Praktikum 8

Pemrograman Berbasis Obyek (PBO)

(Exception Handling)

A. Tujuan

1. Mahasiswa memahami konsep Exception Handling dalam pemrograman
2. Mahasiswa mengaplikasikan penggunaan *try*, *catch*, dan *throw* dalam pemrograman

B. Dasar Teori

Exception adalah suatu kondisi abnormal yang terjadi pada saat menjalankan program. *Exception* merupakan sebuah fitur khusus yang disediakan oleh bahasa pemrograman C++ untuk menangani kondisi *error* pada saat program sedang berjalan untuk mencegah adanya pemberhentian program karena kesalahan kode atau program, sebagai gantinya programmer secara manual melaukan penanganan atas *error* tersebut.

Error pada program terdapat 3 macam diantaranya:

1. Syntax Error, suatu kesalahan dari penulisan syntax pada program sehingga syntax tersebut tidak dapat dieksekusi oleh program yang pasti membuat program tersebut error.
2. Logical Error, suatu kesalahan yang disebabkan oleh kesalahan penulisan atau rumus yang diterapkan oleh programmer. Misal suatu nilai dibagi dengan angka 0.
3. Runtime Error, error ini akan muncul apabila terjadi miss komunikasi antara program dan file atau fungsi yang dipanggil dalam program, misalnya program itu membutuhkan database yang bernama db_siswa yang disimpan pada localhost,tapi kenyatannya ternyata programmer tidak mempunyai atau belum membuat database tersebut di localhost, program akan tetap berjalan, namun saat aksi simpan data, hapus data atau tampil data pasti program akan error.

Pada pernyataan *Exception Handling*, terdapat 3 kata kunci, yaitu:

1. `try`, adalah sebuah blok dari kumpulan kode dimana biasanya sebuah kesalahan muncul, dan tempat dimana kita juga dapat memberikan tugas pengecualian `throw`. Juga memungkinkan untuk menentukan blok kode yang akan diuji untuk kesalahan saat sedang dijalankan.
2. `catch`, adalah sebuah blok yang akan dieksekusi ketika sebuah Pengecualian telah di lemparkan atau untuk mengeksekusi blok kode, jika terjadi kesalahan dalam blok percobaan

3. `throw`, adalah sebuah pertanyaan yang digunakan untuk melemparkan sebuah Pengecualian secara manual oleh *Programmer,* yaitu dengan melemparkan pengecualian saat masalah terdeteksi.

Berikut contoh penulisan ***C++ Exception*** :

```
try {
  //code program
  throw exception; //exception akan melempar jika ada masalah
}
catch () {
   // code untuk menangani masalah
}
```

jika penulisan program benar atau tidak ada error maka exception akan lolos, namun jika salah c++ exception akan melempar dan catch akan menangkap untuk menangani masalah. Contoh lainnya :

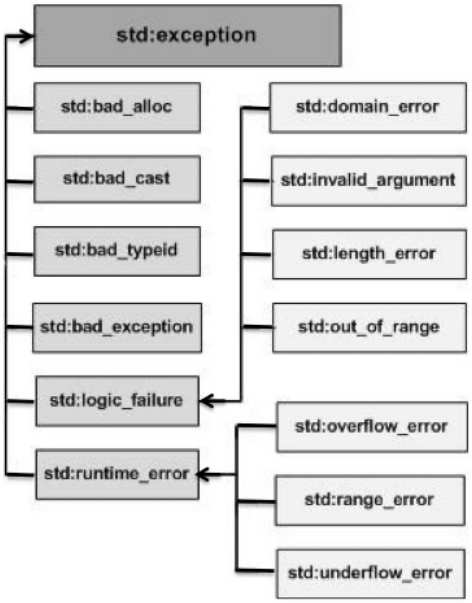
```
try{
    //baris kode
    throw x;//pernyataan pelemparan
}catch(parameter){
    //respon tangkapan
}catch(parameter){
    //bisa mendirikan banyak catch dengan
    //parameter yang berbeda

    //respon tangkapan
}catch(...){
    //default catch, jika tidak ada parameter
    //yang cocok maka ini akan dijalankan

    //respon tangkapan
}
```

C++ Standart Exception

Pada penjelasan di atas kita mencoba untuk melemparkan argumen secara manual kepada penangkap Pengecualian (**catch**), C++ juga menyediakan pustaka Standard Exception yaitu untuk menangani dan menerima argumen berupa obyek **std::exception**. Jadi jika terjadi kesalahan pada ruang ligkup **try** (tanpa melempar), **std::exception** akan menerima obyek dari kesalahan tersebut dan kita dimungkinkan untuk mengakses pesan kesalahan di dalam obyek tersebut.



Standart exception pada library standart C++

Berikut contoh penulisan ***C++ Exception*** :

```
try{
    //kode yang dilindungi
}catch(std::exception &e){
    cout<<"error : "<<e.what()<<endl;
}
```

Contoh 1. Program dengan Exception Handling :

```
#include <iostream>
#include <string>
#include <exception> //untuk obyek exception yang akan digunakan
#include <array> //untuk obyek array yang akan kita gunakan
using namespace std;
int main(int argc, const char * argv[]) {

    cout<<"Awal Program"<<endl; //penanda awal
    try{
        array<int, 4> data = {1, 2, 3,4};
        //pesan array integer 4 elemen
        cout<<data.at(5)<<endl;
        //memanggil array elemen ke 5
    }catch(exception &e){
    //penangkap menggunakan obyek exception
        cout<<e.what()<<endl;
      /*akan dieksekusi karna array data hanya memiliki 4elemen*/
    }
    cout<<"Akhir Program"<<endl; // penanda akhir
    /*penanda akhir: bahwa program berjalan tanpa berhentimeskipun terjadi kesalahan*/
    return 0;
}
```

Contoh 2. Program dengan Exception Handling :

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

int pembagian(int &da, int &db){
    if(db == 0){
        throw "Error: Data Pembagi nol";
    }
    return da/db;
}
int main(int argc, const char * argv[]) {
    int a;
    int b;
    int c;

    while(true){
        cout << "Data A: ";
        cin >> a;
        cout << "Data B: ";
        cin >> b;
        try{
            c = pembagian(a,b);
            cout << c << endl;
        }catch(const char* e){
            cout << e << endl;
        }
    }
    return 0;
}
```

C. Percobaan

1. Buatlah program yang dilengkapi dengan Exception Handling untuk input data dan report berupa struk pada kasir dengan contoh data barang sebagai berikut:

Nama Barang	Harga per kilogram	
Mangga	IDR	15.500
Apel	IDR	34.300
Nanas	IDR	12.300
Anggur	IDR	43.000

Nama Barang	Harga per kilogram	
Semangka	IDR	20.000
Salak	IDR	26.800
Duku	IDR	8.500
Buah Naga	IDR	21.700

2. Buatlah program yang dilengkapi dengan Exception Handling untuk input data Nilai mahasiswa dan output data nilai angka.

Nilai	Bobot	Skala
A	4	75 – 100
AB	3.5	70 – 74,99
B	3	65 – 69,99
BC	2.5	60 – 64,99
C	2	55 – 59,99
D	1	40 – 54,99
E	0	0 – 44,99