Business Application Development – UAS

Nama : Reynaldi O. Rasubala

NIM : 2301906363

1. [LO 2 & L0 3, 20 poin] Berikan penjelasan dari Event Driven Programming, serta berikan contoh program untuk mendeteksi aksi pada klik mouse dan pergerakan mouse.

Event – Driven Programming merupakan sebuah *design pattern* arsitektur yang berfungsi untuk membangun aplikasi perangkat lunak dimana komponen dari *runtime* membuka dan menanggapi respon dari *event* tersebut. Cara kerja dari Event Driven Programming sendiri adalah mengumpulkan kegiatan – kegiatan (ActionEvent, MouseEvent, KeyEvent, dll) yang dilakukan lalu mengirimkan-nya ke Event-Handler melalui *scheduler*. Scheduler ini akan bersifat selalu terbuka sampai ditutup oleh *event* lainnya. Adapun beberapa istilah yang digunakan dalam Event Driven Programming antara lain seperti Event Object, Event Source, dan Event Listener.

Event Object merupakan Events yang dikemas menjadi sebuah objek ketika *event* tersebut dibuat berhubung konsep Event Driven Programming dibangun menggunakan konsep Object – Oriented Programming. Event Source merupakan objek yang memicu *event*. Misalnya, *event* klik dari sebuah tombol pada aplikasi. Event Listener sendiri merupakan bagian dari kode *runtime* yang bersifat terbuka dan mendengar jika adanya *event* yang masuk.

Untuk contoh bentuk penerapan dari Event Driven Programming pada program untuk mendeteksi aksi klik dan pergerakan mouse adalah sebagai berikut:

|  |  |
| --- | --- |
| Graphical user interface, application  Description automatically generated  Figure 1 Tampilan Awal | Figure 2 Tampilan Mouse Memasuki Tombol |
| Figure 3 Tampilan Mouse Keluar Tombol | Figure 4 Tampilan Mouse Klik Tombol |

*Figure 1* menunjukkan tampilan awal dari program. Untuk mendeteksi pergerakan *mouse* jika *mouse* memasuki tombol “Click Me” maka tampilan program akan seperti *figure* 2. Sebaliknya jika *mouse* menjauhi tombol maka akan ditampilkan seperti *figure 3*. Jika tombol “Click Me” sendiri di tekan maka akan ditampilkan seperti *figure 4.*

1. [LO 3 & L0 4, 20 poin] Berikan penjelasan dan contoh program untuk membuat Menu dan Sub Menu berikut:

Dalam membuat menu seperti contoh gambar yang diberikan maka langkah – langkahnya adalah sebagai berikut:

1. Pertama, kita perlu menentukan terlebih dahulu jenis layout yang akan digunakan berhubung Menu dan Sub – Menu merupakan Header dari program yang artinya bersifat tetap walaupun halaman dari program tersebut berganti – ganti. Disini saya menggunakan BorderPane sebagai *layout* utamanya.

Figure 5 BorderPane Layout

1. Kedua, membuat MenuBar dan menentukan isi dari masing – masing Sub Menu. MenuBar merupakan salah satu komponen dari JavaFX yang tujuannya untuk menyediakan *drop down* menu seperti kebanyakan aplikasi *desktop* pada umumnya yang terletak pada posisi kiri atas dari jendela aplikasi. Menu sendiri merupakan *pop up menu* yang dalamnya terdapat beberapa MenuItem yang akan ditampilkan ketika pengguna mengklik menu tersebut. Setelah atribut – atribut yang diperlukan telah di deklarasikan maka kita tinggal menambahkan menu – menu tersebut ke MenuBar dan menentukan *layout*-nya di Top.

Text

Description automatically generated

Figure 6 MenuBar

1. Ketiga, kita tinggal perlu menambahkan listener untuk setiap masing – masing dari menu yang di klik. Untuk cara penggunaannya kurang lebih seperti pada *figure 7.*

Text

Description automatically generated

Figure 7 Action Listener

1. [LO 3 & L0 4, 10 poin] Berikan urutan langkah-langkah program Java untuk dapat membangun koneksi ke Database MySql/Oracle/Sql Server (bebas) dan melalukan query ke tabel Mahasiswa serta menampilkan isi tabel tersebut.

Dalam membangun koneksi Database ke MySQL langkah – langkahnya pengerjaannya adalah sebagai berikut:

* 1. Pertama, sebelum kita mengkonfigurasi Database-nya melalui kodingan terdapat *library external* **mysql-connector-java-8.0.24.jar** yang perlu kita import terlebih dahulu. Cara mengimportnya sendiri pada IDE Eclipse adalah **klik kanan** pada folder project **→ Build Path → Configure Build Path → Libraries → Add External JARs →** Kemudian pilih **mysql-connector-java-8.0.24.jar**. **→ Apply and Close**.
  2. Kedua, setelah *external library* berhasil di *import* maka selanjutnya kita perlu mengkonfigurasi Database-nya. Disini untuk konfigurasi-nya biasa dibuat kelas terpisah. Sebagai contoh saya akan menggunakan nama kelas Connect untuk konfigurasi DB-nya. Adapun tahapan pembuatan konfigurasi DB-nya antara lain adalah sebagai berikut:
     1. Mendeklarasikan variabel – variabel yang akan digunakan. Seperti pada *figure 8* variabel Connection digunakan untuk menghubungkan program dengan local database yang telah dibuat pada MySQL. Statement digunakan untuk mengeksekusi *query* pada *database* yang dimana memerlukan Connection terlebih dahulu agar dapat mengeksekusi *query* dan ResultSet merupakan hasil dari *query* yang berhasil di eksekusi yang dibalikkan dalam bentuk ResultSet Object.

Text

Description automatically generated

Figure 8 Pendeklarasian Variabel

* + 1. Kedua, dilanjutkan dengan pembuatan *constructor* dari Connect dan metode dengan *return type* kelas Connect. Adanya pembuatan *constructor* Connect untuk menghubungkan koneksi dengan database. Seperti pada contoh *figure 9* pada *constructor* Connect terdapat pengambilan koneksi untuk database “Universitas”. Dasarnya cukup dengan adanya *constructor* ini kita sudah menghubungkan program dengan *database*. Namun, kode program ini kurang-lah bagus dikarenakan setiap kali kita akan mengeksekusi *query* maka kita akan selalu membuat koneksi yang baru. Maka dari itu, terdapat tambahan metode bernama getConnection() yang bertujuan untuk melakukan validasi jika kelas Connect telah dibuat sebelumnya atau tidak. Jika sudah ada maka program cukup memanggil koneksi yang telah dibuatkan tersebut.

Graphical user interface, text, application

Description automatically generated

Figure 9 Membuat Metode Connect

* + 1. Ketiga, setelah koneksi *database* terhubung maka saat-nya kita melakukan *query* untuk menampilkan seluruh data yang ada pada tabel **Mahasiswa**.

Graphical user interface, application

Description automatically generated

Figure 10 Pemanggilan Query

Text

Description automatically generated

Figure 11 Kelas Mahasiswa

Untuk melakukan eksekusi *query* maka kita perlu memanggil kelas Connect terlebih dahulu seperti pada *line 5*. Hasil dari eksekusi *query* akan kita tampung ke dalam tipe data ResultSet. Jika hasil eksekusi berhasil maka untuk menampilkan data – data tersebut perlu kita tampung satu – satu menggunakan *loop* dimana kondisinya selama masih terdapat data dalam ResultSet seperti pada *line 13*. Dalam *loop* tersebut kita tambahkan data yang didapat dari *database* ke dalam Vector/ArrayList dengan tipe data yang ditampung adalah objek Mahasiswa (Isi dari kelas Mahasiswa dapat dilihat pada *figure 11*).

1. [LO 4 & L0 5, 50 poin] Buatlah 2 buah form GUI, pertama adalah form GUI Login ke database untuk dapat masuk ke form ke dua, yaitu form Entri Database Mahasiswa, di mana terdapat fitur entri nama, alamat, no telp, Jurusan dan fasilitas button simpan data, Jika data sudah tersimpan maka akan muncul pesan Messagebox bahwa data sudah disimpan. Berikan kode lengkap serta tampilan hasilnya.

Graphical user interface

Description automatically generated

Figure 12 Form Login

Untuk form login diberikan *username: reynaldi.rasubala* dan *password: 123456* seperti pada *figure 12*.

Graphical user interface

Description automatically generated

Figure 13 Form Entri Database Mahasiswa

Graphical user interface

Description automatically generated

Figure 14 Data Berhasil Disimpan

Graphical user interface

Description automatically generated

Figure 15 Tampilan Data Setelah Ditambahkan