**LAPORAN PRAKTIKUM BAHASA PEMROGRAMAN 3**

**MUTIPLE KOMPONEN**

Disusun Untuk Memenuhi Tugas Mata Kuliah Bahasa Pemrograman

Dosen : Agung Nugroho,S.Kom.,M.Kom.



Nama :Fonahia Jisokhi Zagoto

NIM : 311710040

**SEKOLAH TINGGI TEKNOLOGI (STT) PELITA BANGSA**

Jl. Inspeksi Kalimalang Tegal Danas arah Deltamas

Cikarang – Kab.Bekasi

Tahun 2018

**KATA PENGANTAR**

Assalamualaikum Wr.Wb

Alhamdulillah, puji syukur kita panjatkan atas kehadiran Allah SWT, yang di mana telah memberi kita semua kesehatan dan panjang umur sehingga saya selaku penulis dapat menyelesaikan tugas Laporan Praktikum Bahasa Pemograman.

Tak lupa juga saya selaku penulis ingin mengucapkan banyak-banyak terimakasih kepada bapak dosen yang senantiasa mengajarkan ilmu-ilmunya kepada kami semua .

Sebelumnya penulis mohon maaf yang sebesar-besarnya apabila ada kata- kata yang kurang sopan atau kurang berkenan di hati bapak dosen dalam mambuat lapran ini. Dan harapan kami semoga laporan ini bisa mengimbuhkan ilmu yang lebih luas kepada pembaca.

Karna keterbatasan pengetahuan maupun pengalaman kami, kami yakin masih banyak sekali kekurangan dari makalah yang kami buat ini. Oleh karna itu kami sebagai penulis membutuhkan masukan dan kritik yang bersifat membangun yang berasal dari semua pihak, demi perbaikan terhadap tugas selanjutnya.

Wabillahi Taufiq Walhidayah Wassalamualaikum Wr.Wb

Penyusun

**BAB I**

**PENDAHULUAN**

1. **Maksud dan Tujuan Praktikum**

Maksud dan tujuan dari pembuatan laporan Praktikum ini yaitu sebagai berikut :

1. Memahami fungsi JOptionPane pada Netbeans
2. Memahami tentang multiple Komponen, dan Input Dialog pada Java Netbeans
3. Mahasiswa mampu membuat aplikasi desktop sederhana menggunakan multiple komponen dan input dialog.
4. Memahami tentang proses input dan output
5. **Landasan Teori**

Dialog memerankan peran yang penting dalam aplikasi berbasis desktop. Interaksi antara user dengan aplikasi terkadang tidak berjalan dengan baik karena user memberikan aksi yang tidak valid kepada aplikasi. Ketika hal tersebut terjadi, aplikasi harus memberitahukan kepada user apa yang telah terjadi dan bagaimana seharusnya user memperbaikinya. Model interaksi seperti ini tepat dilaksanakan menggunakan dialog.

Skenario lain adalah ketika aplikasi memerlukan input dari user agar aplikasi bisa terus melaksanakan tugasnya, misalnya meminta konfirmasi apakah user yakin akan melaksanakan sebuah aksi penting terhadap aplikasi seperti delete, update atau add data.  
Dialog juga memberikan pembatasan kepada user, sebelum dialog selesai diproses, user tidak akan bisa berinteraksi dengan bagian aplikasi lainya. Dialog mencegah hal ini terjadi dengan memastikan bahwa jendela yang bisa diaktifkan hanyalah jendela dialog, sedangkan jendela aplikasi yang lain tidak dapat diaktifkan selama jendela dialog masih aktif.

Aplikasi sangat sering menggunakan dialog untuk berinteraksi dengan user, tetapi jenis interaksinya selalu seragam dan berulangulang. Swing menyediakan dialog yang didesign untuk keperluan yang sering muncul dalam aplikasi, seperti JOptionPane dan JFileChooser. Swing juga menyediakan class JDialog jika kita ingin membuat dialog custom sesuai keinginan kita.  
Membuat pre-defined dialog dengan JOptionPane  
JOptionPane menyediakan beberapa dialog yang siap pakai dan sering digunakan dalam aplikasi. JOptionPane sangat memudahkan kita dalam meminta user suatu input tertentu atau memberitahu user apa yang terjadi dalam aplikasi.  
JOptionPane mempunyai banyak static method untuk menampilkan popup dialog dengan mudah. Terdapat empat method utama yang dapat kita gunakan sebagai landasan membuat dialog. Keempat method tersebut secara rinci digambarkan dalam table berikut ini:

1. **Komponen JOptionPane**

JOptionPane merupakan sebuah kelas yang menyediakan jendela dialog.

JOptionPane bisa kita gunakan untuk mengambil input, menampilkan informasi, menampilkan pesan error, menampilkan dialog konfirmasi, dan lain-lain.

Sebelum menggunakan JOptionPane, kita harus mengimpornya terlebih dahulu ke dalam program:

import javax.swing.JOptionPane;

Terdapat empat fungsi atau metode yang sering digunakan pada **JOptionPane**:

1. **showConfirmDialog()** : untuk menampilkan dialog konfirmasi;
2. **showInputDialog()** : untuk menampilkan dialog input;
3. **showMessageDialog()** : untuk menampilkan pesan/informasi;
4. **showOptionDialog()** : untuk menampilkan dialog pilihan.

Untuk parameter dari 4 method tersebut mempunyai pattern yang sama. Dan setiap method tersebut memiliki 4 – 7 varian tergantung dari jumlah parameter setiap method ( kecuali showOptionDialog yang cuma memiliki satu variant) .  Berikut ini beberapa parameter yang digunakan pada 4 method tersebut.

parentComponent : mendefinisikan component yang menjadi parent dari dialog box. Jika diisi dengan null, maka secara default Frame akan menjadi parent component.

message : merupakan message yang akan ditampilkan dalam dialog box. Message bisa berupa String. Tapi type sebenarnya dari parameter ini adalah Object class.

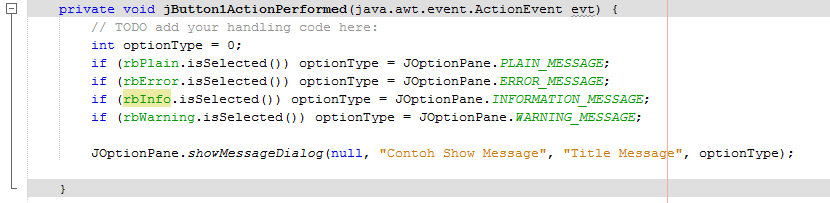
title : merupakan title untuk dialog box

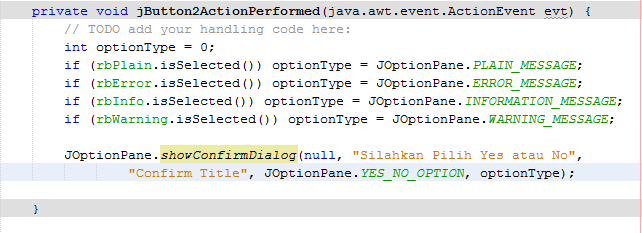
initialValue  : merupakan input value

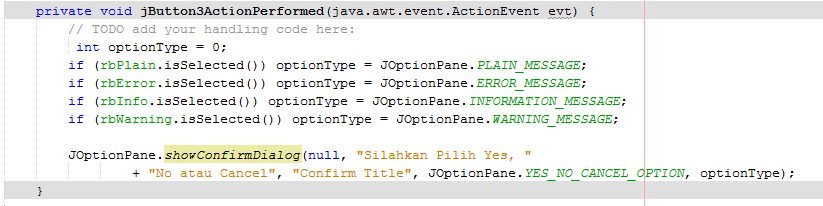
messageType : mendefinisikan style dari message. Jenis dari messageType sebagai berikut ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, PLAIN\_MESSAGE

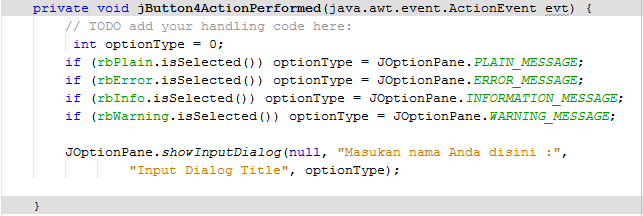
optionType : mendefinisikan kumpulan button option yang muncul pada dialog box. Nilai dari parameter sebagai berikut. DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTION ( Nilai dari variable ini dapat di-customize).

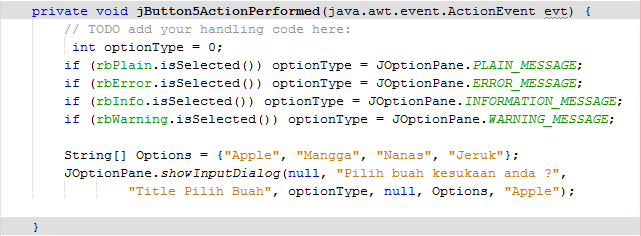
Contoh program :

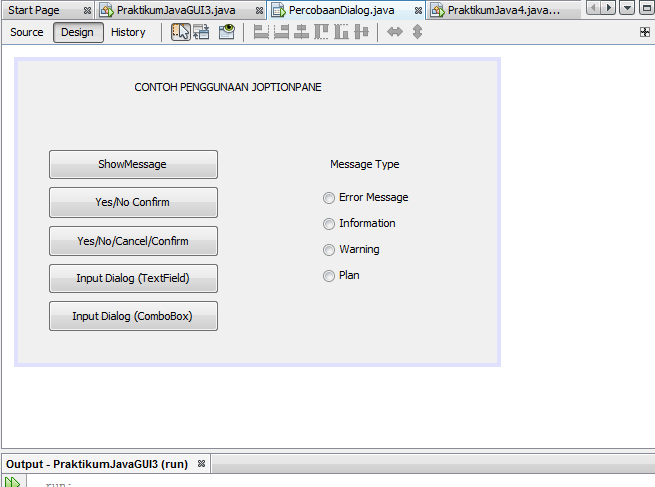




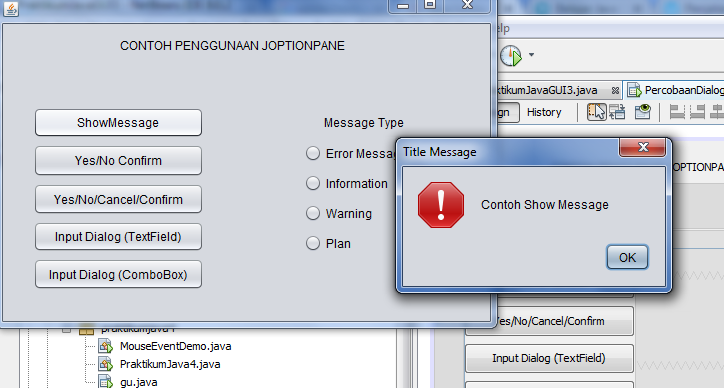




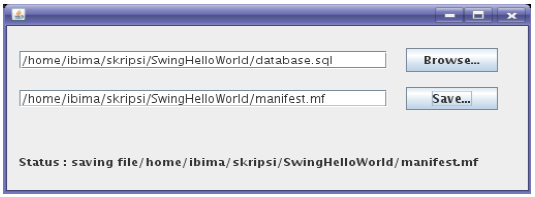




Hasil dari program diatas :

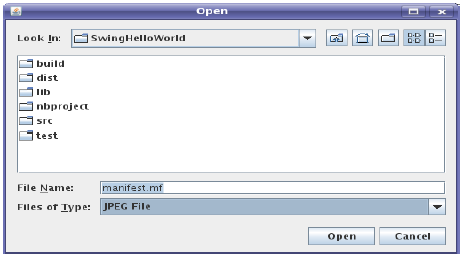


1. **Membuat JFileChooser**  
   JFileChooser digunakan untuk bernavigasi dalam file system, kemudian memilih satu atau lebih file atau folder dari list file dan folder. JFileChooser pada dasarnya adalah pengembangan dari dialog yang dapat digunakan untuk memilih file. JFileChooser dapat digunakan sebagai dialog untuk menyimpan file atau untuk membuka file.  
   JFileChooser hanya memberikan fasilitas untuk memilih file atau folder, sedangkan mekanisme untuk menyimpan atau membuka file dilakukan sendiri menggunakan library I/O.  
   Aplikasi berikut ini adalah contoh penggunaan JFileChooser untuk membuka dan menyimpan file.



                                 Contoh program menggunakan JFileChooser

Tampilan JFileChooser ketika tombol open ditekan adalah seperti di bawah ini :



                                        Tampilan JFileChooser

Untuk membuat aplikasi diatas lakukan langkah-langkah berikut ini :  
1. Buat class JFrame Form baru, beri nama Chooser.java  
2. Masukkan dua buah JTextField : txtOpen dan txtSave, dua buah

Jbutton : btnOpen dan btn save, sebuah JLabel : lblStatus.  
Sesuaikan penataan komponen sesuai dengan gambar diatas.

3. Tambahkan sebuah object JFileChooser sebagai field dari class Chooser, beri nama chooser.

public class Chooser{  
JFileChooser chooser = new JFileChooser();  
//kode lain di sini  
}

4. FileNameExtentionFilter digunakan sebagai file filter dalam JFileChooser. Metode filteringnya adalah mencocokkan ekstensi file dalam file system dengan ekstensi yang ada dalam FileNameExtentionFilter. Contoh kode di bawah ini akan menyebabkan JFileChooser mempunyai pilihan “JPEG File”, dan jika pilihan tersebut dipilih, maka file dengan ekstensi “jpg”, “jpeg”,”JPG” atau“JPEG” saja yang akan ditampilkan oleh JFileChooser.

FileNameExtensionFilter JPEGFilter =  
new FileNameExtensionFilter(  
"JPEG File","jpg","jpeg",”JPG”,”JPEG”);  
chooser.addChoosableFileFilter(JPEGFilter);

5. Set direktori yang akan dituju oleh JFileChooser. Untuk mengetahui dimana direktori aktif aplikasi, kita bisa menggunakan system property “user.dir”. Kode berikut ini akan menyebabkan JFileChooser dibuka pada direktori aktif aplikasi :

String dir = System.getProperty("user.dir");  
chooser.setCurrentDirectory(new File(dir));

6. Menghandle event penekanan tombol btnSave. Ketika tombol  
btnSave ditekan, chooser akan menampilkan dialog save file,  
kemudian mengambil nama file yang dipilih dan  
menampilkannya dalam txtSave, serta menampilkanya dalam  
lblStatus. Berikut ini kodenya :

private void btnSaveActionPerformed(ActionEvent evt) {  
// TODO add your handling code here:  
int ret = chooser.showSaveDialog(this);  
if(ret == JFileChooser.APPROVE\_OPTION){  
File f = chooser.getSelectedFile();  
lblStatus.setText("Status : saving file" +  
f.getAbsolutePath());  
txtSave.setText(f.getAbsolutePath());  
}  
}

7. Menghandle penekanan tombol btnOpen. Kode untuk menangani penekanan tombol btnOpen mirip dengan kode untuk menangani penenakan tombol btnSave, perbedaanya adalah btnOpen akan menampilkan dialog open file, berikit ini kodenya :

private void btnBrowseActionPerformed(ActionEvent evt){  
// TODO add your handling code here:  
int ret = chooser.showOpenDialog(this);  
if(ret == JFileChooser.APPROVE\_OPTION){  
File f = chooser.getSelectedFile();  
lblStatus.setText("Status : opening file" +  
f.getAbsolutePath());  
txtOpen.setText(f.getAbsolutePath());  
}  
}

8. Compile dan jalankan aplikasinya dengan menekan tombol SHIFT + F6

Bekerja dengan JOptionPane dan dengan JFileChooser sangat sederhana. Keduanya menggunakan modal dialog untuk mengambil input dari user. Modal dialog akan mencegah user mengakses bagian aplikasi lain sebelum dialog ditutup, atau dalam hal ini memutuskan pilihan apa yang diambil oleh user.

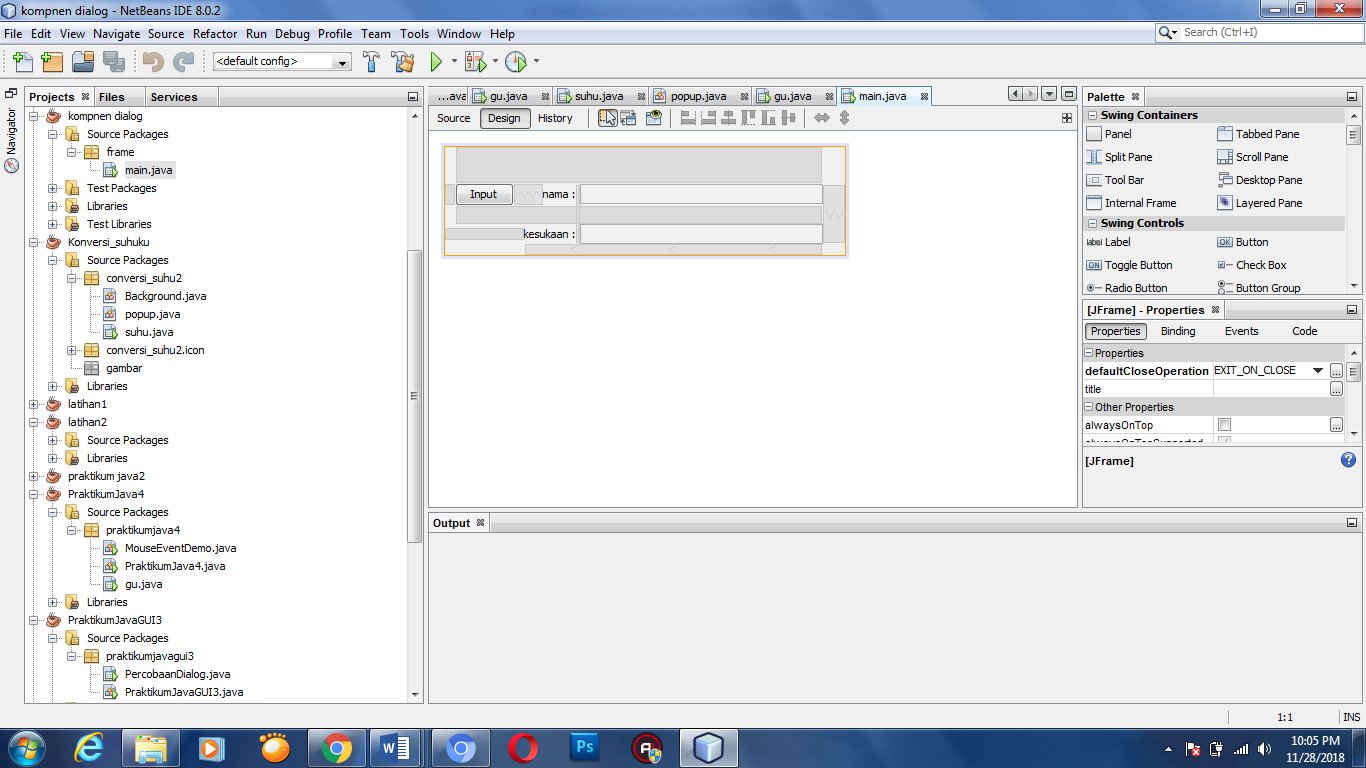
**Latihan II**

1. Membuat Project Baru dengan nama komponen dialog
2. Lalu membuat jframe baru

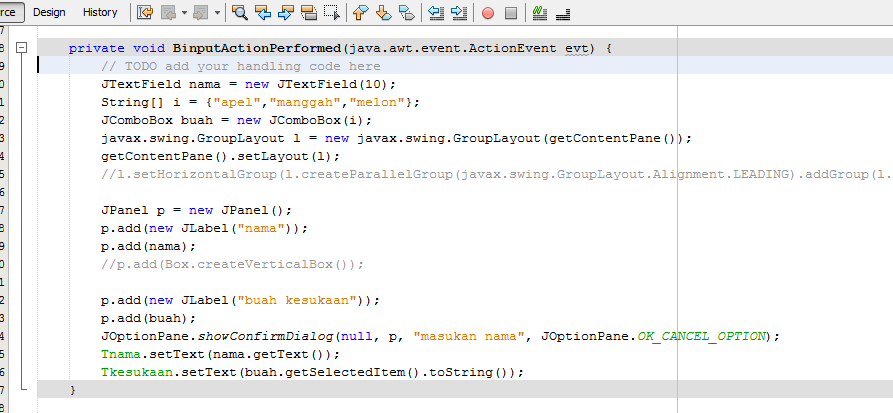
1 Jbutton

2 Jlabel

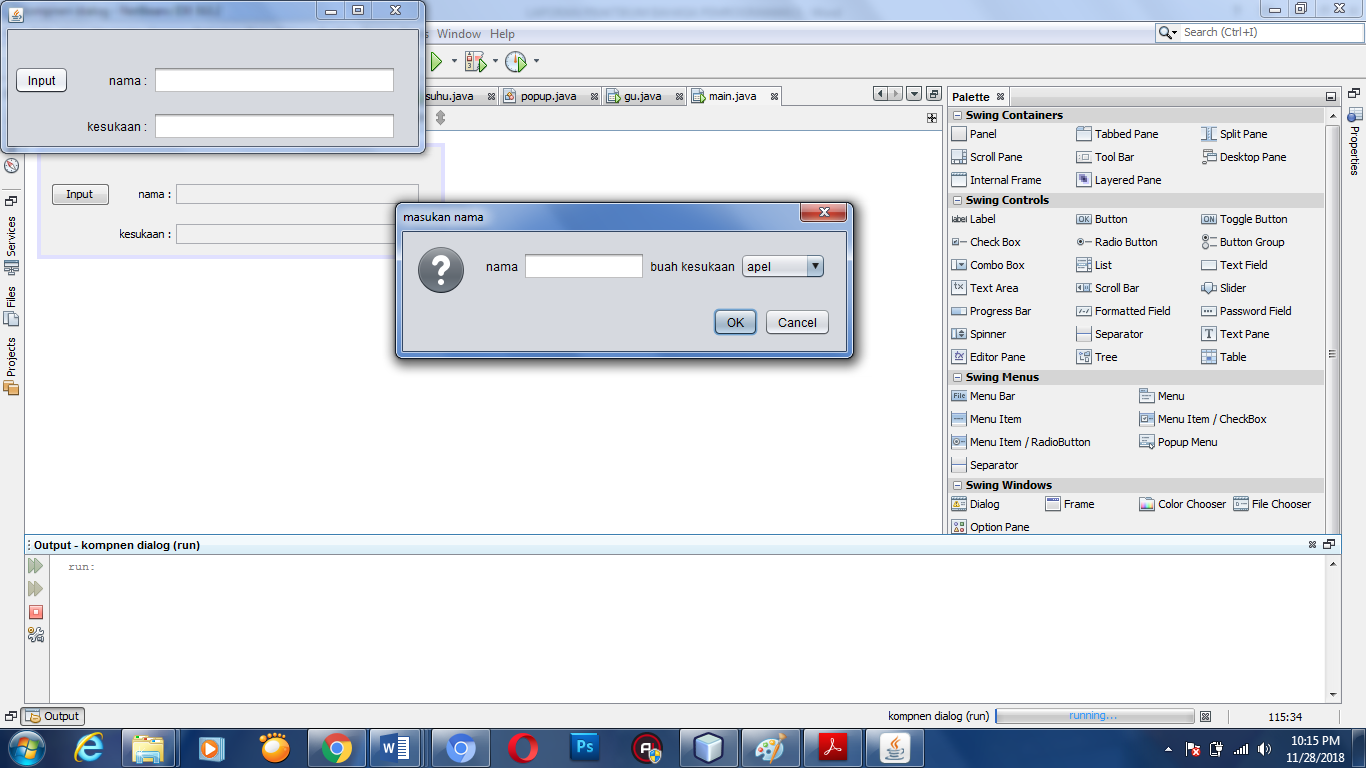
2 TextField



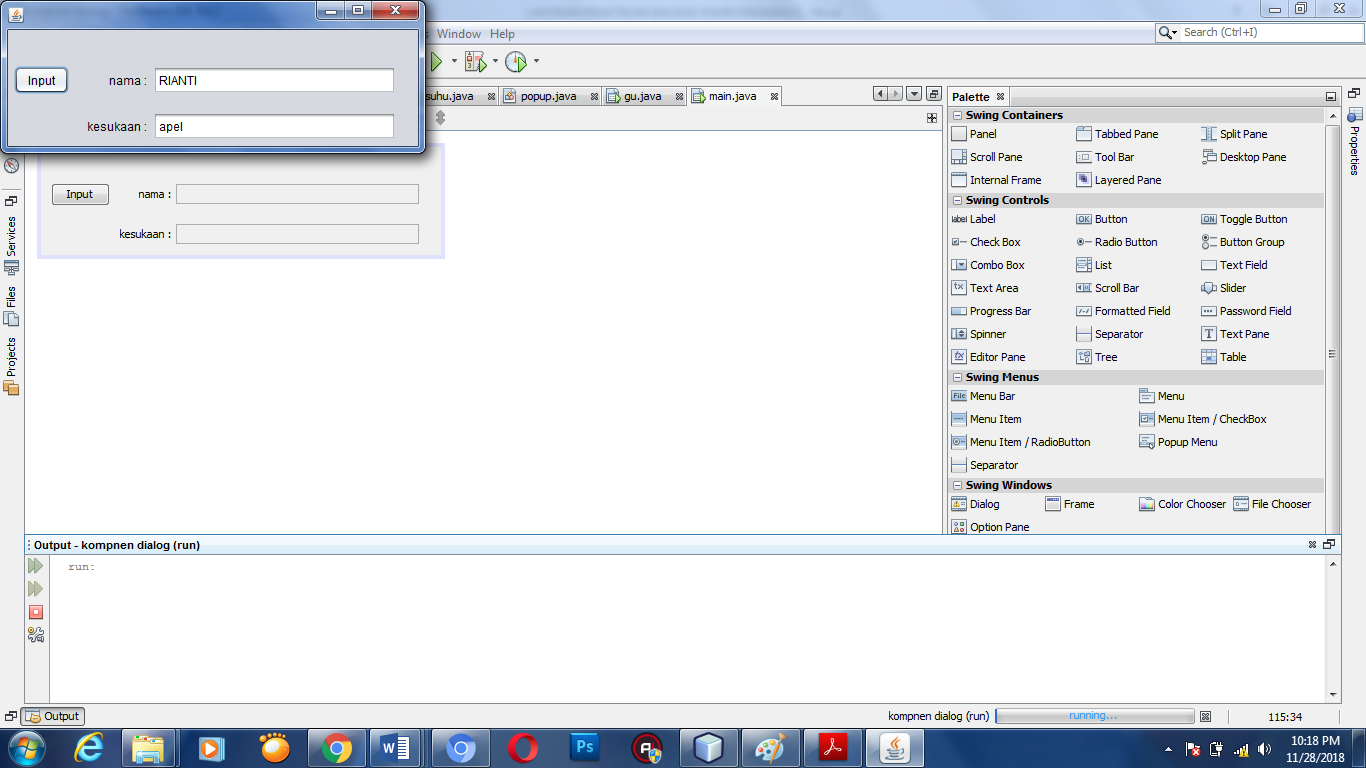
1. Masukan source kode beriku pada jButton Input > Event >Action



1. Berikut ini adalah hasil nya yaitu untuk menampilkan penggunaan Input Dialog dengan multiple komponen



Setelah melakukan input pada textfile di input dialog maka akan muncul di jframe



**BAB III**

**PENUTUP**

**KESIMPULAN**

Dari rangkaian kegiatan praktikum pemrograman java yang telah terselesaikan, maka dapat diambil beberapa kesimpulan diantaranya:

1. Dialog memerankan peran yang penting dalam aplikasi berbasis desktop. Interaksi antara user dengan aplikasi terkadang tidak berjalan dengan baik karena user memberikan aksi yang tidak valid kepada aplikasi. Ketika hal tersebut terjadi, aplikasi harus memberitahukan kepada user apa yang telah terjadi dan bagaimana seharusnya user memperbaikinya. Model interaksi seperti ini tepat dilaksanakan menggunakan dialog.
2. Bekerja dengan JOptionPane dan dengan JFileChooser sangat sederhana. Keduanya menggunakan modal dialog untuk mengambil input dari user. Modal dialog akan mencegah user mengakses bagian aplikasi lain sebelum dialog ditutup, atau dalam hal ini memutuskan pilihan apa yang diambil oleh user.
3. Dengan dilakukannya sebuah praktikum maka kita akan mengetahui tentang cara membuat sebuah program aplikasi berbasis Java. Dan kita juga harus memperhatikan script yang kita tulis atau ketikkan. Dalam bahasa pemrograman Java, kita akan menggunakan pernyataan masukan dan keluaran. Pernyataan masukan adalah pernyataan untuk mendapatkan masukan dari keyboard. Sedangkan pernyataan keluaran adalah pernyataan untuk menampilkan sesuatu nilai ke layar.