

MODUL PRAKTIKUM

STRUKTUR DATA

C/C++

Oleh:

Hermawan ST, M.Kom.

Informatika 2023

Uniersitas Trunojoyo Madura

II. VARIABEL, KOLEKSI ARRAY dan KOMPOSISI

Pada Bab 2 ini, peserta perkuliahan akan mempelajari konsep dasar mengenai variabel Primitif, Pointer, Koleksi array dan Komposisi Struct serta menuliskan program C++ dan visualisasi menggunakan IDE QT Creator, serta mengerjakan tugas latihan.

2.1. Variabel dan Tipe Data

Variabel adalah Register pengenalan data (identifier) yang disimpan didalam memori komputer baik didalam memori *Stack* maupun *Heap*. Secara default nilai didalam variable bisa diubah, adapun identifier lain yang tetap adalah Konstanta (const). Penulisan Variabel dapat dilakukan secara global didalam header dan main program sedangkan pada Void dan Function adalah local.

```
#include <stdio.h>

#define PANJANG 10
#define LEBAR 5
#define ENTER '\n'

int main() {
    int area;

    area = PANJANG * LEBAR;
    printf("value of area : %d", area);
    printf("%c", ENTER);

    return 0;
}
```

Gambar 2.1.

```
#include <stdio.h>

int main() {
    const int LENGTH = 10;
    const int WIDTH = 5;
    const char NEWLINE = '\n';
    int area;

    area = LENGTH * WIDTH;
    printf("value of area : %d", area);
    printf("%c", NEWLINE);

    return 0;
}
```

}

Tipe data adalah klasifikasi dan alokasi memori untuk variable sesuai dengan tipe yang ditentukan.

Standar tipe data didalam Bahasa Pemrograman C/C++.

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4bytes for 32 bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

```
bool b=true;
printf("    Size Of bool_BIT    :   %d\n", sizeof((bool)b));
printf("CHAR_BIT      :   %d\n", CHAR_BIT);
printf("    Size Of CHAR_BIT    :   %d\n", sizeof((bool)CHAR_BIT));
printf("CHAR_MIN-MAX    :   %d-%d\n", CHAR_MIN,CHAR_MAX);
printf("    Size Of CHAR_MAX    :   %d\n", sizeof((char)CHAR_MAX));
printf("INT_MIN-MAX      :   %d-%d\n", (int)INT_MIN,(int)INT_MAX);
printf("    Size Of INT_MAX      :   %d\n", sizeof(INT_MAX));
double d;
printf("    Size Of DOUBLE   :   %d\n", sizeof(d));
```

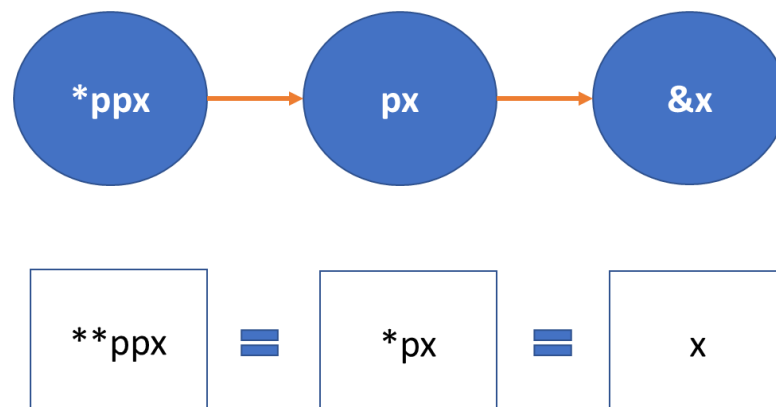
2.2. Pointer

Variabel data didalam memori komputer tidak hanya memiliki nilai tetapi juga referensi alamat memori (Pointer). Pengaksesan data dapat dilakukan melalui akses langsung menggunakan identifier

variable dan juga menggunakan referensi pointer. Jadi Pointer adalah Variabel referensi alamat dari sebuah variable, karena pengaksesan tidak secara langsung maka Pointer bersifat Dinamis dan diletakkan didalam memori Heap.

```
int main()
{
    int x = 1000;
    int *px = &x;
    *px = 0;
    printf("nilai:   %d|%p|%p|%d\n", x, &x, px, *px);
    int **ppx = &px;
    **ppx = 10000;
    printf("nilai ppx   :   %d|%p|%p|%p|%d\n",
x, &x, px, *ppx, **ppx);
    return 0; }
```

- ✓ Jika $\text{int } *px = \&x$, dimana $px = \&x$, $*px$ adalah dereference x , sehingga $*px = x$, jika nilai $*px$ berubah maka nilai x akan ikut berubah, $*px = 0$ maka menyebabkan nilai $x = 0$;
- ✓ Jika $\text{int } **ppx = \&px$; dimana $*ppx = px = \&x$, jika nilai $**ppx$ berubah maka nilai $x = *px = **ppx$.



2.3. Koleksi Array

Array adalah tipe data koleksi dasar untuk mengumpulkan data dengan tipe yang sama. Data Array terkoleksi dengan terhubung secara fisik pada memori berdasarkan indeks-nya. Inisialisasi Array bersifat static konstan sesuai dengan deklarasi inisialisasi yang ditentukan.

```
int ArrayInt[10]=ArrayInt={0,1,2,3,4,5,6,7,8,9};
```

```

int main()
{
    int ArrayInt[10]={0,1,2,3,4,5,6,7,8,9};
    printf(" Array Int indek ke-2=%d\n", ArrayInt[2]);
    char ArrChar[10]={'0','1','2','3','4','5','6','7','8','9'};
    printf(" Array Char indek ke-9=%d-%c\n",
        ArrChar[9], (char)ArrChar[9]);
    return 0;}

```

Tipe data String adalah tipe data default koleksi Char, pointer *c dapat diberikan nilai string dan juga mengakses char dari string.

```

int main()
{
    string teks ="HELLO STRING";
    char cteks[100]="teks";
    teks.copy(cteks,sizeof(teks),0);
    printf(" Array Char indek ke-0=%c = %c\n",
        cteks[0],teks.at(0));

    char* c = "POINTER CHAR";
    printf(" Array Char pointer ke-5=%c\n", *(c+4));
    c=&teks.at(0);
    printf(" Array Char pointer %c\n", *(c+4));
    return 0;}

```

2.4. Instruksi Praktiku

1. Buat Program sesuai dengan tutorial
2. Lakukan pengujian script sesuai dengan tutorial

2.5. Tugas Praktikum

1. Jika terdapat program sebagaimana berikut, buatlah pembenaran program menggunakan pointer.

```

void replace(char* c, std::string text)
{
    .....lengkapi function untuk bisa ubah teks menjadi
    "Millo World"
}

int main()
{
    string teks = "Hello World";
    char* c = &teks.at(0);
    replace(c,string text);
    cout<<"print:"<<teks<<endl;
    return 0;
}

```

2. Jika
terdapat
template
masukan

sebagaimana from Window berikut, lakukan perubahan teks pada input template berdasarkan ketika “Change”

