DASAR PEMROGRAMAN

Pertemuan XIV

Tujuan

- Mahasiswa memahami cara kerja penyimpanan dan pengaksesan file dalam Komputer
- Mahasiswa mampu melakukan manipulasi terhadap file

Materi

Arsip pada C++

Menulis ke dalam arsip

Membuka Arsip

ARSIP (FILE)

ARSIP

Arsip (file) atau berkas adalah struktur penyimpanan data di dalam memori sekuder seperti disk.

Data disimpan di dalam arsip agar sewaktu-waktu dapat dibuka kembali.

Struktur arsip memungkinkan kita menyimpan data secara permanen dan mengaksesnya kembali bila perlu.


Office
W
X
P
O
N

PUSTAKA ARSIP

IOSTREAM

Hingga saat ini, kita telah menggunakan pustaka **<iostream>** sehingga kita dapat menggunakan **cin** dan **cout** untuk membaca input dan menuliskan output.

FSTREAM

Untuk membaca input dan menuliskan output dari/ke arsip, kita membutuhkan pustaka lainnya, yaitu **<fstream>**.

1. Open
2. Close
3. Read
4. Write

OPERASI DASAR PEMROSESAN ARSIP

OBJEK ARSIP

IFSTREAM

Digunakan untuk membaca informasi dari suatu file

OFSTREAM

Digunakan untuk membuat file dan menuliskan informasi pada suatu file

FSTREAM

Memiliki kemampuan ofstream dan ifstream. Artinya objek ini bisa membuat file, menuliskan informasi pada file, dan membaca informasi dari file

MODE MANIPULASI ARSIP

Mode	Jika File Kosong	Jika File Berisi Data
ios::app	Data dituliskan pada file	Menuliskan data ke posisi terakhir data yang sudah ada sebelumnya
ios::ate	Data dituliskan pada file	Menuliskan data pada posisi yang sudah ditentukan
ios::in	Data pada file dibaca sebagai file kosong	Membuka file dan mengakses data pada file untuk diproses lebih lanjut
ios::out	Data dituliskan pada file	Menggantikan data pada file dengan data baru yang dituliskan
ios::trunc	Data dituliskan pada file	Menghapus data saat file dibuka dan menyimpan data yang baru saat menuliskan data pada file

MENULIS ARSIP

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ofstream file;
    string text= "hello world";

    file.open("contoh.txt");
    file << text;

    return 0;
}
```

1 Gunakan pustaka <fstream>

2 Deklarasikan objek ofstream

3 Buat file dengan nama tertentu jika file belum ada buka file jika file sudah ada sebelumnya

4 Tuliskan data ke dalam file

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ofstream file;
    string text= "hello world";

    file.open("contoh.txt");
    file << text;

    return 0;
}
```

Mode manipulasi arsip pada objek ofstream adalah ios::out. Sehingga jika kita tidak menuliskan mode manipulasi, compiler secara otomatis menggunakan mode ini.

Dalam C++, untuk melakukan output, kita menggunakan operator <<.
Hal yang sama juga berlaku dalam pemrosesan arsip , yang membedakan adalah objek yang digunakan bukanlah **cout** namun **fstream** atau **ofstream**

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ofstream file;
    string text= "hello world";

    file.open("contoh.txt");
    file << text;
    file.close();

    return 0;
}
```

MENUTUP ARSIP

Ketika sebuah program C++ selesai dieksekusi, program akan secara otomatis menutup semua arsip.

Walaupun demikian, menutup arsip setelah dibuka dan digunakan adalah good practice.

MEMBACA ARSIP

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream file;
    string data;

    file.open("contoh.txt");
    file >> data;

    cout << data;

    file.close();

    return 0;
}
```

1. Gunakan pustaka <fstream>
2. Deklarasikan objek ifstream
3. Buka file
4. Baca data dari file lalu tampung pada sebuah variable

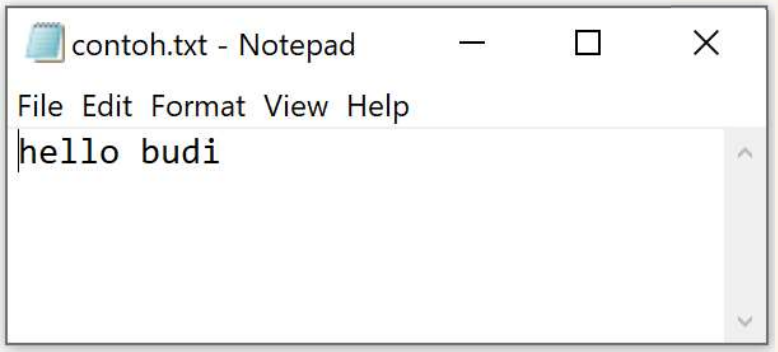
```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream file;
    string data;

    file.open("contoh.txt");
    file >> data;

    cout << data;

    file.close();

    return 0;
}
```

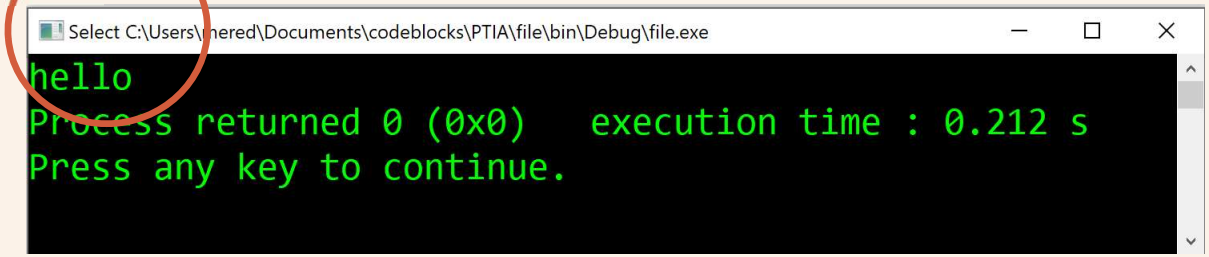
Mode manipulasi arsip pada objek ifstream adalah ios::in. Sehingga jika kita tidak menuliskan mode manipulasi, compiler secara otomatis menggunakan mode ini.

Dalam C++, untuk melakukan input, kita menggunakan operator >>.
Hal yang sama juga berlaku dalam pemrosesan arsip , yang membedakan adalah objek yang digunakan bukanlah **cin** namun **fstream** atau **ifstream**

```
file >> data;
```

Hanya akan membaca data dari file hingga whitespace (spasi atau enter)





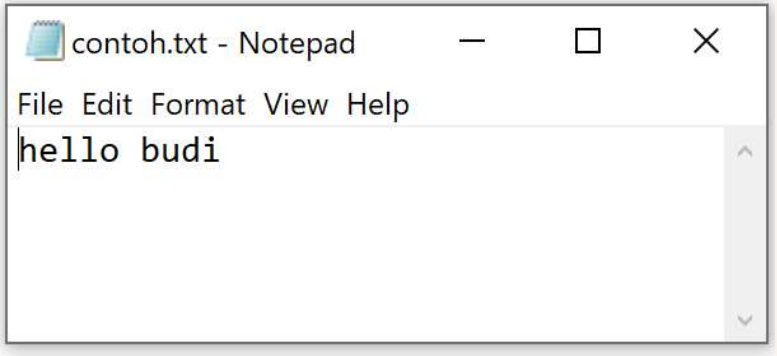
```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream file;
    string data;

    file.open("contoh.txt");

    while (!file.eof())
    {
        file >> data;
        cout << data;
    }

    file.close();

    return 0;
}
```

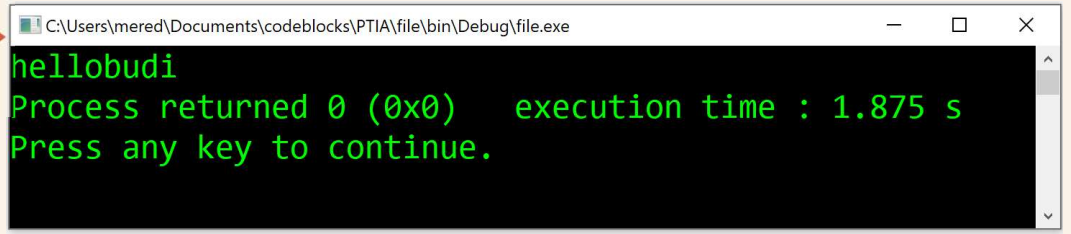
MEMBACA HINGGA EOF

Agar semua data pada file bisa dibaca, perintah `file>>data` harus dilakukan berulang-ulang hingga end of file.

C++ sudah menyediakan fungsi bawaan `eof()` untuk memeriksa end of file.

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream file;
    string data;

    file.open("contoh.txt");

    while (!file.eof())
    {
        file >> data;
        cout << data;
    }

    file.close();

    return 0;
}
```



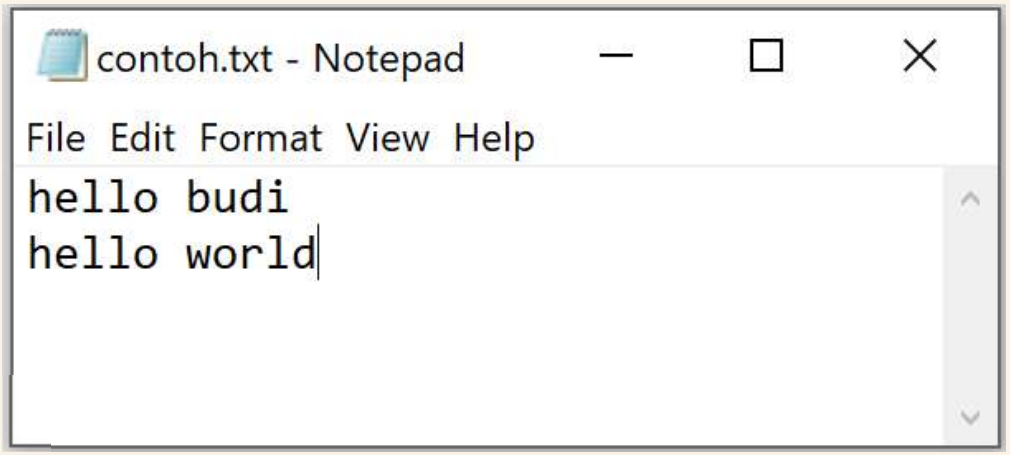


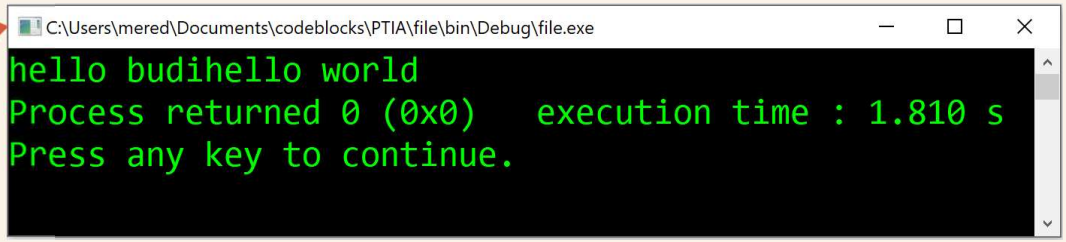
```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    fstream file;
    char data[100];

    file.open("contoh.txt");

    while (!file.eof())
    {
        file.getline(data, 100);
        cout << data;
    }

    file.close();

    return 0;
}
```

MEMBACA ARSIP PER BARIS

Dalam C++, untuk membaca arsip per baris dapat menggunakan `getline`

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    fstream file;
    char data[100];

    file.open("contoh.txt");

    while (!file.eof())
    {
        file.getline(data, 100);
        cout << data;
    }

    file.close();

    return 0;
}
```





Spasi tidak hilang karena data dibaca per baris.