**TUGAS AKHIR**

**APLIKASI PENJUALAN HAMSTER DAN KELENGKAPANNYA BERBASIS ANDROID DENGAN METODE PROTOTYPE**

**(STUDI KASUS TOKO LOMBOK RABBIT PET HOUSE)**

****

**Oleh:**

**AGUNG RAHMITA ILLAH**

**F1D 016 007**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

**TUGAS AKHIR**

**APLIKASI PENJUALAN HAMSTER DAN KELENGKAPANNYA BERBASIS ANDROID DENGAN METODE PROTOTYPE**

**(STUDI KASUS TOKO LOMBOK RABBIT PET HOUSE)**

****

**Oleh:**

**AGUNG RAHMITA ILLAH**

**F1D 016 007**

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

**APLIKASI PENJUALAN HAMSTER DAN**

**KELENGKAPANNYA BERBASIS ANDROID DENGAN**

**METODE PROTOTYPE (STUDI KASUS TOKO**

# LOMBOK RABBIT PET HOUSE)

Oleh :

**AGUNG RAHMITA ILLAH**

F1D016007

Telah diperiksa oleh Tim Pembimbing :

1. Pembimbing Utama

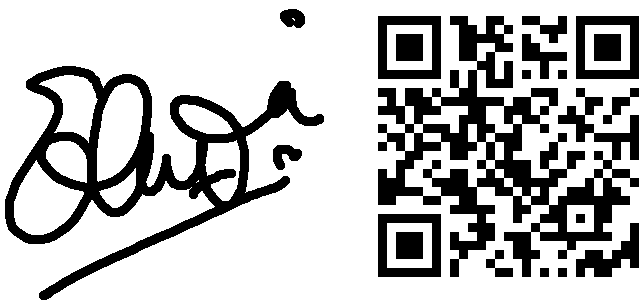


Tanggal: 27/07/2020

**N adiyasari Agitha, S.Kom.,M.MT**

**NIP. 198608132018032001**

1. Pembimbing Pendamping



Tanggal: 27/07/2020

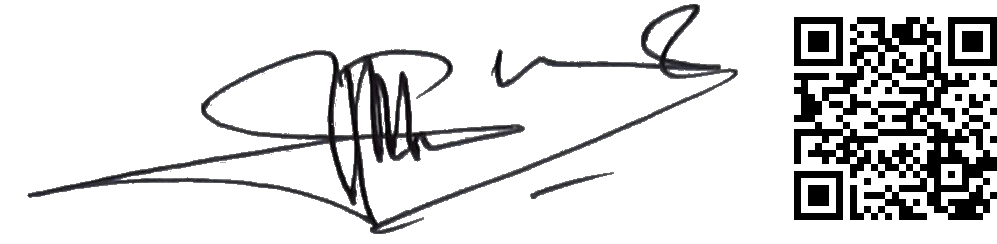
**D r. Eng. Budi Irmawati. S.Kom., MT.**

**NIP. 197210191999032001**

Mengetahui,

Ketua Program Studi Teknik Informatika Fakultas Teknik

Universitas Mataram



**Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.**

**NIP. 197311302000031001**

 **APLIKASI PENJUALAN HAMSTER DAN KELENGKAPANNYA BERBASIS ANDROID DENGAN METODE PROTOTYPE (STUDI KASUS TOKO**

**LOMBOK RABBIT PET HOUSE)**

Oleh :

**AGUNG RAHMITA ILLAH**

F1D016007

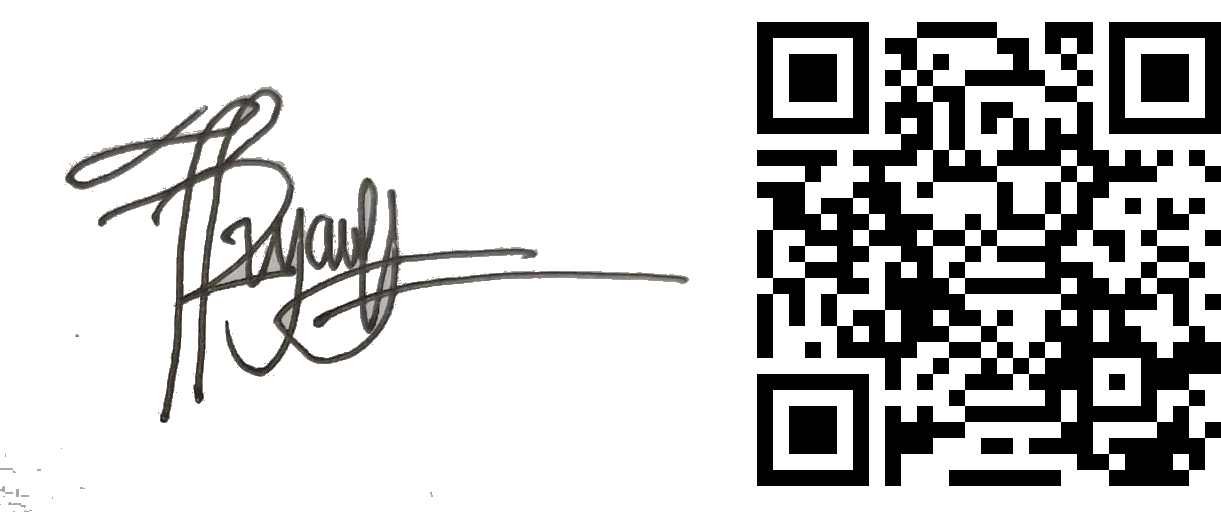
Telah diujikan di depan penguji

Pada tanggal 13 Juli 2020

Dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1

Program Studi Teknik Informatika

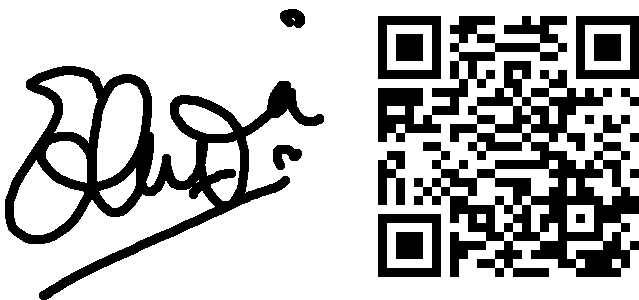
Susunan Tim Penguji :

1. Penguji 1

Tanggal: 27/07/2020

**R oyana Afwani, ST., MT.**

**NIP. 198507072014042001**

1. Penguji 2

Tanggal: 27/07/2020

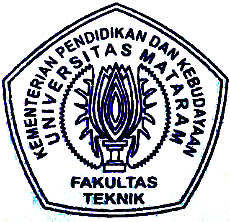
**M oh. Ali Albar, ST., M.Eng.**

**NIP. 198311252015041002**

1. Penguji 3

Tanggal: 27/07/2020

**I r. Sri Endang Anjarwani, M.Kom   
NIP. 196604032006042001**

Mataram, 27 Juli 2020  
 Dekan Fakultas Teknik   
Universitas Mataram

  
 **Akmaluddin, ST., M.Sc Eng., Ph.D.  
 NIP. 196812311994121001**

# HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan bahwa tugas akhir ini dengan judul “Sistem Informasi Geografis Untuk Pemetaan Lokasi Wisata Alam Menggunakan Metode Prototyping Berbasis Mobile” sepenuhnya adalah karya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara yang tidak sesuai dengan etika keilmuan yang berlaku. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Mataram, 27 Juli 2020

Yang membuat pernyataan,

**Agung Rahmita Illah**

# PRAKATA

*Assalamualaikum Wr. Wb.*

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-NYA sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Aplikasi Penjualan Hamster dan Kelengkapannya Berbasis Android Dengan Metode Prototype (Studi Kasus Toko Lombok Rabbit Pet House) ” tepat pada waktunya. Pada kesempatan ini pula, penulis menghaturkan terimakasih yang sebesar-besarnya kepada seluruh pihak yang telah mendukung agar terselesaikanya Tugas Akhir ini.

Penulis tentu menyadari Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan di dalamnya. Untuk itu, diharapkan kritik serta saran dari pembaca untuk Tugas Akhir ini, agar Tugas Akhir ini dapat menjadi lebih baik lagi. Penulis juga mengucapkan banyak-banyak terimakasih kepada semua pihak yang telah membantu penulis sehingga Tugas Akhir ini dapat selesai.

Demikian yang bisa disampaikan, sekali lagi terimakasih atas semua pihak yang telah membantu di dalam penyelesaian Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

*Wassalamualaikum Wr. Wb.*

Mataram, 27 Juli 2020

# UCAPAN TERIMAKASIH

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tentunya bukan hanya dari usaha penulis saja. Tugas Akhir ini bisa selesai tepat waktu tentunya berkat dukungan dari semua pihak yang terlibat juga. Oleh karena itu, pada kesempatan ini penulis menghaturkan terimakasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, Kedua Orang Tua dan keluarga yang selalu memberikan dukungan dari segala lini kehidupan selama perkuliahan.
2. Ibu Nadiyasari Agitha, S.Kom., M.M.T.selaku dosen pembimbing utama yang telah memberikan bimbingan, motivasi, semangat dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
3. Ibu Dr. Eng. Budi Irmawati, S.Kom., MT.selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
4. Gilang Ramdhani Putra selaku teman diskusi, pemotivasi pribadi dan sebagai penyemangat penulis selama pengerjaan Tugas Akhir.
5. M. Ilham Fidatama dan Ari Purnama Aji yang telah bersedia meluangkan waktunya untuk membantu penulis selama pengerjaan.
6. Bella Karina Sari, Amira Aghni H.Z, Amalia, dan Amalia Damayanti selaku supporter pribadi penulis.
7. Anggin Risna W. sebagai supporter terdekat selama penulis mengerjakan Tugas Akhir ini.
8. Toko Lombok Rabbit Pet House selaku client pada Tugas Akhir ini.
9. Responden Ketika pengumpulan kebutuhan dan penelitian yang telah meluangkan waktunya sehingga saran dan masukkannya berguna untuk pengembangan sistem.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan do’a dan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik

Semoga Tuhan Yang Maha Kuasa selalu memberikan rahmat dan hidayah-Nya dan memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis.

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc46835250)

[HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iii](#_Toc46835251)

[PRAKATA iv](#_Toc46835252)

[UCAPAN TERIMAKASIH v](#_Toc46835253)

[DAFTAR ISI vi](#_Toc46835254)

[DAFTAR GAMBAR viii](#_Toc46835255)

[DAFTAR TABEL xi](#_Toc46835256)

[ABSTRAK. xiii](#_Toc46835257)

[ABSTRACT. xiii](#_Toc46835258)

[BAB I 1](#_Toc46835259)

[1.1 Latar Belakang 1](#_Toc46835260)

[1.2 Rumusan Masalah 3](#_Toc46835261)

[1.3 Batasan Masalah 3](#_Toc46835262)

[1.4 Tujuan 4](#_Toc46835263)

[1.5 Manfaat 4](#_Toc46835264)

[1.6 Sistematika Penulisan 4](#_Toc46835265)

[BAB II 6](#_Toc46835266)

[2.1 Tinjauan Pustaka 6](#_Toc46835267)

[2.2 Dasar Teori 8](#_Toc46835268)

[2.2.1 Lombok Rabbit Pet House 8](#_Toc46835269)

[2.2.2 Hamster 8](#_Toc46835270)

[2.2.3 Sistem Informasi 15](#_Toc46835271)

[2.2.4 Penjualan 16](#_Toc46835272)

[2.2.5 Konsultasi 16](#_Toc46835273)

[2.2.6 Penyakit 17](#_Toc46835274)

[2.2.7 Prototype 17](#_Toc46835275)

[2.2.8 UML (Unidentified Modelling Language) 18](#_Toc46835276)

[2.2.9 MySQL 20](#_Toc46835277)

[2.2.10 Android 20](#_Toc46835278)

[2.2.11 Android Studio 20](#_Toc46835279)

[2.2.12 Integrated Development Environment (IDE) 20](#_Toc46835280)

[2.2.13 Android Software Development Kit (Android SDK) 21](#_Toc46835281)

[2.2.14 Java Development Kit (JDK) 21](#_Toc46835282)

[2.2.15 Android *Virtual Device* (AVD) 21](#_Toc46835283)

[2.2.16 CodeIgniter 21](#_Toc46835284)

[2.2.17 Kotlin 22](#_Toc46835285)

[2.2.18 REST (*Representational* *State* *Transfer*) 23](#_Toc46835286)

[BAB III 23](#_Toc46835287)

[3.1 Alat dan Bahan 23](#_Toc46835288)

[3.2 Alur Penelitian 24](#_Toc46835289)

[3.2.1 Studi Literatur 24](#_Toc46835290)

[3.2.2 Analisa Kebutuhan Sistem 25](#_Toc46835291)

[3.2.3 Desain 29](#_Toc46835292)

[3.2.4 Membangun *Prototype* 47](#_Toc46835293)

[3.2.5 Evaluasi *Prototype* 57](#_Toc46835294)

[3.2.6 Pembuatan Aplikasi 57](#_Toc46835295)

[3.2.7 Pengujian Aplikasi 57](#_Toc46835296)

[3.2.8 Implementasi 58](#_Toc46835297)

[BAB IV 59](#_Toc46835298)

[4.1 Evaluasi Prototype 59](#_Toc46835299)

[4.2 Pembuatan Aplikasi 60](#_Toc46835300)

[4.2.1 Arsitektur aplikasi 60](#_Toc46835301)

[4.2.2 Database 61](#_Toc46835302)

[4.2.3 Pembuatan API 66](#_Toc46835303)

[4.2.4 Pembuatan proyek aplikasi 71](#_Toc46835304)

[4.3 Pengujian Aplikasi 91](#_Toc46835305)

[4.3.1 Hasil pengujian *blackbox* 92](#_Toc46835306)

[4.3.2 Hasil pengujian MOS 95](#_Toc46835307)

[BAB V 102](#_Toc46835308)

[5.1 Kesimpulan 102](#_Toc46835309)

[5.2 Saran 103](#_Toc46835310)

[LAMPIRAN 104](#_Toc46835311)

[DAFTAR PUSTAKA 109](#_Toc46835312)

# DAFTAR GAMBAR

[Gambar 2. 1 Fishbone diagram berdasarkan tinjauan pustaka 7](#_Toc46730590)

[Gambar 2. 2 Wet tail 8](#_Toc46730591)

[Gambar 2. 3 *Scabies* 9](#_Toc46730592)

[Gambar 2. 4 *Prolapse* Ani 10](#_Toc46730593)

[Gambar 2. 5 Konstipasi 11](#_Toc46730594)

[Gambar 2. 6 *Skin Mite* 12](#_Toc46730595)

[Gambar 2. 7 *Ear Mite* 13](#_Toc46730596)

[Gambar 2. 8 Infeksi Mata 14](#_Toc46730597)

[Gambar 2. 9 Trauma Fraktur 15](#_Toc46730598)

[Gambar 3. 1 Diagram Alir Perancangan Aplikasi. 24](#_Toc46413903)

[Gambar 3. 2 *Activity* *diagram* sistem yang sedang berjalan. 25](#_Toc46413904)

[Gambar 3. 3 ERD Aplikasi Penjualan Hamster dan Kelengkapannya. 30](#_Toc46413905)

[Gambar 3. 4 *Use* *case* *Diagram* Sistem 33](#_Toc46413906)

[Gambar 3. 5 *Activity* *Diagram* “Pembeli Mendaftar” 34](#_Toc46413907)

[Gambar 3. 6 *Activity* *Diagram* “Proses *Login*” 34](#_Toc46413908)

[Gambar 3. 7 *Activity* *Diagram* “Proses Belanja dan Konfirmasi Pesanan”. 35](#_Toc46413909)

[Gambar 3. 8 *Activity* *Diagram* “Proses konsultasi” 36](#_Toc46413910)

[Gambar 3. 9 *Activity* *Diagram* “Pembeli Lupa Kata sandi”. 36](#_Toc46413911)

[Gambar 3. 10 *Activity* *Diagram* “Ganti Kata sandi” 37](#_Toc46413912)

[Gambar 3. 11 *Activity* *Diagram* “Tambah Barang” 37](#_Toc46413913)

[Gambar 3. 12 *Activity* *Diagram* “Edit Barang” 38](#_Toc46413914)

[Gambar 3. 13 *Activity* *Diagram* “Tentang Kami” 38](#_Toc46413915)

[Gambar 3. 14 *Activity* *Diagram* “Hapus Barang” 39](#_Toc46413916)

[Gambar 3. 15 *Activity* *Diagram* “*Logout*” 39](#_Toc46413917)

[Gambar 3. 16 *Class* *Diagram* 40](#_Toc46413918)

[Gambar 3. 17 *Sequence* *Diagram* “Pembeli Mendaftar” 41](#_Toc46413919)

[Gambar 3. 18 *Sequence* *Diagram* “*Login*” 41](#_Toc46413920)

[Gambar 3. 19 *Sequence* *Diagram* “Lupa Kata Sandi” 42](#_Toc46413921)

[Gambar 3. 20 *Sequence* *Diagram* “Proses Belanja” 42](#_Toc46413922)

[Gambar 3. 21 *Sequence* *Diagram* “Konsultasi” 43](#_Toc46413923)

[Gambar 3. 22 *Sequence* *Diagram* “Lihat tentang Kami” 43](#_Toc46413924)

[Gambar 3. 23 *Sequence* *Diagram* “Ganti kata sandi” 44](#_Toc46413925)

[Gambar 3. 24 *Sequence* *Diagram* “Tambah Barang” 44](#_Toc46413926)

[Gambar 3. 25 *Sequence* *Diagram* “Hapus Barang” 45](#_Toc46413927)

[Gambar 3. 26 *Sequence* *Diagram* “Edit Barang” 46](#_Toc46413928)

[Gambar 3. 27 *Sequence* *Diagram* “Proses Konfirmasi Pesanan” 46](#_Toc46413929)

[Gambar 3. 28 *Sequence* *Diagram* “Melihat laporan” 46](#_Toc46413930)

[Gambar 3. 29 *Splash* Aplikasi 47](#_Toc46413931)

[Gambar 3. 30 *Form Login* 48](#_Toc46413932)

[Gambar 3. 31 *Form* Daftar untuk Pembeli 48](#_Toc46413933)

[Gambar 3. 32 Lupa kata sandi 48](#_Toc46413934)

[Gambar 3. 33 Halaman beranda pembeli 49](#_Toc46413935)

[Gambar 3. 34 Halaman “Hamster” Pembeli 49](#_Toc46413936)

[Gambar 3. 35 Halaman Detail Barang 50](#_Toc46413937)

[Gambar 3. 36 Halaman Keranjang Belanja 50](#_Toc46413938)

[Gambar 3. 37 Halaman Biodata Penerima 51](#_Toc46413939)

[Gambar 3. 38 Halaman “Pembayaran” 51](#_Toc46413940)

[Gambar 3. 39 Halaman “Riwayat Belanja” 52](#_Toc46413941)

[Gambar 3. 40 Halaman “Konsultasi” 52](#_Toc46413942)

[Gambar 3. 41 Halaman “Diagnosa Penyakit” 53](#_Toc46413943)

[Gambar 3. 42 Halaman “Tentang Kami” Pembeli 53](#_Toc46413944)

[Gambar 3. 43 Beranda Penjual 54](#_Toc46413945)

[Gambar 3. 44 Halaman “Tambah Barang” 54](#_Toc46413946)

[Gambar 3. 45 Halaman “Tambah Barang” 55](#_Toc46413947)

[Gambar 3. 46 Halaman “Detail Barang” Penjual 55](#_Toc46413948)

[Gambar 3. 47 Halaman “Daftar Pesanan” 56](#_Toc46413949)

[Gambar 3. 48 Halaman “Detail Pesanan” Penjual 56](#_Toc46413950)

[Gambar 3. 49 Salah Satu Contoh Halaman Statistik Penjualan 57](#_Toc46413951)

[Gambar 4. 1 Arsitektur aplikasi 60](#_Toc46834608)

[Gambar 4. 2. ERD yang telah diperbarui pada aplikasi 61](#_Toc46834610)

[Gambar 4. 3 Tabel barang 62](#_Toc46834612)

[Gambar 4. 4 Tabel gejala 62](#_Toc46834613)

[Gambar 4. 5 Tabel kabupaten 62](#_Toc46834614)

[Gambar 4. 6 Tabel kategori 63](#_Toc46834615)

[Gambar 4. 7 Tabel kecamatan 63](#_Toc46834616)

[Gambar 4. 8 Tabel keranjang 63](#_Toc46834617)

[Gambar 4. 9 Tabel laporan 64](#_Toc46834618)

[Gambar 4. 10 Tabel pembeli 64](#_Toc46834619)

[Gambar 4. 11 Tabel penjual 65](#_Toc46834620)

[Gambar 4. 12 Tabel penyakit 65](#_Toc46834621)

[Gambar 4. 13 Tabel Transaksi 65](#_Toc46834622)

[Gambar 4. 14 *Controllers* yang digunakan pada CI 66](#_Toc46834623)

[Gambar 4. 15 Fungsi “index\_get()” pada *controller* Barang.php 67](#_Toc46834624)

[Gambar 4. 16 Fungsi “index\_delete()” pada *controller* Barang.php 67](#_Toc46834625)

[Gambar 4. 17 Fungsi “index\_post()” pada *controller* Barang.php 68](#_Toc46834626)

[Gambar 4. 18 Fungsi “index\_put()” pada *controller* Barang.php 68](#_Toc46834627)

[Gambar 4. 19 *Models* yang digunakan pada CI 69](#_Toc46834628)

[Gambar 4. 20 *Models* “get()” dan “delete()” pada *model* Barang\_model.php 70](#_Toc46834629)

[Gambar 4. 21 *Models* “create()” dan “update()” pada *model* Barang\_model.php 71](#_Toc46834630)

[Gambar 4. 22 *Package* activity yang digunakan pada aplikasi penjual 72](#_Toc46834631)

[Gambar 4. 23 *Package* activity pada aplikasi pembeli 73](#_Toc46834632)

[Gambar 4. 24 *Package*adapter penjual 75](#_Toc46834633)

[Gambar 4. 25 *Package*adapter pembeli 75](#_Toc46834634)

[Gambar 4. 26 Package API penjual 75](#_Toc46834635)

[Gambar 4. 27 Package API pembeli 75](#_Toc46834636)

[Gambar 4. 28 Main view model penjual 76](#_Toc46834637)

[Gambar 4. 29 Main view model pembeli 76](#_Toc46834638)

[Gambar 4. 30 Session manager 76](#_Toc46834639)

[Gambar 4. 31 *Package* Model pembeli 77](#_Toc46834640)

[Gambar 4. 32 *Package* Model penjual 77](#_Toc46834641)

[Gambar 4. 33*Package* Layout pembeli 78](#_Toc46834642)

[Gambar 4. 34 *Package* Layout penjual 78](#_Toc46834643)

[Gambar 4. 35 Tampilan*login* 79](#_Toc46834644)

[Gambar 4. 36 Tampilan *form* daftar 80](#_Toc46834645)

[Gambar 4. 37 Tampilan lupa kata sandi 80](#_Toc46834646)

[Gambar 4. 38 Tampilan beranda pembeli 81](#_Toc46834647)

[Gambar 4. 39 Tampilan konsultasi 81](#_Toc46834648)

[Gambar 4. 40 Tampilan keranjang 82](#_Toc46834649)

[Gambar 4. 41 Tampilan riwayat belanja 83](#_Toc46834650)

[Gambar 4. 42 Tampilan profil pembeli 84](#_Toc46834651)

[Gambar 4. 43 Tampilan detail barang 84](#_Toc46834652)

[Gambar 4. 44 Tampilan ubah kata sandi 85](#_Toc46834653)

[Gambar 4. 45 Tampilan tentang kami 85](#_Toc46834654)

[Gambar 4. 46 Tampilan *login* penjual 86](#_Toc46834655)

[Gambar 4. 47 Tampilan *splash* 86](#_Toc46834656)

[Gambar 4. 48 Tampilan beranda penjual 87](#_Toc46834657)

[Gambar 4. 49 Tampilan laporan pesanan 87](#_Toc46834658)

[Gambar 4. 50 Tampilan laporan bulanan 88](#_Toc46834659)

[Gambar 4. 51 Tampilan profil penjual 89](#_Toc46834660)

[Gambar 4. 52 Tampilan detail barang penjual 89](#_Toc46834661)

[Gambar 4. 53 Tampilan laporan stok menipis 90](#_Toc46834662)

[Gambar 4. 54 Tampilan tambah barang 91](#_Toc46834663)

# DAFTAR TABEL

[Tabel 2. 1 Tabel nilai keyakinan penyakit diare/*wet Tail* 9](#_Toc46697176)

[Tabel 2. 2 Tabel nilai keyakinan penyakit *scabies* 10](#_Toc46697177)

[Tabel 2. 3 Tabel nilai keyakinan penyakit *prolapse* ani 11](#_Toc46697178)

[Tabel 2. 4 Tabel nilai keyakinan penyakit konstipasi 11](#_Toc46697179)

[Tabel 2. 5 Tabel nilai keyakinan penyakit *skin mite* 12](#_Toc46697180)

[Tabel 2. 6 Tabel nilai keyakinan penyakit *ear mite* 13](#_Toc46697181)

[Tabel 2. 7 Tabel nilai keyakinan penyakit infeksi mata 14](#_Toc46697182)

[Tabel 2. 8 Tabel nilai keyakinan penyakit trauma fraktur 15](#_Toc46697183)

[Tabel 3. 1 Tabel pembeli 30](#_Toc46697184)

[Tabel 3. 2 Tabel penjual 31](#_Toc46697185)

[Tabel 3. 3 Tabel penyakit 31](#_Toc46697186)

[Tabel 3. 4 Tabel gejala 31](#_Toc46697187)

[Tabel 3. 5 Tabel barang 31](#_Toc46697188)

[Tabel 3. 6 Tabel kriteria gejala 31](#_Toc46697189)

[Tabel 3. 7 Tabel Membeli 32](#_Toc46697190)

[Tabel 3. 8 Tabel Konsultasi 32](#_Toc46697191)

[Tabel 4. 1 Pengujian *Blackbox* fungsi halaman laporan pesanan 92](#_Toc46391654)

[Tabel 4. 2 Pengujian *Blackbox* fungsi laporan bulanan 92](#_Toc46391655)

[Tabel 4. 3 Pengujian *Blackbox* fungsi menambah data baru 93](#_Toc46391656)

[Tabel 4. 4 Pengujian *Blackbox* fungsi menghapus data barang 93](#_Toc46391657)

[Tabel 4. 5 Pengujian *Blackbox* fungsi mengubah data barang 93](#_Toc46391658)

[Tabel 4. 6 Pengujian *Blackbox* fungsi menambah pesanan 94](#_Toc46391659)

[Tabel 4. 7 Pengujian *Blackbox* fungsi keranjang belanja 94](#_Toc46391660)

[Tabel 4. 8 Pengujian *Blackbox* fungsi riwayat belanja 94](#_Toc46391661)

[Tabel 4. 9 Pengujian *Blackbox* fungsi pengiriman barang 95](#_Toc46391662)

[Tabel 4. 10 Pengujian *Blackbox* fungsi konsultasi 95](#_Toc46391663)

[Tabel 4. 11 Pengujian MOS penjual (pertanyaan 1) 97](#_Toc46391664)

[Tabel 4. 12 Pengujian MOS penjual (pertanyaan 2) 97](#_Toc46391665)

[Tabel 4. 13 Pengujian MOS penjual (pertanyaan 3) 98](#_Toc46391666)

[Tabel 4. 14 Pengujian MOS penjual (pertanyaan 4) 98](#_Toc46391667)

[Tabel 4. 15 Pengujian MOS penjual (pertanyaan 5) 98](#_Toc46391668)

[Tabel 4. 16 Pengujian MOS penjual (pertanyaan 6) 99](#_Toc46391669)

[Tabel 4. 17 Pengujian MOS pembeli (pertanyaan 1) 99](#_Toc46391670)

[Tabel 4. 18 Pengujian MOS pembeli (pertanyaan 2) 99](#_Toc46391671)

[Tabel 4. 19 Pengujian MOS pembeli (pertanyaan 3) 100](#_Toc46391672)

[Tabel 4. 20 Pengujian MOS pembeli (pertanyaan 4) 100](#_Toc46391673)

[Tabel 4. 21 Pengujian MOS pembeli (pertanyaan 5) 100](#_Toc46391674)

[Tabel 4. 22 Pengujian MOS pembeli (pertanyaan 6) 101](#_Toc46391675)

# ABSTRAK

Lombok Rabbit Pet House adalah salah satu toko hewan yang menjual hamster di Mataram, Lombok. Tidak hanya melayani masyarakat di sekitar Mataram, tetapi toko ini juga merupakan pemasok hamster di beberapa pasar hewan di Mataram. Karena relatif sedikit toko hewan yang menjual hamster di Lombok, tidak jarang bagi pelanggan di sekitar kota Mataram untuk menjadi pelanggan setia toko ini. Namun sayangnya, sistem penjualan di toko ini masih menggunakan sistem manual. Termasuk catatan barang dagangan yang tersisa dan barang yang dijual. Berdasarkan wawancara dengan pemilik toko (Mr Ulul Azmi), ia mengatakan bahwa toko membutuhkan sistem penjualan yang dapat diakses oleh semua masyarakat di Lombok untuk meningkatkan layanan penjualan dan memfasilitasi penjual dan pelanggan dalam melakukan transaksi online. Setelah mendapatkan informasi yang diperlukan tentang toko, penulis mensurvei 37 responden tentang pengetahuan mereka tentang penyakit hamster. Ditemukan bahwa 70,3% responden mengaku memiliki sedikit pengetahuan tentang jenis-jenis penyakit hamster. Pada saat yang sama, 5,4% lainnya mengaku tidak memiliki pengetahuan tentang itu. Selanjutnya, 91,9% responden memilih untuk setuju jika ada sistem informasi yang dapat membantu pemilik hamster dalam mendapatkan informasi tentang berbagai jenis penyakit hamster. Untuk alasan itu, penulis membuat Aplikasi Penjualan Hamster dan Asesorisnya Berbasis Android Menggunakan Metode Prototipe. Dalam desainnya, ia menggunakan bahasa pemrograman Kotlin, juga kerangka kerja CodeIgniter sebagai akses API yang memanggil query pada database.

Kata kunci: Sistem Informasi, Metode Prototyping, Android Apps, Toko Hewan.

# ABSTRACT

Lombok Rabbit Pet House is one of the animal shops that sell hamsters in Mataram, Lombok. Not only serving the community around Mataram, but this shop is also a supplier of hamsters in several animal markets in Mataram. Because there are relatively few animal shops that sell hamsters in Lombok, it is not uncommon for customers around the city of Mataram to become loyal customers of this shop. But unfortunately, the sales system in this store still uses a manual system. Includes records of remaining merchandise and items sold. Based on an interview with the shop owner (Mr. Ulul Azmi), he said that the shop needed a sales system that was accessible to all communities in Lombok to improve sales services and facilitate sellers and customers in conducting online transactions. After obtaining the necessary information about the shop, the authors surveyed 37 respondents about their knowledge of hamster disease. It was found that 70.3% of respondents claimed to have little knowledge about the types of hamster disease. At the same time, another 5.4% claimed to have no knowledge about it. Furthermore, 91.9% of respondents chose to agree if there is an information system that can help the owner of the hamster in getting information about various types of hamster disease. For that reason, the author makes an Android-Based Hamster Sales Application and Its Accessories Using the Prototype Method. In its design, it uses the Kotlin programming language, as well as the CodeIgniter framework as an API access that calls queries on the database.

Keywords: Information Systems, Prototyping Methods, Android Apps, Pet Shop.

# BAB I

**PENDAHULUAN**

## 1.1 Latar Belakang

Hamster merupakan salah satu hewan pengerat yang masih termasuk keluarga *Cricetidae*. Pada beberapa jenisnya, binatang hamster dapat bertahan sebagai hewan peliharaan di antaranya: *Syrian* Hamster, *Dwarf* *Campbells* *Russian* Hamster, *Dwarft* *Winter* *White* *Russian* Hamster, *Shinese* Hamster dan *Roborovski* Hamster. Layaknya binatang peliharaan lainnya, hamster juga membutuhkan perawatan dan peralatan khusus untuk membuatnya tetap nyaman hidup bersama pemiliknya. Beberapa peralatan yang dapat digunakan dalam merawat hamster diantaranya: kandang (kandang plastik, kandang kawat), peralatan kandang (botol air, rumah, blok kayu, sepatu boot plastik), serta makanan hamster (sejenis gandum, pellet, kacang tanah) [1].

Sebagai hewan peliharaan yang cukup popular dan membutuhkan perhatian yang cukup intens terhadap pakan dan habitatnya, beberapa pasar hewan dan toko hewan juga memasukkan hamster dan kelengkapannya ke dalam daftar dagangan mereka. Berdasarkan data dari Google Maps, terdapat lebih dari 35 toko hewan yang terdaftar di Lombok. Sementara itu, hanya empat toko yang mencantumkan hamster pada daftar dagangannya. Tiga di antaranya berlokasi di sekitar Kota Mataram dan satu di Lombok Timur. Salah satu toko hewan di Mataram yang tercantum pada data Google Maps adalah toko Lombok Rabbit Pet House. Toko ini menjual beberapa jenis hewan peliharaan seperti: kelinci, hamster, dan kucing. Beberapa jenis kebutuhan hewan peliharaan juga dapat dijumpai di toko ini. Berdasarkan hasil wawancara yang telah dilakukan, pemilik toko (sdr. Ulul Azmi) mengatakan bahwa dari ketiga jenis hewan yang dijual, hamster adalah hewan yang cukup banyak peminatnya, mulai dari masyarakat biasa hingga anggota dari komunitas reptil. Jangkauannya pembelinya pun tidak hanya masyarakat di sekitar Kota Mataram saja, namun juga hingga Lombok Timur. Bahkan penjual-penjual hamster di sekitar Mataram juga menjadikan toko ini sebagai penyuplai stok hamster mereka, salah satunya adalah Pasar Hewan Sindu Mataram. Dalam sebulan toko ini mampu menjual delapan ekor hamster, ditambah dengan pembelian pakan dan kelengkapan hamster. Namun, penjualan dan transaksi toko ini masih menggunakan sistem manual yang dibantu dengan aplikasi android sederhana sebagai pencatatan laporan hariannya. Dengan jangkauan pembeli yang luas serta penjualan dan pembuatan laporan penjualan hamster yang masih dilakukan secara semi manual, pemilik menginginkan sebuah sistem penjualan yang dapat mempermudah proses transaksi dan pencatatan laporan penjualan yang mudah diakses baik oleh penjual maupun pembeli tanpa perlu bertatap muka secara langsung.

Hamster tidak diragukan tetap pada posisinya sebagai hewan peliharaan yang popular bagi anak-anak dan ia telah diterima sebagai seekor binatang yang kompetitif pada pertunjukan. Walaupun hamster secara relatif mudah untuk dipelihara, tetapi apabila mereka terserang penyakit, kondisi mereka dapat memburuk dengan cepat dan perhatian serta perawatan adalah jalan terbaik untuk mencegahnya [1]. Berdasarkan pernyataan seorang pakar (ibu drh. Lucky Retno Putri), penyakit-penyakit umum yang sering terjadi pada hamster adalah diare, *scabies*, *prolapse* ani, konstipasi, kutu kulit (*skin* *mite*), kutu kuping (*ear* *mite*), infeksi mata, dan trauma fraktur. Sementara itu, pemilik yang memiliki pengetahuan yang minim tentang gejala penyakit hamster, akan mengalami kesulitan dalam melakukan pertolongan pertama pada hamster yang sedang sakit [2]. Setelah dilakukan survei terhadap 37 responden (36 orang memelihara hamster dan satu orang tidak memelihara hamster) mengenai pengetahuannya tentang penyakit hamster, didapatkan rekapitulasi persentase nilai sebesar 70,3% responden mengaku memiliki sedikit pengetahuan mengenai macam-macam penyakit hamster. Sedangkan 5,4% lainnya mengaku tidak memiliki pengetahuan apapum mengenai macam-macam penyakit hamster. Selanjutnya, hasil survei menyatakan sebanyak 91,9% responden memilih setuju apabila terdapat sebuah sistem informasi yang dapat membantu para majikan hamster dalam mendapatkan informasi mengenai macam-macam penyakit hamster.

Di era modern seperti sekarang, internet bukanlah hal asing lagi bagi manusia. Perkembangan aplikasi *mobile* berbasis *smartphone* setiap saat mengalami peningkatan. Kondisi tersebut dapat dilihat dari jumlah pengguna yang mengakses *internet* melalui *smartphone* selalu meningkat dari tahun ke tahun [3]. Penggunaan *internet* untuk bisnis secara *online* telah tumbuh dengan pesatnya. Kemudahan dalam mengoperasionalkan teknologi untuk melakukan pembelanjaan *online* menjadikan konsumen puas [4]. Maka, dari permasalahan-permasalahan di atas, dibutuhkan suatu metode penjualan hamster dan kelengkapannya menggunakan aplikasi *mobile*. Selain itu, disediakan fasilitas berupa sebuah konsultasi penyakit hamster dengan melihat gejala utama yang nampak pada hamsternya.

Untuk itu, pada penelitian ini dibuat suatu aplikasi penjualan yang ditunjang dengan suatu fitur tambahan berupa konsultasi sederhana penyakit hamster yang dapat mempermudah penjual dan calon pembeli toko hewan Lombok Rabbit Pet House untuk dapat membeli hamster tanpa perlu datang ke tokosecara langsung.

Pada pengembangannya, sistem ini akan menggunakan metode *prototype*. Dipilihnya metode *prototype* karena adanya *requirement* *identification* yang akurat dan dilakukan evaluasi secara berkala, *user* *experience* yang meningkat, serta kesalahan dan redudansi yang dapat diminimalisir karena proses identifikasi yang baik pada *interface* [5]. Dengan adanya aplikasi penjualan hamster dan kelengkapannya ini, diharapkan dapat membantu proses jual beli dan pembuatan laporan penjualan pada toko Lombok Rabbit Pet House dan mempermudah para pembelinya untuk melakukan pembelian dengan mudah serta dapat membantu pemilik hamster untuk mengetahui kemungkinan penyakit yang terjadi pada peliharaannya berdasarkan gejala dominan yang terlihat.

## Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan, rumusan masalah yang dapat diambil adalah:

1. Bagaimana perancangan Sistem Informasi Penjualan Hamster dan Kelengkapannya untuk memudahkan toko Lombok Rabbit Pet House dan calon pembelinya melakukan transaksi tanpa perlu bertemu secara langsung?
2. Bagaimana membangun mekanisme sistem untuk mengetahui penyakit hamster melalui gejala dominan yang dipilih?

## Batasan Masalah

Penelitian ini memiliki batasan-batasan masalah untuk memberikan lingkup penelitian agar lebih terfokus ketika pengerjaan. Adapun batasan masalah yang diberikan adalah sebagai berikut:

1. Destinasi pengiriman diisi secara manual oleh pembeli.
2. Dalam proses konsultasi, pembeli memilih salah satu dari gejala-gejala yang diberikan.
3. Pembayaran dilakukan via ATM dengan konfirmasi secara manual dengan melakukan *upload* foto bukti pembayaran via *email*.
4. Pengiriman hamster hanya dapat dilakukan di Pulau Lombok.

## Tujuan

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Untuk dapat merancang Sistem Informasi Penjualan Hamster dan Kelengkapannya dengan *interface* yang mudah diakses oleh penjual di toko Lombok Rabbit Pet House dan calon pembelinya dengan menampilkan beberapa jenis hamster dan kebutuhan hamster yang dijual.
2. Untuk membangun Sistem Informasi Penjualan Hamster dan Kelengkapannya yang dapat memberikan informasi yang dibutuhkan pengguna dalam mengetahui suatu penyakit yang mungkin diderita hamster dilihat dari gejala dominan yang dimasukkan.

## Manfaat

Manfaat dari penelitian bagi pengguna Sistem Informasi Penjualan dan Konsultasi Penyakit Hamster antara lain:

1. Calon pembeli bisa mendapatkan informasi hamster dan kebutuhan hamster yang tersedia.
2. Pengguna mengetahui penyakit yang mungkin diderita hamsternya melalui gejala-gejala yang terjadi.
3. Untuk memudahkan pembeli dalam transaksi pembelian hamster pada toko Lombok Rabbit Pet House.
4. Pihak penjual pada toko Lombok Rabbit Pet House dapat mengetahui laporan mengenai penjualannya.

## Sistematika Penulisan

Sistematika penulisan dari penelitian ini disajikan dalam beberapa bab Antara lain sebagai berikut.

1. Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

1. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian-penelitian terdahulu yang mengimplementasikan metode *prototype* serta teori-teori sebagai referensi penulis ketika melakukan penelitian.

1. Bab III Metode Penelitian

Bab ini membahas tentang metodelogi yang digunakan untuk membangun Sistem Informasi Penjualan dan Konsultasi Penyakit Hamster.

1. Bab IV Pembahasan

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisis masalah, analisis metode, analisis kebutuhan sistem, serta perancangan sistem yang terdiri dari perancangan diagram alir (*flowchart*).

1. Bab V Penutup

Bab ini berisi kesimpulan yang terkait dengan permasalahan dan saran.

# BAB II

**TINJAUAN PUSTAKA DAN DASAR TEORI**

## Tinjauan Pustaka

Dalam pembuatannya, Aplikasi Penjualan Hamster dan Kelengkapannya ini merujuk pada beberapa penelitian yang sebelumnya pernah dilakukan. Namun, penelitian-penelitian tersebut umumnya hanya membahas tentang sistem informasi penjualan saja atau sistem pakar terhadap suatu penyakit tertentu.

Sistem informasi berjudul “Analisis dan Perancangan Sistem Informasi Penjualan Berdasarkan Stok Gudang Berbasis *Client* *Server* (Studi Kasus Toko Grosir “Restu Anda”)” menggunakan teknologi RMI (*Remote* *Method* *Invocation*) yang mempermudah pengguna mengelola data dari dua atau lebih komputer yang berbeda. Dalam pengembangannya, sistem ini menggunakan metode *Prototype*. Hasil yang diperoleh dari sistem ini adalah pemilik toko dapat memprediksi berapa banyak kebutuhan barang yang harus disediakan untuk dijual kepada *downline* dengan melihat data penjualan yang sudah tersinkronisasi dengan data *member* [6].

Sistem penjualan berjudul “Aplikasi *M-Commerce* Bebasis Android Pada Kinza Collection” menggunakan *Waterfall* sebagai metode pengembangannya. Dibuatnya sistem ini agar konsumen tidak perlu lagi datang ke toko langsung untuk belanja, dan dapat melakukan transaksi secara *online* sehingga dapat lebih mempersingkat waktu pencarian dan tidak terbatas ruang dan waktu. Sistem ini menghasilkan sebuah aplikasi M-*commerce* dengan memanfaatkan sistem aplikasi android pada *smartphone*. Serta sistem ini akan dikembangkan dengan menggunakan pengembangan dan pemrograman perangkat lunak yang berorientasi objek (OOP) [7].

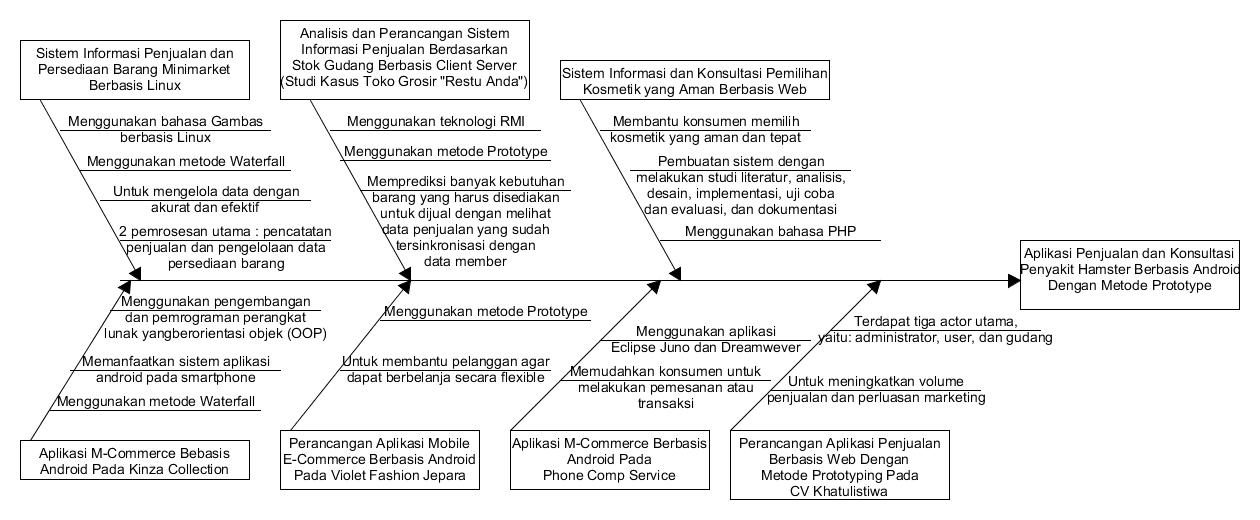
Sistem penjualan berjudul “Perancangan Aplikasi *Mobile E-Commerce* Berbasis Android Pada Violet Fashion Jepara” dibuat dengan tujuan untuk membantu pelanggan agar dapat berbelanja dimana pun dan kapan pun yang diinginkan secara *flexible*. Sistem ini menggunakan metode *Prototype* dalam pengembangannya [8].

Sistem penjualan berjudul “Aplikasi *M-Commerce* Berbasis Android Pada Phone Comp Service” dibuat untuk memudahkan konsumen untuk melakukan pemesanan atau transaksi pada toko Phone Comp Service, sehingga akan lebih mudah efisiensi waktu. Pada pengembangannya, sistem ini menggunakan aplikasi Eclipse Juno dan Dreamwever [9].

Sistem penjualan berjudul “Perancangan Aplikasi Penjualan Berbasis Web Dengan Metode *Prototyping* Pada CV Khatulistiwa” dibuat dengan tujuan untuk meningkatkan volume penjualan dan perluasan *marketing* produk CV Khatulistiwa. Pada aplikasi ini, terdapat tiga aktor utama, yaitu: *administrator*, *user*, dan gudang [10].

Sistem berjudul “Sistem Informasi dan Konsultasi Pemilihan Kosmetik yang Aman Berbasis Web” serupakan sistem yang bertujuan untuk menyediakan informasi, membimbing, memberi prediksi serta mengarahkan pengguna agar dapat melakukan pengambilan keputusan dengan lebih baik dalam memilih jenis kosmetik. Hasil akhir dari sistem ini adalah sebuah aplikasi yang dapat membantu konsumen dalam memilih kosmetik yang aman dan tepat untuk digunakan. Langkah pembuatan sistem ini yaitu dengan melakukan studi literature, analisis, desain, implementasi, uji coba dan evaluasi, dan dokumentasi. Dalam pembuatannya, sistem ini menggunakan bahasa PHP [11].

Berdasarkan jurnal-jurnal yang telah disebutkan, penulis menyimpulkan bahwa aplikasi yang akan dibuat nantinya akan berbentuk sebuah sistem penjualan berbasis android dengan penambahan fitur berupa konsultasi sederhana mengenai 8 penyakit terumum yang dapat menyerang hamster. Penggabungan kedua fungsi tersebut dalam suatu aplikasi bermaksud untuk menambah keefektifan pengguna dalam melakukan pembelian sekaligus konsultasi penyakit hamster.



Gambar 2. Fishbone diagram berdasarkan tinjauan pustaka

## Dasar Teori

Berikut adalah teori-teori dasar atau umum yang digunakan dalam penelitian ini:

### Lombok Rabbit Pet House

Lombok Rabbit Pet House adalah sebuah toko hewan yang beralamat di Jl. Majapahit No.18, Pagesangan, Kec. Mataram, Kota Mataram, Nusa Tenggara Barat. Toko ini didirikan oleh pemiliknya yang bernama saudara Ulul Azmi pada 18 Maret 2017 setelah sebelumnya sempat membuka jasa jual hewan secara *online* lewat media sosial. Fokus dari penjualan toko ini adalah pada penjualan hewan peliharaan, seperti : kucing, kelinci, dan hamster. Selain itu, toko ini juga menyediakan kebutuhan-kebutuhan hewan, seperti : pakan, kandang, vitamin hewan, dll.

### Hamster

Hamster adalah binatang menyusui yang termasuk keluarga binatang pengerat, yang mempunyai gigi seri besar dan secara terus-menerus tumbuh. Hamster senantiasa menggerogoti sesuatu untuk mencegah gigi mereka terus bertambah besar. Kata ‘*rodentia’* diperoleh dari bahasa Latin ‘*rodere’* yang berarti ‘meng-gerogoti’.

Tupai hamster membentuk keluarga *Cricetidae* yang dipecah ke dalam beberapa jenis berbeda (termasuk *Cricetulus*, *Phodopus*, dan *Calomyscus*). Ada banyak jenis tupai hamster yang berbeda di seluruh dunia dan paling banyak tinggal di wilayah *semi-desert*, di mana mereka tinggal di liang-liang. Liang-liang ini terdiri dari banyak terowongan yang memisahkan kamar untuk menyimpan makanan dan kamar tidur. Mereka mempunyai penglihatan yang sangat lemah. Indera yang paling berfungsi adalah insting pendengaran dan insting penciuman yang tajam [1]. Terdapat beberapa penyakit yang umumnya dapat menyerang hamster, seperti berikut [2]:

#### Diare / *Wet* *Tail*



Gambar 2. 2 Wet tail

Diare / *Wet* *Tail* merupakan penyakit yang disebabkan oleh infeksi bakteri yang memicu diare akut. Selain itu, stress karena perjalanan jauh atau faktor lingkungan yang tidak sehat juga dapat menjadi penyebab terserangnya hamster oleh penyakit ini. Ketidakseimbangan porsi makan juga dapat mengakibatkan diare [12]. Berikut merupakan gejala yang ditimbulkan penyakit diare/*wet* *tail* pada hamster [2]:

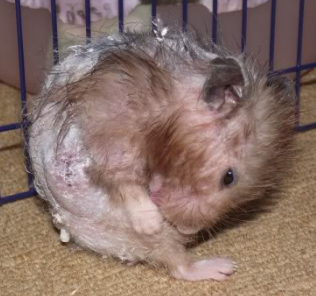
* 1. Area anus hamster menjadi basah dan lengket.
  2. Hamster menjadi tidak aktif.
  3. Hamster tidak nafsu makan.
  4. Kotoran hamster cair dan dalam jumlah banyak.
  5. Hamster tampak kesakitan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. Tabel nilai keyakinan penyakit diare/*wet Tail*

|  |  |  |
| --- | --- | --- |
| No. | Gejala | Nilai |
| 1. | Kotoran hamster cair dan dalam jumlah banyak. | 0.7 |
| 2. | Hamster menjadi tidak aktif. | 0.1 |
| 3. | Hamster tidak nafsu makan. | 0.1 |
| 4. | Area anus hamster menjadi basah dan lengket. | 0.5 |
| 5. | Hamster tampak kesakitan. | 0.1 |

#### *Scabies*



Gambar 2. *Scabies*

Saat ini istilah *scabies* berarti lesi kulit yang muncul oleh aktivitas tungau. *Sarcoptes scabiei* tinggal di dalam stratum korneum (lapisan tanduk) kulit dan memakan cairan sel. Tungau menggali hanya dilapisan bagian atas kulit dan tidak pernahsampai di bawah stratum korneum. Terowongan yang dihasilkan tampak sebagaigaris tipis yang berkelok-kelok yang berwarna abu-abu atau seperti kulit dengan panjang dapat mencapai lebih dari 1 cm [13]. Berikut merupakan gejala yang ditimbulkan penyakit *scabies* pada hamster [2]:

* + - 1. Munculnya keropeng di kulit hamster.
      2. Efek mengelupas pada kulit hamster.
      3. Bulu hamster mengalami kerontokan.
      4. Aksi menggaruk yang berlebihan.
      5. Hamster menjadi tidak aktif.
      6. Hamster tidak nafsu makan.
      7. Hamster tampak kesakitan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. Tabel nilai keyakinan penyakit *scabies*

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Rambut rontok pada hamster. | 0.5 |
| 5. | Aksi menggaruk yang berlebihan pada hamster. | 0.5 |
| 6. | Munculnya keropeng di kulit hamster. | 0.7 |
| 7. | Efek mengelupas pada kulit hamster. | 0.5 |

#### Prolapse Ani



Gambar 2. 4 *Prolapse* Ani

*Prolapse* Ani merupakan penyakit yang disebabkan oleh trauma mekanis atau kogenital [12]. Berikut merupakan gejala yang ditimbulkan penyakit *prolapse* ani pada hamster [2]:

Turunnya posisi organ pencernaan hamster.

Anus hamster menjuntai keluar.

* + - 1. Hamster menjadi tidak aktif.
      2. Hamster tidak nafsu makan.
      3. Hamster tampak kesakitan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. Tabel nilai keyakinan penyakit *prolapse* ani

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Turunnya posisi organ pencernaan hamster. | 0.5 |
| 5. | Anus hamster menjuntai keluar. | 0.5 |

#### Konstipasi



Gambar 2. Konstipasi

Konstipasi atau sembelit pada hamster dapat disebabkan oleh kebiasaan hamster mengunyah *bedding* (alas kandang) yang kotor. Selain itu, kurangnya cairan pada tubuh hamster akan menyebabkan dehidrasi sehingga memicu penyakit ini terjadi [12]. Berikut merupakan gejala yang ditimbulkan oleh penyakit konstipasi pada hamster [2]:

* + - 1. Hamster menjadi tidak aktif.
      2. Hamster tampak kesakitan.
      3. Hamster tidak nafsu makan.
      4. Tidak terdapat kotoran hamster di sekamnya.
      5. Jika diraba bagian perut hamster terasa keras.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. Tabel nilai keyakinan penyakit konstipasi

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Tidak terdapat kotoran hamster di sekamnya. | 0.1 |
| 5. | Jika diraba bagian perut hamster terasa keras. | 0.7 |

#### *Skin* *Mite* (Kutu Kulit)



Gambar 2. 6 *Skin Mite*

*Skin* *mite* pada hamster dapat disebabkan oleh tungau atau kutu mikroskopis yang bersarang di kulit hamster sehingga seakan tidak terlihat sama sekali. Jika tungau ini menyerang salah satu hamster, dengan cepat akan menular pada hamster lain dalam kandang yang sama. Tungau kulit ini juga dapat tertular dari binatang lainnya seperti anjing dan kucing [12]. Berikut merupakan gejala yang ditimbulkan oleh penyakit *skin* *mite* pada hamster [2]:

Rambut rontok pada hamster.

Aksi menggaruk yang berlebihan pada hamster.

* + - 1. Hamster menjadi tidak aktif.
      2. Hamster tidak nafsu makan.
      3. Hamster tampak kesakitan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. 5 Tabel nilai keyakinan penyakit *skin mite*

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Rambut rontok pada hamster. | 0.5 |
| 5. | Aksi menggaruk yang berlebihan pada hamster. | 0.5 |

#### *Ear* *Mite* (Kutu Kuping)



Gambar 2. 7 *Ear Mite*

*Ear* *mite* merupakan penyakit yang hampir sama dengan *skin* *mite*, hanya saja spesifik terserang pada telinga dan sekitarnya. Jenis spesies yang menyebabkan penyakit ini adalah *Notoedres* [12]. Berikut merupakan gejala yang ditimbulkan oleh penyakit *skin* *mite* pada hamster [2]:

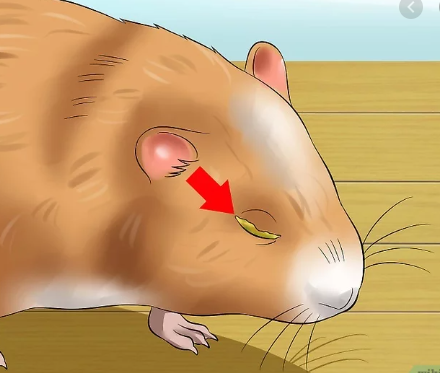
* 1. Hamster menggaruk secara membabi buta di area telinga.
  2. Kulit di sekitar kuping hamster menjadi rusak.
  3. Kulit di sekitar kuping hamster menjadi tampak kemerahan.
  4. Terdapat banyak kotoran pada telinga hamster.
     + 1. Hamster menjadi tidak aktif.
       2. Hamster tidak nafsu makan.
  5. Hamster tampak kesakitan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. 6 Tabel nilai keyakinan penyakit *ear mite*

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Hamster menggaruk secara membabi buta di area telinga. | 0.6 |
| 5. | Kulit di sekitar kuping hamster menjadi rusak. | 0.5 |
| 6. | Kulit di sekitar kuping hamster menjadi tampak kemerahan. | 0.5 |
| 7. | Terdapat banyak kotoran pada telinga hamster. | 0.1 |

#### Infeksi Mata



Gambar 2. Infeksi Mata

Infeksi pada mata dapat disebabkan oleh virus, bakteri, jamur maupun parasit. Infeksi tersebut dapat mengenai seluruh bagian mata, mulai dari kelopak mata hingga lensa [14]. Berikut merupakan gejala yang ditimbulkan oleh penyakit infeksi mata pada hamster [2]:

Hamster menjadi tidak aktif.

Hamster tidak nafsu makan.

Hamster tampak kesakitan.

Mata hamster mengeluarkan air mata terus-menerus.

Kelopak mata hamster memerah.

Mata hamster memicing.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. Tabel nilai keyakinan penyakit infeksi mata

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Mata hamster mengeluarkan air mata terus-menerus. | 0.7 |
| 5. | Kelopak mata hamster memerah. | 0.7 |
| 6. | Mata hamster memicing. | 0.7 |

#### Trauma Fraktur



Gambar 2. 9 Trauma Fraktur

Fraktur adalah gangguan dari kontinuitas yang normal dari suatu tulang. Jika terjadi fraktur, maka jaringan lunak di sekitarnya juga sering kali terganggu. Radiografi (sinar-x) dapat menunjukkan keberadaan cedera tulang, tetapi tidak mampu menunjukkan otot atau ligamen yang robek, saraf yang putus, atau pembuluh darah yang pecah sehingga dapat menjadi komplikasi pemulihan klien [15]. Berikut merupakan gejala yang ditimbulkan oleh penyakit trauma fraktur pada hamster [2]:

Hamster menjadi tidak aktif.

Hamster tidak nafsu makan.

Hamster tampak kesakitan.

Hamster kesulitan berjalan.

Berikut merupakan nilai keyakinan pakar terhadap suatu gejala terjadi pada sebuah penyakit hamster:

Tabel 2. 8 Tabel nilai keyakinan penyakit trauma fraktur

|  |  |  |
| --- | --- | --- |
| No | Gejala | CF *Rule* |
| 1. | Hamster menjadi tidak aktif. | 0.1 |
| 2. | Hamster tidak nafsu makan. | 0.1 |
| 3. | Hamster tampak kesakitan. | 0.1 |
| 4. | Hamster kesulitan berjalan. | 0.7 |

### Sistem Informasi

Sistem informasi adalah kumpulan elemen yang saling berhubungan dan berfungsi untuk memproses, mengumpulkan, mendistribusikan serta menyimpan informasi yang berguna sebagai pendukung dalam pembuatan keputusan juga pengawasan dalamorganisasi. Maka dengan adanya sistem informasi, pemilik toko akan lebih mudah dalam memberikan informasi sehingga penjualan yang dilakukan tidak terganggu [6].

### Penjualan

Yang dimaksud dengan penjualan ialah menjual barang dagang yang menjadi usaha pokok suatu perusahaan dengan dilakukan secara terus menerus dan tepat. penjualan juga adalah salah satu langkah pemasaran dari suatu perusahaan, sehingga perusahaan dapat memperoleh keuntungan yang akibatnya adalah kegiatan operasional perusahaan dapat tetap dijalankan. Berikut pengertian penjualan menurut beberapa ahli [6]:

* 1. Menjual adalah ilmu dan seni mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang dan jasa yang ditawarkan.
  2. Penjualan adalah berkumpulnya seorang pembeli dan penjual dengan tujuan melaksanakan tukar menukar barang dan jasa berdasarkan pertimbangan yang berharga misalnya pertimbangan uang.
  3. Penjualan adalah bagian dari promosi dan promosi adalah salah satu bagian dari keseluruhan sistem pemasaran.

Dalam pengertiannya suatu perusahaan mempunyai tiga tujuan dalam penjualan yaitu [6]:

1. Mencapai batas tertentu dari penjualan.
2. Mendapat keuntungan.
3. Mendukung perkembangan perusahaan.

### Konsultasi

Layanan konsultasi merupakan proses dalam suasana kerja sama dan hubungan antar pribadi dengan tujuan memecahkan suatu masalah dalam lingkup professional dari orang yang meminta konsultasi. Ada tiga unsur di dalam konsultasi, yaitu klien, orang yang minta konsultasi, dan konsultan. Konsultasi *(consultation*) yaitu segala usaha memberikan asistensi kepada seluruh anggota staf pendidikan di sekolah dan kepada orang tua siswa, demi perkembangan siswa yang lebih baik. Konsultasi diartikan sebagai pertukaran pikiran untuk mendapatkan kesimpulan (nasehat, saran) yang sebaik-baiknya.

Konsultasi merupakan kegiatan berbagi pemahaman dan kepedulian antara konselor atau guru bimbingan dan koseling dengan guru mata pelajaran, orang tua, pimpinan satuan pendidikan, atau pihak lain yang relevan dalam upaya membangun kesamaan persepsi dan memperoleh dukungan yang diharapkan dalam memperlancar pelaksanaan program layanan bimbingan dan konseling [16].

### Penyakit

Sakit merupakan suatu keadaan dimana tubuh tidak berada pada kondisi normal yang disebabkan oleh beberapa faktor dari dalam maupun luar tubuh. Berdasarkan karakteristiknya penyakit dapat digolongkan menjadi 2 yaitu penyakit menular dan penyakit tidak menular. Penyakit menular mendapatkan perhatian yang lebih dari pemerintah dibanding dengan penyakit tidak menular. Penyakit menular adalah penyakit yang disebabkan oleh bakteri, virus, atau parasit yang dapat ditularkan melalui media tertentu. Penyakit menular sering juga disebut penyakit infeksi karena penyakit ini diderita melalui infeksi virus, bakteri, atau parasit yang ditularkan melalui berbagai macam media seperti udara, jarum suntik, transfusi darah, tempat makan atau minum, dan lain sebagainya. Penyakit menular merupakan hasil perpaduan berbagai faktor yang saling mempengaruhi. Penyakit tidak menular adalah penyakit yang tidak disebabkan oleh kuman melainkan dikarenakan adanya masalah fisiologis atau metabolisme pada jaringan tubuh manusia [17].

### Prototype

Pendekatan *prototyping* *model* digunakan jika pemakai hanya mendefenisikan objektif umum dari perangkat lunaktanpa memerinci kebutuhan *input*, pemrosesan dan *output*-nya, sementara pengembang tidak begitu yakin akan efesiensi algoritma, adaptasi sistem operasi, atau bentuk antarmuka manusia-mesin yang harus diambil. Cakupan aktivitas dari *prototyping* *model* terdiri dari [5]**:**

1. Mendefinisikan objektif secara keseluruhan dan mengidentifikasi kebutuhan yang sudah diketahui.
2. Melakukan perancangan secara cepat sebagai dasar untuk membuat *prototype*.
3. Menguji coba dan mengevaluasi *prototype* dan kemudian melakukan penambahan dan perbaikan-perbaikan terhadap *prototype* yang sudah dibuat.

### UML (Unidentified Modelling Language)

Berikut adalah teori-teori dasar mengenai UML (*Unidentified Modelling Language*) yang digunakan dalam penelitian ini:

#### Use Case Diagram

*Use Case* diagram merupakan pemodelan untuk mendeskripsikan perilaku dari sistem informasi yang akan dibuat. Dapat juga dipahami sebagai cara untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi yang ada.

Penamaan *Use Case* didefiniskan sesederhana mungkin dan mudah untuk difahami. Ada dua hal yang utama dalam *Use Case*, yaitu aktor dan *Use Case.* Berikut penjelasan mengenai dua utama yang digunakan dalam *Use Case* [19]:

* 1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Meskipun simbol dari aktor berbentuk orang, tetapi aktor belum tentu merupakan orang.
  2. *Use Case* merupakan fungsionalitas dari sebuah sistem. Dengan demikian, antara konsumen dan juga pengendara pada sistem tersebut, akan mengerti atau paham mengenai fungsi sistem yang tengah dibangun.

#### ERD (Entity Relationship Diagram)

ERD (*Entity Relationship Diagram*) merupakan suatu modeluntuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Umumnya setelah perancangan ERD selesai berikutnya adalah mendesain *database* secara fisik yaitu pembuatan tabel, *index* dengan tetap mempertimbangkan *performance*. Kemudian setelah *database* selesai dilanjutkan dengan merancang aplikasi yang melibatkan *database* [19].

#### Class Diagram

*Class* Diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name.* Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Sususan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut [19]:

1. Kelas *Main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

1. Kelas *Interface*

Kelas yang mendefinisikan dan mengatur tampilan ke pengguna. Biasanya juga disebut kelas *boundaries*.

1. Kelas yang diambil dari pendefinisian *Use Case*

Merupakan kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian *Use Case.*

1. Kelas Entitas

Merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

#### Sequence Diagram

*Sequence* Diagram adalah salah satu dari diagram - diagram yang ada pada UML, *Sequence* diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object.* Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Dalam UML, *object* pada *Sequence* diagram digambarkan dengan segi empat yang berisi nama dari *object* yang digaris-bawahi. Pada *object* terdapat 3 cara untuk menamainya yaitu: nama *object*, nama *object* dan *class*, dan nama *class* [19]*.*

#### Activity Diagram

*Activity* Diagram menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. Struktur diagram ini juga mirip dengan flowchart. *Activity* Diagram berfungsi sebagai [19]:

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
2. Memperlihatkan urutan aktifitas proses pada sistem
3. *Activity* Diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *Use Case* *Diagram*.

### MySQL

MySQL adalah sebuah basis data yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel sumber data menjadi berbagai macam informasi yang dibutuhkan oleh pemakai.

Secara sederhana dapat dikatakan bahwa data diolah menjadi suatu informasi. Dan pada tahapan selanjutnya, sebuah informasi akan menjadi data untuk terciptanya informasi yang lain [19].

### Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2005. Dalam usaha mengembangkan Android, pada tahun 2007 dibentuklah *Open* *Handset* *Alliance* (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek Android, termasuk PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc [20].

### Android Studio

Android Studio merupakan sebuah IDE (*Integrated* *Development* *Environment*) untuk pengembangan aplikasi android, aplikasi ini dipublikasikan oleh Google pada tanggal 16 mei 2013 dan tersedia secara gratis dibawah lisensi Apache 2.0, Android Studio ini menggantikan *software* pengembangan android sebelumnya yaitu Eclipse [20].

### Integrated Development Environment (IDE)

*Integrated* *Development* *Environment* adalah aplikasi pengembang perangkat lunak dengan fungsi-fungsi terintegrasi yang dibutuhkan untuk mebangun sebuah perangkat lunak seperti *code* *editor*, *debugger*, *compiler*, dan sebagainya [20]. IDE merupakan salah satu komponen terpenting dalam pembuatan suatu aplikasi.

### Android Software Development Kit (Android SDK)

Android SDK dalah tools API (*Application* *Programming* *Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Pada Android SDK ini terdiri dari *debugger*, *libraries*, *handset* *emulator*, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi untuk *platform* Android SDK, Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan, *emulator* dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis dengan bahasa pemrograman Java dan berjalan di Dalvik, mesin virtual yang dirancang khusus untuk penggunaan *embedded* yang berjalan diatas kernel Linux [20].

### Java Development Kit (JDK)

JDK (Java *Development* *Kit*) adalah Paket fungsi API untuk bahasa pemrograman Java, meliputi Java *Runtime* *Environment* (JRE) dan Java *Virtual* *Machine* (JVM) [20]. JRE berfungsi untuk menjalankan program java seperti Android Studio.

### Android *Virtual Device* (AVD)

Android *Virtual* *Device* merupakan *emulator* untuk menjalankan program aplikasi Android yang kita buat. AVD selanjutnya digunakan sebagai tempat untuk *test* dan menjalankan aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi Android, jenis, dan ukuran layar dan besarnya memori AVD bisa dibuat sebanyak yang kita inginkan [20].

### CodeIgniter

CodeIgniter merupakan sebuah *framework* yang dibuat dengan menggunakan bahasa PHP, yang dapat digunakan untuk pengembangan web secara cepat. Adapun *framework* sendiri dapat diartikan sebagai suatu struktur pustaka-pustaka, kelas-kelas dan infrastruktur *run*-*time* yang dapat digunakan oleh *programmer* untuk mengembangkan aplikasi web secara cepat. Tujuan penggunaan *framework* adalah untuk mempermudah pengembang web mengembangkan aplikasi web yang robust secara cepat tanpa kehilangan fleksibilitas.

Pola desain dalam pengembangan web dengan CodeIgniter menggunakan MVC (*Models*-*View*- *Controller*). Dimana aplikasi yang dibuat akan dipisahkan antara logika bisnis dan presentasinya, sehingga memungkinkan web *programmer* dan web designer bekerja secara terpisah antara satu dengan yang lain. Agar bisa mengembangkan web dengan CodeIgniter, maka perlu dipahami terlebih dahulu konsep MVC dan struktur direktori dari CodeIgniter [21].

### Kotlin

Kotlin adalah bahasa pemrograman berbasis Java *Virtual* *Machine* (JVM) yang dikembangkan oleh JetBrains. Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object* *oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk *backend*. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi diatas *platform* JVM adalah sebagai berikut:

Dapat mengatasi NullPointerException yang umumnya terdapat pada Java.

Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.

Mudah dipelajari

Dukungan IDE untuk mempermudah dalam pemrograman.

Perbandingan Kotlin dan java :

1. Pengukuran dilakukan dengan membangun aplikasi pada masing-masing bahasa pemrograman Java dan Kotlin. Kemudian mengukur performa aplikasi dengan mengoperasikan setiap fitur yang ada pada aplikasi. Dari operasi tersebut didapatkan hasil penggunaan CPU, memori dan kecepatan eksekusi aplikasi pada perangkat android.
2. Kotlin lebih sedikit dalam penggunaan CPU walaupun perbedaan hanya 0.65%
3. Kotlin lebih kecil dalam penggunaan memori dengan perhitungan lebih dari 2 kali lipat penggunaan memori pada Bahasa java.
4. Kotlin lebih cepat dalam mengeksekusi program [21].

### REST (*Representational* *State* *Transfer*)

REST (*Representational* *State* *Transfer*) merupakan seperangkat prinsip arsitektur yang melakukan transmisi data melalui antarmuka yang terstandarisasi seperti HTTP. REST API bekerja layaknya seperti aplikasi web biasa. *Client* dapat mengirimkan permintaan kepada server melalui protokol HTTP dan kemudian server memberikan respons balik kepada *client*. REST dikembangkan oleh Roy Fielding yang merupakan co-*founder* dari Apache HTTP *Server* *Project* [22]. Berikut adalah metode HTTP yang umumnya digunakan dalam arsitektur REST [23]:

1. GET untuk menyediakan akses untuk membaca sumber data

2. PUT untuk memperbarui data yang tersedia.

3. DELETE untuk menghapus data.

4. POST untuk membuat data baru.

# BAB III

**METODE PENELITIAN**

## Alat dan Bahan

Berikut merupakan alat dan bahan yang digunakan dalam merancang aplikasi:

1. Alat

Peralatan yang digunakan dalam merancang aplikasi ini adalah sebagai berikut:

* 1. Laptop asus VivoBook Max AMD Dual Core A9-9420, up to 3.6 GHz
  2. Sistem operasi Windows 10 Home Single Language 64-bit
  3. Android Studio
  4. MySQL (Xampp)
  5. Figma

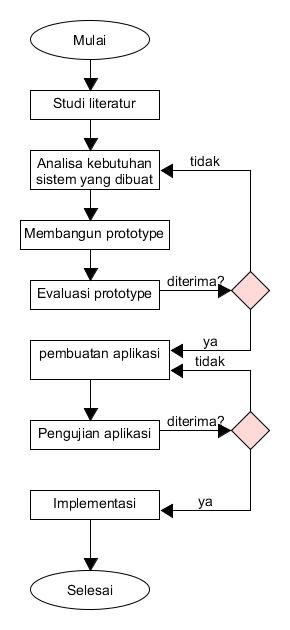
1. Bahan

Bahan-bahan yang akan digunakan dalam pembuatan sistem informasi ini adalah:

1. Literatur-literatur dari jurnal
2. Buku-buku elektronik
3. Penelitian-penelitian sebelumnya
4. Data-data yang diberikan oleh pihak pakar, seperti beberapa penyakit paling umum diderita hamster serta gejala-gejalanya.

## Alur Penelitian

Ada beberapa tahapan yang dilakukan dalam perancangan ini terdiri dari beberapa tahapan. Sistem informasi ini menggunakan metode pengembangan sistem *Prototype* yang terdiri dari: studi literatur, analisa kebutuhan sistem yang dibutuhkan, membangun *prototype*,evaluasi *prototype*, danpembuatanaplikasi*,* pengujianaplikasi*,* danimplementasi. Tahap-tahap pengembangan sistem tersebut diilustrasikan pada diagram alir atau *flowchart* yang dapat dilihat pada **Gambar 3.1**:



Gambar 3. 1 Diagram Alir Perancangan Aplikasi.

Penjelasan diagram alir akan dijelaskan pada sub-sub bab berikut:

### Studi Literatur

Tahap pertama dalam perancangan Aplikasi Penjualan Hamster dan Kelengkapannya adalah studi literatur. Studi literatur dilakukan dengan mempelajari dan memahami literatur yang berkaitan dengan penelitian yang di peroleh dari jurnal, buku, dan penelitian-penelitian sebelumnya. Literatur yang dipelajari adalah yang berkaitan dengan penggunaan metode *Prototype* (sebagai metode pengembangan), penjualan, dan mengenai 8 penyakit terumum yang dapat menyerang hamster serta gejala-gejalanya. Serta hal-hal yang terkait dengan perancangan Aplikasi Penjualan Hamster dan Kelengkapannya lainnya.

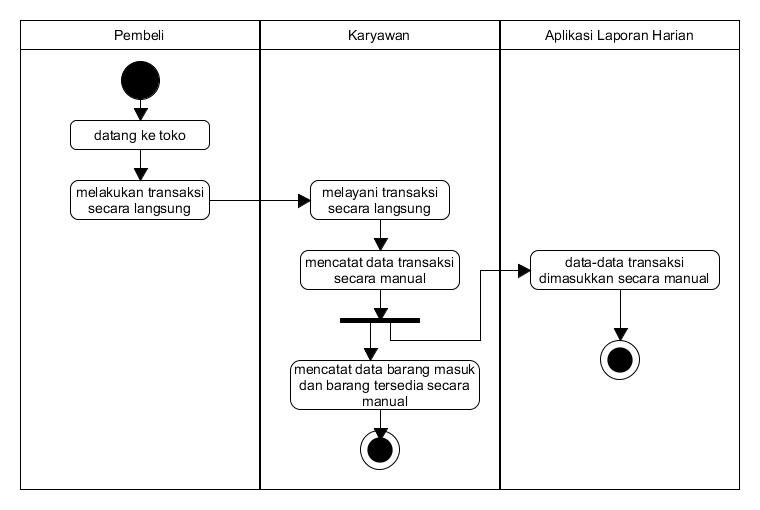
### Analisa Kebutuhan Sistem

Analisa kebutuhan sistem merupakan tahap awal dan salah satu tahap yang paling penting pada metode ini. Tahap ini menentukan bagaimana bentuk sistem yang akan dibangun. Pada tahap ini akan diperlihatkan sistem penjualan yang sedang berjalan serta yang diajukan pada toko Lombok Rabbit Pet House.

#### Analisa sistem yang sedang berjalan

Berikut ini merupakan proses berjalannya sistem penjualan pada toko Lombok Rabbit Pet House:

* + - 1. Pembeli datang ke toko Lombok Rabbit Pet House.
      2. Pembeli melakukan proses belanja secara manual dengan karyawan toko Lombok Rabbit Pet House.
      3. Data transaksi dicatat secara manual oleh karyawan toko Lombok Rabbit Pet House.
      4. Data-data transaksi yang telah dicatat selanjutnya dipindahkan ke aplikasi pembuatan laporan penjualan pada *handphone* pemilik sebagai laporan harian saat toko akan ditutup.
      5. Karyawan melakukan pencatatan barang masuk dan barang tersedia secara manual.



Gambar 3. 2 *Activity* *diagram* sistem yang sedang berjalan.

#### Analisa sistem yang diajukan

Berikut ini merupakan fitur-fitur pada sistem yang diusulkan untuk toko Lombok Rabbit Pet House dan telah disetujui.

* + - 1. **Sisi** **Penjual**

1. Beranda

Merupakan tampilan awal pada saat penjual berhasil *login*. Halaman ini mencakup beberapa menu, seperti: Hamster, Kelengkapan Hamster, Pakan Hamster, Tambah Barang, Laporan, dan daftar pesanan (laporan harian). Serta terdapat menu tambahan yang berisi profil penjual dan *logout*.

1. Hamster

Merupakan halaman yang berisi daftar data barang yang dijual dengan kategori “Hamster”. Pada halaman ini, penjual dapat melakukan pengubahan dan penghapusan data. Pada fitur edit (pengubahan), data yang dapat diganti adalah berupa: nama barang, jumlah barang, kategori barang, harga jual barang, harga beli barang, harga pokok barang, dan gambar.

1. Kelengkapan Hamster

Merupakan halaman yang berisi daftar data barang yang dijual dengan kategori “Kelengkapan Hamster”. Pada halaman ini, penjual dapat melakukan pengubahan dan penghapusan data. Pada fitur edit (pengubahan), data yang dapat diganti adalah berupa : nama barang, jumlah barang, kategori barang, harga jual barang, harga beli barang, harga pokok barang, dan gambar.

1. Pakan Hamster

Merupakan halaman yang berisi daftar data barang yang dijual dengan kategori “Pakan Hamster”. Pada halaman ini, penjual dapat melakukan pengubahan dan penghapusan data. Pada fitur edit (pengubahan) , data yang dapat diganti adalah berupa : nama barang, jumlah barang, kategori barang, harga jual barang, harga beli barang, harga pokok barang, dan gambar.

1. *Login*

Merupakan sebuah halaman *form* yang akan menampung *username* dan *password* penjual yang telah memiliki akun sebelum memasuki halaman beranda. Fitur ini digunakan sebagai validasi akun yang terkait.

1. Tambah Barang

Merupakan halaman *form* yang menampung beberapa informasi data baru yang akan dijual. Penjual hanya dapat menambah data dengan nama yang belum ada pada *database*. Pada *form* ini, data yang harus diisi adalah berupa : nama barang, jumlah barang, kategori barang, harga jual barang, harga beli barang, harga pokok barang, dan gambar.

1. Laporan

Fitur laporan terbagi menjadi tiga, yaitu : laporan pesanan, laporan penjualan terbanyak, dan laporan pembelian terbanyak. Pada laporan pesanan, akan ditampilkan daftar pesanan yang telah dikonfirmasi dan belum dikonfirmasi. Pada laporan penjualan terbanyak berisi tentang statistik berbentuk diagram balok yang menunjukkan 10 nama barang (dari segala kategori) dengan penjualan terbanyak dalam sebulan. Pada laporan pembelian terbanyak berisi tentang statistik berbentuk diagram balok yang menunjukkan 10 nama akun dengan pembelian terbanyak dalam sebulan.

1. Profil Penjual

Merupakan halaman yang memuat nama pengguna dari sisi penjual yang sedang *log* *in*. Pada fitur ini penjual juga dapat mengubah kata sandi-nya.

1. *Logout*

Berfungsi agar penjual dapat keluar dari akunnya setelah sebelumnya melakukan *login*. Fitur ini bertujuan untuk menjaga keamanan sebuah akun agar tidak digunakan orang lain. Penjual yang telah *logout* dari akunnya harus *login* kembali untuk dapat memasuki akun yang sama.

* + - 1. **Sisi Pembeli**

1. Daftar

Merupakan sebuah halaman *form* yang akan menampung *email*, *password*, alamat dan kode unikpembeli yang belum memiliki akun. Fitur ini digunakan sebagai langkah awal pembeli untuk dapat menggunakan layanan pada aplikasi.

1. *Login*

Merupakan sebuah halaman *form* yang akan menampung *email* dan *password* pembeli sebelum memasuki halaman beranda. Fitur ini digunakan sebagai validasi akun yang terkait.

1. *Logout*

Berfungsi agar pembeli dapat keluar dari akunnya setelah sebelumnya melakukan *login*. Fitur ini bertujuan untuk menjaga keamanan sebuah akun agar tidak digunakan orang lain. Pembeli yang telah *logout* dari akunnya harus *login* kembali untuk dapat memasuki akun yang sama.

1. Beranda

Merupakan tampilan awal pada saat pembeli berhasil *login*. Halaman ini mencakup beberapa menu, seperti: Hamster, Kelengkapan Hamster, Pakan Hamster, Konsultasi, Tentang Kami, dan Keranjang Belanja. Serta terdapat menu tambahan yang berisi profil pembeli dan *logout*.

1. Tentang Kami

Merupakan halaman berisi informasi singkat seputar informasi aplikasi. Beberapa ketentuan toko dapat dilihat oleh pembeli di halaman ini.

1. Konsultasi

Merupakan halaman berisi layanan konsultasi sederhana mengenai 8 penyakit terumum yang dapat menyerang hamster. Fitur ini akan dibuat dalam bentuk Q&A (*Question* *and* *Answer*) dimana pembeli hanya perlu menjawab sebuah pertanyaan singkat yang diikuti beberapa pilihan jawaban berupa gejala-gejala paling menonjol yang terjadi pada hamsternya. Gejala-gejala tersebut ditentukan berdasarkan nilai keyakinan terbesar yang diberikan oleh pakar pada suatu gejala terhadap sebuah penyakit. Selanjutnya, setelah memilih salah satu gejala tersebut maka pembeli akan mendapatkan hasil informasi penyakit yang mungkin diderita oleh hamsternya berdasarkan gejala yang dimasukkan beserta gejala-gejala pengikut yang mungkin juga nampak akibat penyakit tersebut. Contoh :

|  |
| --- |
| **Question :**  Apakah hamster mengalami salah satu dari gejala di bawah ini?   * + - 1. Kotoran hamster cair dan dalam jumlah banyak.       2. Munculnya keropeng di kulit hamster.       3. Anus hamster menjuntai keluar.       4. Jika diraba bagian perut hamster terasa keras.       5. Rambut rontok pada hamster.       6. Hamster menggaruk secara membabi buta di area telinga       7. Mata hamster mengeluarkan air mata terus-menerus.       8. Hamster kesulitan berjalan.   **Answer :**   * + - 1. Kotoran hamster cair dan dalam jumlah banyak.   **Dugaan penyakit :**  Wait tail  **Gejala yang mungkin juga akan nampak :**   * + - 1. Area anus hamster menjadi basah dan lengket,       2. Hamster menjadi tidak aktif,       3. Hamster tidak nafsu makan,       4. Hamster tampak kesakitan. |

1. Hamster

Merupakan halaman yang berisi data barang yang dijual dengan kategori “Hamster”. Pada halaman ini, pembeli dapat memilih hamster yang ingin dibeli.

1. Kelengkapan Hamster

Merupakan halaman yang berisi data barang yang dijual dengan kategori “Kelengkapan Hamster”. Pada halaman ini, pembeli dapat memilih kelengkapan hamster yang ingin dibeli.

1. Pakan Hamster

Merupakan halaman yang berisi data barang yang dijual dengan kategori “Pakan Hamster”. Pada halaman ini, pembeli dapat memilih pakan hamster yang ingin dibeli.

1. Profil Pembeli

Merupakan halaman yang memuat nama pengguna dari sisi pembeli yang sedang *log* *in*. Pada fitur ini penjual juga dapat mengubah kata sandi-nya.

### Desain

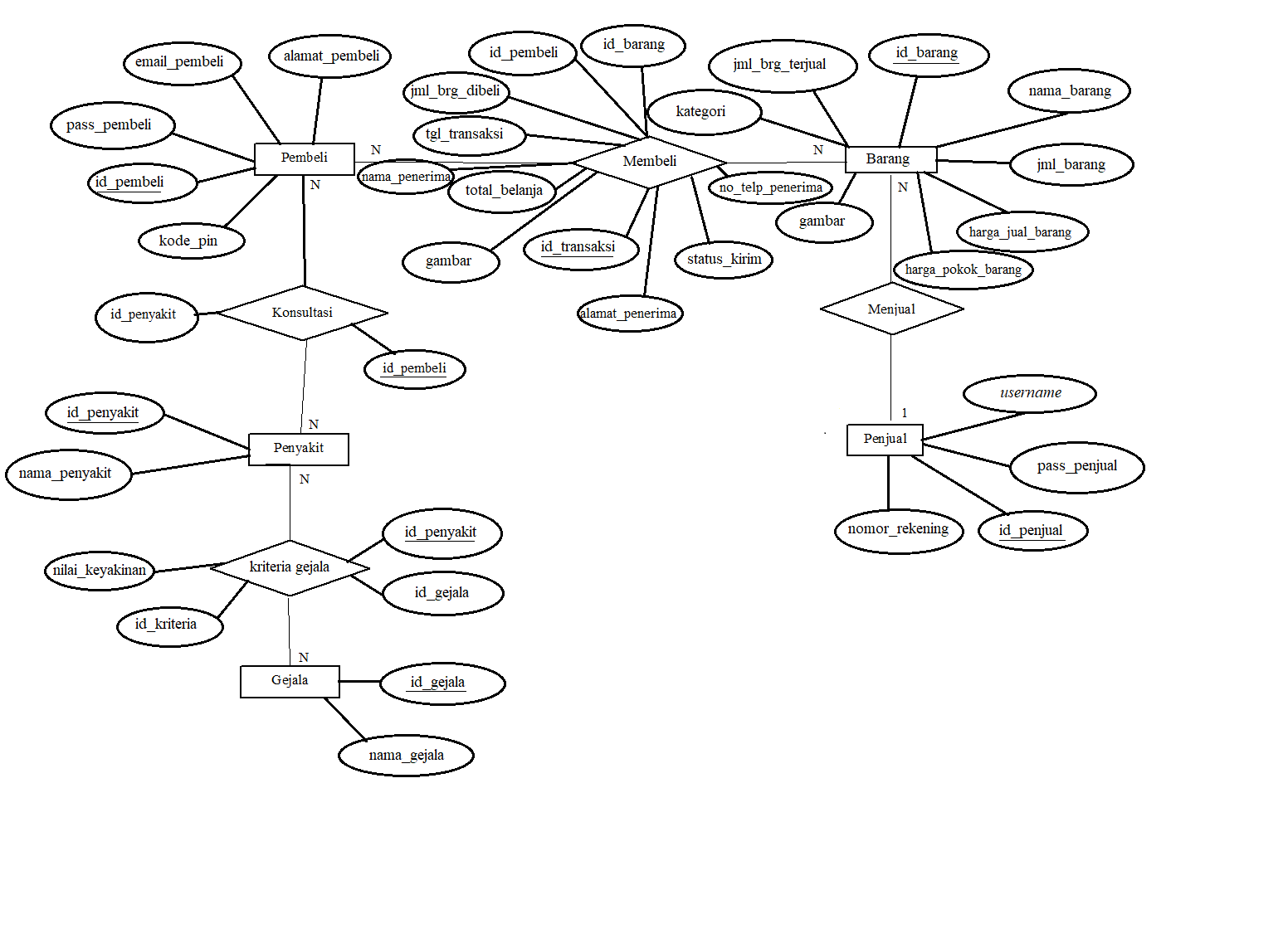
Tahap desain pada aplikasi ini mengacu pada dua hal penting, berupa: desain *database* dan desain sistem. Pada desain sistem berisi ERD dan struktur tabel pada sistem yang dibuat. Sedangkan pada desain sistem berisi UML dan desain *prototype* sistem.

#### Desain Database

Terdapat dua hal yang akan digunakan untuk membangun desain *database*, yaitu ERD dan struktur tabel. Berikut merupakan gambar ERD dan struktur tabel yang akan digunakan:

##### EDR

Berikut merupakan desain sistem yang akan digambarkan dalam bentuk diagram ERD **Gambar 3.3**:



Gambar 3. 3 ERD Aplikasi Penjualan Hamster dan Kelengkapannya.

Berdasarkan **Gambar 3.3**, dapat dilihat bahwa terdapat beberapa entitas pada yang saling berelasi satu sama lain, yaitu: pembeli, barang, penyakit, gejala, dan penjual. Terdapat juga beberapa *primary* *key* yang digunakan, yaitu : id\_pembeli , id\_transaksi, id\_barang, id\_penyakit, id\_penjual, dan id\_gejala. Serta terdapat beberapa *foreign* *key* yang digunakan, seperti: id\_pembeli, id\_barang, gambar, dan id\_penyakit.

##### Struktur Tabel

Berikut merupakan struktur tabel pada Aplikasi Penjualan Hamster dan Kelengkapannya:

Tabel 3. Tabel pembeli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_pembeli | *Int* | 5 | *Primary Key* pada tabel pembeli auto\_increment |
| 2. | email\_pembeli | *Varchar* | 50 |  |
| 3. | alamat\_pembeli | *Varchar* | 100 |  |
| 4. | Pass\_pembeli | *Varchar* | 8 |  |
| 5. | Kode\_pin | *Varchar* | 4 |  |

Tabel 3. Tabel penjual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_penjual | *Int* | 2 | *Primary Key* pada tabel penjual auto\_increment |
| 2. | *Username* | *Varchar* | 20 |  |
| 3. | Nomor\_rekening | *Varchar* | 20 |  |
| 4. | Pass\_penjual | *Varchar* | 8 |  |

Tabel 3. Tabel penyakit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_penyakit | *Varchar* | 3 | *Primary Key* pada tabel penyakit |
| 2. | Nama\_penyakit | *Varchar* | 30 |  |

Tabel 3. Tabel gejala

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_gejala | *Varchar* | 3 | *Primary Key* pada tabel gejala |
| 2. | Nama\_gejala | *Varchar* | 50 |  |

Tabel 3. Tabel barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_barang | *Int* | 5 | *Primary Key* pada tabel barang auto\_increment |
| 2. | Nama\_barang | *Varchar* | 30 |  |
| 3. | Jml\_barang | *Int* | 5 |  |
| 4. | Kategori | *Varchar* | 20 |  |
| 5. | Jml\_brg\_terjual | *Int* | 7 |  |
| 6. | Harga\_jual\_barang | *Int* | 7 |  |
|  | Gambar | *Varchar* | 100 |  |
|  | Harga\_pokok\_barang | *Int* | 7 |  |

Tabel 3. Tabel kriteria gejala

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_kriteria | *Int* | 3 | *Primary Key* pada tabel kriteria gejala auto\_increment |
| 2. | Id\_gejala | *Varchar* | 3 | *Foreignkey* |
| 3. | Id\_penyakit | *Varchar* | 3 | *Foreignkey* |
| 4. | Nilai\_keyakinan | *Double* | 5 |  |

Tabel 3. Tabel Membeli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_transaksi | *Int* | 5 | *Primary Key* pada tabel membeli |
| 2. | Id\_pembeli | *Int* | 5 | *Foreignkey* |
| 3. | Id\_barang | *Int* | 5 | *Foreignkey* |
| 4 | Gambar | *Varchar* | 100 |  |
| 5. | Jml\_brg\_dibeli | *Int* | 4 |  |
| 6. | Tgl\_transaksi | *Date* |  |  |
| 7. | Total\_belanja | *Int* | 4 |  |
| 8. | Status\_kirim | *Boolean* |  |  |
|  | Nama\_penerima | *Varchar* | 30 |  |
|  | Alamat\_penerima | *Varchar* | 50 |  |
|  | No\_telp\_penerima | *Varchar* | 13 |  |

Tabel 3. Tabel Konsultasi

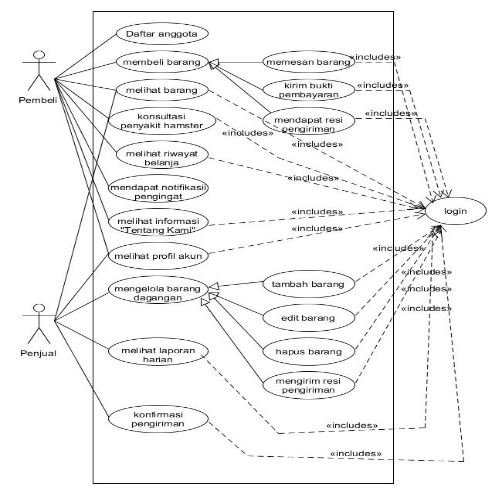
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Field* | *Type* | *Size* | Keterangan |
| 1. | id\_pembeli | *Int* | 5 | *Primary Key* pada tabel konsultasi auto\_increment |
| 2. | Id\_penyakit | *Varchar* | 3 |  |

#### Desain Sistem

Desain sistem akan dibangun dengan menggunakan beberapa diagram UML seperti : *use case* *diagram*, *activity* *diagram*, *class* *diagram*, dan *sequence* *diagram*. Berikut penjelasan mengenai desain sistem Aplikasi Penjualan Hamster dan Kelengkapannya :

##### Use case diagram

Pada *use* *case* *diagram*, perilaku masing-masing pengguna dapat dilihat. Berikut ini *use* *case* diagram pada Aplikasi Penjualan Hamster dan Kelengkapannya pada **Gambar 3.3**:



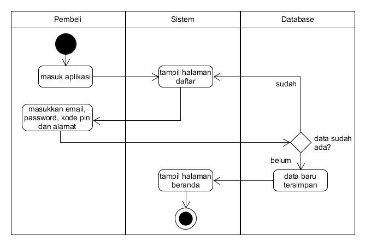
Gambar 3. *Use* *case* *Diagram* Sistem

Pada **Gambar 3.4** dapat dilihat bahwa terdapat dua aktor pada *use* *case*. Dimana aktor pertama adalah pembeli dan aktor kedua adalah penjual. Prilaku aktor pembeli pada *use* *case* adalah pembeli dapat melakukan daftar sebagai anggota, membeli barang, melihat riwayat belanja, melihat barang, konsultasi penyakit hamster, melihat profil akun, mendapat notifikasi pengingat, dan melihat informasi pada menu “Tentang Kami”. Pembeli dapat melakukan aktivitas-aktivitas tersebut (kecuali daftar sebagai anggota) setelah melakukan *log in* terlebih dahulu. Selanjutnya, prilaku aktor penjual adalah penjual dapat melihat barang, melihat profil akun, mengelola barang dagangan, melihat laporan harian, dan melakukan konfirmasi pengiriman. Penjual dapat melakukan aktivitas-aktivitas tersebut setelah melakukan *log in* terlebih dahulu.

##### Activity diagram

Berikut adalah *activity* *diagram* pada Sistem Penjualan dan Konsultasi Penyakit Hamster:

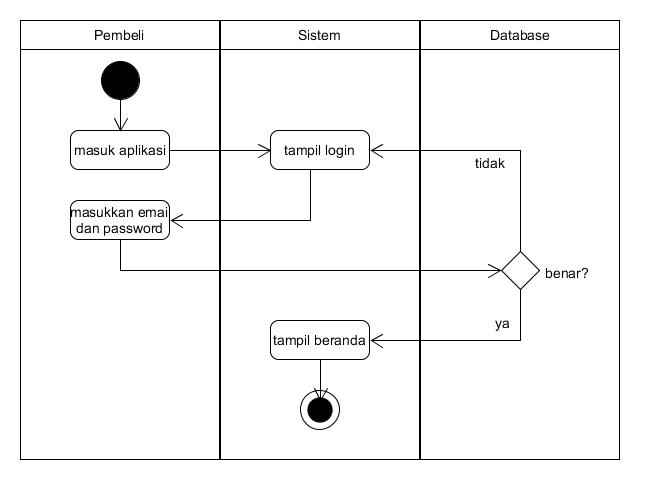
* 1. Pembeli mendaftar



Gambar 3. 5 *Activity* *Diagram* “Pembeli Mendaftar”

Pada **Gambar 3.5** merupakan gambaran pada saat pembeli melakukan pendafatran sebagai anggota. Pada tahap ini, pembeli membutuhkan *email*, *password*, dan alamat sebagai data pribadi yang akan disimpan di *database* sistem. Jika *email* yang dimasukkan sudah terdaftar, maka pembeli akan di arahkan kembali pada laman pendaftaran untuk memasukkan *email* yang berbeda.

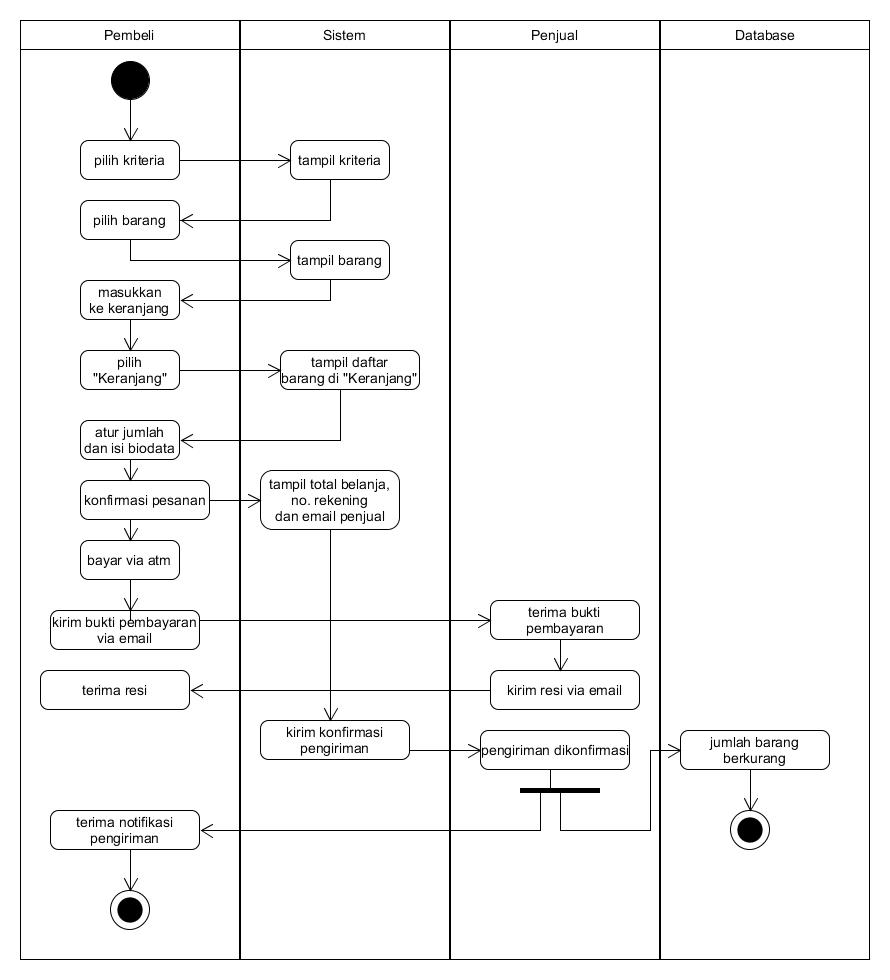
* 1. Proses *login*



Gambar 3. 6 *Activity* *Diagram* “Proses *Login*”

Pada **Gambar 3.6** merupakan gambaran pada saat pembeli melakukan *login*. Pada tahap ini, pembeli cukup memasukkan *email* dan *password* yang sebelumnya digunakan saat mendaftar. Jika *email* atau *password* yang dimasukkan salah, maka pembeli akan diarahkan pada laman *login* untuk memasukkan data yang benar.

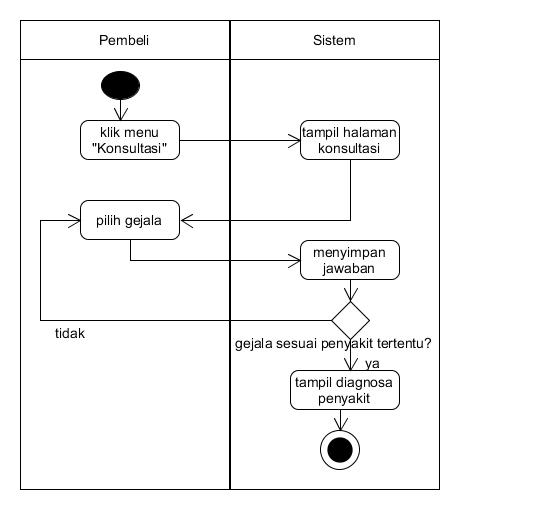
* 1. Proses belanja dan konfirmasi pengiriman



Gambar 3. 7 *Activity* *Diagram* “Proses Belanja dan Konfirmasi Pesanan”.

Pada **Gambar 3.7** merupakan gambaran pada saat pembeli melakukan proses belanja. Pada tahap ini, pembeli dapat memilih kriteria barang yang diinginkan yang selanjutnya akan mengarah pada halaman kriteria tersebut. Kemudian pembeli dapat memilih barang yang hendak dibeli dan mengatur jumlahnya pada halaman keranjang. Setelah mengisi biodata dan mengkonfirmasi pesanan dan membayar via ATM, pembeli harus mengirimkan bukti pembayaran via *email* kepada penjual. Selanjutnya, penjual akan mengirim nomor resi pengiriman via *email* dan melakukan konfirmasi pengiriman via aplikasi. Setelah pengiriman dikonfirmasi, jumlah barang akan otomatis berkurang pada *database*.

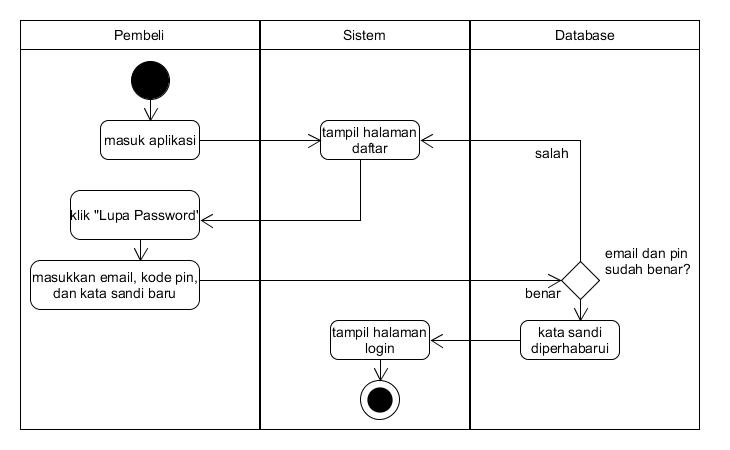
* 1. Proses konsultasi



Gambar 3. *Activity* *Diagram* “Proses konsultasi”

Pada **Gambar 3.8** merupakan gambaran pada saat pembeli melakukan proses konsultasipenyakit hamster. Pada tahap ini, pembeli dapat mengklik menu “Konsultasi”, kemudian halaman konsultasi akan menampilkan pertanyaan demi pertanyaan yang akan dijawab pembeli. Setelah jawaban disimpan dan dilakukan perhitungan, maka sistem akan menampilkan diagnosa penyakit dan beberapa gejala spesifik (yang mungkin juga akan nampak pada penyakit tersebut) yang mengikuti gejala utama.

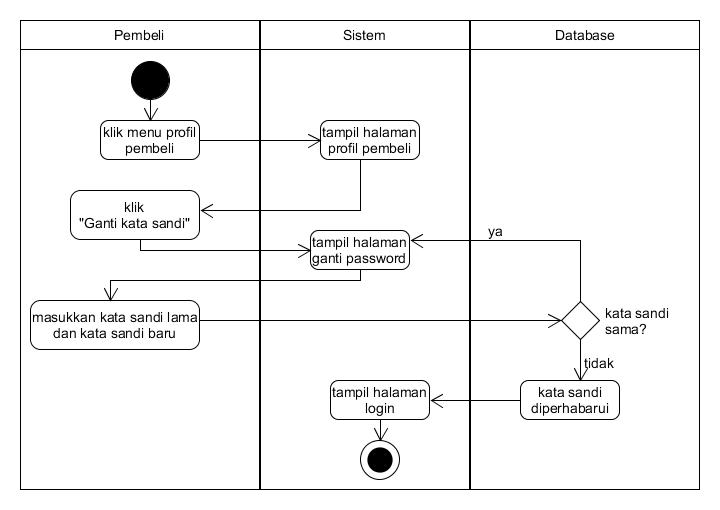
* 1. Lupa kata sandi



Gambar 3. 9 *Activity* *Diagram* “Pembeli Lupa Kata sandi”.

Pada **Gambar 3.9** merupakan gambaran pada saat pembeli mengklik “Lupa Kata sandi”. Pada tahap ini, pembeli diminta untuk mengisi *email*, kode unik/ pin yang sebelumnya telah dibuat saat mendaftar, dan kata sandi baru untuk mengganti kata sandinya.

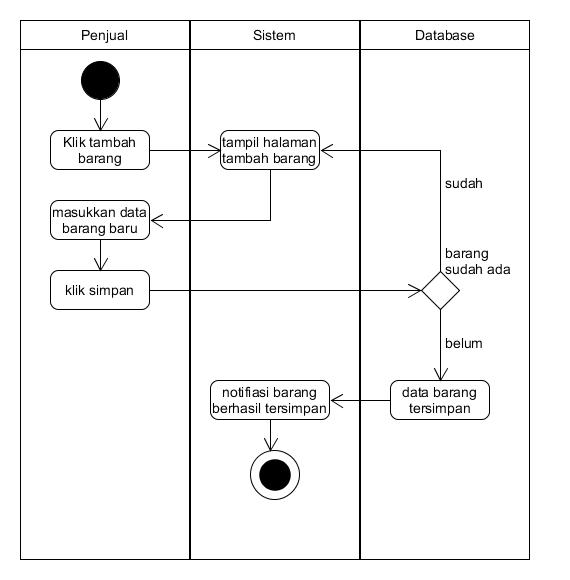
* 1. Ganti kata sandi



Gambar 3. *Activity* *Diagram* “Ganti Kata sandi”

Pada **Gambar 3.10** merupakan gambaran pada saat pembeli mengklik “Ganti Kata sandi”. Pada tahap ini, pembeli diminta untuk mengisi kata sandi baru dan kata sandi lamanya sebagai validasi untuk mengganti kata sandinya.

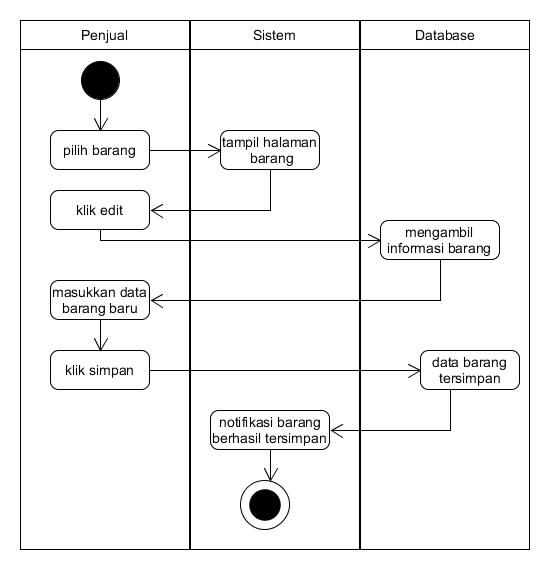
* 1. Proses tambah barang



Gambar 3. 11 *Activity* *Diagram* “Tambah Barang”

Pada **Gambar 3.11** merupakan gambaran pada saat penjual menambah barang dagangan. Pada tahap ini, penjual dapat mengklik menu “Tambah Barang”, kemudian mengisi data barang yang ditambahkan dan mengklik “Simpan”. Apabila data sudah ada, maka penjual akan diarahkan ke halaman “Tambah Barang” untuk mengisi data yang berbeda.

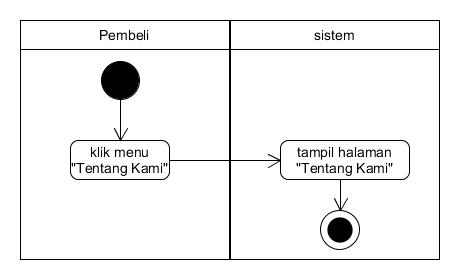
* 1. Proses edit barang



Gambar 3. *Activity* *Diagram* “Edit Barang”

Pada **Gambar 3.12** merupakan gambaran pada saat penjual mengedit data barang dagangan. Pada tahap ini, penjual dapat memilih barang yang akan diedit (baik nama maupun jumlah barang tersedia), kemudian mengklik edit. Selanjutnya sistem akan mengambil data barang pada *database* sehingga penjual dapat melakukan pengeditan. Setelah itu, data akan tersimpan setelah penjual mengklik “Simpan”.

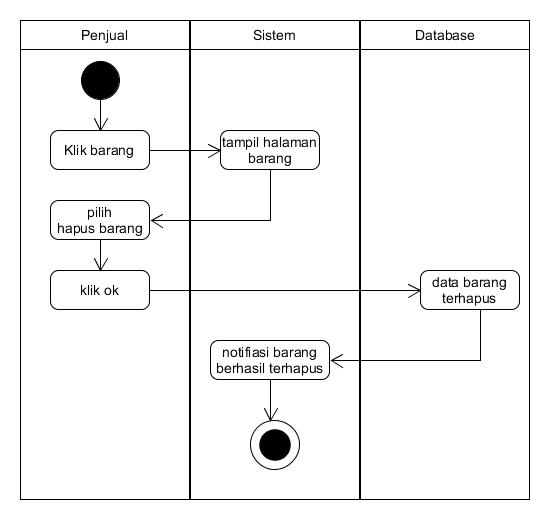
* 1. Melihat tentang kami



Gambar 3. 13 *Activity* *Diagram* “Tentang Kami”

Pada **Gambar 3.13** merupakan gambaran pada saat pembeli melihat informasi aplikasi. Pada tahap ini, pembeli dapat melihat informasi pada halaman “Tentang Kami” dengan memilih menu tersebut.

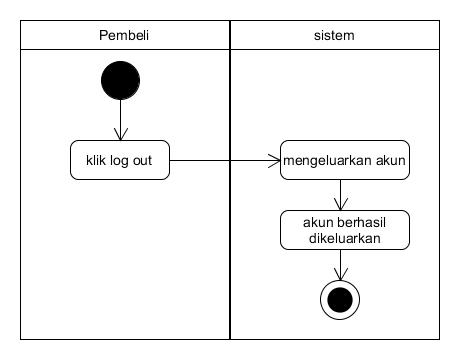
* 1. Proses hapus barang



Gambar 3. 14 *Activity* *Diagram* “Hapus Barang”

Pada **Gambar 3.14** merupakan gambaran pada saat penjual menghapus barang dagangan. Pada tahap ini, penjual dapat memilih barang yang ingin dihapus, kemudian data barang pada *database* akan terhapus.

* 1. Proses *logout*

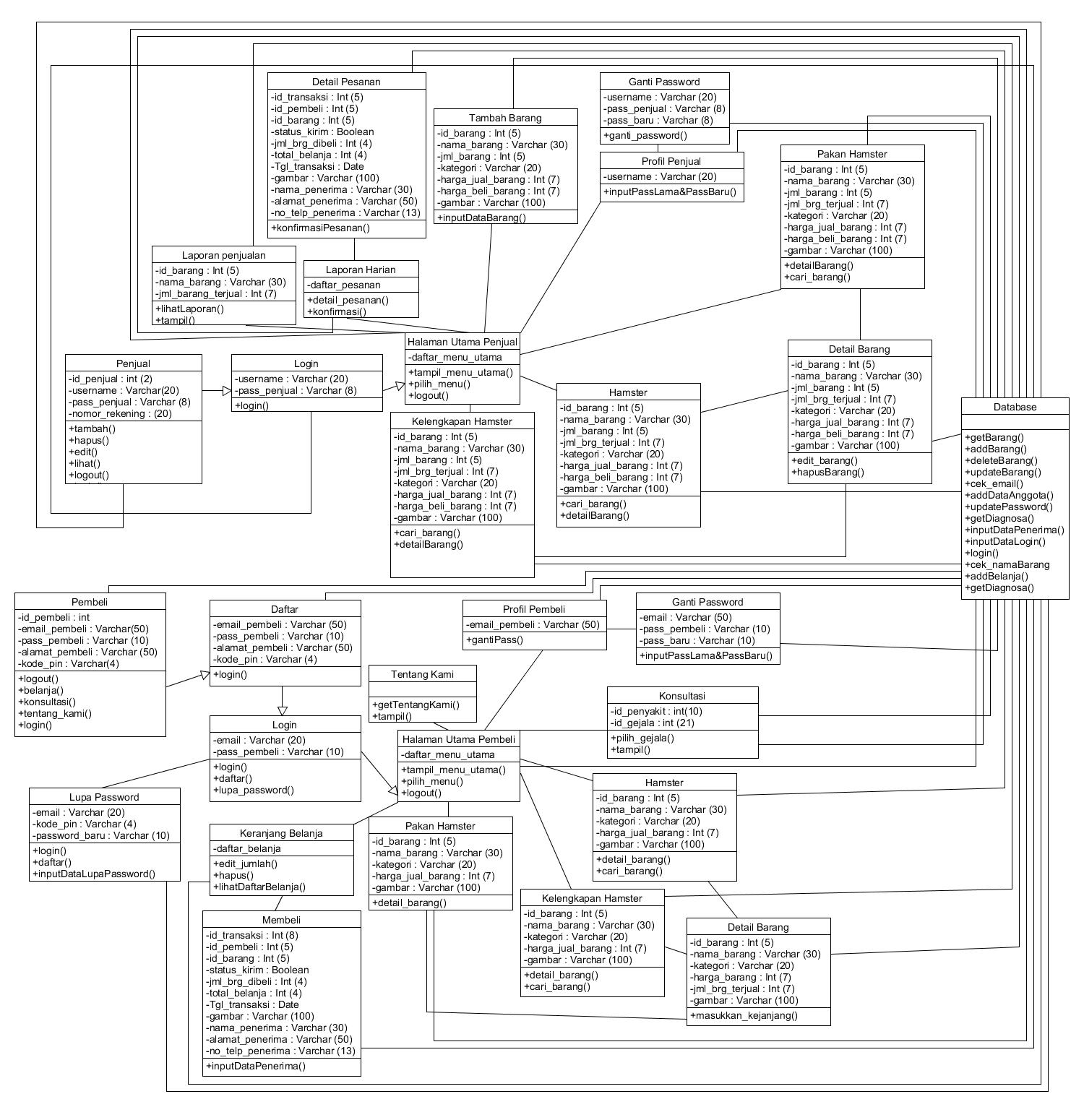


Gambar 3. 15 *Activity* *Diagram* “*Logout*”

Pada **Gambar 3.15** merupakan gambaran pada saat pembeli keluar dari akunnya. Pada tahap ini, pembeli dapat keluar dengan memilih menu “*Logout*”.

##### Class diagram

Pada **Gambar 3.16** merupakan *class* *diagram* pada sistem penjualan dan konsultasi penyakit hamster. Terdapat *class* penjual, *login*, halaman utama penjual, laporan harian, laporan penjualan, detail pesanan, tambah barang, kelengkapan hamster, profil penjual, ganti *password*, pakan hamster, hamster, dan edit barang. Sedangkan untuk pembeli, terdapat *class* pembeli, daftar, *login*, lupa *password*, keranjang belanja, membeli, tentang kami, halaman utama pembeli, pakan hamster, profil pembeli, ganti *password*, konsultasi, diagnosa, hamster, kelengkapan hamster, detail barang.



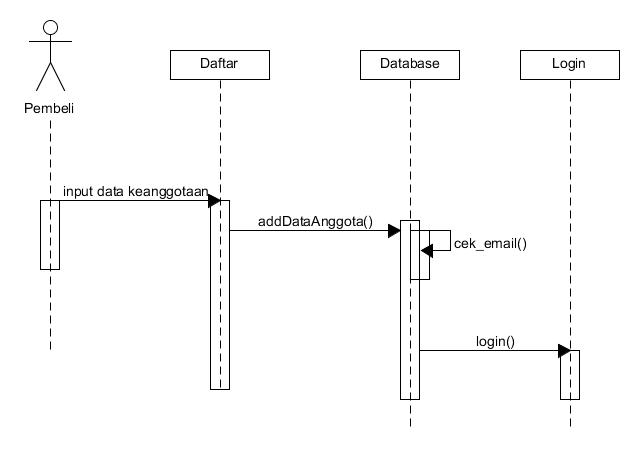
Gambar 3. *Class* *Diagram*

***Sequence* *diagram***

Berikut adalah *Sequence* *Diagram* pada sistem penjualan dan konsultasi penyakit hamster :

Pembeli

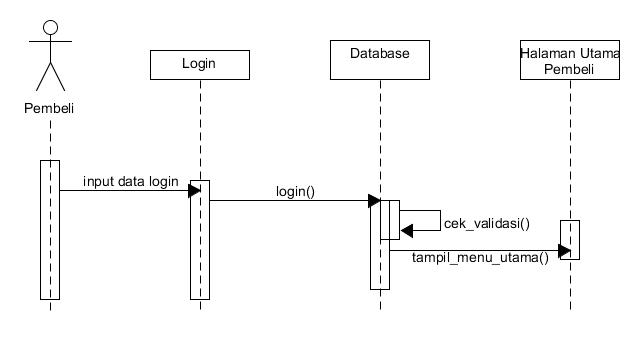
1. Pembeli mendaftar



Gambar 3. 17 *Sequence* *Diagram* “Pembeli Mendaftar”

Pada **Gambar 3.17** merupakan tahap-tahap bagaimana pembeli melakukan pendaftaran. Setelah memasuki layar daftar, pembeli diminta untuk memasukkan *email*, kata sandi, kode unik/ pin, dan alamat. Kemudian, *class* Database akan memanggil *method* “addDataAnggita()”, kemudian akan dilakukan pengecekan oleh *database* dengan dipanggilnya *method* “cek\_email()”, apabila *email* yang digunakan belum ada di *database*, maka data akan tersimpan dan pembeli akan diarahkan pada halaman pembeli dengan *method* “login()” yang dipanggil oleh *class* Login.

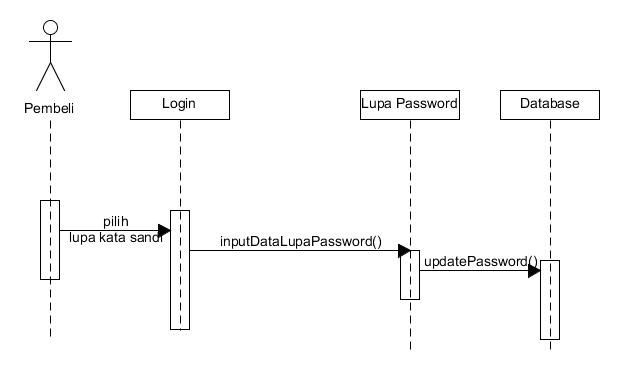
1. Proses *login*



Gambar 3. *Sequence* *Diagram* “*Login*”

Pada **Gambar 3.18** merupakan tahap-tahap bagaimana pembeli melakukan *login* aplikasi. Pada layar *login*, pembeli diminta untuk memasukkan *email* dan kata sandi. Kemudian *class* Database akan memanggil *method* “login()” dan melakukan pengecekan validasi dengan memanggil *method* “cek\_validasi()”, apabila data yang dimasukkan benar, maka *method* “tampil\_menu\_utama()” akan dipanggil oleh *class* Halaman Utama Pembeli dan pembeli akan berhasil masuk ke halaman utama pembeli.

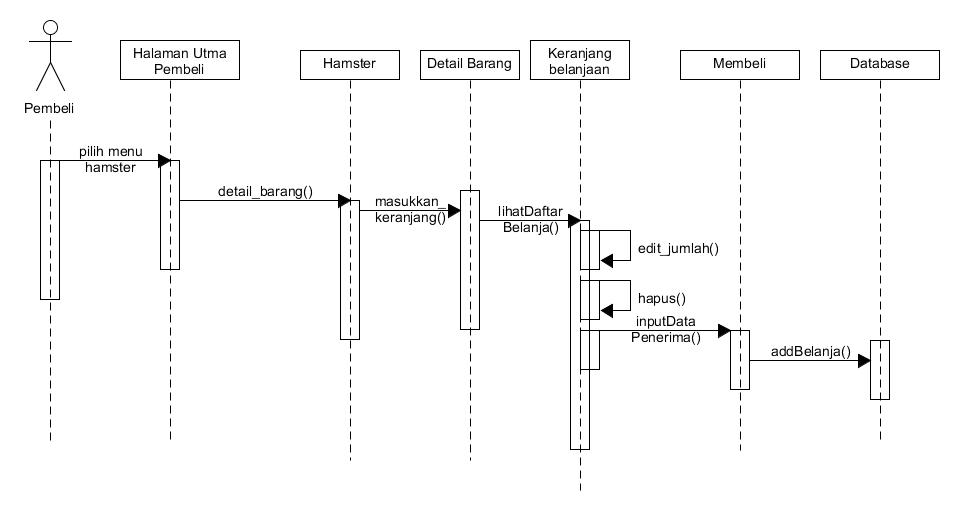
1. Lupa kata sandi



Gambar 3. 19 *Sequence* *Diagram* “Lupa Kata Sandi”

Pada **Gambar 3.19** merupakan tahap-tahap apabila pembeli lupa kata sandinya. Setelah mengklik “lupa kata sandi?” (lupa *password*), maka *class* Lupa Password akan memanggil *method* “inputDataLupaPassword()”. Lalu pembeli diminta untuk memasukkan *email* dan kode pin yang dibuat saat mendaftar, kata sandi lama, dan kata sandi barunya. Kemudian *class* Database akan memanggil *method* “updatePassword()” dan kata sandi baru akan tersimpan.

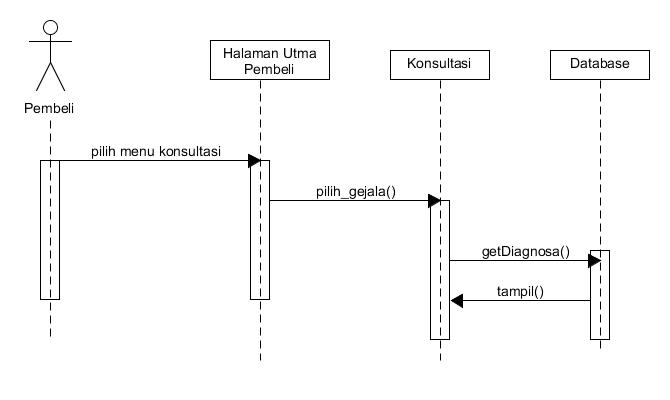
1. Proses belanja



Gambar 3. 20 *Sequence* *Diagram* “Proses Belanja”

**Gambar 3.20** menunjukkan bagaimana pembeli melakukan proses belanja. Setelah berhasil masuk, pembeli dapat memilih menu yang tersedia, misalkan pembeli memilih halaman Hamster. Saat barang yang diinginkan dipilih, *class* Hamster akan memanggil *method* “detail\_barang()”. Pada halaman Detail Barang, *class* Detail Barang akan memanggil *method* “masukkan\_keranjang()” apabila pembeli ingin membeli barang tersebut. Setelah selesai berbelanja, pembeli dapat membuka halaman Keranjang Belanjaan. Pada *class* Keranjang Belanjaan terdapat 3 *method* pilihan berupa “edit\_jumlah()”,“hapus()”, dan “lihatDaftarBelanja()”. Kemudian *class* Membeli memanggil *method* “inputDataPesanan()” apabila data yang dimasukkan sudah dikonfirmasi oleh pembeli. Selanjutnya *class* Database akan memanggil *method* “addBelanja()”. Kemudian, pembeli membayar pesanan via ATM dan melakukan pengiriman bukti pembayaran via *email*.

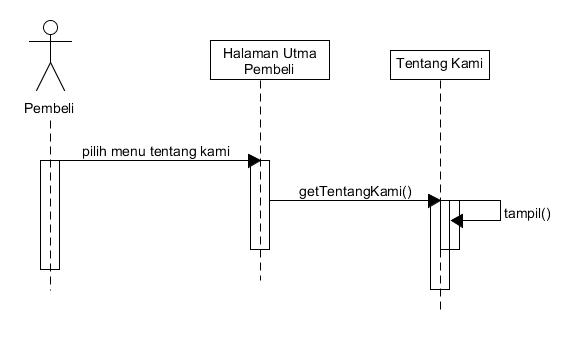
1. Proses konsultasi



Gambar 3. 21 *Sequence* *Diagram* “Konsultasi”

Pada **Gambar 3.21** merupakan tahap-tahap bagaimana pembeli melakukan konsultasi penyakit hamster. Setelah berhasil *login*, pembeli dapat memilih menu konsultasi dan menjawab pertanyaan-pertanyaan seputar gejala yang mungkin terjadi pada hamster. Di sini, *class* Konsultasi akan memanggil *method* “pilih gejala()”. Apabila gejala sudah dipilih, kemudian *class* Database akan memanggil *method* “getDiagnosa()”. Setelah itu, informasi diagnosa dengan *method* “tampil()” yang dipanggil oleh *class* Konsultasi.

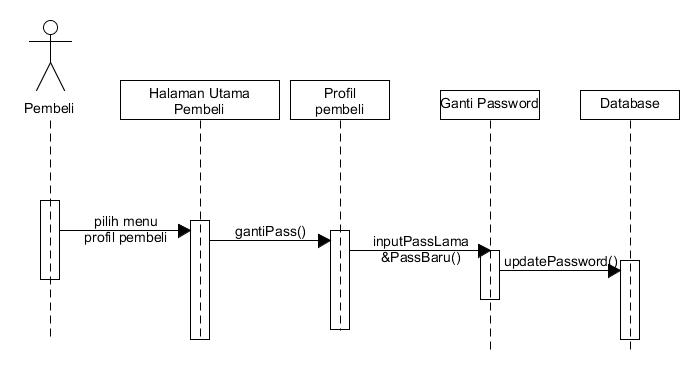
1. Melihat tentang kami



Gambar 3. *Sequence* *Diagram* “Lihat tentang Kami”

Pada **Gambar 3.22** merupakan tahap-tahap bagaimana pembeli melihat informasi pada menu “Tentang Kami”. Setelah berhasil *login*, pembeli dapat memilih menu “Tentang Kami”. Kemudian *class* Tentang Kami akan memanggil *method* “getTentangKami()”, kemudian informasi dari halaman “Tentang Kami” dan ditampilkan dengan *method* “tampil()” .

1. Ganti kata sandi

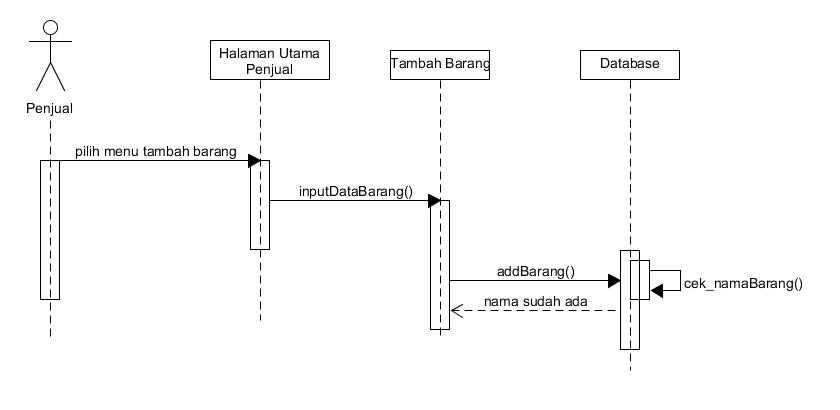


Gambar 3. *Sequence* *Diagram* “Ganti kata sandi”

Pada **Gambar 3.23** merupakan tahap-tahap apabila pembeli ingin mengganti kata sandinya. Setelah berhasil *login* dan memilih menu Profil Pembeli, pembeli mengklik “Ganti Kata sandi” sehingga *method* “gantiPass()” dipanggil, kemudian *class* Ganti Password akan memanggil *method* “inputPassLama&PassBaru()” setelah pembeli diminta untuk memasukkan email, kata sandi lama, dan kata sandi barunya. Kemudian *method* “updatePasword()” akan dipanggil oleh *class* Database untuk menyimpan kata sandi baru.

Penjual

1. Tambah barang



Gambar 3. 24 *Sequence* *Diagram* “Tambah Barang”

Pada **Gambar 3.24** merupakan tahap-tahap bagaimana penjual menambah barang dagangan. Setelah berhasil *login*, penjual dapat memilih menu “Tambah Barang”. Kemudian, *method* “inputDataBarang()” akan dipanggil oleh *class* Tambah Barang dan menampilkan sebuah *form*. Penjual harus mengisi beberapa informasi barang, seperti : nama barang, kategori barang, jumlah barang, harga jual barang, harga pokok barang dan gambar barang pada *form* tersebut. Kemudian *class* Database akan memanggil *method* “addBarang()” setelah data pada *form* yang dibutuhkan telah diisi sebelumnya. Kemudian nama barang akan dicek dengan memanggil *method* “cek\_namaBarang()”. Dan data akan tersimpan apabila nama barang belum terdaftar pada *database*.

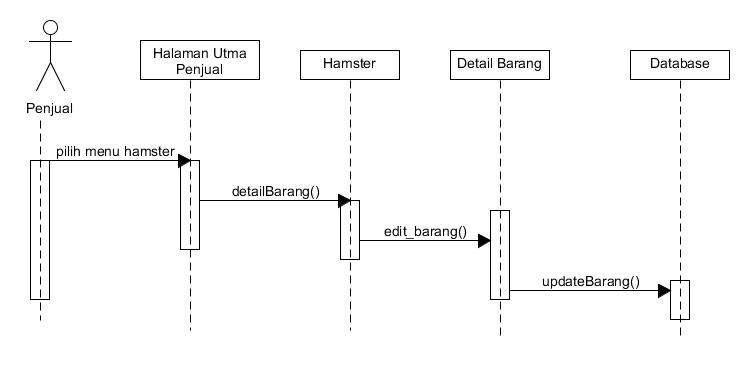
1. Hapus barang



Gambar 3. 25 *Sequence* *Diagram* “Hapus Barang”

Pada **Gambar 3.25** merupakan tahap-tahap bagaimana penjual menghapus barang dagangan. Setelah berhasil *login*, penjual dapat memilih kategori barang yang akan dihapus, kemudian memilih barang yang akan dihapus pada kategori tersebut sehingga *class* kategori, misalkan pada kasus ini adalah *class* Hamster akan memanggil *method* “detailBarang()”. Dan saat dihapus, maka *class* Detail Barang akan memanggil *method* “hapusBarang()”. Selanjutnya, *class* Database akan memanggil *method* “deleteBarang()”.

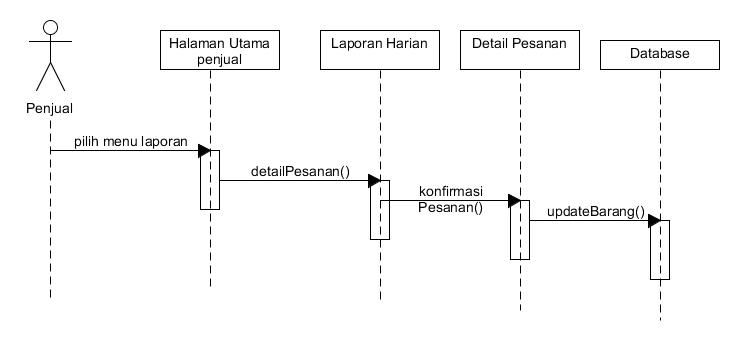
1. Edit barang



Gambar 3. *Sequence* *Diagram* “Edit Barang”

Pada **Gambar 3.26** merupakan tahap-tahap bagaimana penjual mengedit barang dagangan. Setelah berhasil *login*, penjual dapat memilih kategori barang yang akan diedit, dalam kasus ini adalah “Hamster”. Kemudian *class* Hamster akan memanggil *method* “detailBarang()” untuk menampilkandetail barang yang dapat diedit. Lalu, *method* “edit\_barang()” akan dipanggil oleh *class* Detail Barang setelah data pada *form* yang dibutuhkan telah diedit sebelumnya. Selanjutnya, *method* “updateBarang()” akan dipanggil oleh class Database dan data akan tersimpan.

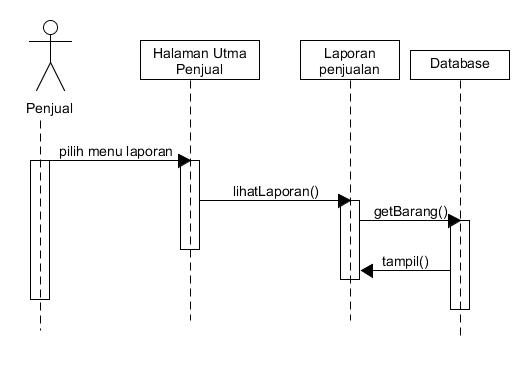
1. Proses konfirmasi pesanan



Gambar 3. 27 *Sequence* *Diagram* “Proses Konfirmasi Pesanan”

**Gambar 3.27** menunjukkan bagaimana penjual melakukan proses konfirmasi pesanan. Setelah melakukan *login*, dan masuk ke halaman laporan harian, *class* Laporan harian akan memanggil *method* “detailPesanan()” untuk melihat detail pesanan. Lalu, apabila pesanan dikonfirmasi penjual, maka *class* Detail Pesanan akan memanggil *method* “konfirmasiPesanan()”. Sehingga *method* “updateBarang()” akan dipanggil oleh *class* Database dan mengakibatkan jumlah barang yang dibeli berkurang. Selanjutnya pengiriman nomor resi barang akan dilakukan via email oleh penjual.

1. Lihat laporan



Gambar 3. *Sequence* *Diagram* “Melihat laporan”

Pada **Gambar 3.28** merupakan tahap-tahap bagaimana penjual melihat laporan. Setelah berhasil *login*, pembeli dapat memilih menu “Laporan Penjualan”. Kemudian *database* akan mengambil data “Laporan Penjualan” dengan memanggil *method* “getBarang()” dan ditampilkan pada halaman penjual dengan memanggil *method* “tampil()”.

### Membangun *Prototype*

Tahap selanjutnya setelah mengetahui dan mendapatkan seluruh kebutuhan yang diperlukan dalam pengembangan adalah pembuatan *prototype* sistem. Pada tahap ini, dilakukan perancangan dengan pembuatan dokumentasi berupa *mockup* seperti berikut :

**Sisi Pembeli**



Gambar 3. *Splash* Aplikasi

**Gambar 3.29** merupakan tampilan pertama saat aplikasi pertama kali dibuka oleh pengguna (pembeli maupun penjual). Halaman ini berisi logo perusahaan.



Gambar 3. *Form Login*

Pada **Gambar 3.30** merupakan tampilan *form login* yang dapat diakses oleh pembeli maupun penjual. Terdapat kolom email dan kata sandi yang harus diisi oleh pengguna.



Gambar 3. *Form* Daftar untuk Pembeli

Pada **Gambar 3.31** merupakan tampilan *form* Daftarbagi pembeli yang ingin membuat akun baru. Terdapat kolom *email*, kata sandi, alamat, dan kode unik / pin yang harus diisi oleh pembeli untuk kebutuhan penggantian kata sandi.



Gambar 3. Lupa kata sandi

Pada **Gambar 3.32** merupakan tampilan apabila pembeli mengklik “lupa kata sandi?” pada halaman *login*. Pada halaman ini pembeli diminta untuk mengisi *email*, kode pin dan kata sandi baru untuk dapat mengganti kata sandinya.



Gambar 3. Halaman beranda pembeli

**Gambar 3.33** merupakan tampilan ketika pembeli berhasil masuk ke akunnya. Terdapat beberapa menu pada beranda pembeli, seperti: Hamster, Pakan Hamster, Kelengkapan Hamster, Tentang Kami, Konsultasi, dan Daftar Pesanan. Pada menu Daftar Pesanan terdapat fitur keranjang belanja dan riwayat pesanan. Kemudian terdapat satu menu tambahan yang berisi menu profil pembeli dan menu *logout*. Menu tambahan ini juga tersedia pada sisi penjual.



Gambar 3. Halaman “Hamster” Pembeli

**Gambar 3.34** merupakan tampilan ketika pembeli mengklik menu “Hamster”. Aplikasi akan menampilkan barang dagangan dengan kategori “Hamster” disertai dengan deskripsi singkat berupa nama dan harga barang. Tampilan pada kategori “Kelengkapan Hamster” dan “Pakan Hamster” juga memuat jenis data yang sama berdasarkan kategorinya. Tampilan serupa juga terjadi pada sisi penjual.



Gambar 3. Halaman Detail Barang

**Gambar 3.35** merupakan tampilan ketika pembeli mengklik salah satu barang pada kategori tertentu. Aplikasi akan menampilkan detail barang dagangan berdasarkan kategori disertai dengan deskripsi yang lebih lengkap dan fitur keranjang dengan mengklik tombol “beli” apabila pembeli ingin membeli barang tersebut.



Gambar 3. Halaman Keranjang Belanja

**Gambar 3.36** merupakan tampilan ketika pembeli mengklik menu “Keranjang belanja”. Aplikasi akan menampilkan barang yang telah dipilih pembeli. Pembeli bisa mengatur jumlah barang yang ingin dibeli di sini.



Gambar 3. Halaman Biodata Penerima

**Gambar 3.37** merupakan tampilan setelah pembeli melanjutkan pemesanan setelah sebelumnya mengatur jumlah barang yang akan dibeli. Pada halaman ini, pembeli diminta untuk melengkapi biodata penerima selengkap mungkin.



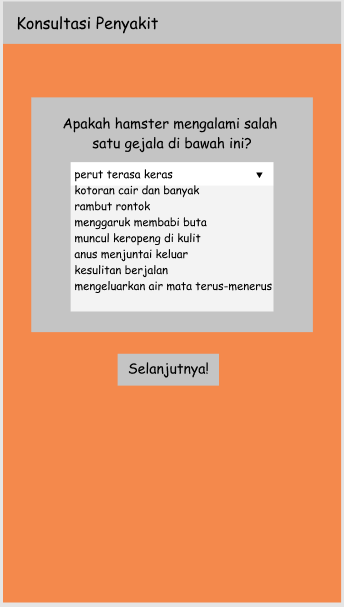
Gambar 3. Halaman “Pembayaran”

**Gambar 3.38** merupakan tampilan ketika pembeli telah mengisi biodata penerima. Aplikasi akan menampilkan nomor rekening penjual dan menginformasikan agar pembeli mengirim bukti pembayaran pada email yang diberikan.



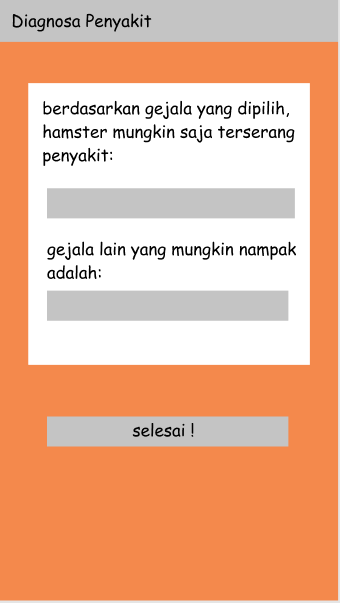
Gambar 3. Halaman “Riwayat Belanja”

**Gambar 3.39** merupakan tampilan dari menu “Riwayat Belanja”. Aplikasi akan merekam barang dagangan yang telah dibeli dan dikonfirmasi pengirimannya oleh penjual pada menu ini.



Gambar 3. Halaman “Konsultasi”

**Gambar 3.40** merupakan tampilan ketika pembeli mengklik menu “Konsultasi”. Aplikasi akan menampilkan beberapa gejala penyakit hamster yang akan dijawab oleh pembeli sesuai dengan kondisi hamster. Gejala-gejala yang ditampilkan adalah gejala dengan nilai keyakinan pakar berskala prioritas tertinggi di tiap penyakit.



Gambar 3. Halaman “Diagnosa Penyakit”

**Gambar 3.41** merupakan tampilan hasil diagnosa penyakit. Aplikasi akan menampilkan nama penyakit dan gejala-gejala yang mungkin timbul juga pada penyakit tersebut.



Gambar 3. Halaman “Tentang Kami” Pembeli

**Gambar 3.42** merupakan tampilan ketika pembeli membuka halaman Tentang Kami. Terdapat beberapa informasi umum mengenai perusahaan pada halaman tersebut.

**Sisi Penjual**



Gambar 3. Beranda Penjual

**Gambar 3.43** merupakan tampilan ketika penjual berhasil masuk ke akunnya. Terdapat beberapa menu pada beranda penjual, seperti: Hamster, Pakan Hamster, Kelengkapan Hamster, Laporan, Daftar Pesanan dan Konsultasi. Serta menu tambahan yang memuat fitur profil pembeli dan *logout*.



Gambar 3. Halaman “Tambah Barang”

**Gambar 3.44** merupakan tampilan ketika penjual ingin menambahkan barang baru. Terdapat beberapa kolom yang harus diisi, seperti: nama barang, jumlah barang, kategori, harga pokok, harga jual, deskripsi, dan gambar. Dan penjual dapat mengklik “simpan!” untuk menyimpan data tersebut.



Gambar 3. Halaman “Tambah Barang”

**Gambar 3.45** merupakan tampilan ketika penjual ingin mengubah data barang tertentu. Terdapat beberapa kolom yang bias diubah, seperti : nama barang, jumlah barang, kategori, harga pokok, harga jual, deskripsi, dan gambar. Dan penjual dapat mengklik “simpan!” untuk menyimpan data tersebut.



Gambar 3. Halaman “Detail Barang” Penjual

**Gambar 3.46** merupakan tampilan ketika penjual mengklik salah satu barang pada kategori tertentu. Aplikasi akan menampilkan detail barang dagangan berdasarkan kategori disertai dengan deskripsi yang lebih lengkap. Disini penjual dapat melakukan edit dan hapus barang sesuai kebutuhan.



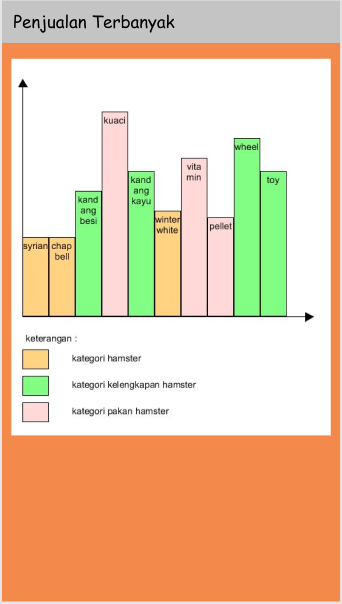
Gambar 3. Halaman “Daftar Pesanan”

**Gambar 3.47** merupakan tampilan ketika penjual membuka halaman Daftar Pesanan. Aplikasi akan menampilkan barang dagangan yang telah dipesan pembeli, serta fitur “dikirim” untuk konfirmasi pengiriman via aplikasi.



Gambar 3. Halaman “Detail Pesanan” Penjual

**Gambar 3.48** merupakan tampilan ketika penjual mengklik salah satu pesanan. Aplikasi akan menampilkan detail pesanan berupa: informasi pembeli, alamat pengiriman, dan pesanan.



Gambar 3. Salah Satu Contoh Halaman Statistik Penjualan

**Gambar 3.49** merupakan tampilan salah satu contoh ketika pembeli mengklik fitur “Statistik Penjualan” pada menu “Laporan”. Contoh halaman ini menunjukkan grafik dari urutan 10 data barang dengan kuantitas penjualan terbanyak dalam bentuk diagram balok. Pada statistik seperti yang ditampilkan di **Gambar 3.49**, laporan yang akan disiapkan adalah data barang dengan jumlah penjualan terbanyak pada periode bulan ke-n.

### Evaluasi *Prototype*

Pada tahap ini, evaluasi dilakukan dengan cara membagikan kuisioner kepada masyarakat sebagai pihak calon pembeli serta pemilik toko dan karyawan “Lombok Rabbit Pet House”. Jika *prototype* sudah sesuai dengan keinginan semua pengguna, maka tahap pembuatan dapat dilanjutkan pada pembuatan aplikasi secara keseluruhan. Jika belum, maka akan dilakukan analisa kebutuhan dan kembali membuat *prototype*.

### Pembuatan Aplikasi

Setelah *prototype* telah lulus evaluasi, selanjutnya dapat dilakukan pembuatan aplikasi secara utuh. Dalam penelitian kali ini, pembuatan aplikasi menggunakan bahasa pemrograman Kotlin dengan memanfaatkan MySQL sebagai pengelolaan data pada *database*.

### Pengujian Aplikasi

Sistem yang telah diselesaikan pada tahap *coding* kemudian akan diuji dari segi tampilan dan sistemnya. Hal ini dilakukan untuk memastikan sistem dapat digunakan dengan baik dan layak atau masih memerlukan perbaikan. Dalam penelitian ini digunakan tiga bentuk pengujian.

Pengujian pertama merupakan *blackbox* *testing* yang berfungsi untuk menguji sistem yang sedang berjalan. Serta pengujian subjektif berupa *Mean* *Opinion* *Score* (MOS) untuk menguji kelayakan sistem yang telah dibuat.

### Implementasi

Setelah pengujian sistem dilakukan, selanjutnya pengimplementasian sistem dalam kehidupan sehari-hari dapat dilakukan. Tahap ini merupakan tahap terakhir dari metode *prototype*.

# BAB IV

**HASIL DAN PEMBAHASAN**

## Evaluasi Prototype

Pada tahap ini, baik basis data maupun *user* *interface* aplikasi dievaluasi oleh pemilik toko untuk informasi toko dan karyawan toko untuk informasi proses jual-beli pada toko yang dilanjutkan dengan konsultasi bersama pakar pada bagian menu QnA. Pada evaluasi dengan pemilik toko, dilakukan baik dengan tatap muka (konsultasi wawancara langsung) maupun via *chatting* *online* (melalui aplikasi *chatting* WhatsApp). Intensitas tatap muka dengan karyawan toko dilakukan sebanyak lima kali dalam waktu kurang lebih tiga bulan. Sedangkan untuk wawancara secara *online* dilakukan sebanyak enam kali bersama pemilik toko dan lima kali bersama karyawan toko, masing-masing dalam kurun waktu tiga bulan. Sedangkan, untuk evaluasi fitur konsultasi berbentuk QnA yang dilakukan bersama pakar dilakukan via *chatting* *online* (melalui aplikasi *chatting* WhatsApp) dengan intensitas sebanyak delapan kali dalam kurun waktu tiga bulan. Pada evaluasi yang telah dilakukan, seluruhnya mengacu pada tujuan penelitian yaitu:

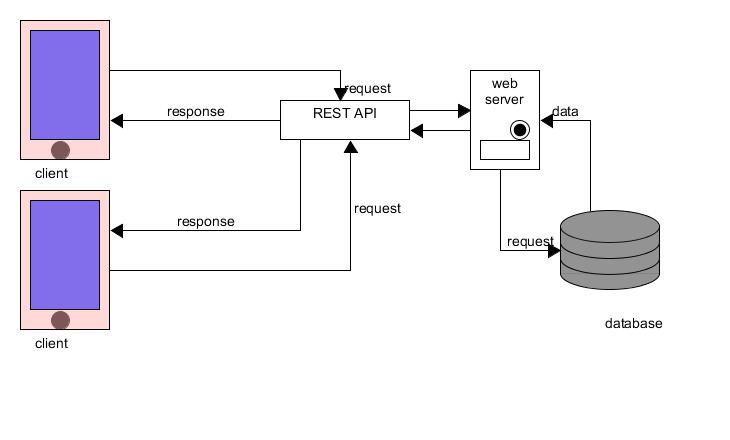
1. Kemampuan aplikasi dalam menyajikan rancangan aplikasi penjualan untuk mempermudah pihak toko dan pembeli yang bertujuan memberi kemudahan menyimpan, pengolahan dan kecepatan dalam aktivitas jual beli dan in*form*asi barang dagangan. Indikator evaluasi: kemampuan aplikasi dalam menyediakan fitur-fitur yang fungsional, jelas, dan mudah dipahami pengguna. Evaluasi yang dilakukan:
2. Penambahan terhadap data harga pokok barang untuk mempermudah penjual dalam menghitung laba.
3. Penambahan biaya ongkir, seperti berikut : sekitar Kecamatan Mataram gratis, di luar Kecamatan Mataram Rp 10.000, di luar Kabupaten Mataram pembayaran dilakukan antar pelangan dan kurir ojek online.
4. Penambahan terhadap fitur aplikasi berupa fitur pengingat jumlah stok barang yang menipis.
5. Statistik yang digunakan pada aplikasi merupakan statistik yang menunjukkan jumlah pembelian terbanyak selama satu bulan untuk kebutuhan toko dalam mempersiapkan suplai barang dan hamster.
6. Kemampuan aplikasi dalam menyajikan kemungkinan penyakit yang menyerang seekor hamster berdasarkan gejala dominan yang dimasukkan oleh pengguna. indikator evaluasi: penetapan gejala dominan yang jelas dan sesuai dengan nilai keyakinan pakar yang telah didapatkan sebelumnya, serta penetapan gejala-gejala pendukung yang sesuai. Evaluasi yang dilakukan:
7. Penyederhanaan tampilan fitur Q&A agar mudah dipahami.
8. Penyederhanaan kosa kata gejala agar dapat dimengerti oleh pengguna.

## Pembuatan Aplikasi

Setelah lolos evaluasi *prototype*, selanjutnya pembuatan aplikasi dapat mulai dilakukan. Pada tahap ini, akan dilakukan pembuatan aplikasi secara utuh agar aplikasi dapat digunakan pada segi fungsionalitasnya. Beberapa hal yang akan dibahas pada tahap ini, antara lain: pembuatan *database*, pembuatan API, dan membangun program secara menyeluruh.

### Arsitektur aplikasi

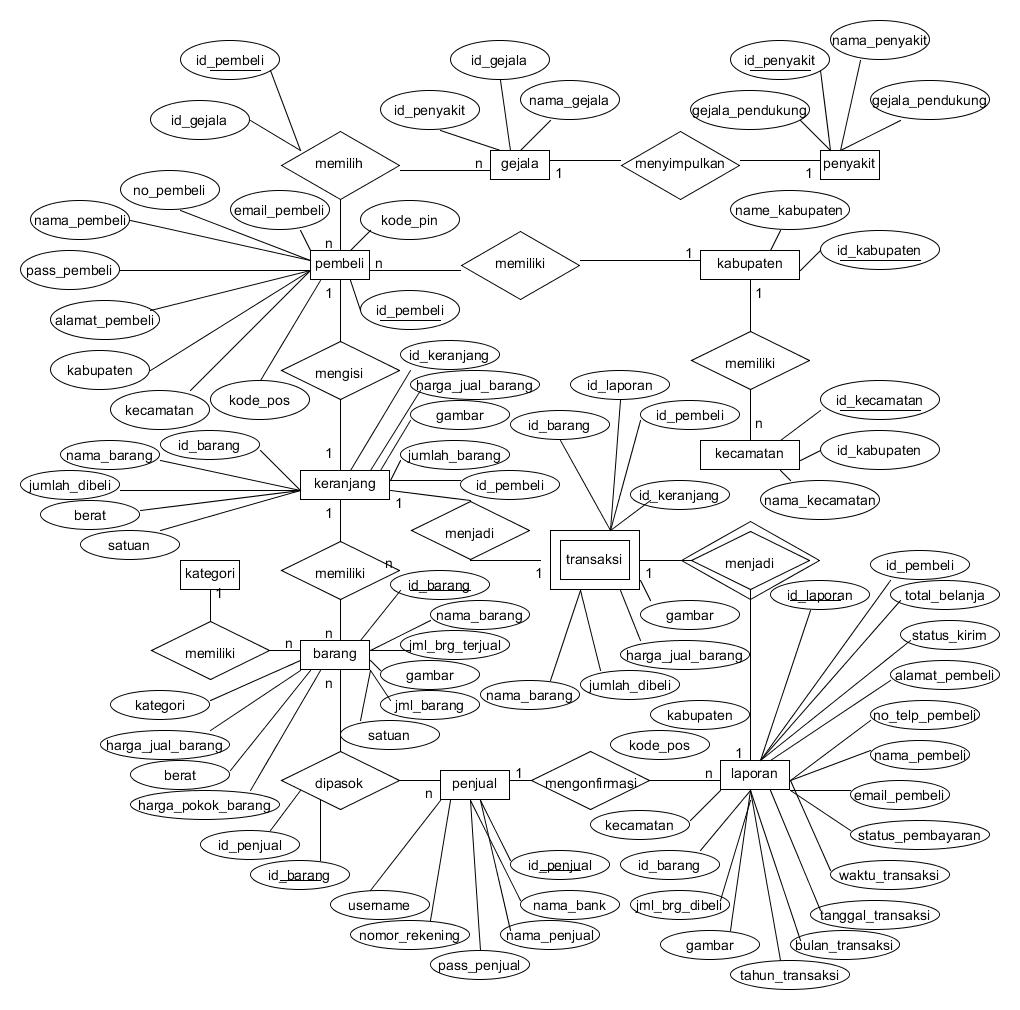
Pada observasi yang dilakukan, penulis mengamati secara langsung bagaimana proses transaksi yang dilakukan pada toko Lombok Rabbit Pet House. Setelah dilakukan observasi dapat ketahui arsitektur sistem yang akan dibangun seperti pada **Gambar** **4.1** berikut :



Gambar 4. Arsitektur aplikasi

Seperti yang dapat dilihat pada **Gambar 4. 1**, data yang akan dimasukkan ke dalam sistem adalah data barang, gejala maupun identitas pengguna. Kemudian diolah di dalam sistem dengan bantuan internet untuk menghasilkan *output* berupa data transaksi, kesimpulan penyakit maupun data diri pengguna.

### Database



Gambar 4. . ERD yang telah diperbarui pada aplikasi

**Gambar 4. 2** merupakan gambar ERD pada aplikasi setelah dilakukan evaluasi secara berkala. Pada pengembangan database aplikasi penjualan hamster dan kelengkapannya untuk toko Lombok Rabbit Pethouse, digunakan 11 tabel berupa: tabel barang, tabel gejala, tabel kabupaten, tabel kategori, tabel kecamatan, tabel keranjang, tabel laporan, tabel pembeli, tabel penjual, tabel penyakit, dan tabel transaksi.

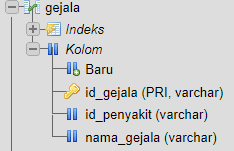
1. Struktur tabel
2. Barang



Gambar 4. Tabel barang

Pada Tabel barang terdapat 9 atribut seperti yang dapat dilihat pada **Gambar 4.2**. Tabel ini berfungsi untuk menampung data barang yang dimasukkan oleh penjual, juga digunakan untuk menampung jumlah barang yang dibeli oleh pembeli secara temporer atau sementara sebelum dipindah ke tabel keranjang.

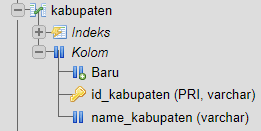
1. Gejala



Gambar 4. Tabel gejala

Pada Tabel gejala terdapat 4 atribut seperti yang dapat dilihat pada **Gambar 4.3**. Tabel ini berfungsi untuk menampung gejala-gejala dominan yang digunakan untuk menyimpulkan penyakit hamster yang mungkin terjadi.

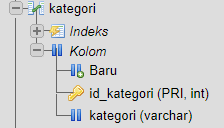
1. Kabupaten



Gambar 4. Tabel kabupaten

Pada Tabel kabupaten terdapat 2 atribut seperti yang dapat dilihat pada **Gambar 4.4**. Tabel ini berfungsi untuk menampung data nama kabupaten.

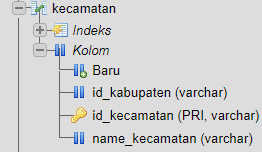
1. Kategori



Gambar 4. Tabel kategori

Pada Tabel kategori terdapat 2 atribut seperti yang dapat dilihat pada **Gambar 4.5**. Tabel ini berfungsi untuk menampung data nama kategori barang.

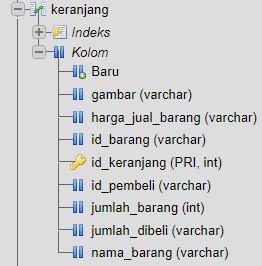
1. Kecamatan



Gambar 4. Tabel kecamatan

Pada Tabel barang terdapat 3 atribut seperti yang dapat dilihat pada **Gambar 4.6**. Tabel ini berfungsi untuk menampung data nama kecamatan.

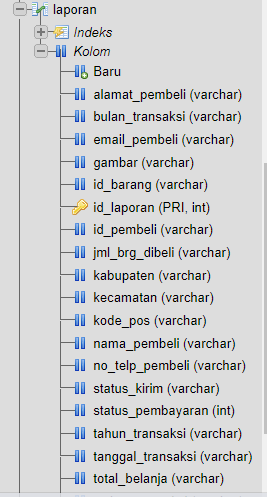
1. Keranjang



Gambar 4. Tabel keranjang

Pada Tabel keranjang terdapat 8 atribut seperti yang dapat dilihat pada **Gambar 4.7**. Tabel ini berfungsi untuk menampung data barang yang telah dipilih oleh pembeli dan tersimpan pada Tabel keranjang untuk dilakukan perhitungan total belanja.

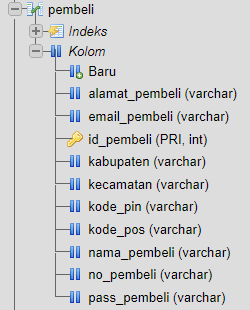
1. Laporan



Gambar 4. Tabel laporan

Pada Tabel laporan terdapat 19 atribut seperti yang dapat dilihat pada **Gambar 4.8**. Tabel ini berfungsi untuk menampung detail data transaksi yang dilakukan oleh pembeli agar pengiriman dapat dilakukan secara terkontrol.

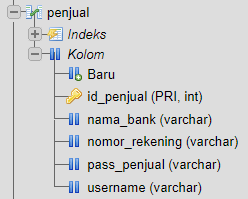
1. Pembeli



Gambar 4. Tabel pembeli

Pada Tabel pembeli terdapat 10 atribut seperti yang dapat dilihat pada **Gambar 4.9**. Tabel ini berfungsi untuk menampung data para pengguna aplikasi dari sisi pembeli.

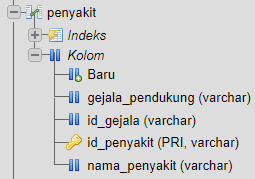
1. Penjual



Gambar 4. Tabel penjual

Pada Tabel penjual terdapat 5 atribut seperti yang dapat dilihat pada gambar **Gambar 4.10**. Tabel ini berfungsi untuk menampung data pengguna aplikasi dari sisi penjual.

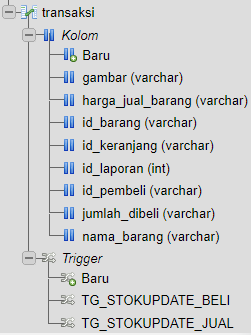
1. Penyakit



Gambar 4. Tabel penyakit

Pada Tabel penyakit terdapat 4 atribut seperti yang dapat dilihat pada **Gambar 4.11**. Tabel ini berfungsi untuk menampung data nama penyakit dan gejala pendukungnya.

1. Transaksi



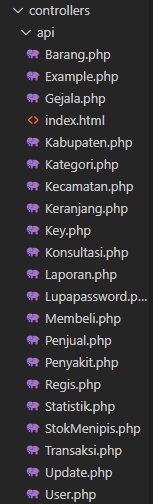
Gambar 4. Tabel Transaksi

Pada Tabel barang terdapat 9 atribut dan 2 *trigger* seperti yang dapat dilihat pada **Gambar 4.13**. Tabel ini berfungsi untuk menampung data transaksi secara mendetail (per barang). *Trigger* pada Tabel ini digunakan untuk dilakukannya *update* stok barang dan jumlah barang yang terjual pada Tabel barang.

### Pembuatan API

Pada penggunaannya, API pada aplikasi ini dibuat menggunakan *platform* CodeIgniter. Fungsi dari pembuatan API ini sendiri adalah untuk mempermudah dalam melakukan pengolahan dan manipulasi data. Berikut merupakan *controllers* dan *models* yang digunakan dalam pembuatan API aplikasi.

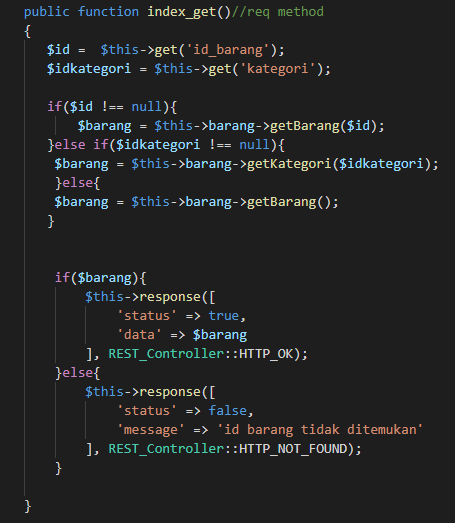
1. *Controllers*



Gambar 4. *Controllers* yang digunakan pada CI

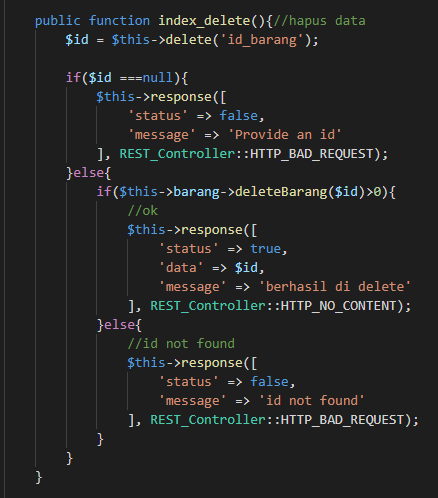
Pada *controllers* yang digunakan (**Gambar 4.14)**, terdapat fungsi-fungsi untuk memproses suatu data dengan label “index\_get()”, “index\_post()”, “index\_put()”, dan “index\_delete()” yang masing-masing fungsinya yaitu untuk mengambil, menambah, mengubah dan menghapus data . Dalam pembuatan API pada aplikasi ini, *controllers* sendiri digunakan sebagai penghubung antara *models* dan view yang akan ditampilkan pada android studio. Berikut ini merupakan contoh *controllers* yang digunakan pada API Aplikasi Penjualan Hamster dan Kelengkapannya:

1. Barang.php



Gambar 4. Fungsi “index\_get()” pada *controller* Barang.php

Fungsi “index\_get()” (**Gambar 4.15**) pada *controller* Barang.php digunakan untuk menghubungkan fungsi “getBarang()” dan “getKategori” dengan fungsi yang ada pada *model* Barang\_model.php. Pada fungsi ini, dilakukan pengambilan data sesuai kebutuhan pengguna, yaitu secara keseluruhan atau berdasarkan kategori yang diinginkan.



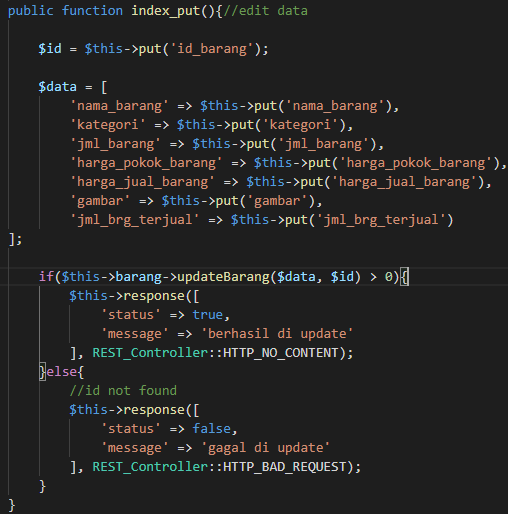
Gambar 4. Fungsi “index\_delete()” pada *controller* Barang.php

Fungsi “index\_delete()” (**Gambar 4.16**) pada *controller* Barang.php digunakan untuk menghubungkan fungsi “deleteBarang()”dengan fungsi yang ada pada *model* Barang\_model.php. Pada fungsi ini, dilakukan penghapusan data sesuai keinginan pengguna.

|  |  |
| --- | --- |
|  |  |

Gambar 4. Fungsi “index\_post()” pada *controller* Barang.php

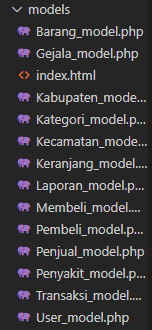
Fungsi “index\_post()” (**Gambar 4. 17**) pada *controller* Barang.php digunakan untuk menghubungkan fungsi “createBarang()”dengan fungsi yang ada pada *model* Barang\_model.php. Pada fungsi ini, dilakukan penambahan data sesuai masukan yang dilakukan pengguna.



Gambar 4. Fungsi “index\_put()” pada *controller* Barang.php

Fungsi “index\_put()” pada *controller* Barang.php (**Gambar 4.18**) digunakan untuk menghubungkan fungsi “updateBarang()” dengan fungsi yang ada pada *model* Barang\_model.php. Pada fungsi ini, dilakukan pembaharuan data berdasarkan pengubahan yang dilakukan pengguna.

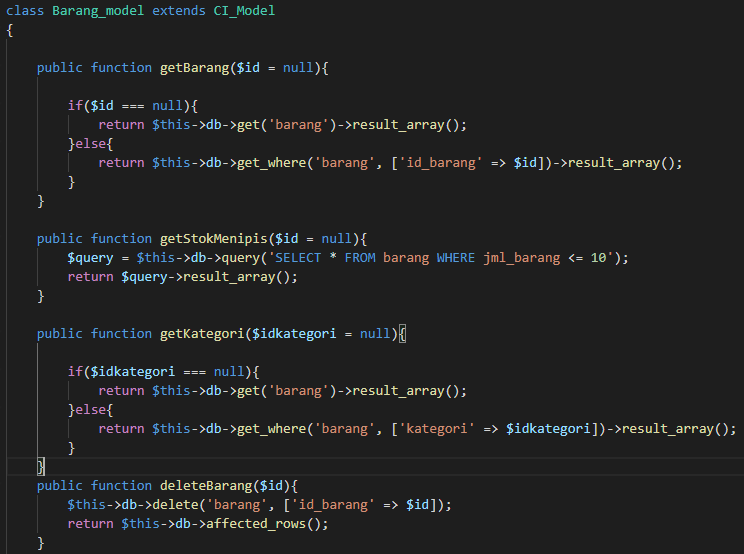
1. *Models*



Gambar 4. *Models* yang digunakan pada CI

Pada *models* yang dibuat untuk API Aplikasi Penjualan Hamster dan Kelengkapannya (**Gambar 4.19**), digunakan beberapa fungsi berlabel create, update, delete, dan get untuk menambah, mengubah, menghapus, dan mengambil data pada Tabel di *database*. Fungsi-fungsi yang dibuat bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisir data dari *database* sesuai dengan instruksi dari *controller* yang telah dibuat.

1. Barang\_model.php



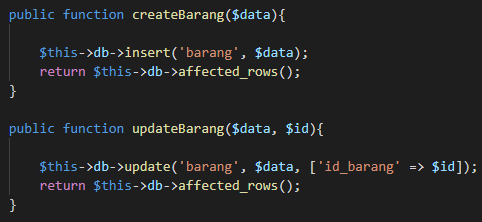
Gambar 4. *Models* “get()” dan “delete()” pada *model* Barang\_model.php

Fungsi “getBarang()” pada *model* Barang\_model.php (**Gambar 4.20**) digunakan untuk menampilkan data barang pada Tabel Barang apabila fungsi “index\_get()” pada *controller* Barang.php dijalankan. Pada fungsi ini dijalankan kondisi pengambilan data barang sesuai kebutuhan, yaitu secara keseluruhan maupun berdasarkan id barang.

Fungsi “getStokMenipis()” pada *model* Barang\_model.php (**Gambar 4.20**) digunakan untuk menampilkan data barang pada Tabel Barang dengan indicator barang memiliki jumlah kurang dari 10pcs apabila fungsi “index\_get()” pada *controller* Barang.php dijalankan.

Fungsi “getKategori()” pada *model* Barang\_model.php (**Gambar 4.20**) digunakan untuk menampilkan data barang pada Tabel Barang sesuai kategori yang diinginkan pengguna apabila fungsi “index\_get()” pada *controller* Barang.php dijalankan. Pada fungsi ini dijalankan kondisi pengambilan data barang sesuai kebutuhan, yaitu secara keseluruhan maupun berdasarkan id kategori.

Fungsi “deleteBarang()” pada *model* Barang\_model.php (**Gambar 4.20**) digunakan untuk menghapus data barang pada Tabel Barang berdasarkan keinginan pengguna apabila fungsi “index\_delete()” pada *controller* Barang.php dijalankan. Pada fungsi ini barang akan dihapus berdasarkan id barang tersebut.



Gambar 4. *Models* “create()” dan “update()” pada *model* Barang\_model.php

Selanjutnya, fungsi “createBarang()” pada *model* Barang\_model.php (**Gambar 4.21**) digunakan untuk menambah data barang pada Tabel Barang apabila fungsi “index\_post()” pada *controller* Barang.php dijalankan. Pada fungsi ini data akan ditambahkan sesuai masukan dari pengguna.

Dan yang terakhir, fungsi “updateBarang()” pada *model* Barang\_model.php (**Gambar 4.21**) digunakan untuk mengubah data barang pada Tabel Barang sesuai id barang yang diinginkan pengguna apabila fungsi “index\_put()” pada *controller* Barang.php dijalankan.

### Pembuatan proyek aplikasi

Berikut ini merupakan tampilan dari aplikasi yang telah dibuat. Tampilan dibagi menjadi dua bagian, yaitu penjual dan pembeli.

#### Android Project

##### Pakage Activity

*Package* Activity merupakan sebuah kelas yang nantinya terhubung ke Layout (XML) atau tampilan dari aplikasi. Activity ini nantinya tempat dilakukan koding untuk membuat fungsi-fungsi yang akan ditampilkan pada sebuah halaman.

1. **Penjual**

|  |
| --- |
|  |

Gambar 4. *Package* activity yang digunakan pada aplikasi penjual

Pada **Gambar 4.22** merupakan daftar activity yang digunakan pada penjual. Berikut penjelasan dari masing-masing fungsi activity tersebut:

1. *Class* beranda\_admin untuk menampilkan beranda pada penjual.
2. *Class* ChartActivity untuk menampilkan grafik laporan bulanan.
3. *Class* detail\_barang untuk menampilkan data detail dari barang berdasarkan id yang dipilih.
4. *Class* DetailPesanan untuk menampilkan data detail dari pesanan berdasarkan id yang dipilih.
5. *Class* GantiSandi untuk menampilkan halaman yang menampilkan fitur untuk mengganti kata sandi penjual.
6. *Class* hamster\_admin untuk menampilkan data barang dengan kategori hamster pada halaman penjual.
7. *Class* kelengkapan\_admin untuk menampilkan data barang dengan kategori kelengkapan hamster pada halaman penjual.
8. *Class* LaporanPesananActivity untuk menampilkan data seluruh laporan barang pesanan.
9. *Class* login\_admin untuk melakukan fungsi login pada aplikasi penjual.
10. *Class* pakan\_hamster untuk menampilkan data barang dengan kategori pakan hamster pada halaman penjual.
11. *Class* ProfilPenjual untuk menampilkan halaman-halaman dengan fungsi ubah kata sandi dan profil penjual.
12. *Class* SplashPenjual untuk menampilkan logo aplikasi pada halaman splash.
13. *Class* StokMenipis untuk menampilkan daftar barang dengan jumlah stok kurang dari 10.
14. *Class* tambah\_barang untuk melakukan fungsi post atau penambahan data barang dagangan.
15. *Class* UbahProfil untuk melakukan fungsi pengubahan data pribadi penjual yang berkaitan dengan proses transaksi.
16. **Pembeli**

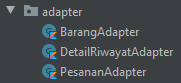
|  |
| --- |
|  |

Gambar 4. *Package* activity pada aplikasi pembeli

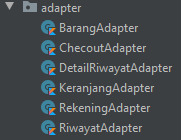
Pada gambar **Gambar 4.23** merupakan daftar activity yang digunakan pada penjual. Berikut penjelasan dari masing-masing fungsi activity tersebut:

1. *Class* CheckoutActivity untuk melakukan fungsi penyimpanan data transaksi serta pengiriman.
2. *Class* DaftarActivity untuk melakukan fungsi penambahan akun pembeli baru.
3. *Class* DetailRiwayat untuk menampilkan data pesanan yang pernah dibuat oleh pembeli.
4. *Class* FragmentTab untuk menampung fragment-fragment yang digunakan dalam pembuatan tab.
5. *Class* GantiPassword untuk melakukan fungsi pengubahan kata sandi pembeli.
6. *Class* InfoCheckout untuk memanggil daftar informasi terkait pembayaran pesanan.
7. *Class* KeranjangActivity untuk melakukan fungsi penyimpanan data barang yang akan dibeli serta estimasi total biaya.
8. *Class* KonsultasiFragment untuk melakukan fungsi penyimpulan penyakit hamster berdasarkan gejala yang dipilih.
9. *Class* LoginActivity untuk melakukan fungsi login pada penjual.
10. *Class* MainActivity untuk menghubungkan fragment-fragment yang digunakan.
11. *Class* ProfilActivity untuk menampilkan data diri pembeli yang dapat diedit.
12. *Class* ProfileFragment untuk menampilkan halaman profil penjual yang di dalamnya terdapat menu-menu yang berfungsi untuk mengubah data pribadi pembeli.
13. *Class* RiwayatBelanja untuk menampilkan daftar pesanan yang pernah dilakukan pembeli.
14. *Class* VerifikasiKodeActivity untuk melakukan fungsi fungsi verifikasi kode pin pada saat pembeli ingin mengubah kata sandi-nya.
15. *Class* DetailBarang untuk menampilkan data detail barang berdasarkan id yang dipilih.
16. *Class* HamsterFragment untuk menampilkan daftar barang dengan kategori hamster.
17. *Class* KelengkapanFragment untuk menampilkan daftar barang dengan kategori kelengkapan hamster.
18. *Class* lupa\_katasandi untuk melakukan fungsi ubah kata sandi berdasarkan email dan kode pin yang dimasukkan.
19. *Class* PakanFragment untuk menampilkan daftar barang dengan kategori pakan hamster.
20. *Class* SectionPagerAdapter untuk membuat nama-nama tab yang akan digunakan.
21. *Class* SplashPembeli untuk menampilkan logo aplikasi.
22. *Class* TentangKami untuk menampilkan informasi toko Lombok Rabbit Pet House.

##### Package Adapter



Gambar 4. *Package*adapter penjual



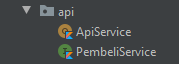
Gambar 4. *Package*adapter pembeli

*Package* adapter (**Gambar 4.24 dan Gambar 4.25**) merupakan kelas yang menghubungkan antara data yang ada pada database ke sebuah listview/recycleview pada aplikasi. Contohnya, seperti BarangAdapter yang akan mengambil data dari Tabel barang, DetailRiwayatAdapter yang akan mengambil data dari Tabel transaksi, PesananAdapter yang akan mengambil data dari Tabel laporan, dan seterusnya. Pada aplikasi ini data yang akan ditampilkan berupa arraylist.

##### Package API



Gambar 4. Package API penjual

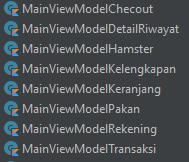


Gambar 4. Package API pembeli

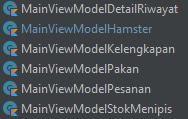
*Package* API (**Gambar 4.26 dan Gambar 4.27**) merupakan cara yang digunakan untuk terhubung dengan API yang sudah dibuat sebelumnya. Package ini nantinya terbagi menjadi 2 yaitu AdminService pada **Gambar 4.26** atau PembeliSevice pada **Gambar 4.27** dan ApiService, dimana ApiService merupakan kelas utama yang nantinya menghubungkan aplikasi dengan API. Dimana pada kelas ini juga mengimplementasikan Gson untuk menjadi data yang semulanya Array menjadi Gson. Kemudian pada kelas ini juga menghubungkan aplikasi dengan Base url dari API yang digunakan. Kemudian, kelas ini juga terhubung dengan kelas *interface* PembeliService atau AdminService.

Sedangkan, AdminService atau PembeliService merupakan kelas yang mengimplementasi endpoint dari base url yang telah dibuat pada API. Pada kelas ini terdapat fungsi GET, POST, dan DELETE. Dimana disetiap fungsi yang dibuat mengimplementasikan kelas Response pada package Model.

##### Package Main view model



Gambar 4. Main view model penjual



Gambar 4. Main view model pembeli

*Package* MainViewModel (**Gambar 4.28 dan Gambar 4.29**) merupakan folder berisi kelas-kelas yang digunakan untuk mengambil data pada *database* dalam bentuk ArrayList. Kelas ini mengimplementasikan Kelas-kelas Model dan Response untuk terhubung dengan API yang digunakan, seperti MainViewModelHamster yang mengimplementasi kelas model Barang dan BarangResponse untuk mengambil data barang berdasarkan kategori hamster. Terdapat fungsi set yang berguna untuk mengambil semua data yang ada pada API, serta fungsi get yang berguna untuk melakukan return dari data yang sudah diambil dari ArrayList.

##### Session Manager



Gambar 4. Session manager

SessionManager (**Gambar 4.30**) digunakan untuk menyimpan data pengguna (pembeli maupun penjual) ketika berhasil melakukan login seperti id, *email*, kata sandi, dan status dari pengguna apakah sedang *login* atau tidak. Jika pengguna sudah *login* dan keluar dari aplikasi, maka aplikasi akan menyimpan data dari pengguna, sehingga tidak perlu melakukan *login* lagi. Pada kelas ini juga terdapat fungsi clearsession untuk menghapus semua data pengguna pada saat melakukan *logout* sehingga tidak dapat masuk ke dalam halaman beranda admin jika belum melakukan *login*.

##### Package Model

|  |
| --- |
|  |

Gambar 4. *Package* Model pembeli

|  |
| --- |
|  |

Gambar 4. *Package* Model penjual

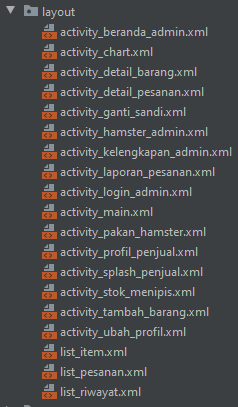
Kelas-kelas model pada pembuatan aplikasi ini (**Gambar 4.31 dan Gambar 4.32**) merupakan sebuah *data* *class* yang berguna untuk mengimplementasikan atribut-atribut yang ada pada Tabel. Contohnya pada model Kategori maka yang akan diakses adalah id\_kategori dan nama\_kategori. Kelas-kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

Kemudian pada kelas-kelas *response*, dapat diartikan bahwa kelas-kelas *response* merupakan sebuah *data* *class* yang terhubung dengan kelas-kelas model dengan nama yang sesuai, seperti kelas model Barang dan BarangResponse. Kelas *response* berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API-API tertentu, jika statusnya *true* maka akan menampilkan data yang dipanggil dalam bentuk array, namun jika *false* akan menampilkan pesan *error* di Array API tersebut.

#### *Package* Layout (Xml)

|  |
| --- |
|  |

Gambar 4. *Package* Layout pembeli



Gambar 4. *Package* Layout penjual

*Layout* yang ditunjukkan pada **Gambar 4.33 dan Gambar 4.34** merupakan sebuah *package* yang berguna untuk tempat membuat sebuah *interface* yang nantinya akan dilihat oleh pengguna. Salah satu pengimplementasian kode *interface* tersebut dapat dilihat pada lampiran dari laporan ini. Pada kode XML tersebut, tampilan yang dihasilkan adalah berupa formulir yang dapat menerima input dari pengguna (pembeli) berupa email, kode pin, dan kata sandi baru. Kemudian disediakan *button*/tombol untuk melakukan aksi selanjutnya apabila kolom-kolom sebelumnya sudah terisi.

##### Pembeli

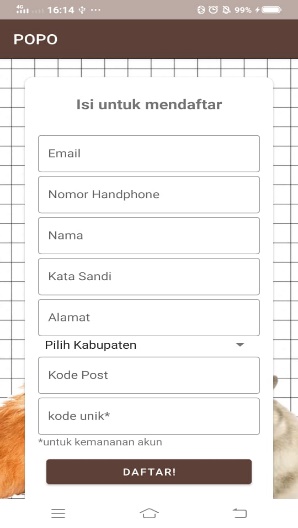
1. ***Login***



Gambar 4. Tampilan*login*

Halaman *login* seperti pada **Gambar 4.35** digunakan untuk dapat masuk ke sistem dengan memasukkan *email* dan kata sandi jika sudah memiliki akun, Sedangkan jika belum maka harus melakukan pendaftaran untuk menjadi member dengan memilih TextView “Belum punya akun?”. Dan jika pembeli lupa kata sandi-nya, maka harus dapat memilih TextView “lupa kata sandi?”

1. **Daftar**



Gambar 4. Tampilan *form* daftar

**Gambar 4.36** merupakan tampilan *form* pendaftaran pada pembeli. Pada tampilan ini, digunakan material seperti Button, CardView, TextInputLayout, dan lain-lain untuk menyusun tampilannya. Pembeli dapat memasukkan data dirinya dan mengkonfirmasi dengan mengklik tombol daftar.

1. **Lupa kata sandi**



Gambar 4. Tampilan lupa kata sandi

**Gambar 4.37** merupakan tampilan *form* lupa kata sandi pada pembeli. Pada tampilan ini, digunakan material seperti button, cardview, TextInputLayout, dan lain-lain untuk menyusun tampilannya. Pembeli dapat mengisi email, kode pin, dan kata sandi baru-nya kemudian mengklik tombol selanjutnya untuk mengganti kata sandinya.

1. **Beranda**



Gambar 4. Tampilan beranda pembeli

**Gambar 4.38** merupakan tampilan beranda pada pembeli. Pada tampilan ini, digunakan material seperti Fragment, CardView, RicycleView, dan melakukan penambahan fitur seperti pencarian, menu atas dan menu bawah, dan lain-lain untuk menyusun tampilannya.

1. **Konsultasi**

|  |
| --- |
|  |

Gambar 4. Tampilan konsultasi

**Gambar 4.39** merupakan tampilan halaman konsultasi pada pembeli. Pada tampilan ini, digunakan material seperti Button, Spinner, ScrollView, dan lain-lain untuk menyusun tampilannya. Pembeli dapat memilih gejala pada spinner kemudian mengklik tombol selanjutnya untuk mendapatkan hasil kesimpulan penyakit beserta gejala-gejala pengikut yang mungkin juga akan nampak apabila hamster mengidap gejala dominan yang dipilih. Gejala yang dipilih merupakan gejala yang paling tampak (dominan) pada seekor hamster. Gejala yang ditampilkan pada spinner hanya akan menyimpulkan atau mengarah kepada satu penyakit saja karena gejala-gejala tersebut bersifat sangat kontras terhadap satu sama lain, sehingga tidak mungkin suatu gejala dominan dimiliki oleh dua penyakit yang sama atau lebih.

1. **Keranjang**

|  |
| --- |
|  |

Gambar 4. Tampilan keranjang

**Gambar 4.40** merupakan tampilan halaman keranjang pada pembeli. Pada tampilan ini, digunakan material seperti Button, Spinner, ScrollView, dan lain-lain untuk menyusun tampilannya. Pembeli dapat mengatur jumlah barang yang dibeli dengan mengklik tombol “+” atau “–”, menghapus daftar barang yang akan dibeli. Selanjutnya pembeli mengatur data diri penerima pada *form* yang disediakan, kemudian menerima nomor rekening penjual untuk melakukan pembayaran via ATM. Seperti pada gambar, terdapat batas waktu pembayaran yang harus dipenuhi pembeli jika ingin melanjutkan transaksi. Pembeli dapat melakukan konfirmasi pembayaran sebelum batas pembayaran yang diberikan habis atau berakhir. Serta terdapat tombol info yang mengarah pada halaman “Tentang Kami” yang menyimpan alamat dan nomor WhatsApp toko.

1. **Riwayat belanja**

|  |
| --- |
|  |

Gambar 4. Tampilan riwayat belanja

**Gambar 4.41** merupakan tampilan halaman tentang kami pada pembeli. Pada tampilan ini, digunakan material seperti Button, CardView, LinearLayout, dan lain-lain untuk menyusun tampilannya. Di sini pembeli dapat melakukan konfirmasi jika sudah melakukan pembayaran dan sekaligus melihat status pengiriman terhadap pesanan yang telah dilakukan dan melihat detail pesanannya. Konfirmasi yang dimaksud merupakan konfirmasi pembayaran yang harus dilakukan sebelum batas pembayaran yang telah diberikan.

1. **Profil**

|  |
| --- |
|  |

Gambar 4. Tampilan profil pembeli

**Gambar 4.42** merupakan tampilan halaman profil pada pembeli. Pada tampilan ini, digunakan material seperti Button, CardView, LinearLayout, dan lain-lain untuk menyusun tampilannya. Di sini pembelu dapat melakukan pengubahan data dirinya serta melakukan *logout*.

1. **Detail barang**



Gambar 4. Tampilan detail barang

**Gambar 4.43** merupakan tampilan halaman keranjang pada pembeli. Pada tampilan ini, digunakan material seperti Button, ImageView, TextInputLayout, TextView, EditText dan lain-lain untuk menyusun tampilannya. Pada halaman ini pembeli bisa menentukan berapa jumlah barang yang ingin dibeli dan memindahkannya pada halaman keranjang dengan mengklik tombok beli.

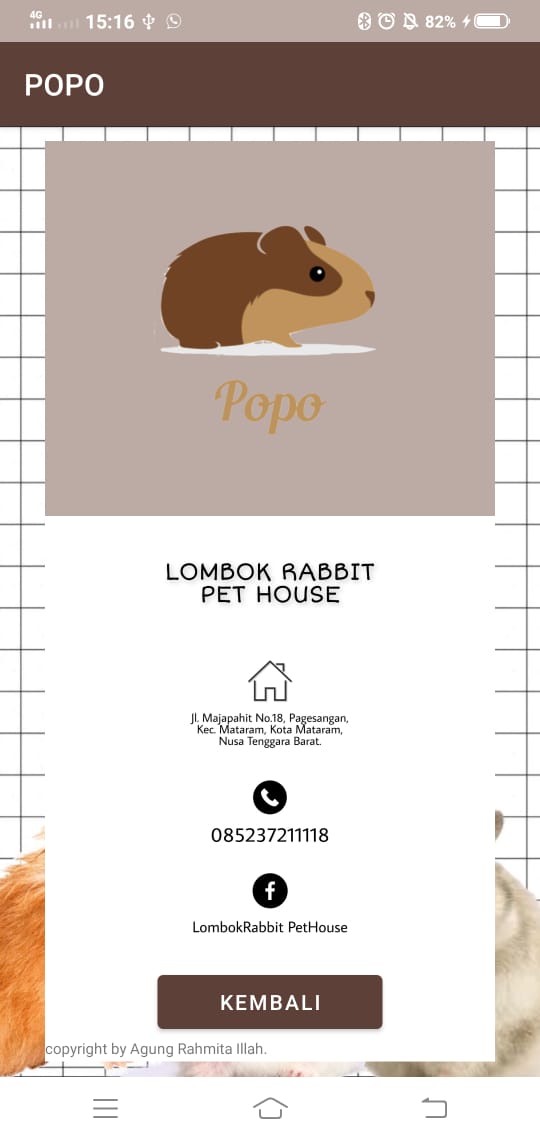
1. **Ubah kata sandi**

|  |
| --- |
|  |

Gambar 4. Tampilan ubah kata sandi

**Gambar 4.44** merupakan tampilan ubah kata sandi pada pembeli. Pada tampilan ini, digunakan material seperti Button, PinEntryEditText, LinearLayout, TextInputEditText dan lain-lain untuk menyusun tampilannya. Sebelum dapat mengubah kata sandi-nya, pembeli harus melakukan verifikasi kode pin terlebih dahulu.

1. **Tentang kami**

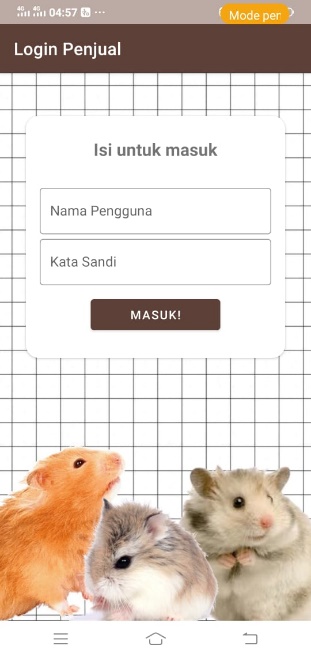


Gambar 4. Tampilan tentang kami

**Gambar 4.45** merupakan tampilan halaman tentang kami pada pembeli. Pada tampilan ini, digunakan material seperti Button, CardView, LinearLayout, dan lain-lain untuk menyusun tampilannya. Pembeli dapat melihat informasi toko seperti nomor WhatsApp toko, alamat dan nama akun Facebook toko di sini.

##### Penjual

1. ***Login***



Gambar 4. Tampilan *login* penjual

**Gambar 4.46** merupakan tampilan halaman *login* pada penjual. Pada tampilan ini, digunakan material seperti Button, TextInputLayout, CardView, dan lain-lain untuk menyusun tampilannya. Penjual dapat memasukkan email dan kata sandi-nya pada *form* yang disediakan untuk login ke aplikasi.

1. ***Splash* aplikasi**



Gambar 4. Tampilan *splash*

**Gambar 4.47** merupakan tampilan *splash* aplikasi baik pada penjual maupun pembeli. Pada tampilan ini, digunakan material seperti ImageView dan LinearLayout untuk menyusun tampilannya. Ini merupakan tampilan awal yang akan muncul pada saat aplikasi pertama kali dibuka.

1. **Beranda**

|  |
| --- |
|  |

Gambar 4. Tampilan beranda penjual

**Gambar 4.48** merupakan tampilan halaman beranda pada penjual. Pada tampilan ini, digunakan material seperti Button, ImageView, menu, TextView dan lain-lain untuk menyusun tampilannya. Ini merupakan halaman utama pada aplikasi penjual.

1. **Laporan pesanan**

|  |
| --- |
|  |

Gambar 4. Tampilan laporan pesanan

**Gambar 4.49** merupakan tampilan halaman laporan pesanan pada penjual. Pada tampilan ini, digunakan material seperti Button, ImageView, ScrolView, TextView dan lain-lain untuk menyusun tampilannya. Di sini penjual bisa melihat detail pesanan seperti identitas pembeli dan alamat pembeli. Kemudian penjual juga dapat melihat barang apa saja yang telah dibeli oleh pembeli serta melakukan konfirmasi pengiriman.

1. **Laporan bulanan**

|  |
| --- |
|  |

Gambar 4. Tampilan laporan bulanan

**Gambar 4.50** merupakan tampilan halaman laporan bulanan pada penjual. Pada tampilan ini, digunakan material seperti Button, Spinner, BarChart, TextView dan lain-lain untuk menyusun tampilannya. Di sini penjual dapat memilih statistik jumlah pembelian barang tertinggi per bulan yang ingin dilihat. Statistik akan tampak setelah penjual memilih bulan dan tahun pada spinner yang telah disediakan, kemudian mengklik tombol “cek”. Gambar statistic dapat diperbesar dengan mengusap dua jari dengan arah yang berlawanan.

1. **Profil**

|  |
| --- |
|  |

Gambar 4. Tampilan profil penjual

**Gambar 4.51** merupakan tampilan halaman profil pada penjual. Pada tampilan ini, digunakan material seperti Button, TextInputLayout, CardView, TextView dan lain-lain untuk menyusun tampilannya. Di sini penjual dapat melakukan pengubahan data pribadinya.

1. **Detail barang**

|  |
| --- |
|  |

Gambar 4. Tampilan detail barang penjual

**Gambar 4.52** merupakan tampilan halaman detail barang pada penjual. Pada tampilan ini, digunakan material seperti Button, Spinner, ImageView, TextInputLayout dan lain-lain untuk menyusun tampilannya. Di sini penjual dapat melakukan pengubahan dan penghapusan data. Pengisian data dilakukan dengan mengisi nama barang, memilih kategori melalui spinner yang diberikan, kemudian mengisi data-data lainnya. Pemilihan gambar dilakukan dengan mengklik “pilih gambar”, kemudian sistem akan mengarahkan penjual untuk memilih folder penyimpan foto. Setelah foto yang diinginkan dipilih, penjual dapat mengklik tombol “perbarui” untuk memperbarui data.

1. **Laporan stok menipis**

|  |
| --- |
|  |

Gambar 4. Tampilan laporan stok menipis

**Gambar 4.53** merupakan tampilan halaman stok menipis pada penjual. Pada tampilan ini, digunakan material seperti Button, Spinner, CardView, ImageView dan lain-lain untuk menyusun tampilannya. Di sini penjual dapat melihat barang apa saja yang jumlahnya kurang dari atau sama dengan 10pcs, serta dapat melakukan pengubahan stok barang secara langsung.

1. **Tambah barang**

|  |
| --- |
|  |

Gambar 4. Tampilan tambah barang

**Gambar 4.54** merupakan tampilan halaman tambah barang pada penjual. Pada tampilan ini, digunakan material seperti Button, Spinner, TextInputLayout, ImageView dan lain-lain untuk menyusun tampilannya. Di sini penjual dapat melakukan penambahan barang sesuai kategori yang diinginkan. Pengisian data dilakukan dengan mengisi nama barang, memilih kategori melalui spinner yang diberikan, kemudian mengisi data-data lainnya. Pemilihan gambar dilakukan dengan mengklik “pilih gambar”, kemudian sistem akan mengarahkan penjual untuk memilih folder penyimpan foto. Setelah foto yang diinginkan dipilih, penjual dapat mengklik tombol “simpan” untuk menyimpan data.

## Pengujian Aplikasi

Pengujian pada aplikasi ini dilakukan dengan menggunakan metode *blackbox* testing dan MOS. Digunakannya metode *blackbox* testing yaitu dengan menguji masing-masing fungsi yang sudah dibuat dalam aplikasi ini. Hal tersebut bermaksud untuk menemukan kesalahan atau bug pada sistem yang sudah dibangun sehingga pada saat aplikasi ini digunakan nantinya dapat berjalan sesuai dengan apa yang diinginkan dan diharapkan oleh pengguna. Pengujian *blackbox* testing sendiri adalah pengujian langsung pada fungsi-fungsi aplikasi tanpa memperhatikan alur eksekusi program. Pengujian ini dilakukan dengan memperhatikan apakah fungsi yang dibuat sudah sesuai atau tidak dengan yang diharapkan.

Selanjutnya, pada pengujuan metode MOS akan dilakukan penyebaran kuesioner kepada 30 responden yang terdiri dari pemilik dan seorang karyawan toko Lombok Rabbit Pet House, dan 28 orang calon pembeli toko Lombok Rabbit Pet House untuk melakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan dengan menunjukkan sistem kepada responden untuk dijalankan. Setelah responden menjalankan sistem, diberikan kuesioner yang berisi 6 pertanyaan dengan 5 pilihan jawaban. Tujuan dari pengujian ini untuk mengetahui kualitas sistem yang dilihat dari sisi pengguna. Metode yang digunakan dalam pengujian kuesioner adalah metode kuantitatif, yaitu hasil pengujian ditentukan dalam suatu nilai angka.

### Hasil pengujian *blackbox*

1. Fungsi pada penjual
2. Fungsi halaman laporan pesanan

Fungsi halaman laporan pesananpada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halamanlaporan pesanan penjual sudah berjalan dengan baik dan benar. Pengujian fungsi laporan pesananakan dijelaskan pada Tabel 4.1.

Tabel 4. Pengujian *Blackbox* fungsi halaman laporan pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual mengonfirmasi pengiriman | Konfirmasi berhasil dan status pengiriman berubah dari “Belum Dikirim” menjadi “Sudah Dikirim” | Sesuai | Valid |

1. Fungsi halaman laporan bulanan

Fungsi laporan bulananpada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman laporan bulananpenjual sudah berjalan dengan baik dan benar. Pengujian fungsi laporan bulanan akan dijelaskan pada Tabel 4.2.

Tabel 4. Pengujian *Blackbox* fungsi laporan bulanan

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual membuka halaman laporan bulanan | Halaman laporan bulanan menampilkan 10 barang dengan jumlah terbeli tertinggi pada bulan tersebut | Sesuai | Valid |

1. Fungsi menambah data baru

Fungsi menambah data barupada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman tambah data barang padapenjual sudah berjalan dengan baik dan benar. Pengujian fungsi menambah data baru akan dijelaskan pada Tabel 4.3.

Tabel 4. Pengujian *Blackbox* fungsi menambah data baru

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Kategori dipilih sesuai kebutuhan pengguna untuk mengelompokkan data | Data barang berhasil dikelompokkan berdasarkan kategori | Sesuai | Valid |
| *Form* tambah barang diisi dengan data barang | Penambahan data berhasil kemudian sistem kembali menampilkan halaman beranda penjual | Sesuai | Valid |

1. Fungsi menghapus data barang

Fungsi menghapus data barangpada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman detail barang penjual sudah berjalan dengan baik dan benar. Pengujian fungsi menghapus data barang akan dijelaskan pada Tabel 4.4.

Tabel 4. Pengujian *Blackbox* fungsi menghapus data barang

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual menghapus data barang dari halaman detail barang | Penghapusan data berhasil lalu sistem kembali menampilkan halaman beranda penjual | Sesuai | Valid |

1. Fungsi mengubah data barang

Fungsi mengubah data barangpada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman detail barang penjual sudah berjalan dengan baik dan benar. Pengujian fungsi mengubah data barang akan dijelaskan pada Tabel 4.5.

Tabel 4. Pengujian *Blackbox* fungsi mengubah data barang

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual mengubah data barang dari halaman detail barang | Pengubahan data berhasil lalu sistem kembali menampilkan halaman beranda penjual | Sesuai | Valid |

1. Fungsi pada pembeli
2. Fungsi menambah pesanan

Fungsimenambah pesanan pembelidilakukan untuk menguji apakah fungsi menambah pesanan yang berjalan pada halamankeranjang belanja pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi menambah pesanan pembeli akan dijelaskan pada Tabel 4.6.

Tabel 4. Pengujian *Blackbox* fungsi menambah pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Pembeli memilih barang yang akan dibeli dan menentukan jumlah barang yang dibeli melalui halaman detail barang | Barang yang akan dibeli berhasil ditambahkan di keranjang belanja beserta jumlah baramg | Sesuai | Valid |

1. Fungsi keranjang belanja

Fungsikeranjang belanja pembelidilakukan untuk menguji apakah fungsi keranjang belanja yang ada sudah berjalan dengan baik dan benar. Pengujian fungsi keranjang belanja pembeli akan dijelaskan pada Tabel 4.7.

Tabel 4. Pengujian *Blackbox* fungsi keranjang belanja

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Pembeli mengatur jumlah barang yang akan dibeli dari halaman keranjang  Belanja | Jumlah barang berhasil diubah | Sesuai | Valid |
| Pembeli menghapus barang yang akan dibeli dari halaman keranjang belanja | Barang berhasil dihapus dari halaman keranjang belanja | Sesuai | Valid |

1. Fungsi riwayat belanja

Fungsiriwayat belanja pembelidilakukan untuk menguji apakah fungsi riwayat belanja yang berjalan pada riwayat belanja pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi riwayat belanja pembeli akan dijelaskan pada Tabel 4.8.

Tabel 4. Pengujian *Blackbox* fungsi riwayat belanja

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Pembeli dapat melihat detail riwayat belanja | Riwayat belanja yang diinginkan berhasil ditampilkan | Sesuai | Valid |

1. Fungsi pengisian data pengiriman barang

Fungsi pengisian data pengiriman barang dilakukan untuk menguji apakah fungsi pengisian data pengiriman barang yang ada sudah berjalan dengan baik dan benar. Pengujian fungsi pengisian data pengiriman barang akan dijelaskan pada Tabel 4.9.

Tabel 4. Pengujian *Blackbox* fungsi pengiriman barang

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| *Form* data pengiriman barang diisi dengan data yang diperlukan dan dikirim | Data pengiriman berhasil dikirim dengan status pengiriman default (belum terkirim) | Sesuai | Valid |

1. Fungsi konsultasi

Fungsi konsultasi dilakukan untuk menguji apakah fungsi konsultasi yang ada sudah berjalan dengan baik dan benar. Pengujian fungsi konsultasi akan dijelaskan pada Tabel 4.10.

Tabel 4. Pengujian *Blackbox* fungsi konsultasi

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Pengguna memilih gejala dominan yang tampak pada hamster | Sistem berhasil menampilkan kemungkinan penyakit sesuai gejala yang terjadi beserta gejala lain yang mungkin Nampak | Sesuai | Valid |

### Hasil pengujian MOS

Setelah dilakukan pengujian MOS berbentuk kuisioner, hasil jawaban yang didapatkan dihitung untuk mendapatkan kesimpulan seberapa layak sistem dapat digunakan dan dipasarkan. Terdapat dua jenis kuisioner yang akan dibagikan, yaitu kuisioner untuk penjual (pihak Lombok Rabbit Pet House) dan calon pembeli. Kuisioner yang diberikan masing-masing terdiri dari enam pertanyaan.

Berikut merupakan isi pertanyaan kedua jenis kuisioner :

1. Penjual
2. Apakah tampilan aplikasi mudah dipahami?
3. Apakah aplikasi mudah digunakan?
4. Apakah aplikasi ini dapat membantu pihak toko Lombok Rabbit Pet House dalam proses penjualan dan pelaporan transaksi?
5. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
6. Apakah aplikasi berjalan dengan baik?
7. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?
8. Pembeli
9. Apakah tampilan aplikasi mudah dipahami?
10. Apakah aplikasi mudah digunakan?
11. Apakah aplikasi ini dapat membantu calon pembeli toko Lombok Rabbit Pet House dalam proses belanja dan konsultasi mengenai penyakit hamsternya?
12. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
13. Apakah aplikasi berjalan dengan baik?
14. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

Responden diminta untuk menjawab kuesioner tersebut dengan pilihan jawaban sebagai berikut:

1 = Tidak setuju.

2 = Kurang setuju.

3 = Cukup.

4 = Setuju.

5 = Sangat setuju

Berdasarkan jawaban tersebut, dilakukan perhitungan persentase dari setiap jawaban yang diberikan oleh pengguna di setiap pertanyaannya. Rumus untuk menghitung persentase jawaban adalah sebagai berikut:

Persentase = J / N\*100%

Keterangan= J : Total nilai jawaban yang diberikan pengguna pada setiap nomor.

N : Jumlah pengguna.

Selanjutnya dilakukan proses perhitungan persentase rata-rata dari setiap poin jawaban, perhitungan yang digunakan adalah menggunakan rumus sebagai berikut:

|  |
| --- |
| Rata-rata = (Persentase ke-1 + ….. + Persentase ke-5) / 6 |

Persentase jawaban responden terhadap kuesioner yaitu, tidak setuju, kurang setuju, cukup, setuju dan sangat setuju dapat dihitung secara keseluruhan dan didefinisikan sebagai nilai rata-rata. Dengan menggunakan rumus perhitungan diatas, didapatkan perhitungan untuk hasil jawaban pengguna yaitu penjual dan pembeli.

1. Penjual

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah tampilan aplikasi mudah dipahami?”, 0% menjawab sangat setuju, 50% menjawab setuju, 50% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.11.

Tabel 4. Pengujian MOS penjual (pertanyaan 1)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 0 | 2 | 0% |
| Setuju | 1 | 2 | 50% |
| Cukup setuju | 1 | 2 | 50% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi mudah digunakan?”, 0% menjawab sangat setuju, 50% menjawab setuju, 50% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.12.

Tabel 4. Pengujian MOS penjual (pertanyaan 2)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 0 | 2 | 0% |
| Setuju | 1 | 2 | 50% |
| Cukup setuju | 1 | 2 | 50% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi ini dapat membantu pihak toko Lombok Rabbit Pet House dalam proses penjualan dan pelaporan transaksi?”, 50% menjawab sangat setuju, 0% menjawab setuju, 50% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.13.

Tabel 4. Pengujian MOS penjual (pertanyaan 3)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 1 | 2 | 50% |
| Setuju | 0 | 2 | 0% |
| Cukup setuju | 1 | 2 | 50% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?”, 0% menjawab sangat setuju, 100% menjawab setuju, 0% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.14.

Tabel 4. Pengujian MOS penjual (pertanyaan 4)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 0 | 2 | 0% |
| Setuju | 2 | 2 | 100% |
| Cukup setuju | 0 | 2 | 0% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi berjalan dengan baik?”, 50% menjawab sangat setuju, 50% menjawab setuju, 0% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.15.

Tabel 4. Pengujian MOS penjual (pertanyaan 5)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 1 | 2 | 50% |
| Setuju | 1 | 2 | 50% |
| Cukup setuju | 0 | 2 | 0% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?”, 50% menjawab sangat setuju, 0% menjawab setuju, 50% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.16.

Tabel 4. Pengujian MOS penjual (pertanyaan 6)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 1 | 2 | 50% |
| Setuju | 0 | 2 | 0% |
| Cukup setuju | 1 | 2 | 50% |
| Kurang setuju | 0 | 2 | 0% |
| Tidak setuju | 0 | 2 | 0% |

1. Pembeli

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah tampilan aplikasi mudah dipahami?”, 39.3% menjawab sangat setuju, 57.1% menjawab setuju, 3.6% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.17.

Tabel 4. Pengujian MOS pembeli (pertanyaan 1)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 16 | 28 | 39.3% |
| Setuju | 11 | 28 | 57.1% |
| Cukup setuju | 1 | 28 | 3.6% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi mudah digunakan?”, 46.4% menjawab sangat setuju, 53.6% menjawab setuju, 0% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.18.

Tabel 4. Pengujian MOS pembeli (pertanyaan 2)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 13 | 28 | 46.4% |
| Setuju | 15 | 28 | 53.6% |
| Cukup setuju | 0 | 28 | 0% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi ini dapat membantu calon pembeli toko Lombok Rabbit Pet House dalam proses belanja dan konsultasi mengenai penyakit hamsternya?”, 35.7% menjawab sangat setuju, 60.7% menjawab setuju, 3.6% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.19.

Tabel 4. Pengujian MOS pembeli (pertanyaan 3)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 10 | 28 | 35.7% |
| Setuju | 17 | 28 | 60.7% |
| Cukup setuju | 1 | 28 | 3.6% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?”, 21.4% menjawab sangat setuju, 71.4% menjawab setuju, 7.1% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.20.

Tabel 4. Pengujian MOS pembeli (pertanyaan 4)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 6 | 28 | 21.4% |
| Setuju | 20 | 28 | 71.4% |
| Cukup setuju | 2 | 28 | 7.1% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah aplikasi berjalan dengan baik?”, 42.9% menjawab sangat setuju, 50% menjawab setuju, 7.1% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.21.

Tabel 4. Pengujian MOS pembeli (pertanyaan 5)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 12 | 28 | 42.9% |
| Setuju | 14 | 28 | 50% |
| Cukup setuju | 2 | 28 | 7.1% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

Hasil rekapitulasi data kuesioner dan persentase jawaban dari pengguna aplikasiuntuk pertanyaan “Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?”, 35.7% menjawab sangat setuju, 60.7% menjawab setuju, 3.6% menjawab cukup, dan 0% menjawab kurang setuju, serta 0% menjawab tidak setuju. Dapat dilihat pada Tabel 4.22.

Tabel 4. Pengujian MOS pembeli (pertanyaan 6)

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Jawaban | Frekuensi Jawaban | Jumlah Populasi Jawaban | Presentase |
| Sangat setuju | 10 | 28 | 35.7% |
| Setuju | 17 | 28 | 60.7% |
| Cukup setuju | 1 | 28 | 3.6% |
| Kurang setuju | 0 | 28 | 0% |
| Tidak setuju | 0 | 28 | 0% |

# BAB V

**PENUTUP**

## Kesimpulan

Berdasarkan uraian dan pembahasan yang telah disajikan sebelumnya, berikut ini kesimpulan yang dapat ditarik:

1. “Aplikasi Penjualan Hamster dan Kelengkapannya Berbasis Android Dengan Metode Prototype (Studi Kasus Toko Lombok Rabbit Pet House)” pada penelitian ini dapat membantu pihak penjual dan pembeli dalam melakukan transaksi secara *online* karena dapat menampung data barang dagang dan melaksanakan fungsi transaksi *online* dengan baik.
2. Fitur konsultasi sederhana berbentuk QnA pada aplikasi ini berjalan dengan baik dapat menyimpulkan suatu penyakit berdasarkan gejala dominan yang telah dipilih pengguna.
3. Hasil *blackbox* *testing* menyatakan bahwa fitur-fitur pada aplikasi ini berjalan sesuai harapan klien.
4. Metode prototype pada pengembangan aplikasi ini menunjang baik dari segi fungsionalitas maupun estetika karena dilakukan pengujian dan konsultasi secara berkala baik dengan klien (pihak toko Lombok Rabbit Pet House) maupun pakar.
5. Hasil pengujian MOS kepada dua orang pegawai toko Lombok Rabbit Pet House dan 28 orang responden yang diambil secara acak dari berbagai kabupaten di Lombok (12 orang berasal dari Lombok Timur, lima orang berasal dari Lombok Barat, dan 11 orang berasal dari Mataram) menyatakan rata-rata responden dengan penilaian terhadap aplikasi menyatakan setuju dan sangat setuju masing – masing bernilai 58.9% dan 36.9% pada *user* pembeli dan 41.6% dan 25% pada *user* penjual. Dan untuk jawaban kurang setuju, rata-rata bernilai 33.34% pada responden pegawai toko, dan 4.16% dari responden acak. Sedangkan pada keduanya, untuk jawaban kurang setuju dan tidak setuju masing-masing memiliki nilai rata-rata 0%

## Saran

Saran yang diberikan agar sistem yang telah dibangun menjadi lebih baik lagi yaitu sebagai berikut:

1. Menambah fitur *wishlist* untuk memudahkan pembeli dalam menyimpan barang yang ingin dibeli.
2. Mengoptimalkan desain tampilan agar lebih nyaman saat digunakan oleh pengguna.

# LAMPIRAN

**Kode activity\_lupa\_katasandi.xml:**

|  |
| --- |
| <?xml version=”1.0” encoding=”utf-8”?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".lupa\_katasandi"  android:background="@drawable/bg">  <androidx.cardview.widget.CardView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="50dp"  android:layout\_marginLeft="30dp"  android:layout\_marginRight="30dp">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Isi untuk ganti kata sandi"  android:layout\_marginTop="10dp"  android:gravity="center"  android:textStyle="bold"  android:textSize="20dp"  android:id="@+id/tv\_titlegantipassword"/>  <LinearLayout  android:layout\_below="@+id/tv\_titlegantipassword"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"  android:orientation="vertical"  android:padding="16dp">  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  android:layout\_height="wrap\_content">  <com.google.android.material.textfield.TextInputEditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/email"  android:hint="Kata Sandi"  android:inputType="textEmailAddress"  /> </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent" style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  android:layout\_height="wrap\_content"> <com.google.android.material.textfield.TextInputEditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/kodepin"  android:hint="Kode Pin"  android:maxLength="4"  android:inputType="number"/> </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  app:passwordToggleEnabled="true" style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  android:layout\_height="wrap\_content">  <com.google.android.material.textfield.TextInputEditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/passwordbaru"  android:hint="Kata Sandi"  android:inputType="textEmailAddress"/>  </com.google.android.material.textfield.TextInputLayout>  <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/btn\_selanjutnya\_lupakatasandi"  android:layout\_marginTop="20dp"  android:text="selanjutnya!"  android:background="#bcaaa4"/>  <TextView  android:id="@+id/tv\_belumpunyaakun1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="belum punya akun?"  android:layout\_margin="20dp"  android:gravity="right"/>  </LinearLayout>  </RelativeLayout>  </androidx.cardview.widget.CardView> </RelativeLayout> |

Pengujian pada fungsi-fungsi lain :

**Fungsi *login* penjual**

Fungsi *login* pada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman *login* penjual sudah berjalan dengan baik dan benar. Pengujian fungsi *login* pada penjual akan dijelaskan pada Tabel 1.

Tabel 1. Pengujian *Blackbox* fungsi *login* penjual

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| *Form username* (nama pengguna) dan kata sandi diisi dengan data yang tidak sesuai | *Login* tidak berhasil | Sesuai | Valid |
| *Form username* (nama pengguna) dan kata sandi diisi dengan data yang sesuai | *Login* berhasil kemudian sistem menampilkan halaman beranda penjual | Sesuai | Valid |

**Fungsi ganti kata sandi penjual**

Fungsi ganti kata sandipada penjual dilakukan untuk menguji apakah fungsi yang berjalan pada halaman profil penjual sudah berjalan dengan baik dan benar. Pengujian fungsi ganti kata sandi akan dijelaskan pada Tabel 2.

Tabel 2. Pengujian *Blackbox* fungsi ganti kata sandi penjual

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual mengubah kata sandi dengan data yang sesuai dari halaman profil | Pengubahan kata sandi berhasil kemudian sistem kembali menampilkan halaman beranda penjual | Sesuai | Valid |
| Penjual mengubah kata sandi dengan data yang tidak sesuai dari halaman profil | Pengubahan kata sandi tidak berhasil | Sesuai | Valid |

**Fungsi *logout* penjual**

Fungsi *logout* pada penjual dilakukan untuk menguji apakah fungsi *logout* yang ada sudah berjalan dengan baik dan benar. Pengujian fungsi ganti kata sandi akan dijelaskan pada Tabel 3.

Tabel 3. Pengujian *Blackbox* fungsi *logout* penjual

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual melakukan *logout* dari halaman profil | *logout* berhasil kemudian sistem kembali menampilkan halaman *login* penjual | Sesuai | Valid |

**Fungsi registrasi pembeli**

Fungsi registrasi pembelidilakukan untuk menguji apakah fungsi yang berjalan pada halaman daftar pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi daftar pembeli akan dijelaskan pada Tabel 4.

Tabel 4. Pengujian *Blackbox* fungsi registrasi pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Pembeli mengisi data *form* registrasi | Pendaftaran pembeli berhasil dan menampilkan halaman *login* | Sesuai | Valid |

**Fungsi *login* pembeli**

Fungsi *login* pembelidilakukan untuk menguji apakah fungsi yang berjalan pada halaman *login* pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi *login* pembeli akan dijelaskan pada Tabel 5.

Tabel 5. Pengujian *Blackbox* fungsi *login* pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| *Form email* dan kata sandi diisi dengan data yang tidak sesuai | *login* tidak berhasil kemudian sistem kembali menampilkan halaman *login* pembeli | Sesuai | Valid |
| *Form email* dan kata sandi diisi dengan data yang sesuai | *login* berhasil kemudian sistem menampilkan halaman beranda pembeli | Sesuai | Valid |

**Fungsi lupa kata sandi pembeli**

Fungsilupa kata sandi pembelidilakukan untuk menguji apakah fungsi lupa kata sandi yang berjalan pada halamanlupa kata sandi pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi lupa kata sandi pembeli akan dijelaskan pada Tabel 6.

Tabel 6. Pengujian *Blackbox* fungsi lupa kata sandi pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| *Form email*, kata sandi lama dan kata sandi baru diisi dengan data yang sesuai | Mengganti kata sandi berhasil kemudian sistem kembali menampilkan halaman *login* pembeli | Sesuai | Valid |
| *Form email*, kata sandi lama dan kata sandi baru diisi dengan data yang tidak sesuai | Mengganti kata sandi tidak berhasil | Sesuai | Valid |

**Fungsi mengubah kata sandi pembeli**

Fungsimengubah kata sandi pembelidilakukan untuk menguji apakah fungsi mengubah kata sandi yang berjalan pada halamanprofil pembeli sudah berjalan dengan baik dan benar. Pengujian fungsi mengubah kata sandi pembeli akan dijelaskan pada Tabel 7.

Tabel 7. Pengujian *Blackbox* fungsi mengubah kata sandi pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| *Form email*, kata sandi lama dan kata sandi baru diisi dengan data yang tidak sesuai | Mengganti kata sandi tidak berhasil | Sesuai | Valid |
| *Form email*, kata sandi lama dan kata sandi baru diisi dengan data yang sesuai | Mengganti kata sandi berhasil | Sesuai | Valid |

**Fungsi *logout* pembeli**

Fungsi *logout* pada pembeli dilakukan untuk menguji apakah fungsi *logout* yang ada sudah berjalan dengan baik dan benar. Pengujian fungsi *logout* akan dijelaskan pada Tabel 8.

Tabel 8 Pengujian *Blackbox* fungsi logout pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Penjual melakukan *logout* dari halaman profil | *logout* berhasil kemudian sistem kembali menampilkan halaman *login* penjual | Sesuai | Valid |

# DAFTAR PUSTAKA

[1] Sahara, Eli. 2009. Referensi Mata Kuliah Aneka Ternak dan Satwa Harapan “Hamster”. Yogyakarta: Paradigma Indonesia (*Group* Elmatera)

[2] P., drh. Lucky Retno. 2020. Gejala 6 Penyakit Umum Pada Hamster. Wawancara oleh Agung Rahmita Illah pada tanggal 18 Februari 2020

[3] Syakti, Firamon. 2019. Metode Pengembangan Perangkat Lunak Berbasis *Mobile*: *A* *Review*, (*online*) JBK, Vol. 1, No. 2, ([http://journal.binadarma.ac.id/ index.php/binakomputer/article/view/440](http://journal.binadarma.ac.id/%20index.php/binakomputer/article/view/440)), diakses pada 10 Oktober 2019.

[4] Supit, Henky, dkk. 2012. Penggunaan Teknologi Internet dalam Sistem Penjualan *Online* untuk Meningkatkan Kepuasan dan Pembelian Berulang Produk Batik pada Usaha Kecil dan Menengah di Jawa Timur, (*Online*) Jurnal Manajemen dan Kewirausahaan, Vol. 14., No. 1, ([http://jurnalmanajemen.petra.ac.id/index.php /man/article/download/18374/ 18199](http://jurnalmanajemen.petra.ac.id/index.php%20/man/article/download/18374/%2018199)), diakses pada 10 Oktober 2019.

[5] Budi, D.S., dkk. 2016. Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak, (*Online*) Teknika, Vol. 5, (http://ejournal.ikado.ac.id /index.php/teknika/article/view/48), diakses pada 12 November 2019.

[6] Anthony, Anthony, dkk. 2017. Analisis dan Perancangan Sistem Informasi Penjualan Berdasarkan Stok Gudang Berbasis *Client* *Server* (Studi Kasus Toko Grosir “Restu Anda”), (*Online*) JTIIK, Vol. 4, No. 2, ([http://jtiik.ub.ac.id/ index.php/jtiik/article/download/321/pdf](http://jtiik.ub.ac.id/%20index.php/jtiik/article/download/321/pdf)), diakses pada 3 November 2019.

[7] Jafiansyah, dkk. 2014. Aplikasi *M-Commerce* Berbasis Android Pada Kinza Collection, (*Online*) Jurnal SISKOM, Vol. 03, No. 01, ([media.neliti.com/media/ publications/265639-aplikasi-m-commerce-berbasis-android-pad-e714deb0.pdf](https://media.neliti.com/media/publications/265639-aplikasi-m-commerce-berbasis-android-pad-e714deb0.pdf)), diakses pada 18 April 2020.

[8] Listiani, Septi, dkk. 2015. Perancangan Aplikasi *Mobile E-Commerce* Berbasis Android Pada Violet Fashion Jepara, (*Online*) Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Dian Nuswantoro, (<http://eprints.dinus.ac.id/14995/>), diakses pada 18 April 2020.

[9] Yulianto, Stefanus. 2014. Aplikasi *E-Commerce* Berbasis Android Pada Phone Comp Service, (*Online*), Jurnal SISKOM , Vol. 03, No. 02, (http://jurnal.atmaluhur.ac. id/index.php/sisfokom/article/download/208/171), diakses pada 18 April 2020.

[10] Iqbal, Izharyan. 2015. Perancangan Aplikasi Penjualan Berbasis Web Dengan Metode *Prototyping* Pada CV Khatulistiwa, (*Online*), e-Proceeding of Engineering, Vol. 2, No. 1, ([https://openlibrary.telkomuniversity.ac.id/pustaka/files/100166/ jurnal\_eproc](https://openlibrary.telkomuniversity.ac.id/pustaka/files/100166/%20jurnal_eproc)), diakses pada 18 April 2020.

[11] Agitha, Nadiyasari. 2009. Sistem Informasi dan Konsultasi Pemilihan Kosmetik yang Aman Berbasis Web, (*Online*) Jurusan Teknik Informatika Fakultas Teknik Universitas Surabaya, ([http://digilib.ubaya.ac.id/index.php?page=data\_ eksemplar&key=216794 &status=ADA](http://digilib.ubaya.ac.id/index.php?page=data_%20eksemplar&key=216794%20&status=ADA)), diakses pada 22 Februari 2020.

[12] Muhammad, Kanda Y., dkk. 2013. Hewan Kesayangan Mini & Eksotis. Jakarta: Penebar Swadaya Grup.

[13] Griana, Tias Pramesti. 2013. *Scabies*: Penyebab, Penanganan, dan Pencegahannya, (*Online*) El-Hayah, Vol. 4, No. 1, ([https://www.researchgate.net/publication/ 309724092\_SCABIES\_PENYEBAB\_PENANGANAN\_DAN\_PENCEGAHANNYA](https://www.researchgate.net/publication/%20309724092_SCABIES_PENYEBAB_PENANGANAN_DAN_PENCEGAHANNYA)), diakses pada 18 Februari 2020.

[14] Muslimawaty, Tiara. 2016. Uji Efektivitas Tobramicin Tetes Mata dan Gatifloxacin Tetes Mata Terhadap *Staphylococcus aureus* Secara *Infitro*, (*Online*) Fakultas Kedokteran Universitas Muhammadiyah Malang, ([http://eprints.umm.ac.id/ 34682/](http://eprints.umm.ac.id/%2034682/)), diakses pada 29 Februari 2020.

[15] Andani, Widiyawati. 2018. Penerapan Mobilisasi Dini Pada Asuhan Keperawatan Pasien *Post* Operasi Fraktur Femur Dengan Gangguan Pemenuhan Kebutuhan Aktivitas di RDUD Sleman, (*Online*) Jurusan Keperawatan Politeknik Kesehatan Kementrian Kesehatan Yogyakarta, (<http://eprints.poltekkesjogja.ac.id/1360/>), diakses pada 29 Februari 2020.

[16] Andriani, Lisa. 2018. Pelaksanaan Layanan Konsultasi di Madrasah Tsanawiyah Swasta Al-Washliyah Kolam, (*Online*) Fakultas Ilmu Tarbiyah dan Keguruan Universitas Islam Negeri Sumatera Utara Medan, (http://repository. uinsu.ac.id/4975/), diakses pada 5 November 2019.

[17] Badaini, Aisah, dkk. 2016. Aplikasi Fuzzy Decision Making untuk Diagnosis Penyakit Tropis, (*Online*) Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta, (http://eprints.uny.ac.id /40665/), diakses pada 12 November 2019.

[18] Illina, Humaira Indreswari. 2019. Sistem Informasi Daya Tampung Area Parkir Pada Pusat Perbelanjaan untuk Meningkatkan Layanan Penggunaan Parkir (Studi Kasus: Lombok Epicentrum Mall). Skripsi. Tidak Diterbitkan. Fakultas Teknik. Universitas Mataram: Mataram.

[19] Maiyana, Efmi. 2018. Pemanfaatan Android dalam Perancangan Aplikasi Kumpulan Doa, (*Online*) Jurnal Sains dan Informatika, Vol. 4, No. 11 (55-67), (http://doi.org/10.22216 /jsi.v4il.3409), diakses pada 12 November 2019.

[20] Afuan, Lasmedi. 2010. Pemanfaatan Framework Codeigniterdalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed (Codeigniter Framework Used in Information System Development for Student’s Report Data Collection Practices Program in Informatics Engineering Program Study of Unsoed). JUITA Vol 1 Nomor 2 Hal 39-44.

[21] Sibarani, Niko Sumanda, dkk. 2018. Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. Politeknik Negeri Bandung: Bandung.

[22] Akbar, Muhammad. 2018. Pengembangan Restful Api Untuk Application Specific High Level Location Service. Universitas Islam Indonesia: Yogyakarta.

[22] Pranata, Beni Adi. 2017. Perancangan *Application* *Programming* *Interface* (Api) Berbasis Web Menggunakan Gaya Arsitektur *Representational* *State* *Transfer* (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. Universitas Lampung: Lampung.