

Nama	: Cahyo Devi putro
NBI	: 1461900333
Kelas	: J4
Praktikum	: Pemrograman Berorientasi Objek
Tugas	: pendahuluan ke-3

2. Application programming Interface (API) merupakan sebuah perangkat lunak yang dapat menerima panggilan atau perintah dari perangkat lunak lainnya seperti aplikasi dan website yang memberikan pelayanan. Dengan kata lain API merupakan sebuah perangkat lunak yang mengizinkan atau menghubungkan dua aplikasi. Tidak hanya itu API juga digunakan untuk membuat software dan aplikasi oleh para developer.
3. Container adalah jenis komponen yang menampung dan mengelola komponen-komponennya. Ideanya suatu komponen harus diletakkan pada sebuah container agar ia dapat digunakan.
3. + Event handling merupakan konsep penanganan suatu action yang terjadi. Jadi suatu program akan berjalan saat sesuatu terjadi, misalnya saat tombol diklik, saat combo box dipilih, dan sebagainya.
- + Event listener digunakan untuk menangkap event yang terjadi pada komponen / even-source.
- + Event Object dimana ketika sebuah event terjadi (misal, ketika user berinteraksi dengan Komponen GUI), sebuah objek event diciptakan. Objek berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah terjadi, seperti ketika mouse di klik.
4. a. Radio Button adalah suatu tombol yang digunakan untuk mengambil salah satu pilihan dari banyak pilihan yang tersedia.
- b. Combo Box adalah Control yang digunakan untuk memasukkan data dengan cara memilih nilai dari daftar pilihan yang tersedia.
- c. Text area adalah sebuah komponen swing yang bertugas untuk menampung inputan berupa string yang cukup panjang.

- g. spinner merupakan komponen Java yang digunakan sebagai input field yang memperbolehkan pengguna (user) untuk memilih sebuah nilai angka atau objek dari daftar yang terdapat.
- h. Text pane merupakan komponen yang digunakan untuk menampilkan text dan membolehkan user untuk mengeditnya.
- i. a. CardLayout merupakan layout manager yang mampu menciptakan efek "tampakkan" komponen. Artinya, layout ini tidak memposisikan komponen di lokasi-lokasi tertentu dalam kontainer. melainkan menampilkan satu demi satu. Contoh:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

```
public class Card extends JPanel {
    CardLayout carts = new CardLayout();
    public Card() {
        setLayout(carts);
```

```
        ActionListener lisfer = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                carts.next(card.this);
            }
        };
    }
```

```
    JButton button;
    button = new JButton("One");
    button.addActionListener(lisfer);
    add(button, "One");
}
```

```
public static void main(String[] args) {
    JFrame frame = new JFrame("Card");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane(new card());
    frame.setVisible(true);
}
```

```
Y
```

b. GridLayout merupakan salah satu dari sekian banyak layout yang ada pada Java. GridLayout berfungsi untuk mengatur komponen-komponen yang ada pada sebuah container (JPanel, JFrame, dll) layaknya sebuah tabel / array dengan 2 dimensi. Contoh :

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
public class Grid extends JFrame {  
    public Grid() {  
        CreateUserInterface();  
    }  
  
    public void CreateUserInterface() {  
        JPanel panel1 = new JPanel();  
        panel1.setLayout(new GridLayout(3,2));  
        panel1.add(new JButton("One"));  
        this.add(panel1);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(200,200);  
        this.setLocation(200,200);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        Grid app = new Grid();  
    }  
}
```

c. GridBagLayout adalah standart layout manager . Dengan tata letak platform java . Yang kompleks dan fleksibel . GridBagLayout menempatkan dan menyingkatkan komponen tersebut untuk mengatur beberapa grid baris dan kolom . Tidak semua baris harus memiliki ketinggian yang sama . Demikian pula , tidak semua kolom harus memiliki lebar yang sama . Contoh :

```
import java.awt.GridBagConstraints;  
import java.awt.GridBagLayout;
```

```

import javax.swing.JButton;
import javax.swing.JFrame;

public class GBL extends JFrame {
    public GBL() {
        setLayout(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        c.gridx = 0;
        c.gridy = 0;
        c.fill = GridBagConstraints.VERTICAL;
        add(new JButton("0,0"), c);

        c.gridx = 1;
        c.gridy = 0;
        add(new JButton("1,0"), c);
        setSize(600, 300);
        setVisible(true);
    }
}

public static void main(String[] args) {
    new GBL();
}

```

8. FlowLayout digunakan untuk menata letak komponen-komponen dalam satu baris dari kiri ke kanan, jika baris tersebut sudah penuh maka komponen-komponen akan dilepas dalam baris berikutnya. Contoh :

```

import java.awt.FlowLayout;
import java.awt.Container;
import javax.swing.JFrame;
import javax.swing.JButton;

public class Flow extends JFrame {
    JButton tombol1, tombol2;

```

```
public Flow(string judul) {
```

```
super(judul);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setVisible(true);
```

```
Container tempat = getContentPane();
```

```
tempat.setLayout(new FlowLayout());
```

```
tombol1 = new JButton("Java");
```

```
tombol2 = new JButton("PHP");
```

```
tempat.add(tombol1);
```

```
tempat.add(tombol2);
```

```
}
```

```
private void () {
```

```
setSize(400,100);
```

```
}
```

```
public static void main (String[] args) {
```

```
Flow f1 = new Flow("flowLayout");
```

```
f1.flow();
```

```
}
```

Q. BorderLayout merupakan layout default untuk Jwindow dan JFrame, karena setiap komponen diasosiasikan dengan satu arah geografis, akibatnya layout ini hanya dapat menampung maksimal 5 komponen. BorderLayout juga berfungsi untuk menyusun komponen berdasarkan lokasi geografis : North, South, East, West, dan Center. Contoh:

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import javax.swing.*;
```

```
public class BorderLayout extends JFrame {
```

```
public BorderLayout () {
```

```
createUserInterface();
```

```
}
```

```
private void createUserInterface () {
```

```
this.setTitle ("Border");
```

```
this.setSize(300, 200);  
this.setLocation(200, 200);  
this.setLayout(new BorderLayout());  
this.add(new JButton("North"), BorderLayout.NORTH);  
this.add(new JButton("South"), BorderLayout.SOUTH);  
this.add(new JButton("East"), BorderLayout.EAST);  
this.add(new JButton("West"), BorderLayout.WEST);  
this.add(new JButton("Center"), BorderLayout.CENTER);  
this.setVisible(true);
```

Y public static void main (String [] args) {

BorderLayout app = new BorderLayout();

app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

Y