**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

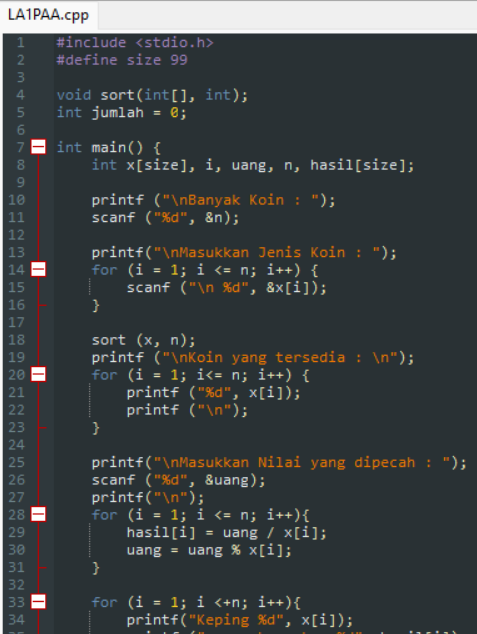


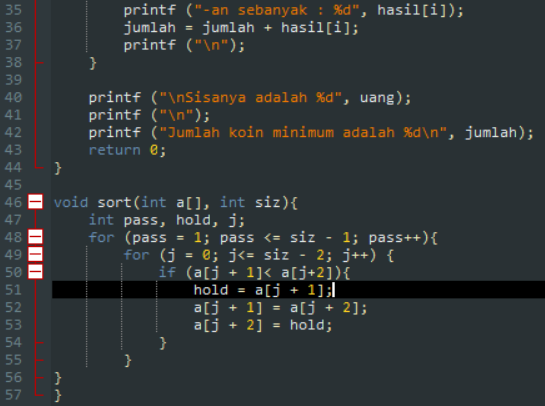
**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2023**

**LISTING**





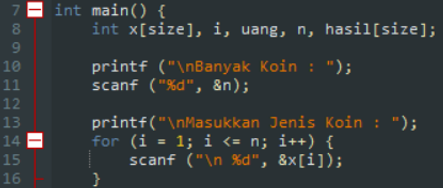
**LOGIKA**

****

**#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

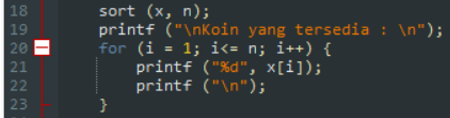
****

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.

****

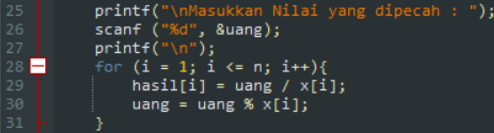
**int main() {** Ini adalah awal dari fungsi main. int x[size], i, uang, n, hasil[size]; Ini adalah deklarasi variabel-variabel x, i, uang, n, dan hasil yang digunakan dalam program. Lalu perintah **printf ("\nBanyak Jenis Koin : ");** berfungsi untuk mencetak teks "Banyak Jenis Koin : " ke layer, kemudian **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**.



**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 layer.

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.



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.

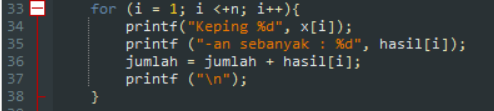
**scanf ("%d", &uang);** berfungsi untuk membaca input dari pengguna, yaitu nilai yang akan dipecah menjadi koin. Nilai ini akan disimpan dalam variabel **uang**.

**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.



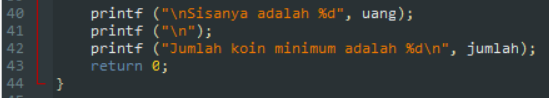
**for (i = 1; i <= n; i++) {** digunakan untuk mencetak hasil dari pemecahan nilai uang menjadi koin dari setiap jenis yang tersedia. Loop ini akan berjalan sebanyak **n** kali, di mana **n** adalah jumlah jenis koin.

**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.

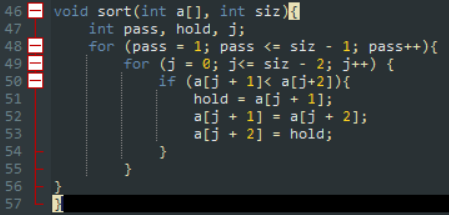
**jumlah = jumlah + hasil[i];** adalah operasi penjumlahan yang digunakan untuk mengakumulasi jumlah total koin yang digunakan dalam pemecahan nilai uang. Nilai ini disimpan dalam variabel **jumlah.**

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



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.

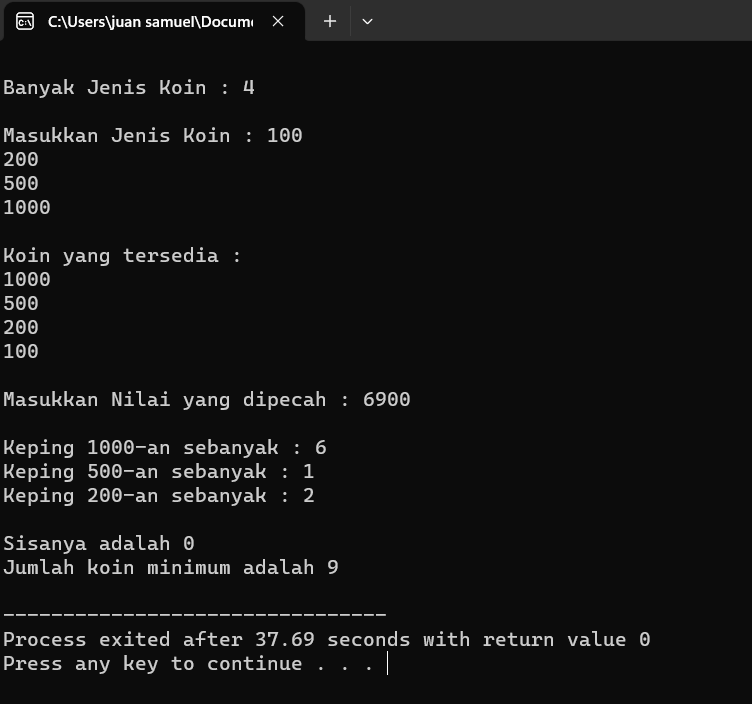
**printf ("Jumlah koin minimum adalah %d\n", jumlah);** berfungsi untuk mencetak pesan yang menunjukkan jumlah total koin minimum yang digunakan dalam pemecahan nilai uang. Kemudian **return 0;** adalah pernyataan return yang mengakhiri fungsi main dan mengembalikan nilai 0, yang menunjukkan bahwa program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



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**

****