**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**MODUL 7**

**JUDUL MODUL**

**DISUSUN OLEH :**

**AMMAR BAGAS FATHURRAHMAN W 2350081008**

Logo

Description automatically generated

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS SAINS DAN INFORMATIKA**

**UNIVERSITAS JENDERAL ACHMAD YANI**

**TAHUN 2023**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc153223470)

[DAFTAR GAMBAR ii](#_Toc153223471)

[BAB I. HASIL PRAKTIKUM 3](#_Toc153223472)

[I.1 Tabint.c 3](#_Toc153223473)

[I.1.A. Source Code 3](#_Toc153223474)

[I.1.B. Hasil 4](#_Toc153223475)

[I.1.C. Analisa 4](#_Toc153223476)

[BAB II. TUGAS PRAKTIKUM 5](#_Toc153223477)

[II.1 Array.c 5](#_Toc153223478)

[II.1.A. Source Code 5](#_Toc153223479)

[II.1.B. Hasil 15](#_Toc153223480)

[II.1.C. Analisa 16](#_Toc153223481)

[BAB III. KESIMPULAN 17](#_Toc153223482)

# DAFTAR GAMBAR

[Gambar 1. 1 Hasil program tabint.c 4](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p7\hasil\MODUL%207.docx#_Toc153223558)

[Gambar 2. 1 Hasil program array.c 15](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p7\hasil\MODUL%207.docx#_Toc153223568)

[Gambar 2. 2 Hasil program array.c 15](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p7\hasil\MODUL%207.docx#_Toc153223569)

# HASIL PRAKTIKUM

## Tabint.c

### Source Code

/\* program : tabint.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date : 10/12/2023

desc : memasukan hari ke tabel

\*/

#include <stdio.h>

#include <math.h>

int isprima (int n);

int main(){

int li;

printf("bilangan prima dari 1 sampe dengan 100 : \n");

for (li = 1; li <= 100; li++){

if(isprima(li)){

printf("<%d>", li);

}

}

return 0;

}

int isprima (int n){

int li;

if (n == 2){

return 1; //true

}else if (n % 2 == 0 || n == 1){

return 0; //false

}

}

### Hasil

Gambar 1. 1 Hasil program tabint.c

### Analisa

Pada program tabint.c digunakan sebuah function dengan algoritma atau code untuk kondisional jika di dalam pengulangan bilangan 1 sampai dengan 100 pada main driver maka return yang akan dikeluarkan adalah true atau jika tidak atau bukan bilangan prima maka akan mereturn false kemudian semua bilangan prima pada pengulangan akan dicetak.

# TUGAS PRAKTIKUM

## Array.c

### Source Code

/\* program : array.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date :10/12/2023

desc : membuat tabel dan di isi oleh user dan akan mencetak beberapa fitur menggunakan procedure dan function

\*/

#include <conio.h>

#include <stdio.h>

#include "boolean.h"

#define nMax 10

typedef struct {

int Tb[nMax + 1];

int neff;

}TabInt;

//prototype

//prototype membvuat tabel

void CreateTab(TabInt \*T);

//prototype memasukan elemen

void AddElm(TabInt \*T, int x);

//prototype mencetak tabel

void CetakTabel(TabInt T);

//prototype banyak elemen

int getJumElm(TabInt T);

//prototype jumlah elemen

int SumElm(TabInt T);

//prototype mencari elemen

boolean CariElm(TabInt T, int x);

//prototype banyak nilai ganjil

int JumElmGanjil(TabInt T);

//prototype menjumlahkan nilai genap

int SumElmGEnap(TabInt T);

//prototype copy tabel

TabInt CopyMytab(TabInt T);

//prototype is prima

boolean IsPrima(int n);

//prototype MaxSort

void MaxSort (TabInt \*T);

//prototype CekPrima

void CekPrima (TabInt T);

//main driver

int main () {

//kamus

TabInt Mytab;

TabInt Mytab1;

TabInt Mytab2;

int N, z, o, k, p;

//algoritma

CreateTab(&Mytab);

printf ("masukan bilangan :"); scanf("%d", &N);

while (N != 999){

AddElm (&Mytab, N);

printf ("masukan bilangan :"); scanf("%d", &N);

}

printf("cetak tabel :\n");

CetakTabel(Mytab);

printf("\n\n");

printf("banyak elemen : %d\n", getJumElm(Mytab));

printf("jumlah elemen : %d\n", SumElm(Mytab));

printf("jumlah elemen genap : %d\n", SumElmGenap(Mytab));

printf("banyak elemen ganjil : %d\n", JumElmGanjil(Mytab));

Mytab1 = CopyMytab(Mytab);

printf("\ncetak tabel negasi :\n");

CetakTabel(Mytab1);

printf("\n\nTabel Maxsort\n");

MaxSort(&Mytab);

CetakTabel(Mytab);

printf("\n\n");

CekPrima(Mytab);

printf("\n");

printf("\nmasukan elemen yang mau dicari :"); scanf("%d", &k);

if (CariElm(Mytab, k)){

printf("elemen ada\n");

}else{

printf("elemen tidak ada\n");

}

printf("\nmasukan bilangan :"); scanf("%d", &p);

p = IsPrima(p);

if(p){

printf("Bilangan prima\n");

}else{

printf("Bukan bilangan prima\n");

}

return 0;

}

//body of prototype

void CreateTab (TabInt \*T){

(\*T).neff = 0;

}

void AddElm (TabInt \*T, int x){

if ((\*T).neff < nMax){

(\*T).neff++;

(\*T).Tb[(\*T).neff] = x;

}

}

void CetakTabel (TabInt T){

//kamus lokal

int i;

for (i = 1; i <= T.neff; i++){

printf ("<%d>", T.Tb[i]);

}

}

int getJumElm(TabInt T){

//kamus

int jum;

//algoritma

jum = T.neff;

return jum;

}

int SumElm(TabInt T){

//kamus

int sum;

int i;

//algoritma

sum = 0;

for (i = 1; i <= T.neff; i++){

sum = sum + T.Tb[i];

}

return sum;

}

boolean CariElm(TabInt T, int x){

//kamus

boolean ketemu;

int i;

//algoritma

ketemu = false;

i = 1;

while(i <= T.neff && !ketemu){

if(T.Tb[i] == x){

ketemu = true;

}else{

i++;

ketemu = ketemu;

}

}

return ketemu;

}

int JumElmGanjil(TabInt T){

//kamus

int i;

int ganjil;

//algoritma

ganjil = 0;

for(i = 1; i <= T.neff; i++){

if(T.Tb[i] %2 != 0){

ganjil = ganjil + 1;

}

}

return ganjil;

}

int SumElmGenap(TabInt T){

//kamus

int i;

int jum;

//algoritma

jum = 0;

for(i = 1; i <= T.neff; i++){

if(T.Tb[i] %2 == 0){

jum = jum + T.Tb[i];

}

}

return jum;

}

TabInt CopyMytab(TabInt T){

//kamus

TabInt Copy;

int i;

//algoritma

Copy.neff = 0;

for(i = 1; i <= T.neff; i++){

Copy.neff++;

Copy.Tb[i] = T.Tb[i] \* -1 ;

}

return Copy;

}

boolean IsPrima(int n){

//kamus

boolean ketemu;

//algortima

ketemu = true;

if (n == 2 ){

ketemu = true;

}else if (n % 2 == 0 || n == 1){

ketemu = false;

}

return ketemu;

}

void MaxSort (TabInt \*T){

//kamus

int j, N, i, k;

int mak, Idx;

int Tmp;

//algoritma

N = (\*T).neff;

for (j = N; j >= 1; j--){

mak = 0;

for (i = 1; i <= j; i++){

if((\*T).Tb[i] > mak){

mak = (\*T).Tb[i];

Idx = i;

}

}

Tmp = (\*T).Tb[j];

(\*T).Tb[j] = mak;

(\*T).Tb[Idx] = Tmp;

}

}

void CekPrima (TabInt T){

//kamus

int i;

int p;

//algoritma

for(i = 1; i <= T.neff; i++){

p = T.Tb[i];

IsPrima(p);

p = IsPrima(p);

if(p){

printf("\n<%d> adalah bilangan prima\n", T.Tb[i]);

}else{

printf("\n<%d> adalah bukan bilangan prima\n", T.Tb[i]);

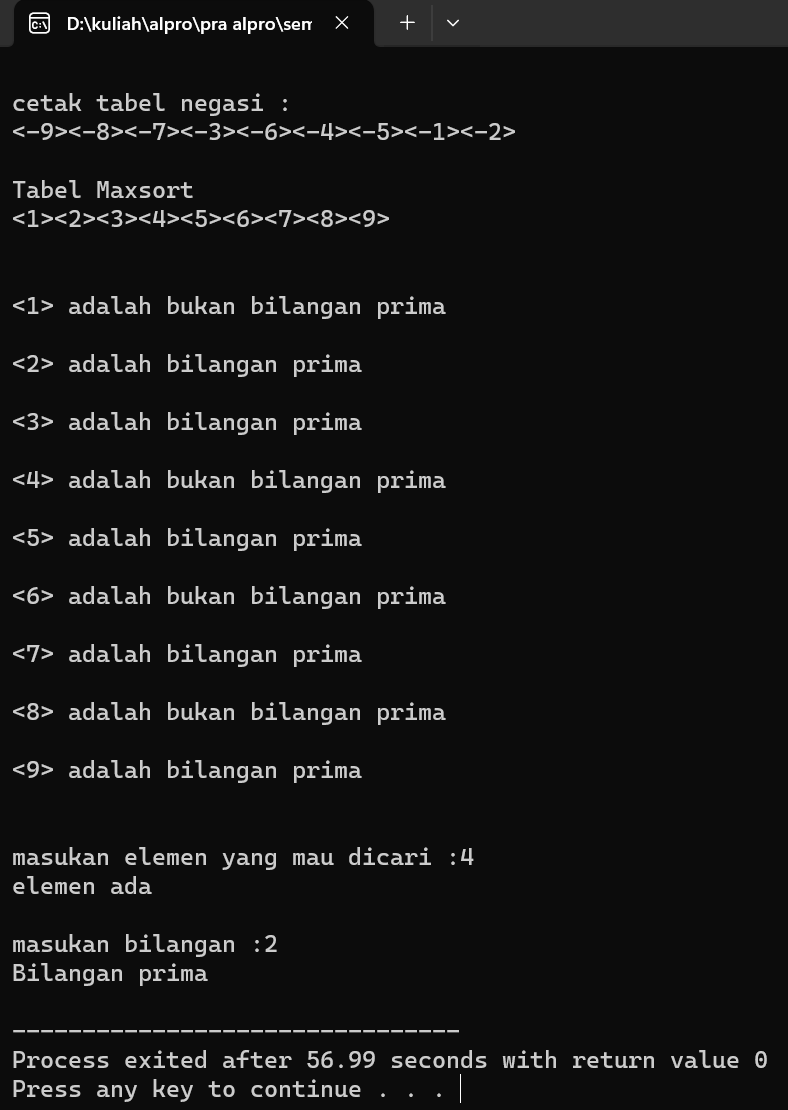
}

}

}

### Hasil

Gambar 2. 1 Hasil program array.c



Gambar 2. 2 Hasil program array.c

### Analisa

Program array.c adalah program yang sama dengan program pertemuan sebelum nya program yang memasukan nilai bilangan interger pada tabel kemudian ditambahkan fitur – fitu atau juga pada pembuatan tabel menggunakan procedure atau function. Pada pertemuan kali ini penambahan fitur ada 3 menggunakan function yaitu boolean isprima, int cekprima, dan maxsort. Fitur boolean isprima adalah untuk pengecekan bilangan yang dimasukan ke function apakah bilangan prima atau bukan jika bilangan prima maka akan mereturn true dan jika buka maka false. Function int cekprima adalah untuk melakukan pengecekan pada isi array yang telah dimasukan pada function ini dilakukan pemanggilan pada function isprima untuk memperpendek codingan dengan begitu output yang akan dihasilkan oleh function ini “<bilangan di dalam array> bilangan prima / bukan bilangan prima”. Kemudian yang terakhir function maxsort untuk melakukan pengurutan pada array dari mulai terkecil hingga yang terbesar.

# KESIMPULAN

Kesimpulan pada praktikum kali ini kita menjadi paham penggunaan function dan kapan function digunakan pada codingan. Dengan adanya function codingan akan menjadi efektif jika terdapat eror dapat lebih mudah ditemukan letak kesalahan nya, dan juga pemanggilan function pada function dapat dilakukan jadi pembuatan code tidak menjadi double.