

PRAKTIKUM 3 SDA KELAS 1 – DOSEN : SN

(*Pointer to Structure dalam bentuk linear list (single linked list)*)

```
/*-----*/
/* File      : PTR10.cpp                               */
/* Deskripsi  : contoh pendefinisian struct dan pointer ke struct */
/* Dibuat oleh : Tim Dosen SDP                               */
/* Tanggal   : 13-09-2001                                   */
/*-----*/

#include<stdlib.h>
#include<stdio.h>

int main()
{
    /* kamus */
    /* cara 3 : Ini yang membuat TYPE baru */
    typedef struct {
        float x;
        float y;
    } Point;          /* nama type */

    Point *P2;          /* deklarasi pointer ke struct */
    Point *P3 = (Point *) malloc (sizeof (Point)); /* deklarasi alokasi */

    /* program */

    /* Cara 1 : mengacu elemen pointer ke struct */
    printf("Titik P2, dengan P2->x dan P2->y : \n");
    P2 = (Point *) malloc (sizeof(Point));          /* Alokasi */
    P2->x = 9.12;          /* Isi nilai komponen */
    P2->y = 2.567;

    printf("P2.x = %f \nP2.y = %f \n", P2->x, P2->y);

    /* Cara 2 : Perhatikanlah tanda kurung, lihat prioritas () dibuku TC */
    (*P3).x = 0.5;          /* Mendefinisikan isi */
    (*P3).y = 10.5;
    printf("P3.x = %f \nP3.y = %f \n", (*P3).x, (*P3).y);

    return 0;
}
```

```

/*-----*/
/* File      : PTR11.cpp                               */
/* Deskripsi  : contoh deklarasi list dan pemakaian makro */
/* Dibuat oleh : Tim Dosen SDP                         */
/* Tanggal   : 13-09-2001                             */
/*-----*/

#include<stdlib.h>
#include<stdio.h>

/*Definisi akses komponen type, standard kuliah Algoritma dan pemrograman*/
#define info(P) (P)->info
#define next(P) (P)->next
#define Nil NULL

/* Definisi TYPE global (sebenarnya untuk soal ini tidak perlu global) */
/* Element list linier */
typedef int infotype;
typedef struct tElmtlist *address;
typedef struct tElmtlist {
    infotype info;
    address next;
} ElmtList;

/* Program Utama */
int main()
{
    /* kamus */
    address First;
    address P, Q;

    /* program */

    /* Create list kosong */
    First = Nil;
    /* Alokasi, insert as first elemen */
    P = (address) malloc(sizeof (ElmtList));
    info(P) = 10;
    next(P) = Nil;
    First = P;

    /* Alokasi, insert as first elemen */
    Q = (address) malloc(sizeof (ElmtList));
    info(Q) = 20;
    next(Q) = Nil;
    next(Q) = First;
    First = Q;

    /* Alokasi, insert as first elemen */
    P = (address) malloc(sizeof (ElmtList));
    info(P) = 30;
    next(P) = Nil;
    next(P) = First;
    First = P;

    printf("%d \n", info(next(next(First))));

    return 0;
}

```