Nama : Muhammad Fikri Ramadhana

NIM : 21091397033

Kelas : 2021 A

**C. TUGAS PENDAHULUAN**

**1)** Rekursif adalah suatu proses ataupun prosedur dari fungsi yang memanggil dirinya secara berulang-ulang dengan kondisi yang membatasi perulangan yang sedang terjadi dalam proses tersebut. Fungsi rekursif juga memakan memori lebih besar dari yang seharusnya karena setiap bagian dari fungsi rekursif dipanggil, maka akan membutuhkan atau memakan ruang penyimpanan memori pada device.

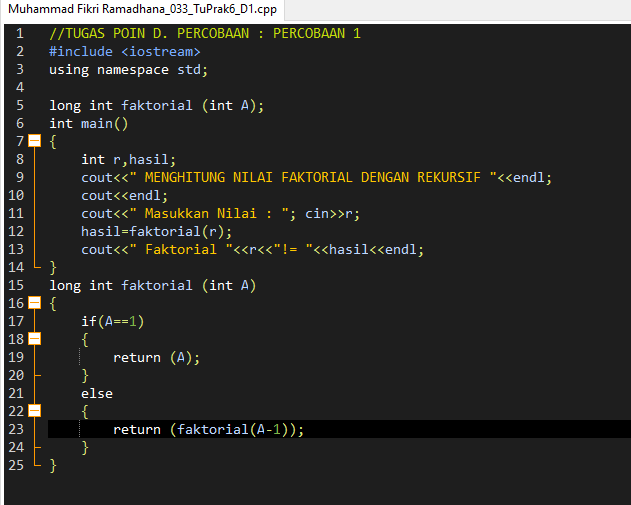
**2)** Dalam penerapa rekursif fungsi dari metode faktorial dapat dijabarkan sebagai berikut :

Melalui penulisan atau pemahaman pemrogaman dapat ditulis sebagai berikut

**Faktorial (0) = 1 (1)**

**Faktorial (1) = N\*Faktorial (N – 1) (2)**

Menghitung rekursif fungsi faktorial :



**3)** Bilangan fibbonaci didefinisikan sebagai berikut :

1 1 2 3 5 8 13 21 34 55 89 …

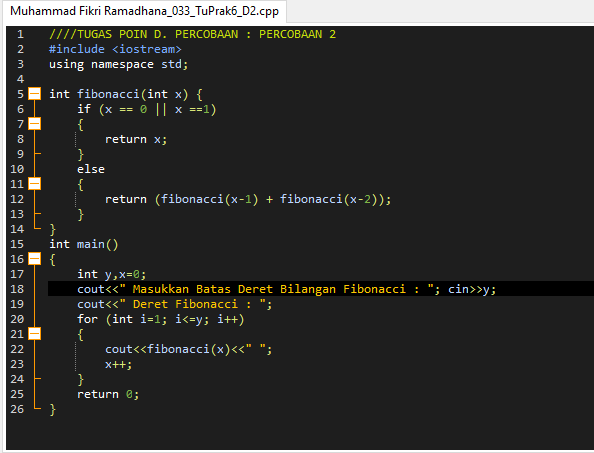
Sehingga didapat dari barisan tersebut jika bilangan ke- N (N>2) pada barisan dapat dicari dari dua bilangan yang tersedia sebelumnya dekat dengan posisi bilangan N, yaitu bilangan (N-1) dan bilangan (N-2), dengan rumus yang dapat dijabarkan sebagai berikut :

**Fibbonaci (1) = 1 (1)**

**Fibbonaci (2) = 1 (2)**

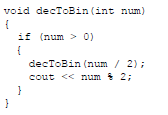
**Fibbonaci (N) = Fibbonaci (N-1) + Fibbonaci (N-2) (1)**

Berikut fungsi dalam menampilkan nilai fibbonaci dari deret fibbonaci :



**E. LATIHAN (1-4 Coding)**

**5) Progam rekursif**

****

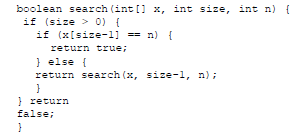
**void decToBin (int num)**

Proses awal mengubah desimal menjadi binary pada variabel tipe integer num.

**if (num > 0)**

Proses perintah dengan kondisi jika (if) variabel num bernilai benar apabila lebih dari 0, dengan instruksi decToBin bernilai variabel num dibagi dua. Kemudian cout variabel num dimodul 2.

**6)** **Progam rekursif**



**bool search (int x[ ], int size, int n)**

Proses tipe data boolean (bool) variabel search dengan kondisi true or false bernilai integer variabel x[ ], size dan n.

**if (size > 0)**

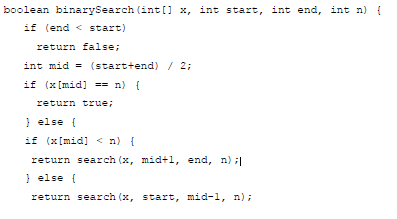
**if (x [size – 1] == n)**

Proses perintah kondisi true berupa variabel x dengan size – 1 sama dengan n didalam kondisi size bernilai benar apabila lebih dari 0.

**else ( )**

Apabila kondisi pertama tidak terpenuhi maka return search bernilai (x, size – 1, n).

**7) Progam rekursif**





**bool binarysearch (int x[ ], int start, int end, int n)**

Proses tipe data boolean binary variabel search dengan kondisi true or false bernilai interger variabel x[ ], start, end, dan n.

**if (end < start)**

Proses kondisi jika (if) variabel end kurang dari start maka statementnya bernilai salah atau return false.

**int mid = (start+end) / 2**

Proses tipe data integer variabel mid sama dengan start ditambah end hasilnya dibagi 2.

**if (x [mid] == n)**

Proses kondisi jika variabel x bernilai operasi mid sama dengan n maka bernilai benar atau true.

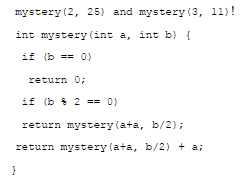
**if (x [mid] < n)**

Proses kondisi jika variabel x bernilai operasi mid kurang dari n maka bernilai return search (x, mid + 1, end, n).

**else ( )**

Apabila kondisi pertama dan kedua tidak terpenuhi maka bernilai return search (x, start, mid – 1, n).

**8) Progam rekursif**



**Int mystery (int a, int b)**

Proses pendeklarasian fungsi mystery dengan parameter formal fungsi tipe data int variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

**if (b == 0)**

**return 0;**

if (b == 0) return 0; merupakan kondisi jika nilai variabel b sama dengan 0 bernilai benar maka nilai balik (return value) adalah 0.

**if (b % 2 == 0)**

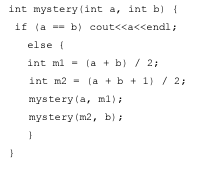
**return mystery(a+a, b/2);**

**return mystery(a+a, b/2);+a**

**}**

Deklarasi if (b % 2 == 0)return mystery(a+a, b/2); merupakan kondisi jika nilai variabel b dimodulo 2 adalah 0 atau bernilai true maka nilai balik (return value) akan dioperasikan ke dalam return mystery. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi int mystery.

**9) Progam rekursif**



**int mystery(int a, int b) {**

Deklarasi integrer mystery (int a, int b) { merupakan fungsi mystery dengan menggunakan parameter formal tipe data int variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

**if (a == b) cout<<a<<endl;**

**else { int m1 = (a+b)/2;**

**int m2 = (a+b+1)/2;**

**mystery(a, m1);**

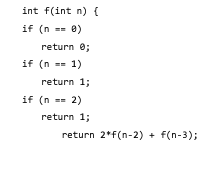
**mystery(m2, b);**

**}**

**}**

Proses kondisi jika (if) variabel a sama dengan nilai variabel b maka bernilai benar. Kemudia jika kondisi satu tidak terpenuhi maka int m1 = (a+b)/2, int m2 = (a+b+1)/2, mystery(a, m1), dan mystery(m2, b).

**10) Progam rekursif**

****

**int f(int n) {**

Deklarasi fungsi f dengan parameter tipe integer variabel n.

**if (n == 0)**

**return 0;**

Proses kondisi jika (if) variabel n sama dengan 0 maka bernilai return 0.

**if (n == 1)**

**return 1;**

Proses kondisi jika (if) variabel n sama dengan 1 maka bernilai return 1.

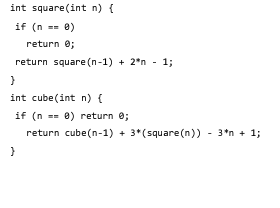
**if (n == 2)**

**return 1;**

**return 2\*f(n-2) + f(n-3);**

Proses kondisi jika (if) variabel n sama dengan 2 maka bernilai return 1 dan return 2\*f(n-2) + f(n-3).

**11) Progam rekursif**



**int square(int n) {**

Pendeklarasikan fungsi square dengan parameter tipe data int variabel n.

**if (n == 0)**

**return 0;**

**return square(n-1) + 2\*n - 1; }**

Proses kondisi jika (if) variabel nilai n sama dengan 0 maka bernilai return 0 dan return square(n-1) + 2\*n – 1.

**int cube(int n) {**

Pendeklarasian fungsi cube dengan parameter fungsi tipe data int variabel n.

**if (n == 0)**

**return 0;**

**return cube(n-1) + 3\*(square(n)) - 3\*n + 1;**

**}**

Proses kondisi jika (if) variabel n sama dengan 0 maka bernilai return 0 dan return cube(n-1) + 3\*(square(n)) – 3\*n + 1 dan diakhri dengan tanda } sebagai penutup.

**F. LAPORAN RESMI**

**a) analisis poin (D)**

**warna kuning ini merupakan hasil analisis dari coding**

**Percobaan 1**

#include <iostream>

using namespace std;

long int faktorial (int A); //tipe data long integer faktorial integer variabel A

int main(){

int r,hasil; //integer variabel r dan hasil

cout<<"MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF"<<endl;

cout<<endl;

cout<<"Masukan Nilai = ";

cin>>r;

hasil=faktorial(r); //variabel hasil sama dengan faktorial (r)

cout<<"Faktorial "<<r<<"!= "<<hasil<<endl;

}

long int faktorial (int A){ //tipe data long integer faktorial variabel A

if (A==1) //proses kondisi variabel A sama dengan 1 maka bernilai return (A)

return(A);

else //apabila kondisi satu tidak terpenuhi maka bernilai return

return

**Percobaan 2**

#include <iostream>

using namespace std;

int fibonacci(int m) { //tipe data integer fibbonaci dengan variabel m

if (m == 0 || m ==1) //proses kondisi m sama dengan 0 AND m sama dengan 1 maka return m

{

return m;

}

else //apabila kondisi satu tidak terpenuhi maka bernilai return (fibonacci(m-1) + fibonacci (m-2))

{

return (fibonacci(m-1) + fibonacci(m-2));

}

}

int main() {

int n, m= 0; //integer n dan m = 0

cout << "Masukan Batas Deret Bilangan Fibonacci : ";

cin >> n;

cout << "Deret Fibonacci: ";

for (int i = 1; i <= n; i++){ \\proses perulangan int i dengan nilai awal 1, int i kurang dari sama dengan n, dan i++ merupakan penambahan sampai nilai int i terpenuhi sampai batas yang ditentukan dengan instruksi fibonacci (m) penambahan m++.

cout << fibonacci(m) <<" ";

m++;

}

return 0;

}

**Percobaan 3**

#include <iostream>

using namespace std;

int ambil(int bil, int i){ //integer variabel ambil bernilai variabel int bil dan int i

if (i == 1) { //proses kondisi variabel i sama dengan 1 maka bernilai return 1

return 1;

}

else if (bil % i == 0) { //apabila kondisi variabel bil modul dengan i sama dengan 0 maka bernilai return 1 ditambah variabel ambil(bil, --i)

return 1 + ambil(bil, --i);

} else { //apabila kondisi satu dan dua tidak terpenuhi maka bernilai return 0 ditambah ambil(bil, --i)

return 0 + ambil(bil, --i);

}

}

int cek(int bil){ //integer variabel cek bernilai integer bil

if (bil > 1) { //proses kondisi variabel bil lebih dari 1 maka bernilai return (ambil(bil, bil) == 2)

return (ambil(bil, bil) == 2);

} else //apabila kondisi satu tidak terpenuhi maka bernilai return salah atau false

return false;

}

int main(){

int bil; //integer variabel bil

cout<<"Masukan Bilangan : ";

cin>>bil;

if (cek(bil)){ //proses kondisi variabel cek bernilai bil maka bernilai "Bilangan Prima" pada tampilan

cout<<"Bilangan Prima"<<endl;

}else { //apabila kondisi satu tidak terpenuhi maka bernilai "Bukan Bilangan Prima" pada tampilan

cout<<"Bukan Bilangan Prima"<<endl;

}

return 0;

}

**Percobaan 4**

#include <iostream>

using namespace std;

long int pangkatrekursif(int x, int y); //tipe data long integer pangkatrekursif bernilai integer x dan y

int main(){

int x,y; //integer variabel x dan y

cout<<"FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT"<<endl;

cout<<endl;

cout<<"Masukan Nilai X = ";

cin>>x; cout<<"Masukan Nilai Y = ";

cin>>y;

cout<<endl;

cout<<x<<" Dipangkatkan "<<y<<" = "<<pangkatrekursif(x,y)<<endl;

}

long int pangkatrekursif(int x, int y){ //tipe data long integer pangkatrekursif bernilai integer x dan y

if (y==0) //proses kondisi variabel y sama dengan 0 maka bernilai return 1

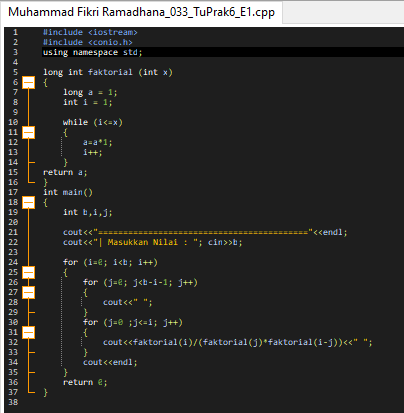
return 1 ;

else //apabila kondisi satu tidak terpenuhi maka bernilai return x\*pangkatrekursif(x,y-1)

return x \* pangkatrekursif(x,y-1); }

**b) analisis poin (E)**

**1)**

****

**long int faktorial (int x)**

Tipe data long integer variabel faktorial bernilai int x dengan tanda { dan diakhiri }.

**long a = 1;**

Tipe data long variabel a = 1 berada dalam fungsi faktorial dengan ditutupi tanda ; .

**int i = 1;**

Tipe data integer variabel i = 1 berada dalam fungsi faktorial dengan ditutupi tanda ; .

**while (i<=x)**

Kondisi while dengan variabel i kurang dari sama dengan x, saat nilai variabel imencapai x maka perulangan akan berhenti.

**a = a \* 1;**

Tipe data variabel a sama dengan a operasi perkalian 1 dengan ditutupi tanda ; .

**i++**

Tipe data variabel i akan mengalami pertambahan terus- menerus namun jika tidak diberi batasan maka akan berlanjut seterusnya, dengan ditutupi tanda ; .

**int main ()**

Tipe data integer variabel main, diawali dengan tanda { dan diakhiri dengan tanda }.

**int b, i, j;**

Tipe data integer variabel b, i, j dengan ditutupi tanda ; .

**for (i=0; i<b; i++) {**

**for (j=0; j<b-i-1; j++) {**

**cout<<” “;**

**}**

**for (j=0 ;j<=i; j++) {**

**cout<<faktorial(i)/(faktorial(j)\*faktorial(i-j))<<" ";**

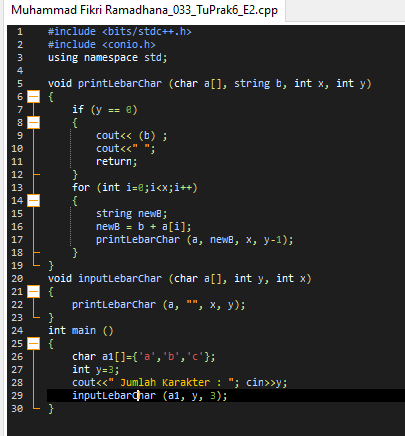
**}**

**cout<<endl;**

**}**

Perulangan fungsi **for (i=0; i<b; i++)**  dengan diawali tanda { dengan nilai perulangan dari i = 0, variabel i kurang dari b sebagai batasan berhenti, dan i++ sebagai perulangan penjumlahan menaik atau ditambah 1. Dengan didalam fungsi tersebut terdapat **for (j=0; j<b-i<1; j++)** sebagai perulangan kedua dengan nilai awal j = 0, variabel j kurang dari b dikurangi variabel i dikurangi 1, dan j++ sebagai perulangan naik yang harus ada batasan perulangannya. Kemudian perulangan ketiga **for (j=0 ;j<=i; j++)** dengan nilai awal j = 0, variabel j kurang dari sama dengan i, dan j++ sebagai perulangan naik yang harus memiliki pembatas. Dalam perulangan ketiga terdapat **cout<<faktorial(i)/(faktorial(j)\*faktorial(i-j))<<" ";** yang berfungsi untuk menampilkan output hasil nilai dari operasi yang dilakukan.

**2)**

****

**#include <bits/stdc++.h>**

File header bits/stdc++.h berfungsi agar progam tidak memasukkan tipe header library lainnya.

**void printLebarChar (char set[ ], string b,int x, int y)**

void printLebarChar (char a[ ], string b,int x, int y) dengan tanda awal { pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe data char variabel a[ ],tipe data string variabel b, tipe data int variabel x dan y.

**if (y == 0)**

**cout << (b) ;**

**cout << " ";**

**return;**

Proses kondisi variabel y sama dengan 0 maka bernilai cout << (b) ; cout << " "; return;.

**for (int i = 0; i < x; i++)**

**string newB;**

**newB = b + a[i];**

**printLebarChar (a, newB, x, y - 1);**

Proses perulangan yang dimulai dari i = 0, perulangan akan berhenti pada saat nilai i < x, dan i++ sebagai perulangan naik. Isi dari perulangan for terdapat variabel string newB, variabel newB = b + a[i], dan printLebarChar yang bernilai (a, newB, x, y-1).

**void inputLebarChar (char a[ ], int y,int x) {**

**printLebarChar (a, "", x , y);**

**}**

Deklarasi fungsi void dengan parameter fungsi char variabel a, integer y, dan x. Didalan fungsi tersebut terdapat deklarasi lebar tipe data char yang ada di dalam fungsi for.

**int main() {**

**char a1[] = {'a', 'b', 'c'};**

**int y = 3;**

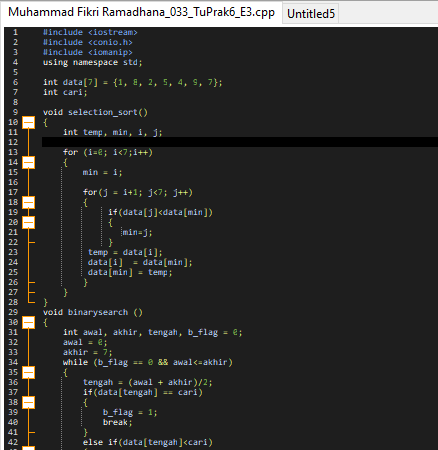
**cout <<" Jumlah karakter :" ; cin>>y;**

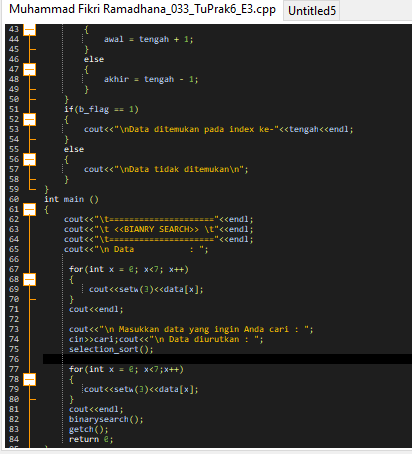
**inputLebarChar(set1, k, 3);**

**}**

Fungsi main dengan isi statement data char a1={‘a’, ‘b’, ‘c’}, integer y = 3, output cout <<" Jumlah karakter :" , cin>>y sebagai proses input nilai ke variabel y, dan inputLebarChar(set1, k, 3); .

**3)**

****

****

**int data[7] = {1, 8, 2, 5, 4, 9, 7};**

**int cari;**

Deklarasi tipe data int [7] = {1, 8, 2, 5, 4, 9, 7}; dan variabel tipe data int cari.

**void selection\_sort() {**

Deklarasi fungsi void selection\_sort merupakan kombinasi antar sorting dan searching.

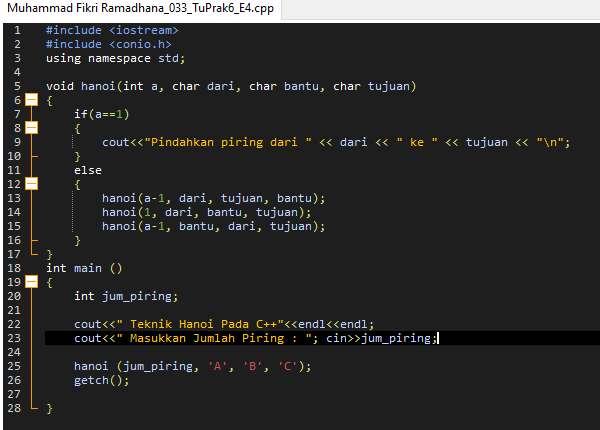
**int temp, min, i, j;**

Deklarasi integer variabel temp, min, i, j.

**for(i=0; i<7;i++) {**

Proses perulangan for dengan nilai awal i = 0, perulangan akan berhenti saat variabel i mencapai kurang dari 7, dan i++ sebagai perulangan bertingkat naik.

**4) Progam rekursif**

****

**void hanoi(int n, char dari, char bantu, char tujuan) {**

Void hanoi(int n, char dari, char bantu, char tujuan){ pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe data char variabel dari, bantu, tujuan, dan tipe data int variabel n.

**if(n==1)**

**cout<<"Pindahkan piring dari " << dari << " ke " << tujuan << "\n";**

**else{ hanoi(n-1, dari, tujuan, bantu);**

**hanoi(1, dari, bantu, tujuan);**

**hanoi(n-1, bantu, dari, tujuan);**

**}**

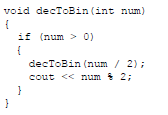
**}**

if(n==1) cout<<"Pindahkan piring dari " << dari << " ke " << tujuan << "\n"; Kondisi jika akan berjalan ketika nilai variabel n=1, maka akan menampilkan kalimat Pindahkan piring dari, nilai variabel dari, menampilkan kalimat ke, nilai variabel tujuan, dan berpindah pada baris berikutnya. else{hanoi(n-1, dari, tujuan, bantu);hanoi(1, dari, bantu, tujuan); hanoi(n-1, bantu, dari, tujuan);}} kondisi tidak (else) maka nilai variabel n yang dimasukkan akan dihitung menggunakan rumus hanoi tersebut. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi void.

**hanoi(jum\_piring, 'A', 'B', 'C');**

Deklarasian variabel hanoi di dalam fungsi int main.

**5) Progam rekursif**

****

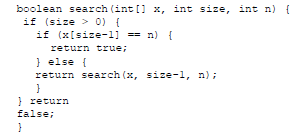
**void decToBin (int num)**

Proses awal mengubah desimal menjadi binary pada variabel tipe integer num.

**if (num > 0)**

Proses perintah dengan kondisi jika (if) variabel num bernilai benar apabila lebih dari 0, dengan instruksi decToBin bernilai variabel num dibagi dua. Kemudian cout variabel num dimodul 2.

**6)** **Progam rekursif**



**bool search (int x[ ], int size, int n)**

Proses tipe data boolean (bool) variabel search dengan kondisi true or false bernilai integer variabel x[ ], size dan n.

**if (size > 0)**

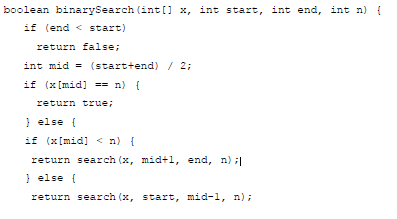
**if (x [size – 1] == n)**

Proses perintah kondisi true berupa variabel x dengan size – 1 sama dengan n didalam kondisi size bernilai benar apabila lebih dari 0.

**else ( )**

Apabila kondisi pertama tidak terpenuhi maka return search bernilai (x, size – 1, n).

**7) Progam rekursif**





**bool binarysearch (int x[ ], int start, int end, int n)**

Proses tipe data boolean binary variabel search dengan kondisi true or false bernilai interger variabel x[ ], start, end, dan n.

**if (end < start)**

Proses kondisi jika (if) variabel end kurang dari start maka statementnya bernilai salah atau return false.

**int mid = (start+end) / 2**

Proses tipe data integer variabel mid sama dengan start ditambah end hasilnya dibagi 2.

**if (x [mid] == n)**

Proses kondisi jika variabel x bernilai operasi mid sama dengan n maka bernilai benar atau true.

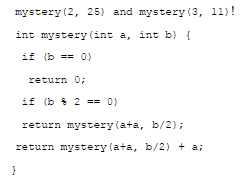
**if (x [mid] < n)**

Proses kondisi jika variabel x bernilai operasi mid kurang dari n maka bernilai return search (x, mid + 1, end, n).

**else ( )**

Apabila kondisi pertama dan kedua tidak terpenuhi maka bernilai return search (x, start, mid – 1, n).

**8) Progam rekursif**



**Int mystery (int a, int b)**

Proses pendeklarasian fungsi mystery dengan parameter formal fungsi tipe data int variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

**if (b == 0)**

**return 0;**

if (b == 0) return 0; merupakan kondisi jika nilai variabel b sama dengan 0 bernilai benar maka nilai balik (return value) adalah 0.

**if (b % 2 == 0)**

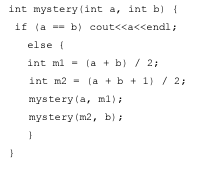
**return mystery(a+a, b/2);**

**return mystery(a+a, b/2);+a**

**}**

Deklarasi if (b % 2 == 0)return mystery(a+a, b/2); merupakan kondisi jika nilai variabel b dimodulo 2 adalah 0 atau bernilai true maka nilai balik (return value) akan dioperasikan ke dalam return mystery. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi int mystery.

**9) Progam rekursif**



**int mystery(int a, int b) {**

Deklarasi integrer mystery (int a, int b) { merupakan fungsi mystery dengan menggunakan parameter formal tipe data int variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

**if (a == b) cout<<a<<endl;**

**else { int m1 = (a+b)/2;**

**int m2 = (a+b+1)/2;**

**mystery(a, m1);**

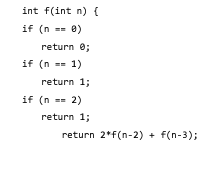
**mystery(m2, b);**

**}**

**}**

Proses kondisi jika (if) variabel a sama dengan nilai variabel b maka bernilai benar. Kemudia jika kondisi satu tidak terpenuhi maka int m1 = (a+b)/2, int m2 = (a+b+1)/2, mystery(a, m1), dan mystery(m2, b).

**10) Progam rekursif**

****

**int f(int n) {**

Deklarasi fungsi f dengan parameter tipe integer variabel n.

**if (n == 0)**

**return 0;**

Proses kondisi jika (if) variabel n sama dengan 0 maka bernilai return 0.

**if (n == 1)**

**return 1;**

Proses kondisi jika (if) variabel n sama dengan 1 maka bernilai return 1.

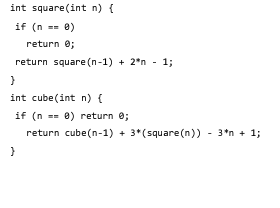
**if (n == 2)**

**return 1;**

**return 2\*f(n-2) + f(n-3);**

Proses kondisi jika (if) variabel n sama dengan 2 maka bernilai return 1 dan return 2\*f(n-2) + f(n-3).

**11) Progam rekursif**



**int square(int n) {**

Pendeklarasikan fungsi square dengan parameter tipe data int variabel n.

**if (n == 0)**

**return 0;**

**return square(n-1) + 2\*n - 1; }**

Proses kondisi jika (if) variabel nilai n sama dengan 0 maka bernilai return 0 dan return square(n-1) + 2\*n – 1.

**int cube(int n) {**

Pendeklarasian fungsi cube dengan parameter fungsi tipe data int variabel n.

**if (n == 0)**

**return 0;**

**return cube(n-1) + 3\*(square(n)) - 3\*n + 1;**

**}**

Proses kondisi jika (if) variabel n sama dengan 0 maka bernilai return 0 dan return cube(n-1) + 3\*(square(n)) – 3\*n + 1 dan diakhri dengan tanda } sebagai penutup.