

MODUL 2

Constructor



Tujuan:

Mahasiswa dapat mengenal dan memahami konsep constructor dan overloading constructor

Materi:

- | | |
|---------------|---------------------------|
| ✓ Pengantar | ✓ Overloading constructor |
| ✓ Constructor | ✓ Soal Latihan |

Referensi:

- ❖ Fikri, Rijalul. 2005. *Pemrograman Java*. Yogyakarta: Penerbit Andi
- ❖ Hermawan, Benny. 2004. *Menguasai Java 2 & Object Oriented Programming*. Yogyakarta: Penerbit Andi
- ❖ Purnama, Rangsang. 2003. *Tuntunan Pemrograman Java Jilid 2*. Surabaya: Prestasi Pustaka Publisher

1.1. Pengantar

Kalau anda melihat gambar yang ada di cover pada modul 2, anda mungkin berpikiran bahwa constructor serupa dengan seorang yang membangun sebuah gedung, apartemen, maupun bangunan-bangunan tinggi. Memang sich, jawaban anda tidak salah untuk yang satu ini. Tapi jawaban yang anda kemukakan akan keliru ketika anda melihat constructor dari segi pemrograman. Lalu sebenarnya apa sich constructor itu? Baiklah akan saya jelaskan dengan bahasa yang mudah anda pahami melalui sebuah ilustrasi.



Setiap manusia pasti memiliki nama bukan? Coba anda bertanya kepada diri anda anda, “Siapakah aku? (atau biasa dikenal dengan *Who am I?*)?”. Sadar atau tidak, anda akan memperkenalkan diri anda dengan menyebutkan nama, alamat, tanggal lahir, hobi, dan sebagainya. Namun, pernahkah anda berpikir mengapa banyak orang suka menghafal nama anda ketimbang alamat, tanggal lahir, hobi, dan sebagainya? Padahal terkadang nama anda tuh banyak dipakai juga pada orang lain? Bahkan nama lengkap saya sendiri pun juga ada yang menggunakannya meskipun ejaannya berbeda. (Berarti saya terkenal donk :p). Itu semua karena nama anda adalah unik. Sekali lagi, coba anda bayangkan bagaimana kalau anda tidak mempunyai nama? Orang lain pasti akan kebingungan memanggil anda. Bisa-bisa anda akan dipanggil “Anonymous” sebagai seseorang tanpa nama. Demikian pula dalam pembuatan constructor.

Ketika anda membuat sebuah objek dari class manusia (sebut saja “Orang 1”), kemudian anda dapat menge-set nilai berupa nama, alamat, tanggal lahir, dan

hobi menggunakan method setter(). Berbeda halnya dengan constructor. **Ketika objek “Orang 1” telah terbentuk, anda langsung memberikan nilai** berupa nama, alamat, tanggal lahir, dan hobi. Hal itu ibarat anda baru lahir di dunia ini dan langsung diberi nama. Itulah merupakan konsep dari Constructor. Untuk lebih jelas tentang konsep Constructor dan penerapannya, silahkan anda simak penjelasannya di bawah ini.

1.2. Constructor

Constructor adalah method yang berfungsi untuk menginisialisasi variabel-variabel instans yang akan dimiliki oleh objek. Constructor dipanggil pada saat proses instansiasi kelas menjadi objek. Jika kelas tidak memiliki method constructor, maka seluruh variabel objek akan diinisialisasi kepada nilai default, sesuai dengan tipe datanya masing-masing. Berikut adalah struktur constructor.

```
1  class Nama_Kelas
2  {
3      Nama_Kelas ()
4      {
5          //isi constructor
6      }
7
8      // isi dari kelas
9  }
```

Contoh penggunaan constructor:

```
1  class Login
2  {
3      Login ()
4      {
5          //isi constructor
6      }
7
8      // isi dari kelas
9  }
```

1.3. Karakteristik Constructor

Berikut ini adalah beberapa karakteristik yang dimiliki oleh constructor:

1. Method constructor harus memiliki nama yang sama dengan nama class
2. Tidak mengembalikan suatu nilai (tidak ada keyword *return*)
3. Satu kelas memiliki lebih dari constructor (*overloading constructor*)
4. Dapat ditambah *access modifier* public, private, protected maupun default
5. Suatu constructor bisa dipanggil oleh constructor lain dalam satu kelas

1.4. Overloading Constructor

Seperti yang telah dijelaskan poin 3 pada karakteristik constructor bahwa dalam sebuah class dapat memiliki lebih dari satu constructor. Yang membedakan antara constructor yang satu dengan yang lainnya adalah jumlah parameter dan tipe data di dalamnya. Struktur Overloading Constructor adalah sebagai berikut.

```
1  class Nama_Kelas
2  {
3      Nama_Kelas ()
4      {
5          //isi constructor
6      }
7
8      Nama_Kelas(parameter)
9      {
10         //isi constructor
11     }
12
13     // isi dari kelas
14 }
```

Contoh penggunaan constructor:

```
1 class Login
2 {
3     private String username, password;
4
5     Login()
6     {
7         username = "admin";
8         password = "12345";
9     }
10
11    Login(String username, String password)
12    {
13        this.username = username;
14        this.password = password;
15    }
16
17    // isi dari kelas
18 }
```

Pada contoh di atas, anda melihat class Login dimana memiliki 2 constructor. Selain itu, terdapat perbedaan cara membuat objek (*instance of class*) dengan menggunakan overloading constructor di class Utama. Jika anda ingin membuat *instance of class* dengan method Login tanpa parameter adalah sebagai berikut:

```
Login user1 = new Login();
```

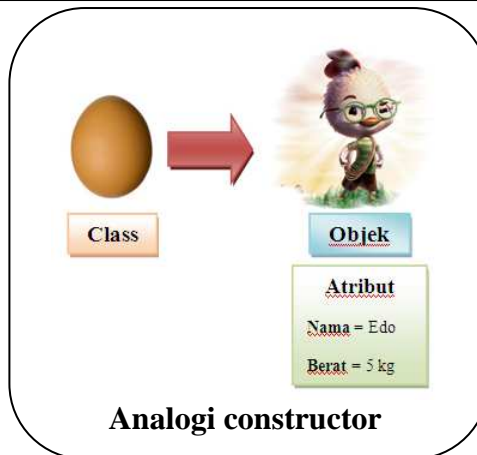
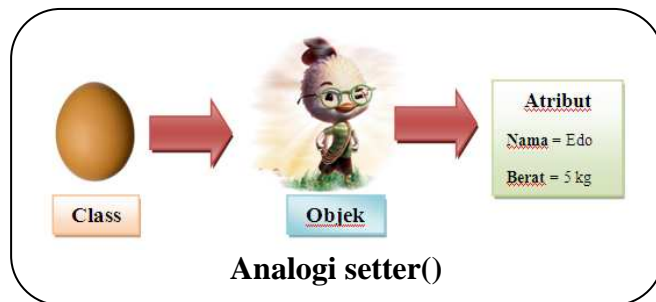
Pembuatan objek di atas akan memberikan nilai set default username dan password berupa admin dan 12345. Sedangkan jika anda ingin membuat user sendiri (misal: username=edo, password=pb0) melalui *instance of class* dengan method Login menggunakan parameter adalah sebagai berikut:

```
Login user1 = new Login("edo", "pb0");
```

1.5. Perbedaan menggunakan Constructor dengan method Setter()

Jika anda melihat dengan teliti, anda pasti akan bertanya-tanya “lalu apa perbedaan penggunaan setter() dengan constructor?”. Dalam method setter(), pertama kali obyek dibuat dari sebuah kelas (*instance of class*). Setelah objek terbentuk, kemudian objek tersebut diberi atribut. Berbeda dengan constructor.

Dalam constructor, obyek yang dibuat dari sebuah kelas (*instance of class*) langsung diberi atribut. Berikut adalah analogi perbedaan setter() dan constructor.



1.6. Soal Latihan

Berdasarkan contoh di atas tentang login, buatlah 2 buah class yang terdiri dari class Login dan class utama. Class Login harus memiliki beberapa ketentuan sebagai berikut:

- ✓ Atribut berisi **username** dan **password**
- ✓ Terdapat 2 buah constructor Login. Constructor pertama tidak memiliki parameter dan memiliki nilai default username="admin" dan password="12345". Sedangkan constructor kedua memiliki parameter untuk mengeset nilai username dan password berdasarkan inputan user.
- ✓ Terdapat method **setter** dan **getter** untuk mengeset/merubah dan mengambil nilai dari **username dan password**

Sedangkan pada Class utama digunakan untuk memanggil class Login.

Ketika class utama dijalankan, hasilnya akan tampak seperti di bawah ini:

```
C:\Program Files\Xinox Softw... - _ X
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
```

- ❖ Jika pilihan = 1, maka akan tampil sebagai berikut:

```
C:\Program Files\Xinox Softw... - _ X
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = 1
Masukan username = admin
Masukan password = 12345
*** Login Sukses ***
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
```

Input

Output

Kembali ke Menu Utama

- ❖ Jika pilihan = 2, maka akan tampil sebagai berikut:

```
C:\Program Files\Xinox Software\U... - _ X
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = 2
Masukan password lama = 12345
Masukan password baru = g3j3
*** Password berhasil dirubah ***
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
```

Input

Output

Kembali ke Menu Utama

- ❖ Jika pilihan = 3, maka akan tampil sebagai berikut:

```

C:\Program Files\Xinox Software\J...
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = 3
Masukan username = edo
Masukan password = 3d0
*** User telah berhasil dibuat ***
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
    
```

- ❖ Jika pilihan = 4, maka akan tampil sebagai berikut:

```

C:\Program Files\Xinox Software\J...
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = 4
=====
DATA USER
User 1
Username = admin
Password = g3j3
User 2
Username = edo
Password = 3d0
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
    
```

- ❖ Jika pilihan = 5, maka akan tampil sebagai berikut:

```

C:\Program Files\Xinox Software\J...
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = 5
Press any key to continue...
=====
MENU LOGIN
=====
1. Login admin
2. Ubah password admin
3. Buat user
4. Lihat Data user
5. Keluar
=====
Masukan pilihan = _
    
```

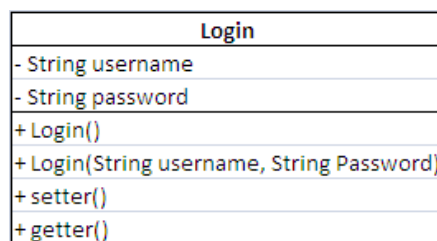
Jawabannya adalah...

Setelah anda membaca soal tersebut dengan baik dan seksama, langkah pertama yang harus anda lakukan adalah menganalisa soal tersebut dan membuat

skema diagram dari soal tersebut. Skema ini nantinya akan membantu anda pembuatan program. Berikut adalah skema diagramnya.

➤ **Langkah 1: Membuat skema**

Skema diagram digunakan untuk membantu anda dalam membantu logika anda untuk pembuatan program. Tanda “-“ dilambangkan sebagai *private*. Sedangkan tanda “+” dilambangkan sebagai *public*. Berikut adalah skema diagramnya.



➤ **Langkah 2: class Login (ketikkan script berikut)**

a. Membuat kerangka class Login

```
1 class Login
2 {
3     //deklarasi
4
5     //constructor
6
7     //setter
8
9     //getter
10
11 }
```

Setelah anda membuat class Login, simpan file tersebut dengan nama **Login.java**. Di dalam class Login, saya juga menyediakan tempat untuk mendeklarasikan variabel, setter dan getter.

b. Mendeklarasi variabel yang dibutuhkan

Setelah kita membuat kerangka class, maka diperlukan pendeklarasian variabel yang nantinya digunakan sebagai tempat menyimpan data yang

bersifat sementara (*temporary*). Gambar di bawah ini menunjukkan pendeklarasian variabel.

```
1 class Login
2 {
3     //deklarasi
4     private String username, password;
5
6     //constructor
7
8     //setter
9
10    //getter
11
12 }
```

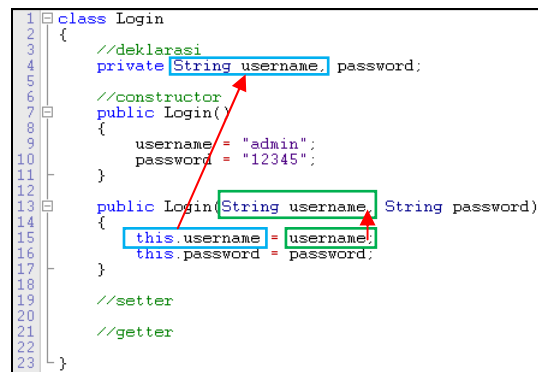
c. Mendeklarasi constructor

Setelah membuat variabel yang dibutuhkan pada class Login, langkah selanjutnya anda membuat constructor login. Constructor ini nantinya akan digunakan dalam class Utama. Gambar di bawah ini menunjukkan deklarasi constructor.

```
1 class Login
2 {
3     //deklarasi
4     private String username, password;
5
6     //constructor
7     public Login()
8     {
9         username = "admin";
10        password = "12345";
11    }
12
13    public Login(String username, String password)
14    {
15        this.username = username;
16        this.password = password;
17    }
18
19    //setter
20
21    //getter
22
23 }
```

Coba perhatikan kembali script yang telah anda buat seperti gambar di atas. Seperti halnya dengan penggunaan method setter(), di dalam pembuatan constructor Login dengan parameter, anda juga dapat menggunakan keyword *this*. Penggunaan keyword *this* akan mengacu kepada variabel yang dideklarasikan pada class Login (lihat *script* yang diberi kotak berwarna biru pada gambar di bawahnya). Apabila variabel tersebut tersebut tidak diberi keyword *this*, maka variabel tersebut akan mengacu kepada variabel yang dideklarasikan pada parameter constructor (lihat *script* yang

diberi kotak berwarna hijau pada gambar di bawahnya). Penggunaan keyword **this** dapat digunakan atau tidak (*optional*) apabila ada perbedaan deklarasi nama variabel pada class Login (untuk lebih jelas mengenai keyword **this**, anda dapat melihat modul 1).



```
1 class Login
2 {
3     //deklarasi
4     private String username, password;
5
6     //constructor
7     public Login()
8     {
9         username = "admin";
10        password = "12345";
11    }
12
13    public Login(String username, String password)
14    {
15        this.username = username;
16        this.password = password;
17    }
18
19    //setter
20
21    //getter
22
23 }
```

d. Membuat method setter

Setelah membuat variabel yang dibutuhkan pada class Login, langkah selanjutnya anda buat method setter untuk masing-masing variabel. Hal ini bertujuan untuk mengeset atau merubah nilai variabel username dan password sesuai dengan permintaan soal pada menu yang ke-2 di class Utama nanti. Perlu diketahui pula, bahwa dalam pembuatan variabel pada parameter di method setter, harus sesuai dengan tipe data pada variabel yang dideklarasikan di kelas. Gambar di bawah ini menunjukkan deklarasi setter.

```

1 class Login
2 {
3     //deklarasi
4     private String username, password;
5
6     //constructor
7     public Login()
8     {
9         username = "admin";
10        password = "12345";
11    }
12
13    public Login(String username, String password)
14    {
15        this.username = username;
16        this.password = password;
17    }
18
19    //setter
20    public void setUsername(String username)
21    {
22        this.username=username;
23    }
24    public void setPassword(String password)
25    {
26        this.password=password;
27    }
28
29    //getter
30
31 }

```

e. Membuat method getter

Setelah membuat method setter, anda tinggal membuat method getter untuk mengambil nilai dari masing-masing variabel. Hal ini bertujuan untuk mengambil nilai dari variabel pada class Login yang nantinya akan kita kembalikan ke dalam class Utama. Perlu diketahui pula, bahwa dalam pembuatan variabel pada parameter di method getter, harus sesuai dengan tipe data pada variabel yang dideklarasikan di kelas. Gambar di bawah ini menunjukkan deklarasi getter.

```

1 class Login
2 {
3     //deklarasi
4     private String username, password;
5
6     //constructor
7     public Login()
8     {
9
10    }
11
12    public Login(String username, String password)
13    {
14
15    }
16
17    //setter
18    public void setUsername(String username)
19    {
20        this.username=username;
21    }
22    public void setPassword(String password)
23    {
24        this.password=password;
25    }
26
27    //getter
28    public String getUsername()
29    {
30        return username;
31    }
32    public String getPassword()
33    {
34        return password;
35    }
36
37 }
38

```

➤ **Langkah 3: class Utama (ketikkan script berikut)**

a. Membuat kerangka class Utama

```
1 import java.io.*;
2 class Utama
3 {
4     public static void main (String [] args) throws Exception
5     {
6         BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
7
8         //instance of class
9
10        //menu
11
12        //input
13
14        //proses + output
15    }
16 }
```

Setelah anda membuat class Utama, simpan file tersebut dengan nama **Utama.java**. Di dalam class inilah, program anda akan dijalankan. Sebagai catatan, dalam pembuatan class di atas, saya sudah menambahkan class `BufferedReader` (line 6) yang berada pada package `java.io.*` (line 1) yang digunakan untuk menerima inputan user.

b. Membuat *instance of class*

Setelah anda membuat class Utama, langkah berikutnya yang anda lakukan adalah membuat sebuah objek yang bertipe class `Login`. Pembuatan variabel dengan bertipe kelas itulah yang dinamakan *instance of class* (untuk penjelasannya, dapat anda lihat pada modul 1). Misalkan, objek yang saya buat adalah **user1** dan **user2**, dimana **user1** menggunakan constructor `Login` tanpa parameter, sedangkan **user2** menggunakan constructor `Login` dengan parameter.

```
1 import java.io.*;
2 class Utama
3 {
4     public static void main (String [] args) throws Exception
5     {
6         BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
7
8         //instance of class
9         Login user1 = new Login();
10        Login user2 = new Login(" ", " ");
11
12        //menu
13
14        //input
15
16        //proses + output
17    }
18 }
```

Coba perhatikan kembali pembuatan *instance of class*. Pada *line 7*, objek “user1” yang telah terbentuk akan mereferens ke constructor Login tanpa parameter. Sedangkan pada *line 8*, kita mendeklarasikan objek “user2” bertipe class Login, dimana nilai yang kita berikan masih belum diketahui. Karena tipe data username dan password bertipe *String*, maka saya menggunakan tanda petik ganda (“ ”) untuk memberi nilai awal berupa kosong. Variabel **user 2** yang telah dibuat nantinya akan digunakan pada **case 3**.

c. Membuat menu dan perulangan menu

Menu digunakan untuk mempermudah user dalam melakukan transaksi, seperti halnya buku menu yang disajikan seorang pelayan di sebuah restoran. Dalam pembuatan menu, diperlukan tombol “next” dan “back” sehingga user dapat leluasa memposisikan diri pada transaksi yang ingin dia lakukan. Untuk itulah, diperlukan perulangan menu guna mengantisipasi hal itu. Gambar di bawah ini menunjukkan pembuatan menu dan perulangan menu.

```

1 import java.io.*;
2 class Utama
3 {
4     public static void main (String [] args) throws Exception
5     {
6         BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
7
8         //instance of class
9         Login user1 = new Login();
10        Login user2 = new Login("", "");
11
12        while(true)
13        {
14            //menu
15            System.out.println("=====");
16            System.out.println("MENU LOGIN");
17            System.out.println("=====");
18            System.out.println("1. Login admin");
19            System.out.println("2. Ubah password admin");
20            System.out.println("3. Buat user");
21            System.out.println("4. Lihat Data user");
22            System.out.println("5. Keluar");
23            System.out.println("=====");
24
25            //input
26
27            //proses + output
28        }
29    }
30 }
31
32

```

Line 15-23 menunjukkan menu yang kita butuhkan dalam contoh soal di atas. Sedangkan proses perulangan menu, saya menggunakan **while** yang

berada di luar menu (bagi anda yang tidak terbiasa menggunakan “while”, anda juga bisa menggunakan “do...while” maupun “for” dalam perulangannya). Di dalam “while”, saya menggunakan kondisi bernilai “true”, dimana program tersebut akan mengulang menu tersebut berulang kali. Untuk keluar dari menu tersebut, akan saya bahas nanti pada langkah poin (h).

d. Membuat inputan yang diisi user

Setelah menu dan perulangan menu selesai kita buat, maka kita membutuhkan inputan user untuk memilih menu tersebut. Berikut adalah contoh *script*-nya.

```
1 import java.io.*;
2 class Utama
3 {
4     public static void main (String [] args) throws Exception
5     {
6         BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
7
8         //instance of class
9         Login user1 = new Login();
10        Login user2 = new Login("", "");
11
12        while(true)
13        {
14            //menu
15            System.out.println("=====");
16            System.out.println("MENU LOGIN");
17            System.out.println("=====");
18            System.out.println("1. Login admin");
19            System.out.println("2. Ubah password admin");
20            System.out.println("3. Buat user");
21            System.out.println("4. Lihat Data user");
22            System.out.println("5. Keluar");
23            System.out.println("=====");
24
25            //input
26            System.out.print("Masukan pilihan = ");
27            int pilih = Integer.parseInt (br.readLine());
28
29            System.out.println();
30
31            //proses + output
32
33        }
34    }
35 }
```

Sekedar tambahan, “System.out.println();” pada line 29 hanya digunakan untuk memberikan jarak antara proses dengan inputan dari user.

e. Mengecek inputan user

Inputan user yang nantinya akan diisi, akan menentukan pilihan yang dieksekusi (kayak teroris aja ya... ☺). Untuk itu, dibutuhkan, pengecekan inputan user dengan menu yang dipilih. Di sini, saya menggunakan

switch...case... dikarenakan penggunaannya lebih mudah dalam mengecek sebuah menu.

```

25 //input
26 System.out.print("Masukan pilihan = ");
27 int pilih = Integer.parseInt (br.readLine());
28
29 //proses + output
30 switch(pilih)
31 {
32     //jika pilih = 1
33     case 1:
34         //isi pilihan bernilai 1 ketika dijalankan
35         break;
36
37     //jika pilih = 2
38     case 2:
39         //isi pilihan bernilai 2 ketika dijalankan
40         break;
41
42     //jika pilih = 3
43     case 3:
44         //isi pilihan bernilai 3 ketika dijalankan
45         break;
46
47     //jika pilih = 4
48     case 4:
49         //isi pilihan bernilai 4 ketika dijalankan
50         break;
51
52     //jika nilai pilih yang dimasukkan bukan 1,2,3 atau 4, maka program akan keluar secara otomatis
53     default:
54         System.exit(0);
55 }
56
57 }
58
59 }
60

```

Sekedar tambahan, dalam setiap case jangan lupa menambahkan **break** yang bertugas untuk menghentikan proses pengecekan menu apabila salah satu case sudah terpenuhi dan telah dieksekusi. Penggunaan **default** ditujukan apabila pilihan 1,2,3 atau 4 tidak sesuai dengan inputan user. Jika anda perhatikan baik-baik, “System.exit(0);” pada line 54 bertujuan untuk keluar dari menu dan mengakhiri program.

f. Mengisi Case 1 (Login Admin)

Ketika user memilih inputan menu no. 1, maka dilakukan beberapa proses sebagai berikut:

- Line 36-39: berisi permintaan inputan username dan password yang nantinya akan diisi oleh user
- Line 44-51: berisi pengecekan apakah user dan password yang diinputkan sesuai dengan isi data user dan password pada class Login. Jika hasilnya bernilai **true**, maka program akan mencetak tulisan ***** Login Sukses *****. Jika hasilnya bernilai **false**, maka program akan mencetak tulisan ***** Login Gagal *****


```

29 //proses + output
30 switch(pilih)
31 {
32     //jika pilih = 1
33     case 1:
34         //isi pilihan bernilai 1 ketika dijalankan
35         System.out.print("Masukan username = ");
36         String my_user = br.readLine();
37         System.out.print("Masukan password = ");
38         String my_password = br.readLine();
39
40         System.out.println();
41
42         //cek apakah user dan password yang diinputkan sesuai dengan isi data user dan password pada class Login
43         if (my_user.equals(user1.getUsername()) && my_password.equals(user1.getPassword()))
44         {
45             System.out.println("*** Login Sukses ***");
46         }
47         else
48         {
49             System.out.println("*** Login Gagal ***");
50         }
51         break;
52

```

g. Mengisi Case 2 (Ubah Password Admin)

Ketika user memilih inputan menu no. 2, maka dilakukan beberapa proses sebagai berikut:

- Line 60-63: berisi permintaan inputan password lama dan password baru yang nantinya akan diisi oleh user
- Line 67-75: berisi pengecekan apakah password lama yang diinputkan sesuai dengan isi password yang sudah ada pada class Login. Jika hasilnya bernilai **true**, maka program akan merubah password lama dengan password baru dan kemudian mencetak tulisan ***** Password berhasil dirubah*****. Jika hasilnya bernilai **false**, maka program akan mencetak tulisan ***** Anda salah memasukkan password lama *****

```

56 //jika pilih = 2
57 case 2:
58     //isi pilihan bernilai 2 ketika dijalankan
59
60     System.out.print("Masukan password lama = ");
61     String old_password = br.readLine();
62     System.out.print("Masukan password baru = ");
63     String new_password = br.readLine();
64
65     System.out.println();
66
67     if (old_password.equals(user1.getPassword()))
68     {
69         user1.setPassword(new_password);
70         System.out.println("*** Password berhasil dirubah ***");
71     }
72     else
73     {
74         System.out.println("*** Anda salah memasukkan password lama ***");
75     }
76     break;

```

h. Mengisi Case 3 (Buat User)

Ketika user memilih inputan menu no. 3, maka dilakukan beberapa proses sebagai berikut:

- Line 82-85: berisi permintaan inputan username dan password yang nantinya akan diisi oleh user
- Line 87: merupakan proses mentransfer data username pada variabel **create_user** dan password pada **create_password** ke dalam constructor Login. Setelah anda membuat username dan password baru, program akan mencetak tulisan ***** User telah berhasil dibuat ***** (line 91). Untuk dapat mengetahui apakah username dan password yang anda buat berhasil masuk ke dalam class Login, anda dapat memilih **case 4** untuk melihat **data user**.

```
78 //jika pilih = 3
79 case 3:
80     //isi pilihan bernilai 3 ketika dijalankan
81
82     System.out.print("Masukan username = ");
83     String create_user = br.readLine();
84     System.out.print("Masukan password = ");
85     String create_password = br.readLine();
86
87     user2 = new Login(create_user, create_password);
88
89     System.out.println();
90
91     System.out.println("*** User telah berhasil dibuat ***");
92     break;
```

i. Mengisi Case 4 (Lihat Data User)

Ketika user memilih inputan menu no. 4, maka dilakukan beberapa proses sebagai berikut:

- Line 102-103: mencetak isi data username dan password yang sudah ada pada variabel **user 1**. Apabila anda melakukan perubahan password **admin** dari nilai *default*-nya adalah **12345** pada **case 2**, berarti data user yang ditampilkan adalah username **admin** dengan **password yang sudah anda rubah**.
- Line 108-109: mencetak isi data username dan password yang sudah anda buat pada **case 3** pada variabel **user 2**.

```
94         //jika pilih = 4
95     case 4:
96         //isi pilihan bernilai 4 ketika dijalankan
97
98         System.out.println("=====");
99         System.out.println("DATA USER");
100
101         System.out.println("User 1");
102         System.out.println("Username = "+user1.getUsername());
103         System.out.println("Password = "+user1.getPassword());
104
105         System.out.println();
106
107         System.out.println("User 2");
108         System.out.println("Username = "+user2.getUsername());
109         System.out.println("Password = "+user2.getPassword());
110
111         System.out.println("=====");
112         break;
```