

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Perancangan dan Analisis Algoritma
Kelas : 3IA13
Praktikum ke- : 1
Tanggal : 19 Oktober 2022
Materi : Metode Greedy
NPM : 51420249
Nama : Tiara Puspita
Ketua Asisten :
Jumlah Lembar : 4

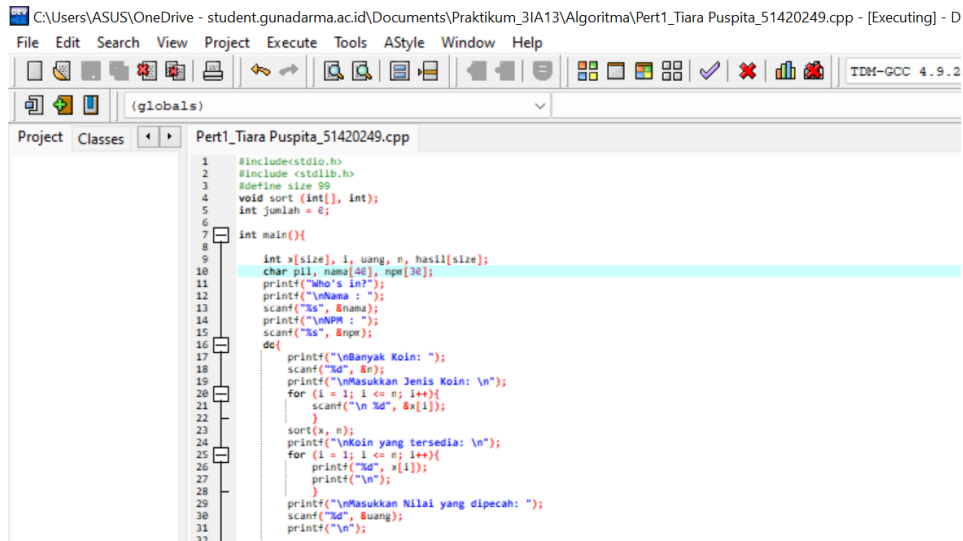


LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

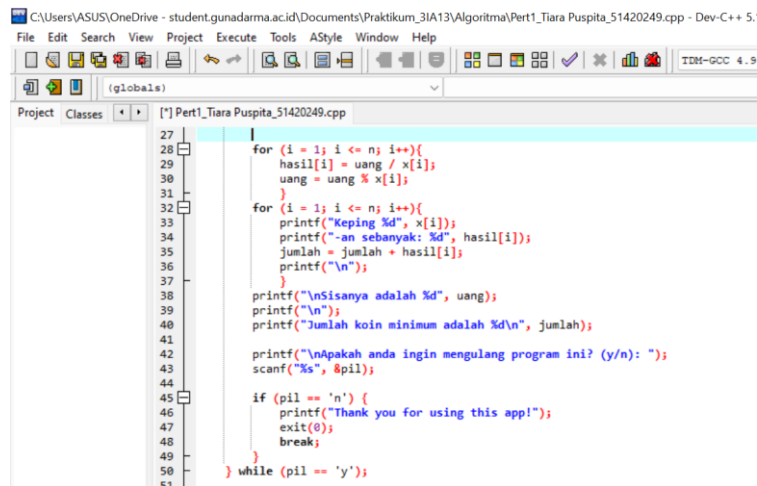
2022

Listing Program



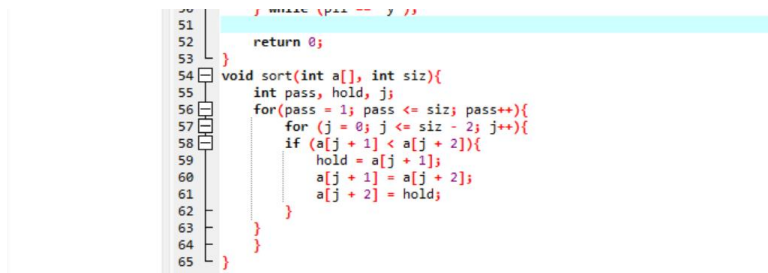
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define size 99
4 void sort (int[], int);
5 int jumlah = 0;
6
7 int main(){
8     int x[size], i, uang, n, hasil[size];
9     char pil, nama[40], npe[30];
10    printf("Who's in?");
11    printf("\nNama : ");
12    scanf("%s", &nama);
13    printf("\nNPM : ");
14    scanf("%s", &npe);
15    do{
16        printf("\nBanyak Koin: ");
17        scanf("%d", &n);
18        printf("\nMasukkan Jenis Koin: \n");
19        for (i = 1; i <= n; i++){
20            scanf("%d", &x[i]);
21        }
22        sort(x, n);
23        printf("\nKoin yang tersedia: \n");
24        for (i = 1; i <= n; i++){
25            printf("%d", x[i]);
26            printf("\n");
27        }
28        printf("\nMasukkan Nilai yang dipecah: ");
29        scanf("%d", &uang);
30        printf("\n");
31    } while (pil != 'y');
```

Gambar 1. Coding split



```
27
28
29     for (i = 1; i <= n; i++){
30         hasil[i] = uang / x[i];
31         uang = uang % x[i];
32     }
33     for (i = 1; i <= n; i++){
34         printf("Keping %d", x[i]);
35         printf("-an sebanyak: %d", hasil[i]);
36         jumlah = jumlah + hasil[i];
37         printf("\n");
38     }
39     printf("\nSisanya adalah %d", uang);
40     printf("\n");
41     printf("Jumlah koin minimum adalah %d\n", jumlah);
42
43     printf("\nApakah anda ingin mengulang program ini? (y/n): ");
44     scanf("%s", &pil);
45
46     if (pil == 'n') {
47         printf("Thank you for using this app!");
48         exit(0);
49         break;
50     } while (pil == 'y');
```

Gambar 2 Coding split



```
50
51
52     return 0;
53 }
54 void sort(int a[], int siz){
55     int pass, hold, j;
56     for(pass = 1; pass <= siz; pass++){
57         for (j = 0; j <= siz - 2; j++){
58             if (a[j + 1] < a[j + 2]){
59                 hold = a[j + 1];
60                 a[j + 1] = a[j + 2];
61                 a[j + 2] = hold;
62             }
63         }
64     }
65 }
```

Gambar 3 Coding split

```
C:\Users\ASUS\OneDrive - student.gunadarma.ac.id\Documents\Praktikum_3\A13\Algoritma\Pert1_Tiara Puspita_51420249.exe
Who's in?
Nama : Tiara_Puspita

NPM : 51420249

Banyak Koin: 2

Masukkan Jenis Koin:
20
90

Koin yang tersedia:
90
20

Masukkan Nilai yang dipecah: 300

Keping 90-an sebanyak: 3
Keping 20-an sebanyak: 1

Sisanya adalah 10
Jumlah koin minimum adalah 4

Apakah anda ingin mengulang program ini? (y/n): n
Thank you for using this app!
-----
Process exited after 28.02 seconds with return value 0
Press any key to continue . . .
```

Gambar 4 Output Program

Algoritma Program

Pada saat memulai program, user diminta untuk memasukkan jumlah koin yang tersedia atau yang user inginkan. Setelah itu, user diminta untuk menyebutkan nilai dari setiap koin yang dimasukan. Kunci dari program ini ada pada sorting nilai koin yang ada pada fungsi sort(). Setelah user menginput data yang diperlukan program, program akan mengurutkan nilai koin mulai dari nilai yang terbesar hingga nilai yang terkecil. Setelah diurutkan, user akan diminta untuk meng-input jumlah uang yang akan atau ingin dipecah. Program memecah koin dengan menggunakan baris kode berikut:

```
for (i = 1; i <= n; i++){
    hasil[i] = uang / x[i];
    uang = uang % x[i];
}
```

Jumlah setiap koin yang diperlukan setelah uang dipecah akan disimpan pada variable hasil[] dalam bentuk array. Jika terdapat sisa, atau koin tidak mencukupi nilai yang dipecah, jumlah sisa akan

disimpan pada variable uang. Pastikan untuk mengurutkan nilai koin secara descending dengan logika program seperti pada baris berikut:

```
void sort(int a[], int siz){
    int pass, hold, j;
    for(pass = 1; pass <= siz; pass++){
        for (j = 0; j <= siz-2; j++){
            if (a[j + 4] < a[j + 2]){
                hold = a[j + 1];
                a[j + 1] = a[j + 2];
                a[j + 2] = hold;
            }
        }
    }
}
```

Algoritma Greedy

Penggunaan algoritma greedy pada program ini terdapat pada descending sort dari setiap nilai koin yang diimplementasikan dengan fungsi sort(). Setelah koin diurutkan secara menurun atau dari nilai terbesar hingga nilai terkecil. Setelah koin diurutkan, program akan mencari jumlah dari setiap koin yang akan digunakan. Dalam hal ini, solusi optimal nya ada pada jumlah koin yang dipakai atau dikeluarkan. Semakin minim koin keping koin yang digunakan, maka itu merupakan solusi yang optimal.