

## #REKURSIF

---

```
//Faktorial.c
#include <stdio.h>
#include <conio.h>

// prototipe fungsi
int Faktorial(int a);

void main()
{
    int x, hasil;

    printf("Masukkan sebuah angka: \n");
    scanf("%d",&x); /*masukan dari pengguna*/

    hasil = Faktorial(x); /*pemanggilan fungsi*/
    printf("Faktorial = %d", hasil);
    getch();
} /*akhir dari main*/

int Faktorial(int a) /*definisi fungsi*/
{
    int hasil;

    if(a <= 0)
        return(1); // Faktorial(0) = 1
    else
        hasil = a*Faktorial(a-1); /*pemanggilan fungsi dengan rekursi*/

    return(hasil); /*mengembalikan nilai dari Faktorial kepada main*/
} /*akhir dari fungsi Faktorial*/
```

## #STACK

```
//tumpukanArray.c
#include <stdio.h>
#include <conio.h>

#define MAKS 10

// prototipe fungsi
int push(int nil);
int pop();
int Tampil();
int apaKosong();
int apaPenuh();

// deklarasi struktur Tumpukan.
struct Tumpukan //mengimplementasikan Tumpukan sebagai suatu struktur
{
    int s[10];
    int sp;
};

typedef struct Tumpukan tmpkan;
tmpkan st;

int main()
{
    int pilihan, item; // item yang akan ditempatkan ke atas Tumpukan
    st.sp = -1; // -1 mengindikasikan bahwa Tumpukan kosong.

    do
    {
        printf("\n\n");
        printf("\n MENU");
        printf("\n 1:push");
        printf("\n 2:pop");
        printf("\n 3:tampil");
        printf("\n 4:keluar\t\t");

        printf("\n Masukkan pilihan Anda: ");
        scanf("%d",&pilihan);

        switch(pilihan)
        {
            case 1:
                printf("\n Masukkan sebuah elemen ke atas tumpukan: ");
```

```

        scanf("%d", &item); //membaca item
        push(item);
        Tampil();
        break;

    case 2:
        pop();
        Tampil();
        break;

    case 3:
        Tampil();
        break;
    }
}while(pilihan !=4);

return 0;
getch();
} // akhir dari main

// Definisi fungsi
int apaKosong()
{
    if(st.sp == -1)
    {
        printf("Tumpukan Kosong");
        return 1;
    }

    else
        return 0;
} // akhir dari apaKosong()

int apaPenuh ()
{
    if(st.sp == MAKS-1)
    {
        printf("Tumpukan Penuh");
        return 1;
    }

    else
        return 0;
} // akhir dari apaPenuh()

```

```

int push(int nil)
{
    /* item akan ditempatkan ke atas tumpukan. Variabel jwb memuat informasi
       jika tumpukan penuh atau kosong yang dihasilkan oleh fungsi apaPenuh()*/
    int jwb;
    jwb = apaPenuh(); // memanggil apaPenuh()

    if ( jwb == 0) // Tumpukan tidak penuh
    {
        st.sp++; //menginkremen sp ke posisi berikutnya
        st.s[st.sp] = nil; //menugaskan nil kepada st[sp]
    }

    else
    {
        printf("Tumpukan Penuh");
    }

    return 0;
} // akhir dari push().

```

```

int pop()
{
    // jwb memuat nilai balik dari apaKosong().
    // Nilai balik 1 jika tumpukan kosong. 0 jika tumpukan tidak kosong.
    int jwb ;

    jwb = apaKosong();

    if (jwb == 0) // Tumpukan tidak kosong
    {
        //menampilkan item yang dihapus dari atas tumpukan
        printf("\nElemen teratas pada tumpukan adalah %d", st.s[st.sp]);
        st.sp--; //mendekremen Tumpukan[sp]
    }

    else
    {
        printf( "\nTumpukan Kosong. Tidak ada item yang dapat dihapus.");
    }

    return 0;
} // akhir dari pop()

```

```

int Tampil()

```

```
{
    int i ;
    printf("\n");

    if (apaKosong()== 0)
    {
        printf("\n ****Elemen-elemen pada Tumpukan****");

        for(i=st.sp;i>=0;i--) //menampilkan item-item Tumpukan
            printf("\n %d\t",st.s[i]);
    }

    else
    {
        printf("\nTumpukan Kosong.");
    }

    return 0;
} //akhir dari Tampil()
```

## #Tumpukan Senarai Berantai

---

```
//TumpukanSenaraiBerantai.c
# include <stdio.h>
# include <stdlib.h>
#include <conio.h>

#define UKURAN 10

struct Tumpukan
{
    int data;
    struct Tumpukan *brktnya; // struktur referensi-diri
};

typedef struct Tumpukan tmpkan;
tmpkan *tos = NULL;

void push();
int pop();
void Tampil();

void main()
{
    int pilihan = 0;
    int nil;

    do
    {
        printf("\n MENU..... \n");
        printf("\n 1. Tempatkan item data ke atas tumpukan dan tampilkan");
        printf("\n 2. Hapus item data dari atas tumpukan dan tampilkan");
        printf("\n 3. Tampilkan Tumpukan");
        printf("\n 4. Keluar");

        printf("\n Masukkan pilihan Anda: ");
        scanf("%d",&pilihan);

        switch(pilihan)
        {
            case 1:
                push();
                Tampil();
                break;
```

```

        case 2: nil = pop();
                printf("\n Menghapus nilai %d", nil);
                printf("\n Tumpukan setelah penghapusan...");
                Tampil();
                break;

        case 3:
                Tampil();
                break;

        case 4:
                printf("Keluar dari program");
                break;

        default:
                printf("\nPilihan salah <Masukkan antara 1 dan 4");
    }
}while(pilihan!=4);

    getch();
} //akhir dari main

// definisi fungsi
void push()
{
    tmpkan *simpul; // simpul merupakan simpul baru sebagai simpul pertama
    simpul = (tmpkan*)malloc(sizeof(tmpkan));

    printf("\n Masukkan data yang akan ditempatkan ke atas Tumpukan: ");
    scanf("%d", &simpul->data);

    //membuat simpul sebagai simpul pertama pada Tumpukan
    simpul->brktnya = tos;
    tos = simpul; // sekarang tos menunjuk ke simpul yang baru disisipkan
}

int pop()
{
    int nil;

    // menyimpan simpul pertama pada temp
    tmpkan *temp;
    temp = tos;

```

```

//memeriksa apakah Tumpukan kosong
if (tos==NULL)
{
    printf("\n Tumpukan Kosong");
    exit(0);
}

else
{
    nil=tos->data;

    // menggeser tos ke elemen berikutnya
    tos = tos->brktnya;
    free(temp); // membebaskan temp
}
return nil;
}

void Tampil()
{
    tmpkan *temp;
    temp = tos; // Menggunakan temp untuk menjelajah Tumpukan

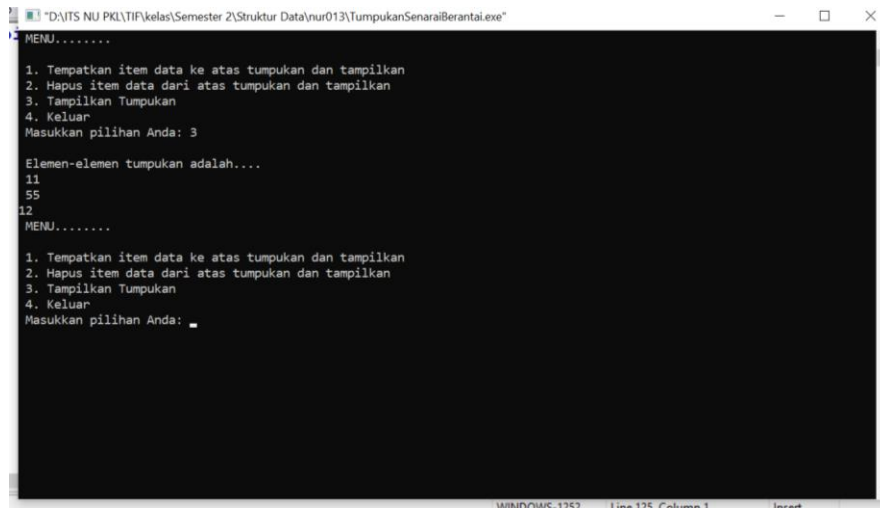
    printf("\n Elemen-elemen tumpukan adalah...\n");
    if (temp == NULL)
        printf("\n Tumpukan Kosong");
    else
    {
        while (temp->brktnya != NULL)
        {
            printf(" %d \n",temp->data);
            temp = temp->brktnya;
        } //akhir dari while

        printf("%d", temp->data);
    }//akhir dari else
}

```



## Hasil Output



```
"D:\ITS NU PKL\TIF\kelas\Semester 2\Struktur Data\nur013\TumpukanSenaraiBerantai.exe"
MENU.....
1. Tempatkan item data ke atas tumpukan dan tampilkan
2. Hapus item data dari atas tumpukan dan tampilkan
3. Tampilkan Tumpukan
4. Keluar
Masukkan pilihan Anda: 3

Elemen-elemen tumpukan adalah...
11
55
12
MENU.....
1. Tempatkan item data ke atas tumpukan dan tampilkan
2. Hapus item data dari atas tumpukan dan tampilkan
3. Tampilkan Tumpukan
4. Keluar
Masukkan pilihan Anda: 
```