

# **LAPORAN MODUL (9)**

## **Pemrograman Berorientasi Objek Lanjut**

**“String”**

Dosen Pengampu: Eduardus Hardika Sandy Atmaja, S.Kom., M.Cs.



**DIBUAT OLEH:**

Nama : Yohanis Celvin D. P. U. Pati

Nim : 245314033

**KELAS : BP**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2025**

## A. TUJUAN

1. Mahasiswa mampu memahami dan menggunakan tipe data String untuk mengolah, memanipulasi, serta menampilkan teks dalam program secara efektif.

## B. TUGAS

### Latihan 1

#### • Source code

##### ▪ Penduduk

The screenshot shows a Java code editor with the 'Source' tab selected. The code defines a class named 'Penduduk' with private fields: nik, nama, jenisKelamin, tanggalLahir, noHP, and id. It includes getters and setters for each field. The code is as follows:

```
10  /*
11  * public class Penduduk {
12  *     private String nik;
13  *     private String nama;
14  *     private String jenisKelamin;
15  *     private String tanggalLahir;
16  *     private String noHP;
17  *     private String id;
18  *     private String password;
19  *
20  *     public String getNik() { return nik; }
21  *     public void setNik(String nik) { this.nik = nik; }
22  *
23  *     public String getNama() { return nama; }
24  *     public void setNama(String nama) { this.nama = nama; }
25  *     public String getJenisKelamin() { return jenisKelamin; }
26  *     public void setJenisKelamin(String jenisKelamin) { this.jenisKelamin = jenisKelamin; }
27  *
28  *     public String getTanggalLahir() { return tanggalLahir; }
29  *     public void setTanggalLahir(String tanggalLahir) { this.tanggalLahir = tanggalLahir; }
30  *
31  *     public String getNoHP() { return noHP; }
32  *     public void setNoHP(String noHP) { this.noHP = noHP; }
33  *
34  *     public String getAlamat() { return alamat; }
35  *     public void setAlamat(String alamat) { this.alamat = alamat; }
36  *     public String getId() { return id; }
37  *     public void setId(String id) { this.id = id; }
38  *
39  *     public String getPassword() { return password; }
40  *     public void setPassword(String password) { this.password = password; }
41  * }
```

#### Penjelasan:

1. Getter dan Setter untuk nik **public String getNik() { return nik; }**  
**public void setNik(String nik) { this.nik = nik; }**  
untuk mengambil dan mengubah nilai NIK Penduduk.
2. Getter dan Setter untuk nama **public String getNama() { return nama; }**  
**public void setNama(String nama) { this.nama = nama; }**  
untuk mengambil dan mengubah nama penduduk.
3. Getter dan Setter untuk jenisKelamin **public String getJenisKelamin() { return jenisKelamin; }**  
**public void setJenisKelamin(String jenisKelamin) { this.jenisKelamin = jenisKelamin; }**  
untuk menyimpan atau ambil informasi jenis kelamin.
4. Getter dan Setter untuk tanggalLahir **public String getTanggalLahir() { return tanggalLahir; }**  
**public void setTanggalLahir(String tanggalLahir) { this.tanggalLahir = tanggalLahir; }**  
untuk mengelola data tanggal lahir penduduk.

5. Getter dan Setter untuk nohp **public String getNohp() { return nohp; }**  
**public void setNohp(String nohp) { this.nohp = nohp; }**  
untuk mengatur nomor HP penduduk.
6. Getter dan Setter untuk alamat **public String getAlamat() { return alamat; }**  
**public void setAlamat(String alamat) { this.alamat = alamat; }**  
Untuk mengambil dan mengubah alamat tempat tinggal penduduk
7. Getter dan Setter untuk id **public String getId() { return id; }**  
**public void setId(String id) { this.id = id; }**  
untuk ID unik dalam sistem login.
8. Getter dan Setter untuk password **public String getPassword() { return password; }**  
**public void setPassword(String password) { this.password = password; }**  
untuk mengambil dan mengatur kata sandi.

- PendudukTableModel

## Penjelasan:

- ```
1. Konstruktor public PendudukTableModel(List<Penduduk> dp) {  
    this.daftarPenduduk = dp;  
}
```

Konstruktor ini menerima daftar penduduk dari luar kelas dan disimpan ke atribut daftarPenduduk agar bisa digunakan tabel.

- ## 2. @Override

```
public int getRowCount() {  
    return daftarPenduduk.size();  
}
```

Mengembalikan jumlah baris yang harus ditampilkan di tabel Jumlah baris = jumlah objek Penduduk di dalam ArrayList.

- ### 3 @Override

```
public int getColumnCount() {  
    return 7;  
}
```

Menentukan jumlah kolom yang ada di tabel

4. Method **getValueAt(int rowIndex, int columnIndex)** digunakan untuk menyesuaikan kolom mana yang mau ditampilkan.
  5. Method **getColumnName(int col)** Method ini menentukan nama header kolom yang muncul di bagian atas tabel (JTable).

## ■ TabelDataPenduduk

The screenshot shows a Java code editor with the following details:

- Title Bar:** Source, History, and various icons for file operations.
- Code Area:** The code is written in Java and defines a class named `TabelDataPenduduk`. The code includes imports for Java AWT and Swing components, a list to store `Penduduk` objects, and components like `JTable`, `JButton`, and `JScrollPane`.
- Code Content:**

```
9  * @author Celvin Pati
10 */
11 import java.awt.BorderLayout;
12 import java.awt.FlowLayout;
13 import java.util.ArrayList;
14 import java.util.List;
15 import javax.swing.*;
16 import javax.swing.table.AbstractTableModel;
17
18 public class TabelDataPenduduk extends JFrame {
19
20     public static List<Penduduk> daftarPenduduk = new ArrayList<>();
21
22     private JTable tabel;
23     private JButton btnSegarkan;
24     private JButton btnTutup;
25     private JScrollPane scrollPane;
26
27     public TabelDataPenduduk() {
28         setTitle("DATA PENDUDUK");
29         setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
30         setSize(900, 400);
31         setLocationRelativeTo(null);
32         initComponents();
33     }
34
35     private void initComponents() {
36
37         tabel = new JTable();
38         scrollPane = new JScrollPane(tabel);
39         btnSegarkan = new JButton("Segarkan");
40         btnTutup = new JButton("Tutup");
41
42         JPanel bottom = new JPanel(new FlowLayout(FlowLayout.CENTER, 12, 8));
43
44         bottom.add(btnSegarkan);
45         bottom.add(btnTutup);
46
47         getContentPane().setLayout(new BorderLayout(8, 8));
48         getContentPane().add(scrollPane, BorderLayout.CENTER);
49         getContentPane().add(bottom, BorderLayout.SOUTH);
50
51         refreshTableModel();
52
53         btnSegarkan.addActionListener(e -> refreshTableModel());
54         btnTutup.addActionListener(e -> dispose());
55     }
56
57     public void refreshTableModel() {
58
59         tabel.setModel(new PendudukTableModel(daftarPenduduk));
60
61         ((AbstractTableModel) tabel.getModel()).fireTableDataChanged();
62     }
63
64     public static void main(String[] args) {
65
66         if (daftarPenduduk.isEmpty()) {
67             Penduduk p1 = new Penduduk();
68             p1.setNik("0123456789");
69             p1.setNama("Budi Santoso");
70             p1.setJenisKelamin("Laki-Laki");
71             p1.setTanggalLahir("01/01/90");
72             p1.setNoHp("081-234-567-890");
73             p1.setAlamat("Jl. Kebon Jeruk");
74             p1.setId("Budi90");
75             p1.setPassword(new StringBuilder(p1.getNoHp()).reverse().toString());
76             daftarPenduduk.add(p1);
77
78             Penduduk p2 = new Penduduk();
79             p2.setNik("9876543210");
80             p2.setNama("Siti Aminah");
81             p2.setJenisKelamin("Perempuan");
82             p2.setTanggalLahir("02/02/91");
83             p2.setNoHp("082-111-222-333");
84             p2.setAlamat("Jl. Merdeka");
85             p2.setId("Siti91");
86             p2.setPassword(new StringBuilder(p2.getNoHp()).reverse().toString());
87             daftarPenduduk.add(p2);
88
89         }
90
91     }
}
```

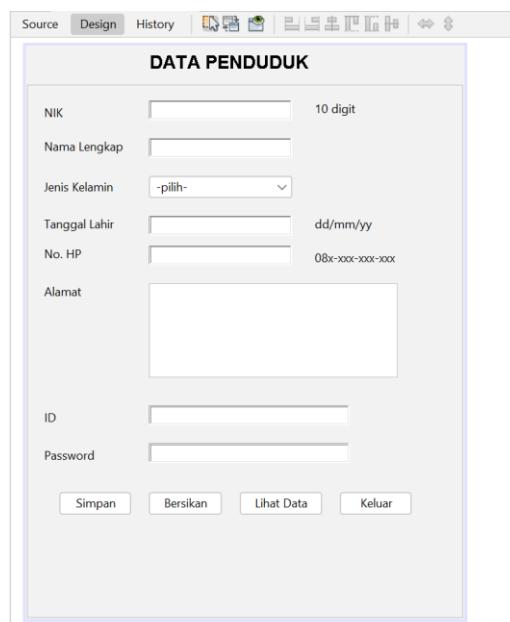
### Penjelasan:

```
1. Konstruktor public TabelDataPenduduk() {  
    setTitle("DATA PENDUDUK");  
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
    setSize(900, 400);  
    setLocationRelativeTo(null);  
    initComponents();  
}
```

Untuk judul, menentukan aksi saat jendela ditutup, menentukan ukuran jendela, menampilkan jendela di tengah layar, memanggil method untuk membangun isi tampilan.

2. **Method initComponents()** untuk Membuat tabel dan tombol-tombol, langsung menampilkan data ke tabel saat pertama dibuka.
3. **Method refreshTableModel()** untuk Membuat model baru berdasarkan data daftarPenduduk, menyegarkan tampilan tabel agar menampilkan data terbaru, fireTableDataChanged() memastikan tabel langsung diperbarui.
4. **Method main()** untuk Mengecek apakah daftarPenduduk kosong, untuk membalik nomor HP menjadi password, dan menjalankan tampilan GUI dengan setVisible(true).

### ■ Jframe DataPenduduk



#### ■ JButton Simpan

```
247 private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
248     Penduduk p = new Penduduk();  
249  
250     p.setNIK(txtNIK.getText());  
251     p.setName(txtNamaLengkap.getText());  
252     p.setJenisKelamin(cbJenisKelamin.getSelectedItem().toString());  
253     p.setTanggalLahir(txtTanggalLahir.getText());  
254     p.setNoHP(txtNoHP.getText());  
255     p.setAlamat(txtAlamat.getText());  
256     p.setId(txtID.getText());  
257     p.setPassword(txtPassword.getText());  
258  
259     TabelDataPenduduk.daftarPenduduk.add(p);  
260  
261     JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");  
262  
263  
264 }  
265  
266
```

#### Penjelasan:

**private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt)**

tombol "Simpan", yaitu tombol yang berfungsi untuk menyimpan data penduduk baru ke dalam daftar utama (daftarPenduduk).

#### ■ JButton Bersihkan

```
267 private void btnBersihkanActionPerformed(java.awt.event.ActionEvent evt) {  
268  
269     txtNIK.setText("");  
270     txtNamaLengkap.setText("");  
271     cbJenisKelamin.setSelectedIndex(0);  
272     txtTanggalLahir.setText("");  
273     txtNoHP.setText("");  
274     txtAlamat.setText("");  
275     txtID.setText("");  
276     txtPassword.setText("");  
277  
278  
279 }  
280  
281 }
```

#### Penjelasan:

**private void btnBersihkanActionPerformed(java.awt.event.ActionEvent evt)**

tombol "Bersihkan", yaitu tombol yang digunakan untuk mengosongkan semua field input pada form data penduduk.

#### ■ JButton Lihat Data

```
281  
282 private void btnLihatDataActionPerformed(java.awt.event.ActionEvent evt) {  
283     new TabelDataPenduduk().setVisible(true);  
284  
285  
286  
287 }  
288  
289 }
```

#### Penjelasan:

**private void btnLihatDataActionPerformed(java.awt.event.ActionEvent evt) {**

```
    new TabelDataPenduduk().setVisible(true);  
}
```

Tombol ini berfungsi untuk menampilkan jendela baru yang berisi tabel data semua penduduk yang sudah disimpan yaitu membuka form TabelDataPenduduk (yang kamu tunjukkan di gambar sebelumnya).

#### ■ JButton Keluar

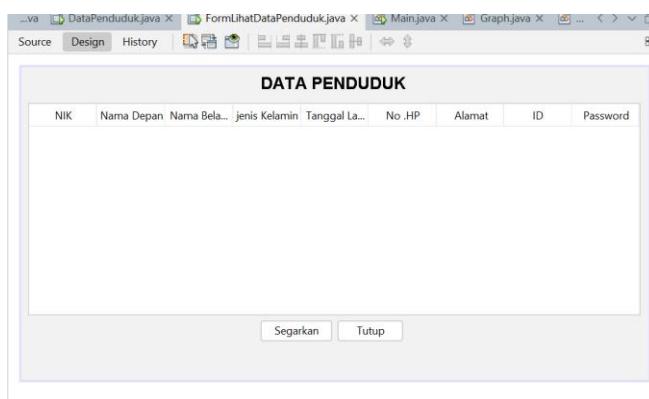
```
288
289     private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
290         System.exit(0);
291     }
292 }
293 */
294 /**
 * ...
 */
```

Penjelasan:

```
private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt)
{
    System.exit(0);
}
```

tombol ini berfungsi untuk menutup program ketika ditekan.

#### ■ JFrame FormLihatDataPenduduk



#### ■ JButton Segarkan

```
53
54     private void btnSegarkanActionPerformed(java.awt.event.ActionEvent evt) {
55
56         jTable1.setModel(new PendudukTableModel(daftarPenduduk));
57         ((AbstractTableModel) jTable1.getModel()).fireTableDataChanged();
58
59
60
61
62
63 }
```

Penjelasan:

```
private void btnSegarkanActionPerformed(java.awt.event.ActionEvent evt) {
```

```
jTable1.setModel(new PendudukTableModel(daftarPenduduk));
((AbstractTableModel) jTable1.getModel()).fireTableDataChanged();
```

```
}
```

tombol btnSegarkan berfungsi untuk refresh data tabel penduduk.

#### ■ JButton Tutup

```
164]
165  private void btnTutupActionPerformed(java.awt.event.ActionEvent evt) {
166      this.dispose();
167  }
168
169
170}
171
```

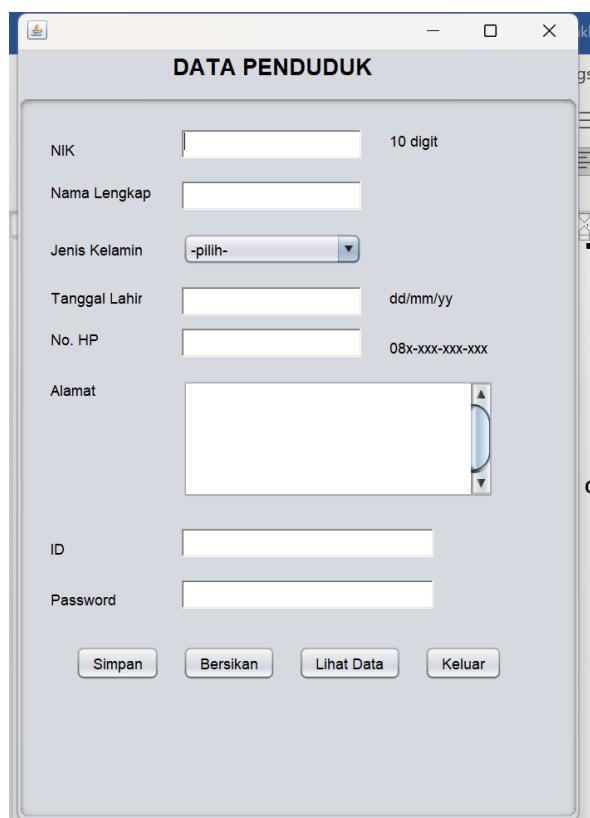
#### Penjelasan:

```
private void btnTutupActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();
}
```

Tombol ini untuk menutup jendela list data penduduk sajah.

#### Output

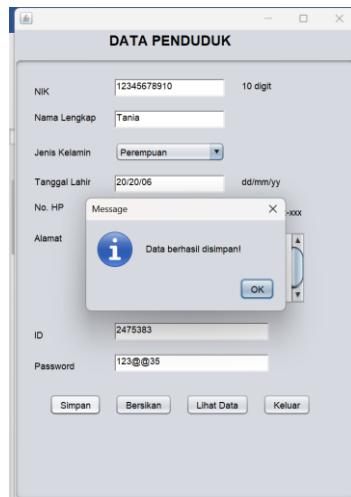
- **Output 1 hasil setelah program diRun**



#### Penjelasan:

Saat program diRun maka akan muncul Form untuk input data penduduk karena ada perintah **new FormPenduduk().setVisible(true);**

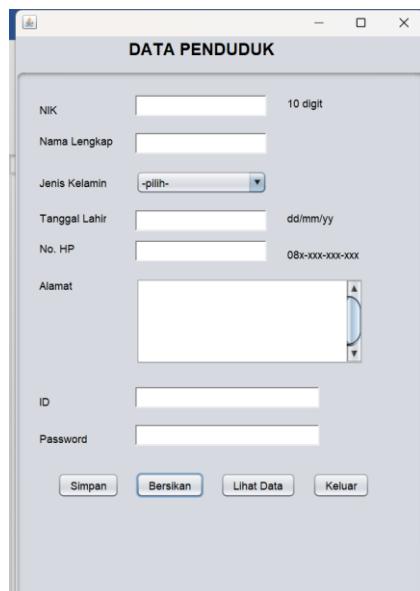
- **Output 2 hasil setelah mengisi data dan klik Simpan**



**Penjelasan:**

Setelah saya mengipun data seorang dan mengklik jButton simpan maka data akan langsung tersimpan karena code ini **private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt)** dan muncul JoptionPane dengan pesan “data berhasil disimpan” karena code **JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");**

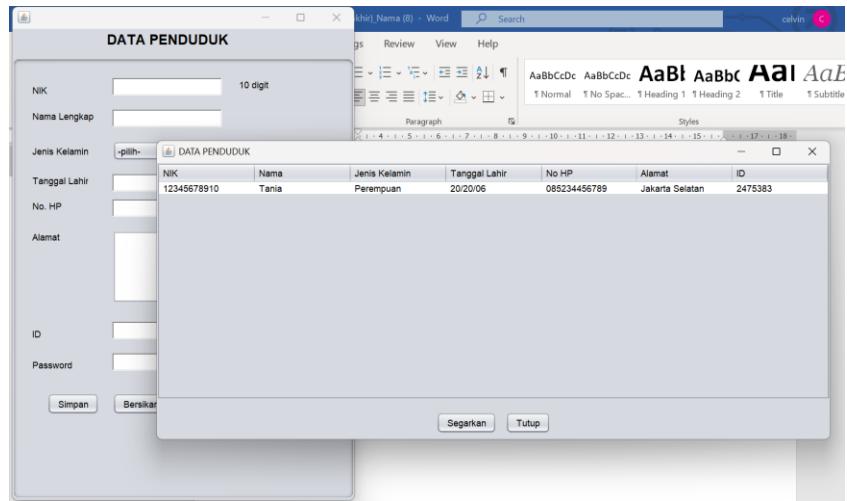
- **Output 3 hasil setelah klik JButton Bersihkan**



**Penjelasan:**

Setelah saya klik jButton bersihkan maka isi field atau data bekas input data penduduk di teks area maka field akan langsung clear karena code ini **private void btnBersihkanActionPerformed(java.awt.event.ActionEvent evt**

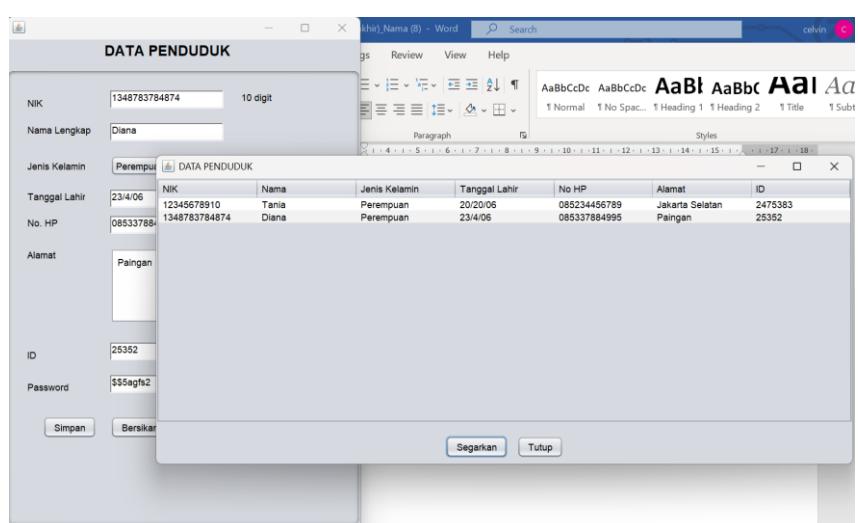
- **Output 4 hasil setelah klik JButton Lihat Data**



**Penjelasan:**

Setelah berhasil menginput dan menyimpan data penduduk tadi maka setelah saya klik JButton Lihat Data maka akan muncul Jframe untuk list data penduduk karena code ini private void **btnLihatDataActionPerformed(java.awt.event.ActionEvent evt)** {  
    new TabelDataPenduduk().setVisible(true);  
}

- **Output 5 hasil setelah klik JButton Segarkan**



**Penjelasan:**

Sebelumnya saya mau isi ladi data pengguna baru saat berhasil disimpan kemudian saya klik segarkan di Jframe list data penduduk itu maka akan refresh dan menampilkan data kedua yang saya baru input lagi, karena code ini

```
private void btnSegarkanActionPerformed(java.awt.event.ActionEvent evt) {  
    jTable1.setModel(new PendudukTableModel(daftarPenduduk));  
    ((AbstractTableModel) jTable1.getModel()).fireTableDataChanged();  
}
```