**PRATIKUM PEMROGRAMAN**

**PERTEMUAN KE - 3**



Disusun Oleh :

**Rizki Pratama (223040017)**

**Kelas B**

**Teknik Informatika**

**Fakultas Teknik**

**Universitas Pasundan**

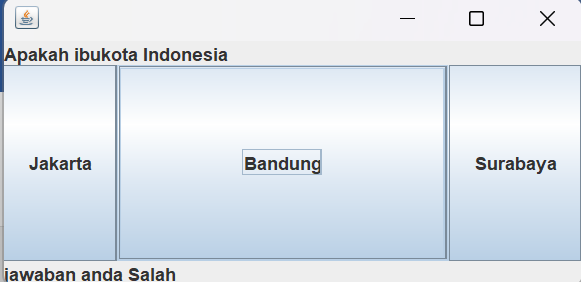
1. HelloBorderLayout



Output







Penjelasan

1. **Pernyataan Impor**:
   * import java.awt.\*;, import java.awt.event.\*;, dan import javax.swing.\*; digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membuat antarmuka pengguna grafis (GUI) dan menangani event.
2. **Deklarasi Kelas**:
   * public class HelloBorderLayout extends JFrame { mendeklarasikan kelas publik bernama HelloBorderLayout yang memperluas kelas JFrame.
3. **Konstruktor Kelas**:
   * public HelloBorderLayout() { adalah konstruktor kelas di mana jendela dan komponennya diatur.
4. **Membuat JFrame**:
   * this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); memastikan aplikasi keluar ketika jendela ditutup.
   * this.setSize(400, 200); menetapkan ukuran jendela menjadi 400x200 piksel.
5. **Membuat JLabel**:
   * JLabel labelpertanyaan = new JLabel("Apakah ibukota Indonesia"); membuat label dengan teks pertanyaan.
   * JLabel labelhasil = new JLabel("Jawab Pertanyaan di atas"); membuat label dengan teks hasil.
6. **Membuat JButton**:
   * JButton buttonA = new JButton("Jakarta");, JButton buttonB = new JButton("Bandung");, dan JButton buttonC = new JButton("Surabaya"); membuat tiga tombol dengan teks kota.
7. **Menambahkan ActionListener ke Tombol**:
   * buttonA.addActionListener(new ActionListener() { ... }); menambahkan pendengar aksi ke tombol buttonA yang akan menampilkan "jawaban anda benar" ketika diklik.
   * buttonB.addActionListener(new ActionListener() { ... }); dan buttonC.addActionListener(new ActionListener() { ... }); masing-masing menampilkan "jawaban anda Salah" ketika tombol diklik.
8. **Menambahkan Komponen ke Frame dengan BorderLayout**:
   * this.add(labelpertanyaan, BorderLayout.NORTH); menambahkan label pertanyaan ke bagian atas frame.
   * this.add(labelhasil, BorderLayout.SOUTH); menambahkan label hasil ke bagian bawah frame.
   * this.add(buttonA, BorderLayout.WEST);, this.add(buttonB, BorderLayout.CENTER);, dan this.add(buttonC, BorderLayout.EAST); menambahkan tombol ke posisi yang sesuai dalam layout border.
9. **Metode Main untuk Menjalankan Aplikasi**:
   * public static void main(String[] args) { javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() { HelloBorderLayout h = new HelloBorderLayout(); h.setVisible(true); } }); } menjalankan aplikasi dalam thread event-dispatching Swing, membuat instance HelloBorderLayout dan menampilkan jendela.

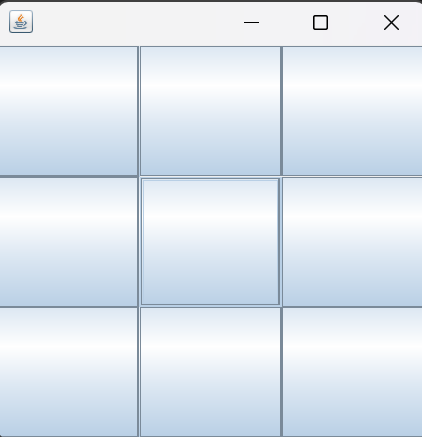
2. HelloGridLayout

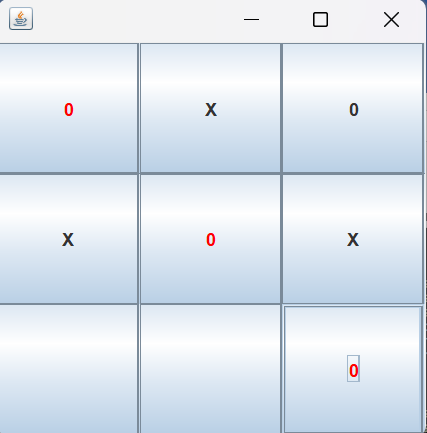


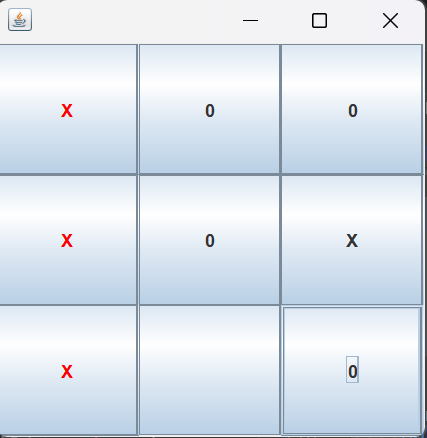




Output







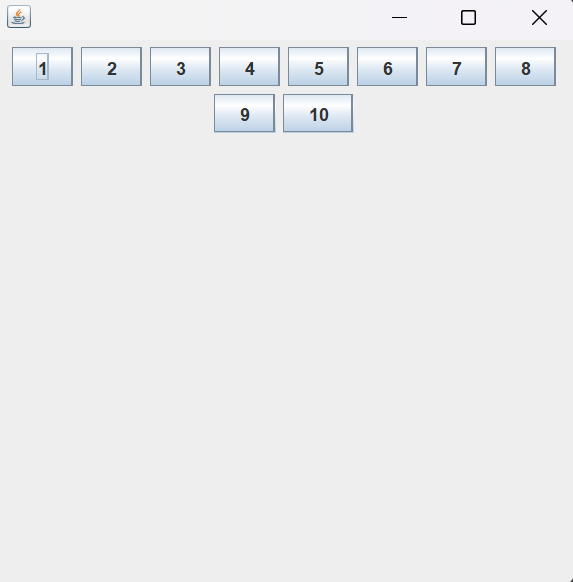
Penjelasan

1. **Pernyataan Impor**:
   * import java.awt.\*;, import java.awt.event.\*;, dan import javax.swing.\*; digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membuat antarmuka pengguna grafis (GUI) dan menangani event.
2. **Deklarasi Kelas**:
   * public class HelloGridLayout extends JFrame implements ActionListener { mendeklarasikan kelas publik HelloGridLayout yang memperluas JFrame dan mengimplementasikan ActionListener.
3. **Deklarasi Variabel**:
   * Kelas ini mendeklarasikan beberapa tombol dan array tombol, serta variabel gameOver untuk menentukan status permainan.
4. **Konstruktor Kelas**:
   * public HelloGridLayout() { adalah konstruktor kelas di mana jendela dan komponennya diatur.
   * this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); memastikan aplikasi keluar ketika jendela ditutup.
   * gameOver = false; menginisialisasi variabel gameOver ke false.
5. **Membuat Tombol**:
   * Membuat tombol kosong ("") dan memasukkannya ke dalam array buttons.
6. **Menambahkan ActionListener ke Tombol**:
   * Menambahkan pendengar aksi ke setiap tombol menggunakan buttonA.addActionListener(this);.
7. **Mengatur Layout dan Menambahkan Tombol ke Frame**:
   * this.setLayout(new GridLayout(3, 3)); menetapkan layout frame menjadi GridLayout dengan 3 baris dan 3 kolom.
   * Menambahkan setiap tombol ke frame.
8. **Metode actionPerformed**:
   * public void actionPerformed(ActionEvent e) { menangani event ketika tombol ditekan.
   * Jika permainan belum berakhir (!gameOver), menandai tombol dengan teks "0" dan memeriksa pemenang (checkWinner()).
   * Jika belum ada pemenang, menandai tombol kosong berikutnya dengan "X" dan memeriksa pemenang lagi.
9. **Metode checkWinner**:
   * public void checkWinner() { memeriksa apakah ada pemenang dengan membandingkan teks tombol dalam pola kemenangan.
   * Jika ditemukan pemenang, mengubah warna teks tombol menjadi merah dan mengatur gameOver ke true.
10. **Metode Main untuk Menjalankan Aplikasi**:
    * public static void main(String[] args) { javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() { HelloGridLayout h = new HelloGridLayout(); h.setVisible(true); } }); } menjalankan aplikasi dalam thread event-dispatching Swing, membuat instance HelloGridLayout dan menampilkan jendela.

3. HelloFlowLayout



Output



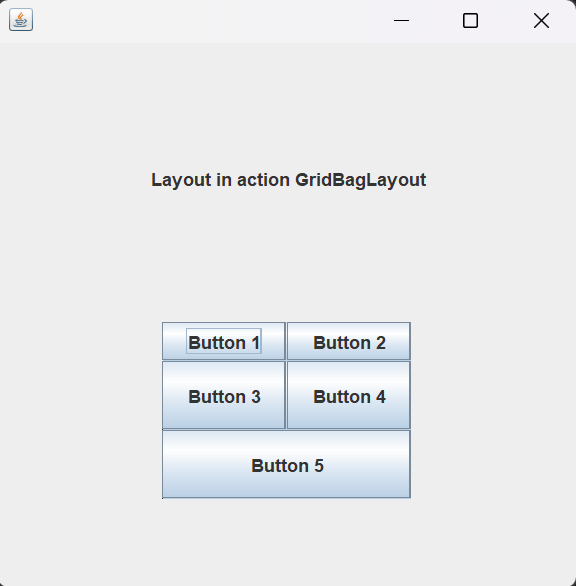
Penjelasan

1. **Pernyataan Impor**:
   * import java.awt.\*; dan import javax.swing.\*; digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membuat antarmuka pengguna grafis (GUI).
2. **Deklarasi Kelas**:
   * public class HelloFlowLayout extends JFrame { mendeklarasikan kelas publik HelloFlowLayout yang memperluas kelas JFrame.
3. **Konstruktor Kelas**:
   * public HelloFlowLayout() { adalah konstruktor kelas di mana jendela dan komponennya diatur.
   * this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); memastikan aplikasi keluar ketika jendela ditutup.
4. **Membuat Tombol**:
   * Membuat sepuluh tombol (JButton button1 = new JButton("1");, JButton button2 = new JButton("2");, ... JButton button10 = new JButton("10");) dengan label angka 1 sampai 10.
5. **Mengatur Layout dan Menambahkan Tombol ke Frame**:
   * this.setLayout(new FlowLayout()); menetapkan layout frame menjadi FlowLayout.
   * Menambahkan setiap tombol ke frame menggunakan this.add(button1);, this.add(button2);, ..., this.add(button10);.
6. **Menetapkan Ukuran Frame**:
   * this.setSize(400, 400); menetapkan ukuran jendela menjadi 400x400 piksel.
7. **Metode Main untuk Menjalankan Aplikasi**:
   * public static void main(String[] args) { javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() { HelloFlowLayout h = new HelloFlowLayout(); h.setVisible(true); } }); }

4. GridBagLayout



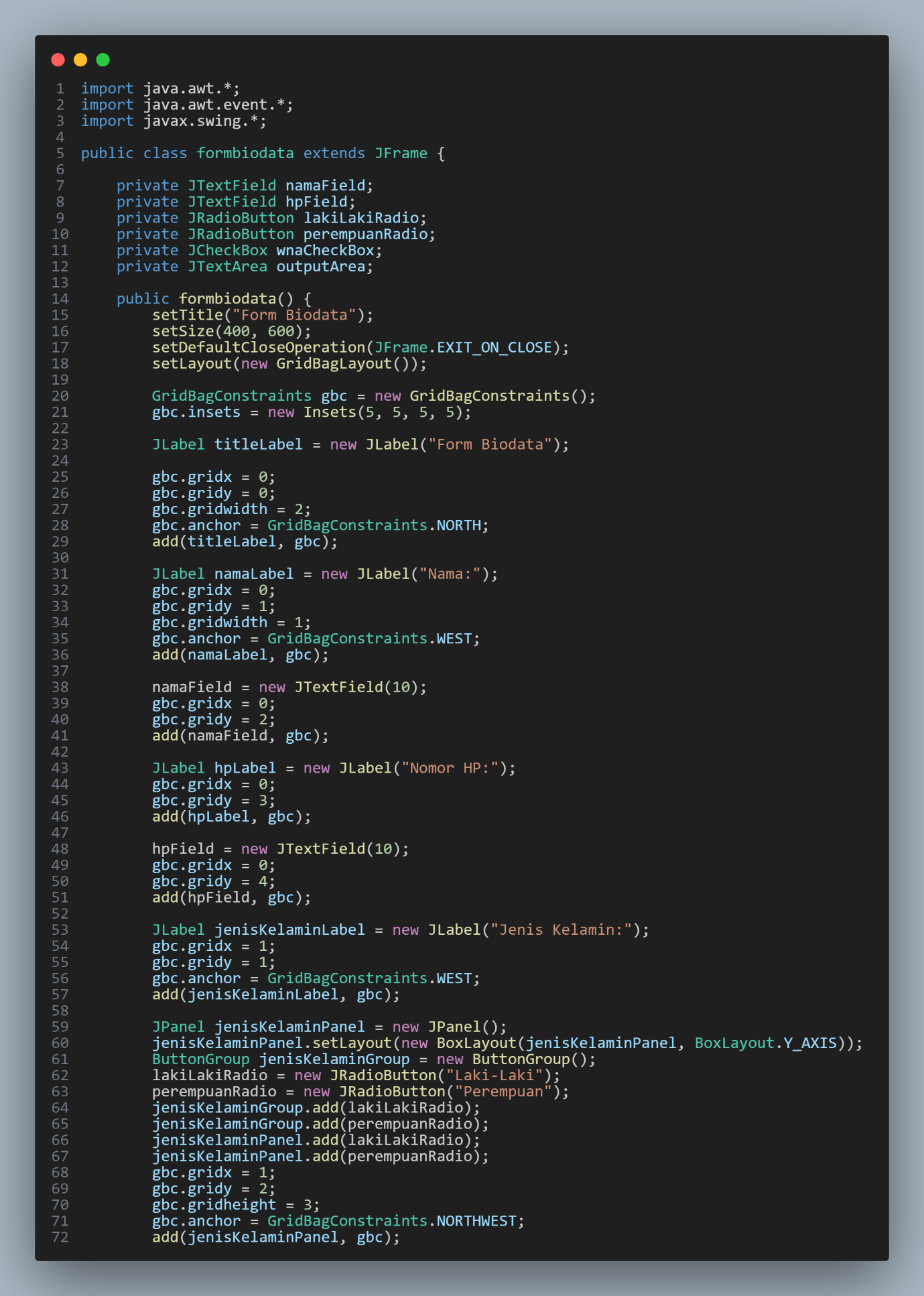
Output

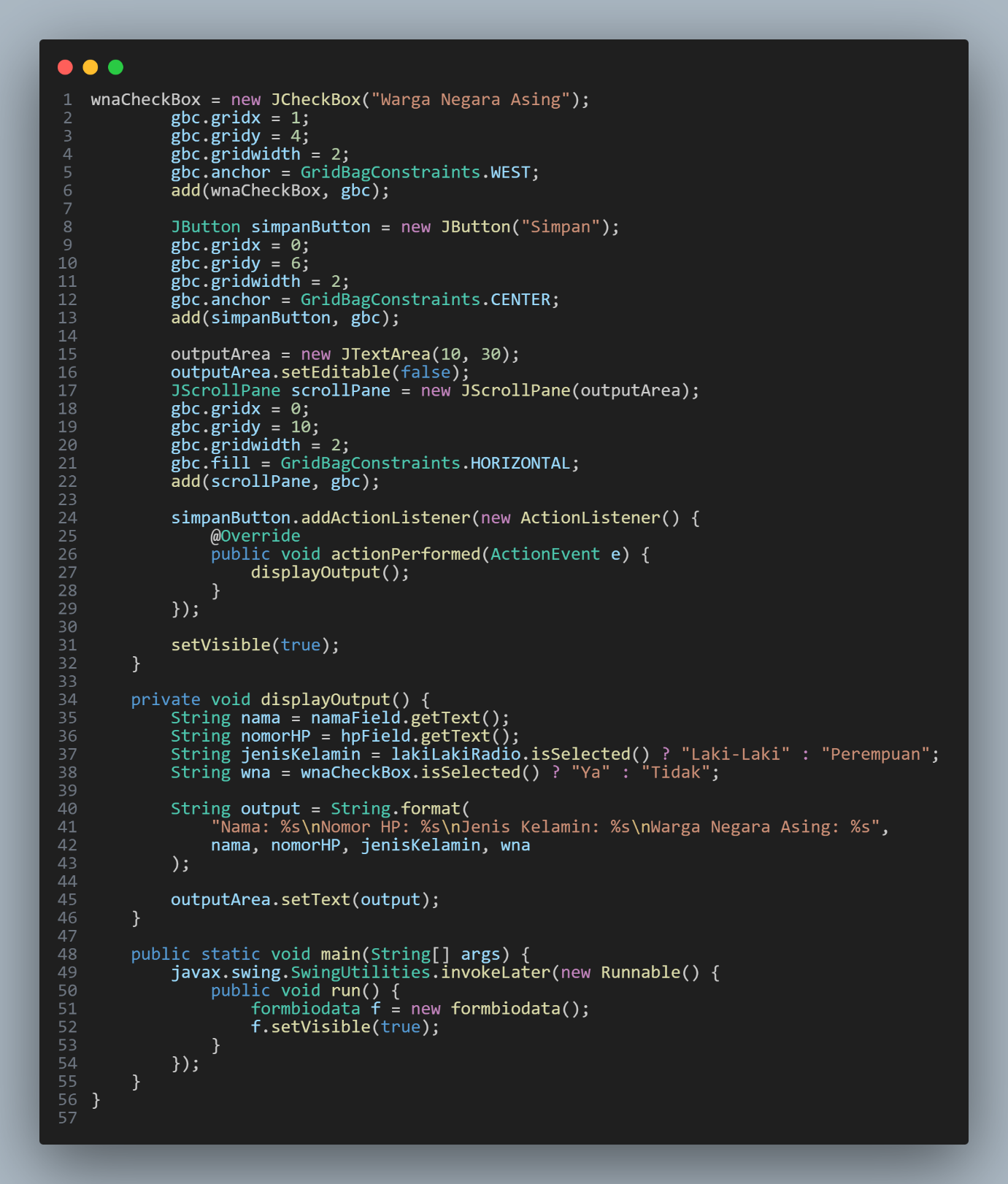


Penjelasan

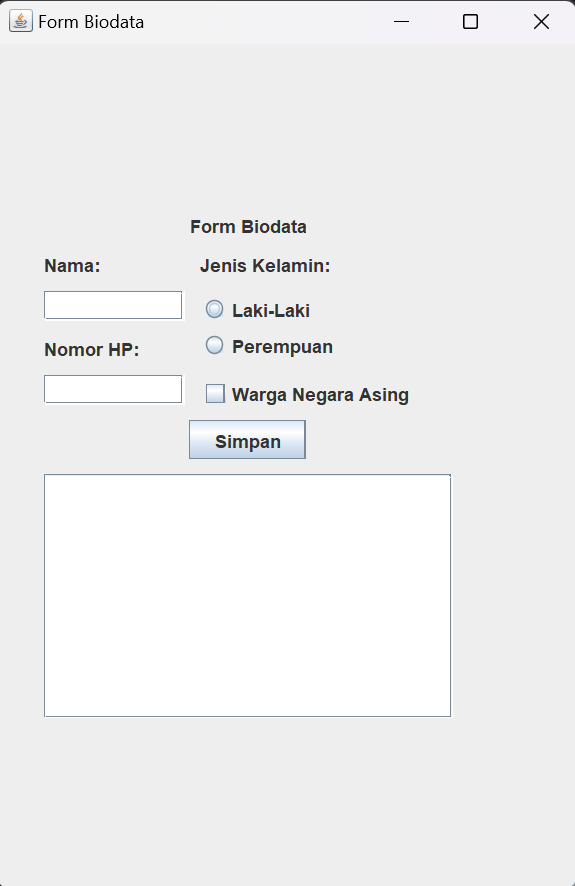
1. **Pernyataan Impor**:
   * import java.awt.\*;, import java.awt.event.\*;, dan import javax.swing.\*; digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membuat antarmuka pengguna grafis (GUI) dan menangani event.
2. **Deklarasi Kelas**:
   * public class HelloGridBagLayout extends JFrame { mendeklarasikan kelas publik HelloGridBagLayout yang memperluas kelas JFrame.
3. **Konstruktor Kelas**:
   * public HelloGridBagLayout() { adalah konstruktor kelas di mana jendela dan komponennya diatur.
   * this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); memastikan aplikasi keluar ketika jendela ditutup.
4. **Membuat Komponen**:
   * JLabel headerLabel = new JLabel("Layout in action GridBagLayout", JLabel.CENTER); membuat label dengan teks "Layout in action GridBagLayout" dan mengatur posisi teks di tengah.
   * JPanel controlPanel = new JPanel(); membuat panel kontrol.
   * controlPanel.setLayout(new FlowLayout()); menetapkan layout panel kontrol menjadi FlowLayout.
   * JPanel panel = new JPanel(); membuat panel lain.
   * panel.setBackground(Color.darkGray); menetapkan warna latar belakang panel menjadi abu-abu gelap.
   * panel.setSize(300, 300); menetapkan ukuran panel menjadi 300x300 piksel.
5. **Mengatur GridBagLayout dan GridBagConstraints**:
   * GridBagLayout layout = new GridBagLayout(); membuat layout GridBagLayout.
   * panel.setLayout(layout); menetapkan layout panel menjadi GridBagLayout.
   * GridBagConstraints gbc = new GridBagConstraints(); membuat objek GridBagConstraints untuk mengatur batasan grid.
   * Menambahkan tombol ke panel dengan menggunakan batasan grid yang sesuai (gbc.gridx, gbc.gridy, dll.).
6. **Menambahkan Komponen ke Frame**:
   * controlPanel.add(panel); menambahkan panel ke panel kontrol.
   * this.setLayout(new GridLayout(2, 1)); menetapkan layout frame menjadi GridLayout dengan 2 baris dan 1 kolom.
   * this.add(headerLabel); menambahkan label header ke frame.
   * this.add(controlPanel); menambahkan panel kontrol ke frame.
   * this.setSize(400, 400); menetapkan ukuran frame menjadi 400x400 piksel.
7. **Metode Main untuk Menjalankan Aplikasi**:
   * public static void main(String[] args) { javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() { HelloGridBagLayout h = new HelloGridBagLayout(); h.setVisible(true); } }); } menjalankan aplikasi dalam thread event-dispatching Swing, membuat instance HelloGridBagLayout dan menampilkan jendela.

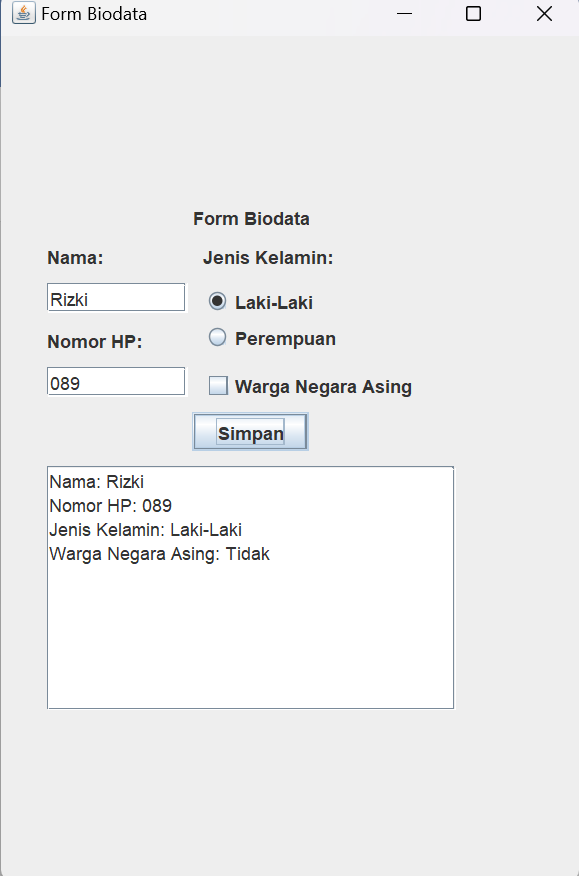
5. formBiodata

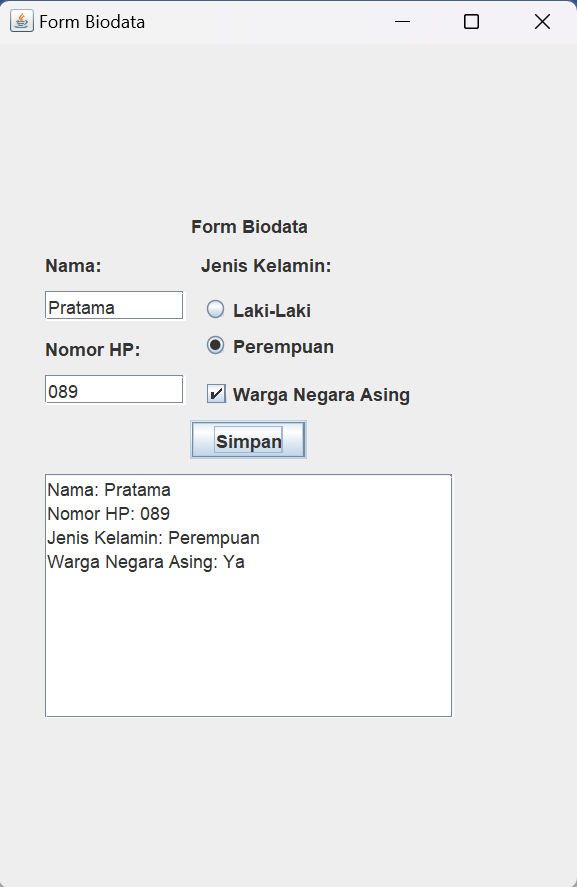




Output







Penjelasan

1. **Pernyataan Impor**:
   * import java.awt.\*;, import java.awt.event.\*;, dan import javax.swing.\*; digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membuat antarmuka pengguna grafis (GUI) dan menangani event.
2. **Deklarasi Kelas**:
   * public class formbiodata extends JFrame { mendeklarasikan kelas publik formbiodata yang memperluas kelas JFrame.
3. **Deklarasi Variabel**:
   * Mendeklarasikan variabel teks, radio button, checkbox, dan area teks yang akan digunakan dalam form.
4. **Konstruktor Kelas**:
   * public formbiodata() { adalah konstruktor kelas di mana jendela dan komponennya diatur.
   * setTitle("Form Biodata"); menetapkan judul jendela menjadi "Form Biodata".
   * setSize(400, 600); menetapkan ukuran jendela menjadi 400x600 piksel.
   * setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); memastikan aplikasi keluar ketika jendela ditutup.
   * setLayout(new GridBagLayout()); menetapkan layout jendela menjadi GridBagLayout.
5. **Menambahkan Komponen ke Jendela**:
   * Menambahkan label judul, label nama, field nama, label nomor HP, field nomor HP, label jenis kelamin, panel jenis kelamin, checkbox WNA, tombol simpan, dan area output menggunakan GridBagConstraints untuk menentukan posisi dan tata letak komponen dalam grid.
   * Contoh: gbc.gridx = 0; gbc.gridy = 1; gbc.anchor = GridBagConstraints.WEST; add(namaLabel, gbc); menambahkan label nama ke grid pada posisi (0,1).
6. **Menambahkan ActionListener ke Tombol Simpan**:
   * simpanButton.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent e) { displayOutput(); } }); menambahkan pendengar aksi ke tombol simpan untuk memanggil metode displayOutput ketika tombol diklik.
7. **Metode displayOutput**:
   * private void displayOutput() { mengambil teks dari field nama, nomor HP, dan pilihan jenis kelamin dan WNA, kemudian menampilkan hasilnya di area output.
   * Contoh: String output = String.format("Nama: %s\nNomor HP: %s\nJenis Kelamin: %s\nWarga Negara Asing: %s", nama, nomorHP, jenisKelamin, wna); menggabungkan informasi menjadi satu string.
8. **Metode Main untuk Menjalankan Aplikasi**:
   * public static void main(String[] args) { javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() { formbiodata f = new formbiodata(); f.setVisible(true); } }); } menjalankan aplikasi dalam thread event-dispatching Swing, membuat instance formbiodata, dan menampilkan jendela.