**LAPORAN PRAKTIKUM VII**

**FUNDAMENTAL OF PROGRAMMING WORKSHOP**



DOSEN PENGAJAR

Indrawati, SST.MT

**DISUSUN OLEH**

**Kelompok 3:**

**Nasywa Deby Azanna**

**Kelas : TRKJ-1B**

**POLITEKNIK NEGERI LHOKSEUMAWE**

**JURUSAN TEKNOLOGI KOMUNIKASI DAN KOMPUTER**

**PRODI D-IV TEKNOLOGI REKAYASA KOMPUTER DAN JARINGAN**

**2022/2023**

* Tujuan Pratikum
* Mengidentifikasi macam-macam fungsi dari Java
* Mengimplementasikan bahasa pemprograman java
* Membiasakan diri menggunakan Java
* Dasar Teori

A. Pengertian Java

JAVA merupakan sebuah bahasa pemrograman yang dijalankan di berbagai komputer dan pada handphone. Java pertama kali dikembangkan oleh Sun Microsystems yang menjelaskan bahwa bahasa java dapat digambarkan sebagai a simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dan dynamic language. Sebuah program dalam bahasa java dirakit (compiled) menjadi sebuah berkas byte-code.

B. Pengertian GUI

GUI ( *Graphical User Interface* ) adalah metode interaksi secara grafis antara pengguna dengan komputer yang berfungsi memudahkan pengguna dalam menggunakan computer secara maksimal. Contoh penggunaan jenis GUI : Microsoft Windows, MacOS dan Xwin.

C. Pengertian Java GUI

JAVA GUI sendiri secara umum sama dengan pemahaman GUI sendiri, namun disini GUI pada JAVA adalah salah satu dari fasilitas Java dalam mengembangkan suatu aplikasi dengan memberikan suatu interface yang nantinya akan disesuaikan dengan kebutuhan suatu aplikasi, baik skala kecil dan besar. Intinya, GUI pada Java adalah komponen-komponen penyempurna dalam sebuah sistem, seperti komponen navigasi, graphic, interface desktop, komponen menu, sinkronisasi antar interface dan semuanya itu dirancang dengan penelitian yang baik, dimana nantinya aplikasi tersebut dapat dengan mudah dioperasikan user. Banyak sudah aplikasi yang telah Java kembangankan dengan menggunakan GUI sebagai nilai penggunaan dan nilai jual. Software perangkat lunak dan keras juga termasuk di dalamnya. Java GUI adalah program pengembangan suatu user interface dengan mengetengahkan graphic sebagai penghubung user dengan sistem, sehingga mudah dalam implementasi aplikasi sistem tersebut.

D. Komponen penyusun Java GUI

Komponen Utama GUI java adalah JFC (Java Fundamental Class)

* Koordinat GUI
* AWT (Abstract Windows Toolkit) merupakan paket dasar dalam pembentukan GUI dengan menggunakan platform yang terdapat dalam masing-masing OS (native resource) Paket dasar : java.awt.component
* Swing Product SUN dalam membangun GUI yang Independent sehingga bersifat cross platform dalam tampilan window yang dihasilkan. Paket dasar: javax.swing.Jcomponent
* Event Handler Aksi-reaksi yang diberikan pada tiap-tiap komponen GUI

E. Fungsi GUI

Dari pemahaman tentang GUI sebelumnya, fungsi-fungsi GUI dalam dunia teknologi dan informasi saat ini sangat beragam. Berikut beberapa fungsi dari penggunaan GUI dalam dunia Iptek khususnya komputer :

* Pointing device dengan perangkat berupa mouse
* Memiliki fungsi sebagai plug and play, yang mana memudahkan user dalam menginstall dan uninstall hardware/sofware (sinkronisasi)
* Membuat software-software yang menuntut aplikasi dalam komputer user yang dapat digunakan untuk kegiatan user dalam proses seperti menulis text, spreadsheet, sound, graphic dan lain-lain
* Untuk digunakan dalam pembuatan System Operasi sebagai User friendly dan ergonomis
* Pembuatan website
* Opensource software/hardware atau pengembang aplikasi-aplikasi skala besar lainnya.

F. Macam – macam GUI

Ini adalah beberapa macam contoh GUI pada Java :

**AWT (Abstract Window Toolkit) – SUN Microsystem -> Oracle Product**

AWT (Abstract Window Toolkit) ini adalah GUI Toolkit pertama pada bahasa pemrograman Java, sayang-nya AWT ini sangat-sangat kekurangan komponen yang biasa digunakan untuk membangun sebuah aplikasi desktop secara lengkap (komponen tabel saja tidak ada). Terlepas dari kurang-nya komponen GUI yang terdapat pada AWT (Abstract Window Toolkit), aplikasi yang dibangun menggunakan AWT (Abstract Window Toolkit) akan tampak seperti aplikasi native. Maksudnya yaitu, jika aplikasi yang dibangun menggunakan AWT (Abstract Window Toolkit) ini dijalankan pada Sistem Operasi Windows. Maka aplikasi ini akan terlihat seperti aplikasi Windows pada umum-nya, dan begitu juga jika dijalankan pada Sistem Operasi Mac ataupun GNU/Linux. Kenapa ini bisa terjadi, karena AWT (Abstract Window Toolkit) ini benar-benar memanggil native subrutin untuk menggambar setiap komponen-nya ke layar.

**v Java Swing – SUN Microsystem -> Oracle**

GUI Toolkit yang mungkin paling banyak dikenal oleh teman-teman yang baru belajar Pemrograman GUI di Java dibandingkan dengan GUI Toolkit yang lain GUI Toolkit ini lebih banyak dikenal dikarenakan dukungan tutorialnya yang cukup banyak bertebaran di Internet, dan juga merupakan standart dari Java yang mana kita tidak perlu melakukan penambahan library lagi kalau ingin menggunakan GUI Toolkit ini. Selain itu, terdapat 2 IDE besar yang menggunakan GUI Toolkit Java Swing yaitu NetBeans IDE(OpenSource) dan IntelliJ IDEA (mempunyai versi OpenSource dan Komersial)

Dibandingkan dengan pendahulu-nya yaitu AWT (Abstract Window Toolkit), Swing mempunyai lebih banyak komponen pendukung untuk membangun sebuah aplikasi yang lengkap untuk keperluan desktop. Selain didukung dengan banyak-nya komponen, Swing ini benar-benar murni 100 % ditulis dengan bahasa pemrograman Java tanpa adanya sebuah wrapper untuk memanggil rutin-rutin native code via JNI (Java Native Interface). Seluruh komponen yang terdapat pada Swing, semuanya murni digambar sendiri menggunakan API (Application Programming Interface) 2D untuk memanggil rutin-rutin dasar penggambaran komponen-nya.

Selain itu, Swing juga mempunyai kemampuan untuk berganti-ganti tampilan menggunakan LAF (Look And Feel) atau themes Sayang-nya, jika kita menginginkan tampilan GUI yang native (tampilan-nya sama seperti aplikasi-aplikasi lain pada sistem operasi target) Swing seperti-nya masih terasa kurang smooth terutama dukungan pada font rendering-nya.

**SWT (Standart Widget Tookit) – IBM Product -> Eclipse Foundation**

SWT (Standart Widget Toolkit) ini adalah sebuah GUI Toolkit yang dikeluaran oleh IBM sebagai alternatif dari AWT/Java Swing milik SUN Microsystem, yang membedakan antara SWT (Standart Widget Toolkit) dan AWT/Java Swing adalah SWT ini benar-benar mengakses native GUI library yang terdapat pada Sistem Operasi melalui JNI (Java Native Interface). Dengan model seperti ini, memungkinkan tampilan aplikasi yang dibangun menggunakan GUI Toolkit SWT menjadi sama persis dengan aplikasi native lain-nya.

**QtJambi – Trolltech -> Nokia Product -> Stopped and Taken By Community**

QtJambi ini merupakan binding Qt Framework dengan bahasa Java, tetapi sayang-nya proyek QtJambi sudah tidak disupport oleh Nokia dan secara resmi telah ditutup Untung-nya, awal tahun ini ada beberapa developer yang peduli dengan kelangsungan proyek ini dan akhir-nya membuat sebuah komunitas untuk melanjutkan pengembangan proyek QtJambi.

**JavaGNOME – Community Product**

Kalau QtJambi diatas ditujukan untuk teman-teman yang sudah akrab dengan API (Application Programming Interface) Qt Framework, berbeda dengan JavaGNOME Proyek ini lebih dikhususkan untuk teman-teman pecinta GTK atau yang paling banyak dikenal yaitu GNOMESama seperti GUI Toolkit SWT dan QTJambi, JavaGNOME ini juga mengakses native library tetapi API yang digunakan adalah API dari GTK. Untuk teman-teman yang sudah terbiasa membangun aplikasi menggunakan Glade, maka teman-teman bisa men-design form-nya menggunakan Glade dan memanggil-nya menggunakan bahasa java melalui JavaGNOME.

G. Kelebihan dan Kekurangan GUI

· Kelebihan GUI :

o Desain Grafis lebih menarik.

o GUI memungkinkan user untuk berinteraksi dengan komputer secara lebih baik.

o Memudahkan pengguna.

o Menarik minat pengguna.

o Resolusi gambar yang tinggi.

· Kekurangan GUI :

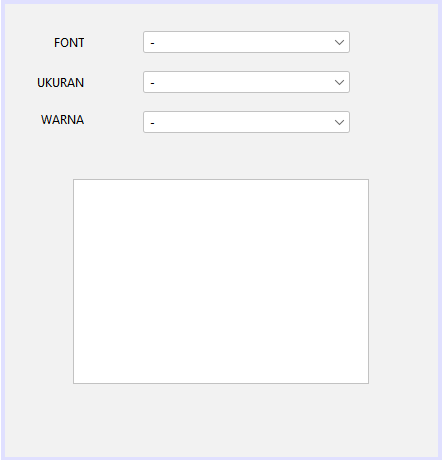
o Memakan memory yang sangat besar.

o Bergantung pada perangkat keras.

o Membutuhkan banyak tempat pada layar komputer.

o Tidak fleksibel.

* Alat dan Bahan
* Laptop
* NeatBeans
* Langkah Percobaan
* Membuat tampilan form seperti berikut ini



* Menambahkan kode berikut pada tombol “ComboBox1” yang digunakan untuk memilih Font

if(jComboBox1.getSelectedItem()=="Arial")

{ jTextArea1.setFont(new Font("Arial",Font.PLAIN,12));

}else if(jComboBox1.getSelectedItem()=="Verdana"){

jTextArea1.setFont(new Font("Verdana",Font.PLAIN,12));

}else if(jComboBox1.getSelectedItem()=="Tahoma"){

jTextArea1.setFont(new Font("Tahoma",Font.PLAIN,12));

}

* Menambahkan kode berikut pada tombol “ComboBox2” yang digunakan untuk memilih Ukuran

if(jComboBox2.getSelectedItem()=="14"){

jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),14));}

else if(jComboBox2.getSelectedItem()=="16"){

jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),16));}

else if(jComboBox2.getSelectedItem()=="18"){

jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),18));

}

* Menambahkan kode berikut pada tombol “ComboBox3” yang digunakan untuk memilih Warna

if(jComboBox3.getSelectedItem()=="Merah"){

jTextArea1.setForeground(Color.red);}

else if(jComboBox3.getSelectedItem()=="Hijau"){

jTextArea1.setForeground(Color.green);}

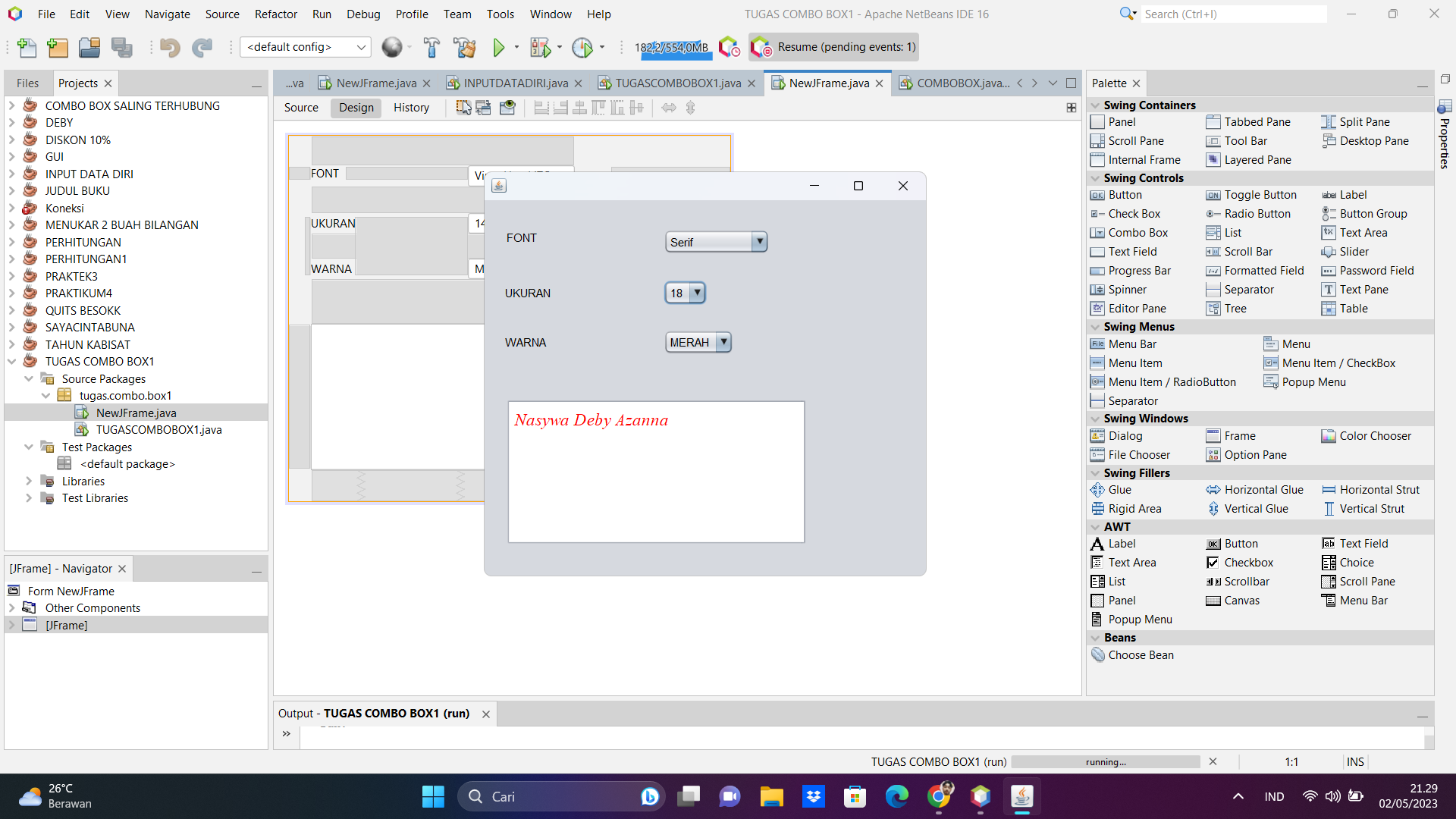
else if(jComboBox3.getSelectedItem()=="Biru"){

jTextArea1.setForeground(Color.blue);

}

* Hasil dan Analisa

Hasil:



Analisa:

|  |  |
| --- | --- |
| if(jComboBox1.getSelectedItem()=="Arial")  { jTextArea1.setFont(new Font("Arial",Font.PLAIN,12));  }else if(jComboBox1.getSelectedItem()=="Verdana"){  jTextArea1.setFont(new Font("Verdana",Font.PLAIN,12));  }else if(jComboBox1.getSelectedItem()=="Tahoma"){  jTextArea1.setFont(new Font("Tahoma",Font.PLAIN,12));  } | Jika salah satu pilihan yang terdapat pada comboBox terpilih, maka tulisan yang ada pada jtextarea1 akan berubah sesuai dengan pilihannya |
| if(jComboBox2.getSelectedItem()=="14"){  jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),14));}  else if(jComboBox2.getSelectedItem()=="16"){  jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),16));}  else if(jComboBox2.getSelectedItem()=="18"){  jTextArea1.setFont(new Font(jTextArea1.getFont().getName(),jTextArea1.getFont().getStyle(),18));  } | Jika salah satu pilihan ukuran yang terdapat pada comboBox terpilih, maka tulisan yang ada pada jtextarea1 akan berubah sesuai dengan ukuran dari pilihannya |
| if(jComboBox3.getSelectedItem()=="Merah"){  jTextArea1.setForeground(Color.red);}  else if(jComboBox3.getSelectedItem()=="Hijau"){  jTextArea1.setForeground(Color.green);}  else if(jComboBox3.getSelectedItem()=="Biru"){  jTextArea1.setForeground(Color.blue);  } | Jika salah satu pilihan warna yang terdapat pada comboBox terpilih, maka tulisan yang ada pada jtextarea1 akan berubah sesuai dengan warna dari pilihannya |

Kesimpulan

* Pada Java GUI kita bisa mendeklarasikan font yang kita inginkan dengan berbagai macam model dan ukuran
* ComboBox berfungsi untuk memberikan beberapa pilihan
* Jika ingin menggunakan Font lain, harus menggunakan librarynya, sama juga hal nya dengan warna dan ukuran