

LAPORAN MODUL (9)
Pemrograman Berorientasi Objek Lanjut

“String”

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



DIBUAT OLEH:

Nama : Yohanis Calvin 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

```
Source History
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 alamat;
18      private String id;
19      private String password;
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

```
Source History
11 import java.util.List;
12 import javax.swing.table.AbstractTableModel;
13
14 public class PendudukTableModel extends AbstractTableModel {
15
16     private List<Penduduk> daftarPenduduk;
17
18     public PendudukTableModel(List<Penduduk> dp) {
19         this.daftarPenduduk = dp;
20     }
21
22     @Override
23     public int getRowCount() {
24         return daftarPenduduk.size();
25     }
26
27     @Override
28     public int getColumnCount() {
29         return 7;
30     }
31
32     @Override
33     public Object getValueAt(int rowIndex, int columnIndex) {
34         Penduduk p = daftarPenduduk.get(rowIndex);
35         switch (columnIndex) {
36             case 0: return p.getNik();
37             case 1: return p.getNama();
38             case 2: return p.getJenisKelamin();
39             case 3: return p.getTanggalLahir();
40             case 4: return p.getNoHp();
41             case 5: return p.getAlamat();
42             case 6: return p.getId();
43             default: return "";
44         }
45     }
46
47     @Override
48     public String getColumnName(int col) {
49         switch (col) {
50             case 0: return "NIK";
51             case 1: return "Nama";
52             case 2: return "Jenis Kelamin";
53             case 3: return "Tanggal Lahir";
54             case 4: return "No HP";
55             case 5: return "Alamat";
56             case 6: return "ID";
57             default: return "";
58         }
59     }
60 }
```

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

```

Source History
9  * @author Calvin 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      private void initComponents() {
35
36          tabel = new JTable();
37          scrollPane = new JScrollPane(tabel);
38          btnSegarkan = new JButton("Segarkan");
39          btnTutup = new JButton("Tutup");
40
41          JPanel bottom = new JPanel(new FlowLayout(FlowLayout.CENTER, 12, 8));
42
43          bottom.add(btnSegarkan);
44          bottom.add(btnTutup);
45
46          getContentPane().setLayout(new BorderLayout(8, 8));
47          getContentPane().add(scrollPane, BorderLayout.CENTER);
48          getContentPane().add(bottom, BorderLayout.SOUTH);
49
50          refreshTableModel();
51
52          btnSegarkan.addActionListener(e -> refreshTableModel());
53          btnTutup.addActionListener(e -> dispose());
54      }
55      public void refreshTableModel() {
56
57          tabel.setModel(new PendudukTableModel(daftarPenduduk));
58
59          ((AbstractTableModel) tabel.getModel()).fireTableDataChanged();
60      }
61      public static void main(String[] args) {
62
63          if (daftarPenduduk.isEmpty()) {
64              Penduduk p1 = new Penduduk();
65              p1.setNik("0123456789");
66              p1.setNama("Budi Santoso");
67              p1.setJenisKelamin("Laki-Laki");
68              p1.setTanggalLahir("01/01/90");
69              p1.setNoHp("081-234-567-890");
70              p1.setAlamat("Jl. Kebon Jeruk");
71              p1.setId("Budi90");
72              p1.setPassword(new StringBuilder(p1.getNoHp()).reverse().toString());
73              daftarPenduduk.add(p1);
74
75              Penduduk p2 = new Penduduk();
76              p2.setNik("9876543210");
77              p2.setNama("Siti Aminah");
78              p2.setJenisKelamin("Perempuan");
79              p2.setTanggalLahir("02/02/91");
80              p2.setNoHp("082-111-222-333");
81              p2.setAlamat("Jl. Merdeka");
82              p2.setId("Siti91");
83              p2.setPassword(new StringBuilder(p2.getNoHp()).reverse().toString());
84              daftarPenduduk.add(p2);
85          }
86          SwingUtilities.invokeLater(() -> {
87              new TabelDataPenduduk().setVisible(true);
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

The screenshot shows a Java Swing window titled "DATA PENDUDUK". The window has a standard Mac OS-style title bar with "Source", "Design", and "History" tabs. The main content area is a light gray panel containing several form elements: a text field for "NIK" with a "10 digit" label, a text field for "Nama Lengkap", a dropdown menu for "Jenis Kelamin" with a "- pilih -" placeholder, a text field for "Tanggal Lahir" with a "dd/mm/yy" label, a text field for "No. HP" with a "08x-xxxx-xxxx-xxxx" label, a text area for "Alamat", a text field for "ID", and a text field for "Password". At the bottom of the panel, there are four buttons: "Simpan", "Bersihkan", "Lihat Data", and "Keluar".

■ JButton Simpan

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

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     cmbJenisKelamin.setSelectedIndex(0);  
272     txtTanggalLahir.setText("");  
273     txtNoHp.setText("");  
274     txtAlamat.setText("");  
275     txtID.setText("");  
276     txtPassword.setText("");  
277  
278  
279  
280 }
```

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  
284     new TabelDataPenduduk().setVisible(true);  
285  
286  
287  
288 }
```

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
291     System.exit(0);
292
293 }
294
295

```

Penjelasan:

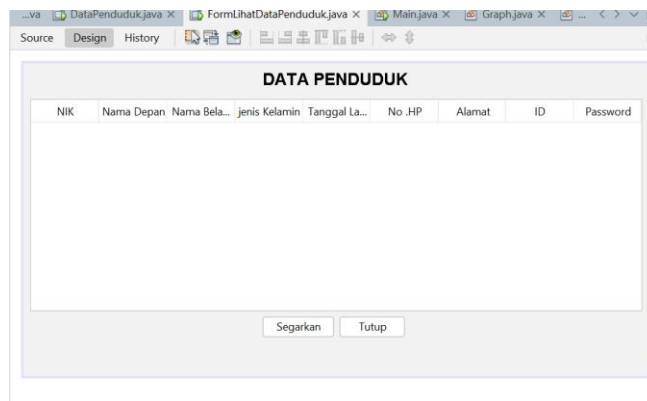
```

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

```

tombol ini berfungsi untuk menutup program ketika ditekan.

▪ JFrame FormLihatDataPenduduk



▪ JButton Segarkan

```

53 private void btnSegarkanActionPerformed(java.awt.event.ActionEvent evt) {
54
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  
167     this.dispose();  
168  
169  
170 }
```

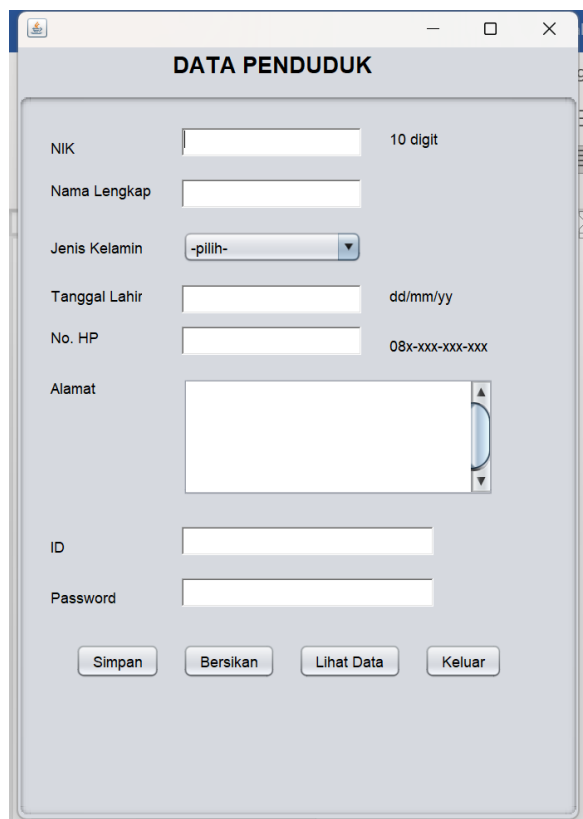
Penjelasan:

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

Tombol ini untuk menutup jendela list data penduduk saja.

Output

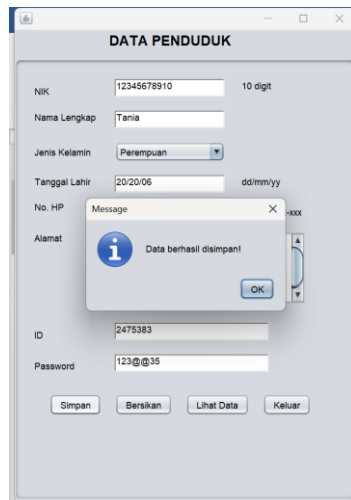
- Output 1 hasil setelah program diRun



Penjelasan:

Saat progrm diRun maka akan muncul From untuk input data penduduk karena ada perintah `new FormPenduduk().setVisible(true);`

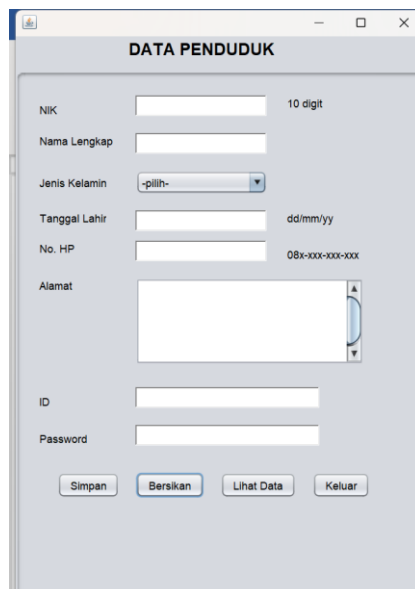
- **Output 2** hasil setelah mengisi data dan klik Simpan



Penjelasan:

Setelah saya menginput 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!");**

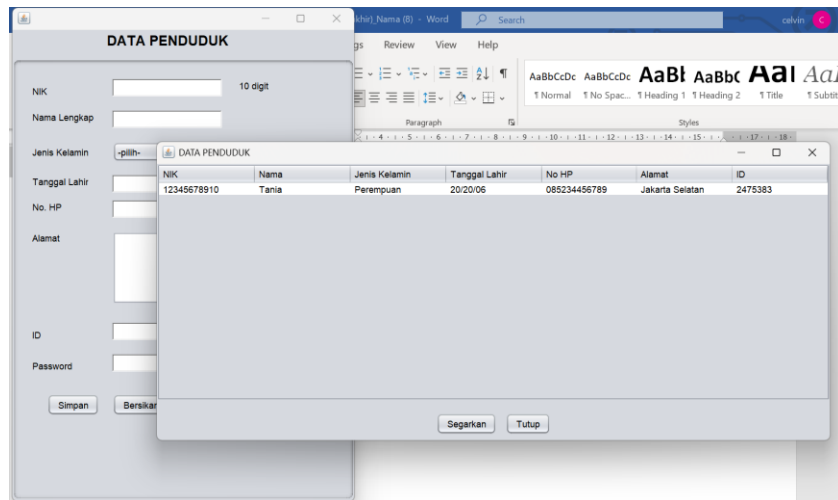
- **Output 3** hasil setelah klik Jbutton Bersihkan



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**

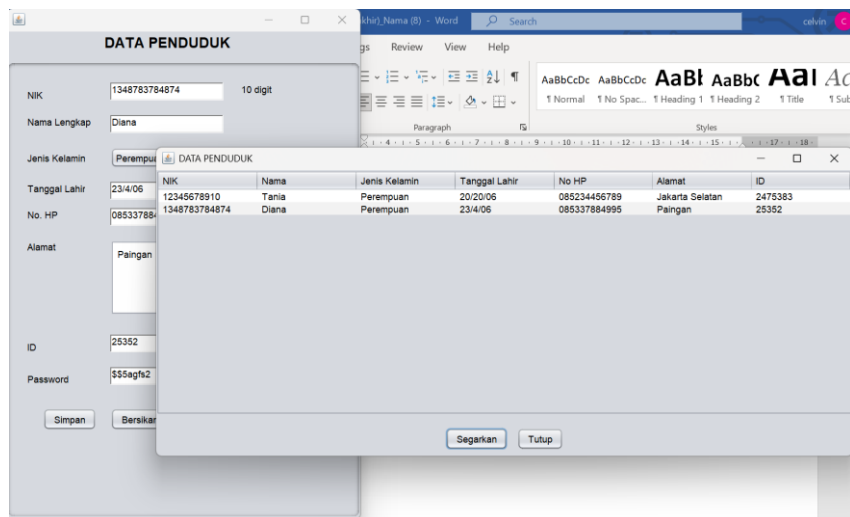


Penjelasan:

Setelah mengisi 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:

Setelah itu saya mau isi lagi data pengguna baru saat sudah 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();  
}
```