

SKPL-OO

**SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK
BERORIENTASI OBJEK**

WareHosea

untuk:


Hosea Nathanael Abetnego

Dipersiapkan oleh:

Kelompok 10

Denise Felicia Tiowanni	13522013
Erdianti Wiga Putri A	13522053
Shazya Audrea Taufik	13522063
Zahira Dina Amalia	13522085
Shulha	13522087
Tazkirah Amaliah	10023608

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132
2024**

	Program Studi Teknik Informatika STEI – ITB	Nomor Dokumen		Halaman
		SKPL-OO		<#>/<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

1 Pendahuluan	5
1.1 Tujuan Penulisan Dokumen	5
1.2 Lingkup Masalah	5
1.3 Definisi, Istilah dan Singkatan	5
Tabel 1.3.1. Definisi Istilah dan Singkatan	5
1.4 Aturan Penomoran	5
Tabel 1.4.1. Aturan Penomoran	5
1.5 Referensi	6
1.6 Deskripsi umum Dokumen (Ikhtisar)	6
2 Pendahuluan	7
2.1 Deskripsi Umum Sistem	7
Gambar 2.1.1 Flowchart Sistem	7
2.2 Deskripsi Umum Perangkat Lunak	7
2.3 Karakteristik Pengguna Perangkat Lunak	8
Tabel 2.3.1. Karakteristik Pengguna Perangkat Lunak	8
2.4 Batasan Perangkat Lunak	8
2.5 Lingkungan Operasi Perangkat Lunak	8
3 Deskripsi Kegiatan	9
3.1 Kebutuhan Fungsional	9
Tabel 3.1.1. Kebutuhan Fungsional Aplikasi	9
3.2 Kebutuhan Non Fungsional	9
Tabel 3.2.1. Kebutuhan Non Fungsional Aplikasi	9
3.3 Model Use Case	11
3.3.1 Diagram Use Case	11
3.3.2 Definisi Aktor	12
Tabel 3.3.2.1. Definisi Actor	12
3.3.3 Definisi Use Case	12
Tabel 3.3.3.1. Definisi Actor	12
3.3.4 Skenario Use Case	12
4 Model Analisis	14
4.1 Identifikasi Kelas	14
Tabel 4.1.1. Identifikasi Kelas	14
4.2 Diagram Kelas	14
Gambar 1.2.1. Diagram Kelas	14
Tabel 4.2.1. Keterangan Elemen Diagram Kelas	14
5 Keruntutan (traceability)	17
5.1 Kebutuhan Fungsional vs Use Case	17
5.2 Use Case vs Kelas Terkait	17

Generate daftar isi dengan cara otomatis. Jika terdapat gambar dan tabel dalam dokumen, maka halaman ini wajib diikuti oleh halaman daftar tabel dan daftar gambar.

1 Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen ini dibuat untuk pembuatan aplikasi WareHosea, yaitu aplikasi pengelola *warehouse* yang dipesan oleh Hosea Nathanael Abetnego. Dokumen ini digunakan oleh developer untuk menyamakan desain dan fungsi dari aplikasi yang dibuat, serta untuk mendesain program dengan model yang berorientasi objek.

1.2 Lingkup Masalah

WareHosea adalah sistem perangkat lunak *storage manager* berbasis aplikasi desktop yang dirancang untuk memudahkan administrator dalam mengelola dan melacak barang-barang di beberapa *warehouse*. Sistem ini akan menampilkan nama barang, kode barang, deskripsi barang, jumlah stok barang, gambar barang, status pengiriman barang, dan tanggal masuk barang. Sistem juga menyajikan informasi tentang lokasi dan kapasitas *warehouse*. Pengguna dapat melakukan pengelolaan stok barang-barang yang ada di berbagai *warehouse* dengan fungsi *create*, *update*, *delete*, dan *read* (CRUD). Selain itu, dengan sistem ini pengguna dapat melakukan pemindahan barang dari *warehouse* yang satu ke *warehouse* yang lainnya, baik satu barang maupun semua barang sekaligus. Sistem ini dapat melakukan pencarian suatu barang berdasarkan kata kunci yang diberikan menggunakan fungsi *search*.

1.3 Definisi, Istilah dan Singkatan

Tabel 1.3.1. Definisi Istilah dan Singkatan

Singkatan, Akronim, atau Istilah	Penjelasan
Spesifikasi Kebutuhan Perangkat Lunak (SKPL)	Dokumen penjelasan deskripsi detail P/L yang akan dibangun yang menjadi bentuk komitmen antara pengembang dan pengguna.
<i>Operating System</i> (OS)	Perangkat lunak yang menjadi jembatan antara perangkat keras komputer dengan program perangkat lunak yang digunakan pengguna
<i>Database Management System</i> (DBMS)	Sistem pengorganisasian dan pengolahan basis data pada komputer.

1.4 Aturan Penomoran

Tabel 1.4.1. Aturan Penomoran

Hal / Bagian	Penomoran	Keterangan
Kebutuhan fungsional	F<XX>	XX berupa nomor terurut
Kebutuhan non fungsional	N<XX>	XX berupa nomor terurut

1.5 Referensi

Materi perkuliahan IF2250 Rekayasa Perangkat Lunak T.A. 2023-2024.

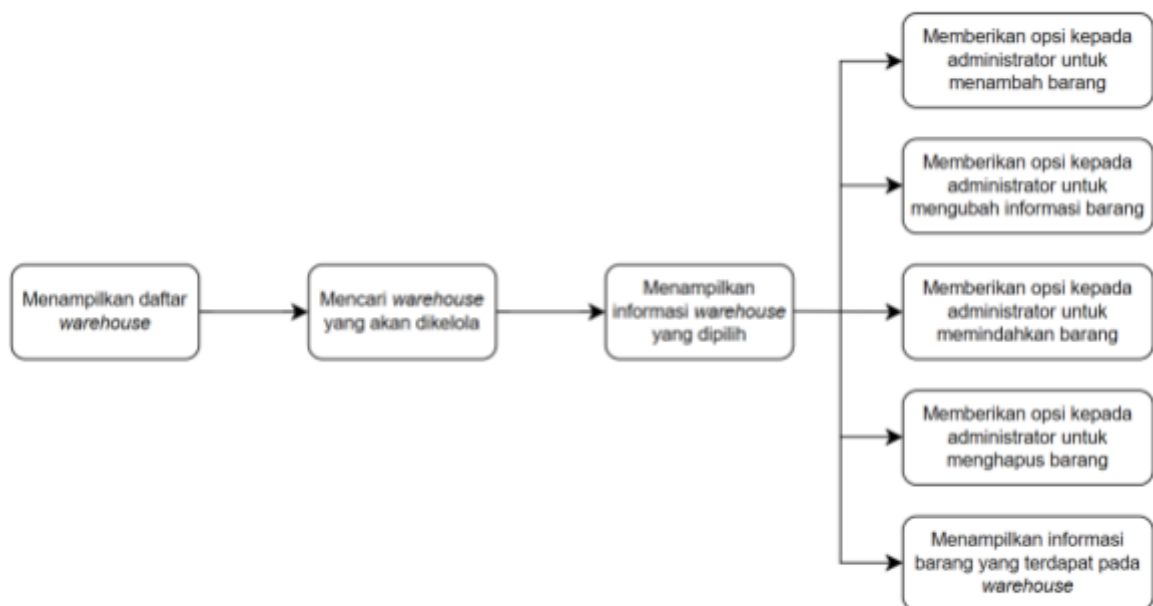
1.6 Deskripsi umum Dokumen (Ikhtisar)

Dokumen ini mencakup gambaran umum mengenai sistem dan perangkat lunak yang akan dikembangkan, termasuk karakteristik pengguna, pembatasan sistem, serta lingkungan operasional perangkat lunak. Selain itu, dokumen ini juga mencantumkan kebutuhan perangkat lunak yang akan dibuat, termasuk kebutuhan fungsional maupun non fungsional, *data flow diagram*, dan kebutuhan data perangkat lunak. Selanjutnya, akan dijelaskan pembatasan dalam desain serta urutan langkah-langkah dalam pengembangan perangkat lunak.

2 Pendahuluan

2.1 Deskripsi Umum Sistem

WareHosea adalah sebuah aplikasi desktop yang berfungsi sebagai storage manager untuk membantu administrator dalam mengelola dan melacak barang-barang di berbagai *warehouse*. Fitur utama yang terdapat pada aplikasi ini adalah menyediakan tampilan berbagai *warehouse* beserta data barang di dalamnya. Aplikasi ini akan menampilkan informasi terkait barang seperti nama barang, kode barang, deskripsi, jumlah stok, gambar, status pengiriman, dan tanggal masuk. Selain itu, aplikasi juga akan menampilkan lokasi *warehouse* beserta kapasitas *warehouse* (kapasitas maksimum dan terkini). Melalui aplikasi ini, administrator dapat melakukan pengelolaan stok barang dengan fungsi *create*, *update*, *delete*, dan *read* (CRUD), artinya administrator dapat menambah, mengubah, menghapus, dan melihat data barang dengan mudah. Fitur ini menjadi primitif untuk fitur selanjutnya, yaitu administrator dapat melakukan pemindahan barang dari satu *warehouse* ke *warehouse* lainnya, baik itu dilakukan untuk satu barang atau semua barang sekaligus. Fitur terakhir yang terdapat pada aplikasi ini adalah menyediakan pencarian *warehouse* dan barang berdasarkan kata kunci yang diberikan, sehingga memudahkan untuk menemukan *warehouse* dan barang tertentu di antara stok yang tersedia.



Gambar 2.1.1 Flowchart Sistem

2.2 Deskripsi Umum Perangkat Lunak

Aplikasi WareHosea memiliki serangkaian fungsional yang memungkinkan administrator untuk mengelola *warehouse* secara efektif. Fungsional pertama adalah sistem memungkinkan administrator untuk membuat *warehouse* baru dengan menambahkannya ke dalam database. Fungsional kedua, yaitu sistem menyediakan daftar lengkap *warehouse* beserta detailnya kepada administrator. Informasi yang disediakan meliputi lokasi, kapasitas maksimum, dan kapasitas terkini dari setiap *warehouse*. Fungsional ketiga adalah administrator dapat menggunakan sistem untuk mengubah informasi *warehouse*, termasuk lokasi dan kapasitas. Selanjutnya terdapat pula fungsional yang memungkinkan administrator untuk menghapus *warehouse* dari database. Tidak hanya dapat mengelola *warehouse*, aplikasi ini juga memiliki fungsional untuk mengelola barang pada *warehouse* juga. Fungsional selanjutnya adalah administrator dapat menambahkan barang baru ke dalam suatu *warehouse*. Fungsional ke-6 yaitu menyediakan detail lengkap tentang setiap barang yang tersedia di setiap *warehouse* kepada administrator. Fungsional ke-7, administrator dapat mengubah informasi setiap barang di setiap *warehouse*, seperti jumlah stok barang, status pengiriman barang, dan tanggal masuk barang sesuai

dengan kondisi terbaru melalui sistem. Sebagai primitif dari fungsional ke-7, terdapat fungsional ke-8 dan ke-9 yaitu penghapusan barang tertentu dari suatu *warehouse* oleh administrator dan pemindahan barang antar *warehouse* oleh administrator, baik satu barang atau seluruh barang sekaligus. Fungsional terakhir adalah fitur pencarian (*search*) yang memungkinkan administrator untuk mencari *warehouse* dan barang tertentu dengan mudah sesuai kata kunci yang diberikan.

2.3 Karakteristik Pengguna Perangkat Lunak

Tabel 2.3.1. Karakteristik Pengguna Perangkat Lunak

Kategori Pengguna	Tugas	Hak Akses ke aplikasi
Administrator	Mengelola seluruh <i>warehouse</i> pada sistem dengan fungsi membuat, melihat, menghapus, memindahkan, dan memperbaharui informasi barang pada <i>warehouse</i> .	Memiliki akses keseluruhan terhadap fungsionalitas serta data dari aplikasi.

2.4 Batasan Perangkat Lunak

- Database yang digunakan bersifat lokal
- Berfungsi secara multiplatform (Windows dan Linux)
- Perangkat tidak memerlukan akses internet untuk dijalankan
- Perangkat lunak tidak memerlukan *browser* untuk dijalankan

2.5 Lingkungan Operasi Perangkat Lunak

Aplikasi WareHosea akan berfungsi dengan spesifikasi sebagai berikut.

OS : Windows, Linux

DBMS : MariaDB

3 Deskripsi Kegiatan

3.1 Kebutuhan Fungsional

Tabel 3.1.1. Kebutuhan Fungsional Aplikasi

ID	Pernyataan Kebutuhan	Penjelasan
F01	P/L memiliki fungsi di mana admin dapat membuat warehouse baru	Admin dapat menambah warehouse baru ke database
F02	P/L dapat menampilkan daftar warehouse beserta detailnya	Admin dapat mengakses informasi dari semua warehouse meliputi lokasi, kapasitas maksimum, dan kapasitas terkini.
F03	P/L memiliki fungsi di mana admin dapat mengubah informasi warehouse	Admin dapat mengubah informasi dari warehouse yang meliputi lokasi, dan kapasitas warehouse.
F04	P/L memiliki fungsi di mana admin dapat menghapus warehouse	Admin dapat menghapus warehouse dari database
F05	P/L memiliki fungsi untuk menambah barang pada warehouse	Admin dapat menambahkan barang baru pada suatu warehouse.
F06	P/L dapat menampilkan detail setiap barang yang berada di warehouse	Admin dapat mengakses informasi dari semua barang yang ada di semua warehouse.
F07	P/L memiliki fungsi di mana admin dapat mengubah informasi setiap barang pada setiap warehouse	Admin mengubah informasi setiap barang meliputi jumlah stok barang, status pengiriman barang, dan tanggal masuk barang sesuai dengan kondisi terbaru.
F08	P/L memiliki fungsi di mana admin dapat menghapus suatu barang dari suatu warehouse	Admin dapat menghapus barang dari suatu warehouse.
F09	P/L memiliki fungsi di mana admin dapat memindahkan barang antar warehouse	Admin dapat menentukan pemindahan barang dari satu warehouse ke warehouse lain, baik itu satu barang ataupun seluruh barang sekaligus.
F10	P/L memiliki fungsi di mana admin dapat melakukan pencarian melalui fitur <i>search</i>	Admin dapat memanfaatkan fitur <i>search</i> untuk mencari warehouse tertentu.

3.2 Kebutuhan Non Fungsional

Tabel 3.2.1. Kebutuhan Non Fungsional Aplikasi

ID	Parameter	Kebutuhan
NF01	Availability	Sistem harus dapat beroperasi terus menerus selama 7 hari 24 jam per hari, 7 hari per minggu, 30 hari per bulan.
NF02	Reliability	Sistem tidak boleh mendapatkan input yang salah dari admin.
NF03	Portability	Sistem dapat dijalankan di laptop/PC.
NF04	Memory	Sistem tidak memberatkan perangkat user dengan maksimal memory yang digunakan adalah 200 MB.
NF05	Response time	Aplikasi harus mampu menampilkan detail informasi baik barang-barang maupun warehouse dalam 3 detik. Aplikasi harus mampu menampilkan hasil <i>search</i> dalam 1 detik. Aplikasi harus mampu mengupdate informasi dalam 3 detik.

NF06	Security	Sistem harus menyediakan <i>security check</i> (semacam <i>password</i>) untuk akses P/L khusus administrator agar sistem tidak dapat diakses oleh selain admin.
NF07	Maintainability	Jangka waktu maintenance P/L setiap bulan dengan durasi maintenance antara 2-4 jam agar selalu sesuai dengan availability sistem.
NF08	Bahasa komunikasi	Sistem dapat merespon Bahasa Indonesia dan Bahasa Inggris.

3.3 *Model Use Case*

3.3.1 *Diagram Use Case*



Gambar 3.3.1.1. Use Case Diagram System

3.3.2 Definisi Aktor

Bagian ini diisi dengan daftar aktor dan deskripsi role untuk aktor tersebut. Deskripsi role harus menjelaskan wewenang pada role tersebut dalam perangkat lunak. Bisa dibuat dalam bentuk tabel berikut:

Tabel 3.3.2.1. Definisi Actor

No	Actor	Deskripsi
1	Administrator	Aktor ini memiliki wewenang untuk mengelola seluruh <i>warehouse</i> pada sistem dengan fungsi membuat, melihat, menghapus, memindahkan, dan memperbaharui informasi barang pada <i>warehouse</i> . Actor ini Memiliki akses keseluruhan terhadap fungsionalitas serta data dari aplikasi.

3.3.3 Definisi Use Case

Bagian ini diisi dengan daftar use case dan deskripsi singkat mengenai use case tersebut. Bisa dibuat dalam bentuk tabel berikut:

Tabel 3.3.3.1. Definisi Actor

ID	Use Case	Deskripsi
U01	Memasukkan ID dan password	Aktor Administrator memasukkan ID dan password untuk dapat mengakses aplikasi.
U02	Mencari <i>warehouse</i>	Aktor Administrator memasukkan kata kunci untuk mencari <i>warehouse</i> yang diinginkan.
U03	Melihat daftar <i>warehouse</i>	Aktor Administrator melihat kumpulan <i>warehouse</i> yang terdaftar di sistem.
U04	Membuat <i>warehouse</i> baru	Aktor Administrator menambahkan suatu <i>warehouse</i> baru ke dalam sistem.
U05	Menghapus <i>warehouse</i>	Aktor Administrator menghapus suatu <i>warehouse</i> dari sistem.
U06	Memperbaharui data <i>warehouse</i>	Aktor Administrator mengubah informasi data <i>warehouse</i> dan mengupdatenya ke sistem.
U07	Mencari barang	Aktor Administrator memasukkan kata kunci untuk mencari barang yang diinginkan.
U08	Menambah barang pada <i>warehouse</i>	Aktor Administrator menambahkan barang ke suatu <i>warehouse</i> dan datanya akan disimpan ke dalam sistem.
U09	Menghapus barang pada <i>warehouse</i>	Aktor Administrator menghapus barang dari suatu <i>warehouse</i> dan datanya akan disimpan ke dalam sistem.
U10	Memperbaharui data barang	Aktor Administrator mengubah informasi suatu data barang dan mengupdatenya ke sistem.
U11	Memindahkan barang dari satu <i>warehouse</i> ke <i>warehouse</i> lain	Aktor Administrator mengubah informasi suatu data barang dari satu <i>warehouse</i> ke <i>warehouse</i> lain.

3.3.4 Skenario Use Case

3.3.4.1 Skenario Use Case Memasukkan ID dan Password

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman <i>login</i> .
2. Admin memasukkan ID beserta <i>password</i> -nya.	
	3. Sistem menampilkan laman utama.
Skenario Alternatif 1	
	1. Sistem menampilkan laman <i>login</i> .
2. Admin belum memiliki akun, sehingga harus membuat akun terlebih dahulu.	
	3. Sistem menampilkan laman pembuatan akun.
4. Admin memasukkan ID dan <i>password</i> untuk akun barunya.	
	5. Sistem menampilkan laman <i>login</i> .
6. Admin memasukkan ID beserta <i>password</i> -nya.	
	7. Sistem menampilkan laman utama.
Skenario Alternatif 2	
	1. Sistem menampilkan laman <i>login</i> .
2. Admin memasukkan ID atau <i>password</i> yang salah.	
	3. Sistem menampilkan notifikasi “ID atau <i>password</i> Anda salah. Harap coba lagi.”
	4. Sistem menampilkan laman <i>login</i> .

3.3.4.2 Skenario Use Case Mencari *Warehouse*

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin memasukkan kata kunci berupa nama <i>warehouse</i> .	
	3. Sistem menampilkan daftar <i>warehouse</i> .

Skenario Alternatif	
	1. Sistem menampilkan laman utama.
2. Admin memasukkan kata kunci berupa nama <i>warehouse</i> yang tidak tersedia pada aplikasi.	
	3. Sistem menampilkan notifikasi “Pencarian tidak ditemukan”.

3.3.4.3 Skenario Use Case Mencari Barang

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin memasukkan kata kunci berupa nama barang.	
	3. Sistem menampilkan daftar <i>barang</i> .
Skenario Alternatif	
	1. Sistem menampilkan laman utama.
2. Admin memasukkan kata kunci berupa nama barang yang tidak tersedia pada aplikasi.	
	3. Sistem menampilkan notifikasi “Pencarian tidak ditemukan”.

3.3.4.4 Skenario Use Case Melihat Daftar Warehouse

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan daftar <i>warehouse</i> .

3.3.4.5 Skenario Use Case Membuat *Warehouse* Baru

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .

4. Admin menekan tombol “Tambah”.	
	5. Sistem menampilkan laman penambahan data <i>warehouse</i> baru.
6. Admin melengkapi data pada laman tersebut meliputi ID <i>warehouse</i> , nama <i>warehouse</i> , kapasitas maksimum, kapasitas terkini, dan alamat <i>warehouse</i> .	
	7. Sistem menambahkan data <i>warehouse</i> baru ke database dan daftar <i>warehouse</i> .
Skenario Alternatif 1	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Tambah”.	
	5. Sistem menampilkan laman penambahan data <i>warehouse</i> baru.
6. Admin menekan tombol ‘X’ untuk membatalkan penambahan <i>warehouse</i> .	
	7. Sistem kembali menampilkan laman daftar <i>warehouse</i> .
Skenario Alternatif 2	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Tambah”.	
	5. Sistem menampilkan laman penambahan data <i>warehouse</i> baru.
6. Admin memasukkan ID <i>warehouse</i> atau nama <i>warehouse</i> yang sudah tersedia di database.	
	7. Sistem menampilkan notifikasi “ <i>Warehouse</i> sudah terdaftar. Harap ubah ID atau nama <i>warehouse</i> .”
	8. Sistem menampilkan laman penambahan data <i>warehouse</i> baru (kembali ke skenario normal no 5).
Skenario Alternatif 3	

	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Tambah”.	
	5. Sistem menampilkan laman penambahan data <i>warehouse</i> baru.
6. Admin tidak memasukkan salah satu/beberapa atribut <i>warehouse</i> .	
	7. Sistem menampilkan notifikasi “Atribut tidak boleh kosong. Harap lengkapi informasi.”
	8. Sistem menampilkan laman penambahan data <i>warehouse</i> baru (kembali ke skenario normal no 5).

3.3.4.6 Skenario Use Case Menghapus *Warehouse*

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Hapus”.	
	5. Sistem menampilkan notifikasi konfirmasi “Apakah Anda yakin untuk menghapus <i>warehouse</i> ini?”.
6. Admin memilih tombol “Ya”.	
	7. Sistem menghapus <i>warehouse</i> dari database.
Skenario Alternatif 2	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Hapus”.	
	5. Sistem menampilkan notifikasi konfirmasi “Apakah Anda yakin untuk menghapus <i>warehouse</i> ini?”.

6. Admin memilih tombol “Tidak”.	
	7. Sistem menampilkan laman daftar <i>warehouse</i> .
Skenario Alternatif 3	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan tombol “Hapus”.	
	5. Sistem menampilkan notifikasi konfirmasi “Apakah Anda yakin untuk menghapus <i>warehouse</i> ini?”.
6. Admin memilih tombol “X”.	
	7. Sistem menampilkan laman daftar <i>warehouse</i> .

3.3.4.7 Skenario Use Case Memperbaharui Data *Warehouse*

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan <i>warehouse</i> yang ingin diedit.	
	5. Sistem menampilkan informasi <i>warehouse</i> yang dipilih.
6. Admin menekan tombol “Edit”.	
	7. Sistem menampilkan laman pengeditan data <i>warehouse</i> .
8. Admin mengubah/melengkapi data pada laman tersebut meliputi kapasitas maksimum, kapasitas terkini, dan alamat <i>warehouse</i> , kecuali ID <i>warehouse</i> dan nama <i>warehouse</i> .	
	9. Sistem memperbaharui data <i>warehouse</i> dan menyimpannya dalam database.
Skenario Alternatif	
	1. Sistem menampilkan laman utama.

2. Admin menekan tombol “Daftar <i>Warehouse</i> ”.	
	3. Sistem menampilkan laman daftar <i>warehouse</i> .
4. Admin menekan <i>warehouse</i> yang ingin diedit.	
	5. Sistem menampilkan informasi <i>warehouse</i> yang dipilih.
6. Admin menekan tombol “Edit”.	
	7. Sistem menampilkan laman pengeditan data <i>warehouse</i> .
8. Admin menekan tombol ‘X’ untuk membatalkan pengeditan data <i>warehouse</i> .	
	9. Sistem kembali menampilkan informasi <i>warehouse</i> .

3.3.4.8 Skenario Use Case Melihat Data Barang

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar Barang”.	
	3. Sistem menampilkan daftar barang.

3.3.4.9 Skenario Use Case Menambah Barang pada *Warehouse*

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar Barang”	
	3. Sistem menampilkan daftar barang <i>warehouse</i> .
4. Admin menekan tombol “Tambah”	
	5. Sistem menampilkan laman penambahan data barang baru.
6. Admin melengkapi data pada laman tersebut meliputi ID barang, nama barang, foto barang, deskripsi barang, stok barang, serta status pengiriman barang.	
	7. Sistem menambahkan data barang baru ke database dan daftar barang.

Skenario Alternatif 1	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar Barang”	
	3. Sistem menampilkan daftar barang <i>warehouse</i> .
4. Admin menekan tombol ‘X’ untuk membatalkan penambahan barang.	
	5. Sistem kembali menampilkan laman daftar barang.
Skenario Alternatif 2	
	1. Sistem menampilkan laman utama.
2. Admin memasukkan ID barang yang sudah tersedia di database.	
	3. Sistem menampilkan notifikasi “Barang sudah terdaftar. Harap ubah ID barang.”
	4. Sistem menampilkan laman penambahan data barang baru (kembali ke skenario normal no 3).
Skenario Alternatif 3 (ada data kosong)	
	1. Sistem menampilkan laman utama.
2. Admin tidak memasukkan salah satu atribut barang.	
	3. Sistem menampilkan notifikasi “Atribut tidak boleh kosong. Harap lengkapi informasi.”

3.3.4.10 Skenario Use Case Menghapus Barang pada *Warehouse*

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman daftar barang.
2. Admin menekan tombol “Hapus”.	
	3. Sistem menampilkan laman penghapusan data barang.
4. Admin mengkonfirmasi penghapusan.	
	5. Sistem menghapus data barang dari database dan daftar barang.
Skenario Alternatif 1	

	1. Sistem menampilkan laman daftar barang.
2. Admin menekan tombol “Hapus”.	
	3. Sistem menampilkan laman penghapusan data barang.
4. Admin menekan tombol ‘X’ untuk membatalkan penghapusan barang.	
	5. Sistem kembali menampilkan laman daftar barang.

3.3.4.11 Skenario Use Case Memperbaharui Data Barang

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar Barang”.	
	3. Sistem menampilkan daftar barang.
4. Admin memilih barang yang ingin diperbaharui.	
	5. Sistem menampilkan informasi barang.
6. Admin menekan tombol “Edit”.	
	7. Sistem menampilkan laman pengeditan data barang.
8. Admin mengubah/melengkapi data pada laman tersebut meliputi nama barang, foto, deskripsi, stok, dan status pengiriman, kecuali ID barang.	
	9. Sistem menambahkan data barang baru ke database dan daftar barang.
Skenario Alternatif	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol “Daftar Barang”.	
	3. Sistem menampilkan daftar barang.
4. Admin memilih barang yang ingin diperbaharui.	
	5. Sistem menampilkan informasi barang.
6. Admin menekan tombol “Edit”.	

	7. Sistem menampilkan laman pengeditan data barang.
8. Admin menekan tombol 'X' untuk membatalkan pengeditan barang.	
	9. Sistem kembali menampilkan informasi barang.

3.3.4.12 Skenario Use Case Memindahkan Barang dari Satu *Warehouse* ke *Warehouse* Lain

Aksi Admin	Reaksi Sistem
Skenario Normal	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol "Daftar Barang".	
	3. Sistem menampilkan daftar barang.
4. Admin mengklik barang yang ingin dipindahkan.	
	5. Sistem menampilkan informasi barang.
6. Admin mengedit status pengiriman barang, seperti memasukkan lokasi barang.	
	7. Sistem menampilkan laman daftar barang dengan status pengiriman barang yang terbaru.
Skenario Alternatif 1	
	1. Sistem menampilkan laman utama.
2. Admin menekan tombol "Daftar Barang".	
	3. Sistem menampilkan daftar barang.
4. Admin mengklik barang yang ingin dipindahkan.	
	5. Sistem menampilkan informasi barang.
6. Admin memasukkan lokasi barang yang sama dengan lokasi barang saat ini	
	7. Sistem menampilkan notifikasi "Lokasi barang setelah dipindahkan tidak boleh sama seperti sebelumnya. Harap masukkan lokasi kembali."
	8. Sistem kembali menampilkan laman daftar barang.

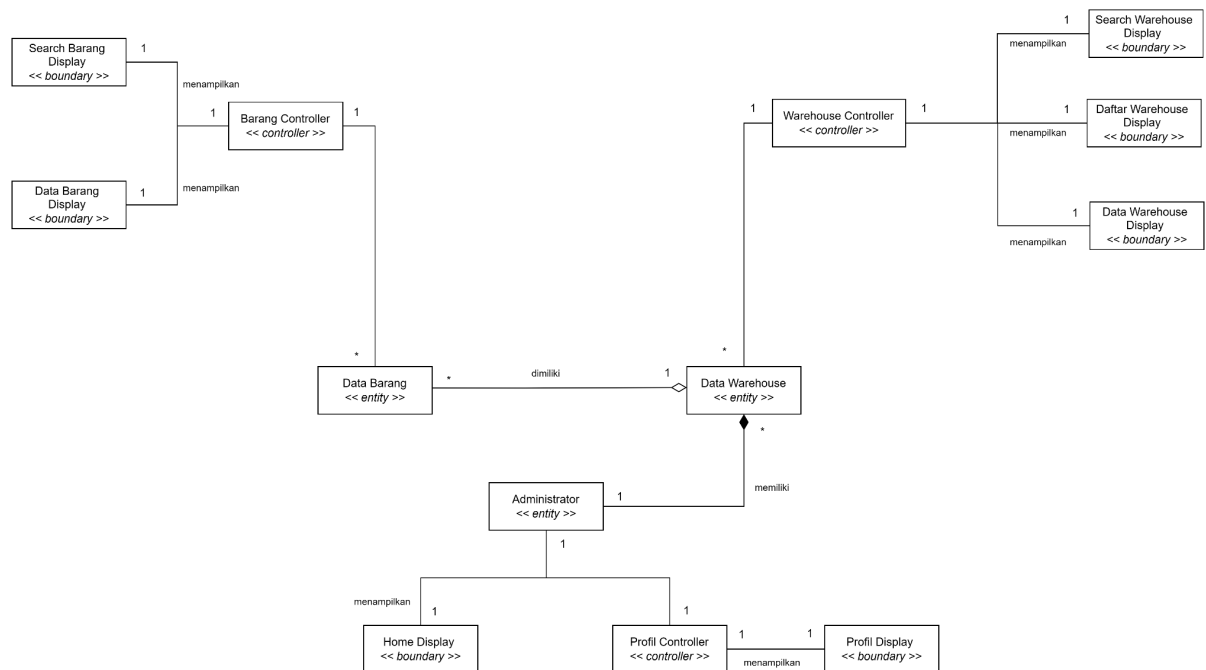
4 Model Analisis

4.1 Identifikasi Kelas

Tabel 4.1.1. Identifikasi Kelas

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	Administrator	Entity
2	Data Warehouse	Entity
3	Data Barang	Entity
4	Barang Controller	Controller
5	Warehouse Controller	Controller
6	Profile Controller	Controller
7	Home Display	Boundary
8	Daftar Warehouse Display	Boundary
9	Data Warehouse Display	Boundary
10	Data Barang Display	Boundary
11	Search Warehouse Display	Boundary
12	Search Barang Display	Boundary
13	Profil Display	Boundary

4.2 Diagram Kelas



Gambar 1.2.1. Diagram Kelas

Tabel 4.2.1. Keterangan Elemen Diagram Kelas

ID Kelas	Nama Kelas	Atribut	Metode
E1	Administrator	- Admin_ID : int - Nama_Lengkap : string	+ getAdmin_ID : int + getNama_Lengkap() : string

		- Surel : string - Password : string - No_Telp : string	+ getSurel(): string + getPassword() : string + getNo_Telp() : string + setAdmin_ID(int) + setNama_Lengkap(string) + setSurel(string) + setPassword(string) + setNo_Telp(string)
E2	Data Warehouse	- Warehouse_ID : int - Nama_Warehouse : string - Kapasitas_Maksimum : int - Kapasitas_Terkini : int - Alamat : string	+ getWarehouse_ID() : int + getNama_Warehouse() : string + getKapasitas_Maksimum() : int + getKapasitas_Terkini() : int + getAlamat() : string + setWarehouse_ID(int) + SetNama_Warehouse(string) + setKapasitas_Maksimum(int) + setKapasitas_Terkini(int) + setAlamat(string)
E3	Data Barang	- Barang_ID : int - Nama_Barang : string - Foto : string - Deskripsi : string - Stok : int - Status_Pengiriman : string	+ getBarang_ID() : int + getNama_Barang() : string + getFoto() : string + getDeskripsil() : string + getStok() : int + getStatus_Pengiriman() : string + setBarang_ID(int) + setNama_Barang(string) + setFoto(string) + setDeskripsi(string) + setStok(int) + setStatus_Pengiriman(string)
C1	Barang Controller		+ Tambah_Barang() + Update_Barang(int) + Search_Barang(string) : Barang + Hapus_Barang() + Lihat_Info_Barang(int) : Barang + Pindahkan_Barang(int, int, int)
C2	Warehouse Controller		+ Tambah_Warehouse() + Update_Warehouse(int) + Search_Warehouse(string) : Warehouse + Hapus_Warehouse() + Lihat_Daftar_Warehouse() : array of Warehouse + Lihat_Info_Warehouse(int) : Warehouse
C3	Profile Controller		+ Edit_Profil() + Lihat_Profil() : Profil
B1	Home Display		+ Tombol_Laman_Utama() + Tampilkan_Laman_Utama()
B2	Daftar Warehouse Display		+ Tombol_Daftar_Warehouse() + Tampilkan_Daftar_Warehouse()

B3	Data Warehouse Display		+ Tombol_Data_Warehouse() + Input_Nama_Warehouse() + Input_Kapasitas_Maksimum() + Input_Kapasitas_Terkini() + Input_Alamat() + Tampilkan_Data_Warehouse()
B4	Data Barang Display		+ Tombol_Data_Barang() + Input_Nama_Barang() + Input_Foto() + Input_Deskripsi() + Input_Stok() + Input_Status_Pengiriman() + Tampilkan_Data_Barang()
B5	Search Warehouse Display		+ Kolom_Pencarian_Warehouse() + Tombol_Pencarian_Warehouse()
B6	Search Barang Display		+ Kolom_Pencarian_Barang() + Tombol_Pencarian_Barang()
B7	Profil Display		+ Tombol_Profil() + Input_Nama_Lengkap() + Input_Surel() + Input_Password() + Input_No_Telp() + Tampilkan_Profil()

5 Keruntutan (traceability)

5.1 Kebutuhan Fungsional vs Use Case

Tabel 5.1.1. Hubungan Kebutuhan Fungsional dan Use Case

ID Kebutuhan Fungsional	ID Use Case Terkait
F01	U04
F02	U03
F03	U06
F04	U05
F05	U08
F06	U09, U10, U11
F07	U10
F08	U09
F09	U11
F10	U02

5.2 Use Case vs Kelas Terkait

Tabel 5.2.1 Hubungan Use Case dan Kelas Terkait

ID Use Case	Kelas Terkait
U01	E1
U02	E2, C2, B2, B3, B5
U03	E2, C2, B2, B3
U04	E2, C2, B3
U05	E2, C2, B3
U06	E2, C2, B3
U07	E3, C1, B4, B6
U08	E3, C1, B4
U09	E3, C1, B4
U10	E3, C1, B4
U11	E3, C1, B4