**LAPORAN RESMI**

**MODUL II**

**(EVENT LISTENER)**

**PRAKTIKUM PEMROGRAMAN VISUAL**

**NAMA : DONY EKA OCTAVIAN PUTRA**

**N.R.P : 230441100041**

**DOSEN : FIRMANSYAH ADIPUTRA, ST., M.CS.**

**ASISTEN : MUHAMMAD IQBAL FIRMANSYAH**

**TGL PRAKTIKUM : 02 OKTOBER 2024**

**Disetujui : 08 OKTOBER 2024**

**ASISTEN**

**MUHAMMAD IQBAL FIRMANSYAH   
21.04.411.00084**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Event listener adalah konsep penting dalam pemrograman visual yang memungkinkan aplikasi untuk merespons interaksi pengguna, seperti klik mouse, penekanan tombol, dan gerakan mouse. Pemrograman visual sendiri merupakan pendekatan pengembangan perangkat lunak yang memudahkan pengguna untuk membangun antarmuka pengguna (UI) melalui alat grafis dan metode drag-and-drop, berbeda dengan pemrograman teks tradisional yang mengandalkan penulisan kode manual. Antarmuka Pengguna Grafis (GUI) menjadi komponen utama dalam pemrograman visual, yang memungkinkan interaksi pengguna dengan perangkat lunak melalui elemen-elemen grafis seperti tombol dan menu. Dalam konteks ini, event adalah peristiwa yang terjadi akibat interaksi pengguna, sementara event handling adalah proses untuk merespons peristiwa tersebut.

Event listener berfungsi sebagai objek atau fungsi yang "mendengarkan" untuk peristiwa tertentu dan memberikan respons saat peristiwa itu terjadi. Contoh umum dari event listener meliputi ActionListener, yang menangani klik tombol, dan MouseListener, yang merespons berbagai peristiwa mouse. Cara kerja event listener dimulai dengan pendaftaran pada elemen GUI yang relevan, diikuti oleh pemicu event saat pengguna berinteraksi, dan diakhiri dengan eksekusi kode di dalam event listener. Kelebihan menggunakan event listener antara lain membuat aplikasi lebih responsif, mengorganisir kode dengan memisahkan logika respons dari elemen UI, serta memungkinkan reusabilitas di berbagai elemen UI. Dengan memahami dan menerapkan event listener, pengembang dapat menciptakan aplikasi yang interaktif dan meningkatkan pengalaman pengguna secara keseluruhan.

## **Tujuan**

* Jelaskan mengenai konsep pemrograman Swing dan Sebutkan apa saja komponen swing dalam pemrograman GUI
* Apa yang dimaksud tool gui builder?
* Jelaskan pengertian dan fungsi implementasi event listener dalam pemrograman GUI!

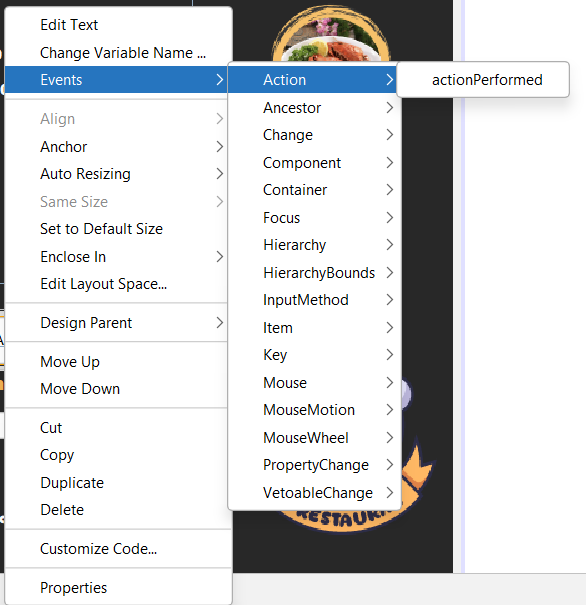
# **BAB II DASAR TEORI**

## **Pengertian Event Handling**

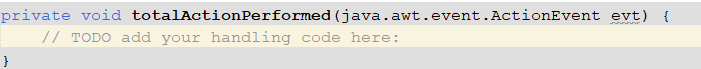
Event Handling merupakan sebuah method yang digenerate oleh pengguna pada saat sesuatu terjadi terhadap suatu komponen. Sebagai contoh, event digenerate pada saat pengguna mengklik sebuah tombol, drag posisi mouse, atau memilih item dari combo box. Event handling dalam menangani event terbagi menjadi 2 macam yaitu Event Source, Event Listener atau Event Handler.

2.1.1 Event Listener

Event listener adalah objek yang diberitahu pada saat suatu event terjadi pada event source. Event listener diimplementasikan ke dalam bentuk interface. Dengan demikian, untuk mengimplementasikan salah satu listener tertentu, perlu mendefinisikan method yang terdapat pada interface bersangkutan. Sebagai contoh, interface ActionListener mendeklarasikan sebuah method abstrak dengan nama actionPerformed(). Jadi, apabila ingin membuat kelas yang mengimplementasikan interface ActionListener, maka harus mendefinisikan method actionPerformed() didalam kelas yang dibuat.



Atau bisa juga dengan meng-klik 2x komponen yang ingin ditambahkan event. Untuk tombol (JButton), event yang terbentuk adalah action  actionPerformed(). Akan muncul bagian source code pada gui seperti berikut.Logika pemrograman dapat dituliskan di method tersebut.

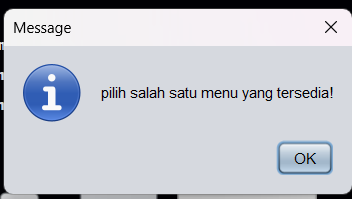


Beberapa komponen swing memiliki method yang bisa digunakan untuk dipakai pada logika pemrograman yang terdapat pada event handler. Beberapa yang paling sering digunakan adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Method | Komponen | Fungsi |
| setText(“X”) | JButton,JTextfield, JLabel | Mengatur tulisan yang muncul pada masing-masing komponen |
| getText() | JButton, JTextfield, JLabel | Mengambil nilai String yang terdapat pada masing masing komponen |
| setToolTipText(“X”) | JButton, JTextfield, JLabel | Mengambil nilai String yang terdapat pada masing masing komponen |
| setEnabled(false) | JButton, JTextfield, JLabel,  JComboBox,JRadioButton,  JcheckBox | Meng-enable suatu komponen atau tidak (dapat di-klik atau tidak) |
| setSelected(true) | JRadioButton, JCheckBox | Membuat masing-masing komponen terpilih atau tidak |
| isSelected() | JRadioButton, JCheckBox | Menge-cek apakah suatu  komponen sedang terpilihatau tidak |
| setSelectedIndex(4) | JcomboBox | Membuat indeks pada angkatertentu sebagai komponenterpilih |
| getSelectedIndex() | JcomboBox | Mengambil indeks terpilih dari komponen.  Indeks dimulai dari 0 |
| getSelectedItem() | JcomboBox | Mengambil objekterpilih  dari komponen. Dapat  langsung ditampilkan jikayang terpilih adalah objekString. |
| insertItemAt(objek, index) | JcomboBox | Memasukkan item pilihanindek) berupa sebuah objek bertipe  Object pada index ke index |
| setValueAt(Objek, baris, kolom) | Jtable | Menge-set table dengan nilai Objek yang bertipe data Object pada baris dan kolom tertentu |
| getSelectedRow() | Jtable | Mengambil indeks baris tabel terpilih. Indeks dimulai dari 0 |
| getSelectedColumn() | JTable | Mengambil indeks kolom tabel terpilih. Indeks dimulai dari 0 |

2.1.2Penggunaan Dialog (JOptionPane)

Dialog di modul ini merujuk pada sebuah komponen yang muncul sebagai informasi atau peringatan bahkan meminta input user setelah melakukan sesuatu.



Pada komponen swing, komponen ini bernama JOptionPane. Terdapat 4 buah option pane:

1. showConfirmDialog : Untuk memberi konfirmasi yes/no/cancel
2. showInputDialog: Untuk menampilkan pop-up sebagai input data
3. showMessageDialog: Untuk menampilkan status/pesan dengan 1 tombol

“OK”.

1. showOptionDialog: Gabungan dari ketiga komponen di atas. Ada kasi

konfirmasi, meminta input dan menampilkan status/pesan

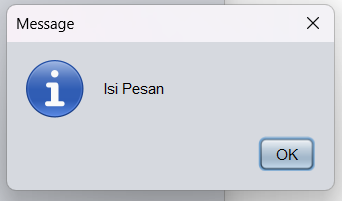
Pada modul praktikum ini hanya dibahas 2 tipe option pane, showConfirmDialog dan showMessageDialog. Setiap option pane dapat dibuat objeknya dengan menggunakan jumlah parameter yang berbeda-beda. Hal ini akan berpengaruh pada tampilan option pane yang dihasilkan.

2.1.3ShowMessageDialog

Contoh Kode showMessageDialog dengan 2 parameter:



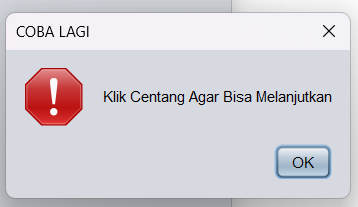
Hasil:



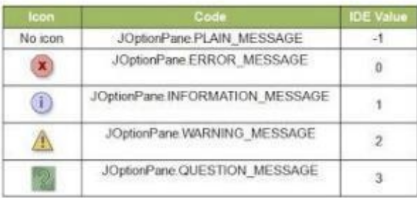
Contoh Kode showMessageDialog dengan 4 parameter:

JOptionPane.showMessageDialog(null, "Isi Pesan", "Judul Pesan", JOptionPane.ERROR\_MESSAGE);

Hasil:



Lambang/symbol/icon yang dihasilkan (silang merah atau huruf “i” dalam lingkaran), tergantung pada parameter terakhir dari contoh kode di atas (untuk kode 4 parameter). Parameter ini disebut “option type”. Jenis-jenis dari option type adalah sebagai berikut:

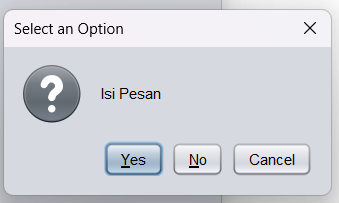


2.1.4ShowConfirmDialog

Contoh Kode showMessageDialog dengan 2 parameter:

JOptionPane.showConfirmDialog(null, "Isi Pesan");

Hasil:



Untuk mengambil nilai dari setiap tombol yang ditekan, gunakan kode sebagaiberikut:

//meminta input yang ditekan sekaligus deklarasi option pane int

hasil = JOptionPane.showConfirmDialog(null, "Isi Pesan");

//membandingkan hasil dengan konstanta yang dimiliki option

pane if(hasil == JOptionPane.YES\_OPTION){

System.out.println("Yes!");

}else if(hasil == JOptionPane.NO\_OPTION){

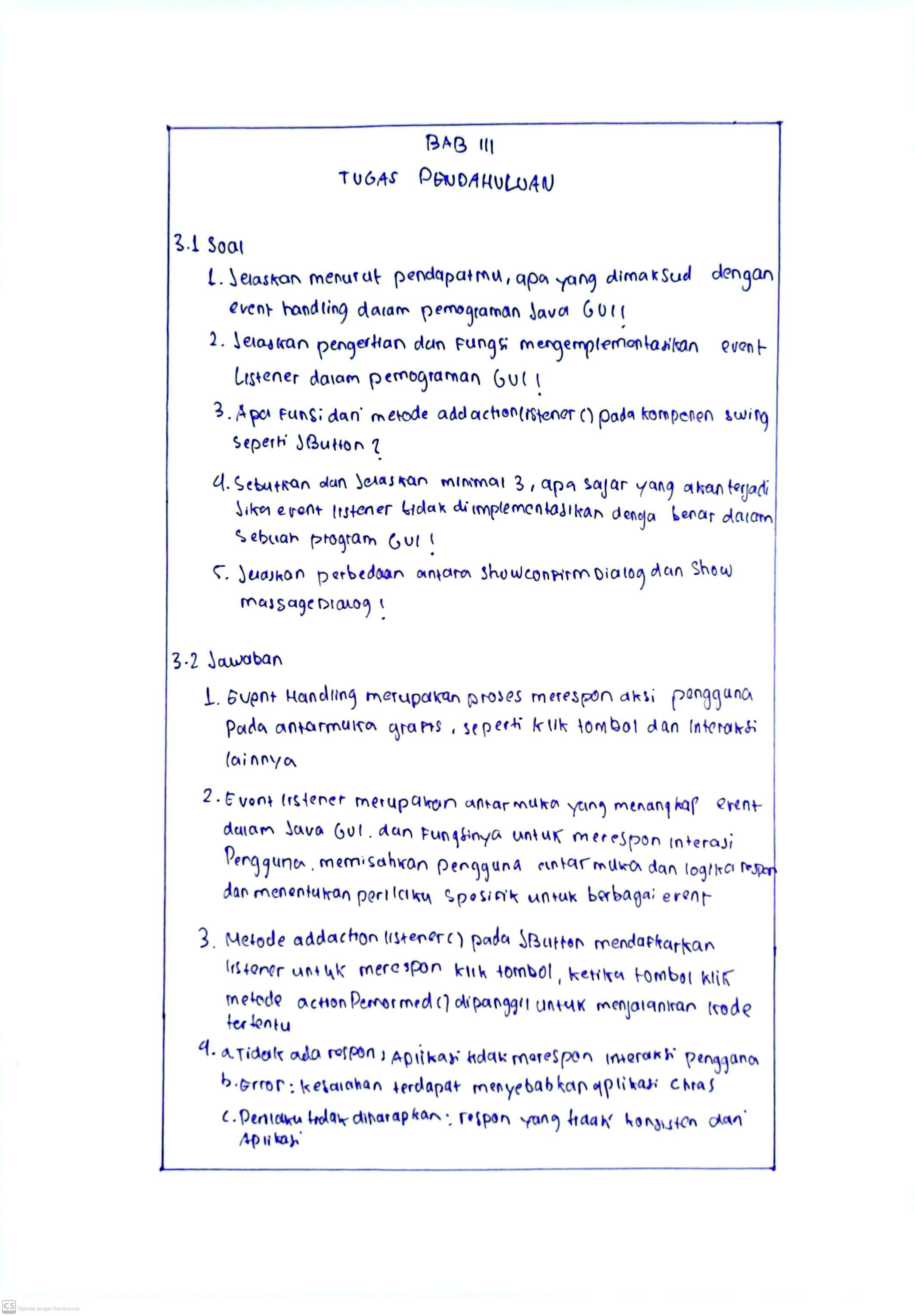
System.out.println("No!");

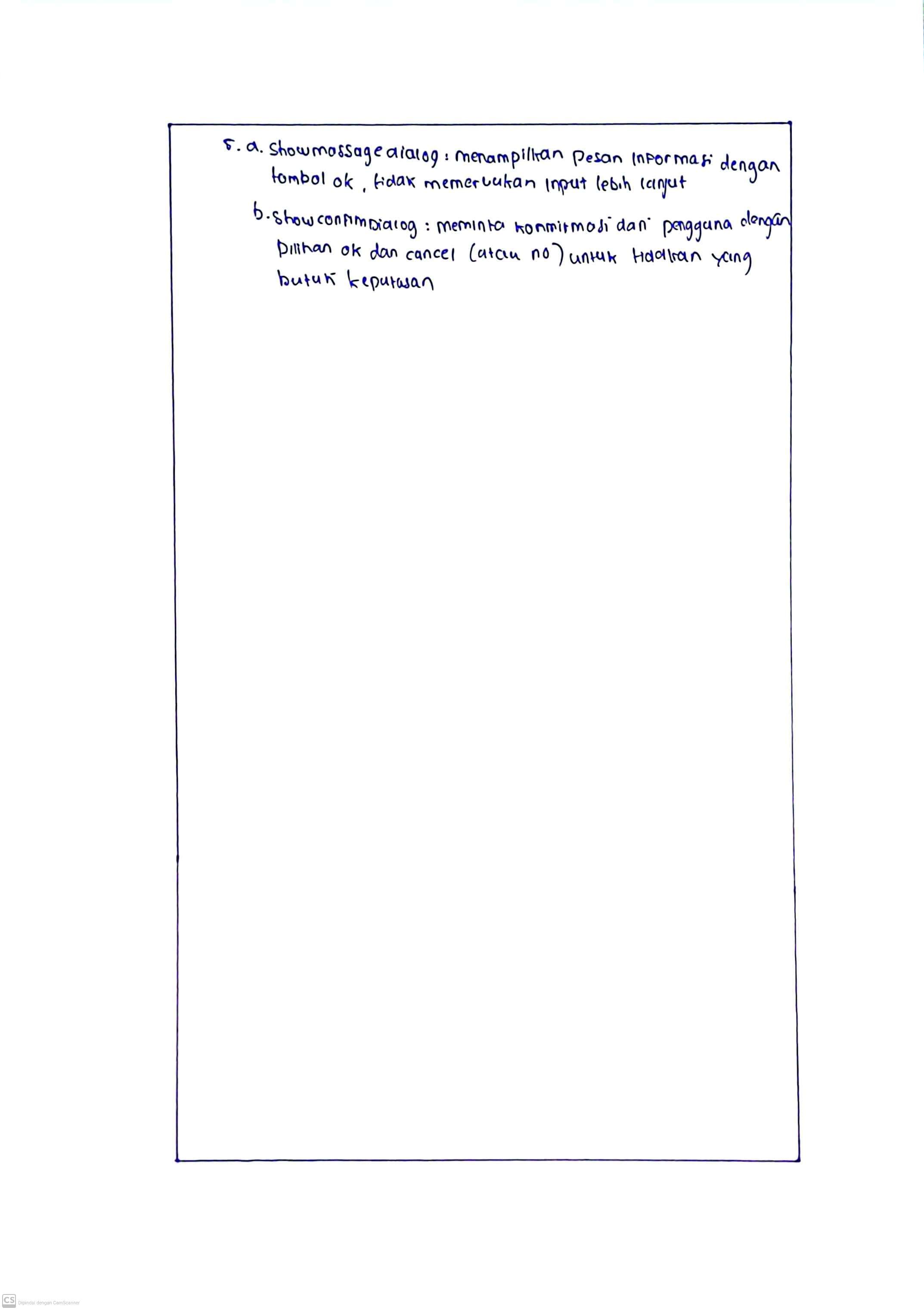
}else if(hasil == JOptionPane.CANCEL\_OPTION){

System.out.println("Cancel!"); }

Pembanding pesan terdiri dari:

* + - YES\_OPTION: Jika memilih tombol “YES”
    - OK\_OPTION: Jika memilih tombol “OK”
    - NO\_OPTION: Jika memilih tombol “NO”
    - CLOSED\_OPTION: Jika memilih tombol “CLOSED”
    - CANCEL\_OPTION: Jika memilih tombol “CANCEL”



****

# **BAB IV IMPLEMENTASI**

## **Source Code**

4.1.1 Soal

1. Buatlah sistem pemesanan sederhana yang terdapat pada sebuah toko roti dengan menggunakan komponen-komponen Swing Java.
2. Buat pilihan 6-8 jenis roti dan tambahkan fitur untuk bisa memilih pilihan tersebut.
3. Tambahkan juga menu 2-4 topping dibawah menu roti yang bisa dipilih.
4. Tambahkan juga harga disamping tiap menunya.
5. Sistem hanya dapat memesan satu menu saja tiap transaksinya (menu roti hanya pilih satu dan topping satu).
6. Sertakan tombol untuk menambah atau mengurangi pesanan.
7. Tambahkan tombol untuk menghitung total harga berdasarkan jumlah pesanan yang dilakukan.
8. Jika total pembelian melebihi Rp. 100.000 maka pembeli berhak mendapatkan diskon 5%.
9. Saat tombol hitung di-klik, jika terdapat menu yang belum dipilih maka akan menampilkan pop-up peringatan untuk memilih menu terlebih dahulu.
10. Tampilkan total harga pada TextField Non-Editable (Tidak bisa diubah nilainya).
11. Tambahkan inputan uang pembeli juga menggunakan TextField dan saat TextField tersebut diisi dengan angka maka otomatis akan menghitung kembaliannya.
12. Tampilkan kembalian pada TextField Non-Editable (Tidak bisa diubah nilainya).
13. Selain itu tambahkan tombol untuk membatalkan pesanan atau menghapusnya.

4.4.2.1 NO 1.

Import Icom :

public class bakar extends javax.swing.JFrame {

   Icon sukses = new javax.swing.ImageIcon(getClass().getResource("/img/check.png"));

    Icon silang = new javax.swing.ImageIcon(getClass().getResource("/img/sign.png"));

Reset :

private void resetActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        grupmenu.clearSelection();

        grupsaus.clearSelection();

        Jpesanan.setText("");

        hasiltotal.setText("");

        isicash.setText("");

        isikembali.setText("");

    }

Total :

private void totalActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        int hasilkabeh = Integer.parseInt(Jpesanan.getText());

        int hargasemua = 0;

        int diskon = 0;

        int hargasaus = 0;

        if(scalop.isSelected()){

            hargasemua=15000 \* hasilkabeh;

        }else if(stikcrab.isSelected()){

            hargasemua=25000 \* hasilkabeh;

        }else if(lobster.isSelected()){

            hargasemua=20000 \* hasilkabeh;

        }else if(oster.isSelected()){

            hargasemua=30000 \* hasilkabeh;

        }else if(sop.isSelected()){

            hargasemua=18000 \* hasilkabeh;

        }else if(squid.isSelected()){

            hargasemua=39000 \* hasilkabeh;

        }else{

            JOptionPane.showMessageDialog(this,"Mohon Pilih Pesanan","COBA LAGI",JOptionPane.WARNING\_MESSAGE,silang);

            return;

    }

        if(padang.isSelected()){

            hargasaus += 5000 \* hasilkabeh;

        }else if (bolones.isSelected()){

            hargasaus += 6000 \* hasilkabeh;

        }else if (tiram.isSelected()){

            hargasaus += 7000 \* hasilkabeh;

    }

    hargasemua += hargasaus;

        if (hargasemua > 100000) {

            diskon = hargasemua \* 5/100;

            hargasemua = hargasemua-diskon;

            JOptionPane.showMessageDialog(this,"Selamat Anda Mendapat Diskon 5%","yeey",JOptionPane.WARNING\_MESSAGE,sukses);

    }

    hasiltotal.setText(String.valueOf(hargasemua));

    isidiskon.setText(String.valueOf(diskon));

    }

Kurang :

private void kurangActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

    int jukuk = Integer.parseInt(Jpesanan.getText());

    if (jukuk > 1){

        jukuk--;

    }

    Jpesanan.setText(String.valueOf(jukuk));

    }

Tambah:

private void tambahActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

    int jukuk;

    if(Jpesanan.getText().isEmpty()){

        jukuk=0;

    }else{

        jukuk=Integer.parseInt(Jpesanan.getText());

    }

    jukuk++;

    Jpesanan.setText(String.valueOf(jukuk));

    }

IsiCash:

private void isicashKeyReleased(java.awt.event.KeyEvent evt) {

        // TODO add your handling code here:

         try{

            int harga\_total = Integer.parseInt(hasiltotal.getText());

            int susuk = Integer.parseInt(isicash.getText());

            if (susuk < harga\_total){

            isikembali.setText("Uang anda kurang");

        }   else {

            int kembalian = susuk - harga\_total;

            isikembali.setText(String.valueOf(kembalian));

        }

        }   catch (NumberFormatException e) {

            isikembali.setText("Masukkan angka saja");

        }

    }

## **4.2 Hasil**

4.2.1 Frame Pertama



## **Penjelasan**

4.3.1 Soal Nomor 1

Pertama saya membuat Jframe dan di dalamnya terdapat Jlabel, RadioButton, ButtonGroup, Button, dan TextFile, kemudian di bagian reset saya mendeklarasikan variabel RadioButton menggunakan fungsi Clearselection dan mendeklarasikan variabael TextFile menggunakan fungsi setText untuk mengeset isi tersebut agar kosong,

Kemudian di bagian total saya mendeklarasikan variabel Jpesanan menjadi integer, dan juga mendeklarasikan variabel hargasemua,diskon, dan harga saus dan dimuali dari 0, lalu saya membuat seleksi kondisi pada menu berfungsi pada saaf kita memilih saat program di run dan di bagian else saya beri JoptionPane agar ketika pengguna tidak memilih akan mendapatkan informasi bahwa pengguna wajib memilih pesanan lebih dulu, kemudian saya juga membuat seleksi kondisi untuk sausnya dan kemudian variabel hargasemua dan hargasaus di jumlahkan, dan juga jika kita membeli di atas Rp.100.000 maka akan mendapatkan diskon 5%

Setelah itu di bagian Button kurang saya mendeklarasikan variabel Jpesanan ke Integer lalu membuat seleksi kondisi jika angka lebih dari 1 maka jika pengguna pencet tanda kurang angka akan mengurang, dan juga di bagian tambah

# **BAB V PENUTUP**

## **Analisa**

Dari hasil praktikum ini pada modul Event Listener dalam Java GUI memberikan pemahaman mendalam tentang bagaimana aplikasi dapat merespons interaksi pengguna. Dengan menggunakan komponen GUI seperti JFrame, JButton, dan JLabel, peserta belajar untuk mengimplementasikan berbagai jenis listener yang memungkinkan aplikasi untuk menjadi lebih interaktif. Konsep dasar yang dipelajari meliputi pengertian berbagai antarmuka listener seperti ActionListener, MouseListener, dan KeyListener, yang masing-masing menangani jenis peristiwa tertentu. Dalam praktikum ini, peserta juga diajarkan cara mendaftarkan listener pada komponen GUI, sehingga ketika pengguna melakukan aksi tertentu, aplikasi dapat memberikan respon yang sesuai.

## **Kesimpulan**

1. Praktikum ini berhasil memberikan pemahaman yang jelas tentang konsep event handling dalam Java GUI, yang esensial untuk menciptakan aplikasi yang responsif.
2. Peserta memperoleh keterampilan praktis dalam mengimplementasikan listener, yang memungkinkan mereka untuk menangani berbagai jenis interaksi pengguna.
3. Melalui penerapan event listener, peserta belajar bahwa desain antarmuka yang baik harus memperhatikan bagaimana pengguna berinteraksi dengan aplikasi, sehingga menciptakan pengalaman pengguna yang lebih baik.
4. Modul ini menjadi dasar yang kuat untuk pengembangan aplikasi GUI lebih lanjut, termasuk eksplorasi framework lain seperti JavaFX.
5. Keterampilan yang diperoleh dari praktikum ini sangat relevan di industri pengembangan perangkat lunak, di mana aplikasi yang interaktif dan responsif sangat dibutuhkan.