



Perangkat Lunak
Berbasis Komponen



SELESAI.IN

SISTEM MANAJEMEN TUGAS

KELOMPOK-22

01.



OUR TEAM



**Fathiya Namira
Fardhi**

2208107010071



Iwani Khairina

2208107010078



PENDAHULUAN

Latar Belakang

Aplikasi Manajemen Tugas Mahasiswa dirancang untuk membantu mahasiswa dalam mencatat, memantau, dan menyimpan tugas-tugas secara terorganisir.

Fitur-fitur utama meliputi:

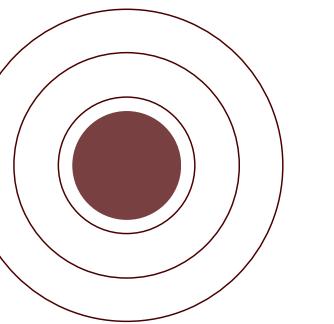
- Menambahkan tugas baru
- Menyimpan data ke file .txt
- Menampilkan seluruh daftar tugas
- Menghapus List Tugas
- Menampilkan Tugas dari Dosen



TUJUAN SISTEM

- 01.** Memberikan sarana bagi mahasiswa untuk mencatat, mengatur, dan memantau tugas-tugas perkuliahan secara sistematis

- 02.** Membantu pengguna dalam mengelola tugas-tugas mereka secara efisien dengan menyediakan platform untuk menambah, mengedit, menampilkan, dan menyimpan tugas.



kelompok 22



WORKFLOW

01.

Component
Identification

02.

Component
Interaction

03.

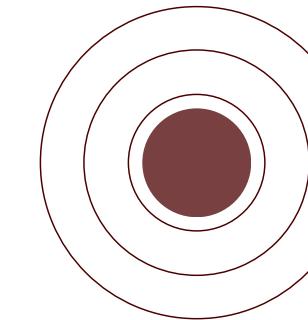
Component
Specification

05





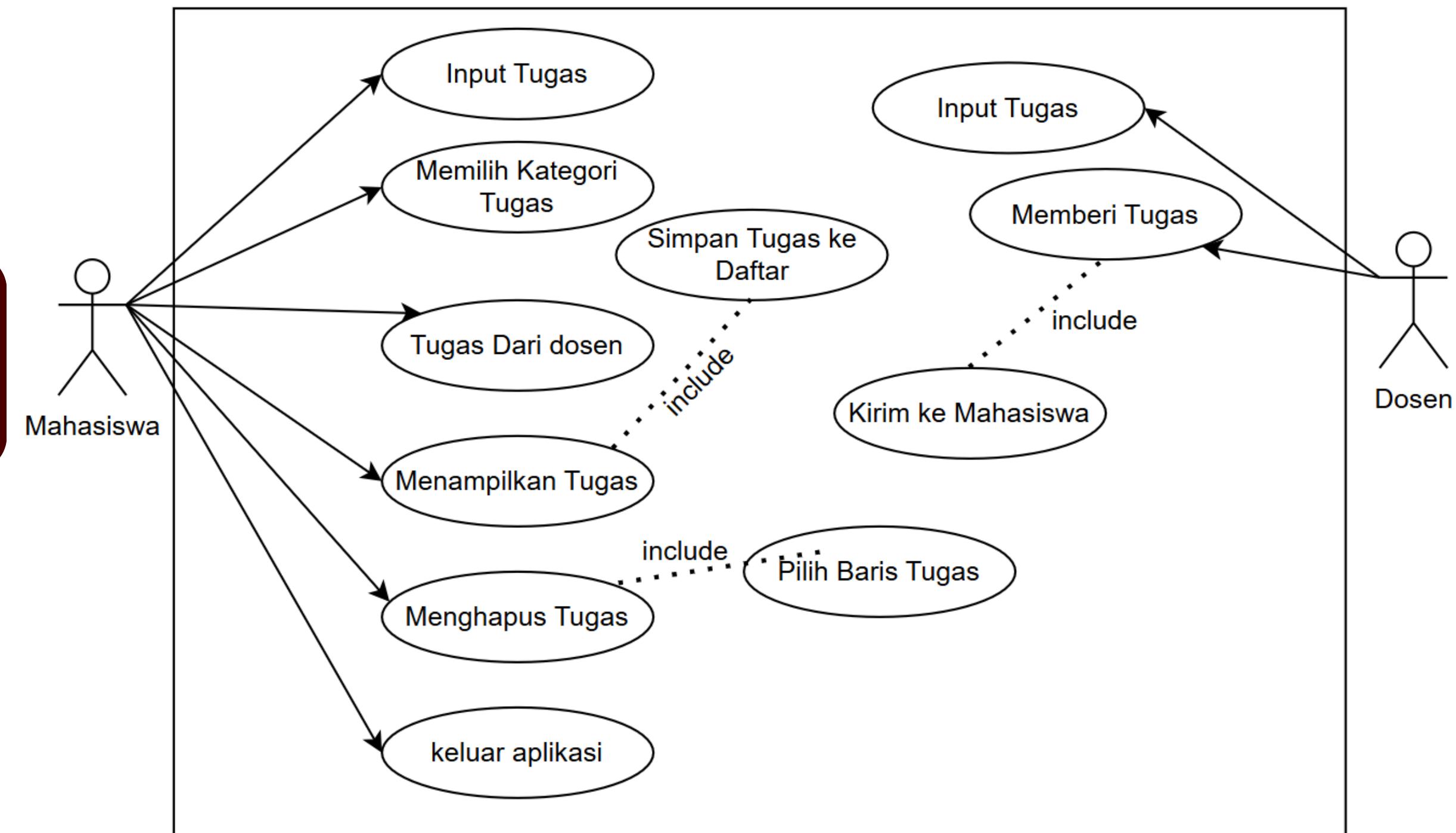
COMPONENT IDENTIFICATION



Kelompok 22



USE CASE DIAGRAM





USE CASE DESC

kelompok 22



Input Tugas

Name: Input Tugas

Initiator: Mahasiswa

Goal: Mahasiswa dapat memasukkan data tugas baru.

Main Success Scenario:

1. Mahasiswa memilih opsi "Input Tugas".
2. Sistem menampilkan form input tugas.
3. Mahasiswa mengisi data tugas (judul, deadline, deskripsi).
4. Sistem melakukan validasi input.
5. Sistem menyimpan tugas ke daftar.
6. Tugas berhasil ditambahkan dan ditampilkan pada daftar tugas.

Extensions:

- 4a. Jika input tidak valid, sistem menampilkan pesan kesalahan.

Memilih Kategori Tugas

Name: Memilih Kategori Tugas

Initiator: Mahasiswa

Goal: Memungkinkan mahasiswa memilih kategori tugas seperti "Tugas Kuliah", "Proyek", dll.

Main Success Scenario:

1. Saat input tugas, sistem menampilkan daftar kategori.
2. Mahasiswa memilih salah satu kategori.
3. Sistem mencatat kategori yang dipilih sebagai bagian dari tugas.



USE CASE DESC

kelompok 22



Tugas dari Dosen

Name: Tugas dari Dosen

Initiator: Sistem (berdasarkan aksi dari dosen)

Goal: Menyimpan tugas yang dikirim oleh dosen ke daftar mahasiswa.

Main Success Scenario:

- Dosen menginput tugas dan mengirim ke mahasiswa.
- Sistem menyimpan tugas ke daftar mahasiswa.
- Tugas dari dosen muncul dalam tampilan mahasiswa.

Memilih Kategori Tugas

Name: Menampilkan Tugas

Initiator: Mahasiswa

Goal: Melihat daftar tugas yang telah dimasukkan.

Main Success Scenario:

- Mahasiswa memilih menu "Menampilkan Tugas".
- Sistem mengambil data tugas dari penyimpanan.
- Sistem menampilkan seluruh tugas dalam tabel.
- **Extensions:**
- 2a. Jika tidak ada tugas, sistem menampilkan pesan "Belum ada tugas".



USE CASE DESC

kelompok 22



Simpan Tugas

Name: Simpan Tugas ke Daftar

Initiator: Sistem

Goal: Menyimpan tugas ke struktur data penyimpanan.

Included by: Input Tugas, Tugas dari Dosen

Main Success Scenario:

- Sistem menerima objek tugas.
- Tugas ditambahkan ke dalam daftar.

Pilih Baris Tugas

Name: Pilih Baris Tugas

Initiator: Mahasiswa

Goal: Memilih tugas dari daftar untuk diproses (hapus, lihat detail).

Included by: Menghapus Tugas, Menampilkan Tugas

Main Success Scenario:

- Mahasiswa mengklik salah satu baris tugas.
- Sistem menyorot baris dan menyimpan ID tugas tersebut.



USE CASE DESC

kelompok 22



Menghapus Tugas

Menghapus Tugas

Initiator: Mahasiswa

Goal: Menghapus tugas dari daftar yang dipilih.

Main Success Scenario:

- Mahasiswa memilih baris tugas dari daftar.
- Mahasiswa klik tombol "Hapus".
- Sistem meminta konfirmasi.
- Mahasiswa mengonfirmasi.
- Sistem menghapus tugas.
- **Extensions:**
- 1a. Jika tidak memilih baris tugas, sistem menampilkan pesan "Pilih tugas terlebih dahulu".
- 3a. Jika mahasiswa membatalkan, tugas tidak dihapus.

Memberi Tugas

Name: Memberi Tugas

Initiator: Dosen

Goal: Mengirimkan tugas ke mahasiswa.

Main Success Scenario:

- Dosen mengisi detail tugas.
- Sistem melakukan validasi.
- Sistem mengirim tugas ke mahasiswa.



USE CASE DESC

kelompok 22



Kirim ke Mahasiswa

Name: Kirim ke Mahasiswa

Initiator: Sistem

Goal: Mengirimkan data tugas ke mahasiswa.

Included by: Memberi Tugas

Main Success Scenario:

- Sistem mencatat dan mendistribusikan tugas ke akun mahasiswa.

Keluar Aplikasi

Name: Keluar Aplikasi

Initiator: Mahasiswa

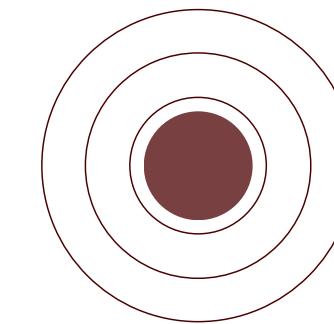
Goal: Menutup aplikasi dengan aman.

Main Success Scenario:

- Mahasiswa memilih opsi "Keluar Aplikasi".
- Sistem menutup aplikasi.



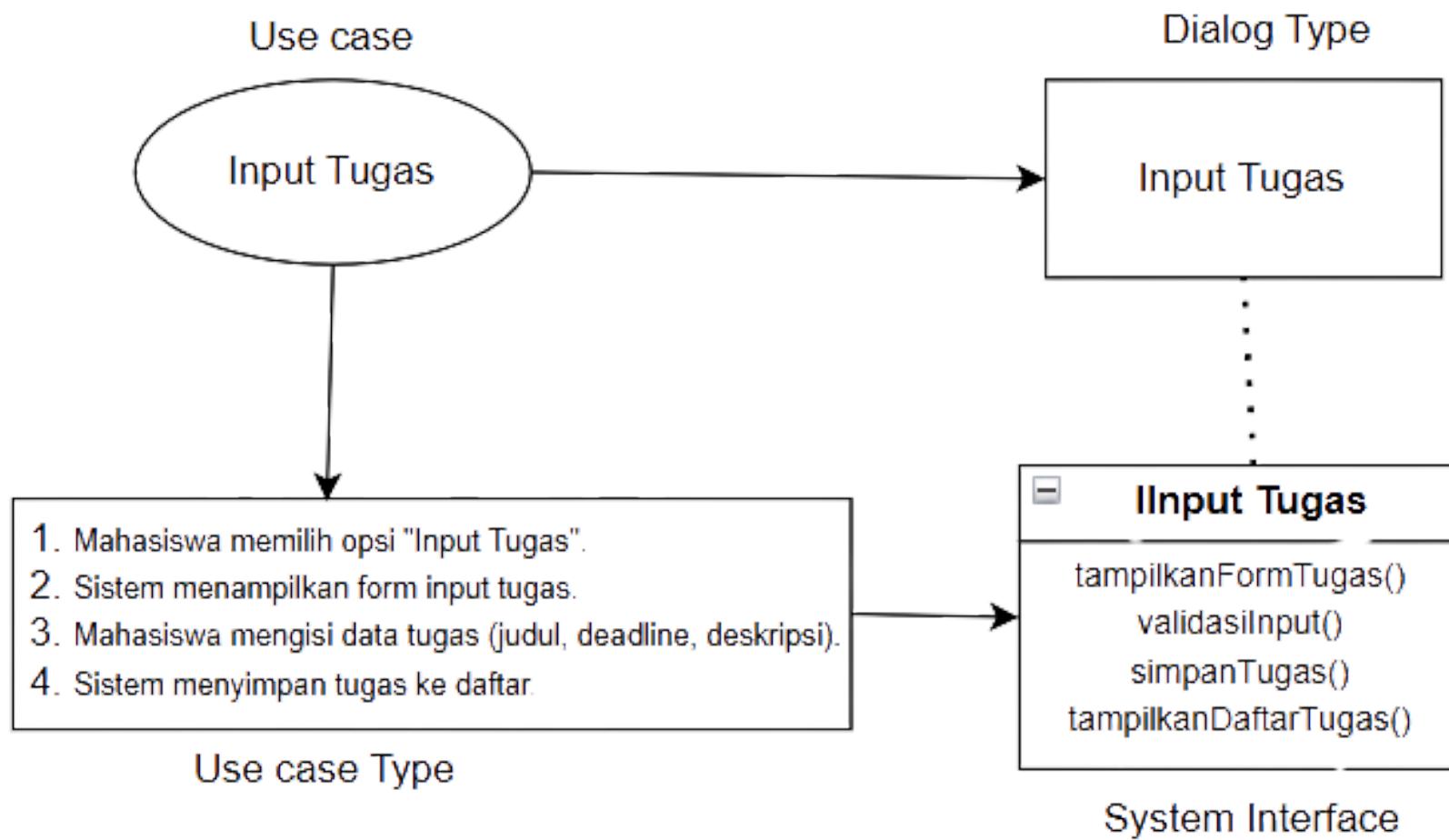
SYSTEM INTERFACE



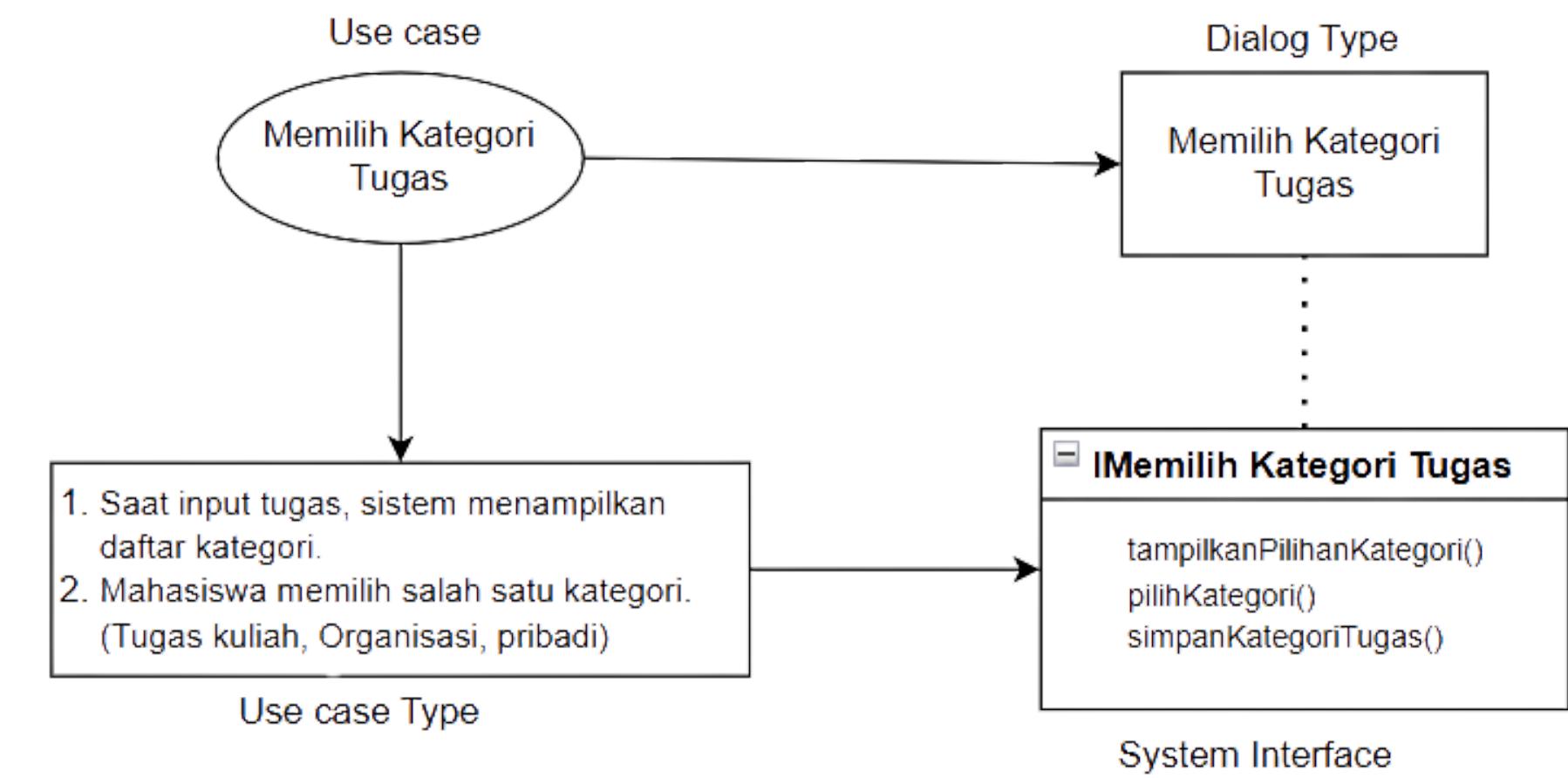
kelompok 22



Input Tugas

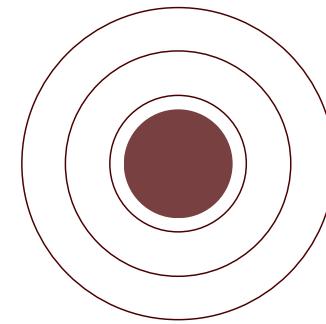


Memilih Kategori Tugas





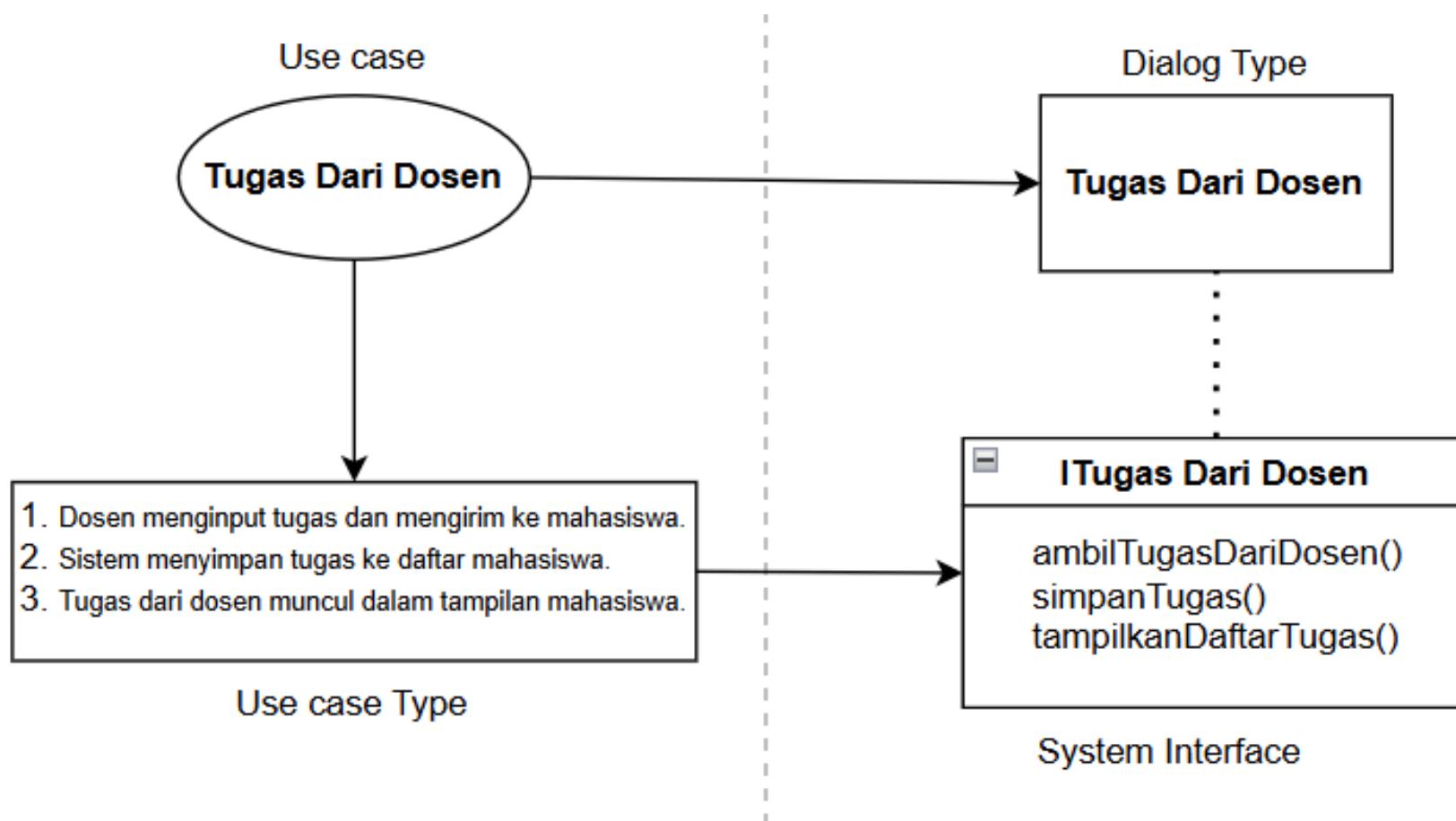
SYSTEM INTERFACE



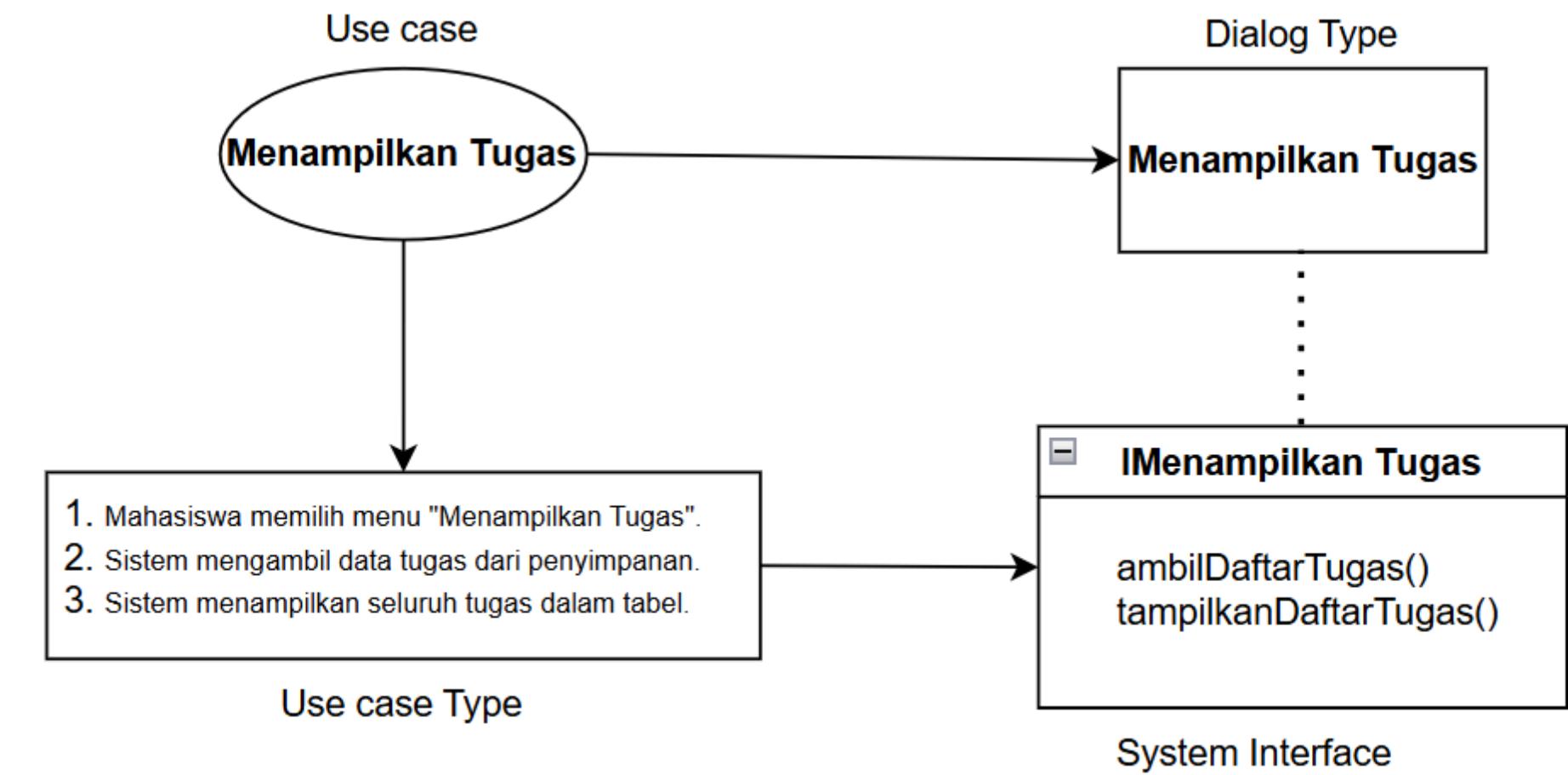
kelompok 22



Tugas Dari Dosen

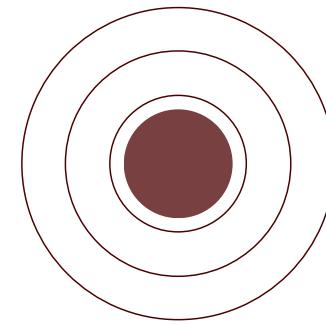


Menampilkan Tugas





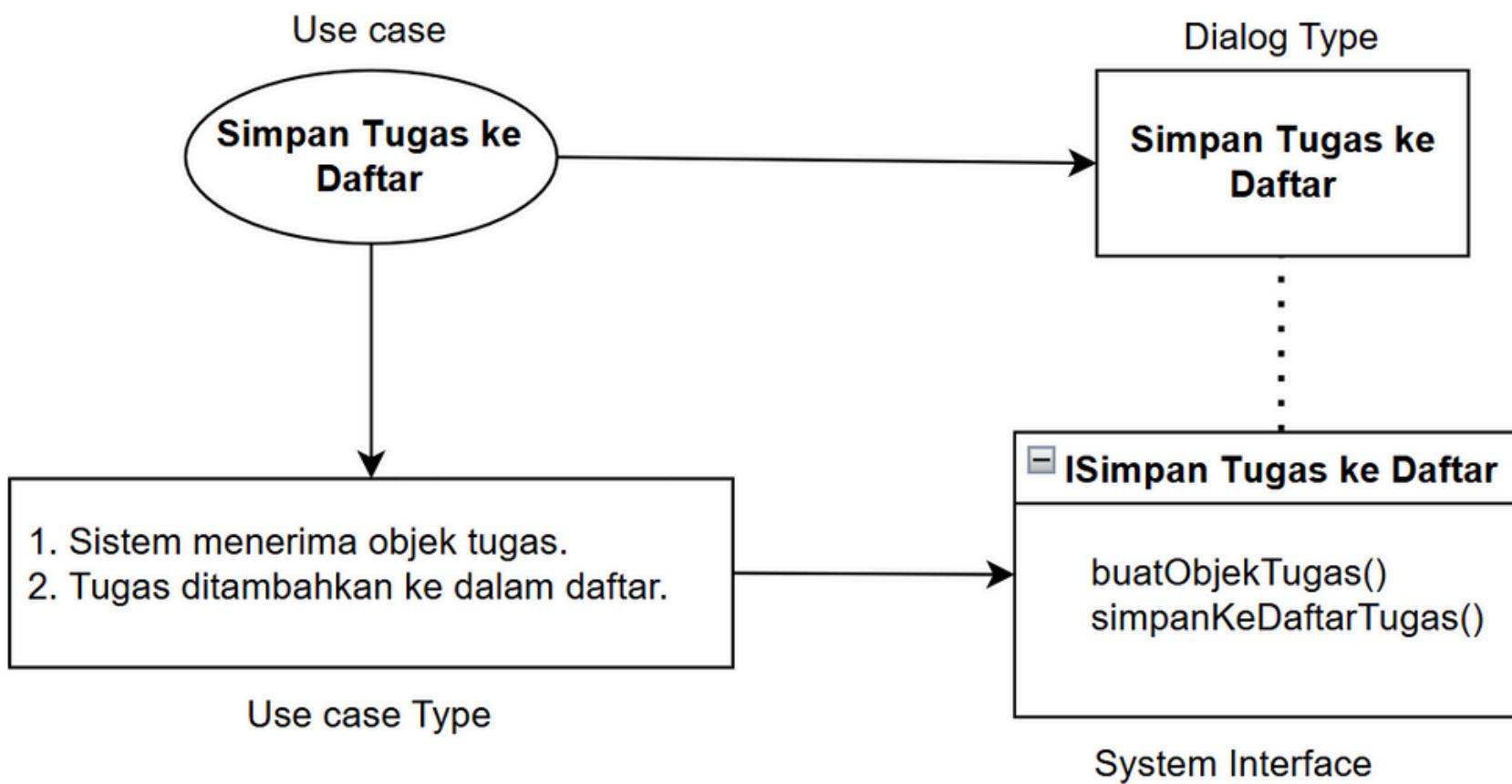
SYSTEM INTERFACE



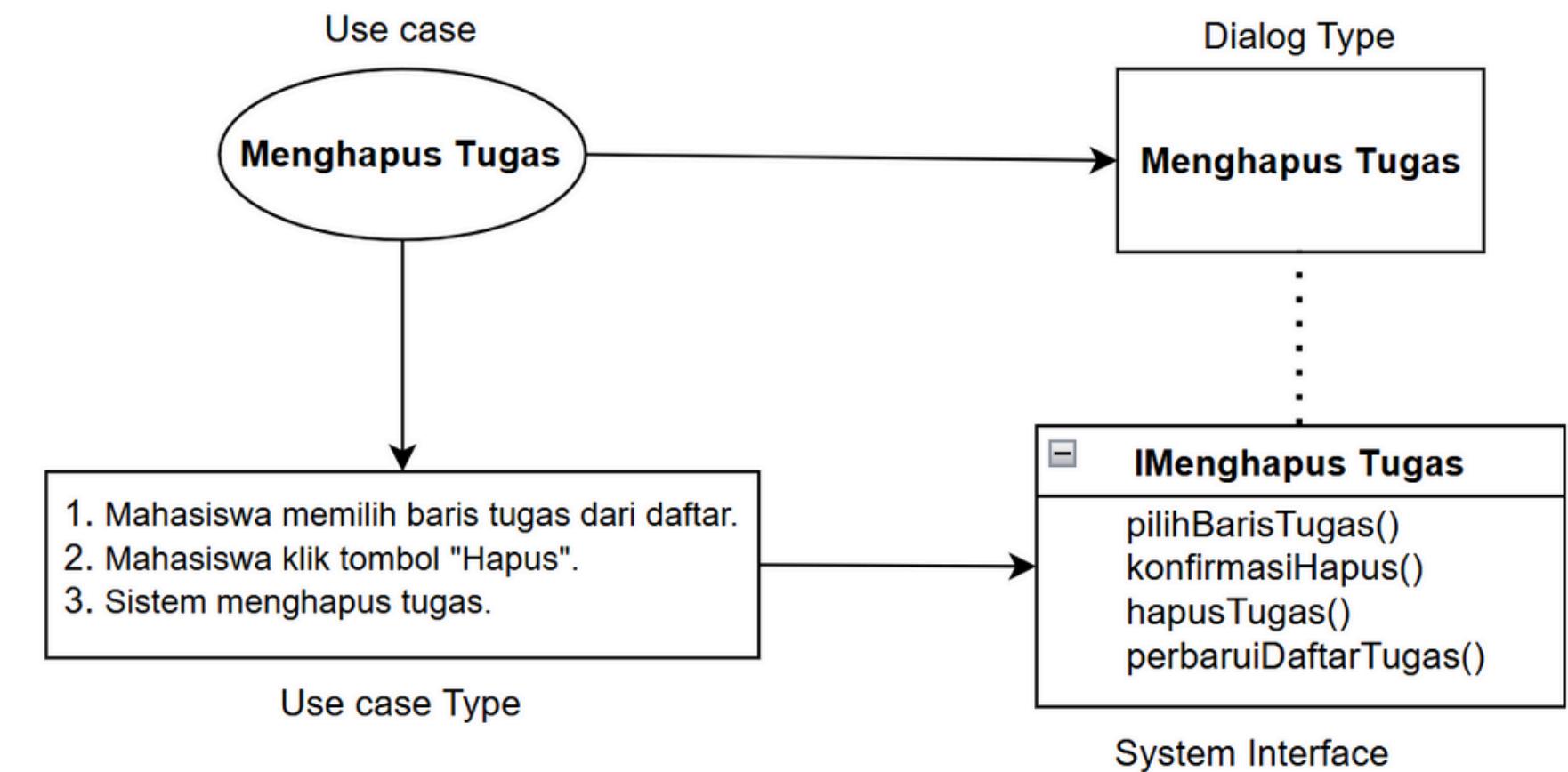
kelompok 22



Simpan Tugas ke Daftar

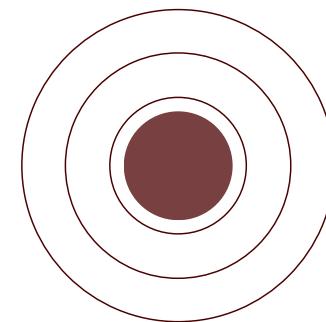


Menghapus Tugas





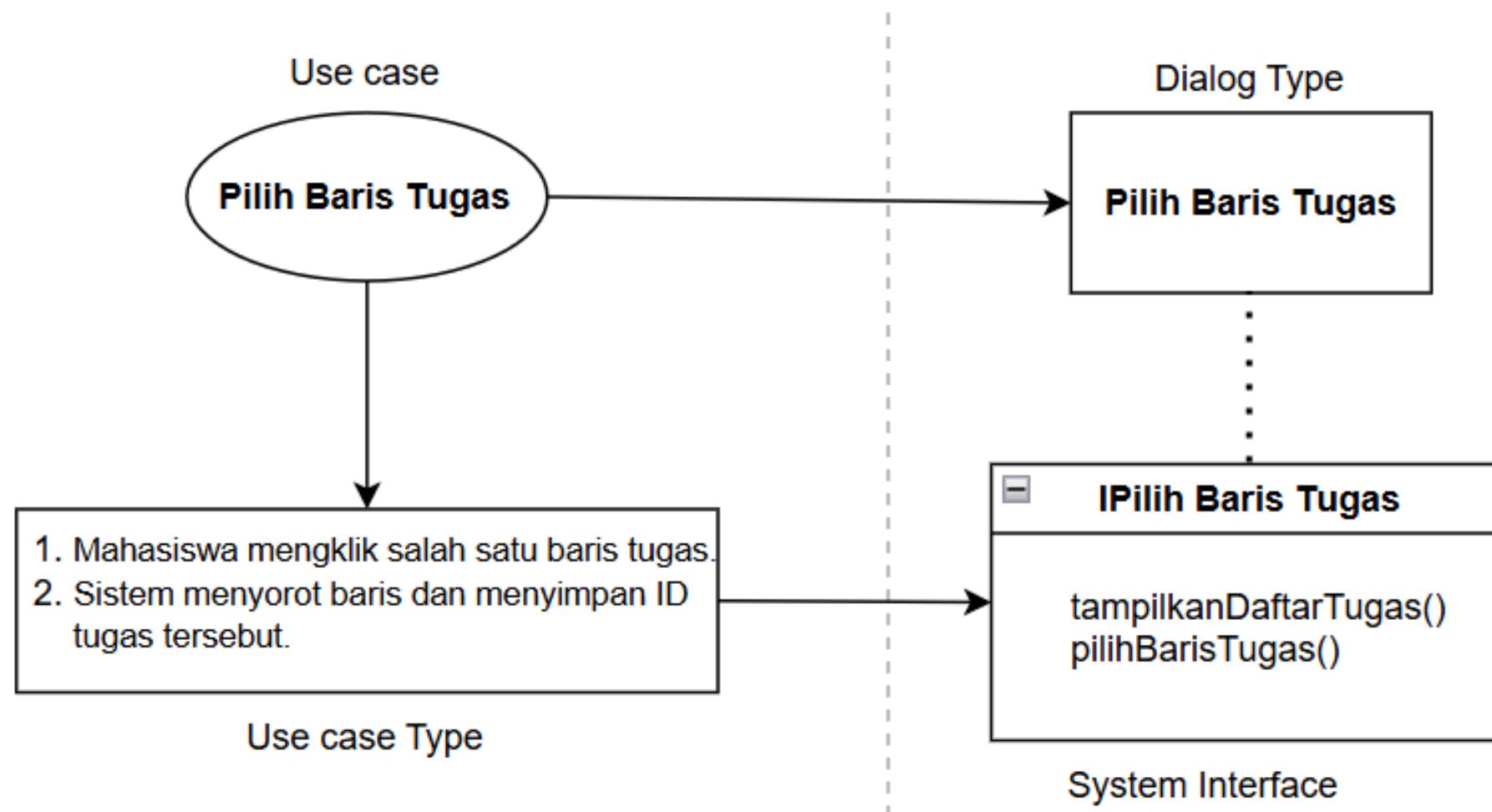
SYSTEM INTERFACE



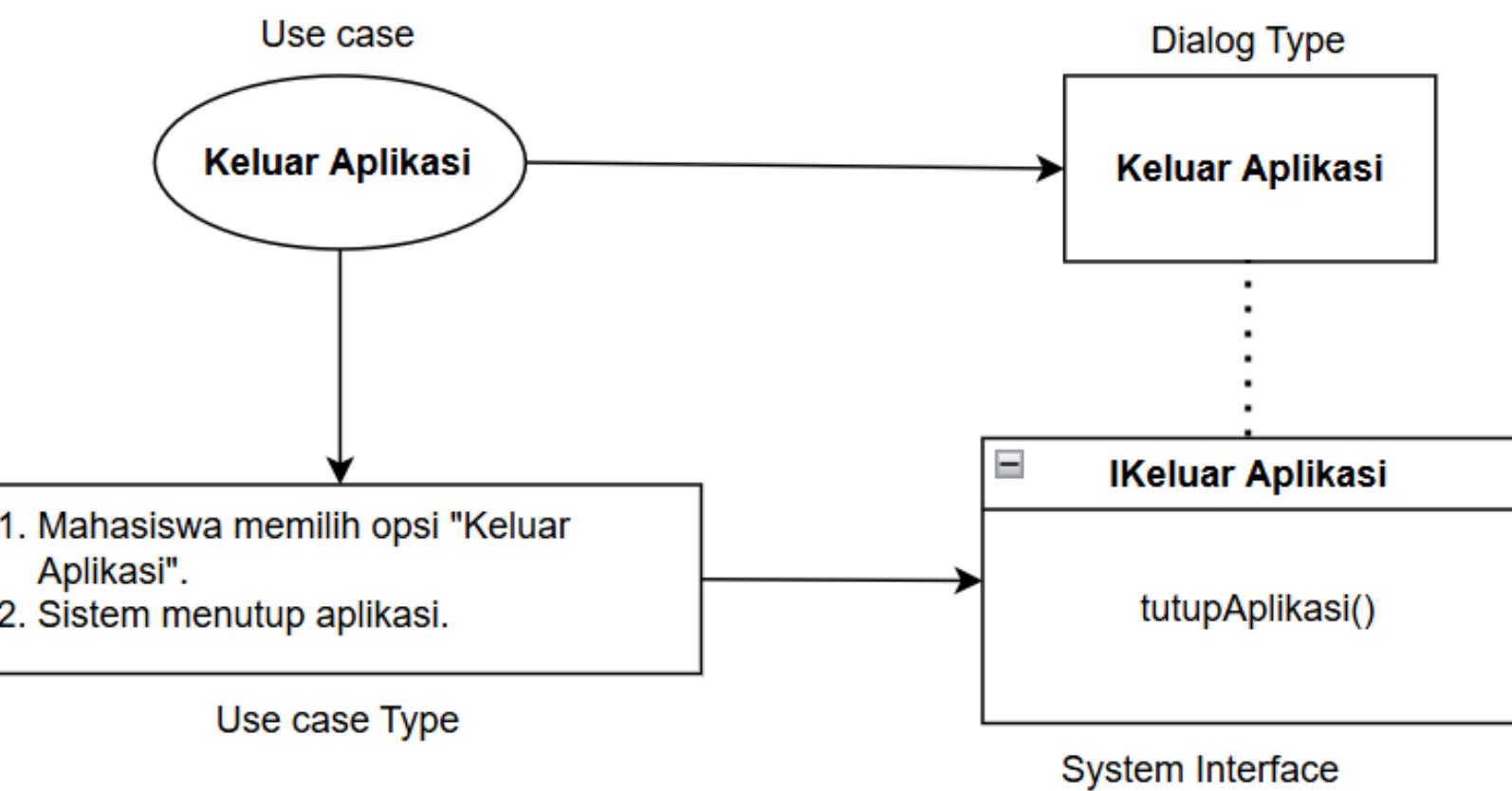
kelompok 22



Pilih Baris Tugas

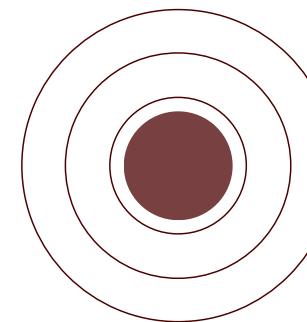


Keluar Aplikasi





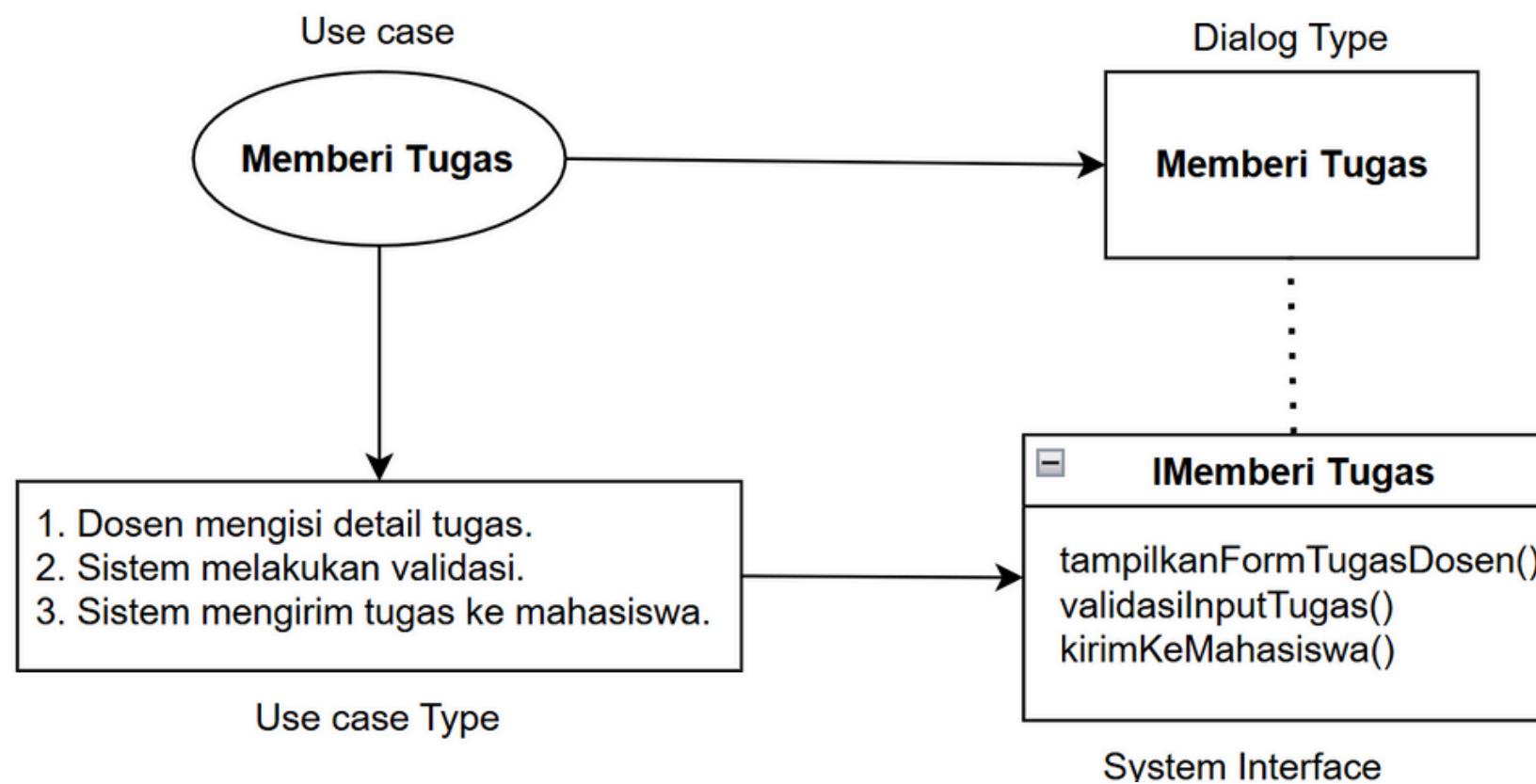
SYSTEM INTERFACE



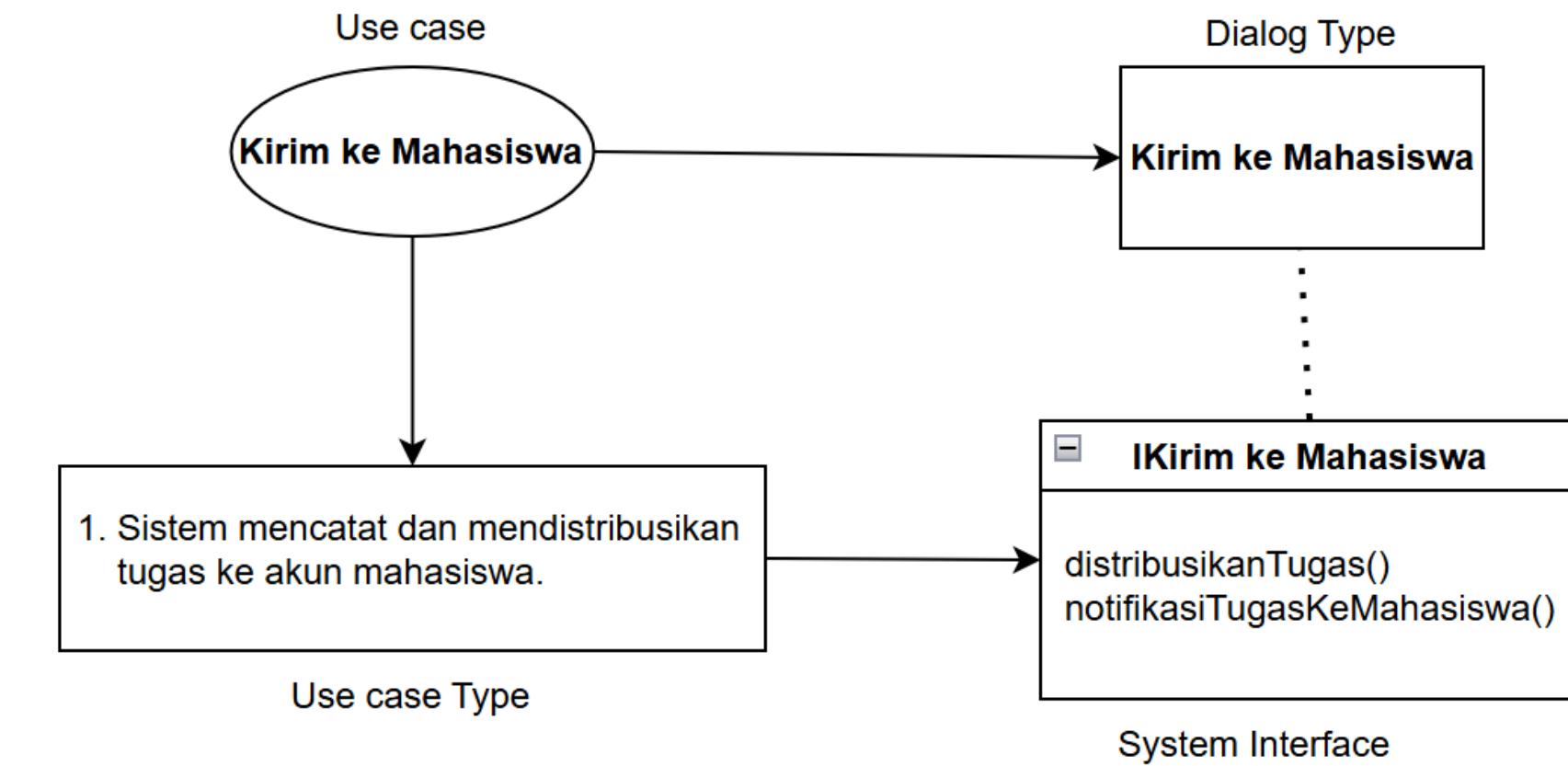
kelompok 22



Memberi Tugas



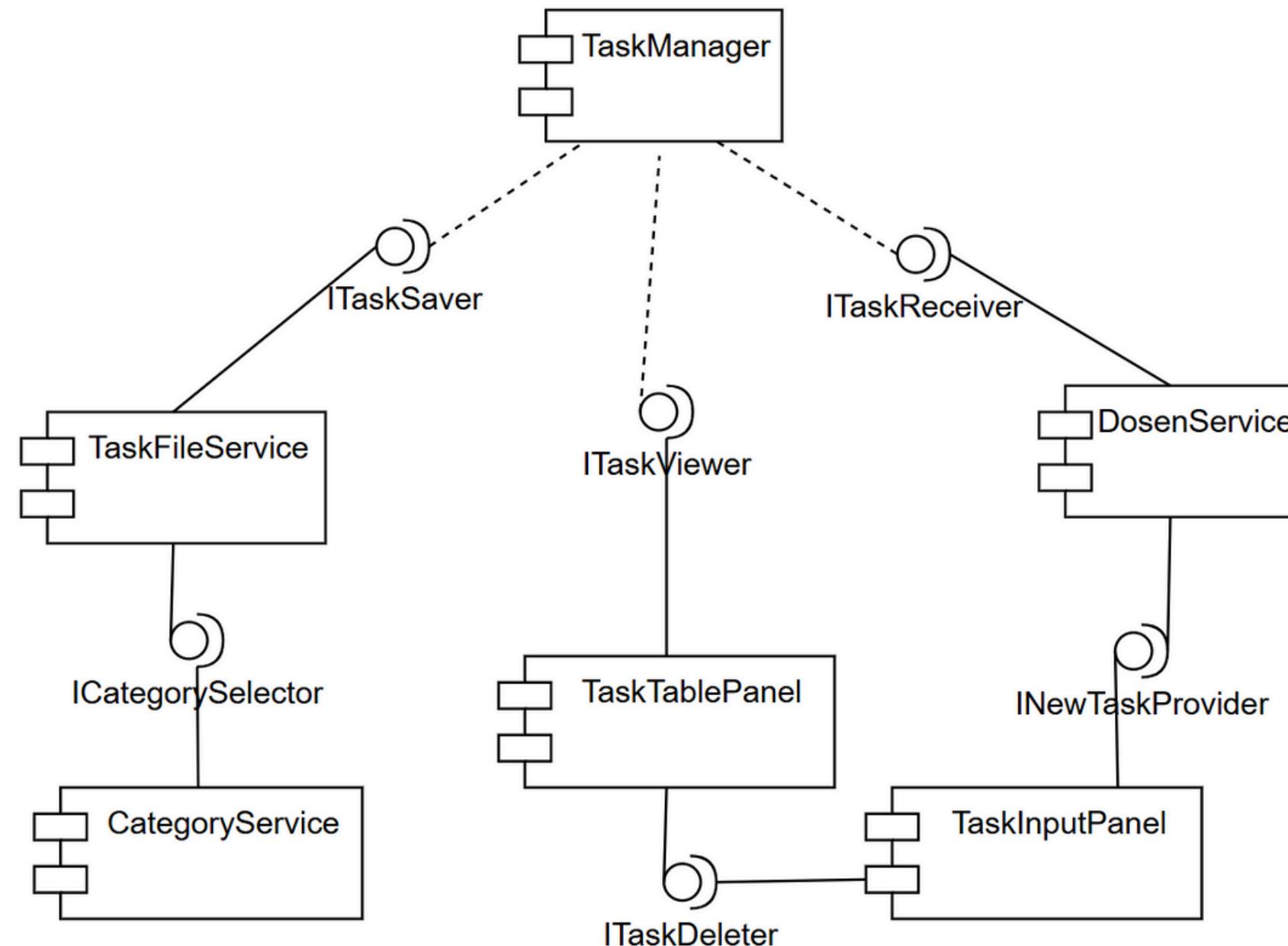
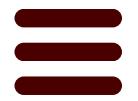
Kirim Ke Mahasiswa





PROVIDE DAN REQUIRE INTERFACE

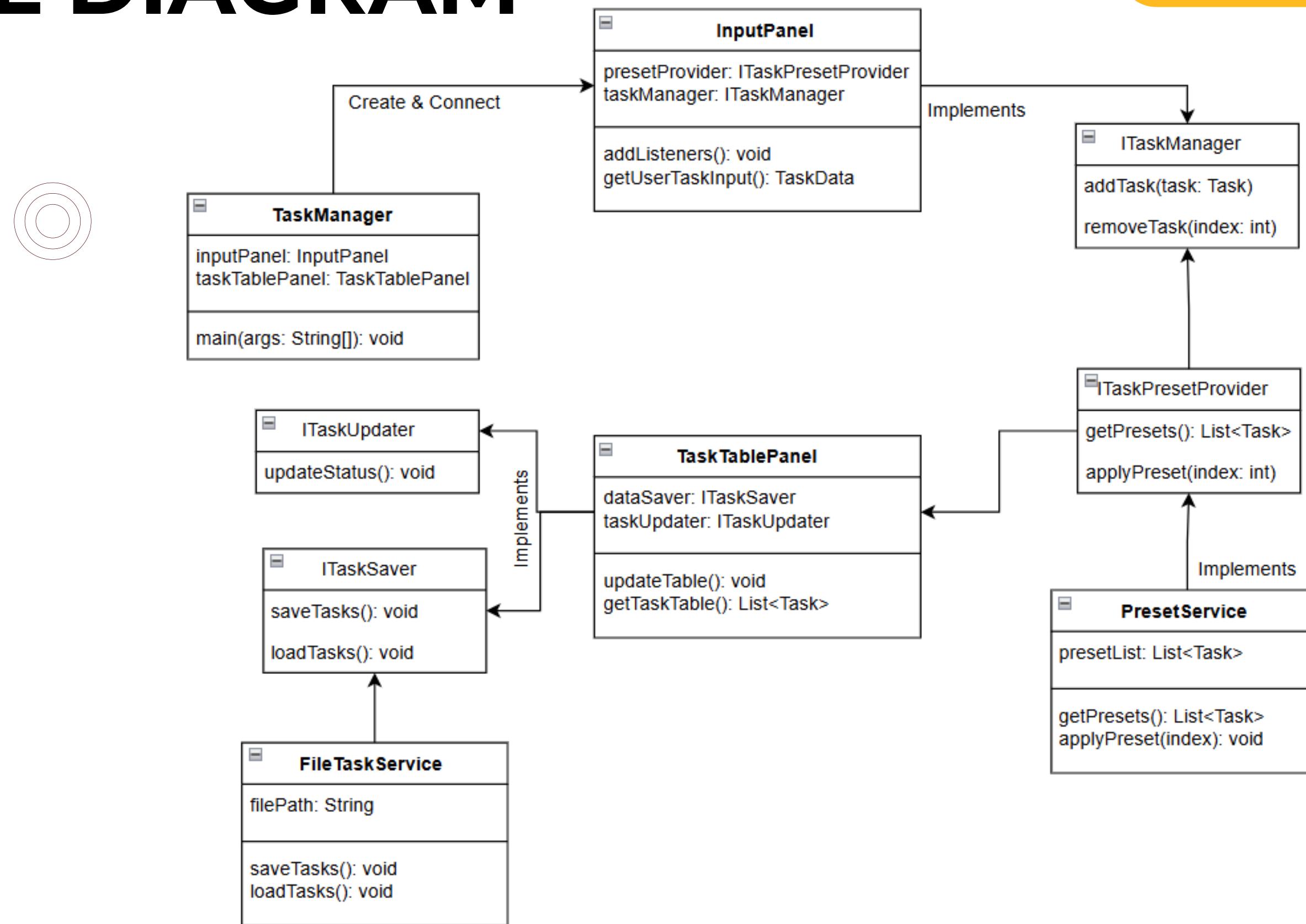
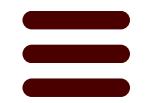
kelompok 22





UML DIAGRAM

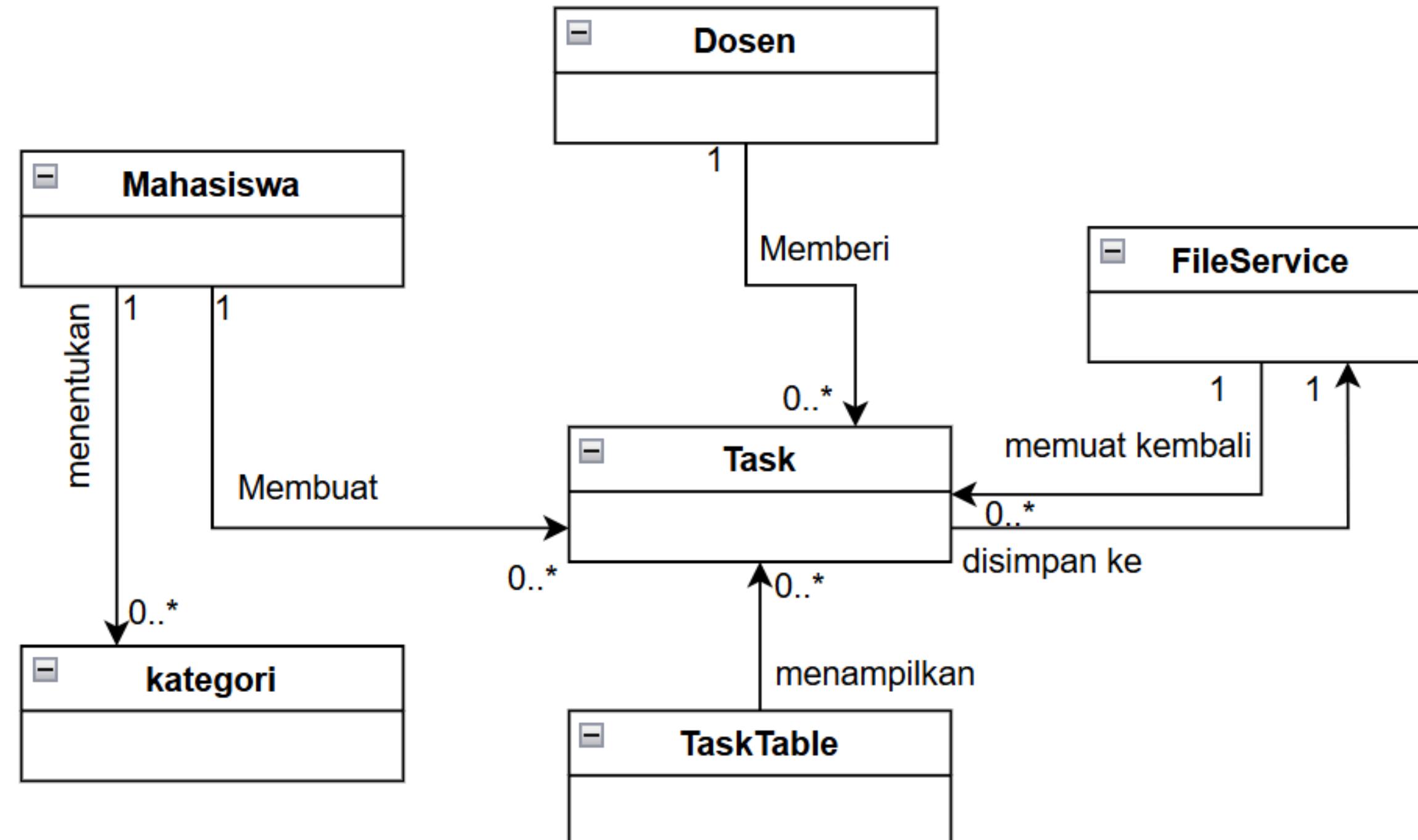
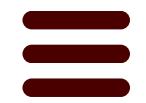
kelompok 22





BUSINESS CONCEPT MODEL

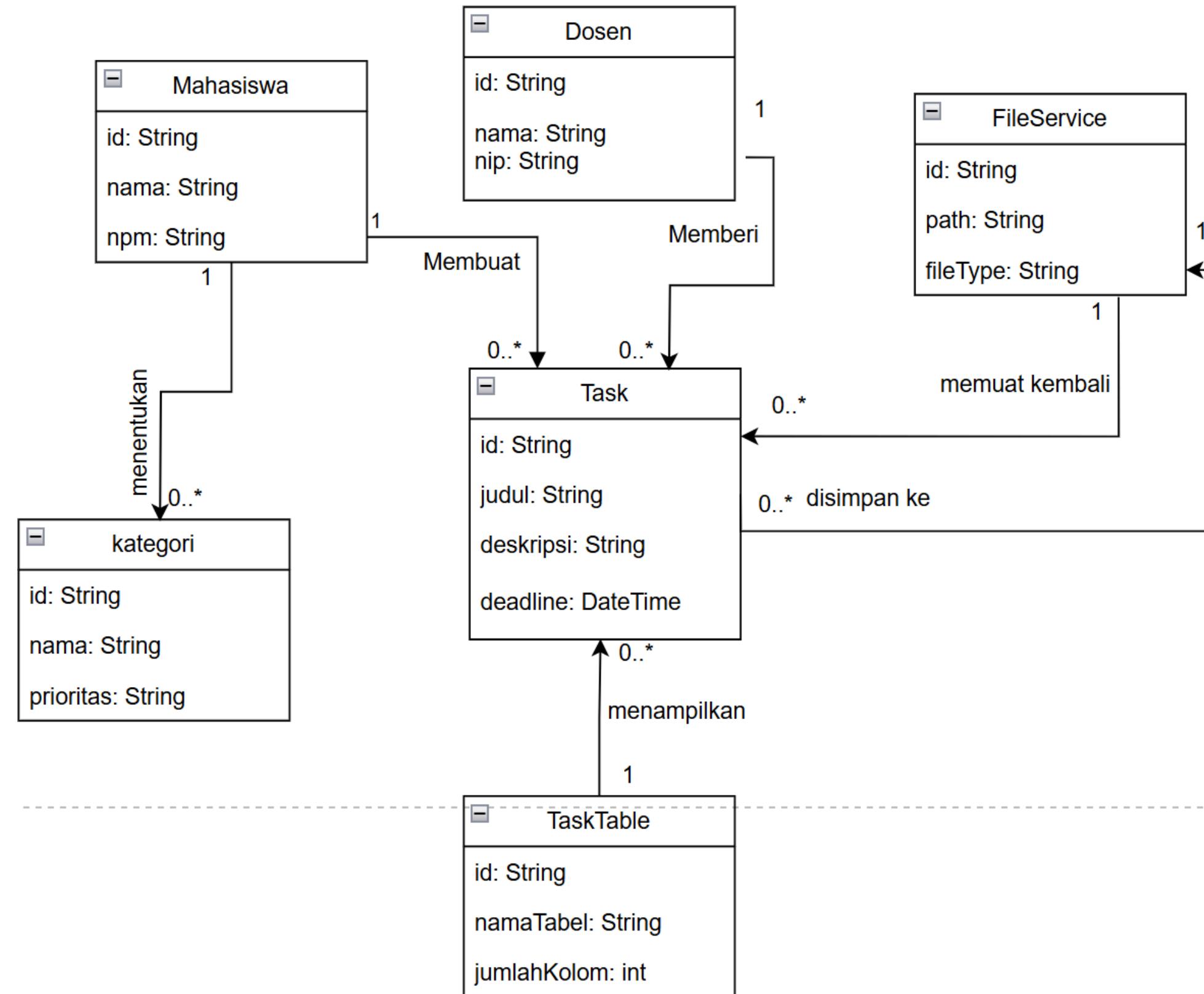
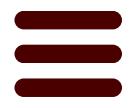
kelompok 22





BUSINESS TYPE MODEL

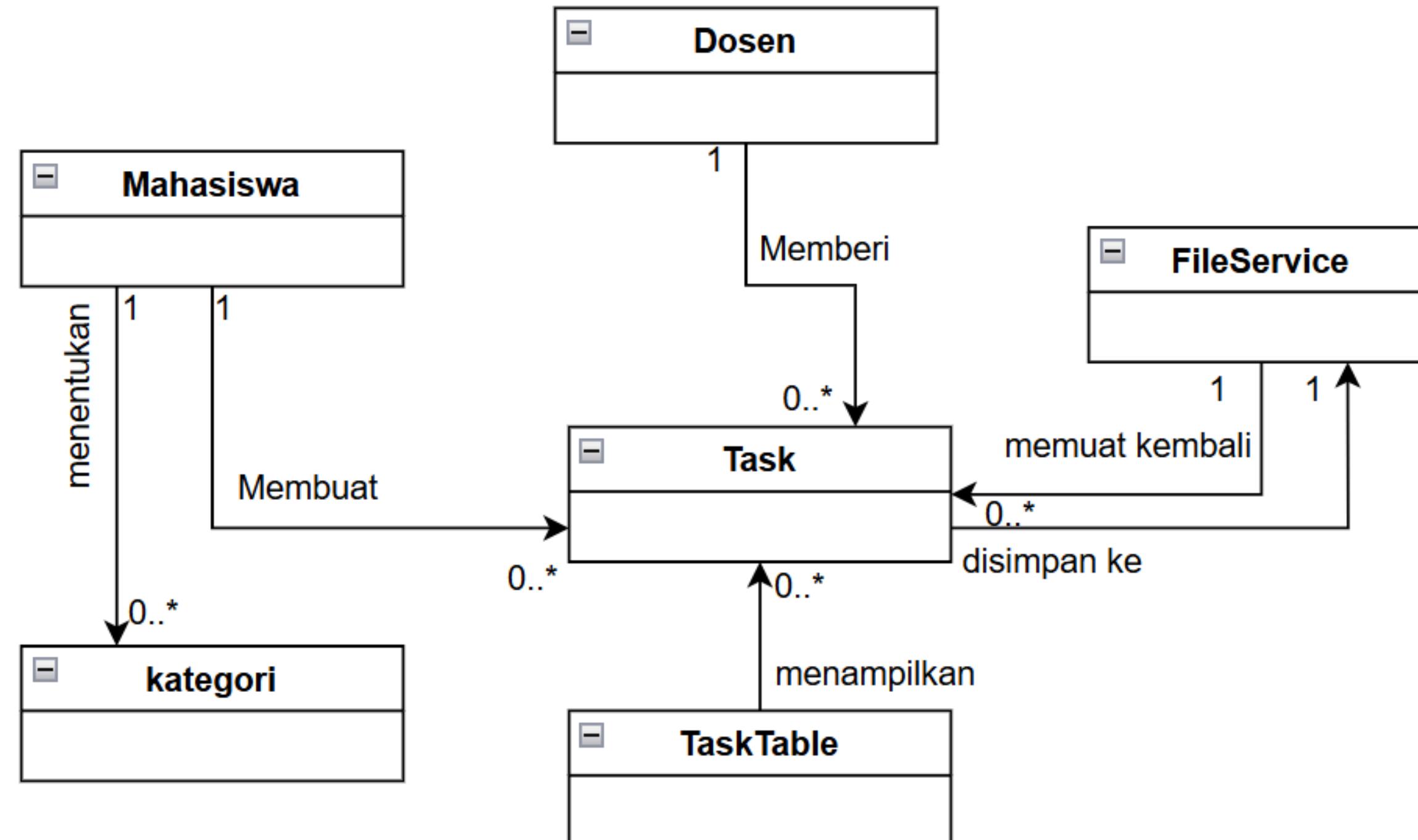
kelompok 22





BUSINESS CONCEPT MODEL

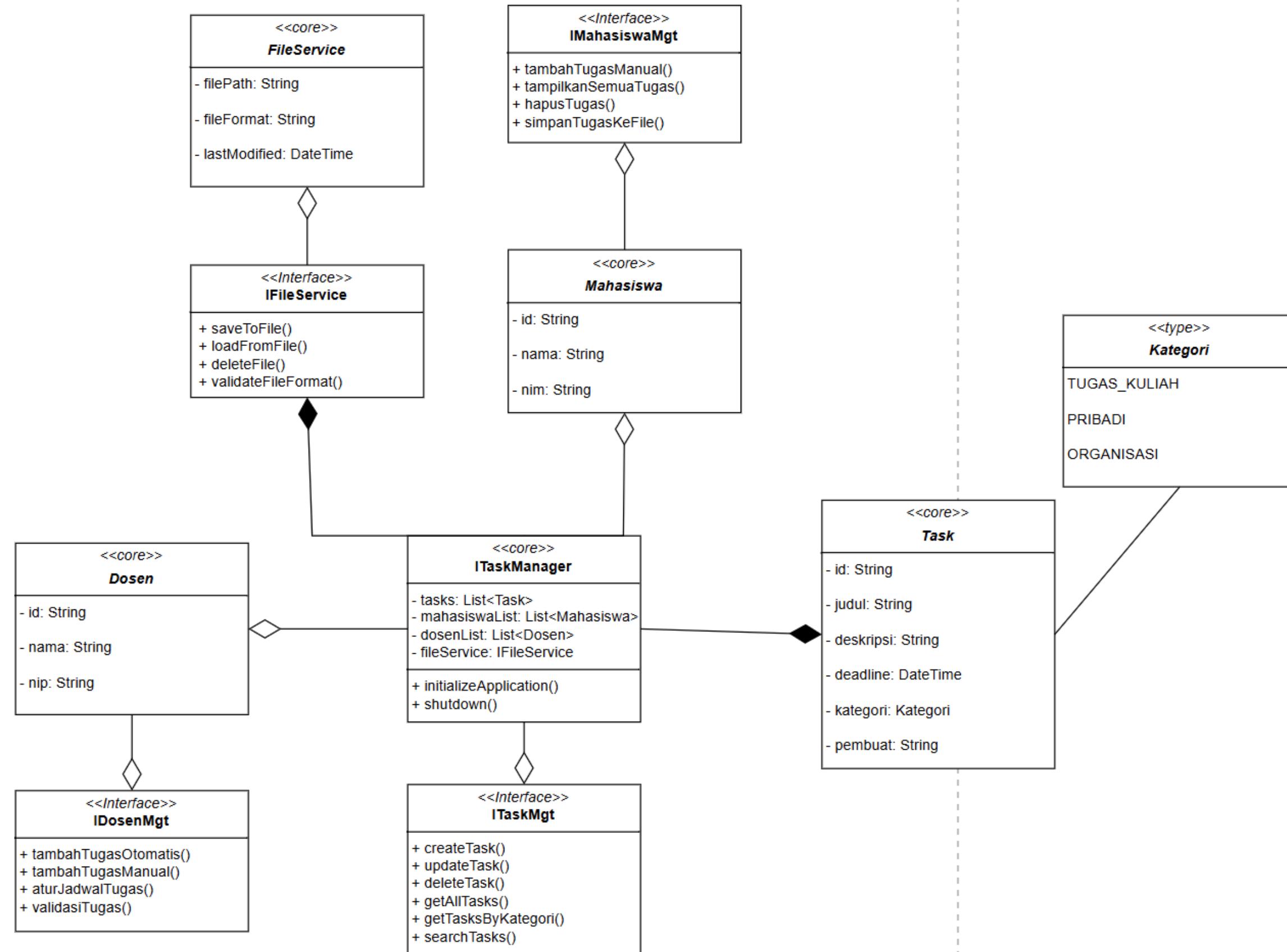
kelompok 22





BUSINESS INTERFACE

kelompok 22

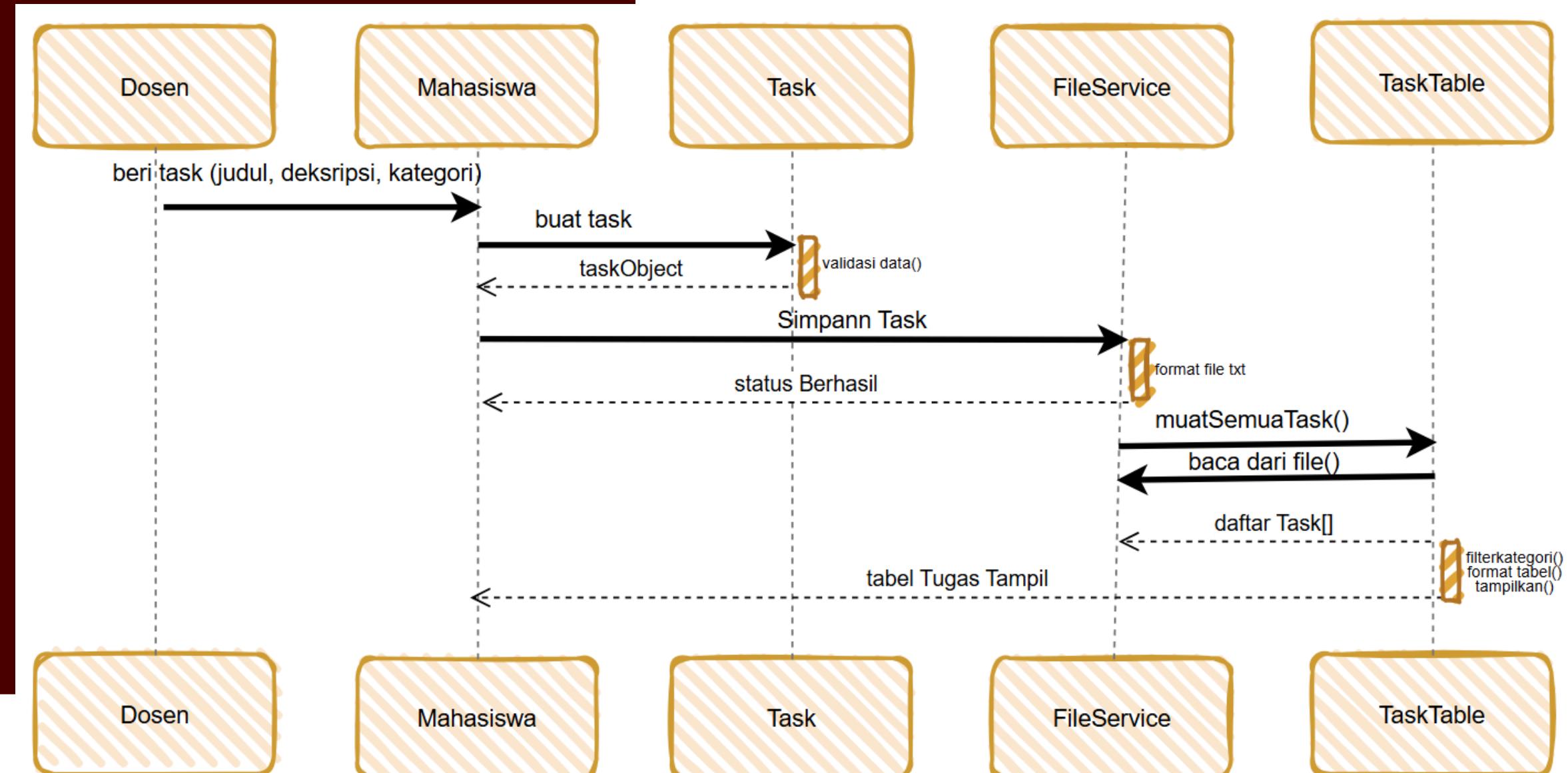


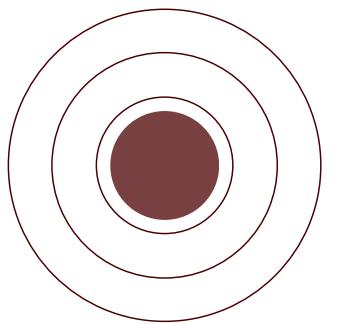


02.

Component Interaction

SEQUENCE DIAGRAM



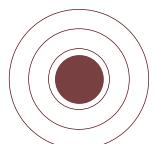


kelompok 22



03.

Component Specification



24



MEMBUAT TUGAS (CREATE TASK)

kelompok 22



```
● ● ●

1 context Task::createTask(id: String, judul: String, deskripsi: String, deadline: DateTime, kategori: Kategori, pembuat: String): Task
2 pre: -- ID tugas harus unik dalam sistem
3   TaskManager.allInstances()->forAll(tm |
4     tm.tasks->forAll(t | t.id <> id)
5   )
6 post: -- Tugas baru berhasil dibuat dan ditambahkan ke TaskManager
7   TaskManager.allInstances()->exists(tm |
8     tm.tasks->includes(result) and
9     result.id = id and
10    result.judul = judul and
11    result.deskripsi = deskripsi and
12    result.deadline = deadline and
13    result.kategori = kategori and
14    result.pembuat = pembuat
15  )
16 context Task::createTask(id: String, judul: String, deskripsi: String, deadline: DateTime, kategori: Kategori, pembuat: String): Task
17 pre: -- Deadline harus di masa depan
18   deadline > DateTime.now()
19 post: -- Deadline tersimpan dengan benar
20   result.deadline = deadline and result.deadline > DateTime.now()
```



MENAMPILKAN TUGAS (VIEW TASK)

kelompok 22



```
1 context TaskManager::getAllTasks() : Set(Task)
2
3 -- Precondition:
4 pre: -- Tidak ada prasyarat khusus untuk melihat tugas
5     true
6
7 -- Postcondition:
8 post: -- Mengembalikan semua tugas yang tersimpan di sistem
9     result = self.tasks->asSet()
10
```

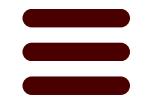


MENAMBAHKAN TIPE/KATEGORI TUGAS

kelompok 22



```
● ● ●  
1 context Task  
2 -- Invariant: Kategori harus salah satu dari enum yang valid  
3 inv KategoriValid:  
4   self.kategori = Kategori::TUGAS_KULIAH or  
5   self.kategori = Kategori::PRIBADI or  
6   self.kategori = Kategori::ORGANISASI  
7  
8 -- Precondition: Saat pembuatan atau pengubahan tugas  
9 pre:  
10  -- Pengguna harus sudah login  
11  self.createdBy <> null and  
12  
13  -- Kategori tidak boleh null  
14  self.kategori <> null  
15  
16 -- Postcondition: Setelah penyimpanan tugas  
17 post:  
18  -- Kategori tugas yang disimpan harus termasuk dalam nilai enum yang diperbolehkan  
19  self.kategori = Kategori::TUGAS_KULIAH or  
20  self.kategori = Kategori::PRIBADI or  
21  self.kategori = Kategori::ORGANISASI
```



DELETE TUGAS



```
1 context TaskManager::deleteTask(taskId: String): Boolean
2 pre: -- Tugas dengan ID tertentu harus ada
3     self.tasks->exists(t | t.id = taskId)
4 post: -- Tugas berhasil dihapus dari daftar
5     not self.tasks->exists(t | t.id = taskId) and
6     self.tasks->size() = self.tasks@pre->size() - 1
7
8 context TaskManager::deleteTask(taskId: String): Boolean
9 post: -- Tugas lain tidak terpengaruh oleh penghapusan ini
10    self.tasks->forAll(t |
11        t.id <> taskId implies
12        self.tasks@pre->exists(oldTask | oldTask = t)
13    )
```



TUGAS DARI DOSEN

```
● ● ●

1 context Dosen::tambahTugasManual(id: String, judul: String, deskripsi: String, deadline: DateTime): Task
2 pre: -- Dosen harus terdaftar dalam sistem
3     Dosen.allInstances()->includes(self) and
4     self.id <> null and self.id.size() > 0
5 post: -- Tugas berhasil dibuat dengan kategori TUGAS_KULIAH
6     result.pembuat = self.id and
7     result.kategori = Kategori::TUGAS_KULIAH
8 context Dosen::tambahTugasOtomatis(): Set(Task)
9 pre: -- Dosen harus memiliki mata kuliah yang valid
10    self.nip <> null and self.nip.size() > 0
11 post: -- Tugas otomatis dibuat berdasarkan jadwal kuliah
12    result->forAll(t |
13        t.pembuat = self.id and
14        t.kategori = Kategori::TUGAS_KULIAH and
15        t.deadline > DateTime.now()
16    )
```



kelompok 22



SEKIAN TERIMA KASIH

Sekali lagi, terima kasih atas kesempatan ini. Mari kita lanjutkan diskusi ini di lain waktu. Sampai jumpa!

SELESAI