LAPORAN PRAKTIKUM

MODUL I
ARRAY

Disusun oleh:
Syafanida Khakiki
NIM: 2311102005

Dosen Pengampu:
Muhammad Afrizal Amrustian, S. Kom., M. Kom

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2023

BAB I

TUJUAN PRAKTIKUM

1. Mahasiswa dapat memahami konsep Array.
2. Mahasiswa dapat mengetahui jenis dimensi Array dan cara penulisannya.
3. Mahasiswa dapat mengimplementasikan Array pada kode program yang dibuat.

BAB II

DASAR TEORI

Array merupakan sekumpulan data yeng memiliki nama sama dan memiliki tipe data yang sama. Array memudahkan kita agar tidak perlu menuliskan variabel secara berulang. Variabel dalam array dibedakan berdasarkan nomor indeks yang dimulai dari 0.

1. Deklarasi Array

Elemen dari array dapat diakses langsung jika dan hanya jika indeks terdefinisi Cara-mengacu sebuah elemen:

```
// Kamus
      int TabJumlahHari[12]; // indeks 0..11
      float TabNilai[l5]; // indeks 0..14
      char TabHuruf[100]; // indeks 0..99
      string TabKatall0071; // indeks 0..99
      Point TabTitik[20]; // indeks 0..19
      int TabJumlahHari[12]; // indeks 0..11
// Algoritma
....
```

Contoh:

```
TabInt [2]
TabInt[i] // jika i terdefinisi
```

2. Mengisi Array

Mengisi array merupakan aktifitas memberi nilai elemen
Contoh : Pemberian nilai pada deklarasi array :

```
char Daftarlururf [b] = {'a'',''b',''c',''d" ,"e"};
```

Pemberian nilai dalam loop :

```
TabInt[0]=31;
for (i=0;i<10;i++)
{

TabInt[i]=1*10;
}
```

3. Mengisi Dan Membaca Isi Array

Elemen array yang telah diberi nilai dapat diakses kembali.

Contoh :

```
#include <iostream>
using namespace std;

      int main ()
      { // Kamus
      int TabInt([10]; int i;
      // Algoritma mengisi array
      for (i=0; i<10; i++) {
      TabInt [1]=1*10;
}
// Algoritma membaca dan menuliskan // isi array ke layar
      for (i=0; 1<10; i++) {
      cout << TabInt[i] << endl;
      }

return 0;
}
```

Berikut ini adalah beberapa jenis array :

1) Array Satu Dimensi

Array satu dimensi adalah tipe variabel yang terdiri dari kumpulan data dengan tipe yang sama yang disusun dalam satu baris atau satu dimensi. Setiap elemen di dalam array memiliki sebuah indeks atau nomor yang digunakan untuk mengakses elemen tersebut. Indeks dimulai dari 0 dan berakhir pada jumlah elemen dikurangi satu.

Contoh :

```
#include <iostream>
using namespace std;int
main() {
    int arr[5] = {9, 3, 5, 2, 1}; //deklarasi arraycout<<
    arr[1] << endl;
    cout<< arr[4];}
```

Output :

```
3
```

```
1
```

2) Array Dua Dimensi

Array dua dimensi adalah variable yang terdiri dari kumpulan array satu dimensi dengan tipe yang sama yang disusun dalam baris dan kolom. Dalam array dua dimensi, setiap elemen memiliki dua indeks, yaitu indeks baris dan indeks kolom. Indeks baris menunjukkan posisi elemen dalam baris, sementara indeks kolom menunjukkan posisi elemen dalam kolom.

Contoh :

```
#include <iostream>
using namespace std;int
main() {
   int arr[2][2] = {{3, 2}, {2, 5}};
   for (int i=0; i<2; i++) { //baris for(int j=0;
     j<2; j++) { //kolom
        cout<< arr[i][j] << ends;
     };
     cout << endl;
   };
}
```

Output :

```
3 2
2 5
```

3) Array Multidimensi

Array multidimensi memiliki kesamaan dengan array satu dimensi dan dua dimensi, namun memiliki kapasitas memori yang lebih besar. Array ini digunakan untuk merepresentasikan array dengan dimensi lebih dari dua atau array yang memiliki lebih dari dua indeks, seperti array tiga dimensi, array empat dimensi, array lima dimensi, dan seterusnya.

Contoh :

```
#include <iostream>
using namespace std;int
main() {
   int arr[2][2][3] = {{{2, 8, 7}, {6, 5, 1}}, {{8,
5, 2}, {9, 2 ,7}}};
   for (int i=0; i<2; i++) {
     for(int j=0; j<2; j++) {
        for(int k=0; k<3; k++) { cout<<
          arr[i][j][k] << ends;
        };
        cout<< endl;
     };
     cout<< endl;
   };
```

Output :

```
2 8 7
6 5 1

8 5 2
9 2 7
```

4) Array Empat Dimensi

Contoh :

```
int arr [3][2][4][4];
```

5) Array Lima Dimensi

Contoh :

```
int arr [2][4][4][3][3];
```

BAB III

GUIDED

1. Guided 1

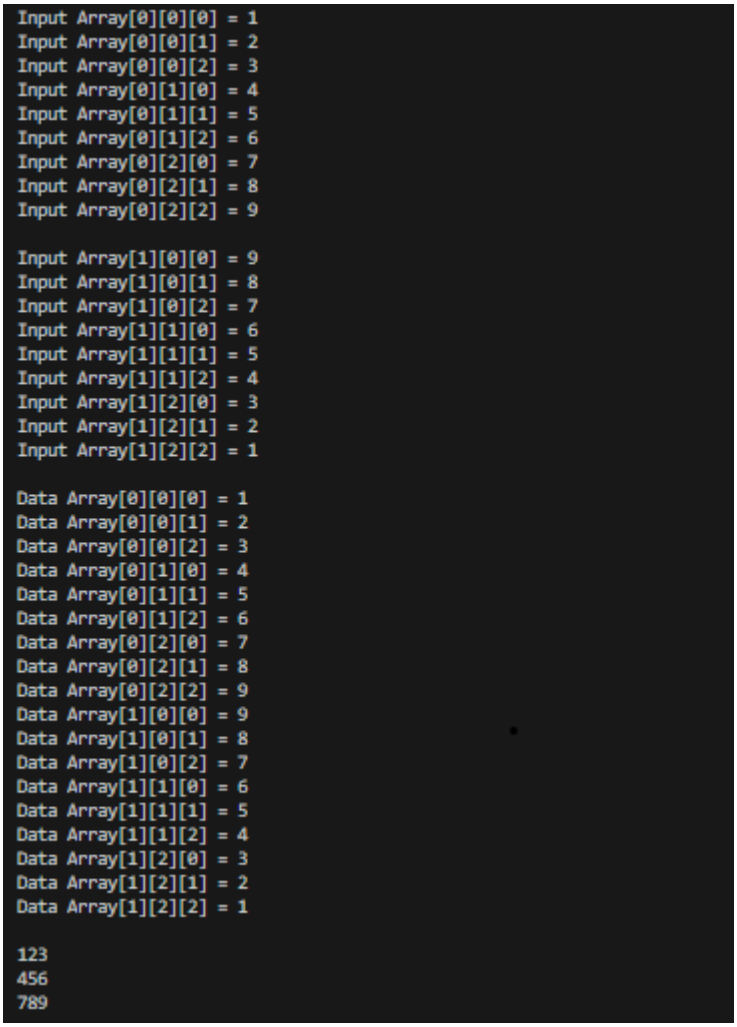
Source code :

Program Input Data Array Tiga Dimensi

```
#include <iostream>
using namespace std;

int main()
{

// Deklarasi array
int arr[2][3][3];
// Input elemen
for (int x = 0; x < 2; x++)
{
    for (int y = 0; y < 3; y++)
    {
        for (int z = 0; z < 3; z++)
        {
        cout << "Input Array[" << x << "][" << y << "][" << z <<"] = ";
        cin >> arr[x][y][z];
        }
    }
    cout << endl;
}
// Output Array
for (int x = 0; x < 2; x++)
{
    for (int y = 0; y < 3; y++)
    {
        for (int z = 0; z < 3; z++)
        {
        cout << "Data Array[" << x << "][" << y << "][" << z <<"] = "
       << arr[x][y][z] << endl;
        }
    }
}
```

```
cout << endl;
// Tampilan array
for (int x = 0; x < 2; x++)
{
    for (int y = 0; y < 3; y++)
    {
        for (int z = 0; z < 3; z++)
        {
        cout << arr[x][y][z] << ends;
        }
        cout << endl;
        }
    cout << endl;
    }

return 0;
}
```

Output :



Deskripsi program :

Program ini menggunakan array 3 dimensi untuk menyimpan dan menampilkan bilangan integer.

1. **Deklarasi Array**

   Program mendeklarasikan array bernama arr dengan 3 dimensi: 2 (lapisan), 3 (baris), dan 3 (kolom). Ini berarti arr dapat menyimpan 2 x 3 x 3 = 18 bilangan integer.

2. **Input Elemen**

   Program menggunakan tiga loop for bersarang untuk meminta pengguna memasukkan nilai untuk setiap elemen array. Loop terluar iterasi melalui 2

lapisan, loop tengah iterasi melalui 3 baris dalam setiap lapisan, dan loop dalam iterasi melalui 3 kolom dalam setiap baris.

- Saat program meminta input, Anda akan melihat format Input Array[x][y][z] =, di mana x, y, dan z mewakili indeks lapisan, baris, dan kolom saat ini.

3. **Output Detail Array**

   Setelah input selesai, program menggunakan tiga loop for bersarang lagi untuk menampilkan isi array. Kali ini, program mencetak tidak hanya nilai elemen, tetapi juga indeksnya (lapisan, baris, dan kolom) menggunakan format Data Array[x][y][z] = value.

4. **Output Padat Array**

   Terakhir, program menggunakan tiga loop for bersarang sekali lagi untuk menampilkan isi array dalam format yang lebih ringkas. Kali ini, hanya nilai elemen yang dicetak, disusun dalam baris dan kolom, tanpa informasi indeks.

   - Program menggunakan ends manipulator untuk menghilangkan karakter baris baru setelah setiap elemen, sehingga menghasilkan output seperti tabel.

2. Guided 2

Source Code :
Program Mencari Nilai Maksimal pada Array

```
#include <iostream>
using namespace std;
int main()

{
    int maks, a, i = 1, lokasi;
    cout << "Masukkan panjang array: ";
    cin >> a;
    int array[a];
    cout << "Masukkan " << a << " angka\n";
    for (i = 0; i < a; i++)
    {
        cout << "Array ke-" << (i) << ": ";
        cin >> array[i];
    }
    maks = array[0];
    for (i = 0; i < a; i++)
```

```
    {
        if (array[i] > maks)
        {
            maks = array[i];
            lokasi = i;
        }
    }
    cout << "Nilai maksimum adalah " << maks << " berada di Array ke " << lokasi
<< endl;
return 0;
```

Output :

```
Masukkan panjang array: 5
Masukkan 5 angka
Array ke-0: 1
Array ke-1: 4
Array ke-2: 7
Array ke-3: 3
Array ke-4: 2
Nilai maksimum adalah 7 berada di Array ke 2
```

Deskripsi Program :

Program ini mencari nilai maksimum dalam sebuah array dan menampilkan lokasinya.

1. Deklarasi Variabel:

- maks: Menyimpan nilai maksimum yang ditemukan dalam array.
- a: Menyimpan panjang array yang diinputkan pengguna.
- i: Digunakan sebagai loop counter.
- lokasi: Menyimpan lokasi (indeks) elemen array dengan nilai maksimum.

2. Input Data:

- Pengguna diminta untuk memasukkan panjang array.
- Pengguna kemudian diminta untuk memasukkan nilai untuk setiap elemen array.

3. Pencarian Nilai Maksimum:

- Loop for digunakan untuk iterasi melalui setiap elemen array.
- Di dalam loop, nilai elemen dibandingkan dengan maks.

- Jika nilai elemen lebih besar dari maks, maka nilai elemen tersebut menjadi nilai maksimum baru dan lokasinya disimpan dalam lokasi.

4. Output Hasil:

- Nilai maksimum dan lokasinya (indeks) dicetak ke layar.

BAB IV
UNGUIDED

1. Unguided 1

Source code :

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int daftar_bilangan[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int bilangan_ganjil[10], bilangan_genap[10];
    int indeks_ganjil = 0, indeks_genap = 0;

    cout << "Data Array:" << endl;
    for (int i = 0; i < 10; i++)
    {
        cout << setw(3) << daftar_bilangan[i];
        if (i < 9)
        {
            cout << ", ";
        }
    }
    cout << endl
         << endl;

    for (int i = 0; i < 10; i++)
    {
        if (daftar_bilangan[i] % 2 != 0)
        {
            bilangan_ganjil[indeks_ganjil++] = daftar_bilangan[i];
        }
        else
        {
            bilangan_genap[indeks_genap++] = daftar_bilangan[i];
        }
    }

    cout << "Nomor genap :" << endl;
```

```
    for (int i = 0; i < indeks_genap; i++)
    {
        cout << setw(3) << bilangan_genap[i];
        if (i < indeks_genap - 1)
        {
            cout << ", ";
        }
    }

    cout << endl
         << endl;

    cout << "Nomor ganjil :" << endl;
    for (int i = 0; i < indeks_ganjil; i++)
    {
        cout << setw(3) << bilangan_ganjil[i];
        if (i < indeks_ganjil - 1)
        {
            cout << ", ";
        }
    }
    cout << endl;

    return 0;
}
```

Output :

```
Data Array:
  1,   2,   3,   4,   5,   6,   7,   8,   9,  10

Nomor genap :
  2,   4,   6,   8,  10

Nomor ganjil :
  1,   3,   5,   7,   9
```

Deskripsi program :

Program ini memisahkan bilangan genap dan ganjil dari sebuah array yang sudah didefinisikan sebelumnya. Berikut penjelasannya:

1. Deklarasi Array:

- daftar_bilangan: Array berisi 10 bilangan integer yang sudah diinisialisasi (1 sampai 10).
- bilangan_ganjil: Array kosong untuk menyimpan bilangan ganjil yang akan dipisahkan.
- bilangan_genap: Array kosong untuk menyimpan bilangan genap yang akan dipisahkan.
- indeks_ganjil: Variabel untuk menyimpan indeks elemen array bilangan_ganjil (awalnya 0).
- indeks_genap: Variabel untuk menyimpan indeks elemen array bilangan_genap (awalnya 0).

2. Menampilkan Array Awal:

- Program mencetak isi daftar_bilangan dengan format kolom menggunakan setw(3).
- Setiap elemen dipisahkan dengan koma (", ") kecuali elemen terakhir.

3. Pemisahan Bilangan Genap dan Ganjil:

- Loop for iterasi melalui setiap elemen daftar_bilangan.
- Kondisi if (daftar_bilangan[i] % 2 != 0) memeriksa apakah elemen tersebut ganjil (sisa bagi pembagian 2 tidak sama dengan 0).
    - Jika ganjil, elemen ditambahkan ke bilangan_ganjil pada indeks indeks_ganjil yang kemudian ditambah 1.
    - Jika genap, elemen ditambahkan ke bilangan_genap pada indeks indeks_genap yang kemudian ditambah 1.

4. Menampilkan Hasil Pemisahan:

- Program mencetak judul "Nomor Genap :".
- Loop for iterasi melalui elemen bilangan_genap hingga indeks indeks_genap.
    - Setiap elemen dicetak dengan format kolom menggunakan setw(3).
    - Elemen dipisahkan dengan koma (", ") kecuali elemen terakhir.
- Program mencetak baris kosong.
- Program mencetak judul "Nomor Ganjil :".
- Loop for iterasi melalui elemen bilangan_ganjil hingga indeks indeks_ganjil.
    - Setiap elemen dicetak dengan format kolom menggunakan setw(3).
    - Elemen dipisahkan dengan koma (", ") kecuali elemen terakhir.

2. **Unguided 2**

Source Code :

```
      #include <iostream>
#include <iomanip> // for setw manipulator
using namespace std;

int main()
{
    int jumlahElemen;

    cout << "Masukkan jumlah elemen array (1-100): ";
    cin >> jumlahElemen;

    while (jumlahElemen < 1 || jumlahElemen > 100)
    {
        cout << "Jumlah elemen tidak valid! Masukkan kembali (1-100): ";
        cin >> jumlahElemen;
    }

    int daftar_bilangan[jumlahElemen];
    int bilangan_ganjil[jumlahElemen], bilangan_genap[jumlahElemen];
    int indeks_ganjil = 0, indeks_genap = 0;

    cout << "Masukkan " << jumlahElemen << " bilangan bulat: " << endl;
    for (int i = 0; i < jumlahElemen; i++)
    {
        cout << "Bilangan ke-" << i + 1 << ": ";
        cin >> daftar_bilangan[i];
    }

    for (int i = 0; i < jumlahElemen; i++)
    {
        if (daftar_bilangan[i] % 2 != 0)
        {
            bilangan_ganjil[indeks_ganjil++] = daftar_bilangan[i];
        }
        else
        {
            bilangan_genap[indeks_genap++] = daftar_bilangan[i];
        }
    }
```

```
    cout << "Nomor Genap :" << endl;
    for (int i = 0; i < indeks_genap; i++)
    {
        cout << setw(3) << bilangan_genap[i];
        if (i < indeks_genap - 1)
        {
            cout << ", ";
        }
    }
    cout << endl
         << endl;

    cout << "Nomor Ganjil :" << endl;
    for (int i = 0; i < indeks_ganjil; i++)
    {
        cout << setw(3) << bilangan_ganjil[i];
        if (i < indeks_ganjil - 1)
        {
            cout << ", ";
        }
    }
    cout << endl;

    return 0;
}
```

Output :

```
Masukkan jumlah elemen array (1-100): 5
Masukkan 5 bilangan bulat:
Bilangan ke-1: 4
Bilangan ke-2: 2
Bilangan ke-3: 5
Bilangan ke-4: 7
Bilangan ke-5: 2
Nomor Genap :
  4,   2,   2

Nomor Ganjil :
  5,   7
```

Deskripsi Program :

Program ini memisahkan bilangan genap dan ganjil dari sebuah array dengan jumlah elemen yang ditentukan oleh pengguna.

1. Mendapatkan Jumlah Elemen:

- Pengguna diminta untuk memasukkan jumlah elemen array (1-100).
- Input divalidasi dengan loop while untuk memastikan nilainya dalam rentang yang valid.

2. Deklarasi Array:

- daftar_bilangan: Array dinamis dengan jumlahElemen sebagai ukurannya untuk menyimpan bilangan integer yang diinputkan.
- bilangan_ganjil: Array dinamis dengan jumlahElemen sebagai ukurannya untuk menyimpan bilangan ganjil yang dipisahkan.
- bilangan_genap: Array dinamis dengan jumlahElemen sebagai ukurannya untuk menyimpan bilangan genap yang dipisahkan.
- indeks_ganjil: Variabel untuk menyimpan indeks elemen array bilangan_ganjil (awalnya 0).
- indeks_genap: Variabel untuk menyimpan indeks elemen array bilangan_genap (awalnya 0).

3. Input Bilangan:

- Program meminta pengguna untuk memasukkan jumlahElemen bilangan bulat.
- Setiap bilangan disimpan dalam daftar_bilangan.

4. Pemisahan Bilangan Genap dan Ganjil:

- Loop for iterasi melalui setiap elemen daftar_bilangan.
- Kondisi if (daftar_bilangan[i] % 2 != 0) memeriksa apakah elemen tersebut ganjil (sisa bagi pembagian 2 tidak sama dengan 0).
    - Jika ganjil, elemen ditambahkan ke bilangan_ganjil pada indeks indeks_ganjil yang kemudian ditambah 1.
    - Jika genap, elemen ditambahkan ke bilangan_genap pada indeks indeks_genap yang kemudian ditambah 1.

5. Menampilkan Hasil Pemisahan:

- Program mencetak judul "Nomor Genap :".
- Loop for iterasi melalui elemen bilangan_genap hingga indeks indeks_genap.
    - Setiap elemen dicetak dengan format kolom menggunakan setw(3).

    - Elemen dipisahkan dengan koma (", ") kecuali elemen terakhir.
- Program mencetak baris kosong.
- Program mencetak judul "Nomor Ganjil :".
- Loop for iterasi melalui elemen bilangan_ganjil hingga indeks indeks_ganjil.
    - Setiap elemen dicetak dengan format kolom menggunakan setw(3).
    - Elemen dipisahkan dengan koma (", ") kecuali elemen terakhir.

3. Unguided 3

Source Code :

```
            #include <iostream>
using namespace std;

int main()
{
    int pilih;
    int daftar_bil[10];
    int total = 0;
    int bilangan;
    int n;
    int maksimum = daftar_bil[0];
    int minimum = daftar_bil[0];

    cout << "--------MENU-------  " << endl;
    cout << " " << endl;
    cout << "1. Cari Nilai Minimum" << endl;
    cout << "2. Cari Nilai Maksimum" << endl;
    cout << "3. Hitung Rata-rata" << endl;
    cout << " " << endl;
    cout << "Masukkan pilihan (1-3): ";
    cin >> pilih;
    cout << " " << endl;

    switch (pilih)
    {

    case 1:
        cout << "Cari Nilai Minimum" << endl;

        cout << "Masukkan panjang Array : ";
        cin >> n;

        for (int i = 0; i < n; i++)
        {
            cout << " Masukkan Bilangan Bulat ke-" << i + 1 << " : ";
```

```
            cin >> daftar_bil[i];
            if (daftar_bil[i] < minimum)
            {
                minimum = daftar_bil[i];
            }
        }
        cout << "Nilai Minimum: " << minimum << endl;
        break;
    case 2:
        cout << "Cari Nilai Maksimum" << endl;
        cout << "Masukkan panjang Array : ";
        cin >> n;

        for (int i = 0; i < n; i++)
        {
            cout << " Masukkan Bilangan Bulat ke-" << i + 1 << " : ";
            cin >> daftar_bil[i];

            if (daftar_bil[i] > maksimum)
            {
                maksimum = daftar_bil[i];
            }
        }
        cout << "Nilai Maksimum: " << maksimum << endl;
        break;
    case 3:
        cout << "Hitung Rata-rata" << endl;
        for (int i = 0; i < 10; i++)
        {
            cout << " Masukkan Bilangan Bulat ke-" << i + 1 << " : ";
            cin >> daftar_bil[i];
            total += daftar_bil[i];
        }
        float rata_rata = static_cast<float>(total) / 10;
        cout << "Rata-rata: " << rata_rata << endl;
        break;
    }

    return 0;
}
```

Output :

```
--------MENU-------

1. Cari Nilai Minimum
2. Cari Nilai Maksimum
3. Hitung Rata-rata

Masukkan pilihan (1-3): 1

Cari Nilai Minimum
Masukkan panjang Array : 4
 Masukkan Bilangan Bulat ke-1 : 5
 Masukkan Bilangan Bulat ke-2 : 7
 Masukkan Bilangan Bulat ke-3 : 14
 Masukkan Bilangan Bulat ke-4 : 8
Nilai Minimum: 5
```

Deskripsi Program :

Program ini menyediakan menu interaktif untuk melakukan operasi dasar pada array bilangan bulat:

1. Mencari Nilai Minimum:

- Pengguna memasukkan panjang array (n).
- Loop for digunakan untuk memasukkan nilai ke dalam array daftar_bil.
- Nilai minimum dihitung dengan membandingkan setiap elemen array dengan minimum dan memperbarui minimum jika menemukan nilai yang lebih kecil.
- Nilai minimum dicetak ke layar.

2. Mencari Nilai Maksimum:

- Pengguna memasukkan panjang array (n).
- Loop for digunakan untuk memasukkan nilai ke dalam array daftar_bil.
- Nilai maksimum dihitung dengan membandingkan setiap elemen array dengan maksimum dan memperbarui maksimum jika menemukan nilai yang lebih besar.
- Nilai maksimum dicetak ke layar.

3. Menghitung Rata-rata:

- Loop for digunakan untuk memasukkan 10 nilai ke dalam array daftar_bil.

- Total nilai dihitung dengan menjumlahkan semua elemen array.
- Rata-rata dihitung dengan membagi total nilai dengan 10.
- Nilai rata-rata dicetak ke layar.

BAB IV

KESIMPULAN

Array merupakan struktur data yang digunakan untuk menyimpan sekumpulan data dengan tipe yang sama secara berurutan dalam memori komputer.

Karakteristik Array:

- **Tipe Data:** Elemen array harus memiliki tipe data yang sama (misalnya, integer, float, string).
- **Ukuran:** Array memiliki ukuran tetap yang ditentukan saat deklarasi.
- **Indeks:** Setiap elemen array diakses melalui indeksnya, yang dimulai dari 0.

Operasi Dasar Array:

- **Deklarasi:** Menentukan nama, tipe data, dan ukuran array.
- **Input:** Mengisi nilai ke dalam elemen array.
- **Output:** Menampilkan nilai elemen array.
- **Pencarian:** Menemukan elemen array dengan nilai tertentu.
- **Pengurutan:** Mengurutkan elemen array berdasarkan nilai tertentu.
- **Manipulasi:** Melakukan operasi matematika pada elemen array.

Keuntungan Array:

- **Efisiensi Memori:** Menyimpan data dengan tipe sama secara berurutan.
- **Akses Cepat:** Elemen array dapat diakses dengan mudah melalui indeksnya.
- **Kemudahan Pengolahan:** Memudahkan operasi pada sekumpulan data.

Kekurangan Array:

- **Ukuran Tetap:** Ukuran array tidak dapat diubah setelah deklarasi.
- **Kompleksitas Implementasi:** Operasi kompleks pada array membutuhkan algoritma yang lebih rumit.

Array merupakan struktur data fundamental dalam pemrograman yang memungkinkan penyimpanan dan manipulasi data secara efisien. Memahami konsep array dan operasi dasarnya penting untuk menyelesaikan berbagai masalah pemrograman.

DAFTAR PUSTAKA

Muhardian, Ahmad. (18 Mei 2018). Belajar C++ #10: Mengenal Struktur data Array pada C++. Petanikode. https://www.petanikode.com/cpp-percabangan/

Asisten Praktikum (2024). Array. Learning Management System