**DESAIN DOKUMEN UNTUK**

**SISTEM PENJUALAN ANAK AYAM (SPAY)**

****

**KELOMPOK 7 :**

**STEVEN 2125240006**

**STEVEN NEVILE 2125240043**

**RICHARD SATYA 2125240044**

**IRFAN ANDIKA 2125240080**

**FAKULTAS ILMU KOMPUTER & REKAYASA**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS MULTI DATA PALEMBANG**

**TAHUN AKADEMIK 2022 – 2023**

**DESAIN DOKUMEN UNTUK**

**SISTEM PENJUALAN ANAK AYAM (SPAY)**

1. **Ulasan**

Setelah meninjau analisis Use Case, berikut adalah kelas dasar dan tindakan yang harus dilakukan :

Kelas : (Blok Bangunan Dasara SPAY)

|  |  |  |
| --- | --- | --- |
| **No** | **Kelas** | **Prinsip Tanggung Jawab** |
| 1 | AdminController | Mengatur tampilan yang akan dituju oleh pengguna tergantung jabatannya |
| 2 | Register | Mendaftarkan akun pengguna |
| 3 | LoginController | Mengautentikasi data user yang masuk |
| 4 | LupaPassword Controller | Mengubah password user |
| 5 | StokAyamController | Mengelola data ayam yang disimpan |
| 6 | AyamMasukController | Mengelola data ayam yang masuk |
| 7 | AyamKeluarController | Mengelola data ayam yang keluar |
| 8 | StokPakanController | Mengelola data pakan yang disimpan |
| 9 | PakanMasukController | Mengelola data pakan yang masuk |
| 10 | PakanKeluarController | Mengelola data pakan yang keluar |
| 11 | PelangganController | Mengatur tampilan dan fitur fitur penting untuk akun pelanggan |

Catatan: Kelas tambahan lainnya dapat ditambahkan ke daftar selama implementasi untuk

tujuan penyeimbang dan modularitas.

Aksi :

|  |  |
| --- | --- |
| **No** | **Aksi** |
| 1 | Tambah/Ubah/Hapus AyamMasukController/AyamKeluarController/PakanAyamController,  PakanKeluarController |
| 2 | Tambah/Membaca/Ubah/Hapus StokAyamController/StokPakanController |
| 3 | Validasi pengguna |

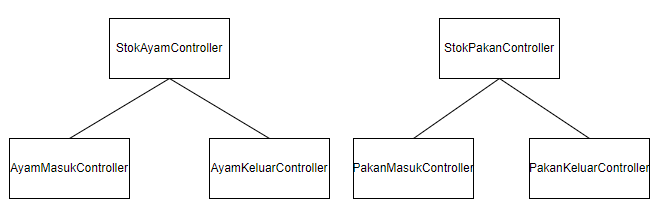
Catatan: Ada tindakan kecil lainnya yang tidak berperan besar dalam pemodelan.

1. Struktur Sistem

Di sini kami menjelaskan struktur akhir. Namun, perlu diingat bahwa mendapatkan struktur akhir adalah latihan berulang – struktur awal disempurnakan sebagai desain berlangsung. Secara khusus, pemodelan dinamis berdampak pada struktur.

* 1. Struktur Pewarisan

Kelihatannya tidak ada struktur pewarisan karena kurangnya kesamaan antar kelas. Di beberapa tempat pewarisan tampak intuitiff, misalnya dalam mengkhususkan StokAyamController ke AyamMasukController dan AyamKeluarController dan StokPakanController ke PakanMasukController dan PakanKeluarController. Gambar dibawah ini menunjukkan struktur pewarisan,

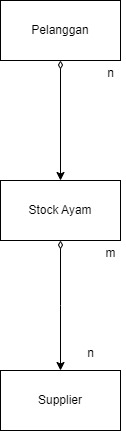


Gambar 2.1.1 Pewarisan yang memungkinkan

Namun struktur warisan ini tidak diperlukan. Kita dapat memodelkannya menggunakan atribut ekstra StokAyamType dan StokPakanType di kelas StokAyamController dan StokPakanController masing-masing.

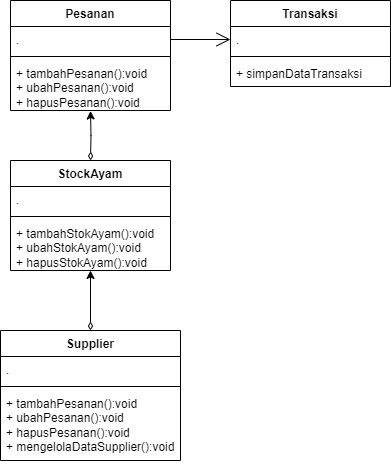
* 1. Agregasi

Struktur logis Investasi menyarankan agregasi berikut antara kelas.



Gambar 2.2.1 Struktur Agregasi

* 1. Asosiasi

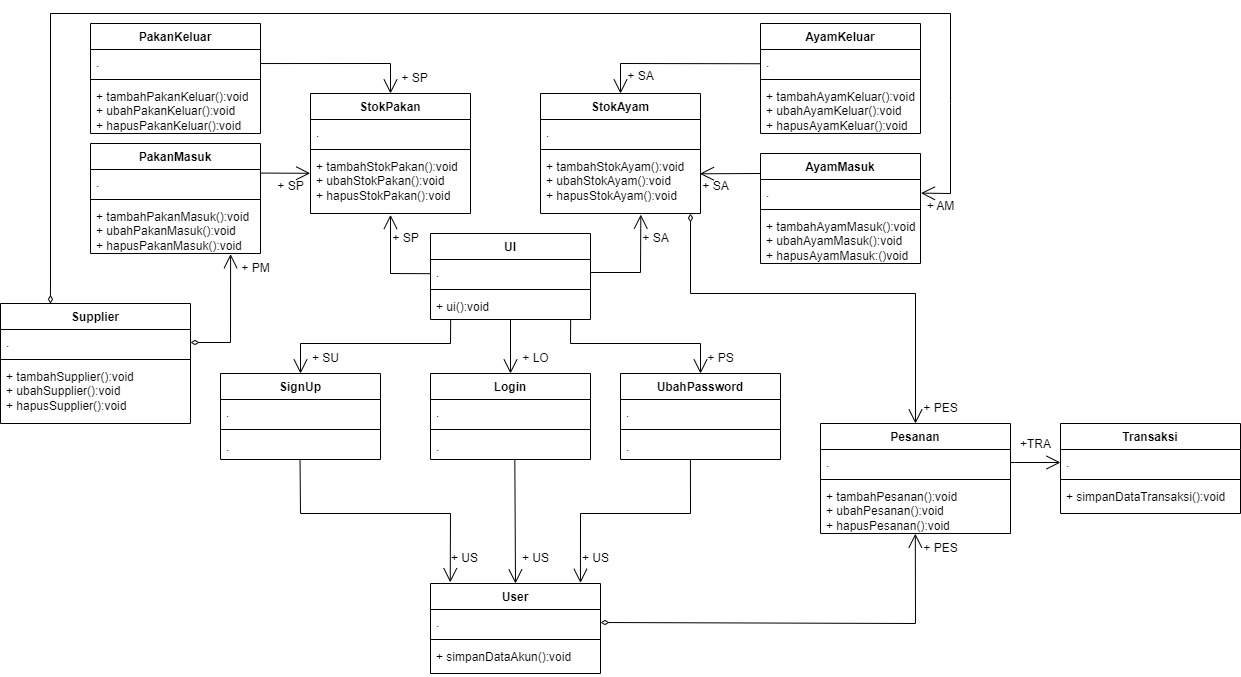
Kami mencari tahu hubungan antar kelas dalam proses pemodelan prinsip tindakan.

Contoh:

Gambar 2.3.1 Diagram Asosiasi

* 1. Melengkapi Kelas Diagram

Akhirnya setelah mempertimbangkan semua tindakan utama, asosiasi lengkap + agregasi struktur tercapai.

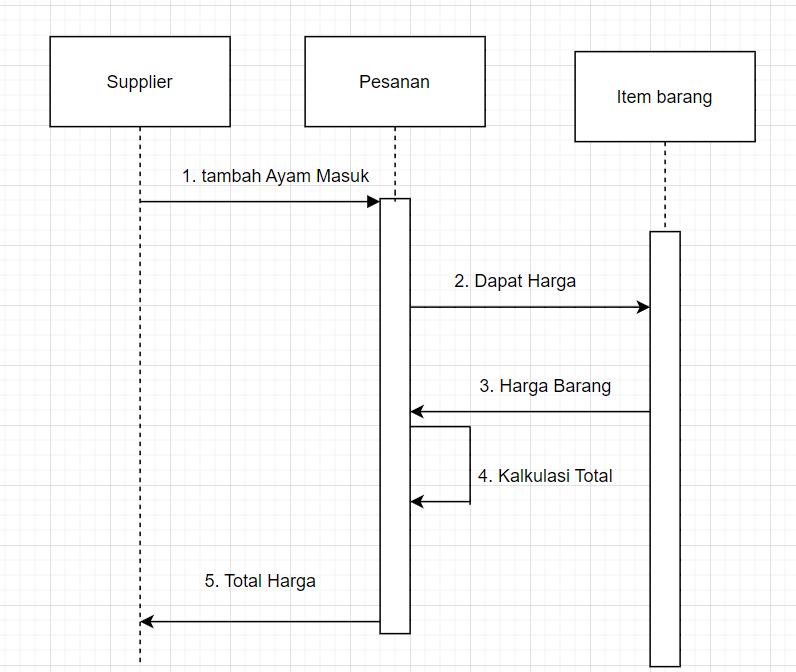


Gambar 2.4.1 Kelas Diagram menampilkan semua kelas dan asosiasi sistem

1. Perilaku Sistem

Perilaku dinamis dari sistem dimodelkan dengan mencari tahu interaksi antara kelas yang terlibat dalam setiap tindakan utama. Kami menunjukkan diagram akhir di sini. Dia harus diingat bahwa model-model ini berdampak pada penyempurnaan dan peningkatan diagram kelas - kita tidak membahas aspek-aspek ini di sini

* 1. Tindakan Prinsip: Tambah/ Ubah/ Hapus/AyamMasuk/AyamKeluar/PakanMasuk/PakanKeluar



Gambar 3.1.1: Sequence diagram untuk tindakan prinsip menambah ayam masuk & pakan masuk

* 1. Tindakan Prinsip: Tambah/Membaca/ Ubah/ Hapus/StokAyam / StokPakan

Gambar 3.2.1: Sequence diagram untuk aksi prinsip Tambah Ayam Keluar & Pakan Keluar

Sekarang kita berada dalam posisi untuk memulai dengan spesifikasi desain karena kita memiliki semuanya atribut dan metode dari semua kelas.

4. Detail Desain Spesifikasi:

Ini terdiri dari daftar kelas utama dan atribut serta metodenya dengan tepat komentar.

1. class AdminController{

//atribut//

//metode//

void index();

void pelanggan();

void owner();

}

1. class Register{

//atribut//

//metode//

Request create(request);

void index();

Request inputAdmin(request);

Request ubahAdmin(request);

Request hapusAdmin(request);

}

1. class LoginController{

//atribut//

//metode//

void index();

Request postLogin();

void postLogout();

}

1. class LupaPasswordController{

//atribut//

//metode//

Request submitForgetPasswordForm(request);

void showResetPasswordForm(token);

void submitResetPasswordForm(token);

}

1. class StokAyamController{

//atribut//

//metode//

void index();

Request inputStockAyam(request);

Request editStockAyam(request);

Request hapusStockAyam(request);

void exportStock();

void show(id);

}

1. class AyamMasukController{

//atribut//

//metode//

void index();

Request inputAyamMasuk(request);

Request editAyamMasuk(request);

Request hapusAyamMasuk(request);

}

1. classAyamKeluarController{

//atribut//

//metode//

void index();

Request inputAyamKeluar(request);

Request inputAyamKeluar(request);

Request inputAyamKeluar(request);

}

1. classStokPakanController{

//atribut//

//metode//

void index();

Request inputStockPakan(request);

Request editStockPakan(request);

Request hapusStockPakan(request);

}

1. class PakanMasukController{

//atribut//

//metode//

void index();

Request inputPakanMasuk(request);

Request editPakanMasuk(request);

Request hapusPakanMasuk(request);

}

1. class PakanKeluarController{

//atribut//

//metode//

void index();

Request inputPakanKeluar(request);

Request editPakanKeluar(request);

Request hapusPakanKeluark(request);

}

1. class PelangganController{

//atribut//

//metode//

void index();

void produk();

}

Catatan: *Kelas Investasi memiliki daftar nama portofolio sebagai atribut dan bukan daftar*

*objek portofolio. Ini dilakukan untuk mengurangi tekanan pada RAM, mempertahankan semua objek portofolio, surat berharga dan transaksi hidup artinya kita memiliki seluruh database di RAM ini mungkin sangat mempengaruhi efisiensi. Objek portofolio dapat dibuat saat dijalankan sebagaimana dan kapan pun diperlukan. Hal serupa juga dilakukan untuk portofolio.*