

# Dasar-Dasar Pemrograman 2

## Tugas Pemrograman 04

### GUI Sistem Akademik

**Deadline: 04 Juni 2021, 23.55 WIB**



FAKULTAS  
**ILMU**  
**KOMPUTER**

#### Daftar Isi

<b>Dasar-Dasar Pemrograman 2</b>	<b>1</b>
Tugas Pemrograman 04	1
Daftar Isi	1
Pengantar	2
Ketentuan Umum:	2
Class SistemAkademikGUI	2
Class HomeGUI	3
Class TambahMahasiswaGUI	4
Class TambahMataKuliahGUI	7
Class TambahIRSGUI	10
Class HapusIRSGUI	17
Class RingkasanMahasiswaGUI	23
Class DetailRingkasanMahasiswaGUI	26
Class RingkasanMataKuliahGUI	29
Class DetailRingkasanMataKuliahGUI	32
Komponen Penilaian	33

## Pengantar

Tugas Pemrograman kali ini merupakan kelanjutan dari Tugas Pemrograman 2 di mana Anda diminta untuk membuat GUI dari Sistem Akademik Fasilkom. GUI yang dibuat merepresentasikan halaman untuk administrator dari Sistem Akademik Fasilkom di mana seorang administrator dapat mendaftarkan mahasiswa dan matakuliah, menambah dan menghapus IRS mahasiswa, serta melihat ringkasan mahasiswa dan mata kuliah. Secara umum, *flow* dari Tugas Pemrograman kali ini sama dengan Tugas Pemrograman 2 dengan sedikit perubahan minor. Kode *backend* (Kode pada TP 2) sudah disediakan pada template **(kecuali implementasi algoritma *sorting*, hal ini perlu diimplementasikan oleh Anda)** sehingga pada Tugas Pemrograman ini Anda dapat fokus pada implementasi GUI.

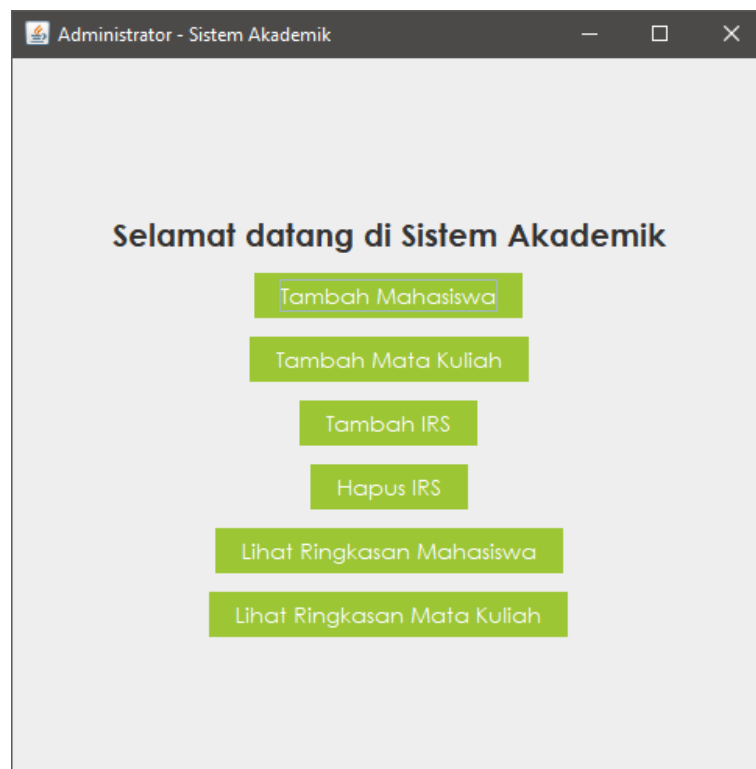
### Ketentuan Umum:

- a. Judul Halaman diletakkan di tengah halaman
- b. Ketika melakukan pergantian halaman, halaman sebelumnya “dihapus” terlebih dahulu. **Jangan melakukan penumpukan halaman.**
- c. *Styling* dan *Layout* dibebaskan selama tetap **memperhatikan segi estetika tampilan *User Interface***
- d. Silakan lakukan perubahan pada template yang telah disediakan jika Anda merasa perubahan tersebut dapat membantu anda dalam mengerjakan Tugas Pemrograman ini

### 1. *Class* SistemAkademikGUI

- a. *Class* untuk program *GUI* utama di mana *frame* akan diinisialisasi pada *Class* ini.
- b. Pada *Class* ini akan disimpan *ArrayList* daftar mahasiswa dan *ArrayList* daftar mata kuliah yang berisi daftar mahasiswa dan daftar mata kuliah di Fasilkom yang ditambahkan.
- c. Ketentuan *GUI*:
  - i. Ketika *window* ditutup, maka program harus berhenti.
  - ii. *Title bar* dari *window* adalah: **Administrator - Sistem Akademik**
  - iii. Halaman *default* yang ditampilkan saat program pertama kali dijalankan adalah halaman **Home**

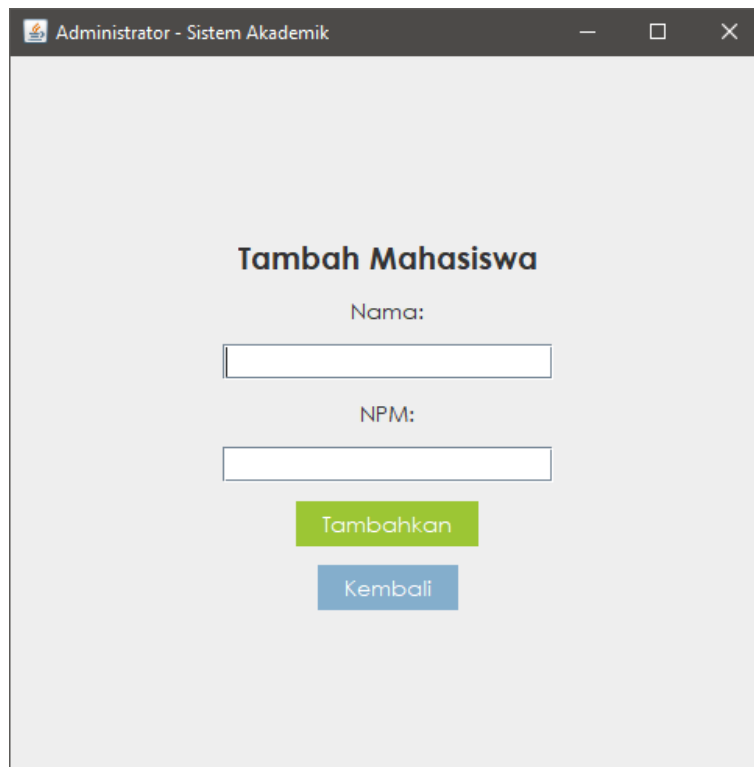
## 2. Class HomeGUI



Gambar 1. Contoh Tampilan Halaman *Home*

- a. Merupakan *landing page* dari program TP4 di mana terdapat menu untuk:
  - i. Tambah Mahasiswa: menambahkan mahasiswa ke daftar mahasiswa
  - ii. Tambah Mata Kuliah: menambahkan mata kuliah ke daftar mata kuliah
  - iii. Tambah IRS: menambahkan IRS mahasiswa
  - iv. Hapus IRS: menghapus IRS mahasiswa
  - v. Lihat Ringkasan Mahasiswa: melihat ringkasan seorang mahasiswa
  - vi. Lihat Ringkasan Mata Kuliah: melihat ringkasan suatu mata kuliah.
- b. Ketentuan *GUI*:
  - i. Terdapat judul halaman: **Selamat datang di Sistem Akademik**
  - ii. Saat *button* di klik, maka akan terjadi perpindahan ke halaman yang bersesuaian di *window* yang sama (bukan membuat *window* yang baru untuk setiap halaman)

### 3. Class TambahMahasiswaGUI



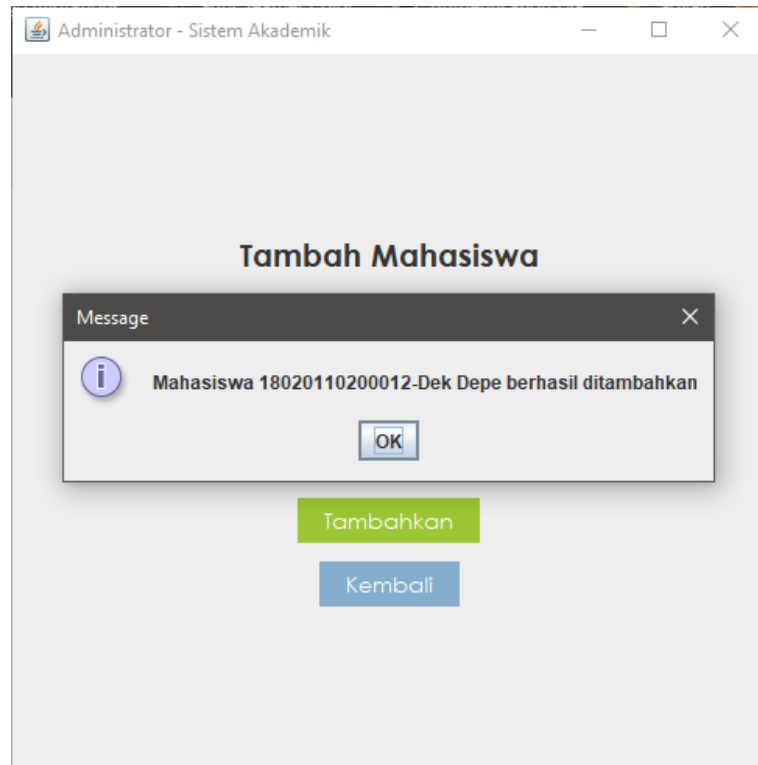
Gambar 2. Contoh Tampilan Halaman Tambah Mahasiswa

- a. Merupakan halaman untuk menambahkan (mendaftarkan) mahasiswa ke daftar mahasiswa yang ada.
- b. Ketentuan *GUI*:
  - i. Terdapat judul halaman: **Tambah Mahasiswa**
  - ii. Terdapat *form* yang berisi:

- *Label* dan *Field* untuk nama mahasiswa
- *Label* dan *Field* untuk npm mahasiswa
- *Button* Tambahkan untuk menambahkan mahasiswa

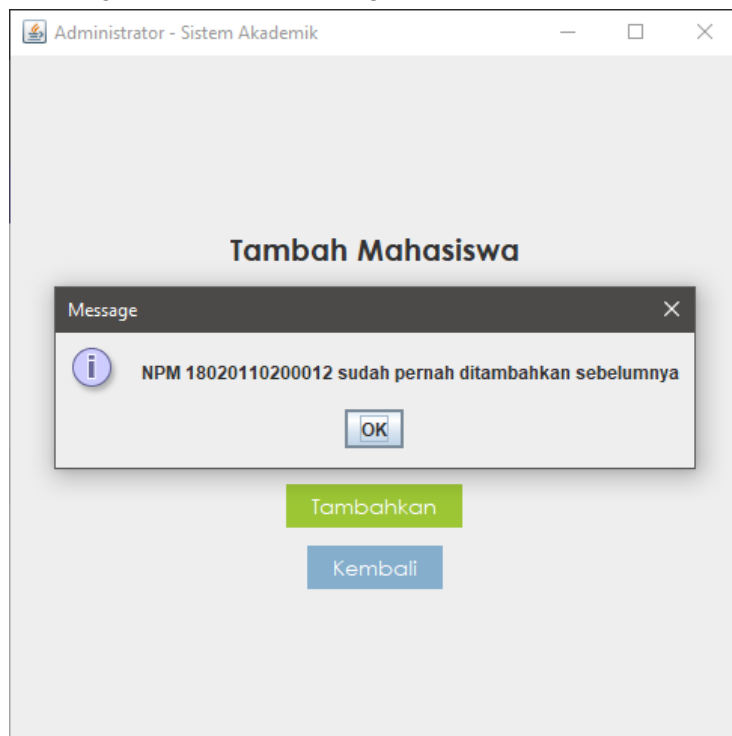
**Catatan:**

- Input *form* sudah dipastikan valid (nama hanya mengandung *alphabet* dan npm hanya mengandung angka, npm yang dimasukkan valid sesuai ketentuan TP2) sehingga Anda tidak perlu meng-*handle* kasus input yang tidak sesuai format.
- Apabila mahasiswa berhasil ditambahkan, maka tampilkan kotak dialog yang berisi pesan: **Mahasiswa [NPM]-[Nama] berhasil ditambahkan**. Selain itu, seluruh field akan kembali dikosongkan setelah mata kuliah berhasil ditambahkan.



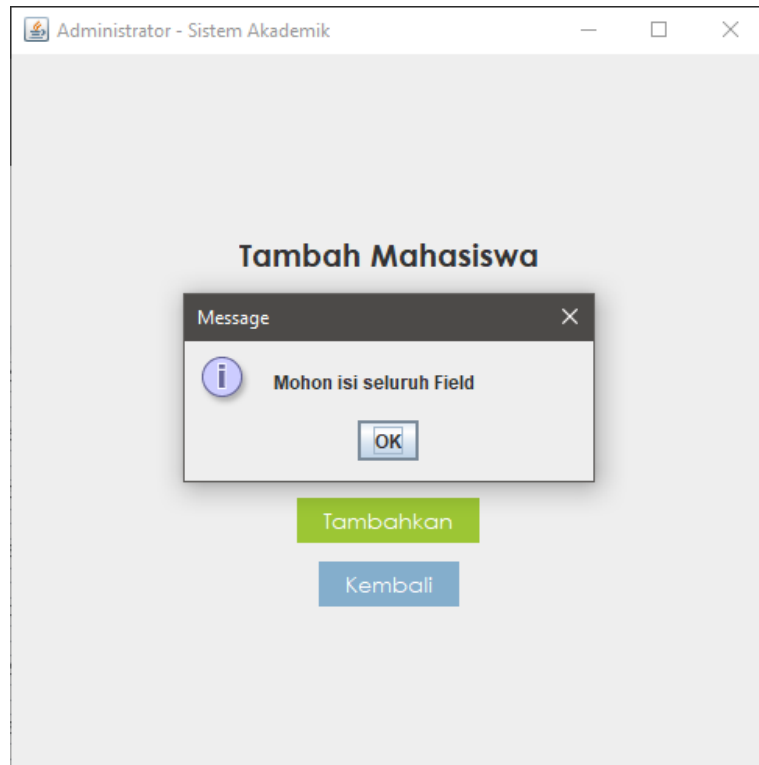
Gambar 3. Contoh Tampilan Ketika Mahasiswa Berhasil Ditambahkan

→ NPM Mahasiswa bersifat unik sehingga Anda harus meng-handle pengecekan apakah sudah ada mahasiswa dengan npm yang sama. Jika sudah pernah ditambahkan, maka penambahan gagal dan tampilkan kotak dialog berisi pesan: **NPM [NPM] sudah pernah ditambahkan sebelumnya**. Selain itu, kosongkan kembali seluruh field.



Gambar 4. NPM Sudah Pernah Dimasukkan Sebelumnya

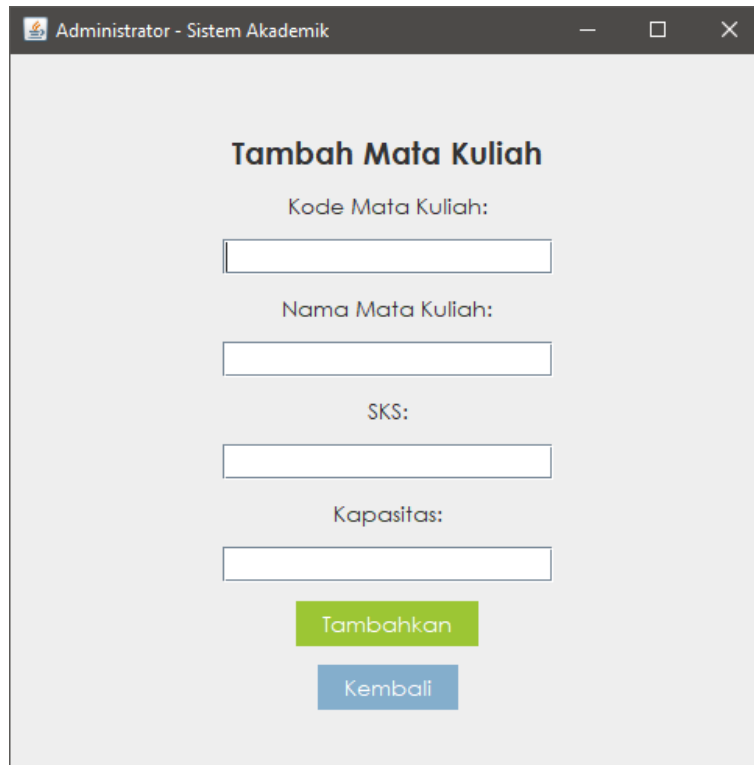
- Seluruh field harus terisi sehingga Anda perlu meng-*handle* kasus apabila terdapat minimal 1 *field* yang kosong dengan cara menampilkan kotak dialog yang berisi pesan: **Mohon isi seluruh Field**. Field yang sudah terisi tidak perlu dikosongkan.



Gambar 5. Contoh Tampilan Ketika Terdapat Field yang Tidak Terisi

- iii. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.

#### 4. Class TambahMataKuliahGUI

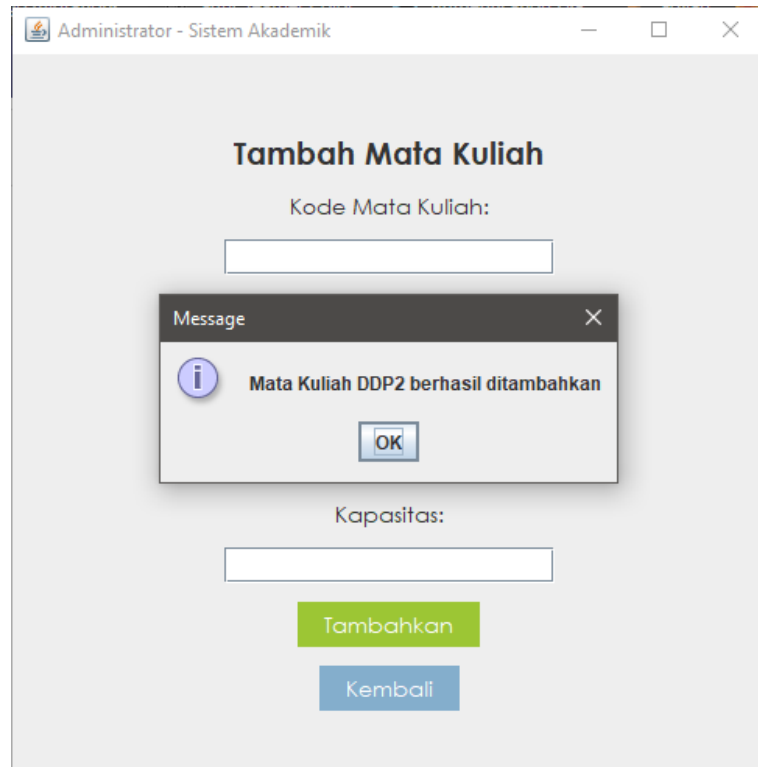


Gambar 6. Contoh Tampilan Halaman Tambah Mata Kuliah

- a. Merupakan halaman untuk menambahkan (mendaftarkan) mata kuliah ke daftar mata kuliah yang ada.
- b. Ketentuan *GUI*:
  - i. Terdapat judul halaman: **Tambah Mata Kuliah**
  - ii. Terdapat *form* yang berisi:
    - *Label* dan *Field* untuk kode mata kuliah
    - *Label* dan *Field* untuk nama mata kuliah
    - *Label* dan *Field* untuk sks mata kuliah
    - *Label* dan *Field* untuk kapasitas mata kuliah
    - *Button* Tambahkan untuk menambahkan mata kuliah

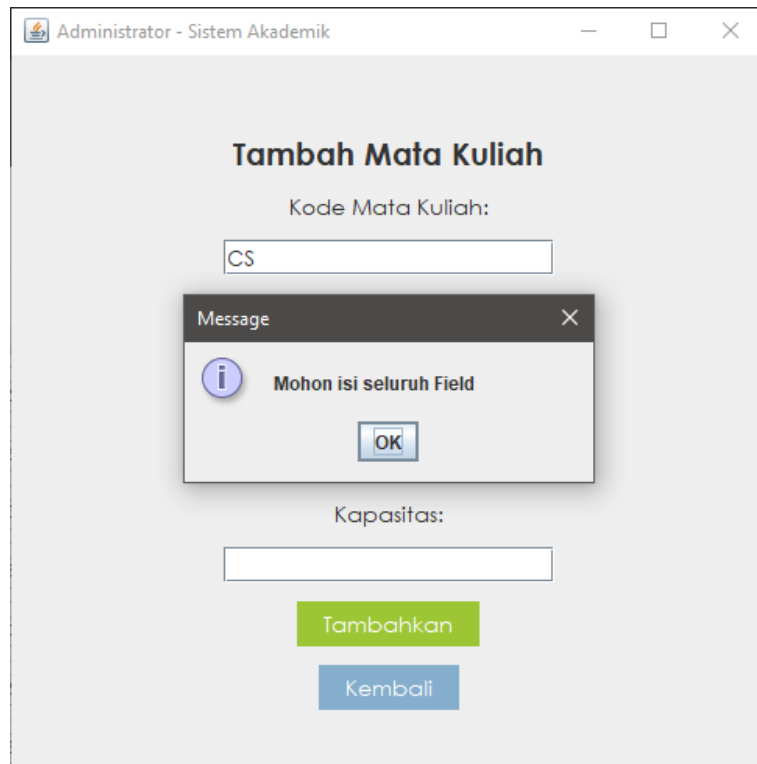
**Catatan:**

- Input *form* sudah dipastikan valid (kode hanya mengandung *alphabet* serta sks dan kapasitas hanya mengandung angka, kode mata kuliah pasti CS, IK atau SI, dan sebagainya di mana seluruh input dipastikan valid sesuai ketentuan TP 2) sehingga Anda tidak perlu meng-*handle* kasus input yang tidak sesuai format.
- Apabila mata kuliah berhasil ditambahkan, maka tampilkan kotak dialog yang berisi pesan: **Mata kuliah [Nama Mata Kuliah] berhasil ditambahkan**. Selain itu, seluruh field akan kembali dikosongkan setelah mata kuliah berhasil ditambahkan.



Gambar 7. Contoh Tampilan Ketika Mata Kuliah Berhasil Ditambahkan

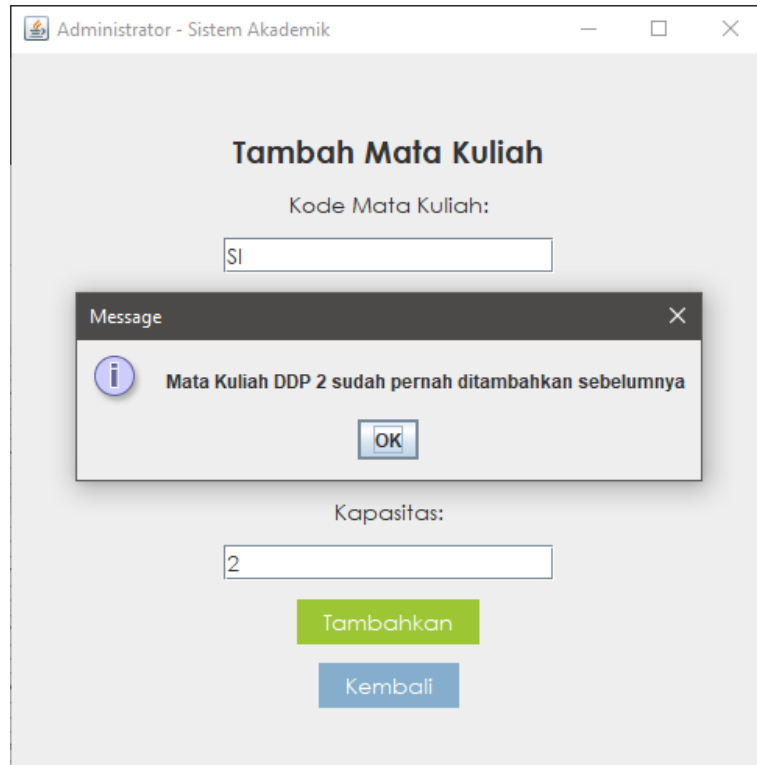
- Seluruh field harus terisi sehingga Anda perlu meng-*handle* kasus apabila terdapat minimal 1 *field* yang kosong dengan cara menampilkan kotak dialog yang berisi pesan: **Mohon isi seluruh Field**. Field yang sudah terisi tidak perlu dikosongkan.



Gambar 8. Contoh Tampilan Ketika Terdapat Field yang Tidak Terisi



- Nama Mata Kuliah bersifat unik sehingga Anda harus meng-handle pengecekan apakah sudah ada mata kuliah dengan nama yang sama. Jika sudah pernah ditambahkan, maka penambahan gagal dan tampilkan kotak dialog berisi pesan: **Mata Kuliah [Nama Mata Kuliah] sudah pernah ditambahkan sebelumnya.** Seluruh *field* kembali dikosongkan.



Gambar 9. Mata Kuliah Sudah Pernah Ditambahkan Sebelumnya

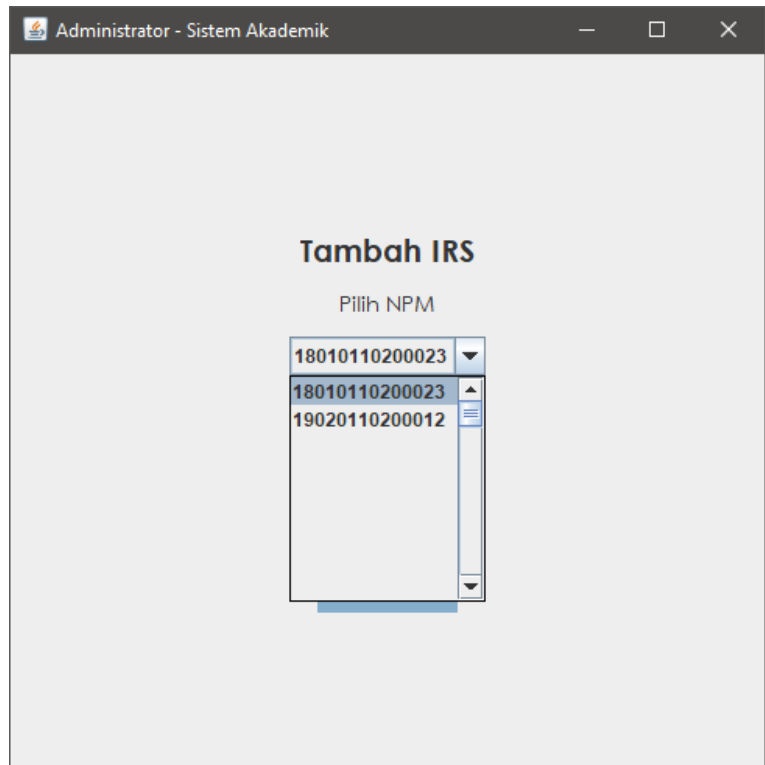
- iii. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.

## 5. Class TambahIRSGUI



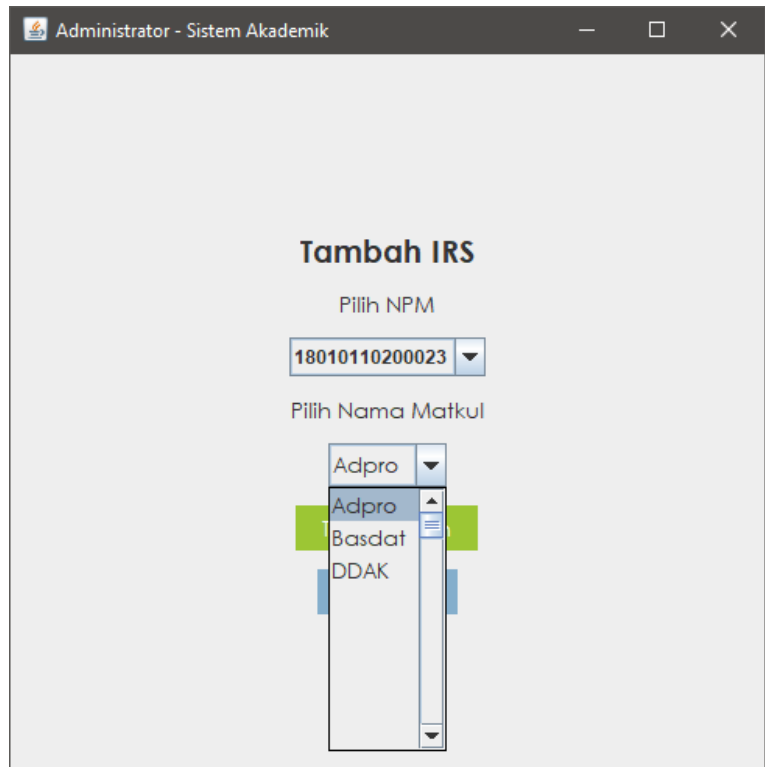
Gambar 10. Halaman Tambah IRS

- a. Merupakan halaman untuk menambahkan IRS seorang mahasiswa. Terdapat perbedaan dengan TP 2 di mana pada TP 2 saat memilih menu Add IRS maka akan ditanya berapa mata kuliah yang akan dimasukkan. Sedangkan pada TP kali ini, Anda hanya bisa menambahkan satu mata kuliah pada satu waktu..
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Tambah IRS**
  - ii. Terdapat form yang berisi:
    - Dropdown untuk memilih NPM Mahasiswa dengan ketentuan:
      - Pilihan pada dropdown ditampilkan secara terurut dari NPM kecil ke besar. Silakan implementasikan **algoritma *sorting*** agar NPM ditampilkan secara terurut. **Tidak diperkenankan menggunakan built-in method `sort` dari Java. Anda harus mengimplementasikan *sorting* secara manual.**



Gambar 11. NPM ditampilkan secara teratur

- Dropdown untuk memilih mata kuliah dengan ketentuan:
  - Pilihan pada dropdown ditampilkan teratur secara *alphabetical order*. Silakan implementasikan **algoritma sorting** agar Mata Kuliah ditampilkan secara teratur. **Tidak diperkenankan menggunakan built-in method sort dari Java. Anda harus mengimplementasikan sorting secara manual.**
  - Cukup tampilkan daftar mata kuliah yang ada dalam sistem. Jika ada mata kuliah yang sudah ditambahkan mahasiswa, di-handle dengan menampilkan kotak dialog setelah menekan tombol tambahkan.

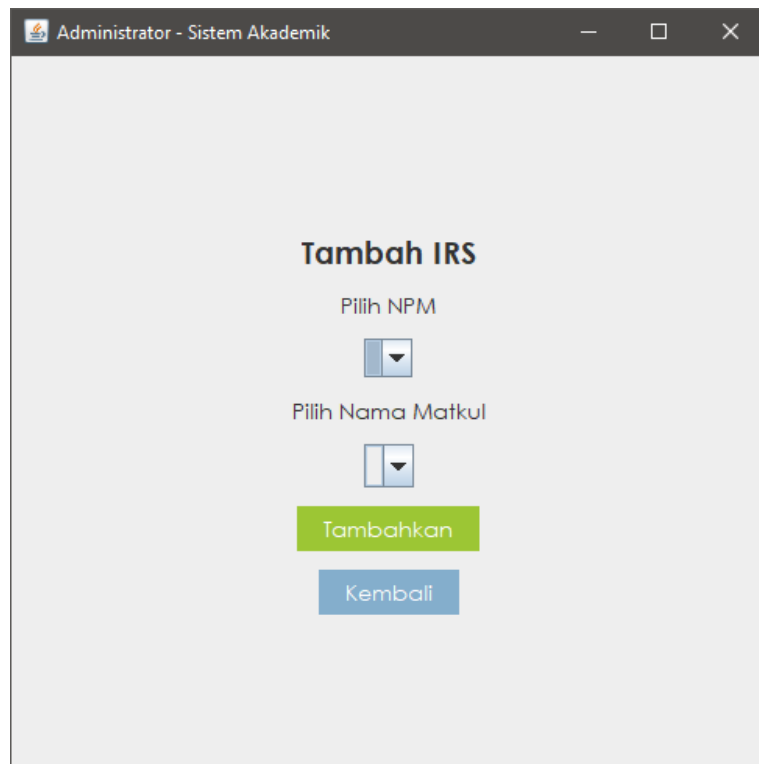


Gambar 12. Dropdown untuk Nama Mata Kuliah

- *Button* Tambahkan untuk menambahkan IRS

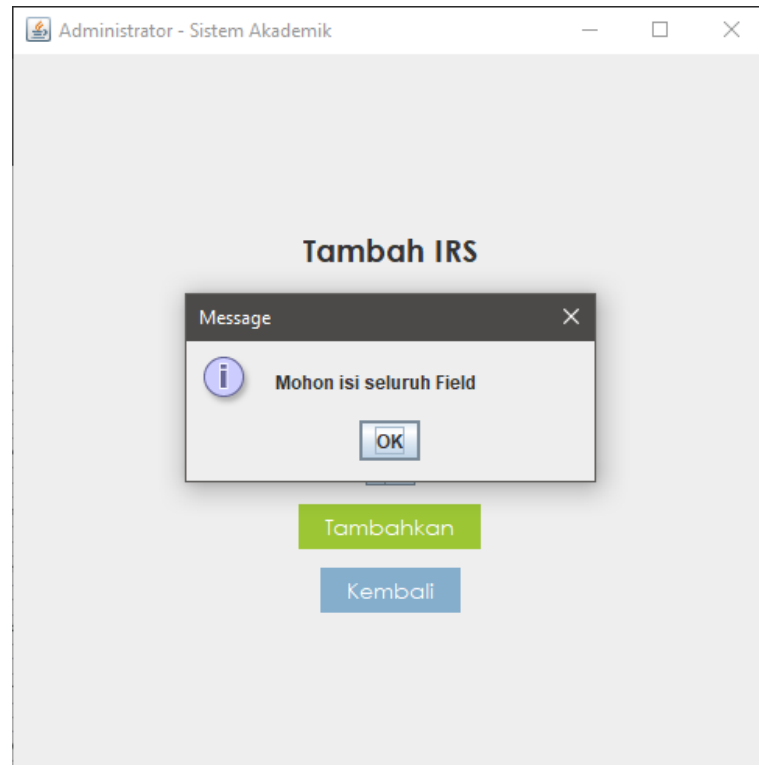
**Catatan:**

- Jika daftar mahasiswa atau mata kuliah masih kosong, maka cukup tampilkan dropdown yang kosong



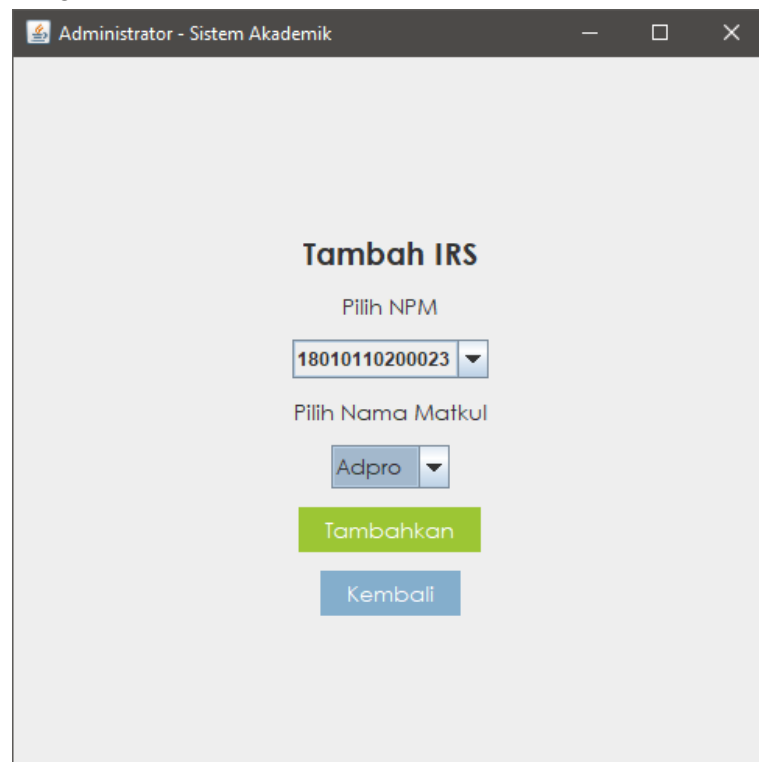
Gambar 13. Daftar Mata Kuliah dan Mahasiswa masih kosong

- Jika ada *field* yang belum terisi, munculkan kotak dialog dengan pesan: **Mohon isi seluruh field**



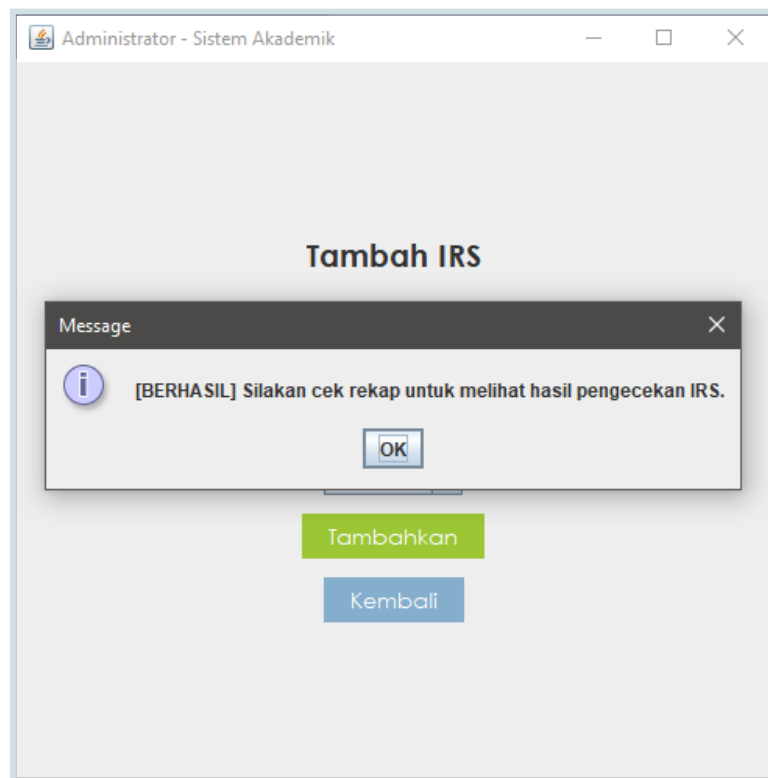
Gambar 14. Terdapat *Field* yang Masih Kosong

- Jika daftar mahasiswa atau mata kuliah sudah terisi, dropdown secara *default* memilih npm terkecil dan nama mata kuliah dengan *alphabet* terkecil.



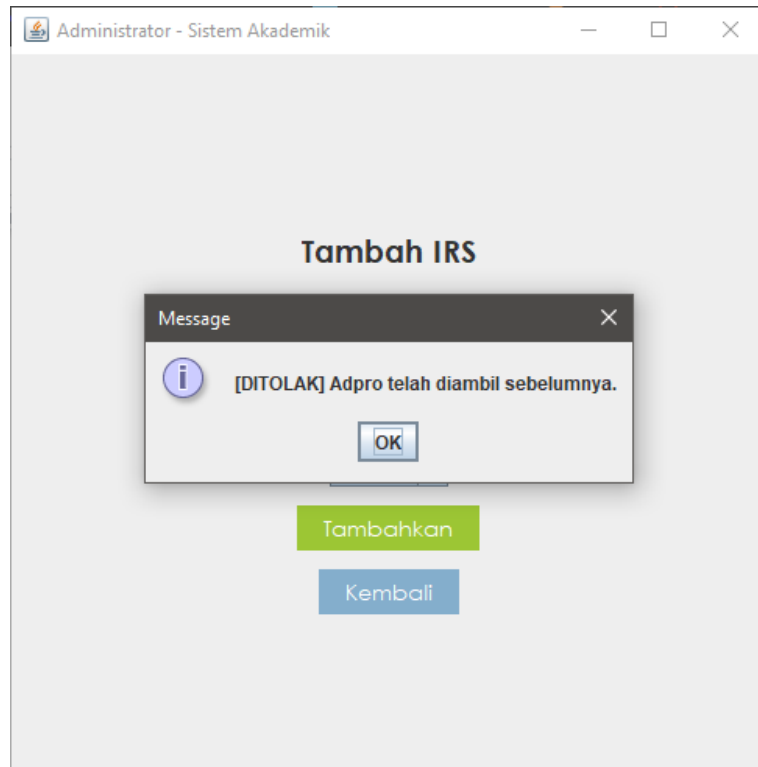
Gambar 15. Tampilan Apabila daftar mata kuliah atau mahasiswa tidak kosong

- iii. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.
- c. *Constraint* yang ada sama dengan TP 2 (tidak perlu Anda *handle* jika sudah menggunakan template karena sudah diatasi pada program *backend*). Namun, Anda diminta untuk menampilkan pesan dari *constraint* tersebut pada sebuah kotak dialog. Berikut adalah ketentuannya:
  - i. Jika IRS berhasil ditambahkan, maka tampilkan kotak dialog berisi pesan: **[BERHASIL] Silakan cek rekap untuk melihat hasil pengecekan IRS.**



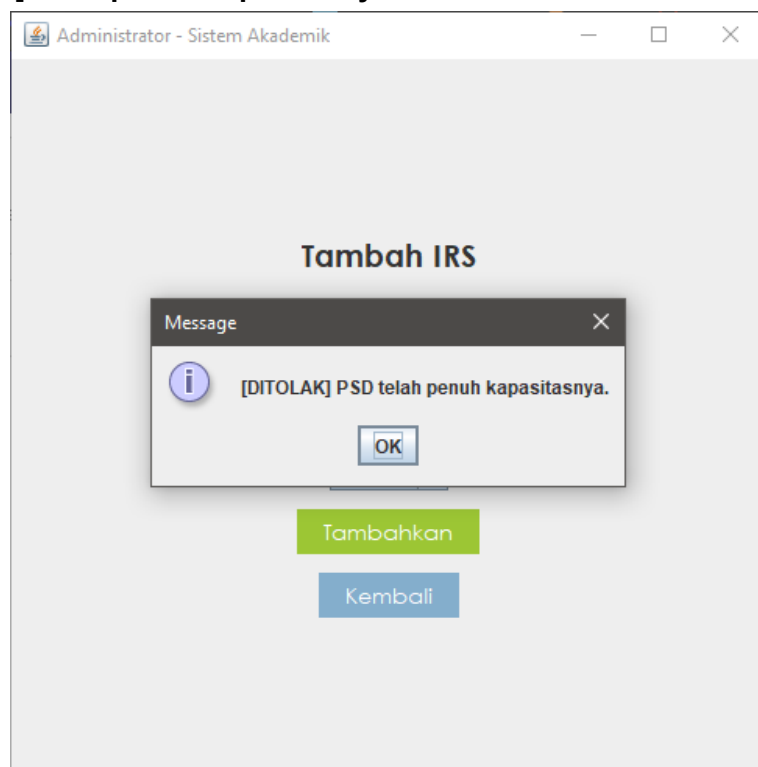
Gambar 16. IRS Berhasil Ditambahkan

- ii. Jika Mata Kuliah pernah diambil sebelumnya, maka tampilkan kotak dialog dengan pesan: **[DITOLAK] [Nama mata kuliah] sudah diambil sebelumnya.**



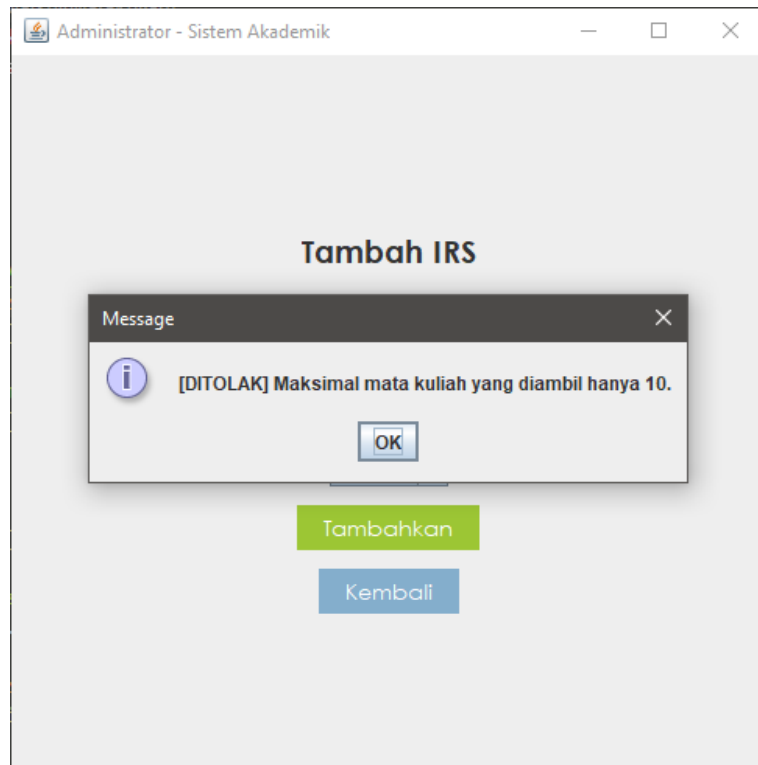
Gambar 17. Penambahan IRS Ditolak karena Mata Kuliah sudah Diambil Sebelumnya

- iii. Jika kapasitas mata kuliah yang ingin diambil sudah penuh, maka tampilkan kotak dialog dengan pesan: **[DITOLAK] [nama mata kuliah] telah penuh kapasitasnya.**



Gambar 18. Penambahan IRS Ditolak karena Mata Kuliah Sudah Penuh Kapasitasnya

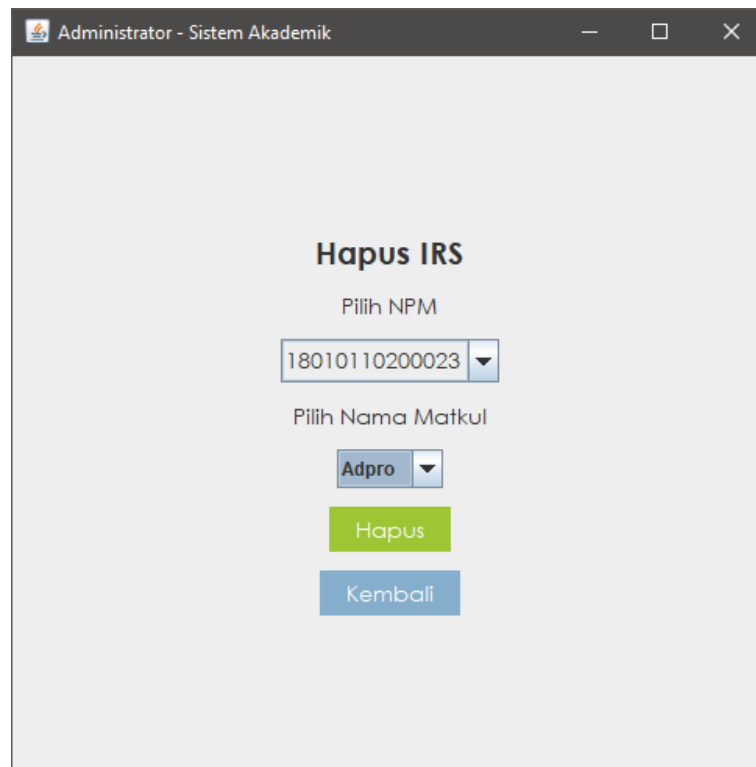
- iv. Jika mahasiswa sudah mengambil lebih dari 10 mata kuliah, maka tampilkan kotak dialog dengan pesan: **[DITOLAK] Maksimal mata kuliah yang diambil hanya 10.**



Gambar 19. Penambahan IRS Ditolak karena Sudah Mengambil Sepuluh Mata Kuliah

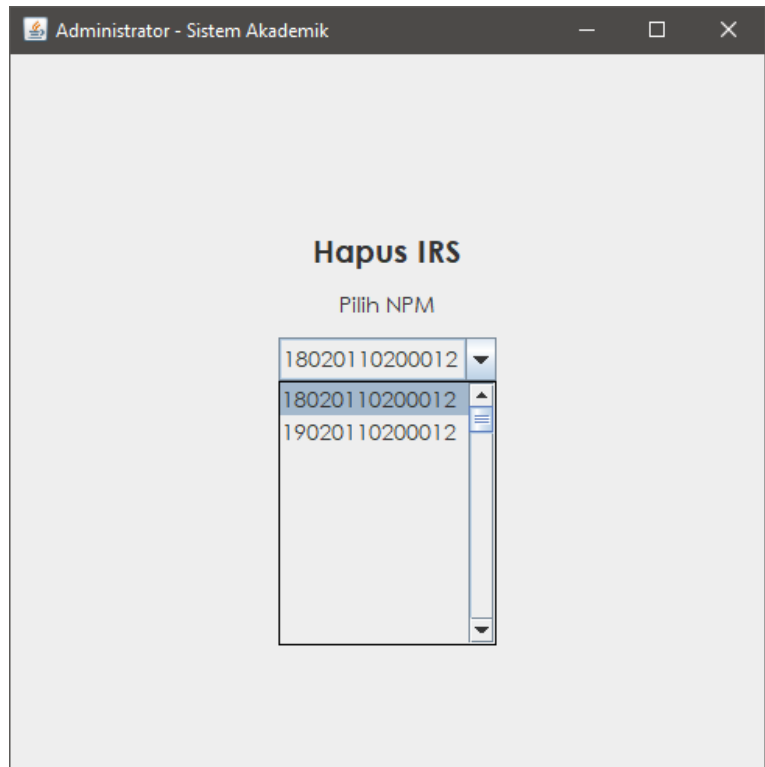


## 6. Class HapusIRSGUI



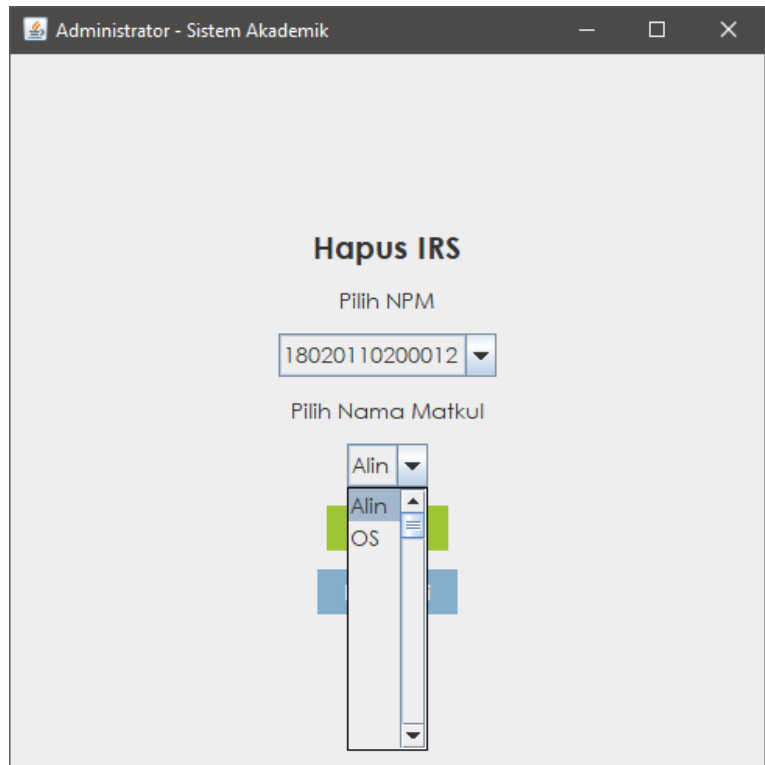
Gambar 20. Halaman Hapus IRS

- a. Merupakan halaman untuk menghapus IRS seorang mahasiswa.
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Hapus IRS**
  - ii. Terdapat form yang berisi:
    - Dropdown untuk memilih NPM Mahasiswa dengan ketentuan:
      - Pilihan pada dropdown ditampilkan secara terurut dari NPM kecil ke besar. Silakan implementasikan **algoritma sorting** agar NPM ditampilkan secara terurut. **Tidak diperkenankan menggunakan built-in method sort dari Java. Anda harus mengimplementasikan sorting secara manual.**



Gambar 21. NPM ditampilkan secara teratur

- Dropdown untuk memilih mata kuliah dengan ketentuan:
  - Pilihan pada dropdown ditampilkan teratur secara *alphabetical order*. Silakan implementasikan **algoritma sorting** agar mata kuliah ditampilkan secara teratur. **Tidak diperkenankan menggunakan built-in method sort dari Java. Anda harus mengimplementasikan sorting secara manual.**
  - Cukup tampilkan daftar mata kuliah yang ada dalam sistem. Jika ada mata kuliah yang sudah ditambahkan mahasiswa, di-handle dengan menampilkan kotak dialog setelah menekan tombol Hapus..

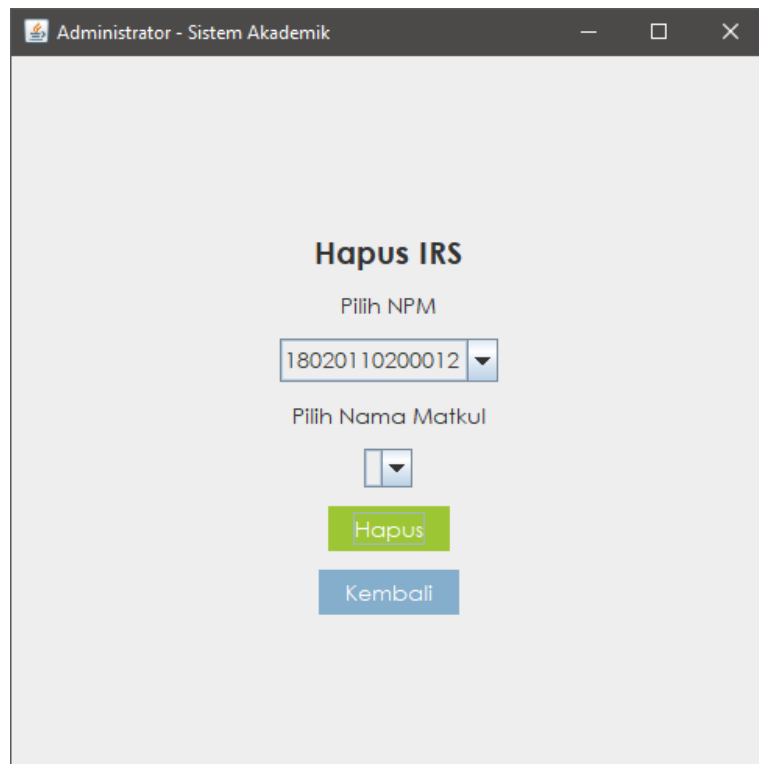


Gambar 22. Dropdown untuk Nama Mata Kuliah

- *Button* Hapus untuk menambahkan IRS

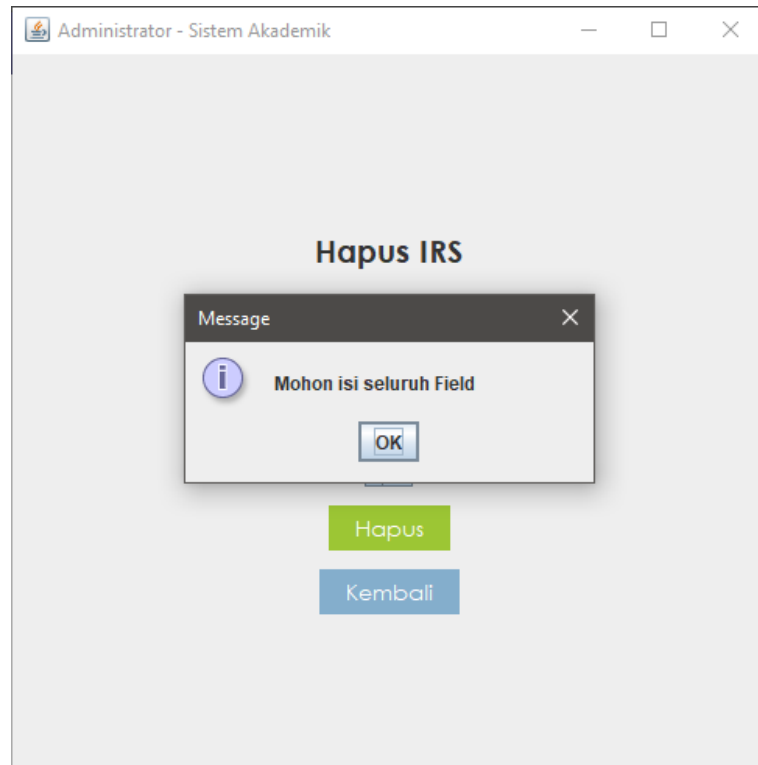
**Catatan:**

- Jika daftar mahasiswa atau mata kuliah masih kosong, maka cukup tampilkan dropdown yang kosong



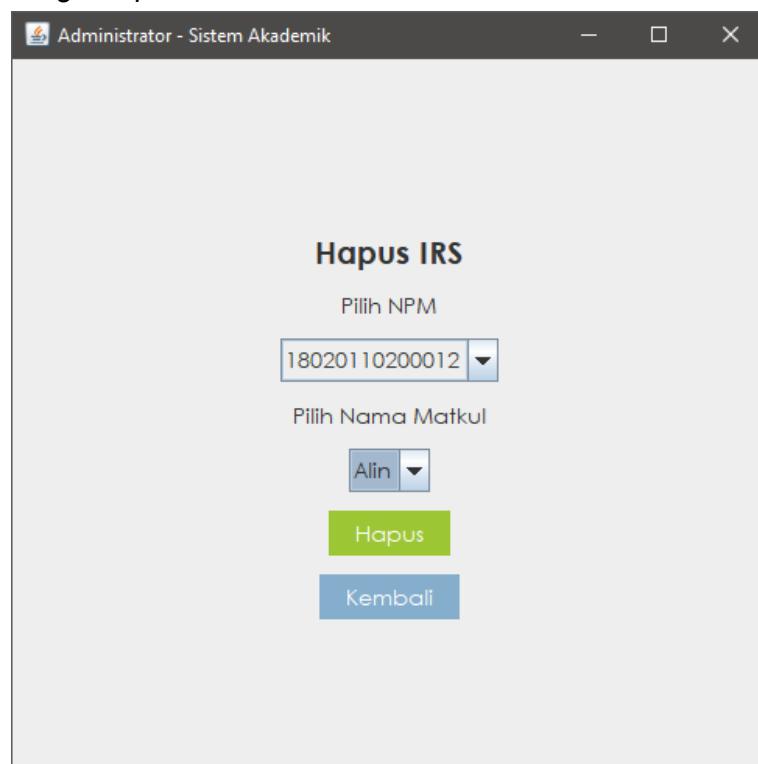
Gambar 23. Daftar Mata Kuliah Masih Kosong

- Jika ada *field* yang belum terisi, munculkan kotak dialog dengan pesan: **Mohon isi seluruh field**



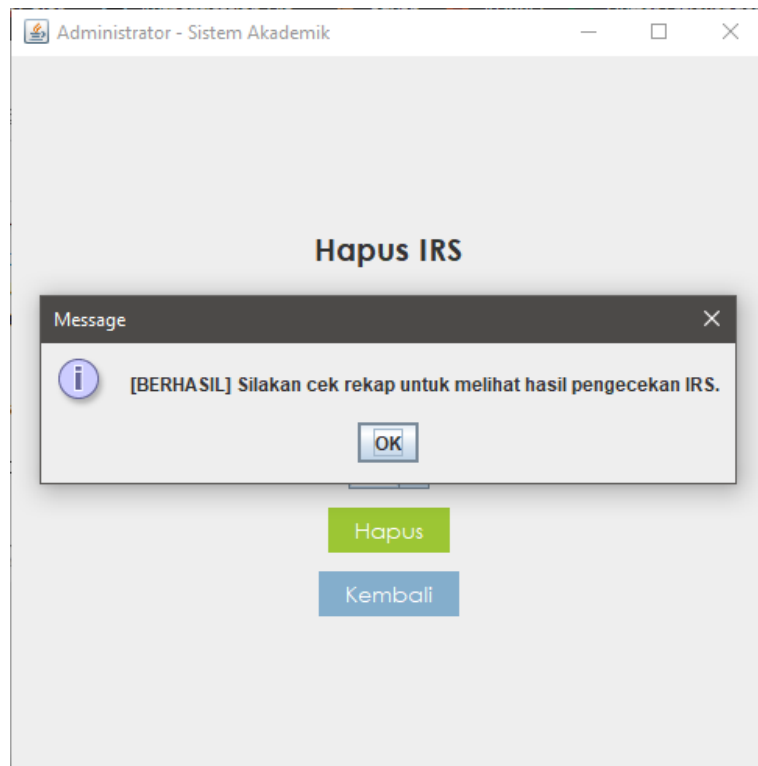
Gambar 24. Terdapat *Field* yang Masih Kosong

- Jika daftar mahasiswa atau mata kuliah sudah terisi, dropdown secara *default* memilih npm terkecil dan nama mata kuliah dengan *alphabet* terkecil.



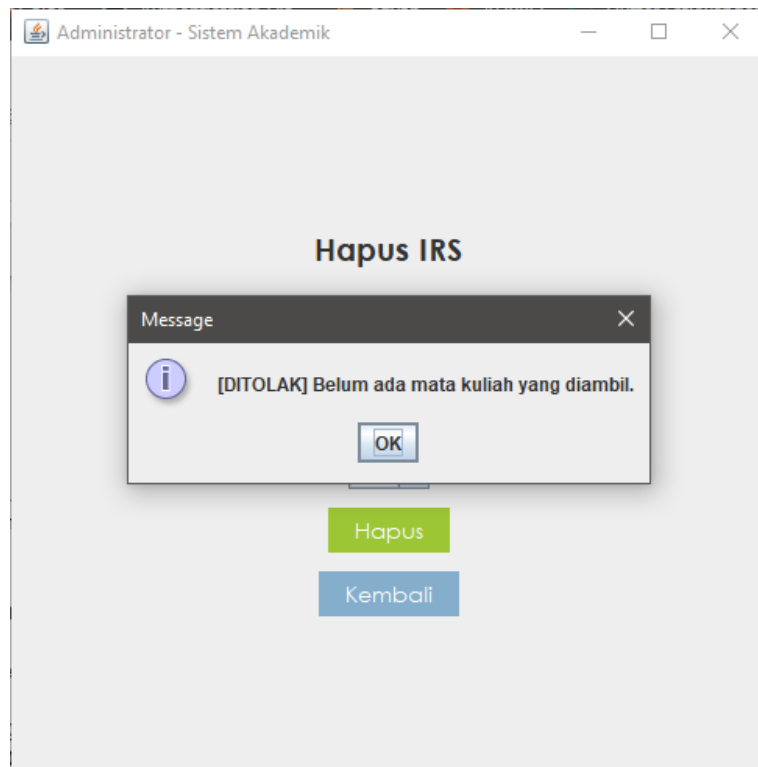
Gambar 25. Tampilan Apabila daftar mata kuliah atau mahasiswa tidak kosong

- iii. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.
- c. *Constraint* yang ada sama dengan TP 2 (tidak perlu Anda *handle* jika sudah menggunakan template karena sudah diatasi pada program *backend*). Namun, Anda diminta untuk menampilkan pesan dari *constraint* tersebut pada sebuah kotak dialog. Berikut adalah ketentuannya:
  - i. Jika IRS berhasil dihapus, maka tampilkan kotak dialog berisi pesan: **[BERHASIL] Silakan cek rekap untuk melihat hasil pengecekan IRS.**



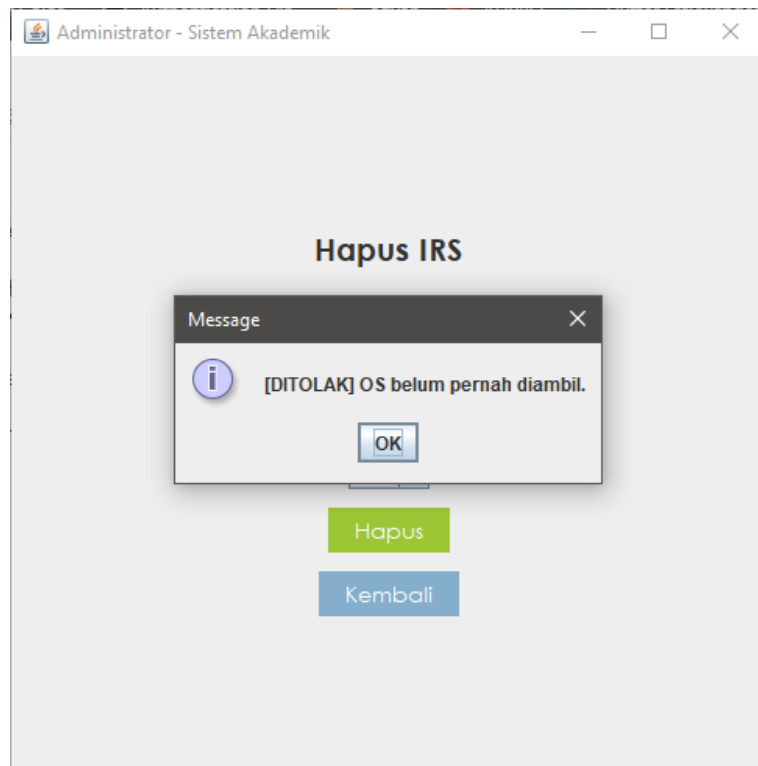
Gambar 26. IRS Berhasil Dihapus

- d. Jika belum ada Mata Kuliah yang diambil, maka tampilkan kotak dialog dengan pesan: **[DITOLAK] belum ada mata kuliah yang diambil.**



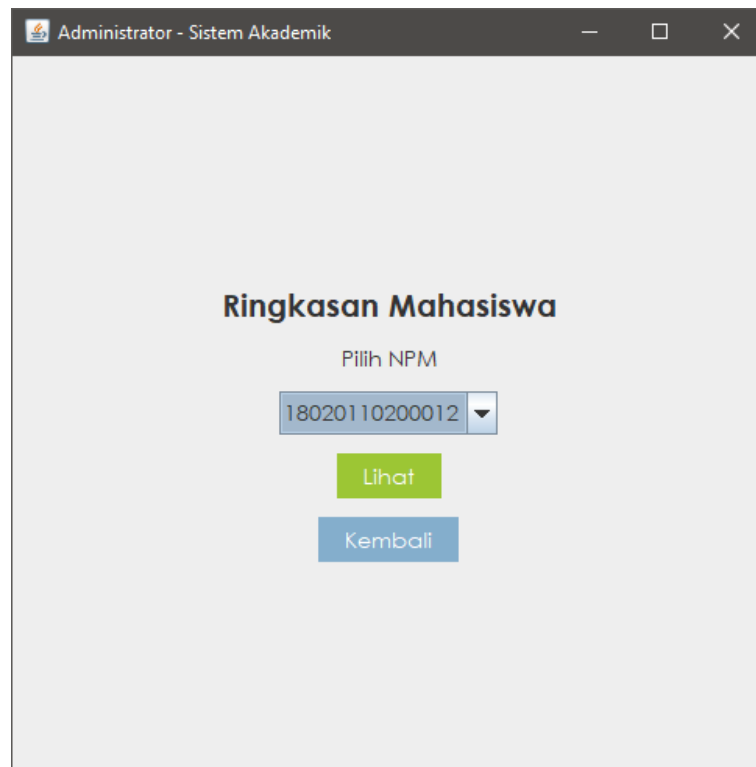
Gambar 27. Belum Ada Mata Kuliah yang Diambil

- e. Jika mata kuliah yang ingin dihapus belum diambil, maka tampilkan kotak dialog dengan pesan: **[DITOLAK] [nama mata kuliah] belum diambil.**



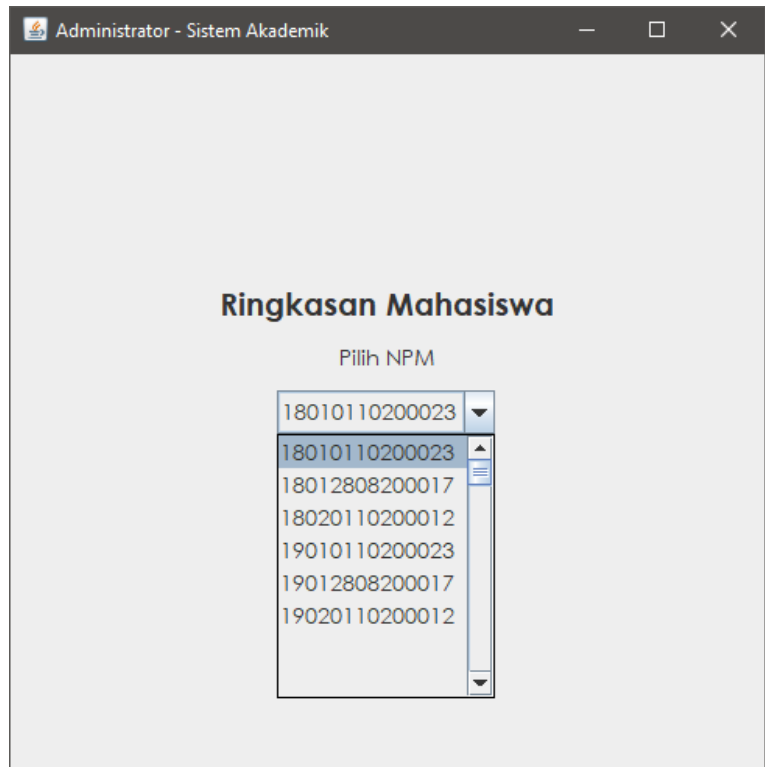
Gambar 28. Mata Kuliah Belum Diambil

## 7. Class RingkasanMahasiswaGUI



Gambar 29. Halaman Ringkasan Mahasiswa

- a. Merupakan halaman untuk memasukan NPM mahasiswa yang ingin dilihat ringkasannya
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Ringkasan Mahasiswa**
  - ii. Terdapat form yang berisi:
    - Dropdown untuk memilih NPM Mahasiswa dengan ketentuan:
      - Pilihan pada dropdown ditampilkan secara terurut dari NPM kecil ke besar. Silakan implementasikan **algoritma *sorting*** agar NPM ditampilkan secara terurut. **Tidak diperkenankan menggunakan built-in method `sort` dari Java. Anda harus mengimplementasikan *sorting* secara manual.**

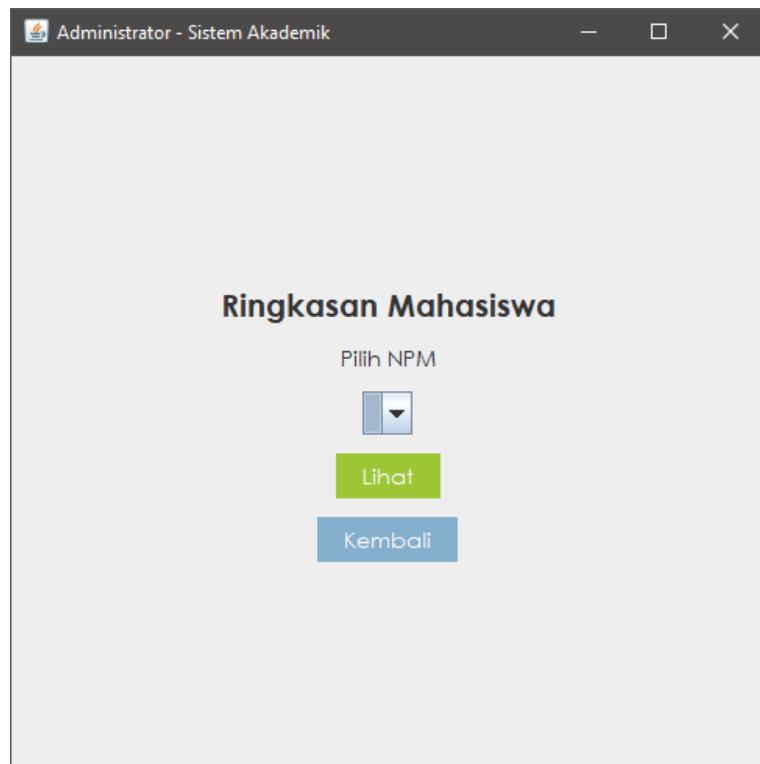


Gambar 30. Dropdown NPM Terurut dari Kecil ke Besar

- *Button Lihat* yang akan mengarahkan ke halaman **Detail Ringkasan Mahasiswa**.

**Catatan:**

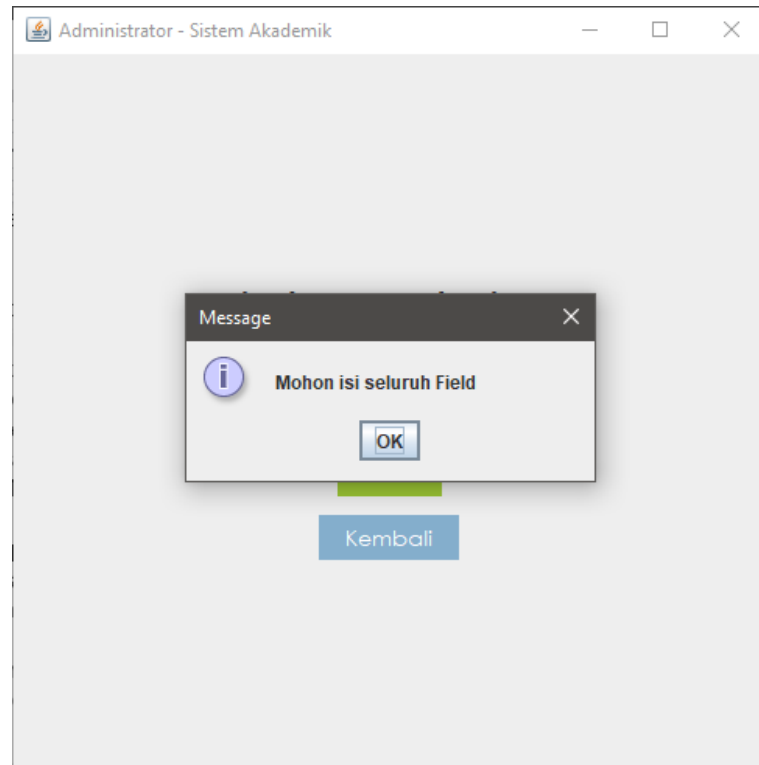
- Jika daftar mahasiswa masih kosong, maka cukup tampilkan dropdown yang kosong



Gambar 31. Dropdown NPM Kosong



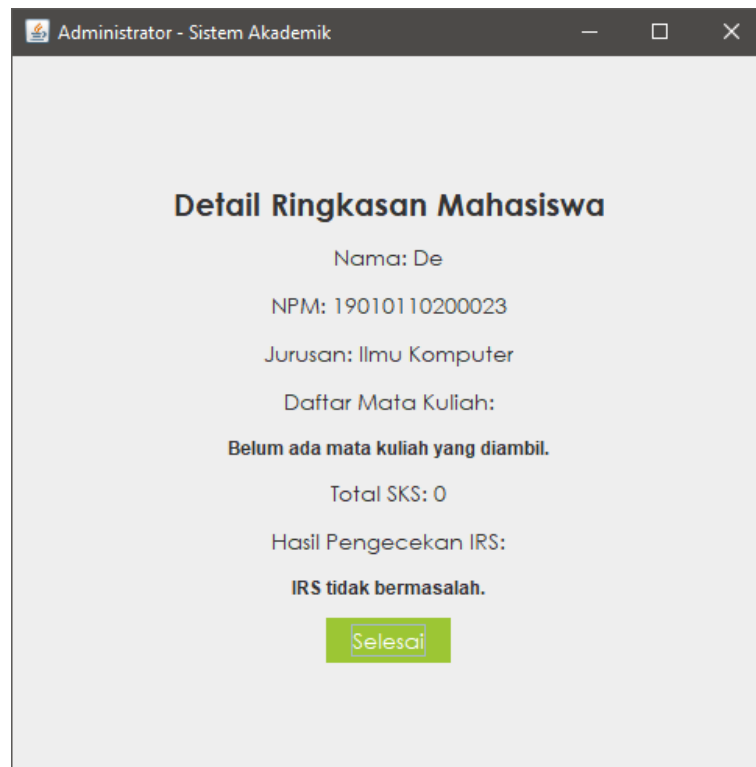
- Seluruh field harus terisi, jika ada field yang belum terisi maka tampilkan kotak dialog dengan pesan: **Mohon isi seluruh field**



Gambar 32. Terdapat field yang masih kosong

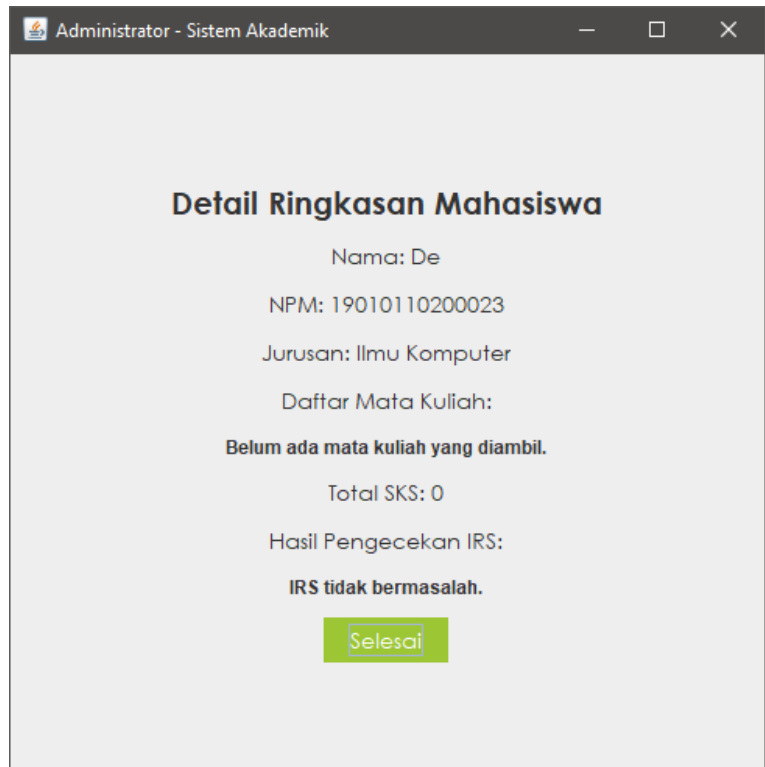
- iii. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.

## 8. Class DetailRingkasanMahasiswaGUI



Gambar 33. Detail Ringkasan Mahasiswa

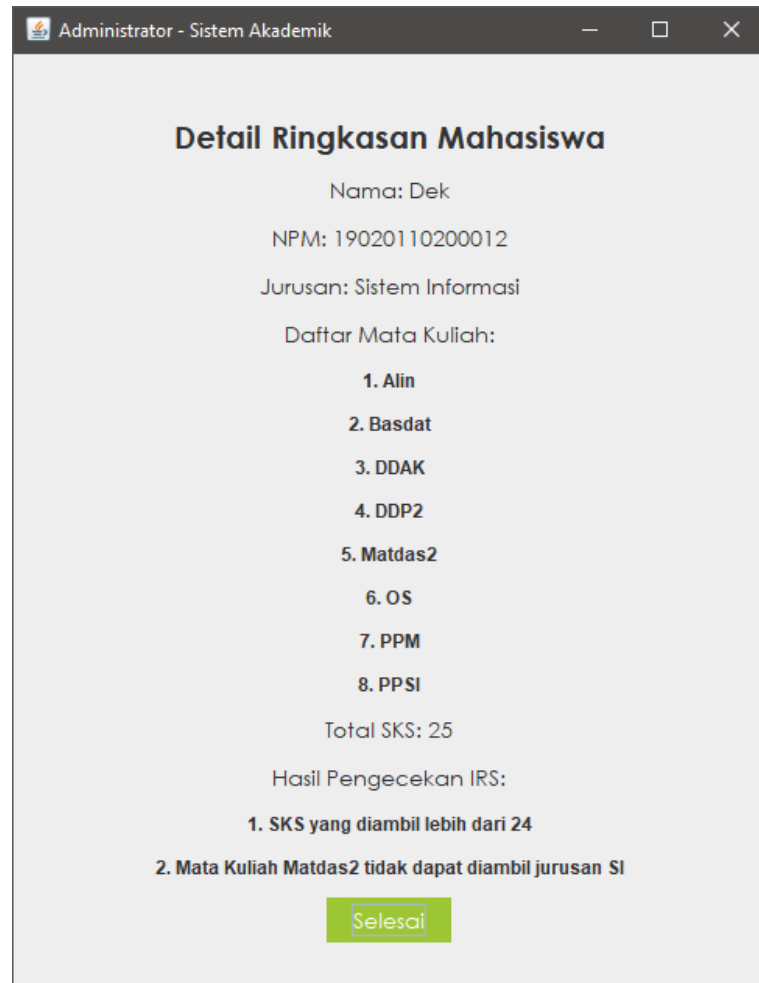
- a. Merupakan halaman untuk menampilkan ringkasan mahasiswa yang NPM-nya dimasukkan pada halaman Ringkasan Mahasiswa.
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Detail Ringkasan Mahasiswa**
  - ii. Terdapat informasi mengenai:
    - Nama mahasiswa
    - NPM
    - Jurusan
    - Daftar Mata Kuliah
      - Jika belum ada mata kuliah yang diambil, tampilkan: **Belum ada mata kuliah yang diambil**



Gambar 34. Belum Ada Mata Kuliah yang Diambil

- Total SKS
- Hasil Pengecekan IRS
  - Jika IRS tidak bermasalah, tampilkan: **IRS tidak bermasalah.**
  - Jika SKS yang diambil melebihi 24 sks, masukan ke daftar hasil pengecekan bahwa: **SKS yang diambil lebih dari 24.**
  - Jika terdapat mata kuliah yang diambil tidak sesuai jurusan (Mata kuliah dengan kode IK hanya dapat diambil mahasiswa Ilmu Komputer, mata kuliah dengan kode SI hanya dapat diambil mahasiswa Sistem Informasi, mata kuliah CS dapat diambil mahasiswa Ilmu Komputer maupun Sistem Informasi), masukan ke daftar hasil pengecekan bahwa: **Mata kuliah [nama mata kuliah] tidak dapat diambil jurusan [kode jurusan mahasiswa]**

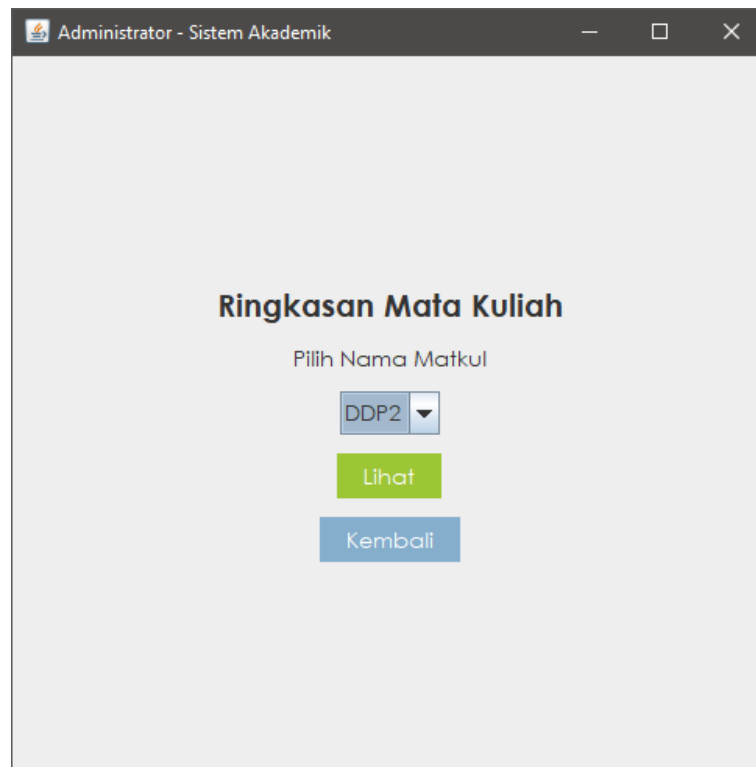
**Catatan:** Poin ke-2 dan ke-3 sudah di-handle pada template backend sehingga Anda hanya perlu menampilkannya.



Gambar 35. Terdapat Beberapa Masalah IRS

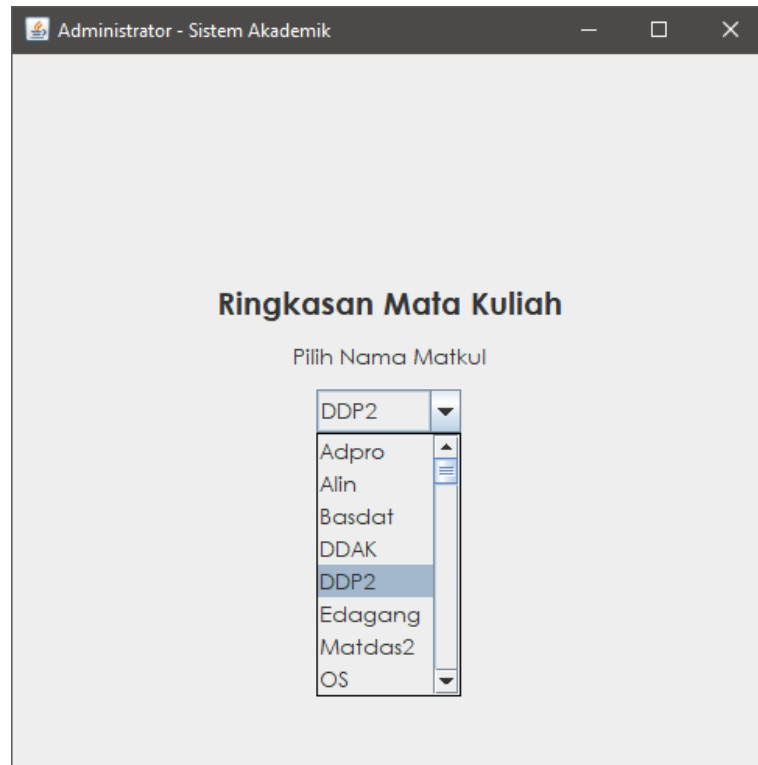
- iii. Terdapat tombol **Selesai** yang akan mengarahkan ke halaman **Home**

## 9. Class RingkasanMataKuliahGUI



Gambar 36. Halaman Ringkasan Mata Kuliah

- a. Merupakan halaman untuk memasukkan nama mata kuliah yang ingin dilihat ringkasannya
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Ringkasan Mata Kuliah**
  - ii. Terdapat form yang berisi:
    - Dropdown untuk memilih mata kuliah dengan ketentuan:
      - Pilihan pada dropdown ditampilkan ditampilkan terurut secara *alphabetical order*. Silakan implementasikan **algoritma sorting** agar NPM ditampilkan secara terurut. **Tidak diperkenankan menggunakan built-in method sort dari Java. Anda harus mengimplementasikan sorting secara manual.**

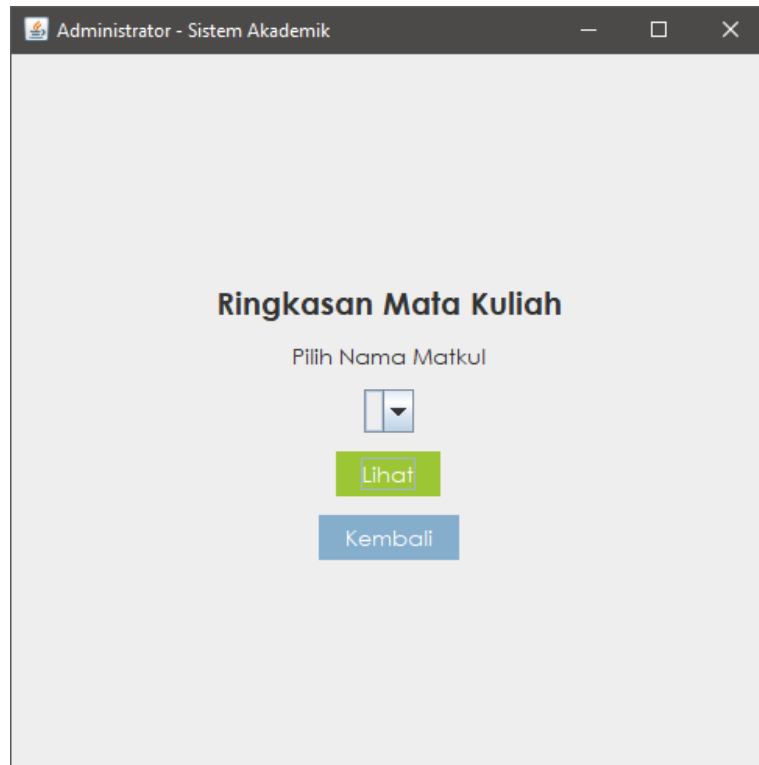


Gambar 37. *Dropdown Nama Mata Kuliah Ditampilkan Secara Alphabetical Order*

- **Button Lihat** yang akan mengarahkan ke halaman **Detail Ringkasan Mahasiswa**.

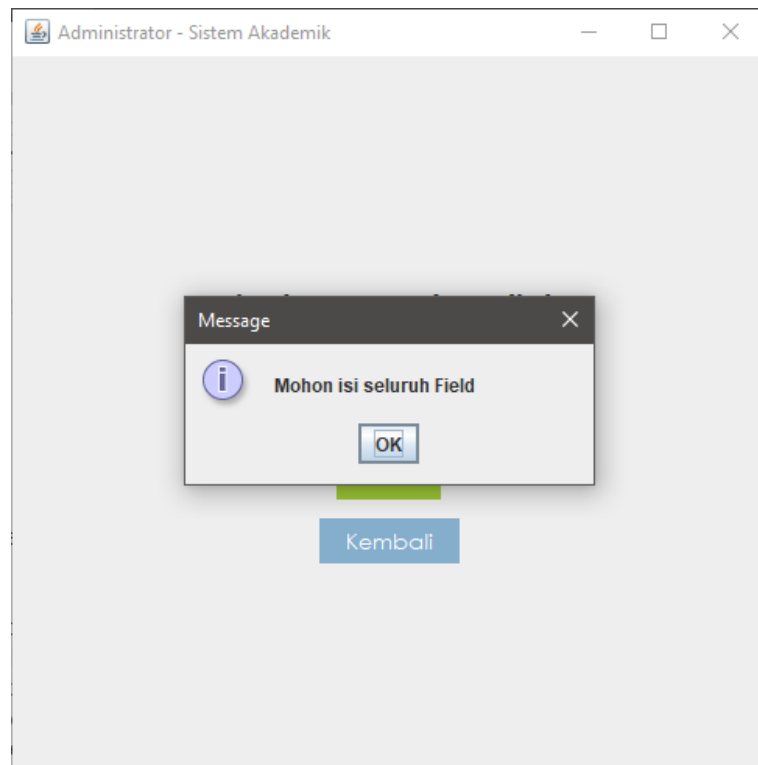
**Catatan:**

- Jika daftar mata kuliah masih kosong, maka cukup tampilkan dropdown yang kosong



Gambar 38. Dropdown Nama Mata Kuliah Kosong

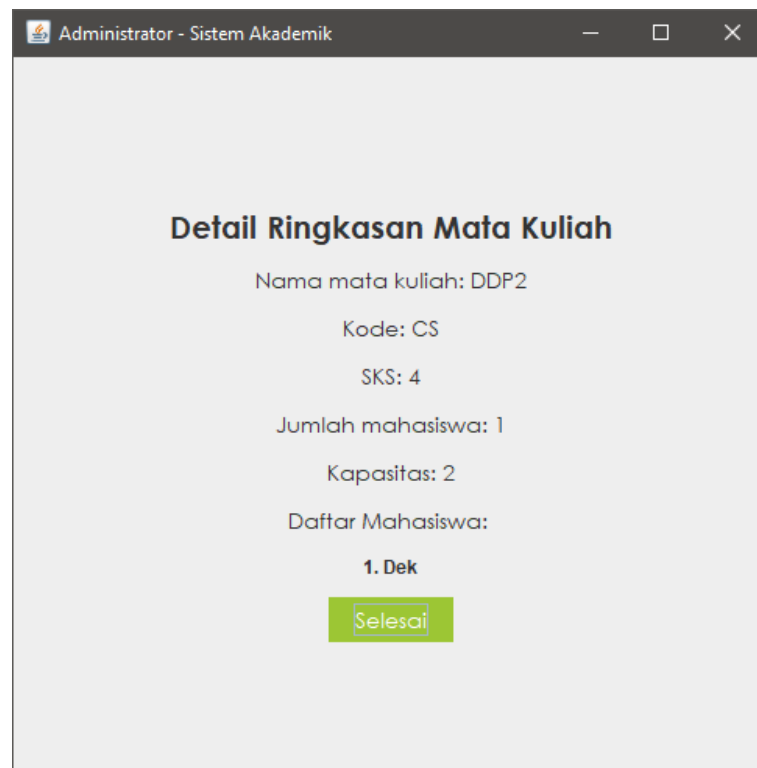
- Seluruh field harus terisi, jika ada field yang belum terisi maka tampilkan kotak dialog dengan pesan: **Mohon isi seluruh field**



Gambar 39. Terdapat field yang masih kosong

- c. Terdapat *button* **Kembali** yang akan mengarahkan ke halaman **Home**.

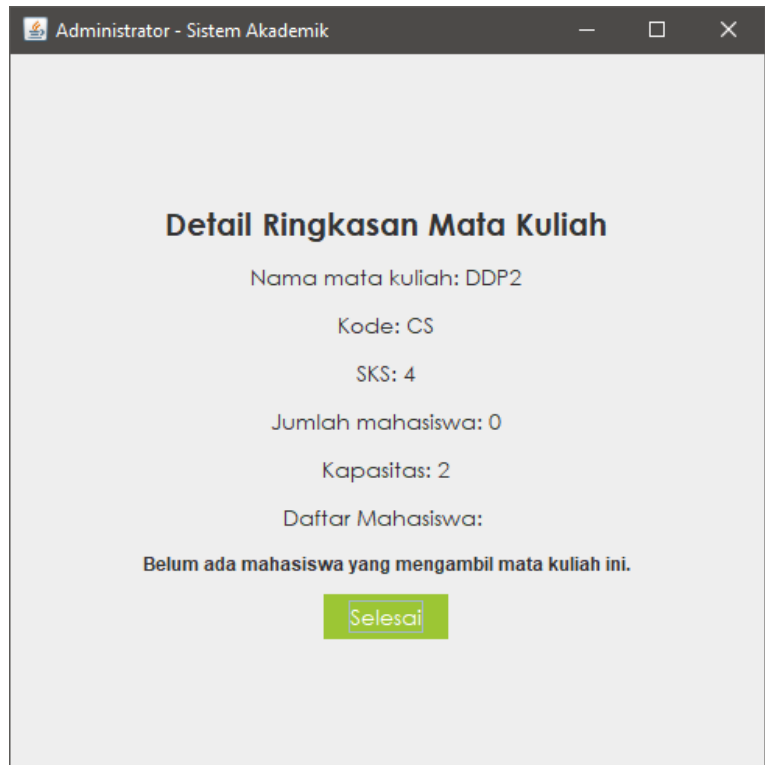
## 10. Class DetailRingkasanMataKuliahGUI



Gambar 40. Detail Ringkasan Mata Kuliah

- a. Merupakan halaman untuk menampilkan ringkasan mata kuliah yang namanya dimasukkan pada halaman Ringkasan Mata Kuliah.
- b. Ketentuan GUI:
  - i. Terdapat judul halaman: **Detail Ringkasan Mata Kuliah**
  - ii. Terdapat informasi mengenai:
    - Nama mata kuliah
    - Kode mata kuliah
    - SKS
    - Jumlah Mahasiswa yang mengambil
    - Kapasitas
    - Daftar Mahasiswa
      - Jika belum ada mata kuliah yang diambil, tampilkan: **Belum ada mahasiswa yang mengambil mata kuliah ini**





Gambar 41. Belum Ada Mahasiswa yang Mengambil Mata Kuliah

- iii. Terdapat tombol **Selesai** yang akan mengarahkan ke halaman **Home**

## Komponen Penilaian

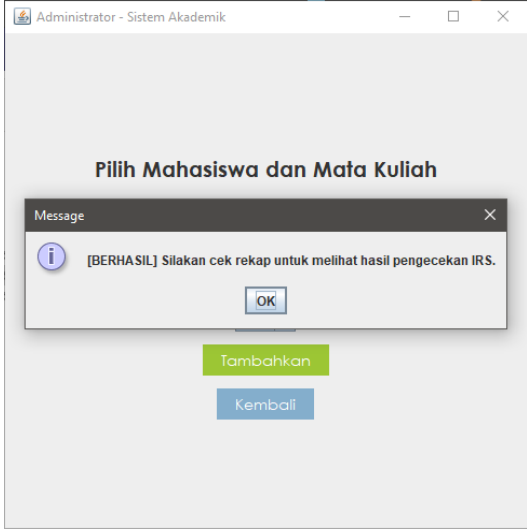
- **40%** Kesesuaian Flow GUI dengan penjelasan soal
- **10%** Implementasi Algoritma *Sorting*
- **10%** Estetika GUI
- **40%** Demonstrasi Program ke Asisten

## Revisi

- Revisi 1 (22 May 2021)
  - Komponen Penilaian:

Sebelum	Sesudah
<ul style="list-style-type: none"> <li>• <b>45%</b> Kesesuaian Flow GUI dengan penjelasan soal</li> <li>• <b>5%</b> Implementasi Algoritma <i>Sorting</i></li> </ul>	<ul style="list-style-type: none"> <li>• <b>40%</b> Kesesuaian Flow GUI dengan penjelasan soal</li> <li>• <b>10%</b> Implementasi Algoritma <i>Sorting</i></li> </ul>

- Revisi 2 (28 May 2021)

Sebelum	Sesudah
<p>Judul pada Tambah IRS saat berhasil melakukan penambahan IRS: <b>Pilih Mahasiswa dan Mata Kuliah</b></p> 	<p>Judul pada Tambah IRS saat berhasil melakukan penambahan IRS: <b>Tambah IRS</b></p> 