

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN**

**MODUL 6
FUNCTION**

DISUSUN OLEH :

AMMAR BAGAS FATHURRAHMAN WANTORO 2350081008



**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN INFORMATIKA
UNIVERSITAS JENDERAL ACHMAD YANI
TAHUN 2023**

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB I. HASIL PRAKTIKUM	3
I.1 Program Fungsi_1.c	3
I.1.A. Source Code	3
I.1.B. Hasil	4
I.1.C. Analisa	4
I.2 Program Function_1.c	5
I.2.A. Source code	5
I.2.B. Hasil	9
I.2.C. Analisa	10
BAB II. TUGAS PRAKTIKUM	11
II.1 Program function.c	11
II.1.A. Source Code	11
II.1.B. Hasil	18
II.1.C. Analisa	18
II.2 Program akarkdrt.c	19
II.2.A. Source code	19
II.2.B. Hasil	21
II.2.C. Analisa	21
BAB III. KESIMPULAN	23

DAFTAR GAMBAR

Gambar 1. 1 Hasil fungsi_1.c	4
Gambar 1. 2 hasil program function_1.c	9
Gambar 2. 1 Hasil program function.c	18
Gambar 2. 2 Hasil program akarkdrt.c	21

BAB I. HASIL PRAKTIKUM

I.1 Program Fungsi_1.c

I.1.A. 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;
    //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);

    return 0;
}
```

```

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

    //kamus

    int d;

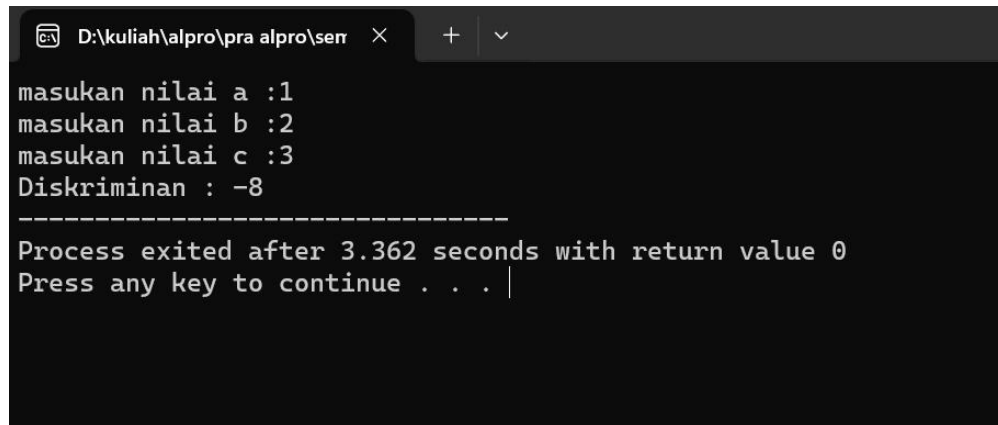
    //algoritma

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

    return d;

}

```



```

D:\kuliah\alpro\pra alpro\sen >
masukan nilai a :1
masukan nilai b :2
masukan nilai c :3
Diskriminan : -8
-----
Process exited after 3.362 seconds with return value 0
Press any key to continue . . .

```

Gambar 1. 1 Hasil fungsi_1.c

I.1.B. Hasil

I.1.C. Analisa

Program fungsi_1.c merupakan program mencari nilai diskriminan dari 3 nilai dan 3 buah variabel menggunakan function. Function berfungsi sebagai pemisah kode seperti prosedur yang membedakan function dengan prosedur 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 prosedur. Seperti pada source code di atas nilai diskriminan dicari menggunakan persamaan $d = ((b*b) - (4*a*c))$ setelah nilai diskriminan diketahui nilai dari diskriminan tersebut di return.

I.2 Program Function_1.c

I.2.A. 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;

}

```

I.2.B. Hasil

```

D:\kuliah\alpro\pra alpro\sen >
masukan bilangan :1
masukan bilangan :2
masukan bilangan :3
masukan bilangan :4
masukan bilangan :5
masukan bilangan :999
cetak tabel :
<1><2><3><4><5>

banyak elemen : 5
jumlah elemen : 15

masukan elemen yang mau dicari :10
elemen tidak ada

-----
Process exited after 8.939 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.2 hasil program function_1.c

I.2.C. Analisa

Program `function_1.c` merupakan program membuat tabel, mengisi tabel, mencetak tabel menggunakan prosedur dan mencari banyak elemen, jumlah elemen, dan mencari nilai 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 elemen array maka akan di output “elemen ada” jika tidak ada akan di output “elemen tidak ada”.

BAB II. TUGAS PRAKTIKUM

II.1 Program function.c

II.1.A. 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 ;

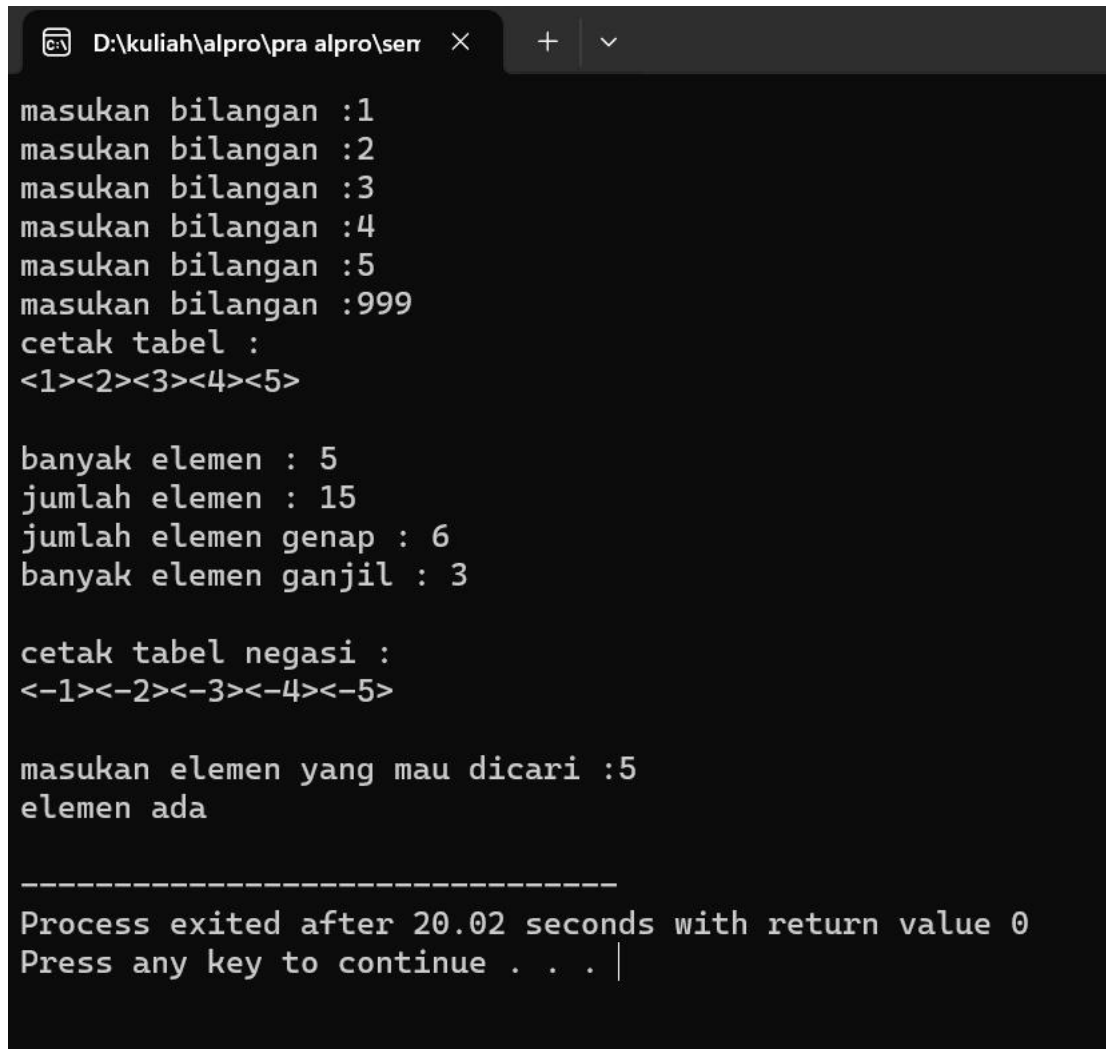
    }

}

```

```
        return Copy;
    }
```

II.1.B. Hasil



```
D:\kuliah\alpro\pra alpro\sen
masukan bilangan :1
masukan bilangan :2
masukan bilangan :3
masukan bilangan :4
masukan bilangan :5
masukan bilangan :999
cetak tabel :
<1><2><3><4><5>

banyak elemen : 5
jumlah elemen : 15
jumlah elemen genap : 6
banyak elemen ganjil : 3

cetak tabel negasi :
<-1><-2><-3><-4><-5>

masukan elemen yang mau dicari :5
elemen ada

-----
Process exited after 20.02 seconds with return value 0
Press any key to continue . . . |
```

Gambar 2. 1 Hasil program function.c

II.1.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.

II.2 Program akarkdrt.c

II.2.A. 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;

}

```

```

D:\kuliah\alpro\pra alpro\sen >
masukan nilai a :1
masukan nilai b :2
masukan nilai c :-3
Diskriminan : 16
hasil dari persamaan akar kuadrat : -2.00
-----
Process exited after 6.246 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 2 Hasil program akarkdrt.c

II.2.B. Hasil

II.2.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 dihitunghlah menggunakan persamaan akar kuadrat tambahan fitur menghitung juga menggunakan function pada c.

BAB III. 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.