**DPPL-004**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

INVENTARIS GUDANG

untuk:

Gudang Pensupply Barang

Dipersiapkan oleh:

Kelompok 4

Azhar Kurniawan (1301174023)

Fitrayada Ade Perdana (1301174372)

Muhammad Athariq Purbaya (1301174155)

Muhammad Ilham Akbari (1301164050)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-004* | | *<#>/<jml #* |
| Revisi | *02* | *Tgl: 1 Mei 2019* |

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 10](#_Toc8069313)

[Daftar Gambar 11](#_Toc8069314)

[1. Pendahuluan 12](#_Toc8069315)

[1.1 Tujuan Penulisan Dokumen 12](#_Toc8069316)

[1.2 Lingkup Masalah 12](#_Toc8069317)

[1.3 Definisi dan Istilah 12](#_Toc8069318)

[1.4 Referensi 14](#_Toc8069319)

[1.5 Sistematika Pembahasan 15](#_Toc8069320)

[2 Deskripsi Perancangan Global 15](#_Toc8069321)

[2.1 Deskripsi Arsitektural 16](#_Toc8069322)

[2.2 Deskripsi Komponen 16](#_Toc8069323)

[3 Perancangan Rinci 18](#_Toc8069324)

[3.1 Realisasi Use Case 18](#_Toc8069325)

[3.1.1 Use Case #1 18](#_Toc8069326)

[3.1.1.1 Perancangan Antarmuka Usecase #1 20](#_Toc8069327)

[3.1.1.1.1 Tabel Identifikasi Antarmuka / Layar / Page 20](#_Toc8069328)

[3.1.1.2 Identifikasi Object Baru 21](#_Toc8069329)

[3.1.1.3 Robustness Diagram 22](#_Toc8069330)

[3.1.1.4 Diagram Kelas 23](#_Toc8069331)

[3.1.1.5 Sequence Diagram 23](#_Toc8069332)

[3.1.2 Use Case #2 23](#_Toc8069333)

[3.1.2.1 Perancangan Antarmuka Usecase #2 25](#_Toc8069334)

[3.1.2.1.1 Tabel Identifikasi Antarmuka / Layar / Page 26](#_Toc8069335)

[3.1.2.2 Identifikasi Object Baru 27](#_Toc8069336)

[3.1.2.3 Robustness Diagram 28](#_Toc8069337)

[3.1.2.4 Diagram Kelas 28](#_Toc8069338)

[3.1.2.5 Sequence Diagram 28](#_Toc8069339)

[3.1.3 Use Case #3 28](#_Toc8069340)

[3.1.3.1 Perancangan Antarmuka Usecase #3 30](#_Toc8069341)

[3.1.3.1.1 Tabel Identifikasi Antarmuka / Layar / Page 30](#_Toc8069342)

[3.1.3.2 Identifikasi Object Baru 31](#_Toc8069343)

[3.1.3.3 Robustness Diagram 33](#_Toc8069344)

[3.1.3.4 Diagram Kelas 34](#_Toc8069345)

[3.1.3.5 Sequence Diagram 34](#_Toc8069346)

[3.1.4 Use Case #4 34](#_Toc8069347)

[3.1.4.1 Perancangan Antarmuka Usecase #4 35](#_Toc8069348)

[3.1.4.1.1 Tabel Identifikasi Antarmuka / Layar / Page 36](#_Toc8069349)

[3.1.4.2 Identifikasi Object Baru 37](#_Toc8069350)

[3.1.4.3 Robustness Diagram 38](#_Toc8069351)

[3.1.4.4 Diagram Kelas 40](#_Toc8069352)

[3.1.4.5 Sequence Diagram 40](#_Toc8069353)

[3.1.5 Use Case #5 40](#_Toc8069354)

[3.1.5.1 Perancangan Antarmuka Usecase #5 41](#_Toc8069355)

[3.1.5.1.1 Tabel Identifikasi Antarmuka / Layar / Page 42](#_Toc8069356)

[3.1.5.2 Identifikasi Object Baru 42](#_Toc8069357)

[3.1.5.3 Robustness Diagram 43](#_Toc8069358)

[3.1.5.4 Diagram Kelas 45](#_Toc8069359)

[3.1.5.5 Sequence Diagram 45](#_Toc8069360)

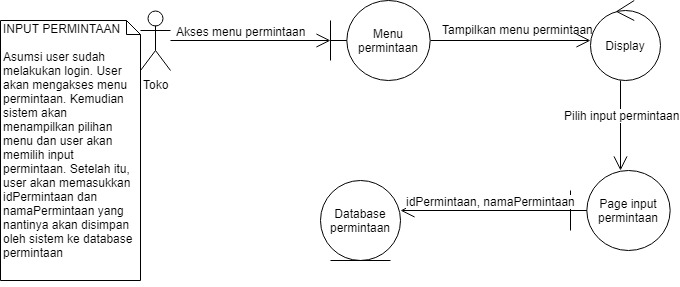
[3.1.6 Use Case #6 45](#_Toc8069361)

[3.1.6.1 Perancangan Antarmuka Usecase #6 46](#_Toc8069362)

[3.1.6.1.1 Tabel Identifikasi Antarmuka / Layar / Page 47](#_Toc8069363)

[3.1.6.2 Identifikasi Object Baru 47](#_Toc8069364)

[3.1.6.3 Robustness Diagram 48](#_Toc8069365)

[3.1.6.4 Diagram Kelas 49](#_Toc8069366)

[3.1.6.5 Sequence Diagram 49](#_Toc8069367)

[3.1.7 Use Case #7 49](#_Toc8069368)

[3.1.7.1 Perancangan Antarmuka Usecase #7 49](#_Toc8069369)

[3.1.7.1.1 Tabel Identifikasi Antarmuka / Layar / Page 49](#_Toc8069370)

[3.1.7.2 Identifikasi Object Baru 49](#_Toc8069371)

[3.1.7.3 Robustness Diagram 49](#_Toc8069372)

[3.1.7.4 Diagram Kelas 49](#_Toc8069373)

[3.1.7.5 Sequence Diagram 49](#_Toc8069374)

[3.1.8 Use Case #8 49](#_Toc8069375)

[3.1.8.1 Perancangan Antarmuka Usecase #8 49](#_Toc8069376)

[3.1.8.1.1 Tabel Identifikasi Antarmuka / Layar / Page 49](#_Toc8069377)

[3.1.8.2 Identifikasi Object Baru 49](#_Toc8069378)

[3.1.8.3 Robustness Diagram 49](#_Toc8069379)

[3.1.8.4 Diagram Kelas 49](#_Toc8069380)

[3.1.8.5 Sequence Diagram 49](#_Toc8069381)

[3.1.9 Use Case #9 49](#_Toc8069382)

[3.1.9.1 Perancangan Antarmuka Usecase #9 50](#_Toc8069383)

[3.1.9.1.1 Tabel Identifikasi Antarmuka / Layar / Page 50](#_Toc8069384)

[3.1.9.2 Identifikasi Object Baru 51](#_Toc8069385)

[3.1.9.3 Robustness Diagram 52](#_Toc8069386)

[3.1.9.4 Diagram Kelas 54](#_Toc8069387)

[3.1.9.5 Sequence Diagram 54](#_Toc8069388)

[3.1.10 Use Case #10 54](#_Toc8069389)

[3.1.10.1 Perancangan Antarmuka Usecase #10 55](#_Toc8069390)

[3.1.10.1.1 Tabel Identifikasi Antarmuka / Layar / Page 56](#_Toc8069391)

[3.1.10.2 Identifikasi Object Baru 56](#_Toc8069392)

[3.1.10.3 Robustness Diagram 57](#_Toc8069393)

[3.1.10.4 Diagram Kelas 58](#_Toc8069394)

[3.1.10.5 Sequence Diagram 58](#_Toc8069395)

[4 Perancangan Detil 58](#_Toc8069396)

[4.1 Perancangan Detil Kelas 58](#_Toc8069397)

[4.2 Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel) 59](#_Toc8069398)

[4.3 Perancangan Algoritma 59](#_Toc8069399)

[4.3.1 Algoritma #1 59](#_Toc8069400)

[4.3.1 Algoritma #2 60](#_Toc8069401)

[4.3.2 Algoritma #3 61](#_Toc8069402)

[4.3.3 Algoritma #4 61](#_Toc8069403)

[4.3.4 Algoritma #5 61](#_Toc8069404)

[4.3.5 Algoritma #6 62](#_Toc8069405)

[4.3.6 Algoritma #7 63](#_Toc8069406)

[4.4 Perancangan Query 63](#_Toc8069407)

[5 Matriks Kerunutan (Requirement Traceability Matrix) 64](#_Toc8069408)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# Daftar Tabel

# Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi Dokumen Perancangan Perangkat Lunak (DPPL) untuk *website* Inventaris Gudang. Tujuan dari penulisan dokumen ini adalah untuk memberikan penjelasan mengenai desain perangkat lunak yang akan dibangun. Desain yang dimaksud mencakup seluruh aspek desain perangkat lunak. Baik berupa gambaran umum maupun penjelasan yang detail. Pengguna dari Dokumen ini adalah pengembang perangkat lunak untuk website Inventaris Gudang. Dokumen ini akan digunakan sebagai bahan acuan dalam pengembangan dan sebagai bahan evaluasi pasca-pengembangan *website* Inventaris Gudang.

## Lingkup Masalah

Nama dari website kami adalah Inventaris Gudang yang merupakan sebuah website dimana nantinya dapat menunjang proses pendataan barang pada Gudang yang menggunakan aplikasi ini. Hal yang dapat dilakukan oleh aplikasi ini adalah:

1. Dapat melakukan *login* pada website Inventaris Gudang.
2. Dapat melakukan manajemen barang (lihat, tambah, hapus dan ubah data barang).
3. Dapat digunakan oleh pihak Toko untuk melakukan permintaan barang ke Gudang.
4. Dapat digunakan oleh pihak Gudang untuk memesan barang ke Logistik.
5. Dapat digunakan oleh pihak Logistik untuk memproses pesanan dari Gudang.

## Definisi dan Istilah

|  |  |  |
| --- | --- | --- |
| Istilah | Singkatan | Definisi |
| Spesifikasi Kebutuhan Perangkat Lunak | SKPL | Spesifikasi perangkat lunak yang akan di kembangkan. |
| Deskripsi Perancangan Perangkat Lunak | DPPL | merupakan dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses pengkodean *website* inventaris gudang. |
| *Functional Requirements* | FR | Kebutuhan fungsional mendefinisikan apa yang system lakukan, dan disimpan dalam database |
| *Non-Functional Requirements* | NFR | Kebutuhan fungsional mendefinisikam kualitas secara keseluruhan. |
| *Use case* | UC | Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara ‘aktor’ — inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana. |
| *Database* | DB | Kumpulan data-data yang tersimpan, tersusun, dan saling terhubung satu sama lain pada suatu komputer serta digunakan perangkat lunak untuk mengakses maupun mengelolanya sehingga dapat dihasilkan informasi yang berguna. |
| *Website* | *Website* | *Website* adalah fasilitas internet penghubung dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web* *page* sementara link dalam *website* memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper* *text*), baik diantara page yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. |
| *Admin* | *Admin* | Aktor yang mengendalikan, memantau segala kegiatan dalam aplikasi. |

## Referensi

* Sora,2017. Pengertian Database. 8 April 2017. Web Site.
* Novitasari, Candra, 2018. Pengertian Class Diagram. 28 September 1018. Web Site Pelajarindo.com.
* Koishikunar, 2008. DPPL. 30 Juni 2008. Web Site.
* Pinandito, Aryo, 2012. Functional Requirements (FR) dan Non-Functional Requirements (NFR), Program Vokasi Universitas Brawijaya 2012.
* Lukmanul, Hakim, 2004. Website. Web site Dewaweb.com
* Wicaksana, Arif, 2016. Pengertian Use Case. Web Site Medium Corp.
* Spesifikasi Kebutuhan Perangkat lunak

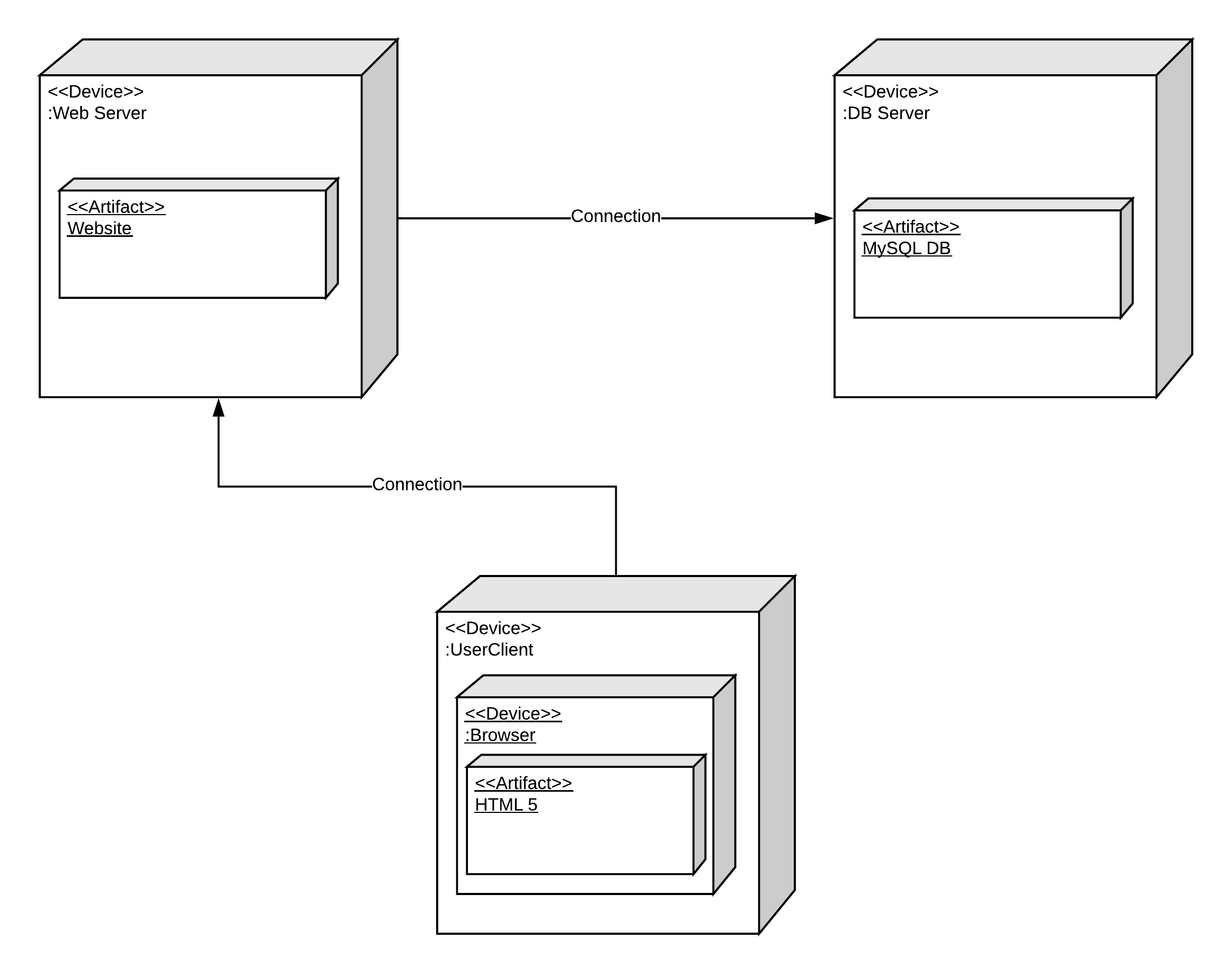
## Sistematika Pembahasan

Dokumen DPPL untuk Perangkat Lunak ini berisi deskripsi kebutuhan pengembangan perangkat lunak secara rinci. Perorganisasian dokumen dikelompokkan dalam 5 bagian, yaitu diantaranya :

1. Bab 1 Pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang di tangani pada perangkat lunak yang akan dibangun, dan deskripsi umum dokumen.
2. Bab 2 Deskripsi Perancangan Global Perangkat Lunak, berisi penjelasan perangkat lunak yang akan di implementasikan di lingkungan pengguna secara umum, pada bagian ini terdiri dari deskripsi aarsitektural,deskripsi komponen, fungsi produk, dan batasan-batasan.
3. Bab 3 Deskripsi Rinci Kebutuhan, berisi hasil analisi terhadap kebutuhan perangkat lunak secara rinci.
4. Bab 4 Deskripsi detil
5. Bab 5 Matriks Ketenurutan.

# Deskripsi Perancangan Global

## Deskripsi Arsitektural



## Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1 | Login | Proses login harus dilakukan agar pengguna bias mengakses aplikasi ini |
| 2 | Management Barang | Proses memperbarui data barang. |
| 3 | Tambah barang | Proses tambah barang ketika toko menginginkan barang |
| 4 | Hapus barang | Proses hapus barang ketika ada barang yang jumlahnya berkurang akibat permintaan. |
| 5 | Lihat Barang Tersedia | Proses melihat barang untuk pihak took untuk melihat barang yang tersedia |
| 6 | Proses Pemesanan | Proses pihak logistik dalam menanggapi pemesanan oleh pihak gudang. |
| 7 | Input Permintaan | Proses membuat permintaan barang oleh pihak toko ke pihak gudang. |
| 8 | Input Pemesanan | Proses membuat pemesanan barang oleh pihak gudang ke pihak toko. |
| 9 | Registrasi | Proses mendaftarkan diri ke halaman web. |

# Perancangan Rinci

## Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Use Case | Deskripsi Use Case | |
| 1 | Login | Untuk masuk ke halaman web. | |
| 2 | Registrasi | Untuk mendaftarkan akun baru agar dapat masuk ke halaman web. | |
| 3 | Manajemen Barang | Untuk memperbarui data barang. | |
| 4 | Hapus Barang | Untuk menghapus data barang. | |
| 5 | Tambah Barang | Untuk menambahkan data barang. | |
| 6 | Input Permintaan | Untuk memesan barang ke pihak gudang oleh pihak toko | |
| 7 | Input Pemesanan | Untuk memesan barang ke pihak logistik oleh pihak gudang jika terdapat barang atau stok barang yang kurang. | |
| 8 | Lihat Barang Tersedia | Untuk melihat barang yang dapat dipesan oleh pihak toko. | |
| 9 | Proses Pemesanan | Untuk memproses pemesanan dari pihak gudang | |
|  | | | |

### Use Case #1

*Skenario Use Case #1 : Login*

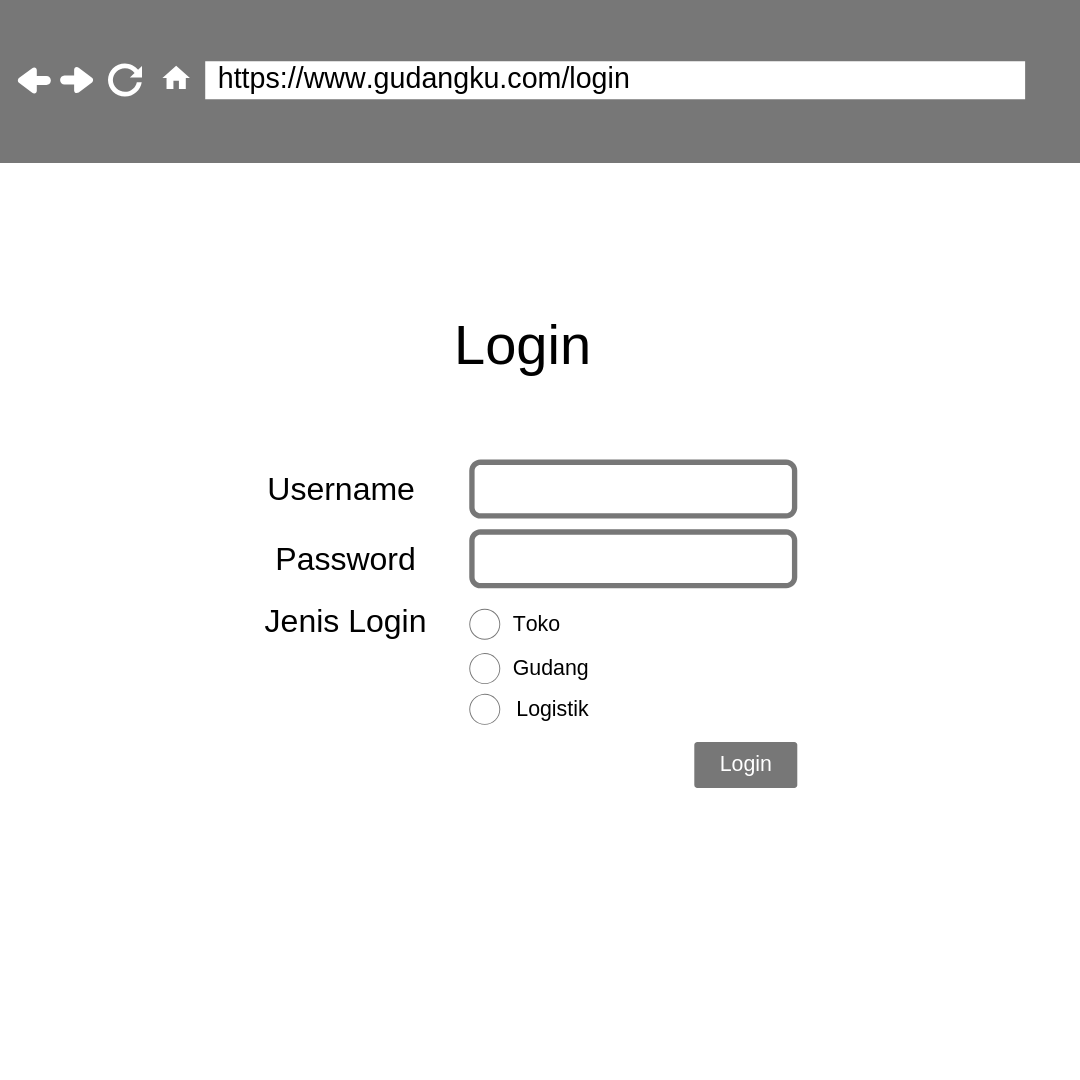
*Primary Flow :*

* + - 1. Pengguna membuka *website*.
      2. Sistem menampilkan form *login* yang berisikan *username*/*email* dan *password.*
      3. Pengguna memasukkan *username* dan *password* yang sesuai dengan akunnya dan memilih jenis login.
      4. Sistem memeriksa kecocokan *username* dan *password.*
      5. Jika *username* dan *password* tepat, maka sistem akan mengijinkan *user* untuk *login.*

*Alternate Flow :*

* + - 1. Pengguna membuka *website*.
      2. Sistem menampilkan form *login* yang berisikan *username*/*email* dan *password.*
      3. Pengguna memasukkan *username* dan *password* yang sesuai dengan akunnya dan memilih jenis login.
      4. Sistem memeriksa kecocokan *username* dan *password.*
      5. Jika *username* dan/atau *password* salah, maka sistem akan kembali menampilkan halaman *login.*

#### Perancangan Antarmuka Usecase #1



#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
|  | *Home Page* |  |
|  | *Page Registrasi* |  |
|  | *Page Transaksi* |  |

*Page HOMEPAGE*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Function….. |
| *Buuton 2* | *Button* | *SAVE* | Jika diklik akan mengaktifkan …… |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |

*Page REGISTRASI*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *TextBox1* | Textbox | Username | Jika diklik, akan mengaktifkan Function….. |
| *Buuton 2* | *Button* | *SAVE* | Jika diklik akan mengaktifkan …… |

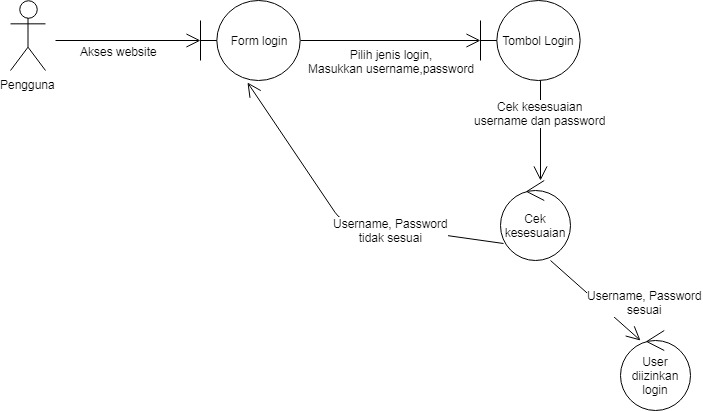
#### Identifikasi Object Baru

*TABEL OBJECT PERANCANGAN*

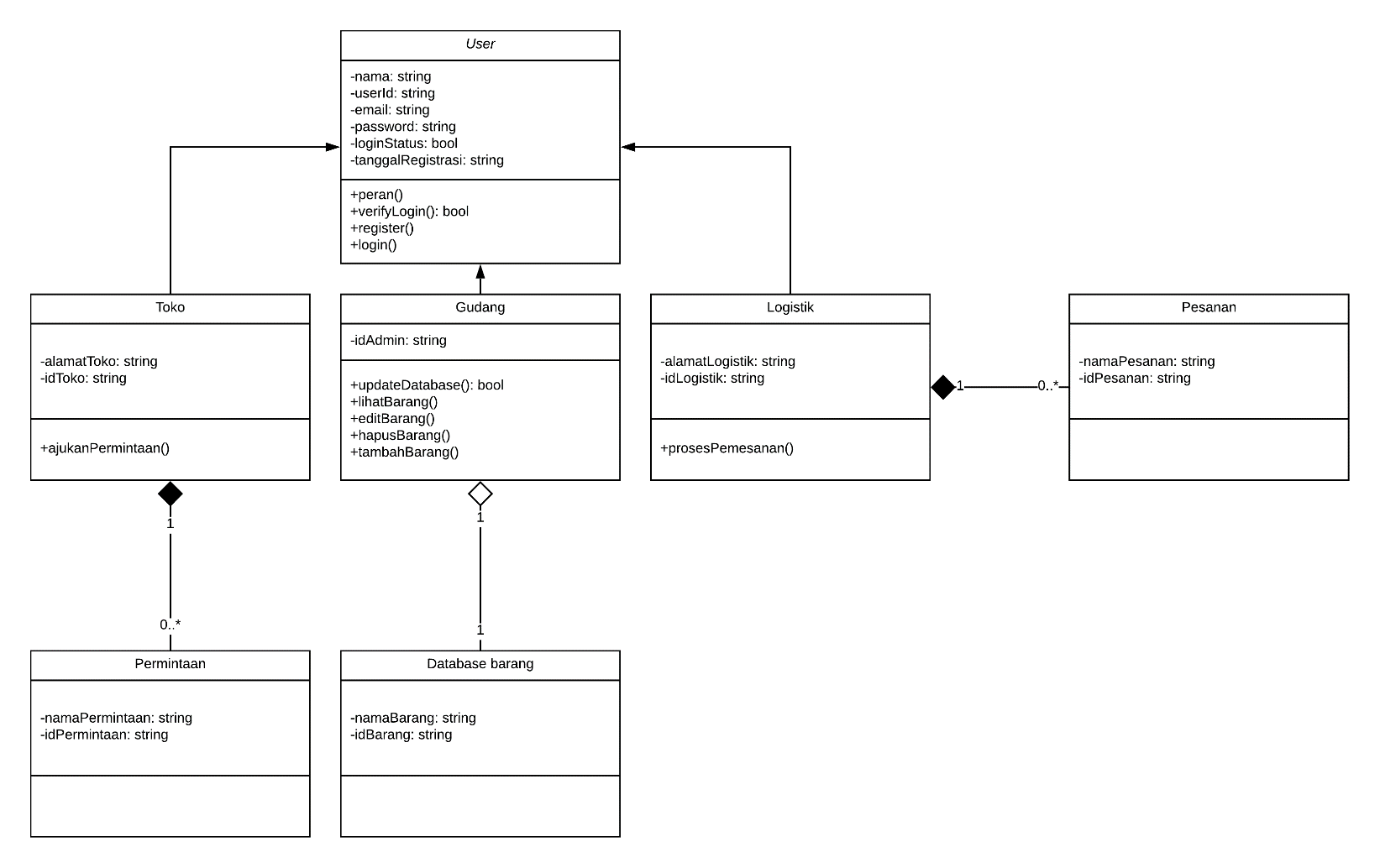
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu Login* | *Boundary* |
| *2* | *Validasi* | *Controller* |
| *3* | *Database User* | *Entitiy* |
| *4* | *Masuk sistem* | *controller* |
| *5* | *Keluar* | *controller* |

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

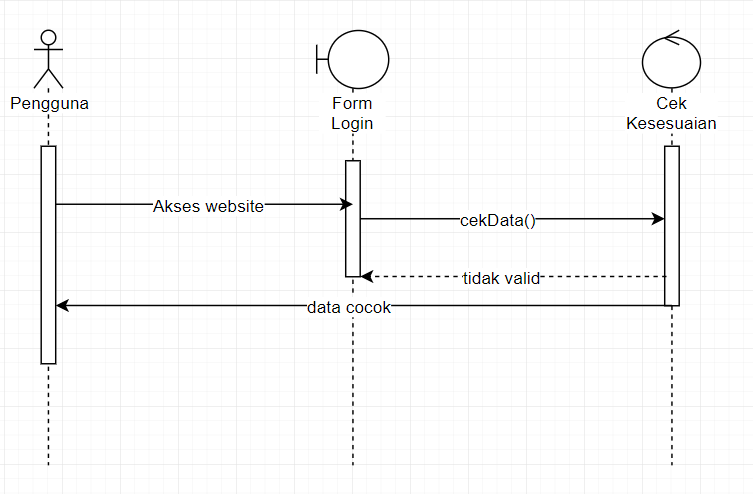
#### Robustness Diagram

**

#### Diagram Kelas



#### Sequence Diagram

******

### Use Case #2

*Skenario Use Case #2 : Registrasi*

*Primary Flow :*

Pengguna mengklik tombol ‘registrasi’.

Sistem mengarahkan ke halaman registrasi.

Pengguna memilih jenis pengguna.

Pengguna mengisi data diri.

Pengguna mengklik tombol ‘registrasi’.

Sistem memeriksa apakah email telah terdaftar. Jika belum, maka sistem akan menambahkan data.

Akun dibuat dan pengguna diarahkan ke halaman login

*Alternate Flow :*

Pengguna mengklik tombol ‘registrasi’.

Sistem mengarahkan ke halaman registrasi.

Pengguna memilih jenis pengguna.

Pengguna mengisi data diri.

Pengguna mengklik tombol ‘registrasi’.

Sistem memeriksa apakah email telah terdaftar.

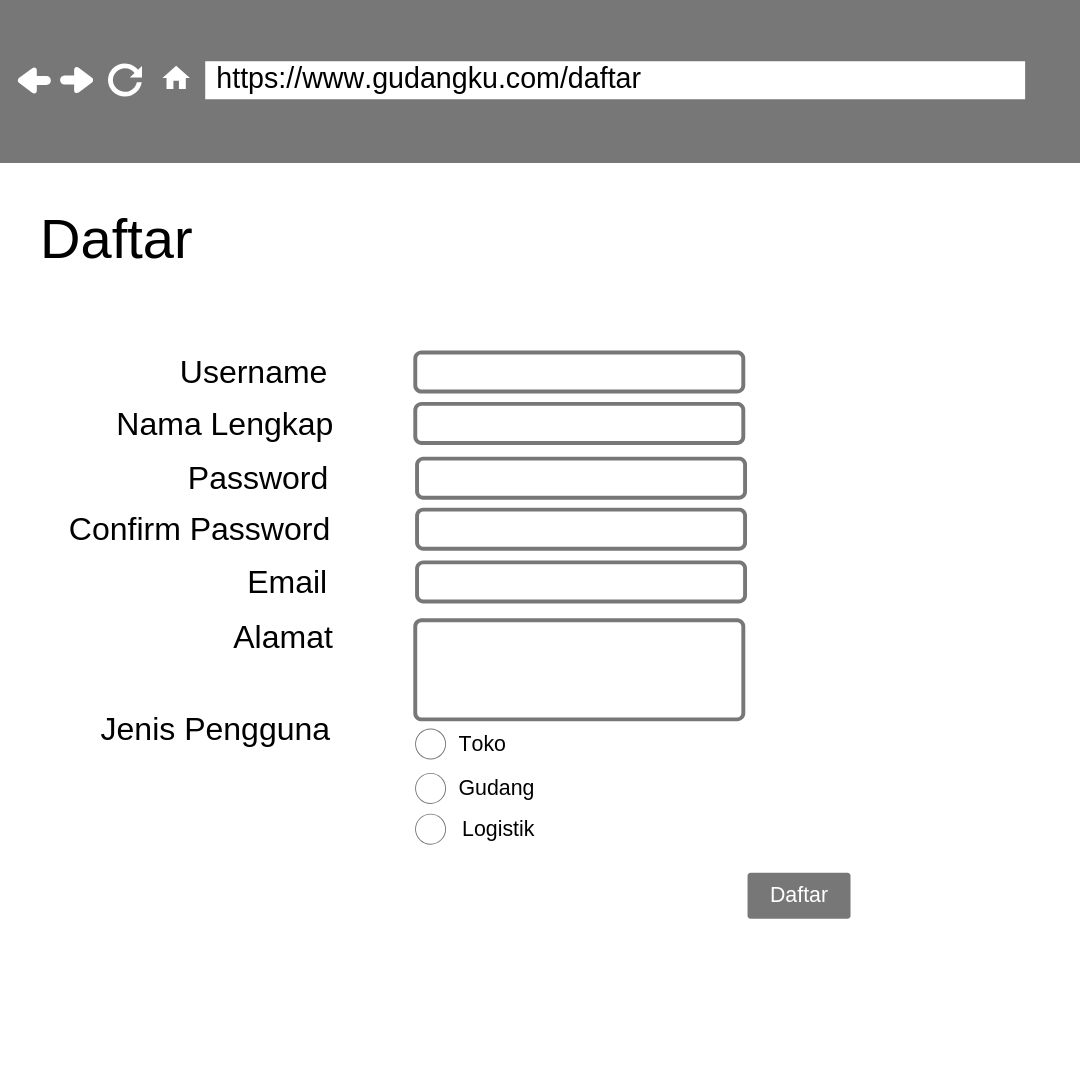
Email telah terpakai.

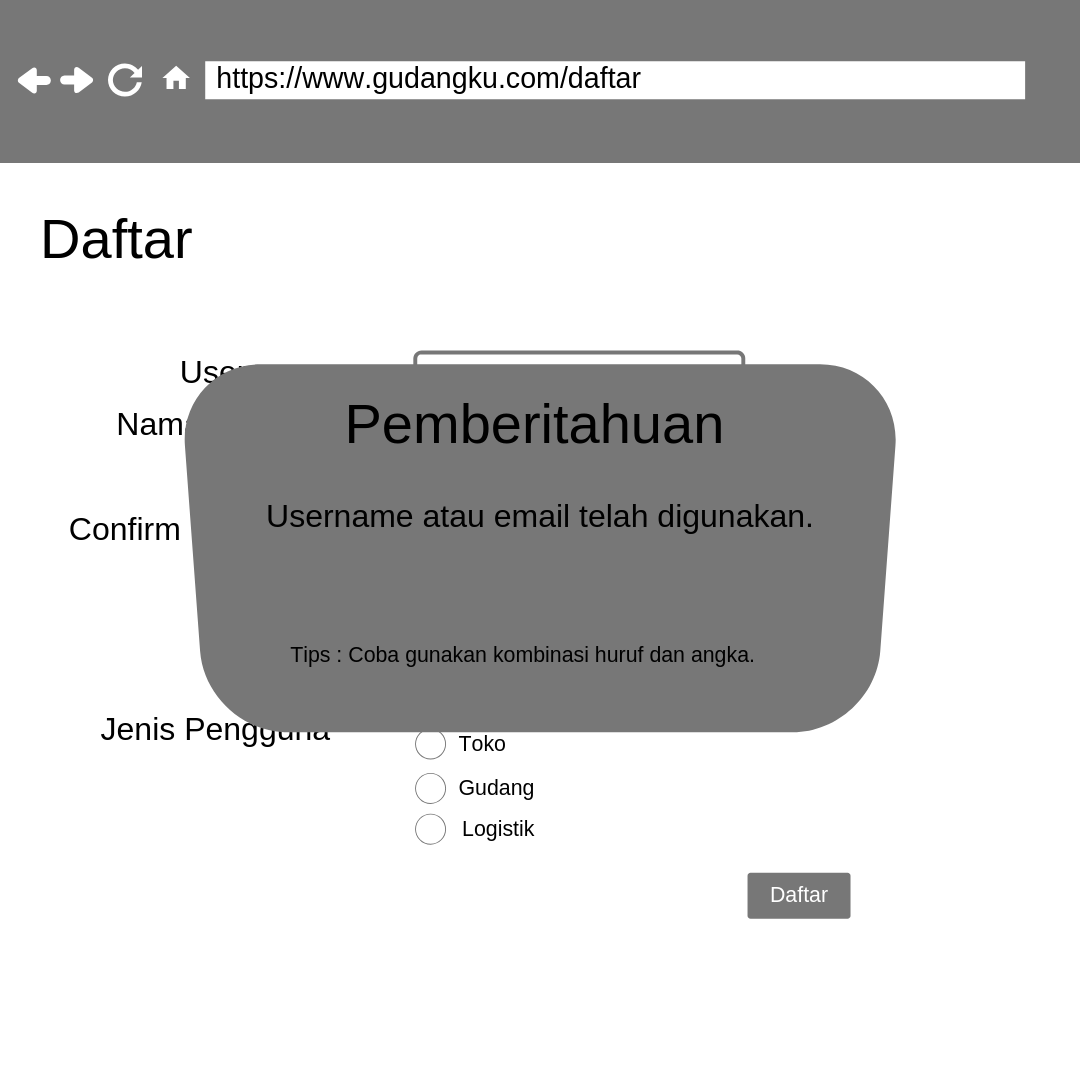
Pengguna mengganti email yang akan dipakai. dan mengklik tombol ‘registrasi’.

Email belum terpakai dan sistem menabmbahkan data.

Pengguna diarahkan ke halaman login.

#### Perancangan Antarmuka Usecase #2





#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
|  | *Home Page* |  |
|  | *Page Registrasi* |  |
|  | *Page Transaksi* |  |

*Page HOMEPAGE*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Button1* | Button | OK | Jika diklik, akan mengaktifkan Function….. |
| *Buuton 2* | *Button* | *SAVE* | Jika diklik akan mengaktifkan …… |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |
|  |  |  |  |

*Page REGISTRASI*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *TextBox1* | Textbox | Username | Jika diklik, akan mengaktifkan Function….. |
| *Buuton 2* | *Button* | *SAVE* | Jika diklik akan mengaktifkan …… |

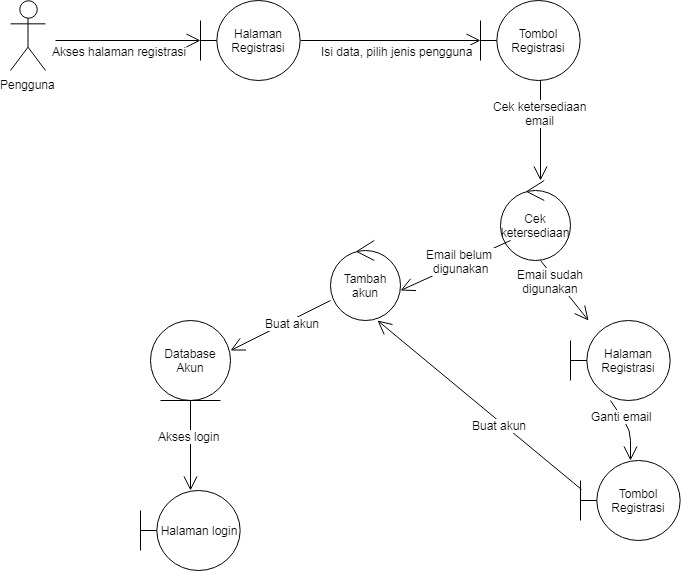
#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

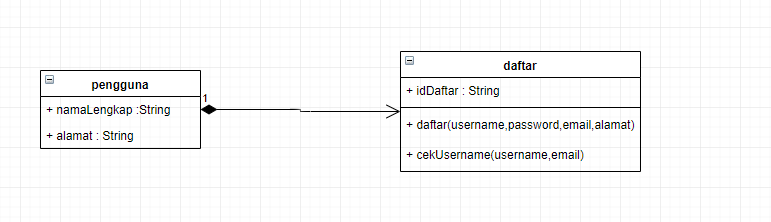
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu daftar* | *boundary* |
| *2* | *Cek ketersediaan* | *controller* |
| *3* | *Database user* | *entity* |
| *4* | *Disimpan pada database* | *controller* |
| *5* | *Notifikasi error* | *controller* |

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

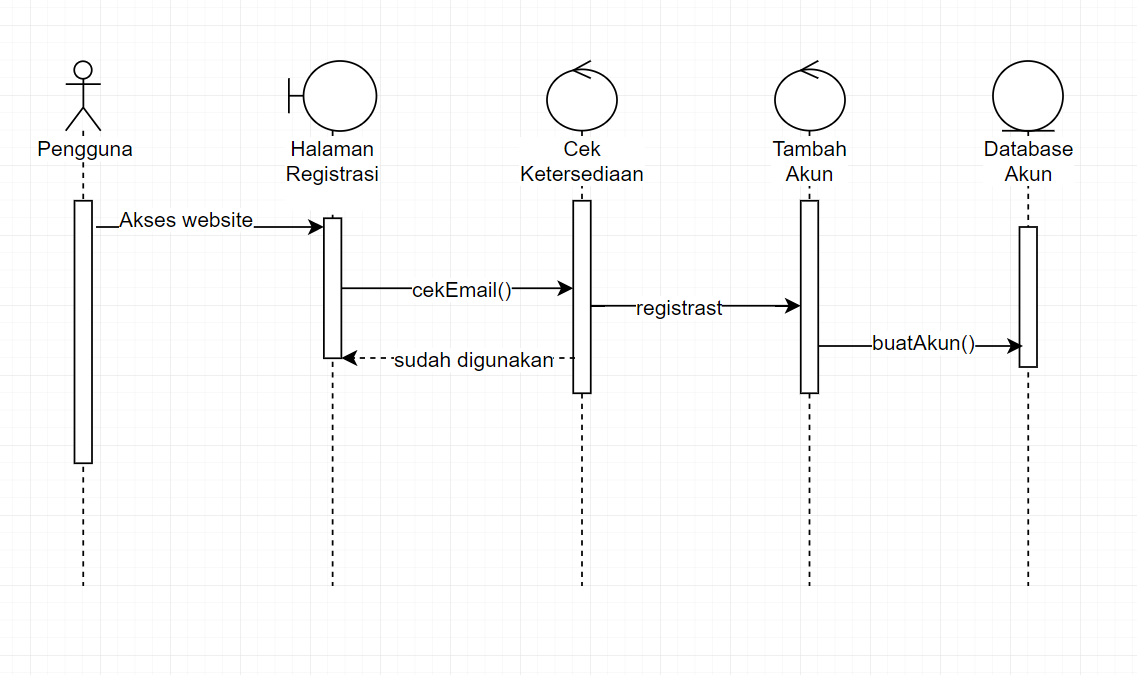
#### Robustness Diagram



#### Diagram Kelas

**

#### Sequence Diagram



### Use Case #3

*Skenario Use Case #3 : Manajemen Barang*

*Primary Flow :*

Setelah berhasil login sebagai gudang, pengguna diarahkan ke halaman manajemen barang.

Pengguna mengklik tombol ‘edit’ dan mengedit barang. Kemudian mengklik tombol ‘selesai’.

Sistem merekam perubahan yang dilakukan

Data barang berhasil diubah dan pengguna diarahkan ke halaman lihat barang.

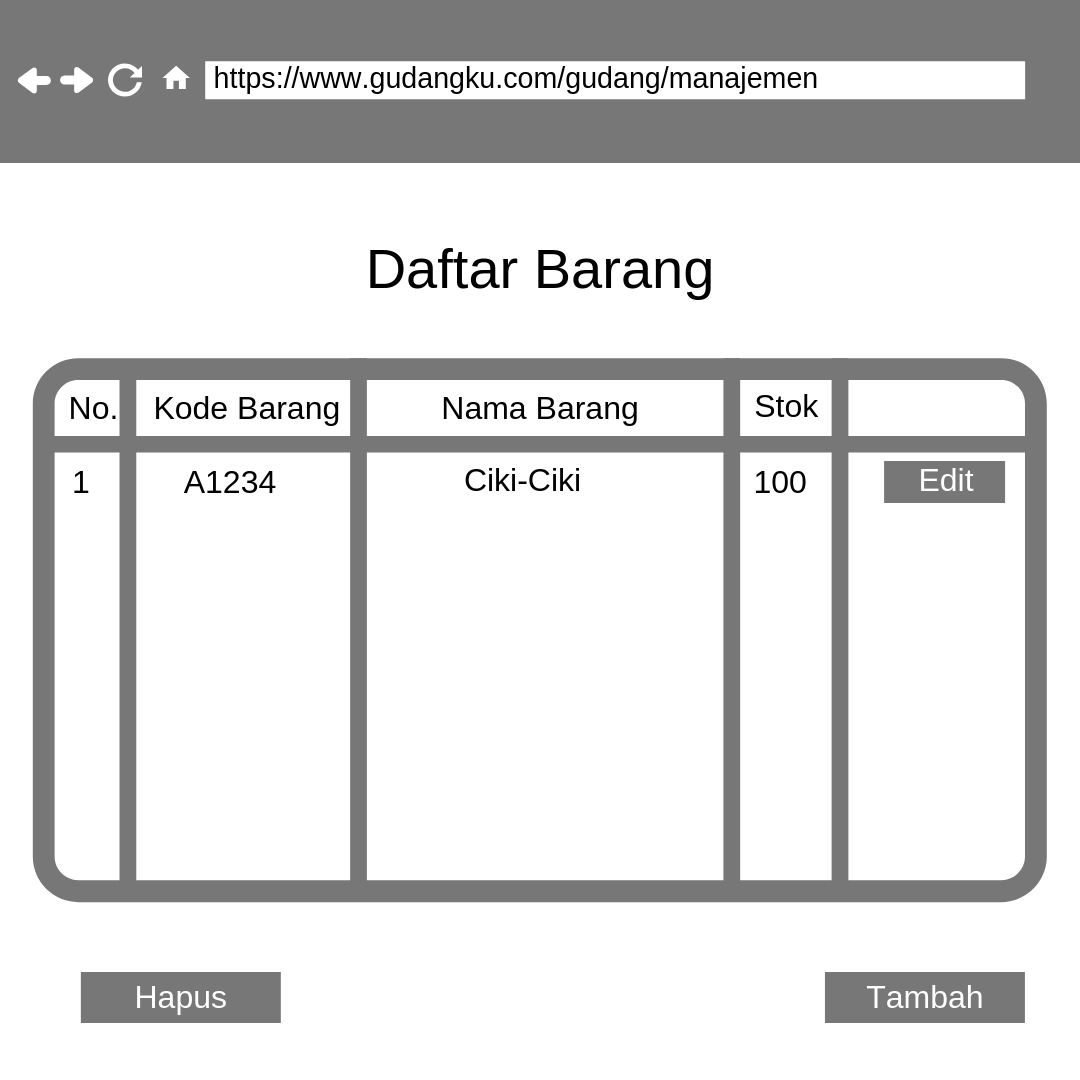
*Alternate Flow 1 :*

* + - 1. Setelah berhasil login sebagai gudang, pengguna diarahkan ke halaman manajemen barang.
      2. Pengguna mengklik tombol ‘tambah barang’.
      3. Sistem mengarahkan ke halaman tambah barang

*Alternate Flow 2 :*

* + - 1. Setelah berhasil login sebagai gudang, pengguna diarahkan ke halaman manajemen barang.
      2. Pengguna mengklik tombol ‘hapus barang’.
      3. Sistem mengarahkan ke halaman hapus barang

#### Perancangan Antarmuka Usecase #3



#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 3.1 | Page Manajemen Barang | Menampilkan data barang yang tersedia dan memberikan pilihhan untuk memperbarui, menambah atau menghapus data. |

*Page Manajemen Barang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | TAMBAH | Jika diklik, sistem akan mengarahkan pengguna ke halaman tambah barang. |
| Button2 | Button | HAPUS | Jika diklik, sistem akan mengarahkan pengguna ke halaman hapus barang. |
| Button3 | Button | EDIT | Jika diklik, sistema akan memberikan pengguna akses untuk memperbarui data barang (kode barang, nama barang maupun stok). |

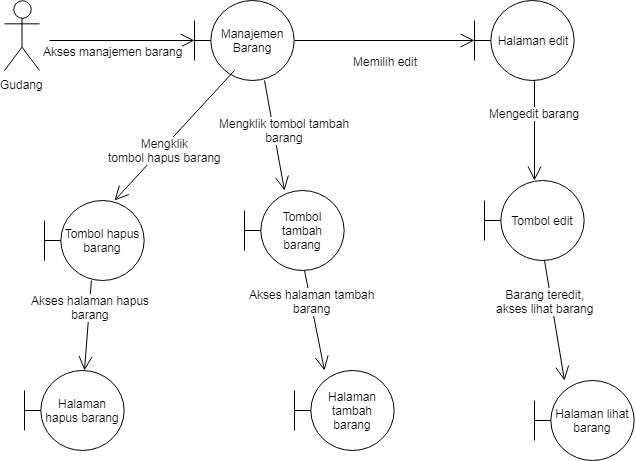
#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

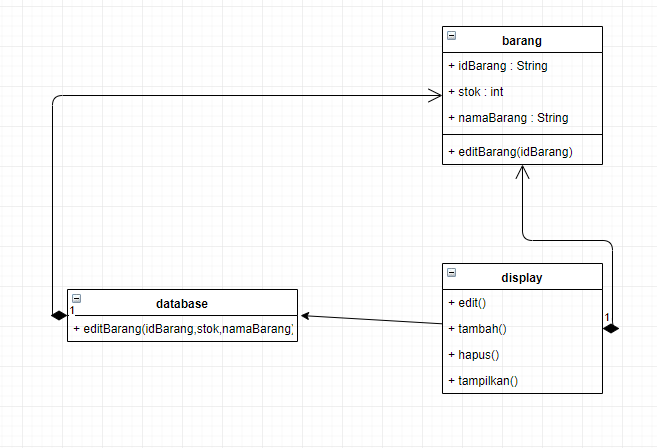
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Login Page* | *boundary* |
| *2* | *Validasi* | *contoller* |
| *3* | *Keluar* | *controller* |
| *4* | *Menu manajemen barang* | *boundary* |
| *5* | *Page input barang* | *boundary* |
| *6* | *Display* | *controller* |
| *7* | *Cari data barang* | *controller* |
| *8* | *Edit Page* | *boundary* |
| *9* | *Page delete barang* | *boundary* |
| *10* | *Notifikasi error* | *controller* |
| *11* | *Database barang* | *entity* |

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

#### Robustness Diagram

**

#### Diagram Kelas

**

#### Sequence Diagram

### Use Case #4

*Skenario Use Case #4 : Hapus Barang*

*Primary Flow :*

Pengguna berhasil mengakses halaman hapus barang.

Pengguna memasukan kode barang yang akan dihapus. Dan mengklik tombol ‘cari’.

Sistem mencari data barang yang dicari dan menampilkannya.

Pengguna melihat dan mengkonfirmasi barang yang akan dihapus.

Pengguna mengklik tombol ‘hapus’.

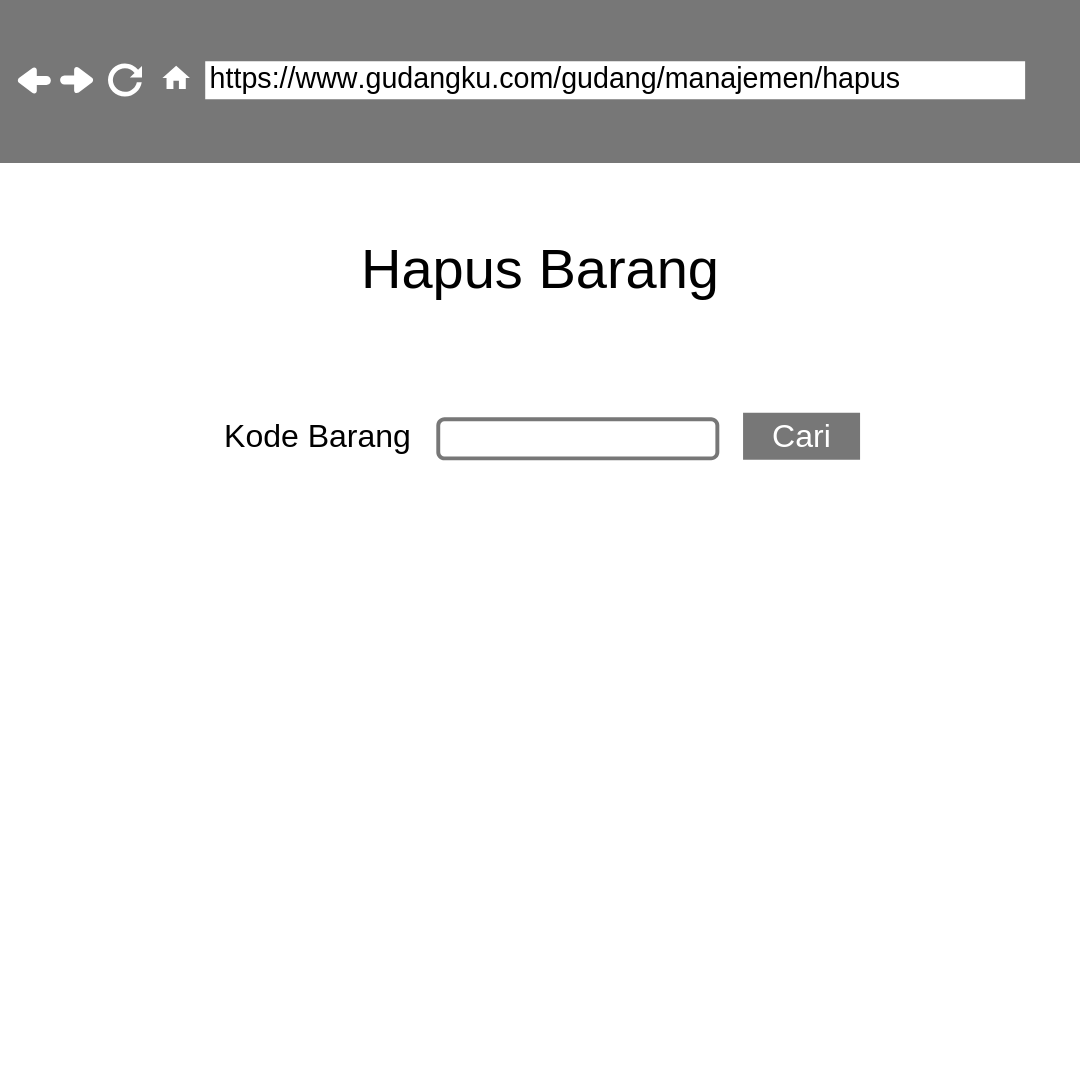
Data barang dihapus.

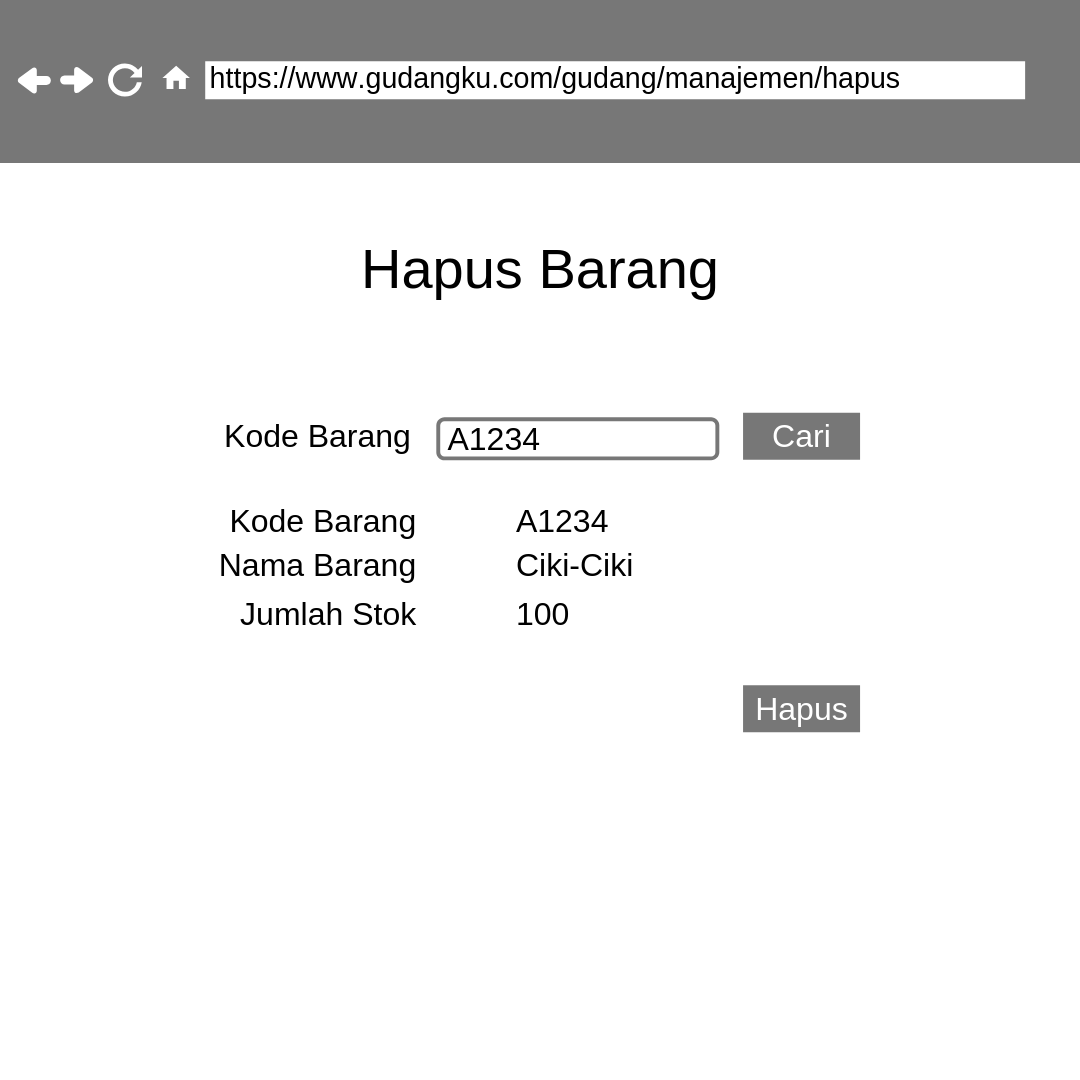
*Alternate Flow :*

* + - 1. Pengguna berhasil mengakses halaman hapus barang.
      2. Pengguna memasukan kode barang yang akan dihapus. Dan mengklik tombol ‘cari’.
      3. Sistem mencari data barang yang dicari dan barang yang dicari tidak ada.
      4. Pengguna mengulang proses pencarian.

#### Perancangan Antarmuka Usecase #4

*Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case*





#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 4.1 | Hapus Barang | Menyediakan fitur mencari barang dengan informasi barang tersebut. |
| 4.2 | Hapus Barang (Hasil Pencarian) | Menampilkan hasil pencarian barang yang siap dihapus. |

*Page Hapus Barang*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | CARI | Jika diklik, sistem akan mencari data dengan kode yang sama. |

*Page Hapus Barang (Hasil [encarian)*

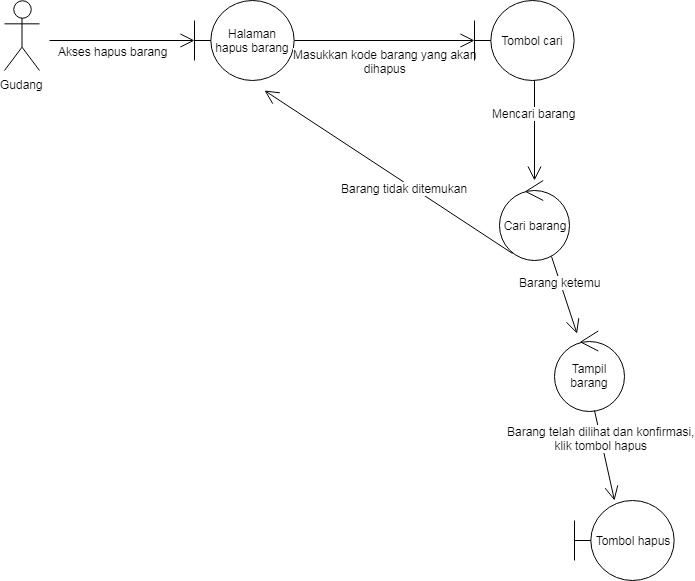
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | CARI | Jika diklik, sistem akan mencari data dengan kode yang sama. |
| Button2 | Button | HAPUS | Jika diklik, sistem akan menghapus data barang yang berhasil dicari. |

#### Identifikasi Object Baru

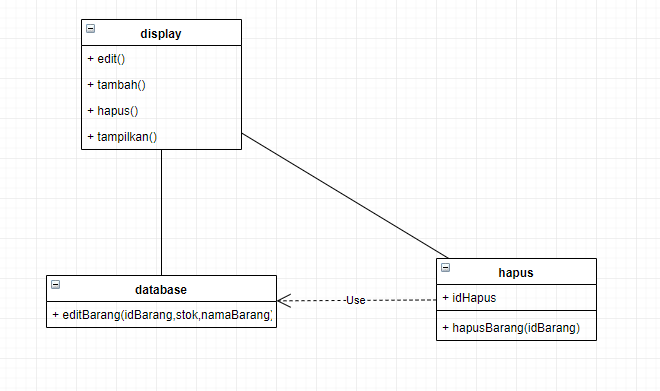
TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Login Page* | *boundary* |
| *2* | *Validasi* | *contoller* |
| *3* | *Keluar* | *controller* |
| *4* | *Menu manajemen barang* | *boundary* |
| *5* | *Page input barang* | *boundary* |
| *6* | *Display* | *controller* |
| *7* | *Cari data barang* | *controller* |
| *8* | *Edit Page* | *boundary* |
| *9* | *Page delete barang* | *boundary* |
| *10* | *Notifikasi error* | *controller* |
| *11* | *Database barang* | *entity* |

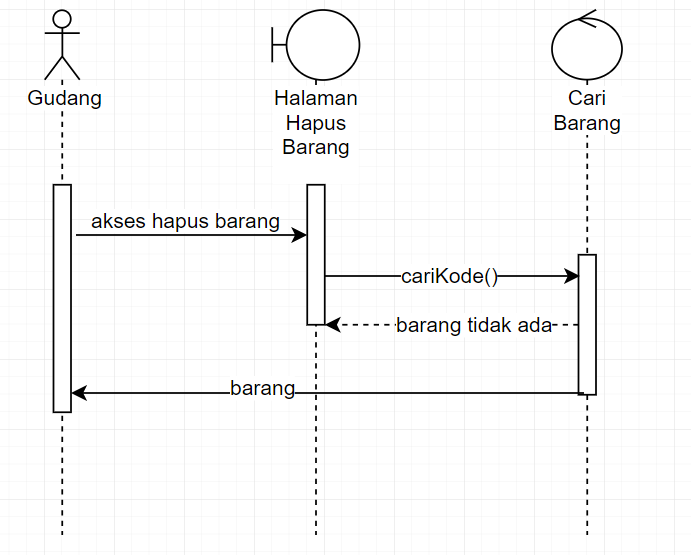
#### Robustness Diagram



#### Diagram Kelas

**

#### Sequence Diagram



### Use Case #5

*Skenario Use Case #5 : Tambah Barang*

*Primary Flow :*

Pengguna berhasil login sebagai gudang kemudian diarahkan ke halaman tambah barang.

Pengguna mengisi formulir untuk menambah data barang.

Pengguna mengklik tombol ‘tambah’.

Sistem merekam pemesanan dan menyimpan data barang jika barang dengan kode yang sama belum tersedia

Alternative  *Flow :*

Pengguna berhasil login sebagai gudang kemudian diarahkan ke halaman tambah barang.

Pengguna mengisi formulir untuk menambah data barang.

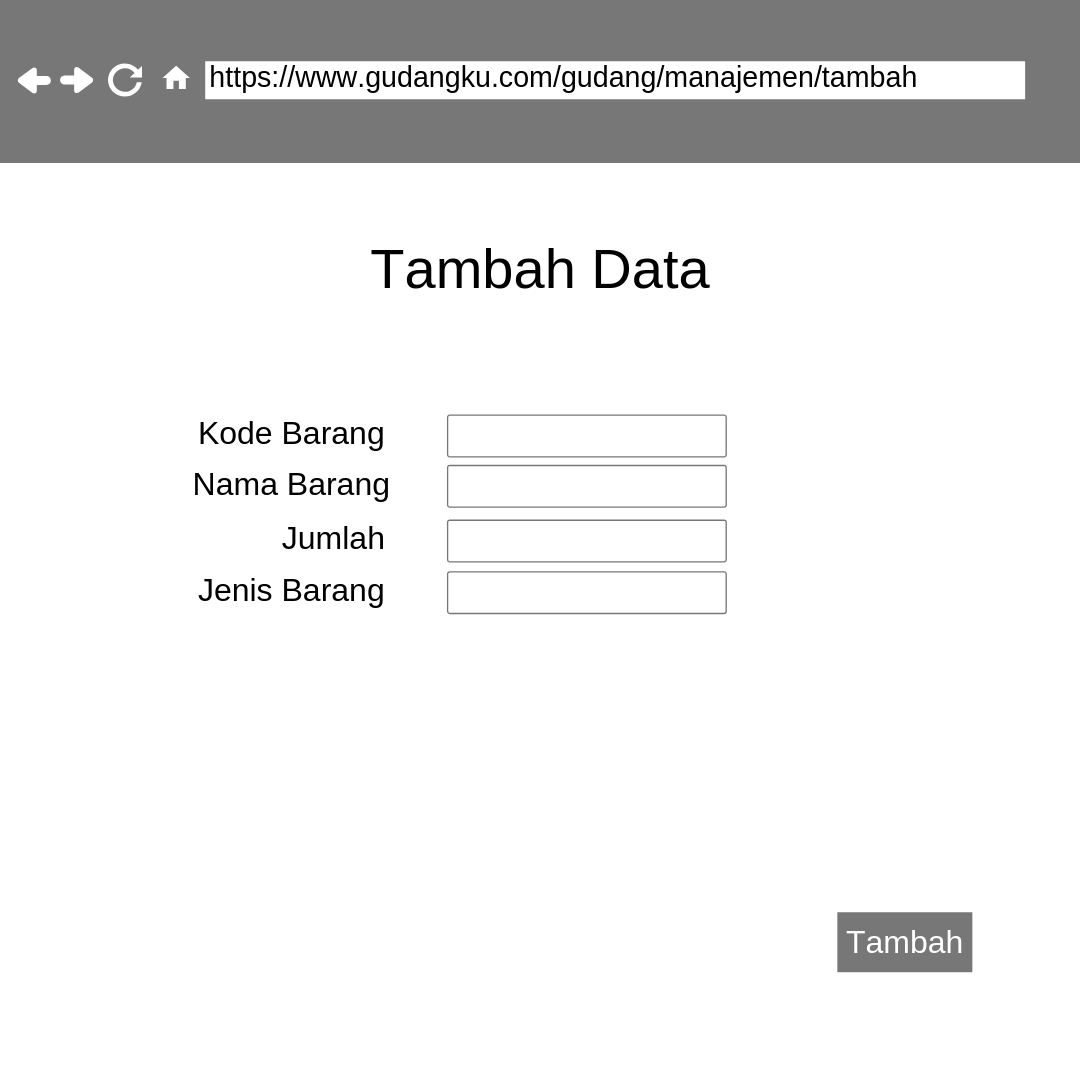
Pengguna mengklik tombol ‘tambah’.

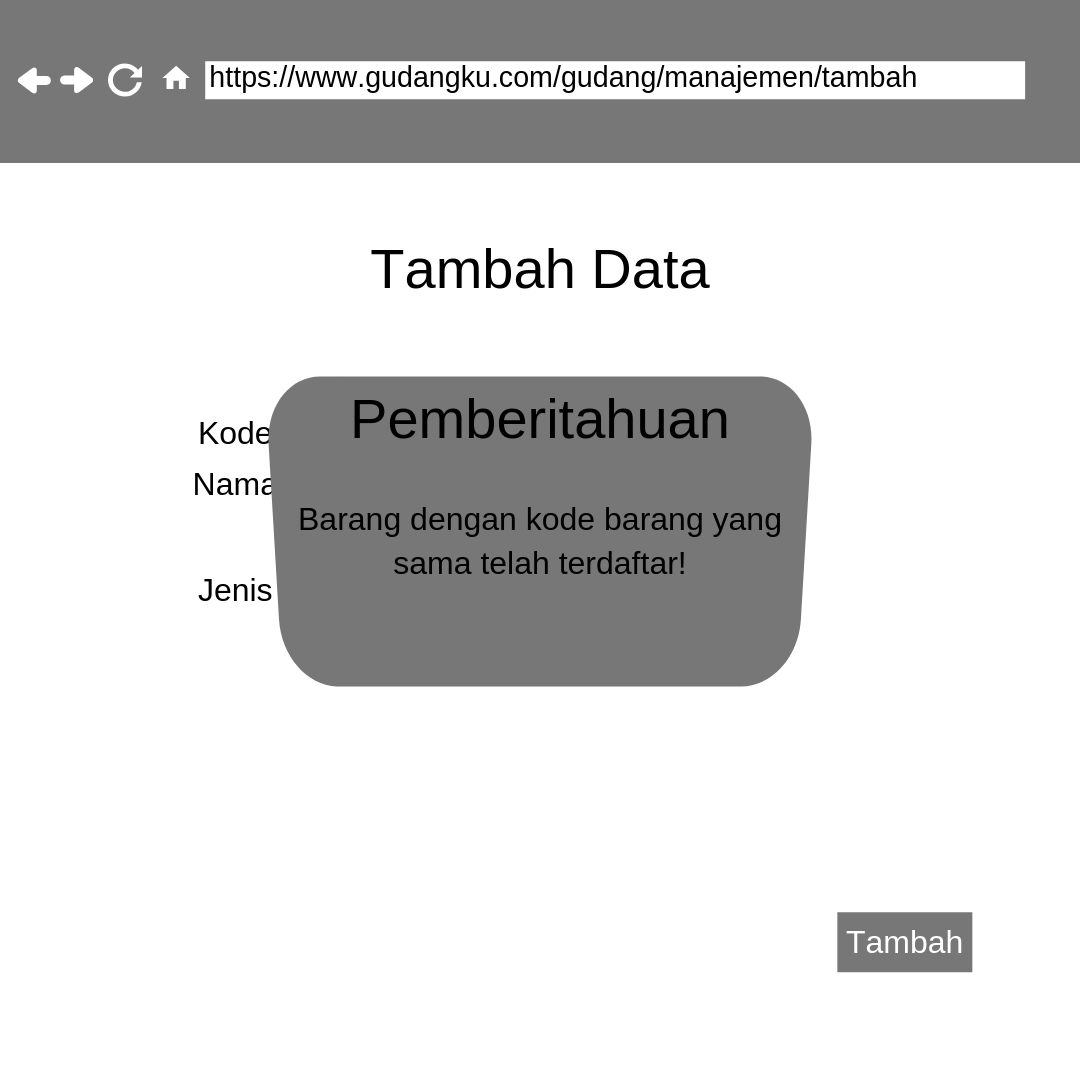
Barang dengan kode yang sama ada.

Pengguna mengganti kode barang.

Barang dengan kode yang dimasukkan belum ada dan sistem menambahkannya ke *database*.

#### Perancangan Antarmuka Usecase #5





#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 5.1 | Page Tambah Barang | Menyediakan form yang digunakan untuk memasukkan data barang baru. |

*Page Tambah Barang*

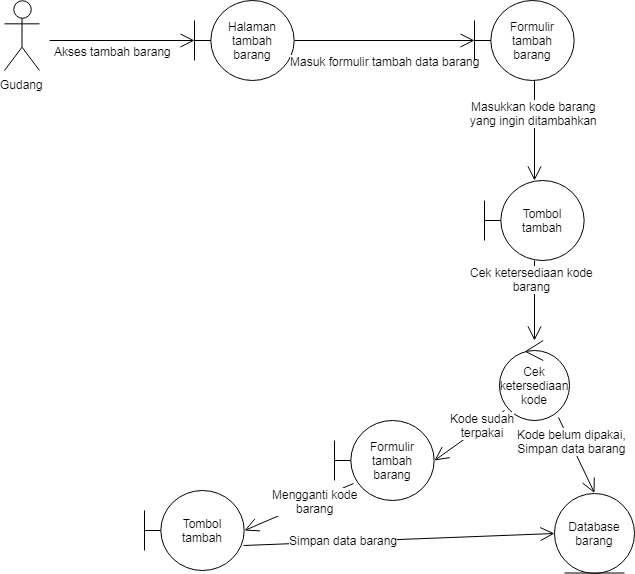
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| TextBox1 | TextBox | KODE BARANG | Kotak teks untuk memasukkan kode barang dari barang yang akan ditambahkan. |
| TextBox2 | TextBox | NAMA BARANG | Kotak teks untuk memasukkan nama barang yang akan ditambahkan |
| TextBox3 | TextBox | JUMLAH | Kotak teks untuk memasukkan jumlah stok barang. |
| TextBox4 | TextBox | JENIS BARANG | Kotak teks untuk memasukkan jenis barang. (makanan ringan, produk kecantikan dan lain-lain) |
| Button1 | Button | TAMBAH | Jika diklik, sistem akan menambahkan data barang jika data dengan kode yang sama belum ada, jika ada sistem menampilkan alert box. |

#### Identifikasi Object Baru

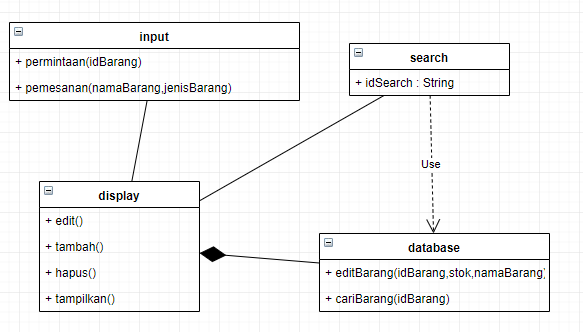
TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu permintaan* | *boundary* |
| *2* | *Display* | *controller* |
| *3* | *Page input permintaan* | *boundary* |
| *4* | *Database permintaan* | *entity* |

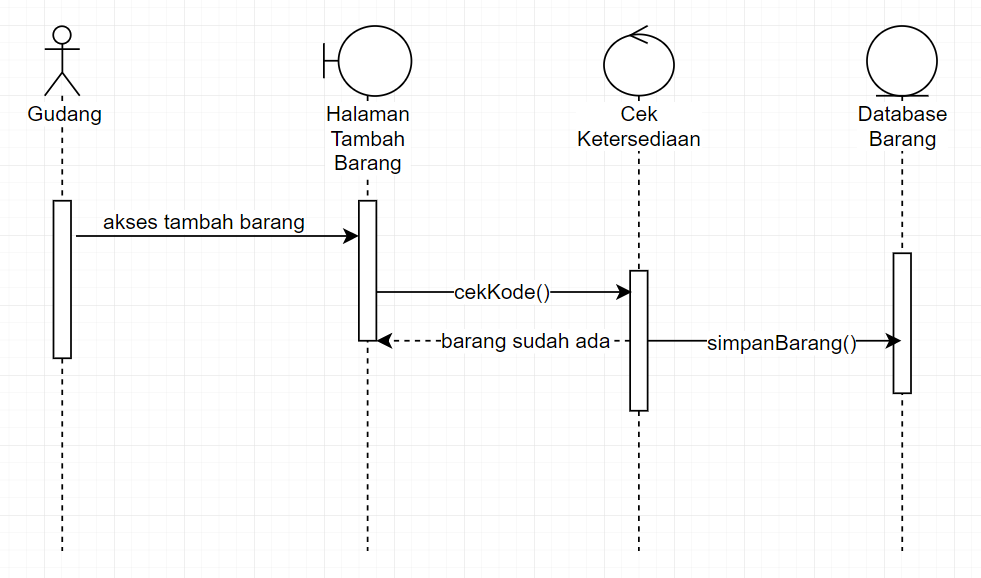
#### Robustness Diagram



#### Diagram Kelasy

**

#### Sequence Diagram



### Use Case #6

*Skenario Use Case #6 : Input Permintaan*

*Primary Flow :*

Pengguna berhasil login sebagai toko. Diarahkan ke halaman lihat barang tersedia dan mengklik tombol ‘input permintaan’.

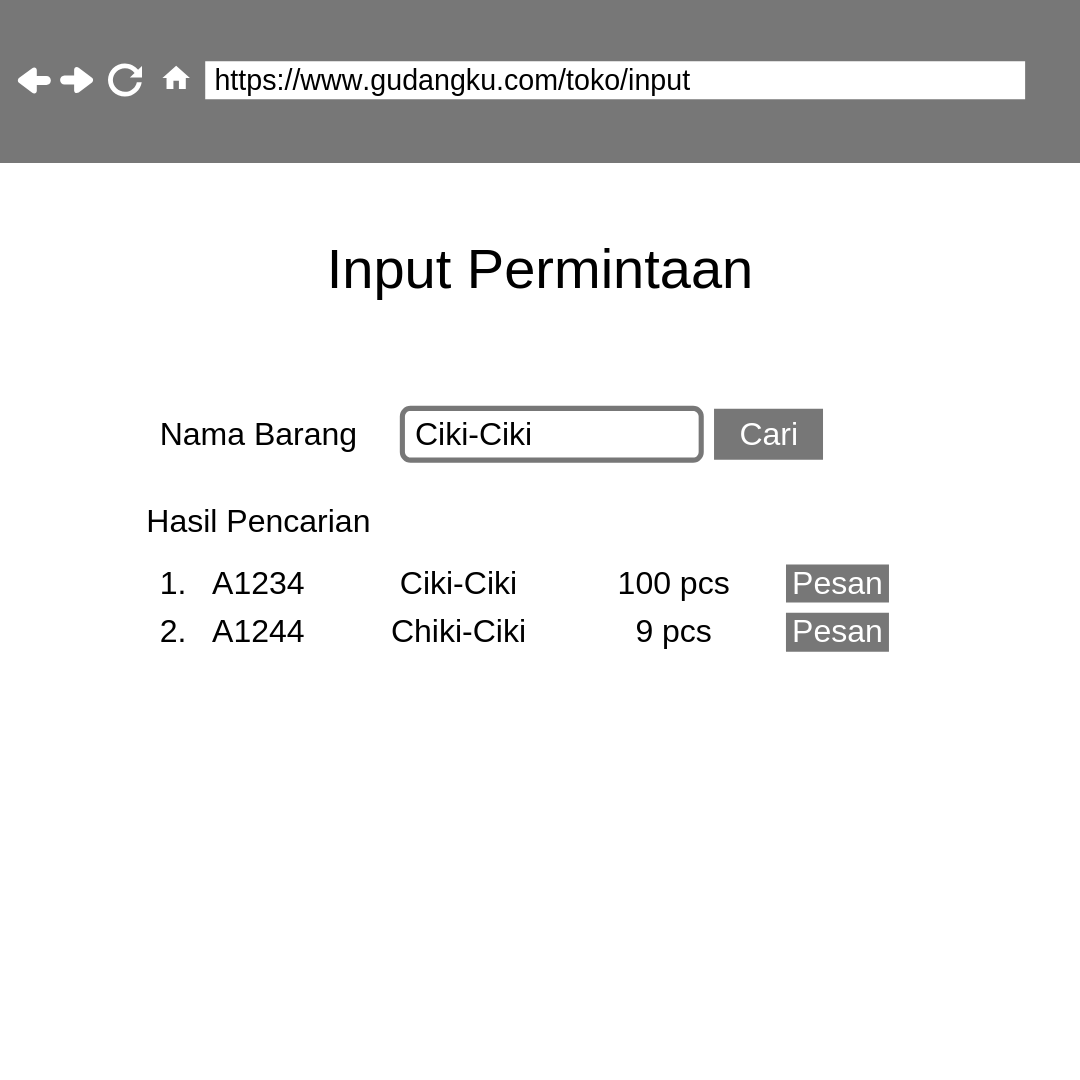
Sistem mengarahkan pengguna ke halaman input permintaan.

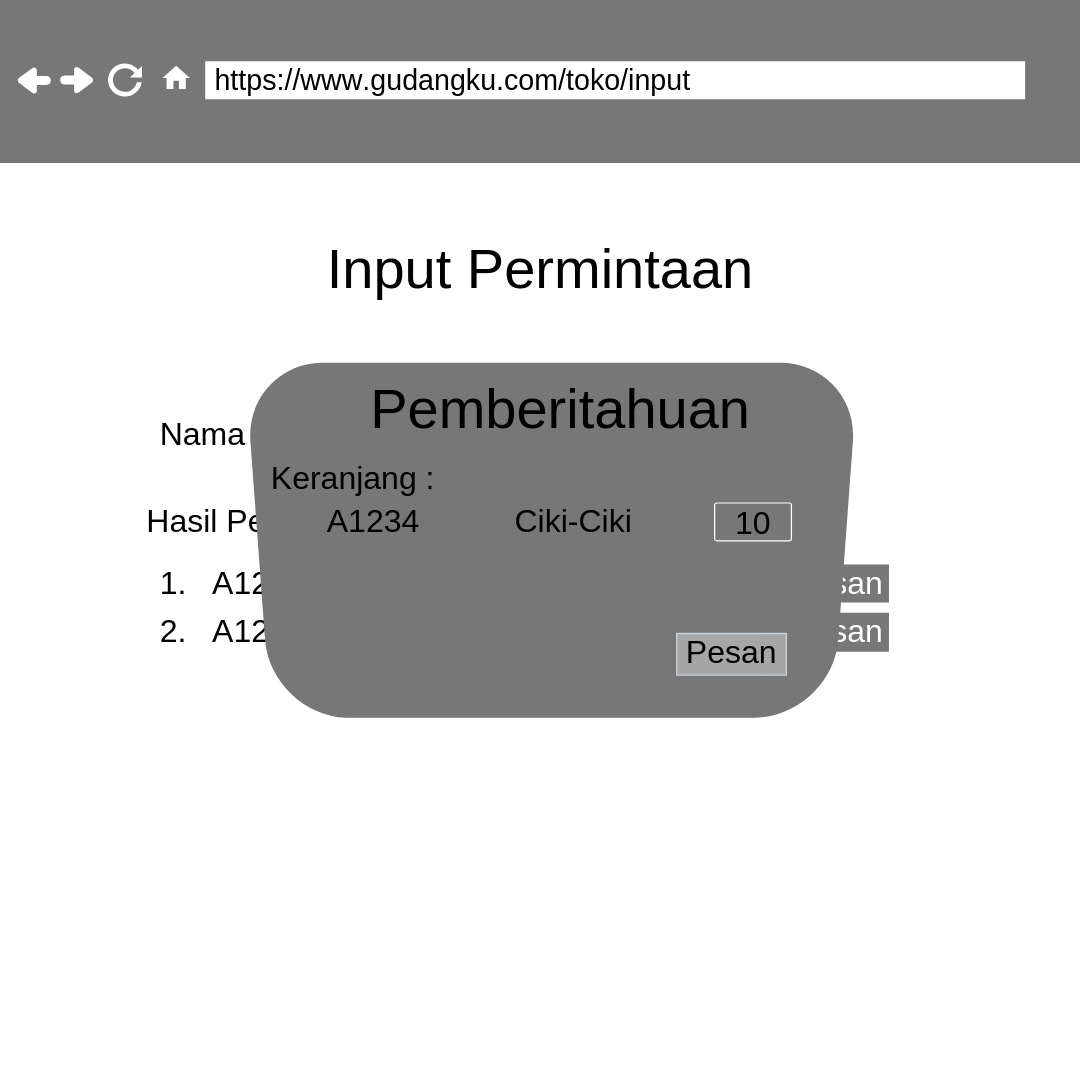
Pengguna mengisi formulir untuk meminta barang.

Pengguna mengklik tombol ‘pesan’.

Sistem merekam permintaan dan mengirimkannya ke pihak gudang yang dituju.

#### Perancangan Antarmuka Usecase #6





#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 6.1 | Page Input Permintaan | Menyediakan fitur pencarian barang yang ingin dicari dan dapat dipesan, serta membuat permintaan ke gudang. |

*Page Input Permintaan*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | CARI | Jika diklik, sistem akan mencari data di databse yang memiliki nama yang sama dan mendekati dengan apa yang dimasukkan. |
| Button2 | Button | PESAN | Jika diklik, sistem akan menampilkan alert box yang berguna untuk memasukkan jumlah barang yang ingin dipesan |
| Button3 | Button | PESAN (ALERT) | Jika diklik, sistem akan merekam permintaan dan dikirimkan ke gudang. |
| TextBox1 | TextBox | NAMA BARANG | Kotak teks untuk memasukkan nama barang yang ingin dicari dan dipesan. |

#### Identifikasi Object Baru

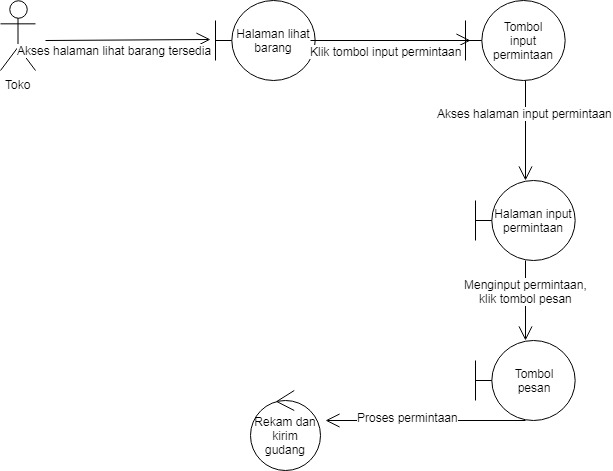
TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu permintaan* | *boundary* |
| *2* | *Display* | *controller* |
| *3* | *Page input permintaan* | *boundary* |
| *4* | *Database permintaan* | *entity* |

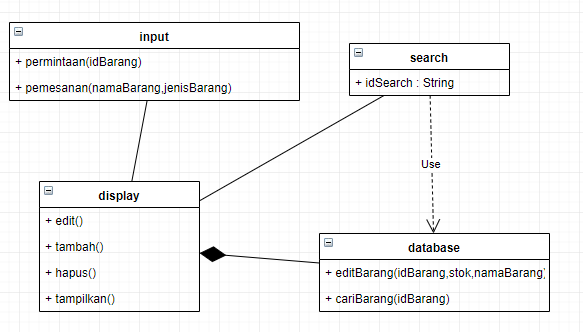
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Robustness Diagram

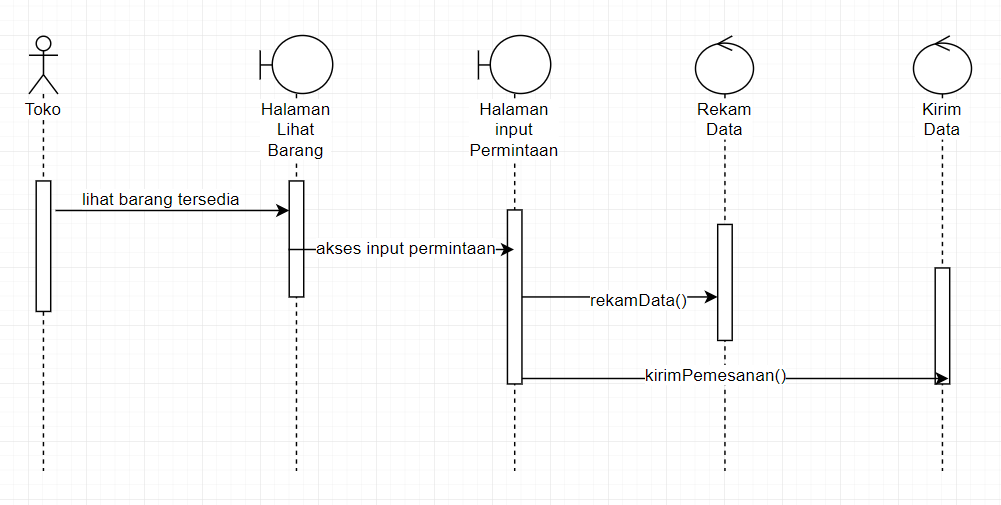
*Buatlah diagram robustness untuk masing – masing use case*



#### Diagram Kelasy

**

#### Sequence Diagram



### Use Case #7

*Skenario Use Case #7: Proses Pemesanan*

*Primary Flow :*

Pengguna berhasil login dan memasuki halaman input pemesanan.

Sistem menampilkan formulir pemesanan barang ke pihak logistik.

Pengguna mengisi formulir pemesanan barang dan mengklik ‘pesan’.

Sistem merekam data yang dimasukkan dan dikirimkan ke pihak logistik.

#### Perancangan Antarmuka Usecase #7



#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 7.1 | Page Proses Pemesanan | Menampilkan pemesanan yang dilakukan oleh gudang yang kekurangan stok/tidak ada barang. |

*Page Proses Pemesanan*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | YA | Jika diklik, sistem mengirimkan konfirmasi penerimaan pemesanan ke toko dan pemesanan diproses. |
| Button 2 | Button | TIDAK | Jika diklik, sistem akan mengirimkan konfirmasi penolakan pemesanan. |

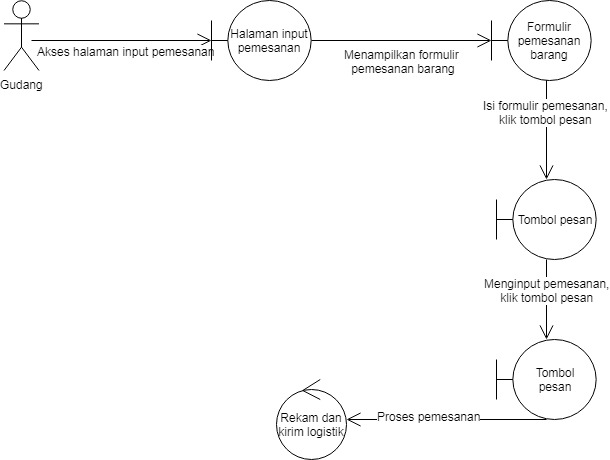
#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

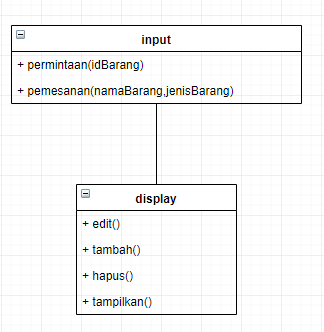
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu pemesanan* | *boundary* |
| *2* | *Display* | *controller* |
| *3* | *Page input pemesanan* | *boundary* |
| *4* | *Database pemesanan* | *entity* |

#### Robustness Diagram

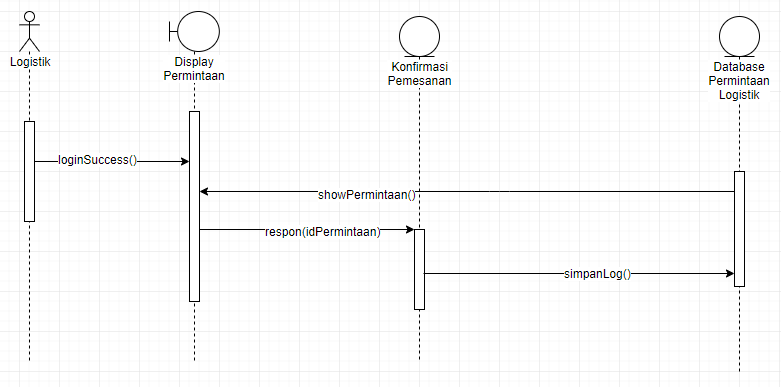
*Buatlah diagram robustness untuk masing – masing use case*

**

#### Diagram Kelas



#### Sequence Diagram

**

### Use Case #8

*Skenario Use Case #8 : Lihat Barang Tersedia*

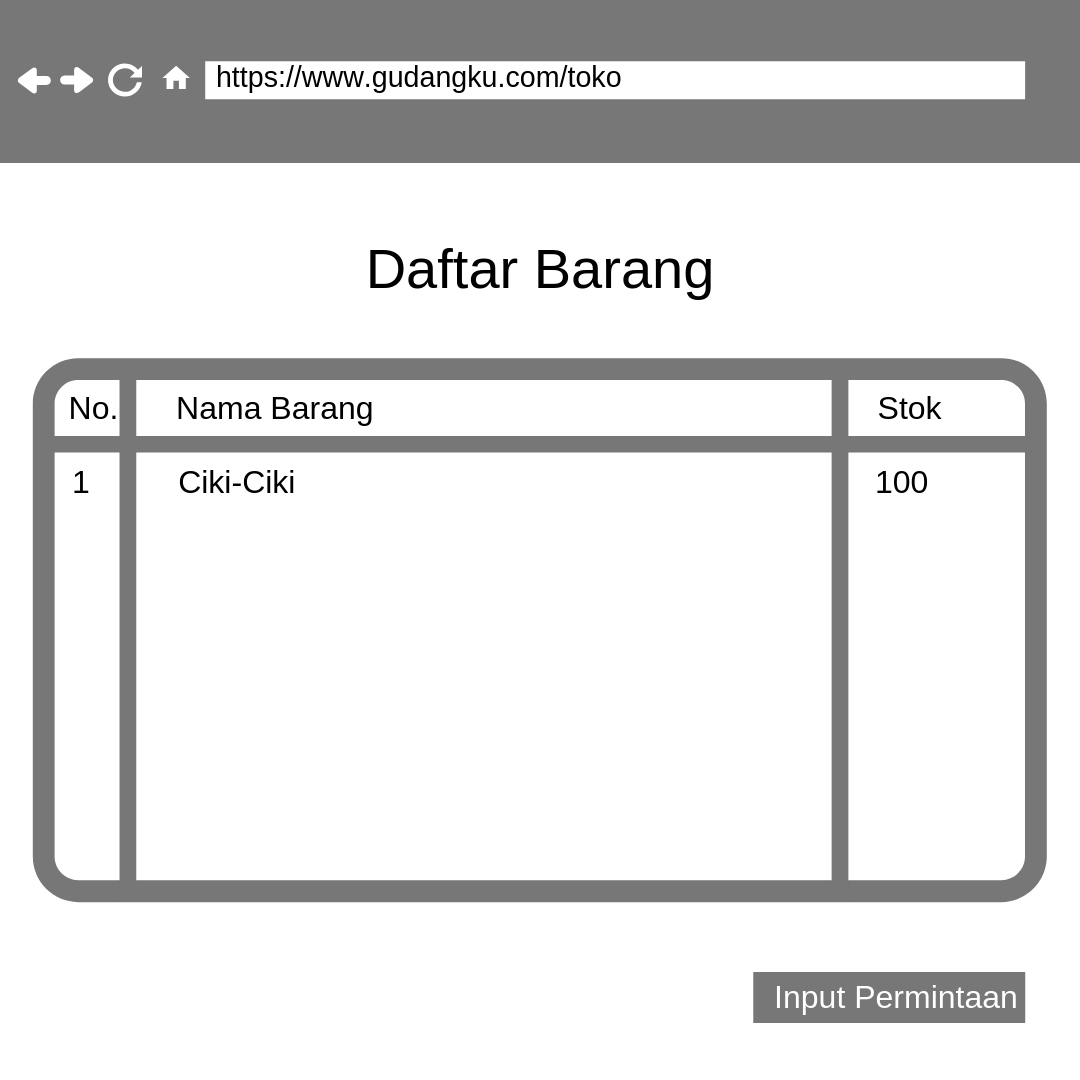
*Primary Flow :*

1. Pengguna berhasil login sebagai toko.
2. Sistem menampilkan data barang yang dapat dipesan.

*Alternate Flow :*

1. Pengguna berhasil login sebagai toko.
2. Sistem menampilkan data barang yang dapat dipesan.
3. Pengguna mengklik tombol ‘input pemesanan’.
4. Sistem mengarahkan pengguna ke halaman pemesanan.

#### Perancangan Antarmuka Usecase #8



#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 8.1 | Page Lihat Barang Tersedia | Menampilkan data barang yang dapat dipesan. |

*Page HOMEPAGE*

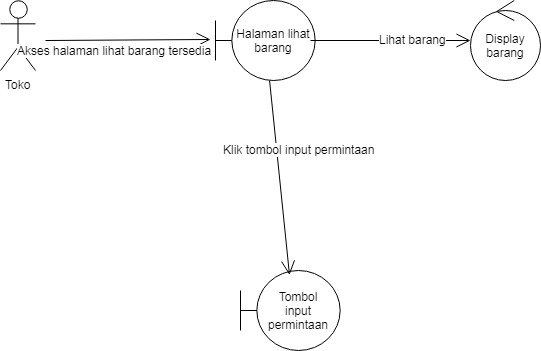
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| Button1 | Button | INPUT PERMINTAAN | Jika diklik, sistem akan mengarahkan pengguna ke halaman input permintaan. |

#### Identifikasi Object Baru

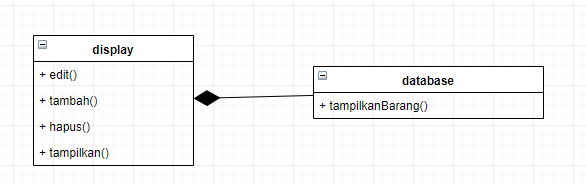
TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
|  |  |  |
|  |  |  |
|  |  |  |

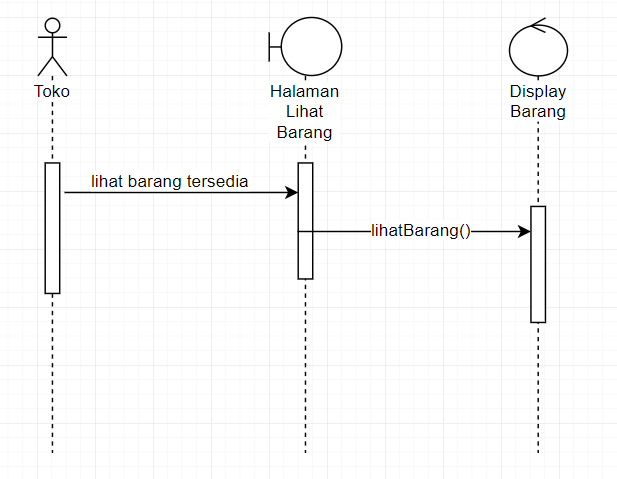
#### Robustness Diagram

**

#### Diagram Kelas

**

#### Sequence Diagram



### Use Case #9

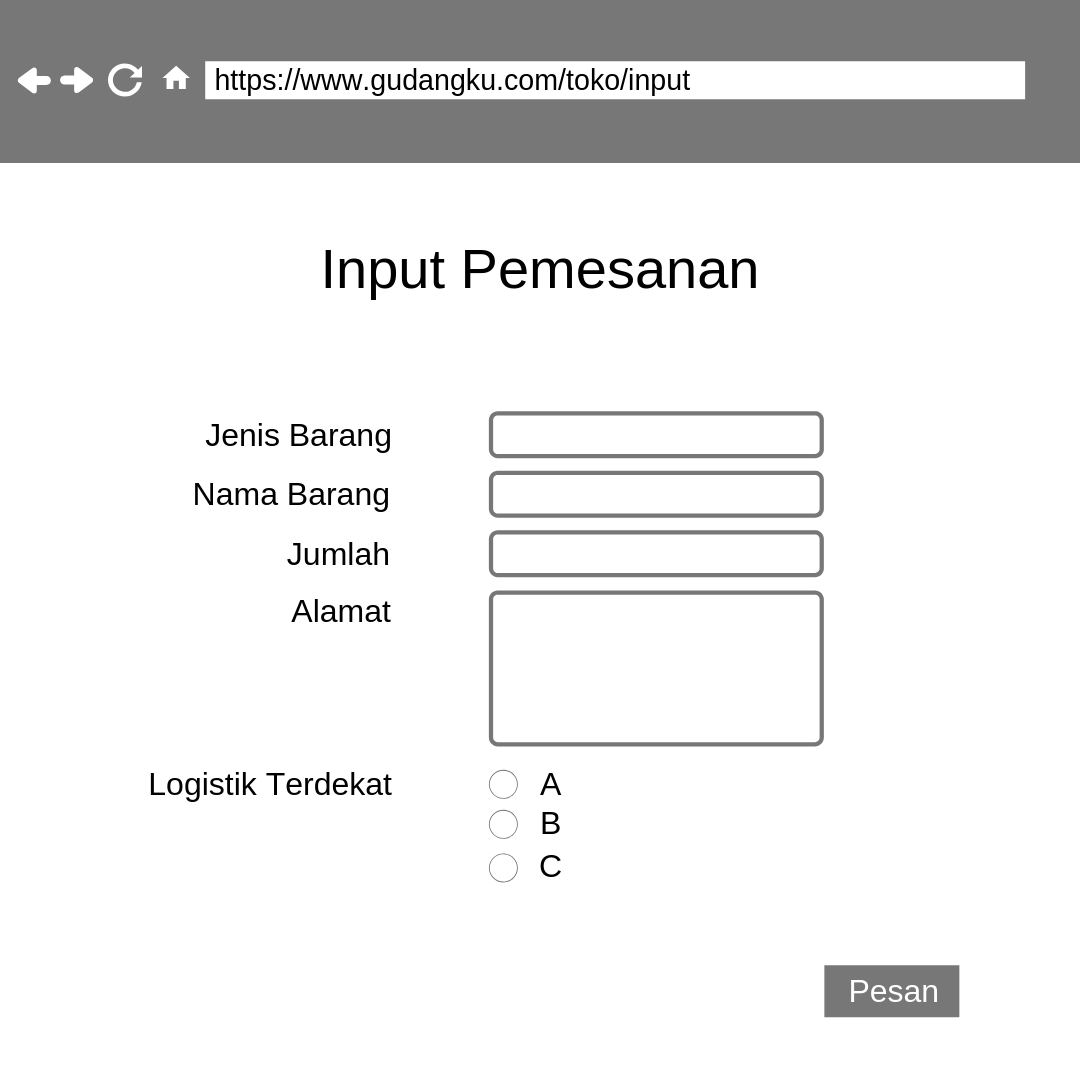
*Skenario Use Case #9 : Input Pemesanan*

*Primary Flow :*

1. Pengguna berhasil login sebagai gudang.
2. Pengguna masuk ke halaman input pemesanan.
3. Sistem menampilkan form pemesanan barang.
4. Pengguna mengisi form dan mengklik tombol ‘pesan’.
5. Sistem merekam data pemesanan.

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Pengguna berhasil login sebagai logistik. |  |
|  | 1. Sistem menampilkan pemesanan yang dikirimkan oleh pihak gudang. |
| 1. Logistik mengkonfirmasi pemesanan. |  |

#### Perancangan Antarmuka Usecase #10



#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 9.1 | Page Input Pemesanan | Menampilkan form yang dapat disini untuk melakukan pemesanan ke pihak logistik. |

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil

*Antarmuka XXX: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

*Page HOMEPAGE*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Button1* | Button | PESAN | Jika diklik, sistem akan menyimpan data pemesanan dan mencarikan logistik yang menjual barang yang dicari. |
| *TextBox1* | *TextBox* | *JENIS BARANG* | Kotak teks untuk memasukkan jenis barang yang ingin dipesan, |
| *TextBox2* | *TextBox* | *NAMA BARANG* | Kotak Teks untuk memasukkan nama barang yang ingin dipesan. |
| *TextBox3* | *TextBox* | *JUMLAH* | Kotak teks untuk memasukkan jumlah barang pemesanan. |
| *TextBox4* | *TextBox* | *ALAMAT* | Kotak teks untuk memasukkan alamat gudang. |
| *RadioButton* | *RadioButton* | *LOGISTIK TERDEKAT* | Menampilkan logistik yang paling dekat dengan alamat gudang. |

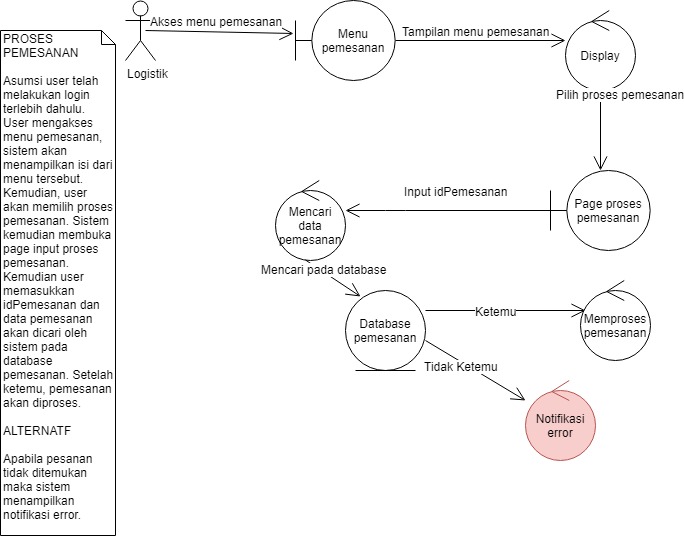
#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

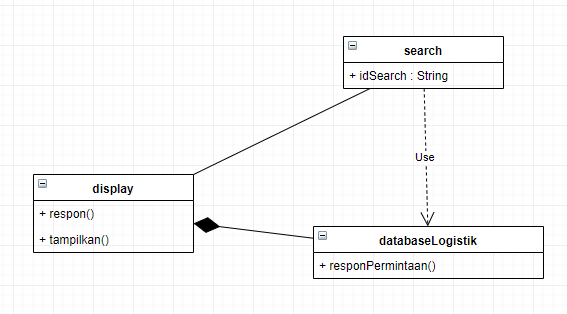
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *Menu pemesanan* | *boundary* |
| *2* | *Display* | *controller* |
| *3* | *Page proses pemesanan* | *boundary* |
| *4* | *Cari data pemesanan* | *controller* |
| *5* | *Database pemesanan* | *entity* |
| *6* | *Memproses pemesanan* | *controller* |
| *7* | *Notifikasi Error* | *controller* |

#### Robustness Diagram

*Buatlah diagram robustness untuk masing – masing use case*



#### Diagram Kelas

**

#### Sequence Diagram

# Perancangan Detil

#### Perancangan Detil Kelas

*Bagian ini diisi dengan daftar keseluruhan kelas yang akan digunakan dalam PL. menggunakan model MVC*

**TABEL KELAS :**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID Kelas** | **Nama Kelas Perancangan** | **Atribute (visibility)** | **Method / Operation** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*Untuk setiap kelas:*

* *identifikasi operasi (mengacu pada tanggung-jawab kelas),*
* *identifikasi atribut, termasuk visibility-nya*

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

Skema Relasi database yang digunakan

#### Perancangan Algoritma

#### Algoritma #1

*Nama Kelas : Login*

*Nama Operasi  : Login*

*Algoritma      :*

|  |
| --- |
| Input username  Input pass  if  (user klik login) then          i<-1          While (i<>nMax) do                   //cari username                  //cari pass          {endwhile}          If (Username dan pass ketemu) then                  //Buka menu selanjutnya          Else                 //Tidak Valid          {endif}  {endif} |

#### Algoritma #2

*Nama Kelas : Managemen Barang*

*Nama Operasi  : Menu*

*Algoritma      :*

|  |
| --- |
| repeat    Output //Tampilan menu//     Input pil     Case pil of:         1. //fungsi inputbarang         2. //fungsi viewbarang         3. //fungsi editbarang         4. //fungsi deletebarang         5. //fungsi inputpemesanan         6. //fungsi prosespemesanan  Until pil=0  {endrepeat} |

#### Algoritma #3

*Nama Kelas : Manajemen Barang*

*Nama Operasi  : Input Barang*

*Algoritma      :*

|  |
| --- |
| Input kodebarang  Input namabarang  If (klik tombol input) then    //simpan ke database  {endif} |

#### Algoritma #4

*Nama Kelas : Manajemen Barang*

*Nama Operasi  : View Barang*

*Algoritma      :*

|  |
| --- |
| //buka database  Output //list barang |

#### Algoritma #5

*Nama Kelas : Manajemen Barang*

*Nama Operasi  : Edit Barang*

*Algoritma      :*

|  |
| --- |
| Input kodebarang  i<-1  while(i<=nMax) do    //cari kodebarang  {endwhile}  if(kodebarang ketemu) then   //pilih yang ingin diganti   Input pil   Case pil of:       1. Input kodebarangnew           kodebarang<-kodebarangnew       2. Input namabarangnew           namabarang<-namabarangnew   {endcase}  {endif} |

#### Algoritma #6

*Nama Kelas : Manajemen Barang*

*Nama Operasi  : delete Barang*

*Algoritma      :*

|  |
| --- |
| Input kodebarang  i<-1  while(i<=nMax) do     //cari kodebarang  {endwhile}  If (kodebarang ketemu) then     //delete data barang  {endif} |

#### Algoritma #7

*Nama Kelas : Pemesanan*

*Nama Operasi  : Input Pemesanan*

*Algoritma      :*

|  |
| --- |
| Input idPesanan  Input namaPesanan  If (klik tombol input) then  Simpan ke database  {endif} |

#### Perancangan Query

Query :

|  |  |  |  |
| --- | --- | --- | --- |
| No Query | Query | | Keterangan |
| Q-001 | Select \* from barang | | Query untuk memanggil seluruh data barang |
| Q-002 | Select \* from admin | | Query untuk memanggil semua data admin |
| Q-003 | Select \* from users | | Query untuk memanggil semua data pengguna *website*. |
| Q-004 | Insert into orders values (‘namaBarang’, ‘jumlah’); | Query untuk toko melakukan permintaan barang ke gudang | |
| Q-005 | Insert into orderLogistic values (‘namaBarang’, ‘jumlah’); | Query untuk gudang melakukan pemesanan barang ke logistik | |
| Q-006 | Insert into users values (‘email’,’nama’,’username’,’password’, clientType); | Query untuk melakukan registrasi pengguna baru. | |
| Q-007 | update barang set ‘namaBarang’ to ‘namaBarangNew’ | Query untuk mengubah nama barang di *database*. | |
| Q-008 | Update barang set ‘sisaStock’ to ‘sisaStockNew’ | Query untuk mengubah sisa stock barang di *database*. | |
| Q-009 | delete from barang where namaBarang=’namaBarang’ | Query untuk menghapus data barang | |
| Q-010 | Update barang set namaBarang=’namaBarang’, sisaStock=’sisaStock’ | Query untuk mengubah data barang di *database* | |
| Q-011 | Insert into barang (namaBarang,sisaStock) values (‘namaBarang’,’sisaStock’) | Query untuk menambah barang di *database*. | |

# Matriks Kerunutan (Requirement Traceability Matrix)

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Fungsional Requirement** | **Nama Use Case** |
| FR1 | Login | Login |
| FR2 | Menambahkan barang | Tambah Barang |
| FR3 | Lihat data barang | Lihat Barang |
| FR4 | Edit data barang | Manajemen Barang |
| FR5 | Hapus data barang | Hapus Barang |
| FR6 | Memilih barang yang ingin dipesan | Input Pemesanan |
| FR7 | Memproses pemesanan barang | Proses Pemesanan |

# 