

PRAKTIKUM 2 #2 SDP KELAS 1 – DOSEN : SN

(Bagian ke-2 : Pointer dan fungsi/prosedur)

- Tulis kembali naskah program berikut kemudian eksekusi!
- Beri komentar program '*secukupnya*' sehingga bila dibaca kembali akan mudah untuk dipahami!
- Tulis hasil pemahaman Anda terhadap maksud dari program-program berikut !
- Tulis hal baru apa saja yang Anda peroleh setelah mempelajari program-program berikut !

```
/* ----- */
/* File      : PTR10.cpp                      */
/* Deskripsi  : Pointer dan fungsi/prosedur   */
/* ----- */

#include <stdio.h>
/* Prototype */
void f1(void);
void f2(void);
void f3(void);
void f4(void);

/* kamus Global */
#define true 1
#define false 0
int quit = false;

int main()
{
    /* kamus lokal */
    /* Definisi tabel yang elemennya adalah pointer ke fungsi */
    /* Elemen tabel yang ke - i akan mengakses fungsi ke - i */
    /* Pilihan menjadi indeks tabel, dan dipakai untuk mengaktifkan fungsi */
    /* yang sesuai */

    void (*tab[4]) () = {f1, f2, f3, f4};          /* Pointer ke procedure */

    /* program */
    printf("Pointer to function : \n");
    /* Menu */
    do
    {
        printf("Pilih salah satu : \n");
        printf("1. Buka file hanya untuk baca \n");
        printf("2. Tutup file \n");
        printf("3. Edit file \n");
        printf("4. Quit \n");
        tab[getchar() - '1'] () ;
        getchar();          /* untuk membuang return karakter */
    } while (!quit);

    return 0;
}

/* Body Function */
void f1 ()
{
    printf("Ini fungsi f1 \n");
}
```

```

void f2 ()
{
    printf("Ini fungsi f2 \n");
}

void f3 ()
{
    printf("Ini fungsi f3 \n");
}

void f4 ()
{
    quit = true;
    printf("Quit ... \n");
}

/* ----- */
/* File      : PTR11.cpp                               */
/* Deskripsi  : Pointer ke function (Function sebagai parameter) */
/*           Melakukan offset terhadap tabel tergantung fungsi f */
/* ----- */

/* Kamus Global */
int N;                                /* ukuran efektif */

/* Prototype, fungsi yang diberikan sebagai parameter aktual */
int succ (int i);                     /* succ -> singkatan dari suksesor */
int pred (int i);                     /* pred -> singkatan dari predesesor */

/* Prosedur dengan parameter sebuah fungsi */
void geser (int *TT, int (*f) (int));

/* Program Utama */
int main()
{
    /* kamus lokal */
    int MyTab[10];
    int i;

    /* program */
    N = 10;
    for (i = 0; i < N; i++)
    {
        MyTab[i] = i;
    }
    printf("Isi tabel dalam main sebelum pemanggilan : \n");
    for (i = 0; i < N; i++)
    {
        printf(" %d ", MyTab[i]);
    }
    printf("\n");

    /* Pakai geser dengan parameter succ */
    printf("Geser dengan succ \n");
    geser (MyTab, succ);
    printf(" dalam main \n");
    for (i = 0; i < N; i++)

```

```

        {
            printf(" %d ", MyTab[i]);
        }
        printf("\n");

        /* Pakai geser dengan parameter pred */
        printf("Geser dengan pred \n");
        geser (MyTab, pred);
        printf(" dalam main setelah aplikasi pred \n");
        for (i = 0; i < N; i++)
        {
            printf(" %d ", MyTab[i]);
        }
        printf("\n");
        return 0;
    }

    /* Body Function */
    int succ (int i)
    {
        return (i+1);
    }

    int pred (int i)
    {
        return (i-1);
    }

    void geser (int *TT, int (*f) (int))
    {
        int i;

        printf(" dalam geser \n");
        for (i = 0; i < N; i++)
        {
            TT[i] = f (TT[i]);
            printf(" %d ", TT[i]);
        }
        printf("\n");
    }

    /* ----- */
    /* File      : PTR12.cpp                               */
    /* Deskripsi  : Pointer ke function (Procedure dengan parameter input/output sebagai parameter) */
    /*                                           Melakukan offset terhadap tabel tergantung fungsi f */
    /* ----- */

    /* Kamus Global */
    int N;                                /* ukuran efektif */

    /* Prototype, fungsi yang diberikan sebagai parameter aktual */
    void succ (int *i);                    /* suksesor, berupa procedure by ref */
    void pred (int *i);                    /* predesesor */

    /* Prosedur dengan parameter sebuah fungsi */
    void geser (int *TT, void (*f) (int *));

    /* Program Utama */

```

```

int main()
{
/* kamus lokal */
    int MyTab[10];
    int i;

/* program */
    N = 10;
    for (i = 0; i < N; i++)
    {
        MyTab[i] = i;
    }
    printf("Isi tabel dalam main sebelum pemanggilan : \n");
    for (i = 0; i < N; i++)
    {
        printf(" %d ", MyTab[i]);
    }
    printf("\n");

/* Pakai geser dengan parameter succ */
    printf("Geser dengan succ \n");
    geser (MyTab, succ);
    printf(" dalam main \n");
    for (i = 0; i < N; i++)
    {
        printf(" %d ", MyTab[i]);
    }
    printf("\n");

/* Pakai geser dengan parameter pred */
    printf("Geser dengan pred \n");
    geser (MyTab, pred);
    printf(" dalam main setelah aplikasi pred \n");
    for (i = 0; i < N; i++)
    {
        printf(" %d ", MyTab[i]);
    }
    printf("\n");
    return 0;
}

/* Body Function */
void succ (int *i)
{
    *i = *i+1;
}

void pred (int *i)
{
    *i = *i-1;
}

void geser (int *TT, void (*f) (int *))
{
    int i;

    printf(" dalam geser \n");
    for (i = 0; i < N; i++)

```

```

        {
            f (&TT[i]);
            printf(" %d ", TT[i]);
        }
    printf("\n");
}

/* ----- */
/* File      : PTR13.cpp */
/* Deskripsi  : Pointer ke function (Procedure dengan parameter input/output sebagai parameter) */
/*                               Melakukan offset terhadap tabel tergantung fungsi f */
/* ----- */

/* Kamus Global */
    int N;                                /* ukuran efektif */
    enum MyType
    {
        bulat, karakter
    };

/* Prototype, fungsi yang diberikan sebagai parameter aktual */
void succI (int *i);                      /* suksesor, berupa procedure by ref */
void predI (int *i);                      /* predesesor */
void succC (char *c);                     /* suksesor, berupa procedure by ref */
void predC (char *c);                     /* predesesor */

/* print tabel yang belum ketahuan typenya */
void printtab (void *T, enum MyType Ty);

/* Prosedur dengan parameter sebuah fungsi */
void geser (int *TT, void (*f) (void *));

/* Program Utama */
int main()
{
    /* kamus lokal */
    void *MyTabInt;
    void *MyTabC;
    int i;

    /* program */
    N = 10;
    MyTabInt = (int *) malloc (N * sizeof(int));
    MyTabC = (char *) malloc (N * sizeof(char));
    *MyTabInt = 1;
    for (i = 0; i < N; i++)
    {
        *(MyTabInt + i) = i;
        *(MyTabC + i) = 'Z';
    }
    printf("Isi tabel dalam main sebelum pemanggilan : \n");

    printf("  Tabel integer \n");
    printtab ((int *) MyTabInt, 0);
    printf("  Tabel karakter \n");
    printtab ((char *) MyTabC, 1);

    printf("\n");
}

```

```

/* Pakai geser dengan parameter succ */
printf("Geser dengan succ \n");
geser ((int *) MyTabInt, (int *) succI);
geser ((char *) MyTabC, (char *) succC);
printf(" dalam main \n");
printf("  Tabel integer \n");
printtab ((int *) MyTabInt, 0);
printf("  Tabel karakter \n");
printtab ((char *) MyTabC, 1);
printf("\n");

/* Pakai geser dengan parameter pred */
printf("Geser dengan pred \n");
geser ((int *) MyTabInt, (int *) predI);
geser ((char *) MyTabC, (char *) predC);
printf(" dalam main \n");
printf("  Tabel integer \n");
printtab ((int *) MyTabInt, 0);
printf("  Tabel karakter \n");
printtab ((char *) MyTabC, 1);
printf("\n");
return 0;
}

/* Body Function */
void succI (int *i)
{
    *i = *i+1;
}

void predI (int *i)
{
    *i = *i-1;
}

void succC (char *c)
{
    *c = *c+1;
}

void predC (char *c)
{
    *c = *c-1;
}

void geser (int *TT, void (*f) (void *))
{
    int i;

    printf(" dalam geser \n");
    for (i = 0; i < N; i++)
    {
        f (&TT[i]);
        printf(" %d ", TT[i]);
    }
    printf("\n");
}

```

```
void printtab (void *T, enum MyType Ty)
{
    int i;

    for (i = 0; i < N; i ++)
    {
        switch (Ty)
        {
            case bulat : printf("%d ", (int *) *(T + i)); break;
            case karakter : printf("%c ", (char *) *(T + i)); break;
        }
    }
}
```