

MINGGU 1

Array & Pointer

DESKRIPSI TEMA

Pada minggu ini, mahasiswa akan belajar mengenai array, baik itu array 1 dimensi dan 2 dimensi, juga penggunaan pointer, baik pada variabel dasar, array, maupun sebagai function parameter.

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

1. Mahasiswa mampu menerapkan array dalam membuat program dalam Bahasa C.
2. Mahasiswa mampu menerapkan pointer dalam membuat program dalam Bahasa C.

PENUNJANG PRAKTIKUM

1. Software CodeBlocks

LANGKAH-LANGKAH PRAKTIKUM

A. Array

Array adalah sebuah tipe data yang digunakan untuk menyimpan data dengan tipe data yang sama dalam jumlah yang banyak dalam satu nama (variabel).

- Tutorial 1.0 – 1 Dimensional Array Part 1
 1. Buatlah sebuah file dengan nama Wo1_NIM_1DArray_1.c
 2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
#include <stdio.h>

int main(){

    // Menentukan nilai pada array saat pembuatan
    int randomNumber[5] = {10, 24, 234, 345, 234};
    int i; // Variabel untuk melakukan iterasi

    printf("Isi awal array\n");
    for (i=0; i<5; i++){
        printf("randomNumber[%d] = %d\n", i, randomNumber[i]);
    }

    printf("\n");
    randomNumber[3] = 10000;

    printf("Isi array setelah berubah\n");
    for (i=0; i<5; i++){
        printf("randomNumber[%d] = %d\n", i, randomNumber[i]);
    }

    return 0;
}
```

- Tutorial 1.1 – 1 Dimensional Array Part 2
 1. Buatlah sebuah file dengan nama Wo1_NIM_1DArray_2.c
 2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
#include <stdio.h>

int main(){

    int randomNumber[5] = {10, 24, 234, 345, 234};
    int i;

    printf("Isi awal array\n");
    for (i=0; i<6; i++){
        printf("randomNumber[%d] = %d\n", i, randomNumber[i]);
    }

    return 0;
}
```

- Tutorial 1.2 – Passing 1 Dimensional Array to Function
 1. Buatlah sebuah file dengan nama Wo1_NIM_1DArrayToFunc.c
 2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
1  #include <stdio.h>
2
3  int sum(int recievedArr[]){
4      int i, result=0;
5      for(i=0; i<5; i++){
6          result += recievedArr[i];
7      }
8      return result;
9  }
10
11 int main(){
12     int randomNumber[] = {10, 24, 234, 345, 234};
13     int i;
14
15     // Jika hasil penjumlahan mau disimpan ke dalam
16     // Variabel
17     int total;
18     total = sum(randomNumber);
19     printf("Hasil penjumlahan = %d\n", total);
20
21     // Jika hasil penjumlahan hanya akan ditampilkan
22     printf("Hasil penjumlahan = %d\n", sum(randomNumber));
23
24     return 0;
25 }
```

- Tutorial 1.3 – 2 Dimensional Array
 1. Buatlah sebuah file dengan nama Wo1_NIM_2DArray.c
 2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
1  #include <stdio.h>
2
3  void printArray(int arr[][3]){
4      // Indeks untuk baris dan kolom
5      int row, col;
6
7      printf("Isi dari array:\n");
8      for(row=0; row<3; row++){
9          for(col=0; col<3; col++){
10             printf("%d ", arr[row][col]);
11         }
12         printf("\n");
13     }
14     printf("\n");
15 }
16
17 int main(){
18     // Inisialisasi langsung
19     int randomNumbers[3][3] =
20     {
21         {1, 2, 3},
22         {6, 5, 4},
23         {7, 8, 9}
24     };
25     printArray(randomNumbers);
26
27     return 0;
28 }
```

*Catatan: terdapat dua cara dalam melempar sebuah variabel ke sebuah function, yaitu call by value dan call by reference. Data yang dikirimkan, pada Tutorial 1.2 dan Tutorial 1.3, merupakan reference dari array.

B. Pointer

Pointer adalah sebuah variabel yang isinya merupakan **alamat memori**. Memori yang dimaksud adalah RAM (Random Access Memory) pada komputer. Pointer menggunakan dua simbol dalam pengoperasiannya, yaitu ampersand (&) untuk merepresentasikan alamat dan operand perkalian (*) untuk melihat isi alamat tersebut.

- Tutorial 2.0 – Basic Pointer

1. Buatlah sebuah file dengan nama Wo1_NIM_BasicPointer.c
2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
1  #include <stdio.h>
2
3  int main(){
4      int number = 14026;
5
6      printf("Content of number = %d\n", number);
7      printf("Address of number = %p\n", &number);
8
9      return 0;
10 }
```

- Tutorial 2.1 –Pointer in Array

1. Buatlah sebuah file dengan nama Wo1_NIM_BasicPointerArr1.c
2. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
#include <stdio.h>

int main(){
    int randomNumber[] = {10, 24, 234, 345, 234};

    printf("randomNumber = %p\n", randomNumber);
    printf("&randomNumber = %p\n", &randomNumber);
    printf("&randomNumber[0] = %p\n\n", &randomNumber[0]);

    // Pointer dalam array menunjuk ke array ke 0
    printf("randomNumber = %d\n", randomNumber[0]);
    printf("*randomNumber = %d\n\n", *randomNumber);

    // Pointer menunjuk ke dalam array index ke 2
    // dengan menjumlahkan sesuai index
    printf("randomNumber[2] = %d\n", randomNumber[2]);
    printf("*(randomNumber+2) = %d\n\n", *(randomNumber+2));

    // Bagaimana dengan alamat array index ke 2?
    printf("&randomNumber[2] = %p\n", &randomNumber[2]);
    // Apa yang berbeda dengan &randomNumber?

    return 0;
}
```

- Tutorial 2.2 – Pointer to function

1. Pengiriman sebuah *value* ke parameter *function* biasanya dilakukan dengan cara *call by value*, di mana *value* hanya diproses di *receiver function* dan nilainya tidak berubah pada sender function. Penggunaan Pointer mengubah pengiriman *value* ke parameter menjadi *call by reference*. *Value* yang diterima *receiver function* hanyalah alamat, sehingga nilai sebenarnya yang dikirimkan terdapat pada variabel saat pengiriman *value*.
2. Buatlah sebuah *file* dengan nama Wo1_NIM_PointerArr.c
3. Ketik ulang *code* di bawah ini, jalankan, dan perhatikan hasilnya.

```
#include <stdio.h>

int hitungJumlah(int *jumlah, int numbers[]){
    int i=0;
    for(i=0; i<5; i++){
        *jumlah += numbers[i];
    }
}

int main(){
    int randomNumber[5] = {1, 2, 3, 4};
    int jumlah = 0;

    hitungJumlah(&jumlah, randomNumber);
    printf("Jumlah = %d\n", jumlah);

    return 0;
}
```

4. Lihat bahwa penjumlahan terjadi di *function* hitungJumlah menggunakan **total*, namun nilai variabel jumlah pada *main* juga berubah.

C. Tugas

Tugas harus menggunakan array dan pointer!

1. Buatlah sebuah program untuk uppercase, lowercase, dan switching sebuah kalimat.

- a. Menu utama

```
1. Uppercase
2. Lowercase
3. Switching
4. Keluar
Pilihan :
```

- b. Tampilan menu jika opsi 1 dipilih. Pada opsi ini, kalimat menjadi huruf kapital, sementara huruf kapital tidak berubah.

```
1. Uppercase
2. Lowercase
3. Switching
4. Keluar
Pilihan : 1

Masukkan kata: Algoritma dan Struktur Data
ALGORITMA DAN STRUKTUR DATA
```

- c. Tampilan menu jika opsi 2 dipilih. Pada opsi ini, huruf kapital menjadi huruf kecil, sementara huruf kecil tidak berubah.

```
1. Uppercase
2. Lowercase
3. Switching
4. Keluar
Pilihan : 2

Masukkan kata: ALGORitmA dan STRuktur Data
algoritma dan struktur data
```

- d. Tampilan menu jika opsi 3 dipilih. Pada opsi ini, huruf kapital menjadi huruf kecil, sementara huruf kecil menjadi huruf kapital.

```
1. Uppercase
2. Lowercase
3. Switching
4. Keluar
Pilihan : 3

Masukkan kata: Algoritma dan Struktur Data
aLGORITMA DAN sTRUKTUR dATA
```

- e. Simpanlah tugas dengan nama Wo1_NIM_T1.c

2. Buatlah sebuah program untuk melakukan penjumlahan, perkalian, dan transpose dari dua buah matrix berukuran 3x3.

- a. Program pada awalnya akan meminta kedua matrix tersebut

```
Masukkan Matrix Pertama(3 x 3)
1 2 3
2 2 1
1 3 4

Masukkan Matrix Kedua(3 x 3)
1 2 1
2 2 2
2 1 2_
```

- b. Setelah itu, akan terdapat menu

```
1. Perkalian Dua Buah Matrix
2. Penjumlahan Dua Buah Matrix
3. Transpose Dua Buah Matrix
4. Keluar
Pilihan: _
```

- c. Gambar di bawah adalah jika menu 1 dipilih

```
1. Perkalian Dua Buah Matrix
2. Penjumlahan Dua Buah Matrix
3. Transpose Dua Buah Matrix
4. Keluar
Pilihan: 1
11 9 11
8 9 8
15 12 15
```

- d. Gambar di bawah adalah jika menu 2 dipilih

```
1. Perkalian Dua Buah Matrix
2. Penjumlahan Dua Buah Matrix
3. Transpose Dua Buah Matrix
4. Keluar
Pilihan: 2
2 4 4
4 4 3
3 4 6
```

- e. Gambar di bawah adalah jika menu 3 dipilih

```
1. Perkalian Dua Buah Matrix
2. Penjumlahan Dua Buah Matrix
3. Transpose Dua Buah Matrix
4. Keluar
Pilihan: 3
Transpose Matrix Pertama
1 2 1
2 2 3
3 1 4
Transpose Matrix Kedua
1 2 2
2 2 1
1 2 2
```


- f. Simpanlah tugas dengan nama Wo1_NIM_T2.c

REFERENSI