

## LEMBAR PRAKTIKUM – IMPLEMENTASI DASAR PADA ALOKASI DINAMIS

### PTR1

```
/* ----- */
/* File Program : PTR1.CPP */
/* Contoh pemakaian pointer */
/* Tujuan : mengetahui panjang dan posisi variabel di memory */
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    int x, y;
    int *px;
    x = 87;
    px = &x;
    y = *px;
    printf("Alamat x = %p\n", &x);
    printf("Isi px = %d\n", x);
    printf("Nilai yang ditunjuk oleh px = %d\n", *px);
    printf("Nilai y = %d\n", y);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Tambahkan pernyataan untuk menampilkan Alamat varabel y, eksekusi, dan perhatikan hasilnya
- 4) Bila pernyataan `px = &x;` diganti dengan `y = *px;`, apa yang terjadi?
- 5) Tuliskan hasil pemahaman Anda

### PTR2

```
/* ----- */
/* File Program : PTR2.CPP */
/* Contoh pemakaian pointer yang salah */
/* Tujuan : mengetahui panjang dan posisi variabel di memory */
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    float *pu;
    float nu;
    int u = 1234;
    pu = &u;
    nu = *pu;
    printf("u = %d\n", u);
    printf("nu = %f\n", nu);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Tuliskan hasil pemahaman Anda

### PTR3

```
/* ----- */
/* File Program : PTR3.CPP */
/* Contoh pengubahan isi suatu variabel melalui pointer*/
/* Tujuan : menggunakan type data pointer dan manipulasinya*/
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    float d, *pd;
    d = 54.5;
    printf("Isi d semula = %g\n", d);
    pd = &d;
    *pd = *pd + 10;
    printf("Isi d kini = %g\n", d);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Tuliskan hasil pemahaman Anda

### PTR4

```
/* ----- */
/* File Program : PTR4.CPP */
/* Contoh operasi pemakaian pointer*/
/* Tujuan : melakukan operasi pada nilai yang ditunjuk pada pointer*/
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    int z, s, *pz, *ps;
    z = 20;
    s = 30;
    pz = &z;
    ps = &s;
    *pz = *pz + *ps;
    printf("z = %d, s = %d\n", z , s);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Modifikasilah pernyataan di atas, agar nilai Z (20) tidak ter-replace dengan nilai hasil operasi pertambahan. Pengubahan dilakukan sedemikian rupa sehingga terdapat nilai Z, S dan hasil pertambahannya.
- 4) Tuliskan hasil pemahaman Anda

## PTR5

```
/* ----- */
/* File Program : PTR5.CPP */
/* Contoh pointer ke type dasar, mendeklarasikan & alokasi variabel dinamik */
/* Tujuan : memahami oprasi pada pointer*/
/* ----- */
```

```
#include <stdlib.h>
#include <stdio.h>
#include int main() {
    /* Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    int i = 5, j;
    char c = 'X';
    int *Ptri = (int *) malloc(4);
    int *Ptrj = (int *) malloc(sizeof(int));
    float *fx = (float *) malloc(sizeof(float));
    int A[5];
    float f = 7.23;

    /* program */
    *Ptri = 8;
    *Ptrj = 0;
    *fx = 3;
    printf("Alamat i = %x \n", &i);
    printf("Nilai i = %d \n", i);
    printf("Ukuran int = %d byte\n\n", sizeof(int));
    printf("Alamat j = %x \n", &j);
    printf("Nilai j = %d \n", j);
    printf("Alamat c = %x \n", &c);
    printf("Nilai c = %c \n", c);
    printf("Ukuran char = %d byte\n\n", sizeof(char));
    printf("Alamat Ptri = %x \n", &Ptri);
    printf("Isi var Ptri = %x \n", Ptri);
    printf("Nilai yang ditunjuk oleh Ptri = %d \n", *Ptri);
    printf("Ukuran pointer int = %d byte\n\n", sizeof(int *));
    printf("Alamat Ptrj = %x \n", &Ptrj);
    printf("Isi var Ptrj = %x \n", Ptrj);
    printf("Nilai yang ditunjuk oleh Ptrj = %d \n", *Ptrj);

    Ptrj = Ptri;
    printf("Isi var Ptrj sekarang = %x \n", Ptrj);
    printf("Nilai yang ditunjuk oleh Ptrj sekarang = %d \n", *Ptrj);
    printf("Alamat A = %x \n", &A);
    printf("Isi var A = %x \n", A[0]);
    printf("Ukuran array A = %d byte\n\n", sizeof(A));
    printf("Alamat f = %x \n", &f);
    printf("Nilai f = %f \n", f);
    printf("Ukuran float = %d byte\n\n", sizeof(float));
    return 0;
}
```

### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Tuliskan hasil eksekusi setiap statement program dan
- 3) Tuliskan hasil pemahaman Anda

## LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS PADA ARRAY

### PTR6

```
/* ----- */
/* File Program : PTR6.CPP */
/* Contoh pointer menunjuk ke array*/
/* Tujuan : memahami operasi pointer yang menunjuk kepada array*/
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    /* Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    static int tgl_lahir[] = {18, 6, 1989};
    int *ptgl;

    ptgl = tgl_lahir;
    printf("Nilai yang ditunjuk oleh ptgl = %d\n", *ptgl);
    printf("Nilai dari tgl_lahir[0] = %d\n", tgl_lahir[0]);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Modifikasi program tersebut agar bisa menampilkan seluruh elemen array tgl\_lahir !
- 4) Tuliskan hasil pemahaman Anda

## LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS PADA STRING Bag 1

### PTR7

```
/* ----- */
/* File Program : PTR7.CPP */
/* Contoh Pointer dan String bagian ke-1*/
/* Tujuan : memahami operasi pointer yang menunjuk ke data string*/
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    /*Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    char *pkota = "BANDUNG";

    puts(pkota);
    return 0;;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Apa perbedaan deklarasi variabel char \*pkota='BANDUNG'; dengan char kota[]='BANDUNG'
- 4) Tuliskan hasil pemahaman Anda

## LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS PADA STRING Bag 2

### PTR8a

```
/* ----- */
/* File Program : PTR8a.CPP */
/* Contoh Pointer dan String bagian ke-2*/
/* Tujuan : memahami operasi menukarkan isi dua buah string TANPA pemakaian
pointer */
/* ----- */
```

```
#include <stdio.h>
#include <string.h>
#define PANJANG 20
#include int main() {
    /* Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    char nama1[PANJANG] = "DEWI SARTIKA";
    char nama2[PANJANG] = "SULTAN HASANUDDIN";
    char namaX[PANJANG];
    puts("Nama semula : ");
    printf("Nama 1 --> %s\n", nama1);
    printf("Nama 2 --> %s\n", nama2);

    /* Pertukaran string */
    strcpy(namaX, nama1);
    strcpy(nama1, nama2);
    strcpy(nama2, namaX);
    puts("Nama sekarang : ");
    printf("Nama 1 --> %s\n", nama1);
    printf("Nama 2 --> %s\n", nama2);
    return 0;
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Tuliskan hasil pemahaman Anda

## LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS PADA STRING Bag 3

### PTR8b

```
/* ----- */
/* File Program : PTR8b.CPP */
/* Contoh Pointer dan String bagian ke-3*/
/* Tujuan : memahami operasi menukarkan isi dua buah string yang ditunjuk
oleh pointer */
/* ----- */
```

```
#include <stdio.h>
#include int main() {
    /* Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    char nama1[PANJANG] = "DEWI SARTIKA";
    char nama2[PANJANG] = "SULTAN HASANUDDIN";
    char namaX[PANJANG];
    puts("Nama semula : ");
    printf("Nama 1 --> %s\n", nama1);
    printf("Nama 2 --> %s\n", nama2);
```

```

/* Pertukaran string */
strcpy(namaX, nama1);
strcpy(nama1, nama2);
strcpy(nama2, namaX);
puts("Nama sekarang : ");
printf("Nama 1 --> %s\n", nama1);
printf("Nama 2 --> %s\n", nama2);
return 0;
}

```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Amatilah pernyataan di bawah ini. Kemudian berikan pendapat Anda mengenai pernyataan tsb.  
*"PERHATIKAN! Bila dibandingkan dengan program PTR8a.cpp, maka dengan pointer, penyalinan isi array tidak perlu dilakukan. Sehingga bisa menghemat waktu eksekusi."*
- 4) Tuliskan hasil pemahaman Anda

### LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS ADVANCE (POINTER NUNJUK POINTER)

#### PTR9

```

/* ----- */
/* File Program : PTR9.CPP */
/* Contoh Pointer yang menunjuk ke pointer*/
/* Tujuan : memahami operasi pointer yang menunjuk ke pointer */
/* ----- */

#include <stdio.h>
#include int main() {
    /* Kamus Data (deklarasi, inisialisasi nilai variabel)*/
    int var_x = 273;
    int *ptr1; /* ptr1 adl pointer yang menunjuk ke objek bertipe integer */
    /* (Boleh disebut pointer integer saja) */
    int **ptr2; /* ptr2 adl pointer yang menunjuk ke pointer yang menunjuk */
    /* ke objek bertipe integer (Boleh disebut : pointer menunjuk ke pointer
    integer saja) */

    ptr1 = &var_x;
    ptr2 = &ptr1;

    /* Mengakses nilai var_x melalui ptr1 dan ptr2 */
    printf("Nilai var_x = %d\n", *ptr1);
    printf("Nilai var_x = %d\n", **ptr2);
    return 0;
}

```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Komentari setiap baris program (maksud tiap pernyataan di tiap baris program)
- 3) Tuliskan hasil pemahaman Anda

## LEMBAR PRAKTIKUM – IMPLEMENTASI ALOKASI DINAMIS PADA FUNGSI/PROSEDUR

### PTR10

```
/* ----- */
/* File Program : PTR10.CPP */
/* Contoh Pointer yang menunjuk ke prosedur*/
/* Tujuan : memahami operasi pointer yang menunjuk ke 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;

#include 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(void){
    printf("Ini prosedur f1 \n");
}
void f2(void){
    printf("Ini prosedur f2 \n");
}
void f3(void){
    printf("Ini prosedur f3 \n");
}
}
```

```
void f4(void){
    printf("Ini prosedur f4 \n");
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Beri komentar program 'secukupnya' sehingga bila dibaca kembali akan mudah untuk dipahami!
- 3) Tuliskan hasil pemahaman Anda

#### PTR11

```
/* ----- */
/* File Program : PTR11.CPP */
/* Contoh Pointer yang menunjuk ke fungsi yang memiliki parameter)*/
/* Tujuan : memahami operasi pointer yang menunjuk ke fungsi. Melakukan
offset terhadap tabel tergantung fungsi f */
/* ----- */
```

```
#include <stdio.h>
```

#### /\* Prototype \*/

```
/* 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));
```

#### /\* kamus Global \*/

```
int N; /* ukuran efektif */
```

```
#include 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)){
    /* Kamus Lokal */
    int i;

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

```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Beri komentar program 'secukupnya' sehingga bila dibaca kembali akan mudah untuk dipahami!
- 3) Tuliskan hasil pemahaman Anda

#### PTR12

```

/* ----- */
/* File Program : PTR12.CPP */
/* Contoh Pointer yang menunjuk ke Procedure dengan parameter input/output*/
/* Tujuan : memahami operasi pointer yang menunjuk ke prosedur yang memiliki parameter. Melakukan offset terhadap tabel tergantung fungsi f */
/* ----- */

```

```

#include <stdio.h>

```

```

/* Prototype */

```

```

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

```

```

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

```

```

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

#include 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) (void)){
    /* Kamus Lokal */
    int i;

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

```

```
printf("\n")
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Beri komentar program 'secukupnya' sehingga bila dibaca kembali akan mudah untuk dipahami!
- 3) Tuliskan hasil pemahaman Anda

#### PTR13

```
/* ----- */
/* File Program : PTR13.CPP */
/* Contoh Pointer yang menunjuk ke Procedure dengan parameter input/output*/
/* Tujuan : memahami operasi pointer yang menunjuk ke prosedur yang memiliki
parameter. Melakukan offset terhadap tabel tergantung fungsi f */
/* ----- */

#include <stdio.h>

/* Prototype */
/* 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 */

/* Prosedur dengan parameter sebuah fungsi */
void geser (int *TT, void (*f) (void *));
/* print tabel yang belum ketahuan typenya */
void printtab (void *T, enum MyType Ty);

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

#include int main() {
    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)){
    /* Kamus Lokal */
    int i;

    /*Program*/
    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;  
}  
}
```

#### CASE STUDY

- 1) Eksekusi file program tersebut dan pahami maksud program ini
- 2) Beri komentar program 'secukupnya' sehingga bila dibaca kembali akan mudah untuk dipahami!
- 3) Tuliskan hasil pemahaman Anda