

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : M. Akbar Amiruddin
Nim : 13020220023

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2024

1. Buat aplikasi bahasa Java memasukkan dan menampilkan nim, nama, jurusan dan fakultas anda, data tersebut dimasukkan melalui keyboard!

```
J Scan.java X
C: > Users > ASUS > Documents > PBO > Tugas 2 > Tugas2_SourceCode > J Scan.java > ..
1  import java.util.Scanner;
2
3  public class Scan{
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          System.out.println("Masukkan NIM:");
8          String nim = scanner.nextLine();
9
10         System.out.println("Masukkan nama:");
11         String nama = scanner.nextLine();
12
13         System.out.println("Masukkan jurusan:");
14         String jurusan = scanner.nextLine();
15
16         System.out.println("Masukkan fakultas:");
17         String fakultas = scanner.nextLine();
18
19         System.out.println("\n");
20
21         System.out.println("Data yang dimasukkan: ");
22         System.out.println("NIM: " + nim);
23         System.out.println("Nama: " + nama);
24         System.out.println("Jurusan: " + jurusan);
25         System.out.println("Fakultas: " + fakultas);
26
27         scanner.close();
28     }
29 }
30
```

Coding tersebut merupakan contoh program sederhana dalam bahasa pemrograman Java yang menggunakan kelas Scanner untuk menerima input dari pengguna melalui keyboard dan menampilkannya kembali ke layar. Mari kita bahas secara detail:

1. `import java.util.Scanner;`: Baris ini mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
2. `public class Scan{`: Ini adalah deklarasi kelas utama yang disebut Scan. Nama kelas harus sesuai dengan nama file Java tempat kode ini disimpan.

3. `public static void main(String[] args) {`: Ini adalah metode utama yang akan dieksekusi ketika program dimulai. Semua kode yang ingin dieksekusi harus berada di dalam metode main.
4. `Scanner scanner = new Scanner(System.in);`: Membuat objek Scanner yang bernama 'scanner'. Objek ini akan digunakan untuk membaca input dari pengguna melalui keyboard (System.in).
5. Kemudian program meminta pengguna untuk memasukkan NIM, nama, jurusan, dan fakultas satu per satu menggunakan metode `scanner.nextLine()`. Input dari pengguna disimpan dalam variabel yang sesuai: nim, nama, jurusan, dan fakultas.
6. Setelah menerima input dari pengguna, program mencetak pesan baru dan menampilkan data yang dimasukkan oleh pengguna ke layar dengan menggunakan perintah `System.out.println()`.
7. Terakhir, metode `scanner.close()` digunakan untuk menutup objek Scanner setelah selesai digunakan. Ini adalah praktik yang baik untuk membersihkan sumber daya setelah selesai digunakan.

Program ini memungkinkan pengguna untuk memasukkan data NIM, nama, jurusan, dan fakultas, dan kemudian menampilkan data tersebut ke layar.

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>javac Scan.java

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>java Scan
Masukkan NIM:
13020220023
Masukkan nama:
M. Akbar Amiruddin
Masukkan jurusan:
Teknik Informatika
Masukkan fakultas:
Ilmu Komputer

Data yang dimasukkan:
NIM: 13020220023
Nama: M. Akbar Amiruddin
Jurusan: Teknik Informatika
Fakultas: Ilmu Komputer

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>
```

2. Menggunakan Class BufferedReader

```
> Users > ASUS > Documents > PBO > Tugas 2 > Tugas2_SourceCode > BuffR.java > BuffR > main()
1  import java.io.BufferedReader;
2  import java.io.IOException;
3  import java.io.InputStreamReader;
4
5  public class BuffR{
6      public static void main(String[] args) throws IOException {
7          BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
8
9          System.out.print("Masukkan NIM: ");
10         String nim = reader.readLine();
11
12         System.out.print("Masukkan nama: ");
13         String nama = reader.readLine();
14
15         System.out.print("Masukkan jurusan: ");
16         String jurusan = reader.readLine();
17
18         System.out.print("Masukkan fakultas: ");
19         String fakultas = reader.readLine();
20
21         System.out.println("\n");
22
23         System.out.println("Data yang dimasukkan:");
24         System.out.println("NIM: " + nim);
25         System.out.println("Nama: " + nama);
26         System.out.println("Jurusan: " + jurusan);
27         System.out.println("Fakultas: " + fakultas);
28
29         reader.close();
30     }
31 }
```

Coding tersebut juga merupakan contoh program sederhana dalam bahasa pemrograman Java yang mengambil input dari pengguna dan menampilkannya kembali ke layar. Akan tetapi, dalam contoh ini, penggunaan kelas `BufferedReader` digunakan untuk membaca input dari pengguna.

1. `import java.io.BufferedReader;`: Baris ini mengimpor kelas `BufferedReader` dari paket `java.io`. `BufferedReader` digunakan untuk membaca teks dari input karakter, seperti keyboard.
2. `import java.io.IOException;`: Ini mengimpor kelas `IOException` dari paket `java.io`. `IOException` adalah pengecualian yang dapat terjadi selama operasi input/output (I/O) dalam Java.
3. `import java.io.InputStreamReader;`: Mengimpor kelas `InputStreamReader` dari paket `java.io`. `InputStreamReader` menghubungkan aliran input byte ke aliran input karakter.

4. `public class BuffR{``}```: Ini adalah deklarasi kelas utama yang disebut BuffR. Nama kelas harus sesuai dengan nama file Java tempat kode ini disimpan.`
5. `public static void main(String[] args) throws IOException {``}```: Ini adalah metode utama yang akan dieksekusi ketika program dimulai. Throws IOException menunjukkan bahwa metode ini dapat menghasilkan IOException dan mengharuskan pemanggilannya untuk menanganinya.`
6. `BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));``}```: Membuat objek BufferedReader yang bernama 'reader'. Objek ini menggunakan InputStreamReader untuk membaca input dari pengguna melalui keyboard (System.in).`
7. Program meminta pengguna untuk memasukkan NIM, nama, jurusan, dan fakultas satu per satu menggunakan metode `reader.readLine()`````. Input dari pengguna disimpan dalam variabel yang sesuai: nim, nama, jurusan, dan fakultas.`
8. Setelah menerima input dari pengguna, program mencetak pesan baru dan menampilkan data yang dimasukkan oleh pengguna ke layar dengan menggunakan perintah `System.out.println()`````.`
9. Terakhir, metode `reader.close()````` digunakan untuk menutup objek BufferedReader setelah selesai digunakan. Ini adalah praktik yang baik untuk membersihkan sumber daya setelah selesai digunakan.`

Program ini memungkinkan pengguna untuk memasukkan data NIM, nama, jurusan, dan fakultas, dan kemudian menampilkan data tersebut ke layar menggunakan BufferedReader.

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>javac BuffR.java

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>java BuffR
Masukkan NIM: 13020220023
Masukkan nama: M. Akbar Amiruddin
Masukkan jurusan: Teknik Informatika
Masukkan fakultas: Ilmu Komputer

Data yang dimasukkan:
NIM: 13020220023
Nama: M. Akbar Amiruddin
Jurusan: Teknik Informatika
Fakultas: Ilmu Komputer

C:\Users\ASUS\Documents\PBO\Tugas 2\Tugas2_SourceCode>
```

3. Menggunakan Class JOptionPane

```
C: > Users > ASUS > Documents > PBO > Tugas 2 > Tugas2_SourceCode > J JOpti.java > ...
1  import javax.swing.JOptionPane;
2
3  public class JOpti{
4      public static void main(String[] args) {
5          String nim = JOptionPane.showInputDialog("Masukkan NIM:");
6          String nama = JOptionPane.showInputDialog("Masukkan nama:");
7          String jurusan = JOptionPane.showInputDialog("Masukkan jurusan:");
8          String fakultas = JOptionPane.showInputDialog("Masukkan fakultas:");
9
10         String message = "Data yang dimasukkan:\n" +
11             "NIM: " + nim + "\n" +
12             "Nama: " + nama + "\n" +
13             "Jurusan: " + jurusan + "\n" +
14             "Fakultas: " + fakultas;
15
16         JOptionPane.showMessageDialog(null, message);
17     }
18 }
19
```

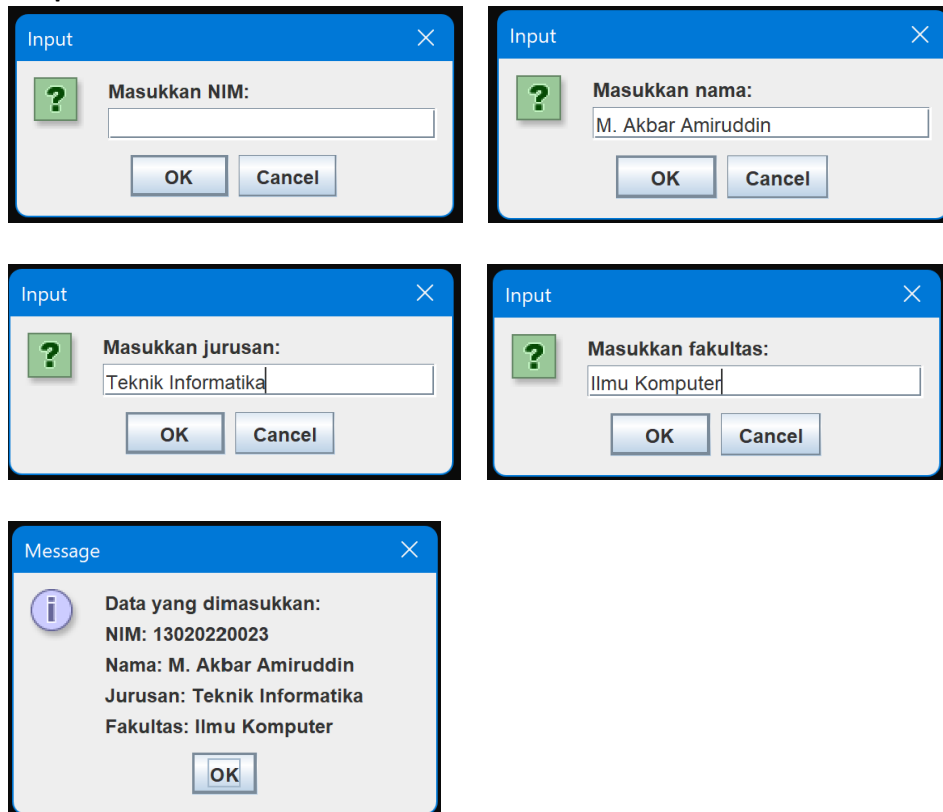
Coding tersebut adalah contoh program sederhana dalam bahasa pemrograman Java yang menggunakan JOptionPane dari paket javax.swing untuk menampilkan dialog input dan pesan kepada pengguna.

1. `import javax.swing.JOptionPane;`: Baris ini mengimpor kelas JOptionPane dari paket javax.swing. JOptionPane digunakan untuk menampilkan dialog input dan pesan kepada pengguna dalam bentuk kotak dialog.
2. `public class JOpti{`: Ini adalah deklarasi kelas utama yang disebut JOpti. Nama kelas harus sesuai dengan nama file Java tempat kode ini disimpan.
3. `public static void main(String[] args) {`: Ini adalah metode utama yang akan dieksekusi ketika program dimulai.
4. Program menggunakan metode `JOptionPane.showInputDialog()` untuk menampilkan kotak dialog yang meminta pengguna untuk memasukkan NIM, nama, jurusan, dan fakultas. Setiap panggilan metode ini akan mengembalikan string yang dimasukkan oleh pengguna.
5. Data yang dimasukkan oleh pengguna disimpan dalam variabel yang sesuai: nim, nama, jurusan, dan fakultas.
6. Kemudian, program menggabungkan data yang dimasukkan oleh pengguna ke dalam satu string yang ditampilkan dalam kotak dialog menggunakan metode `JOptionPane.showMessageDialog()`.
7. Variabel 'message' digunakan untuk menyimpan pesan yang akan ditampilkan dalam kotak dialog.
8. Metode `JOptionPane.showMessageDialog()` menampilkan kotak dialog dengan pesan yang telah disiapkan, menampilkan data NIM, nama, jurusan, dan fakultas yang dimasukkan oleh pengguna.

- Parameter null dalam ``JOptionPane.showMessageDialog()`` menunjukkan bahwa kotak dialog akan ditampilkan di tengah layar.

Program ini memungkinkan pengguna untuk memasukkan data NIM, nama, jurusan, dan fakultas menggunakan kotak dialog dan kemudian menampilkan data tersebut dalam kotak dialog lainnya.

Output:



4. Buat Program menggunakan bahasa java untuk Konversi Waktu (Jam:Menit:Detik) dari masukan/input detik!

```
C:\Users\ASUS> Documents > PBO > Tugas 2 > Tugas2_SourceCode > J KonversiWaktu.java > ...
1  import java.util.Scanner;
2
3  public class KonversiWaktu {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          System.out.print("Masukkan total detik: ");
8          int totalDetik = scanner.nextInt();
9          int detikSekarang = totalDetik % 60;
10         int totalMenit = totalDetik / 60;
11         int menitSekarang = totalMenit % 60;
12         int totalJam = totalMenit / 60;
13         int jamSekarang = totalJam % 24;
14
15
16         System.out.println("Waktu sekarang: " + jamSekarang + ":" + menitSekarang + ":" + detikSekarang);
17
18         scanner.close();
19     }
20 }
```

Coding tersebut adalah program sederhana dalam bahasa pemrograman Java yang mengkonversi total detik menjadi format jam:menit:detik.

1. `import java.util.Scanner;`: Baris ini mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
2. `public class KonversiWaktu {`: Ini adalah deklarasi kelas utama yang disebut KonversiWaktu. Nama kelas harus sesuai dengan nama file Java tempat kode ini disimpan.
3. `public static void main(String[] args) {`: Ini adalah metode utama yang akan dieksekusi ketika program dimulai. Semua kode yang ingin dieksekusi harus berada di dalam metode main.
4. Program menggunakan objek Scanner untuk meminta pengguna memasukkan total detik.
5. `int totalDetik = scanner.nextInt();`: Meminta pengguna untuk memasukkan total detik dan menyimpannya dalam variabel totalDetik.
6. Program kemudian melakukan konversi total detik ke jam, menit, dan detik.
 - `detikSekarang` adalah sisa detik setelah membagi total detik dengan 60.
 - `totalMenit` adalah hasil dari pembagian total detik dengan 60.
 - `menitSekarang` adalah sisa menit setelah membagi total menit dengan 60.
 - `totalJam` adalah hasil dari pembagian total menit dengan 60.
 - `jamSekarang` adalah sisa jam setelah membagi total jam dengan 24.
7. Setelah melakukan konversi, program mencetak waktu saat ini dalam format jam:menit:detik menggunakan perintah `System.out.println()`.
8. Metode `scanner.close()` digunakan untuk menutup objek Scanner setelah selesai digunakan. Ini adalah praktik yang baik untuk membersihkan sumber daya setelah selesai digunakan.

Program ini memungkinkan pengguna untuk memasukkan total detik dan kemudian menampilkan waktu saat ini dalam format jam:menit:detik.

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PB0\Tugas 2\Tugas2_SourceCode>javac KonversiWaktu.java

C:\Users\ASUS\Documents\PB0\Tugas 2\Tugas2_SourceCode>java KonversiWaktu
Masukkan total detik: 7000
Waktu sekarang: 1:56:40

C:\Users\ASUS\Documents\PB0\Tugas 2\Tugas2_SourceCode>
```