

Nama : Muhammad Azhar Alauddin

Kelas : 1A – D4 Teknik Informatika

NIM : 201524013

LATIHAN ALOKASI DINAMIS (POINTER)

1. Pengisian pointer dengan nilai

```
#include <stdio.h>

int main()
{
    int *ptr;
    ptr = 2;
    printf("%d", ptr);
    return 0;
}
```

Final State :
Alamat *ptr berada di 2

Visualisasi Memori :

2	?	<-- *ptr
	?	
&ptr	ptr = 2	

2. Operasi pointer

```
#include <stdio.h>

int main()
{
    int *ptr;
    ptr = 2;
    ptr++;
    printf("%d", ptr);
    return 0;
}
```

Final State :
Alamat *ptr berada di 6

Visualisasi Memori :

	6 ?	<-- *ptr
	?	
&ptr	ptr = 6	

3. Akses pointer ke variabel primitif

```
#include <stdio.h>

int main()
{
    int y, x = 87;
    int *px;

    px = &x;
    y = *px;
    printf("Alamat x = %p\n", &x);
    printf("Isi px = %p\n", px);
    printf("Isi x = %d\n", x);
    printf("Nilai yang ditunjuk oleh px = %d\n", *px);
    printf("Nilai y = %d\n", y);
    return 0;
}
```

Final State :

Alamat x = 000000000062FE10

Isi px = 000000000062FE10

Isi x = 87

Nilai yang ditunjuk oleh px = 87

Nilai y = 87

Visualisasi Memori :

&x	x = 87	<-- *px
&y	y = 87	
&px	px = &x	

4. Pointer vs Assignment biasa

```
#include <stdio.h>

int main()
{
    int x = 6;
    int p = x;
    int q = p;

    p = 10;
    printf("x : %d p : %d, q : %d \n",x,p,q);

    int x1 = 6;
    int *ptrp = &x1;
    int q1 = *ptrp;
    //ubah nilai x
    *ptrp = 10;

    printf("x : %d p : %d, q : %d ",x1,*ptrp,q1);
    return 0;
}
```

Final State :

Saat Menggunakan Assignment Biasa :

x = 6 p = 10, q = 6

Saat Menggunakan Pointer :

x = 10, p = 10, q = 6

Visualisasi Memori :

&q1	q1 = 6	
&x1	x1 = 10	<--*ptrp
&ptrp	ptrp = &x1	
&q	q = 6	
&p	p = 10	
&x	x = 6	

5. Pointer pada Array

```
#include <stdio.h>

int main()
{
    int my_arr[] = {1,23,17,4,-5,100};
    int *ptr;

    int i;
    ptr = &my_arr[0];
    for(i = 0; i < 6; i++){
        printf("my_arr[%d] = %d ",i,my_arr[i]);
        printf("ptr + %d = %d\n",*(ptr+i)); // coba ubah jadi *ptr++ dan *(++ptr)
    }
    return 0;
}
```

Final State :

saat `*(ptr+i)`

```
my_arr[0] = 1 ptr + 0 = 1
my_arr[1] = 23 ptr + 1 = 23
my_arr[2] = 17 ptr + 2 = 17
my_arr[3] = 4 ptr + 3 = 4
my_arr[4] = -5 ptr + 4 = -5
my_arr[5] = 100 ptr + 5 = 100
```

saat `*ptr++`

```
my_arr[0] = 1 ptr + 0 = 1
my_arr[1] = 23 ptr + 1 = 23
my_arr[2] = 17 ptr + 2 = 17
my_arr[3] = 4 ptr + 3 = 4
my_arr[4] = -5 ptr + 4 = -5
my_arr[5] = 100 ptr + 5 = 100
```

saat `*(++ptr)`

```
my_arr[0] = 1 ptr + 0 = 23
my_arr[1] = 23 ptr + 1 = 17
my_arr[2] = 17 ptr + 2 = 4
my_arr[3] = 4 ptr + 3 = -5
my_arr[4] = -5 ptr + 4 = 100
my_arr[5] = 100 ptr + 5 = 90 //pindah alamat, alamatnya berisi nilai 90
```

Visualisasi Memori :

*(ptr++)		
&ptr	ptr = &my_arr[0]	
&my_arr[5]	my_arr[5]	<--*(ptr+5)
&my_arr[4]	my_arr[4]	<--*(ptr+4)
&my_arr[3]	my_arr[3]	<--*(ptr+3)
&my_arr[2]	my_arr[2]	<--*(ptr+2)
&my_arr[1]	my_arr[1]	<--*(ptr+1)
&my_arr[0]	my_arr[0]	<--*(ptr+0)

Keterangan : ***(ptr++)**

*(ptr+i)		
&ptr	ptr = &my_arr[0]	
&my_arr[5]	my_arr[5]	<--*(ptr+5)
&my_arr[4]	my_arr[4]	<--*(ptr+4)
&my_arr[3]	my_arr[3]	<--*(ptr+3)
&my_arr[2]	my_arr[2]	<--*(ptr+2)
&my_arr[1]	my_arr[1]	<--*(ptr+1)
&my_arr[0]	my_arr[0]	<--*(ptr+0)

Keterangan : ***(ptr+i)**

*(++ptr)		
	?	<--*(ptr+6)
&my_arr[5]	my_arr[5]	<--*(ptr+5)
&my_arr[4]	my_arr[4]	<--*(ptr+4)
&my_arr[3]	my_arr[3]	<--*(ptr+3)
&my_arr[2]	my_arr[2]	<--*(ptr+2)
&my_arr[1]	my_arr[1]	<--*(ptr+1)
&my_arr[0]	my_arr[0]	<--*(ptr+0)

Keterangan : ***(++ptr)**

6. Pointer pada Array2

```
#include<stdio.h>

int main(){
    int arr[] = {10,11,12};
    int *p = arr;

    int i;
    for(i = 0; i < 4; i++){
        printf("%d\n",*p++);
    }
    return 0;
}
```

Final State :

pada alamat 6487552, nilai yang ditunjuk p adalah 10

pada alamat 6487556, nilai yang ditunjuk p adalah 11

pada alamat 6487560, nilai yang ditunjuk p adalah 12

Visualisasi Memori :

	?	<--*(ptr+3)
&arr[2]	arr[0] = 8	<--*(ptr+2)
&arr[1]	arr[0] = 9	<--*(ptr+1)
&arr[0]	arr[0] = 10	<--*(ptr+0)
&ptr	ptr = &arr[0]	

PROGRAM MYSTERI

```
/*
Nama Program : Mysteri.cpp
Deskripsi :
Programmer : 1A - 201524013 - Muhammad Azhar Alauddin
Tanggal : 2 April 2021
Versi : 1
*/

#include <stdio.h>
int main()
{
    int *intPtr;
    double *dubPtr;

    //deklarasi var integer dan double
    int int1, int2, int3, int4, int5;
    double dub1, dub2, dub3, dub4, dub5;

    /*Inisialisasi nilai variabel*/
    int1 = 54;
    int2 = 10;
    int3 = 96;
    int4 = 21;
    int5 = 88;
    dub1 = 64.32;
    dub2 = 143.3;
    dub3 = 79.12;
    dub4 = 76.78;
    dub5 = 52.5;

    intPtr = &int1;
    dubPtr = &dub1;

    //Output 1
    printf("Output 1 : Variabel bertipe integer dan double \n\n");
    printf("Tipe integer : \n\n");
    printf("int1 = %d, int2 = %d, int3 = %d, int4 = %d, int5 = %d ", int1, int2,
        int3, int4, int5);

    printf("Tipe double : \n\n");
    printf("dub1 = %g, dub2 = %g, dub3 = %g, dub4 = %g, dub5 = %g ", dub1, dub2,
        dub3, dub4, dub5);
}
```



```

//Menampilkan alamat (dan pointer) dalam hexa
printf("\n\nOutput 2 : Alamat dan variabel pointer bertipe integer : \n");
printf("Alamat memori int1, int2, dan int3 : %x, %x %x", &int1, &int2, &int3
);

printf("\n\n Menampilkan data integer menggunakan pointer (+decrement) : *(i
ntPtr) *(--intPtr) *(--intPtr)\n\n");

printf("--> *intPtr = %d", *intPtr);
printf("\n--> *(--intPtr) = %d", *(--intPtr));
printf("\n--> *(--intPtr) = %d", *(--intPtr));

printf("\n\n Menampilkan data integer menggunakan pointer (+increment) : *(i
ntPtr) *(++intPtr) *(++intPtr)\n\n");
printf("--> *intPtr = %d", *intPtr);
printf("\n--> *(++intPtr) = %d", *(++intPtr));
printf("\n--> *(++intPtr) = %d", *(++intPtr));

//Pertanyaan : alamat &int4 dan &int5 ?
//Output 3 : Alamat dan variabel pointer bertipe double
printf("\n\nOUTPUT 3 : Alamat dan Variabel Pointer bertipe double ");
printf("\nAlamat Memori : dub3 = %x , dub4 = %x , dub5 = %x ", &dub3, &dub4,
&dub5);

dubPtr = &dub5;

printf("\n\nMenampilkan data double menggunakan pointer (+increment) : *(dub
Ptr), *(++dubPtr), *(++dubPtr)");
printf("\n*(dubPtr) = %g, *(++dubPtr) = %g, *(++dubPtr) = %g", *(dubPtr), *(
++dubPtr), *(++dubPtr));

printf("\n\nMenampilkan data double menggunakan pointer (+decrement) : *(dub
Ptr), *(--dubPtr), *(--dubPtr)");
printf("\n*(dubPtr) = %g, *(--dubPtr) = %g, *(--
dubPtr) = %g", *(dubPtr), *(--dubPtr), *(--dubPtr));

return 0;
}
Final State :
IntPtr = 62fdfc = &int1
dubPtr = 62fdd0 = &dub5

Alamat &int4 = &int3 - 1
Alamat &int5 = &int4 - 1 = &int3 - 2

```

Visualisasi Memori :

&int1	int1 = 54	<--*intPtr
&int2	int2 = 10	<--*(intPtr-1)
&int3	int3 = 96	<--(*intPtr-2)
&int4	int4 = 21	<--(*intPtr-3)
&int5	int5 = 88	<--(*intPtr-4)
&dub1	dub1 = 64.32	<--*dubPtr
&dub2	dub2 = 143.3	<--*(dubPtr-1)
&dub3	dub3 = 79.12	<--*(dubPtr-2)
&dub4	dub4 = 76.78	<--*(dubPtr-3)
&dub5	dub5 = 52.5	<--(*dubPtr-4)
&dubPtr	dubPtr = &dub1	
&intPtr	intPtr = &int1	

LESSON LEARNED

Ada banyak sekali pelajaran yang saya bisa dapatkan, setelah mengerjakan tugas *Case Study* Pemahaman Alokasi Dinamis ini. Mungkin yang awalnya hanya tau dan mengerti sebatas teori mengenai pointer dan juga alokasi, tetapi dengan adanya tugas ini saya jadi mengetahui dan mengerti juga bagaimana secara praktisnya dalam implementasi algoritma. Berikut ringkasan dari beberapa hal yang saya baru pahami mengenai materi terkait :

- Operasi pada pointer yang dilakukan berfungsi untuk mengubah alamat memori yang ditunjuk oleh pointer
- Mengetahui perbedaan antara : $*(ptr + i)$, $*(ptr++)$, dan $*(++ptr)$
- Pointer dapat menunjuk nilai yang terdapat pada suatu array dan juga dapat menampilkannya. Dan apabila penunjukan nilai yang ada pada array, maka akan diambil nilai random dari memori...