**LAPORAN PRAKTIKUM**

**PEMROGRAMAN II**

**MODUL 6**

****

**Graphical User Interfce (GUI)**

**Oleh:**

**NIZAR ALI NIM. 2210817310012**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**DESEMBER 2023**

# LEMBAR PENGESAHAN

**LAPORAN PRAKTIKUM PEMROGRAMAN II**

**MODUL 6**

Laporan Praktikum Pemrograman II Modul 6 : Grapichal User Interface ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Prakitkum ini dikerjakan oleh:

Nama Praktikan : Nizar Ali

NIM : 2210817310012

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Muhammad Aulia Akbar  NIM. 2010817210023 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Andreyan Rizky Baskara, S.Kom., M.Kom.  NIP. 19930703 201903 1 011 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 1](#_Toc153352739)

[DAFTAR ISI 2](#_Toc153352740)

[DAFTAR GAMBAR 3](#_Toc153352741)

[DAFTAR TABEL 4](#_Toc153352742)

[SOAL 1 5](#_Toc153352743)

[A. Source Code 7](#_Toc153352744)

[B. Output Program 10](#_Toc153352745)

[C. Pembahasan 10](#_Toc153352746)

[a. Mahasiswa 10](#_Toc153352747)

[b. Main 11](#_Toc153352748)

[LINK GIT HUB 11](#_Toc153352749)

# DAFTAR GAMBAR

[output gambar modul 6 soal 1 10](#_Toc153541662)

# DAFTAR TABEL

[source code modul 6 soal 1 7](#_Toc153541637)

[source code modul 6 soal 2 9](#_Toc153541638)

# SOAL 1

Diberikan class diagram seperti berikut: (isi program harus sesuai dengan class diagram)



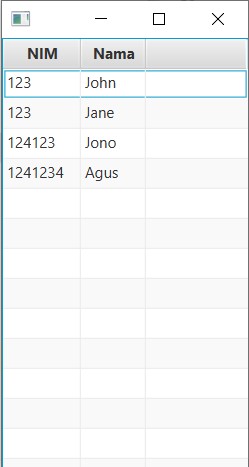
Implementasikan class diagram diatas menjadi class pada Bahasa pemrograman java. Class mahasiswa harus menerapkan setter dan getter.

Program harus menampilkan list data dalam bentuk tabel.

Kolom dari tabel adalah :

* NIM
* Nama

Kemudian buatlah 10 data secara *hardcode* untuk ditampilkan pada tabel Contoh program dapat dilihat sebagai berikut:



## Source Code

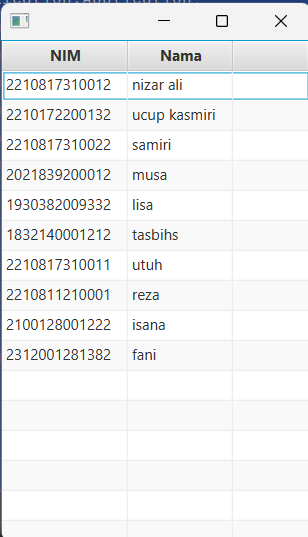
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  25  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package com.example.nizar; import javafx.fxml.FXML; import javafx.scene.control.Label;  public class mahasiswa {  private int id;  private String name;  private String nim;   public mahasiswa(int id, String name, String nim) {  this.id = id;  this.name = name;  this.nim = nim;  }   public int getId() {  return id;  }   public void setId(int id) {  this.id = id;  }   public String getName() {  return name;  }   public void setName(String name) {  this.name = name;  }   public String getNim() {  return nim;  }   public void setNim(String nim) {  this.nim=nim;  } } |

source code modul 6 soal 1

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | package com.example.nizar;  import javafx.application.Application; import javafx.scene.Scene; import javafx.scene.control.TableColumn; import javafx.scene.control.TableView; import javafx.scene.control.cell.PropertyValueFactory; import javafx.scene.layout.VBox; import javafx.stage.Stage;  public class main extends Application {   public static void main(String[] args) {  *launch*(args);  }   @Override  public void start(Stage primaryStage) {   TableView tableView = new TableView();   TableColumn<mahasiswa, String> column1 =  new TableColumn<>("NIM");   column1.setCellValueFactory(  new PropertyValueFactory<>("nim"));    TableColumn<mahasiswa, String> column2 =  new TableColumn<>("Nama");   column2.setCellValueFactory(  new PropertyValueFactory<>("name"));   tableView.getColumns().add(column1);  tableView.getColumns().add(column2);   tableView.getItems().add(  new mahasiswa(1, "nizar ali", "2210817310012"));  tableView.getItems().add(  new mahasiswa(2, "ucup kasmiri", "2210172200132"));  tableView.getItems().add(  new mahasiswa(3, "samiri", "2210817310022"));  tableView.getItems().add(  new mahasiswa(4, "musa", "2021839200012"));  tableView.getItems().add(  new mahasiswa(5, "lisa", "1930382009332"));  tableView.getItems().add(  new mahasiswa(6, "tasbihs", "1832140001212"));  tableView.getItems().add(  new mahasiswa(7, "utuh", "2210817310011"));  tableView.getItems().add(  new mahasiswa(8, "reza", "2210811210001"));  tableView.getItems().add(  new mahasiswa(9, "isana", "2100128001222"));  tableView.getItems().add(  new mahasiswa(10, "fani", "2312001281382"));  VBox vbox = new VBox(tableView);  Scene scene = new Scene(vbox);  primaryStage.setScene(scene);  primaryStage.show();  } } |

source code modul 6 soal 2

## Output Program



output gambar modul 6 soal 1

## Pembahasan

### Mahasiswa

Baris 1: package com.example.nizar; - Menentukan paket tempat kelas mahasiswa berada (bisa diganti sesuai struktur project Anda).

Baris 2: import javafx.fxml.FXML; - Mengimpor library FXML dari JavaFX untuk menghubungkan elemen Java code dengan elemen UI dalam file FXML.

Baris 3: import javafx.scene.control.Label; - Mengimpor library Label dari JavaFX untuk menampilkan teks dalam UI.

Baris 4: public class mahasiswa { - Deklarasi kelas publik bernama mahasiswa.

Baris 5-7: Deklarasi atribut privat:

private int id; - Variabel id bertipe integer untuk menyimpan ID mahasiswa.

private String name; - Variabel name bertipe String untuk menyimpan nama mahasiswa.

private String nim; - Variabel nim bertipe String untuk menyimpan NIM mahasiswa.

Baris 8: public mahasiswa(int id, String name, String nim) { - Deklarasi konstruktor publik dengan tiga parameter:

int id - Parameter untuk menerima dan menyimpan ID mahasiswa.

String name - Parameter untuk menerima dan menyimpan nama mahasiswa.

String nim - Parameter untuk menerima dan menyimpan NIM mahasiswa.

Baris 9-11: Assign nilai parameter ke atribut:

this.id = id; - Mengassign nilai parameter id ke atribut id.

this.name = name; - Mengassign nilai parameter name ke atribut name.

this.nim = nim; - Mengassign nilai parameter nim ke atribut nim.

Baris 12-15: Deklarasi getter untuk masing-masing atribut:

public int getId() { return id; } - Mengembalikan nilai id.

public void setName(String name) { this.name = name; } - Mengubah nilai name dengan parameter name.

public String getNim() { return nim; } - Mengembalikan nilai nim.

public void setNim(String nim) { this.nim = nim; } - Mengubah nilai nim dengan parameter nim.

### main

Baris 1: package com.example.nizar; - Menetapkan paket tempat kelas main berada.

Baris 2: import javafx.application.Application; - Mengimpor class Application dari library JavaFX untuk membuat aplikasi desktop.

Baris 3: import javafx.scene.Scene; - Mengimpor class Scene dari JavaFX untuk menampung elemen UI aplikasi.

Baris 4: import javafx.scene.control.TableColumn; - Mengimpor class TableColumn dari JavaFX untuk membuat kolom dalam sebuah tabel.

Baris 5: import javafx.scene.control.TableView; - Mengimpor class TableView dari JavaFX untuk membuat tabel yang dapat menampilkan data.

Baris 6: import javafx.scene.control.cell.PropertyValueFactory; - Mengimpor class PropertyValueFactory dari JavaFX untuk menghubungkan data model dengan sel tabel.

Baris 7: import javafx.scene.layout.VBox; - Mengimpor class VBox dari JavaFX untuk mengatur tata letak elemen secara vertikal.

Baris 8: import javafx.stage.Stage; - Mengimpor class Stage dari JavaFX untuk mewakili jendela utama aplikasi.

Baris 9: public class main extends Application { - Deklarasi class publik main yang mewarisi class Application.

Baris 10: public static void main(String[] args) { - Deklarasi method main sebagai titik awal eksekusi program.

Baris 11: launch(args); - Memanggil method launch dari class Application untuk memulai aplikasi.

Baris 13: @Override public void start(Stage primaryStage) { - Deklarasi method start yang di-override dari class Application dan menerima objek Stage untuk jendela utama.

Baris 14: TableView tableView = new TableView(); - Menginisialisasi objek tableView bertipe TableView untuk menampilkan data mahasiswa.

Baris 15-17: Deklarasi dua objek TableColumn untuk menampilkan NIM dan Nama:

TableColumn<mahasiswa, String> column1 = new TableColumn<>("NIM"); - Membuat kolom "NIM" untuk data bertipe String dengan label "NIM".

TableColumn<mahasiswa, String> column2 = new TableColumn<>("Nama"); - Membuat kolom "Nama" untuk data bertipe String dengan label "Nama".

Baris 18-19: Mengatur nilai yang ditampilkan dalam sel menggunakan PropertyValueFactory:

column1.setCellValueFactory(new PropertyValueFactory<>("nim")); - Mengatur agar isi sel kolom "NIM" diisi oleh nilai atribut nim dari objek mahasiswa.

column2.setCellValueFactory(new PropertyValueFactory<>("name")); - Mengatur agar isi sel kolom "Nama" diisi oleh nilai atribut name dari objek mahasiswa.

Baris 21: tableView.getColumns().add(column1); - Menambahkan kolom "NIM" ke tabel.

Baris 22: tableView.getColumns().add(column2); - Menambahkan kolom "Nama" ke tabel.

Baris 24-33: Menambahkan sepuluh objek mahasiswa baru ke tableView:

Setiap baris membuat objek mahasiswa baru dengan ID, nama, dan NIM tertentu.

Metode getItems().add() menambahkan objek mahasiswa tersebut ke daftar data yang ditampilkan dalam tabel.

Baris 35: VBox vbox = new VBox(tableView); - Menginisialisasi objek vbox bertipe VBox dan menambahkan tableView ke dalamnya.

Baris 36: Scene scene = new Scene(vbox); - Menginisialisasi objek scene bertipe Scene dengan vbox sebagai kontennya.

Baris 37: primaryStage.setScene(scene); - Mengatur scene sebagai konten dari jendela utama primaryStage.

Baris 38: primaryStage.show(); - Menampilkan jendela utama aplikasi.

# LINK GIT HUB