

PBO - Modul Praktik 4

Pada modul praktikum ini kita akan membuat beberapa komponen User Interface berbasis CLI yang dapat digunakan berulang-ulang, serta membuat class `Page` yang memanfaatkan komponen tersebut. Komponen yang akan kita buat adalah: Horizontal Line, Header, Label, Input, Selection Input, dan Space. Adapun class page yang akan memanfaatkan komponen tersebut adalah: Main Page, Input Page, dan Password List Page.

Terdapat 2 jenis class komponen yang kita gunakan pada praktikum ini, yaitu komponen output dan komponen input. Komponen output yang perlu dibuat adalah class `HLine`, `Space`, dan `Label`. Adapun jenis class input yang akan kita buat adalah class `Input` dan `SelectedInput`.

Class Komponen Output

Class yang bertanggung jawab untuk memunculkan tampilan pada terminal (CLI) sesuai dengan spesifikasi tampilan agar lebih rapi dan seragam. Di bawah ini adalah class diagram dari komponen output.

HLine	Space	Label
- int width	- int width	- int width + String text
+HLine(width) +void draw()	-Space(width) -void draw()	+Label(text, width) +void draw()

Class `HLine`

Bertugas untuk menampilkan garis horizontal (*horizontal line*) sebagai pemisah antara satu bagian tampilan dengan bagian yang lain. Class `HLine` memiliki **spesifikasi sebagai berikut**:

- 1 atribut yaitu `width` untuk menentukan lebar garis / jumlah separator yang digunakan untuk baris.
- 1 konstruktor dengan parameter `width` untuk mengeset atribut `width`.
- Method `draw` untuk menampilkan String garis di layar terminal dengan format sebagaimana ditunjukkan di bawah.
- Contoh pemanggilan `new HLine(50).draw()` akan menghasilkan output sebagai berikut

```
+=====+
```

- Jumlah karakter `=` yang dicetak mengikuti nilai `width`

Class `Space`

Berfungsi untuk memberikan 1 baris jeda kosong antara komponen sebelum (di atasnya) dan komponen setelahnya dengan tetap menggunakan aturan tampilan yang ditentukan. Class `Space` memiliki **spesifikasi sebagai berikut**:

- 1 attribute `width` untuk menentukan lebar spasi
- 1 konstruktor dengan parameter `width` untuk mengeset atribut `width`.
- Method `draw` untuk menampilkan baris kosong dengan garis depan & belakang
- Contoh pemanggilan `new Space(50).draw()` akan menghasilkan output seperti berikut:

```
|                                     |
```

- Jumlah karakter (spasi) yang dicetak mengikuti nilai `width`

Class `Label`

Berguna untuk menampilkan suatu teks dalam 1 baris menggunakan aturan tampilan yang ditentukan. Class `Label` memiliki **spesifikasi sebagai berikut**:

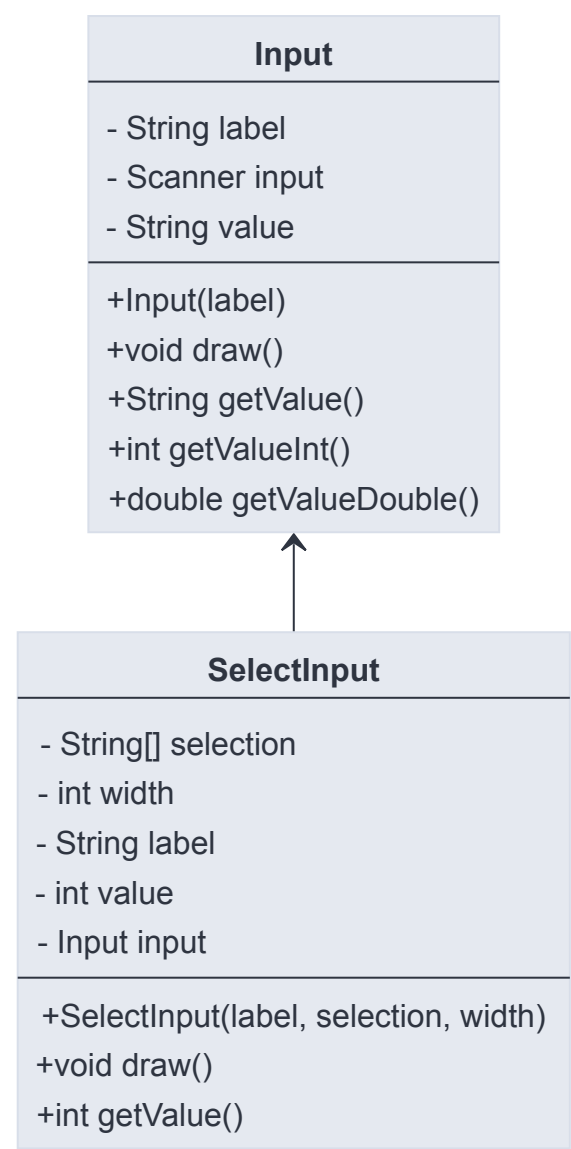
- atribut `width` untuk menentukan lebar minimal area
- atribut `text` untuk menentukan teks yang dicetak pada baris tersebut
- 1 konstruktor dengan parameter `text` untuk mengset atribut `text` dan `width` untuk mengeset atribut `width`
- method `draw` untuk menampilkan `text` pada area dengan lebar minimal `width-2` (karena terdapat 2 spasi sebelum teks).
- contoh pemanggilan `new Label("Teks tercetak", 50).draw()` akan menghasilkan output seperti berikut:

```
|  Teks tercetak  |
```

- Jumlah karakter (spasi) yang tercetak mengikuti nilai `width - 2 - panjangTeks` . Namun jika panjang teks melebihi nilai `width` maka, hanya 1 spasi yang tercetak.

Class Komponen Input

Class yang memungkinkan penggunaan untuk memberikan inputan pada terminal (CLI) serta menampilkan label atau informasi yang relevan seputar inputan tersebut. Di bawah ini adalah class diagram dari class `Input`.



Class `Input`

Class ini memungkinkan user untuk memberikan inputan melalui terminal (CLI) serta menampilkan label dengan format yang sesuai. Class `Input` memiliki **spesifikasi sebagai berikut**:

- Atribut `label` untuk menyimpan teks label dari inputan
- Atribut `input` untuk menyimpan object stream input yang digunakan (`Scanner(System.in)`)
- Atribut `value` untuk menyimpan nilai hasil inputan pengguna
- Konstruktor dengan parameter `label` untuk mengeset atribut `label` (NB: tidak ada pengesetan width, karena belum ditemukan cara untuk menampilkan karakter penutup setelah input)
- Method `draw()` untuk menampilkan label di tampilan serta membuat kursor CLI siap menerima inputan
- Method `getValue()` untuk mendapatkan nilai (`String`) yang diinputkan pengguna
- Method `getValueInt()` untuk mendapatkan nilai integer yang diinputkan pengguna
- Method `getValueDouble()` untuk mendapatkan nilai double yang diinputkan pengguna

Contoh Penggunaan:

```
Input input = new Input("Masukkan nama");
Input input2 = new Input("Usia anda");
input.draw();
String nama = input.getValue();
input2.draw();
int usia = input2.getValueInt();
```

Output dari perintah tersebut adalah:

```
|  Masukkan nama : Fahri Firdausillah
|  Usia anda : 30
```

Catatan: `Fahri Firdausillah` dan `30` adalah inputan dari pengguna.

Class `SelectInput`

Class ini memungkinkan user untuk memberikan input berdasarkan pilihan yang ditampilkan, sehingga 2 hal yang ditampilkan adalah label dan (String array) pilihan, serta inputan terminal. Class `SelectInput` memiliki **spesifikasi sebagai berikut**:

- Atribut `label` merupakan teks label yang berisikan informasi yang relevan dengan input
- Atribut `width` untuk menentukan lebar tampilan dari pilihan inputan
- Atribut `selection` berisi *Array of String* teks pilihan inputan
- Atribut `value` digunakan untuk menampung hasil inputan pengguna
- Atribut `input` merupakan komponen class `Input` yang digunakan untuk mendapatkan pilihan inputan pengguna
- Konstruktor digunakan untuk inisialisasi atribut `label`, `width`, `selection`, dan komponen `input`.
- Method `draw()` bertanggung jawab memberikan tampilan pilihan dan input sesuai format yang ditentukan.
- Method `getValue()` dapat digunakan untuk mendapatkan nilai pilihan yang diinputkan pengguna.

Contoh Penggunaan:

```
String [] pilihan = {"Pilihan 1", "Pilihan 2", "Pilihan 3", "Pilihan 4"};
SelectInput pilSelect = new SelectInput("Inputkan pilihan anda:", pilihan, 70);
pilSelect.draw();
int value = pilSelect.getValue();
```

Output dari perintah tersebut adalah:

```
|   Inputkan pilihan anda:   |
|   [1] Pilihan 1           |
|   [2] Pilihan 2           |
|   [3] Pilihan 3           |
|   [4] Pilihan 4           |
|   Pilihan : 3             |
```

Catatan: 3 adalah inputan dari pengguna.

Testing Semua Komponen

Contoh pemanggilan untuk semua komponen adalah sebagai berikut:

```
new HLine(50).draw();
new Space(50).draw();
new Label("Uji coba komponen yang dibuat", 50).draw();
new Space(50).draw();

new HLine(50).draw();
new Space(50).draw();
Input input = new Input("Masukkan nama");
Input input2 = new Input("Usia anda");
input.draw();
String nama = input.getValue();
input2.draw();
int usia = input2.getValueInt();

new Space(50).draw();
String [] pilihan = {"Pilihan 1", "Pilihan 2", "Pilihan 3", "Pilihan 4"};
SelectInput pilSelect = new SelectInput("Inputkan pilihan anda:", pilihan, 50);
pilSelect.draw();
int value = pilSelect.getValue();
new Space(50).draw();
new HLine(50).draw();

new Space(50).draw();
new Label("Nama anda: "+nama, 50).draw();
new Label("Usia anda: "+usia, 50).draw();
new Label("Anda memilih menu: "+value, 50).draw();

new Space(50).draw();
new HLine(50).draw();
```

Pemanggilan tersebut akan menghasilkan output tampilan seperti di bawah ini:

```
+=====+
|                                     |
|   Uji coba komponen yang dibuat   |
|                                     |
+=====+
|                                     |
|   Masukkan nama : James Gosling   |
```

```

|  Usia anda : 80
|
|  Inputkan pilihan anda:
|  [1] Pilihan 1
|  [2] Pilihan 2
|  [3] Pilihan 3
|  [4] Pilihan 4
|  Pilihan : 2
|
+=====+
|
|  Nama anda: James Gosling
|  Usia anda: 80
|  Anda memilih menu: 2
|
+=====+

```

Class Page (Halaman)

Class Page berfungsi sebagai penampung komponen user interface yang dengan satu tujuan yang relevan. Sebuah page selayaknya memiliki beberapa komponen default yaitu header (termasuk judul halaman), footer, serta konten. Selain itu, attribute page juga digunakan untuk menentukan default width untuk semua komponen yang ada di dalamnya, sehingga default atribut dan method yang harus dimiliki tiap page adalah:

- Attribute: `title` , `width`
- Method: `draw` , `drawContent` , `drawHeader` , dan `drawFooter`

Atribut dan method lain dapat ditambahkan pada class tersebut sesuai kebutuhan.

Pada praktikum ini kita akan membuat 3 page yaitu:

- `MainPage` : Halaman utama yang pertama kali dipanggil saat aplikasi dijalankan
- `InputPage` : Halaman input data password
- `ListPasswordPage` : Halaman untuk menampilkan list password yang tersimpan

Catatan Khusus

Untuk menyimpan data password ke dalam sebuah variabel global, kita dapat membuat 1 class khusus seperti berikut:

```

public class DataPassword {
    public static final ArrayList<PasswordStore> passData = new ArrayList<>();
}

```

Class `MainPage`

Halaman utama dan yang pertama kali dibuka saat aplikasi dijalankan. Spesifikasi yang harus dipenuhi pada halaman ini adalah:

- Menampilkan pesan selamat datang ke aplikasi
- Menampilkan pilihan menu yang terdiri dari:
 1. Halaman input password
 2. Halaman tampil password
 3. Keluar dari aplikasi
- Jika pilihan menu adalah `1` maka program akan membuka halaman Input Password
- Jika pilihan menu adalah `2` maka program akan membuka halaman Tampil data Password

- Jika pilihan menyunya adalah 3 maka program akan keluar
- Selain itu, program akan menampilkan kembali halaman MainPage

Implementasi koding dari class MainPage adalah sebagai berikut:

```
public class MainPage {

    public String title;
    public int width;

    private final HLine hline;
    private final Space space;
    private final Label label;

    public MainPage(String title, int width){
        this.width = width;
        this.title = title;
        this.hline = new HLine(width);
        this.space = new Space(width);
        this.label = new Label(title.toUpperCase(), width);
    }

    public void draw(){
        this.drawHeader();
        this.space.draw();
        this.drawContent();
    }

    public void drawHeader(){
        this.hline.draw();
        this.space.draw();
        this.label.draw();
        this.space.draw();
        this.hline.draw();
    }

    private void drawContent(){
        String [] pages = {"Input Password", "Tampil Password", "Keluar Aplikasi"};
        SelectInput pageSelect = new SelectInput("Pilih halaman berikut:",
                                                pages, this.width);

        int select;

        this.label.text = "Selamat datang di aplikasi Password Vault";
        this.label.draw();
        this.label.text = "Simpan password anda dengan aman di sini";
        this.label.draw();
        this.space.draw();

        pageSelect.draw();
        select = pageSelect.getValue() - 1;
        switch (select) {
            case 0 -> {
```

```

        drawFooter();
        new InputPage("Inputan Password", this.width).draw();
    }
    case 1 -> {
        drawFooter();
        new PasswordListPage("List Password Tersimpan", this.width).draw();
    }
    case 2 -> {
        new Label("Terima kasih telah menggunakan aplikasi", this.width).draw();
        drawFooter();
    }
    default -> {
        new MainPage(this.title, this.width).draw();
    }
}

public void drawFooter(){
    this.space.draw();
    this.hline.draw();
}
}

```

Class InputPage

Halaman untuk menginput data password. Spesifikasi yang harus dipenuhi adalah sebagai berikut:

- Memiliki 3 inputan teks untuk menginput data judul (atau nama) data, username, dan password
- Serta memiliki 1 inputan seleksi untuk menginput kategori password. Perlu diingat ada 4 kategori password yang sudah didefinisikan di class `PasswordStore` yaitu `UNCATEGORIZED`, `CAT_WEBAPP`, `CAT_MOBILEAPP`, dan `CAT_OTHER`.
- Setelah semua inputan terisi, program akan menyimpan data password tersebut pada variable static `DataPassword.passData` yang sudah kita buat sebelumnya.
- Setelah selesai input, program akan kembali menampilkan `MainPage`
- Format tampilan halaman ini adalah sebagai berikut:

```

+=====+
|                                             |
|   Inputan Password                         |
|                                             |
+=====+
|                                             |
|   Inputan Password Baru                   |
|   Judul Password : Login Instagram        |
|   Username : @bagusbanget                 |
|   Password : RahasiaPasswordnya          |
|   Kategori                               |
|   [1] Belum terkategori                   |
|   [2] Aplikasi Web                       |
|   [3] Aplikasi Mobile                    |
|   [4] Akun Lainnya                       |
|   Pilihan : 3                             |

```

```
|  -----  |
|  Input password berhasil dibuat  |
|  |
+=====+
```

Class ListPasswordPage

Halaman untuk menampilkan list password yang sudah tersimpan. Spesifikasi yang harus dipenuhi adalah sebagai berikut:

- Program membaca data password dari variabel static `DataPassword.dataPass` yang sudah dibuat sebelumnya.
- Program menampilkan jumlah password yang tersimpan.
- Program menampilkan semua list password dalam bentuk semi tabel seperti dengan format di bawah ini.

```

+=====+
|
| List Password Tersimpan
|
+=====+
|
| Terdapat 3 tersimpan.
|
| ----
|
| | Login Instagram | @bagusbanget | Aplikasi Mobile |
| | Login SiAdin    | A12345678    | Aplikasi Web    |
| | Login Gmail     | akun@gmail.com | Aplikasi Web    |
|
+=====+

```

- Usahakan ukuran masing-masing kolom bisa seragam antar barisnya.