**PERANCANGAN APLIKASI *E*-*RENT (ELECTRONIC RENTAL)***

**BERBASIS ANDROID**

**TUGAS AKHIR**

*Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana Pendidikan (S1) Pada Jurusan Teknik Elektronika Program Studi Pendidikan Teknik Informatika Universitas Negeri Padang*

****

**HARIS ELFIAN**

**NIM. 16076034/2016**

**PROGRAM STUDI S1 PENDIDIKAN TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2021**

**HALAMAN PERSETUJUAN TUGAS AKHIR**

**PERANCANGAN APLIKASI *E*-*RENT (ELECTRONIC RENTAL)***

**BERBASIS ANDROID**

Nama : Haris Elfian

NIM/TM : 16076034/2016

Program Studi : Pendidikan Teknik Informatika

Jurusan : Teknik Elektronika

Fakultas : Teknik

Padang, Februari 2021

Disetujui Oleh :

Dosen Pembimbing

Drs. Denny Kurniadi, M.Kom.

NIP.19630606 198903 1 001

Mengetahui,

Ketua Jurusan Teknik Elektronika

FT-UNP

Thamrin, S.Pd., MT

NIP. 19770101 200812 1 001

**PENGESAHAN TIM PENGUJI**

**Nama : Haris Elfian**

**NIM/TM : 16076034/2016**

**Dinyatakan Lulus setelah mempertahankan Tugas Akhir di depan Tim Penguji Program Studi Pendidikan Teknik Informatika**

**Jurusan Teknik Elektronika Fakultas Teknik**

**Universitas Negeri Padang**

**Dengan judul**

**PERANCANGAN APLIKASI *E*-*RENT (ELECTRONIC RENTAL)***

**BERBASIS ANDROID**

Padang, Februari 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tim Penguji | | | | Tanda Tangan |
| 1. | Ketua | : | Drs. Denny Kurniadi, M.Kom. |  |
| 2. | Anggota | : | Dr. Dedy Irfan, S.Pd., M.Kom. |  |
| 3. | Anggota | : | Geovanne Farell, S.Pd., M.Pd.T. |  |

# ABSTRAK

**Haris Elfian: Perancangan Aplikasi *E-Rent (Electronic Rental)* Berbasis Android**

Perkembangan globalisasi informasi yang didukung oleh kemajuan teknologi kini telah lumrah di masyarakat. Salah satu hasil dari kemajuan teknologi adalah lahirnya berbagai macam alat elektronik yang mampu membantu pekerjaan manusia seperti komputer, laptop, *smartphone, smartwatch,* dll. Alat elektronik ini digunakan sebagai media untuk bertukar informasi yang didukung dengan internet sebagai sarana penghubung. *E-Rent* (*Electronic Rental)* merupakan aplikasi yang memberikan layanan secara elektronik mengenai informasi penyewaan (*rent)*. Pada sistem yang sedang berjalan saat ini, masyarakat yang ingin menyewa mendapatkan informasi mengenai barang atau jasa sewa tidak mendetail karena tersebarnya sumber informasi. Dengan tersebarnya informasi ini membuat masyarakat yang ingin menyewa barang atau jasa membutuhkan waktu sedikit lama untuk menemukan barang yang dibutuhkan. Aplikasi *E-Rent (Electronic Rental)* ini dibuat menggunakan bahasa pemrograman Java dan menggunakan *CodeIgniter framework*, serta menggunakan MYSQL untuk *database*. Perancangan aplikasi *E-Rent* ini bertujuan untuk menghasilkan sebuah aplikasi berbasis web yang dapat memberikan informasi barang atau jasa, lokasi, harga, dan stok barang atau jasa yang bisa disewa sehingga memudahkan masyarakat yang ingin menyewa barang menemukan barang atau jasa sesuai kebutuhan. Aplikasi ini memberikan fitur pemberian *testimony* dan *rating* atau penilaian terhadap barang dan jasa yang sudah selesai disewa untuk mengapresiasi dan juga menjadi pembanding bagi masyarakat yang ingin memilih barang untuk disewa.

Kata Kunci: *E-Rent*, Rental, Sewa, Android, *Framework CodeIgniter*

# KATA PENGANTAR

Alhamdulillaahirobbil’aalamiin. Puji dan syukur penulis ucapkan atas kehadirat Allah SWT yang telah melimpahkan rahmat, karunia, beserta hidayah-Nya kepada penulis sehingga penulis mampu menyelesaikan Tugas Akhir penulis yang berjudul **“Perancangan Aplikasi *E-Rent (Electronic Rental)* Berbasis Android”**. Tugas Akhir ini dibuat sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Pendidikan Strata Satu (S1) di Program Studi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Padang.

Penyelesaian Tugas Akhir ini tidak terlepas dari bimbingan, bantuan, serta dukungan berbagai pihak. Oleh karena itu pada kesempatan kali ini, penulis mengucapkan terima kasih kepada:

1. Bapak Drs. Denny Kurniadi, M.Kom., selaku dosen pembimbing dan dosen penasehat akademik yang telah membimbing dalam perencanaan, pelaksanaan dan pelaporan Tugas Akhir ini.
2. Bapak Dr. Dedy Irfan, S.Pd., M.Kom. dan Bapak Geovanne Farell, S.Pd., M.Pd.T., selaku dosen penguji yang akan banyak meluangkan waktu untuk memberikan arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
3. Bapak Khairi Budayawan, S.Pd., M.Kom., selaku Ketua Program Studi Pendidikan Teknik Informatika dan Komputer Fakultas Teknik Universitas Negeri Padang.
4. Bapak Thamrin, S.Pd., M.T., selaku Ketua Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
5. Bapak Dr. Fahmi Rizal, M.Pd. M.T, selaku Dekan Fakultas Teknik Universitas Negeri Padang.
6. Bapak dan Ibu dosen Pendidikan Teknik Informatika, Teknisi dan Pegawai Jurusan Teknik Elektronika Universitas Negeri Padang.
7. Orang Tua tercinta yang senantiasa memberikan dukungan, kasih sayang, dan doa yang tiada hentinya kepada penulis.
8. Teman-teman Pendidikan Teknik Informatika 2016 yang telah memberikan motivasi selama mengerjakan Tugas Akhir ini.
9. Semua pihak yang telah membantu dalam menyelesaikan Tugas Akhir ini.

Dalam penyusunan dan penulisan Tugas Akhir ini penulis menyadari masih banyak kekurangan. Untuk itu penulis mengharapkan kritik dan saran yang nantinya dapat menyempurnakan Tugas Akhir ini. Akhir kata, dengan niat yang tulus penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penulis, semoga Allah SWT memberikan balasan setimpal.

Padang, Februari 2021

Penulis

# DAFTAR ISI

**HALAMAN PERSETUJUAN TUGAS AKHIR**

**HALAMAN PENGESAHAN TUGAS AKHIR**

**SURAT PERNYATAAN**

[ABSTRAK i](#_Toc64241114)

[KATA PENGANTAR ii](#_Toc64241115)

[DAFTAR ISI iv](#_Toc64241116)

[DAFTAR GAMBAR vii](#_Toc64241117)

[DAFTAR TABEL ix](#_Toc64241118)

[BAB I PENDAHULUAN 1](#_Toc64241119)

[A. Latar Belakang Masalah 1](#_Toc64241120)

[B. Identifikasi Masalah 5](#_Toc64241121)

[C. Batasan Masalah 6](#_Toc64241122)

[D. Rumusan Masalah 7](#_Toc64241123)

[E. Tujuan Tugas Akhir 8](#_Toc64241124)

[F. Manfaat Tugas Akhir 9](#_Toc64241125)

[BAB II LANDASAN TEORI 10](#_Toc64241126)

[A. *E-Rent (Electronic Rental)* 10](#_Toc64241127)

[B. Sewa Menyewa 11](#_Toc64241128)

[C. Perangkat Pemodelan Sistem 14](#_Toc64241129)

[1. Use Case Diagram 15](#_Toc64241130)

[2. Context Diagram 16](#_Toc64241131)

[3. Activity Diagram 17](#_Toc64241132)

[4. Sequence Diagram 18](#_Toc64241133)

[5. Entity Relationship Diagram 19](#_Toc64241134)

[6. Database 20](#_Toc64241135)

[D. Antarmuka Pengguna (*User* *Interfaces*) 22](#_Toc64241136)

[E. Perangkat Pengembangan 24](#_Toc64241137)

[1. Java 24](#_Toc64241138)

[2. PHP 24](#_Toc64241139)

[3. Extensible Markup Language (XML) 25](#_Toc64241140)

[4. JavaScript Object Notation (JSON) 26](#_Toc64241141)

[5. My Structured Query Language (MySQL) 27](#_Toc64241142)

[6. XAMPP 28](#_Toc64241143)

[7. Android Studio 28](#_Toc64241144)

[F. Penelitian Terdahulu 29](#_Toc64241145)

[BAB III ANALISIS DAN PERANCANGAN 32](#_Toc64241146)

[A. Analisis Sistem 32](#_Toc64241147)

[1. Analisis Sistem Berjalan 32](#_Toc64241148)

[a. Analisis Proses Bisnis/Aktivitas 32](#_Toc64241149)

[b. Analisis Aturan Bisnis (*Business Rule*) 45](#_Toc64241150)

[c. Analisis Pelaku Bisnis 46](#_Toc64241151)

[d. Analisis Dokumen I/O 47](#_Toc64241152)

[e. Analisis Masalah dan Solusi 50](#_Toc64241153)

[f. Flow Map Sistem Berjalan 51](#_Toc64241154)

[2. Sistem Yang Disusulkan 54](#_Toc64241155)

[a. Analisis *User* 54](#_Toc64241156)

[b. Analisis Proses dan Prosedur 57](#_Toc64241157)

[c. Analisis Dokumen I/O 62](#_Toc64241158)

[d. Analisis Persyaratan (*Requirement*) 65](#_Toc64241159)

[e. Flow Map Diusulkan 69](#_Toc64241160)

[B. Perancangan Sistem 71](#_Toc64241161)

[*1.* *Context Diagram* 72](#_Toc64241162)

[2. *Use case* *Diagram* 73](#_Toc64241163)

[*3.* *Activity Diagram* 75](#_Toc64241164)

[*4.* *Sequence Diagram* 83](#_Toc64241165)

[C. Perancangan *Database* 87](#_Toc64241166)

[1. Normalisasi 87](#_Toc64241167)

[*2.* *Entity Relationship Diagram (ERD)* 92](#_Toc64241168)

[3. Struktur Tabel 93](#_Toc64241169)

[D. Perancangan Antarmuka (*Interfaces*) 100](#_Toc64241170)

[1. Perancangan *Input* 100](#_Toc64241171)

[2. Perancangan *Output* 105](#_Toc64241172)

[3. Perancangan Pengelolaan Admin 113](#_Toc64241173)

[E. Rancangan Keamanan 114](#_Toc64241174)

[1. Perancangan Level Pengguna (*Session*) 115](#_Toc64241175)

[2. Perancangan Teknik Keamanan Enkripsi Data 116](#_Toc64241176)

[BAB IV HASIL DAN PEMBAHASAN 117](#_Toc64241177)

[A. Hasil Rancangan Tampilan 117](#_Toc64241178)

[1. Halaman Tampilan *Input* 117](#_Toc64241179)

[2. Halaman Tampilan *Output* 128](#_Toc64241180)

[3. Halaman Tampilan Lainnya 135](#_Toc64241181)

[B. Pengujian Sistem 139](#_Toc64241182)

[BAB V PENUTUP 142](#_Toc64241183)

[A. Kesimpulan 142](#_Toc64241184)

[B. Saran 143](#_Toc64241185)

[DAFTAR PUSTAKA 144](#_Toc64241186)

# DAFTAR GAMBAR

[**Gambar 1.** Penyebaran Informasi Penyewaan. 3](#_Toc64240973)

[**Gambar 2.** Simbol *Diagram* *Use case* 16](#_Toc64240974)

[**Gambar 3.** Tampilan Android Studio 3.5.3 29](#_Toc64240975)

[**Gambar 4.** *Flow-map* sistem saat ini 53](#_Toc64240976)

[**Gambar 5.** *Flow-map* Sistem diusulkan 70](#_Toc64240977)

[**Gambar 6.** *Context Diagram* 72](#_Toc64240978)

[**Gambar 7.** *Use Diagram* 74](#_Toc64240979)

[**Gambar 8.** *Register* *User* 75](#_Toc64240980)

[**Gambar 9.** *Activity* *Diagram* *Login* Sistem 76](#_Toc64240981)

[**Gambar 10.** *Activity* *Diagram* barang dan jasa 77](#_Toc64240982)

[**Gambar 11.** *Activity* *Diagram* CRUD Data Kategori 78](#_Toc64240983)

[**Gambar 12.** *Activity* *Diagram* *View* Data Jasa Sewa 79](#_Toc64240984)

[**Gambar 13.** *Activity Diagram* mencari barang dan jasa sewa 80](#_Toc64240985)

[**Gambar 14.** *Sequence Diagram Registration* 83](#_Toc64240986)

[**Gambar 15.** *Sequence Diagram Login* 84](#_Toc64240987)

[**Gambar 16.** *Sequence Diagram Administrator* 85](#_Toc64240988)

[**Gambar 17.** *Sequence Diagram viewer/public* 86](#_Toc64240989)

[**Gambar 18.** *Sequence Diagram* Penyedia sewa 86](#_Toc64240990)

[**Gambar 19.** ERD Sistem 92](#_Toc64240991)

[**Gambar 20.** Tampilan Halaman *Register* 100](#_Toc64240992)

[**Gambar 21.** Tampilan Halaman *Login* 101](#_Toc64240993)

[**Gambar 22.** Tampilan Halaman *Edit Profil* 102](#_Toc64240994)

[**Gambar 23.** Tampilan Halaman Sewa 103](#_Toc64240995)

[**Gambar 24.** Tampilan Halaman *Form Store* 104](#_Toc64240996)

[**Gambar 25.** Tampilan Halaman *Form* Pasang Sewa 104](#_Toc64240997)

[**Gambar 26.** Tampilan Halaman *Form* Input Testimony 105](#_Toc64240998)

[**Gambar 27.** Tampilan Halaman *Home* Aplikasi 106](#_Toc64240999)

[**Gambar 28.** Tampilan Halaman Kategori 107](#_Toc64241000)

[**Gambar 29.** Tampilan Halaman Bagaimana cara? 107](#_Toc64241001)

[**Gambar 30.** Tampilan Halaman Utama 108](#_Toc64241002)

[**Gambar 31.** Tampilan Halaman Store 109](#_Toc64241003)

[**Gambar 32.** Tampilan Halaman *Booked* 110](#_Toc64241004)

[**Gambar 33.** Tampilan Halaman Profil 110](#_Toc64241005)

[**Gambar 34.** Tampilan Halaman Profil 111](#_Toc64241006)

[**Gambar 35.** Gambar Detail Laporan Barang 111](#_Toc64241007)

[**Gambar 36.** Halaman Barang dan Jasa 112](#_Toc64241008)

[**Gambar 37.** Halaman Pemesanan 113](#_Toc64241009)

[**Gambar 38.** Halaman Pengelolaan Admin 114](#_Toc64241010)

[**Gambar 39.** Teknik Keamanan Level Pengguna (*Session*) 115](#_Toc64241011)

[**Gambar 40.** Teknik Keamanan Enkripsi MD5 116](#_Toc64241012)

[**Gambar 41.** Tampilan Halaman Registrasi 117](#_Toc64241013)

[**Gambar 42.** Halaman *Login* 119](#_Toc64241014)

[**Gambar 43.** Gambar Halaman Edit Profil 120](#_Toc64241015)

[**Gambar 44.** Tampilan Halaman *Input Store* 122](#_Toc64241016)

[**Gambar 45.** Halaman *Input* Barang 123](#_Toc64241017)

[**Gambar 46.** Halaman Pemesanan 125](#_Toc64241018)

[**Gambar 47.** Halaman *Checkout* Pemesanan 125](#_Toc64241019)

[**Gambar 48.** Bukti Pemesanan 126](#_Toc64241020)

[**Gambar 49.** Halaman *Input Review* dan *Rating* 127](#_Toc64241021)

[**Gambar 50.** Tampilan Halaman Utama 129](#_Toc64241022)

[**Gambar 51.** Halaman *Home* 130](#_Toc64241023)

[**Gambar 52.** Tampilan Halaman Profil 132](#_Toc64241024)

[**Gambar 53.** Halaman *Store* 133](#_Toc64241025)

[**Gambar 54.** Halaman Barang 134](#_Toc64241026)

[**Gambar 55.** Halaman Tampilan *Index* Admin 136](#_Toc64241027)

[**Gambar 56.** Halaman *Left* Admin 137](#_Toc64241028)

[**Gambar 57.** Halaman Pencarian 138](#_Toc64241029)

# DAFTAR TABEL

[**Tabel 1.** Notasi pada *Activity* Diagram 18](#_Toc64241187)

[**Tabel 2.** Karakteristik GUI 23](#_Toc64241188)

[**Tabel 3.** Penelitian Terdahulu 30](#_Toc64241189)

[**Tabel 4.** Analisis Proses Bisnis/Aktivitas 32](#_Toc64241190)

[**Tabel 5.** Analisis Aturan Bisnis 45](#_Toc64241191)

[**Tabel 6.** Analisis Pelaku Bisnis 46](#_Toc64241192)

[**Tabel 7.** Analisis Dokumen *Input* 47](#_Toc64241193)

[**Tabel 8.** Analisis Dokumen *Output*. 49](#_Toc64241194)

[**Tabel 9.** Analisis Masalah dan Solusi. 50](#_Toc64241195)

[**Tabel 10.** Analisis *User*. 55](#_Toc64241196)

[**Tabel 11.** Analisis Prosedur 58](#_Toc64241197)

[**Tabel 12.** Analisis Dokumen *Input* 62](#_Toc64241198)

[**Tabel 13.** Analisis Dokumen *Output*. 64](#_Toc64241199)

[**Tabel 14.** Tabel *Functional Requirements* 66](#_Toc64241200)

[**Tabel 15.** Analisis *Non-Functional Requirement* 68](#_Toc64241201)

[**Tabel 16.** Analisis *Hardware* *Requirements* 69](#_Toc64241202)

[**Tabel 17.** Analisis *Software Requirements* 69](#_Toc64241203)

[**Tabel 18.** Bentuk Tabel Tidak Normal 87](#_Toc64241204)

[**Tabel 19.** Bentuk Tabel Normal 1 88](#_Toc64241205)

[**Tabel 20.** Tabel Bentuk Normal Kedua 2NF 90](#_Toc64241206)

[**Tabel 21.** Tabel Bentuk Normal Ketiga (3NF) 91](#_Toc64241207)

[**Tabel 22.** Tabel *User* 93](#_Toc64241208)

[**Tabel 23.** Tabel Admin 94](#_Toc64241209)

[**Tabel 24.** Tabel *Store* 94](#_Toc64241210)

[**Tabel 25.** Tabel *Member* 95](#_Toc64241211)

[**Tabel 26.** Tabel Penyewa 95](#_Toc64241212)

[**Tabel 27.** Tabel Kategori 96](#_Toc64241213)

[**Tabel 28.** Tabel Barang dan Jasa 96](#_Toc64241214)

[**Tabel 29.** Tabel Foto Barang 96](#_Toc64241215)

[**Tabel 30.** Tabel *Block* *User* 97](#_Toc64241216)

[**Tabel 31.** Tabel Testimoni 97](#_Toc64241217)

[**Tabel 32.** Tabel Pemesanan 98](#_Toc64241218)

[**Tabel 33.** Tabel Pembayaran 99](#_Toc64241219)

[**Tabel 34.** Tabel Bintang 99](#_Toc64241220)

[**Tabel 35.** Pengujian *Test* Unit dan Uji Coba Sistem pada Public 139](#_Toc64241221)

[**Tabel 36.** Pengujian *Test* Unit dan Uji Coba Sistem pada *user.* 140](#_Toc64241222)

# BAB I PENDAHULUAN

## Latar Belakang Masalah

Perkembangan globalisasi informasi yang didukung oleh kemajuan teknologi kini telah lumrah di masyarakat. Salah satu hasil dari kemajuan teknologi adalah lahirnya berbagai macam alat elektronik yang mampu membantu pekerjaan manusia seperti komputer, laptop, *smartphone*, *smartwatch,* dan lain-lain. Alat elektronik ini digunakan sebagai media untuk bertukar informasi yang didukung dengan internet sebagai sarana penghubung. *E-Rent* (*Electronic Rental*) merupakan aplikasi yang menyediakan layanan secara elektronik mengenai informasi penyewaan(*rent*). Menurut Kamus Besar Bahasa Indonesia, rental berarti persewaan. Kata rental sendiri merupakan kata serapan yang diambil dari bahasa inggris yaitu *rent*.

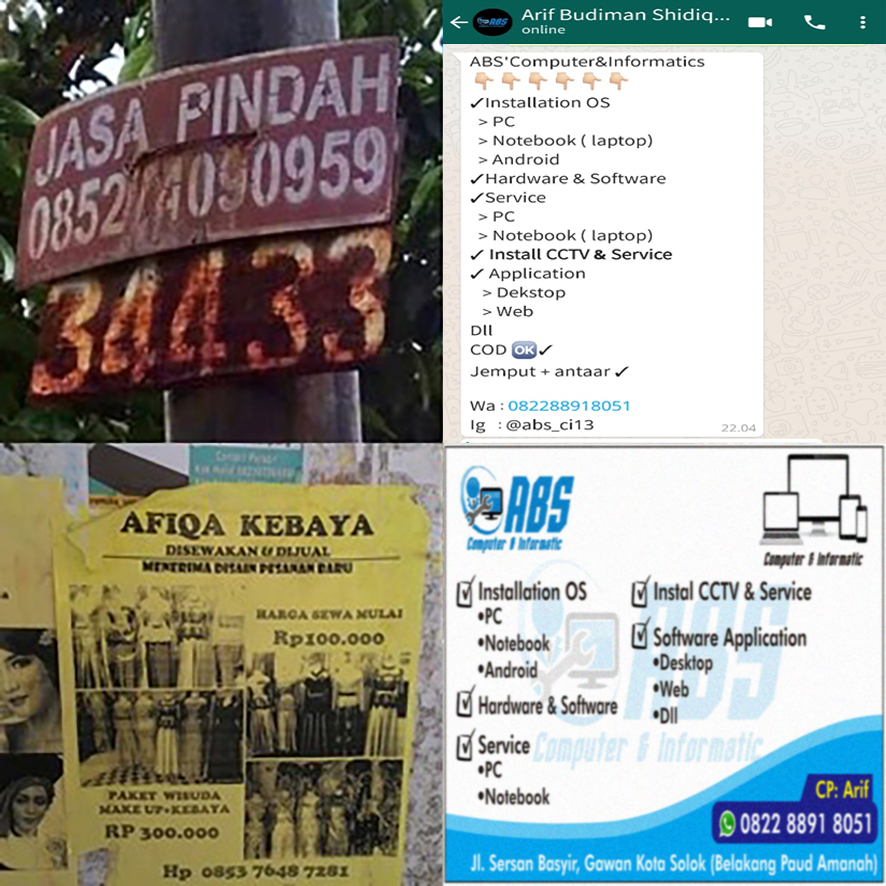
Penyewaan (*rent*) merupakan salah satu bisnis yang mudah dan juga dibutuhkan saat ini. Penyewaan adalah sebuah persetujuan antara dua belah pihak penyewa dan penyedia sewa terhadap sebuah pembayaran yang dilakukan atas penggunaan suatu barang atau properti atau jasa secara sementara. Hal ini juga diperjelas oleh Subekti (2008) yang mengatakan bahwa penyewaan adalah pemindahan hak guna pakai suatu barang, benda atau jasa dari pihak pemilik barang kepada pihak penyewa dalam jangka waktu tertentu, dengan pembayaran uang sewa oleh pihak penyewa kepada pemilik barang atau benda sesuai perjanjian kedua belak pihak.

Jenis barang yang disewa, harga, dan juga lamanya waktu sewa bermacam-macam. Rumah umumnya disewa dalam setahun, mobil dalam sehari, dan komputer disewa dalam hitungan jam. Waktu penggunaan barang atau jasa yang tidak lama dan biaya yang harus dikeluarkan lebih ekonomis dibandingkan membeli produk yang disewa. Inilah yang pada akhirnya mendorong masyarakat lebih memilih melakukan penyewaan daripada membeli.

Sementara itu, dalam melakukan penyewaan, masyarakat perlu mengetahui lokasi, harga, stok barang serta foto barang atau jasa yang disewa. Jika diamati, beberapa penyewa barang dan jasa mengalami kesulitan mendapatkan informasi tentang barang yang disewa. Informasi barang cenderung tidak mendetail dan menjawab rasa ingin tahu tentang barang yang disewa. Misalnya, saat seseorang ingin menyewa sebuah gedung. Adakalanya penyewa mengetahui lokasi gedung, tetapi tidak mengetahui harga sewa dari gedung tersebut. Akhirnya, mereka harus mendatangi satu per satu lokasi sampai menemukan harga yang sesuai. Situasi lain misalnya saat ingin menyewa mobil. Penyewa mengetahui lokasi harga sewa mobil tersebut, tetapi tidak mengetahui jumlah mobil tersedia atau tidak untuk disewa.

Namun, dengan memanfaatkan media sosial sebagai daya dukung, penyewa dan penyedia sewa memungkinkan berbagai informasi tentang barang yang mereka perbincangkan. Masyarakat (penyedia sewa dalam hal ini) yang mempunyai barang atau jasa untuk disewakan mempromosikan barang ke berbagai media sosial seperti aplikasi *WhatsApp*, *Facebook*, atau *Instagram.* Akan tetapi, kendala yang sering diterima oleh penyedia sewa dalam melakukan promosi melalui media sosial adalah harus menjawab pertanyaan yang sama untuk beberapa penyewa yang akan menyewakan barang. Hal ini dirasa kurang efektif.

Di samping menggunakan media sosial sebagai promosi, informasi mengenai barang dan jasa yang akan disewakan juga sebarkan melalui brosur, poster yang di tempel di mading-mading kampus bahkan dipinggir jalan. Seperti yang terlihat pada Gambar 1.



**Gambar 1.** Penyebaran Informasi Penyewaan.

Gambar 1 membuktikan media penyebaran informasi barang dan jasa yang ada masih tidak merata.

Proses penyewaan umumnya diawali dengan pengisian formulir data diri di kertas oleh penyewa dan memberikan beberapa jaminan berupa KTP, KK, atau SIM. Data dan jaminan yang diberikan ini digunakan oleh penyedia sebagai bukti untuk dapat menyerahkan kepemilikan sementara barang untuk disewakan. Akan tetapi, kendala lain yang sering terjadi terkait hal ini adalah kerusakan atau hilangnya formulir berisi data dan jaminan tersebut sehingga tidak ada bukti fisik untuk penyedia sewa terhadap data barang yang sedang disewa.

Berdasarkan beberapa kendala yang sudah dijabarkan di atas, upaya memudahkan penyewa dalam mencari informasi barang untuk disewa dan memberikan wadah khusus bagi penyedia sewa menginformasikan barang atau jasa mereka serta meminimalisir kehilangan bukti penyewaan, maka penulis membuat sebuah aplikasi *e-rent* yang dapat menampung informasi mengenai barang dan jasa dari berbagai jenis barang yang umum disewakan. Aplikasi ini menyediakan berbagai sewaan baik jasa maupun barang dari berbagai penyedia sewa. Artinya, aplikasi ini bersifat sewa harian yang perpanjangannya disesuaikan kesepakatan kedua belah pihak.

*Android* merupakan salah satu sistem operasi yang ditanamkan di dalam sebuah *Smartphone*, *Smartphone* dengan sistem operasi *Android* merupakan telepon pintar yang paling banyak digunakan pada waktu ini. Selain itu Android juga menyediakan kerangka kerja aplikasi yang kaya dan memungkinkan Anda membangun aplikasi dan *game* inovatif untuk perangkat seluler di lingkungan bahasa pemrograman Java. Dokumen yang tercantum di navigasi sebelah kiri menyediakan detail tentang cara membangun aplikasi menggunakan beragam Android API.

Aplikasi Android dibangun sebagai kombinasi beragam komponen yang bisa dipanggil satu per satu. Misalnya, satu aktivitas individual menyediakan satu layar untuk antarmuka pengguna, dan layanan yang secara terpisah melakukan tugas di latar belakang. Android menyediakan kerangka kerja aplikasi adaptif yang memungkinkan Anda menyediakan sumber daya unik bagi konfigurasi perangkat yang berbeda-beda. Misalnya, Anda bisa membuat berbagai *file* *layout* XML untuk ukuran layar yang berbeda-beda dan sistem akan menentukan *layout* yang akan diterapkan berdasarkan ukuran layar perangkat saat ini

Berdasarkan latar belakang di atas, maka diajukan tugas akhir yang saya angkat yaitu dengan judul **“Perancangan Aplikasi *E-Rent* (*Electronic Rental*) Berbasis Android”.**

## Identifikasi Masalah

Berdasarkan latar belakang di atas, dirumuskan permasalahan yang akan diselesaikan, yaitu sebagai berikut.

1. Masyarakat membutuhkan barang atau jasa dalam suatu waktu, tetapi belum cukup finansial untuk memiliki barang tersebut dengan jalan membeli sehingga *rent* ‘sewa’ menjadi alternatif.
2. Masyarakat yang ingin menyewakan barang membutuhkan lebih banyak waktu untuk mencari informasi mengenai barang yang akan mereka sewa karena informasi yang tersedia belum menggambarkan dengan baik barang yang disewakan sehingga informasi yang diperoleh penyewa tidak mendetail.
3. Layanan informasi tentang barang yang disewakan antara penyewa dan penyedia sewa di media sosial kurang efektif karena memungkinkan seringnya penyedia sewa melayani penyewa dengan pertanyaan yang sama dengan penyewa lainnya.
4. Media promosi dalam bentuk cetak baik brosur maupun pamflet tidak ekonomis dan tidak merata sehingga tidak menjangkau masyarakat dalam jumlah besar.
5. Potensi kehilangan bukti fisik kepemilikan barang bagi penyedia sewa tinggi karena proses *input* data penyewa dilakukan secara manual dengan media kertas.

## Batasan Masalah

Berdasarkan identifikasi masalah di atas, dirumuskan batasan masalah sebagai berikut.

1. Merancang aplikasi *electronic rental* *(e-rent)* atau rental elektronik menggunakan berbasis Android di Kota Payakumbuh.
2. Aplikasi ini menyediakan fitur pencarian berdasarkan nama barang yang ingin dicari.
3. Aplikasi ini menyediakan layanan pemesanan barang, *input testimonial*, dan *rating* atau penilaian terhadap barang dan jasa yang disewa.
4. Transaksi untuk penyewaan serta serah terima barang tidak dilakukan di dalam aplikasi.
5. Pengguna dari aplikasi ini adalah masyarakat umum yang ingin menyewakan barang atau mencari barang untuk disewakan di Kota Payakumbuh.
6. Aplikasi ini berbasis arsitektur *Model-View-Controller* atau MVC yang dibangun menggunakan *Retrofit* *API* di *Android*.
7. Sistem dibangun menggunakan bahasa pemrograman *Java,* *MySQL* sebagai *Database Management System* (DBMS), dan *Android Studio* sebagai *Integrated Development Environment* (IDE).

## Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah, dirumuskan rumusan masalah sebagai berikut.

1. Bagaimana merancang aplikasi *electronic rental (e-rent)* berbasis ­­­­­Android di Kota Payakumbuh ?
2. Bagaimana membuat aplikasi dengan fitur pencarian berdasarkan nama barang yang ingin dicari berbasis Android di Kota Payakumbuh ?
3. Bagaimana membuat aplikasi yang dapat menyediakan layanan pemesanan barang, *input* *testimonial* dan *rating* atau penilaian terhadap barang dan jasa yang di sewa berbasis Android di Kota Payakumbuh ?
4. Bagaimana menampilkan informasi data penyewa, data penyedia sewa, data barang atau jasa seperti harga, stok, dan foto barang yang akan disewakan berbasis Android di Kota Payakumbuh ?
5. Bagaimana membuat aplikasi Android berbasis arsitektur *Model*-*View*-*Controller* atau MVC yang di bangun dengan *Retrofit* *API* berbasis Android di Kota Payakumbuh ?
6. Bagaimana mengembangkan aplikasi Android menggunakan bahasa pemrograman Java, *MySQL* sebagai *Database Management System* (DBMS), dan *Android Studio* *Integrated Development Environment* (IDE). berbasis Android di Kota Payakumbuh ?

## Tujuan Tugas Akhir

Adapun tujuan dari tugas akhir ini adalah sebagai berikut.

1. Menghasilkan rancangan aplikasi *electronic rental (e-rent)* berbasis Android di Kota Payakumbuh.
2. Menghasilkan aplikasi dengan fitur pencarian berdasarkan nama barang yang ingin dicari berbasis Android di Kota Payakumbuh.
3. Menghasilkan aplikasi yang dapat menyediakan layanan pemesanan barang, *input* *testimonial* dan *rating* atau penilaian terhadap barang dan jasa yang di sewa berbasis Android di Kota Payakumbuh.
4. Menyajikan informasi data penyewa, data penyedia sewa, data barang atau jasa seperti harga, stok, dan foto barang yang akan disewakan berbasis Android di Kota Payakumbuh.
5. Menghasilkan aplikasi Android berbasis arsitektur *Model*-*View*-*Controller* atau MVC yang di bangun dengan *Retrofit* *API* berbasis Android di Kota Payakumbuh.
6. Menghasilkan aplikasi Android menggunakan bahasa pemrograman Java, *MySQL* sebagai *Database Management System* (DBMS), dan *Android Studio* *Integrated Development Environment* (IDE). berbasis Android di Kota Payakumbuh.

## Manfaat Tugas Akhir

Adapun manfaat dari tugas akhir ini adalah sebagai berikut.

1. Manfaat Teoritis
2. Menyumbangkan *best* *practice* dalam pengembangan rekayasa perangkat lunak.
3. Mengimplementasikan teknik arsitektur MVC ke dalam sebuah sistem informasi untuk memberikan kemudahan dalam membuat sebuah aplikasi.
4. Manfaat Praktis
5. Bagi masyarakat (penyedia sewa) yang ingin menyewakan barang dapat mempromosikan harga dan lokasi barang / jasa akan disewakan dalam satu tempat dan menekan tingkat kehilangan data penyewa melalui *input* data secara digital.
6. Bagi masyarakat (penyewa) yang membutuhkan barang untuk disewa (penyewa) mendapatkan informasi perbandingan harga dan lokasi dari barang dan jasa yang akan disewa dengan mudah, lengkap, dan cepat.

# BAB II LANDASAN TEORI

1. *E-Rent (Electronic Rental)*

Sistem Layanan Elektronik atau E-layanan (bahasa Inggris: *Electronic Services* di singkat E-Services) merupakan satu aplikasi terkemuka memanfaatkan Teknologi Informasi dan Komunikasi (TIK) di daerah yang berbeda. Namun, definisi yang tepat dari layanan elektronik sulit didapat sebagian peneliti telah menggunakan definisi yang berbeda untuk menggambarkan layanan elektronik. Meskipun definisi ini berbeda, dapat dikatakan bahwa mereka semua sepakat tentang peran teknologi dalam memfasilitasi pelayanan yang membuat mereka lebih dari layanan elektronik.

Menurut Rowley (dalam Ihsan, Mohammad Siregar, 2019:170), layanan elektronik di definisikan sebagai: “...perbuatan, usaha atau pertunjukan yang pengiriman di mediasi oleh teknologi informasi. Layanan elektronik tersebut meliputi unsur layanan e-tailing, dukungan pelanggan, dan pelayanan”. Definisi ini mencerminkan tiga komponen utama, penyedia layanan, penerima layanan dan saluran pelayanan (yaitu, teknologi). Misalnya, sebagai yang bersangkutan untuk layanan elektronik publik, badan publik adalah penyedia layanan dan warga negara serta bisnis penerima layanan. Saluran pelayanan adalah persyaratan ketiga dari layanan elektronik. Internet adalah saluran utama dari layanan elektronik pengiriman sementara saluran klasik lainnya juga.

Layanan berbasis elektronik ini juga dapat dimanfaatkan dalam bisnis penyewaan (*rent*). *E-Rent (Electronic Rental)* merupakan aplikasi yang memberikan layanan secara elektronik mengenai informasi penyewaan (*rent*). Informasi seperti nama barang, bentuk barang, harga, stok hingga pemesanan terhadap barang yang akan disewakan disajikan dalam bentuk digital yang dapat di akses melalui alat elektronik seperti komputer, laptop, ponsel, dll.

Menurut Lu (dalam Buchari, Rd. Ahmad, 2016:238) mengidentifikasi sejumlah manfaat dari penggunaan layanan elektronik. Pertama, Mengakses basis pelanggan yang lebih besar. Kedua dapat memperluas jangkauan pasar. Ketiga, menurunkan penghalang masuk ke pasar baru dan biaya mendapatkan pelanggan baru. Keempat, alternatif saluran komunikasi ke pelanggan. Kelima, meningkatkan pelayanan kepada pelanggan. Keenam, meningkatkan citra perusahaan. Ketujuh, mendapatkan keunggulan kompetitif. Kedelapan, potensi peningkatan pengetahuan pelanggan.

1. Sewa Menyewa

Perjanjian sewa-menyewa diatur di dalam bab VII Buku III KUHP yang berjudul “Tentang Sewa-Menyewa” yang meliputi pasal 1548 sampai dengan pasal 1600 KUH Perdata. Definisi perjanjian sewa-menyewa menurut Pasal 1548 KUHP menyebutkan bahwa: “Perjanjian sewa-menyewa adalah suatu perjanjian, dengan mana pihak yang satu mengikatkan dirinya untuk memberikan kepada pihak yang lainya kenikmatan dari suatu barang, selama waktu tertentu dan dengan pembayaran suatu harga, yang oleh pihak tersebut belakangan telah disanggupi pembayarannya.”

Sewa-menyewa dalam bahasa Belanda disebut dengan *huurenverhuur* dan dalam bahasa Inggris disebut dengan *rent* atau *hire*. Sewa-menyewa merupakan salah satu perjanjian timbal balik. Menurut Kamus Besar Bahasa Indonesia sewa berarti pemakaian sesuatu dengan membayar uang sewa dan menyewa berarti memakai dengan membayar uang sewa.

Menurut Fitria, Z. I. (2018) Sewa menyewa adalah suatu bentuk hubungan kerja sama antara dua pihak, di mana pihak pertama memberikan barangnya untuk pihak kedua dengan jangka waktu tertentu. Berdasarkan beberapa pendapat di atas, dapat disimpulkan sewa-menyewa merupakan perjanjian yang dilakukan oleh dua pihak penyewa dan penyedia sewa atas kepemilikan sementara suatu barang dengan membayar uang sewa sesuai harga yang sudah disepakati sebelumnya.

Pihak-pihak yang terlibat dalam Perjanjian sewa-menyewa adalah:

1. Pihak yang menyewakan

Pihak yang menyewakan adalah orang atau badan hukum yang menyewakan barang atau benda kepada pihak lainya untuk dinikmati kegunaan benda tersebut kepada penyewa. Pihak yang menyewakan barang atau benda tidak harus pemilik benda sendiri tetapi semua orang yang atas dasar hak penguasaan untuk memindahkan pemakaian barang ke tangan orang lain. Hal tersebut dikarenakan di dalam sewa-menyewa yang diserahkan kepada pihak penyewa bukanlah hak milik atas suatu barang melainkan hanya pemakaian atau pemungutan atas hasil dari barang yang disewakan. Pihak yang menyewakan barang memiliki hak dan kewajiban dalam menyewakan barang. Hak dari pihak yang menyewakan adalah menerima harga sewa yang telah ditentukan. Kewajiban dari yang menyewakan barang diatur dalam Pasal 1551-1552 KUH Perdata. Pertama, barang yang disewakan harus diserahkan dalam keadaan baik. Kedua, barang yang disewakan harus terus dijaga baik-baik dan yang rusak wajib diperbaiki (apabila hal tersebut menjadi tanggung jawabnya). Ketiga, menjamin terhadap penyewa untuk dapat memakai dan menggunakan barang yang disewa dengan aman selama berlaku perjanjian sewa menyewa. Keempat, menanggung segala kekurangan pada benda yang disewakan, yaitu kekurangan-kekurangan yang dapat menghalang-halangi pemakaian benda itu, walaupun ia sejak berlakunya perjanjian itu tidak mengetahui adanya kekurangan atau cacat tersebut.

1. Pihak Penyewa

Pihak penyewa adalah orang atau badan hukum yang menyewa barang atau benda dari pihak yang menyewakan. Penyewa memilik hak dan kewajiban dalam menyewa barang. Hak dari penyewa adalah menerima barang yang disewakan dalam keadaan baik. Kewajiban dari penyewa diatur Pasal 1560-1566 KUH Perdata. Pertama, membayar uang sewa pada waktu yang telah ditentukan. Kedua, tidak diperkenankan mengubah tujuan barang yang disewakan. Ketiga, mengganti kerugian apabila terjadi kerusakan yang disebabkan oleh penyewa sendiri, atau oleh orang-orang yang diam di dalam rumah yang disewa. Keempat, mengembalikan barang yang disewa dalam keadaan semua ketika perjanjian sewa menyewa tersebut telah habis waktunya. Kelima, menjaga barang yang disewa sebagai seseorang yang bertanggung jawab. Keenam, tidak boleh menyewakan lagi barang sewaannya kepada orang lain. Apabila telah ditentukan demikian, dan ketentuan tersebut dilanggar, maka perjanjian dapat dibubarkan dan penyewa dapat dituntut mengganti perongkosan, kerugian, serta bunga.

Tujuan dari diadakannya perjanjian sewa-menyewa adalah untuk memberikan hak pemakaian kepada pihak penyewa sehingga benda yang bukan berstatus hak milik dapat disewakan oleh pihak yang mempunyai hak atas benda tersebut. Jadi benda yang dapat disewakan oleh pihak yang menyewakan dapat berupa hak milik, hak guna usaha, hak pakai, hak menggunakan hasil, hak pakai, hak sewa (hak sewa kedua) dan hak guna bangunan.

1. Perangkat Pemodelan Sistem

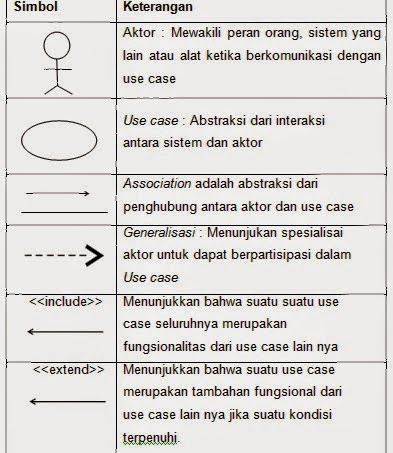
Pemodelan adalah gambaran dari Realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan perangkat lunak digunakan untuk mempermudah langkah berikutnya dari pengembangan sebuah sistem informasi sehingga lebih terencana. Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Dalam perkembangannya, muncullah sebuah standarisasi bahasa pemodelan yaitu *Unified Modeling Language* (UML).

UML adalah sebuah standar penulisan atau penotasian *blueprint* perangkat lunak yang digunakan untuk membantu para software developer memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak (Pressman, 2010). UML memiliki beberapa diagram, di antaranya sebagai berikut.

1. Use Case Diagram

*Use Case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang dibuat. Secara kasarnya, *use case* diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Rosa dan Shalahuddin, 2014: 155).

*Use case* menyatakan sebuah aktivitas atas pekerjaan tertentu, misalnya *login* ke sistem, melakukan *create* sebuah daftar mahasiswa, dan lain sebagainya. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Elemen-elemen yang digunakan dalam pembuatan diagram *use case* yaitu:



**Gambar 2.** Simbol *Diagram* *Use case*

Langkah-langkah yang dibutuhkan untuk menyusun diagram *use case*: mengidentifikasi pelaku bisnis, mengidentifikasi *use case* persyaratan bisnis, membuat diagram model *use case*, dan mendokumentasikan naratif use case persyaratan bisnis.

1. Context Diagram

*Context diagram* menurut Hanif (2007:109), adalah sistem yang digambarkan dengan sebuah proses, dan menunjukkan semua entitas yang menerima informasi dari atau memberikan informasi ke sistem. *Context Diagram* merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran dari sistem diketahui penganalisis dari wawancara dengan *user* dan sebagai hasil analisis dokumen. *Context diagram* menggaris bawahi sejumlah karakteristik penting dari suatu sistem. Pertama, kelompok pemakai, organisasi atau sistem lain, di mana sistem melakukan komunikasi yang disebut juga sebagai *terminator*. Kedua, data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu. Ketiga, data yang dihasilkan sistem kita dan diberikan ke dunia luar. Keempat, penyimpanan data (*data store*). Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Kelima, batasan antara sistem dengan dunia luar (*rest of the world*).

1. Activity Diagram

*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*. *Activity* *Diagram* digunakan untuk memodelkan aspek dinamis dari sistem. Dan juga sistem-sistem yang dapat dieksekusi, baik untuk *forward engineering* atau *reverse engineering.*

**Tabel 1.** Notasi pada *Activity* Diagram

|  |  |  |
| --- | --- | --- |
| **Notasi** | **Nama** | **Keterangan** |
|  | *Start State* | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| **activity** | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan /  *decision* | Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu. |
|  | Penggabungan  / *join* | Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | *Stop State* | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

1. Sequence Diagram

Sequence Diagram melambangkan interaksi antar objek di dalam dan di sekitar sistem, yang digunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respons dari sebuah kegiatan untuk menghasilkan *output* tertentu. *Sequence* *Diagram* menjelaskan interaksi antar objek yang disusun sesuai dengan urutan kejadian yang dilakukan oleh seorang aktor.

*Sequence* *Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Dimulai dari apa aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Langkah awal yang perlu kita lakukan untuk membuat sebuah *Sequence* Diagram yaitu menentukan aktor, kemudian menentukan proses apa saja yang akan dilakukan oleh aktor tersebut terhadap sistem, dengan adanya *use case* diagram kita menjadi lebih mudah untuk merancang hak-hak *user* terhadap sistem serta mengetahui proses yang dijalankan sistem.

1. Entity Relationship Diagram

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Jadi, jelaslah bahwa ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data. Pada dasarnya ada tiga macam simbol yaitu:

* 1. *Entity*

*Entity* adalah suatu obyek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. *Entity* di gambarkan menggunakan persegi empat.

* 1. Atribut

*Entity* memiliki elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter *entity*. Atribut bisa digambarkan dengan menggunakan simbol elips.

* 1. Hubungan (Relasi)

Sebagai mana halnya entity maka dalam hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antar *entity* dengan isi dari hubungan itu sendiri.

1. Database

Para pengguna sistem informasi bisa memperoleh informasi untuk berbagai kepentingan berkat keberadaan *database*. Data yang tersimpan dalam *database* dapat diambil dan diproses sehingga menghasilkan informasi.

1. Pengertian *Database*

Menurut Fathansyah (dalam Novita, R., & Sari, N. 2015), mengatakan *database* atau basis data terdiri dari 2 kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan dan lain sebagainya yang diwujudkan dalam bentuk angka, huruf, simbol, text, bunyi, atau kombinasinya.

Menurut Mohamad Subhan (2012:209) *database* adalah kumpulan data, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelaksasi. Sedangkan Menurut Yudi Priyadi (2014:2), *database* adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital. Dari kedua pendapat tersebut, dapat disimpulkan bahwa *database*.

1. Komponen *Database*

Menurut Yudi Priyadi (2014:83), *database* terbagi dalam tiga komponen, yaitu:

1. *Data Definition Language* (DDL)

DDL merupakan perintah SQL yang digunakan untuk melakukan definisi awal suatu basis data dan tabel pada konsep RDBMS. Perintahnya adalah:

*Create*: untuk membuat/menciptakan *object database*.

*Alter*: untuk memodifikasi/mengubah *object database*.

*Drop*: untuk menghapus *object database*.

*Object database* yang dimaksud adalah terdiri dari *database*, *table*, *index* dan *view*.

1. *Data Manipulation Language* (DML)

DML merupakan perintah SQL yang digunakan untuk melakukan pengolahan *record* atau memanipulasi dan mengambil data pada suatu basis data. Manipulasi data misalnya:

*Select*: digunakan untuk mengambil data dari *database*

*Delete*: digunakan untuk menghapus data pada *database*

*Insert*: digunakan untuk menambahkan data pada *database*

*Update*: digunakan untuk memodifikasi data pada *database*

1. *Data Control Language* (DCL)

DCL adalah merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna dalam basis data.

1. Antarmuka Pengguna (*User* *Interfaces*)

Antarmuka pemakai (*User Interface*) merupakan mekanisme komunikasi antara pengguna (*user*) dengan sistem. Antarmuka pemakai (*User Interface*) dapat menerima informasi dari pengguna (*user*) dan memberikan informasi kepada pengguna (*user*) untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi. *User Interface* terdiri dari *Text-Based* dan *Graphical User Interface* (GUI). *Text-Based* menggunakan *syntax*/rumus yang sudah ditentukan untuk memberikan perintah. GUI menggunakan unsur-unsur multimedia (seperti gambar, suara, video) untuk berinteraksi dengan pengguna.

Saat ini interface yang banyak digunakan dalam *software* adalah GUI (*Graphical User Interface*). GUI memberikan keuntungan seperti: Pertama, mudah dipelajari oleh pengguna yang pengalaman dalam menggunakan komputer cukup minim. Kedua, berpindah dari satu layar ke layar yang lain tanpa kehilangan informasi. Ketiga, akses penuh pada layar dengan segera untuk beberapa macam tugas/keperluan. *Graphical User Interface* (GUI) memiliki beberapa karakteristik. Karakteristik tersebut dapat dilihat pada tabel berikut :

**Tabel 2.** Karakteristik GUI

|  |  |
| --- | --- |
| **Karakteristik** | **Penjelasan** |
| *Window* | Beberapa *window* bisa menampilkan informasi-informasi berbeda sekaligus pada layar. |
| *Icon* | Mewakili informasi yang berbeda seperti *icon* untuk *file*, *icon* *folder* atau *icon* program tertentu. |
| Halaman | Menawarkan perintah-perintah yang disusun dalam halaman tanpa harus mengetik. |
| *Pointing* | Alat penujuk seperti *mouse* untuk memilih pilihan pada layar. |
| *Graphic* | Gambar yang bisa dicampur dengan teks pada *display* yang sama untuk menyajikan informasi. |

Tujuan sebuah *User Interface* adalah mengkomunikasikan fitur-fitur sistem yang tersedia agar *user* mengerti dan dapat menggunakan sistem tersebut. Dalam hal ini penggunaan bahasa amat efektif untuk membantu pengertian, karena bahasa merupakan alat komunikasi tertua kedua *gesture*, yang dipakai orang untuk berkomunikasi sehari-harinya.

Tanpa bahasa pun kadang ikon bisa tidak jelas maknanya, sebab tidak semua lambang ikon bisa bersifat universal. Meski penting, namun sayangnya kadang penggunaan bahasa, seperti pemilihan istilah, sering dianggap kurang begitu penting. Bahasa sering menjadi sesuatu yang nomor dua ketimbang elemen-elemen *interface* lainnya.

1. Perangkat Pengembangan
2. Java

Menurut Liang (dalam Sari, D., dkk. 2019:9), Java merupakan bahasa pemrograman yang dikembangkan oleh *James Gosling* beserta tim di *Sun Microsystem* pada tahun 1991. Awalnya Java disebut dengan “Oak”. Namun pada tahun 1995, nama “Oak” diganti menjadi dengan Java. Bahasa pemrograman Java dirancang untuk menjadi bahasa pemrograman *multi-platform* yang cukup aman dan tangguh. Java dapat digunakan untuk pemrograman web, maupun aplikasi mandiri di seluruh platform pada server, desktop, dan perangkat *mobile*.

Pada Java Acuan pelaksanaan *compiler*, *virtual machine*, dan *class libraries* yang dapat berjalan pada *Java Virtual Machine* (JVM) terlepas dari arsitektur komputer. Hal ini dimaksudkan agar pengembang aplikasi dapat "*write once, run anywhere*".

1. PHP

Menurut Bunafit Nugroho (dalam Susianto, Didi 2019:60), PHP termasuk program yang hanya bisa berjalan di sisi *server* atau sering disebut *Side Server Language.* Jadi, program yang dibuat dengan kode PHP tidak bisa berjalan kecuali bila dijalankan pada *server* web*,* tanpa adanya *server* webyang terus berjalan, maka web tidak akan bisa dijalankan. Penerapan PHP dalam pemrograman secara umum ditulis sebagai berikut :

<?php

echo “Hello World! ”;?>

PHP merupakan bahasa *Server Side Scripting*, di mana PHP selalu membutuhkan *webserver* dalam menjalankan aksinya. Secara prinsip, *server* akan bekerja apabila ada permintaan dari *client* berupa kode-kode PHP. *Client* tersebut akan dikirimkan ke *server*, kemudian *server* akan mengembalikan pada halaman sesuai instruksi yang diminta

Menurut Edy Winarno (dalam Susianto, Didi 2019:60), berikut beberapa kelebihan dari PHP. Pertama, mudah dipelajari dan memiliki tingkat akses yang cepat. Kedua, *multiplatform*, yang berarti PHP dapat digunakan ke dalam berbagai *platform OS* (*Operating System*) dan hampir semua *browser* juga mendukung PHP. Ketiga, *Free* atau gratis, bersifat *open source*. Keempat, didukung oleh beberapa macam *web server* dan *database.*

1. Extensible Markup Language (XML)

(Sudirman, 2016) Mengemukakan XML adalah bahasa *markup* yang menggunakan *tag* sebagai penanda untuk mengategorikan, menjelaskan data lebih spesifik. Pada XML terdapat *tag* pembuka dan *tag* penutup. *Tag* pembuka dan *tag* penutup hanya mempunyai perbedaan karakter / pada *tag* penutup. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standardisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah antar satu dengan yang lain.

1. JavaScript Object Notation (JSON)

(Sudirman, 2016) Mengemukakan JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah di terjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1996. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apa pun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON adalah salah satu bahasa markup yang dapat melakukan pertukaran data di mana JSON ini dibuat berdasarkan *JavaScript* dan pastinya sintaknya lebih ke *JavaScript*. Dengan membuat sebuah JSON sama halnya dengan kita membuat sebuah object pada JavaScript itu sendiri. Di dalam membuat JSON pasti kita bakal berkenalan dengan yang namanya *array* pada JavaScript sehingga memudahkan bagi para developer/programmer.

JSON terbuat dari dua struktur:

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

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

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

1. My Structured Query Language (MySQL)

*MySQL* adalah sebagai berikut: *MySQL* (*My Structured Query Language*) adalah sebuah program pembuatan dan pengelola *Database* atau yang disering disebut dengan DBMS (*Database* Management System). Kelebihan lain dari *MySQL* adalah menggunakan bahasa *query* (permintaan) standar SQL. SQL adalah suatu bahasa permintaan yang terstruktur. *MySQL* dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. Hingga saat ini, *MySQL* sudah berkembang hingga versi 8. *MySQL 8* sudah lebih memudahkan pengelolaan tabel dalam *database.*

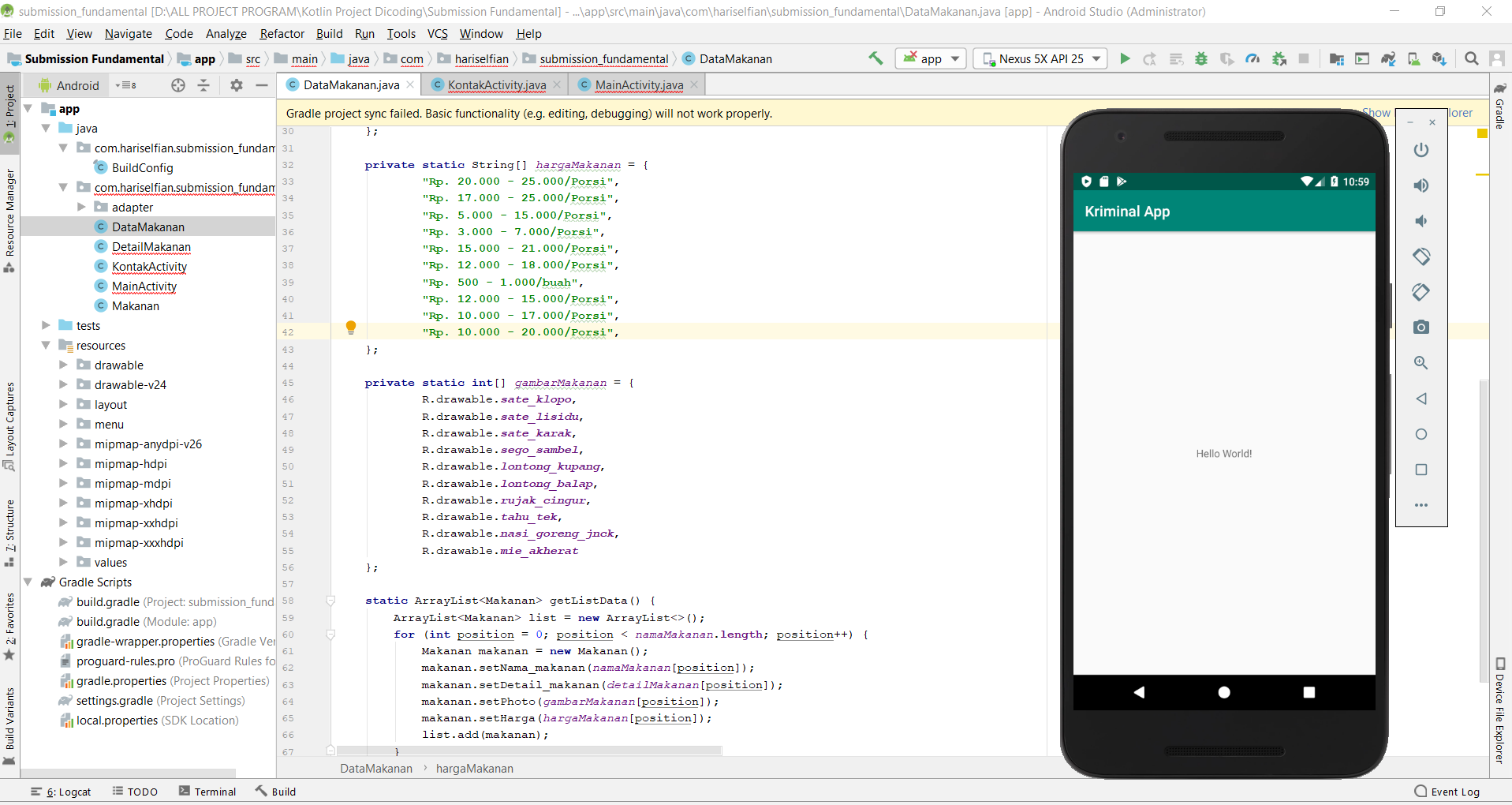
1. XAMPP

Menurut Sidik (2014:72), XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *server web* PHP *database* MySQL yang paling populer di kalangan pengembang Android dengan menggunakan PHP dan MySQL sebagai *database* Nya.

1. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

1. Sistem versi berbasis *Gradle* yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. *Instant* Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. *Template* kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine.*



**Gambar 3.** Tampilan Android Studio 3.5.3

Android Studio menyediakan banyak sekali fitur yang sangat berguna bagi pengembang perangkat lunak untuk menuliskan programnya. Beberapa di antaranya adalah *Editor* Tata letak *Visual* yang menampilkan tata letak aplikasi yang dibuat dan bisa juga untuk mengubah pratinjau ukuran layout, *APK Analyzer, Fast Emulator, Intelligent code editor, Flexible build system* dan lain-lain.

1. Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji tugas akhir yang dilakukan. Pada pembuatan tugas akhir ini, penulis menggunakan lima penelitian terdahulu yang sangat bermanfaat dan dapat penulis gunakan sebagai rujukan. Berikut merupakan penelitian terdahulu yang berupa beberapa jurnal yang terkait dengan tugas akhir yang dilakukan oleh penulis.

**Tabel 3.** Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama**  **Peneliti** | **Judul**  **Penelitian** | **Hasil** |
| 1. | Evan Rosiska & Puspita Rama Nopiana, 2018 | Aplikasi Sistem Informasi Akuntansi Jasa Penyewaan Kapal Berbasis Web Pada PT Sikumbang Permata Engineering | Hasil dari penelitian adalah peneliti membangun aplikasi sistem informasi yang dapat memberikan kemudahan dalam melakukan penyewaan. Perusahaan dapat memberikan halaman web sehingga penyewa dapat melihat spesifikasi kapal yang akan disewa. Proses penyewaan kapal dalam negosiasi harga kapal sampai kontrak penyewaan siap dilakukan, sehingga pihak penyewa dapat membayar melalui rekening yang telah ditentukan atau datang langsung ke perusahaan. |
| 2. | Intan Septavia, Rd.Erwin Gunadhi & Rina Kurniawati | Sistem Informasi Penyewaan Mobil Berbasis Web Di Jasa Karunia *Tour And Travel* | Hasil dari penelitian adalah penulis melakukan perencanaan perancangan sistem informasi yang akan menghasilkan aplikasi ini yang dapat memberikan informasi secara detail dan akurat mengenai mobil-mobil yang disewakan oleh jasa karunia tour dan travel. |
| 3. | Sukmawati & Didi Susianto | Perancangan Sistem Pemesanan E-Tiket Pada Wisata Di Lampung Berbasis Web Mobil | Hasil dari penelitian adalah penulis memudahkan pelanggan untuk mendapatkan mengetahui informasi wisata secara Online, memberikan informasi yang cepat mengenai info lokasi dan informasi yang tersedia pada wisata yang ada di Lampung yang bermanfaat bagi wisatawan. |
| 4. | Muh Fakhrurrozi Yasin, Hero Wintolo, & Astika Ayuningtyas | Aplikasi Penyewaan Mobil Berbasis Android ( Studi Kasus : Tom Transport) | Hasil dari penelitian ini adalah penulis membuat aplikasi penyewaan aplikasi berbasis Android admin dapat melihat proses pemesanan, laporan penyewaan mobil, memproses transaksi data *customer* atau konsumen dalam aplikasi android. |
| 5. | Larissa Navia Rani | Perancangan Aplikasi Sistem Informasi Sebagai Media Promosi Menggunakan Bahasa Pemrograman PHP dan *Database* *MySQL* Pada Kinali Auto Rental | Hasil dari penelitian ini adalah penerapan sistem informasi dapat membantu pelanggan mendapatkan informasi rental mobil dengan cepat dan tepat dan memberikan kemudahan bagian *administrator* dan servis personel dengan informasi pelanggan yang lebih baik dan lengkap. Aplikasi ini di rancang dengan bahasa pemrograman PHP dan MySQL sebagai *database*  dari aplikasi tersebut*.* |

# BAB III ANALISIS DAN PERANCANGAN

1. Analisis Sistem

Dalam pengembangan sebuah sistem ada beberapa hal yang harus kita perhatikan terlebih dahulu, untuk itu diperlukan adanya analisis sistem yang akan dibuat dan akan dikembangkan.

1. Analisis Sistem Berjalan

Analisis sistem yang berjalan bertujuan memberikan gambaran tentang saat ini yang bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui. Adapun analisis sistem yang berjalan saat ini sebagai berikut:

1. Analisis Proses Bisnis/Aktivitas

Analisis proses bisnis adalah suatu kumpulan aktivitas atau pekerjaan terstruktur yang saling terkait untuk menyelesaikan suatu masalah tertentu atau yang menghasilkan produk atau layanan. Untuk lebih jelasnya proses bisnis yang berjalan saat ini dapat dilihat pada tabel berikut:

**Tabel 4.** Analisis Proses Bisnis/Aktivitas

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Proses Bisnis** | **Rincian Aktivitas** | **Pelaku Terkait** |
|  | Pengelolaan Pendaftaran  Penyedia Sewa/Jasa | * Calon penyedia sewa/jasa datang ke tempat pengelola sewa/jasa. * Calon penyedia sewa/jasa menginformasikan ingin mendaftar bergabung sebagai Member ke panitia sewa/jasa. * Calon penyedia sewa/jasa meminta formulir pendaftaran. * Panitia pengelola sewa/jasa memberikan formulir pendaftaran beserta rincian persyaratan berupa berkas-berkas yang harus di lengkapi. * Calon penyedia sewa/jasa mengisi formulir pendaftaran. * Calon penyedia sewa/jasa melengkapi berkas persyaratan. * Fotocopy Kartu Keluarga * Fotocopy KTP/SIM * Foto barang yang akan disewakan atau jasa yang ditawarkan. * Pas foto warna 3x4 (2 foto) * Calon penyedia sewa/jasa menyerahkan dokumen persyaratan. * Panitia mencatat data calon penyedia sewa/jasa dan membuat surat tanda pendaftaran. * Calon penyedia sewa/jasa menerima kartu tanda bukti pendaftaran. * Panitia mengarsipkan berkas-berkas surat pendaftaran. | * Panitia Pengelola Sewa/Jasa * Calon Penyedia Sewa/Jasa |
|  | Pengelolaan  Pendaftaran Barang Sewaan | * Penyedia sewa datang ke lokasi panitia pengelola sewa untuk melakukan pendaftaran barang sewaannya. * Penyedia sewa meminta fotocopy surat izin usaha tempat usaha (SITU) ke pengelola sewa. * Pengelola memberikan fotocopy surat izin usahanya. * Penyedia sewa memperlihatkan barang yang akan di titip sewakan ke panitia pengelola sewa. * Panitia pengelola sewa mengecek kondisi barang yang diberikan. * Panitia memberikan formulir pendaftaran khusus titip sewa barang. * Penyedia sewa mengisi formulir pendaftaran barang sewa. * Penyedia sewa memberikan STNK terkhusus untuk barang sewaannya adalah Mobil atau Honda. * Penyedia sewa menyerahkan dokumen persyaratan. * Pengelola sewa menerima bukti pendaftaran. * Panitia pengelola sewa mencatat data barang yang di sewakan oleh penyedia sewa. * Panitia pengelola sewa membuat bukti/surat tanda pendaftaran. * Penyedia sewa menerima kartu bukti pendaftaran. * Pengelola sewa mengarsipkan surat berkas-berkas dokumen dari penyedia sewa. | * Panitia Pengelola Sewa * Penyedia Sewa |
|  | Pengelolaan  Pendaftaran Jasa | * Penyedia jasa datang ke lokasi panitia pengelola jasa untuk melakukan pendaftaran jasanya. * Penyedia jasa meminta fotocopy surat izin usaha tempat usaha (SITU) ke pengelola. * Pengelola memberikan fotocopy surat izin usahanya. * Penyedia jasa mempromosikan jasanya agar di daftarkan ke panitia pengelola jasa. * Panitia pengelola jasa menanyakan jasa apa yang akan ditawarkan oleh penyedia jasa. * Panitia pengelola jasa memberikan formulir pendaftaran khusus penyedia jasa. * Penyedia jasa mengisi formulir pendaftaran. * Penyedia jasa menyerahkan dokumen persyaratan pendaftaran. * Pengelola Jasa menerima bukti pendaftaran. * Panitia pengelola mencatat data jasa yang di tawarkan oleh penyedia jasa. * Panitia Pengelola membuat bukti/surat tanda pendaftaran. * Penyedia sewa menerima kartu bukti pendaftaran. * Pengelola sewa mengarsipkan surat berkas-berkas dokumen dari penyedia jasa. | * Panitia Pengelola Jasa * Penyedia Jasa |
|  | Pengelolaan Informasi Promosi Sewa / Jasa | * Panitia pengelola sewa / jasa mengecek data barang yang di titip sewakan dan jasa yang ditawarkan penyedia. * Panitia pengelola sewa / jasa menginformasikan barang/jasa kepada tetangga. * Panitia pengelola sewa / jasa membuat iklan tersebut dalam bentuk spanduk/pamflet atau selebaran sesuai dengan ketersediaan barang sewaan dan jasa yang tersedia. * Panitia pengelola sewa / jasa memasang iklan tersebut dipasang pada tempat-tempat strategis, sehingga dapat dilihat oleh orang banyak. * Panitia pengelola sewa / jasa mengevaluasi iklan barang/jasa miliknya. | * Panitia Pengelola Sewa/Jasa |
|  | Pengelolaan Pemesanan Penyewaan Barang | * Calon penyewa barang mendengar dan melihat iklan tentang barang yang telah dipasang. * Calon penyewa mencari informasi lebih lanjut mengenai barang yang diinginkan. * Calon penyewa menghubungi nomor kontak yang tertera pada iklan. * Calon penyewa pergi ke tempat pengelola barang sewa. * Calon penyewa melakukan survei terhadap barang yang ingin disewanya. * Panitia pengelola sewa barang memperlihatkan barang yang ingin disewa oleh calon penyewa. * Calon penyewa melakukan kesepakatan dengan pengelola sewa mengenai harga dan lama barang akan di sewa. * Calon penyewa meminta formulir penyewaan barang. * Panitia pengelola sewa memberikan formulir penyewaan beserta rincian persyaratan berupa berkas-berkas yang harus di lengkapi. * Calon penyewa barang mengisi formulir penyewaan. * Calon penyewa barang melengkapi berkas persyaratan. * Kartu Keluarga Asli * Fotocopy KTP/SIM * Pas foto warna 3x4 (1 foto) * Calon penyewa barang menyerahkan dokumen persyaratan. * Panitia pengelola mencatat data calon penyewa barang dan membuat surat tanda penyewaan. * Penyewa barang menerima kartu tanda bukti penyewaan. * Panitia memberikan barang yang disewakan. * Penyewa menerima barang yang disewakannya. * Pengelola sewa mengarsipkan surat berkas-berkas dokumen dari penyewa barang. | * Calon Penyewa Barang * Panitia Pengelola Sewa Barang |
|  | Pengelolaan Pemesanan Penyewaan Jasa | * Calon penyewa jasa mendengar dan melihat iklan tentang jasa yang telah dipasang. * Calon penyewa mencari informasi lebih lanjut mengenai jasa yang diinginkan. * Calon penyewa menghubungi nomor kontak yang tertera pada iklan. * Calon penyewa pergi ke tempat pengelola jasa. * Calon penyewa melakukan survei terhadap jasa yang ingin disewanya. * Panitia pengelola sewa jasa menjelaskan jasa yang ingin disewa oleh calon penyewa. * Calon penyewa melakukan kesepakatan dengan pengelola sewa mengenai harga jasa akan di sewa. * Calon penyewa meminta formulir penyewaan jasa. * Panitia pengelola sewa memberikan formulir penyewaan. * Calon penyewa jasa mengisi formulir penyewaan. * Calon penyewa jasa menyerahkan dokumen formulir. * Panitia pengelola mencatat data calon penyewa jasa dan membuat surat tanda penyewaan. * Penyewa jasa menerima kartu tanda bukti penyewaan. * Panitia menelepon orang penyedia jasa yang bersangkutan. * Penyewa jasa datang ke tempat pengelola jasa. * Penyewa jasa dan penyedia jasa pergi ke tempat penyewa untuk melakukan tugasnya. * Pengelola sewa mengarsipkan surat berkas-berkas dokumen dari penyewa. | * Calon Penyewa Jasa * Panitia Pengelola Jasa |
|  | Pengelolaan Pembayaran Pemesanan | * Calon penyewa memesan barang / jasa yang tersedia. * Panitia pengelola melihat stok barang/jasa yang tersedia. * Panitia pengelola barang/jasa menginformasikan stok kepada penyewa. * Panitia pengelola memberikan surat formulir data pemesanan. * Calon penyewa mengisi formulir data pemesanan. * Calon penyewa menyerahkan formulir data pemesanan. * Panitia pengelola memberikan detail data bukti pemesanan. * Penyewa menerima detail data bukti pemesanan. * Penyewa melakukan pembayaran sesuai dengan harga yang telah ditetapkan. * Pengelola sewa mengarsipkan surat berkas-berkas dokumen dari penyewa. | * Calon Penyewa Barang/Jasa * Panitia Pengelola Barang/Jasa |
|  | Pengelolaan Pembatalan Sewa Barang/ Jasa | * Penyewa menghubungi pihak panitia penyewaan. * Panitia pengelola melakukan pembatalan penyewaan barang/ jasa yang telah di sewa. * Panitia penyewaan menghapus pemesanan barang/jasa yang telah dicatat. * Proses selesai. | * Penyewa Barang/Jasa * Panitia Pengelola Barang/Jasa |
|  | Pengelolaan Komplain Sewa Barang/Jasa | * Penyewa menghubungi pihak panitia penyewaan. * Penyewa mengajukan keluhan terhadap produk yang di sewa. * Panitia pengelola mencatat keluhan dari penyewa. * Panitia pengelola mencarikan solusi atas permasalahan yang dikeluhkan. * Panitia pengelola membuat surat keluhan dan di arsipkan. | * Penyewa Barang/Jasa * Panitia Pengelola Barang/Jasa |
|  | Pengelolaan Penilaian Pelayanan Oleh Penyewa | * Setelah proses penyewaan selesai panitia pengelola memberikan kuesioner pelayanan. * Penyewa mengisi kuesioner. * Panitia pengelola merangkap data kepuasan penyewa terhadap layanan yang disediakan, * Panitia pengelola mengarsipkan kuesioner data penilaian penyewa. | * Penyewa Barang/Jasa * Panitia Pengelola Barang/Jasa |
|  | Pengelolaan Pendaftaran Penyedia Sewa Barang  Secara Online | * Calon penyedia sewa menghubungi nomor kontak pengelola sewa. * Calon penyedia sewa menginformasikan barang yang akan disewakannya dipasang iklan oleh pengelola sewa. * Panitia pengelola sewa mengirimkan formulir beserta berkas-berkas dokumen yang diperlukan melalui *WhatsApp*. * Calon penyedia sewa mengisi formulir pendaftaran. * Calon penyedia sewa melengkapi berkas persyaratan. * *Soft File* Kartu Keluarga * *Soft File* KTP/SIM * *Soft File* Foto barang yang akan disewakan atau jasa yang ditawarkan. * *Soft File* Pas foto warna 3x4 (2 foto) * Calon penyedia sewa mengirimkan berkas dokumen ke *WhatsApp* pengelola sewa. * Pengelola sewa menerima *WhatsApp* dan mengirimkan kembali surat bukti pendaftaran. * Calon penyedia menerima surat bukti pendaftaran. | * Calon Penyedia Sewa Online * Panitia Pengelola Barang |
|  | Pengelolaan Pengiriman Barang Secara Online | * Setelah melakukan pendaftaran penyedia sewa Online kembali menginformasikan produk barang yang akan di daftarkannya ke panitia pengelola sewa. * Panitia pengelola sewa meminta detail barang dan harga yang akan disewakan. * Penyedia sewa mengirimkan foto-foto barang dan detail spesifikasi barang serta harga yang akan di sepakati untuk di sewakan. * Setelah sepakat keduanya penyedia sewa akan mengirimkan barangnya melalui pengiriman barang Online *JNE Express*. * Penyedia sewa mengirimkan resi pengimanan barang ke panitia pengelola barang. * Panitia pengelola menerima barang dari *JNE Express.* * Panitia pengelola mendaftarkan data barangnya. * Panitia pengelola sewa menginformasikan ke penyedia sewa barang bahwa barangnya telah di terima dan didaftarkan. * Panitia pengelola sewa membuat surat bukti penerimaan barang. * Panitia pengelola sewa mengarsipkan berkas faktur sewa. | * Penyedia Sewa Online * Panitia Pengelola Barang |
|  | Pengelolaan Deposit | * Penyewa mengisi saldo deposit awal minimal sebesar Rp.100.000. * Panitia pengelola sewa menerima uang deposit yang telah di terima dari penyewa. * Deposit bisa digunakan penyewa untuk melakukan pembayaran. * Penyewa yang menggunakan pembayaran dengan deposit akan mendapatkan diskon 5%. * Panitia mendapatkan saldo deposit tambahan dari pembayaran penyewa dengan saldo deposit. * Panitia pengelola membukukan total deposit penyewa dengan pengelola. * Panitia pengelola sewa membuat surat faktur bukti deposit. * Penyewa menerima surat faktur bukti deposit. * Panitia pengelola sewa mengarsipkan berkas faktur pembayaran deposit. | * Penyewa Barang/Jasa * Panitia Pengelola Barang |

1. Analisis Aturan Bisnis (*Business Rule*)

Analisis aturan bisnis (*Business Rule*) adalah sebuah pernyataan yang menjelaskan kebijakan bisnis atau keputusan prosedur. Untuk lebih jelasnya dapat dilihat pada tabel berikut:

**Tabel 5.** Analisis Aturan Bisnis

|  |  |  |
| --- | --- | --- |
| **No.** | **Pelaku Bisnis** | **Aturan** |
|  | Panitia Pengelola Sewa Barang | * Pengelola harus mengenal jenis dan kategori penyewaan barang/jasa apa saja yang disewakan. * Pengelola harus memenuhi pemesanan penyewa sesuai data pemesanan penyewa. * Lokasi penyewaan yang strategis dan penataan barang yang baik sehingga penyewa tertarik untuk menyewa barang di toko. * Pengelola harus memiliki surat izin usaha. * Pengiriman barang harus pihak yang sah baik manajemen penyewa maupun pihak ketiga. |
|  | Penyedia Sewa Barang / Jasa | * Penyedia sewa harus mengenal jenis dan kategori penyewaan barang/jasa apa saja yang disewakan. * Penyedia harus menyediakan barang/jasa pemesanan penyewa sesuai data pemesanan penyewa. |
|  | Penyewa Barang / Jasa | * Penyewa harus menyebutkan / mencantumkan alamat tempat tinggal penyewa yang jelas dan lengkap. * Penyewa harus mengetahui jenis barang/jasa yang dicari agar mempermudah pengelola dalam mencari barang/jasa sewa yang diinginkan. * Penyewa harus memiliki uang yang cukup ketika melakukan transaksi. |

1. Analisis Pelaku Bisnis

Pelaku bisnis merupakan orang yang terlibat dalam sistem beserta fungsi dan tugasnya masing-masing. Untuk lebih jelasnya dapat dilihat pada tabel berikut:

**Tabel 6.** Analisis Pelaku Bisnis

|  |  |  |
| --- | --- | --- |
| **No.** | **Pelaku Bisnis** | **Keterangan** |
|  | Penyewa Barang / Jasa | * Bagian terpenting dalam usaha yang memberikan perhatian penuh terhadap produk barang/jasa yang ditawarkan dan melakukan penyewaan secara berulang-ulang |
|  | Penyedia Sewa Barang / Jasa | * Pemilik barang/jasa yang menyediakan sewaan yang ingin di titip sewakan sesuai jenis dan kategori. |
|  | Panitia Pengelola Sewa Barang | * Pemilik ataupun pengelola toko yang menyediakan barang/jasa sewaan yang ingin disewakan sesuai jenis dan kategori. * Bertanggung jawab penuh atas semua kelangsungan usaha. * Panitia pengelola melakukan proses tawar menawar dengan calon pembeli jika harga sewa barang/jasa tidak dipatok. * Panitia pengelola dapat menetapkan harga tiap barang/jasa sewa yang tersedia di toko untuk dipasarkan dan diiklankan. |

1. Analisis Dokumen I/O

Analisis dokumen terkait merupakan analisis terhadap dokumen yang dimasukkan (dokumen *input*) dan dokumen yang dihasilkan (dokumen *output*).

1. **Dokumen *Input***

Dokumen *input* merupakan dokumen yang dimasukkan oleh pengguna ke dalam sistem. Untuk lebih jelasnya akan dibahas sebagai berikut:

**Tabel 7.** Analisis Dokumen *Input*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Dokumen** | **Keterangan** | **Pelaku Terkait** |
|  | Formulir Pendaftaran Penyedia Sewa | * Dokumen yang berisi tentang biodata penyedia sewa berupa nama, alamat, tempat tinggal, No. *WhatsApp,* No. handphone. | * Panitia Pengelola Sewa * Calon Penyedia Sewa |
|  | Formulir Pendaftaran Barang Sewa | * Dokumen yang berisi tentang detail dan spesifikasi barang yang akan di titip sewakan kepada pengelola sewa. | * Panitia Pengelola Sewa Barang * Calon Penyedia Sewa Barang |
|  | Formulir Pendaftaran Sewa Jasa | * Dokumen yang berisi tentang detail dan jasa apa yang akan disediakan kepada pengelola sewa. | * Panitia Pengelola Sewa Jasa * Calon Penyedia Sewa Jasa |
|  | Formulir Data Pemesanan | * Berisi nama lengkap penyewa, alamat lengkap, nomor yang bisa dihubungi, jaminan. | * Panitia Pengelola Sewa * Calon Penyewa |
|  | Formulir Data Transaksi | * Berisi lama penyewaan pemilihan jenis pembayaran. | * Panitia Pengelola Sewa * Penyewa |
|  | Formulir Pembatalan Pemesanan | * Berisi pembatalan pemesanan | * Panitia Pengelola Sewa * Penyewa |
|  | Penilaian Layanan | * Berisi komentar kepuasan atau keluhan dari barang/jasa yang disewa. | * Penyewa |

1. **Dokumen *Output***

Dokumen *output* merupakan dokumen yang dihasilkan oleh sistem setelah melakukan proses. Dokumen *output* akan menghasilkan informasi yang bermanfaat untuk pengguna.

**Tabel 8.** Analisis Dokumen *Output*.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Dokumen** | **Keterangan** | **Pelaku Terkait** |
|  | Informasi barang dan jasa yang akan disewakan. | * Merupakan dokumen yang memberikan informasi tentang barang dan jasa yang ingin disewakan. | * Semua Pelaku |
|  | Daftar kategori | * Dokumen yang memberikan informasi barang berdasarkan kategori yang dipilih | * Semua Pelaku |
|  | Informasi profil | * Merupakan dokumen yang memberikan informasi biodata setiap pelaku, baik itu pengelola penyedia sewa, maupun penyewa mempunyai biodata sendiri. | * Penyewa * Penyedia Sewa * Pengelola Sewa |
|  | Laporan pemesanan | * Merupakan dokumen yang berisi detail pemesanan dari penyewa. | * Penyedia Sewa |
|  | Bukti pemesanan | * Dokumen yang berisi data barang dan jumlah harga yang harus dibayar penyewa. | * Penyewa |
|  | Laporan pembatalan pemesanan | * Dokumen yang berisi pembatalan pemesanan | * Penyewa |
|  | Informasi Penilaian Layanan | * Berisi informasi keluhan dan kepuasan dari barang dan jasa | * Semua Pelaku |

1. Analisis Masalah dan Solusi

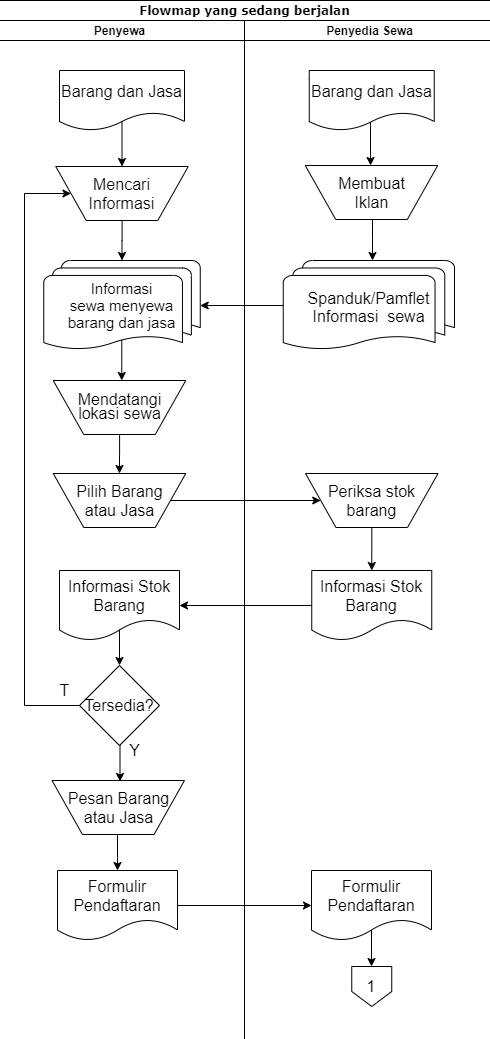
Analisis permasalahan dan solusi merupakan penganalisisan terhadap permasalahan yang terjadi di lapangan dan solusi yang diberikan untuk menyelesaikan permasalahan tersebut. Pada sistem ini, permasalahan dan solusinya adalah sebagai berikut:

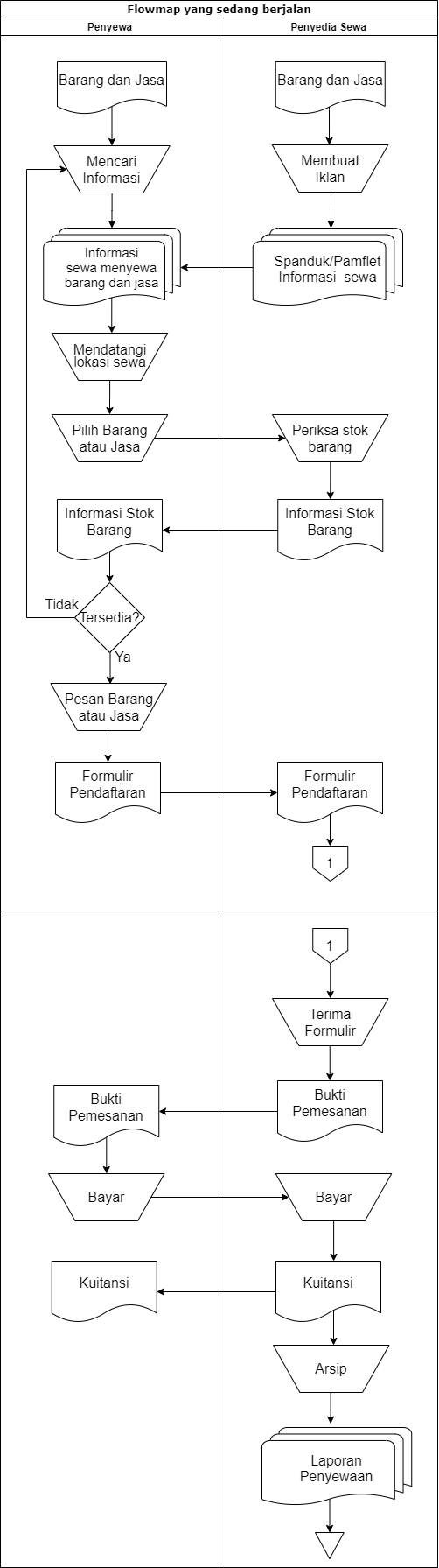
**Tabel 9.** Analisis Masalah dan Solusi.

|  |  |  |
| --- | --- | --- |
| **No.** | **Masalah** | **Solusi** |
|  | Terbatasnya informasi yang didapatkan secara konvensional. | Sistem memberikan informasi yang lebih lengkap mulai dari lokasi, jasa sewa, fasilitas, harga, foto kondisi barang atau jasa yang diberikan. |
|  | Dalam mencari barang dan jasa yang ingin disewa, bisa dilakukan dengan cara mencari atau menanyakan secara langsung jasa sewa yang dibutuhkan. | Sistem ini menyediakan formulir pencarian barang dan jasa yang ingin disewakan di kota Payakumbuh. Penyewa cukup duduk di rumah atau di mana saja dengan menggunakan perangkat seluler atau PC. |
|  | Banyaknya media informasi untuk mempromosikan barang sehingga informasi yang dihasilkan tersebar. | Sistem menyediakan media promosi barang dan jasa yang ditawarkan untuk di informasikan kepada masyarakat yang membutuhkan dalam satu aplikasi. |
|  | Dibutuhkan waktu yang lama karena harus bertanya sana sini untuk mencocokkan harga dengan ketersediaan dana untuk menyewa. | Sistem memberikan pilihan harga dari jenis barang yang sama sehingga pengguna hanya perlu memilih yang sesuai dengan *budget* yang dimiliki. |
|  | Pendaftaran dilakukan menggunakan kertas yang rentan hilang atau rusak. Selain itu penyewa harus mem*fotokopi* jaminan KTP/SIM setiap kali melakukan penyewaan dengan barang yang berbeda. | Sistem menyediakan pendaftaran terkomputerisasi yang datanya tersimpan di dalam *database* dan juga *upload* jaminan digital yang bisa digunakan untuk banyak jenis penyewaan barang. |

1. Flow Map Sistem Berjalan

Menggambarkan diagram atau bagan prosedur *system* yang sedang berjalan. Diagram hendaknya memperlihatkan *input*, proses dan *output* dari setiap prosedur dalam *system* yang sedang berjalan. Untuk lebih jelasnya, sistem penyewaan yang berjalan saat ini dapat digambarkan pada *flow*-*map* berikut:





**Gambar 4.** *Flow-map* sistem saat ini

Sistem yang sedang berjalan pada saat pencarian informasi sewa barang dan jasa di mulai dari pembuatan iklan dalam bentuk spanduk/pamflet atau selebaran oleh penyedia sewa, kemudian iklan tersebut di pasang pada tempat-tempat yang strategis. Penyewa yang membutuhkan barang atau jasa untuk disewakan mencari informasi dengan bertanya atau melihat di tempat-tempat tertentu. Setelah mendapatkan informasi yang dibutuhkan, penyewa mendatangi lokasi dengan melihat barang atau jasa tersebut dan menyesuaikan harga, jika harga tidak sesuai maka penyewa mencari informasi lain. Jika harga sesuai maka penyewa memesan barang atau jasa yang dibutuhkan. Selanjutnya, penyedia sewa melihat stok barang dan menginformasikan kepada penyewa. Jika barang tersedia, maka penyewa mengisi formulir pemesanan. Formulir tersebut di arsipkan oleh penyedia sewa kemudian penyedia sewa memberikan bukti pemesanan. Selanjutnya, penyewa membayar dan membawa barang yang disewakan. Kuitansi pembayaran diarsipkan oleh penyedia sewa menjadi data pemesanan.

Berdasarkan analisis permasalahan dan solusi, maka dibuatlah sebuah sistem yang dapat menjadi media untuk dalam penerapan solusi dari permasalahan yang ada.

1. Sistem Yang Disusulkan

Setelah menganalisis sistem berjalan, maka analisis sistem yang diusulkan sebagai berikut:

1. Analisis *User*

Pada sistem ini melibatkan 5 (lima) tingkat (level) pengguna dalam sistem ini yaitu Admin, *viewer/public*, penyedia sewa, penyewa, dan *member*. Setiap pengguna memiliki tugas dan fungsi yang berbeda di dalam sistem. Untuk lebih jelasnya dapat pada tabel berikut ini:

**Tabel 10.** Analisis *User*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama *User*** | **Aktivitas** | **Dokumen Terkait** | |
| ***Input*** | ***Output*** |
| 1. | Admin | Mengelola sistem, validasi pengguna, memblokir pengguna. | Data konfirmasi pengguna, data blokir pengguna dan testimony, data kategori,  data, data FAQ, *about,* ketentuan, *user guide* | Informasi pengguna, Informasi kategori, Informasi *store*, Informasi barang, ketentuan aplikasi, informasi FAQ. |
| 2 | *Viewer/ public* | Mencari barang/jasa yang akan disewakan, melihat barang atau jasa yang tersedia sesuai kategori, melihat profil toko, melakukan registrasi, memberikan *feedback* aplikasi. | Data pencarian, form registrasi, form *feedback.* | Informasi barang dan jasa, *rating*, *testimony*, daftar kategori, konfirmasi registrasi, informasi profil penyedia sewa, *user guide.* |
| 3 | Penyedia sewa | Melakukan *login* ke sistem, mengubah biodata, membuat *store*, menambahkan barang/ jasa, menerima dan menolak pemesanan, melihat data penyewa, memberikan *feedback* aplikasi. | *Form login, form* edit *profil, form* barang/jasa, *form store,* data penerimaan pemesanan, *form feedback.* | Informasi barang/jasa, *profil,* dokumen pemesanan, bukti transfer, data *rating*, *testimony*, biodata penyewa/ *member*, *user guide*. |
| 4 | Penyewa | Melakukan *login* ke dalam sistem, mengubah biodata, mencari barang/jasa, melihat barang/jasa berdasarkan kategori, melihat profil penyedia sewa, melakukan pemesanan, membatalkan pemesanan, memberi rating, memberi *testimony*, memberikan *feedback* aplikasi | *Form login, form edit profil, form* pemesanan, *form* pembatalan, *form transaksi,* *form rating*, *testimony, form feedback.* | Informasi barang/jasa, *profil,*  biodata *store*, bukti pemesanan, *user guide.* |
| 5 | *Member* | Melakukan *login* ke dalam sistem, mengubah biodata, mencari barang/jasa, melihat barang/jasa berdasarkan kategori, melihat profil penyedia sewa, melakukan pemesanan, membatalkan pemesanan, memberi rating, memberi *testimony*, memberikan *feedback* aplikasi. | *Form login, form edit profil, form* transaksi, *form* pembatalan, *form rating*, *testimony, form feedback.* | Informasi barang/jasa, *profil,* biodata penyedia sewa, bukti pemesanan, *user guide.* |

1. Analisis Proses dan Prosedur

Analisis prosedur atau proses sistem, memberikan gambaran tentang sistem yang saat ini sedang berjalan. Analisis sistem bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui. Pada analisis prosedur ini dijelaskan beberapa langkah dan aktivitas serta mekanisme yang sedang berjalan.

**Tabel 11.** Analisis Prosedur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Aktivitas** | ***User* Terkait** | **Proses** | **Dokumen Terkait** |
| 1. | Pendaftaran | *Viewer/*  *public* | *Viewer/public* yang ingin melakukan proses penyewaan barang atau ingin memberikan layanan penyewaan harus melakukan pendaftaran terlebih dahulu ke dalam sistem. | *Form* pendaftaran |
| 2. | *Login* | Semua *user* | Setelah melakukan pendaftaran, pengguna dapat masuk ke dalam sistem sesuai dengan data yang dimasukkan saat *registrasi.* | *Form* registrasi |
| 3. | Mengubah biodata | Penyewa, *member*, penyedia sewa | Setelah melakukan *login,* pengguna bisa mengubah atau melengkapi data yang belum lengkap. | *Form* biodata |
| 4. | Menginformasikan barang/jasa | Penyedia sewa | Penyedia sewa akan mengisi beberapa *form* seperti harga barang, stok barang, deskripsi barang, bentuk/gambar barang. | *Form* barang / jasa, informasi barang dan jasa. |
| 5. | Pencarian barang/jasa | *Viewer/ public, member,* penyewa | *User* akan membuka aplikasi dan mencari barang atau jasa yang dibutuhkan. pencarian dapat dilakukan berdasarkan nama atau kategori dari barang dan jasa tersebut. | *Form* pencarian, data barang/jasa, informasi barang dan jasa, daftar kategori, biodata penyewa, rating, *testimony.* |
| 6. | Melakukan pemesanan barang/jasa | Penyewa atau  *Member* | Setelah menemukan barang yang ingin disewakan, pengguna dapat memesan barang/jasa tersebut yang nantinya akan di konfirmasi oleh penyedia sewa. | *Form* pemesanan, *form* transaksi, bukti pemesanan. |
| 7. | Pembatalan pemesanan | Penyewa atau  *Member,* penyedia sewa. | Jika pengguna salah melakukan pemilihan barang/jasa dan maka pengguna dapat melakukan pembatalan pemesanan. | *Form* pembatalan pemesanan. |
| 8. | Menerima pemesanan | Penyedia sewa | Penerimaan pemesanan dilakukan oleh penyedia sewa berdasarkan ketersediaan. | Profil penyewa atau member, data pemesanan. |
| 9 | Memberikan *rating* | Penyewa atau  *Member* | Pemberian *rating* dilakukan oleh penyewa atau *member* yang akan ditampilkan pada informasi barang dan jasa | *Form rating*, informasi barang/jasa. |
| 10. | *Testimony* | Penyewa atau *member* | *Testimony* dapat berisi kepuasan atau keluhan yang diisi oleh penyewa atau *member* berkaitan dengan barang dan jasa. | *Form testimony,* informasi barang/jasa. |
| 12. | *Block* pengguna | Admin | Jika ada pengguna yang menyalahi aturan dari aplikasi, Admin dapat memblokir pengguna tersebut. | *Form blokir,* data pengguna |
| 13 | *Logout* | Semua *user* | Setelah melakukan berbagai aktivitas di dalam sistem *user* dapat keluar dari sistem untuk menjaga keamanan data dan informasi. | Tombol *logout* |

1. Analisis Dokumen I/O

Analisis dokumen terkait merupakan analisis terhadap dokumen yang dimasukkan (dokumen *input*) dan dokumen yang dihasilkan (dokumen *output*).

1. **Dokumen *Input***

Dokumen *input* merupakan dokumen yang dimasukkan oleh pengguna ke dalam sistem. Untuk lebih jelasnya akan dibahas sebagai berikut:

**Tabel 12.** Analisis Dokumen *Input*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Dokumen** | ***User* Terkait** | **Keterangan** |
| 1. | Registrasi | *Viewer/public* | Dokumen yang berisi data pengguna berupa email*, username,* dan *password*. |
| 2. | Form *Login* | Semua *user* | Dokumen yang berisi data *email* dan *password* untuk masuk ke dalam sistem. |
| *3.* | *Data pencarian* | *Penyedia sewa, penyewa, member, viewer/public* | *Keyword yang di masukkan untuk mencari informasi barang/jasa berdasarkan nama atau kategori dari barang dan jasa.* |
| 4. | Data profil | Penyedia sewa, penyewa, *member* | Berisi biodata pengguna. |
| 5. | Data barang atau jasa | Penyedia sewa | Berisi nama barang/jasa, harga atau tarif penyewaan, stok barang, deskripsi lengkap barang dan jasa, gambar. |
| 6. | Data pemesanan | Penyewa | Berisi nama lengkap penyewa, alamat lengkap, nomor yang bisa dihubungi, jaminan. |
| 7. | Data transaksi | Penyewa dan *member* | Berisi lama penyewaan, pemilihan pengiriman dan pemilihan jenis pembayaran. |
| 8. | Pembatalan pemesanan | Penyewa dan *member* | Berisi pembatalan pemesanan |
| 9. | Data konfirmasi pemesanan | Penyedia sewa | Berisi konfirmasi penyewaan. |
| 10. | *Rating* | Penyewa dan *member* | Berisi *rating* berupa *rating* bintang. |
| 11. | *Testimony* | Penyewa dan *member*. | Berisi komentar kepuasan atau keluhan dari barang/jasa yang disewa. |

1. **Dokumen *Output***

Dokumen *output* merupakan dokumen yang dihasilkan oleh sistem setelah melakukan proses. Dokumen *output* akan menghasilkan informasi yang bermanfaat untuk pengguna.

**Tabel 13.** Analisis Dokumen *Output*.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Dokumen** | ***User* Terkait** | **Keterangan** |
| 1. | Informasi barang dan jasa yang akan disewakan | Semua *user* | Merupakan dokumen yang memberikan informasi tentang barang dan jasa yang ingin disewakan. |
| 2. | Daftar kategori | Semua *user* | Dokumen yang memberikan informasi barang berdasarkan kategori yang dipilih. |
| 3. | Informasi profil | Penyewa, penyedia sewa, *admin*, *member* | Merupakan dokumen yang memberikan informasi biodata setiap *user*, baik itu admin, penyedia sewa, penyewa maupun *member* mempunyai biodata sendiri. |
| 4. | Laporan pemesanan | Penyedia sewa | Merupakan dokumen yang berisi detail pemesanan dari penyewa atau *member.* |
| 5. | Bukti pemesanan | Penyewa dan *member* | Dokumen yang berisi data barang dan jumlah harga yang harus dibayar penyewa atau *member*. |
| 6. | Laporan pembatalan pemesanan | Penyewa dan *member* | Dokumen yang berisi pembatalan pemesanan. |
| 7. | Informasi *rating* | Semua *user* | Berisi informasi *rating* dari sebuah barang/jasa. |
| 8. | *Testimony* | Semua *user* | Berisi informasi keluhan dan kepuasan dari barang dan jasa. |

1. Analisis Persyaratan (*Requirement*)

Setelah mengetahui permasalahan pada sistem yang sedang berjalan, selanjutnya penulis melakukan analisis persyaratan (*requirement*) sebagai solusi masalah tersebut. Persyaratan yang ada dibagi menjadi 2 (dua) bagian. Bagian pertama adalah *functional* *requirement* yaitu aktivitas dan layanan yang harus disediakan oleh sistem yang akan dibangun. Bagian kedua adalah *non-functional requirement* yaitu fitur-fitur lain yang diperlukan oleh sistem agar sistem lebih maksimal penggunaannya.

1. ***Functional Requirements***

*Functional* *requirements* merupakan aktivitas dan layanan yang harus disediakan oleh sistem yang akan dibangun. Sistem yang dibangun harus mempunyai *functional* *requirements* sebagai berikut:

**Tabel 14.** Tabel *Functional Requirements*

|  |  |  |
| --- | --- | --- |
| **No.** | **Functional** | **Keterangan** |
|  | Pendaftaran | Sistem menyediakan layanan pendaftaran untuk masyarakat yang ingin menyewakan barang dan mencari barang untuk disewa. Pendaftaran ada 3 bentuk yaitu pertama, pendaftaran menjadi penyedia sewa untuk masyarakat yang mempunyai barang untuk disewakan. Kedua, pendaftaran menjadi penyewa untuk masyarakat yang membutuhkan barang atau jasa untuk disewa. Ketiga, pendaftaran menjadi *member* untuk penyewa yang ingin melakukan lebih banyak penyewaan. |
|  | Layanan pemasangan iklan barang/jasa | Sistem menyediakan pemasangan iklan yang dilakukan oleh masyarakat yang ingin menyewakan barang/jasa mereka dengan mengisi *form* yang berkaitan dengan barang dan jasa. |
|  | Layanan pencarian | Sistem menyediakan layanan pencarian informasi mengenai barang dan jasa yang bisa disewakan. Fitur pencarian dilakukan dengan meng*entry*kan nama dari barang atau jasa yang ingin disewakan, atau mencari barang dan jasa berdasarkan kategori yang disediakan di sistem |
|  | Pemesanan barang dan jasa | Sistem menyediakan layanan pemesanan barang/jasa dengan mengisi *form* pemesanan dan beberapa data yang terkait dengan pemesanan. Sistem juga melayani pengiriman antar/jemput serta pembayaran dilakukan di tempat atau transfer via rekening. |
|  | Pembatalan pemesanan | Sistem menyediakan pembatalan pemesanan yang dilakukan oleh pengguna. |
|  | Pemberian *rating* | *Rating* diberikan oleh pengguna atas kepuasan terhadap barang atau jasa. |
|  | *Testimony* | Sistem menyediakan fitur *testimony* yang berisi keluhan ataupun kepuasan yang dituangkan dalam bentuk tulisan. |
|  | Blokir pengguna | Sistem juga akan melakukan pemblokiran kepada pengguna yang menyalahi aturan dalam penggunaan aplikasi. |

1. ***Non-functional Requirements***

*Non-functional requirements* merupakan fitur-fitur lain yang diperlukan oleh sistem agar sistem yang dibangun lebih maksimal penggunaannya.

**Tabel 15.** Analisis *Non-Functional Requirement*

|  |  |  |
| --- | --- | --- |
| **No.** | **Non Functional** | **Keterangan** |
|  | Modul pendaftaran | Berisi email*, username,* dan *password* untuk melakukan pendaftaran ke dalam sistem. |
|  | Modul *login* | Form untuk masuk ke dalam sistem |
|  | Modul kategori | Berisi kategori-kategori yang bisa digunakan untuk pencarian barang dan jasa. |
|  | Modul pemesanan | Berisi barang dan jasa yang akan dipesan, profil penyewa, lama penyewaan, alamat penyewaan, nomor yang bisa dihubungi |
|  | Modul pembayaran | Berisi, jenis pengiriman, jenis pembayaran, jaminan. |
|  | Modul FAQ | Berisi kumpulan pertanyaan dan juga jawaban yang berkaitan dengan aktivitas yang dilakukan di dalam sistem. |
|  | Modul *user guide* | Berisi cara-cara menyewa dan menyewakan barang di dalam aplikasi. |
|  | Model tampilan | Tampilan yang menarik dan lebih *friendly* sehingga mudah dimengerti dan digunakan oleh pengguna. |
|  | Model penyimpanan data | Data disimpan dalam *database* untuk mencegah kehilangan atau kerusakan. |

1. ***Hardware Requirements***

Perangkat keras (*hardware*) yang diperlukan dalam menjalankan sistem yang akan dibangun. Sistem yang dibangun harus mempunyai hardware sebagai berikut:

**Tabel 16.** Analisis *Hardware* *Requirements*

|  |  |  |
| --- | --- | --- |
| **No.** | **Hardware** | **Merek/*Type*** |
|  | Laptop | Asus *Vivo Book* A442U |
|  | Processor | Intel Core i7- Generasi ke-8 |
|  | RAM | Minimal 8GB |
| 4. | *Harddisk* | Berkapasitas 1TB |
| 5. | *Smartphone* | Android versi 6.0 ke atas |

1. ***Software Requirements***

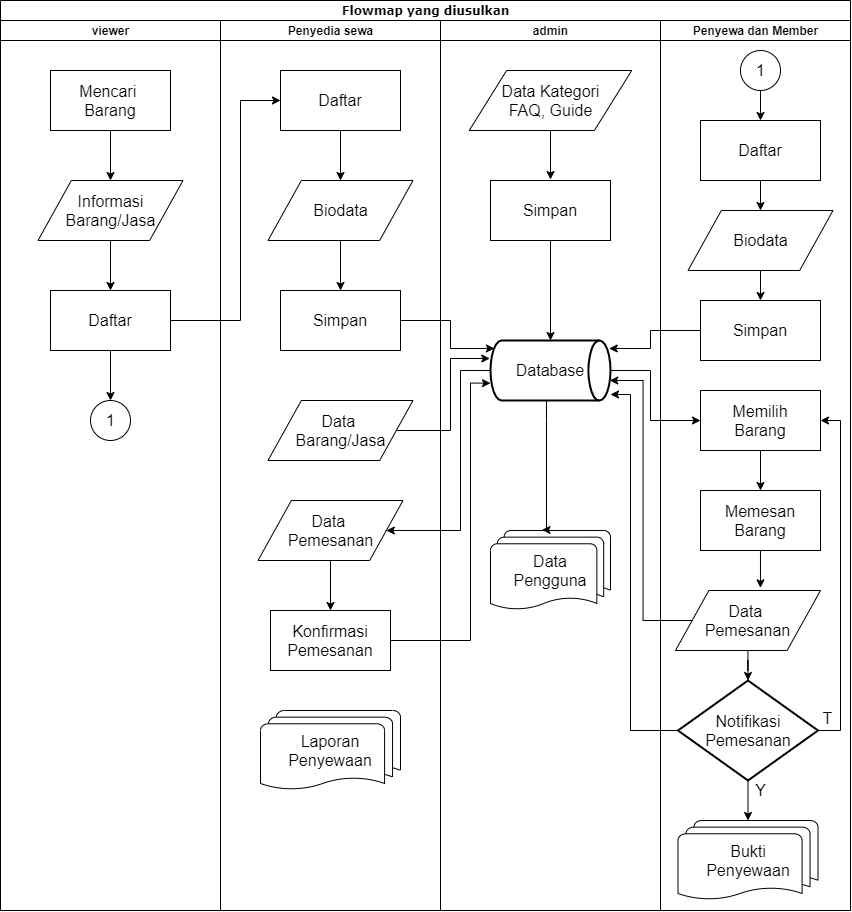
Perangkat lunak (*software*) yang diperlukan dalam menjalankan sistem yang akan dibangun. Sistem yang dibangun harus mempunyai software sebagai berikut:

**Tabel 17.** Analisis *Software Requirements*

|  |  |  |
| --- | --- | --- |
| **No.** | **Software** | **Fungsi Pengembangan** |
|  | Android Studio (Ver.4.0) | Digunakan untuk membuat *user* *interface* dengan XML sebagai *layout.* |
|  | Android SDK | *Tools* Pengembangan *Libraries* |
|  | Bahasa Pemrograman | Bahasa yang digunakan adalah bahasa Java, PHP dan Kotlin. |
| 4. | Adobe Photoshop | Untuk mendesain *interface* aplikasi yang akan dikembangkan seperti logo. |
| 5. | Emulator Android | Untuk menjalankan aplikasi Android secara virtual di laptop. |

1. Flow Map Diusulkan

Menggambarkan diagram atau bagan prosedur *system* yang akan diusulkan. Diagram hendaknya memperlihatkan *input*, proses dan *output* dari setiap prosedur dalam *system* yang sedang berjalan. Untuk lebih jelasnya, sistem penyewaan yang diusulkan dapat digambarkan pada *flow*-*map* berikut:



**Gambar 5.** *Flow-map* Sistem diusulkan

Proses yang diusulkan terjadi dimulai saat *viewer*/*public* mencari informasi barang atau jasa yang dibutuhkan. Ketika *viewer*/*public* mendapatkan barang yang ingin disewa, *viewer*/*public* melakukan pendaftaran terlebih dahulu. Pendaftaran dilakukan dengan satu akun dan bisa menjadi penyewa atau penyedia sewa. Jika *viewer* ingin menjadi penyewa ataupun penyedia sewa maka ada beberapa data yang harus di masukkan ke dalam sistem. Jika *viewer* sudah menjadi penyedia sewa, maka penyedia sewa harus membuat *store* dan bisa memasukkan data barang atau jasa ke dalam sistem. Data tersebut tersimpan di dalam *database* yang kemudian menjadi informasi bagi penyewa dan *member.* Setelah mendapatkan informasi dari barang dan jasa, penyewa dan *member* memilih barang yang disewa sesuai kebutuhan. Kemudian penyewa dan *member* memasukkan data penyewaan yang disimpan di *database*. Data penyewaan ini di konfirmasi oleh penyedia sewa. Jika status penyewaan tidak di konfirmasi, maka penyewa dan *member* harus memilih barang lain. Jika penyewaan di konfirmasi, maka penyewa dan *member* mendapatkan bukti penyewaan.

1. Perancangan Sistem

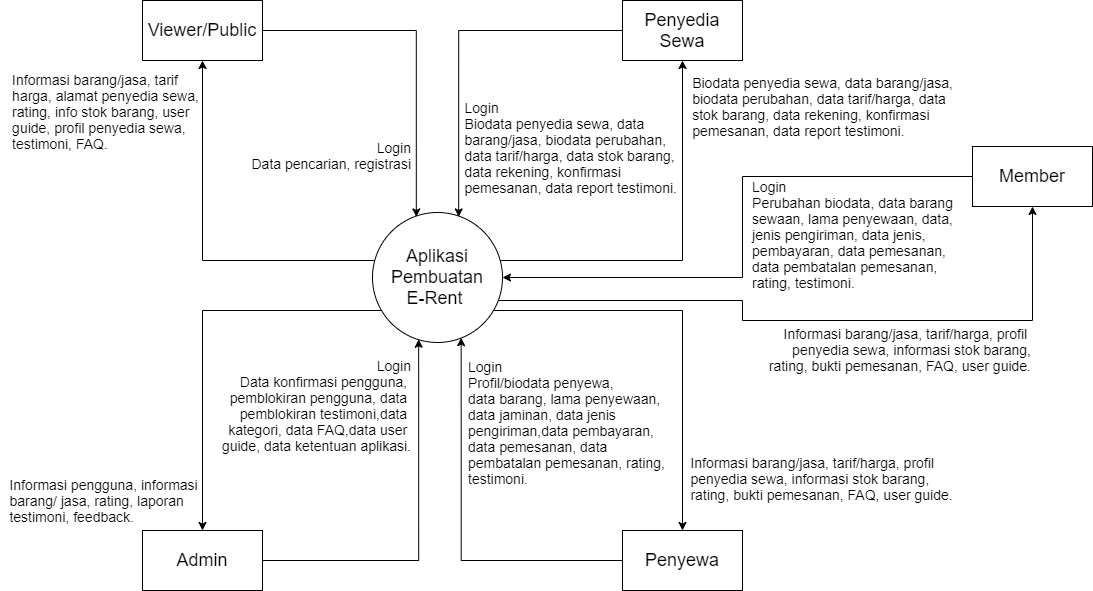
Tahapan selanjutnya adalah melakukan perancangan sistem yaitu untuk membuat pemodelan terhadap aplikasi/sistem sehingga dapat mengatasi masalah yang terdapat pada sistem yang berjalan saat ini, Perancangan sistem ini menggunakan diagram *Unified Modelling Language* (UML). Berikut perancangan *E-Rent* berbasis Android:

Pada perancangan aplikasi ini digambarkan model dari aplikasi yang akan dibuat melalui *diagram konteks, use case diagram, Activity Diagram dan Sequence Diagram* sebagai berikut:

1. *Context Diagram*

Diagram konteks (*context diagram*) merupakan bagan yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem yang akan dibangun. Secara uraian dapat dikatakan bahwa diagram konteks itu berisi siapa saja yang memberikan data masukkan ke sistem serta kepada siapa data informasi yang harus dihasilkan sistem. Diagram konteks berfungsi untuk memetakan bagaimana interaksi yang terjadi antara sistem dengan lingkungan luarnya.

Berdasarkan analisis pengguna (*user*), maka dapat digambarkan diagram konteks diagram konteks sistem pada *E-Rent* berbasis Android seperti terlihat pada gambar 6 di bawah ini :



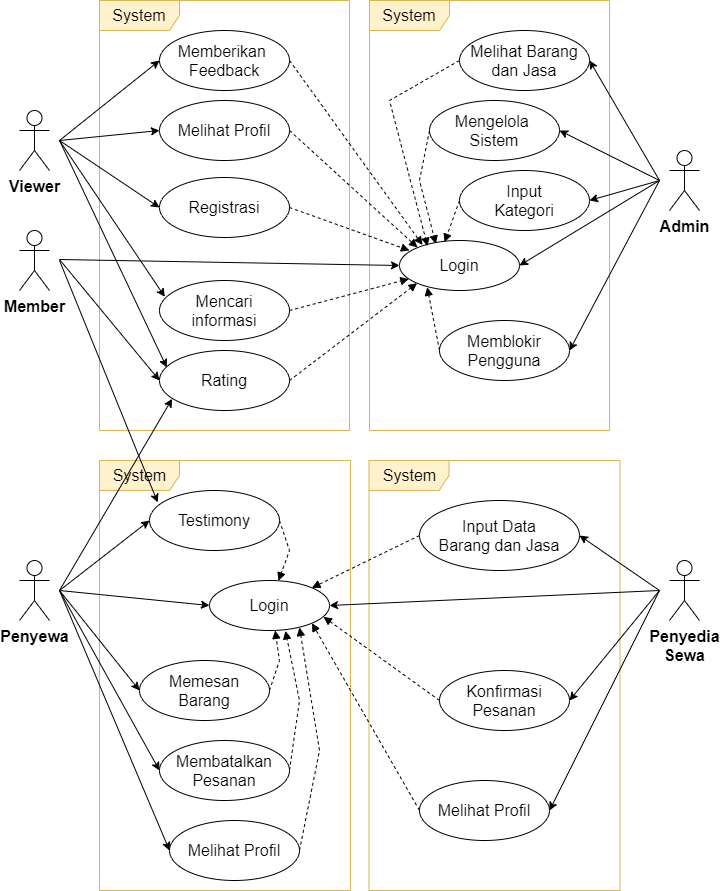
**Gambar 6.** *Context Diagram*

Diagram konteks pada Gambar 6. Menjelaskan tentang masukan dan keluaran yang dilakukan *user* kepada sistem. Administrator dapat memasukkan data yang berhubungan dengan aplikasi, dan mendapatkan keluaran berupa informasi dari *user. User* terdiri dari penyewa, penyedia sewa maupun *viewer/public* dan *member*. *Viewer/public* dapat mencari barang dan jasa yang dibutuhkan di sistem serta melakukan registrasi. Registrasi dilakukan berdasarkan kebutuhan, apakah, *viewer/public* melakukan registrasi sebagai penyedia sewa atau penyewa.

Penyedia sewa dapat memasukkan data barang dan jasa yang disewakan. Data ini diterima sebagai informasi oleh penyewa. Penyewa dapat melakukan pemesanan penyewaan di dalam sistem. Penyewa juga dapat melakukan pembatalan pemesanan. Jika penyewa ingin melakukan penyewaan lebih dari satu kali, penyewa juga bisa mendaftarkan diri sebagai *member.* Di sini perbedaannya, jika pengguna berstatus sebagai *member*, pengguna hanya perlu memasukkan lama sewa dan tidak perlu meng upload jaminan ke dalam sistem. Selain melakukan pemesanan, penyewa dan *member* juga memasukkan *rating* dan *testimony*. *Rating* dan *testimony* ini dimasukkan per jenis barang.

1. *Use case* *Diagram*

Perancangan *use case diagram* untuk aplikasi *E-Rent* berbasis Android memperlihatkan interaksi dari aktor yang ada yaitu Admin, *viewer*, *member*, penyewa, dan penyedia sewa. Selengkapnya dapat dilihat pada gambar 7.



**Gambar 7.** *Use Diagram*

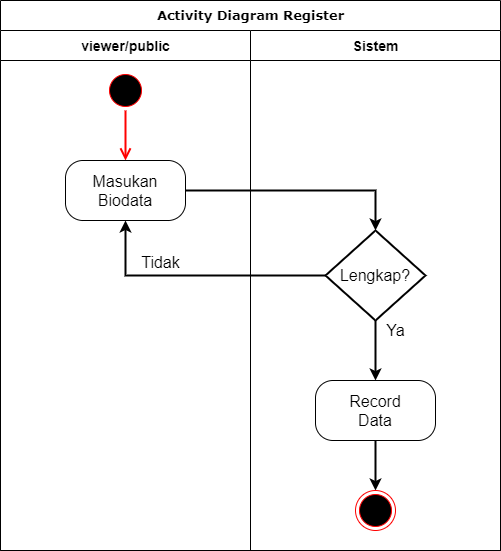
Diagram *use case* pada Gambar 7. Dapat dijelaskan aktivitas dari aktor yang berperan dalam sistem ini. Administrator melihat data penyewa, *member* dan penyedia sewa. Penyedia sewa mengiklankan barang dan jasa yang akan disewakan. Penyewa dan *member* melakukan pencarian berdasarkan kriteria yang diinginkannya pada sistem. Selain mencari informasi barang dan jasa, penyewa juga memberikan *rating* sesuai dengan kepuasan terhadap barang dan jasa yang disewa dan melakukan pemesanan atau *booking*. *Viewer* hanya bisa melakukan pencarian informasi sewa barang dan jasa.

1. *Activity Diagram*

*Activity Diagram* pada sistem penyewaan barang di Payakumbuh menunjukkan uraian dari setiap proses yang terjadi di sistem informasi, seperti pemesanan, aktivitas *login*, aktivitas pendaftaran pengguna dan lain-lain. Berikut adalah diagram aktivitas aplikasi *E-Rent* berbasis Android:

1. Registrasi *user*

*Activity Diagram* pada saat pengguna melakukan registrasi pada sistem dapat dilihat pada gambar 8 berikut:

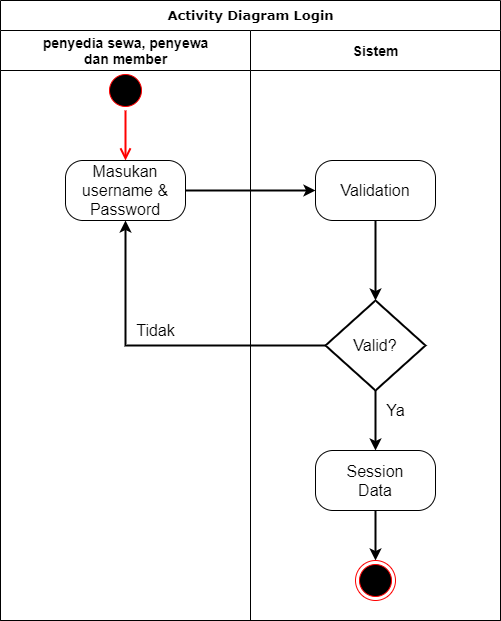


**Gambar 8.** *Register* *User*

Diagram aktivitas pada gambar 8. Dapat dijelaskan bahwa pengguna melakukan registrasi dengan memasukkan biodata selengkap mungkin sesuai dengan permintaan yang disediakan pada sistem. Jika data yang dimasukkan sudah lengkap dan benar, maka data biodata tersebut disimpan ke dalam *database.* Jika tidak, pengguna harus mengulangi kembali.

1. *Login*

Perancangan *Activity Diagram login* ke sistem dapat dilihat pada gambar 9 berikut:

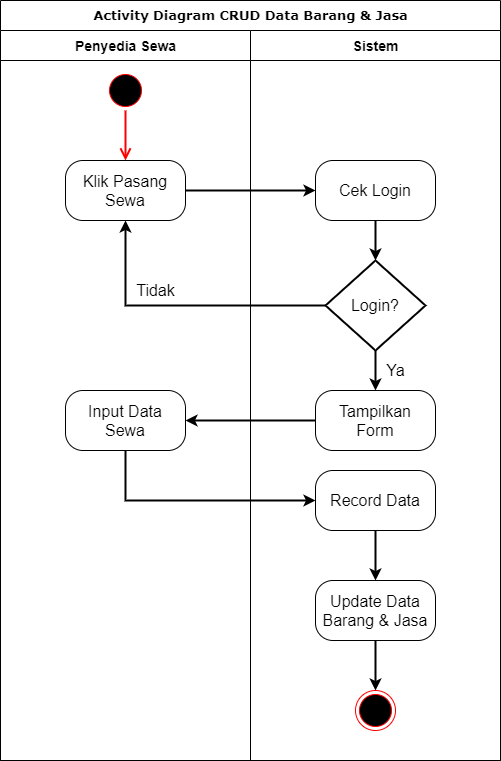


**Gambar 9.** *Activity* *Diagram* *Login* Sistem

Pengguna melakukan *login* ke sistem, harus memasukkan *username* dan *password*. *Username* dan *password* yang dimasukkan tersebut dicek terlebih dahulu keabsahan (validasi) data yang dimasukkan oleh pengguna. Jika data yang dimasukkan valid, maka sistem akan membuat *session* data sesuai dengan hak akses masing-masing pengguna. Jika tidak, pengguna harus mengulangi kembali.

1. CRUD data barang dan jasa

Perancangan *Activity Diagram create, read, update*, dan *delete* (CRUD) data barang dan jasa sebagai berikut ini:

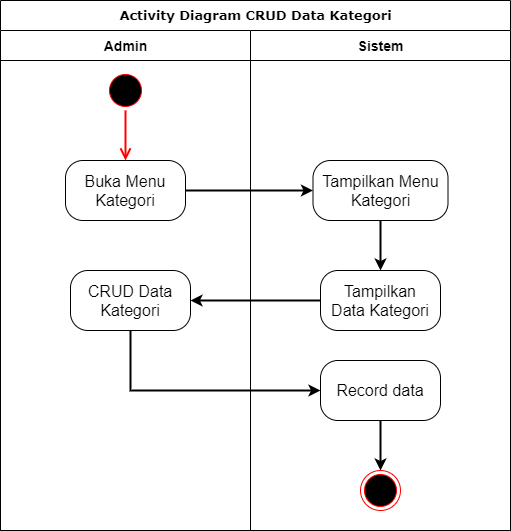


**Gambar 10.** *Activity* *Diagram* barang dan jasa

Pada *activity* ini, Penyedia sewa mengeklik tombol pasang sewa pada halaman. Setelah itu, sistem akan mengecek penyedia sewa tersebut apakah sudah *login* atau belum. Jika belum melakukan *login*, maka penyewa harus *login* terlebih dahulu. Jika sudah *login*, kemudian penyewa memasukkan data sesuai yang diinginkan sistem. Kemudian data sewa disimpan ke dalam *database*. Setelah itu sistem akan melakukan *update* pada data barang dan jasa tersebut.

1. CRUD Data Kategori

Perancangan *Activity Diagram create, read, update*, dan *delete* (CRUD) data kategori sebagai berikut:

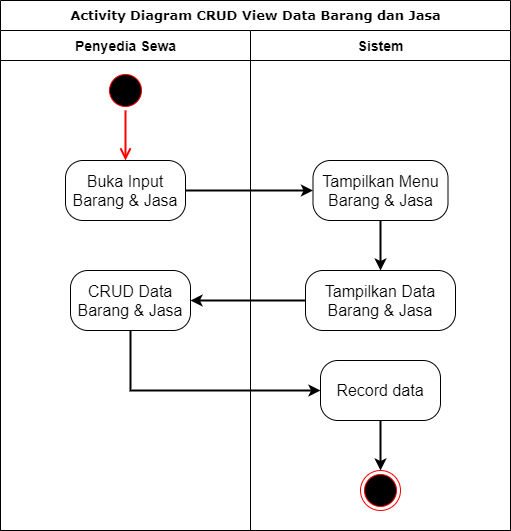


**Gambar 11.** *Activity* *Diagram* CRUD Data Kategori

Administrator membuka Halaman kategori dan sistem akan menampilkan data kategori. Administrator melakukan CRUD data kategori dan disimpan ke dalam *database.*

1. Menampilkan Data Barang dan Jasa

Perancangan *Activity Diagram* menampilkan barang dan jasa adalah sebagai berikut:

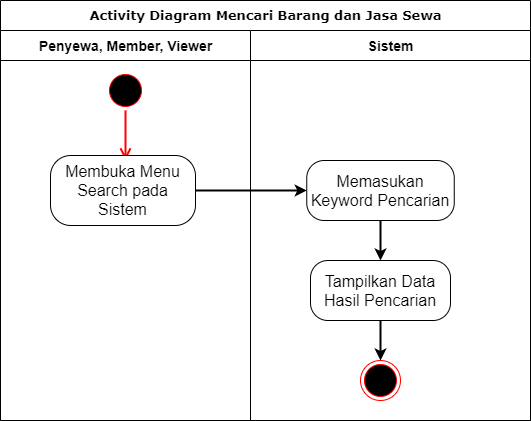


**Gambar 12.** *Activity* *Diagram* *View* Data Jasa Sewa

Penyedia sewa membuka Halaman barang dan jasa dan sistem akan menampilkan data jasa sewa. Pemilik jasa sewa melakukan CRUD data sewa dan disimpan ke dalam *database*.

1. Mencari Barang dan Jasa

Perancangan *Activity Diagram* mencari barang dan jasa dapat dilihat pada gambar 13 sebagai berikut:

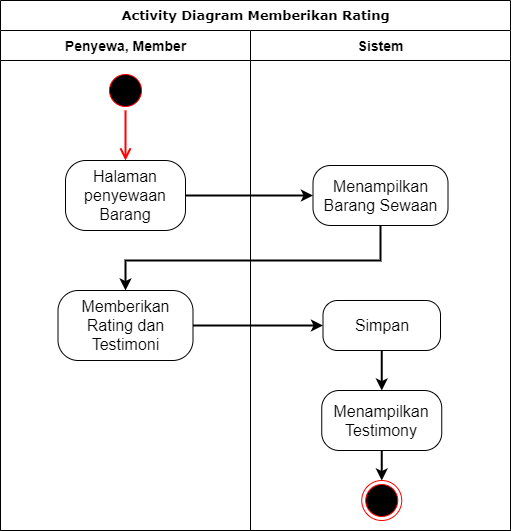


**Gambar 13.** *Activity Diagram* mencari barang dan jasa sewa

Penyewa dapat melakukan pencarian barang dan jasa, maupun lokasi tersedianya barang dan jasa menggunakan *keyword* sesuai dengan kebutuhan.

1. Memberikan *Rating* dan testimoni

Pemberian *rating* dan testimoni dilakukan oleh penyewa yang ingin memberikan apresiasi terhadap pelayanan atau kepuasan barang dan jasa yang telah disewa. Berikut ini *Activity* *Diagram* dalam memberikan *rating* pada sistem dapat di lihat pada gambar 14 di bawah ini:

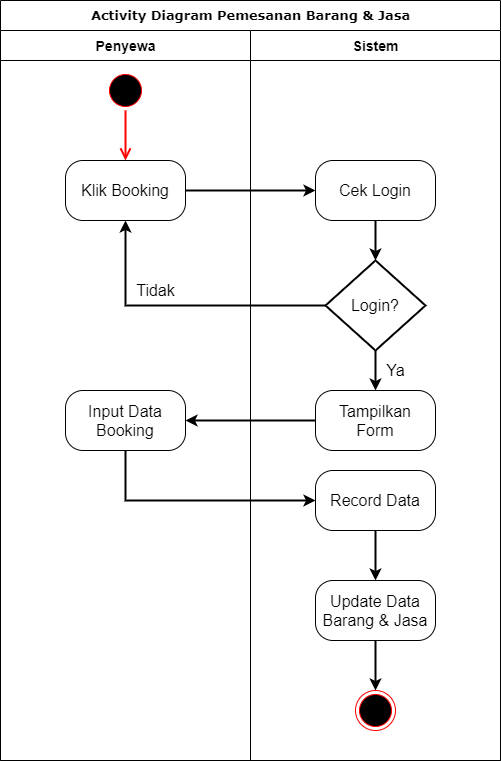


**Gambar 14.** *Activity Diagram* Memberikan *Rating*

Untuk dapat memberikan *rating* dan testimoni, pengguna harus menyelesaikan pemesanan barang terlebih dahulu. Hal ini dilakukan untuk mencegah terjadinya spam pada *rating* dan testimoni dan mengurangi terjadi pemalsuan isi testimoni.

1. *Activity* Pemesanan Barang dan Jasa

Perancangan *Activity Diagram* pemesanan barang dan jasa adalah sebagai berikut:



**Gambar 15.** *Activity Diagram* Pemesanan/*Booking*

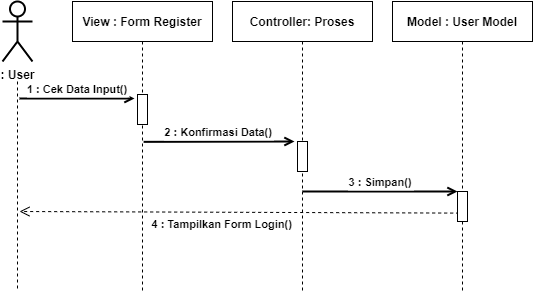
Penyewa mengeklik tombol pemesanan pada halaman *home* atau pencarian barang dan jasa. Setelah itu, sistem akan mengecek penyewa tersebut apakah sudah *login* atau belum. Jika belum melakukan *login*, maka penyewa harus *login* terlebih dahulu. Jika sudah *login*, kemudian penyewa memasukkan data sesuai kebutuhan sistem. Kemudian data pemesanan disimpan ke dalam *database*. Setelah itu sistem akan melakukan *update* pada data barang dan jasa tersebut.

1. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan proses yang dilakukan oleh pengguna (*user*) terhadap sistem. Pada sistem yang akan dibangun terdapat beberapa *Sequence Diagram*, di antaranya adalah sebagai berikut:

1. *Sequence Diagram* Registrasi

*Sequence Diagram* ini menggambarkan proses yang terjadi pada sistem di saat pengguna (*user*) melakukan registrasi terhadap sistem.

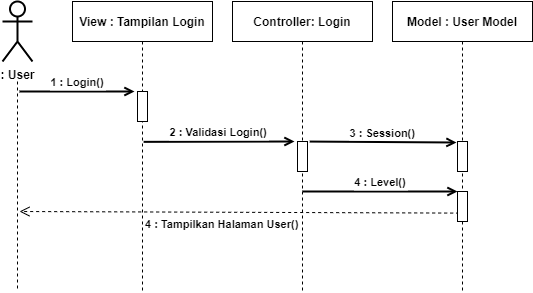


**Gambar 14.** *Sequence Diagram Registration*

Berdasarkan pada gambar 16. Dapat dijelaskan proses *Sequence* Diagram dimulai dari penyedia sewa(pengguna) melakukan *input* biodata lengkapnya sesuai dengan permintaan sistem, kemudian data yang dimasukkan tersebut dicek oleh sistem, kemudian data yang telah dicek tersebut di konfirmasi, setelah itu biodata yang dimasukkan tersebut disimpan.

1. *Sequence* *Diagram Login*

*Sequence Diagram Login* ini menggambarkan proses yang terjadi pada sistem di saat aktor atau *user* melakukan proses *login* terhadap sistem. *Sequence diagram login* pada sistem yang akan dibangun ini adalah sebagai berikut:



**Gambar 15.** *Sequence Diagram Login*

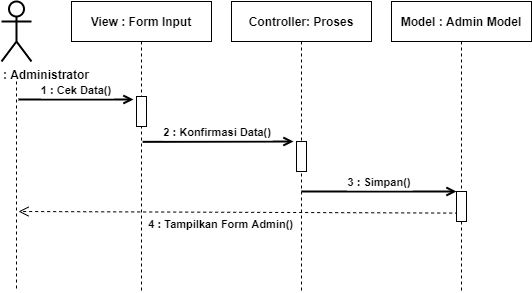
Berdasarkan pada Gambar 17. dapat dijelaskan proses *sequence* diagram berawal dari aktor melakukan *login*, maka sistem akan melakukan validasi data *login* yang dimasukkan oleh aktor berupa *username* dan *password*. Jika data yang dimasukkan tersebut valid, maka sistem akan membuat *session* sesuai dengan level aktor yang *login* sehingga aktor sukses memasuki sistem.

1. *Sequence Diagram Actor*

*Sequence Diagram* aktor ini menggambarkan proses-proses yang dilakukan oleh masing-masing aktor terhadap sistem. Adapun aktor-aktor yang melakukan interaksi terhadap sistem dimulai dari administrator, penyedia sewa, dan penyewa untuk lebih jelasnya sebagai berikut:

1. Administrator

*Sequence Diagram* administrator ini menggambarkan proses yang terjadi pada sistem saat administrator melakukan aktivitasnya.

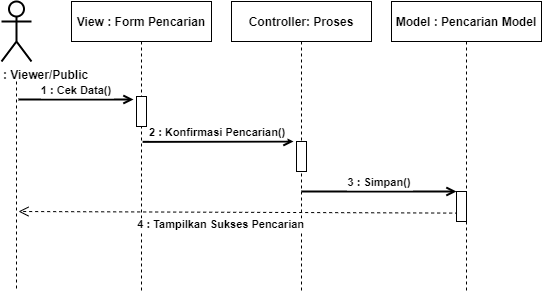


**Gambar 16.** *Sequence Diagram Administrator*

Setelah melakukan *login*, maka administrator melakukan proses *input* data, maka sistem akan melakukan proses penyimpanan data.

1. *Viewer/public*

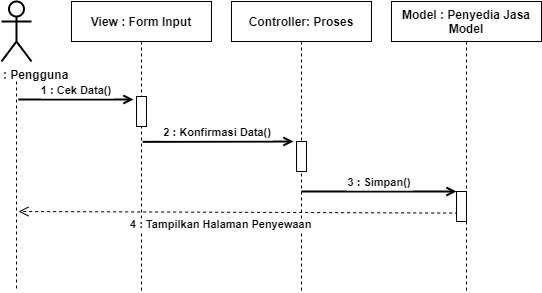
*Sequence Diagram* ini menggambarkan proses yang terjadi pada sistem saat *viewer/public* mencari barang atau jasa.



**Gambar 17.** *Sequence Diagram viewer/public*

1. Penyedia Sewa, penyewa, dan *member*

*Sequence Diagram* penyedia sewa, penyewa, dan *member* ini menggambarkan proses yang terjadi pada sistem saat penyedia sewa melakukan aktivitasnya.



**Gambar 18.** *Sequence Diagram* Penyedia sewa

Setelah melakukan *login*, maka pengguna dapat melakukan aktivitas berkaitan dengan barang dan jasa yang akan di sewakan ke dalam sistem, dan sistem akan melakukan proses penyimpanan data.

1. Perancangan *Database*
2. Normalisasi

Struktur tabel merupakan *database* dari aplikasi informasi barang dan jasa yang terdiri dari beberapa tabel yang saling berhubungan. Tabel tersebut merupakan tempat penyimpanan semua data yang diperlukan dalam pembuatan program. Sebelum memperoleh struktur *database* yang akan digunakan pada aplikasi, dilakukan proses normalisasi, yaitu sebagai berikut:

* + 1. Bentuk Tidak Normal (*Unnormalize Form*)

Pada tahap ini, semua data yang ada, ditulis tanpa format tertentu, data bisa jadi mengalami duplikasi. Di bawah ini ditampilkan tabel *universal* hasil rangkuman untuk menyewa suatu barang.

**Tabel 18.** Bentuk Tabel Tidak Normal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Pengguna | Barang/jasa | Tanggal pemesanan | Tanggal Kembali | Biaya/hari |  |
| Widia | Rental Mobil | 10/5/2020 | 12/5/2020 | 250K |  |
| Rental Kamera | 9/5/2020 | 13/5/2020 | 25K |  |
| Ovilia | 15/5/2020 | 16/5/2020 |  |
| Perbaikan AC | 13/5/2020 | 14/5/2020 | 100K |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Lama Pemakaian | Jaminan | Alamat Store | No. Telepon Store |
|  | 3 hari | KTP | Payakumbuh utara | 54321 |
|  | 4 hari | Payakumbuh barat | 12345 |
|  | 2 hari |
|  | - | - | Payakumbuh selatan | 53241 |

Pada tabel 18 terdapat *attribute* dalam bentuk yang tidak normal (UNF) yang memiliki *attribute* yang bersifat *multivalued* (bernilai banyak). *Attribute* nama\_pengguna, untuk 1 orang nama pengguna dapat menyewa 2 barang dan atau jasa, dan 1 barang atau jasa disewakan oleh 2 orang pengguna. Untuk menghindari *attribute multivalued* maka tabel dalam bentuk ini dinormalisasikan menjadi tabel dalam bentuk normal pertama (1NF).

* + 1. Bentuk Normal Pertama (1NF)

Suatu relasi 1NF jika dan hanya jika sifat dari setiap relasi atributnya bersifat atomik. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya. Berikut ini adalah bentuk tabel normal kesatu (1NF).

**Tabel 19.** Bentuk Tabel Normal 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Pengguna | Barang/jasa | Tanggal pemesanan | Tanggal kembali | Biaya/hari |  |
| Widia | Rental Mobil | 10/5/2020 | 12/5/2020 | 250K |  |
| Widia | Rental Kamera | 9/5/2020 | 13/5/2020 | 25K |  |
| Ovilia | Rental Kamera | 15/5/2020 | 16/5/2020 | 25K |  |
| Ovilia | Perbaikan AC | 13/5/2020 | 14/5/2020 | 100K |  |

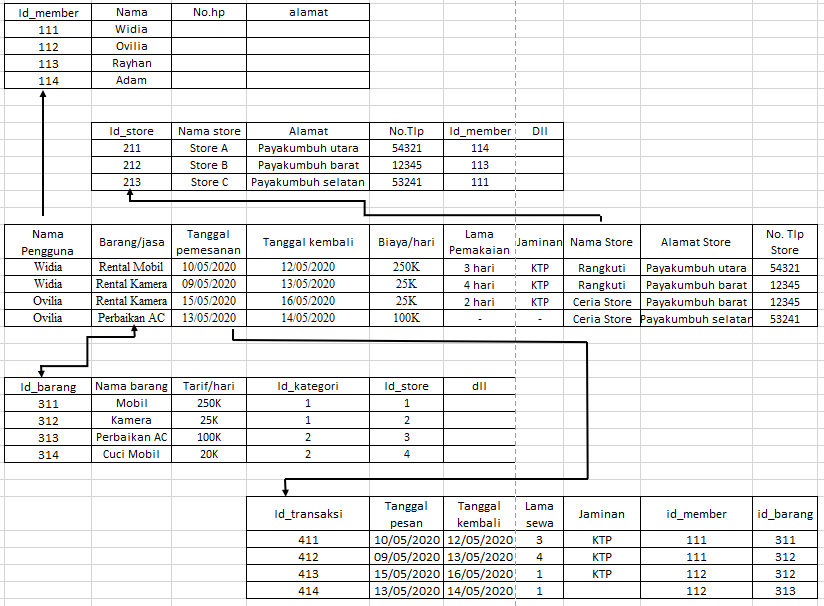
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lama Pemakaian | Jaminan | Nama Store | Alamat Store | No. Telepon Store |
|  | 3 hari | KTP | Rangkuti | Payakumbuh utara | 54321 |
|  | 4 hari | KTP | Rangkuti | Payakumbuh barat | 12345 |
|  | 2 hari | KTP | Ceria Store | Payakumbuh barat | 12345 |
|  | - | - | Ceria Store | Payakumbuh selatan | 53241 |

Pada tabel 19 tidak terdapat atribut/kolom yang bersifat *multivalued*  dan data sudah mengandung 1 nilai. Kolom nama penyewa, barang/jasa, biaya per hari, jaminan yang sebelumnya memuat satu *value* untuk beberapa penyewa, pada normal pertama (1NF) telah memuat satu untuk tiap penyewaan. Tabel 19 masih memiliki data yang bersifat anomali yang akan berpengaruh pada proses *insert*, *delete*, dan *update*. Misalnya, jika Widia menghapus rental kamera, dan ingin menggantinya menjadi rental mobil maka 1 *record* dari *attribute* Ovilia akan terhapus. Untuk menghilangkan terjadinya anomali, tabel yang telah dinormalisasi bentuk pertama akan dinormalisasi menjadi tabel bentuk normal kedua (2NF).

* + 1. Bentuk Normal Kedua (2NF)

Suatu model data dikatakan memenuhi bentuk normal kedua apabila data tersebut telah memenuhi bentuk normal pertama. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama atau *primary key*. Sehingga untuk membentuk normal kedua harus ditentukan kunci-kunci *field*. Kunci *field* harus bersifat unik dan dapat mewakili atribut lain yang menjadi anggotanya. Maka tabel dalam bentuk normal pertama (1NF), dikelompokkan menjadi beberapa tabel berdasarkan ketergantungan fungsi seperti tabel berikut ini:

**Tabel 20.** Tabel Bentuk Normal Kedua 2NF



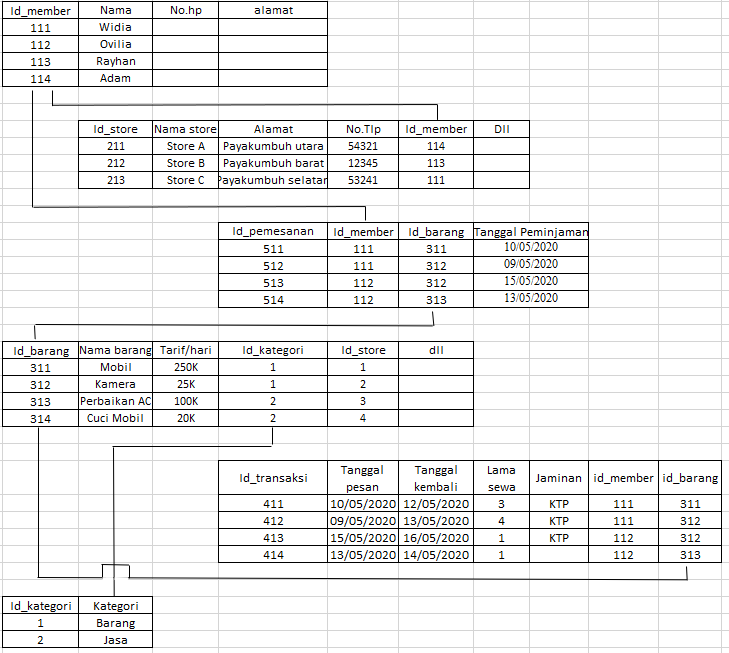
Tabel 20 merupakan normalisasi kedua untuk *database* aplikasi yang akan di rancang. Untuk normalisasi kedua tabel-tabel tidak memiliki ketergantungan fungsional karena sudah dilakukan dekomposisi. Setiap tabel memiliki *record* yang bergantung pada satu *field key* saja. Nama penyewa bergantung pada id\_member, nama barang bergantung pada id\_store, tanggal sewa bergantung pada id\_transaksi. Pada bentuk normal kedua (2NF), masih terjadi ketergantungan transitif, oleh sebab itu proses normalisasi dilanjutkan ke bentuk normal ketiga (3NF).

* + 1. Bentuk Normal Ketiga (3NF)

Untuk menjadi bentuk normal ketiga maka relasi harus dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* secara menyeluruh. Pada tabel bentuk normal kedua di atas termasuk juga bentuk normal ke tiga karena seluruh atribut yang ada bergantung penuh pada kunci utamanya.

Namun Relasi dalam bentuk normal kedua ini tidak menyimpan fakta tentang bagian dari kunci relasi yang menghubungkan satu *field* pada suatu tabel dengan *field* lain pada tabel yang berbeda. Maka tabel 2NF tersebut dapat dibentuk menjadi 3NF seperti berikut:

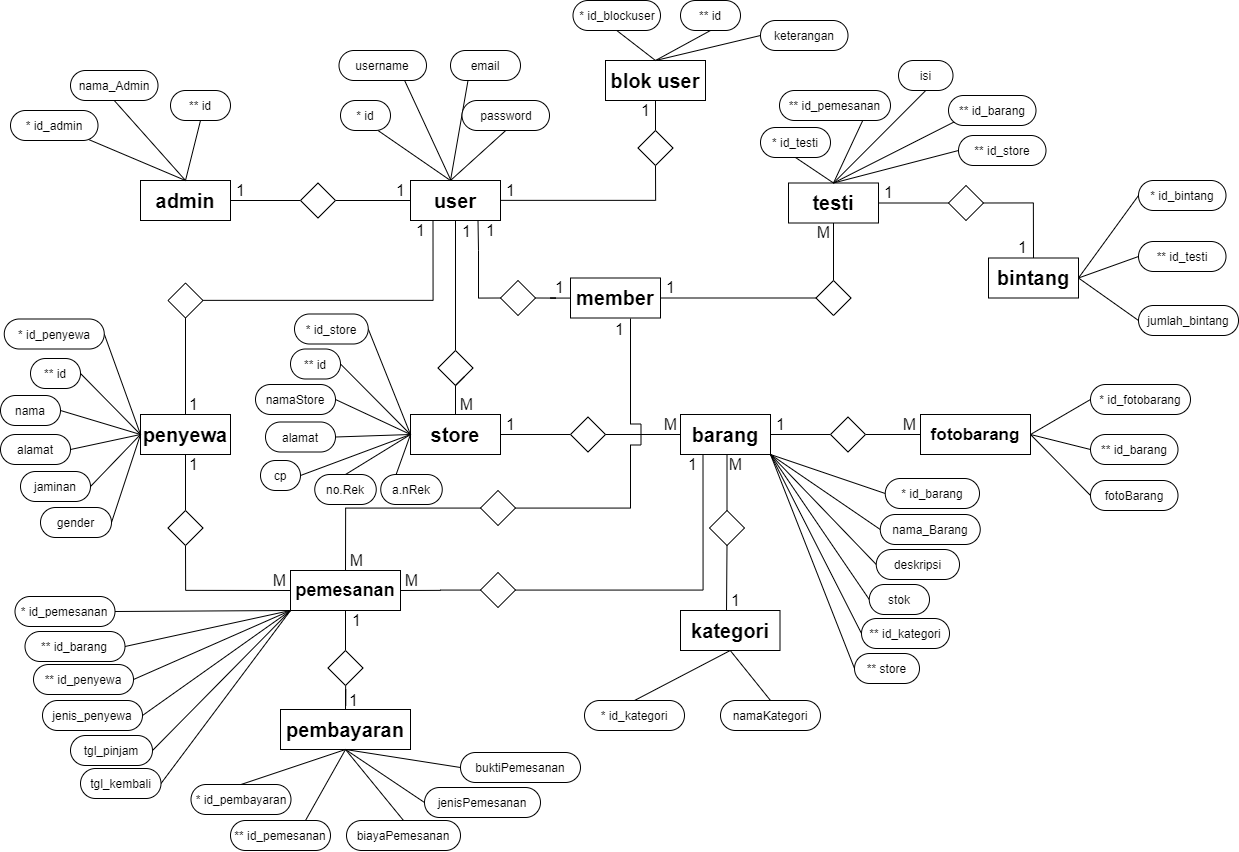
**Tabel 21.** Tabel Bentuk Normal Ketiga (3NF)



Pada tabel 21, sudah memuat fakta tentang bagian dari kunci relasi yang menghubungkan satu *field* pada suatu tabel dengan *field* lain pada tabel yang berbeda. Di sini juga sudah terlihat relasi yang jelas antar tabel-tabel. Selain itu, tabel transaksi pada 3NF ini, *field* lama sewa dihapus karena lama sewa bisa dilihat dari tanggal pemesanan dan tanggal pengembalian.

1. *Entity Relationship Diagram (ERD)*

Berikut perancangan basis data diperlukan *Entity Relationship Diagram* (ERD) pada aplikasi informasi penyewaan barang dan jasa di Kota Payakumbuh:



**Gambar 19.** ERD Sistem

Perancangan basis data diperlukan *Entity Relationship Diagram* (ERD) pada aplikasi informasi penyewaan barang dan jasa di Kota Payakumbuh sudah melalui tahap normalisasi bentuk ketiga (3NF) dengan proses yang sama seperti proses normalisasi yang dibahas sebelumnya. Dari hasil akhir normalisasi bentuk ketiga (3NF) terbentuklah 20 tabel yang memiliki relasi satu sama lainnya. Relasi dari tabel tersebut memiliki *cardinalities* yang berbeda sesuai dengan fungsinya.

1. Struktur Tabel

Struktur tabel merupakan *database* dari aplikasi informasi sewa menyewa barang jasa di Kota Payakumbuh yang terdiri dari beberapa tabel yang saling berhubungan dan merupakan hasil dari 3NF. Tabel tersebut merupakan tempat penyimpanan semua data yang diperlukan dalam pembuatan program. Struktur tabel yang sesuai dengan kebutuhan sebagai media *entry* dan media penyimpanan data adalah sebagai berikut:

1. Tabel *User*

Tabel *user* adalah tabel yang berisikan tentang data-data *user* untuk aplikasi penyewaan.

**Tabel 22.** Tabel *User*

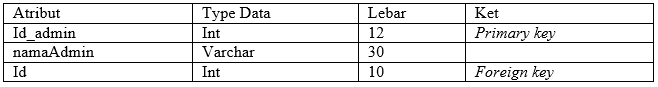


Pada tabel *user, username* bisa terdiri dari huruf dan angka, *password* akan diisi dengan kombinasi huruf dan angka, dan *email* juga terdiri atas kombinasi angka, huruf, dan simbol. Oleh sebab itu ketiga atribut menggunakan tipe data varchar.

1. Admin

Tabel Admin adalah tabel yang berisikan tentang data-data Admin untuk aplikasi penyewaan.

**Tabel 23.** Tabel Admin

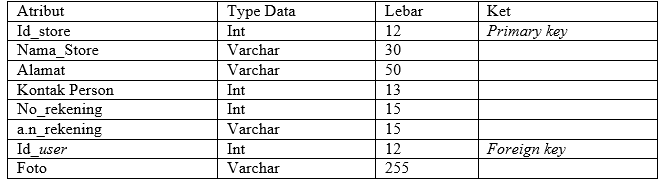


Pada tabel admin, id\_admin merupakan identitas dari admin. Nama admin bertipe *varchar*, dan tabel admin berelasi dengan tabel *user.*

1. Tabel *Store*

Tabel *store* adalah tabel yang berisikan tentang data-data *store* untuk aplikasi penyewaan.

**Tabel 24.** Tabel *Store*

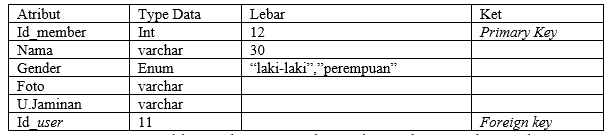


Pada tabel *store*, *attribute* yang menggunakan tipe data *varchar* yaitu nama\_Store, alamat, a.n\_rekening. A.n. rekening merupakan *record* yang menyimpan data nama dari rekening. Tabel *store* berelasi dengan tabel *user*.

1. Tabel *Member*

Tabel *member* adalah tabel yang berisikan tentang data-data *member* untuk aplikasi penyewaan.

**Tabel 25.** Tabel *Member*



Id\_member merupakan *primary key* untuk *member*. Nama menggunakan tipe *varchar* dengan lebar data 30, gender menggunakan tipe data *enum* karena hanya 2 pilihan data, yaitu laki-laki dan perempuan. Sedangkan untuk foto dan u.jaminan menggunakan tipe data blob (*binary large object*). Tabel *member* berelasi dengan tabel *user*.

1. Tabel Penyewa

Tabel penyewa adalah tabel yang berisikan tentang data-data penyewa untuk aplikasi penyewaan.

**Tabel 26.** Tabel Penyewa

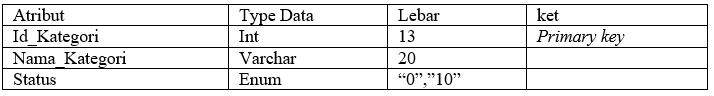


Tabel penyewa berisi data dari penyewa. Id\_penyewa merupakan *primary* *key* dari tabel ini. Tabel penyewa berelasi dengan tabel *user*.

1. Tabel Kategori

Tabel kategori adalah tabel yang berisikan tentang data-data kategori untuk aplikasi penyewaan.

**Tabel 27.** Tabel Kategori

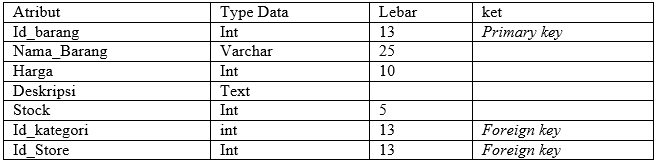


Tabel kategori berisi jenis kategori yang dimasukkan oleh admin. Nama kategori seperti *photography, property, fashion*, jasa.

1. Tabel Barang dan jasa

Tabel barang dan jasa adalah tabel yang berisikan tentang data-data barang dan jasa untuk aplikasi penyewaan.

**Tabel 28.** Tabel Barang dan Jasa

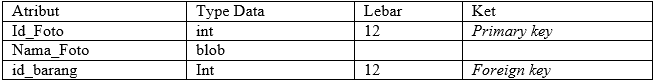


Tabel barang dan jasa berisi data mengenai barang dan jasa yang akan di promosikan. Data barang dan jasa meliputi nama barang, harga, deskripsi dari barang, *stock* barang. Tabel barang dan jasa berelasi dengan tabel kategori dan tabel *store.*

1. Tabel Foto Barang

Tabel foto barang adalah tabel yang berisikan tentang data-data foto barang, jika penyedia sewa men *upload* lebih dari satu barang

**Tabel 29.** Tabel Foto Barang

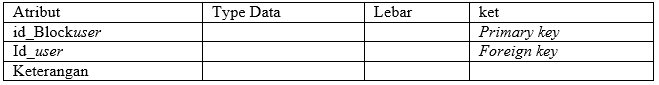


Tabel foto barang merupakan tabel yang berisi data foto barang yang di *upload* oleh penyedia sewa. Foto barang mempunyai tabel sendiri karena penyedia sewa bisa men *upload* lebih dari satu foto untuk satu barang. Tabel foto berelasi dengan tabel barang

1. Tabel *Block User*

Tabel *block user* adalah tabel yang berisikan tentang data-data *block user* untuk aplikasi penyewaan.

**Tabel 30.** Tabel *Block* *User*

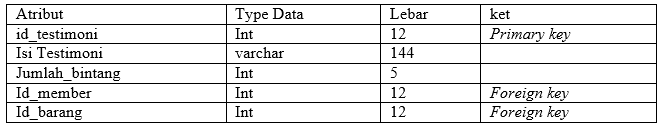


Tabel *block user* merupakan tabel yang berisi data *user* yang sudah di *block* oleh admin. Untuk keterangan akan diisi oleh admin bila dibutuhkan.

1. Tabel Testimoni

Tabel testimoni adalah tabel yang berisikan tentang data-data testimoni untuk aplikasi penyewaan.

**Tabel 31.** Tabel Testimoni

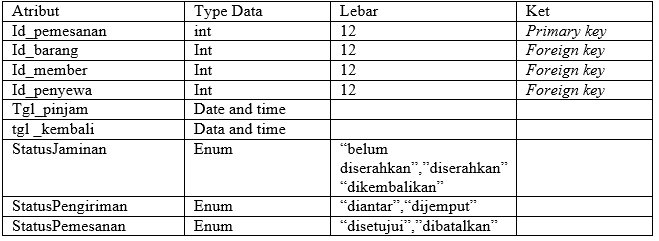


Tabel *testimony* menampung data komentar yang dimasukkan oleh penyewa dan *member*. Setiap komentar yang diisi, bisa diberi *rating*. *Rating* tersebut disimpan di jumlah\_rating. Kumpulan dari rating-rating ini akan di kalkulasi oleh sistem menjadi *rating* dari barang. Jadi *rating* dari suatu barang ditentukan dari banyaknya ulasan/komentar. Tabel *testimony* berelasi dengan tabel *member* dan tabel barang.

1. Tabel Pemesanan

Tabel pemesanan adalah tabel yang berisikan tentang data-data pemesanan untuk aplikasi penyewaan.

**Tabel 32.** Tabel Pemesanan

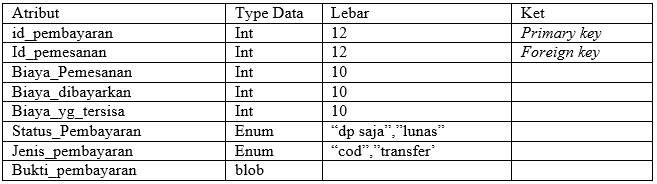


Tabel pemesanan berisi hal-hal yang berkaitan dengan proses pemesanan. Tanggal\_peminjaman dan tanggal\_kembali merupakan lama penyewaan yang dilakukan. Status\_jaminan menyimpan data status jaminan, apakah jaminan sudah diserahkan atau belum. Status\_pengiriman diisi oleh penyewa atau *member*, yaitu diantar atau dijemput. Status\_pemesanan merupakan status dari pemesanan apakah pemesanan disetujui atau dibatalkan. Tabel pemesanan berelasi dengan tabel barang, tabel *member*, dan tabel penyewa.

1. Tabel Pembayaran

Tabel pembayaran adalah tabel yang berisikan tentang data-data pembayaran untuk aplikasi penyewaan.

**Tabel 33.** Tabel Pembayaran

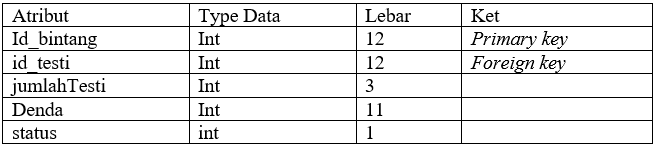


Tabel pembayaran berisi data yang bersangkutan dengan pembayaran. Semua biaya di simpan di sini. Biaya\_pemesanan, biaya\_dibayarkan, dan biaya\_sisa. Biaya\_sisa digunakan jika penyewa dan *member* hanya membayar uang muka saja. Status\_pembayaran yaitu status pembayaran dibayar lunas atau cicil. Jenis\_pembayaran apakah dibayar di tempat atau ditransfer. Jika penyewa dan *member* memilih jenis pembayaran transfer, maka penyewa dan *member* akan mengupload bukti transfer yang disimpan di bukti\_pembayaran. Tabel pembayaran berelasi dengan tabel pemesanan.

1. Tabel Bintang

Tabel bintang berisi data yang menampung informasi *rating* yang di masukan oleh *user* berupa integer. Tabel bintang berelasi dengan tabel testimony.

**Tabel 34.** Tabel Bintang



1. Perancangan Antarmuka (*Interfaces*)

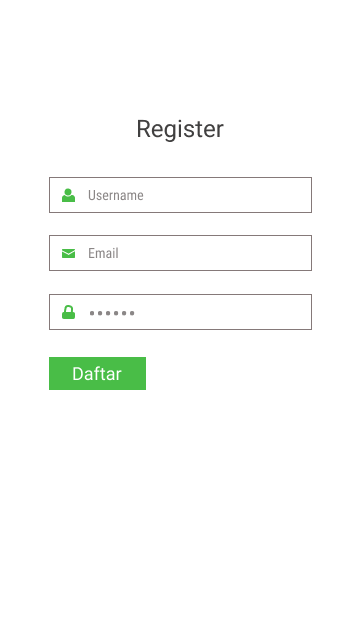
Antarmuka pemakai (*User* *Interface*) merupakan mekanisme komunikasi antara *user* dengan sistem. *User Interface* dapat menerima informasi dari *user* dan memberikan informasi kepada *user* untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi. Tahapan perancangan *interface* merupakan tahapan menentukan dan merancang tampilan dari sistem yang akan dibuat. Berikut ini tampilan aplikasi *E-Rent* berbasis Android:

1. Perancangan *Input*

Perancangan *input* merupakan halaman pendaftaran bagi pengguna yang ingin melakukan sewa barang dan menyewakan barang Pengguna mengisi data berupa *username*, email, dan *password.*

1. Halaman Registrasi

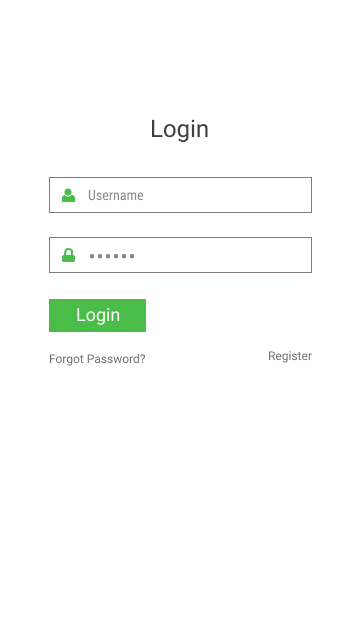
Halaman register merupakan halaman pendaftaran bagi pengguna lain yang ingin melakukan sewa barang dan menyewakan barang. Pengguna mengisi data *username,* email, dan *password*.



**Gambar 20.** Tampilan Halaman *Register*

1. Halaman *Login*

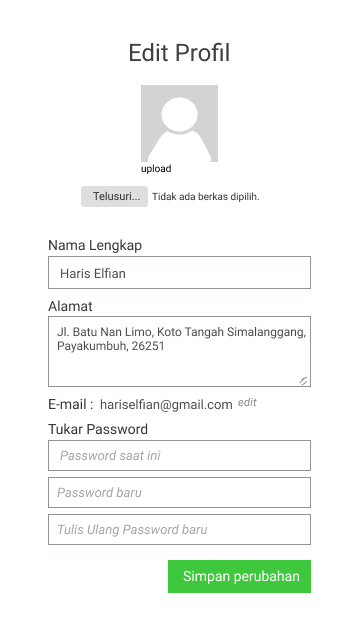
Halaman *Login* digunakan oleh pengguna yang sebelumnya sudah melakukan registrasi untuk masuk ke dalam sistem. Pengguna memasukkan *username* dan *password*, lalu klik *login*. Maka pengguna akan masuk ke halaman utama.



**Gambar 21.** Tampilan Halaman *Login*

1. Halaman Edit Profil

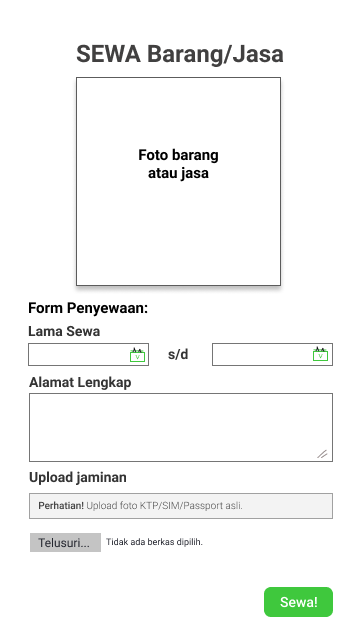
Halaman edit profil berguna jika pengguna ingin men update data diri, ingin mengubah email, ataupun mengganti *password*.



**Gambar 22.** Tampilan Halaman *Edit Profil*

1. Halaman *Form* Pemesanan

Pada halaman ini akan disediakan *form* yang berfungsi untuk melengkapi registrasi yang harus diisi oleh pengguna jika ingin menyewa barang atau jasa. Berikut tampilan *form* sewa barang.

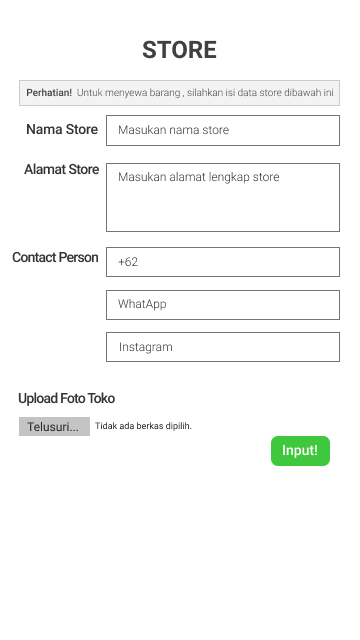


**Gambar 23.** Tampilan Halaman Sewa

Setelah mengisi *form* sewa barang, maka akan muncul rincian harga dan lama sewa berupa *billing* yang dapat di *print* oleh penyewa yang nantinya akan di bawa saat melakukan transaksi dengan penyedia sewa.

1. Halaman *Form Input* Barang dan Jasa

Pada halaman ini akan disediakan *form*  yang berfungsi untuk melengkapi registrasi yang harus diisi oleh pengguna jika ingin menyewakan barang atau jasa. Sebelum mengisi informasi barang dan jasa yang akan disewakan, pengguna harus membuat sebuah *store*. Tampilan halaman *form store* seperti berikut:



**Gambar 24.** Tampilan Halaman *Form Store*

Setelah pengguna mengisi *form store,* selanjutnya pengguna akan diarahkan untuk mengisi *form* informasi barang dan jasa yang akan disewakan. Berikut tampilan form pasang sewa.



**Gambar 25.** Tampilan Halaman *Form* Pasang Sewa

1. Halaman *Form Input Testimony*

Pada halaman ini, disediakan *form* untuk memberikan komentar/*review/testimony*. Pengguna yang memasukkan *review* datanya akan tersimpan di dalam *database* dan akan ditampilkan pada halaman barang dan jasa.

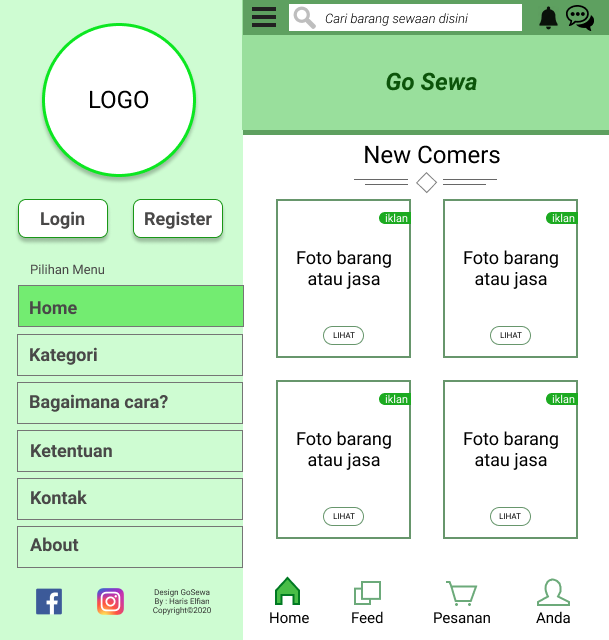


**Gambar 26.** Tampilan Halaman *Form* Input Testimony

1. Perancangan *Output*

Perancangan *output* merupakan desain yang dirancang untuk menampilkan data yang sudah dimasukkan oleh pengguna. Rancangan *output* dapat berupa *display* yang ditampilkan dilayar, ataupun dapat disimpan di dalam berbagai format data seperti format dokumen (.doc), format *excel*, format pdf, format gambar, dll. Berikut ini perancangan *output* aplikasi *E-Rent* berbasis Android.

1. **Halaman *Home***



**Gambar 27.** Tampilan Halaman *Home* Aplikasi

Pada saat pengguna membuka aplikasi *android*, halaman pada gambar merupakan halaman pertama yang akan terbuka, Pada halaman ini, terdapat beberapa pilihan menu halaman di antaranya:

1. Menu Halaman *Login,* yaitu halaman masuk bagi pengguna yang sudah mendaftar.
2. Halaman *Register*, yaitu halaman untuk melakukan pendaftaran bagi pengguna yang ingin menyewa barang atau ingin menyewakan barang.
3. Halaman *Home,* berisi informasi barang dan jasa yang sudah dimasukkan oleh penyedia sewa.
4. Halaman Kategori, merupakan halaman *dropdown* untuk mencari barang dan jasa sesuai dengan kategori. Berikut tampilan halaman kategori :



**Gambar 28.** Tampilan Halaman Kategori

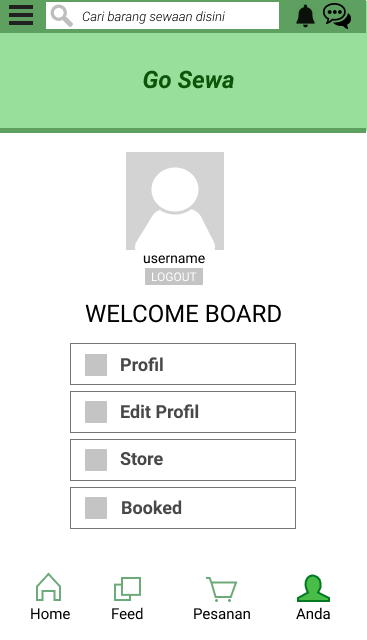
1. Halaman bagaimana cara?, merupakan halaman untuk mengetahui langkah-langkah menyewakan barang ataupun menyewa jasa. Berikut tampilan Halaman bagaimana cara?



**Gambar 29.** Tampilan Halaman Bagaimana cara?

1. Halaman ketentuan, merupakan halaman yang berisi ketentuan sewa barang dan jasa serta ketentuan untuk menyewakan barang dan jasa serta beberapa ketentuan khusus yang ada pada sistem.
2. Halaman Kontak, merupakan halaman yang berisi kontak yang dapat dihubungi terkait dengan aplikasi informasi.
3. Halaman *about*, yaitu halaman yang berisi segala sesuatu tentang aplikasi informasi.
4. Halaman pencarian, yaitu sebuah kolom yang digunakan untuk mencari barang. Barang akan tampil berdasarkan *keyword* yang ditulis.
5. **Halaman Utama**

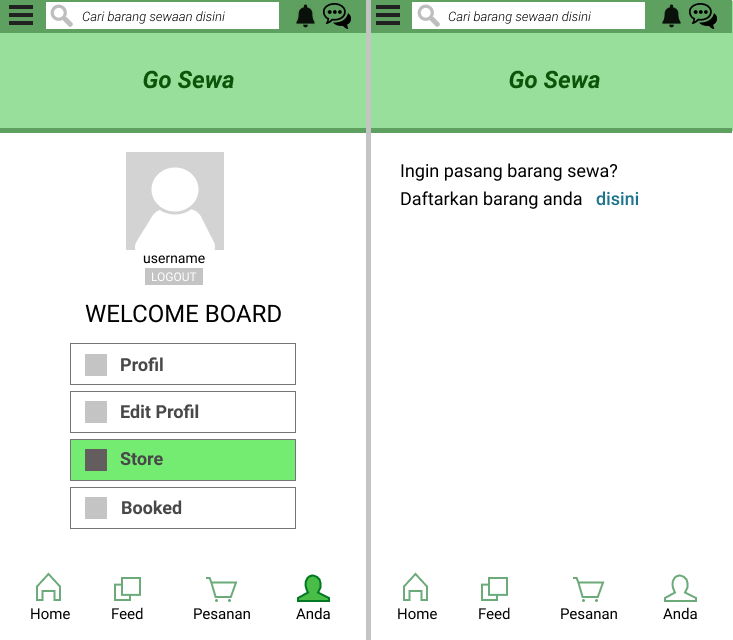
Halaman utama merupakan halaman setelah pengguna melakukan *login.* Berikut tampilan halaman utama.



**Gambar 30.** Tampilan Halaman Utama

Pada halaman utama terdapat beberapa halaman yaitu *profil*, *edit profil*, *store*, dan *booked*.

1. Halaman *profil* akan menampilkan detail *profil* pada saat melakukan pendaftaran.
2. Halaman edit *profil* merupakan Halaman yang disediakan untuk pengguna mengganti informasi *profil*.
3. Halaman *store* berisikan informasi *store* / toko dan informasi barang yang akan disewakan.
4. Halaman *booked* merupakan Halaman yang berisikan informasi barang atau jasa yang sudah di pesan. Berikut ini tampilan saat membuka Halaman *store* dan Halaman *booked*.



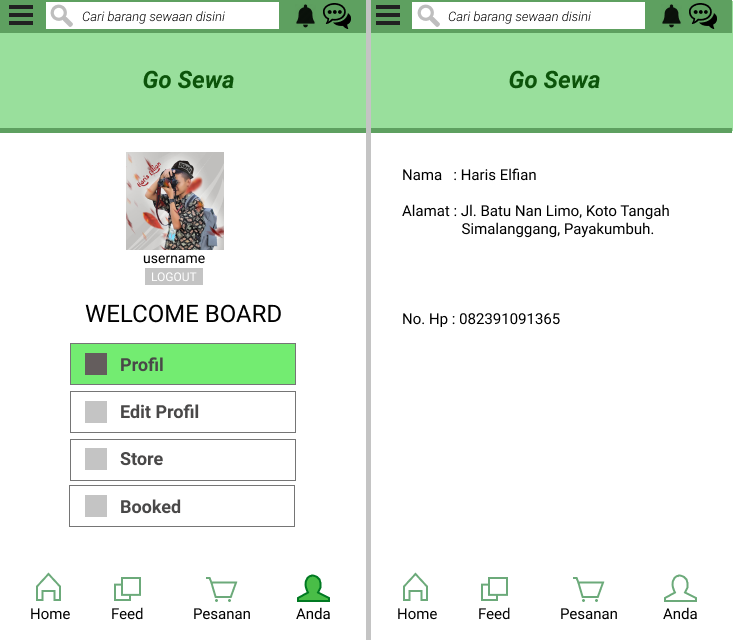
**Gambar 31.** Tampilan Halaman Store



**Gambar 32.** Tampilan Halaman *Booked*

1. **Halaman Tampilan Profil**

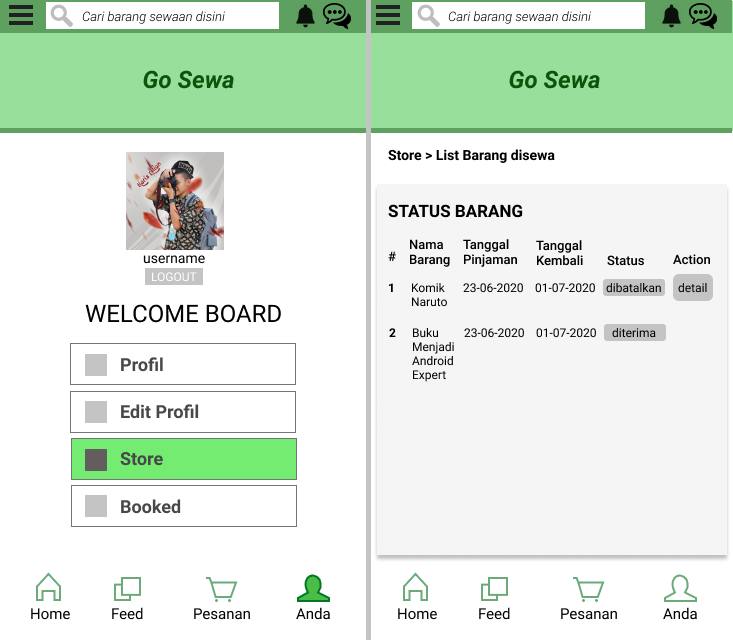
Halaman profil dapat di akses melalui Halaman utama. Halaman profil berisi data-data umum dari pengguna seperti nama, alamat, No. Hp, jenis kelamin, dll. Berikut ini perancangan halaman profil dari aplikasi *E-Rent* berbasis Android.



**Gambar 33.** Tampilan Halaman Profil

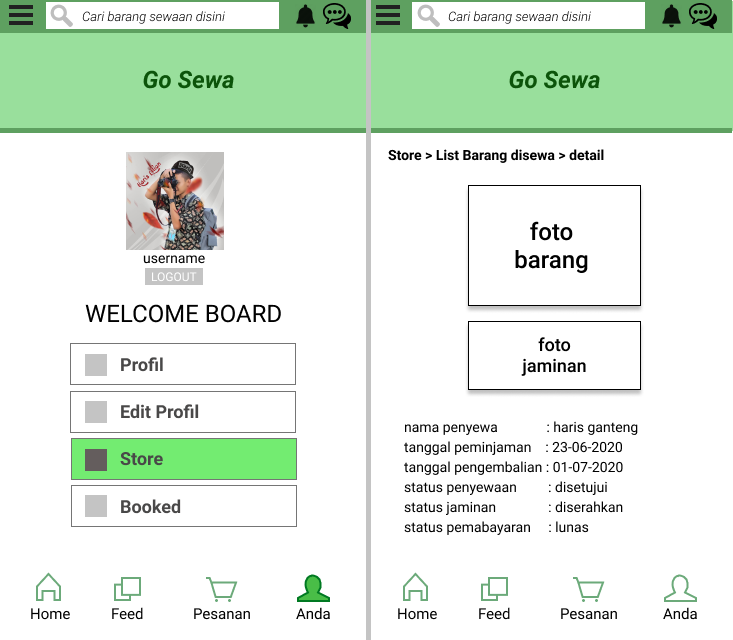
1. **Halaman Laporan Barang**

Pada halaman laporan barang ini, penyedia sewa dapat mengetahui barang apa saja yang sedang disewa:



**Gambar 34.** Tampilan Halaman Profil

Untuk melihat detail barang yang disewa disediakan 1 halaman detail barang seperti berikut:



**Gambar 35.** Gambar Detail Laporan Barang

Pada halaman ini terdapat beberapa informasi mengenai barang yang sedang disewa. Foto barang yang disewa, kemudian nama penyewa. Jika nama penyewa di klik maka akan muncul halaman profil penyewa. Selain itu juga terdapat informasi lama penyewaan, status penyewaan, status jaminan beserta gambar *upload* jaminan, dan status pembayaran.

1. **Halaman Barang dan Jasa**

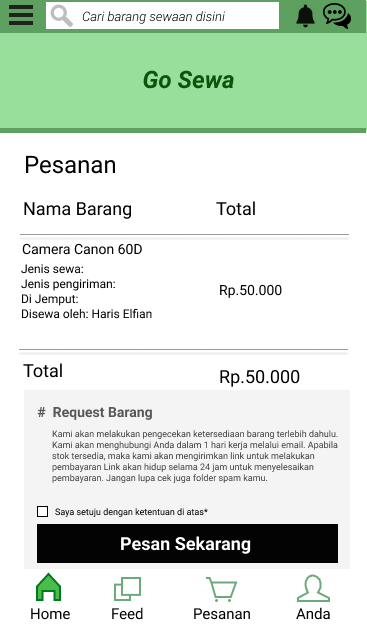
Halaman barang dan jasa menampilkan data barang dan jasa yang telah dimasukkan oleh pengguna. Data berupa foto, nama barang, tarif penyewaan, stok barang, dan *rating*. Halaman ini dapat di akses oleh semua pengguna. Pada halaman barang dan jasa selain menampilkan informasi barang dan jasa juga menampilkan nama penyedia sewa yang menyewakan barang dan jasanya.



**Gambar 36.** Halaman Barang dan Jasa

1. **Halaman Pemesanan**

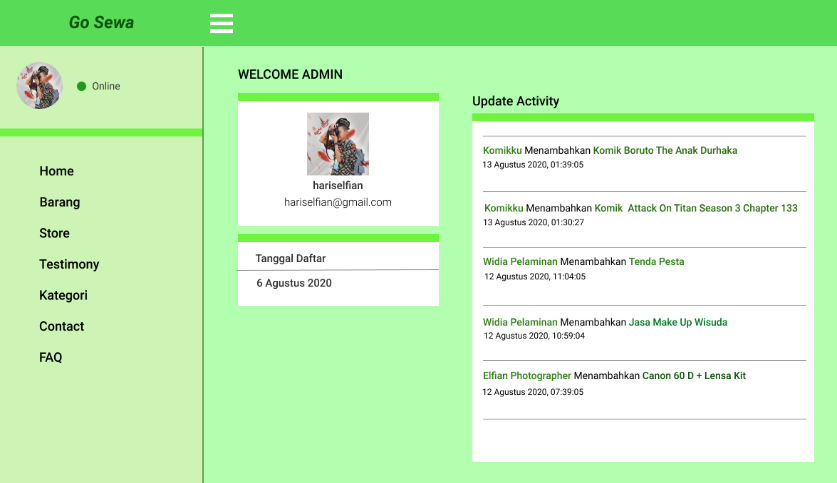
Sebelum melakukan transaksi, penyewa harus menentukan pemesanan barang dan jasa terlebih dahulu . tampilan pesanan akan tampil seperti pada gambar 37.



**Gambar 37.** Halaman Pemesanan

1. Perancangan Pengelolaan Admin

Perancangan pengelolaan Admin merupakan desain yang dirancang untuk menampilkan data yang akan dimasukkan oleh administrator untuk aplikasi *E-Rent* ini. Halaman admin digunakan oleh administrator sistem untuk mengelola sistem secara keseluruhan. Sebagian besar pengaturan sistem bisa dilakukan oleh admin dari halaman admin, kecuali hal-hal yang terkait dengan proses bisnis. Dari halaman ini, administrator sistem bisa melakukan manajemen *user*, mengubah keterangan mengenai sistem, melakukan perubahan terhadap kontak, mengetahui informasi barang, *store*, memblokir *user*, dan lain-lain Berikut ini perancangan pengelolaan Admin aplikasi *E-Rent* akan tampil seperti gambar 38 :



**Gambar 38.** Halaman Pengelolaan Admin

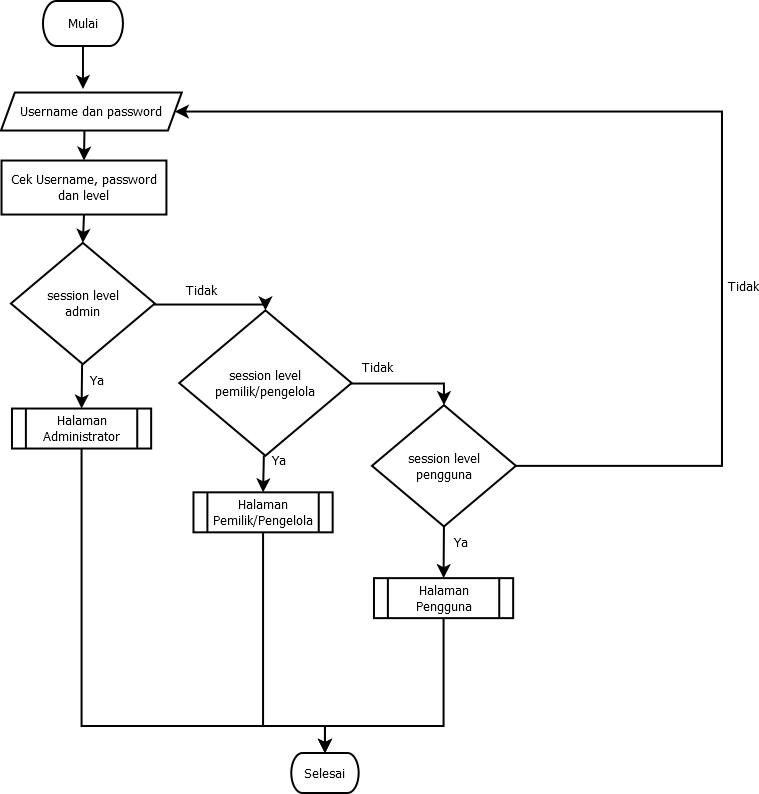
Pada halaman pengelolaan Admin ini, ditampilkan informasi berupa profil singkat admin berupa foto, *username* dan email yang diambil dari tabel *user*, dan beberapa aktivitas yang terjadi di dalam sistem. Kemudian ada beberapa menu untuk admin dalam mengelola sistem dan *user* dan menambahkan kategori barang.

1. Rancangan Keamanan

Sistem yang memiliki data atau transaksi yang bersifat rahasia harus memiliki keamanan yang baik. Keamanan yang baik untuk sistem informasi menggunakan berbagai macam metode atau teknik keamanan seperti yang telah dijelaskan sebelumnya. Pada sistem informasi ini penulis menggunakan teknik keamanan sistem menggunakan (1) Level Pengguna (*Session*), (2) Enkripsi Data. Perancangan teknik keamanan yang digunakan pada sistem ini seperti berikut:

1. Perancangan Level Pengguna (*Session*)

Teknik keamanan Level Pengguna (*Session)* merupakan sebuah teknik yang dapat membatasi pengguna dalam mengakses suatu halaman. Secara garis besar aplikasi *E-Rent* memiliki 3 *session* yaitu administrator, *viewer/public*, dan penyedia sewa/penyewa/*member*. Berikut ini teknik keamanan *session*.

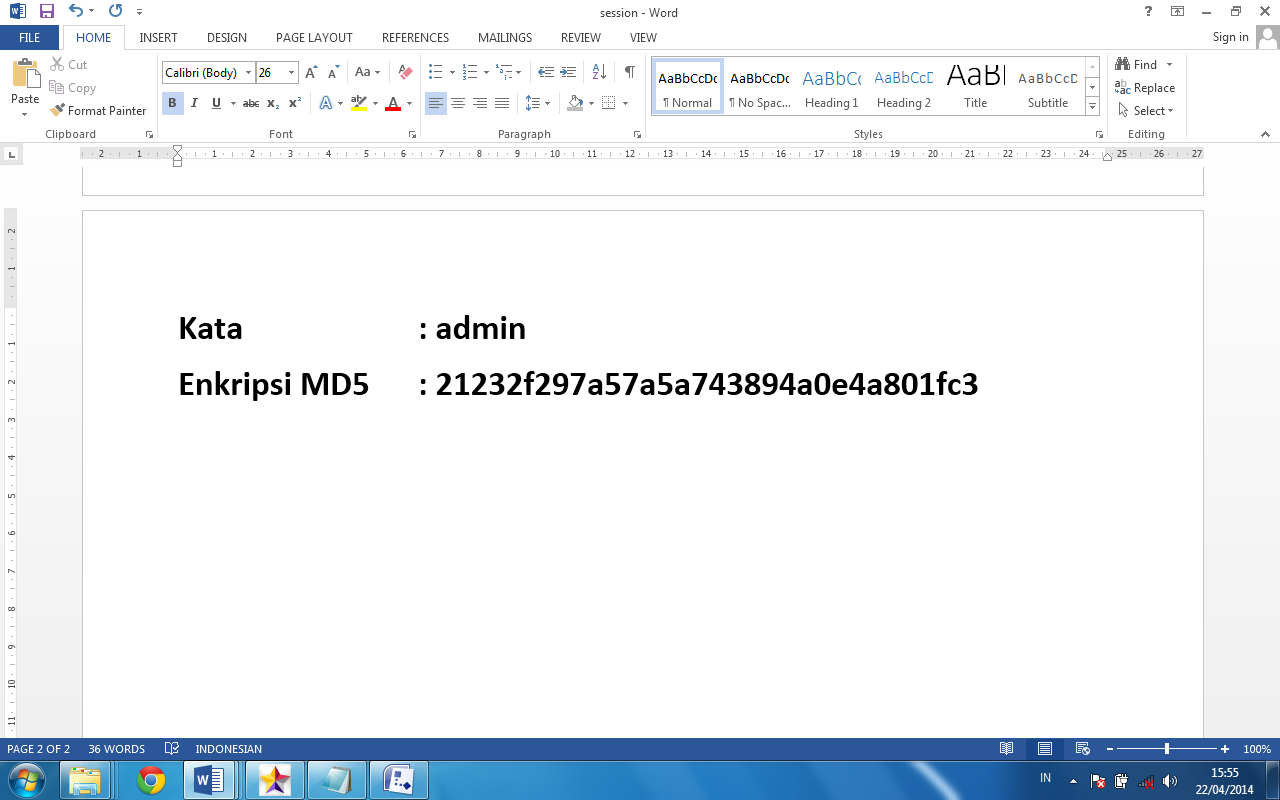


**Gambar 39.** Teknik Keamanan Level Pengguna (*Session*)

Berdasarkan gambar di atas, sistem akan menampilkan data berdasarkan *session level* pengguna. Jika *session* *level*=admin, tampilkan halaman administrator, jika *session* level=penyewa/*member*, tampilkan halaman penyewa/*member* sarana, dan jika *session* *level*= *viewer*, tampilkan halaman *viewer.*

1. Perancangan Teknik Keamanan Enkripsi Data

Perancangan teknik enkripsi data yang digunakan pada aplikasi ini yaitu teknik enkripsi MD5 dari *MySQL* tanpa melakukan modifikasi teknik dan metode. Enkripsi MD5 ini digunakan pada *password*. Saat pengguna memasukkan *password* ke dalam sistem, maka *password* akan terenkripsi sesuai dengan teknik yang digunakan dalam MD5. Contoh pengenkripsian dapat dilihat seperti pada Gambar 41.



**Gambar 40.** Teknik Keamanan Enkripsi MD5

# BAB IV HASIL DAN PEMBAHASAN

## Hasil Rancangan Tampilan

Setelah tahapan perancangan selesai, rancangan tersebut diubah menjadi kode program untuk menampilkan sistem yang dibuat. Berdasarkan rancangan *interface* pada bab 3, maka dibuatlah aplikasi *E-Rent (Electronic Rental)* dengan hasil sebagai berikut:

1. Halaman Tampilan *Input*

Halaman tampilan *input* merupakan halaman untuk menerima perintah dan masukan dari pengguna berupa form dan data *upload*. Berikut ini halaman *input* aplikasi *E-Rent* berbasis Android:

* 1. **Halaman Registrasi**

Halaman registrasi merupakan tampilan untuk pengguna mendaftar ke dalam sistem. Halaman ini akan muncul setelah pengguna menekan tombol register pada halaman *home*. Berikut tampilan halaman registrasi:



**Gambar 41.** Tampilan Halaman Registrasi

Pada halaman registrasi terdapat beberapa *form* diantaranya Nama Lengkap, Alamat, Email, Telepon, *Username* dan *Password* yang harus diisi pengguna agar dapat masuk ke dalam sistem. Pada saat tombol daftar dipilih, maka data pengguna akan dicek apakah ada kesalahan atau tidak. Jika ada, akan tampil pesan kesalahan sesuai jenis kesalahan yang dilakukan. Jika tidak ada kesalahan, maka data pengguna akan disimpan pada tabel *user* di *database.*

Setelah mendaftar, pengguna akan mendapat konfirmasi bahwa pendaftaran sudah berhasil. Kemudian sistem akan mengarahkan ke halaman *login*.

Untuk tampilan dalam halaman ini diatur oleh *activity* *RegisterActivity.java.* Berikut *source code* pada halaman registrasi:

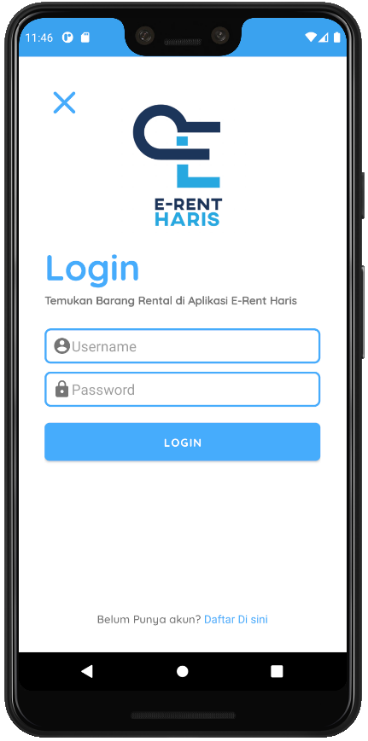
public void tambahData(String nama\_user, String alamat\_user, String email\_user, String telp\_user, String username, String pass){  
 *//koneksi ke file create.php* AndroidNetworking.*post*(api.URL\_REGISTER)  
 .addBodyParameter("nama\_user", nama\_user).addBodyParameter("alamat\_user",alamat\_user) .addBodyParameter("email\_user", email\_user) .addBodyParameter("telp\_user", telp\_user) .addBodyParameter("username", username) .addBodyParameter("password", pass).setPriority(Priority.*MEDIUM*)  
 .build()  
 .getAsJSONObject(new JSONObjectRequestListener() {  
 @Override  
 public void onResponse(JSONObject response) {  
 *//Handle Response* Log.*d*(*TAG*, "onResponse: " + response);  
 Toast.*makeText*(getApplicationContext(),"Data berhasil ditambahkan" , Toast.*LENGTH\_SHORT*).show();  
 Intent i = new Intent(RegisterActivity.this,LoginActivity.class);  
 startActivity(i);

}

*Source code* di atas menunjukkan bahwa untuk melakukan registrasi, diperlukan beberapa *file* yang saling terkait satu sama lain. Baik *file* tersebut berupa *view*, *model*, atau *controller*.

* 1. **Halaman Login**

Halaman *Login* merupakan halaman yang digunakan oleh pengguna yang sudah terdaftar agar dapat masuk ke sistem. Pengguna hanya perlu memasukkan data berupa *username* dan *password* yang dibuat pengguna saat melakukan registrasi. Tampilan halaman *login* dapat dilihat pada gambar berikut:



**Gambar 42.** Halaman *Login*

Setelah semua data diisi, pengguna mengeklik tombol *Login*. Selanjutnya sistem akan melakukan pengecekan terhadap data dan mencocokkannya dengan data yang ada pada tabel *user*. Jika cocok, maka pengguna akan diarahkan ke halaman utama. Jika ada data yang salah, maka akan muncul pesan kesalahan sesuai dengan jenis kesalahan yang dilakukan.

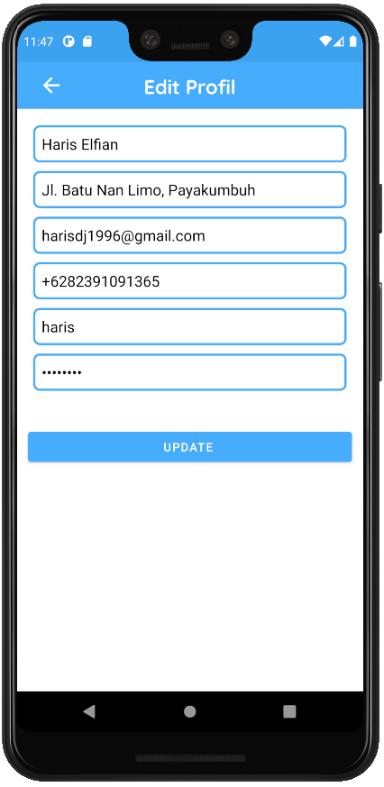
Untuk tampilan dalam halaman ini diatur oleh *activity LoginActivity.java.* Berikut *source code* untuk halaman *login*:

private void getLogin() {  
 alertDialog.show();  
 AndroidNetworking.*post*(api.URL\_LOGIN)  
 .addBodyParameter("username", edtusername.getText().toString())  
 .addBodyParameter("password", edtpass.getText().toString())  
 .setPriority(Priority.*MEDIUM*)  
 .build()  
 .getAsJSONObject(new JSONObjectRequestListener() {  
 @Override  
 public void onResponse(JSONObject response) {  
 try {  
 alertDialog.hide();  
 int stat = response.getInt("status");  
 String message = response.getString("message");  
 Log.*d*("sukses", "code" + response);

Data pengguna yang digunakan untuk *login* ke dalam sistem diambil dari *database* yang diatur pada model *helper.java.*

* 1. **Halaman Edit Profil**

Halaman edit profil merupakan halaman yang disediakan untuk mengubah data umum pengguna seperti Nama Lengkap, Alamat, E-Mail, Nomor Telepon, *username* dan *password*. Tampilan halaman edit profil dapat dilihat pada gambar berikut:



**Gambar 43.** Gambar Halaman Edit Profil

Halaman edit *profil* akan muncul ketika pengguna mengeklik tombol edit *profil* pada halaman *profil*. Setelah pengguna mengisi semua data perubahan pada halaman ini, pengguna akan mengeklik tombol *update* perubahan untuk menyimpan data tersebut. Pengguna akan kembali ke halaman *profil* dan data perubahan ini akan tersimpan ke dalam *database* untuk ditampilkan pada halaman *profil.*

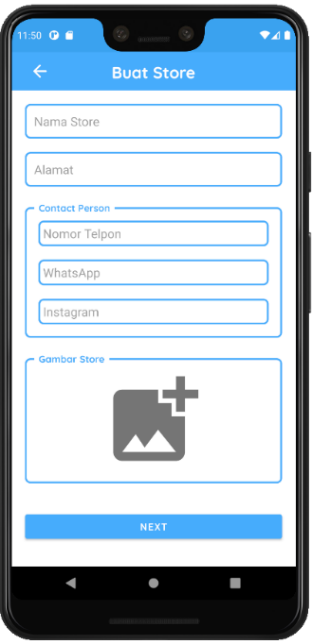
Untuk tampilan dan perubahan data dalam halaman ini diatur oleh *Activity* *EditProfilAvtivity.java.* Berikut ini *source code* untuk perubahan data profil:

public void onResponse(JSONObject response) {  
 Intent i = new Intent(EditProfilActivity.this, ProfilActivity.class);  
 startActivity(i);  
 tinyDB.putString("id\_user", id);  
 tinyDB.putString("keyNamaUser", edt\_namauser\_register.getText().toString());  
 tinyDB.putString("keyAlamatUser", edt\_alamatuser\_register.getText().toString());  
 tinyDB.putString("keyEmailUser", edt\_emailuser\_register.getText().toString());  
 tinyDB.putString("keyTelpUser", edt\_telpuser\_register.getText().toString());  
 Toast.*makeText*(getApplicationContext(), "Data berhasil di update..."  
 , Toast.*LENGTH\_LONG*).show();}

Proses penyimpanan dan pengambilan data ini diatur dalam *ProfilActivity.java.*

* 1. **Halaman *Input* Store**

Halaman *input store* adalah halaman yang disediakan untuk pengguna yang ingin menyewakan barangnya ke dalam sistem. Setelah sebelumnya pengguna masuk ke dalam sistem, pengguna yang ingin menyewakan barang akan masuk ke portal penyedia sewa. Untuk dapat menyewakan barang, pengguna harus membuat toko dengan memasukkan beberapa data, seperti gambar berikut:



**Gambar 44.** Tampilan Halaman *Input Store*

Pengguna memasukkan nama toko, alamat toko, *contact person* yaitu nomor yang bisa dihubungi, serta foto untuk toko. Setelah semua data diisi, pengguna akan mengeklik tombol *next*, selanjutnya data diproses, dan disimpan di dalam *database* untuk ditampilkan pada halaman *store* dan halaman *home*.

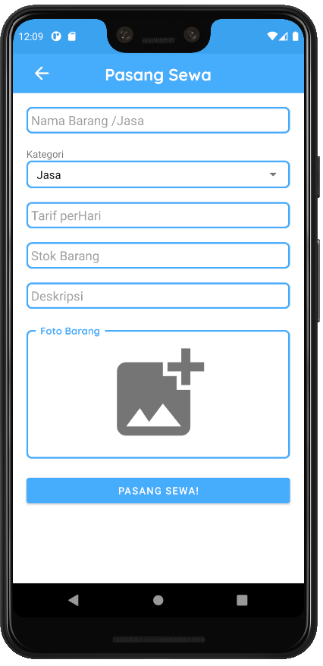
Untuk tampilan dalam halaman ini diatur oleh *FormStoreActivity.java.* Berikut ini *source code* untuk halaman *input store*:

public void aksiTambahStore(){  
 next.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String nama\_store = edt\_namaStore.getText().toString();String alamat\_store = edt\_alamatStore.getText().toString();String telp\_Store = edt\_telpStore.getText().toString();String wa\_store = edt\_WAStore.getText().toString();String ig\_store = edt\_IGStore.getText().toString();if (nama\_store.equals("") || alamat\_store.equals("") || telp\_Store.equals("") || wa\_store.equals("") || ig\_store.equals("")) {  
 Toast.*makeText*(getApplicationContext(), "Semua data harus diisi", Toast.*LENGTH\_SHORT*).show();} else {  
 tambahData(nama\_store, alamat\_store, telp\_Store, wa\_store, ig\_store); *//memanggil fungsi tambahData()*

Halaman *input store* juga terkait dengan Model\_*DetailStore.java* untuk menyimpan data ke dalam *database* dan *view DetailStoreActivity.java* untuk menampilkan hasil data yang sudah di *input.*

* 1. **Halaman *Input* data barang**

Halaman *input* data barang berisi *form* yang digunakan oleh pengguna untuk memasukkan data barang yang disewakan. Pengguna yang ingin memasukkan data barang diharuskan sudah *login* ke dalam sistem, dan sudah mempunyai toko. Jika sudah, maka pengguna dapat memasukkan data barang dengan menekan tombol *Store* pada halaman portal profil. Berikut tampilan halaman *input* data barang:



**Gambar 45.** Halaman *Input* Barang

Pengguna memasukkan data berupa nama barang, pilih kategori barang/jasa, harga barang per hari, deskripsi tentang barang yang akan disewakan. Pengguna juga memasukkan jumlah stok barang yang tersedia dan juga foto barang. Setelah pengguna memasukkan data, pengguna mengeklik tombol Pasang Sewa!, data diproses dan disimpan di dalam *database* untuk selanjutnya akan ditampilkan pada halaman barang dan halaman *home*.

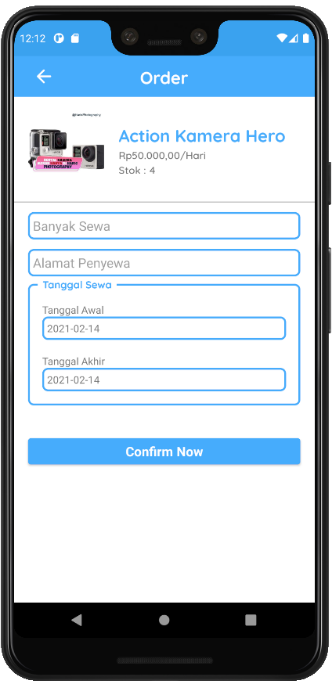
Halaman *input* data barang diatur di *Activity* *PasangSewaActivity.java.* Berikut *source code* untuk menampilkan halaman *input* data barang:

public void aksiPasangSewa() {  
 pasang\_sewa.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String namaBarang = edt\_namaBarang.getText().toString();String tarifBarang = edt\_tarif.getText().toString();String stokBarang = edt\_stokBarang.getText().toString();String deskripsi = edtdeskripsi.getText().toString();if (namaBarang.equals("") || tarifBarang.equals("") || stokBarang.equals("") || deskripsi.equals("")) {  
 Toast.*makeText*(getApplicationContext(), "Semua data harus diisi", Toast.*LENGTH\_SHORT*).show();} else {  
 tambahData(namaBarang, tarifBarang, stokBarang, deskripsi); *//memanggil fungsi tambahData()*

Halaman *input* barang terkait dengan *Model\_Barang.java* untuk menyimpan dan memanggil data dari *database* dan *BarangActivity.java* untuk menampilkan data barang yang sudah berhasil masuk ke dalam sistem. Selain ini juga terkait dengan *Model\_Kategori.java* dan *TinyDB.java.*

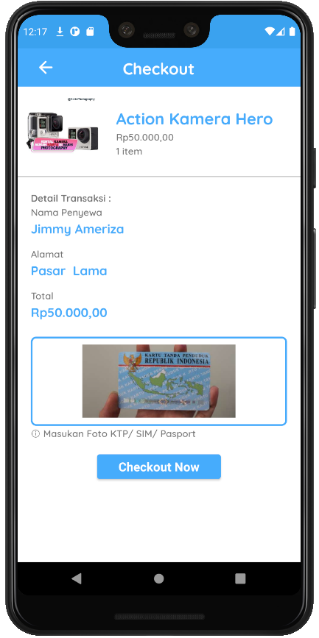
* 1. **Halaman Pemesanan**

Halaman pemesanan merupakan halaman yang berfungsi untuk melakukan pemesanan barang yang akan di sewa sesuai kebutuhan pengguna. Halaman pemesanan hanya dapat di akses oleh pengguna yang sudah terdaftar ke dalam sistem. Pada halaman pemesanan ini, terdapat beberapa data yang harus dimasukkan sebelum akhirnya pengguna mendapatkan barang yang sudah dipesan. Berikut tampilan langkah-langkah dalam pemesanan barang yang akan disewa:



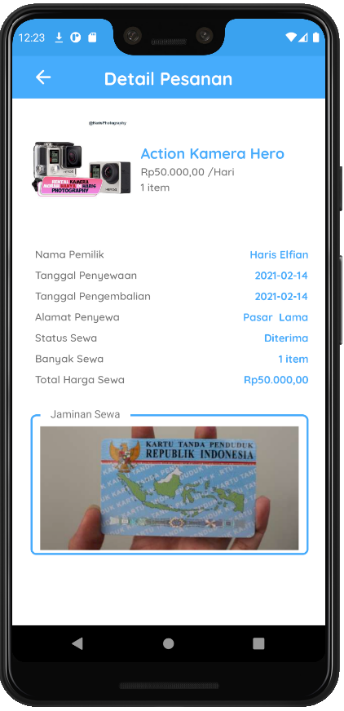
**Gambar 46.** Halaman Pemesanan

Setelah pengguna memasukkan jumlah barang, alamat penyewa, tanggal peminjaman, dan tanggal pengembalian barang yang ingin disewa, selanjutnya pengguna mengeklik tombol *Confirm Now* akan tampil gambar seperti berikut:



**Gambar 47.** Halaman *Checkout* Pemesanan

Pada halaman detail pemesanan diperlihatkan semua data terkait pemesanan yang dilakukan oleh pengguna dan menginputkan foto KTP sebagai jaminan digital. Data pemesanan ini sudah masuk ke penyedia sewa. Penyewa dapat membatalkan atau tetap melanjutkan pemesanan. Jika ingin melanjutkan, pengguna dapat mengeklik tombol *Checkout Now* dan akan mendapatkan bukti pemesanan di bagian menu Pesanan Saya. Berikut tampilan bukti pemesanan:

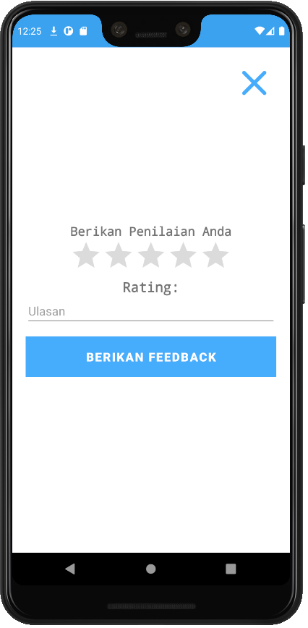


**Gambar 48.** Bukti Pemesanan

Semua tampilan pada halaman pemesanan, diatur dalam dua *package* yaitu *package checkout dan Order* yang di dalamnya terdapat beberapa *file* java mulai dari memasukkan data pemesanan sampai detail pemesanan.

* 1. **Halaman *Input Review* dan *Rating***

Halaman *input review (testimony)* dan *rating* merupakan halaman yang digunakan oleh pengguna untuk memberikan kesan atau pesan, dan penilaian tentang kepuasan pengguna terhadap barang atau pelayanan yang diberikan penyedia sewa. Untuk dapat masuk ke halaman *input review* dan *rating*, pengguna harus menyelesaikan pemesanan barang terlebih dahulu. Jika sudah, maka akan keluar menu beri *testimony* pada “Pesanan Saya”, pengguna mengeklik tombol beri *testimony*, maka akan muncul tampilan halaman input review dan rating sebagai berikut:



**Gambar 49.** Halaman *Input Review* dan *Rating*

Pengguna memasukkan pesan yang akan ditulis sebagai *testimony* dan memberikan jumlah bintang yang sesuai untuk *rating*. Setelah data terisi, pengguna mengeklik tombol berikan Feedback. Jika sudah, maka akan muncul pesan *testimony* sudah berhasil dimasukkan. Data *testimony* dan *rating* tersimpan di dalam *database*, dan ditampilkan pada halaman detail barang.

Untuk memasukkan data *review* dan *rating* melibatkan dua tabel yaitu tabel *testimony* dan tabel bintang untuk menyimpan data *rating*. Setelah data dimasukkan, data tersebut disimpan ke dalam *database* dan ditampilkan pada halaman barang. Berikut *source code* untuk halaman *input review* dan *rating*:

public void tambahData(String ulasan, String rating){AndroidNetworking.*post*(api.URL\_RATING)  
 .addBodyParameter("id\_ulasan","")  
 .addBodyParameter("id\_barang",id\_barang).addBodyParameter("id\_user",id\_user).addBodyParameter("review", ulasan).addBodyParameter("bintang",rating).setPriority(Priority.*MEDIUM*)  
 .build()  
 .getAsJSONObject(new JSONObjectRequestListener() {  
 @Override  
 public void onResponse(JSONObject response) {  
 Intent intent = new Intent(RatingActivity.this, MainActivity.class);  
 startActivity(intent);  
 *//Handle Response* Log.*e*(*TAG*, "onResponse: " + response);  
 Toast.*makeText*(getApplicationContext(),"Data berhasil ditambahkan" , Toast.*LENGTH\_SHORT*).show();

//memunculkan Toast saat data berhasil ditambahkan

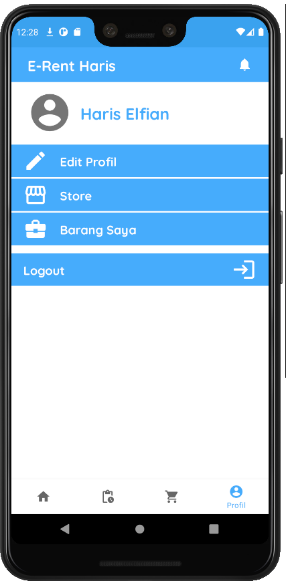
Halaman *input review* dan *rating* terkait dengan model *ModelRating.java* untuk menyimpan data ke *database* dan *RatingActivity.java* untuk menampilkan data. Halaman ini juga terkait dengan data barang.

1. Halaman Tampilan *Output*

Halaman tampilan *output* merupakan halaman untuk menampilkan data yang sebelumnya telah dimasukkan. Berikut ini halaman *output* aplikasi E-Rent berbasis Android:

1. **Halaman Utama**

Halaman utama adalah halaman yang tampil setelah pengguna masuk ke dalam sistem. Halaman utama berisi beberapa sub menu seperti pada gambar berikut ini:



**Gambar 50.** Tampilan Halaman Utama

Pada halaman utama ini, pengguna Akan ada pilihan menu edit profil, *store* dan barang saya. Pada halaman barang saya pengguna dapat melihat barang yang akan, sedang, dan sudah disewa. Data yang tampil pada halaman ini diambil dari *database* pada tabel pemesanan.

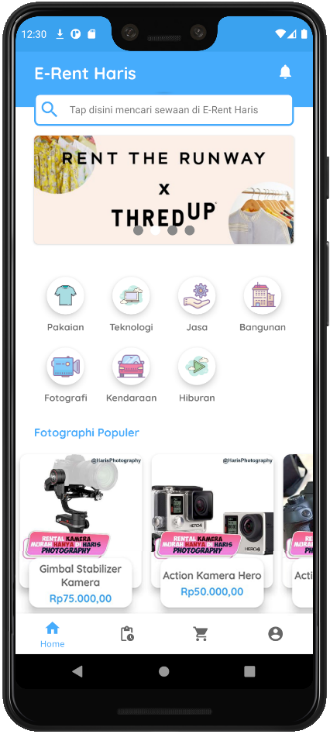
Selain menu barang Saya, juga terdapat beberapa menu diantaranya profil, edit profil dan *log out*. Tampilan untuk menu tersebut diatur pada *Activity FragmentActivity.java*. Berikut ini *source code* untuk menampilkan menu:

tinyDB = new TinyDB(getContext());  
txt\_username\_profil = view.findViewById(R.id.*username\_profl*);  
txt\_username\_profil.setText(tinyDB.getString("keyNamaUser"));  
if (!tinyDB.getBoolean("keyLogin")){  
 Intent intent = new Intent(getContext(),LoginActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);}  
logout = view.findViewById(R.id.*btn\_logout*);  
logout.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 tinyDB.clear();  
 Intent i = new Intent(getContext(), MainActivity.class);  
 i.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 i.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(i);

Halaman ini terkait dengan beberapa tabel yaitu *user, store*, barang, dan pemesanan.

1. **Halaman** ***Home***

Halaman *home* merupakan halaman yang pertama kali dilihat oleh pengguna saat mengakses sistem. Komponen yang tampil di halaman *home* ini diatur dalam *MainActivity.java*. Tampilan halaman utama dapat dilihat pada gambar berikut:



**Gambar 51.** Halaman *Home*

Pada halaman *home* terdapat beberapa menu dan sub-sub menu, yaitu:

* + 1. Menu kategori, berisi sub-sub menu berupa kategori barang yang dimasukkan oleh admin
    2. *Form* pencarian, merupakan *form* yang digunakan untuk melakukan pencarian terhadap barang atau jasa.
    3. Toko populer, merupakan informasi yang disajikan dalam bentuk *thumbnail* untuk menginformasikan barang yang populer terdaftar.
    4. Jasa populer, merupakan informasi yang disajikan dalam bentuk *thumbnail* untuk menginformasikan jasa yang populer terdaftar.

Untuk menampilkan halaman utama, digunakan fungsi *Navigation Bottom* yang terdapat di *MainActivity*.java. Berikut *source code* halamannya:

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);title = findViewById(R.id.*tv\_toolbar*);  
 title.setText("E-Rent Haris");  
 notif = findViewById(R.id.*ib\_notif*);  
 notif.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MainActivity.this, NotificationActivity.class);  
 startActivity(intent);  
 }});  
 getFragmentPage(new Fragment\_Home());

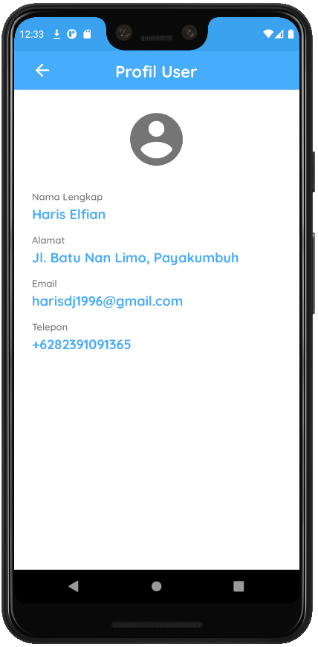
BottomNavigationView bottomNavigation = findViewById(R.id.*bottomNavigationView*);  
bottomNavigation.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 Fragment fragment = null;  
 *//Menantukan halaman Fragment yang akan tampil* switch (item.getItemId()) {  
 case R.id.*home*:  
 fragment = new Fragment\_Home();  
 break;  
 case R.id.*profil*:  
 fragment = new Fragment\_Profil();  
 break;  
 case R.id.*pesanan*:  
 fragment = new Fragment\_Pesanan();  
 break;  
 case R.id.*pesanan\_masuk*:  
 fragment = new Fragment\_PesananMasuk();  
 break;}

return getFragmentPage(fragment);

}});

1. **Halaman Profil**

Halaman profil di sini hanya berisi informasi umum dari pengguna berupa nama lengkap, alamat, email dan telepon. Berikut tampilan halaman profil:



**Gambar 52.** Tampilan Halaman Profil

Pada halaman profil menampilkan informasi umum, seperti nama lengkap*,* alamat,emaildan telepon pengguna. Informasi *profil* diambil dari *database* pada tabel *user*. Berikut *source code* untuk menampilkan data pada halaman profil:

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profil*);  
 tinyDB = new TinyDB(this);  
 nama = findViewById(R.id.*nama*);  
 nama.setText(tinyDB.getString("keyNamaUser"));  
 alamat = findViewById(R.id.*alamat*);  
 alamat.setText(tinyDB.getString("keyAlamatUser"));  
 email = findViewById(R.id.*email*);  
 email.setText(tinyDB.getString("keyEmailUser"));  
 telp = findViewById(R.id.*telp*);  
 telp.setText(tinyDB.getString("keyTelpUser"));  
 title = findViewById(R.id.*tv\_toolbar*);  
 title.setText("Profil User");  
 btnClose = findViewById(R.id.*ib\_back*);  
 btnClose.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent i = new Intent(ProfilActivity.this,

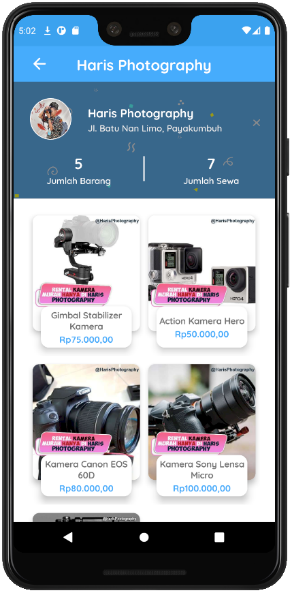
MainActivity.class);  
 startActivity(i);  
 finish();

}});

Halaman profil dihubungkan oleh api *select\_user.php* dan untuk memanggil data dari database diatur dalam *TinyDB.*

1. **Halaman Store**

Halaman *store*/tokomerupakan halaman untuk menampilkan toko yang sudah didaftarkan oleh pengguna ke dalam sistem. Halaman ini hanya bisa diakses oleh pengguna yang sudah terdaftar dan masuk ke dalam sistem. Untuk dapat masuk ke halaman *store* pengguna terlebih dahulu harus masuk ke dalam menu penyedia sewa, lalu klik toko. Berikut tampilan halaman *store:*



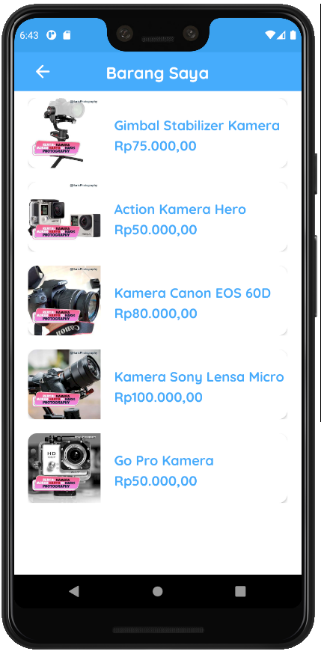
**Gambar 53.** Halaman *Store*

Halaman *store* di desain dengan bentuk kotak yang menampilkan data toko dan barang apa saja yang terdaftar dalam toko tersebut. Berikut *source code* untuk menampilkan halaman *store*:

public void getDataStore() {  
 AndroidNetworking.*get*(api.URL\_DETAIL\_STORE + id\_store)  
 .setPriority(Priority.*LOW*)  
 .build()  
 .getAsJSONObject(new JSONObjectRequestListener() {  
 @Override  
 public void onResponse(JSONObject response) {  
 try {  
 Log.*d*("tampil Store", "response:" + response + id\_store);  
 dataStore.clear();  
 binding.jmlSewa.setText(response.getString("sewa"));  
 JSONArray res = response.getJSONArray("res");  
 Gson gson = new Gson();  
 for (int i = 0; i < res.length(); i++) {  
 JSONObject data = res.getJSONObject(i);  
binding.namaStore.setText(data.getString("nama\_store")); binding.alamatStore.setText(data.getString("alamat\_store")); Picasso.*get*().load(api.URL\_GAMBAR\_U + data.getString("gambar\_store")).into(binding.fotoStore);  
Model\_IsiKategori Isi = gson.fromJson(data + "", Model\_IsiKategori.class);  
 dataStore.add(Isi);  
binding.toolbar.tvToolbar.setText(data.getString("nama\_store")); }  
 binding.jmlBarang.setText(dataStore.size() + "");  
 Adapter\_IsiKategori adapter = new Adapter\_IsiKategori(dataStore);  
 recycler\_detailStore.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
 } catch (JSONException e) {  
 e.printStackTrace();}}

1. **Halaman Barang**

Halaman barang merupakan halaman yang menampilkan *list* barang apa saja yang akan disewakan oleh penyedia sewa. Halaman ini hanya bisa diakses oleh pengguna yang sudah terdaftar dan masuk ke dalam sistem serta sudah memiliki toko. Untuk dapat masuk ke halaman barang pengguna terlebih dahulu harus masuk ke dalam menu penyedia sewa, lalu klik barang. Berikut tampilan *list* halaman barang:



**Gambar 54.** Halaman Barang

Pada halaman barang ditampilkan foto, nama barang, harga sewa barang per hari. Informasi pada halaman barang di ambil dari *database* pada tabel barang. Berikut *source code* untuk menampilkan barang:

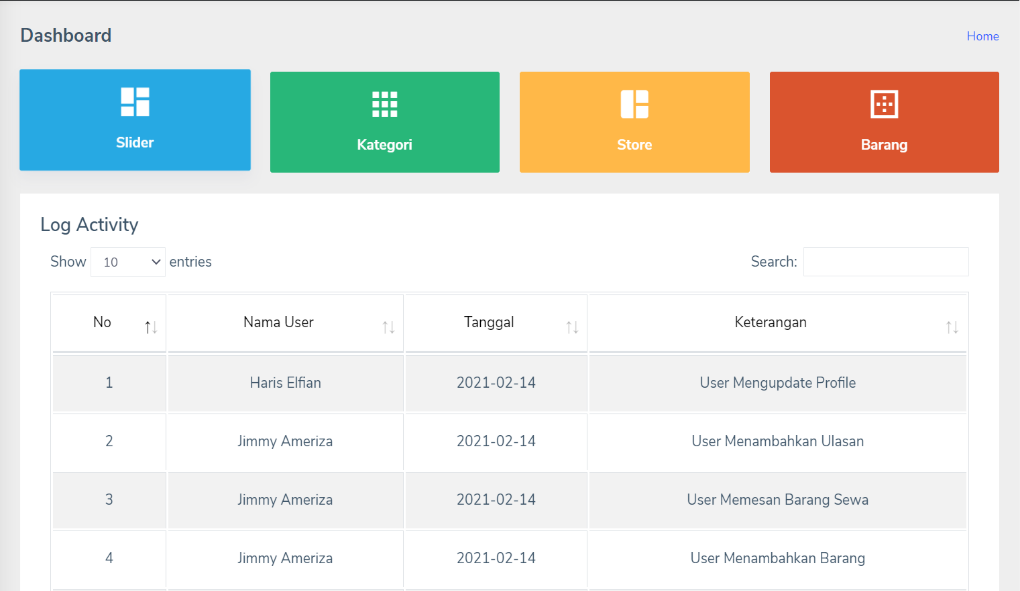
public void getBarang(){  
 AndroidNetworking.*get*(api.URL\_BARANG\_DISEWAKAN+id\_user)  
 .setPriority(Priority.*LOW*)  
 .build()  
 .getAsJSONObject(new JSONObjectRequestListener() {  
 @Override  
 public void onResponse(JSONObject response) {  
 try{  
 if (response.getString("status").equalsIgnoreCase("sukses")) {  
 kosong.setVisibility(View.*GONE*);  
 recycler\_barang.setVisibility(View.*VISIBLE*);  
 Log.*d*("tampil Store", "response:" + response + id\_user);  
 JSONArray res = response.getJSONArray("res");  
 Gson gson = new Gson();  
 dataBarang.clear();  
 for (int i = 0; i < res.length(); i++) {  
 JSONObject data = res.getJSONObject(i);  
Model\_Barang Isi = gson.fromJson(data + "", Model\_Barang.class);  
 dataBarang.add(Isi);}  
 getAdapter();else {  
 kosong.setVisibility(View.*VISIBLE*);  
recycler\_barang.setVisibility(View.*GONE*);  
 Toast.*makeText*(BarangActivity.this, "Data Kosong", Toast.*LENGTH\_SHORT*).show();}  
 } catch (JSONException e) {e.printStackTrace();}}

1. Halaman Tampilan Lainnya
   1. **Halaman Admin**

Halaman admin merupakan halaman yang ditampilkan untuk pengguna dengan hak akses admin. Admin memiliki tanggung jawab mengelola sistem dan fitur-fitur yang ada. Komponen – komponen yang ada pada halaman ini diatur oleh *controller* *admin.php*. Ada beberapa halaman yang terkait dengan admin yaitu :

1. Halaman *Indeks*

Halaman indeks adalah halaman yang akan muncul setelah admin masuk ke dalam sistem. Berikut tampilan indeks admin:



**Gambar 55.** Halaman Tampilan *Index* Admin

Pada halaman *index* ini, ditampilkan informasi beberapa aktivitas yang terjadi di dalam sistem. Berikut source code untuk menampilkan halaman *index* admin:

<?php

class Dashboard extends CI\_Controller

{

    public function \_\_construct()

    {

        parent::\_\_construct();

        $this->load->model('Log\_Model', 'logModel');

        if (!$this->session->userdata('username')) {

            $this->session->set\_flashdata('error', 'Anda Harus Login.');

            redirect('login');

        }

    }

    public function index()

    {

        $data['log'] = $this->logModel->getAll();

        $data['join\_log'] = $this->logModel->join\_all();

        $this->load->view('admin/component/header');

        $this->load->view('admin/component/top');

        $this->load->view('admin/component/menu');

        $this->load->view('admin/index', $data);

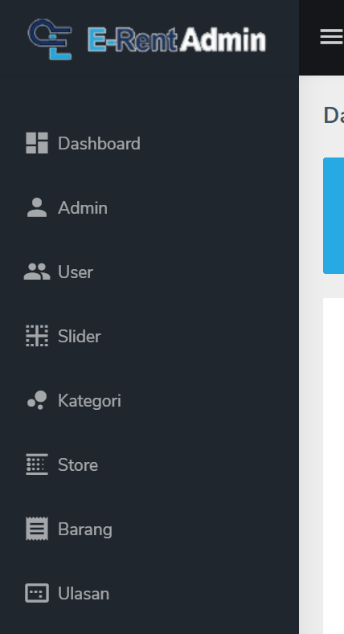
        $this->load->view('admin/component/footer');

    }

}

1. Halaman *left*

Halaman *left* berisi beberapa menu untuk admin dalam mengelola sistem dan *user*. Berikut tampilan *left* pada admin.

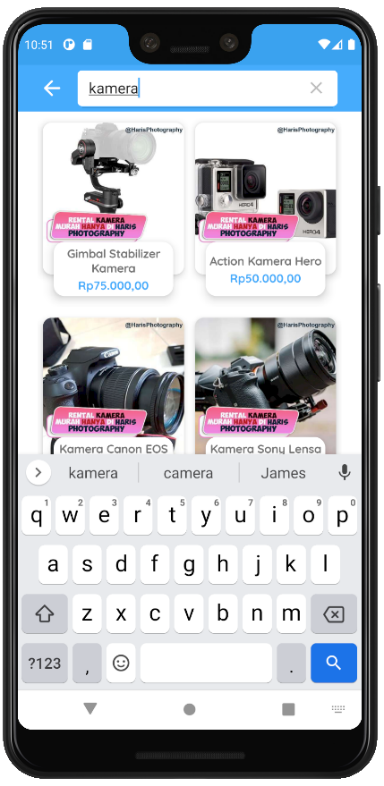


**Gambar 56.** Halaman *Left* Admin

Halaman *left* admin terdiri atas beberapa menu seperti menu barang untuk melihat barang apa saja yang sudah masuk ke dalam sistem, menu *store* yaitu data *store* yang terdaftar dalam sistem, *user, slider, kategori, barang*, hingga menu Ulasan.

* 1. **Halaman Pencarian**

Halaman pencarian merupakan salah satu fitur utama dari aplikasi ini. Halaman pencarian memudahkan pengguna dalam menemukan barang yang dicari sesuai *keyword* yang dimasukkan. Semua pengguna dapat menggunakan fitur pencarian ini tanpa harus daftar atau masuk ke dalam sistem. Berikut tampilan halaman pencarian aplikasi *E-Rent*:



**Gambar 57.** Halaman Pencarian

Halaman pencarian dapat dilakukan oleh pengguna dengan menekan tombol pencarian dengan lambang kaca pembesar. Pengguna bisa mencari barang berdasarkan nama barang atau nama toko, kemudian setelah hasil pencarian muncul, pengguna dapat mengurutkan hasil pencarian berdasarkan nama. Berikut *source code* untuk menampilkan halaman pencarian:

**private void** getProduk(String query) {  
 Log.*d*(**"masuk key"**,**"url | "**+**api**.**URL\_SEARCH**+query);  
 **binding**.**progres**.setVisibility(View.***VISIBLE***);  
 AndroidNetworking.*get*(**api**.**URL\_SEARCH**+query)  
 .setPriority(Priority.***HIGH***)  
 .build()  
 .getAsJSONObject(**new** JSONObjectRequestListener() {  
 @Override  
 **public void** onResponse(JSONObject response) {  
 **try**{  
 Log.*d*(**"tampil barang key"**,**"response:"**+response+query);  
 **binding**.**progres**.setVisibility(View.***GONE***);  
 **dataIsiKategori**.clear();  
 **if** (response.getString(**"status"**).equalsIgnoreCase(**"gagal"**)){  
 }**else** {  
 JSONArray res = response.getJSONArray(**"res"**);  
 Gson gson = **new** Gson();  
 **for** (**int** i=0; i<res.length(); i++){  
 JSONObject data = res.getJSONObject(i);  
 Model\_IsiKategori Isi = gson.fromJson(data + **""**, Model\_IsiKategori.**class**);  
 **dataIsiKategori**.add(Isi);}}  
 Adapter\_IsiKategori adapter = **new** Adapter\_IsiKategori(**dataIsiKategori**);  
 **binding**.**rvProdukS**.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
 } **catch** (JSONException e) {  
 e.printStackTrace();}}

## Pengujian Sistem

Setelah melakukan perancangan dari aplikasi *E-Rent* berbasis Android, maka pada tahap ini dilakukan pengujian terhadap masing-masing halaman dan menu yang ada pada sistem untuk mengetahui apakah sistem yang telah dibuat sesuai dengan rancangan yang diinginkan. Pengujian juga dimaksudkan guna melakukan evaluasi terhadap sistem untuk melihat apakah sudah layak untuk digunakan.

1. **Pengujian *Public***

Pengujian *public* merupakan pengujian yang dilakukan kepada menu yang bisa diakses oleh public (*user* dan bukan *user*).

**Tabel 35.** Pengujian *Test* Unit dan Uji Coba Sistem pada Public

| **No** | **Modul** | **Unit** | **Hasil Pengujian** | |
| --- | --- | --- | --- | --- |
| **Berhasil** | **Tidak Berhasil** |
| 1. | *View Home* | 1. Halaman *Home* |  |  |
| 1. Tampil Halaman Informasi Barang dan Jasa |  |  |
| 1. Halaman *Detail* Barang |  |  |
| 1. Halaman Informasi Penyedia sewa |  |  |
| 1. Halaman *Detail* Barang |  |  |
| 2. | *View* Filter Pencarian Barang/Jasa | 1. Halaman Informasi Barang |  |  |
| 1. Halaman *Detail* Barang |  |  |
| 1. Halaman Informasi Pemilik |  |  |
| 1. Halaman *Detail* Review Barang |  |  |
| 3. | *View* Filter Pencarian Jasa | 1. Halaman Informasi Jasa |  |  |
| 1. Halaman Detail Jasa |  |  |
| 1. Halaman Informasi Pemilik |  |  |
| 1. Halaman *Detail* Review Jasa |  |  |
| 4. | *Button* daftar | 1. *Form* pendaftaran |  |  |
| 1. Proses pendaftaran |  |  |

Berdasarkan tabel 35, dapat disimpulkan bahwa keseluruhan uji coba yang dilakukan pada halaman publik berhasil di mana setiap modul yang di uji coba tidak ada yang mengalami kegagalan.

1. **Pengujian Server**

**Tabel 36.** Pengujian *Test* Unit dan Uji Coba Sistem pada *user.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Modul** | **Unit** | **Hasil Pengujian** | |
| **Berhasil** | **Tidak Berhasil** |
| 1. | Registrasi | 1. *Form* registrasi 2. Proses registrasi |  |  |
| 2 | *Login* | 1. *Form* *login* 2. Proses *login* |  |  |
| 3 | *Profil* | 1. *Form* Edit Profil 2. Proses penyimpanan pembaruan profil 3. Tampilan profil |  |  |
| 4 | CRUD Kategori | 1. *Form* *input* kategori 2. Proses *input* katagori 3. *Form* edit kategori 4. *Delete* kategori 5. Halaman tampilan kategori |  |  |
| 5 | CRUD Barang dan Jasa | 1. Form *input* data barang dan jasa 2. Proses penyimpanan 3. *Form* edit barang dan jasa 4. *Delete* barang jasa 5. Halaman tampilan barang dan jasa |  |  |
| 6. | CRUD Store | 1. Form *input* data *store* 2. Proses penyimpanan 3. Halaman tampilan *store* |  |  |
| 7. | Pencarian | 1. Kolom pencarian 2. Halaman tampilan pencarian |  |  |
| 8. | Pemesanan | 1. *Form* *input* pemesanan 2. Proses penyimpanan 3. Halaman bukti pemesanan 4. Pembatalan pemesanan |  |  |
| 9. | *Rating* | 1. Memberikan *rating* 2. Menampilkan *rating* |  |  |
| 10. | *Testimony* | 1. Form *testimony* 2. Halaman tampilan *testimony* |  |  |
| 12 | *Logout* | 1. Button logout 2. Proses *logout* |  |  |

Berdasarkan Tabel 36, dapat disimpulkan bahwa keseluruhan uji coba yang dilakukan pada menu masing-masing menu yang ada pada halaman *user* berhasil di mana setiap modul yang di uji coba tidak ada yang mengalami kegagalan.

# BAB V PENUTUP

## Kesimpulan

Berdasarkan hasil perancangan Aplikasi *E-Rent (Electronic Rental),* maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi *E-Rent (electronic rental)* dibangun dengan menggunakan bahasa pemrograman Java dan menggunakan *CodeIgniter framework* yang diimplementasikan di Kota Payakumbuh.
2. Aplikasi *E-Rent (electronic rental)* menyediakan informasi barang atau jasa dari penyedia sewa di Kota Payakumbuh dan memberikan layanan untuk memesan atau menyewa barang bagi penyewa yang membutuhkan.
3. Aplikasi *E-Rent (electronic rental)* memiliki fitur yang menunjang pengguna dalam menggunakan sistem seperti fitur untuk mempermudah pengguna mencari barang yang akan disewa, fitur *rating* untuk mengapresiasi dan memberi penilaian terhadap barang yang telah disewa.
4. Aplikasi ini dirancang berbasis aplikasi Android menggunakan bahasa pemrograman Java, *MySQL* sebagai *Database Management System* (DBMS), dan *Android Studio* *Integrated Development Environment* (IDE), dan Visual Studio Code dengan *CodeIgniter Framework*.

## Saran

Adapun saran yang diberikan setelah merancang dan membangun aplikasi ini, antara lain

1. Dalam pengembangan berikutnya, aplikasi ini diharapkan tidak hanya untuk penyewaan barang di Kota Payakumbuh saja, tapi bisa mencakup seluruh Indonesia.
2. Pengguna yang ingin menyewakan barang diharapkan dapat memanfaatkan aplikasi ini sebagai alternatif dalam mempromosikan barang yang akan disewa.
3. Diharapkan aplikasi ini dapat memberikan kemudahan dalam proses pencarian dan pemesanan barang untuk disewa

# DAFTAR PUSTAKA

A.S, Rosa dan M. Shalahuddin. (2014). Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung

Akhbar, S., Firmansyah, R., & Yulistia. (2016). Sistem Informasi Pemesanan dan Penyewaan Mobil Berbasis Web Pada CV. Louhan Mobilindo.

Anonim. 2020. Kamus Besar Bahasa Indonesia *Online*. *Kbbi.web.id*. diakses pada

Betha Sidik. *(*2014), *Pemrograman Web dengan PHP*. Informatika:. Bandung

Buchari, R. A. (2016). Implementasi E-Service Pada Organisasi Publik Di Bidang Pelayanan Publik Di Kelurahan Cibangkong Kecamatan Batununggal Kota Bandung. *Sosiohumaniora*, *18*(3), 225. https://doi.org/10.24198/sosiohumaniora.v18i3.8762

Bunafit Nugroho. (2013). *Dasar Pemrograman Web PHP-MySQL dengan Dreamweaver*. Yogyakarta: Penerbit Gava Media.

Fitria, Z. I. (2018). Legalitas Sewa-Menyewa Kios di Hutan Negara Saree ditinjau menurut Konsep Ijarah ‘Ala Al-Manfa’ah (Doctoral dissertation, UIN Ar-Raniry Banda Aceh).

Hakim, M. L., Bagye, W., Fahmi, H., Imtihan, K., Basuki, J., No, R., & Lombok, P. (2019). Pendetaksian Lokasi Rawan Kriminalitas Berbasis Android Kecamatan Praya Timur , Kabupaten Lombok Tengah. *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, *2*(1), 52–59.

Hanif Al Fatta. (2007). *Analisis & Perancangan Sistem Informasi.* Yogyakarta: Penerbit ANDI.

Heriyanto, Y. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car. *Jurnal Intra-Tech*, *2*(2), 64–77.

Ihsan, M., & Siregar, A. P. (2019). Pengaruh E-service Quality Terhadap Keputusan Pengguna Jasa Ojek Online Go-Ride dalam Aplikasi Go-Jek melalui Kepercayaan Konsumen di Kota Jambi. *J-MAS (Jurnal Manajemen Dan Sains)*, *4*(1), 170. https://doi.org/10.33087/jmas.v4i1.85

Mahpudin, E., & Sugiyanto, S. (2018). Perancangan Sistem Informasi Verifikasi Pajak PPH 22, PPH 23, dan PPH Pasal 4(2) di PT. Pindad (Persero) Bandung. *Jurnal Edunomic*, *6*(01), 11–22.

Novita, R., & Sari, N. (2015). Sistem Informasi Penjualan Pupuk Berbasis E-Commerce. Teknoif, 3(2), 1–6.

Nugroho, A. (2009). *rekayasa perangkat lunak menggunakan UML dan JAVA*. Penerbit Andi.

Pressman, S Roger. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: ANDI

Priyadi, Yudi. (2014). Kolaborasi SQL & ERD Dalam Implementasi Database. Yogyakarta: Penerbit Andi.

Rani, L. N. (2017). Perancangan Aplikasi Sistem Informasi Sebagai Media Promosi Menggunakan Bahasa Pemrograman Php Dan Database Mysql Pada Kinali Auto Rental, *4*(2), 224–233.

Rian, H., & Fuandytama, A. (2019). Pelayanan Jasa Laundry Pada Mamah Laundry and Cleaners Serang. *Jurnal Teknologi Informatika Dan Komputer*, *5*(2), 64–69. http://journal.thamrin.ac.id/index.php/jtik/article/view/168/112

Rosiska, E., & Nopiana, P. R. (2017). Aplikasi Sistem Informasi Akutansi Jasa Penyewaan Kapal Berbasis Web Pada PT.Sekumbang Permata Engineering. *Jurnal Akuntansi*, *5*(2), 134–144.

Rowley, J. (2006). An Analysis of the E-service Literature. Toward a Research Agenda. Internet Research, 16(3), 339-359

Ryan Zulham Ramadhani, Hengki Rusdianto, V. Y. (2019). Rancang Bangun Aplikasi Pusat Informasi Sekolah Dengan Penerapan Chatbot Menggunakan Aiml Berbasis Android. *Jurnal Teknik Informatika (JIKA) Univ. Muhammadiyah Tangerang*, *November*, 27–33.

Sari, D., Winantu, A. (2019). Aplikasi Game Edukasi Puzzle Pengenalan Hewan Berbasis Android. *Jurnal Stimik El Rahma.*

Septavia, I., Gunadhi, E., & Kurniawati, R. (2016). Sistem Informasi Penyewaan Mobil Berbasis Web di Jasa Karunia Tour And Travel. *Jurnal Algoritma*, *12*(2), 534–540. https://doi.org/10.33364/algoritma/v.12-2.534

Subekti, D. R. S. I., & Si, M. (2005). *ASOSIASI ANTARA PRAKTIK PERATAAN LABA DAN REAKSI PASAR MODAL DI INDONESIA DRS. IMAM SUBEKTI, M.Si., AK Dosen Fak. Ekonomi Unibraw*. *September*, 15–16.

Subhan, Muhammad. 2007. Pengenalan *Database*. Jakarta: Ilmu Komputer.Com

Sudirman, S.(2016) ANALISIS KOMUNIKASI DATA DENGAN XML DAN JSON PADA WEBSERVICE. *Computer Engineering, Science and System Journal*, *1*(2), 1-6.

Suryana, D. (Ed.). (2018). *Android Studio: Belajar Android Studio* (Vol. 1). Dayat Suryana Independent.

Susianto, D. (2019). *Perancangan Sistem Pemesanan E-Tiket Pada Wisata Di Lampung Berbasis Web Mobil*. *2*, 60–71.

tanggal 6 Februari 2020, pukul 9.22 WIB. Payakumbuh.

UNP. (2014). *Panduan Tugas Akhir / Skripsi Univeristas Negeri Padang.* Padang: UNP.

Verbintenissen, Van. Buku Ketiga. Kitab Undang-Undang Hukum Perdata.

Wati, E. F., & Kusumo, A. A. (2016). Penerapan Metode Unified Modeling Language ( UML ) Berbasis Desktop Pada Sistem Pengolahan Kas Kecil Studi Kasus Pada PT Indo Mada Yasa Tangerang. *Jurnal Informatika*, *5*(1), 24–36

Yasin, M. F., Wintolo, H., & Ayuningtyas, A. (2018). APLIKASI PENYEWAAN MOBIL BERBASIS ANDROID ( STUDI KASUS : TOM TRANSPORT )

Zericka, M.Dhenda. (2013).Penerapan Electronic Service dalam Pengembangan Informasi di Kabupaten Kutai Kartanegara. *E-Journal Ilmu Komunikasi,*ejournal.ilkom.fisip-unmul.org (diakses tanggal 7 Maret 2020)