

LAPORAN PRAKTIKUM
DASAR PROGRAMMING



1031101 – Dasar Programming

Disusun Oleh:

Nama : Putri Geraldine Alexsandra Sihombing

NIM : 11323009

Diploma III Teknologi Informasi IT DEL 2023

Situoluama, Laguboti 2023

2023/2024

A. Tugas Pendahuluan

1. Apakah yang dimaksud dengan pointer dalam pemrograman bahasa C dan bagaimana deklarasi pointer dalam bahasa C?
 - Pointer adalah variable yang berisi address (alamat) dari variable lain.
 - Dideklarasikan dengan menambahkan * atau & didepan nama atau sebelum nama variabel (int *ptr).
 - Pointer dideklarasikan dengan :
<tipe> * <nama>
 - Operator dalam pointer yaitu:
 - a. * (dereference), mengakses isi alamat yang ditunjuk oleh pointer.
 - b. & (address of), memberikan alamat dari sebuah variable.
 - Saat dideklarasikan, pointer belum menunjukkan ke sebuah alamat tertentu (berisi sebuah alamat acak)
2. Analisi kode

```
#include <stdio.h>
void fun(int x)
{
    x = 30;
}

int main()
{
    int y = 20;
    fun(y);
    printf("%d", y);
    return 0;
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10>gcc kode1.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10>n
20
```

Pada code ini, telah dideklarasikan $x = 30$. Sehingga pada `fun(int x)`, maka x bernilai 30. Selanjutnya diinteger lagi $y = 20$. Pada `fun(y)`, sehingga y bernilai 20.

```
#include <stdio.h>
void fun(int *ptr)
{
    *ptr = 30;
}

int main ()
{
    int y = 20;
    fun(&y);
    printf("%d", y);

    return 0;
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10>gcc kode2.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10>n
30
D:\KULIAH\SEMESTER 1\Daspro\Week 10>|
```

Pada code ini, memakai pointer dimana ditanda dengan adanya tanda * serta tanda &

3. Apakah yang dimaksud double pointer dalam pemrograman bahasa C dan bagaimana deklarasi pointer dalam bahasa C?
 - Double pointer adalah variable yang digunakan untuk menyimpan alamat memori pointer lainnya. Pointer 1, menyimpan alamat variable. Pointer 2 digunakan untuk menyimpan alamat pointer pertama.

B. Praktikum

1. Program 1: simple pointer

```
#include <stdio.h>

//Deklarasikan dan sekaligus inisialisasi variabel tipe integer
int a_var = 10;

//deklarasikan variabel pointer ke variabel tipe integer
int *a_ptr;

int main()
{
    //inisialisasi nilai pointer menunjuk ke alamat a_var
    a_ptr = &a_var; //simbol "&" dibaca address of (alamat dari)

    //akses dan tampilkan alamat memori yang ditunjukkan pointer
    printf("\n Alamat memori yang ditunjuk *a_ptr    = %d", a_ptr);
    printf("\n Alamat memori variabel &a_var        = %d\n", &a_var);

    //akses dan tampilkan nilai variabel: a_var
    printf("\n Akses langsung (a_var)           : a_var = %d", a_var);
    printf("\n Akses tak langsung (*a_ptr)        : a_var = %d\n", *a_ptr);
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>gcc simple_pointer.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>n

Alamat memori yang ditunjuk *a_ptr    = 4210692
Alamat memori variabel &a_var        = 4210692

Akses langsung (a_var)           : a_var = 10
Akses tak langsung (*a_ptr)        : a_var = 10
```

2. Program 2: pointer dan variabel berukuran multi-byte

```
#include <stdio.h>

int main (int argc, char *argv[])
{
    //deklarasikan dan inisialisasi variabel
    int var_int = 123455;
    float var_float = 780.478;
    char var_char = 65; //huruf A

    //deklarasikan dan inisialisasi pointer
    int *ptr_int = &var_int;
    float *ptr_float = &var_float;
    char *ptr_char = &var_char;

    printf("Ukuran char      = %d\n", sizeof(char));
    printf("Ukuran Float     = %d\n", sizeof(float));
    printf("Ukuran Integer    = %d\n", sizeof(int));

    /*printf alamat variabel, alamat menunjuk kepada byte pertama*/

    printf(" Alamat var_char ditunjukkan ptr_char    = %d\n", ptr_char);
    printf(" Alamat var_float ditunjukkan ptr_float    = %d\n", ptr_float);
    printf(" Alamat var_int ditunjukkan ptr_int         = %d\n", ptr_int);
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>gcc PointerAndVarSize.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>n

Ukuran char      = 1
Ukuran Float     = 4
Ukuran Integer    = 4

Alamat var_char ditunjukkan ptr_char    = 6422283
Alamat var_float ditunjukkan ptr_float  = 6422284
Alamat var_int ditunjukkan ptr_int      = 6422288
```

3. Program 3: Double Pointer (Pointer to Pointer)

```
#include <stdio.h>

//Program C untuk mendemonstrasikan pointer ke pointer
int main()
{
    int var = 789;

    // pointer untuk var
    int *ptr2;

    // double pointer untuk ptr 2
    int **ptr1;

    // menyimpan alamat var pada ptr2
    ptr2 = &var;

    // menyimpan alamat ptr2 pada ptr1
    ptr1 = &ptr2;

    // menampilkan nilai var menggunakan pointer tunggal dan ganda
    printf("Value of var = %d\n", var);
    printf("Value of var using single pointer = %d\n", *ptr2);
    printf("Value of var using double pointer = %d\n", **ptr1);

    return 0;
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>gcc double_pointer.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>n
Value of var = 789
Value of var using single pointer = 789
Value of var using double pointer = 789
```

4. Program 4: Void free malloc

```
// program to calculate the sum of n numbers entered by the user
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, *ptr, sum = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    ptr = (int*) malloc(n * sizeof(int));

    // if memory cannot be allocated
    if(ptr == NULL) {
        printf("Error! memory not allocated.");
        exit(0);
    }

    printf("Enter elements: \n");
    for(i = 0; i < n; ++i){
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("sum = %d", sum);

    // deallocating the memory
    free(ptr);

    return 0;
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>gcc void_free_malloc.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>n
Enter number of elements: 5
Enter elements:
5
4
3
2
1
sum = 15
```

5. Program 7: Null Pointer

```
#include <stdio.h>

int main(void){
    int num = 10;
    int *ptr = &num;

    // we can also check with 0 instesd of NULL
    if(ptr == NULL)
        printf("ptr : NULL\n");
    else
        printf("ptr : NOT NULL\n");

    // assigning 0
    ptr = 0;
    if(ptr == NULL)
        printf("ptr : NULL\n");
    else
        printf("ptr : NOT NULL\n");

    return 0;
}
```

```
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>gcc null_pointer.c -o n
D:\KULIAH\SEMESTER 1\Daspro\Week 10\Prak_Daspro_W11S02_11323009>n
ptr : NOT NULL
ptr : NULL
```

C. Analisis

1. What operator is used to determine the address of a variable?
 - Operator yang digunakan untuk alamat variabel adalah & (address of). Namun belum menunjukkan alamat yang spesifik (menunjuk alamat yang acak)
2. What operator is used to determine the value at the location pointed to by a pointer?
 - Operator yang digunakan untuk menunjukkan nilai dari lokasi yang ditunjuk pointer adalah * (dereference).
3. What is a pointer?
 - Pointer adalah variable yang berisi address (alamat) dari variable lain.
 - Pointer digunakan untuk mengkolasikan objek baru, meneruskan fungsi dan melakukan iterasi diatas elemen dadlam array atau struktur data.