**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**MODUL 6**

**FUNCTION**

**DISUSUN OLEH :**

**AMMAR BAGAS FATHURRAHMAN WANTORO 2350081008**

Logo

Description automatically generated

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS SAINS DAN INFORMATIKA**

**UNIVERSITAS JENDERAL ACHMAD YANI**

**TAHUN 2023**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc152167831)

[DAFTAR GAMBAR ii](#_Toc152167832)

[BAB I. HASIL PRAKTIKUM 3](#_Toc152167833)

[I.1 Program Fungsi\_1.c 3](#_Toc152167834)

[I.1.A. Source Code 3](#_Toc152167835)

[I.1.B. Hasil 4](#_Toc152167836)

[I.1.C. Analisa 4](#_Toc152167837)

[I.2 Program Function\_1.c 5](#_Toc152167838)

[I.2.A. Source code 5](#_Toc152167839)

[I.2.B. Hasil 10](#_Toc152167840)

[I.2.C. Analisa 10](#_Toc152167841)

[BAB II. TUGAS PRAKTIKUM 11](#_Toc152167842)

[II.1 Program function.c 11](#_Toc152167843)

[II.1.A. Source Code 11](#_Toc152167844)

[II.1.B. Hasil 18](#_Toc152167845)

[II.1.C. Analisa 18](#_Toc152167846)

[II.2 Program akarkdrt.c 19](#_Toc152167847)

[II.2.A. Source code 19](#_Toc152167848)

[II.2.B. Hasil 21](#_Toc152167849)

[II.2.C. Analisa 21](#_Toc152167850)

[BAB III. KESIMPULAN 22](#_Toc152167851)

# DAFTAR GAMBAR

[Gambar 1. 1 Hasil fungsi\_1.c 4](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p6\hasil\laporan.docx#_Toc152167863)

[Gambar 1. 2 hasil program function\_1.c 10](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p6\hasil\laporan.docx#_Toc152167864)

[Gambar 2. 1 Hasil program function.c 18](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p6\hasil\laporan.docx#_Toc152167868)

[Gambar 2. 2 Hasil program akarkdrt.c 21](file:///D:\kuliah\alpro\pra%20alpro\semester%201\p6\hasil\laporan.docx#_Toc152167869)

# HASIL PRAKTIKUM

## Program Fungsi\_1.c

### Source Code

/\* program : fungsi\_1.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date :28/11//2023

desc : mencari nilai diskriminan menggunakan function

\*/

#include<stdio.h>

#include<conio.h>

//prototpe diskriminant

int Diskriminant(int a, int b, int c);

int main(){

//kamus

int a, b, c;

int D;

//algortima

printf("masukan nilai a :"); scanf("%d", &a);

printf("masukan nilai b :"); scanf("%d", &b);

printf("masukan nilai c :"); scanf("%d", &c);

D = Diskriminant(a, b, c);

printf("Diskriminan : %d", D);

return 0;

}

int Diskriminant(int a, int b, int c){

//kamus

int d;

//algoritma

d = ((b\*b) - (4\*a\*c));

return d;

}

### Hasil

Gambar 1. 1 Hasil fungsi\_1.c

### Analisa

Program fungsi\_1.c merupakan program mencari nilai diskriminan dari 3 nilai dan 3 buah variabel menggunakan function. Fucntion berfungsi sebagai pemisah kode seperti proseccedur yang membedakan function dengan proseccedur adalah function memerlukan input dan juga function harus di return atau di outputkan berbeda dengan prosedur yang default nya hanya input. Pada program fungsi\_1.c input pengguna diperlukan dengan 3 buah variabel a, b, c kemudian mencari diskriminan dengan function yang di tulis pada bagian atas main driver dan juga algoritma atau kode function nya ditulis di bagian bawah main driver sama seperti proseccedur. Seperti pada source code di atas nilai diskriminan dicari menggunakan persamaan d = ((b\*b) – (4\*a\*c)) setelah nilai diskirminan diketahui nilai dari diskriminan tersebut di return.

## Program Function\_1.c

### Source code

/\* program : function\_1.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date :28/11//2023

desc : membuat dan mencetak tabel menggunakan prosedur dengan tambahan fitur 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);

//main driver

int main () {

//kamus

TabInt Mytab;

TabInt Mytab1;

int N, z, o, k;

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

if (CariElm(Mytab, k)){

printf("elemen ada\n");

}else{

printf("elemen tidak ada\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;

i = 1;

//algoritma

ketemu = false;

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

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

ketemu = true;

}else{

i++;

}

}

return ketemu;

}

### Hasil

Gambar 1. 2 hasil program function\_1.c

### Analisa

Program function\_1.c merupakan program membuat tabel, mengisi tabel, mencetak tabel menggunakan prosedur dan mencari banyak elemen, jumlah elemen, dan mencari nila elemen menggunakan function. Prototype pada code yang menggunakan prosedur adalah CreateTab, AddElm, CetakTabel. Kemudian pada function adalah getJumElm yang berfungsi sebagai mencari jumlah elemen yang terdapat pada array dengan di dalam code nya mendeklarasikan variabel dan variabel tersebut sama dengan nilai efektif dari array kemudian variabel tersebut di return. Kemudian function ke-2 adalah SumElm yang berfungsi menjumlah semua elemen pada array dengan cara menggunakan pengulangan dari elemen array ke-1 sampai dengan nilai efektif nya, kemudian menjumlahkan antar elemen kemudian terakhir di return variabelnya. Function ke-3 cariElm yang berfungsi mencari elemen yang di input oleh pengguna di dalam array dengan code di dalam nya dilakukan pencarian pengulangan menggunakan while dan menggunakan kondisional pada main driver yang dimana jika nilai yang di input pengguna ada di dalam elemem array maka akan di output “elemen ada” jika tidak ada akan di output “elemen tidak ada”.

# TUGAS PRAKTIKUM

## Program function.c

### Source Code

/\* program : function.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date :28/11//2023

desc : membuat dan mencetak tabel menggunakan prosedur dengan tambahan fitur 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);

//main driver

int main () {

//kamus

TabInt Mytab;

TabInt Mytab1;

int N, z, o, k;

//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");

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

if (CariElm(Mytab, k)){

printf("elemen ada\n");

}else{

printf("elemen tidak ada\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++;

}

}

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++;

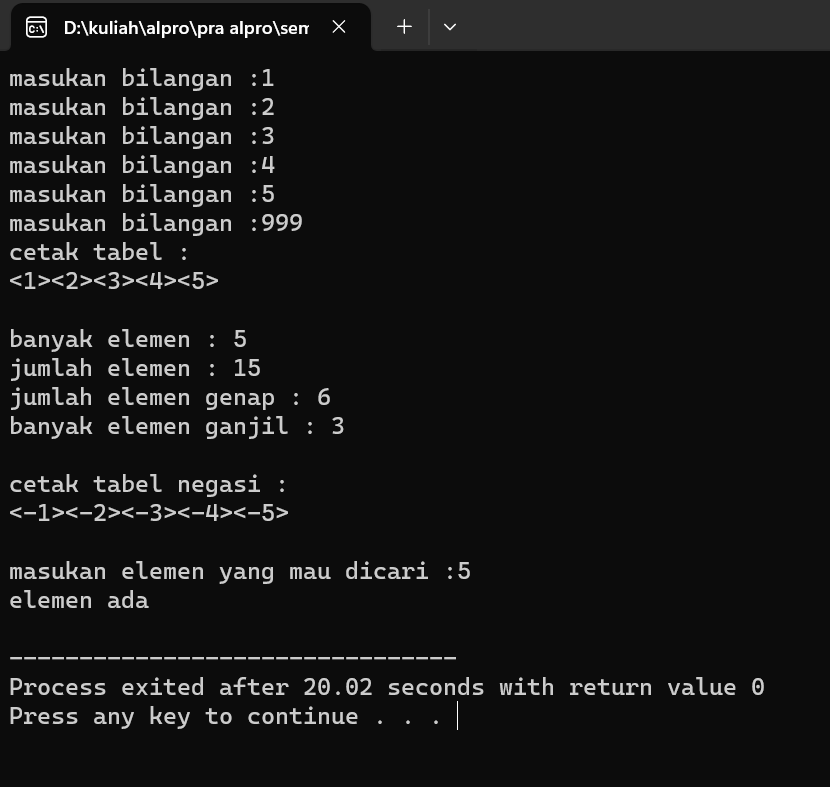
Copy.Tb[i] = T.Tb[i] \* -1 ;

}

return Copy;

}

### Hasil



Gambar 2. 1 Hasil program function.c

### Analisa

Prigram function.c merupakan program yang sama dengan function\_1.c tetapi ditambahkan fitur menghitung banyak elemen ganjil, menjumlahkan elemen genap pada array kemudian mengcopy array yang sama lalu nilai dalam array tersebut di negasikan atau dikalikan dengan negatif 1, setelah itu mencari elemen pada array semua fitur itu menggunakan function pada c agar dapat memudahkan atau dapat mengorganisasikan code lebih mudah.

## Program akarkdrt.c

### Source code

/\* program : function.c

author : Ammar Bagas Fathurrahman Wamtoro

nim : 12350081008

date :28/11//2023

desc : mencari nilai persamaan akar kuadrat dari nilai diskriminan dengan meggunakan fungsi

\*/

#include<stdio.h>

#include<conio.h>

#include<math.h>

//prototype

int akarkdrt(int x, int y, int z);

int Diskriminant(int a, int b, int c);

int main() {

//kamus

int a, b, c;

int D;

double hasil;

//algoritma

printf("masukan nilai a :"); scanf("%d", &a);

printf("masukan nilai b :"); scanf("%d", &b);

printf("masukan nilai c :"); scanf("%d", &c);

D = Diskriminant(a, b, c);

printf("Diskriminan : %d", D);

if(D > 0){

hasil = akarkdrt(a, b, D);

}else{

hasil = 999;

}

printf("\nhasil dari persamaan akar kuadrat : %.2lf", hasil);

return 0;

}

int Diskriminant(int a, int b, int c){

//kamus

int d;

//algoritma

d = ((b\*b) - (4\*a\*c));

return d;

}

int akarkdrt(int x, int y, int z){

//kamus

double hasiltambah;

double hasilkurang;

double hasil;

//algoritma

hasiltambah = (-y + sqrt(z)) / (2 \* x);

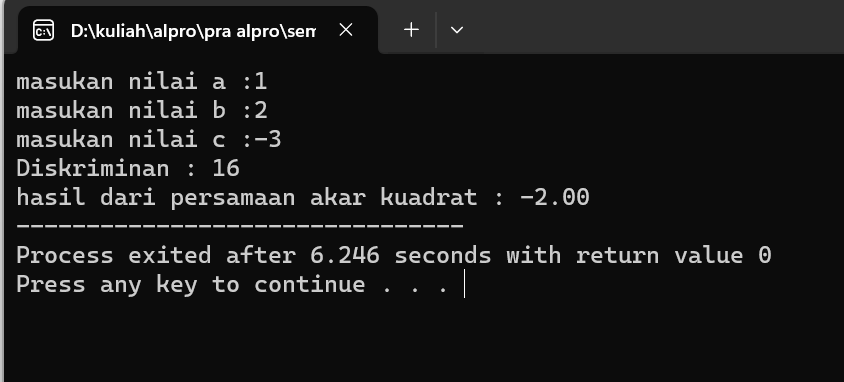
hasilkurang = (-y - sqrt(z)) / (2 \* x);

hasil = hasiltambah + hasilkurang;

return hasil;

}

### Hasil



Gambar 2. 2 Hasil program akarkdrt.c

### Analisa

Program akarkdrt.c merupakan program yang sama dengan program fungsi\_1.c yang berbeda adalah ditambahkan fitur menghitung persamaan kuadrat dengan memanfaatkan diskriminan yang telah di hitung sebelumnya kemudian di masukan ke dalam function akarkdrt setelah itu dihitunglah menggunakan persamaan akar kuadrat tambahan fitur menghitung juga menggunakan function pada c.

# KESIMPULAN

Kesimpulan pada praktikum kali ini adalah kita dapat memahami bagaimana cara function dan procedure bekerja. Dengan function yang membutuhkan input dan juga return dari code nya. Function juga dapat memudahkan pembuat code karena dengan memisahkan beberapa fungsi dari code nya jika terjadi eror ataupun kesalahan logika dapat dicari letak kesalahan lebih mudah, dan juga function dapat dipakai di program yang lain.