

LAPORAN PRAKTIKUM
PEMROGRAMAN TERSTRUKTUR
PRAKTIKUM I – PENDAHULUAN
KELAS B



Disusun Oleh:

Nama : Andro Syahreza
NIM : 175090807111007
Hari/Tgl. Praktikum : Selasa, 23 April 2019

LABORATORIUM KOMPUTASI
JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2019

A. Percobaan 1:

Buatlah program untuk sorting data, serta tentukan mean, median, dan modus pada data tersebut!

Solusi :

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, swap=0, arr[n];
6      int frek[n], max=-100, modus;
7      int med_1, med_2;
8      float sum, mean, median;
9
10     printf("masukkan jumlah data: "); //Input nilai n
11     scanf("%d", &n);
12
13     for(int i=0; i<n; i++){           //Input data
14         printf("data ke-%d: ", i+1);
15         scanf("%d", &arr[i]);
16     }
17
18     for(int i=0; i<n-1; i++){         //Sorting
19         for (int j=i+1; j<n; j++){
20             if (arr[j]<arr[i]){
21                 swap=arr[i];
22                 arr[i]=arr[j];
23                 arr[j]=swap;
24             }
25         }
26
27         for(int i=0; i<n; i++){       //Mean
28             sum+=arr[i];
29             mean=sum/n;
30         }
31
32         for(int i=0; i<n; i++){       //Median
33             if(n%2==1){
34                 med_2=(n/2);
35                 median=arr[med_2];
36             }
37             else if(n%2==0){
38                 med_1=((n/2)-1);
39                 med_2=(n/2);
40                 median=((float)arr[med_1]+(float)arr[med_2])/2;
41             }
42         }
43     }
```

```

41
42     for(int i=0; i<10; i++){           //Modus
43         frek[i]=0;
44         for (int j=0; j<n; j++){
45             if(arr[j]==i){
46                 frek[i]++;}
47             if(frek[i]>max){
48                 max=frek[i];
49                 modus=i;}
50         }
51
52     printf("Hasil Sorting \t= "); //Print Hasil Sorting
53     for(int i=0; i<n; i++){
54         printf("%d ", arr[i]);
55         printf("\n");
56
57     printf("Mean\t\t= %.2f\n", mean); //Print Hasil Mean
58     printf("Median\t\t= %.2f\n", median); //Print Hasil Median
59     printf("Modus\t\t= %d\n", modus); //Print Hasil Modus
60
61
62     return 0;
63 }

```

Gambar 1.1 (Kode Program Percobaan 1)

```

masukkan jumlah data: 10
data ke-1: 1
data ke-2: 3
data ke-3: 2
data ke-4: 5
data ke-5: 8
data ke-6: 1
data ke-7: 4
data ke-8: 6
data ke-9: 7
data ke-10: 9

```

Gambar 1.2 (Masukan 1 dari Program)

```

Hasil Sorting   = 1 1 2 3 4 5 6 7 8 9
Mean            = 4.60
Median          = 4.50
Modus           = 1

```

Gambar 1.3 (Keluaran 1 dari Program)

```
masukkan jumlah data: 11
data ke-1: 1
data ke-2: 1
data ke-3: 3
data ke-4: 2
data ke-5: 6
data ke-6: 5
data ke-7: 4
data ke-8: 9
data ke-9: 8
data ke-10: 1
data ke-11: 7
```

Gambar 1.4 (Masukan 2 dari Program)

```
Hasil Sorting = 1 1 1 2 3 4 5 6 7 8 9
Mean          = 4.27
Median        = 4.00
Modus         = 1
```

Gambar 1.5 (Keluaran 2 dari Program)

Penjelasan :

Pada program 1 (Gambar 1.1) masukan user menggunakan perintah `scanf` yang akan disimpan pada variable 'n' yang bertipe integer sebagai banyaknya jumlah data. Lalu dibuat looping dari 0 sampai batas n untuk memasukkan data input sebanyak n dan disimpan pada variable `arr[n]`. Digunakan bubble sort untuk mensorting data input yang acak, dalam sorting bubble sort dilakukan nested loop. Untuk mendapatkan mean (rata-rata) dari data input dilakukan perintah `sum+=arr[i]` dan membaginya dengan n. Nilai tengah atau median didapatkan dengan menggunakan logika if else, jika nilai n adalah ganjil ($n\%2==1$) maka n akan dibagi dengan 2, hasil baginya akan dijadikan nilai indeks yang akan dicetak. Jika nilai n adalah genap ($n\%2==0$) maka n akan dibagi dengan 2 dan dikurangi dengan 1 lalu akan dijumlahkan dengan nilai n yang dibagi dengan 2, hasil penjumlahan tersebut akan dijadikan nilai indeks yang akan dicetak. Untuk mencari modus dibuat array pembandingan untuk dapat menentukan frekuensi yaitu 'frek[n]', dibuat nested loop lalu jika `arr[j]` adalah sama dengan nilai i maka nilai dari frekuensi ditambah 1, jika nilai frekuensi lebih dari `max(-100)` maka `max` sama dengan `frek[i]`, lalu modus didapatkan dari nilai 'i'. Gambar 1.2 menunjukkan input dengan jumlah yang genap, sedangkan gambar 1.4 menunjukkan input dengan jumlah yang ganjil. Keluaran yang dihasilkan sesuai dengan kode program yang diinginkan pada percobaan 1.