**DPPL-xx**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Lelang Online Preloved (LeBaBe)

untuk:

Mata Kuliah Analisis Perancangan Perangkat Lunak

Dipersiapkan oleh:

Imaduddin M. Fadhil (1301184115)

Irfan Budi Prakoso (1301184088)

Muhammad Faishal Darma Putra (1301183483)

Muhammad Shiba Kabul (1301183457)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-xx <xx:no grp>* | | *<#>/<jml #* |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

[Daftar Tabel 8](#_Toc38733769)

[Daftar Gambar 9](#_Toc38733770)

[1. Pendahuluan 11](#_Toc38733771)

[1.1 Tujuan Penulisan Dokumen 11](#_Toc38733772)

[1.2 Lingkup Masalah 11](#_Toc38733773)

[1.3 Definisi dan Istilah 11](#_Toc38733774)

[1.4 Referensi 12](#_Toc38733775)

[1.5 Sistematika Pembahasan 12](#_Toc38733776)

[2 Deskripsi Perancangan Global 13](#_Toc38733777)

[2.1 Deskripsi Arsitektural 13](#_Toc38733778)

[2.2 Deskripsi Komponen 14](#_Toc38733779)

[3 Perancangan Rinci 16](#_Toc38733780)

[3.1 Realisasi Use Case 16](#_Toc38733781)

[3.1.1 Use Case #1 <Registrasi> 17](#_Toc38733782)

[3.1.1.1 Perancangan Antarmuka Usecase #1 17](#_Toc38733783)

[3.1.1.1.1 Tabel Identifikasi Antarmuka / Layar / Page 18](#_Toc38733784)

[3.1.1.2 Identifikasi Object Baru 18](#_Toc38733785)

[3.1.1.3 Robustness Diagram 19](#_Toc38733786)

[3.1.1.4 Diagram Kelas 19](#_Toc38733787)

[3.1.1.5 Sequence Diagram 20](#_Toc38733788)

[3.1.2 Use Case #2 <Login> 21](#_Toc38733789)

[3.1.2.1 Perancangan Antarmuka Usecase #2 21](#_Toc38733790)

[3.1.2.1.1 Tabel Identifikasi Antarmuka / Layar / Page 22](#_Toc38733791)

[3.1.2.2 Identifikasi Object Baru 22](#_Toc38733792)

[3.1.2.3 Robustness Diagram 23](#_Toc38733793)

[3.1.2.4 Diagram Kelas 24](#_Toc38733794)

[3.1.2.5 Sequence Diagram 24](#_Toc38733795)

[3.1.3 Use Case #3 <inputBarang> 25](#_Toc38733796)

[3.1.3.1 Perancangan Antarmuka Usecase #3 26](#_Toc38733797)

[3.1.3.1.1 Tabel Identifikasi Antarmuka / Layar / Page 26](#_Toc38733798)

[3.1.3.2 Identifikasi Object Baru 27](#_Toc38733799)

[3.1.3.3 Robustness Diagram 27](#_Toc38733800)

[3.1.3.4 Diagram Kelas 28](#_Toc38733801)

[3.1.3.5 Sequence Diagram 28](#_Toc38733802)

[3.1.4 Use Case #4 <updateBarang> 29](#_Toc38733803)

[3.1.4.1 Perancangan Antarmuka Usecase #4 29](#_Toc38733804)

[3.1.4.1.1 Tabel Identifikasi Antarmuka / Layar / Page 30](#_Toc38733805)

[3.1.4.2 Identifikasi Object Baru 30](#_Toc38733806)

[3.1.4.3 Robustness Diagram 31](#_Toc38733807)

[3.1.4.4 Diagram Kelas 32](#_Toc38733808)

[3.1.4.5 Sequence Diagram 32](#_Toc38733809)

[3.1.5 Use Case #5 <hapusBarang> 33](#_Toc38733810)

[3.1.5.1 Perancangan Antarmuka Usecase #5 34](#_Toc38733811)

[3.1.5.1.1 Tabel Identifikasi Antarmuka / Layar / Page 34](#_Toc38733812)

[3.1.5.2 Identifikasi Object Baru 35](#_Toc38733813)

[3.1.5.3 Robustness Diagram 35](#_Toc38733814)

[3.1.5.4 Diagram Kelas 36](#_Toc38733815)

[3.1.5.5 Sequence Diagram 36](#_Toc38733816)

[3.1.6 Use Case #6 <viewBarangDijual untuk Penjual> 37](#_Toc38733817)

[3.1.6.1 Perancangan Antarmuka Usecase #6 37](#_Toc38733818)

[3.1.6.1.1 Tabel Identifikasi Antarmuka / Layar / Page 38](#_Toc38733819)

[3.1.6.2 Identifikasi Object Baru 38](#_Toc38733820)

[3.1.6.3 Robustness Diagram 39](#_Toc38733821)

[3.1.6.4 Diagram Kelas 39](#_Toc38733822)

[3.1.6.5 Sequence Diagram 40](#_Toc38733823)

[3.1.7 Use Case #7 <viewBarangDijual untuk Pembeli> 41](#_Toc38733824)

[3.1.7.1 Perancangan Antarmuka Usecase #7 41](#_Toc38733825)

[3.1.7.1.1 Tabel Identifikasi Antarmuka / Layar / Page 41](#_Toc38733826)

[3.1.7.2 Identifikasi Object Baru 42](#_Toc38733827)

[3.1.7.3 Robustness Diagram 43](#_Toc38733828)

[3.1.7.4 Diagram Kelas 43](#_Toc38733829)

[3.1.7.5 Sequence Diagram 44](#_Toc38733830)

[3.1.8 Use Case #8 <input bidBarang untuk Pembeli> 45](#_Toc38733831)

[3.1.8.1 Perancangan Antarmuka Usecase #8 45](#_Toc38733832)

[3.1.8.1.1 Tabel Identifikasi Antarmuka / Layar / Page 46](#_Toc38733833)

[3.1.8.2 Identifikasi Object Baru 46](#_Toc38733834)

[3.1.8.3 Robustness Diagram 46](#_Toc38733835)

[3.1.8.4 Diagram Kelas 47](#_Toc38733836)

[3.1.8.5 Sequence Diagram 47](#_Toc38733837)

[3.1.9 Use Case #9 <TransaksiBarang> 48](#_Toc38733838)

[3.1.9.1 Perancangan Antarmuka Usecase #9 48](#_Toc38733839)

[3.1.9.1.1 Tabel Identifikasi Antarmuka / Layar / Page 49](#_Toc38733840)

[3.1.9.2 Identifikasi Object Baru 49](#_Toc38733841)

[3.1.9.3 Robustness Diagram 50](#_Toc38733842)

[3.1.9.4 Diagram Kelas 51](#_Toc38733843)

[3.1.9.5 Sequence Diagram 51](#_Toc38733844)

[3.1.10 Use Case #10 <TopUp Saldo> 52](#_Toc38733845)

[3.1.10.1 Perancangan Antarmuka Usecase #10 52](#_Toc38733846)

[3.1.10.1.1 Tabel Identifikasi Antarmuka / Layar / Page 53](#_Toc38733847)

[3.1.10.2 Identifikasi Object Baru 53](#_Toc38733848)

[3.1.10.3 Robustness Diagram 53](#_Toc38733849)

[3.1.10.4 Diagram Kelas 54](#_Toc38733850)

[3.1.10.5 Sequence Diagram 54](#_Toc38733851)

[3.1.11 Use Case #11<Tarik Saldo> 55](#_Toc38733852)

[3.1.11.1 Perancangan Antarmuka Usecase #11 55](#_Toc38733853)

[3.1.11.1.1 Tabel Identifikasi Antarmuka / Layar / Page 56](#_Toc38733854)

[3.1.11.2 Identifikasi Object Baru 56](#_Toc38733855)

[3.1.11.3 Robustness Diagram 57](#_Toc38733856)

[3.1.11.4 Diagram Kelas 57](#_Toc38733857)

[3.1.11.5 Sequence Diagram 58](#_Toc38733858)

[3.1.12 Use Case #12<view Biodata> 59](#_Toc38733859)

[3.1.12.1 Perancangan Antarmuka Usecase #12 59](#_Toc38733860)

[3.1.12.1.1 Tabel Identifikasi Antarmuka / Layar / Page 59](#_Toc38733861)

[3.1.12.2 Identifikasi Object Baru 60](#_Toc38733862)

[3.1.12.3 Robustness Diagram 60](#_Toc38733863)

[3.1.12.4 Diagram Kelas 61](#_Toc38733864)

[3.1.12.5 Sequence Diagram 61](#_Toc38733865)

[3.1.13 Use Case #13 <view editBiodata> 62](#_Toc38733866)

[3.1.13.1 Perancangan Antarmuka Usecase #13 62](#_Toc38733867)

[3.1.13.1.1 Tabel Identifikasi Antarmuka / Layar / Page 63](#_Toc38733868)

[3.1.13.2 Identifikasi Object Baru 63](#_Toc38733869)

[3.1.13.3 Robustness Diagram 64](#_Toc38733870)

[3.1.13.4 Diagram Kelas 64](#_Toc38733871)

[3.1.13.5 Sequence Diagram 65](#_Toc38733872)

[4 Perancangan Detil 66](#_Toc38733873)

[4.1 Perancangan Detil Kelas 66](#_Toc38733874)

[4.2 Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel) 67](#_Toc38733875)

[4.3 Perancangan Algoritma 67](#_Toc38733876)

[4.3.1 Algoritma #1 67](#_Toc38733877)

[4.3.2 Algoritma #2 68](#_Toc38733878)

[4.3.3 Algoritma #3 68](#_Toc38733879)

[4.3.4 Algoritma #4 69](#_Toc38733880)

[4.3.5 Algoritma #5 69](#_Toc38733881)

[4.3.6 Algoritma #6 69](#_Toc38733882)

[4.3.7 Algoritma #7 70](#_Toc38733883)

[4.3.8 Algoritma #8 70](#_Toc38733884)

[4.3.9 Algoritma #9 70](#_Toc38733885)

[4.3.10 Algoritma #10 71](#_Toc38733886)

[4.3.11 Algoritma #11 71](#_Toc38733887)

[4.3.12 Algoritma #12 71](#_Toc38733888)

[4.3.13 Algoritma #13 72](#_Toc38733889)

[4.4 Perancangan Query*:* 73](#_Toc38733890)

[5 Matriks Kerunutan (Requirement Traceability Matrix) 74](#_Toc38733891)

# Daftar Tabel

[Tabel 1 Deskripsi Komponen 15](#_Toc38737308)

[Tabel 2 Realisasi Use Case 16](#_Toc38737309)

[Tabel 3 Use Case Sekanrio Registrasi 17](#_Toc38737310)

[Tabel 4 antar muka layer Registrasi 18](#_Toc38737311)

[Tabel 5 identifikasi objek baru Registrasi 18](#_Toc38737312)

[Tabel 6 Use case skenario Login 21](#_Toc38737313)

[Tabel 7 Tabel identifikasi antarmuka Login 22](#_Toc38737314)

[Tabel 8 Identifikasi objek baru 22](#_Toc38737315)

[Tabel 9 use case skenario inputBarang 25](#_Toc38737316)

[Tabel 10 identifikasi antarmuka layer inputBarang 26](#_Toc38737317)

[Tabel 11 identifikasi objek baru inputBarang 27](#_Toc38737318)

[Tabel 12 usecase scenario updateBarang 29](#_Toc38737319)

[Tabel 13 identifikasi antarmuka layar updateBarang 30](#_Toc38737320)

[Tabel 14 identifikasi objek baru updateBarang 30](#_Toc38737321)

[Tabel 15 use case scenario hapusBarang 33](#_Toc38737322)

[Tabel 16 identifikasi antarmuka hapusBarang 34](#_Toc38737323)

[Tabel 17identifikasi objek baru hapusBarang 35](#_Toc38737324)

[Tabel 18 use case scenario viewBarangDijual 37](#_Toc38737325)

[Tabel 19 identifikasi antarmuka layar viewBarangDijual 38](#_Toc38737326)

[Tabel 20 identifikasi objek baru viewBarangDijual 38](#_Toc38737327)

[Tabel 21 usecase scenario viewBarangDijualP 41](#_Toc38737328)

[Tabel 22 identifikasi antarmuka layar viewBarangDijualP 42](#_Toc38737329)

[Tabel 23 identifikasi objek baru viewBarangDijualP 42](#_Toc38737330)

[Tabel 24 use case scenario bidBarang 45](#_Toc38737331)

[Tabel 25 identifikasi antar muka layar bidBarang 46](#_Toc38737332)

[Tabel 26 identifikasi objek baru bidBarang 46](#_Toc38737333)

[Tabel 27 use case scenario transaksiBarang 48](#_Toc38737334)

[Tabel 28 identifikasi antarmuka layar transaksiBarang 49](#_Toc38737335)

[Tabel 29 identifikasi objek baru transaksiBarang 49](#_Toc38737336)

[Tabel 30 use case scenario topUp 52](#_Toc38737337)

[Tabel 31 identifikasi antarmuka layar topUp 53](#_Toc38737338)

[Tabel 32 identifikasi objek baru 53](#_Toc38737339)

[Tabel 33 use case scenario tarikSaldo 55](#_Toc38737340)

[Tabel 34 identifikasi antarmuka layar tarikSaldo 56](#_Toc38737341)

[Tabel 35 identifikasi objek baru 56](#_Toc38737342)

[Tabel 36 use case scenario viewBiodata 59](#_Toc38737343)

[Tabel 37 identifikasi antarmuka layar viewBiodata 59](#_Toc38737344)

[Tabel 38 identifikasi objek baru viewBiodata 60](#_Toc38737345)

[Tabel 39use case scenario editBiodata 62](#_Toc38737346)

[Tabel 40 identifikasi antarmuka layar editBiodata 63](#_Toc38737347)

[Tabel 41 identifikasi objek baru editBiodata 63](#_Toc38737348)

[Tabel 42 perancangan detil kelas 66](#_Toc38737349)

[Tabel 43 Matrik kerunutan 74](#_Toc38737350)

# Daftar Gambar

# 

[Gambar 1 Deskripsi Arsitektural 13](#_Toc38737351)

[Gambar 2 Deskripsi Komponen 14](file:///D:\TELKOM%20UNIVERSITY\TEKNIK%20INFORMATIKA\SEMESTER%204\APPL\TUGAS\DPPL\fixie.docx#_Toc38737352)

[Gambar 3 perancangan antarmuka Usecase 1 17](#_Toc38737353)

[Gambar 4 Robustness Diagram Registrasi 19](#_Toc38737354)

[Gambar 5 Diagram Kelas Registrasi 19](#_Toc38737355)

[Gambar 6 Sequence Diagram Registrasi 20](#_Toc38737356)

[Gambar 7 Antar Muka use case Login 21](#_Toc38737357)

[Gambar 8 Robustness diagram Login 23](#_Toc38737358)

[Gambar 9 class diagram login 24](#_Toc38737359)

[Gambar 10 sequnce diagram login 24](#_Toc38737360)

[Gambar 11 AntarMuka usecase inputBarang 26](#_Toc38737361)

[Gambar 12 robustness Diagram inputBarang 27](#_Toc38737362)

[Gambar 13 diagram kelas inputBarang 28](#_Toc38737363)

[Gambar 14 sequence diagram inputBarang 28](#_Toc38737364)

[Gambar 15 antarmuka updateBarang 29](#_Toc38737365)

[Gambar 16 robustness diagram updateBarang 31](#_Toc38737366)

[Gambar 17 diagram kelas updateBarang 32](#_Toc38737367)

[Gambar 18 sequence diagram updateBarang 32](#_Toc38737368)

[Gambar 19 antarmuka hapusBarang 34](#_Toc38737369)

[Gambar 20 robustness diagram hapusBarang 35](#_Toc38737370)

[Gambar 21 diagram kelas hapusBarang 36](#_Toc38737371)

[Gambar 22sequence diagram hapusBarang 36](#_Toc38737372)

[Gambar 23 antar muka viewBarangDijual 37](#_Toc38737373)

[Gambar 24 robustness diagram viewBarangDijual 39](#_Toc38737374)

[Gambar 25 diagram kelas viewBarangDijual 39](#_Toc38737375)

[Gambar 26 sequence diagram viewBarangDijual 40](#_Toc38737376)

[Gambar 27 antar muka viewBarangDijualP 41](#_Toc38737377)

[Gambar 28 robustness diagram viewBarangDijualP 43](#_Toc38737378)

[Gambar 29 diagram kelas viewBarangDijualP 43](#_Toc38737379)

[Gambar 30 sequence diagram viewBarangDijualP 44](#_Toc38737380)

[Gambar 31 antarmuka bidBarang 45](#_Toc38737381)

[Gambar 32 robustness diagram bidBarang 46](#_Toc38737382)

[Gambar 33 diagram kelas bidBarang 47](#_Toc38737383)

[Gambar 34 sequence diagram bidBarang 47](#_Toc38737384)

[Gambar 35 antar muka transaksiBarang 48](#_Toc38737385)

[Gambar 36 robustness diagram transaksiBarang 50](#_Toc38737386)

[Gambar 37 diagram kelas transaksiBarang 51](#_Toc38737387)

[Gambar 38 sequence diagram transaksiBarang 51](#_Toc38737388)

[Gambar 39 antar muka topUp 52](#_Toc38737389)

[Gambar 40 robustness diagram topUp 53](#_Toc38737390)

[Gambar 41 diagram kelas topUp 54](#_Toc38737391)

[Gambar 42 sequence diagram topUp 54](#_Toc38737392)

[Gambar 43 antar muka tarikSaldo 55](#_Toc38737393)

[Gambar 44 robustness diagram tarikSaldo 57](#_Toc38737394)

[Gambar 45 diagram kelas tarikSaldo 57](#_Toc38737395)

[Gambar 46 sequence diagram tarikSaldo 58](#_Toc38737396)

[Gambar 47 antar muka viewBiodata 59](#_Toc38737397)

[Gambar 48 robustness diagram viewBiodata 60](#_Toc38737398)

[Gambar 49 diagram kelas viewBiodata 61](#_Toc38737399)

[Gambar 50 sequence diagram viewBiodata 61](#_Toc38737400)

[Gambar 51 antar muka editBiodata 62](#_Toc38737401)

[Gambar 52 robustness diagram editBiodata 64](#_Toc38737402)

[Gambar 53 diagram kelas editBiodata 64](#_Toc38737403)

[Gambar 54 sequence diagram editBiodata 65](#_Toc38737404)

[Gambar 55 perancangan kelas persistensi 67](file:///D:\TELKOM%20UNIVERSITY\TEKNIK%20INFORMATIKA\SEMESTER%204\APPL\TUGAS\DPPL\fixie.docx#_Toc38737405)

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

*Dokumen spesifikasi kebutuhan perangkat lunak (SKPL) merupakan spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahapan selanjutnya.*

## Lingkup Masalah

*LeBabe adalah perangkat lunak untuk lelang pakaian bekas berbasis web yang digunakan sebagai penampung aktivitas lelang pakaian bekas yang saat ini masih umum dilakukan pada media social, maka dalam hal ini kita mencoba mengembangkan sebuah website yang menampung berbagai macam pakaian bekas dan didalam program ini meliputi konektivitas internet dan sisteminformasi yang dapat memudahkan transaksi lelang pakaian bekas antar penjual dan pembeli.*

## Definisi dan Istilah

* ***SKPL***

*Spesifikasi Kebutuhan Perangkat Lunak, atau dalam Bahasa Inggris-nyasering juga disebut sebagai Software Requirements Specification (SRS), dan merupakan spesifikasi perngkat lunak yang akan dikembangkan.*

* ***DFD***

*Data Flow Program, diagram dan notasi yang digunakan untuk menunjukan aliran data pada perangkat lunak.*

* ***PSPEC***

*level abstraksi yang paling rendah (di DFD).*

* ***USE CASE***

*Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau system informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara ‘aktor’ — inisiator dari interaksi system itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana.*

* ***CLASS DIAGRAM***

*Class diagram adalah diagram yang menggambarkan struktur system dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelasmemiliki*

*3 bagian utama yaitu attribute, operation, dan name. kelas-kelas yang ada pada struktur system harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhansistem.*

* ***ERD***

*Entity Relationship Diagram,diagram dan notasi yang digunakan untuk mempresentasikan struktur data statis pada perangkat.*

* ***HTML***

*Hyper Text Markup Language, sintaks bahasa yang digunakan dalam pembuatan website.*

* ***CSS***

*Cascading Style Sheet, Merupakan salah satu Bahasa pemrograman web untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur.*

## Referensi

*Dalam pengerjaan dokumentasi PL ini, referensi-referensi yang digunakan adalah sebagai berikut:*

* *SKPL PRELOVED*
* *DPPL-03 Advanced Restaurant System*

## Sistematika Pembahasan

Dokumen DPPL untuk Online Lelang Perloved (LeBabe) ini berisi deskripsi kebutuhan

pengembangan perangkat lunak secara rinci. Dokumen ini dibagi menjadi 3 bagian utama,

yaitu:

1. Bab 1 Pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang ditangani

pada perangkat lunak Online Lelang Perloved (LeBabe), definisi (beserta istilah dan

singkatan), aturan penomoran, referensi yang digunakan untuk dokumen ini, serta

deskripsi umum dokumen (ikhtisar)

1. Bab 2 Deskripsi Perancangan Global, berisi deskripsi rancangan lingkungan

implementasi, deskripsi basis data yang dikelola oleh Online Lelang Perloved (LeBabe),

serta deskripsi hirarki dari modul proses yang akan dibangun

1. Bab 3 Deskripsi Perancangan Rinci, berisi deskripsi rinci tabel,

deskripsi rinci masing-masing modul yang mencakup fungsi beserta spesifikasispesifikasinya, dan matriks kerunutan dari perancangan Online Lelang Perloved (LeBabe)ini.

# Deskripsi Perancangan Global

## Deskripsi Arsitektural

*Aplikasi perangkat lunak berbasis website ini adalah aplikasi yang dapat digunakan oleh user pada halaman browser, dengan konektivitas internet dan menggunakan SQL sebagai database.*



Gambar 1Deskripsi Arsitektural

## Deskripsi Komponen

Gambar 2 Deskripsi Komponen

Tabel 1 Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1. | User | Merupakan class untuk user |
| 2. | Index.php | Merupakan interface awal |
| 3. | Login.php | Merupakan componen untuk login |
| 4. | Register.php | Merupakan componen untuk register |
| 5. | Profile.php | Merupakan interface profile setelah log in |
| 6. | editProfile.php | Merupakan component untuk meng edit profile |
| 7. | topUp.php | Merupakan component untuk top up saldo |
| 8. | tarikSaldo.php | Merupakan component untuk menarik saldo |
| 9. | Pembeli | Merupakan class untuk pembeli |
| 10. | Pembeli.php | Merupakan componen untuk pembeli |
| 11. | viewListLelang.php | Merupakan komponen untuk menampilkan seluruh lelang aktif |
| 12. | bidBarang.php | Merupakan kompenen untuk melakukan fungsi bid barang lelang |
| 13 | viewBarangTerbeli.php | Merupakan komponen untuk menampilkan barang terbeli |
| 14 | Penjual | Merupakan class untuk penjual |
| 15 | Penjual.php | Merupakan component untuk penjual |
| 16 | inputBarangLelang.php | Merupakan komponen untuk melakukan fungsi membuat lelang baru |
| 17 | viewBarangDijual.php | Merupakan komponen untuk menampilkan barang yan dijual |

# Perancangan Rinci

## Realisasi Use Case

Tabel 2 Realisasi Use Case

|  |  |  |
| --- | --- | --- |
| No | Nama Use Case | Deskripsi Use Case |
| 1 | Registrasi | Fungsiinidigunakan oleh user untukmelakukanregistrasi agar dapatmengaksesfitur di website. |
| 2 | Login | Fungsiinidigunakan oleh user untukmengaksesfitur di website |
| 3 | Top Up dan Tarik Saldo | Fungsiinidigunakan oleh user untukmelakukanpenambahansaldountukpembelian dan biasditarikkembalijikasudahtidakdigunakan |
| 4 | MenjualBarang | Fungsiinidigunakan oleh user(penjual) untukmelakukanpenjualan oleh penjual |
| 5 | Melihat Status Barang Yang user Jual | Fungsiinidigunakan oleh user(penjual) untukmelakukanpenglihatan status barang yang user jual |
| 6 | Meletakan Bid | Fungsiinidigunakan oleh user(pembeli) untukmeletakan bid pada sitempelelangan |
| 7 | Melihat Daftar Barang yang SedangDilelang | Fungsiinidigunakan oleh user untukmelihat daftar barang yang sedangdilelang |
| 8 | Input Barang | Fungsiinidigunakan oleh user untukmelakukan Input baranguntukdijualke system |
| 9 | ValidasiTransaksi | Fungsiinidigunakan oleh user untukmelakukanvalidasitransaksi |
| 10 | Pendataanterhadapbarang yang dilelang | Fungsiinidigunakan oleh user untukmelakukanpendataanbarang yang dilelang |

### Use Case #1 <Registrasi>

*Skenario Use Case #1 :*

Tabel 3 Use Case Sekanrio Registrasi

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : Registrasi |
| Actor : User |
| Pre-Condition : User inginmelakukanregistrasi |
| Post-Condition : User sudahterdaftardalam database  sebagaipengguna |
| Deskripsi : Untukmendaftarkandirisebagaipenggunawebsite |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka Website Pre-Loved |  |
| 2. Mengklik Tab Registrasi |  |
| 3. Mengisi data user |  |
|  | 4. Menambahkankedalam database bahwa user sudahterdaftar |
|  | 5. Menampilkanbahwa user sudahterdaftar |
|  | 6. Menampilkanhalaman login |
| 7. Melakukan Login |  |

#### Perancangan Antarmuka Usecase #1



Gambar 3 perancangan antarmuka Usecase 1

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 4 antar muka layer Registrasi

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page I | *Halaman Registrasi* | Page ini berisi halaman registrasi dari menu Registrasi |

*Page LOGIN*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonRegist* | Button | REGISTRASI | Jika diklik, akanmengaktifkan Functionregistrasi |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu home |
| *txtNama* | *Textbox* | *Nama* | Isi Teks yang disimpan pada File c\_registrasi |
| *txtUsername* | *Textbox* | *Username* | Isi Teks yang disimpan pada File c\_registrasi |
| *txtEmail* | *Textbox* | *Email* | Isi Teks yang disimpan pada File c\_registrasi |
| *txtTtl* | *Textbox* | *TTL* | Isi Teks yang disimpan pada File c\_registrasi |
| *txtPassword* | *Textbox* | *Password* | Isi Teks yang disimpan pada File c\_registrasi |

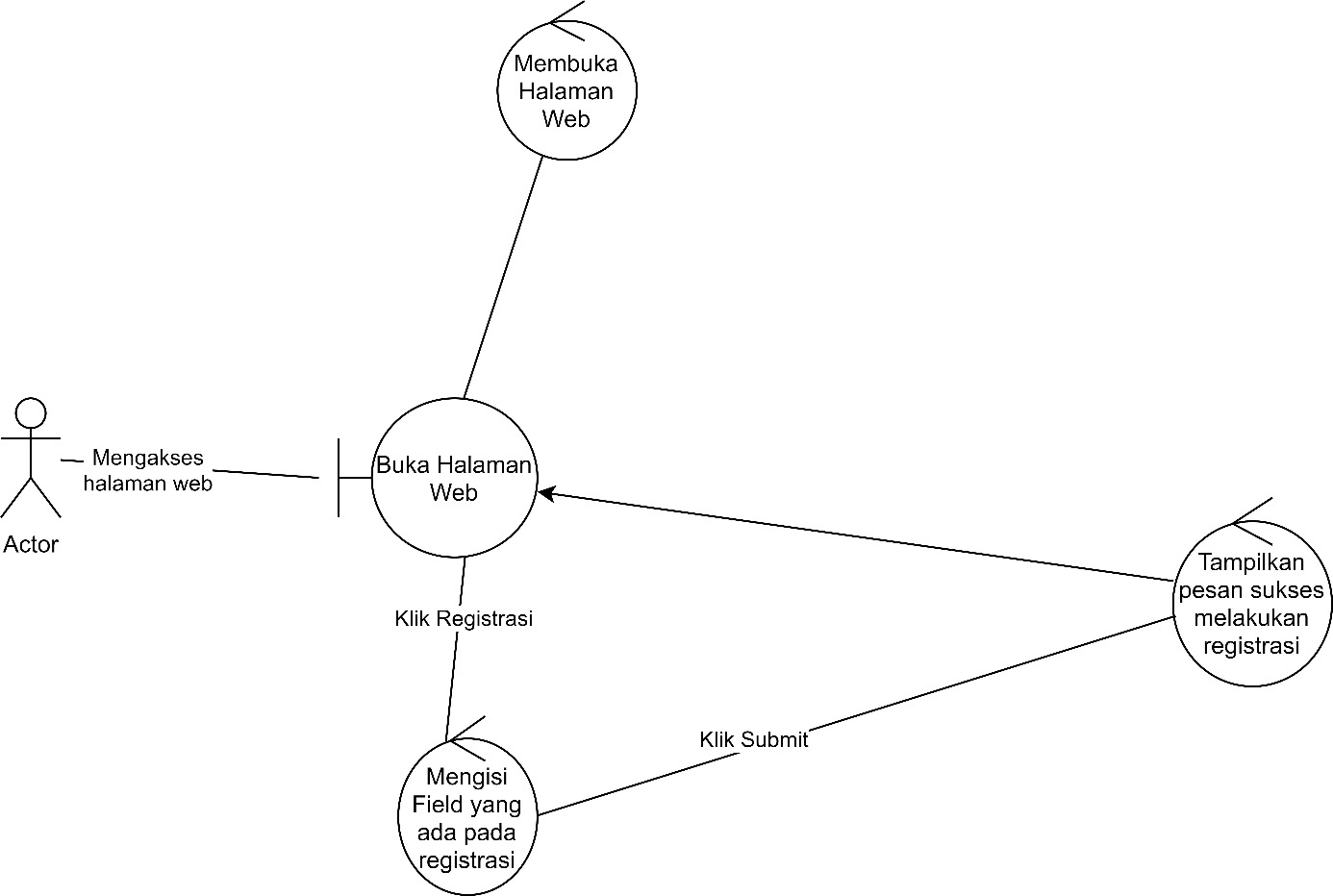
#### Identifikasi Object Baru

Tabel 5 identifikasi objek baru Registrasi

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User* | *User* |
| *2.* | *Database* | *Database* |

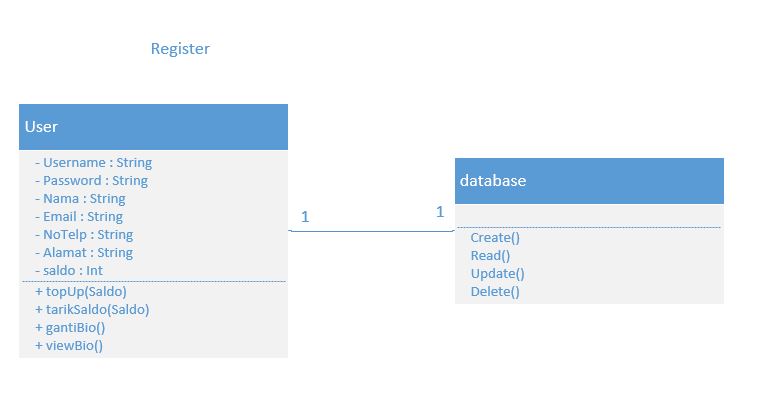
#### 

#### Robustness Diagram



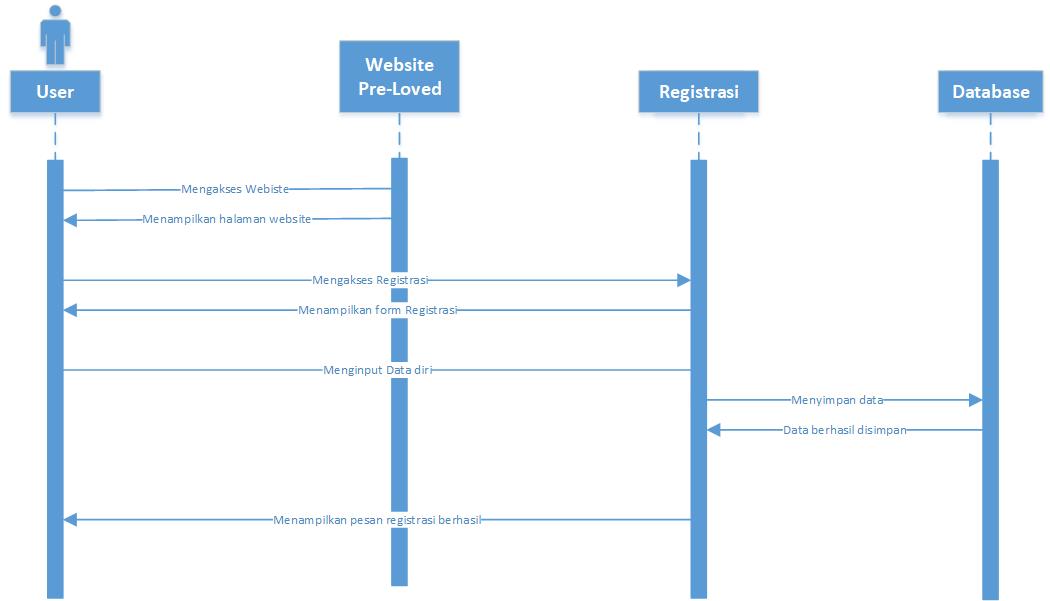
Gambar 4 Robustness Diagram Registrasi

#### Diagram Kelas

**

Gambar 5 Diagram Kelas Registrasi

#### Sequence Diagram



Gambar 6 Sequence Diagram Registrasi

### Use Case #2<Login>

*Skenario Use Case #2*

Tabel 6 Use case skenario Login

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : Login |
| Actor : User |
| Pre-Condition : User ingin menggunakan program dan belum login |
| Post-Condition : User sudah masuk kedalam halaman utama |
| Deskripsi : User melakukan login untuk kegiatan |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Memasukan Usernama dan password |  |
| 2. Mengklik Login |  |
|  | 3. Eksekusi Username dan Password yang telah ada didatabse |
|  | 4. Bila terjadi kesalahan kembali ke no. 1 |
|  | 5. Jika sesuai maka dapat melanjutkan ke no.6 |
|  | 6. Menampilkan halaman utama |
| 7. Melakukan Aktifitas |  |

#### Perancangan Antarmuka Usecase #2



Gambar 7 Antar Muka use case Login

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 7 Tabel identifikasi antarmuka Login

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 2 | *Halaman Login* | Page ini berisi halaman login dari menu Login |

*Page LOGIN*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonLogin* | Button | LOGIN | Jika diklik, akanmengaktifkan Function….. |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu home |
| *txtUsername* | *Textbox* | *Username* | Isi Teks yang disimpan pada File c\_login |
| *txtUsername* | *Textbox* | *Password* | Isi Teks yang disimpan pada File c\_login |

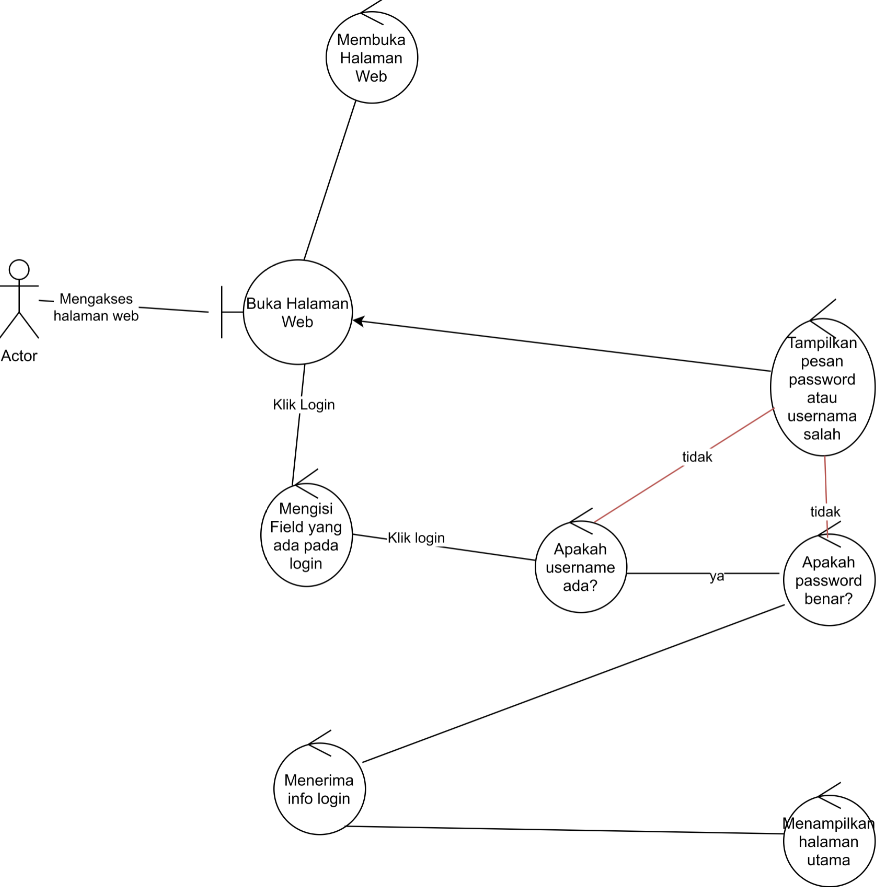
#### Identifikasi Object Baru

Tabel 8 Identifikasi objek baru

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User* | *User* |
| *2.* | *Database* | *Databse* |

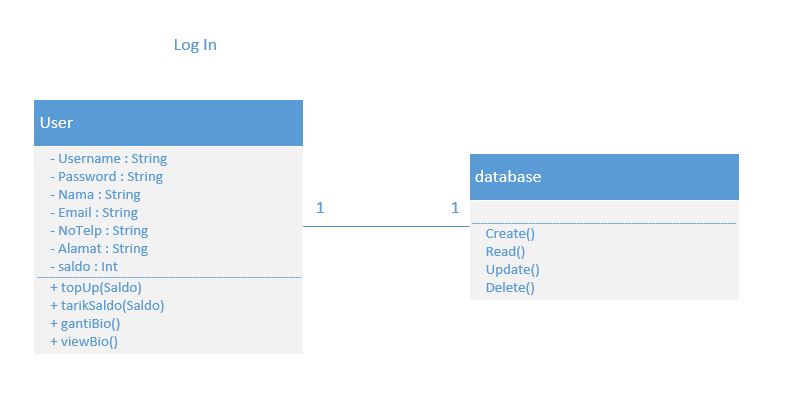
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



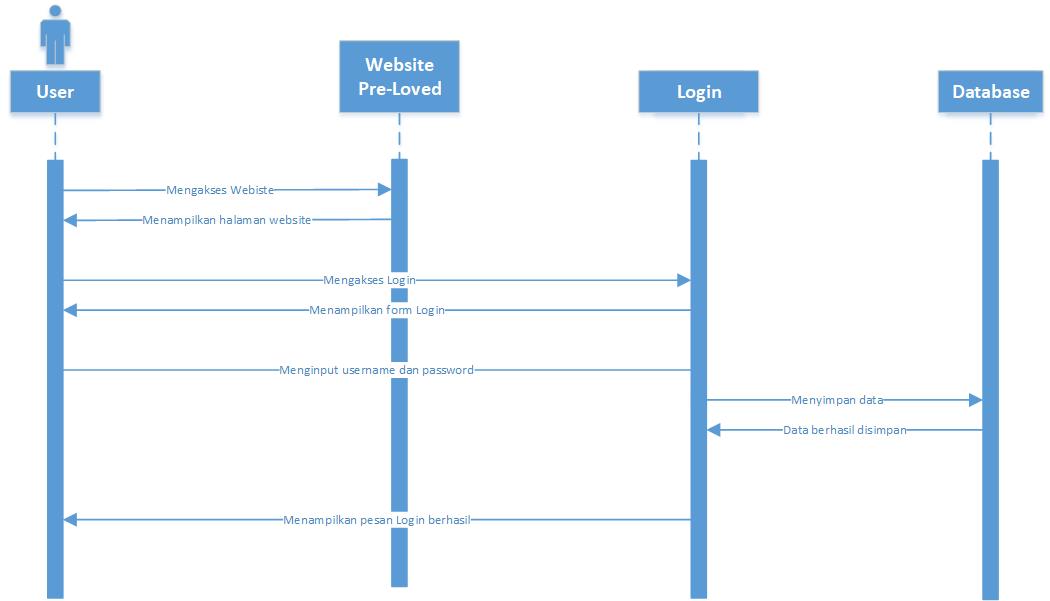
Gambar 8 Robustness diagram Login

#### Diagram Kelas

**

Gambar 9 class diagram login

#### Sequence Diagram



Gambar 10 sequnce diagram login

### Use Case #3 <inputBarang>

*Skenario Use Case #3*

Tabel 9 use case skenario inputBarang

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : Input Barang |
| Actor : Penjual |
| Pre-Condition : User ingin menginput |
| Post-Condition : User sudah menginput |
| Deskripsi : Untuk mencatat data barang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab penjual |  |
| 2. Mengklik data barang |  |
|  | 3. Menampilkan kelola barang |
| 4. Memilih input barang |  |
|  | 5. keluar tampilan input |
| 6. memasukkan barang |  |
| 7. Menekan tombol simpan |  |
|  | 8. Memproses data yang telah disimpan dan masuk ke database |
|  | 9. Menampilkan data yang sudah di simpan |
| 10. Mengecek Kembali data yang disimpan |  |

#### Perancangan Antarmuka Usecase #3



Gambar 11 AntarMuka usecase inputBarang

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 10 identifikasi antarmuka layer inputBarang

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 3 | *Halaman inputBarang* | Page ini berisi halaman inputBarang dari menu Jual Barang |

*Page inputBarang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonListBarangDijual* | Button | List Barang Dijual | Jika diklik, akanmengaktifkan FunctionviewListBarangDijual |
| *ButtonInputList* | *Button* | *Input ke List Barang* | Jika diklik, akanmengaktifkan FunctionjualBarang |
| *ButtonUploadFoto* | *Button* | *UPLOAD* | Jika diklik, akanmengaktifkanuploadPhoto |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu penjual |
| *txtNamaBr* | *Textbox* | *Nama Barang* | Isi Teks yang disimpan pada Filec\_jualBarang |
| *txtMinBid* | *Textbox* | *MIN Bid* | Isi Teks yang disimpan pada File c\_jualBarang |

#### Identifikasi Object Baru

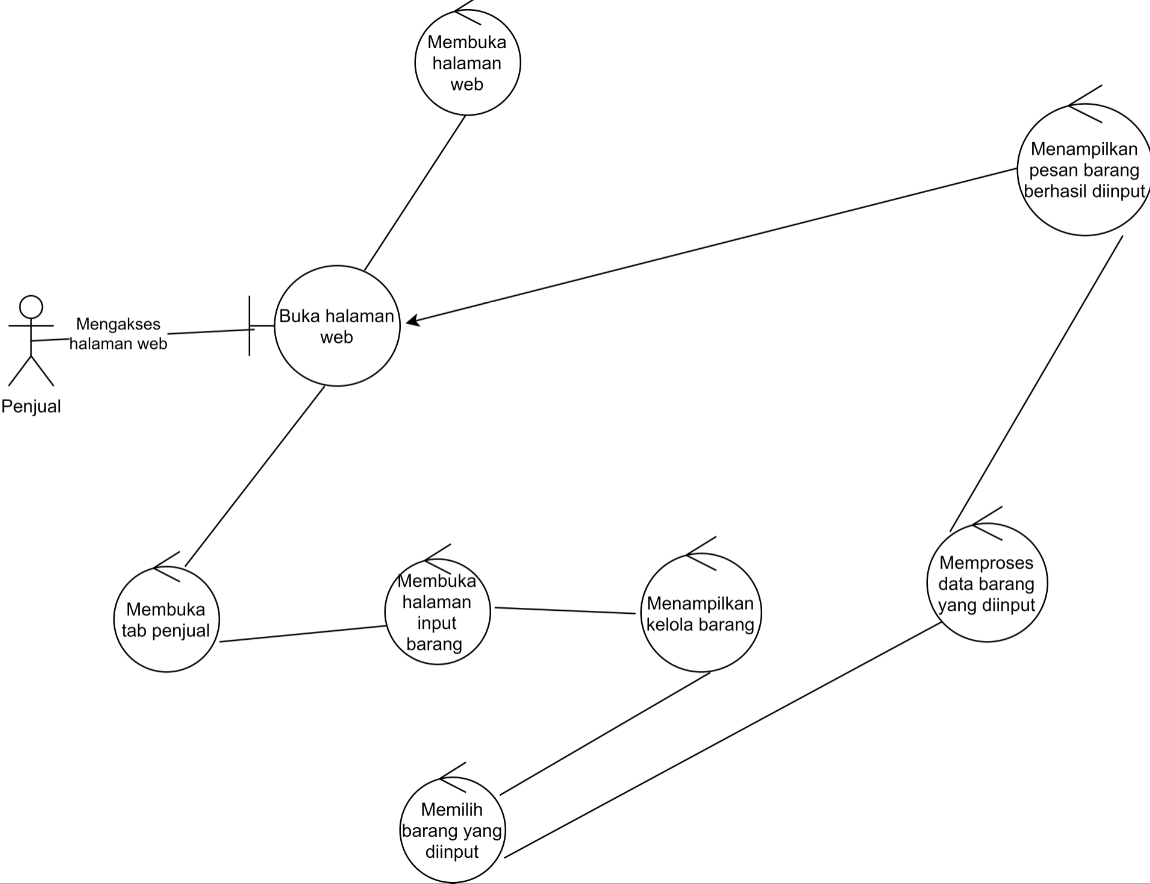
TABEL OBJECT PERANCANGAN

Tabel 11 identifikasi objek baru inputBarang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |

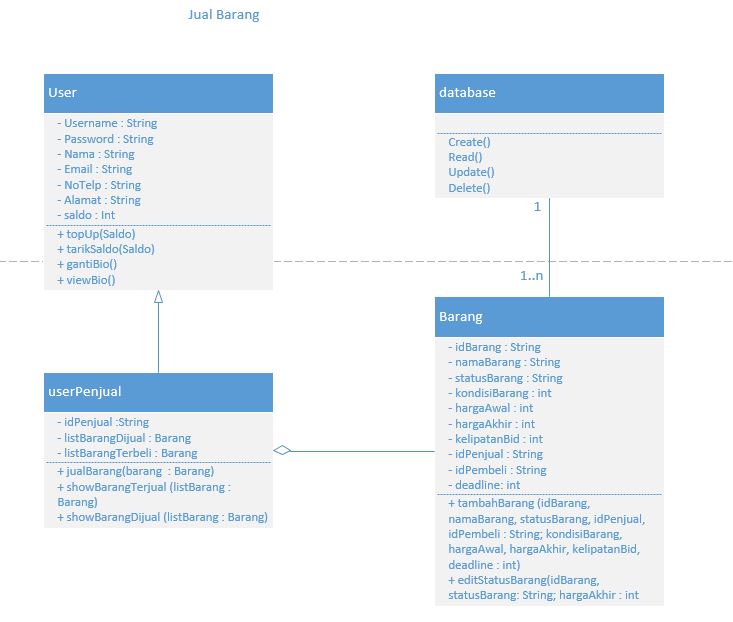
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



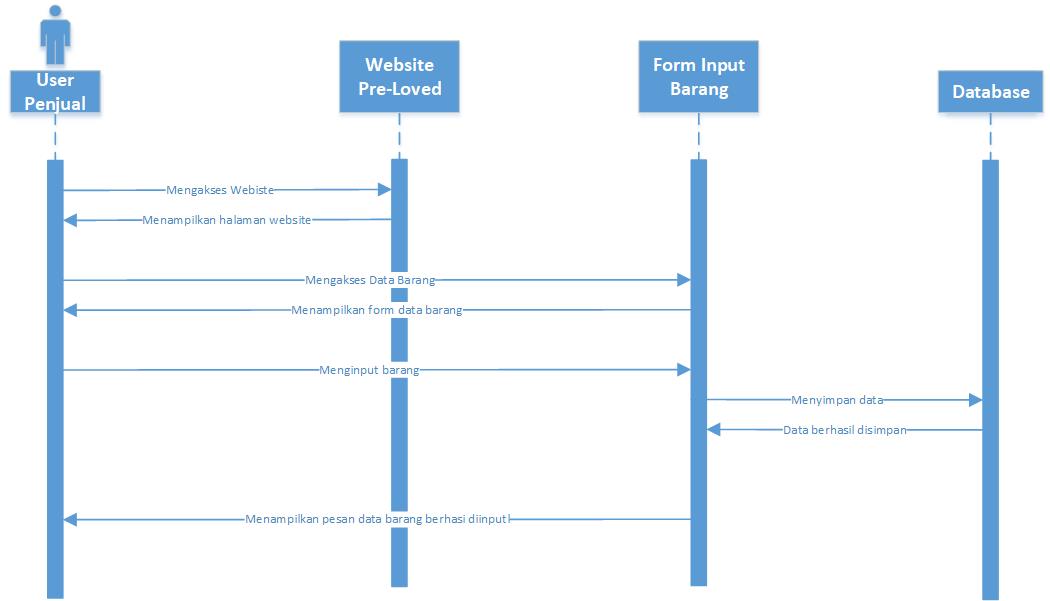
Gambar 12 robustness Diagram inputBarang

#### Diagram Kelas

**

Gambar 13 diagram kelas inputBarang

#### Sequence Diagram



Gambar 14 sequence diagram inputBarang

### Use Case #4<updateBarang>

*Skenario Use Case #4*

Tabel 12 usecase scenario updateBarang

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : Update Barang |
| Actor : Penjual |
| Pre-Condition : User ingin mengedit barang |
| Post-Condition : User sudah mengedit barang |
| Deskripsi : Untuk mengedit data barang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab penjual |  |
| 2. Mengklik data barang |  |
|  | 3. Menampilkan kelola barang |
| 4. Memilih update barang |  |
|  | 5. keluar tampilan update |
| 6.mengedit barang |  |
| 7. Menekan tombol simpan |  |
|  | 8. Memproses data yang telah disimpan dan masuk ke database |
|  | 9. Menampilkan data yang sudah di simpan |
| 10. Mengecek Kembali data yang disimpan |  |

#### Perancangan Antarmuka Usecase #4



Gambar 15 antarmuka updateBarang

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 13 identifikasi antarmuka layar updateBarang

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 4 | *Halaman updateBarang* | Page ini berisi halaman updateBarang dari menu Update Barang |

*Page updateBarang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu penjual |
| *ButtonUploadFoto* | *Button* | *UPLOAD* | Jika diklik, akanmengaktifkan FunctionuploadPhoto |
| *txtNamaBr* | *Textbox* | *Nama Barang* | Isi Teks yang disimpan pada File c\_updateBarang |
| *txtMinBid* | *Textbox* | *MIN Bid* | Isi Teks yang disimpan pada File c\_updateBarang |
| *ButtonUpdate* | *Button* | *update* | Jika diklik akan menjalankan c\_updateBarang |

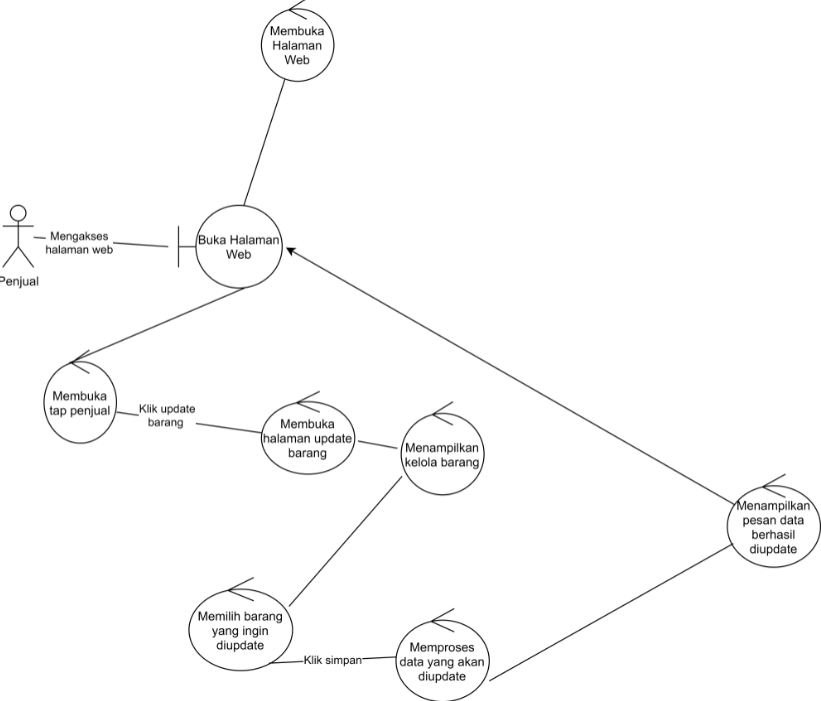
#### Identifikasi Object Baru

Tabel 14 identifikasi objek baru updateBarang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |

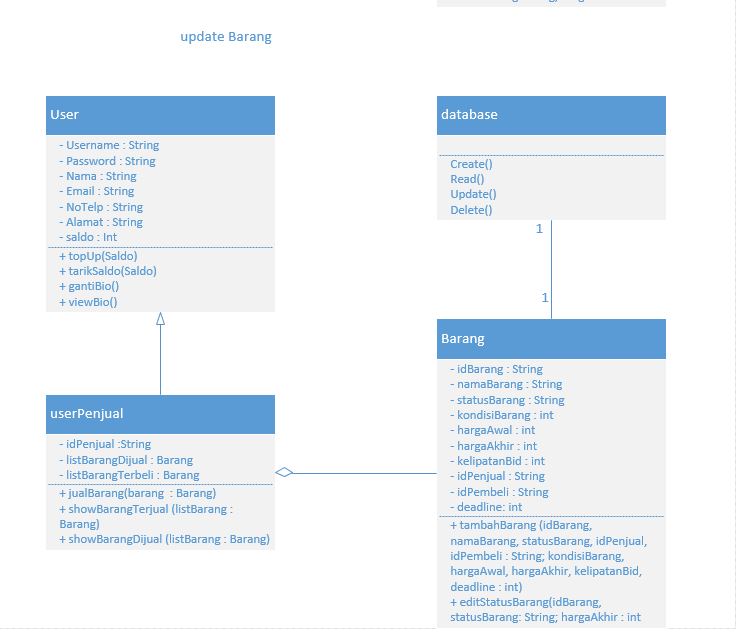
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



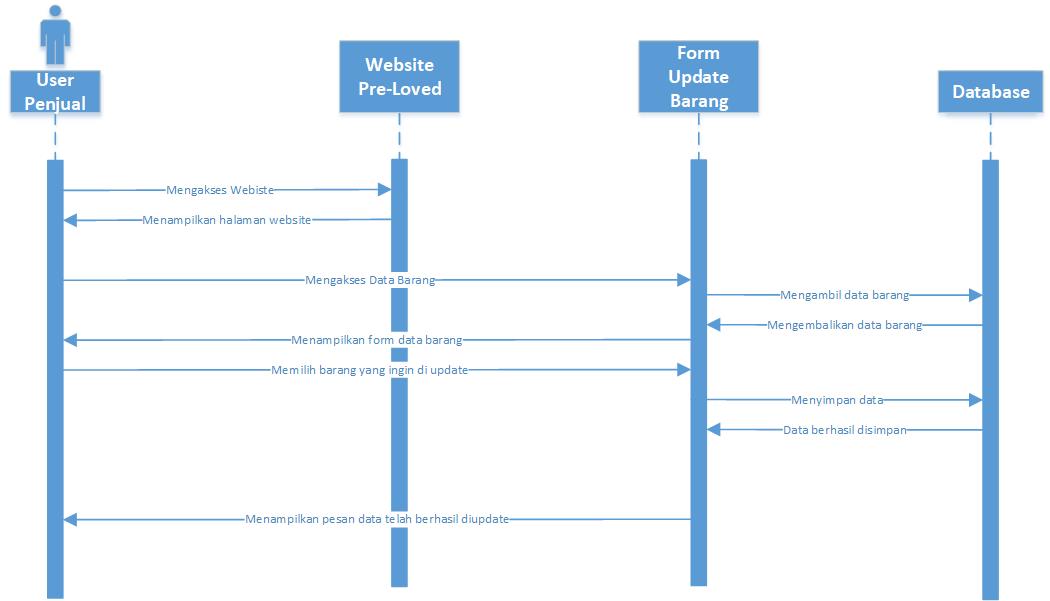
Gambar 16 robustness diagram updateBarang

#### Diagram Kelas

**

Gambar 17 diagram kelas updateBarang

#### Sequence Diagram



Gambar 18 sequence diagram updateBarang

### Use Case #5 <hapusBarang>

*Skenario Use Case #5*

Tabel 15 use case scenario hapusBarang

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : Hapus Barang |
| Actor : Penjual |
| Pre-Condition : User ingin menghapus barang |
| Post-Condition : User sudah menghapus barang |
| Deskripsi : Untuk menghapus data barang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab penjual |  |
| 2. Mengklik data barang |  |
|  | 3. Menampilkan kelola barang |
| 4. Memilih hapus barang |  |
|  | 5. keluar tampilan hapus |
| 6. menghapus barang |  |
| 7. Menekan tombol simpan |  |
|  | 8. Memproses data yang telah disimpan dan masuk ke database |
|  | 9. Menampilkan data yang sudah di simpan |
| 10. Mengecek Kembali data yang disimpan |  |

#### Perancangan Antarmuka Usecase #5



Gambar 19 antarmuka hapusBarang

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 16 identifikasi antarmuka hapusBarang

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 5 | *Halaman hapusBarang* | Page ini berisi halaman hapusBarang dari menu Barang |

*Page hapusBarang*

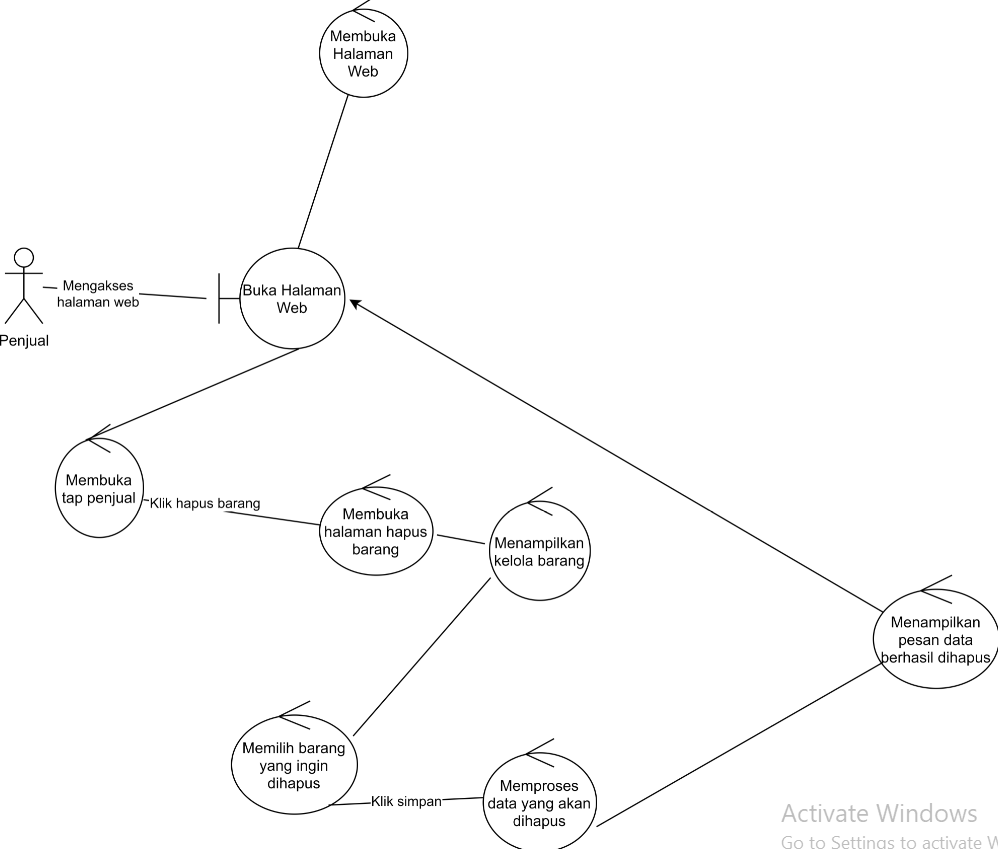
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu penjual |
| *ButtonHapusBr* | *Button* | *Hapus Barang* | Jika diklik, akanmengaktifkan FunctionhapusBarang |
| *ButtonUpdateBr* | *Button* | *Update Barang* | Jika diklik, akanmembuka menu update barang |
| *TxtsearchBarang* | *Textbox* | *SEARCH BARANG* | Jika di isiakanmenjalankanfungsiSearchBarang |

#### Identifikasi Object Baru

Tabel 17identifikasi objek baru hapusBarang

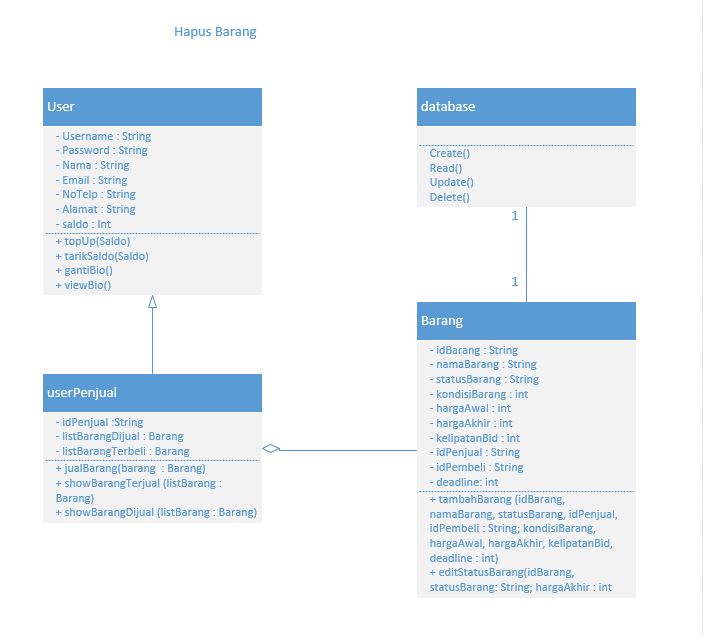
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |

#### Robustness Diagram



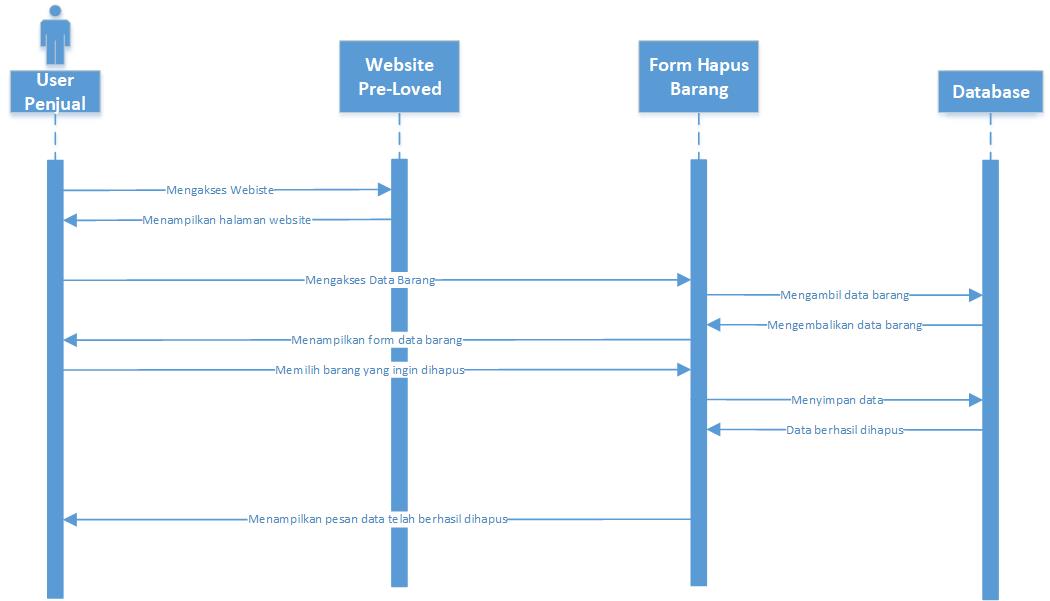
Gambar 20 robustness diagram hapusBarang

#### Diagram Kelas

**

Gambar 21 diagram kelas hapusBarang

#### Sequence Diagram



Gambar 22sequence diagram hapusBarang

### Use Case #6 <viewBarangDijual untuk Penjual>

*Skenario Use Case #6*

Tabel 18 use case scenario viewBarangDijual

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : View Barang dijual |
| Actor : Penjual |
| Pre-Condition : User ingin melihat barang |
| Post-Condition : User sudah melihat barang |
| Deskripsi : Untuk melihat data barang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab penjual |  |
| 2. Mengklik data barang |  |
|  | 3. Menampilkan kelola barang |
| 4. Memilih lihat barang dijual |  |
|  | 5. keluar tampilan list barang dijual |
| 6. melihat data barang dijual |  |

#### Perancangan Antarmuka Usecase #6



Gambar 23 antar muka viewBarangDijual

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 19 identifikasi antarmuka layar viewBarangDijual

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 6 | *Halaman viewBarangDijual* | Page ini berisi halaman viewBarangDijual dari menu Barang |

*Page viewBarangDijual*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke meu penjual |
| *ButtonHapusBr* | *Button* | *Hapus Barang* | Jika diklik, akanmengaktifkan FunctionhapusBarang |
| *ButtonUpdateBr* | *Button* | *Update Barang* | Jika diklik, akanmembuka menu update barang |
| *TxtsearchBarang* | *Textbox* | *SEARCH BARANG* | Jika di isiakanmenjalankanfungsiSearchBarang |

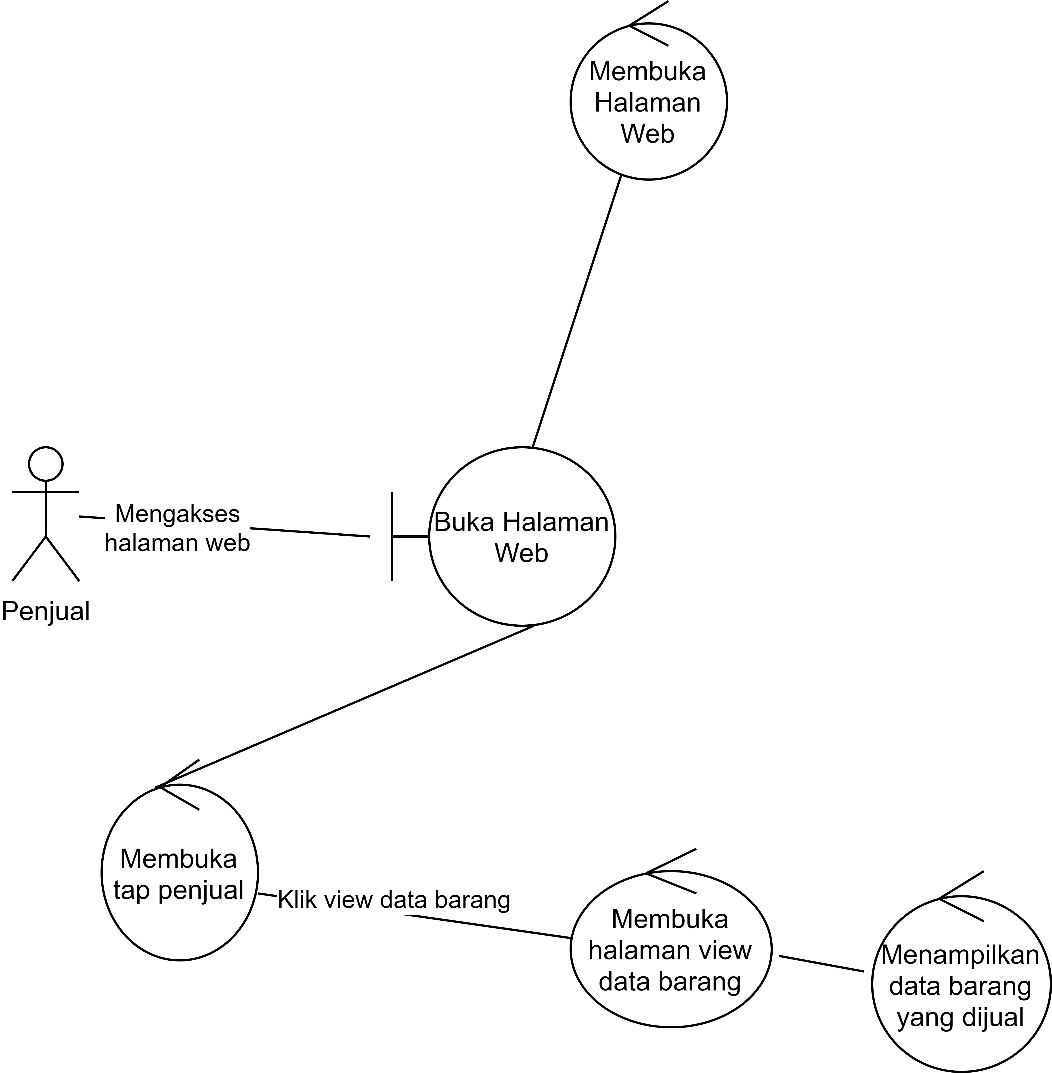
#### Identifikasi Object Baru

Tabel 20 identifikasi objek baru viewBarangDijual

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |

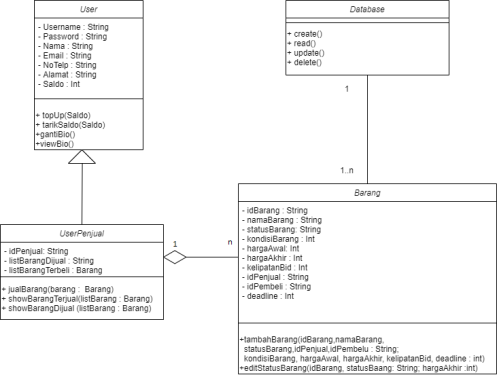
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



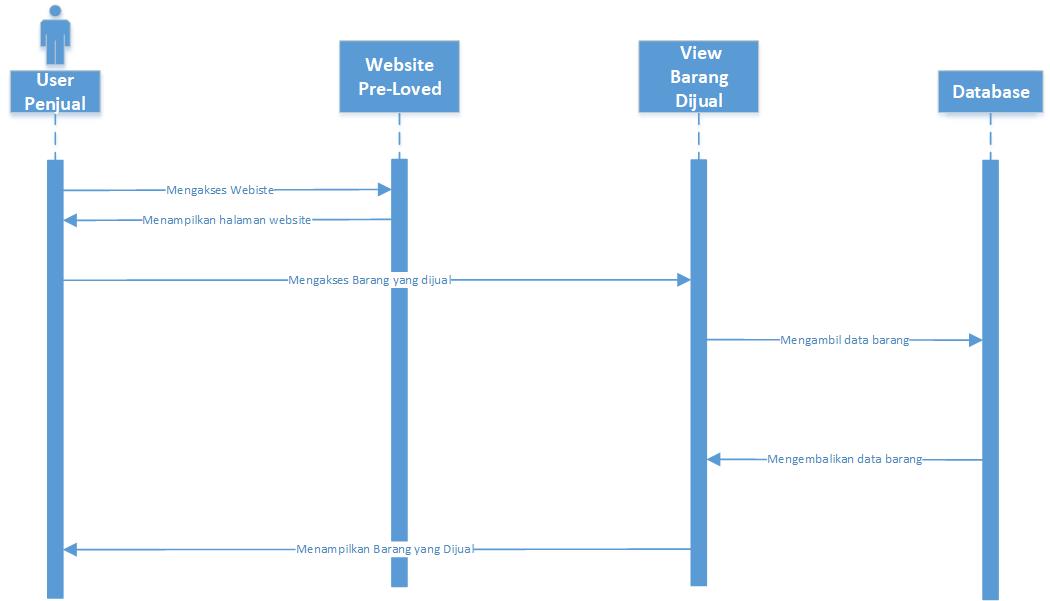
Gambar 24 robustness diagram viewBarangDijual

#### Diagram Kelas



Gambar 25 diagram kelas viewBarangDijual

#### Sequence Diagram



Gambar 26 sequence diagram viewBarangDijual

### Use Case #7 <viewBarangDijual untuk Pembeli>

*Skenario Use Case #7*

Tabel 21 usecase scenario viewBarangDijualP

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : View Barang dijual |
| Actor : Pembeli |
| Pre-Condition : User ingin melihat barang |
| Post-Condition : User sudah melihat barang |
| Deskripsi : Untuk melihat data barang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pembeli |  |
| 2. Mengklik data barang lelang |  |
|  | 3. Menampilkan kelola barang |
| 4. Memilih lihat barang dijual |  |
|  | 5. keluar tampilan list barang dijual |
| 6. melihat data barang dijual |  |

#### Perancangan Antarmuka Usecase #7



Gambar 27 antar muka viewBarangDijualP

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 22 identifikasi antarmuka layar viewBarangDijualP

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 7 | *Halaman viewBarangDijual* | Page ini berisi halaman viewBarangDijual dari menu Barang pada pembeli |

*Page viewBarangDijual*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pembeli |
| *txtBidBarang* | *Textbox* | *Bid Barang* | Isi Teks yang disimpan pada File c\_lelangBarang |
| *txtHargaBr* | *Textbox* | *Harga Tertinggi* | Isi Teks yang disimpan pada File c\_lelangBarang |
| *TxtsearchBarang* | *Textbox* | *SEARCH BARANG* | Jika di isiakanmenjalankanfungsiSearchBarang |
| *ButtonBid* | *Button* | *Bid* | Jika di klik akan menjalankan c\_bidBarang |

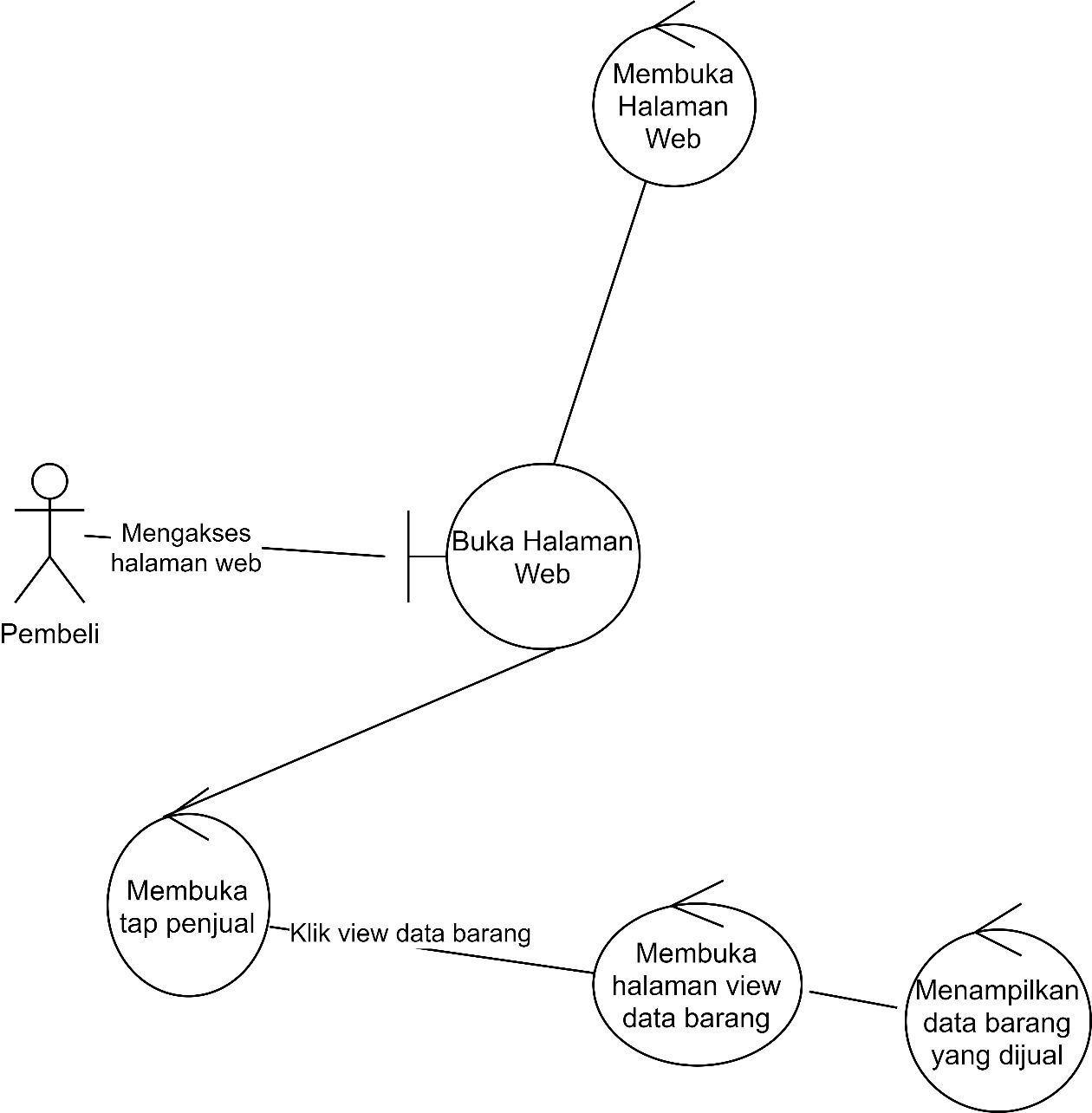
#### Identifikasi Object Baru

Tabel 23 identifikasi objek baru viewBarangDijualP

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(pembelil)* | *User(pembeli)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |
| *4/* | *Harga Tertinggi* | *Harga Tertinggi* |
| *5.* | *Bid* | *Bid* |

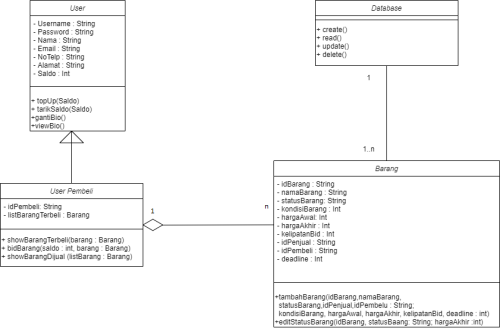
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



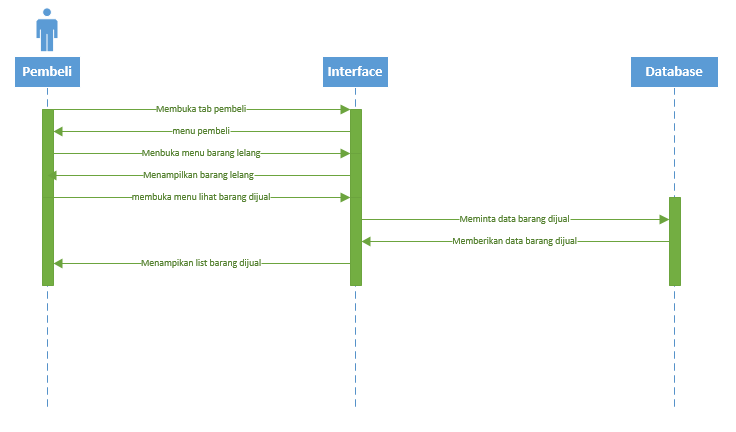
Gambar 28 robustness diagram viewBarangDijualP

#### Diagram Kelas



Gambar 29 diagram kelas viewBarangDijualP

#### Sequence Diagram



Gambar 30 sequence diagram viewBarangDijualP

### Use Case #8 <input bidBarang untuk Pembeli>

*Skenario Use Case #8*

Tabel 24 use case scenario bidBarang

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : input bid barang |
| Actor : Pembeli |
| Pre-Condition : User ingin bid barang |
| Post-Condition : User sudah melakukan bid barang |
| Deskripsi : Untuk bid barang yang dilelang |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pembeli |  |
| 2. Mengklik bid barang |  |
|  | 3. Menampilkan barang lelang |
| 4. Memilih lihat barang lelang |  |
|  | 5. keluar tampilan untuk bid barang |
| 6. melakukan bid pada barang tersebut |  |

#### Perancangan Antarmuka Usecase #8



Gambar 31 antarmuka bidBarang

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 25 identifikasi antar muka layar bidBarang

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 8 | *Halaman inputBidBarang* | Page ini berisi halaman inputBidBarang dari menu Beli Barang |

*Page inputBidBarang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pembeli |
| *txtBidBarang* | *Textbox* | *Bid Barang* | Isi Teks yang disimpan pada File c\_beliBarang |
| *txtHargaBr* | *Textbox* | *Harga Tertinggi* | Isi Teks yang disimpan pada File c\_beliBarang |
| *TxtsearchBarang* | *Textbox* | *SEARCH BARANG* | Jika di isiakanmenjalankanfungsiSearchBarang |
| *ButtonBid* | *Button* | *Bid* | Jika di klik akan menjalankan c\_bidBarang |

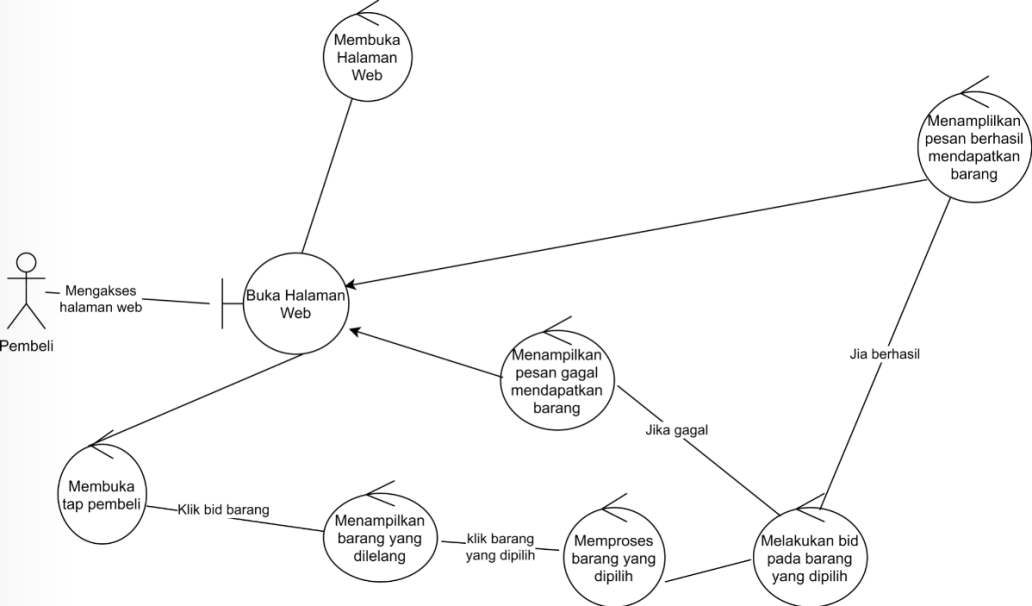
#### Identifikasi Object Baru

Tabel 26 identifikasi objek baru bidBarang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(pembelil)* | *User(pembeli)* |
| *2.* | *Barang* | *Barang* |
| *3.* | *Database* | *Database* |
| *4.* | *Harga Tertinggi* | *Harga Tertinggi* |
| *5.* | *Bid* | *Bid* |

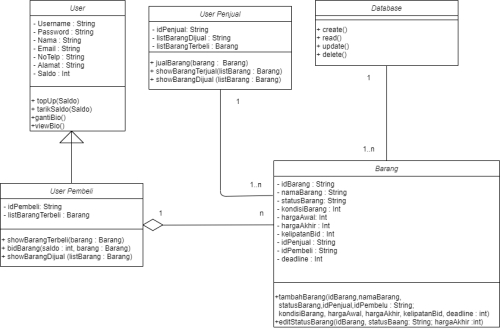
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



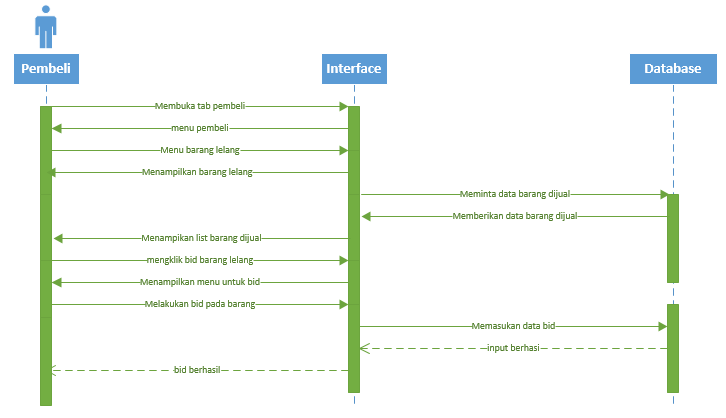
Gambar 32 robustness diagram bidBarang

#### Diagram Kelas

**

Gambar 33 diagram kelas bidBarang

#### Sequence Diagram



Gambar 34 sequence diagram bidBarang

### Use Case #9 <TransaksiBarang>

*Skenario Use Case #9*

Tabel 27 use case scenario transaksiBarang

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : transaksi barang |
| Actor : Pembeli |
| Pre-Condition : User ingin melakukan pembayaran barang terbeli |
| Post-Condition : User sudah melakukan pembayaran |
| Deskripsi : Untuk membayar barang yang telah dimenangkan |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pembeli |  |
| 2. Mengklik bayar barang |  |
|  | 3. Menampilkan barang yang belum terbayar |
| 4. klik tombol bayar |  |
|  | 5. jika saldo mencukupi pembayaran berhasil |

#### Perancangan Antarmuka Usecase #9



Gambar 35 antar muka transaksiBarang

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 28 identifikasi antarmuka layar transaksiBarang

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 9 | *Halaman TransaksiBarang* | Page ini berisi halaman transaksiBarang dari menu List Barang Terbeli |

*Page transaksiBarang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | Button | HOME | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pembeli |
| *txtNamaBarang* | *Textbox* | *Nama Barang* | Isi Teks yang disimpan pada File c\_beliBarang |
| *txtHargaBr* | *Textbox* | *Harga Tertinggi* | Isi Teks yang disimpan pada File c\_beliBarang |
| *buttonTransaksi* | *Button* | *Bayar* | Jika di isiakanmenjalankanfungsiTransaksiBarang |

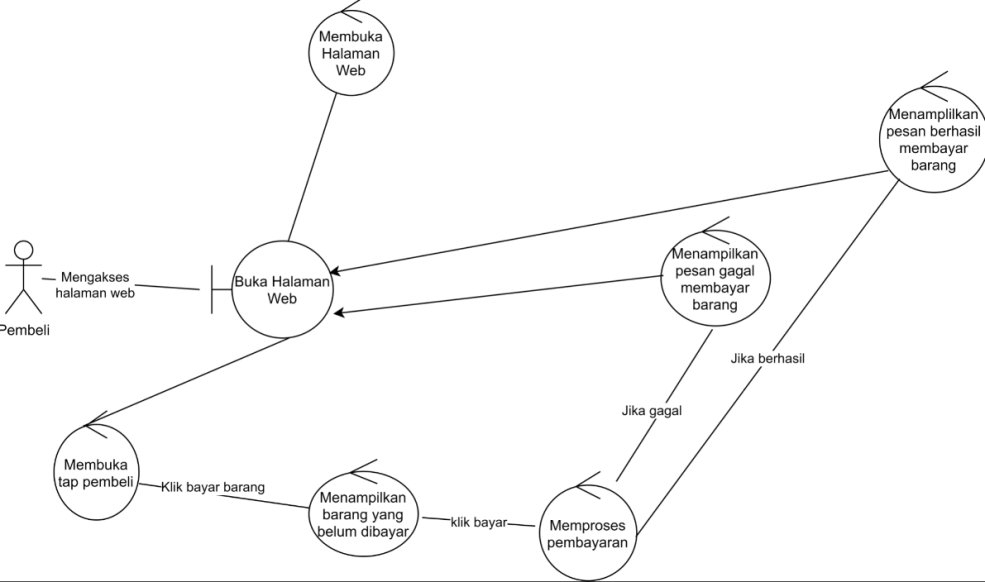
#### Identifikasi Object Baru

Tabel 29 identifikasi objek baru transaksiBarang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(pembeli)* | *User(pembeli)* |
| *2.* | *Database* | *Database* |
| *3.* | *Bayar* | *Bayar* |

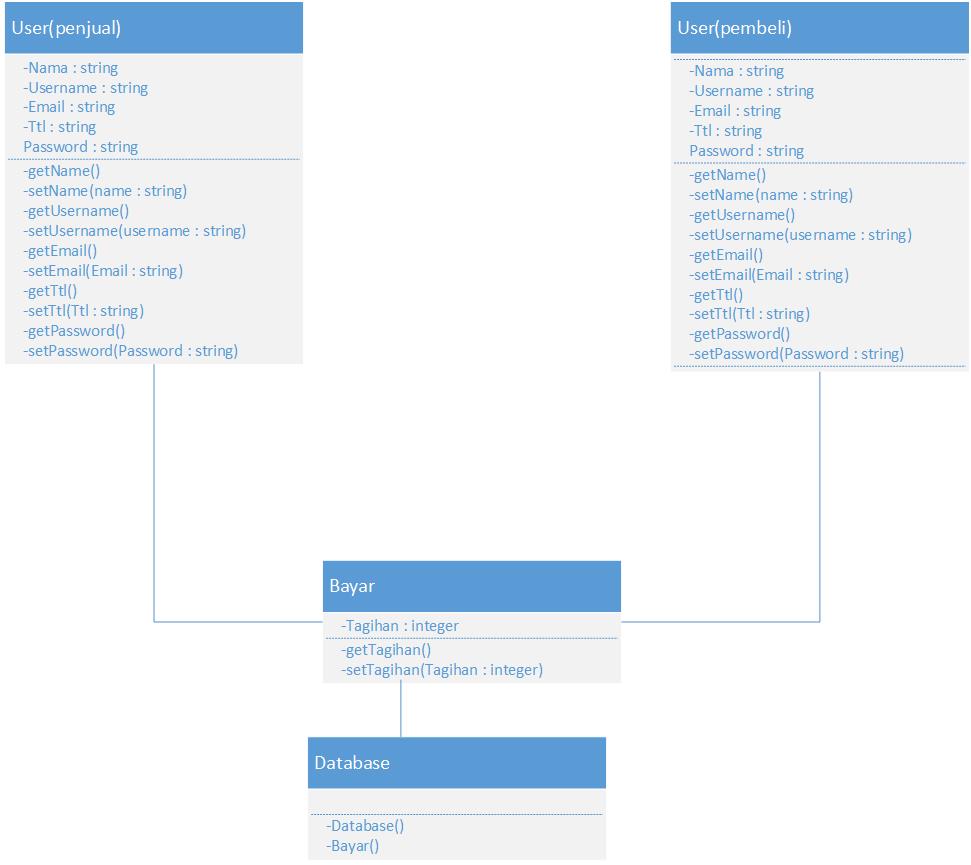
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



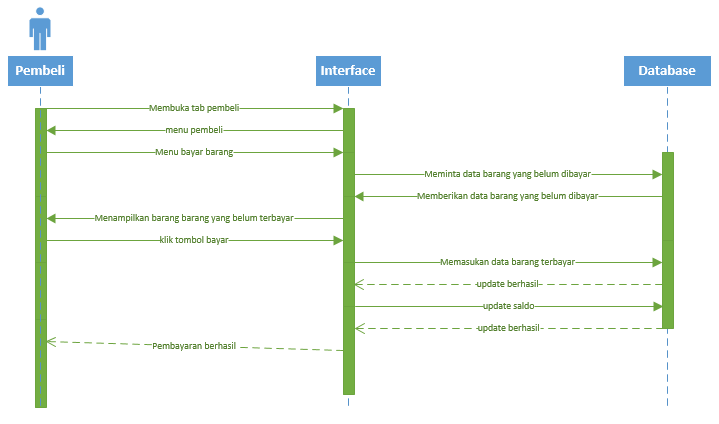
Gambar 36 robustness diagram transaksiBarang

#### Diagram Kelas

**

Gambar 37 diagram kelas transaksiBarang

#### Sequence Diagram



Gambar 38 sequence diagram transaksiBarang

### Use Case #10 <TopUp Saldo>

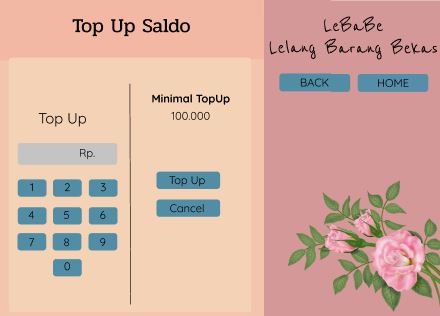
*Skenario Use Case #10*

Tabel 30 use case scenario topUp

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : top up |
| Actor : user |
| Pre-Condition : User ingin mengisi saldo |
| Post-Condition : User sudah mengisi saldo |
| Deskripsi : Untuk mengisi saldo |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pengguna |  |
| 2. Mengklik top up |  |
|  | 3. Menampilkan menu top up |
| 4. memasukan nominal top up |  |
|  | 5. keluar tampilan top up berhasil |

#### Perancangan Antarmuka Usecase #10



Gambar 39 antar muka topUp

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 31 identifikasi antarmuka layar topUp

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 10 | *Halaman Saldo* | Page ini berisi halaman topUpSaldo dari menu Saldo |

*Page topUpSaldo*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonTopUp* | Button | Top Up | Jika diklikakanmenjalankanfungsitopUp |
| *ButtonCancel* | *Button* | *Cancel* | Jika diklikakanbalik ke home pembeli |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu homr |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pengguna |
| *TxtNominalSaldoDiambil* | *Textbox* | *Masukan Nominal Barang* | Isi Teks yang disimpan pada File c\_saldo |
| *TxtNominalSaldo* | *Textbox* |  | Akan muncul nominal saldo saat ini |

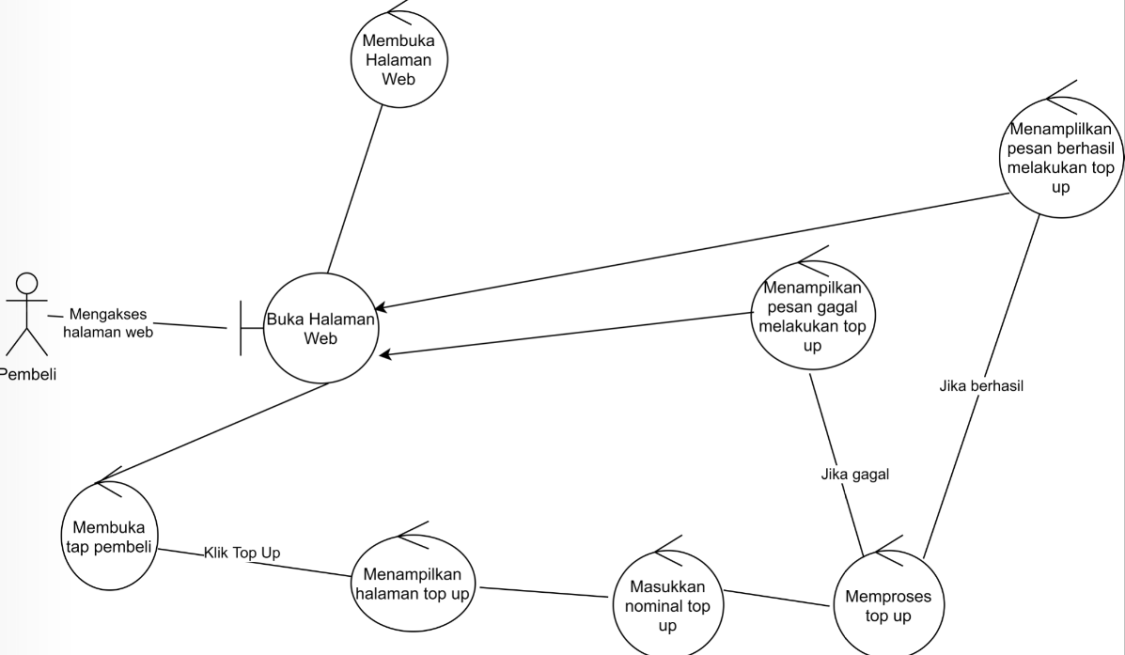
#### Identifikasi Object Baru

Tabel 32 identifikasi objek baru

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(pembeli)* | *User(pembeli)* |
| *2.* | *Database* | *Database* |
| *3.* | *Top Up* | *Top Up* |

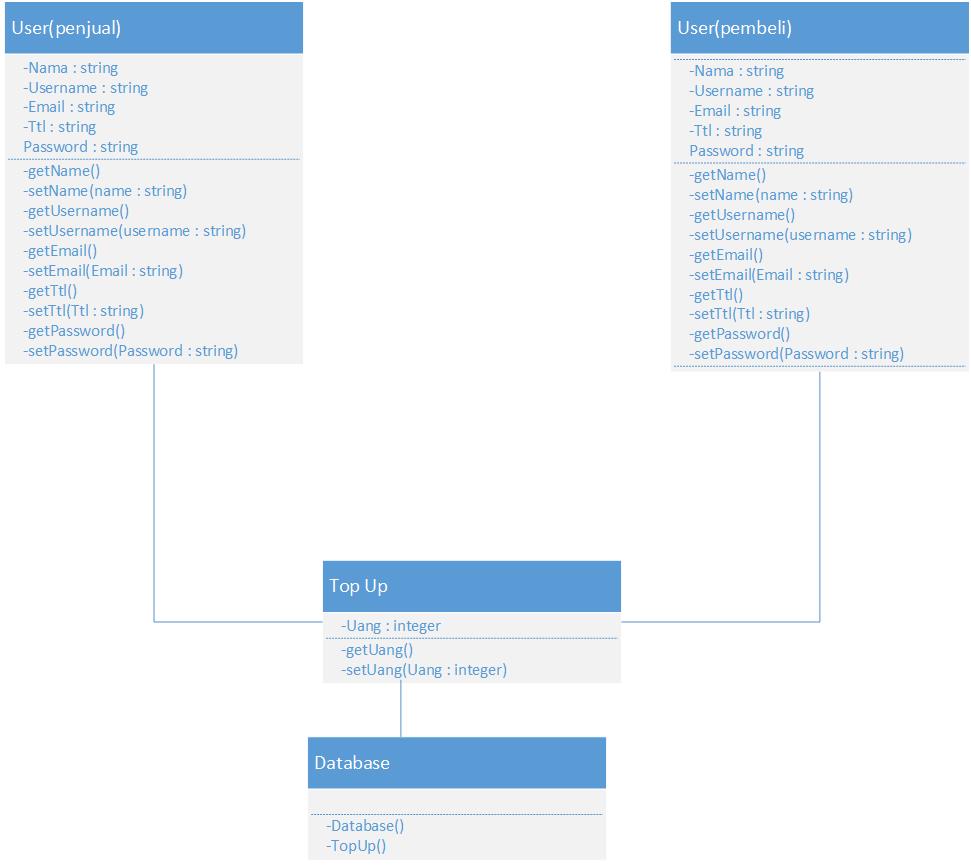
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



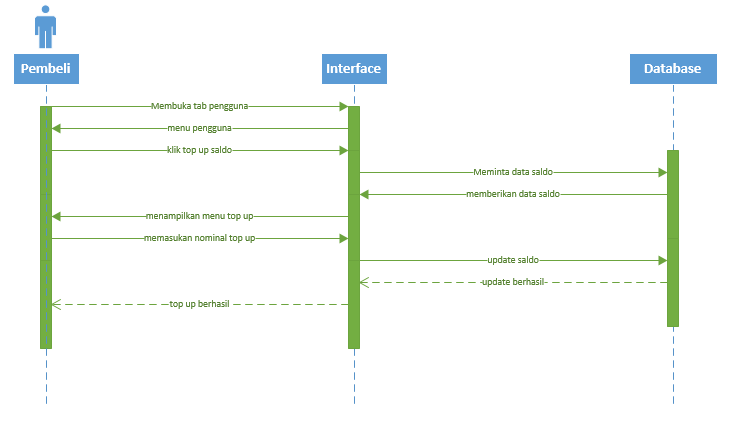
Gambar 40 robustness diagram topUp

#### Diagram Kelas

**

Gambar 41 diagram kelas topUp

#### Sequence Diagram



Gambar 42 sequence diagram topUp

### Use Case #11<Tarik Saldo>

*Skenario Use Case #11*

Tabel 33 use case scenario tarikSaldo

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : tarik saldo |
| Actor : user |
| Pre-Condition : User ingin menarik saldo |
| Post-Condition : User sudah menarik saldo |
| Deskripsi : Untuk menarik saldo |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pengguna |  |
| 2. Mengklik tarik saldo |  |
|  | 3. Menampilkan menu tarik saldo |
| 4. memasukan nominal yang ditarik |  |
|  | 5. keluar tampilan penarikan saldo berhasil |

#### Perancangan Antarmuka Usecase #11



Gambar 43 antar muka tarikSaldo

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 34 identifikasi antarmuka layar tarikSaldo

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 11 | *Halaman Saldo* | Page ini berisi halaman topUpSaldo dari menu Saldo |

*Page tarikSaldo*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonAccept* | Button | Accept | Jika diklikakanmenjalankanfungsiTarikSaldo |
| *ButtonCancel* | Button | Cancel | Jika diklikakanbalik ke home pembeli |
| *ButtonTarikSaldo* | *Button* | *Tarik Saldo* | Jika diklikakanmenjalankanfungsitarikSaldo |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu homr |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pengguna |
| *TxtNominalSaldoDiambil* | *Textbox* | *Masukan Nominal Barang* | Isi Teks yang disimpan pada File c\_saldo |
| *TxtNominalSaldo* | *Textbox* |  | Akan muncul nominal saldo saat ini |

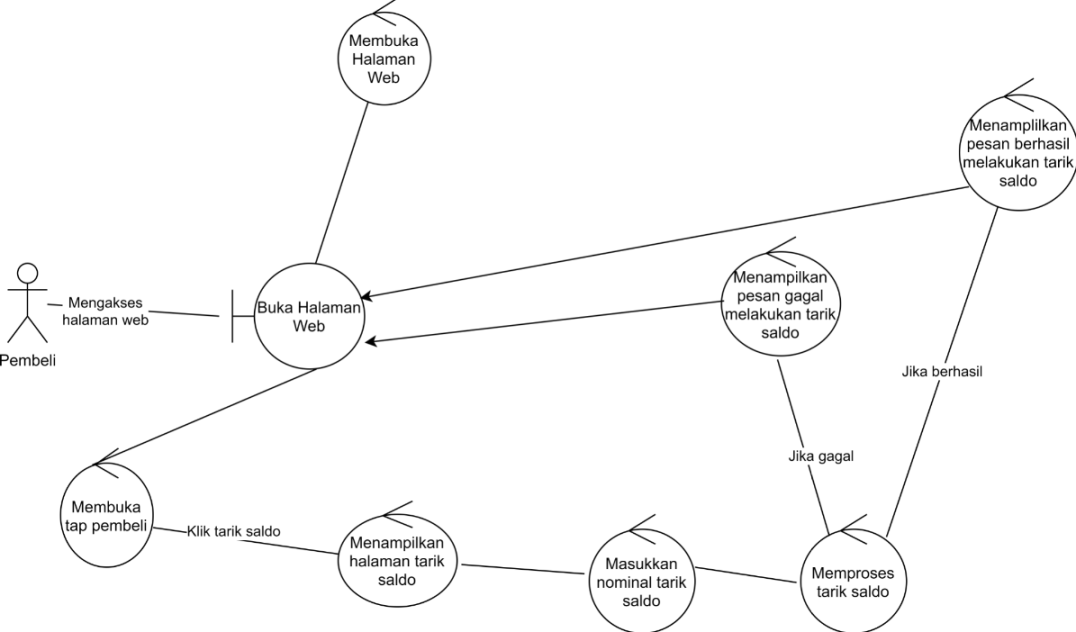
#### Identifikasi Object Baru

Tabel 35 identifikasi objek baru

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *User(penjual)* | *User(penjual)* |
| *3.* | *Tarik Saldo* | *Tarik Saldo* |
| *4.* | *Database* | *Database* |

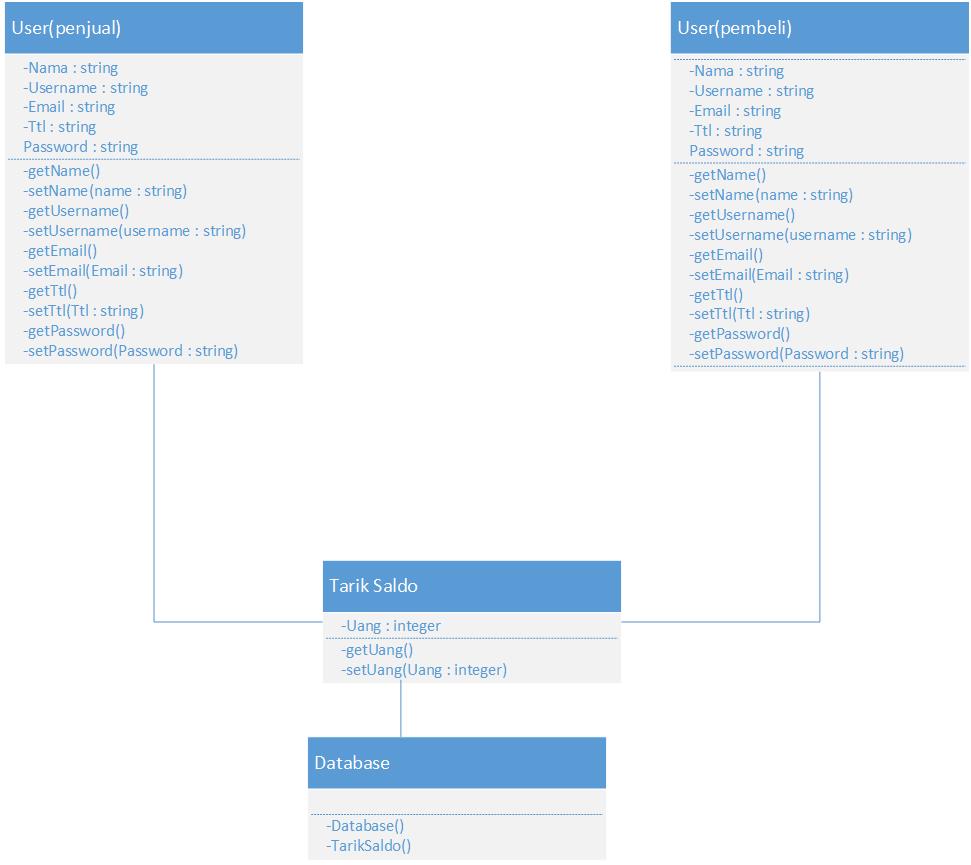
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



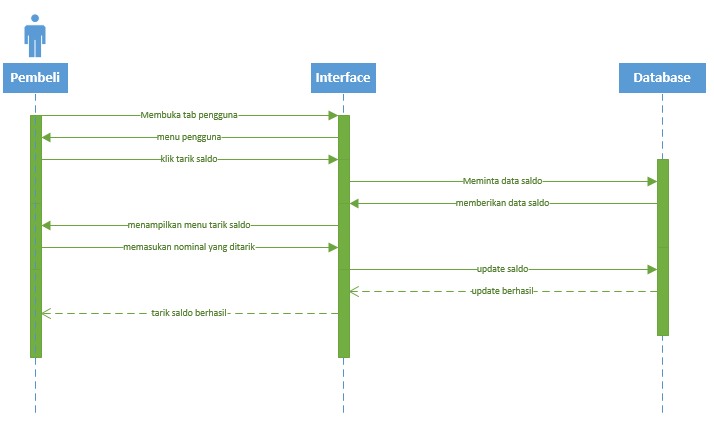
Gambar 44 robustness diagram tarikSaldo

#### Diagram Kelas

**

Gambar 45 diagram kelas tarikSaldo

#### Sequence Diagram



Gambar 46 sequence diagram tarikSaldo

### Use Case #12<view Biodata>

*Skenario Use Case #12*

Tabel 36 use case scenario viewBiodata

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : view biodata |
| Actor : user |
| Pre-Condition : User ingin melihat biodata |
| Post-Condition : User sudah melihat biodata |
| Deskripsi : Untuk melihat biodata |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pengguna |  |
| 2. Mengklik view biodata |  |
|  | 3. Menampilkan biodata pengguna |

#### Perancangan Antarmuka Usecase #12



Gambar 47 antar muka viewBiodata

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 37 identifikasi antarmuka layar viewBiodata

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 12 | *Halaman Profile* | Page ini berisi halaman profile dari menu profile |

*Page Profile*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu home |
| *ButtonBack* | *Button* | *BACK* | Jika diklikakanmengembalikanke menu pengguna |
| *TxtProfile* | *Textbox* | *Nama,username,email,ttl* | Akan memunculkan biodata |
| *buttonTopUp* | *Button* | *Top Up* | Jika di klik akan ke view top up |
| *Button tariksaldo* | *Button* | *Tarik Saldo* | Jika diklik akan ke view tarik saldo |
| *Button editprofile* | *Button* | *Edit profile* | Jika di klik akan ke view edit profile |

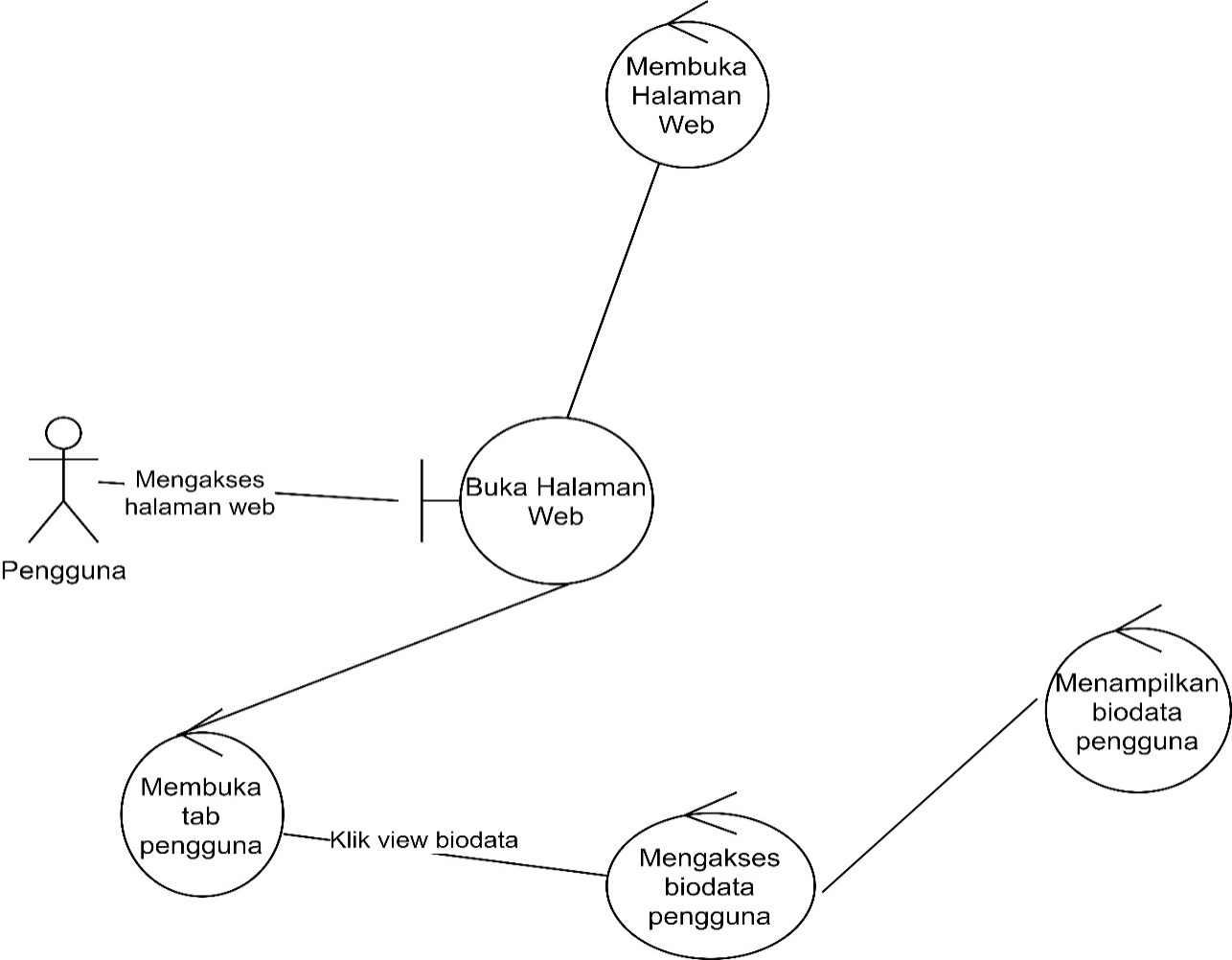
#### Identifikasi Object Baru

Tabel 38 identifikasi objek baru viewBiodata

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *User(pembeli)* | *User(pembeli)* |
| *3.* | *Database* | *Database* |

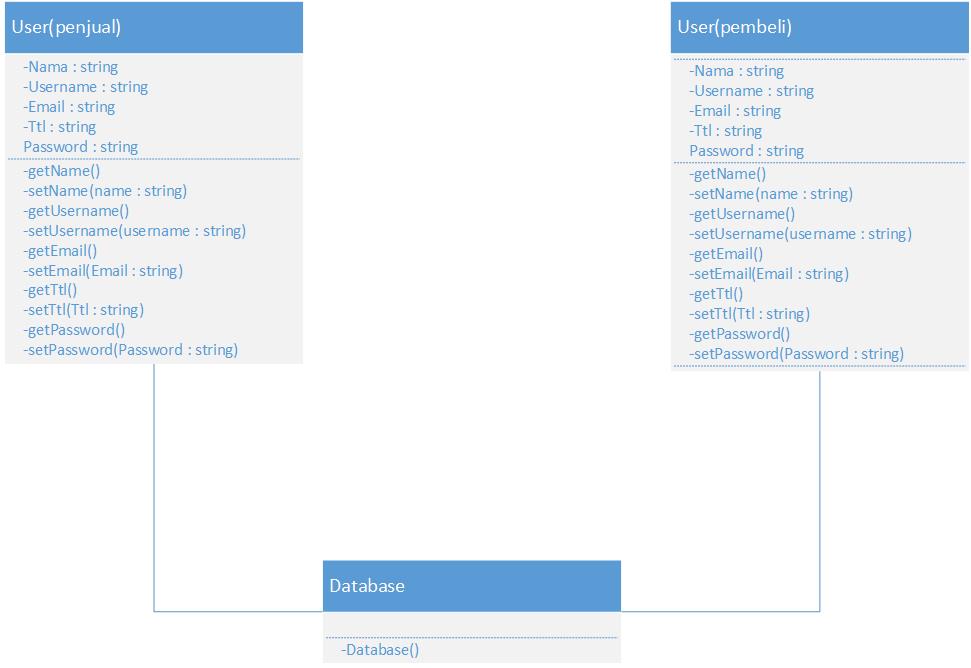
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



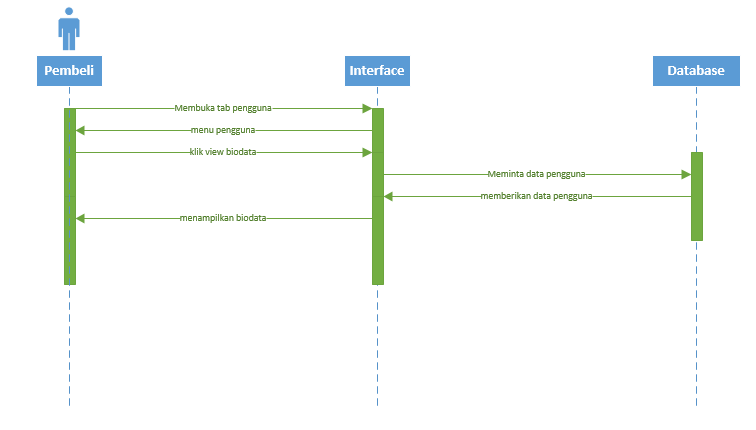
Gambar 48 robustness diagram viewBiodata

#### Diagram Kelas

**

Gambar 49 diagram kelas viewBiodata

#### Sequence Diagram



Gambar 50 sequence diagram viewBiodata

### Use Case #13<view editBiodata>

*Skenario Use Case #13*

Tabel 39use case scenario editBiodata

|  |
| --- |
| **Use case Skenario** |
|  |
| Use Case : edit biodata |
| Actor : user |
| Pre-Condition : User ingin mengubah biodata |
| Post-Condition : User sudah mengubah biodata |
| Deskripsi : Untuk mengubah biodata |

|  |  |
| --- | --- |
| **Actor** | **Activity** |
| 1. Membuka tab pengguna |  |
| 2. Mengklik edit biodata |  |
|  | 3. Menampilkan menu edit biodata |
| 4. Mengisi biodata yang ingin diganti |  |
|  | 5. mengganti biodata |
|  | 6. Menampilkan pesan bahwa biodata telah diganti |

#### Perancangan Antarmuka Usecase #13



Gambar 51 antar muka editBiodata

#### Tabel Identifikasi Antarmuka / Layar / Page

Tabel 40 identifikasi antarmuka layar editBiodata

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Page 13 | *Halaman Update Profile* | Page ini berisi halaman update profile dari menu profile |

*Page update Profile*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *ButtonUpdate* | Button | Submit | Jika diklik, akanmengaktifkan FunctionupdateProfile |
| *ButtonHome* | *Button* | *HOME* | Jika diklikakanmengembalikanke menu home |
| *txtNama* | *Textbox* | *Nama* | Isi Teks yang disimpan pada File c\_updateProfile |
| *txtUsername* | *Textbox* | *Username* | Isi Teks yang disimpan pada File c\_updateProfile |
| *txtEmail* | *Textbox* | *Email* | Isi Teks yang disimpan pada File c\_updateProfile |
| *txtTtl* | *Textbox* | *Ttl* | Isi Teks yang disimpan pada File c\_updateProfile |
| *txtPassword* | *Textbox* | *password* | Isi Teks yang disimpan pada File c\_updateProfile |

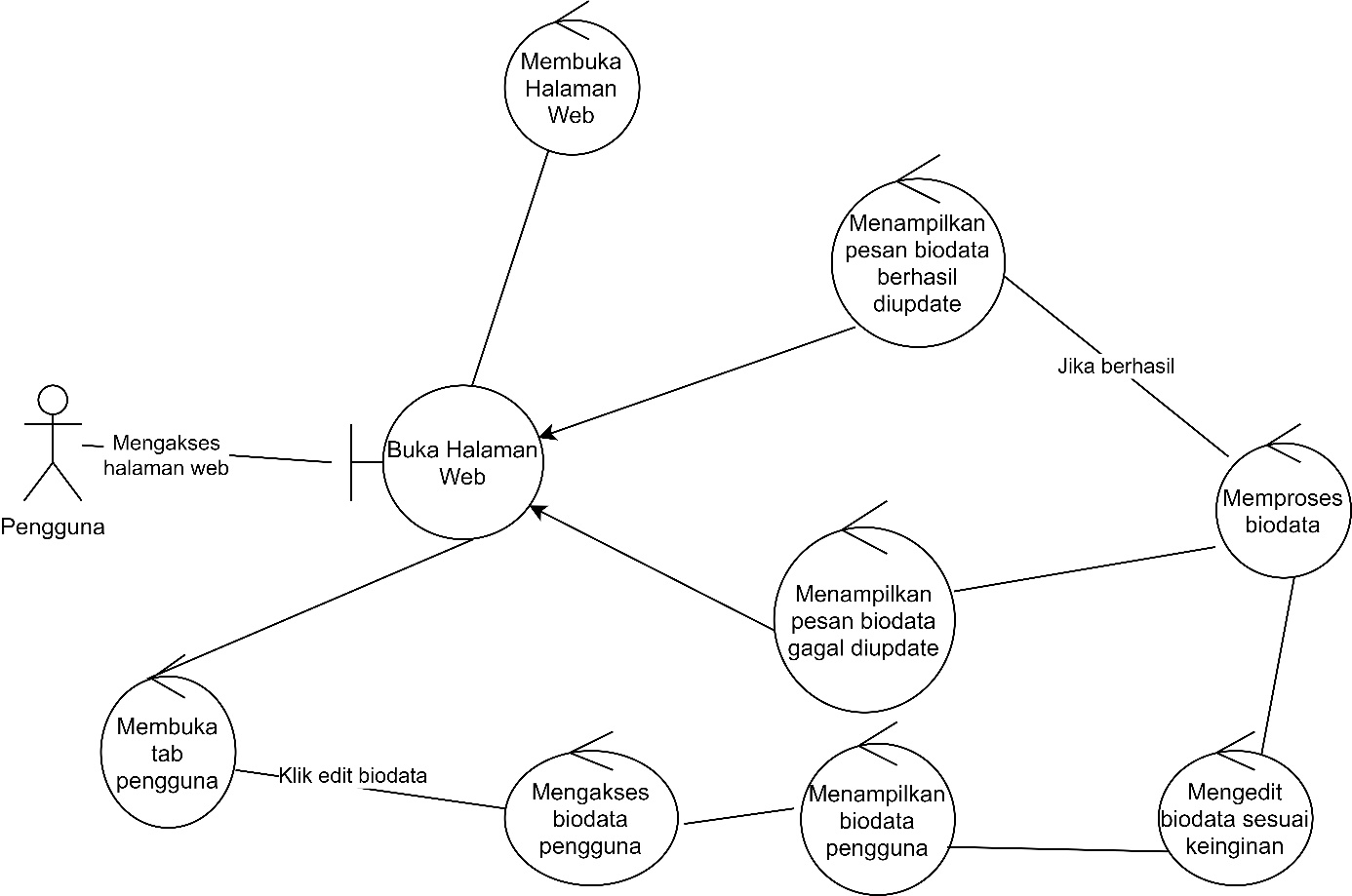
#### Identifikasi Object Baru

Tabel 41 identifikasi objek baru editBiodata

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1.* | *User(penjual)* | *User(penjual)* |
| *2.* | *User(pembeli)* | *User(pembeli)* |
| *3.* | *Database* | *Database* |

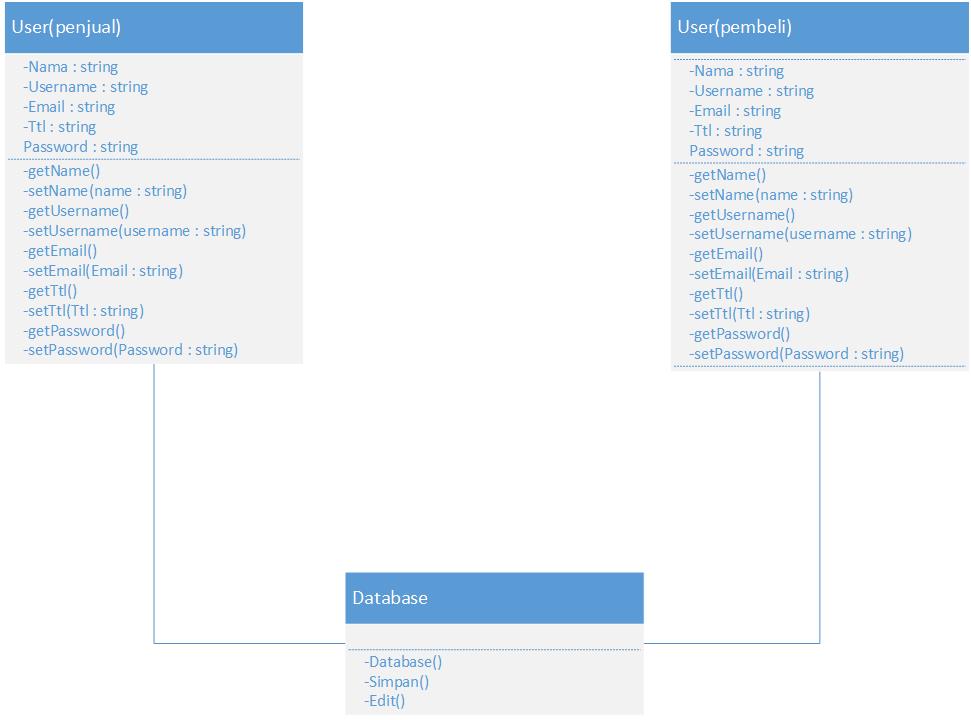
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram



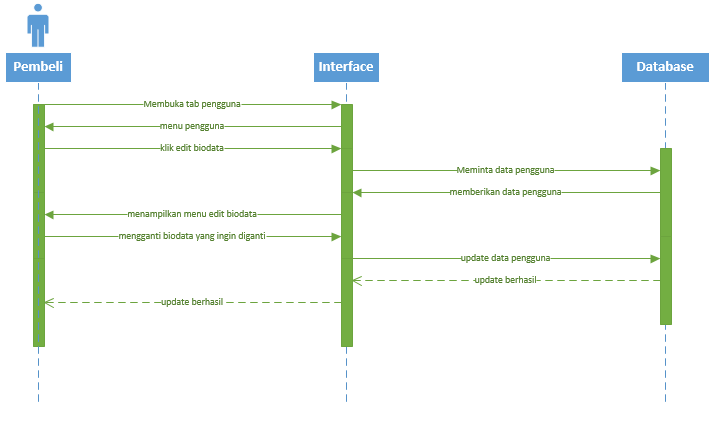
Gambar 52 robustness diagram editBiodata

#### Diagram Kelas

**

Gambar 53 diagram kelas editBiodata

#### Sequence Diagram



Gambar 54 sequence diagram editBiodata

# Perancangan Detil

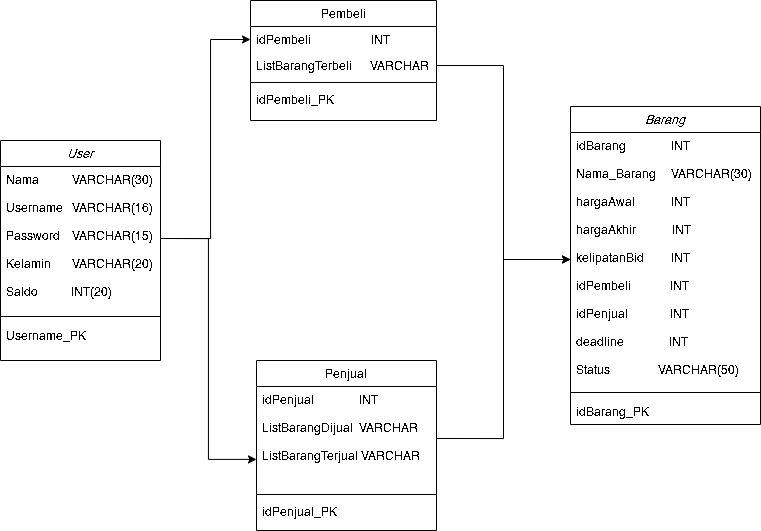
#### Perancangan Detil Kelas

**TABEL KELAS :**

Tabel 42 perancangan detil kelas

|  |  |  |  |
| --- | --- | --- | --- |
| **ID Kelas** | **Nama Kelas Perancangan** | **Atribute (visibility)** | **Method / Operation** |
| K01 | User | -Nama (private)  -Username (private)  -Password (private)  -Kelamin(private)  -Saldo(private) | +getUsername()  +setUsername()  +getPassword()  +setPassword() +setNama()  +getNama()  +getKelamin()  +TopUpSaldo() +TarikSaldo() |
| K02 | Penjual | -idPembeli  -ListBarangTerbeli | +setIdPembeli()  +getIdPembeli()  +showBarangTerbeli()  +BidBarang() |
| K03 | Pembeli | -idPenjual  -ListBarangDijual  -ListBarangTerjual | +getIdPenjual()  +setIdPenjual  +JualBarang()  +showBarangTerjual() |
| K04 | Barang | -NamaBarang  -idBarang  -hargaAwal  -hargaAkhir  -kondisiBarang  -status  -kelipatanBid  -deadline  -idPenjual  -idPembeli | +setNamaBarang  +getNamaBarang  +setIdBarang  +getIdBarang  +setKondisiBarang  +getKondisiBarang  +setHargaAkhir  +getHargaAkhir  +setHargaAwal  +getHargaAwal |

#### Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel)

Skema Relasi database yang digunakan

Gambar 55 perancangan kelas persistensi

#### Perancangan Algoritma

*Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.*

#### Algoritma #1

*Nama Kelas :Registrasi*

*Nama Operasi :Registrasi*

*Algoritma :*

|  |
| --- |
| **Function** registrasi*($user*)  Kamus  usernameNew =String;  passwordNew = String;  emailNew= String;  kelaminNew = String;  SaldoNew = int;  Algoritma  Begin  If (btnRegistrasi) {  Registrasi () {  Username = usernameNew ;  Password = passwordNew;  Email = emailNew;  Kelamin = kelaminNew;  Saldo = SaldoNew;  }  }  End; |

#### Algoritma #2

*Nama Kelas :Login*

*Nama Operasi :Login*

*Algoritma :*

|  |
| --- |
| **Function** login*($user*)  Kamus  usernameNew =String;  passwordNew = String;  Algoritma  Begin  If (btnLogin) {  Login () {  If (username == ‘$username’) && (password == ‘$password’){  Return True  }else  Return False  }  }  End; |

#### Algoritma #3

*Nama Kelas :inputBarang*

*Nama Operasi :inputBarang*

*Algoritma :*

|  |
| --- |
| Function inputBarang()  kamus  nama : string  gambar : file  minBid : int  deadline : date  algoritma  nama = input(namaBarang)  gambar = input(gambar)  minBid = input(minBid)  dealine = input(deadline)  if(btnSimpan)  //query insert to db Barang  endif  end |

#### Algoritma #4

*Nama Kelas :hapusBarang*

*Nama Operasi :hapusBarang*

*Algoritma :*

|  |
| --- |
| Function hapusBarang()  kamus  barang : Barang  idBarang : String  algoritma  barang = Q-007  foreach(barang)  output(barang)  endforeach  if(btnBarang == 'clicked')  idBarang = barang['idBarang']  endif  //query Q-018  end |

#### Algoritma #5

*Nama Kelas :viewBarangDijualPenjual*

*Nama Operasi :viewBarangDijualPenjual*

*Algoritma :*

|  |
| --- |
| Function viewBarangDijualPenjual(Username : string)  kamus  barang : Barang  algoritma  barang = //Q-010  foreach(barang)  output(barang)  endforeach |

#### Algoritma #6

*Nama Kelas :viewBarangDijualPembeli*

*Nama Operasi :viewBarangDijualPembeli*

*Algoritma :*

|  |
| --- |
| Function viewBarangDijualPembeli()  kamus  barang : Barang  algoritma  barang = //Q-011  foreach(barang)  output(barang)  endforeach  end |

#### Algoritma #7

*Nama Kelas :transaksiBarang*

*Nama Operasi :transaksiBarang*

*Algoritma :*

|  |
| --- |
| Function transaksiBarang(uPembeli, idBarang)  kamus  saldo, hargaAkhir : int  uPenjual : String  algoritma  if(tombolBayarClicked)  saldo = //Q-019  hargaAkhir = //Q-013  if(hargaAkhir <= saldo)  tarikSaldo(uPembeli, hargaAkhir)  uPenjual = //Q-013  topUp(uPenjual, hargaAkhir)  //Q-012  endif  endif  end |

#### Algoritma #8

*Nama Kelas :topUp*

*Nama Operasi :topUp*

*Algoritma :*

|  |
| --- |
| Function topUp(username, jumlah)  kamus  saldo, saldoAkhir : int  algoritma  saldo = //Q-014  saldoAkhir = saldo + jumlah  //Q-015  return true;  end |

#### Algoritma #9

*Nama Kelas :tarikSaldo*

*Nama Operasi :tarikSaldo*

*Algoritma :*

|  |
| --- |
| Function tarikSaldo(username, jumlah)  kamus  saldo, saldoAkhir : int  algoritma  saldo = Q-014  if (saldo > jumlah)  saldoAkhir = saldo - jumlah  //query Q-014  return true  else  return false  endif  end |

#### Algoritma #10

*Nama Kelas :viewBiodata*

*Nama Operasi :viewBiodata*

*Algoritma :*

|  |
| --- |
| Function viewBiodata(username)  kamus  biodata = User  algoritma  biodata = Q-016  output(biodata)  end |

#### Algoritma #11

*Nama Kelas :editBiodata*

*Nama Operasi :editBiodata*

*Algoritma :*

|  |
| --- |
| Function editBiodata(username)  kamus  biodata = User  algoritma  biodata Q-016  output(biodata)  input(newBio)  /Q-017  end |

#### Algoritma #12

*Nama Kelas :editBarang*

*Nama Operasi :editBarang*

*Algoritma :*

|  |
| --- |
| **Function**editBarang*($idBarang*)  Kamus  namaBrgNew =String;  hargaNew = String;  stokNew= integer;  fotoNew = file;  Algoritma  Begin  input(namaBarang);  Q-006  if (btnAddNewBarang){  tambahBarang();  } else {  if (namaBarang == true) {  if (btnEdit){  $idBarang;  $namaBarang = namaBrgNew;  $harga = hargaNew;  $foto = fotoNew;  } else {  hapusBarang($idBarang)  }  } else {  output(“BarangTidakTersedia”) ;  }  }  End; |

#### Algoritma #13

*Nama Kelas :bidBarang*

*Nama Operasi :bidBarang*

*Algoritma :*

|  |
| --- |
| **Function** bidBarang*($idBarang*)  Kamus  namaBrgNew =String;  hargaNew = String;  stokNew= integer;  fotoNew = file;  Algoritma  Begin  input(namaBarang);  Q-006  if (btnBidBarang){  bidBarang();  }else {  output(“Kembali ke Menu Utama”) ;  }  }  End; |

#### Perancangan Query*:*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *Select \* from User* | *Deskripsi query untukmengambilsemua daftar user yang ada.* |
| *Q-002* | *Select \* from barang where status = ‘$status’* | *Deskripsi query untukmengecek status barang* |
| *Q-003* | *Select \* from user where email=’$Email’* | *Deskripsi query untukmengecekapakah email user tersebutsudahterdaftarsebelumnya di dalam database* |
| *Q-004* | *Select \* from barang where idBarang = $idBarang* | *Deskripsi query untukmencariinformasibarangberdasarkanidBarang, apakahtelahtersediaataubelum* |
| *Q-005* | *Select password from user where username = ‘$Username’* | *Deskripsi query untukmengecekapakah username pembelicocokdengan password yang dimiliki* |
| *Q-006* | *Select \* from barang where namaBarang = $barang* | *Deskripsi query untukmengambilinformasi /data barang pada namabarangtertentu* |
| *Q-007* | *Select \* from Barang* | *Deskripsi query untuk mengambil semua daftar barang yang ada* |
| *Q-008* | *Select \* from Barang where idPembeli = ‘$idPembeli’* | *Deskripsi query untuk mengambil daftar barang berdasarkan id Pembeli* |
| *Q-009* | *Select \* from Barang where idPembeli = ‘$idPenjuali’* | *Deskripsi query untuk mengambil daftar barang berdasarkan id Penjual* |
| *Q-010* | *select \* from barang where username = $username* | *Deskripsi query untuk mengambil daftar barang berdasarkan username* |
| *Q-011* | *select \* from barang where status = 'dijual'* | *Deskripsi query untuk mengambil daftar barang*  *Berdasarkan status = dijual* |
| *Q-012* | *update Barang set status = 'terjual' where idBarang = '$idBarang* | *Deskripsi query untuk mengupdate status menjadi terjual berdsarakan idBarang* |
| *Q-013* | *select idPenjual from barang where idBarang = '$idBarang'* | *Deskripsi query untuk mengambil semua idPenjual dari barang berdsarakan idBarang* |
| *Q-014* | *select saldo from user where username = username* | *Deskripsi query untuk mengambil semua saldo dari user berdasarkan username* |
| *Q-015* | *update User set saldo = SaldoAkhir where username = username* | *Deskripsi query untuk mengupdate saldo menjadi saldoAkhir berdasarkan username di User* |
| *Q-016* | *select \* from user where username = username* | *Deskripsi query untuk mengambil data user berdasarkan username* |
| *Q-017* | *update user set*  *username = newBio['username'],*  *nama = newBio['nama'],*  *NoTelp = newBio['noTelp'],*  *alamat = newBio['alamat']*  *where username = username* | *Deskripsi query untuk mengupdate user dengan bio yang baru* |
| *Q-018* | *delete from db barang where idBarang = idBarang* | *Deskripsi query untuk menghapus barang berdasarkan idBarang* |
| *Q-019* | *select saldo from user where username = uPembeli* | *Deskripsi query untuk mengambil semua saldo berdsarkan username = username pembeli* |

# Matriks Kerunutan(Requirement Traceability Matrix)

Tabel 43 Matrik kerunutan

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Functional Requirement** | **Nama Use Case** |
| FR-01 | Fungsi ini digunakan oleh user untuk melakukan registrasi agar dapat mengakses fitur diwebsite. | Registrasi |
| FR-02 | Fungsi ini digunakan oleh  user untuk mengakses fitur di website | Login |
| FR-03 | Fungsi ini digunakan oleh user untuk melakukan penambahan saldo untuk pembelian dan bias ditarik kembali jika sudahtidak  digunakan | Top Up dan Tarik Saldo |
| FR-04 | Fungsi ini digunakan oleh user(penjual) untuk melakukan penjualan oleh penjual | MenjualBarang |
| FR-05 | Fungsi ini digunakan oleh user(penjual) untuk melakukan penglihatanstatus  barang yang user jual | View status barang |
| FR-06 | Fungsi ini digunakan oleh user(pembeli) untuk meletakan bid pada sitem pelelangan | Meletakan Bid |
| FR-07 | Fungsi ini digunakan oleh user untuk melihat daftar  barang yang sedang dilelang | View barang Lelang |
| FR-08 | Fungsi ini digunakan oleh user untuk melakukan Input  barang untuk dijual kesystem | Input Barang |
| FR-09 | Fungsi ini digunakan oleh user untuk melakukan validasi transaksi | ValidasiTransaksi |
| FR-10 | Fungsi ini digunakan oleh user untuk melakukan  pendataan barang yang dilelang | Pendataan barang lelang |

# 