

**PERANCANGAN APLIKASI TABEL DISTRIBUSI FREKUENSI
OBJECT ORIENTED PROGRAMMING**



INSTIKI
INSTITUT BISNIS DAN
TEKNOLOGI INDONESIA

Oleh:

Nama: Laila Nur Azizah
NIM: 2201010652

INSTITUT BISNIS DAN TEKNOLOGI INDONESIA
2024

1. Aplikasi Tabel Distribusi Frekuensi

Aplikasi Tabel Distribusi Frekuensi merupakan aplikasi yang dirancang untuk membantu pengguna dalam menyajikan data tunggal tabel distribusi frekuensi. Tabel distribusi frekuensi merupakan sebuah tabel yang menampilkan kelompok data dalam beberapa kategori dengan tujuan agar data menjadi informatif dan mudah dipahami. Aplikasi ini menggunakan form sebagai tampilannya.

Data tunggal yang dimasukkan oleh pengguna akan diolah agar data tersebut dapat disajikan dalam tabel distribusi frekuensi. Proses-proses tersebut, yaitu menentukan jumlah kelas, interval atau panjang, frekuensi, tepi atas, tepi bawah, dan nilai tengah kelas.

Selain menyajikan data dengan tabel distribusi frekuensi, aplikasi ini juga akan membantu pengguna dalam mencari ukuran pemusatan. Ukuran pemusatan merupakan nilai tunggal yang mewakili semua data, contohnya seperti rata-rata (mean), median, dan modus. Aplikasi ini juga membantu dalam mencari dispersi data. Dispersi menggambarkan penyebaran nilai pada data. Dikarenakan data yang dimasukkan oleh pengguna merupakan data tunggal, maka dispersi yang ditampilkan merupakan dispersi data tunggal. Begitu juga dengan ukuran pemusatan, sehingga menggunakan rumus-rumus pada umumnya, bukan rumusan untuk data berkelompok. Data yang dimasukkan akan diproses sehingga dapat menampilkan dispersi data. Proses-proses tersebut, yaitu menghitung simpangan rerata, simpangan baku, dan varians nya.

2. Method dan UML

Dalam aplikasi ini, proses dibagi menjadi 3, yaitu proses membuat tabel distribusi frekuensi, menghitung ukuran pemusatan data dan dispersi, dan mengubah data. Semua aktivitas tersebut berada pada satu *class* yang bernama “operasi”. Sedangkan untuk menyimpan data dan mengisi tabel diproses di form yang digunakan sebagai tampilan. Hal tersebut dilakukan untuk memisahkan proses-proses yang mengolah data dan aritmatika dari proses-proses yang memanipulasi tampilan form. Proses untuk menyimpan data dan mengisi tabel berisi proses seperti mengubah editable *text field* dan enabled tombol pada form. Data yang dimasukkan oleh pengguna akan disimpan pada sebuah array integer bernama data yang berada pada form.

Berikut merupakan UML dari *class* operasi.

| operasi |
|---|
| + double: mean(int [] data) + double: median(int [] data) + String: modus(int [] data) + double: kelas(int [] data) + double: interval(int [] data) + hapus (int sample, int [] data) + hapus (int sample, int [] data,int jmlh) + double: rerata(int [] data) + double: varians(int [] data) + double baku (int [] data,double varians) |

1. Method mean

Method ini menghitung mean atau rata-rata dari array data dengan tipe data integer. Semua method pada class proses memiliki parameter array data agar dapat

memproses data array yang dituju, seperti array data yang berada pada form. Memiliki tipe return berupa double dikarenakan hasil dari rata-rata dapat berupa bilangan desimal. Semua method pada class operasi, termasuk method mean, memiliki modifier public agar dapat diakses dari mana saja.

2. Method median

Method ini menghitung nilai tengah dari data yang telah diurutkan dari terkecil ke terbesar. Memiliki tipe return double karena nilai tengah dapat berupa desimal. Dalam prosesnya, data akan diurutkan menggunakan arrays sort, kemudian akan dicari data yang berada di tengah. Data yang berada di tengah dapat dicari dengan jumlah data dibagi 2. Apabila banyak datanya genap, maka data pada bagian tengah tersebut ditambah satu data sebelumnya dan dibagi dua. Menggunakan method yang memiliki nilai return agar dapat memberikan suatu hasil yang nantinya mengisi form hasil perhitungan.

3. Method modus

Method ini mencari nilai yang muncul terbanyak. Memiliki tipe return string, hal ini dikarenakan ada kemungkinan tidak adanya modus pada suatu data. Apabila masing-masing data berjumlah sama, maka hasil yang akan di-return adalah “tidak ada”. Namun, apabila terdapat modus yang akan di-return, modus harus diubah menjadi string terlebih dahulu.

4. Method kelas

Method ini menentukan jumlah kelas atau baris dari tabel distribusi frekuensi yang akan dihasilkan. Method ini sangat penting dikarenakan akan menjadi batas dalam mengisi tabel. Rumus untuk mencari jumlah kelas, yaitu $1 + 3,322(\log n)$ dimana n merupakan banyak data. Log yang dimaksud disini adalah $10\log n$. Namun, karena hasil dari method ini menentukan jumlah baris tabel, maka hasilnya harus berupa bilangan bulat. Oleh karena itu, hasil yang di-return harus dibulatkan.

5. Method interval

Method ini menentukan interval untuk masing-masing kelas. Untuk menghitung interval, diperlukan jangkauan data, yaitu data terbesar dikurangi data terkecil. Selanjutnya, jangkauan data dibagi dengan banyaknya kelas. Sama seperti method kelas, hasil dari method ini harus berupa bilangan bulat. Oleh karena itu, hasil pembagian perlu dibulatkan. Hasil interval kemudian digunakan untuk menentukan batas-batas pada setiap kelas. Contohnya seperti, kelas 16-20 yang memiliki interval 5. Kemudian dilanjutkan pada kelas selanjutnya, 21-25 dan seterusnya.

6. Method hapus

Terdapat dua method hapus, yaitu method hapus dengan parameter `int sample, int [] data, int jmlh`; dan dengan parameter `int sample, int [] data`. Kedua method sama-sama memiliki fungsi untuk mengganti data pada array. Method ini berupa void, karena method ini akan mempengaruhi data pada array dan frekuensi tabel nantinya.

a. hapus (`int sample, int [] data, int jmlh`)

Merupakan method yang digunakan untuk mengganti data (`int sample`) pada array, dengan jumlah frekuensi yang ditentukan. Contohnya, pada array terdapat angka 2 yang muncul sebanyak 5 kali (frekuensi). Method ini akan menanyakan pengguna, dari total 5 frekuensi, berapa yang ingin diubah. Pengguna dapat hanya mengubah 2 saja (`int jmlh`), sehingga sisa angka 2 yang muncul hanya 3, dan 2 data yang diubah (awalnya angka 2) tersebut akan masuk ke frekuensi angka yang dituju.

b. hapus (`int sample, int [] data`)

[illegible]

Tabel di sebelah kiri merupakan tabel distribusi frekuensi yang nantinya akan terisi dengan data yang telah diproses. Bagian kanan merupakan kumpulan field text yang tidak dapat diedit kecuali bagian memasukkan data ubah. Hal tersebut dikarenakan kumpulan field text tersebut digunakan untuk menampilkan hasil perhitungan.

Pada bagian memasukkan data ubah, pengguna dapat mengisi field text dengan data yang ingin diubah.

Ubah data yang di atas digunakan untuk mengubah data yang dimasukkan hanya sebanyak yang diperlukan. Ketika pengguna mengisi text field dengan data dan menekan tombol ganti, pengguna akan mengisi inputan untuk banyak data yang akan diubah dari data tersebut. Seluruh inputan di aplikasi hanya dapat dimasukkan satu data saja, tidak berupa deretan data sekaligus.

Jumlah Dari Frekuensi Data Tersebut Yang Diubah

OK Cancel

Selanjutnya, pengguna akan mengisi data baru untuk menggantikan data tersebut sebanyak yang ditentukan.

Masukkan Data Baru

OK Cancel

Sedangkan pada ubah data yang di bawah, pengguna langsung mengisi data baru. Pada bagian menambahkan data, pengguna akan mengisi banyak data yang ada.

Masukkan Banyak Data

OK Cancel

Setelah itu, pengguna akan memasukkan data satu persatu, hingga mencapai banyak data yang telah dimasukkan sebelumnya.

Masukkan Data Ke-: 1

OK Cancel