

TUGAS PEKAN 8 PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

Nama : Muhammad Yasin Habiburrahman

NIM : 2511532016

1. Pseudocode

Deskripsi

Program yang menerapkan operator assignment, yaitu suatu variabel yang nilainya berubah-ubah berdasarkan operasi aritmatika yang dimasukkan pengguna

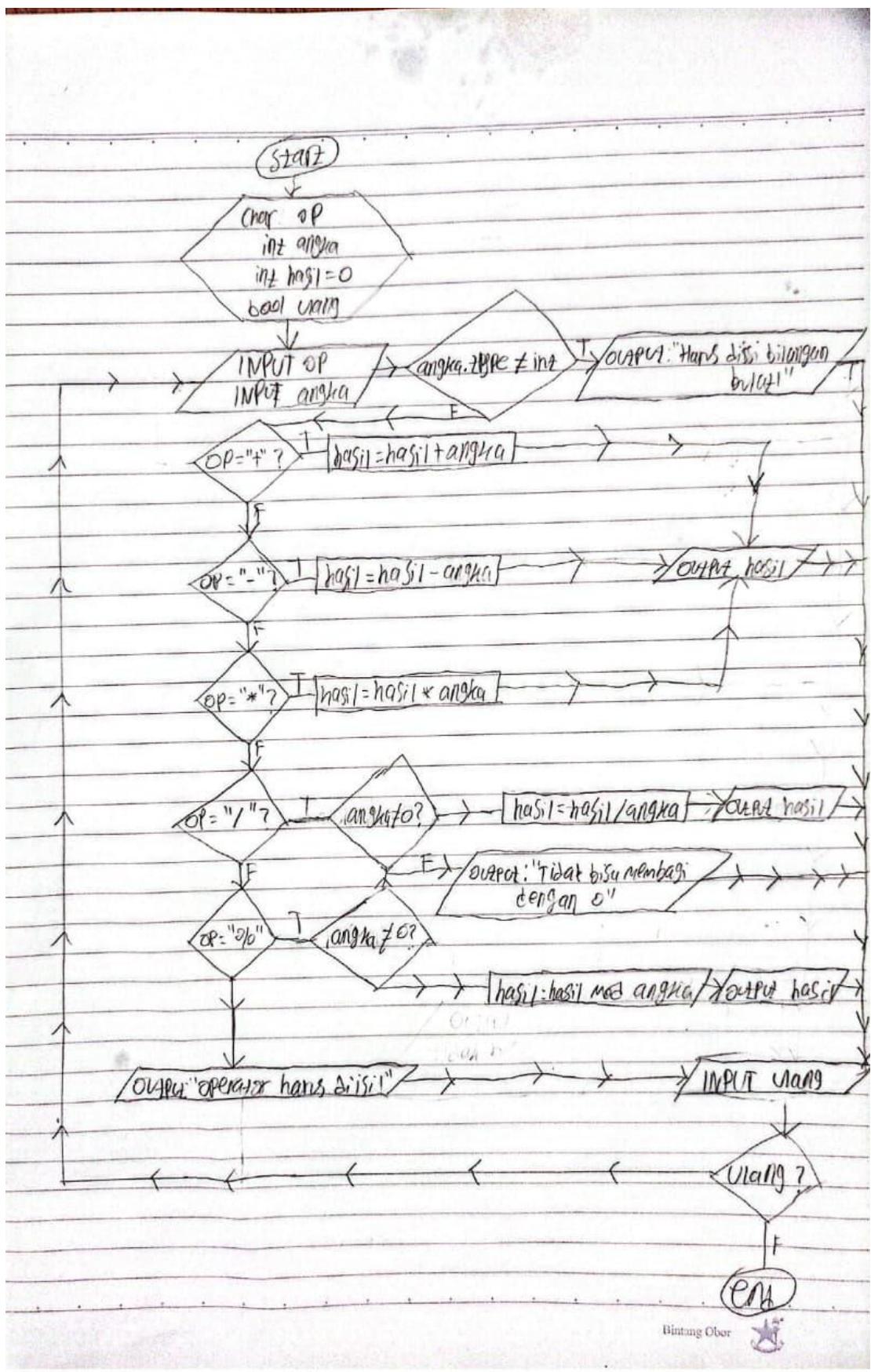
Deklarasi

```
Var op; ←CHAR  
Var angka; ←INT  
Var hasil=0; ←INT  
Var ulang=true; ← BOOLEAN
```

Algoritma

```
WHILE(ulang) {  
    INPUT angka  
    INPUT op  
    IF(op == '+') THEN  
        hasil = hasil + angka  
  
    ELSE IF(op == '-') THEN  
        hasil = hasil - angka  
    END ELSE IF  
  
    ELSE IF(op == '*') THEN  
        hasil = hasil * angka  
    END ELSE IF  
  
    ELSE IF(op == '/') THEN  
        IF(angka==0)THEN  
            OUTPUT("TIDAK BISA MEMBAGI DENGAN 0!")  
        ELSE hasil = hasil / angka  
        END IF  
    END ELSE IF  
  
    ELSE IF(op == '%') THEN  
        IF(angka==0)THEN  
            OUTPUT("TIDAK BISA MODULUS DENGAN 0!")  
        ELSE hasil = hasil % angka  
    END ELSE IF  
END IF  
  
OUTPUT hasil  
INPUT ulang  
END WHILE
```

2. Flowchart



3. Program Java

3.1 Library

```
1 package pekan8_2511532016;
2 import java.awt.BorderLayout;
3 import java.awt.EventQueue;
4 import javax.swing.JFrame;
5 import javax.swing.JPanel;
6 import javax.swing.border.EmptyBorder;
7 import javax.swing.JLabel;
8 import javax.swing.JOptionPane;
9 import java.awt.Font;
10 import javax.swing.SwingConstantsConstants;
11 import javax.swing.JTextField;
12 import javax.swing.JComboBox;
13 import javax.swing.DefaultComboBoxModel;
14 import javax.swing.JButton;
15 import java.awt.event.ActionListener;
16 import java.awt.event.ActionEvent;
```

Di awal program, terdapat sejumlah pernyataan *import* yang digunakan untuk memasukkan *package* dan *class* bawaan dari Java yang diperlukan dalam pembuatan GUI. Bagian *import* ini memungkinkan program menggunakan komponen-komponen GUI seperti *JFrame*, *JPanel*, *JLabel*, *JButton*, serta berbagai kelas pendukung dari *Abstract Window Toolkit* (AWT) dan *Swing*.

3.2 Deklarasi Variabel, Atribut, dan Metode-Metode Pendukung

```
17 public class OperatorAssignmentGUITugasPekan8_2511532016 extends JFrame {
18     private static final long serialVersionUID = 1L;
19     private JPanel contentPane;
20     private JTextField txtBil1;
21     private JTextField txtHasil;
22     private int hasil=0;
23     private void pesanPeringatan(String pesan) {
24         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
25     }
26     private void pesanError(String pesan) {
27         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
28     }
```

Bagian kode tersebut merupakan deklarasi awal kelas *OperatorAssignmentGUITugasPekan8_2511532016* yang berfungsi sebagai kelas utama program GUI dan diturunkan dari *JFrame*, sehingga objek yang dihasilkan akan berupa sebuah jendela aplikasi. Pada bagian ini juga terdapat deklarasi variabel *serialVersionUID* yang digunakan sebagai identitas versi kelas saat proses serialisasi, meskipun tidak memengaruhi logika program secara langsung.

Selanjutnya, beberapa atribut kelas didefinisikan, seperti *contentPane*, *txtBil1*, dan *txtHasil*, yang masing-masing berperan sebagai panel utama serta komponen *input* dan *output* pada

antarmuka grafis. Variabel hasil juga dideklarasikan untuk menyimpan nilai perhitungan yang akan terus diperbarui berdasarkan operator assignment yang dipilih pengguna. Selain itu, terdapat dua metode utilitas, yaitu pesanPeringatan() dan pesanError(), yang masing-masing digunakan untuk menampilkan kotak dialog peringatan dan kesalahan menggunakan JOptionPane.

3.3 Metode main() sebagai Awal dari Program

```
30      //Launch the application.
31  public static void main(String[] args) {
32      EventQueue.invokeLater(new Runnable() {
33          public void run() {
34              try {
35                  OperatorAssignmentGUITugasPekan8_2511532016 frame = new OperatorAssignmentGUITugasPekan8_2511532016();
36                  frame.setVisible(true);
37              } catch (Exception e) {
38                  e.printStackTrace();
39              }
40          }
41      });
42 }
```

Kode di atas merupakan titik masuk utama program, yaitu metode main() yang bertugas menjalankan aplikasi GUI. Eksekusi program dimulai melalui pemanggilan EventQueue.invokeLater(), yang memastikan bahwa proses pembuatan dan penampilan antarmuka grafis dilakukan di dalam *Event Dispatch Thread*, yaitu thread khusus yang digunakan oleh Swing untuk menangani seluruh aktivitas GUI. Pendekatan ini merupakan praktik yang benar dalam pemrograman Swing untuk menjaga stabilitas antarmuka dan mencegah konflik thread. Di dalam blok run(), dibuat sebuah objek OperatorAssignmentGUITugasPekan8_2511532016 yang mewakili jendela utama aplikasi, kemudian jendela tersebut ditampilkan menggunakan setVisible(true). Seluruh proses pembuatan antarmuka dibungkus dalam struktur try–catch untuk menangani kemungkinan kesalahan yang terjadi pada saat inisialisasi GUI, dan jika terjadi error, detailnya akan dicetak melalui e.printStackTrace(). Dengan demikian, bagian ini memastikan bahwa aplikasi dijalankan dengan aman, terstruktur, dan mengikuti standar eksekusi GUI yang benar dalam Java Swing.

3.4 Konstruktor Kelas dan Komponen-Komponen GUI

```
44     //Create the frame
45     public OperatorAssignmentGUITugasPekan8_2511532016() {
46         setTitle("OPERATOR ASSIGNMENT");
47         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
48         setBounds(100, 100, 394, 295);
49         contentPane = new JPanel();
50         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
51         setContentPane(contentPane);
52         contentPane.setLayout(null);
53
54         JLabel lblNewLabel = new JLabel("OPERATOR ASSIGNMENT");
55         lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
56         lblNewLabel.setFont(new Font("Formula1 Display Bold", Font.PLAIN, 12));
57         lblNewLabel.setBounds(38, 32, 213, 31);
58         contentPane.add(lblNewLabel);
59
60         JLabel lblNewLabel_1 = new JLabel("Angka");
61         lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
62         lblNewLabel_1.setBounds(10, 73, 78, 31);
63         contentPane.add(lblNewLabel_1);
64
65         JLabel lblNewLabel_1_1_1 = new JLabel("Operator");
66         lblNewLabel_1_1_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
67         lblNewLabel_1_1_1.setBounds(10, 126, 78, 31);
68         contentPane.add(lblNewLabel_1_1_1);
69
70         JLabel lblNewLabel_1_1_1_1 = new JLabel("Hasil");
71         lblNewLabel_1_1_1_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
72         lblNewLabel_1_1_1_1.setBounds(10, 167, 78, 31);
73         contentPane.add(lblNewLabel_1_1_1_1);
74
75         txtBill1 = new JTextField();
76         txtBill1.setHorizontalAlignment(SwingConstants.CENTER);
77         txtBill1.setBounds(98, 80, 56, 19);
78         contentPane.add(txtBill1);
79         txtBill1.setColumns(10);
80
81
82         JComboBox cbOperator = new JComboBox();
83         cbOperator.setModel(new DefaultComboBoxModel(new String[]{"+", "-", "*", "/", "%"}));
84         cbOperator.setBounds(98, 132, 41, 21);
85         contentPane.add(cbOperator);
86
87         txtHasil = new JTextField();
88         txtHasil.setEditable(false);
89         txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
90         txtHasil.setColumns(10);
91         txtHasil.setBounds(98, 174, 56, 19);
92         contentPane.add(txtHasil);
93
94         JButton btnNewButton = new JButton("Proses");
```

Bagian kode pada konstruktor OperatorAssignmentGUITugasPekan8_2511532016() ini berfungsi untuk membangun seluruh komponen GUI.

Pertama, jendela diberi judul menggunakan setTitle() dan pengaturan seperti aksi ketika jendela ditutup serta ukuran frame dilakukan melalui setDefaultCloseOperation() dan setBounds(). Setelah itu, sebuah JPanel ditempatkan sebagai *content pane* yang akan menampung semua elemen GUI, dilengkapi *padding* menggunakan EmptyBorder. Panel ini kemudian diatur dengan *layout* null, yang memungkinkan setiap komponen ditempatkan secara manual menggunakan koordinat absolut.

Komponen berikutnya berupa beberapa JLabel yang berfungsi sebagai teks keterangan, seperti judul “OPERATOR ASSIGNMENT”, label input angka, label operator, dan label hasil. Masing-masing label diatur jenis font, posisi, serta ditambahkan ke panel. Di bagian input,

sebuah JTextField disediakan untuk menerima angka dari pengguna, sementara JComboBox disusun berisi daftar lima operator assignment dasar, yaitu +, -, *, /, dan %, yang memungkinkan pengguna memilih operasi yang ingin diterapkan.

Komponen output berupa txtHasil juga dibuat dalam bentuk JTextField tetapi dengan mode *read-only* untuk mencegah pengguna mengubah hasil secara manual.

Terakhir, sebuah tombol JButton bernama “Proses” diletakkan di bagian bawah untuk menjalankan perhitungan ketika ditekan. Secara keseluruhan, konstruktor ini menjadi inti dari pembuatan antarmuka pengguna, mengatur tata letak komponen, serta mempersiapkan struktur utama GUI sebelum logika perhitungan dijalankan.

3.5 Logika Tombol Proses dan Penerapan Operator Assignment

```
96  btnNewButton.addActionListener(new ActionListener() {
97      public void actionPerformed(ActionEvent e) {
98
99          if(txtBill1.getText().trim().isEmpty()) pesanPeringatan("Angka harus diisi");
100         else {
101             int nilai;
102             try {
103                 nilai = Integer.parseInt(txtBill1.getText().trim());
104                 int op = cbOperator.getSelectedIndex(); //memilih operator
105
106             switch (op) {
107                 case 0: hasil += nilai; break; // +=
108                 case 1: hasil -= nilai; break; // -=
109                 case 2: hasil *= nilai; break; // *=
110                 case 3:
111                     if (nilai == 0) {
112                         JOptionPane.showMessageDialog(null, "Tidak bisa bagi dengan 0!");
113                         return;
114                     }
115                     hasil /= nilai;
116                     break; // ==
117                 case 4:
118                     if (nilai == 0) {
119                         JOptionPane.showMessageDialog(null, "Tidak bisa modulus dengan 0!");
120                         return;
121                     }
122                     hasil %= nilai;
123                     break; // !=
124             }
125         } catch (NumberFormatException ex) {
126             pesanError("Input harus berupa angka!");
127         }
128     }
129
130     txtHasil.setText(String.valueOf(hasil));
131 }
132 });
133 btnNewButton.setBounds(166, 132, 63, 21);
134 contentPane.add(btnNewButton);
135
136
137
138 }
```

Bagian kode ini mencakup logika utama program yang berjalan saat pengguna menekan tombol Proses. Di sini, sebuah objek JButton dibuat dengan label “Proses”, lalu kita pasang event handler menggunakan addActionListener() supaya program bisa merespons ketika tombol tersebut diklik. Di dalam metode actionPerformed(), pertama-tama kita periksa apakah kolom input txtBill1 kosong; jika memang kosong, program akan menampilkan pesan peringatan agar pengguna tidak menjalankan operasi tanpa angka. Jika input sudah terisi, program berusaha mengonversi teks menjadi bilangan bulat dengan Integer.parseInt(), dan proses ini dibungkus dalam blok try–catch untuk menangkap kemungkinan kesalahan format input.

Setelah nilai berhasil dibaca, program mengambil operator yang dipilih pengguna dari JComboBox dengan menggunakan getSelectedIndex(). Masing-masing indeks kemudian diproses menggunakan struktur switch, di mana setiap case mewakili satu operator assignment: $+=$, $-=$, $*=$, $/=$, dan $\%=$. Operasi pembagian dan modulus dilengkapi dengan pengecekan khusus untuk mencegah pembagian dengan nol. Jika terjadi, program akan menampilkan pesan kesalahan dan menghentikan eksekusi kalkulasi tanpa mengubah hasil.

Jika semua proses berjalan tanpa error, nilai hasil akan diperbarui sesuai dengan operator yang dipilih dan angka yang dimasukkan. Di akhir, nilai terbaru dari variabel hasil ditampilkan pada txtHasil dengan setText(), sehingga pengguna bisa langsung melihat perubahan nilai setelah setiap operasi dijalankan.

3.6 GUI dari Operator Assignment

