

RECIPE SHARE SYSTEM

DISUSUN OLEH KELompok 1:
CUT DAHLIANA 2208107010027
ZAHRA ZAFIRA 2208107010040



RECIPE SHARE SYSTEM

Sistem berbasis konsol yang dirancang untuk mempermudah pengguna dalam berbagi, mencari, dan mengelola resep masakan. Pengguna dapat mendaftar, masuk, menambahkan resep sendiri, mencari resep berdasarkan bahan, diet, atau waktu persiapan, serta melihat detail resep yang dibagikan oleh pengguna lain. Sistem ini juga menyertakan data resep awal dan mendukung eksplorasi resep secara menyeluruh.

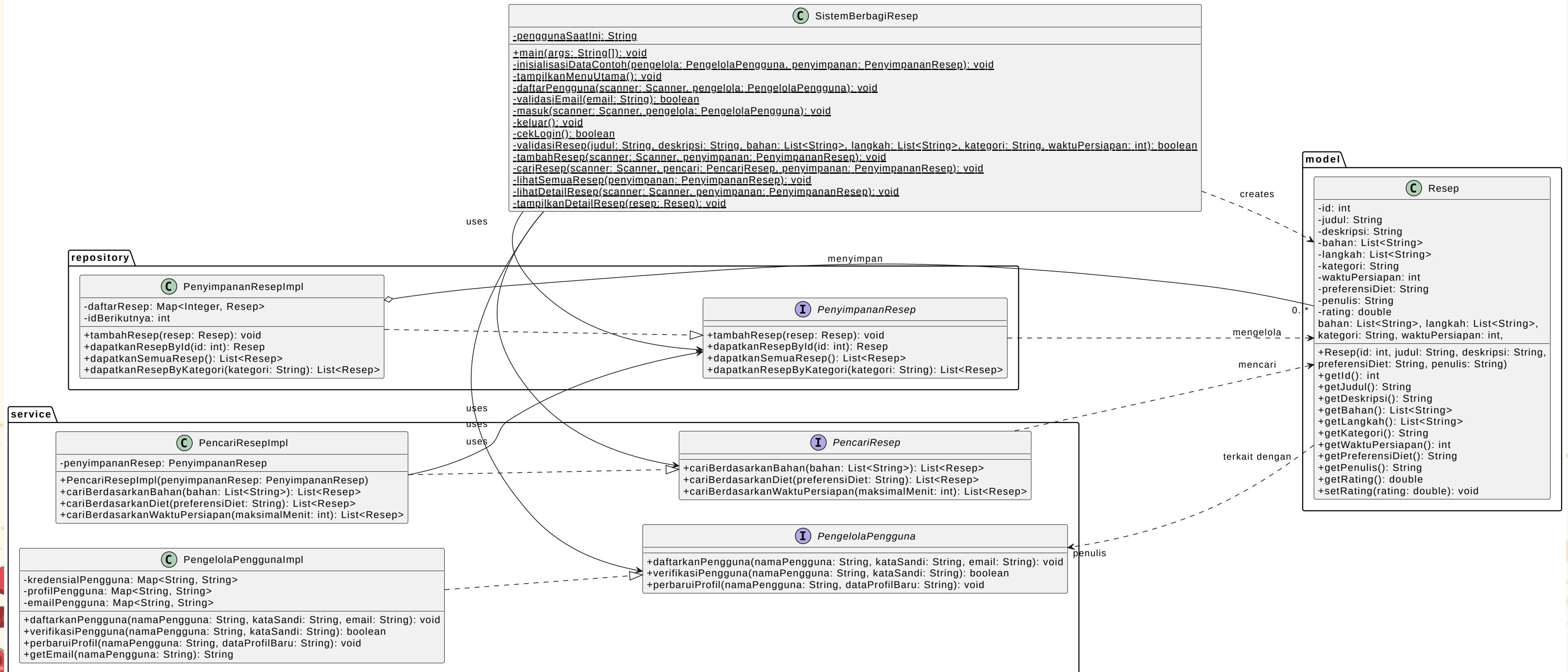
PRE AND POST-CONDITION

Use Case Name	Menambah Resep (AddRecipe)
Aktor	User (Pengguna)
Deskripsi	Pengguna dapat menambahkan resep baru ke dalam sistem.
Precondition	Pengguna telah terdaftar dan login ke dalam sistem.
Postcondition	Resep baru berhasil disimpan dalam database dan tersedia untuk dibagikan.
Flow Utama	<ol style="list-style-type: none"> 1. Pengguna memilih opsi "Tambah Resep". 2. Sistem menampilkan form untuk mengisi detail resep. 3. Pengguna mengisi nama, bahan, langkah-langkah, dan kategori resep. 4. Pengguna mengkonfirmasi penambahan resep. 5. Sistem menyimpan resep baru dan mengkonfirmasi keberhasilan.

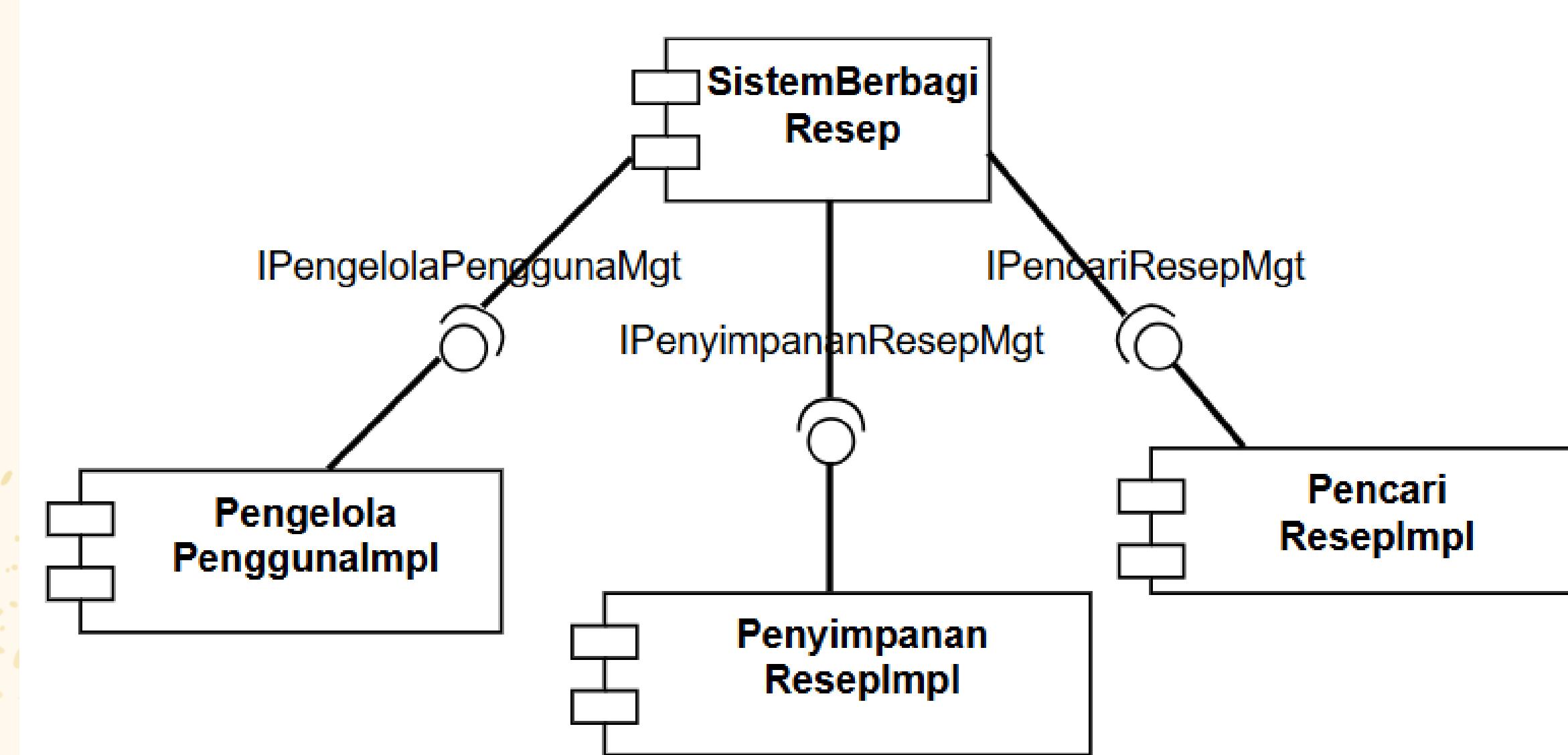
Use Case Name	Menyimpan Resep (SaveRecipe)
Aktor	User (Pengguna)
Deskripsi	Pengguna dapat menyimpan resep yang disukai untuk akses mudah di kemudian hari.
Precondition	Pengguna telah menemukan resep yang ingin disimpan.
Postcondition	Resep berhasil disimpan dalam koleksi pribadi pengguna.
Flow Utama	<ol style="list-style-type: none"> 1. Pengguna menemukan resep yang ingin disimpan. 2. Pengguna memilih opsi "Simpan Resep". 3. Sistem menambahkan resep ke koleksi pribadi pengguna. 4. Sistem mengkonfirmasi bahwa resep telah berhasil disimpan. 5. Resep tersimpan dapat diakses melalui menu "Resep Tersimpan" pengguna.

Use Case Name	Mencari Resep (SearchRecipe)
Aktor	User (Pengguna)
Deskripsi	Pengguna dapat mencari resep berdasarkan kata kunci atau kategori.
Precondition	Terdapat resep yang tersimpan dalam database.
Postcondition	Sistem menampilkan daftar resep yang sesuai dengan kriteria pencarian.
Flow Utama	<ol style="list-style-type: none"> 1. Pengguna mengakses fitur pencarian resep. 2. Sistem menampilkan form pencarian. 3. Pengguna memasukkan kata kunci atau memilih kategori. 4. Sistem memproses pencarian dan menampilkan hasil yang relevan. 5. Pengguna dapat melihat detail resep yang dicari.

UML CLASS DIAGRAMS



UML COMPONENT DIAGRAMS



ARSITEKTUR SISTEM

"Sistem ini menggunakan arsitektur modular dengan pembagian sebagai berikut:"

Representasi objek bisnis

inti

MODEL:

Resep.java - Class utama yang merepresentasikan struktur data resep

Penyimpanan data dan operasi CRUD

REPOSITORY:

- IPenyimpananResepMgt.java (interface)
- PenyimpananResepImpl.java (implementasi) - Menangani penyimpanan data resep

Menangani logika bisnis kompleks

SERVICE:

- IPengelolaPenggunaMgt.java (interface)
- PengelolaPenggunaImpl.java (implementasi)
- IPencariResepMgt.java (interface)
- PencariResepImpl.java (implementasi)

Antarmuka pengguna dan koordinasi alur

MAIN CLASS:

SistemBerbagiResep.java - Menghubungkan semua komponen dan menampilkan menu

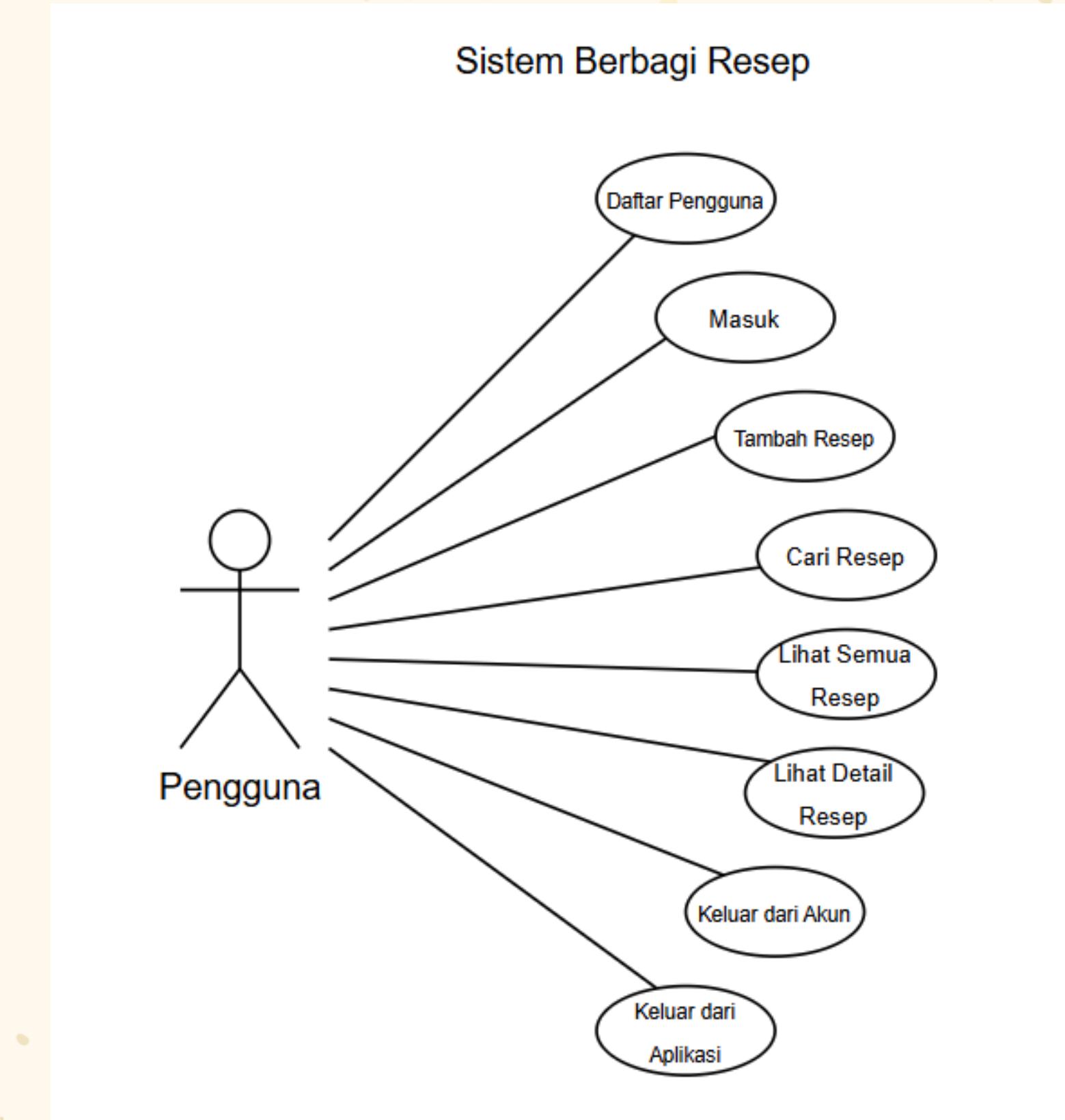


SPECIFICATION WORKFLOW

1. COMPONENT IDENTIFICATION
2. COMPONENT INTERACTION
3. COMPONENT SPECIFICATION

I. COMPONENT IDENTIFICATION

USE CASE DIAGRAM



1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

1. Daftar Pengguna

Name: Daftar Pengguna

- Initiator: Pengguna
- Goal: Mendaftarkan akun pengguna baru ke sistem.

Main Success Scenario:

1. Pengguna memilih opsi "Daftar Pengguna".
2. Sistem meminta data: nama, email, dan kata sandi.
3. Pengguna memasukkan data yang diminta.
4. Sistem memvalidasi data.
5. Sistem menyimpan data pengguna baru.
6. Sistem menampilkan pesan "Pengguna berhasil didaftarkan!" dan kembali ke menu utama.

Extensions:

3. Pengguna membatalkan pendaftaran.
→ Kembali ke menu utama.
4. Data yang dimasukkan tidak valid (contoh: email salah format).
→ Sistem menampilkan pesan "Kredensial tidak valid atau pengguna belum terdaftar!. Periksa kembali." dan kembali ke langkah 2.
5. Sistem gagal menyimpan data.
→ Sistem menampilkan pesan "Nama pengguna sudah digunakan! Silahkan pilih nama lain." dan kembali ke menu utama.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

2. Masuk

2. Name: Masuk

- Initiator: Pengguna
- Goal: Masuk ke akun pengguna.

Main Success Scenario:

1. Pengguna memilih opsi "Masuk".
2. Sistem meminta nama dan kata sandi.
3. Pengguna memasukkan nama dan kata sandi.
4. Sistem memverifikasi kredensial.
5. Sistem mengizinkan pengguna masuk
6. Sistem menampilkan pesan "Berhasil masuk! Selamat datang, Pengguna!" dan menampilkan menu utama.

Extensions:

3. Pengguna membatalkan proses masuk.
→ Kembali ke menu utama.
4. Nama atau kata sandi salah
→ Sistem menampilkan pesan "Kredensial tidak valid atau pengguna belum terdaftar!" dan kembali ke langkah 2.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

3. Tambah Resep

Name: Tambah Resep

- Initiator: Pengguna
- Goal: Menambahkan resep baru ke sistem.

Main Success Scenario:

1. Pengguna memilih opsi "Tambah Resep".
2. Sistem meminta input judul resep, deskripsi, bahan-bahan, langkah pembuatan, kategori, waktu persiapan, dan memilih preferensi diet.
3. Pengguna mengisi data resep.
4. Sistem memvalidasi input.
5. Sistem menyimpan resep ke dalam database.
6. Sistem menampilkan konfirmasi "Resep berhasil ditambahkan!"

Extensions:

3. Pengguna membatalkan proses tambah resep.
 - Kembali ke menu utama.
4. Data yang diisi tidak lengkap.
 - Sistem menampilkan pesan "Input tidak valid." dan kembali ke langkah 2.
5. Sistem gagal menyimpan resep.
 - Sistem menampilkan pesan "Resep tidak dapat ditambahkan karena data tidak valid. Silahkan coba lagi." dan kembali ke menu utama.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

4. Cari Resep

4. Name: Cari Resep

- Initiator: Pengguna
- Goal: Mencari resep berdasarkan bahan, preferensi diet dan waktu persiapan.

Main Success Scenario:

1. Pengguna memilih opsi "Cari Resep".
2. Sistem meminta input kata kunci pencarian.
3. Pengguna memasukkan kata kunci.
4. Sistem menampilkan daftar resep yang sesuai.

Extensions:

3. Pengguna membatalkan pencarian.
→ Kembali ke menu utama.
4. Tidak ada resep ditemukan.
→ Sistem menampilkan pesan "Tidak ada resep yang sesuai dengan kriteria pencarian." dan kembali ke menu utama.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

5. Lihat semua Resep

5. Name: Lihat Semua Resep

- Initiator: Pengguna
- Goal: Melihat daftar semua resep yang tersedia.

Main Success Scenario:

1. Pengguna memilih opsi "Lihat Semua Resep".
2. Sistem menampilkan daftar semua resep.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

6. Lihat Detail Resep

6. Name: Lihat Detail Resep

- Initiator: Pengguna
- Goal: Melihat detail informasi dari suatu resep.

Main Success Scenario:

1. Pengguna memilih salah satu resep dari daftar.
2. Sistem menampilkan detail resep: id, judul, penulis, deskripsi, kategori, waktu persiapan, preferensi diet, rating, bahan, dan langkah-langkah pembuatan.
3. Sistem bertanya kepada pengguna: "Apakah Anda ingin memberi rating untuk resep ini?"
4. Jika pengguna memilih "Ya", sistem meminta input nilai rating (misal: 1-5 bintang).
5. Pengguna memberikan rating.
6. Sistem menyimpan rating baru.
7. Sistem menampilkan pesan "Rating berhasil diberikan!"

Extensions:

1. Pengguna batal memilih resep.
→ Kembali ke menu utama.
2. Sistem gagal menampilkan detail.
→ Sistem menampilkan pesan "Gagal memuat detail resep." dan kembali ke menu utama.
4. Pengguna membatalkan pemberian rating.
→ Sistem kembali ke menu utama tanpa perubahan.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

7. Keluar Dari Akun

7. Name: Keluar dari Akun

- Initiator: Pengguna
- Goal: Keluar dari akun pengguna yang sedang aktif.

Main Success Scenario:

1. Pengguna memilih opsi "Keluar dari Akun".
2. Sistem menampilkan pesan "Berhasil keluar dari akun pengguna."
3. Sistem menghapus sesi login pengguna dan kembali ke halaman masuk.

1. COMPONENT IDENTIFICATION

USE CASE DESCRIPTION

8. Keluar Dari Aplikasi

8. Name: Keluar dari Aplikasi

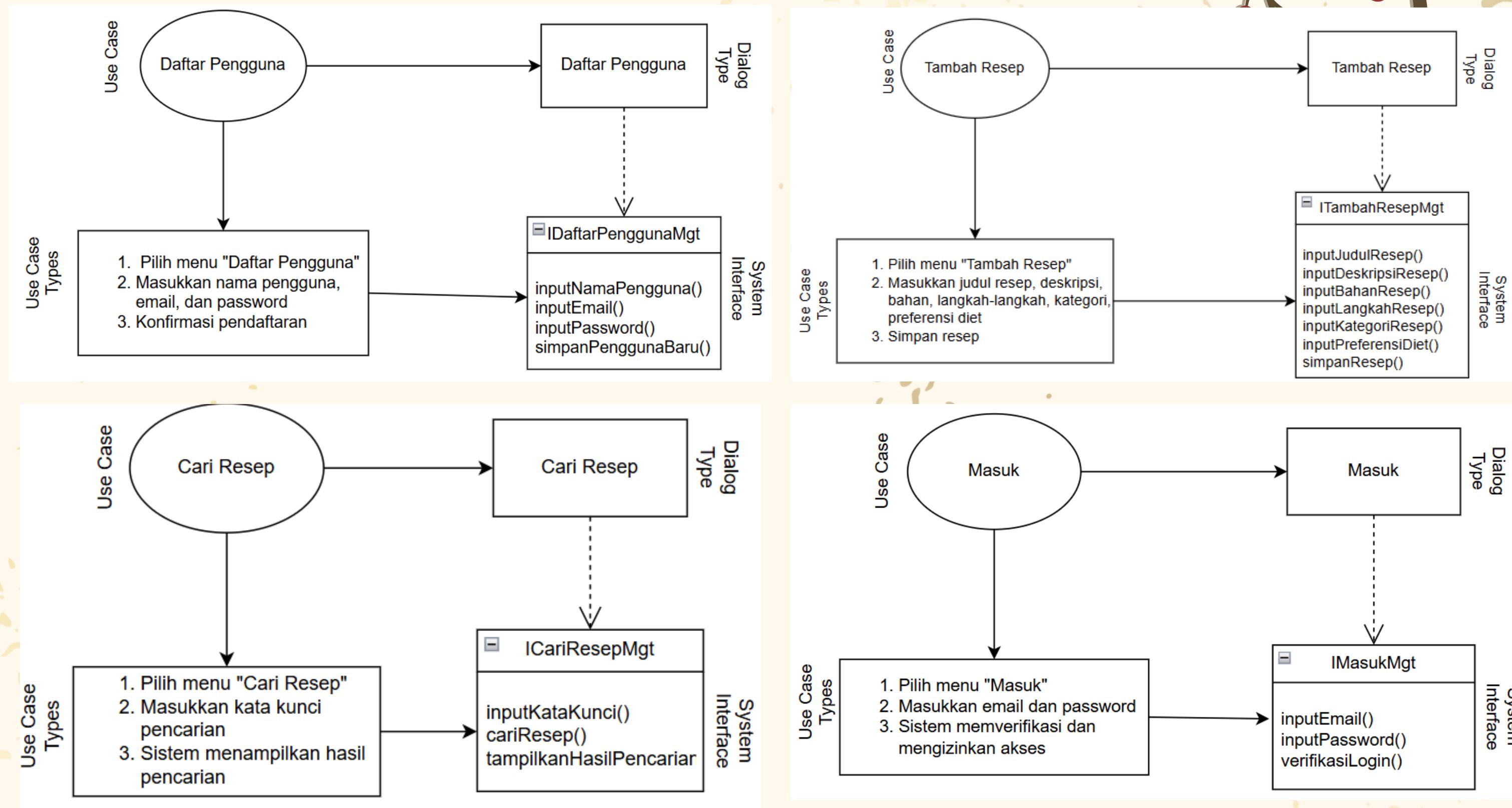
- Initiator: Pengguna
- Goal: Menutup aplikasi berbagi resep.

Main Success Scenario:

1. Pengguna memilih opsi "Keluar dari Aplikasi".
2. Sistem menampilkan pesan "Keluar dari Sistem Berbagi Resep. Sampai Jumpa!"
3. Sistem menutup aplikasi.

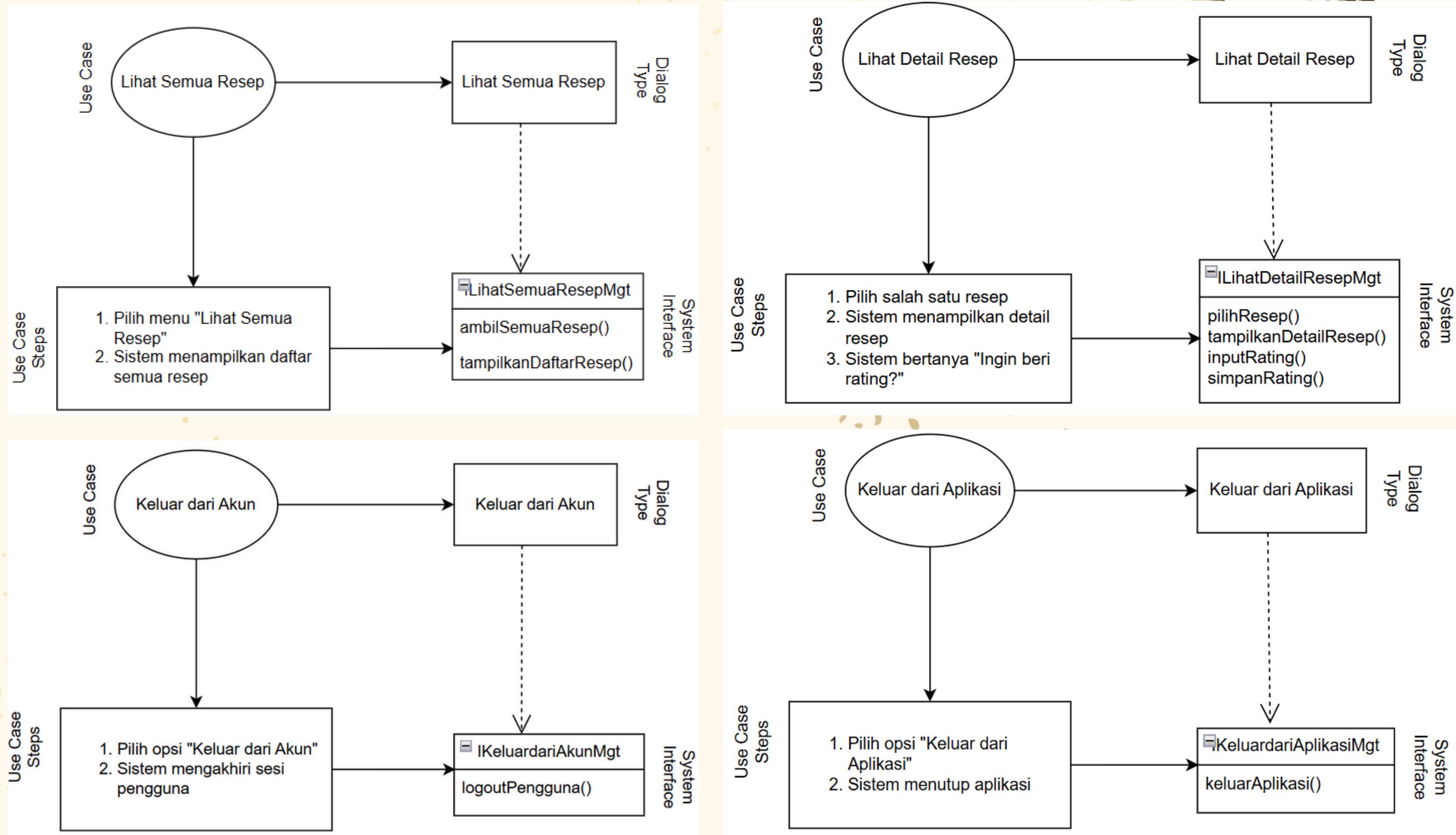
1. COMPONENT IDENTIFICATION

SYSTEM INTERFACE



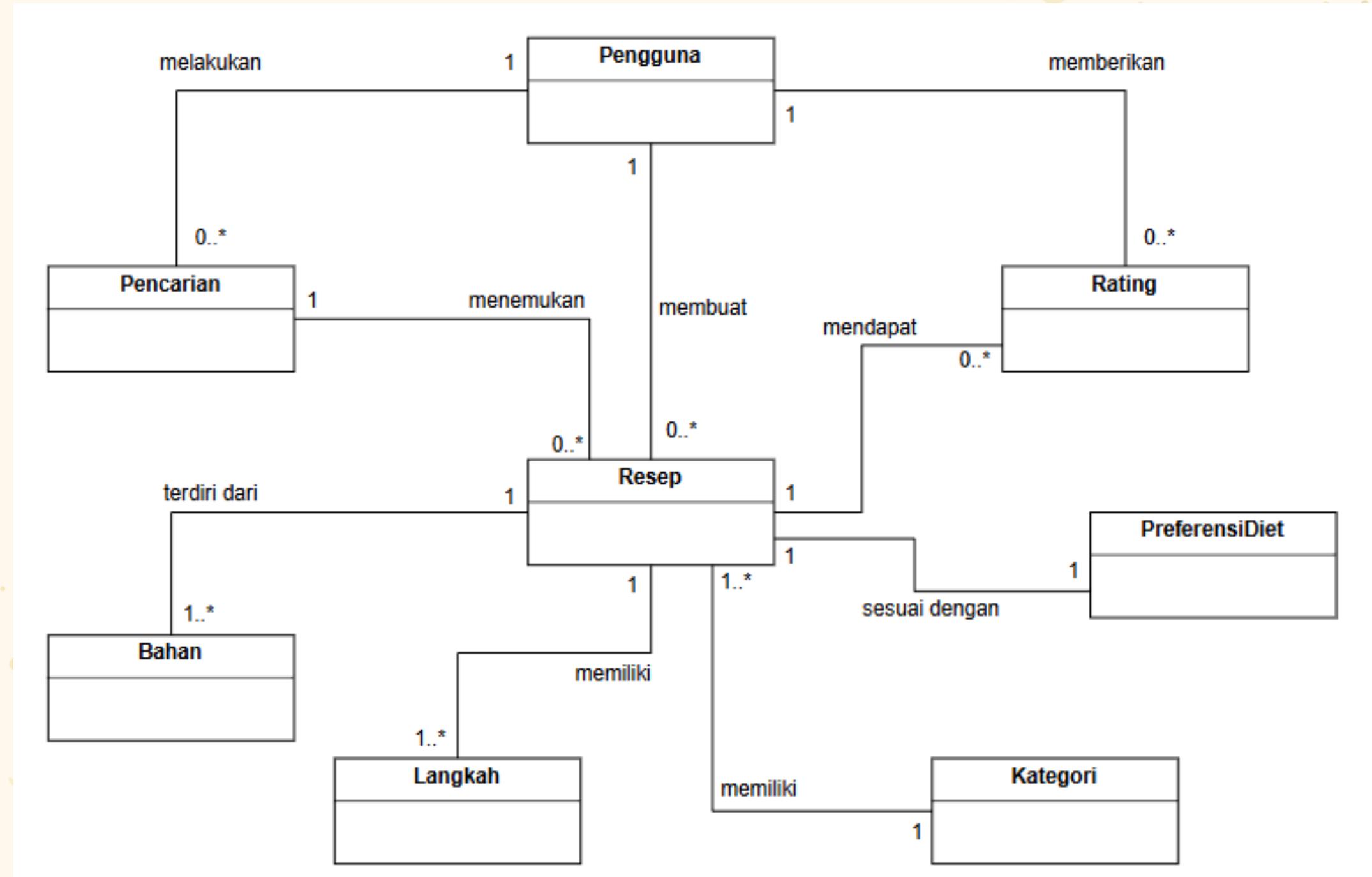
1. COMPONENT IDENTIFICATION

SYSTEM INTERFACE



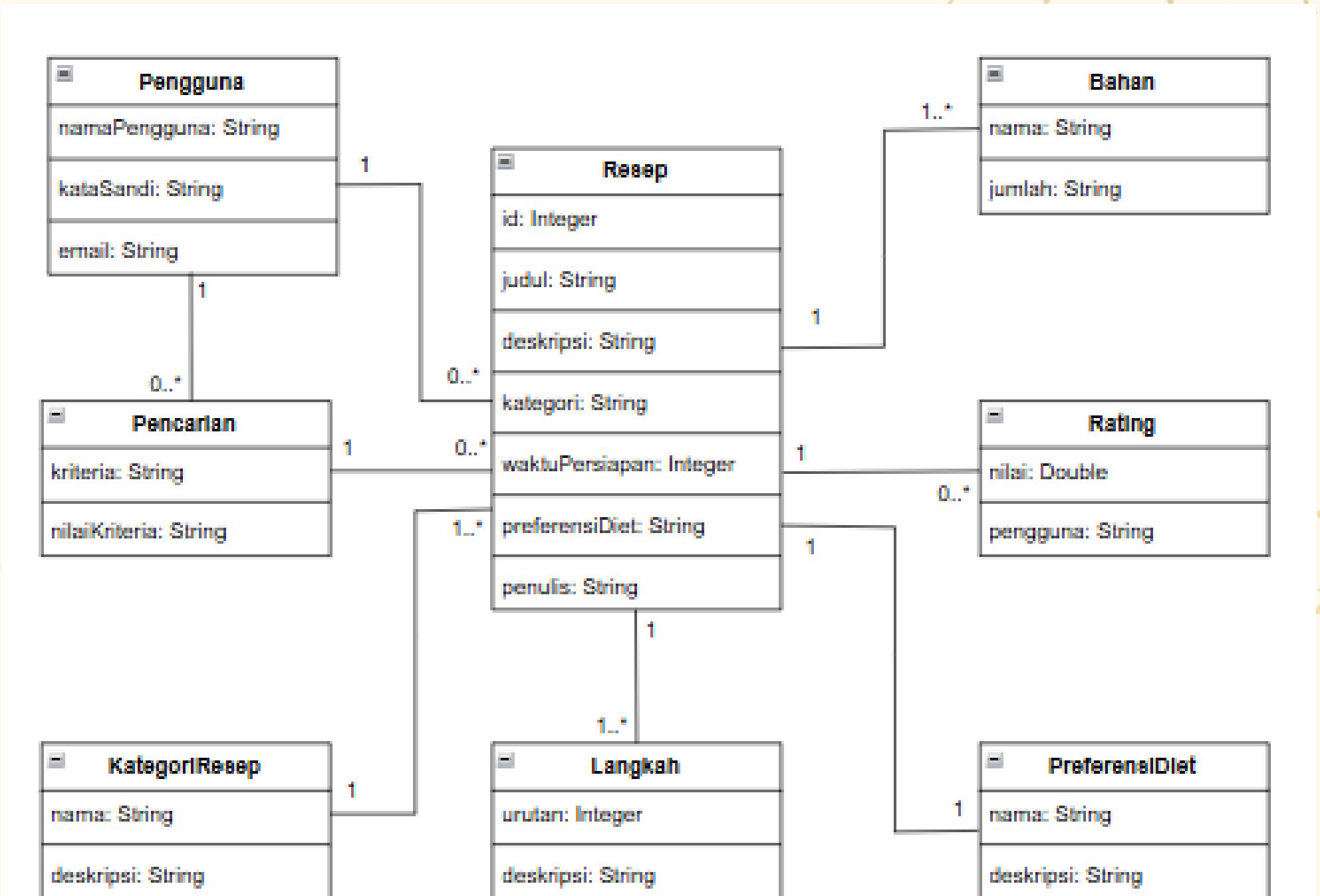
1. COMPONENT IDENTIFICATION

BUSINESS CONCEPT MODEL



1. COMPONENT IDENTIFICATION

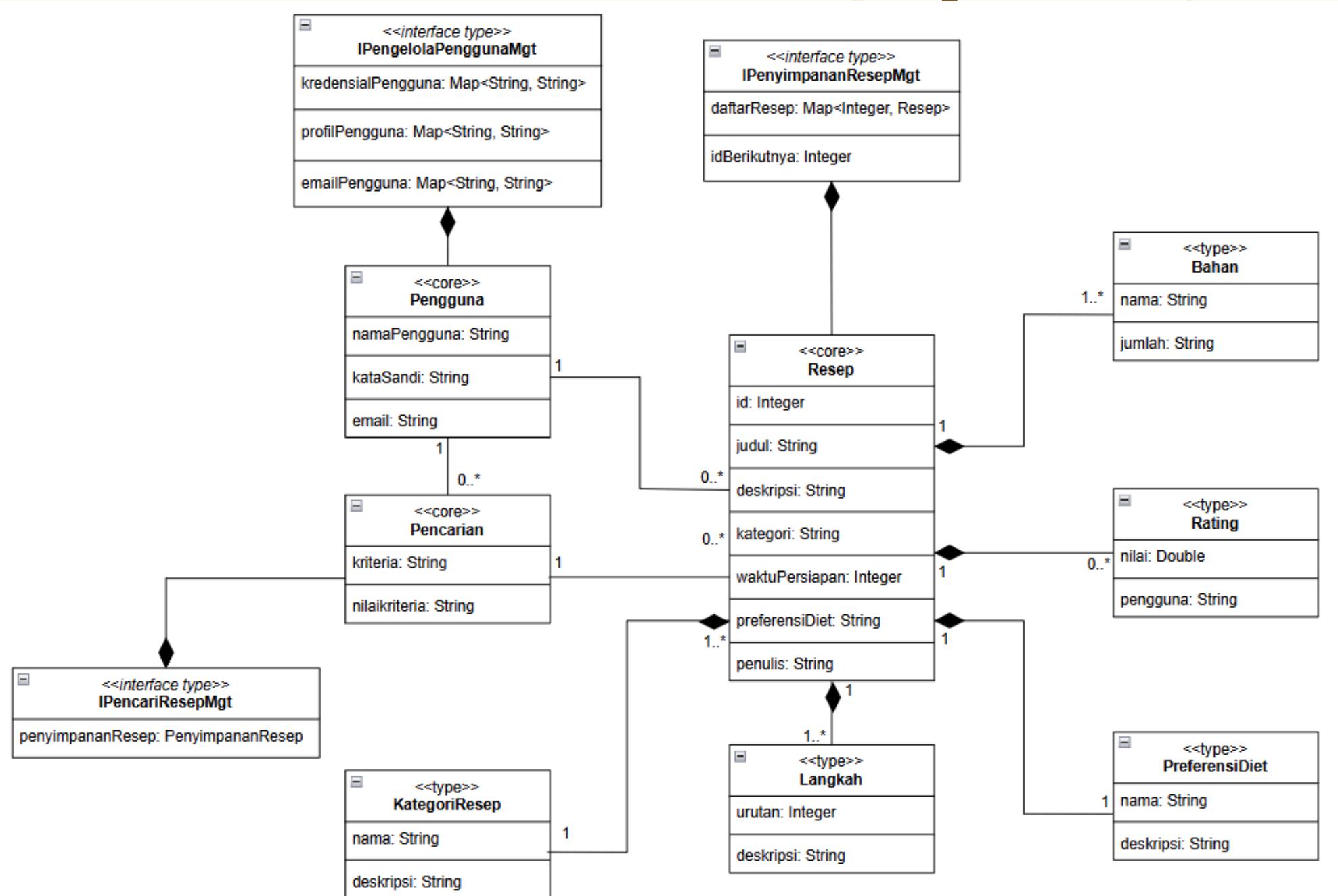
BUSINESS
TYPE
MODEL



1. COMPONENT IDENTIFICATION

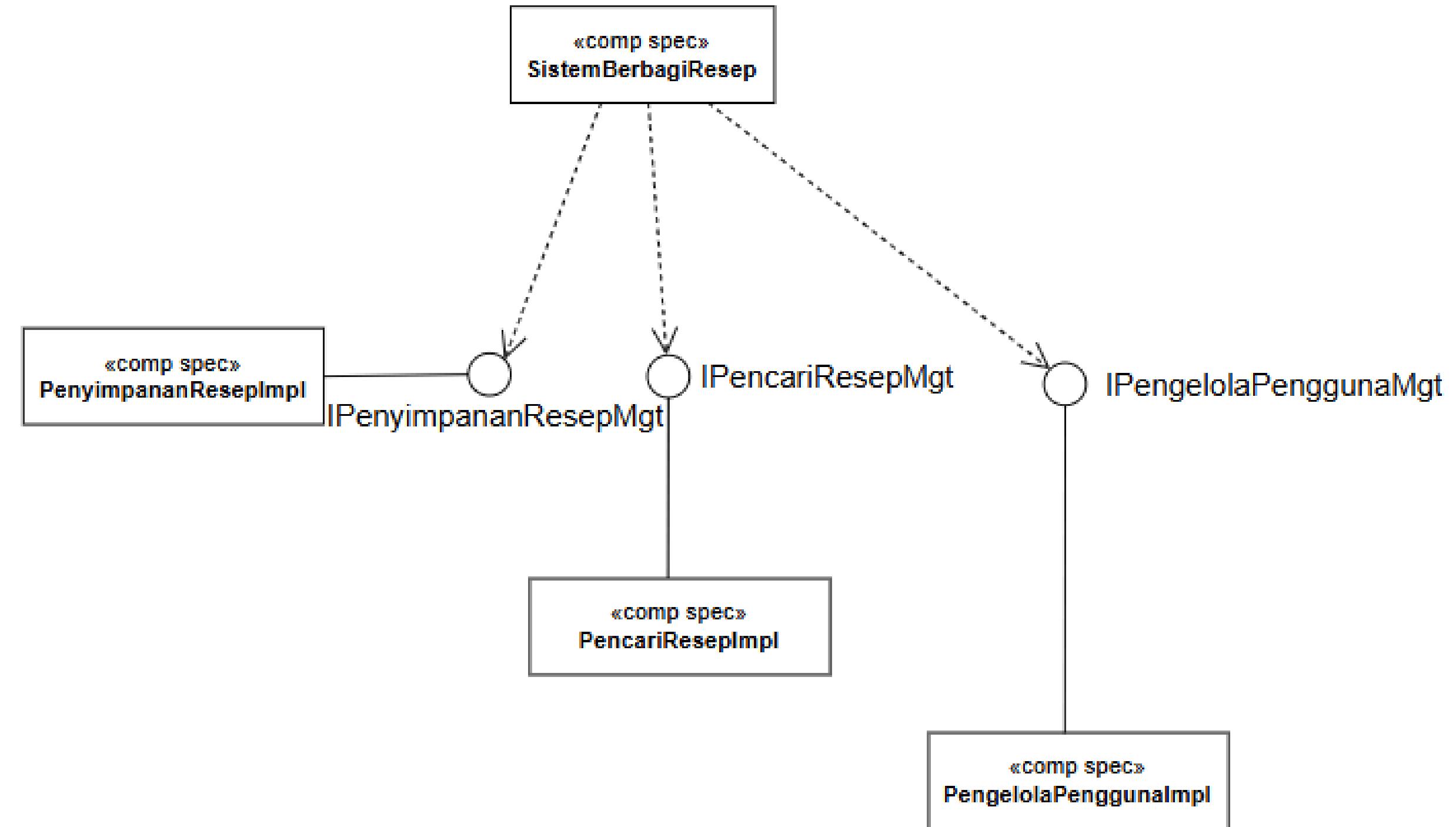
BISNIS INTERFACE

interface responsibility diagram



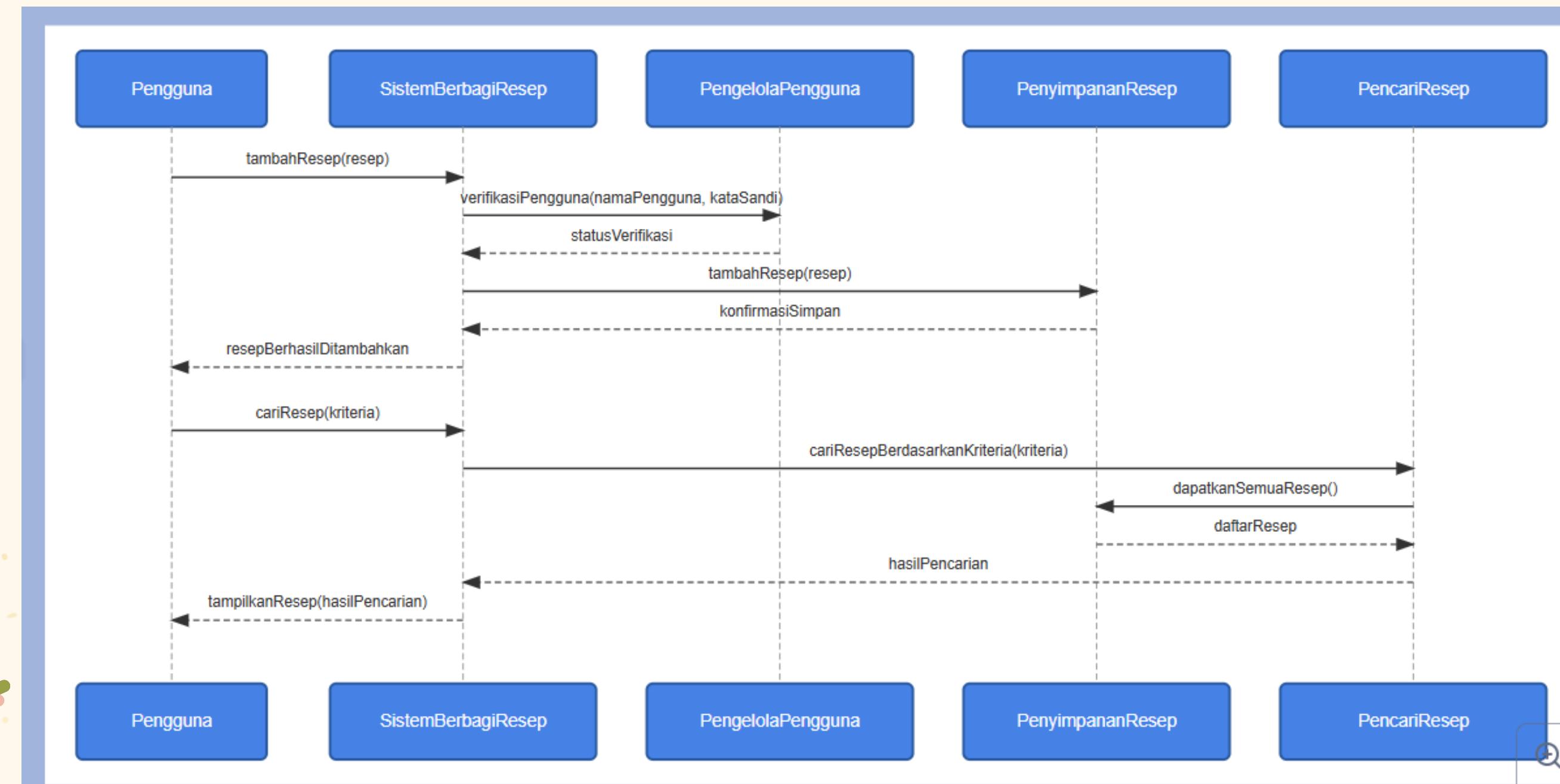
1. COMPONENT IDENTIFICATION

INITIAL COMPONENT DIAGRAM



2. COMPONENT INTERACTION

DIAGRAM
SEQUENCE
DIAGRAM



3. COMPONENT SPECIFICATION

OCL

```
1 context IPenyimpananResepMgt::tambahResep(resep: Resep): void
2 pre:
3   -- resep tidak boleh null
4   resep <> null and
5   -- judul resep tidak boleh kosong
6   resep.judul <> null and resep.judul.size() > 0 and
7   -- deskripsi resep tidak boleh kosong
8   resep.deskripsi <> null and resep.deskripsi.size() > 0 and
9   -- resep harus memiliki minimal satu bahan
10  resep.bahan <> null and resep.bahan->size() > 0 and
11  -- resep harus memiliki minimal satu langkah
12  resep.langkah <> null and resep.langkah->size() > 0 and
13  -- waktu persiapan harus positif
14  resep.waktuPersiapan > 0 and
15  -- penulis tidak boleh kosong
16  resep.penulis <> null and resep.penulis.size() > 0
17 post:
18  -- resep berhasil ditambahkan
19  self.dapatkanResepById(resep.id) = resep
```

3. COMPONENT SPECIFICATION

OCL

```
1 context IPencariResepMgt::cariBerdasarkanBahan(bahan: List<String>): List<Resep>
2 pre:
3   -- list bahan tidak boleh null
4   bahan <> null and
5   -- minimal satu bahan untuk pencarian
6   bahan->size() > 0
7 post:
8   -- hasil pencarian berisi resep yang memiliki minimal satu bahan yang dicari
9   result->forAll(r | r.bahan->exists(b | bahan->exists(cari | b.toLowerCase().contains(cari.toLowerCase()))))
```

```
1 context IPencariResepMgt::cariBerdasarkanDiet(preferensiDiet: String): List<Resep>
2 pre:
3   -- preferensi diet tidak boleh null atau kosong
4   preferensiDiet <> null and preferensiDiet.size() > 0
5 post:
6   -- hasil pencarian berisi resep dengan preferensi diet yang sama
7   result->forAll(r | r.preferensiDiet.equalsIgnoreCase(preferensiDiet))
```

3. COMPONENT SPECIFICATION

```
● ● ●  
1 context IPencariResepMgt::cariBerdasarkanWaktuPersiapan(maksimalMenit: Integer): List<Resep>  
2 pre:  
3   -- waktu maksimal harus positif  
4   maksimalMenit > 0  
5 post:  
6   -- hasil pencarian berisi resep dengan waktu persiapan tidak melebihi batas maksimal  
7   result->forAll(r | r.waktuPersiapan <= maksimalMenit)  
8
```

OCL

```
● ● ●  
1 context IPengelolaPenggunaMgt::daftarkanPengguna(namaPengguna: String, kataSandi: String, email: String): void  
2 pre:  
3   -- nama pengguna tidak boleh kosong  
4   namaPengguna <> null and namaPengguna.size() > 0 and  
5   -- kata sandi tidak boleh kosong  
6   kataSandi <> null and kataSandi.size() > 0 and  
7   -- email harus valid (mengandung @ dan diakhiri dengan gmail.com atau .id)  
8   email <> null and email.size() > 0 and  
9   email.contains('@') and (email.endsWith('gmail.com') or email.endsWith('.id')) and  
10  -- nama pengguna belum terdaftar  
11  not self.kredensialPengguna->containsKey(namaPengguna)  
12 post:  
13  -- pengguna berhasil didaftarkan  
14  self.kredensialPengguna->containsKey(namaPengguna) and  
15  self.kredensialPengguna->get(namaPengguna) = kataSandi and  
16  self.emailPengguna->containsKey(namaPengguna) and  
17  self.emailPengguna->get(namaPengguna) = email  
18
```

3. COMPONENT SPECIFICATION

OCL

```
1 context IPengelolaPenggunaMgt::verifikasiPengguna(namaPengguna: String, kataSandi: String): Boolean
2 pre:
3   -- nama pengguna dan kata sandi tidak boleh kosong
4   namaPengguna <> null and namaPengguna.size() > 0 and
5   kataSandi <> null and kataSandi.size() > 0
6 post:
7   -- hasil true jika kredensial cocok
8   result = (self.kredensialPengguna->containsKey(namaPengguna) and
9             self.kredensialPengguna->get(namaPengguna) = kataSandi)
10
```

```
1 context IPengelolaPenggunaMgt::perbaruiProfil(namaPengguna: String, dataProfilBaru: String): void
2 pre:
3   -- nama pengguna dan data profil tidak boleh kosong
4   namaPengguna <> null and namaPengguna.size() > 0 and
5   dataProfilBaru <> null and dataProfilBaru.size() > 0 and
6   -- pengguna harus terdaftar
7   self.profilPengguna->containsKey(namaPengguna)
8 post:
9   -- profil berhasil diperbarui
10  self.profilPengguna->get(namaPengguna) = dataProfilBaru
11
```

3. COMPONENT SPECIFICATION

```
● ● ●  
1 context SistemBerbagiResep::tambahResep(): void  
2 pre:  
3   -- pengguna harus sudah login  
4   penggunaSaatIni <> null  
5 post:  
6   -- jumlah resep bertambah jika data valid  
7   penyimpananResep.dapatkanSemuaResep()>size() = penyimpananResep.dapatkanSemuaResep()>size()@pre + 1  
8
```

OCL

```
● ● ●  
1 context Resep::setRating(rating: Double): void  
2 pre:  
3   -- rating harus antara 1 sampai 5  
4   rating >= 1.0 and rating <= 5.0  
5 post:  
6   -- rating berhasil diperbarui  
7   self.rating = rating
```

**THANK YOU
FOR YOUR
ATTENTION**

