

# Dasar-Dasar Pemrograman 2

## TP 04

### GUI, Event-driven programming



FAKULTAS  
ILMU  
KOMPUTER

Deadline:

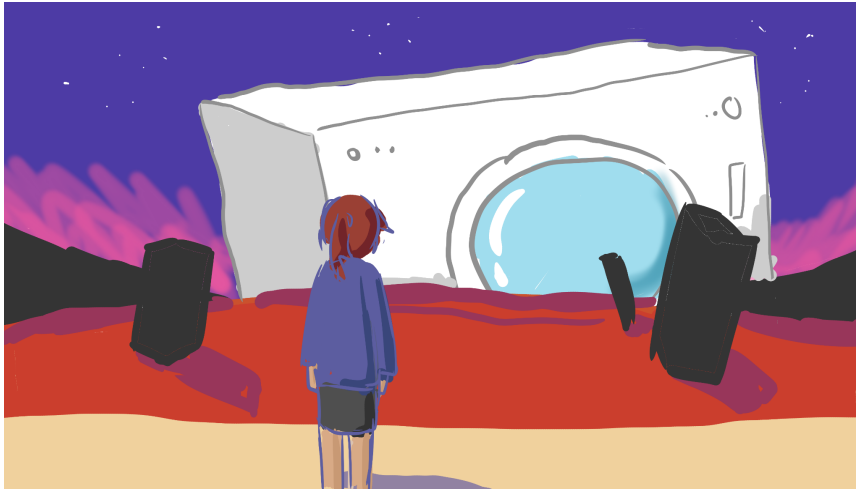
19 Mei 2023 (23.55 waktu Scele)

---

#### Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1	08-05-2023; 12.00 WIB	Rilis Pertama	-
2	10-05-2023; 20.58 WIB	edit requirement untuk employee	yellow
3	13-05-2023; 19.24 WIB	Typo panjang list pada snippet code	blue
4			

## CuciCuci 4: Goodbye, Dek Depe!



CuciCuci sekarang sudah sukses! dan kamu sudah siap untuk berpaling dari CuciCuci untuk mengejar mimpi kamu di matkul lain. Apakah kamu setelah ini akan mengCP di SDA? Memperdalam ilmu ~~wengdev~~ web developer bersama PBP? Mengambil langkah pertama untuk menjadi seorang **[HACKERMAN]** di Jarkom? atau berubah the next corporate Nostradamus pada KASDAD?

Apapun jalan yang kamu pilih di masa depan Dek Depe, Dek Depram, employee CuciCuci dan tim asdos akan mendoakan kesuksesan dan juga kebahagiaan kalian. Namun untuk yang terakhir kalinya kamu akan membuat program GUI sebagai hadiah selamat tinggal.

## Goodbye, Dek Depe!

Nothing more to see here.



Terima kasih kepada kawan-kawan yang telah menyumbangkan karyanya untuk TP ini!

## Ketentuan

Feel free to make any changes to the template! Be creative and have fun!

Tidak usah 100% sama, asalkan fungsionalitas dasarnya masih ada.

### Class MainFrame

- Sebuah class utama yang akan memuat frame.
- Constructor MainFrame bersifat private.
- Class ini memuat semua panel dari class-class yang lain.
- Class ini juga memiliki akses pada LoginManager, MemberSystem, dan EmployeeSystem.
- Terdapat beberapa method abstract yang akan kamu implementasikan pada class ini:

Method	Return Type	Deskripsi
public static MainFrame getInstance()	MainFrame	<ul style="list-style-type: none"><li>• Method untuk mendapatkan instance MainFrame.</li><li>• Instance Class MainFrame harus diambil melalui method ini agar menjamin hanya terdapat satu Frame pada program.</li></ul>
public void navigateTo(String page)	void	<ul style="list-style-type: none"><li>• Method untuk pergi ke panel sesuai dengan yang diberikan pada param.</li><li>• Param:<ul style="list-style-type: none"><li>◦ <b>page</b>: key dari halaman yang diinginkan.</li></ul></li></ul>
public boolean login(String id, String password)	boolean	<ul style="list-style-type: none"><li>• Method untuk login pada sistem.</li><li>• Jika gagal login akan mengembalikan boolean false dan jika berhasil login:<ul style="list-style-type: none"><li>◦ return boolean true</li><li>◦ menampilkan halaman yang sesuai</li></ul></li><li>• Param:<ul style="list-style-type: none"><li>◦ id: ID dari pengguna</li><li>◦ password: password dari pengguna</li></ul></li></ul>
public void logout()	void	<ul style="list-style-type: none"><li>• Method untuk logout dari sistem, kemudian menampilkan halaman Home.</li></ul>

- Tidak wajib, tapi constructor MainFrame kamu bisa ditambahkan dua Employee baru. Silakan implementasikan .addEmployee(Employee[]) pada class EmployeeSystem menggunakan snippet code di bawah ini dan tambahkan juga namamu sebagai employee CuciCuci!

```
public void addEmployee(Employee[] employees) {
    Member[] result = new Member[employees.length + memberList.length];

    System.arraycopy(memberList, 0, result, 0, memberList.length);
    System.arraycopy(employees, 0, result, memberList.length,
employees.length);

    memberList = result;
}
```

## Package gui

### Class HomeGUI

- Di paling atas akan ada sambutan.
- Di paling bawah akan menampilkan hari ini
- Terdapat 3 Button:
  - Login : Akan membawa user pada halaman Login
  - Register: Akan membawa user pada halaman Register
  - Next Day:
    - Akan menskip ke hari selanjutnya
    - menampilkan dialog box yang menandakan hari berganti
    - Mengupdate label hari di bawah

- Berikut adalah fungsionalitas minimal untuk HomeGUI

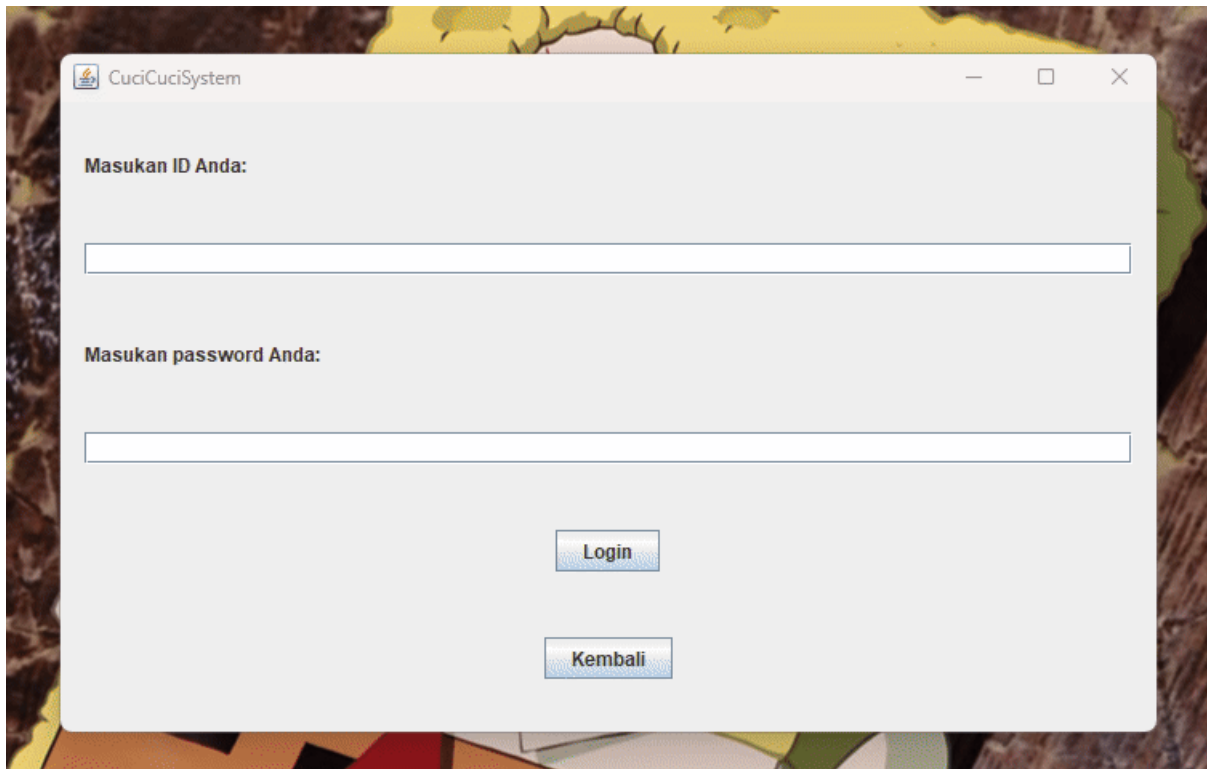


- Silakan menambahkan method lain jika diperlukan.

## Class LoginGUI

- Mewarisi (*inherits*) class JPanel.
- Class ini terdiri atas:
  - TextField sebagai input ID
  - PasswordField sebagai input password
- Setelah menekan tombol login akan ada dua skenario yang mungkin terjadi:
  - Akan langsung mengarahkan user ke halaman yang benar jika ID dan passwordnya valid (Login berhasil).
  - User akan diberikan sebuah notifikasi berupa dialogue box (Login gagal).
- TextField & PasswordField akan kembali bersih jika user keluar dari halaman login.
- Terdapat tombol kembali yang jika ditekan akan mengembalikan user ke halaman Home.

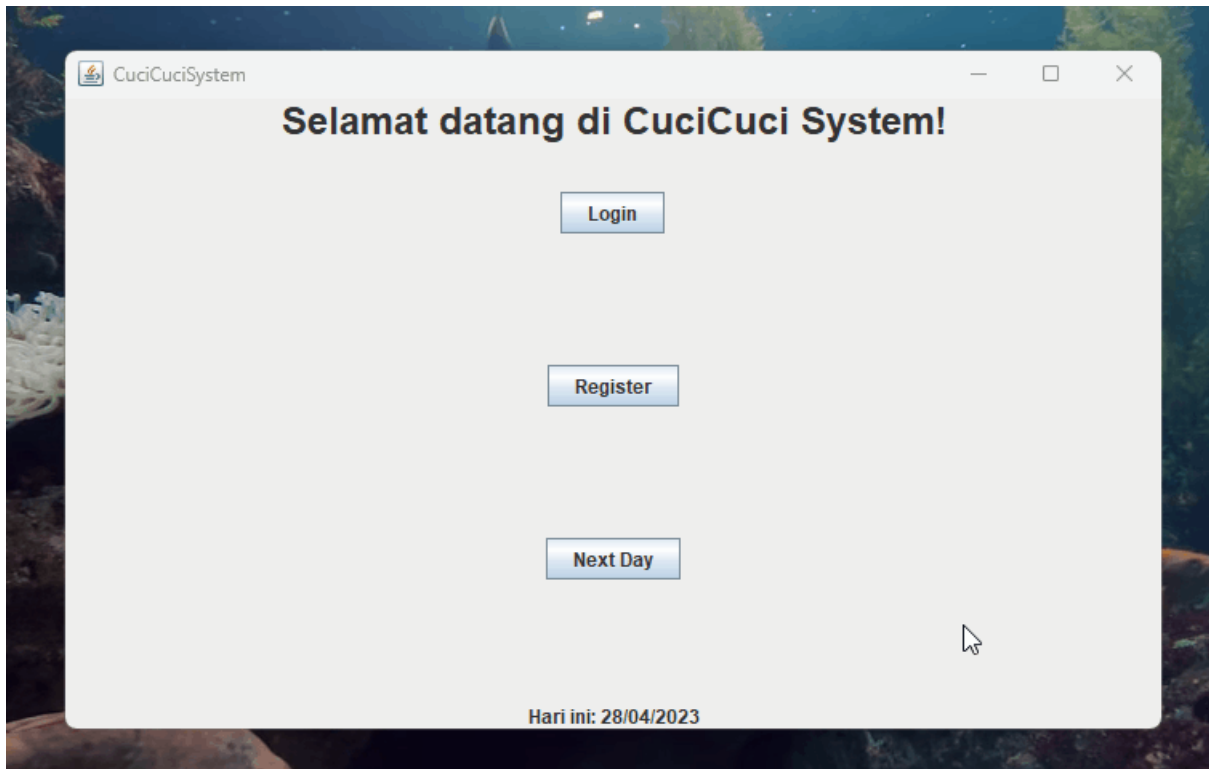
- Berikut adalah contoh fungsionalitas untuk LoginGUI



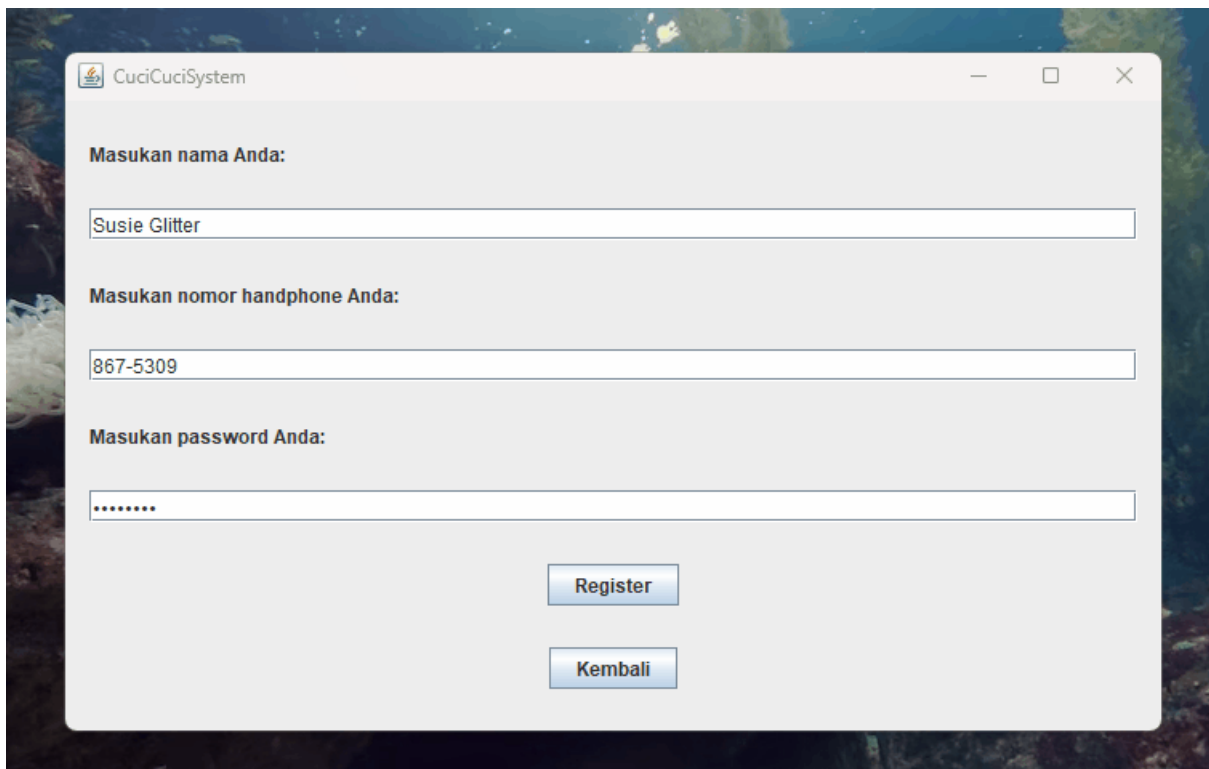
- Silakan menambahkan method lain jika diperlukan.

## Class RegisterGUI

- Mewarisi (*inherits*) class JPanel.
- Class ini terdiri atas:
  - TextField sebagai input Nama
  - TextField sebagai input Nomor Handphone
    - Kamu harus melakukan pengecekan input apabila user memasukkan hal selain angka pada field ini
  - PasswordField sebagai input password
- Jika tombol register ditekan, ada beberapa hal yang dapat terjadi:
  - Jika berhasil, user akan diberikan sebuah notifikasi pada dialogue box berisi ID miliknya.
  - Register gagal karena input nomor handphone tidak sesuai, akan diberikan dialogue box dan kemudian field nomor handphone akan kosong.
  - Register gagal karena member sudah ada, user akan diberikan sebuah notifikasi berupa dialogue box, kemudian akan kembali ke Home.
- Semua TextField akan kembali bersih jika user keluar dari halaman register.
- Berikut adalah contoh interaksi minimal yang ada pada panel ini.



*Contoh kasus register berhasil, gagal karena nomor HP, gagal karena field kosong*



*Contoh kasus register gagal jika juga sudah pernah daftar.*

- Silakan menambahkan method lain jika diperlukan.

## Package gui.member

### Abstract Class AbstractMemberGUI

- Sebuah abstract class yang akan diwarisi oleh 2 concrete class.
- abstract class ini bertujuan memberikan abstraksi yang nantinya akan diimplementasikan secara spesifik pada subclassnya.
- Terdapat label yang menyambut user dan juga footer yang menunjukkan id user.
- Class Member memiliki properti core sebagai berikut:

Properti	Type	Deskripsi
systemCLI	systemCLI	SystemCLI dengan tujuan untuk dapat mengakses Class yang sudah dibuat pada TP3
loggedInMember	Member	Member yang telah login pada sistem.

- Semua properti di atas hanya dapat diakses secara internal dari class yang mewarisi Class AbstractMemberGUI, kecuali SystemCLI yang secara default memiliki akses private, namun **jika kamu membutuhkannya, bisa kamu ubah menjadi protected.**
- Terdapat dua method abstract yang akan kamu implementasikan pada class ini:

Method	Return Type	Deskripsi
createButtons()	JButton[]	<ul style="list-style-type: none"><li>• Method ini mensupply buttons apa saja yang akan dimuat pada panel ini.</li><li>• Button yang disediakan method ini BELUM memiliki ActionListener.</li></ul>
createActionListeners()	ActionListener[]	<ul style="list-style-type: none"><li>• Method ini mensupply ActionListener korespondensi dengan button yang dibuat createButtons().</li></ul>

- Terdapat tiga method biasa. Dua sudah kami implementasikan, dan satu lagi harus kamu implementasikan sendiri. *Feel free to make any change to the template.*

Method	Return Type	Deskripsi
initializeButtons()	JPanel	<ul style="list-style-type: none"><li>• Membuat panel button yang akan ditampilkan pada Panel ini.</li><li>• Buttons dan ActionListener akan disupply oleh method createButtons() &amp; createActionListeners() respectively.</li></ul>



		<ul style="list-style-type: none"> <li>• Feel free to make any changes. Be creative and have fun!</li> </ul>
login(String id, String password)	boolean	<ul style="list-style-type: none"> <li>• Method untuk login pada panel.</li> <li>• Method ini akan melakukan pengecekan apakah ID dan password yang telah diberikan dapat login pada panel ini.</li> <li>• Jika bisa, member akan login pada panel ini, method akan: <ul style="list-style-type: none"> <li>◦ mengupdate Welcome &amp; LoggedInAs label</li> <li>◦ mengupdate LoggedInMember sesuai dengan instance pemilik ID dan password.</li> </ul> </li> </ul>
logout()	void	<ul style="list-style-type: none"> <li>• Method untuk logout pada panel ini.</li> <li>• Akan mengubah loggedInMember menjadi null</li> </ul>

- Silakan menambahkan method lain yang diperlukan

### Interface Loginable

- Interface ini adalah interface yang akan diimplementasikan oleh AbstractMemberGUI
- Terdapat tiga method abstract yang di-*override* pada class ini:

Method	Return Type	Deskripsi
login()	boolean	<ul style="list-style-type: none"> <li>• Sesuai dengan AbstractMemberGUI</li> </ul>
logout()	void	<ul style="list-style-type: none"> <li>• Sesuai dengan AbstractMemberGUI</li> </ul>
getPageName()	String	<ul style="list-style-type: none"> <li>• Mengembalikan nama dari Panel yang mengimplementasikan method ini.</li> </ul>

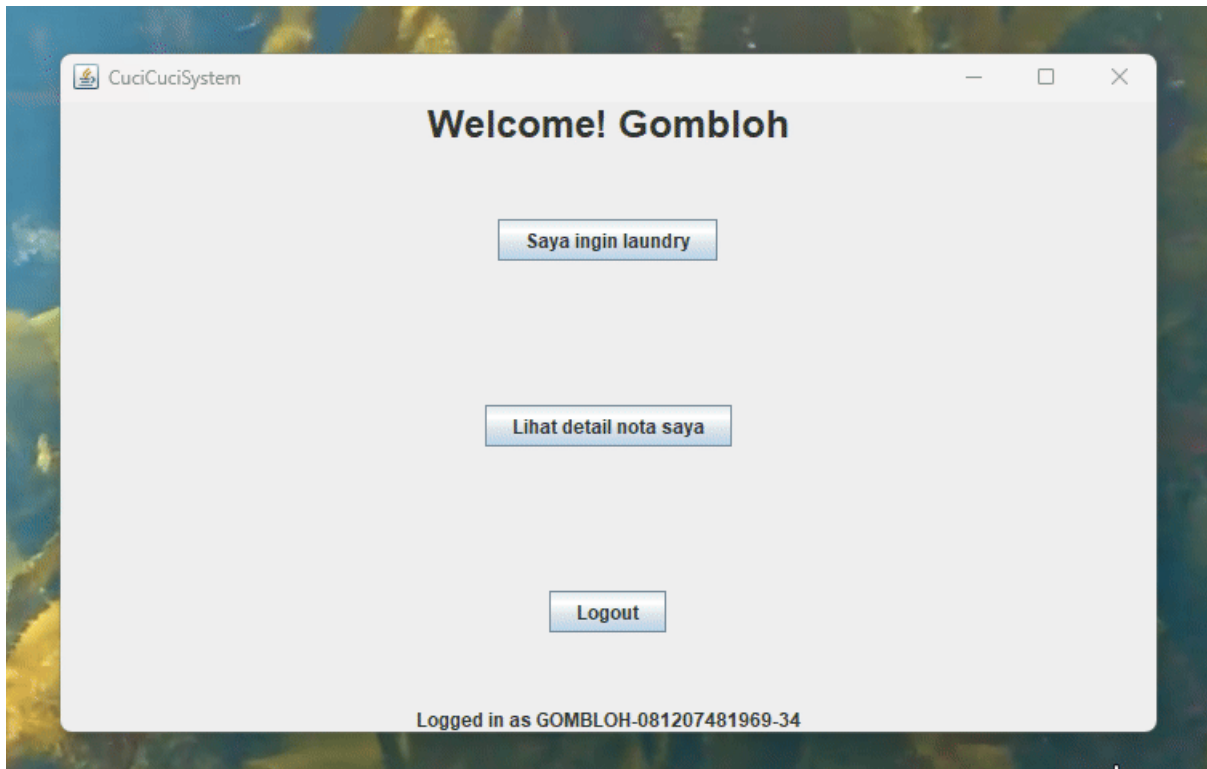
## Package gui.member.member

### Class MemberSystemGUI

- Class yang akan mengimplementasikan button dan logic untuk member biasa.
- Menginherit class AbstractMemberGUI
- Terdapat empat method yang akan kamu implementasikan:

Method	Return Type	Deskripsi
createButtons()	JButton[]	<ul style="list-style-type: none"><li>• Method ini mensupply buttons yang sesuai dengan requirements MemberSystem</li><li>• Button yang disediakan method ini BELUM memiliki ActionListener.</li></ul>
createActionListeners()	ActionListener[]	<ul style="list-style-type: none"><li>• Method ini mensupply ActionListener korespondensi dengan button yang dibuat createButtons() sesuai dengan requirements MemberSystem.</li></ul>
showDetailNota()	void	<ul style="list-style-type: none"><li>• Menampilkan detail Nota milik loggedInMember.</li><li>• Akan dipanggil jika pengguna menekan button pertama pada createButtons</li></ul>
createNota()	void	<ul style="list-style-type: none"><li>• Pergi ke halaman CreateNotaGUI.</li><li>• Akan dipanggil jika pengguna menekan button kedua pada createButtons</li></ul>

- Berikut adalah fungsionalitas minimal untuk MemberSystemGUI

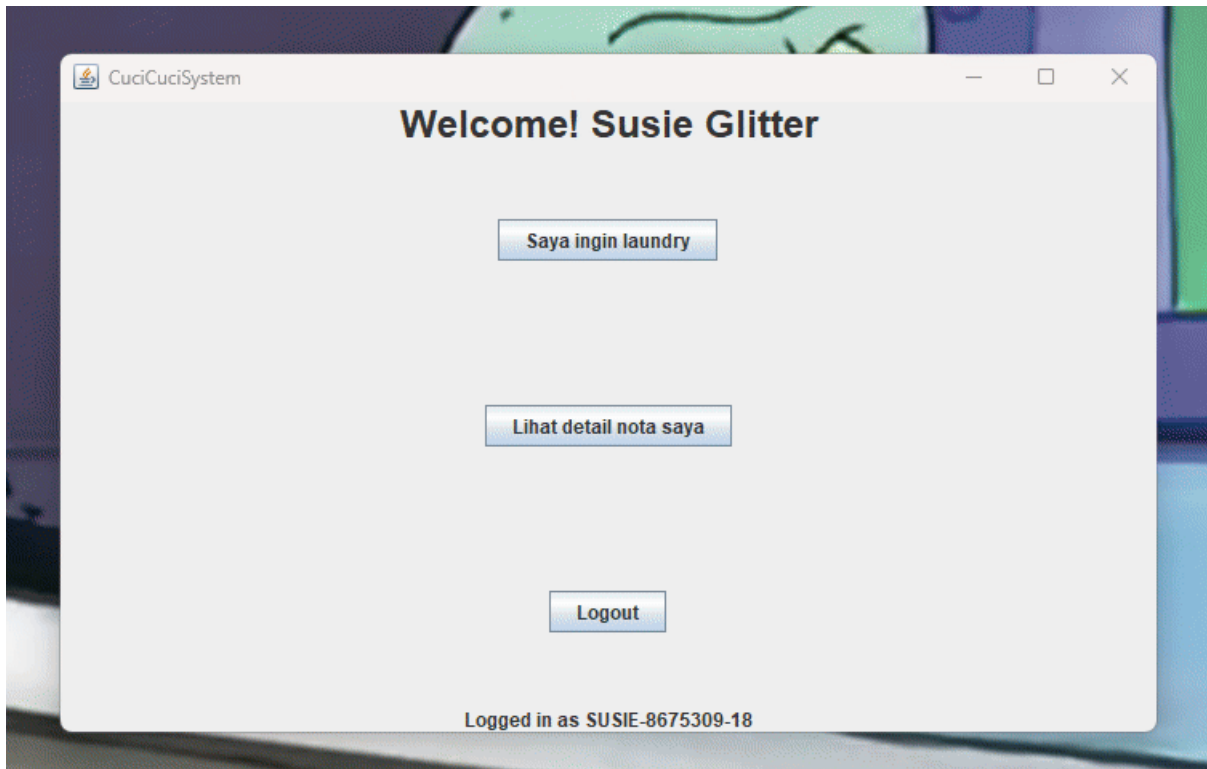


- Silakan menambahkan method lain jika diperlukan.

### Class CreateNotaGUI

- Panel untuk membuat nota.
- Terdapat combo box untuk memilih paket, disebelahnya ada tombol show paket yang bila ditekan akan menampilkan paket yang tersedia.
- Terdapat TextField sebagai input berat cucian, kamu masih harus melakukan pengecekan input disini:
  1. Input harus selalu angka. Ini berarti - (tanda negatif) tidak bisa masuk.
  2. Jika validasi gagal, field ini akan dikosongkan kembali
  3. Ketentuan berat cucian sebelumnya masih berlaku.
- Jika pengguna menekan tombol "Buat Nota" nota akan dibuat seperti pada ketentuan TP sebelumnya dan semua field akan dikosongkan (untuk combo box akan kembali pada "Express").
- Jika pengguna menekan tombol kembali, pengguna akan kembali ke halaman MemberSystem dan semua field akan dikosongkan (untuk combo box akan kembali pada "Express").

- Berikut adalah fungsionalitas minimal untuk CreateNotaGUI



## Package gui.member.employee

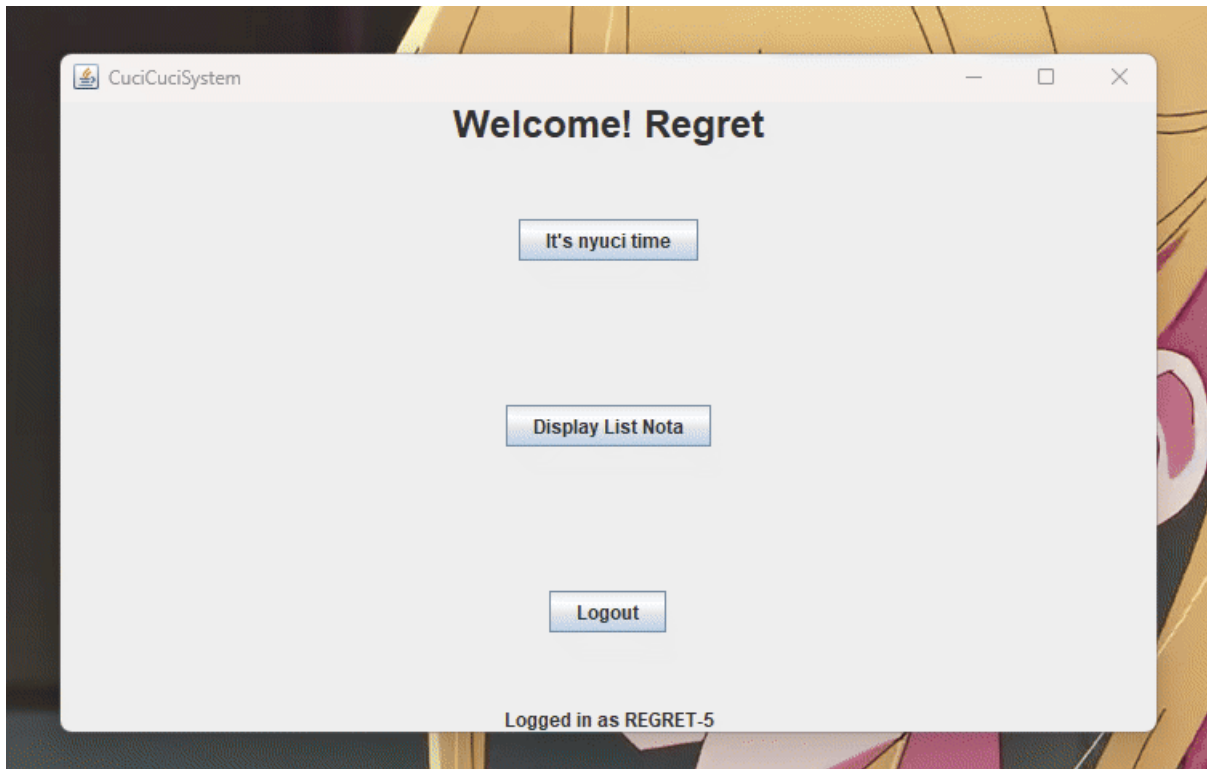
### Class EmployeeSystemGUI

- Class yang akan mengimplementasikan button dan logic untuk **employee**.
- Menginherit class AbstractMemberGUI
- Terdapat empat method yang akan kamu implementasikan:

Method	Return Type	Deskripsi
createButtons()	JButton[]	<ul style="list-style-type: none"> <li>• Method ini mensupply buttons yang sesuai dengan requirements MemberSystem</li> <li>• Button yang disediakan method ini BELUM memiliki ActionListener.</li> </ul>
createActionListeners()	ActionListener[]	<ul style="list-style-type: none"> <li>• Method ini mensupply ActionListener korespondensi dengan button yang dibuat createButtons() sesuai dengan requirements MemberSystem.</li> </ul>
<b>displayNota()</b>	void	<ul style="list-style-type: none"> <li>• <b>Menampilkan status semua Nota.</b></li> <li>• Akan dipanggil jika pengguna menekan button pertama pada createButtons</li> </ul>

cuci()	void	<ul style="list-style-type: none"> <li>• melakukan proses mencuci dan menampilkan textnya, seperti pada TP sebelumnya.</li> <li>• Akan dipanggil jika pengguna menekan button kedua pada createButtons</li> </ul>
--------	------	---

- Berikut adalah fungsionalitas minimal untuk EmployeeSystemGUI



## Test Case

Video: [click disini!](#)

## Komponen Penilaian

- 40% Fungsionalitas program
- 10% Validasi input & tampilan error message
- 5% Informasi prompt program
- 5% Kerapian GUI dan output program
- 5% Dokumentasi (kejelasan kode)
- 5% Standar penulisan kode\* (kerapian kode)
- 30% Demo program dengan asisten dosen

\*Standar penulisan kode yang harus dipenuhi yaitu:

1. Indentasi yang konsisten ([sumber](#))
2. Aturan penamaan harus mengikuti Java Naming Convention ([sumber](#))

### 3. Penamaan Module, Class, Method, dan Variabel yang tidak ambigu

#### Catatan

- **Kabarkan asdosmu jika menggunakan package di luar java package.**  
Pastikan jika kamu ingin menambahkan sesuatu di luar java package, package tersebut hanya untuk unsur estetika dengan persetujuan asdos kamu tentunya.

#### Pengumpulan Berkas

Tidak ada berkas yang perlu dikumpulkan di SceLe untuk TP 4, pengumpulan code dilakukan melalui commit di GitHub