

## ADT Stack Dinamis

```
// -----  
// File stackdin.h (library)  
// -----  
  
#ifndef stackdin_H  
#define stackdin_H  
#include "boolean.h"  
  
#define Nil NULL  
#define next(P) (P)->next  
#define data(P) (P)->data  
#define top(L) (L).top  
#include <limits.h>  
#include <malloc.h>  
  
typedef struct telStack *address;  
typedef int infotype;  
typedef struct telStack{  
    infotype data;  
    address next;  
} elStack;  
typedef struct{  
    address top;  
} list;  
  
// PROTOTYPE  
// Predikat untuk keadaan list  
boolean listEmpty (list L);  
/* Fungsi : Mengirim true jika List Kosong */  
  
// Konstruktor/Kreator List Kosong  
void createList (list *L);  
/* IS : L sembarang */  
/* FS : Terbentuk List Kosong */  
  
// Manajemen Memory  
address alokasi (infotype X);  
/* Fungsi : Mengirimkan address hasil alokasi sebuah elemen */  
/* Jika alokasi berhasil, maka address != Nil, */  
/* dan misalnya menghasilkan P, maka data(P) = X, next(P) = Nil */  
/* Jika alokasi gagal, mengirimkan Nil */  
  
void deAlokasi (address P);  
/* IS : P terdefinisi */  
/* FS : P dikembalikan ke sistem */  
/* Fungsi : Melakukan dealokasi / pengembalian address P ke system */  
  
void PUSH (list *L, infotype X);  
/* IS : X adalah data yang akan diinput */  
/* FS : Stack bertambah */  
/* Fungsi : Melakukan penambahan stack*/  
  
void POP (list *L);  
/* IS : terdapat list of Stack */  
/* FS : stack terhapus satu */  
/* Fungsi : Melakukan penghapusan salah satu stack */  
  
void printStack (list L);  
/* IS : terdapat list of stack */  
/* FS : stack ditampilkan */  
/* Fungsi : Melakukan penampilan stack */  
  
list XXXXX(infotype data);
```

```

/* IS : ..... */
/* FS : ..... */
/* Fungsi : ..... */

#endif

// -----
// File b_stack.c (body stack)
// -----

#include "stackdin.h"
#include <stdio.h>

// PROTOTYPE
boolean isEmpty (list L)
{
    return (top(L) == Nil);
}

void createList (list *L)
{
    top(*L)=Nil;
}

// Manajemen Memory
address alokasi (infotype X)
{
    /* Kamus Lokal */
    address P;
    /* Algoritma */
    P = (address) malloc (sizeof (elStack));
    if (P != Nil) /* Alokasi berhasil */
    {
        data(P) = X;
        next(P) = Nil;
    }
    return (P);
}

void deAlokasi (address P)
{
    if (P != Nil)
    {
        free (P);
    }
}

void PUSH (list *L, infotype X)
{
    /* Kamus Lokal */
    address P;
    /* Algoritma */
    P = alokasi(X);
    if (P != Nil)
    {
        next(P) = top(*L);
        top(*L) = P;
        deAlokasi(P);
    }
}

void POP (list *L)
{
    address P;
    P=top(*L);

```

```

        top(*L)=next(top(*L));
        next(P)=Nil;
    }

void printStack (list L)
{
    address P;
    P = top(L);
    while (P != Nil)
    {
        printf("%d ", data(P));
        P=next(P);
    }
}

list XXXXX(infotype data)
{
    infotype data2;
    list L;
    createList(&L);
    while (data!=0)
    {
        if (data%2==0)
        {
            data2=data%2;
            PUSH(&L, data2);
            data=data-(data/2);
        }
        else
        {
            data2=data%2;
            PUSH(&L, data2);
            data=(data-(data%2))/2;
        }
    }
    return (L);
}

//-----
// File m_stack.c (driver / main program)
//-----

#include "stackdin.h"
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int main()
{
    // deklarasi
    infotype data;
    list L;
    char pilihan;

    do{
        // menu
        clrscr();
        printf("\n UTS SDA\n\n");
        printf("Pilihan:\n");
        printf("1. Proses Utama \n");
        printf("2. Keluar atau Tekan Q \n");
        pilihan = getch();

        // proses utama
        if (pilihan=='1' || pilihan == 't' || pilihan == 'T')

```

```

    {
        do{
            clrscr();
            printf("Masukkan nilai yang akan dikonversi= ");
            scanf("\n%d", &data);
            L=XXXXX(data);
            printf("\n\nHasil Proses = ");
            printStack(L);

            printf("\n\nTekan: \nC: Coba lagi \nQ: Keluar \nMenu utama:
Home/Esc \n ");
            pilihan=getch();
        }while ((pilihan=='C') || (pilihan=='c'));
    }

} while (pilihan != '2' && pilihan != 'Q' && pilihan != 'q');

clrscr();
printf("Thank you.. \n\nPress any key ...");
getch();
_exit(0);
return 0;
}

```