

Modul Dasar Komputer dan Pemrograman

BAB VI

Class dan Constructor

1. Tujuan Praktikum

1. Praktikkan mampu memahami konsep Pemrograman Berorientasi Objek.
2. Praktikkan mampu memahami penggunaan *Class* dan *Constructor*.
3. Praktikkan mampu mengimplementasikan penggunaan *Class* dan *Constructor*.

2. Dasar Teori *Class* dan *Constructor*

2.1. Pemrograman Berbasis Objek

Pemrograman berorientasi objek (Object Oriented Programming atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah entitas yang memiliki atribut, dan perilaku (behaviour).

Ide dasar pada OOP adalah menyelesaikan suatu masalah bukan menggunakan perintah program tetapi dengan membuat object-object yang saling berinteraksi. OOP memiliki konsep dasar yang lain yaitu Class, Constructor, Object, Inheritance, Encapsulaton, Polymorphism dan Abstraction.

2.2. *Class*

Class adalah cetak biru (rancangan) atau prototype dari objek. Kita bisa membuat banyak objek dari satu macam class. Class mendefinisikan sebuah tipe dari objek.

Di dalam class kita dapat mendeklarasikan variabel dan menciptakan objek (instansiasi). Sebuah class mempunyai anggota yang terdiri dari atribut dan method.

Atribut adalah semua field identitas yang kita berikan pada suatu class, missal class manusia memiliki field Atribut berupa nama dan umur. Method dapat kita

artikan sebagai semua fungsi ataupun prosedur yang merupakan perilaku (behavior) dari suatu class. Berikut bentuk umum dari penggunaan class.

```
[modifier] class <nama_class> {  
  
    <class body>  
  
}
```

Keterangan :

- 1) Modifier ini hampir sama seperti enkapsulasi pada sebuah method/function. Namun jarak jangkauan aksesnya berada pada package. Umumnya ada 3 jenis Modifier :
 - (1) Kosong / default / not specified : dapat diakses oleh class lain tpi dalam satu package.
 - (2) public : Dapat dipakai di class manapun atau package lain.
 - (3) Private : Tidak bisa dipakai oleh oleh class manapun.
- 2) <class_name> bisa diganti dengan nama class yang diinginkan.
- 3) <class_body> ini biasanya berisi variabel,method,function, constructor maupun data dari class lain.

Adapun cara untuk mengakses class di class yang lain (object) dapat menggunakan format berikut.

```
<nama_kelas> <nama_objek> = new <nama_kelas>(<parameter>);
```

Keterangan:

- (1) <nama_kelas> ini merupakan nama dari class lain yang mempunyai modifier public maupun default.
- (2) <nama_objek> ini sebenarnya seperti variabel bebas yang mewakili class yang akan dipakai.
- (3) <parameter> yang ada dalam new <nama_class> ini bersifat opsional. Dan parameter ini nanti akan berhubungan dengan sebuah constructor.

2.3. Constructor

Tipe khusus method yang digunakan untuk menginstansiasi atau menciptakan sebuah objek. Nama constructor = nama kelas. Constructor TIDAK BISA mengembalikan nilai. Tanpa membuat constructor secara eksplisit-pun, Java akan menambahkan constructor default secara implisit. Tetapi jika kita sudah mendefinisikan minimal sebuah constructor, maka Java tidak akan menambah constructor default. Constructor default tidak punya parameter. Constructor bisa digunakan untuk membangun suatu objek, langsung mengeset atribut-atributnya. Konstruktor seperti ini harus memiliki parameter masukan untuk mengeset nilai atribut. Access Modifier constructor selayaknya adalah public, karena constructor akan diakses di luar kelasnya.

Cara panggil constructor adalah dengan menambah keyword “new”. Keyword new dalam deklarasi ini artinya kita mengalokasikan pada memory sekian blok memory untuk menampung objek yang baru kita buat.

```
<modifier> <nama_class> (<parameter>) {  
    <body_constructor>  
}
```

Keterangan :

- (1) <modifier> ini bisa menggunakan salah satu dari 3 modifier yang ada.
- (2) <nama_class> merupakan bagian nama sebuah class dipakai karena nanti saat dipanggil dari class lain akan langsung otomatis dijalankan.
- (3) <parameter> disini menyesuaikan dari object class tersebut dipakai oleh di class lain.

Contoh

Main Class

```
userService dataLogin = new userService(username,password);
```

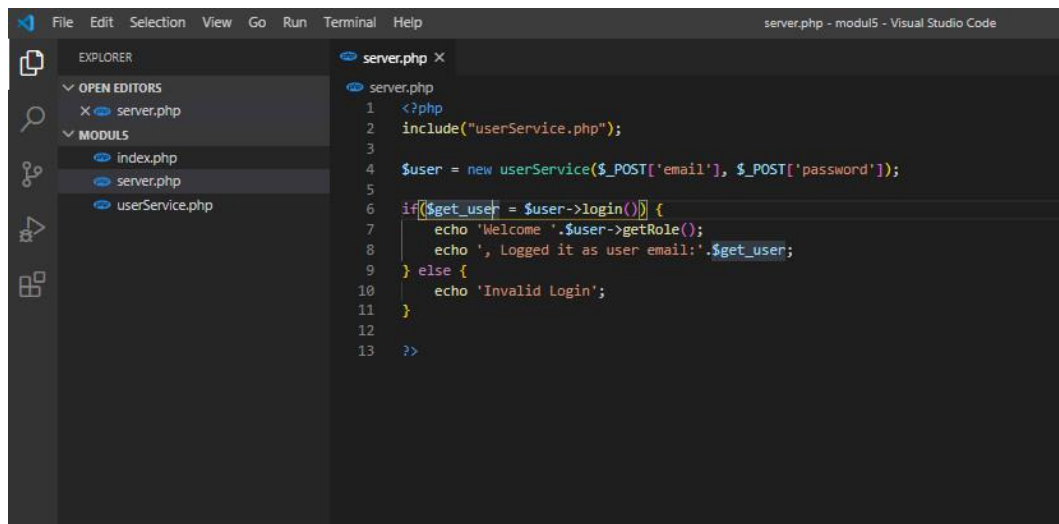
Class userService

```
class userService {  
    public userService(String username, String Password) {}  
}
```

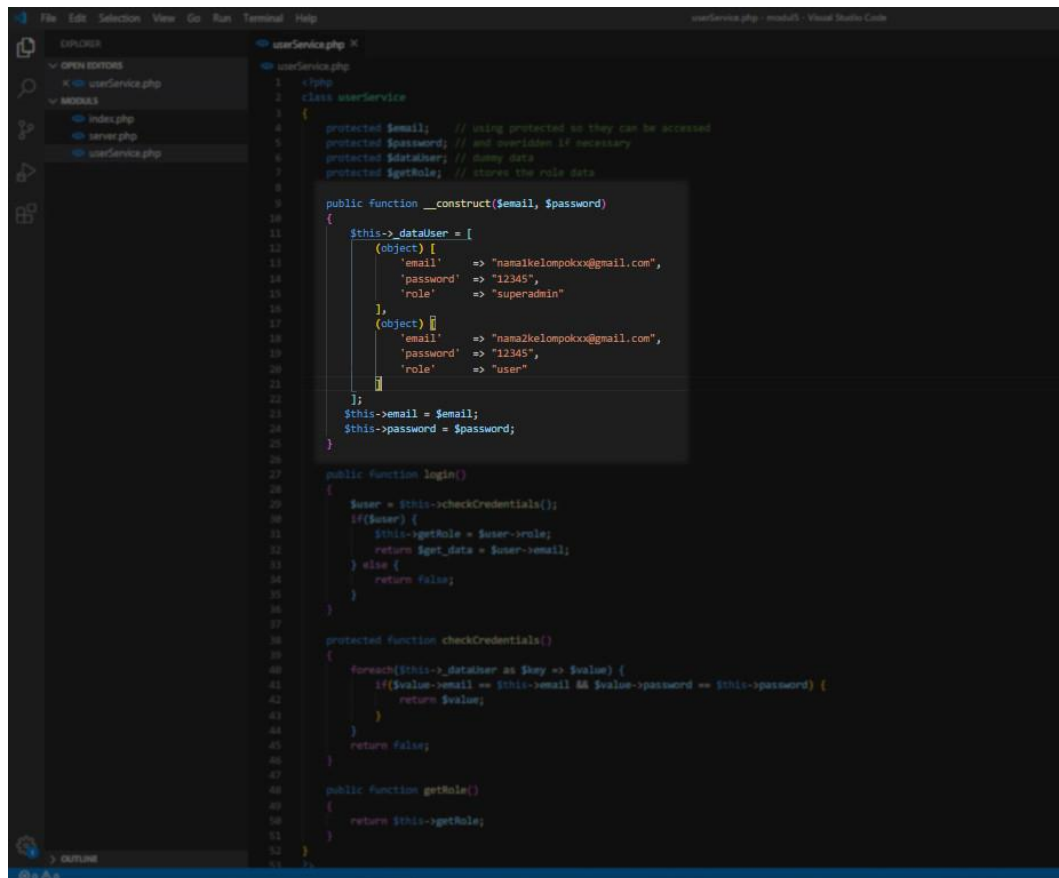
3. Contoh Program Class dan Constructor

3.1. PHP

Penggunaan Class dan Constructor pada bagian PHP cukup mudah. Disini fokuskan pada file server.php dan userService.php. File index.php hanyalah sebagai tampilan input oleh php agar bisa memberikan data.



Include("userService.php") merupakan perintah untuk memanggil berkas, bukan memanggil class. Dan pada bagian \$user disini baru memanggil class dari userService(). Parameter ini diambil dari input yang ada pada index.php. \$user->login() ini merupakan cara akses method/function yang ada pada class userService.



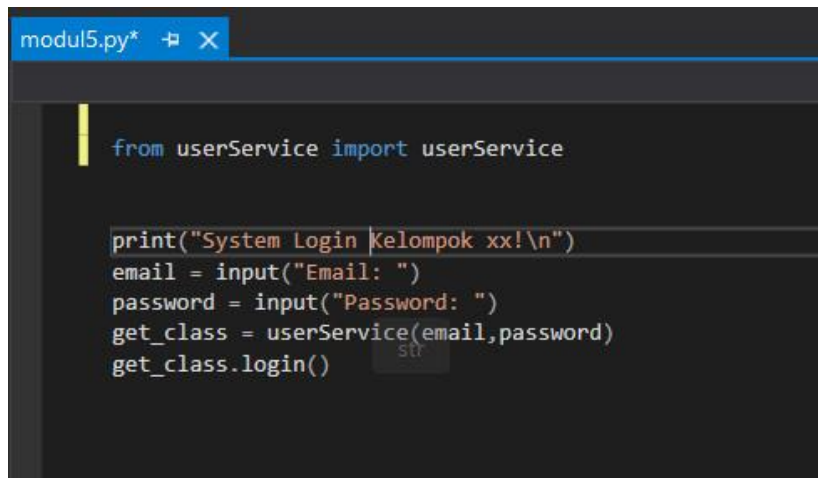
```
1 <?php
2 class userService
3 {
4     protected $email; // using protected so they can be accessed
5     protected $password; // and overridden if necessary
6     protected $dataUser; // dummy data
7     protected $getRole; // stores the role data
8
9     public function __construct($email, $password)
10     {
11         $this->dataUser = [
12             (object) [
13                 'email' => "namaKelompokxx@gmail.com",
14                 'password' => "12345",
15                 'role' => "superadmin"
16             ],
17             (object) [
18                 'email' => "namaKelompokxx@gmail.com",
19                 'password' => "12345",
20                 'role' => "user"
21             ]
22         ];
23         $this->email = $email;
24         $this->password = $password;
25     }
26
27     public function login()
28     {
29         $user = $this->checkCredentials();
30         if($user) {
31             $this->getRole = $user->role;
32             return $get_data = $user->email;
33         } else {
34             return false;
35         }
36     }
37
38     protected function checkCredentials()
39     {
40         foreach($this->dataUser as $key => $value) {
41             if($value->email == $this->email && $value->password == $this->password) {
42                 return $value;
43             }
44         }
45         return false;
46     }
47
48     public function getRole()
49     {
50         return $this->getRole;
51     }
52 }
```

Untuk mengimplementasikan constructor pada PHP adalah seperti pada spotlight diatas. Constructor tersebut akan dijalankan saat class dari userService dipanggil. Yang menjadi poin disini bukan masalah pembuatan array namun pada penggunaan \$this. \$this ini menunjuk pada class ini lalu tanda “->” adalah untuk mengakses kedalam class. Lalu dalam format penggunaan constructor pada PHP selalu diawali dengan <modifier> __construct().

Constructor ini hanya bisa dilakukan apabila sudah berada pada suatu class tertentu. Disini pada line 2 sudah dibuat class userService dengan modifier default. Pada praktikum ini menggunakan constructor berparameter. Namun ada juga bentuk constructor normal tanpa parameter.

3.2. Python

Penggunaan class dan constructor pada Python hampir sama seperti halnya pada PHP namun yang agak berbeda saat penggunaan constructor.

A screenshot of a code editor window titled 'modul5.py*'. The code is written in Python and includes a line to import 'userService' from the 'userService' module. It then prints a login message, prompts for email and password, and creates an instance of the 'userService' class named 'get_class'. Finally, it calls the 'login()' method on 'get_class'.

```
modul5.py*  
  
from userService import userService  
  
print("System Login Kelompok xx!\n")  
email = input("Email: ")  
password = input("Password: ")  
get_class = userService(email,password)  
get_class.login()
```

From userService import userService disini merupakan memanggil berkas sekaligus memanggil class yang ada pada userService. Lalu membuat objek baru bernama get_class untuk menjalankan constructor dengan menggunakan nama class tersebut.

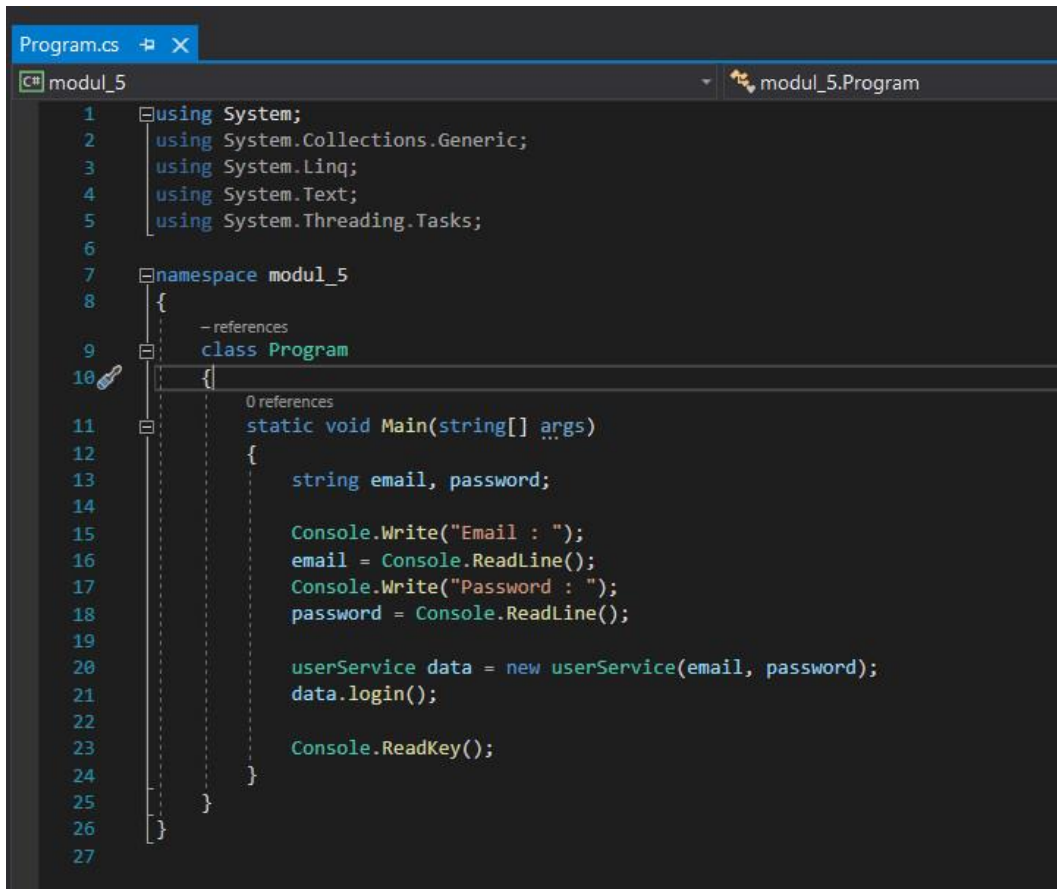
```
userService.py  [X]

class userService(object):
    """description of class"""
    def __init__(self, email, password):
        self.email = email
        self.password = password
        self.data = {
            "nama1kelompokxx@gmail.com" : {
                "email" : "nama1kelompokxx@gmail.com",
                "password" : "12345",
                "role" : "superadmin"
            },
            "nama2kelompokxx@gmail.com" : {
                "email" : "nama2kelompokxx@gmail.com",
                "password" : "12345",
                "role" : "user"
            }
        }
    def login(self):
        get_data = self.checkCredentials()
        if get_data:
            print("\nWelcome ", get_data['role'])
            print("logged it as user email: ", get_data['email'])
        else:
            print("\nInvalid Login!\n")

    def checkCredentials(self):
        for value in self.data:
            if value == self.email:
                get_data_user = self.data[value]
                if self.password == get_data_user['password']:
                    return get_data_user
                else:
                    return False
```

Dari bahasa Python cukup unik dalam pembuatan constructor yaitu dengan memberi nama fungsinya adalah `__init__` dengan parameter yang harus ada `self`. Parameter `self` ini seperti halnya `this` pada bahasa PHP. Jadi `self` ini harus di berikan disetiap parameter function/method yang memerlukan akses ke dalam constructor di class tersebut.

3.3. C#



```
Program.cs
modul_5
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace modul_5
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             string email, password;
14
15             Console.Write("Email : ");
16             email = Console.ReadLine();
17             Console.Write("Password : ");
18             password = Console.ReadLine();
19
20             userService data = new userService(email, password);
21             data.login();
22
23             Console.ReadKey();
24         }
25     }
26 }
27
```

Dalam bahasa C# sendiri dalam membuat objek class sendiri cukup mudah yaitu seperti `userService data = new userService(email, password)`. Nilai dari parameter didapatkan dari inputan user.

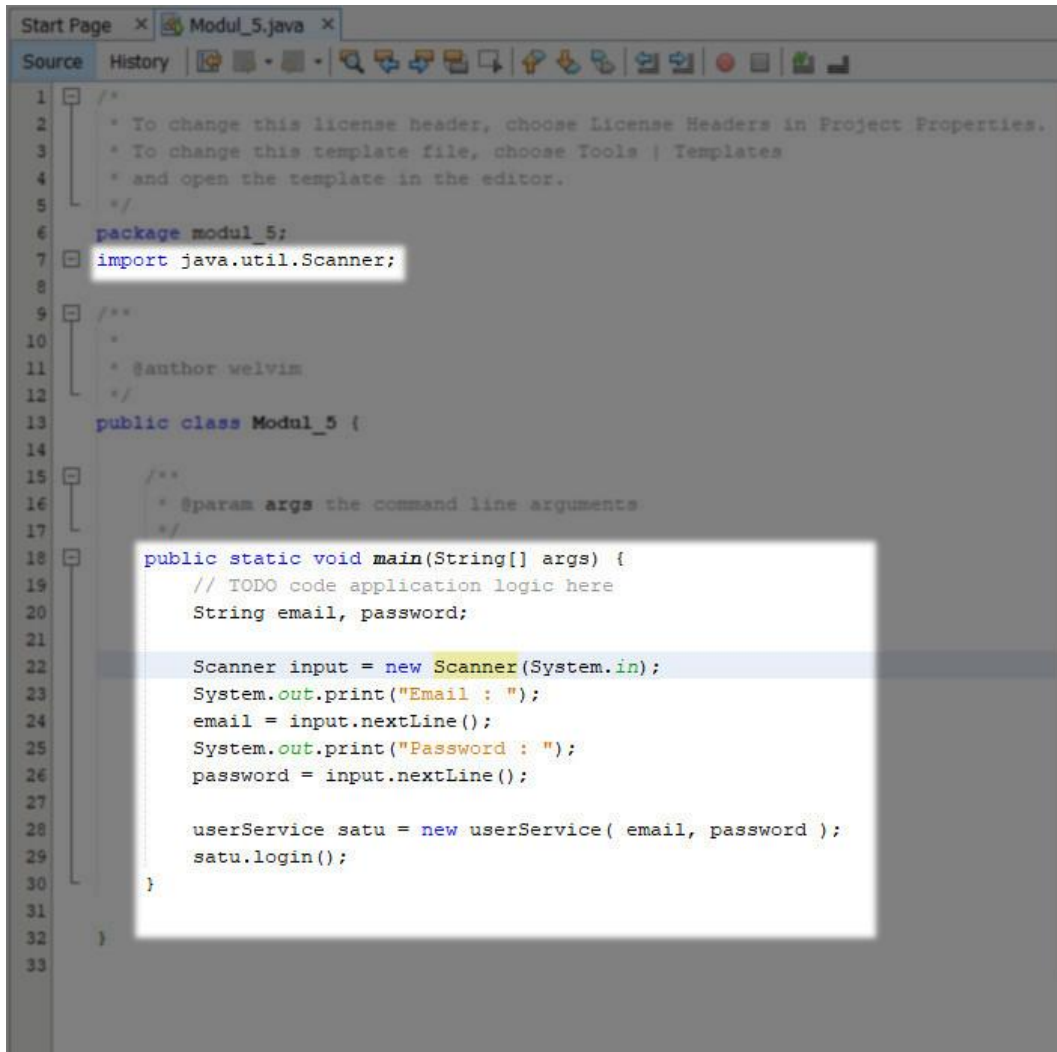
Constructor akan otomatis dijalankan saat objek pada class `userService` dibuat lalu untuk memanggil method `login()` cukup dengan `data.login()` seperti pada spotlight diatas.


```
userService.cs [X]
modul_5 modul_5.userService

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace modul_5
8 {
9     1 reference
10     class userService
11     {
12         private string[,] data;
13         private string email, password, roles = "";
14
15         1 reference
16         public userService(string emails, string passwords)
17         {
18             email = emails;
19             password = passwords;
20             data = new string[2, 3] {
21                 {"nama1kelompokxx@gmail.com", "12345", "superadmin" },
22                 {"nama2kelompokxx@gmail.com", "12345", "user" }
23             };
24
25         1 reference
26         public void login()
27         {
28             var (status, role) = checkCredentials();
29             if (status == true)
30             {
31                 Console.WriteLine("\nwelcome " + role);
32                 Console.WriteLine("logged it as user email: " + email);
33             }
34             else
35             {
36                 Console.WriteLine("\ninvalid login");
37             }
38
39         1 reference
40         private (bool, string) checkCredentials()
41         {
42             for (int i = 0; i < data.GetLength(0); i++)
43             {
44                 if (data[i, 0] == email && data[i, 1] == password)
45                 {
46                     roles = data[i, 2];
47                     return (true, roles);
48                 }
49             }
50             return (false, roles);
51         }
52     }
53 }
```

Pada constructor terdapat deklarasi 3 variabel yaitu email dan password yang nilai nya diambil dari parameter constructor. Dan variabel yang ketiga yaitu data berbentuk array 2x3 yang nilai sudah ditetapkan.

3.4. Java



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package modul_5;
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author welvim
12   */
13  public class Modul_5 {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          String email, password;
21
22          Scanner input = new Scanner(System.in);
23          System.out.print("Email : ");
24          email = input.nextLine();
25          System.out.print("Password : ");
26          password = input.nextLine();
27
28          userService satu = new userService( email, password );
29          satu.login();
30      }
31  }
32
33  }
```

Tidak jauh berbeda dengan C# untuk membuat objek class pada Java cukup dengan `userService satu = new userService(email, password)` dan nilai dari email dan password didapatkan dari inputan user

Constructor akan otomatis dijalankan saat objek pada class `userService` dibuat lalu untuk memanggil method `login()` pada class `userService` cukup dengan `satu.login()` seperti pada spotlight diatas.

```
//ini namanya constructor yang akan dijalankan setiap class diinisialisasikan  
public userService(String emails, String passwords)
```

```
{  
    email = emails;  
    password = passwords;  
    String[][] data =  
    {  
        {"nama1kelompokxx@gmail.com", "12345", "supedAdmin"},  
        {"nama2kelompokxx@gmail.com", "12345", "user"}  
    };  
    this.data = data;  
}
```

```
private boolean checkCredential()  
{  
    for(int i = 0; i < data.length; i++ )  
    {  
        if(data[i][0].equals(email))  
        {  
            if(data[i][1].equals(password))  
            {  
                roles = data[i][2];  
                return true;  
            }  
        }  
    }  
    return false;  
}
```

```
public void login()  
{  
    boolean status = checkCredential();  
    if(status == true)  
    {  
        System.out.println("\nWelcome " + roles);  
        System.out.println("Logged it as user email " + email);  
    }  
    else  
    {  
        System.out.println("\nInvalid Login ");  
    }  
}
```

Pada class userService.java terdapat 2 method yaitu checkCredential untuk melihat apakah inputan yang diberikan user sama dengan data pada constructor. Method login akan memberikan output berupa teks apabila method checkCredential() bernilai true.