

## **LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Perancangan Analisis Algoritma  
Kelas : 3IA24  
Praktikum ke- : 1  
Tanggal : 20 Oktober 2023  
Materi : Algoritma Greedy  
NPM : 50421704  
Nama : Juan Samuel Christopher  
Ketua Asisten : Sherina Permata  
Nama Asisten :  
Paraf Asisten :  
Jumlah Lembar : 7



**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2023**

## LISTING

LA1PAA.cpp

```
1  #include <stdio.h>
2  #define size 99
3
4  void sort(int[], int);
5  int jumlah = 0;
6
7  int main() {
8      int x[size], i, uang, n, hasil[size];
9
10     printf ("\nBanyak Koin : ");
11     scanf ("%d", &n);
12
13     printf("\nMasukkan Jenis Koin : ");
14     for (i = 1; i <= n; i++) {
15         scanf ("\n %d", &x[i]);
16     }
17
18     sort (x, n);
19     printf ("\nKoin yang tersedia : \n");
20     for (i = 1; i <= n; i++) {
21         printf ("%d", x[i]);
22         printf ("\n");
23     }
24
25     printf("\nMasukkan Nilai yang dipecah : ");
26     scanf ("%d", &uang);
27     printf("\n");
28     for (i = 1; i <= n; i++){
29         hasil[i] = uang / x[i];
30         uang = uang % x[i];
31     }
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
40     printf ("\nSisanya adalah %d", uang);
41     printf ("\n");
42     printf ("Jumlah koin minimum adalah %d\n", jumlah);
43     return 0;
44 }
45
46 void sort(int a[], int siz){
47     int pass, hold, j;
48     for (pass = 1; pass <= siz - 1; pass++){
49         for (j = 0; j <= siz - 2; j++) {
50             if (a[j + 1] < a[j+2]){
51                 hold = a[j + 1];
52                 a[j + 1] = a[j + 2];
53                 a[j + 2] = hold;
54             }
55         }
56     }
57 }
```

## LOGIKA

```
1  #include <stdio.h>
2  #define size 99
```

**#include <stdio.h>** Ini adalah library (Kumpulan instruksi yang digunakan dalam bahasa pemrograman C dan C++ sebelum kompilasi) yang digunakan untuk menyertakan file header "**stdio.h**" yang berisi fungsi-fungsi dasar untuk input dan output. Lalu **#define size 99** Ini adalah penggunaan preprocessor directive yang mendefinisikan konstanta **size** dengan nilai 99

```
4  void sort(int[], int);
5  int jumlah = 0;
```

Selanjutnya ada fungsi `void sort(int[], int);` adalah deklarasi fungsi **sort** yang akan menerima sebuah array **integer** dan sebuah **integer** sebagai argumen. `int jumlah = 0;` adalah deklarasi variabel **jumlah** dengan nilai awal 0.

```
7  int main() {
8      int x[size], i, uang, n, hasil[size];
9
10     printf ("\nBanyak Koin : ");
11     scanf ("%d", &n);
12
13     printf ("\nMasukkan Jenis Koin : ");
14     for (i = 1; i <= n; i++) {
15         scanf ("\n %d", &x[i]);
16     }
```

**int main() {** Ini adalah awal dari fungsi main. `int x[size], i, uang, n, hasil[size];` Ini adalah deklarasi variabel-variabel `x`, `i`, `uang`, `n`, dan `hasil` yang digunakan dalam program. Lalu perintah **printf ("\nBanyak Jenis Koin : ");** berfungsi untuk mencetak teks "Banyak Jenis Koin : " ke layar, kemudian **scanf ("%d", &n);** berfungsi untuk membaca masukan dari pengguna, yaitu jumlah jenis koin yang akan dimasukkan. Lalu Nilai ini disimpan dalam variabel **n**

**printf ("\nMasukkan Jenis Koin : ");** berfungsi untuk mencetak pesan "Masukkan Jenis Koin : " ke layar, yang akan meminta pengguna untuk memasukkan jenis-jenis koin. Lalu ada **for (i = 1; i <= n; i++) {** yang merupakan awal dari loop for yang akan digunakan untuk membaca jenis-jenis koin dari pengguna sebanyak **n** kali. Kemudian **scanf ("\n %d", &x[i]);** berfungsi untuk membaca jenis koin dari pengguna dan menyimpannya dalam array **x** pada indeks **i**.

```
18     sort (x, n);
19     printf ("\nKoin yang tersedia : \n");
20     for (i = 1; i <= n; i++) {
21         printf ("%d", x[i]);
22         printf ("\n");
23     }
```

**sort (x, n);** berfungsi untuk mengurutkan jenis koin yang ada dalam array **x**. Lalu **printf ("\nKoin yang tersedia : \n");** berfungsi untuk mencetak teks "**Koin yang tersedia :** " pada layar.

Lalu ada perulangan **for (i = 1; i <= n; i++) {** yang merupakan awal dari loop for yang digunakan untuk mencetak jenis-jenis koin yang sudah diurutkan. Kemudian **printf ("%d", x[i]);** berfungsi untuk mencetak jenis koin pada indeks **i**. Lalu **printf ("\n");** berfungsi untuk mencetak karakter baris baru.

```
25     printf("\nMasukkan Nilai yang dipecah : ");
26     scanf ("%d", &uang);
27     printf("\n");
28     for (i = 1; i <= n; i++){
29         hasil[i] = uang / x[i];
30         uang = uang % x[i];
31     }
```

Lanjut pada blok program diatas ada **printf("\nMasukkan Nilai yang dipecah : ");** yang berfungsi untuk mencetak pesan "**Masukkan Nilai yang dipecah :** " di layar, yang akan meminta pengguna untuk memasukkan nilai yang akan dipecah menjadi koin.

**scanf ("%d", &uang);** berfungsi untuk membaca input dari pengguna, yaitu nilai yang akan dipecah menjadi koin. Nilai ini akan disimpan dalam variabel **uang**.

**printf("\n");** berfungsi untuk mencetak karakter baris baru, sehingga hasil cetakan selanjutnya akan dimulai dari baris baru.

**for (i = 1; i <= n; i++) {** adalah loop for yang akan digunakan untuk menghitung berapa banyak koin dari setiap jenis yang diperlukan untuk memecah nilai uang.

**hasil[i] = uang / x[i];** berfungsi untuk menentukan berapa banyak koin dari jenis **x[i]** yang diperlukan untuk memecah nilai uang. Hasilnya disimpan dalam array **hasil**.

**uang = uang % x[i];** berfungsi untuk menentukan sisa nilai uang setelah sebagian nilai telah dipecah menjadi koin jenis **x[i]**. Sisanya akan digunakan dalam iterasi berikutnya.

```
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     }
```

**for (i = 1; i <= n; i++) {** digunakan untuk mencetak hasil dari pemecahan nilai uang menjadi koin dari setiap jenis yang tersedia. Loop ini akan berjalan sebanyak **n** kali, di mana **n** adalah jumlah jenis koin.

**printf("Keping %d", x[i]);** berfungsi untuk mencetak pesan "**Keping**" diikuti dengan jenis koin **x[i]**, yang akan dicetak.

**printf("-an sebanyak : %d", hasil[i]);** berfungsi untuk mencetak jumlah koin yang diperlukan untuk jenis koin **x[i]**. Jumlah ini diambil dari array hasil yang telah dihitung sebelumnya.

**jumlah = jumlah + hasil[i];** adalah operasi penjumlahan yang digunakan untuk mengakumulasi jumlah total koin yang digunakan dalam pemecahan nilai uang. Nilai ini disimpan dalam variabel **jumlah**.

**printf ("\n");** berfungsi untuk mencetak karakter baris baru, sehingga setiap jenis koin dan hasilnya dicetak dalam baris terpisah.

```
40     printf ("\nSisanya adalah %d", uang);
41     printf ("\n");
42     printf ("Jumlah koin minimum adalah %d\n", jumlah);
43     return 0;
44 }
```

Lanjut , disini ada **printf ("\nSisanya adalah %d", uang);** berfungsi untuk mencetak pesan "Sisanya adalah" diikuti dengan nilai uang yang belum terpecah menjadi koin. Lalu **printf ("\n");** untuk mencetak karakter baris baru.

**printf ("Jumlah koin minimum adalah %d\n", jumlah);** berfungsi untuk mencetak pesan yang menunjukkan jumlah total koin minimum yang digunakan dalam pemecahan nilai uang. Kemudian **return 0;** adalah pernyataan return yang mengakhiri fungsi main dan mengembalikan nilai 0, yang menunjukkan bahwa program selesai dijalankan tanpa masalah.

```
46 void sort(int a[], int siz){
47     int pass, hold, j;
48     for (pass = 1; pass <= siz - 1; pass++){
49         for (j = 0; j <= siz - 2; j++) {
50             if (a[j + 1] < a[j + 2]){
51                 hold = a[j + 1];
52                 a[j + 1] = a[j + 2];
53                 a[j + 2] = hold;
54             }
55         }
56     }
57 }
```

Pada blok program terakhir ada **void sort(int a[], int siz) {** Ini adalah awal dari definisi fungsi sort, yang digunakan untuk mengurutkan jenis-jenis koin dalam array a dengan ukuran siz. Lalu **int pass, hold, j;** adalah deklarasi variabel **pass**, **hold**, dan **j** yang digunakan dalam proses pengurutan.

Kemudian **for (pass = 1; pass <= siz - 1; pass++) {** adalah loop for luar yang digunakan untuk mengulangi proses pengurutan sebanyak **siz - 1** kali. Lalu **for (j = 0; j <= siz - 2; j++) {** adalah loop for dalam yang digunakan untuk membandingkan elemen-elemen array dan menukar mereka jika diperlukan. Dan **if (a[j + 1] < a[j + 2]) {** adalah pernyataan if yang memeriksa apakah elemen di indeks j + 1 lebih kecil dari elemen di indeks j + 2. Jika benar,

maka elemen tersebut akan ditukar. Kemudian **hold = a[j + 1];** adalah pernyataan yang menyimpan nilai elemen di indeks **j + 1** dalam variabel **hold**. Lanjut **a[j + 1] = a[j + 2];** adalah pernyataan yang menukar nilai elemen di indeks **j + 1** dengan nilai elemen di indeks **j + 2**. Dan terakhir ada **a[j + 2] = hold;** adalah pernyataan yang mengembalikan nilai yang disimpan di **hold** ke elemen di indeks **j + 2**, sehingga terjadi pertukaran.

## OUTPUT

```
C:\Users\juan samuel\Docum... X + v
Banyak Jenis Koin : 4
Masukkan Jenis Koin : 100
200
500
1000
Koin yang tersedia :
1000
500
200
100
Masukkan Nilai yang dipecah : 6900
Keping 1000-an sebanyak : 6
Keping 500-an sebanyak : 1
Keping 200-an sebanyak : 2
Sisanya adalah 0
Jumlah koin minimum adalah 9
-----
Process exited after 37.69 seconds with return value 0
Press any key to continue . . . |
```