Nama : Li Wei

NIM : 825220054

Mata Kuliah : Algoritma & Programming (Teori)

Remedial UTS (Saya NPM Genap, maka nomor yang dikerjakan adalah nomor ganjil)

Integer A = 9, B = 4, C = 2, Real I = 2.5, J = 6.5, Logika P = FALSE dan Q = TRUE

1a. R = SQRT(A) + B \* I –TRUNC( J+B ) / MOD((A\*C),B) + ROUND (I \* A) =

R = SQRT(9) + 4 \* 2.5 – TRUNC(6.5+4) / MOD ((9\*2), 4) + ROUND (2.5\*9) R = 3 + 10 – 10 / 2 + 23

R = 13 – 5 + 23

R = 31

1c. X = (A+B) < TRUNC(C+I) AND J > MOD(A,C) OR NOT (P AND Q) AND A DIV C != B

X = (9+4) < TRUNC(2+2.5) AND 6.5 > MOD(9,2) OR NOT (P AND Q) AND 9 DIV 2 != 4

X = 13 < 4 AND 6.5 > 1 OR NOT (P AND Q) AND 4 != 4 X = FALSE AND TRUE OR NOT FALSE AND FALSE X = TRUE AND FALSE

X= FALSE

3. Buatlah algoritma untuk menginputkan nama mahasiswa dan nilai UTS. Jumlah mahasiswa tidak diketahui, tetapi program akan berhenti jika nama mahasiswa diinputkan adalah "STOP". Output yang dihasilkan adalah mencetak nilai tertinggi, jumlah mahasiswa yang mendapatkan nilai tertinggi, nilai terendah dan jumlah mahasiswa yangmendapatkan nilai terendah. Berikut adalah contoh ilustrasi dari soal ini.

Jawab:

Psuedocode:

Deklarasi Void :

Void untuk mencari nilai tertinggi dan menghitung jumlah siswa yang mendapatkan nilai

tertinggi. Void memiliki parameter x[] bertipe double& untuk memasukkan nilai array dan

variable maks bertipe double& untuk nilai tertinggi, y bertipe integer untuk Batasan looping,

dan z bertipe int& untuk menghitung banyak siswa yang mendapatkan nilai tertinggi

Void max (double x[], int y, double& maks, int& z){

1. [Deklarasi nilai maks]

maks = 0

2. [Melakukan looping]

for (int i = 0; i < y; i++){

if (maks <= x[i]){

maks = x[i];

z++;

}

}

}

3. [Menunjukkan hasil]

Write (“Nilai tertinggi = “, maks)

Write (“Jumlah mahasiswa dengan nilai tertinggi = “, k)

Void untuk mencari nilai terendah dan menghitung jumlah siswa yang mendapatkan nilai

terendah. Void memiliki parameter x[] bertipe double& untuk memasukkan nilai array dan

variable min bertipe double& untuk nilai terendah, y bertipe integer untuk Batasan looping,

dan z bertipe int& untuk menghitung banyak siswa yang mendapatkan nilai terendah.

Void mini (double x[], int y, double& min, int& z){

1. [Deklarasi nilai maks]

min = x[0]

2. [Melakukan looping]

for (int i = 0; i < y; i++){

if (min >= x[i]){

min = x[i];

z++;

}

}

}

3. [Menunjukkan hasil]

Write (“Nilai terendah = “, min)

Write (“Jumlah mahasiswa dengan nilai tertinggi = “, k)

Algoritma MENENTUKAN\_NILAI\_PTS\_SISWA\_TERTINGGI\_DAN\_TERENDAH

Algoritma untuk menentukan nilai siswa dengan nilai tertinggi dan terendah. Algoritma

memiliki variable nilai[] bertipe double sebagai array memasukkan nilai, variable maks dan

min bertipe double untuk nilai maksimum dan minimum, variable nama bertipe string,

variable k untuk menentukan jumlah siswa dengan nilai tertinggi atau terendah, dan variable I

bertipe integer.

1. [Algoritma Utama]

[Mendeklarasikan Variabel]

Double nilai[]

Double maks

Double min

String nama

Integer k

Interger l

[Memasukkan input dengan rumus do while]

do {

int i = 0;

Write ("Nama Mahasiswa = ")

Read(nama)

if(nama != "STOP"){

Write( "Nilai UTS : ")

Read(nilai[i])

i++;

}

else{

continue;

}

}

while(nama != "STOP");

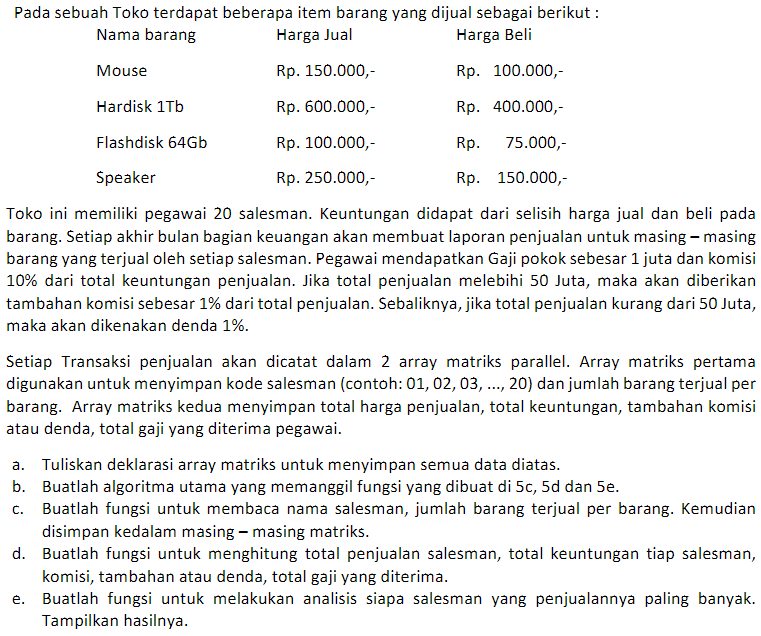
2. [Memanggil void max]

max (double x[], int y, double& maks, int& z)

3. [Memanggil void mini]

mini (double x[], int y, double& min, int& z)

4. Halt.

5. 

Jawab:

a. Deklarasi array matriks untuk menyimpan semua data diatas

1. string nama[20]; untuk nama
2. int jumlah[20]; untuk jumlah tiap barang
3. int m[i], h[i], f[i], s[i]; untuk mouse, hardisk, flashdisk, speaker
4. int total[i]; untuk total penjualan
5. int untung[i]; untuk keuntungan
6. int beli[i]; untuk harga beli
7. int komisi[i]; untuk total komisi
8. int komisi2[i]; untuk bonus komisi
9. int denda[i]; untuk denda
10. int gaji[i]; untuk total gaji

b. Algoritma JUMLAH\_KEUNTUNGAN

Algoritma untuk menghitung jumlah keuntungan penjualan dan menampilkan siapa yang paling

besar penjualannya. Variabel nama[20] adalah array bertipe string, jumlah[20] adalah array

bertipe integer untuk jumlah barang, variable maks bertipe double untuk maksimum

penjualan.

1. [membuat looping untuk pengulangan dan void]

for (int i = 1; i <= 20; i++){

[Memanggil void penjual()]

void penjual(nama, jumlah, i);

[Memanggil void total()]

void total(nama, jumlah, i);

[Memanggil void paling()]

void paling(nama, jumlah, i, maks);

2. Halt

}

c. VOID penjual(string a[],int b[], int i)

Untuk memasukkan nama penjual dan jumlah barang. Parameter terdiri atas string a[],int b[],

int i.{

1. [Mendeklarasikan variable]

int m[i], h[i], f[i], s[i];

Write ("Masukkan nama penjual = ")

Read ( a[i])

Write ( "Masukkan jumlah mouse yang terjual = ";

cin ( m[i])

Write ( "Masukkan jumlah Hardisk yang terjual = ";

cin ( h[i])

Write ( "Masukkan jumlah Flashdisk yang terjual = ";

cin ( f[i])

Write ( "Masukkan jumlah Speaker yang terjual = ";

cin ( s[i])

d. VOID total(string a[], int b[], int i)

Untuk menghitung total, untung, komisi, komisi/denda, dan gaji, Parameter terdiri atas string

a[],int b[], int i.{

1.[Mendeklarasikan variable]

int total[i];

int untung[i];

int beli[i];

int komisi[i];

int komisi2[i];

int denda[i];

int gaji[i];

int m[i], h[i], f[i], s[i];

2. [Memasukkan rumus]

[rumus menghitung total]

total[i]=m[i] \* 150000 + h[i]\*600000 + f[i]\*100000 + s[i]\*250000;

[rumus menghitung untung]

beli [i]=m[i] \* 100000 + h[i]\*400000 + f[i]\*75000 + s[i]\*150000;

untung[i]=total[i] - beli[i];

[rumus menghitung komisi]

komisi[i]= untung[i]\* 0.1;

[rumus menghitung tambahan/denda]

if (total[i]>50000000){

komisi2[i] = total[i] \* 0.01;

}

else{

denda[i] = total[i] \* 0.01;

}

[menghitung total gaji]

gaji[i] = total[i] + komisi[i] + komisi2[i] - denda[i];

}

e. VOID paling(string a[], int b[], int& i, double& maks)

Untuk menghitung penjualan tertinggi. Parameter terdiri atas string a[],int b[], int i, double&

maks.

{

1. [menggunakan looping dan memassukkan rumus maksimum untuk menentukan siapa

yang paling tinggi]

maks = 0;

int = l;

for (int i = 1; i <= 20; i++){

if (maks <= b[i]){

maks = b[i];

i = k;

}

}

2. [Menampilkan hasil]

Write( "Penjualan paling banyak oleh " << a[k] << endl)

Write( "Sebesar = Rp " << b[k] << endl