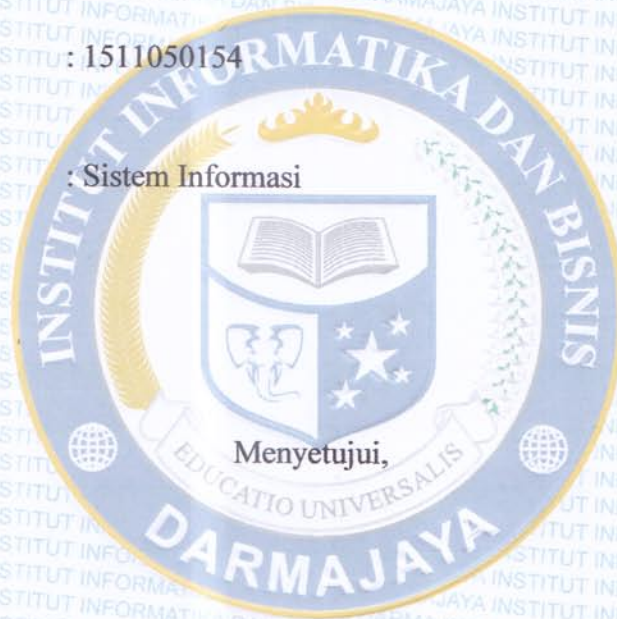
HALAMAN PERSETUJUAN

Judul Skripsi : Rancang Bangun Sistem Informasi E-Ticketing Bagi Para Pengunjung Wisnus dan Wisman Pada Destinasi Wisata Pesawaran Berbasis Android

Nama Mahasiswa : RESTI RIANA PUTRI

NPM : 1511050154

Program Studi : Sistem Informasi



Dosen Pembimbing,

Ketua Program Studi

Sistem Informasi

Indera, S.kom, M.TI

Nurjoko, S.Kom., M.Kom

NIK. 00371101

NIK.00440702

HALAMAN PENGESAHAN

Telah diuji dan dipertahankan didepan Tim Penguji Skripsi Program Studi Sistem Informasi IIB Darmajaya dan dinyatakan diterima untuk memenuhi syarat guna memperoleh gelar Sarjana Muda Komputer.

Mengesahkan,

1. Tim Penguji :

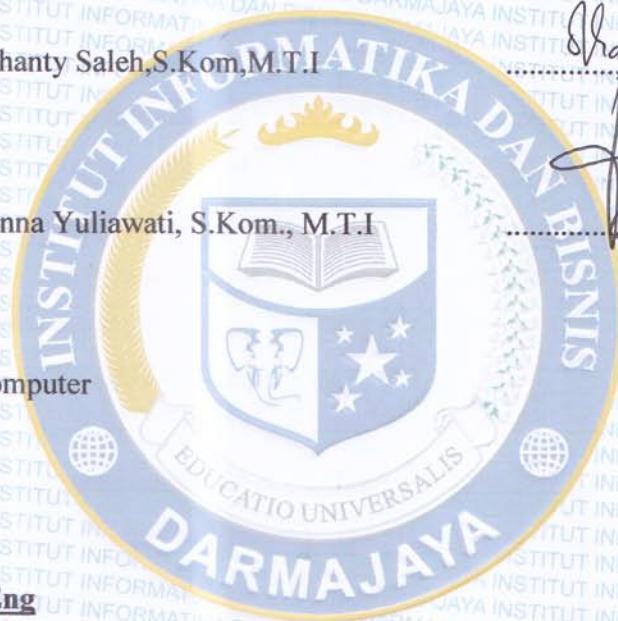
Tanda tangan

Ketua : Sushanty Saleh, S.Kom, M.T.I



Anggota : Donna Yuliawati, S.Kom., M.T.I

2. Dekan Fakultas Ilmu Komputer



Tanggal Lulus Ujian Skripsi : 17 Septemeber 2019

ABSTRAK

Rancang Bangun Sistem Informasi E-Ticketing Bagi Para Pengunjung Wisnus dan Wisman Pada Destinasi Wisata Pesawaran Berbasis Android

Oleh:

RESTI RIANA PUTRI

1511050154

Dengan perkembangan pemerintahan Kabupaten Pesawaran khususnya Dinas Parawisata Pesawaran telah meningkatkan dan memperluas destinasi wisata pesawaran dimana terdapat destinasi yang membuat pengunjung terpesona akan keindahan destinasi tersebut, termasuk pulau pahawang dan tegal mas yang akhir-akhir ini sangat ramai di perbincangkan karena keindahan alam dan bentuk penginapan seperti di pedesaan yang masih hangat akan suasana alam dan hotel maldivis untuk memudahkan para pengunjung local maupun pengunjung baru (*tourism*) maupun yang sudah menetap di lampung-pesawaran agar tidak terdapat kesulitan memilih destinasi wisata yang harus dikunjungi. Namun informasi yang didapatkan oleh masing-masing wisatawan sangat disayangkan dengan sedikitnya informasi serta estimasi biaya dan penggunaan yang cepat untuk menuju tempat wisata tersebut. Ditengah perkembangan Teknologi sekarang semakin berkembang *smartphone* selalu dalam genggamannya setiap waktunya, informasi yang begitu cepat ditangkap oleh kalangan jaman sekarang. Dibalik itu semua sekarang telah hadir *technology industry* level 4.0 dimana kalangan bisnis atau era modern semua aktivitas menggunakan teknologi demi mempercepat suatu informasi dan dapat menyelesaikan permasalahan-permasalahan yang sedang menjadi polemik dalam setiap organisasi, perusahaan, atau instansi perusahaan khususnya para pecinta wisata. Maka dengan membangun sebuah aplikasi akan memudahkan para pengunjung untuk mengunjungi wisata.

Kata kunci : Wisataku,Android,Pesawaran.

ABSTRACT

The Designe Of The *E-Ticketing On* Android-Based Information System For The Wisnus And Wisman Tourism Of Pesawaran Destinastion

By : Resti Riana Putri

1511050154

Following By The Government Of The Pesawaran District Government, Especially The Tourism Office Has Increased And Expanded Tourist Destinastions, Where There Are Destinastions That Makes Visitors Fascinated By The Beuty Of These Destinastions, Including The Island Of Pahawang And Tegal Mas Which Lately Is Very Lively Discussed Because Of The Beauty Of Nature And Forms Of Lodging Such As In The Countryside The Feel Still Warmer To Become Nature. The Atmosphere And New Visitors As Well As Those Who Have Settled In Lampung. So That There Are Will Not Get Difficulties When Choosing Tourist Destinastions To Visit. However, The Information Obtained By Each Tourist Is Unfortunate Cost And Rapid Use To Get Tourist Site. Amid The Development Of Technology Is Now Increasingly Developing Smartphone Are Always On Our Hand In The Same Time. Information That Is Being Quickly Captured By All Circles, Behind It All ,Now Has Industry Technology 4.0 Where Business Circles Or Modern Areas All Activities Using Technology For The Sake Of To Speed Up Information And Be Able To Solve Problems That Are Becoming Polemic In Every Organization And Company Especially The Lovera Of Tourism. By The Build An Application Program It Will Makes Be Easier For Visitors To Visit Tourism Destinastion.

Keyword : Wisataku,Android,Pesawaran



DAFTAR ISI

PERNYATAAN ORISINILITAS PENELITIAN	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSEMBAHAN	iv
MOTTO	v
ABSTRAK	vi
ABSTRACT	vii
RIWAYAT HIDUP	viii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 E-Ticketing	5
2.2 Destinasi Wisata	5
2.3 Wisata Pesawaran	5
2.4 Aplikasi	6
2.5 Android	7
2.6 Perangkat Lunak yang digunakan	7
2.7 Metode Pengembangan Perangkat Lunak	11
2.8 Pengujian Black-Box Testing	12
2.9 Sistem Pemodelan	13
2.10 Sistem Informasi `	14
2.11 Basis Data	14
2.12 Kamus Data	15

2.13 Entity Relation Diagram (ERD)	16
2.14 Flowchart	17
BAB III METODOLOGI PENELITIAN.....	20
3.1 Diagram Alir Pemecah Masalah	20
3.2 Metode Pengumpulan Data	21
3.3 Perencanaan Secara Cepat	22
3.4 Analisis Sistem Yang Berjalan	23
3.5 Desain Global Sistem Baru	26
3.6 Desain Rancangan Web	28
3.7 Desain Aplikasi.....	31
3.8 Rancangan Database	35
3.9 Rancangan Logika Program	41
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	45
4.1 Hasil Implementasi Pada Aplikasi	45
4.2 Hasil Implementasi Pada Web	50
4.3 Kelebihan dan Kelemahan Aplikasi	53
BAB V SIMPULAN DAN SARAN	55
5.1 Simpulan	55
5.2 Saran	55
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

- Gambar 2.1 Tampilan *Balsamiq*mockup
- Gambar 2.2 Layar Kerja Visio Untuk Flowchart
- Gambar 2.3 Layar Kerja Visio Untuk Relasi Antar Table
- Gambar 2.4 Gambar Waterfall
- Gambar 2.5 Simbol Dfd
- Gambar 2.6 Simbol Kamus Data
- Gambar 2.7 Simbol Flowchart
- Gambar 3.1 *Waterfall*
- Gambar 3.2 Flowchart System Promosi
- Gambar 3.3 Alur Proses Booking System Berjalan
- Gambar 3.4 Konteks Diagram
- Gambar 3.5 DFD Level 0.0
- Gambar 3.6 DFD Level 1 Registrasi Data Customer
- Gambar 3.7 DFD Level 2 Booking Wisata
- Gambar 3.8 DFD Level 3 Pembayaran
- Gambar 3.9 Halaman Login Pada Web
- Gambar 3.10 Halaman Home Pada Web
- Gambar 3.11 Halaman User Feedback
- Gambar 3.12 Input Wisata Pada Web
- Gambar 3.13 Fasilitas Input Wisata
- Gambar 3.14 Halaman Log In
- Gambar 3.15 Halaman Registrasi
- Gambar 3.16 Menu Home Pada Aplikasi
- Gambar 3.17 Halaman Registrasi
- Gambar 3.18 Menu Home
- Gambar 3.19 Halaman Recommended Wisata
- Gambar 3.20 Halaman Fasilitas Wisata Pada Aplikasi

Gambar 3.21 Booking Wisata

Gambar 3.23 ERD

Gambar 3.24 Relasi Antar Table

Gambar 3.25 HIPO

Gambar 3.26 Logika Log In

Gambar 3.25 Logika Home

Gambar 3.26 Logika Recommended Wisata

Gambar 3.27 Logika Booking

Gambar 3.28 Logika Input Admin

Gambar 3.29 Logika Simpan Data

DAFTAR TABEL

Tabel 3.1 tabel user	17
Tabel 3.2 Tabel booking	26
Tabel 3.3 Tabel review.....	26
Tabel 3.4 Tabel hotel.....	27
Tabel 3.5 Tabel room	28

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dengan perkembangan pemerintahan Kabupaten Pesawaran khususnya Dinas Parawisata Pesawaran telah meningkatkan dan memperluas destinasi wisata pesawaran dimana terdapat destinasi yang membuat pengunjung terpesona akan keindahan destinasi tersebut, termasuk pulau pahawang dan tegal mas yang akhir-akhir ini sangat ramai di perbincangkan karena keindahan alam dan bentuk penginapan seperti di kota dan hotel maldivis . Namun informasi yang didapatkan oleh masing-masing wisatawan sangat disayangkan dengan sedikitnya informasi serta estimasi biaya dan penggunaan yang cepat untuk menuju tempat wisata tersebut.

Ditengah perkembangan Teknologi sekarang semakin berkembang *smartphone* selalu dalam genggam setiap waktunya, informasi yang begitu cepat ditangkap oleh kalangan jaman sekarang. Dibalik itu semua sekarang telah hadir *technology industry* level 4.0 dimana kalangan bisnis atau era modern semua aktivitas menggunakan teknologi demi mempercepat suatu informasi dan dapat menyelesaikan permasalahan-permasalahan yang sedang menjadi polemik dalam setiap organisasi, perusahaan, atau instansi perusahaan khususnya para pecinta wisata.

Maka dalam hal ini diperlukan sebuah aplikasi sistem informasi yang dapat menampilkan sebuah tampilan layar destinasi wisata dan fasilitas yang tersedia ,seluruh data informasi masing-masing destinasi untuk kepuasan pelanggan dan pengunjung, dengan memanfaatkan teknologi berupa *smartphone* sehingga dapat digunakan dengan cepat.

Oleh karena itu dibutuhkan aplikasi tersebut untuk memudahkan para pengunjung local maupun pengunjung baru (*tourism*) maupun yang sudah menetap di lampung-pesawaran agar tidak terdapat kesulitan memilih destinasi wisata yang harus dikunjungi. Berdasarkan latar belakang yang telah dijelaskan maka pada penelitian skripsi ini mengangkat Judul “**RANCANG BANGUN SISTEM INFORMASI E-TICKETING BAGI PARA PENGUNJUNG WISNUS DAN**

WSIMAN PADA DESTINASI WISATA PESAWARAN BERBASIS ANDROID". maka penulis membangun sebuah sistem informasi destinasi wisata yang ada di Pesawaran. Diharapkan pengunjung wisata baik dari Lampung dan luar Lampung bahkan luar Indonesia pun mendapatkan pemesanan dan informasi dengan cepat dan puas sehingga destinasi Pesawaran menjadi destinasi wisata yang Populer di Indonesia-Lampung.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas telah dijelaskan, berikut adalah beberapa rumusan masalah yang ditemukan :

1. Pengunjung wisata terdapat kesulitan dan estimasi dana yang tidak tertuda(diluar konsep) serta informasi yang berbeda.
2. Bagaimana cara memberikan informasi tentang destinasi pesawaran dengan tepat.
3. Bagaimana membangun sistem informasi E-Ticket destinasi wisata pesawaran berbasis android?.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini agar pembahasan tidak melebihi batas yang sudah ditetapkan oleh penulis :

1. Data obyek yang di input oleh admin wisata dalam database ialah data destinasi wisata pesawaran yaitu pahawang dan tegal mas.
2. Aplikasi ini menggunakan konfirmasi melalui *e-mail* untuk konfirmasi ticket yang sudah di *booking* sehingga memudahkan dan meyakinkan pengunjung.
3. Aplikasi ini membutuhkan koneksi internet untuk dapat Registrasi dan pemesanan wisata.
4. Deadline konfirmasi masih berjalan dalam *calling*.

1.4 Tujuan Penelitian

Tujuan penelitian ini ialah menghasilkan dalam bentuk sistem berbasis *android* yang dapat menampilkan destinasi wisata dengan kelengkapan fasilitas dan estimasi biaya berupa *e-ticket* dan perbandingan keberangkatan dengan menggunakan aplikasi dan yang tidak menggunakan aplikasi serta memudahkan para pecinta wisata.

1.5 Manfaat Penelitian

Adapun Manfaat penelitian dari penelitian ini, antara lain :

1. Perancangan aplikasi android ini dapat membantu dan memudahkan para pengunjung local maupun non local (*tourism*) untuk menemukan solusi terbaik di akhir pekan destinasi mana yang akan dikunjungi dan memberikan informasi harga serta tempat yang tersedia pada tiap destinasi masing-masing.
2. Menghemat biaya bagi para pengunjung karena biaya sudah termasuk dalam paket wisata di aplikasi
3. Efisien dan efektif.

1.6 Sistematika Penulisan

Dalam penulisan skripsi ini terdiri dari 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pada bab ini berisi latar belakang masalah, identifikasi masalah, rumusan masalah, ruanglingkup penelitian, tujuan dan manfaat penelitian serta sistematika penulisan.

BAB II Tinjauan Pustaka

Bab ini berisikan teori-teori dasar yang berhubungan dengan penelitian yang dilakukan oleh penulis.

BAB III Metodologi Penelitian

Bab ini menjelaskan metode-metode penelitian yang akan digunakan dan membahas tentang perangkat lunak dan perangkat keras yang akan digunakan.

BAB IV Hasil Pembahasan

Bab ini berisikan hasil Analisis sitem yang sedang berjalan, rancangan sistem yang diusulkan, rancangan input dan output, dan hasil penggambaran atau bentuk dari website tersebut.

BAB V Penutup

Bab ini berisi kesimpulan dan saran-saran yang terkait dengan pembahasan dalam laporan.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1 E-Ticket

E-Ticket (electronic ticketing) adalah suatu tiket dalam bentuk digital. Semua informasi mengenai electronic ticketing di simpan secara digital dalam sistem komputer.

2.2 Destinasi Wisata

Dalam kamus besar Bahasa Indonesia (KBBI), disebutkan bahwa destinasi merupakan kata benda yang artinya lokasi / tempat tujuan ataupun tempat tujuan pengiriman. Istilah ini menunjukkan suatu tempat yang signifikan yang akan dituju ketika seorang melakukan perjalanan, maksud signifikan ialah bahwa destinasi atau tempat yang benar-benar dituju.

Destinasi pariwisata adalah suatu tempat yang dikunjungi dengan periode waktu yang cukup signifikan selama masa perjalanan seseorang jika dibandingkan dengan tempat lainnya yang dilalui selama perjalanan. (Pitana & I ketut Surya Diarta : 2009)

2.3 Wisata Pesawaran

Dinas pariwisata pesawaran yang berlokasikan di Gedung Tataan, Pesawaran Regency, Lampung 35366, Indonesia, memiliki tempat yang tidak kalah menarik dengan wisata lainnya, hal ini untuk meningkatkan kualitas kinerja pesawaran di wilayah pesawaran memiliki destinasi yang patut dibanggakan. Seperti halnya terdapat wisata alam, wisata buatan, wisata budaya, wisata kuliner. Pada beberapa tahun ini pesawaran telah meningkatkan atau menambahkan obyek wisata, yang sebelumnya pesawaran dikenal dengan obyek wisata pantai. Dinas pariwisata memiliki obyek wisata yang berkembang seperti halnya, pantai sari ringgung dapat menyebrang pulau di mana di tengah pulau terdapat obyek wisata lagi yang bernama pulau tegal mas, pasir timbul, pahwang dan cottage yang di tengah lautan, pulau nemo.

2.4 Aplikasi

Penelitian ini membangun aplikasi untuk pemesanan paket wisata. Aplikasi adalah sekumpulan atribut yang terdiri dari beberapa form, report yang disusun sedemikian rupa agar dapat mengakses data. Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data, secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal (Rahman dan Santoso 2015:79).

Aplikasi yang di bangun berbasiskan mobile yang dapat diterjemahkan sebagai perpindahan yang mudah dari suatu tempat ketempat yang lain, misalnya penggunaa layanan telepon nirkabel atau selular pintar. Dengan kata lain, aplikasi mobile adalah program yang digunakan melalalui telepon nirkabel atau seluler pintar (Ekawati 2017:7)

Karakteristik perangkat mobile adalah sebagai berikut:

1. Ukuran yang kecil : Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat mobile juga memiliki memory yang kecil, yaitu *primary* (RAM) dan *secondary* (disk).
3. Daya proses yang terbatas : Sistem mobile tidak setangguh rekan mereka yaitu desktop.
4. Mengonsumsi daya yang rendah : Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop
5. Kuat dan dapat diandalkan : Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek :Perangkat-perangkat konsumen ini menyala dalam hitungan detik.

2.5 Android

Sistem Operasi yang digunakan dalam aplikasi ini adalah android. Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Dikarenakan bersifat terbuka, maka pengguna dapat mengembangkan aplikasi berdasarkan kebutuhannya (Ekawati 2017:8).

2.6 Perangkat Lunak Yang Digunakan

2.6.1 Android Studio

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 Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan (Android Development Tools) ADT plugin (Juansah 2015).

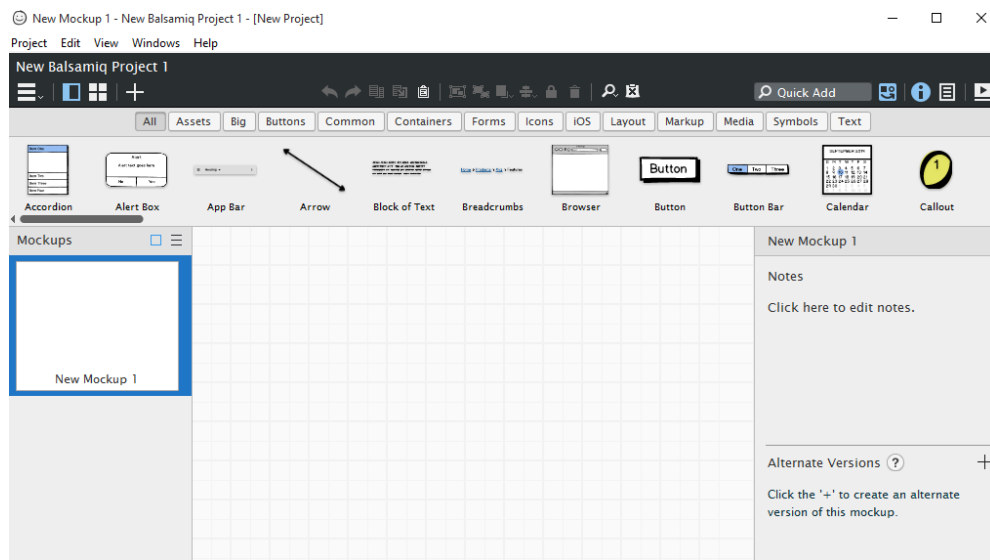
2.6.2 JAVA

Java adalah Bahasa yang dapat dijalankan disembarang platform, di beragam lingkungan internet, consumer electronic products, dan computer applications. Jadi berdasarkan pengertian diatas Java merupakan pemrograman yang dapat dijalankan multi platform (bisa berjalan di berbagai macam sistem operasi) yang mendukung Java dan menjalankan perangkat lunak ataupun jaringan (Afrizal 2017:3).

2.6.3 BalsamiqMockup

Balsamiq Mockup adalah program aplikasi yang digunakan dalam pembuatan tampilan User Interface sebuah aplikasi. Software ini sudah menyediakan tools yang dapat memudahkan dalam membuat desain Prototyping Website atau aplikasi yang akan dibuat. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna. Balsamiq Mockup juga dapat

diartikan sebagai salah satu Software yang berfungsi sebagai pembuatan desain. Software yang dapat mempermudah dalam menggambar sebuah tampilan User Interface. Berikut adalah tampilan halaman balsamiqmockup.

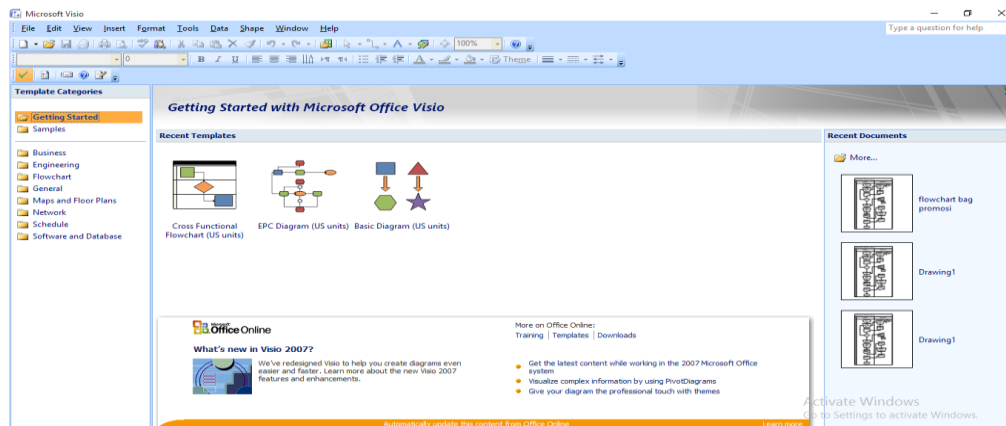


Gambar 2.1 tampilan *balsamiqmockup*

Pada bagian atas terdapat menu toolbar adalah bagian dimana kita dapat memilih untuk menu yang kita buat.

2.6.4 Microsoft Office Visio

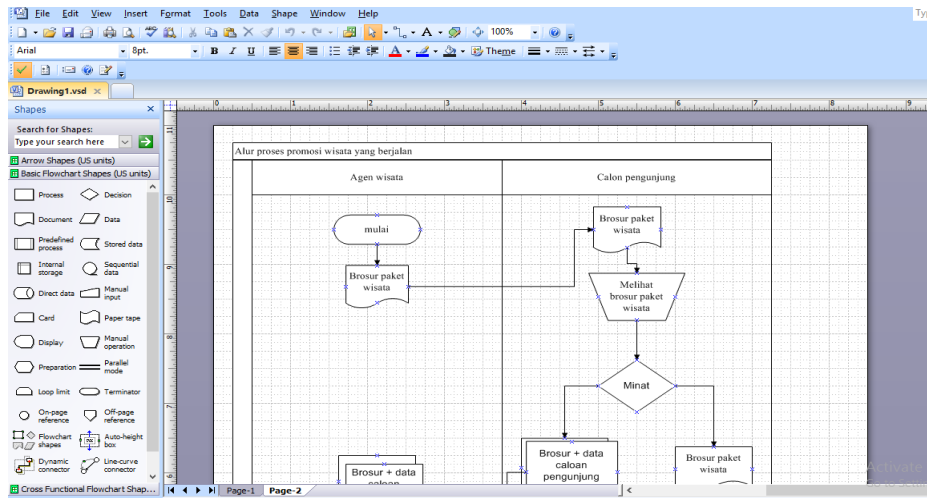
Microsoft Visio merupakan salah satu perangkat pemodelan untuk membuat diagram-diagram yang dibutuhkan dalam penggambaran proses bisnis. Visio dapat membantu *system analyst* sebagai pengguna dalam mempersiapkan penggambaran diagram yang biasa diperlukan, seperti ERD, DFD, rancangan *User Interface*, jaringan, Gantt Chart, dan masih banyak lainnya. Ketika pengguna memulai untuk melakukan aktivitas permodelan menggunakan Visio dengan membuka aplikasi, aplikasi akan menampilkan *interface* seperti di bawah ini:



Seperti yang terdapat pada gambar di atas, Visio mengkategorikan diagram-dia diagram berdasarkan kegunaannya. Pada artikel ini, yang akan lebih dikenalkan adalah flowchart, ERD, dan DFD karena ketiga diagram ini menggunakan pendekatan terstruktur.

1. Flowchart

Pada kategori "Business", pengguna dapat memilih Basic Flowchart template untuk memulai membuat diagram berupa flowchart. Untuk membuat Cross functional flowchart, pada menu bar, klik File-New-Flowchart-Cross Functional Flowchart-pilih Band orientation. Pada bagian kiri tampilan kerja Visio, terdapat Shapes yang di dalamnya terdapat notasi yang digunakan dalam membuat Flowchart. Notasi-notasi tersebut antara lain Process yang digunakan untuk menggambarkan proses yang terjadi dalam proses bisnis, Document yang digunakan untuk menggambarkan dokumen yang terdapat dalam proses bisnis, Decision yang digunakan untuk menggambarkan adanya alternatif pilihan, dan lain-lain yang berada pada Shapes dengan kategori Basic Flowchart Shapes.

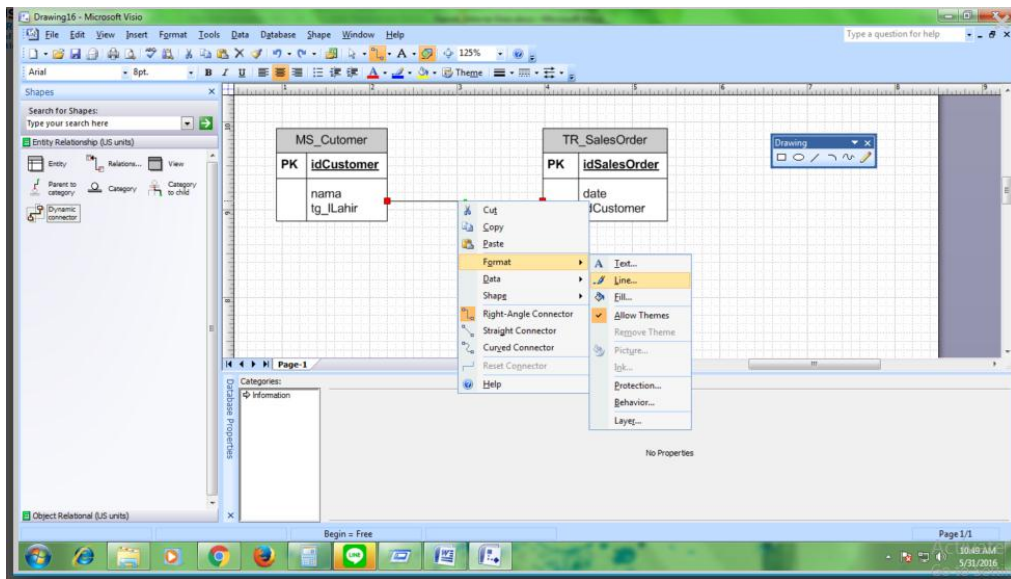


Gambar 2.2 layar kerja visio untuk *flowchart*

Untuk memberikan koneksi antarnotasi yang menggambarkan urutan aliran proses, pengguna dapat menggunakan Connector Tool yang terletak pada toolbar

2. ERD (*Entity Relationship Diagram*)

Pada kategori “Software and Database”, pengguna dapat memilih Database Model Diagram template untuk memulai membuat diagram berupa ERD. Pada bagian kiri tampilan kerja, terdapat notasi-notasi yang digunakan dalam membuat ERD, seperti Entity, Dynamic Connector, dan lain-lain. Untuk entity, pengguna dapat *men-define field* pada Database Properties. Dynamic Connector digunakan untuk mendefinisikan hubungan antarentitas dengan memberikan cardinality. Cara memberikan *cardinality* pada garis connector adalah dengan mengklik kanan connector-pilih Format-klik Line

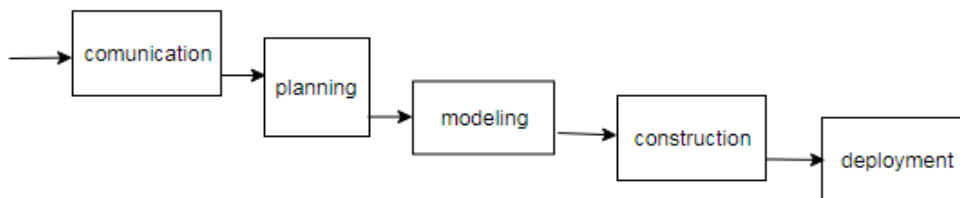


Gambar 2.3 layar kerja visio untuk relasi database

2.7 Metode Pengembang Perangkat Lunak

2.7.1 Metode Waterfall

Menurut Pressman (2015:42), model waterfall adalah model yang bersifat sistematis atau terstruktur seperti bentuk air terjun atau pun tangga jalan, Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 2.4 gambar *waterfall*

- a. Communication langkah awal untuk melakukan penelitian yaitu komunikasi. Hasil dari komunikasi tersebut adalah inisialisasi proyek,

seperti menganalisis permasalahan yang sedang berjalan dan mengumpulkan data-data yang diperlukan,.

- b. Planning (estimasi, penjadwalan, pekerjaan yang sejalan) tahapan perencanaan yang menjelaskan tentang teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, hasil akhir yang dihasilkan atau yang di tampilkan, penjadwalan pembuatan sistem yang akan dilakukan, dan tracking (pekerjaan yang sedang berjalan) proses pengerjaan sistem.
- c. Modeling (Analysis & Design) Tahapan ini adalah tahap penggambaran atau perancangan suatu model program atau sistem yang akan di bangun dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface, untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
- d. Construction (Code & Test) dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.
- e. Deployment (Delivery, Support, Feedback) Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software, dan pengembangan software diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

2.8 Pengujian *Black-Box Testing*

Metode pengujian pada aplikasi ini menggunakan *Black Box Testing* yaitu yang berfokus pada gambaran program. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing* (Pressman 2012). *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).

3. Kesalahan pada struktur data dan akses basis data.

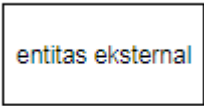
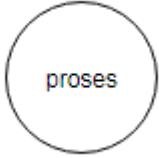
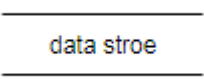
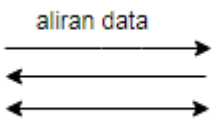
4. Kesalahan performansi (*performance errors*).

5. Kesalahan inisialisasi dan terminasi.

2.9 Sistem Pemodelan

2.9.1 Definisi Data Flow Diagram (DFD)

Data Flow Diagram atau DFD ialah suatu perangkat lunak yang menunjukkan alur data pada sistem dalam bentuk grafik. (Irwansyah, 2014:240). DFD ini merupakan alat perancangan sistem yang berada pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Simbol Data Flow Diagram Data Flow Diagram (DFD) merupakan salah satu alat untuk mengetahui aliran data yang mengalir dalam sistem. DFD menggunakan beberapa simbol sebagai berikut:

Yourdon/De Marco	Keterangan
	Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem.
	Orang, unit yang menggunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
	Aliran data dengan arah khusus dari sumber ke tujuan.
	Penyimpanan data atau tempat data direfer oleh proses.

Gambar 2.5 simbol dfd sumber Srimulyani 2016

2.10 Sistem informasi

2.10.1 Sistem

Sistem adalah suatu proses/ prosedur yang dibuat menurut pola yang ditentukan untuk melaksanakan suatu kegiatan. (Mulyadi 2016).

Ada 3 elemen yang membentuk sebuah sistem yaitu :

- *Input*
Segala sesuatu atau data yang masuk kedalam sistem dan selanjut menjadi bahan untuk di proses selanjutnya.
- *Proses*
Bagian yang melakukan perubahan dari *input* menjadi *output* yang berguna, misalnya berupa informasi berbahan matang, hasil sebuah informasi atau hasil sebuah proses.
- *Output*
Hasil dari pemrosesan, misalnya sesuatu yang dapat dilihat dari sebuah proses, laporan , hasil ini akhir dari ke tiga elemen.

2.10.2 Informasi

Informasi adalah “data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat”.(Krismaji (2015:14)

Sistem informasi adalah “sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai”.(Kadir (2014:9)

2.11 Basis data

Basis data merupakan tempat pengolahan informasi dan penyimpanan informasi secara terdistribusi dalam media komputer. sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah menjaga data yang sudah diolah atau informasi agar terjaga dan serta membuat informasi tersedia saat dibutuhkan”. Rosa dan Shalahuddin (2015:43).

Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang sering digunakan dalam pengolahan basis data yaitu MySQL dan phpMyAdmin.

1. MySQL adalah aplikasi basis data dan Bahasa yang sering digunakan untuk mengolah dan menata file-file dalam database.

Menurut Manurung (2015:39)“MySQL merupakan turunan salah satu konsep utama basis data yang sudah ada sebelumnya SQL yaitu pengoperasian basisdata”

2. PHP MyAdmin selain MySQL, aplikasi yang dapat mengolah basis data yaitu phpMyAdmin. PhpMyAdmin memiliki fungsi yang sama dengan MySQL, namun untuk pengaksesan aplikasi menggunakan browser. Menurut Rahman (2013:21) “phpMyAdmin adalah sebuah software berbasis pemrograman yang dipergunakan sebagai administrator MySQL melalui browser(web) yang digunakan untuk manajemen database”

2.12 Kamus Data

Menurut Rosa dan Shalahudin (2013:73), Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur.

Kamus data biasanya berisi :

- a. Nama-nama dari datab.Digunakan pada-merupakan proses-proses yang terkait datac.
- b. Deskripsi –merupakan deskripsi data.
- c. Informasi tambahan –seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

No.	Simbol	Keterangan
1	=	Disusun atau terdiri dari
2	+	Dan
3	[]	Baik ... atau ...
4	{ } ⁿ	n kali diulang/bernilai banyak
5	()	Data opsional
6	*...*	Batas komentar

Sumber : Sukamto dan Shalahudin (2013:74)

Gambar 2.6 simbol kamus data

2.13 *Entity Relationship Diagram*(ERD)

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut. Menurut Rosa dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”.

Simbol-simbol atau komponen-komponen yang digunakan dalam penggambaran *Entity Relationship Diagram*(ERD) yaitu:

1. Entitas (entity) Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

2. Atribut (Attribute) Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data.
3. Atribut kunci primer Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id.
4. Atribut multivalued (multivalued) Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5. Relasi (relation) Relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua entity yang keterangannya perlu disimpan dalam basis data.

2.14 Flowchart

2.14.1 Definisi Flowchart


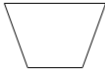
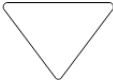
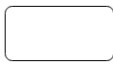


Flowchart menjelaskan alur jalannya suatu proses sebelum dibuatnya program, tahapan dan urutannya secara grafis. Flowchart berisi simbol yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart dapat didefinisikan sebagai sebuah gambaran yang menjelaskan proses yang akan dibentuk. Selain itu, flowchart biasanya digunakan untuk merencanakan tahapan suatu kegiatan yang dilakukan. Jadi, Flowchart atau bagan alur merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah (prosedur) beserta liran data dengan symbol-simbol standar yang mudah dipahami. (Astuti, 2016:31)..



Menurut Astuti (2016: 35) Flowchart yang baik setidaknya mencakup hal-hal berikut:

- memberi gambaran seluruh tahapan proses
- Dapat menjelaskan dan membantu pemecahan masalah
- Mengidentifikasi titik proses yang kritis untuk pengendalian
- Menggambarkan peluang perbaikan.

2.14.2 Jenis Flowchart dan simbol flowchart

Menurut Astuti (2016: 38) flowchart dapat dibedakan ke dalam tiga kategori, yaitu flowchart dokumen, flowchart sistem dan flowchart program. Flowchart dokumen menggambarkan aliran data dan informasi antar- area (divisi) dalam sebuah organisasi. Sedangkan flowchart sistem lebih menekankan pada gambaran tentang aliran input, prosedur pemrosesan dan output yang dihasilkan. Dan flowchart program lebih banyak digunakan untuk menggambarkan logika suatu prosedur penyelesaian masalah dalam pemrograman komputer.

Simbol	Keterangan
 Dokumen manual	Suatu dokumen yang berupa kumpulan dokumen yang terlihat bentuk nya
 Proses manual	Menyatakan suatu kegiatan atau tindakan proses yang tidak dilakukan oleh komputer
 Arsip manual	Penyimpanan data secara bersusun dalam bentuk dokumen atau file yang tidak menggunakan komputer
 Arsip komputer	Penyimpanan data secara bersusun dalam bentuk dokumen atau file yang menggunakan komputer
 Input manual	Memasukan data informasi tanpa menggunakan komputer yaitu tulis tangan
 Start/end	Suatu kegiatan untuk memulai dan berhenti

 decision	Suatu tindakan pilihan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak
 Garis alir	Menyatakan jalannya arus suatu proses

Gambar 2.7 simbol flowchart

BAB III METODE PENELITIAN

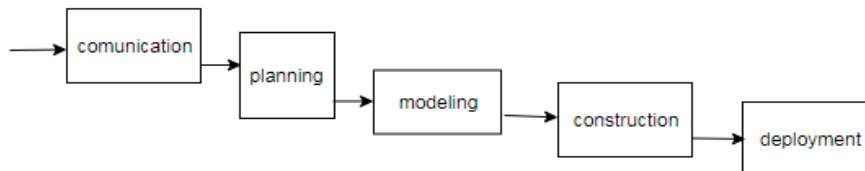
3.1 Diagram Alir Pemecah Masalah

Penulis menyusun metode penelitian yang akan digunakan dalam penelitian, dalam bentuk alur untuk menunjukkan tahapan yang dilakukan untuk mendapatkan informasi.

3.1.1 Metode Pengembangan Sistem

1. Metode Waterfall

Menurut Pressman (2015:42), model waterfall adalah model yang bersifat sistematis atau terstruktur seperti bentuk air terjun atau pun tangga jalan. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 3.1 gambar *waterfall*

- a. Communication (Project Initiation & Requirements Gathering) Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi software. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

- b. Planning (estimasi ,penjadwalan,dan proses jalan psistem)
- c. Modeling (Analysis & Design) Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
- d. Construction (Code & Test) Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.
- e. Deployment (Delivery, Support, Feedback) Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

3.2 Metode Pengumpulan Data

Metode penelitian yang dilakukan untuk memperoleh data yang dibutuhkan adalah sebagai berikut:

1. Wawancara
Melakukan tanya jawab pada salah satu pengelola wisata Pesawaran agar penelitian yang dilakukan sesuai dengan masalah yang terjadi serta mengumpulkan data-data yang dibutuhkan untuk menyelesaikan suatu permasalahan.
2. Observasi
Melakukan observasi dengan mengamati proses apa saja yang terjadi dan berdasarkan pengalaman sebagai pengunjung destinasi Pesawaran.
3. Studi literatur
Metode ini dilakukan dengan mencari sumber-sumber dan data-data yang mendukung dan diperlukan dalam pengembangan “Sistem booking wisata”. Berdasarkan pengumpulan data diatas maka dapat disimpulkan

bahwa permasalahan yang di dapatkan yaitu pada pelanggan yang mengeluhkan mengenai pembayaran yang di luar paket serta lebih mahal tanpa tour ,untuk itu diperlukan sistem yang dapat memfasilitasi pelanggan untuk melakukan pemesanan secara *online* dan meminimalisir keuangan serta hemat transportasi.

3.3 Perencanaan Secara Cepat

Berdasarkan pengamatan pendahuluan rencana cepat merupakan tahapan dengan melakukan analisis dan perencanaan setelah kita mendapatkan data-data dari tahapan komunikasi.

3.3.1 Analisis Kebutuhan Perangkat Lunak

- a. Perangkat lunak sistem operasi pada Laptop Asus adalah *Microsoft Windows 10 64 bit*.
- b. Perangkat lunak sistem operasi pada android minimal adalah Android versi 5.0 (*Lollipop*).
- c. Perangkat lunak untuk pembuatan program adalah Android Studio.

3.3.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yang digunakan untuk membuat dan menjalankan aplikasi adalah sebagai berikut :

1. Processor Intel Core i5.
2. Ram 4 Gb.
3. Harddisk 500 Gb.
4. Keyboard dan Mouse.

Spesifikasi diatas menurut peneliti sudah memenuhi standar untuk membuat dan menjalankan aplikasi Sistem aplikasi wisata di pesawat.

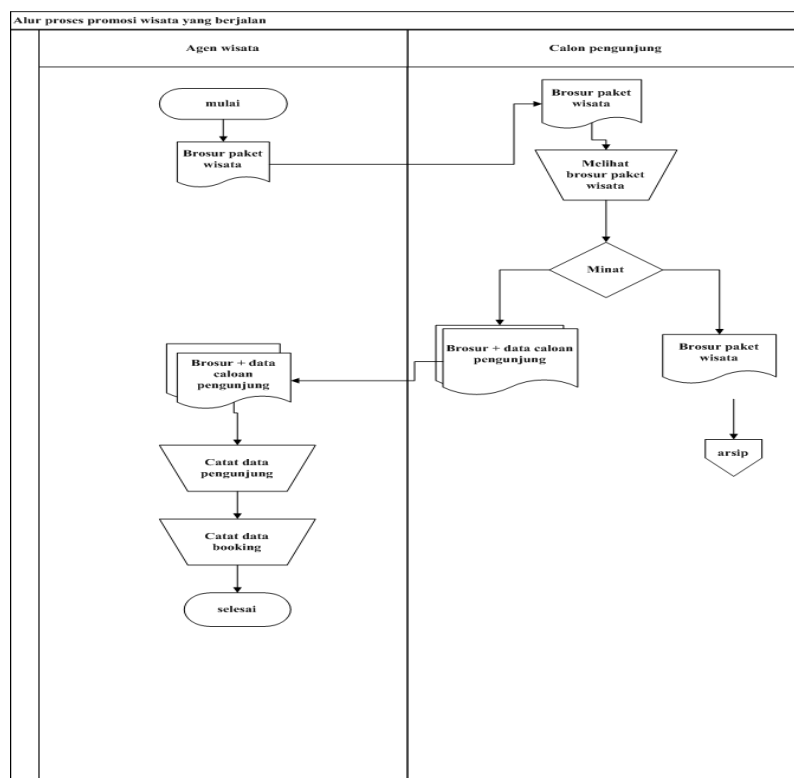
3.4 Analisis Sistem Yang Berjalan

Untuk mengetahui alur proses sistem berjalan secara rinci pada *booking* wisata menggunakan dokumen *flowchart*. *Flowchart* dibuat untuk menerangkan proses apa saja yang dikerjakan.

a. Analisis Flowchart pemasaran yang berjalan

1. Saat melakukan promosi atau penyebar informasi paket wisata para agen yang berhubungan dengan wisata harus membagikan brosur langsung kepada calon pengunjung atau customer dengan menunjukkan dan menjelaskan paket wisata dan no telfon.
2. Calon pengunjung/customer melihat brosur yang di bagikan,apabila minat berkunjung ke wisata maka akan menghubungi dan segera *booking*,apabila tidak berminat maka akan disimpan sebagai arsip bila suatu saat akan berkunjung.
3. Setelah ada yang berminat maka data pengunjung di catat serta kwitansi untuk pembuktian pembayaran.

Document *flowchart promosi* yang berjalan dapat dilihat pada gambar 3.2



Gambar 3.2 *flowchart sistem promosi*

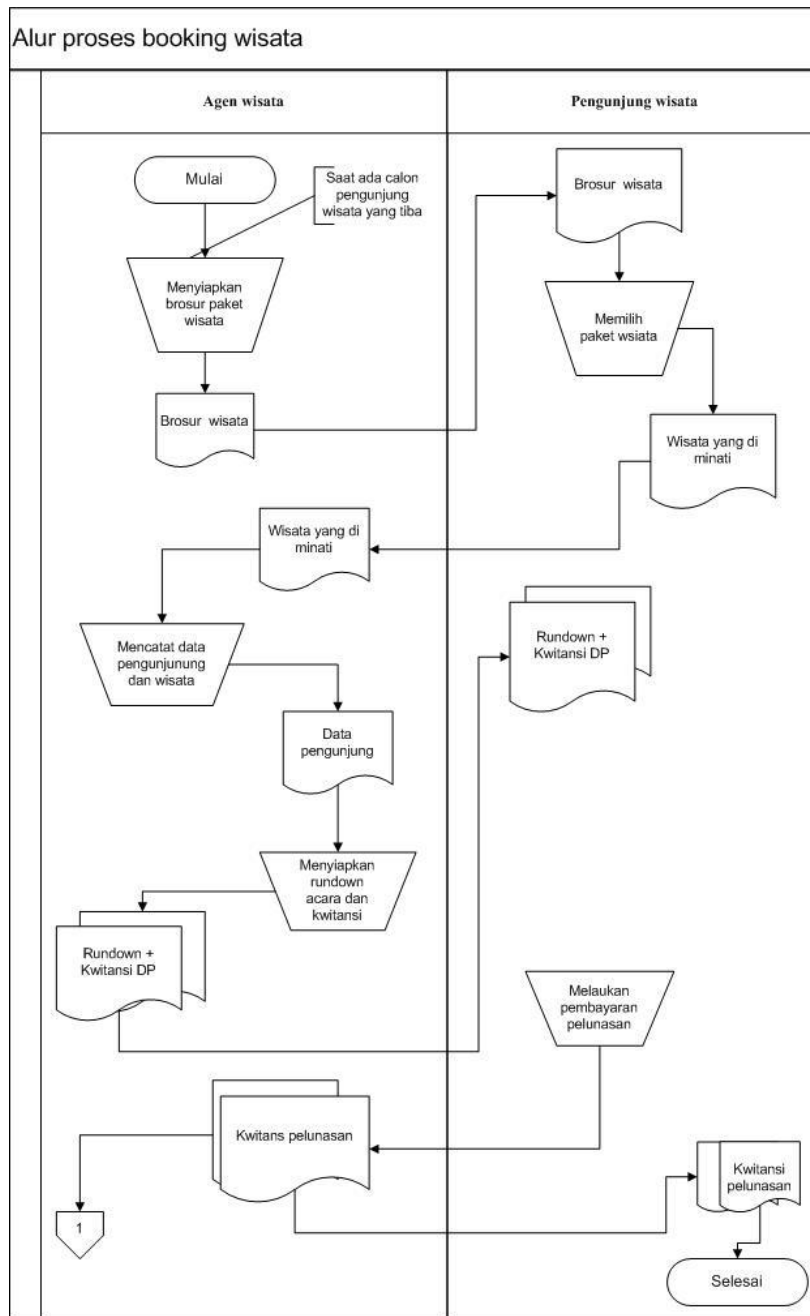
Dari analisa proses promosi yang berjalan terdapat beberapa masalah yang ditemukan ,yaitu :

1. Proses promosi yang cukup memakan waktu harus memungkinkan para calon pengunjung mencari informasi yang detail tentang wisata
2. Membuat brosur dengan sebanyak mungkin yang tidak tahu kedepannya aka nada pengunjung yang berminat atau tidak.

b. Analisis prosedur *booking wisata*

Berikut adalah alur proses *booking wisata* yang berjalan.

1. Saat ada customer atau calon pengunjung ingin *booking wisata* maka pihak agen memberikan brosur.
2. Customer memilih paket wisata serta harga dan fasilitas yang tersedia,jika customer berminat *booking* maka agen memberikan rincian pembayaran.
3. Jika sudah sepakat antara kedua pihak maka agen akan membuatkan *rundown* acara serta kwitansi untuk pembayaran awal.
4. Agen wisata menyiapkan *rundown* acara keberangkatan dan membuat kwitansi sebagai bukti pembayaran dan *pembookingan* untuk kepastian selanjutnya.
5. Setelah itu agen wisata memberikan *rundown* acara kepada pengunjung untuk mengikuti rule yang ada dan mengetahui keberangkatan
6. Tiba di lokasi wisata maka pengunjung melakukan pembayaran kedua yaitu pelunasan
7. Maka agen wisata membuatkan kembali kwitansi pelunasan dengan 2 rangkap,1 untuk pengunjung ,2 untuk arsip agen wisata,
8. Pengunjung dan agen wisata siap untuk mengelilingi wsiata yang telah di *booking*.



Gambar 3.3 alur proses *booking* pada sistem berjalan

Hasil analisis pada proses *booking* wisata yang sedang berjalan.

1. Proses pencatatan data pengunjung ,rundown acara fasilitas acara ,penghitungan harga yang dilakukan agen wisata dilakukan secara tertulis dengan bantuan alat tulis.

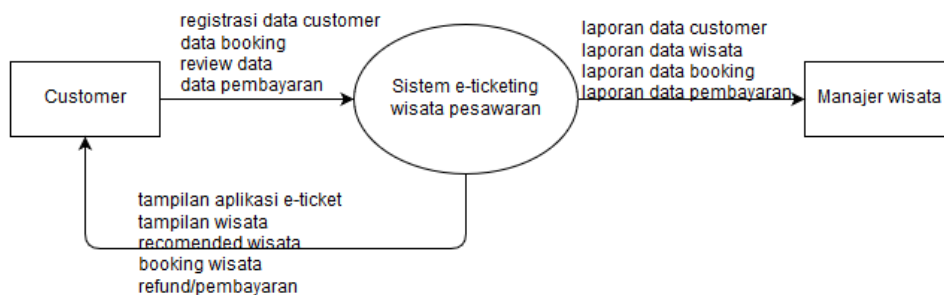
2. Proses pemilihan paket wisata, pengunjung harus datang kepada agen atau pihak orang lain sehingga terjadi informasi yang tidak akurat dan memakan waktu yang lama.
3. Tiba dilokasi agen tidak langsung memesan perahu atau wisata yang sudah di pesan oleh pengunjung karena belum adanya pelunasan maka pihak agen wisata mendapatkan ketrtundaan untuk *booking* wisata sehingga hanya mendapatkan keuntungan sebagian saja sehingga pembayaran diluar prediksi.

Maka dari itu perlunya ada sistem yang mampu mempermudah pengunjung melakukan *review* wisata tanpa harus tiba langsung ke wisata.

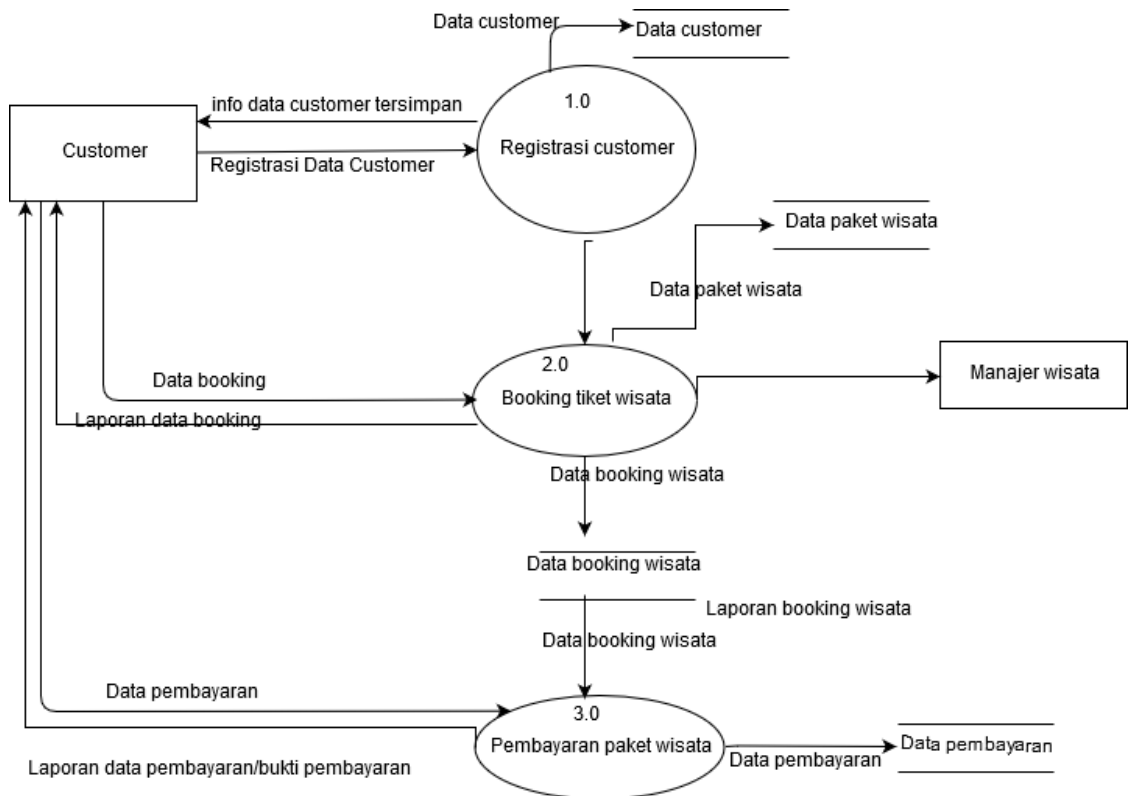
3.5 Desain Global Sistem Baru

3.5.2 Konteks Diagram

Konteks diagram gambar 3.3 merupakan alur sistem *booking* wisata terdapat entitas customer ,admin dan manajer wisata serta sub sistem *booking* wisata.

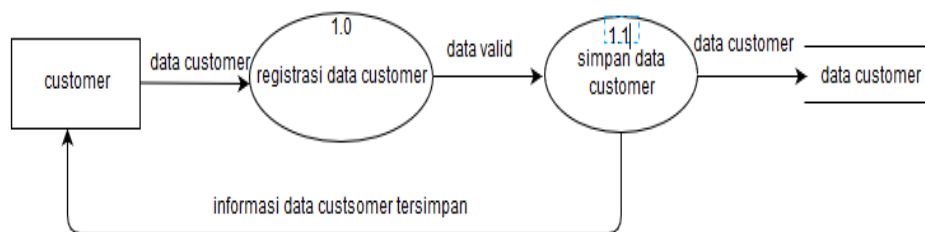


Gambar 3.4 konteks diagram



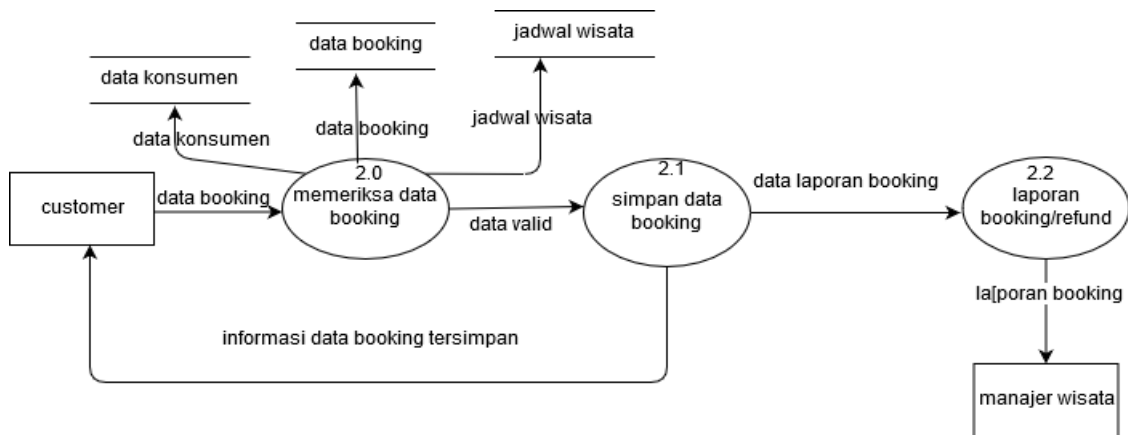
Gambar 3.5 DFD level 0.0

Gambar 3.6 menerangkan alur system registrasi untuk dapat booking e-ticket wisata, berikut gambar dibawah ini:



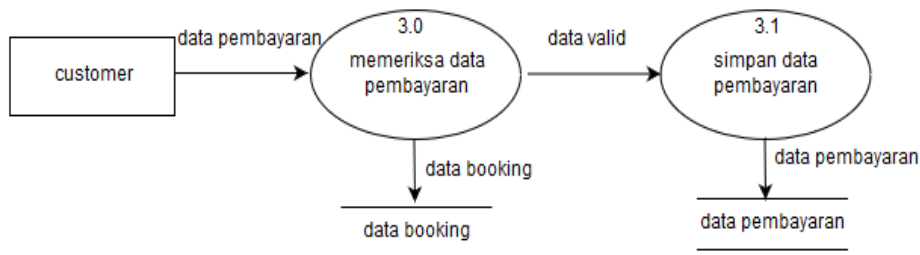
Gambar 3.6 DFD level 1 registrasi data customer

Gambar 3.7 menerangkan alur system *booking* wisata ,berikut gambar dibawah ini



Gambar 3.7 DFD level 2 *booking wisata*

Pada gambar 3.8 menerangkan alur system akhir dari semua proses yaitu pembayaran setelah melakukan *booking* maka customer melakukan pembayaran.



Gambar 3.8 DFD level 3 *pembayaran*

3.6 Desain Rancangan Web

a. Desain rancangan web halaman *log in*

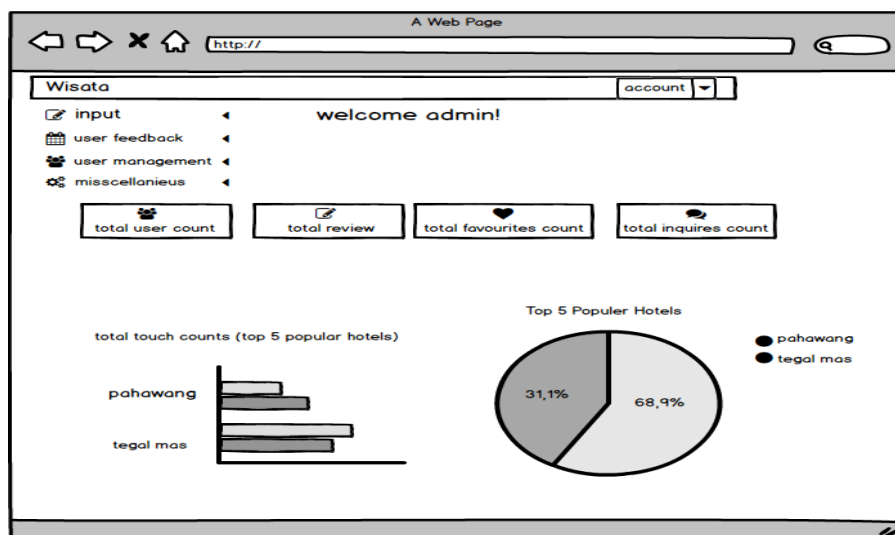
halaman inputan data sebagai admin ,sistem dimana untuk mengelola data dan menyimpan data serta data laporan ketika user *log in* dengan dataya maka admin mengelola data *user* sebagai bahan *reviews* pada tampilan aplikasi,serta mengelola data pemesanan dan ketiak pengunjung wisata *booking* ,maka tugas admin lah mengelola data pengunjung dan input wisata untuk bahan pertimbangan si pengunjung.



Gambar 3.8 halaman *login*

a. Halaman desain *home* pada web

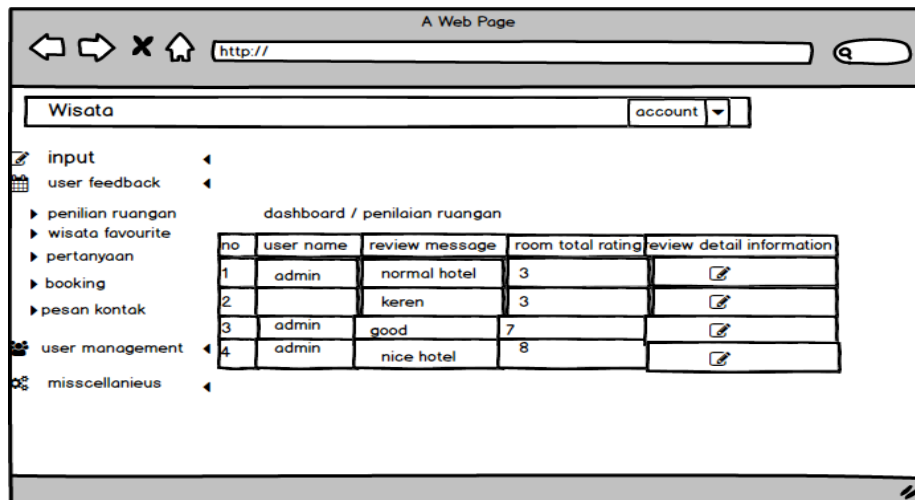
Setelah admin *log in input* username dan password maka akan tampil tampilan dashboard yaitu pilihan menu dimana tugas admin mengelola data dalam web, melakukan inputan data wisata, data negara, fitur wisata, fitur ruangan dan lainnya, selain wisata terdapat inputan negara, serta user feedback, user management.



Gambar 3.9 halaman *home*

b. Halaman tampilan desain userfeedback

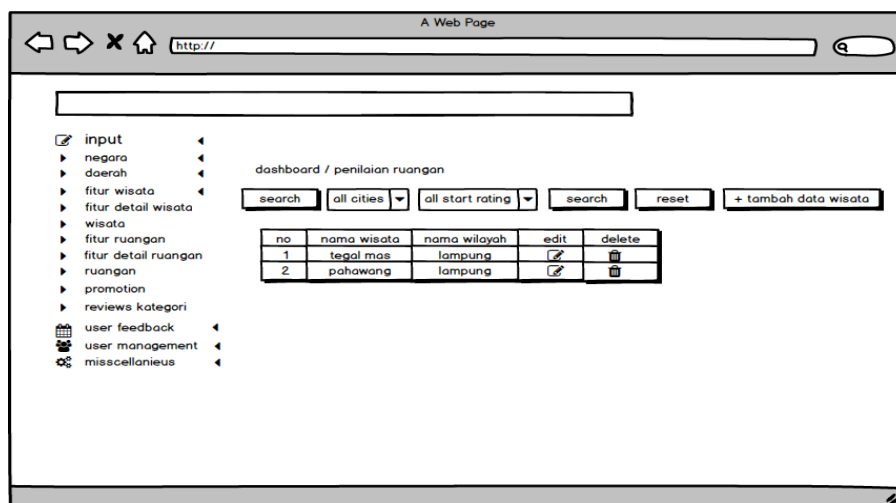
Untuk mengelola reviews pada tampilan tampilan web, melihat jumlah pengunjung, rating dan informasi lainnya terdapat di tampilan user feedback.



Gambar 3.10 halaman *user feedback*

c. Halaman input wisata untu *booking* wisata

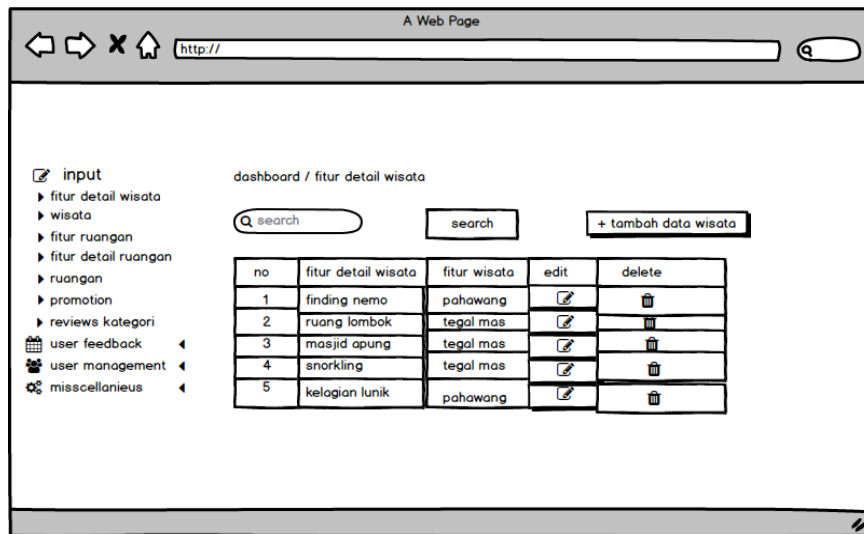
Halaman input wisata pada web dimana admin menginputkan data wisata berupa fasilitas ruangan nama wisata,paket wisata,harga wisata yang mengenai data wisata.



Gambar 3.11 halaman input wisata

d. Halaman fasilitas wisata

Fasilitas wisata terdapat jenis fasilitas yang akan di kunjungi sampai diwisata

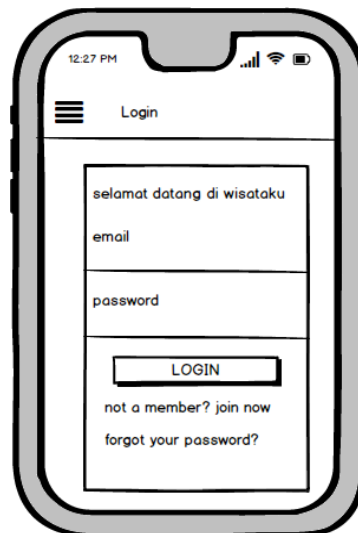


Gambar 3.12 halaman fasilitas input wisata

3.7 Desain Aplikasi

a. tampilan desain menu log in

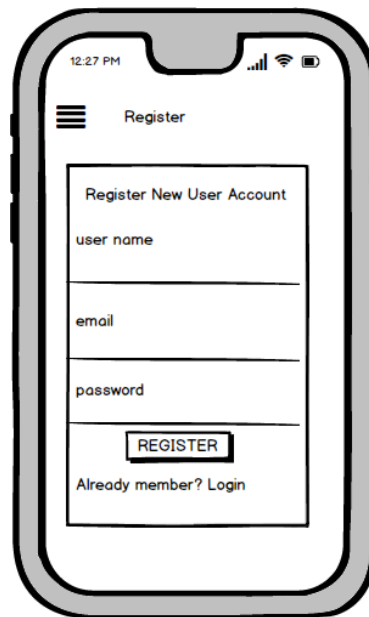
menu *login* pada aplikasi yaitu user harus mengisi email dan password sesuai yang telah di registrasi/ didaftarkan sebelumnya .



Gambar 3.13 *login*

b. tampilan desain registrasi

menu registrasi yaitu untuk masuk ke login jika belum memiliki data email.



Gambar 3.14 halaman registrasi

c. tampilan desain menu *home*

tampilan menu *home* wisata pada aplikasi terdapat cari wisata, recommended wisata, promotion wisata, profile, booking list menu tersebut untuk melihat wisata yang akan di booking, melihat listan hasil akhir booking.

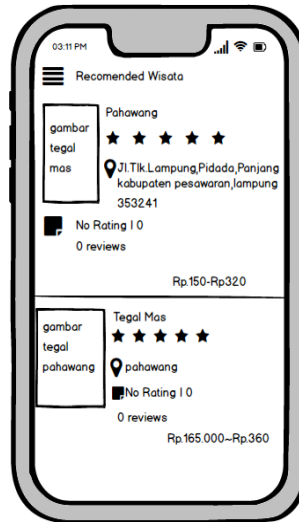


Gambar 3.15 menu *home*

d. tampilan desain recommended wisata

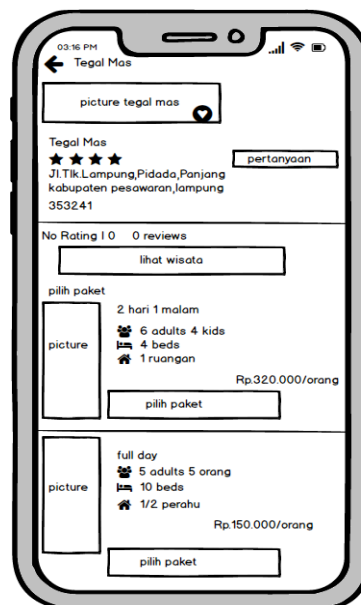
recommended wisata terdapat menu wisata yang akan di *booking* yaitu pahawang dan tegal mas, dalam menu tersebut tertera harga dari yang

terkecil hingga yang terbesar, dan terdapat fasilitas wisata serta ruangan dan paket wisata.

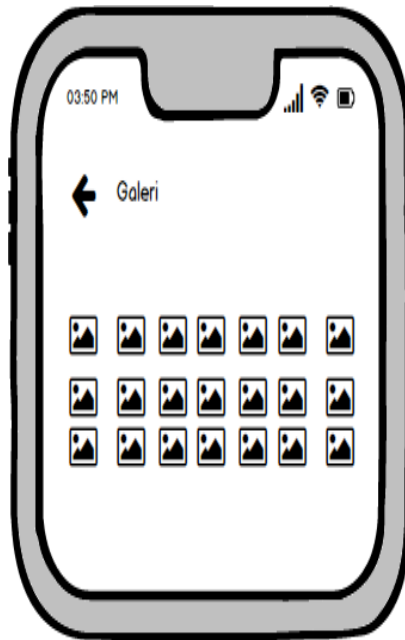


Gambar 3.16 menu recomended wisata

- e. tampilan desain *fasilitas* wisata
tampilan wisata *booking* customer dapat memilih paket sesuai kebutuhan, lalu klik pilih paket lalu mengisi data untuk verifikasi.



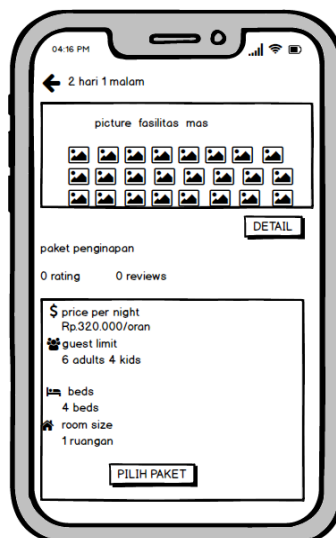
3.17 *fasilitas* wisata (1)



Gambar 3.18 gambar *fasilitas wisata*

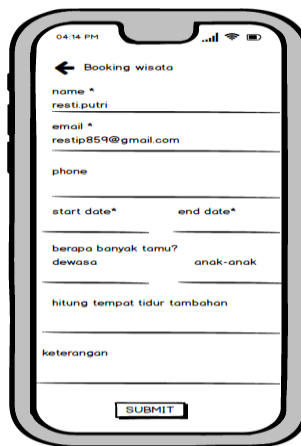
f. tampilan desain *booking* wisata

pada desain fasilitas wisata terdapat fasilitas yang ada di pulau atau wisata yang akan di *booking* seperti fasilitas kamar tidur,kamar mandi,halaman ruangan depan,tampilan kamar,tampilan luar,tempat wisata yang akan dikunjungi setelah tiba di wisata.



Gambar 3.19 *booking* wisata

- g. tampilan desain *input data booking*
pada desain ini menampilkan setelah *booking*/memilih paket langkah selanjutnya ialah input data untu *booking* seperti email,nama,no telephone start tanggal keberangkatan untuk memastikan customer dan memudahkan admin menghubungi pihak terkait.



04:14 PM

← Booking wisata

name *
resti.putri

email *
restip859@gmail.com

phone

start date* end date*

berapa banyak tamu?
dewasa anak-anak

hitung tempat tidur tambahan

keterangan

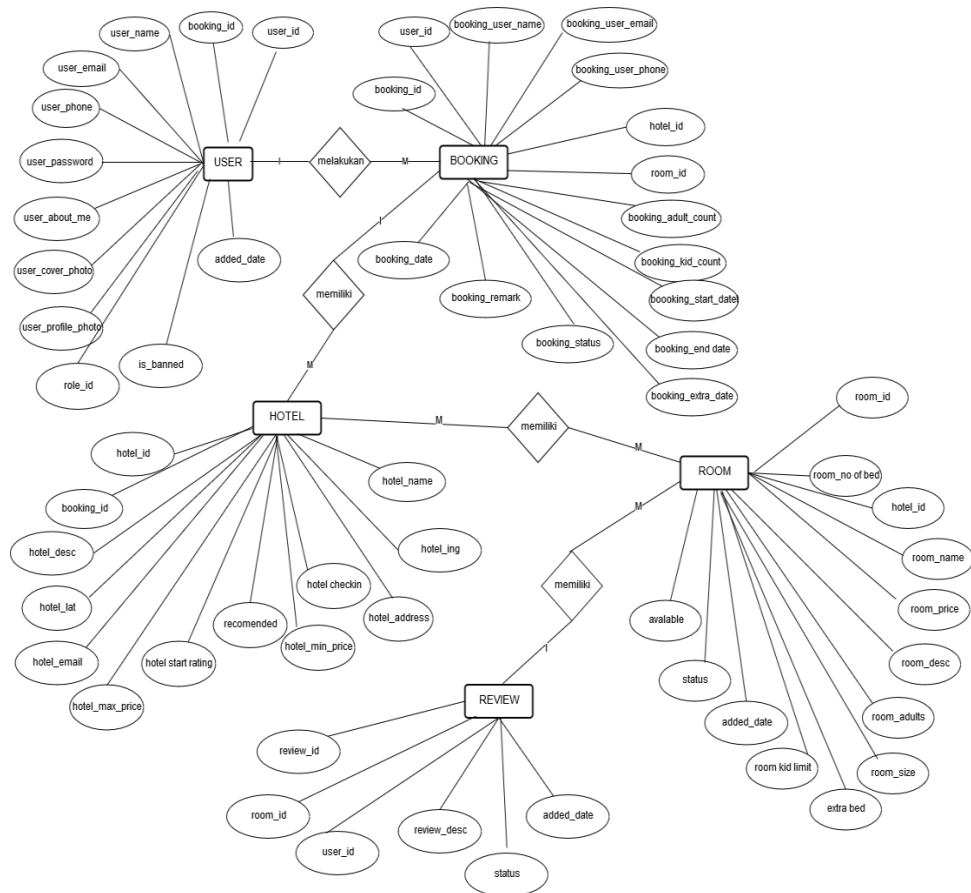
SUBMIT

Gambar 3.20 *input data booking* wisata

3.8 Rancangan *Database*

Rancangan database/basis data merupakan suatu desain terinci yang menjelaskan hubungan antar tabel didalam suatu sistem.

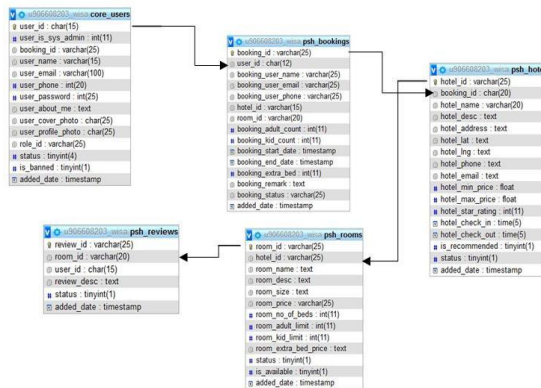
1. Rancangan *Entity Relation Diagram*(ERD)



Gambar 3.21 Entity Relation Diagram

2. Relasi Antar Tabel

Berikut ini adalah relasi antar tabel sistem informasi *digital booking and promotioion* wisata dapat dilihat pada gambar



Gambar 3.22 Relasi Antar Tabel

3. Rancangan Kamus Data

Kamus data merupakan penjabaran dari relasi antar tabel. Di dalam kamus data terdapat penjelasan dari nama-nama *field*, *type data*, *size*, maupun keterangannya.

a. Kamus Data User

Nama *Database* : db_Wisata

Nama *Tabel* : *User*

Primery Key : *User_Id*

Media Penyimpanan : *Hardisk*

No	Nama field	Type data	Size
1	User_id	Char	15
2	user_is_sys_admin	Int	11
3	booking_id	Varchar	25
4	user_name	Varchar	15
5	user_email	varchar	100
6	user_phone	Int	20
7	user_password	int	25
8	user_about_me	text	
9	user_cover_photo	Char	25
10	user_profile_photo	Char	25
`11	role_id	Varchar	25
12	Status	tinyint	4
13	is_banned	Tinyint	1
14	added_date	timestamp	
15	Status	tinyint	

Tabel 3.1 tabel *user*

b. Kamus data *booking*

Nama *Database* : db_Wisata

Nama *Tabel* : *booking*

Primery Key : *booking_id*

Media Penyimpanan : *Hardisk*

No	Nama field	Type data	Size
1	booking_id	Char	20
2	hotel_name	varchar	20
3	hotel_desc	text	
4	hotel_address	text	
5	hotel_lat	text	
6	hotel_lng	text	
7	hotel_phone	text	
8	hotel_email	text	
9	hotel_min_price	float	
10	hotel_max_price	float	
11	hotel_star_rating	Int	11
12	hotel_check_in	time	
13	hotel_check_out	Time	
14	is_recommended	tinyint	
15	Status	tinyint	
16	added_date	timestamp	

Tabel 3.2 tabel *booking*

c. Kamus Data Review

Nama *Database* : *db_Wisata*

Nama *Tabel* : *review*

Primery Key : *review_id*

Media Penyimpanan : *Hardisk*

No	Nama field	Type data	Size
1	review_id	Varchar	25

2	room_id	Varchar	20
3	user_id	Varchar	15
4	review_desc	Text	
5	Status	Tinyint	1
6	added_date	Timestamp	

Tabel 3.3 tabel *review*

d. Kamus data hotel

Nama *Database* : db_Wisata

Nama *Tabel* : *hotel*

Primery Key : *hotel_id*

Media Penyimpanan : *Hardisk*

No	Nama field	Type data	Size
1	hotel_id	varchar	25
2	booking_id	Char	20
3	hotel_name	varchar	20
4	hotel_desc	text	
5	hotel_address	text	
6	hotel_lat	text	
7	hotel_lng	text	
8	hotel_phone	text	
9	hotel_email	text	
10	hotel_min_price	float	
11	hotel_max_price	float	
12	hotel_star_rating	Int	11
13	hotel_check_in	time	5
14	hotel_check_out	time	5
15	is_recommended	Tinyint	
16	Status	tinyint	
17	added_date	timestamp	

Tabel 3.4 tabel *hotel*

e. Kamus data Tabel room

Nama Database : db_Wisata

Nama Tabel : room

Primary Key : room_id

Media Penyimpanan : Hardisk

No	Nama field	Type data	Size
1	room_id	varchar	25
2	hotel_id	varchar	25
3	room_name	text	20
4	room_desc	text	
5	room_size	text	
6	room_price	Varchar	25
7	room_no_of_beds	Int	11
8	room_adult_limit	int(11)	
9	room_kid_limit	int(11)	
10	room_extra_bed_price	text	
11	Status	tinyint(1)	
12	is_available	tinyint(1)	
13	added_date	timestamp	

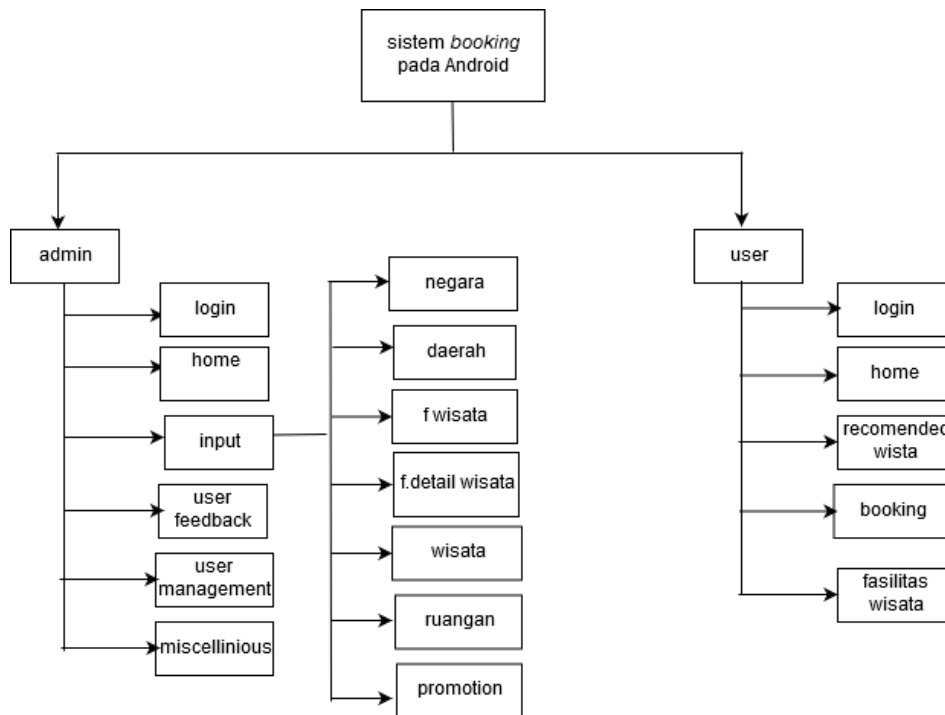
Tabel 3.5 room

3.9 Rancangan logika program

Bagian alir logika program merupakan bagan yang menjelaskan secara rinci Langkah-langkah dari proses program. Bagan alir logika program

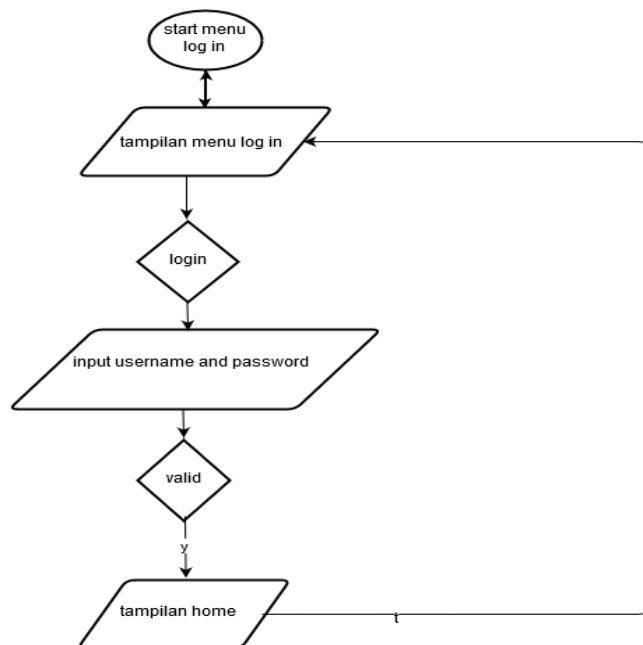
a. hierarchy plus input-proses-output (HIPO)

hierarchy plus input-proses-output (HIPO) merupakan alat dokumentasi program. HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam silus pengembangan sistem. HIPO berbasis pada fungsi,yait tiap-tiap modul di dalam sistem digambarkan oleh fungsi utamanya.



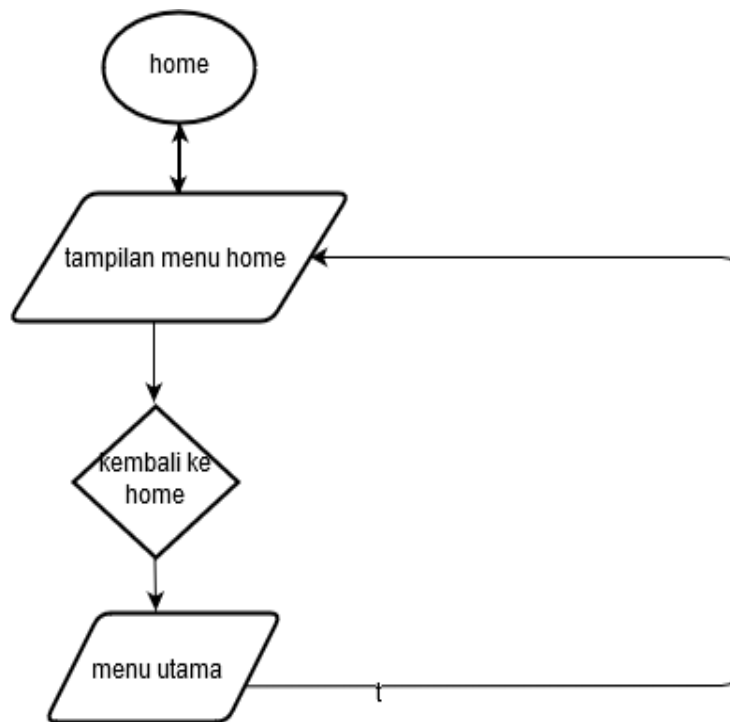
Gambar 3.23 *Hierarchy Plus Input-Proses-Output (HIPO)*

a. logika *login*



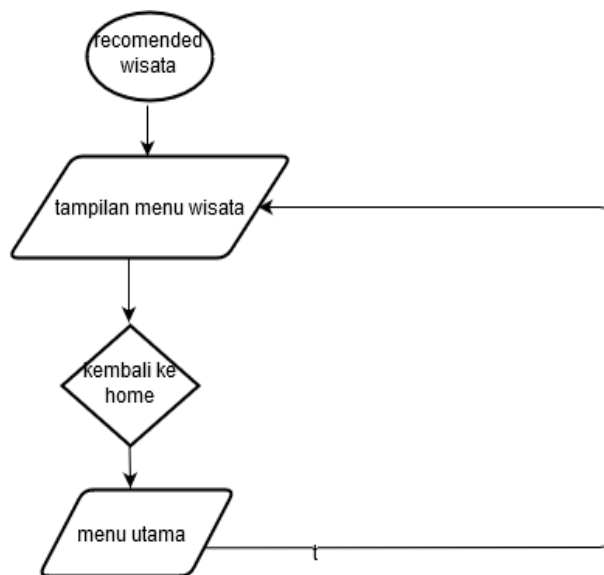
Gambar 3.24 logika *log in*

b. Logika home



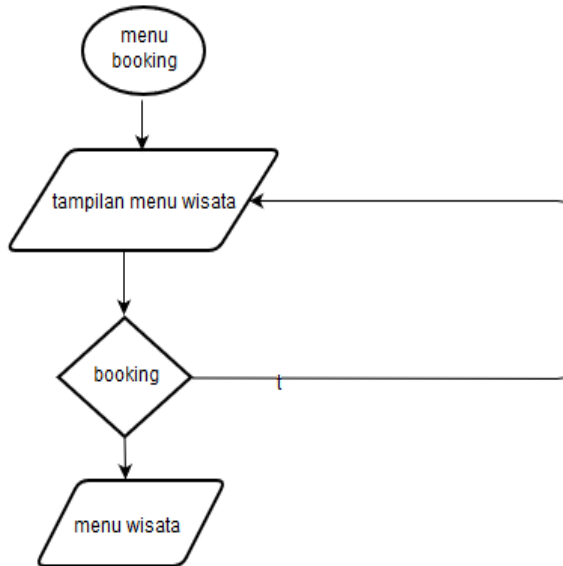
Gambar 3.25 logika *home*

c. Logika recommended wisata



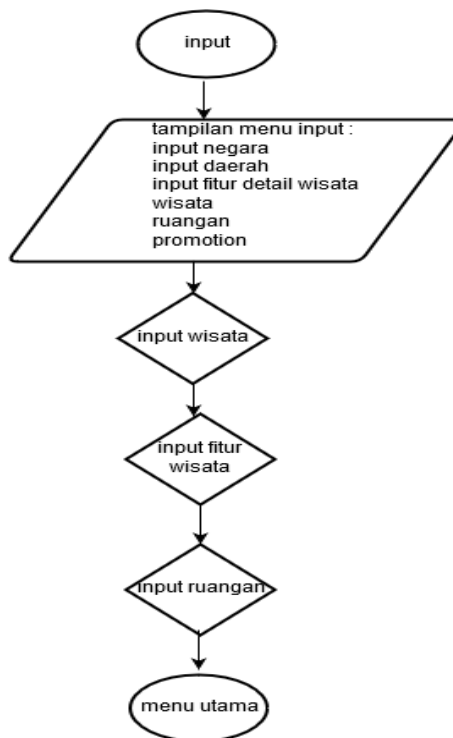
Gambar 3.26 logika *recommended wisata*

d. Logika *booking*

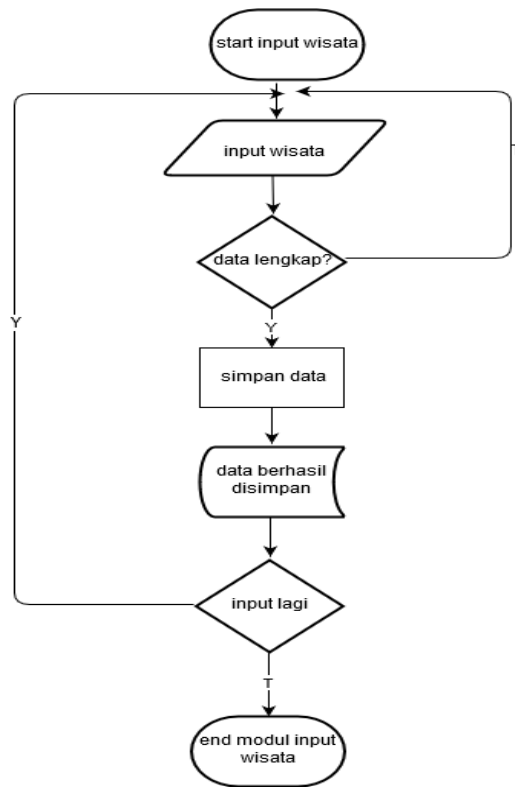


Gambar 3.27 logika *booking*

e. logika input pada admin



Gambar 3.28 logika inpunt admin

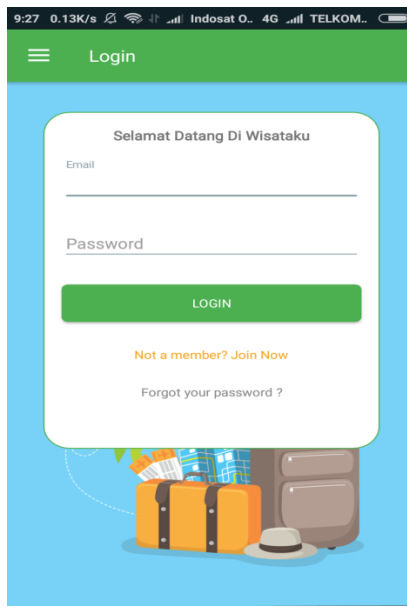


Gambar 3.29 logika simpan data

BAB IV HASIL DAN PEMBAHASAN

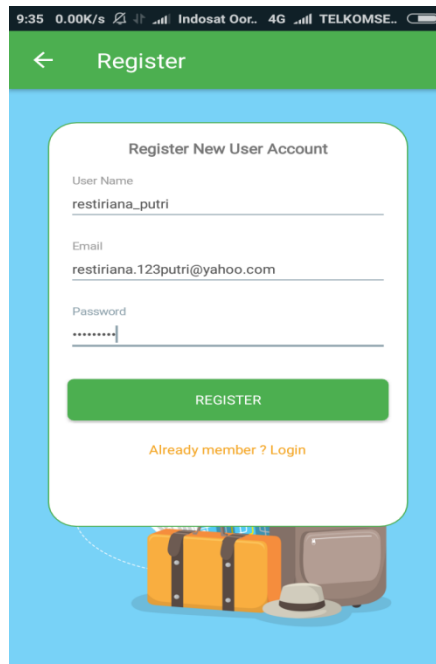
4.1 Hasil Implementasi pada aplikasi

a. Tampilan menu log in pada aplikasi



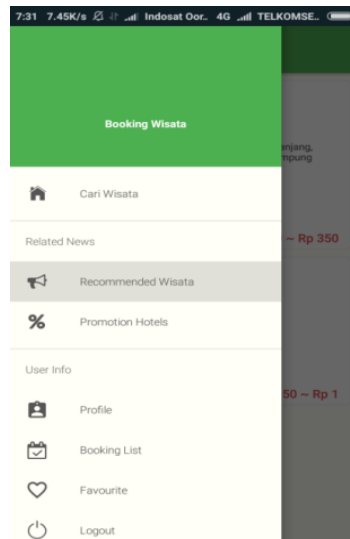
Gambar 4.1 menu *login*

b. Tampilan registrasi user



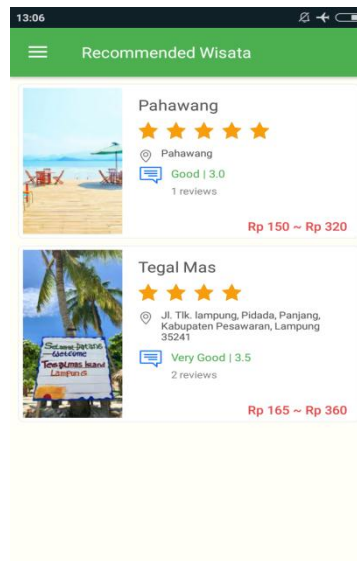
Gambar 4.2 *registrasi user*

c. Tampilan menu *home* pada aplikasi tampilan menu *home* wisata pada aplikasi terdapat cari wisata, recommended wisata, promotion wisata ,profile, booking list menu tersebut untuk melihat wisata yang akan di booking, melihat listan hasil akhir booking.



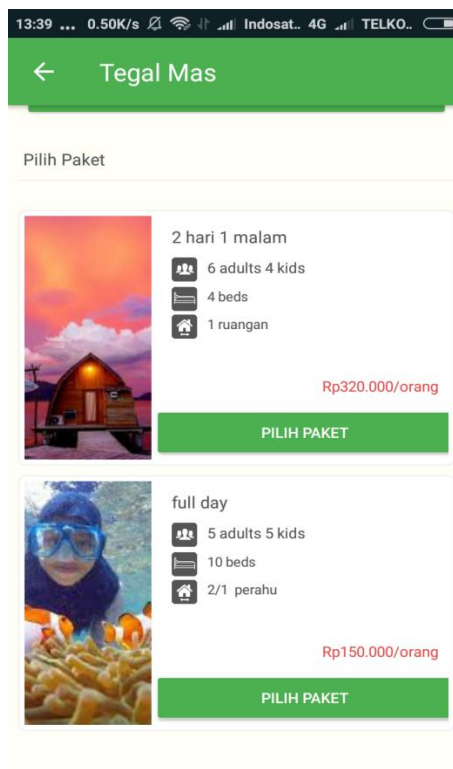
Gambar 4.3 tampilan *home*

d. tampilan recommended wisata recommended wisata terdapat menu wisata yang akan di *booking* yaitu pahawang dan tegal mas, dalam menu tersebut tertera harga dari yang terkecil hingga yang terbesar, dan terdapat fasilitas wisata serta ruangan dan paket wisata.

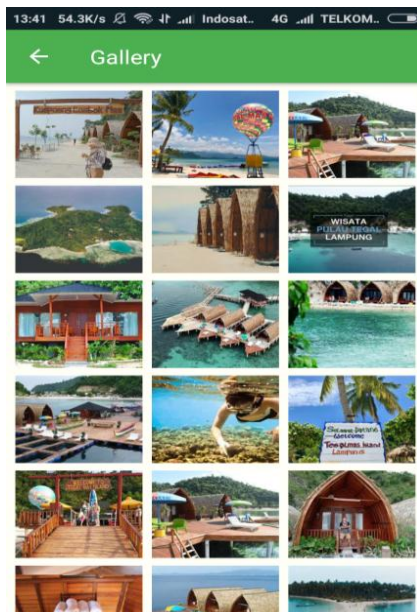


Gambar 4.3 *recommended wisata*

- e. tampilan *fasilitas wisata*
- tampilan *booking wisata* customer dapat memilih paket sesuai kebutuhan, lalu klik pilih paket lalu mengisi data untuk verifikasi.



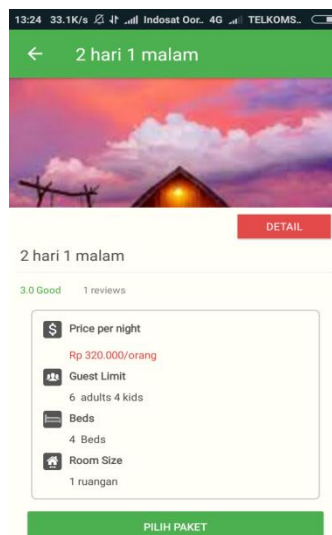
Gambar 4.4 *fasilitas wisata (1)*



Gambar 4.5 gambar *fasilitas* wisata (2)

f. tampilan fasilitas *booking* wisata

pada desain fasilitas wisata terdapat fasilitas yang ada di pulau atau wisata yang akan di *booking* seperti fasilitas kamar tidur,kamar mandi,halaman ruangan depan,tampilan kamar,tampilan luar,tempat wisata yang akan dikunjungi setelah tiba di wisata.



Gambar 4.6 *booking* wisata

- a. tampilan desain *input data booking*
- pada desain ini menampilkan setelah *booking*/memilih paket langkah selanjutnya ialah input data untuk *booking* seperti email, nama, no telephone start tanggal keberangkatan untuk memastikan customer dan memudahkan admin menghubungi pihak terkait.

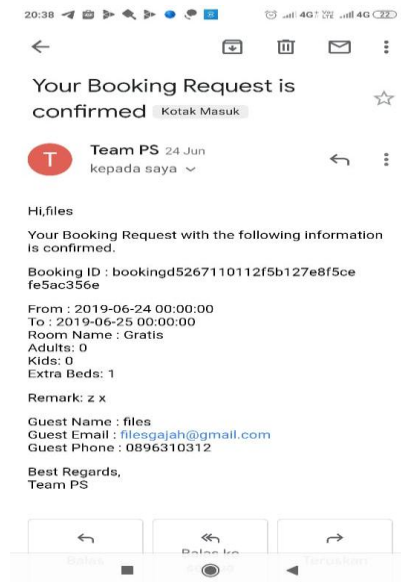
The screenshot shows a mobile application interface for "Booking Wisata". The form contains the following fields and elements:

- Name ***: Input field with the value "resti.putri".
- Email ***: Input field with the value "restip859@gmail.com".
- Phone**: Input field with the placeholder text "Phone".
- Start Date *** and **End Date ***: Two date selection fields.
- Berapa banyak tamu??**: Section header for the number of guests.
- Dewasa** and **Anak-anak**: Two input fields for the number of adults and children, both with the value "0".
- Hitungan tempat tidur tambahan.**: Input field for additional bed calculations, with the value "0".
- Keterangan**: Input field for additional notes, with the placeholder text "Keterangan".
- SUBMIT**: A green button at the bottom of the form.

Gambar 4.7 *input data booking*

g. Refund ticket

Refund ticket ialah hasil akhir setelah booking wisata, sistem akan memverifikasi pembookingan customer setelah semua selesai maka akan ada notification bahwa booking telah di konfirmasi

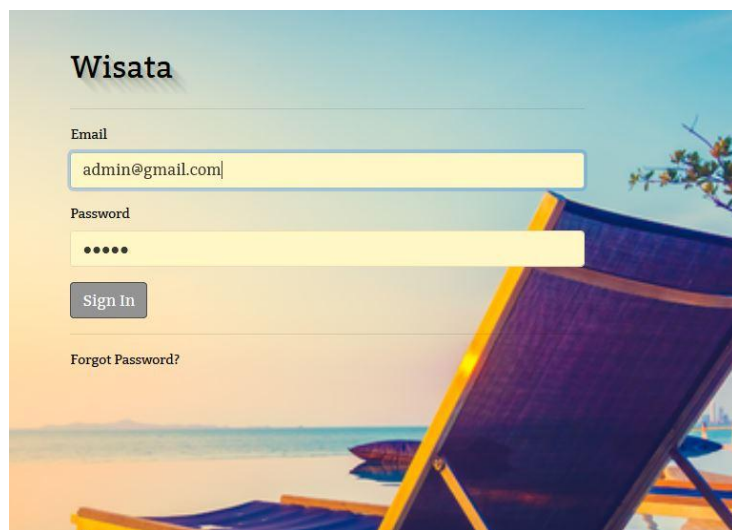


Gambar 4.8 refund ticket

4.2 Hasil implementasi pada Web

a. Tampilan log in admin

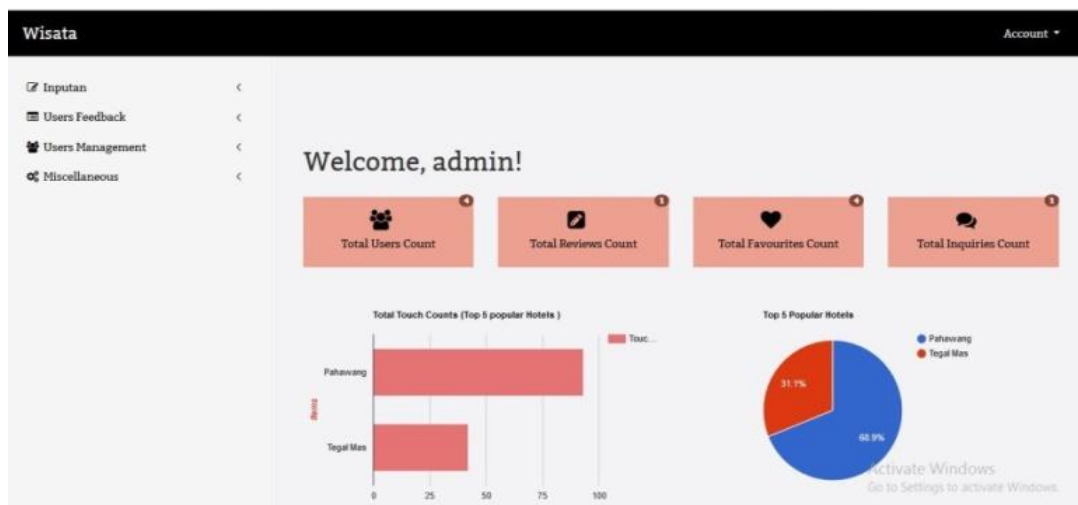
Halaman admin untuk masuk ke halaman selanjutnya ,untuk melakukan inputan data wisata.



Gambar 4.9 halaman log in

b. Tampilan *home* pada web

Setelah admin *log in input* username dan password maka akan tampil tampilan dashboard yaitu pilihan menu dimana tugas admin mengelola data dalam web,melakukan inputan data wisata,data negara,fitur wisata,fitur ruangan dan lainnya,selain wisata terdapat inputan negara,serta user feedback,user management



Gambar 4.10 halaman *home*

c. Tampilan *userfeedback*

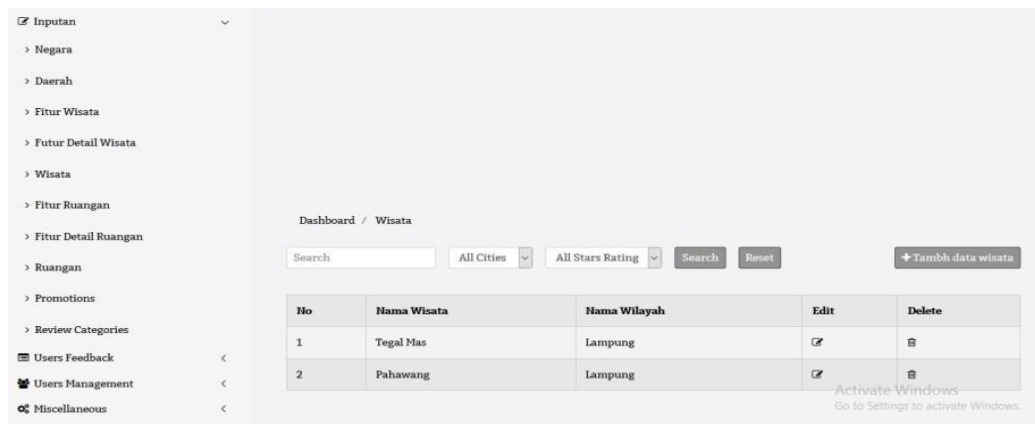
Untuk mengelola reviews pada tampilan tampilan web ,melihat jumlah pengunjung,rating dan informasi lainnya terdapat di tampilan user feedback.

No	User Name	Review Message	Room Total Rating	Review Detail Information
1	admin	Normal Hotel	3	📄
2		keren	3	📄
3	files	jdjdj	5	📄
4	admin	Good	7	📄
5	admin	Nice Condition Hotel	2	📄
6	admin	Nice Location	7	📄
7	admin	Nice Hotel	8	📄

Gambar 4.11 *user feedback*

d. Tampilan input wisata untuk *booking* wisata

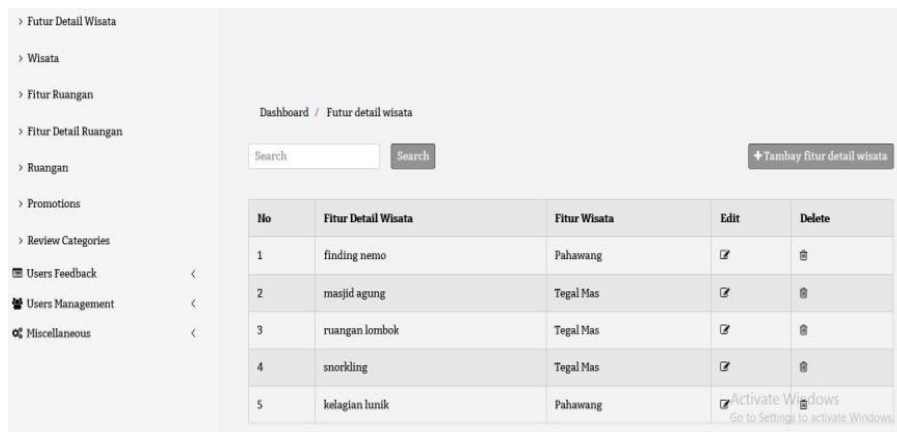
Halaman input wisata pada web dimana admin menginputkan data wisata berupa fasilitas ruangan nama wisata, paket wisata, harga wisata yang mengenai data wisata.



Gambar 4.12 input wisata

e. Tampilan fasilitas wisata

Pada tampilan ini terdapat fasilitas yang akan di kunjungi pada saat keberangkatan perjalanan di pulau wisata yang dikunjungi.



Gambar 4.13 fasilitas wisata



Gambar 4.14 fasilitas wisata

4.3 Kelebihan dan Kelemahan Aplikasi

1. Kelebihan Aplikasi

- a. Aplikasi dapat *booking* wisata dimanapun.
- b. Customer wisata dapat melihat harga yang tertera beserta informasi dan fasilitas yang ada pada aplikasi sehingga dapat menyiapkan biaya tanpa ada biaya tambahan
- c. Terdapat notifikasi jika telah *booking* wisata melalui email
- d. Adanya *Refund* / bukti untuk masuk wisata

2. Kelemahan Aplikasi

- a. Aplikasi ini hanya dapat di instal pada sistem operasi android versi 5.0 (*Lollipop*) keatas.
- b. Aplikasi ini hanya tersedia dua wisata
- c. Harus memiliki koneksi internet

- d. Notifikasi tidak langsung muncul pada layar handphone ataupun web pada admin
- e. Hanya bisa memalui email untuk konfrimasi *booking*

BAB V

KESIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan Penelitian yang sudah di lakukan maka telah di ambil kesimpulan sebagai berikut:

1. Aplikasi Booking Wisata dapat mempermudah transaksi pemesanan antara Customer dan pelayan wisata.
2. Aplikasi Booking wisata memiliki fasilitas yang dapat di lihat oleh pengunjung dan dapat memilih paket mana yang akan di *booking*.

5.2 Saran

Sebagai perbaikan untuk penelitian yang lebih lanjut, maka peneliti memberikan saran diantaranya :

1. Penelitian selanjutnya dapat menambahkan fitur wisata lebih lagi agar pelanggan mengetahui wisata pesawaran bukan sekedar pahawang dan tegal mas.
2. Tata letak yang kurang sedikit rapih.
3. Ditingkatkan kembali sub sub sistem dan metode perangkat lunak

DAFTAR PUSTAKA

Alfiyatullathifah, I. and Husni Thamrin, S.T., 2016. *Sistem Informasi Paket Wisata Sebagai Media Promosi Dan Reservasi Studi Kasus Pada Reyhan Tour* (Doctoral dissertation, Universitas Muhammadiyah Surakarta).

Ardianto, A. and Ardianto, A.E.B.A., Rancang Bangun Aplikasi Pariwisata Malang Berbasis Android. *Bimasakti*.

Eldita, B.R., 2018. Aplikasi Sistem Informasi Geografis Objek Wisata Berbasis Mobile Android Pada Dinas Kepemudaan, Olahraga Dan Pariwisata (Bidang Pariwisata) Kota Metro.

Herlina, H., Yulmaini, Y., Karnila, S. and Yusendra, M.A.E., 2015. Pengembangan Aplikasi E-Tourism Berbasis Android Sebagai Strategi Promosi Pariwisata Provinsi Lampung. *Proceedings Konferensi Nasional Sistem dan Informatika (KNS&I)*.

Maulidin, R.N., 2015. *Sistem Informasi Paket Wisata Pada Cv. Khans Tour Berbasis Web* (Doctoral dissertation, Universitas Muhammadiyah Surakarta).

Nur Hidayati Putri, P.R.I.M.A., 2019. Penerapan Software Balsamiq Untuk Meningkatkan Hasil Belajar Pada Mata Pelajaran Kerja Proyek Siswa Kelas Xii Multimedia Di Smk Negeri 1 Jombang. *Jurnal Mahasiswa Teknologi Pendidikan*, 9(2).

Purnomo, Y.A., 2015. Rancang Bangun Aplikasi Informasi Pariwisata Berbasis Android Dengan Optimasi Biaya. In *Seminar Informatika Aplikatif Polinema*.



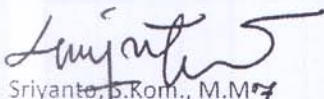
SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.0605/DMJ/DFIK/BAAK/XII-18
Tentang
Dosen Pembimbing Skripsi
Program Studi S1 Sistem Informasi
REKTOR IIB DARMAJAYA

- Memperhatikan :** 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi**.
- Menimbang :** 1. Laporan dan usulan Ketua Program Studi **S1 Sistem Informasi**.
2. Bahwa untuk mengaktifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
- Mengingat :** 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

- Pertama :** Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Sistem Informasi.
- Kedua :** Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga :** Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.
- Keempat :** Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 10 Desember 2018
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer


Sriyanto, S.Kom., M.M.
NIK. 00210800

1. Ketua Jurusan S1 Sistem Informasi
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
 Nomor : SK. 0605/DMJ/DFIK/BAAK/XII-18
 Tanggal : 10 Desember 2018
 Perihal : Pembimbing Penulisan Skripsi
 Program Studi Strata Satu (S1) Sistem Informasi

JUDUL SKRIPSI DAN DOSEN PEMBIMBING
 PROGRAM STUDI STRATA SATU (S1) SISTEM INFORMASI

No	NAMA	NPM	JUDUL	PEMBIMBING
32	Deni Fitri	1611059035p	Rancangan Bangunan Sistem Informasi Pelayanan Kesehatan Pada Klinik Pratama Ardhito Medika	Indera, S.Kom, M.TI
33	Rio Fedrica Maldhan	15110500061	Rancang Bangun Sistem Informasi Monitoring Unit Kegiatan Mahasiswa IIB Darmajaya Berbasis Android	
34	Resti Riana Putri	1511050154	Rancangan Bangun Sistem Informasi <i>E-Ticketing</i> Bagi Para Pengunjung Wisnus Dan Wisma Pada Destinasi Wisata Pesawaran Berbasis Android	
35	Modyta Anggraini	1511050071	Penerapan Data Mining Untuk Menganalisa pola peminjaman Buku Pada Perpustakaan IIB Darmajaya Menggunakan Algoritma Apriori	Melda Agharina, S.Kom, M.TI
36	Ferdhy Apriawan	1611058009p	Sistem Informasi Layanan Terpadu Futsal Pada Club Center Lampung Berbasis Web	
37	Rizky Amalia	1511050027	Sistem Informasi PBPU (Pekerja Bukan Penerima Upah) Berbasis Website dengan Pemanfaatan SMS Gateway Pada BPJS Kesehatan Bandar Lampung	
38	Inas Yuni Tsamarah	1511950084	Sistem Informasi Pada Aiesec Lampung Berbasis Web	Neni Purwanti, S.Kom, M.TI
39	Agnes Dwi Januanti	1511050046	Penerapan Teknik Data Mining Untuk Memprediksi Tingkat Kelulusan Mahasiswa dengan Metode Naive Bayesian di IIB Darmajaya	
40	Rejipah	1611058019p	Sistem Informasi Layanan Umum Terpadu Berbasis Web Pada Kecamatan Labuhan Ratu Kota Bandar Lampung	Nursiyanto, S.Kom, M.TI
41	Putri Nurbaiti	1511050101	Sistem Informasi E-Document Peserta BPJS Kesehatan Bandar Lampung Berbasis Web	
42	Sutono	1511050142	Sistem Informasi CRM (Customer Relationship Management) PT. Sumber Trada Motor (Dealer Kawasaki) Berbasis Android	Nurjoko, S.Kom., M.TI
43	Devi Eliza	1511050018	Aplikasi E-Learning Sekolah Menengah Atas SMA Muhammadiyah Gisting Berbasis Web	
44	Muamar Nabila	1511050011	Sistem Informasi Jasa Customer Service pada Shoes and Care Lampung Berbasis Android	Ochi Marsella F, S.Kom, M.TI
45	Yunita	1511050092	Sistem Informasi Tracer Study Berbasis Mobile Pada Sma Kemala Bhayangkari Kotabumi	
46	Roni Habibi	1511010108	Sistem Informasi Pelayanan E-Customer Service Pada PT Indosat Tbk Berbasis Mobile	
47	I Made Edoar Zenner SY	1511050044	Sistem Informasi Pimpinan Cabang KMHDl Bandar Lampung	Ruki Rizal, S.Kom, M.TI

RESTI RIANANDA PUTRI
151150154 1511050154

Rancang Bangun Sistem Informasi E-ticketing Bagi para Pengunjung Wisata Dan Wisata pada Destinasi Wisata perawakan Berbasis Android.

Tahun	Revisi / Keterangan	Status
		✓
2018	Bimbingan Draft	✓
2018	Bimbingan BAB 2,3	✓
2018	Revisi BAB 1,2,3	U
2018	Revisi awal, dan BAB I	U
2018	Ace sesuai	U
2019	BAB BAB 1, 2, 3 (Bimbingan)	U
2019	Revisi BAB 2,3,4	U
2019	Bimbingan bab, 3,4	U
2019	Bimbingan program	U
2019	Ace sesuai	U

21 November 2018

Nurika, S.Kom M.T.I

Lampiran 1 : Koding

1. Koding Login Activity

```
package id.ac.darmajaya.gosalon;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

import android.support.design.widget.TextInputEditText;

import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.pixplicity.easyprefs.library.Prefs;

import java.util.ArrayList;
import java.util.List;

import id.ac.darmajaya.gosalon.Model.User.DataUser;
```

```
public class LoginActivity extends
AppCompatActivity {

    private TextInputEditText email,
password;

    private ProgressBar progress_bar;

    private FloatingActionButton fab;

    private ProgressDialog pDialog;

    private Context mContext;

    private ResponseLogin dataUsers;

    private List<DataUser> listUsers = new
ArrayList<>();

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

setContentView(R.layout.acitivity_login);

        email = (TextInputEditText)
findViewById(R.id.email);

        password = (TextInputEditText)
findViewById(R.id.password);

        progress_bar = (ProgressBar)
findViewById(R.id.progress_bar);
```



```

        fab = (FloatingActionButton)
findViewById(R.id.fab);

        mContext = this;

        if (Prefs.getInt(SPref.getUserstatus(),
0) == 1) {

            Toasty.success(mContext, "Selamat
Datang " + Prefs.getString(SPref.getNama(),
null), Toast.LENGTH_LONG).show();

            startActivity(new Intent(mContext,
MainActivity.class));

            finish();

        }

        fab.setOnClickListener(new
View.OnClickListener() {

            @Override

            public void onClick(View v) {

                if (validate_login())

                    login();

            }

        });

```

```

        ((TextView)
findViewById(R.id.signup)).setOnClickListener() {

            @Override

            public void onClick(View v) {

```

```

                startActivity(new
Intent(LoginActivity.this,
RegistrationActivity.class));

            }

        });

        public boolean validate_login() {

            return (!Validate.cek(email) &&
!Validate.cek(password)) ? true : false;

        }

        public void login() {

            pDialog = new ProgressDialog(this);

            //

            pDialog.setProgressHelper().setBarColor(C
olor.parseColor("#A5DC86"));

            pDialog.setMessage("Loading");

            pDialog.setCancelable(false);

            // pDialog.setIndeterminate(false);

            pDialog.show();

            Call<ResponseLogin> user =
client.getApi().getresponauth(email.getText(
).toString().trim(),
password.getText().toString().trim());

            user.enqueue(new
Callback<ResponseLogin>() {

```

```

@Override

public void
onResponse(Call<ResponseLogin> call,
Response<ResponseLogin> response) {

    pDialog.hide();

    if (response.isSuccessful()) {

        dataUsers = response.body();

listUsers.addAll(dataUsers.getData());

        setPreference(listUsers.get(0));

        Toasty.success(mContext,
"Selamat Datang " +
listUsers.get(0).getNama(),
Toast.LENGTH_LONG).show();

        finish();

        startActivity(new
Intent(getApplicationContext(),
MainActivity.class));

    } else {

        Toasty.error(mContext,
"Username dan password salah",
Toast.LENGTH_LONG).show();

        System.out.println("data user "
+ response.code());

    }

}

```

```

@Override

public void
onFailure(Call<ResponseLogin> call,
Throwable t) {

    pDialog.hide();

        Toasty.error(mContext, "Koneksi
Tidak ada",
Toast.LENGTH_LONG).show();

        if (pDialog.isShowing())

            pDialog.dismiss();

            System.out.println("data user" +
t.getMessage());

        }

    });

}

private void setPreference(DataUser user)
{

    Prefs.putString(SPref.getId(),
user.getId());

    Prefs.putString(SPref.getPassword(),
user.getPassword());

    Prefs.putString(SPref.getNama(),
user.getNama());

    Prefs.putString(SPref.getAlamat(),
user.getAlamat());

    Prefs.putString(SPref.getEmail(),
user.getEmail());

    Prefs.putString(SPref.getTelp(),
user.getTelp());

```

```

        Prefs.putInt(SPref.getUserstatus(), 1);
    }
}

```

2. Koding Maps

@Override

```

protected void onCreate(@Nullable
Bundle savedInstanceState) {

```

```

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

```

```

    btnmaps =
    findViewById(R.id.btnmap);

```

```

    getLocationPermission();

```

```

    Toasty.info(getApplicationContext(),
"Tentukan Lokasi User",
Toast.LENGTH_SHORT).show();

```

```

}

```

@Override

```

public void onMapReady(GoogleMap
googleMap) {

```

```

    mMap = googleMap;

```

```

    fetchfrominternet();

```

```

    if (mLocationPermissionsGranted) {

```

```

//        getLocation();

```

```

        if
(ActivityCompat.checkSelfPermission(this,

```

```

Manifest.permission.ACCESS_FINE_LOC
ATION)

```

```

!=

```

```

PackageManager.PERMISSION_GRANTED &&

```

```

ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_L
OCATION) !=

```

```

PackageManager.PERMISSION_GRANTED) {

```

```

    return;

```

```

}

```

```

    mMap.setMyLocationEnabled(true);

```

```

    mMap.getUiSettings().setMyLocationButton
Enabled(true);

```

```

}

```

```

        LatLng peta = new LatLng(-5.4219919,
105.2590389);

```

```

    mMap.animateCamera(CameraUpdateFacto
ry.newLatLngZoom(peta, 15f));

```

```

    mMap.setOnMapClickListener(new
GoogleMap.OnMapClickListener() {

```

@Override

```

    public void onMapClick(LatLng
latLng) {

```

```

        marker2.remove();

```

```

        garageLocation = latLng;

```

```

        moveCamera(latLng,
DEFAULT_ZOOM);

```

```

    }
});

    btnmaps.setOnClickListener(new
View.OnClickListener() {

        @Override

        public void onClick(View v) {

            if (garageLocation == null){

                Toasty.info(getApplicationContext(),
                "Tentukan Lokasi User",
                Toast.LENGTH_SHORT).show();

            }else{

                Intent intent = getIntent();

                intent.putExtra("location_lat",
                garageLocation.latitude);

                intent.putExtra("location_lng",
                garageLocation.longitude);

                setResult(Activity.RESULT_OK, intent);

                finish();

            }

        }

    });

}

    private void moveCamera(LatLng latLng,
float zoom) {

```

```

//    Log.d(TAG, "moveCamera: moving
the camera to: lat: " + latLng.latitude + ",
lng: " + latLng.longitude);

    mMap.animateCamera(CameraUpdateFacto
ry.newLatLngZoom(latLng, zoom));

        MarkerOptions options = new
MarkerOptions()

        .icon(BitmapDescriptorFactory.defaultMark
er(BitmapDescriptorFactory.HUE_AZURE)
)

        .position(latLng)

        .title("Lokasi Saya");

        marker2 = mMap.addMarker(options);

    }

    private void fetchfrominternet() {

        Call<ResponseToko> user =
client.getApi().getdatatoko();

        user.enqueue(new
Callback<ResponseToko>() {

            @Override

            public void
onResponse(Call<ResponseToko> call,
Response<ResponseToko> response) {

                if (response.isSuccessful()) {

                    responseToko =
response.body();

```

```
dataTokos.addAll(responseToko.getData());
```

```
    for(int i = 0; i <
dataTokos.size(); i++){
```

```
        String[] koordinat =
dataTokos.get(i).getKordinat().split(",");
```

```
        LatLng latLng = new
LatLng(Double.parseDouble(koordinat[0]),
Double.parseDouble(koordinat[1]));
```

```
        System.out.println("kordinat
" + koordinat[0] + "," + koordinat[1]);
```

```
        MarkerOptions options =
new MarkerOptions()
```

```
.icon(BitmapDescriptorFactory.defaultMark
er(BitmapDescriptorFactory.HUE_RED))
```

```
        .position(latLng)
```

```
.title(dataTokos.get(i).getNama_toko());
```

```
        marker2 =
mMap.addMarker(options);
```

```
    }
```

```
    } else {
```

```
        Toasty.error(getApplicationContext(), "Data
Tidak ditemukan",
Toast.LENGTH_LONG).show();
```

```
        System.out.println("data user "
+ response.code());
```

```
    }
```

```
}
```

```
@Override
```

```
public void
onFailure(Call<ResponseToko> call,
Throwable t) {
```

```
    Toasty.error(getApplicationContext(),
"Koneksi Tidak ada",
Toast.LENGTH_LONG).show();
```

```
        if (pDialog.isShowing())
```

```
            pDialog.dismiss();
```

```
            System.out.println("data user" +
t.getMessage());
```

```
        }
```

```
    });
```

```
}
```

```
private void initMap() {
```

```
        SupportMapFragment mapFragment =
(SupportMapFragment)
getSupportFragmentManager().findFragmen
tById(R.id.map);
```

```
        mapFragment.getMapAsync(MapsActivity.t
his);
```

```
    }
```

```
private void getLocationPermission() {
```

```

String[] permissions =
{Manifest.permission.ACCESS_FINE_LOCATION,

Manifest.permission.ACCESS_COARSE_LOCATION};

if
(ContextCompat.checkSelfPermission(this,
getApplicationContext(),

    FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

    if
(ContextCompat.checkSelfPermission(this,
getApplicationContext(),

        COURSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

        mLocationPermissionsGranted =
true;

        initMap();

    } else {

ActivityCompat.requestPermissions(this,

        permissions,

LOCATION_PERMISSION_REQUEST_CODE);

    }

} else {

```

```

ActivityCompat.requestPermissions(this,

        permissions,

LOCATION_PERMISSION_REQUEST_CODE);

    }

}

private void getDeviceLocation() {

    mFusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);

    try {

        if (mLocationPermissionsGranted) {

            final Task location =
mFusedLocationProviderClient.getLastLocation();

            Task completeListener =
location.addOnCompleteListener(new
OnCompleteListener() {

                @Override

                public void
onComplete(@NonNull Task task) {

                    if (task.isSuccessful()) {

```

```

        Location currentLocation
= (Location) task.getResult();

mMap.animateCamera(CameraUpdateFacto
ry.newLatLngZoom(new
LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude()),
DEFAULT_ZOOM));

        MarkerOptions options =
new MarkerOptions()

        .icon(BitmapDescriptorFactory.defaultMark
er(BitmapDescriptorFactory.HUE_RED))

        .position(new
LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude()))

        .title("Lokasi Saya");

mMap.addMarker(options);

        } else {

Toast.makeText(MapsActivity.this, "unable
to get current location",
Toast.LENGTH_SHORT).show();

        }

});

}

} catch (SecurityException e) {

```

```

        Log.e(TAG, "getDeviceLocation:
SecurityException: " + e.getMessage());

        }

}

private void hideSoftKeyboard() {

this.getWindow().setSoftInputMode(Windo
wManager.LayoutParams.SOFT_INPUT_S
TATE_ALWAYS_HIDDEN);

}

}

```

3.Koding Transaksi

```

@Override

protected void onCreate(@Nullable
Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_transaksi)
;

        combobox = (CheckBox)
findViewById(R.id.combobox);

        nama = (EditText)
findViewById(R.id.nama);

        alamat = (EditText)
findViewById(R.id.alamat);

        notelp = (EditText)
findViewById(R.id.notelp);

        harga = (EditText)
findViewById(R.id.harga);

```

```

        edDate = (EditText)
findViewById(R.id.date);

        koordinat = (EditText)
findViewById(R.id.koordinat);

        edClock = (EditText)
findViewById(R.id.clock);

        btntranskasi = (Button)
findViewById(R.id.btntransaksi);

        mContext = this;

        mySharedPreferences = new
MySharedPreferences(this);

        Locale localeID = new Locale("in",
"ID");

        final NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(localeI
D);

        harga.setText((formatRupiah.format(Integer.
parseInt(String.valueOf(mySharedPreferenc
e.retrieveTotalHarga()))));

        harga.setEnabled(false);

        combobox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

            @Override

```

```

        public void
onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {

            if (combobox.isChecked()) {

                nama.setText(Prefs.getString(SPref.getNam
a(), null));

                alamat.setText(Prefs.getString(SPref.getAla
mat(), null));

                notelp.setText(Prefs.getString(SPref.getTelp
(), null));

            } else {

                nama.setText("");
                alamat.setText("");
                notelp.setText("");

            }

        }

    });

    setDateTextField();

    setwaktu();

    dismissKeyboard(edDate, this);

    edDate.setFocusable(false);

    edDate.clearFocus();

    dismissKeyboard(edClock, this);

    @Override

```



```

        public void onClick(View v) {
            mDatePickerDialog.show();

        }
    });

    edClock.setOnClickListener(new
    View.OnClickListener() {

        @Override

        public void onClick(View v) {

            timePickerDialog.show();

        }

    });

    btntransaksi.setOnClickListener(new
    View.OnClickListener() {

        @Override

        public void onClick(View v) {

            if (validate_login())

                posttransaksi();

        }

    });

    koordinat.setOnClickListener(new
    View.OnClickListener() {

        @Override

        public void onClick(View v) {

```

```

            Intent intent = new
            Intent(getApplicationContext(),
            MapsActivity.class);

                startActivityForResult(intent,
            MAP);

            }

        });

    }

    public void posttransaksi() {

        pDialog = new ProgressDialog(this);

        //

        pDialog.getProgressHelper().setBarColor(C
        olor.parseColor("#A5DC86"));

        pDialog.setMessage("Loading");

        pDialog.setCancelable(false);

        // pDialog.setIndeterminate(false);

        pDialog.show();

        String tanggaldate;

        tanggaldate =

        edDate.getText().toString() + " " +

        edClock.getText().toString();

        final MySharedPreferences mShared =

        new

        MySharedPreferences(TransaksiActivity.this)

        ;

```

```

        GsonBuilder builder = new
GsonBuilder();

        Gson gson = builder.create();

        DataProduk[] addCartProducts =
gson.fromJson(mShared.retrieveProductFromCart(), DataProduk[].class);

        final List<DataProduk> productList =
convertObjectArrayToListObject(addCartProducts);

        final List<ListProduk> list = new
ArrayList<>();

        for (int i = 0; i < productList.size();
i++) {

            list.add(new
ListProduk(productList.get(i).getId()));

        }

        long millis = new Date().getTime();

        PostTransaksi dataTransaksi = new
PostTransaksi(

            String.valueOf(millis).substring(0,
8),

            Prefs.getString(SPref.getId(),
null),

            productList.get(0).getId_toko(),

            list,

```

```

            null,

            nama.getText().toString(),

            alamat.getText().toString(),

            notelp.getText().toString(),

            koordinat.getText().toString(),

            tanggaldate,

            "0"

        );

        Call<ResponTransaksi> user =
client.getApi().posttransaksi(Prefs.getString(
SPref.getEmail(), null),
Prefs.getString(SPref.getPassword(), null),
dataTransaksi);

        user.enqueue(new
Callback<ResponTransaksi>() {

            @Override

            public void
onResponse(Call<ResponTransaksi> call,
Response<ResponTransaksi> response) {

                pDialog.hide();

                if (response.isSuccessful()) {

                    System.out.println("Transaksi
Sukses " + response.code());

                    Toasty.success(mContext,
"Transaksi Berhasil",
Toast.LENGTH_LONG).show();

                    startActivity(new
Intent(TransaksiActivity.this,
MainActivity.class));

                    mShared.deletedata();

```

```

        finish();

    } else {

        Toasty.error(mContext,
"Koneksi Tidak ada",
Toast.LENGTH_LONG).show();

        System.out.println("data user "
+ response.code());

    }

}

@Override

public void
onFailure(Call<ResponTransaksi> call,
Throwable t) {

    pDialog.hide();

    Toasty.error(mContext, "Koneksi
Tidak ada",
Toast.LENGTH_LONG).show();

    if (pDialog.isShowing())

        pDialog.dismiss();

    System.out.println("data user" +
t.getMessage());

}

});

}

```

```

private List<DataProduk>
convertObjectArrayToListObject(DataProdu
k[] allProducts) {

    List<DataProduk> mProduct = new
ArrayList<DataProduk>();

    Collections.addAll(mProduct,
allProducts);

    return mProduct;

}

private boolean validate_login() {

}

private void setwaktu() {

    calendar = Calendar.getInstance();

    currentHour =
calendar.get(Calendar.HOUR_OF_DAY);

    currentMinute =
calendar.get(Calendar.MINUTE);

    timePickerDialog = new
TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {

        @Override

        public void onTimeSet(TimePicker
timePicker, int hourOfDay, int minutes) {

/*            if (hourOfDay >= 12) {

                amPm = "PM";

            } else {

```

```

        amPm = "AM";
    }*/
}

edClock.setText(String.format("%02d:%02d",
hourOfDay, minutes)/* + amPm*/);
}

}, currentHour, currentMinute, false);
}

private void setDateTextField() {

    Calendar newCalendar =
Calendar.getInstance();

    mDatePickerDialog = new
DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {

        public void onDateSet(DatePicker
view, int year, int monthOfYear, int
dayOfMonth) {

            Calendar newDate =
Calendar.getInstance();

            newDate.set(year, monthOfYear,
dayOfMonth);

            SimpleDateFormat sd = new
SimpleDateFormat("yyyy-MM-dd");

            final Date startDate =
newDate.getTime();

            String fdate =
sd.format(startDate);

            edDate.setText(fdate);

```

```

    }

    }, newCalendar.get(Calendar.YEAR),
newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MON
TH));

//
mDatePickerDialog.getDatePicker().setMax
Date(System.currentTimeMillis());

mDatePickerDialog.getDatePicker().setMin
Date(System.currentTimeMillis());

}

@Override

public void onActivityResult(int
requestCode, int resultCode, Intent data) {

    super.onActivityResult(requestCode,
resultCode, data);

    if (requestCode == MAP) {

        if (resultCode ==
Activity.RESULT_OK) {

            double lat = (double)
data.getDoubleExtra("location_lat", 0);

            double lng = (double)
data.getDoubleExtra("location_lng", 0);

            System.out.println("mantap soul "
+ lat + " " + lng);

            koordinat.setText(String.valueOf(lat) + ", "
+ String.valueOf(lng));

        }

```

```
    }  
  }  
}
```

4. Koding Loading Karyawan

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout  
xmlns:android="http://schemas.android.com  
/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk  
/res-auto"
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

```
<ImageView
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:scaleType="fitStart"  
    android:src="@drawable/loginbg"  
/>
```

```
<View
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"
```

```
    android:background="@color/pink_500_ov  
erlay" />
```

```
<LinearLayout
```

```
    android:layout_width="330dp"  
    android:layout_height="match_parent"  
    android:layout_centerInParent="true"  
    android:orientation="vertical"
```

```
    android:padding="@dimen/spacing_large">
```

```
<View
```

```
    android:layout_width="0dp"  
    android:layout_height="50dp" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:layout_marginTop="@dimen/spaci  
ng_medium"  
    android:text="Admin Salon"
```

```
android:textAppearance="@style/Base.Text  
Appearance.AppCompat.Display2"
```

```
android:textColor="@android:color/white"  
/>
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_gravity="center_horizontal"
```

```
android:layout_marginTop="@dimen/spaci  
ng_medium"
```

```
android:fontFamily="sans-serif-  
light"
```

```
android:text="Hope you enjoy the  
stay"
```

```
android:textAppearance="@style/Base.Text  
Appearance.AppCompat.Large"
```

```
android:textColor="@android:color/white"  
/>
```

```
<View
```

```
android:layout_width="0dp"
```

```
android:layout_height="0dp"
```

```
android:layout_weight="1" />
```

```
<View
```

```
android:layout_width="50dp"
```

```
android:layout_height="3dp"
```

```
android:layout_gravity="center_horizontal"
```

```
android:background="@android:color/white"  
/>
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_gravity="center_horizontal"
```

```
android:layout_marginTop="@dimen/spaci  
ng_middle"
```

```
android:text="Log In"
```

```
android:textAppearance="@style/Base.Text  
Appearance.AppCompat.Medium"
```

```
android:textColor="@android:color/white"
```

```
android:textStyle="bold" />
```

```
<View
```

```
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_weight="1" />

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/EditText.White.Hint">

    <android.support.design.widget.TextInputEditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="-4dp"
        android:layout_marginRight="-4dp"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:maxLength="50"
        android:maxLines="1"
```

```
        android:singleLine="true"
        android:textColor="@color/overlay_light_90"
        android:theme="@style/EditText.White" />

</android.support.design.widget.TextInputLayout>

    <View
        android:layout_width="0dp"
        android:layout_height="@dimen/spacing_medium_large" />

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/EditText.White.Hint">

        <android.support.design.widget.TextInputEditText
            android:id="@+id/password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="-
4dp"
        android:layout_marginRight="-
4dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:maxLength="50"
        android:maxLines="1"
        android:singleLine="true"
        android:textColor="@color/overlay_light_9
0"
        android:theme="@style/EditText.White" />
</android.support.design.widget.TextInputL
ayout>
```

```
<View
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="@dimen/spacing_m
xlarge" />
```

```
<RelativeLayout
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:padding="@dimen/spacing_small"
```

```
        android:layout_gravity="center_horizontal"
>
```

```
<android.support.design.widget.FloatingAct
ionButton
```

```
    android:id="@+id/fab"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:clickable="true"
```

```
    android:textStyle="bold"
```

```
    android:src="@drawable/login"
```

```
    app:fabSize="normal" />
```

```
<ProgressBar
```

```
    android:id="@+id/progress_bar"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_centerInParent="true"
```

```
    android:theme="@style/ProgressBar.White"
```

```
    android:visibility="gone" />
```



```

</RelativeLayout>
<View
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_weight="1" />
<LinearLayout
    android:id="@+id/sign_up_for_account"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/spacing_medium"
    android:gravity="center"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/signup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Belum punya
akun?"
        android:textAppearance="@style/Base.Text
Appearance.AppCompat.Subhead"

```

```

    android:textColor="@color/overlay_light_90" />
</LinearLayout>
<View
    android:layout_width="0dp"
    android:layout_height="@dimen/spacing_medium" />
</LinearLayout>
</RelativeLayout>

```

5. Koding Karyawan Terima Pesanan

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_ho
rizontal_margin"

```

```
android:paddingTop="@dimen/activity_ventical_margin"
```

```
    android:orientation="vertical"
```

```
android:background="@color/colorPrimaryLight">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:layout_weight="9">
```

```
<android.support.v7.widget.RecyclerView
```

```
    android:id="@+id/recyclerview"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:scrollbars="none"
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:orientation="vertical"
```

```
    android:gravity="bottom">
```

```
<TextView
```

```
    android:id="@+id/totalharga"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
    android:textSize="20sp"
```

```
    android:textStyle="bold"
```

```
    android:text="Total Pembayaran"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```