

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN**

**MODUL 4
PROCEDURE**

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 pros_1.c	3
I.1.A. Source Code	3
I.1.B. Hasil	5
I.1.C. Analisa	5
I.2 Program pros_2.c	5
I.2.A. Source code	5
I.2.B. Hasil	8
I.2.C. Analisa	8
I.3 Program procedure.c	8
I.3.A. Source code	8
I.3.B. Hasil	14
I.3.C. Analisa	14
BAB II. TUGAS PRAKTIKUM	16
II.1 Sapa.c	16
II.1.A. Source Code	16
II.1.B. Hasil	17
II.1.C. Analisa	17
BAB III. KESIMPULAN	19

DAFTAR GAMBAR

Gambar 1. 1 Hasil dari program pros_1.c	5
Gambar 1. 2 Hasil dari program pros_2.c	8
Gambar 1. 3 hasil dari program procedure.c	14
 Gambar 2. 1 Hasil dari program sapa.c	 17

BAB I. HASIL PRAKTIKUM

I.1 Program pros_1.c

I.1.A. Source Code

```
*      program      : pros_1.c
      author        : Ammar Bagas Fathurrahman Wamtoro
      nim           : 12350081008
      date          : 31/10//2023
      desc          : program memanggil sebuah procedure dalam bahasa c dengan
fungsi menukar nilai variabel
*/

#include <stdio.h>
#include <conio.h>

//program untuk memanggil sebuah procedure

//kamus global
//definis prototype

void tukar (int *a, int *b);

/* I.S : a,b terdefinisi sembarang
   F.S : a dan b saling bertukar nilai */

int main () {
    //kamus lokal pada main drivr
    int bil1, bil2;
```

```

//algoritma

printf("masukan bilangan 1 :"); scanf("%d", &bil1);
printf("masukan bilangan 2 :"); scanf("%d", &bil2);


//procedure

tukar(&bil1, &bil2);

printf("hasil pertukaran bil 1 : %d, bil 2 : %d", bil1, bil2);

return 0;

}


//algoritma procedure

void tukar (int *a, int *b){

    //kamus lokal

    int tmp;

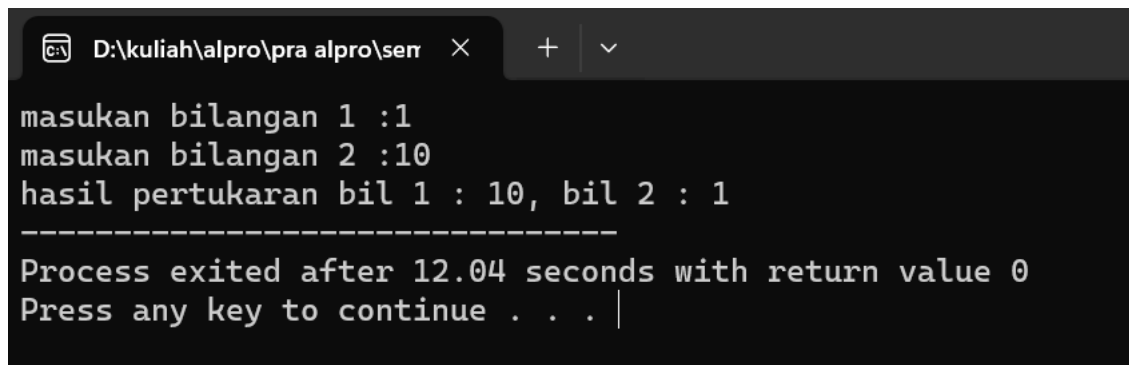
    //algoritma

    tmp = *a;
    *a = *b;
    *b = tmp;

}

```

I.1.B. Hasil



```
D:\kuliah\alpro\pra alpro\sen × + v
masukan bilangan 1 :1
masukan bilangan 2 :10
hasil pertukaran bil 1 : 10, bil 2 : 1
-----
Process exited after 12.04 seconds with return value 0
Press any key to continue . . . |
```

Gambar 1. 1 Hasil dari program pros_1.c

I.1.C. Analisa

Program pros_1.c adalah program yang menukar nilai 2 buah variabel menggunakan operator procedure. Yaitu algoritma untuk menukar nilai variabel berbeda dengan algoritma pada main driver. Dapat dilihat pada source code penggunaan procedure menggunakan void kemudian nama dari procedure nya. Lalu isi dari procedure hampir mirip dengan main driver yang membutuhkan suatu deklarasi dan juga algoritma. Kemudian jika algoritma pada procedure sudah terbuat pemanggilan procedure terjadi 2x letak nya berada di atas main driver dan saat pemanggilan ke 2 pada saat di dalam algoritma main driver. Kemudian letak dari algoritma procedure di bawah main driver.

I.2 Program pros_2.c

I.2.A. Source code

```
/*    program      : pros_1.c
      author       : Ammar Bagas Fathurrahman Wamtoro
      nim          : 12350081008
      date        : 31/10//2023
      desc       : program memannggil sebuah procedure dalam bahasa c dengan
fungsi menukar nilai variabel
*/

#include <stdio.h>
```

```

#include <conio.h>

//program untuk memanggil sebuah procedure

//kamus global

//definis prototype

void tukar (int *a, int *b, int *c);

/* I.S : a,b terdefinisi sembarang
   F.S : a dan b saling bertukar nilai */

int main () {
    //kamus lokal pada main drivr
    int bil1, bil2, bil3;

    //algoritma
    printf("masukan bilangan 1 :"); scanf("%d", &bil1);
    printf("masukan bilangan 2 :"); scanf("%d", &bil2);
    printf("masukan bilangan 3 :"); scanf("%d", &bil3);

    //procedure
    tukar(&bil1, &bil2, &bil3);

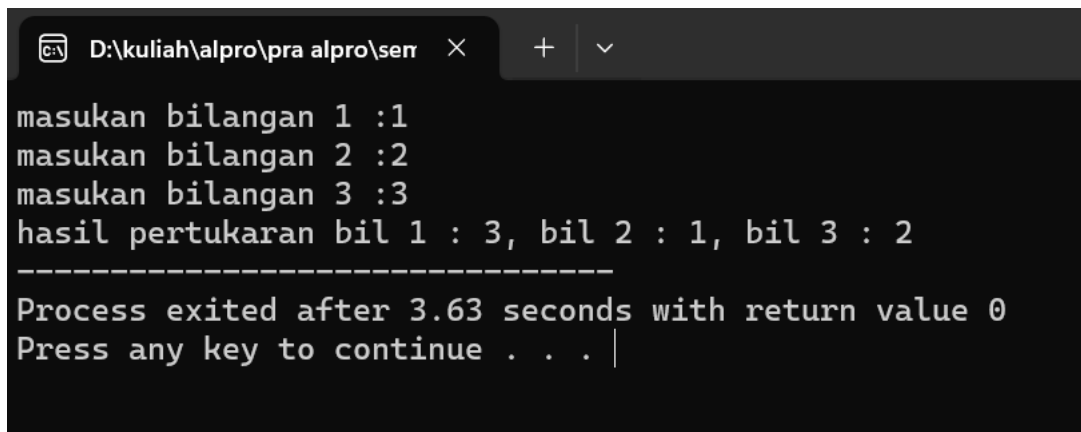
    printf("hasil pertukaran bil 1 : %d, bil 2 : %d, bil 3 : %d",
bil1, bil2, bil3);

    return 0;
}

```

```
//algoritma procedure  
void tukar (int *a, int *b, int *c){  
  
    //kamus lokal  
    int tmp1;  
    int tmp2;  
    int tmp3;  
  
    //algortima  
    tmp1 = *a;  
    tmp2 = *b;  
    tmp3 = *c;  
  
    *a = tmp3;  
    *b = tmp1;  
    *c = tmp2;  
}
```


I.2.B. Hasil



```
D:\kuliah\alpro\pra alpro\sen
masukan bilangan 1 :1
masukan bilangan 2 :2
masukan bilangan 3 :3
hasil pertukaran bil 1 : 3, bil 2 : 1, bil 3 : 2
-----
Process exited after 3.63 seconds with return value 0
Press any key to continue . . . |
```

Gambar 1. 2 Hasil dari program pros_2.c

I.2.C. Analisa

Program pros_2.c merupakan program yang sama dengan program pros_1.c bedanya pada program pros_2.c menukar nilai dari 3 buah variabel dengan menggunakan metode procedure. Pada code algoritma procedure di deklarasikan 3 buah tmp pada kamus lokal procedure agar pertukaran antara 3 variabel dapat di cetak dengan akurat.

I.3 Program procedure.c

I.3.A. Source code

```
/*   program       :  procedur.c
    author        :  Ammar Bagus Fathurrahman Wamtoro
    nim           :  12350081008
    date          :  7/11//2023
    desc          :  cetak tabel menggunakan procedure (Cetak tabel genap),
(Cetak Invers), (InsertFirst), (Dellast)
*/

#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 mencetak tabel genap
void CetakGenap(TabInt T);

//prototype cetak tabel invers
void CetakInvers(TabInt T);

//protoype menyisipkan variabel ke paling depan array
void InsertFirst(TabInt *T, int x);

//menghapus variabel di dalam array yang paling belakang
void DellLast(TabInt *T, int *y);

//main driver
int main () {
    //kamus

```

```

TabInt Mytab;

int N, z, o;


//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");


//cetak tabel genap
printf("\nmencetak tabel genap :\n");
CetakGenap(Mytab);


printf("\n");


//mencetak tabel invers
printf("\nmencetak tabel invers :\n");
CetakInvers(Mytab);


printf("\n");


//insert first
printf("\nmasukan nilai z :"); scanf("%d", &z);

```

```

        InsertFirst (&Mytab, z);

        printf("mencetak tabel yang di sisipkan z di depan :\n");
        CetakTabel(Mytab);

        printf("\n");

        //menghapus nilai akhir dan dimasukan ke dalam variabel o
        o = 0;
        DellLast(&Mytab, &o);

        printf ("\nnilai o : %d\n", o);
        printf("mencetak tabel DellLast :\n");
        CetakTabel(Mytab);

        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]);
        }
    }

void CetakGenap (TabInt T){
    //kamus lokal
    int i;

    //algoritma
    for (i = 1; i <= T.neff; i++){
        if (T.Tb[i] %2 == 0){
            printf ("<%d>", T.Tb[i]);
        }
    }
}

void CetakInvers (TabInt T){
    //kamus lokal
    int i;

    //algoritma
    for (i = T.neff; i >= 1; i--){

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

    }
}

```

```

void InsertFirst (TabInt *T, int x){

    //kamus lokal

    int i, j, last;

    last = (*T).neff;

    //algoritma

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

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

            (*T).Tb[j + 1] = (*T).Tb[j];

            last--;

        }

        (*T).Tb[last + 1] = x;

        (*T).neff++;

    }

}

void DellLast (TabInt *T, int *y){

    //algoritma

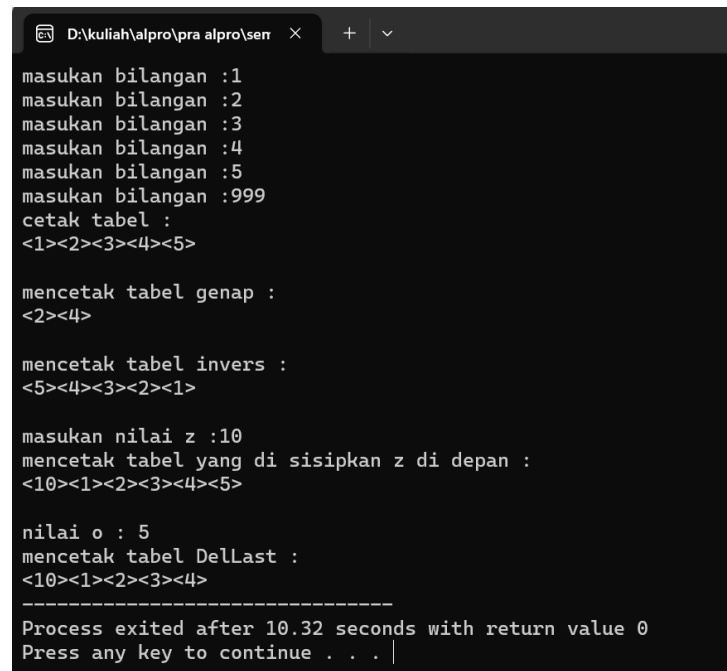
    *y = (*T).Tb[(*T).neff];

    (*T).neff--;

}

```

I.3.B. Hasil



```
D:\kuliah\alpro\pra alpro\sen x + v
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>

mencetak tabel genap :
<2><4>

mencetak tabel invers :
<5><4><3><2><1>

masukan nilai z :10
mencetak tabel yang di sisipkan z di depan :
<10><1><2><3><4><5>

nilai o : 5
mencetak tabel Dellast :
<10><1><2><3><4>
-----
Process exited after 10.32 seconds with return value 0
Press any key to continue . . . |
```

Gambar 1. 3 hasil dari program procedure.c

I.3.C. Analisa

Pada program procedure.c merupakan program membuat tabel dan mencetak nilai genap dari tabel, kemudian ada mencetak tabel invers, lalu ada juga menginput nilai z dan menyisipkannya pada awal tabel, kemudian mencetak tabel yang sudah di delete nilai akhirnya dan mencetaknya pada nilai o itu semua menggunakan procedure. Pada pembuatan tabel yang pertama adalah membuat type data bentukan TabInt yang berarti tabel interger dan yang berisi array dan neff (nilai efektif dari array). Kemudian menggunakan atau memanggil Creatab pada main driver untuk membuat tabel dengan memasukkan nilai 0 pada neff. Setelah itu menggunakan perulangan untuk memasukkan nilai pada array menggunakan procedure AddElm yang berisi kondisional jika neff kurang dari nilai nMax maka neff akan di increament dan dimasukkan nilai yang di input oleh pengguna pada array yang di tunjuk oleh array. Kemudian pada pencetakan tabel menggunakan perulangan for untuk mencetak nilai dari array yang dimulai dari 1 sampai dengan nilai neff. Kemudian jika ingin mencetak nilai genap saja mana tinggal tambakan kondisional jika nilai pada array di bagi dengan 2 dengan modulus 0 mka akan di cetak. Lalu selanjutnya pada cetak invers nilai dari array di cetak terbalik yang di mulai dari neff sampai dengan 1. Pada procedure insertfirst berarti membutuhkan variabel baru untuk menampung sebuah nilai dari input pengguna

dan jika sudah di input tersisip pada bagian awal array, dengan begitu pada array nilai – nilai di dalam nya di geser terlebih dahulu mulai dari last yang sama dengan neff kemudian sampai dengan nilai paling awal di tambah 1 yang berarti bergeser, setelah itu dimasukan nilai dari z yang merupakan nilai dari input pengguna. Kemudian procedure dari DeLast yaitu menghapus nilai bagian akhir dari array dan mencetak nya pada variabel o, masukan dulu variabel neff atau nilai akhir dari array pada variabel o jika sudah neff di decreament

BAB II. TUGAS PRAKTIKUM

II.1 Sapa.c

II.1.A. Source Code

```
.c

    author      : Ammar Bagas Fathurrahman Wamtoro

    nim         : 12350081008

    date        : 7/11//2023

    desc       : program mencetak jika lebih dari umur 17 tahun
menggunakan kak dan jika kurang dari 17 tahun menggunakan dik

    */


#include <stdio.h>

#include <stdlib.h>

#include <string.h>


void sapa (char nama[], int umur);


int main (){

    //kamus

    int umur;

    char nama[100];


    //algoritma

    //input

    printf ("masukan nama :"); scanf ("%s", &nama);


    printf ("masukan umur :"); scanf ("%d", &umur);
```

```

        sapa(nama, umur);

        return 0;
    }

void sapa (char nama[], int umur){

    if (umur >= 17){

        printf("halo kak-%s\n", nama);

        printf("anda sudah cukup dewasa");

    }else{

        printf("halo dik-%s\n", nama);

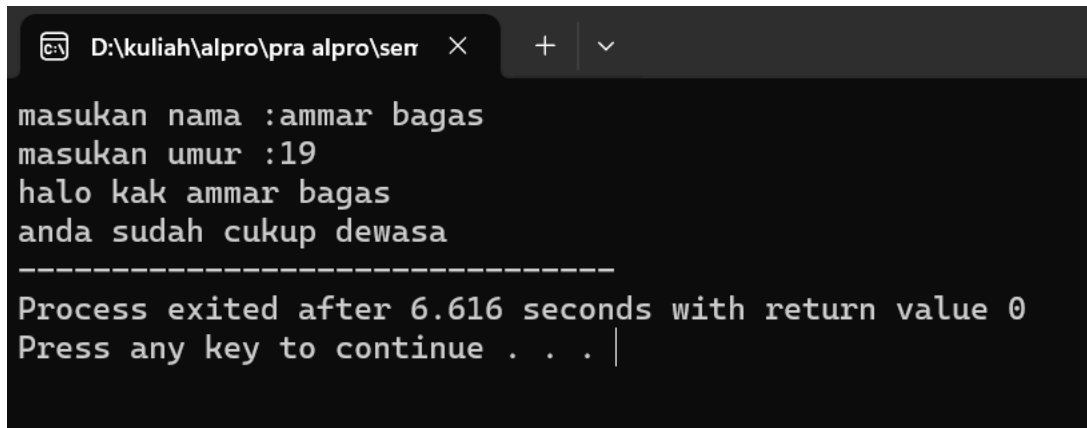
        printf("anda belum cukup dewasa");

    }

}
}

```

II.1.B. Hasil



```

D:\kuliah\alpro\pra alpro\sen
masukan nama :ammar bagas
masukan umur :19
halo kak ammar bagas
anda sudah cukup dewasa
-----
Process exited after 6.616 seconds with return value 0
Press any key to continue . . . |

```

Gambar 2. 1 Hasil dari program sapa.c

II.1.C. Analisa

Sapa.c merupakan program yang membuat atau mencetak sebuah sapaan disaat pengguna memasukkan nama dan umur dengan menggunakan procedure dan kondisional di dalam prordcedure. Dapat dilihat pada algoritma code procedure yang pada bagian algoritma nya menggunakan kondisional jika pengguna menginput umur

lebih dari sama dengan 17 tahun maka sapaan akan menjadi “halo kak (nama yang di input), Anda sudah cukup dewasa” lalu jika umur yang di input adalah kurang dari 17 tahun maka output yang dikeluarkan adalah “halo dik (nama yang di input), Anda belum cukup dewasa”.

BAB III. KESIMPULAN

Kesimpulan pada praktikum kali ini kita menjadi memahami bagaimana fungsi dari sebuah procedure, cara meletakkan procedure dengan benar pemanggilan procedure pada main driver. Dengan begitu menurut saya penggunaan procedure jauh lebih mudah karena code – code yang akan di jalani sudah mendapatkan bagian – bagian tertentu. Jadi jika terjadi eror pada code akan jauh lebih mudah untuk memperbaiki nya.