LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

ENKAPSULASI dan GUI



Oleh :

Tiara Azizah

2411533001

MATA KULIAH ALGORITMA PEMROGRAMAN

DOSEN PENGAMPU : DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, DESEMBER 2024

1. **Pendahuluan**
2. **Enkapsulasi**

Enkapsulasi adalah mekanisme pembungkus yang bisa menyembunyikan informasi. Mekanisme ini membungkus semua informasi menjadi satu seperti pembungkus kapsul. Enkapsulasi merujuk pada penggabungan metode di dalam satu kelas. Hal ini bisa mencegah kelas luar mengakses dan mengubah metode suatu kelas, dengan kata lain enkapsulasi membantu penyembunyian data.

Kelas data yang ada di Java akan bersifat privat atau tersembunyi ketika semua kelas data di buat menjadi terekapsulasi. Menggunakan metode setter dan getter membuat pengguna bisa mengatur dan mengambil data didalamnya. Salah satu contoh kelas yang terenkapsulasi sepenuhnya adalah kelas Java Bean.

1. **Switch Case**

Kode switch-case pada java sama halnya seperti pernyataan kondisional yang menguji beberapa kondisi dan memberikan satu bentuk output. Kode switch-case mengambil Kesimpulan berdasarkan nilai dari suatu variabel. Dibandingkan dengan menggunakan if-else, struktur ini lebih rapi digunakan ketika menggunakan banyak kondisi yang akan dinilai.

1. Pemograman Berorientasi Objek(OOP)

OOP adalah pendekatan pemrograman yang memanfaatkan objek untuk mengelola data dan fungsi secara modular. Konsep utama dalam OOP meliputi enkapsulasi, pewarisan, polimorfisme, dan abstraksi.

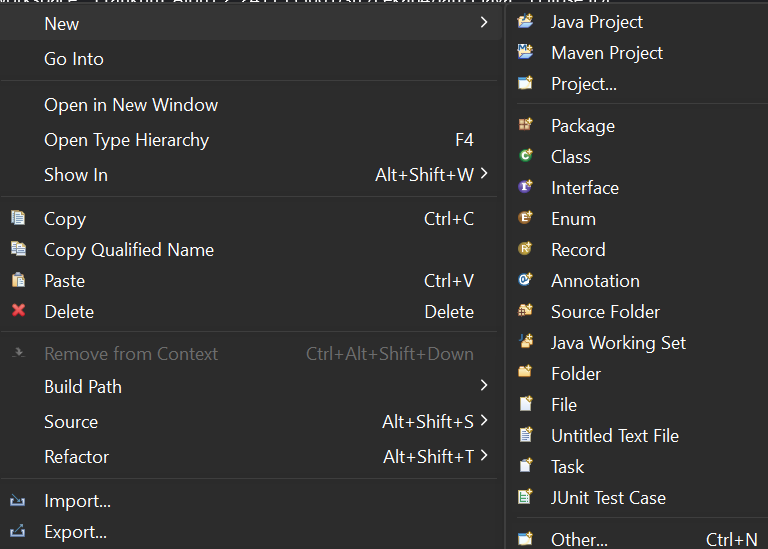
1. **GUI**

GUI adalah pemograman antarmuka menggunakan metode interaksi dengan tampilan visual yang menghubungkan komputer dan penggunanya dengan menggunakan Bahasa pemograman java untuk menampilkan visual dari pogramnya.

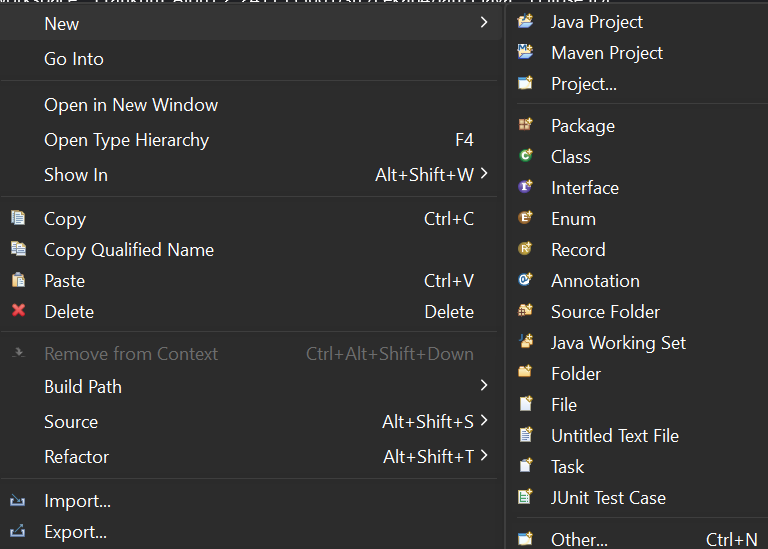
1. **Tujuan**
2. Memahami konsep enkapsulasi dan penerapan dalam pogram berorientasi objek.
3. Mempelajari metode getter dan setter menyediakan akses baca-saja dan tulis-saja di bidang kelas.
4. Menerapkan struktur kontorl switch-case untuk membuat logika kondisional.
5. Mendesain suatu *interface* sederhana pada Java.
6. **Langkah Kerja Pembuatan Enkapsulasi**

Pratikum di pekan tujuh berisikan tentang pengimplementasi sederhana penggunaan enkapsulasi. Program yang akan dibuat pada kali ini berisikan tentang saling berkaitan antarclassnya.

1. Pembuatan class enkapsulasi
2. Pertama buatlah package seperti biasa di java, beri nama package “Pekan7”



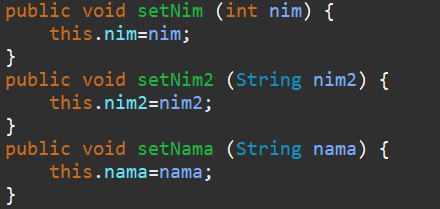
1. Kemudian tambahkan class di package yang telah dibuat. Beri nama sesuai dengan kebutuhan atau keinginan. Dalam kasus ini class diberi nama “mahasiswa”.



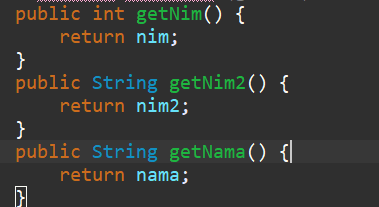
1. Buat variabel global yang hanya bisa diaksses oleh metode kelas “mahasiswa”. Pembuatan variabel merupakan salah satu bagian dari pembuatan enkapsulasi untuk melindungi data dari akses di luar kelas.



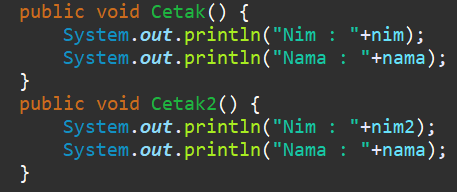
1. Tambahkan mutator *Setter*. Mutator setter bertujuan untuk pengubahan nilai atribut dari luar kelas dengan validasi yang diperlukan. “this” pada pogram digunakan untuk pembeda antara variabel kelas dan parameter metode.



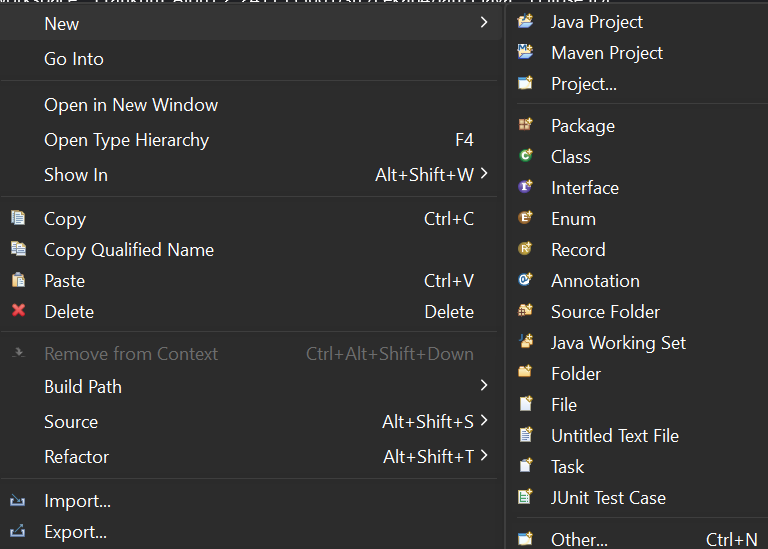
1. Setelah memuat mutator *Setter*, langakah selanjutnya yaitu pembuatan accessor *Getter* yang bertujuan untuk pengambilan nilai atribut dari luar kelas dengan kontrol tambahan.



1. Kemudian buatlah perintah untuk mencetak. Perintah “cetak” berujuan untuk mencetak metode nim dan nama di tipe *int*. Sedangkan “cetak2” akan mencetak menggunakan metode nim2 dan nama pada tipe *String*.



1. Langkah Menggunakan Kelas “Mahasiswa”
2. Langkah pertama buat class di package yang sama dengan class “mahasiswa”. Beri nama dengan kebutuhan yang ada.



1. Setelah membuat metode “main”, buat objek “a” dari class “mahasiswa”. Guna membuat objek adalah memanggil metode dalam class “mahasiswa”.

****

1. Atur data “mahasiswa” menggunakan *setter*. Pada class yang ke dua ini akan mengatur variabel nim dengan angka yang telah ditentukan dan nama yang sudah ditetapkan.



1. Kemudian ambil data mahasiswa menggunakan *getter.* Mengambil variabel nim dan nama yang telah disimpan pada Langkah sebelumnya.



1. Terakhir memanggil metode cetak dengan kode di bawah ini,



1. Output dari langkah kerja tadi adalah sebagai berikut,



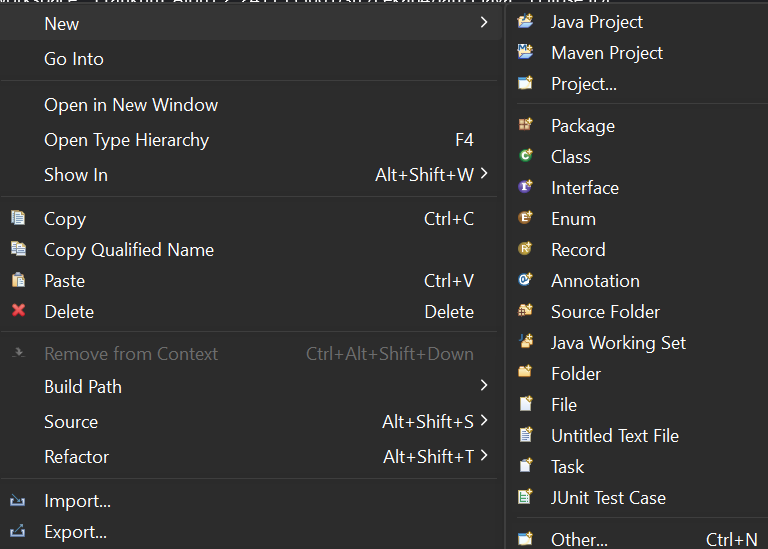
Bagian ini adalah output dari pogram tanpa memanggil metode cetak. Jika menggunakan metode cetak output akan menjadi seperti gambar di bawah ini.



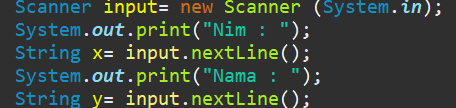
1. Penggunaan Kelas “Mahasiswa”

Pogram berikut ini merupakan cara lain menggunakan kelas enkapsulasi “mahasiswa”. Berikut langkahnya.

1. Buat kelas baru pada package yang sama. Beri nama sesuai kebutuhan atau keiinginan.



1. Buat scanner agar pogram bisa membaca data yang diinputkan oleh pengguna dengan mengikuti cara berikut ini.



1. Kemudian membuat objek “a” dari kelas “mahasiswa”.





1. Buat kode penyimpanan data ke objek dengan cara menggunakan metode *setter*.



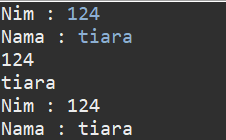
1. Setelah data disimpan, tambahkan metode getter untuk menampilkan data yang telah disimpan sebelumnya.



1. Setelah selesai dengan semua metode di atas, terakhir buatlah metode cetak2. Guna memanggil metode cetak2 adalah mencetak informasi yang telah diinputkan oleh pengguna menggunakan atribut *String*.



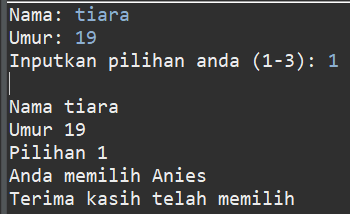
1. Untuk output pogram ini adalah sebagai berikut ini,



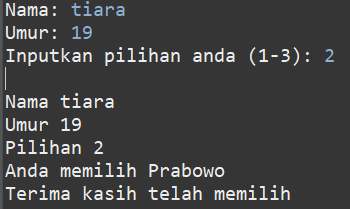
1. Pemilu

Class ke empat pada package pekan tujuh ini, berisikan tentang bagaimana cara mengaplikasikan pogram yang menggunakan switch-case. Berikut langkahnya

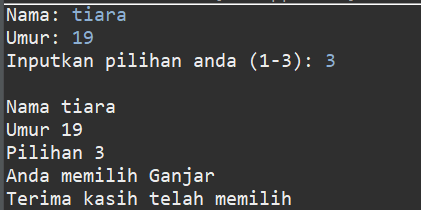
1. Buatlah class baru terlebih dahulu di package tujuh. Beri nama class sesuai kebutuhan. Pada pratikum kali ini karena akan membuat salah satu bentuk pengaplikasian switch-case pogram yang paling cocok adalah pemilu. Maka class ini akan diberi nama class pemilu.
2. Tambahkan scanner agar pogram bisa membaca inputan dari pengguna
3. Setelah menambahkan scanner, buatlah perintah pada pogram untuk memperlihatkan data yang telah dimasukan oleh pengguna.
4. Setelah menambahkan perintah pada pogram, buatlah logika pengambilan keputusan menggunakan switch-case.
5. Setelah menyelesaikan semua kode tutuplah scanner agar tidak terjadi kebocoran sumber daya.
6. Untuk tambahan terakhir buatlah perintah pada pogram untuk mencetak pesan penutup.
7. Output untuk pogram pada class pemilu adalah sebagai berikut



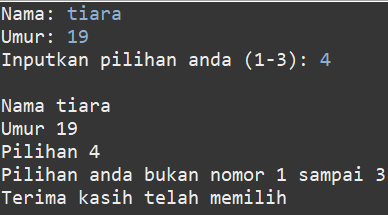
Output pertama merupakan logika yang dihasilkan ketika pengguna menginputkan data pilihan satu.



Output kedua merupakan logika yang dihasilkan ketika pengguna menginputkan data pilihan dua.



Output ketiga merupakan logika yang dihasilkan ketika pengguna menginputkan data pilihan tiga.

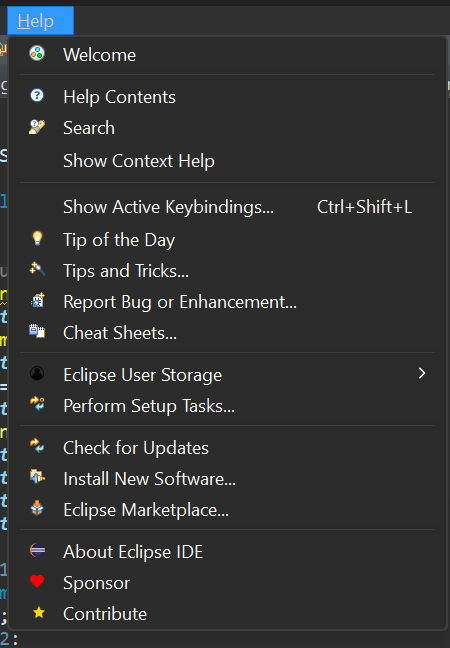


Output terakhir merupakan logika yang dihasilkan ketika pengguna menginputkan data pilihan yang tidak termasuk pada rentan logika yang didaftarkan.

1. GUI

Untuk menggunakan GUI pertama kali di aplikasi eclipse, terlebih dahulu harus menginstal fitur tambahan Bernama WindowBuilder. Berikut langkah mengistal WindowBuilder.

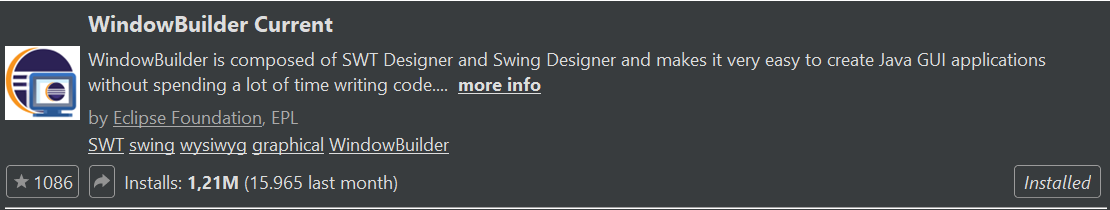
1. Pertama buka aplikasi eclipse di komputer ataupun laptop.
2. Selanjutnya, klik menu “Help” di bagian atas eclipse. Pilih eclipse marketplace.



1. Di kolom pencarian eclipse marketplace, ketikan WindowBuilder. Kemudian pilih tombol “Go” untuk memulai pencarian.



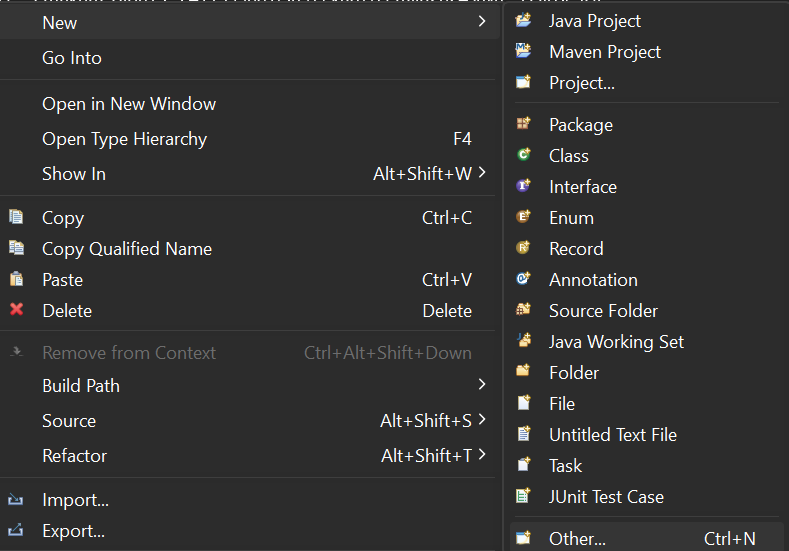
1. Setelah muncul daftar hasil, tarik ke bawah untuk menemukan plugin dengan nama WindowBuilder. Ketika telah menemukannya klik tombol *instal* pada bagian sebelah plugin.



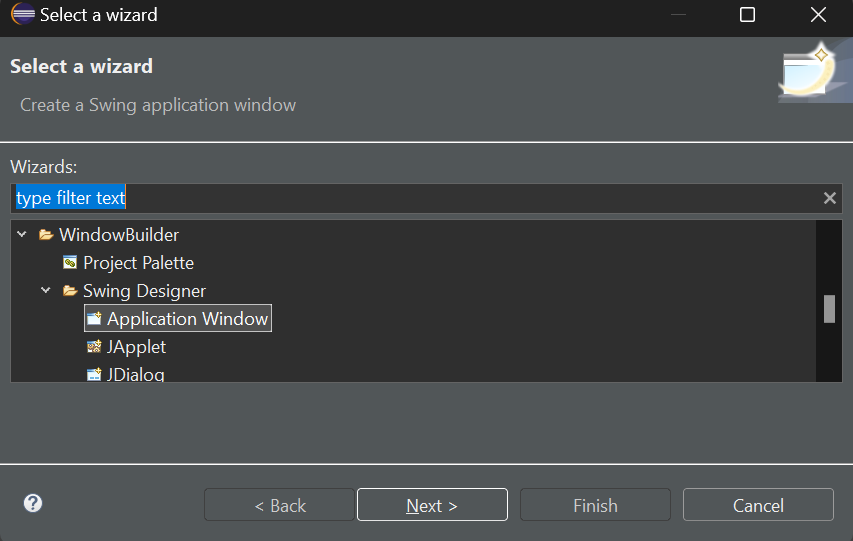
1. Tunggu proses instalasi. Setelah proses instalasi selesai akan muncul dialog berisikan perjanjian mempercayai sertifikat, pilih Trsut All atau OK.
2. Setelah menyetujui sertifikat, *restart* eclipse untuk menerapkan perubahan.

Setelah mengistal fitur Windowbuilder pada eclips langkah selanjut pada pratikum ini adalah mempraktekan cara menggunakan fitur ini. Berikut langkahnya,

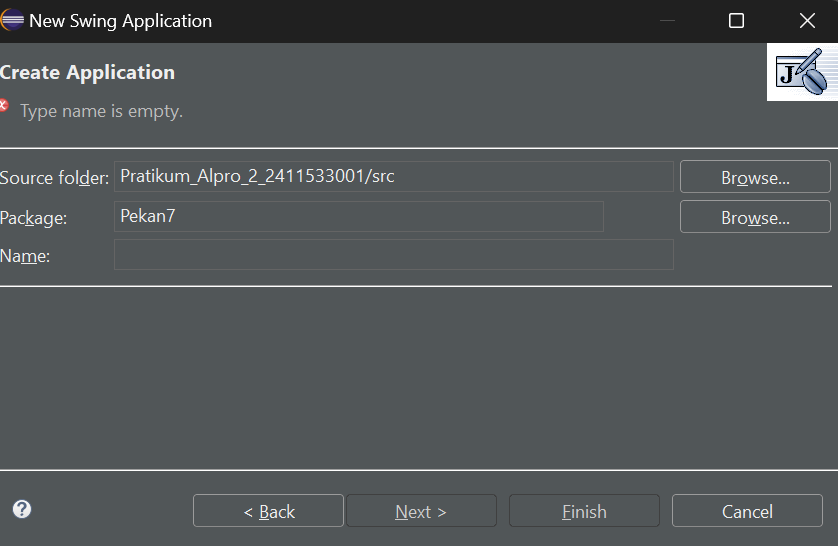
1. Klik kanan pada menu *package*, pilih *other*.



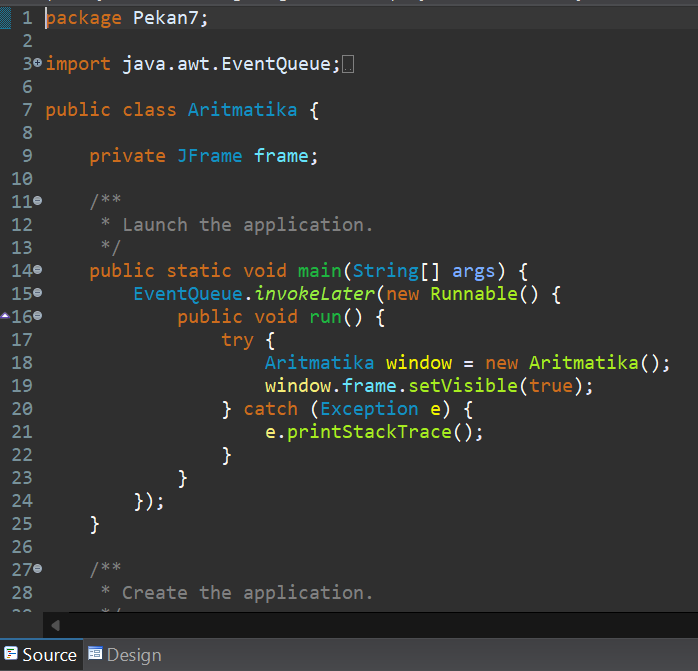
1. Setelah memilih menu *other*, pengguna akan diarahkan ke halaman menu berikutnya. Pilih swing desainer kemudian pilih *application window*.



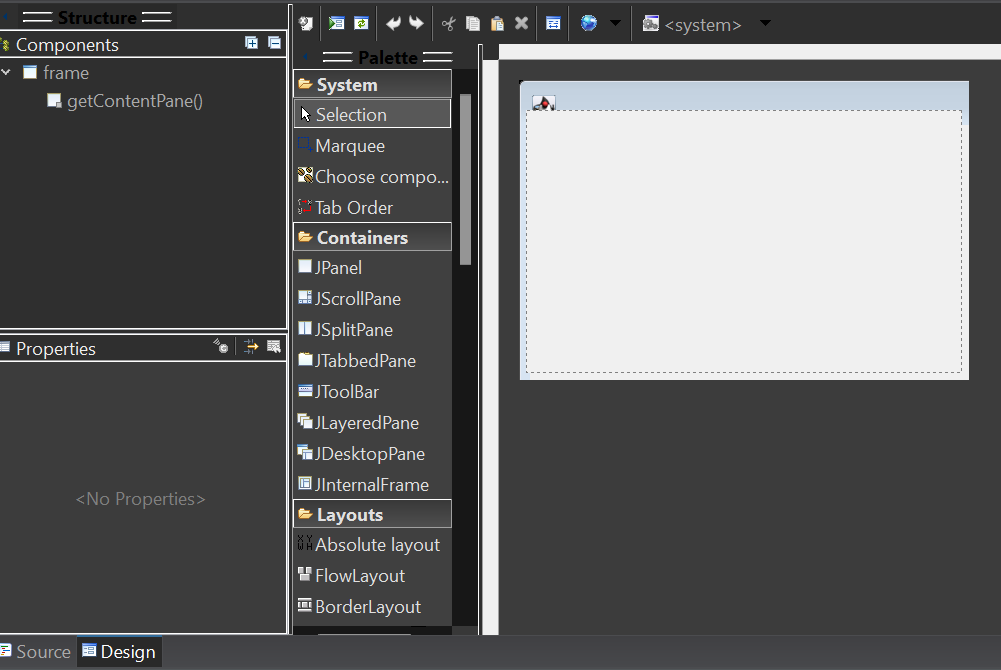
1. Berikutnya beri nama untuk pogram pada tampilan menu seperti ini,



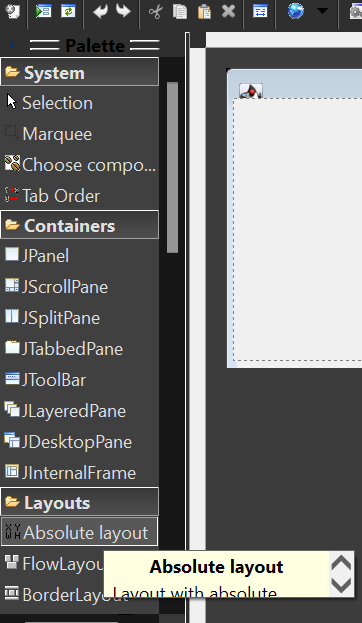
Setelah memberi nama, maka tampilan eclipse akan seperti gambar di bawah ini.



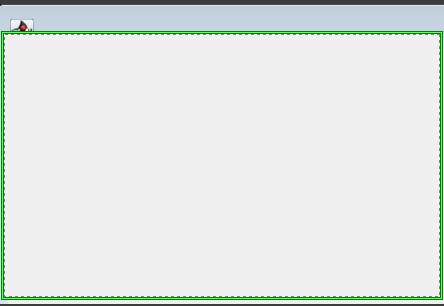
1. Pilih menu *design* yang ada di bagian bawah. Pogram akan membawa pengguna pada menu untuk membuat desain aplikasi antarmuka.



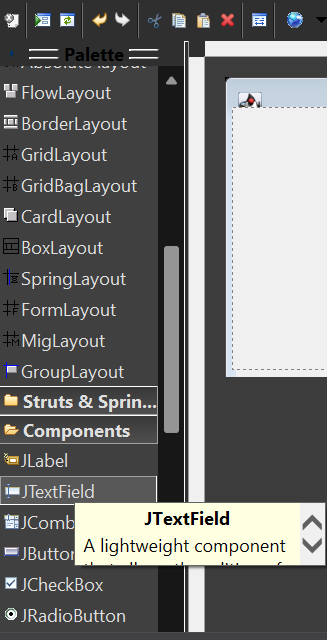
1. Ketika tampilan menu sudah berubah seperti ini, gerek bagian fitur pada kolom di bawah ini, pilih *absolute layout*.



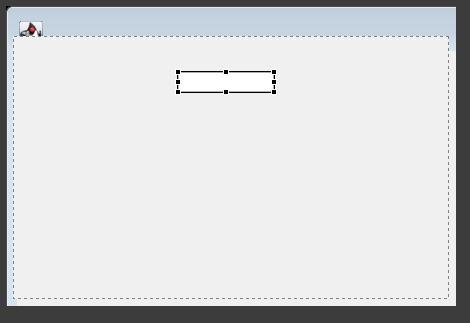
1. Klik *absolute layout* kemudian bawa atau tarik kursor dan lepaskan di ruang kosong bewarna putih di samping kolom menu fitur.



1. Gerek kembali fitur menu tadi. Pilih JTextField. Fungsi fitur ini adalah membuat kolom teks yang bisa diinputkan oleh pengguna.

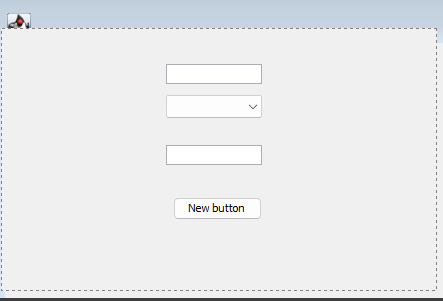


1. Klik menu tersebut, kemudian gerek atau tarik kursor ke bagian ruang kosong di samping kolom menu fitur. Lepaskan.

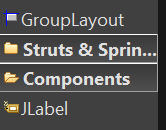
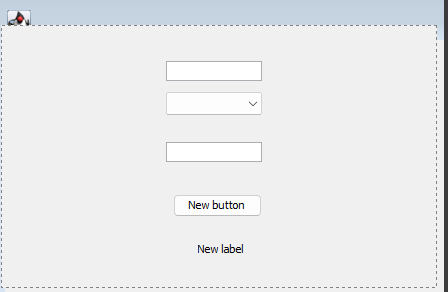


Tampilan ruang kosong akan seperti gambar di atas setelah menyeret kursor JTextField. Buat dua JTextField dengan posisi sejajar secara vertikal.

1. Kembali ke kolom menu fitur, cari fitur JComboBox dan JButon. Lakukan hal yang sama pada dua menu fitur tersebut, seret ke bagian ruang kosong dan lepaskan.
2. Letakkan JComboBox diantara dua JTexField. JButon di bawah JTextField yang ke dua. Pengguna bisa mendesain tampilan masing-masing kolom sesuai dengan selera dan kebutuhan.



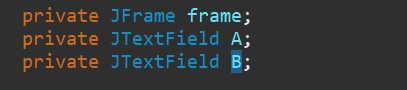
1. Terakhir untuk desainnya, tambahkan JLabel bada bagian bawah JButon. Atur sesuai dengan keinginan

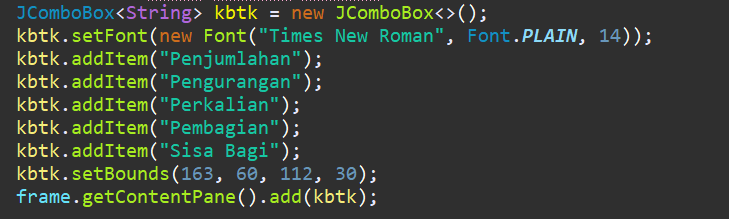
1. Setelah selesai mendesain tampilan, pindah ke bagian *source*.



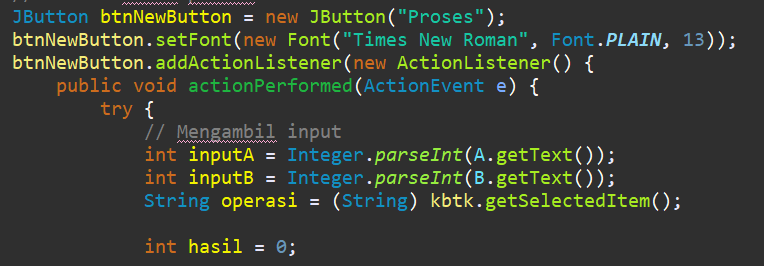
1. Selanjutnya pada bagian *source*, buatlah deklarasi variabel pada JTextField pertama dan kedua.



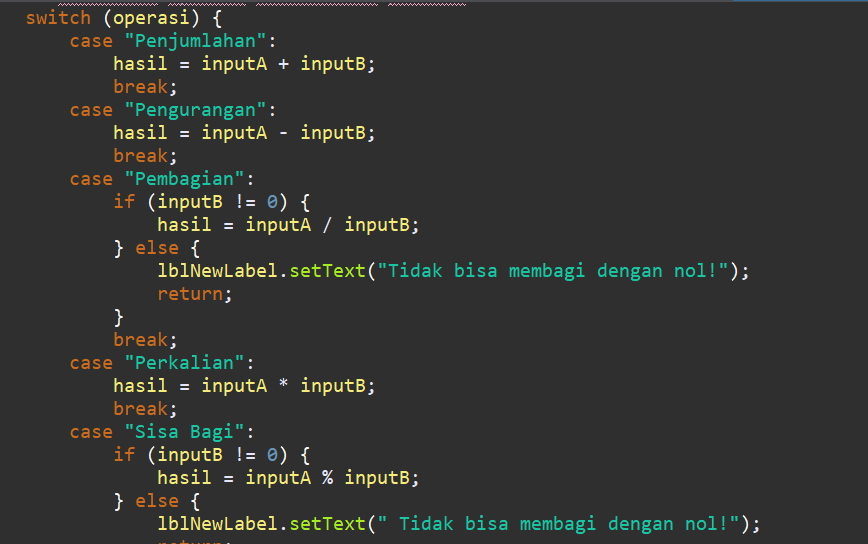
1. Setelah deklarasi variabel, buatlah perintah kode pada JComboBox untuk menyimpan beberapa item yang bertipe *string*.



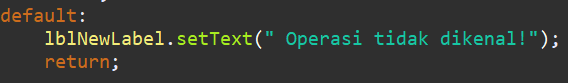
1. Buatlah perintah pada pogram bagian JButon untuk mengambil isi input variabel yang telah dimasukan oleh pengguna.



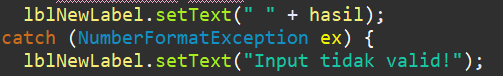
1. Berikutnya pada kode JButon buatlah pernyataan kondisional mengenai operasi aritmatika menggunakan switch-case berupa gambar di bawah ini.



1. Untuk mengakhiri logi pernyataan kondisionalnya terakhir buatlah default untuk pogram kode *switch-case*.



1. Untuk mengahasilkan pogram buatlah kode berikut ini.



1. Kesimpulan

Pratikum ini berfokus kepada konsep enkapsulasi dalam pogram berorientasi objek. Enkapsulasi dilakukan untuk membungkus data dan metode yang dibuat oleh pengguna agar tidak bocor ke kelas luar. Selain itu, pratikum ini memberikan langkah-langkah bagaimana cara membuat antarmuka pengguna grafis(GUI) dalam pembuatan kalkulator sederhana. Sehingga, pengguna mendapatkan interaksi dengan pogram secara visual.