



IPB University
— Bogor Indonesia —

KOM120C -- BAHASA PEMROGRAMAN

Template Programming

- Function Template
- Standard Template Library

Tim Pengajar Bahasa Pemrograman IPB University

Template

Dasar untuk pemrograman generik (general programming)

Ide sederhana dalam template adalah mengirimkan tipe data sebagai parameter sedemikian rupa sehingga kita tidak perlu menuliskan kode program untuk berbagai tipe data yang berbeda.

Sebagai contoh, kita ingin dapat menentukan bilangan terbesar (maksimum) dari 2 bilangan int, juga 2 bilangan float, juga 2 bilangan double, dsb. Maka kita tidak perlu menuliskan kode untuk setiap tipe data, seperti yang kita lakukan pada saat membahas topik *function overloading*.

Template akan di-ekspansi pada saat kompilasi. Mirip *macro*. Bedanya untuk template adalah kompilator tidak memeriksa tipe data sebelum di-ekspansi.

Function Overloading

Pembeda fungsi adalah elemen prototipe fungsinya (return type, function-name, argument)

```
int maks(int x, int y)
{
    return (x > y)? x: y;
}
double maks(double x, double y)
{
    return (x > y)? x: y;
}
char maks(char x, char y)
{
    return (x > y)? x: y;
}
```

```
int main()
{
    cout << maks(3, 7) << endl;
    cout << maks(3.0, 7.0) << endl;
    cout << maks('g', 'e') << endl;

    return 0;
}
```

Function Template

Fungsi generik yang dapat digunakan untuk berbagai tipe data

```
template <typename T>
T maks(T x, T y)
{
    return (x > y)? x: y;
}

int main()
{
    cout << maks<int>(3,7) << endl;
    cout << maks<double>(3.0,7.0) << endl;
    cout << maks<char>('g','e') << endl;

    return 0;
}
```

Function Templates. Kita membuat fungsi generik yang dapat digunakan untuk berbagai tipe data.

```
template <typename T>
T maks(T x, T y)
{
    return (x > y)? x: y;
}
```

Class Template

Mendefinisikan class yang bebas dari tipe data. Umumnya digunakan dalam class struktur data. Contoh : membuat class Array, LinkedList, Stack, Queue, etc.

```
template <typename T>
class myArray {
private:
    T *ptr;
    int size;
public:
    myArray(T arr[], int s) {
        ptr = new T[s];
        size = s;
        for(int i=0; i<size; i++)
            ptr[i] = arr[i];
    }
    void print();
};
```

```
template <typename T>
void myArray<T>::print() {
    for (int i=0; i<size; i++)
        cout<<*(ptr + i);
    cout<<endl;
}

int main() {
    int arr[5] = {1,2,3,4,5};
    myArray<int> a(arr, 5);
    a.print();
    return 0;
}
```

Template Specialization

Sifat generik dari suatu *function and class template* dapat dibuat perkecualian

```
template <typename T>
T maks(T x, T y) {
    return (x > y)? x: y;
}

template <>
char maks(char x, char y) {
    return (x < y)? x: y;
}
```

Template Specialization.

Fungsi `maks` didefinisikan secara generik, kecuali untuk tipe data `char`.

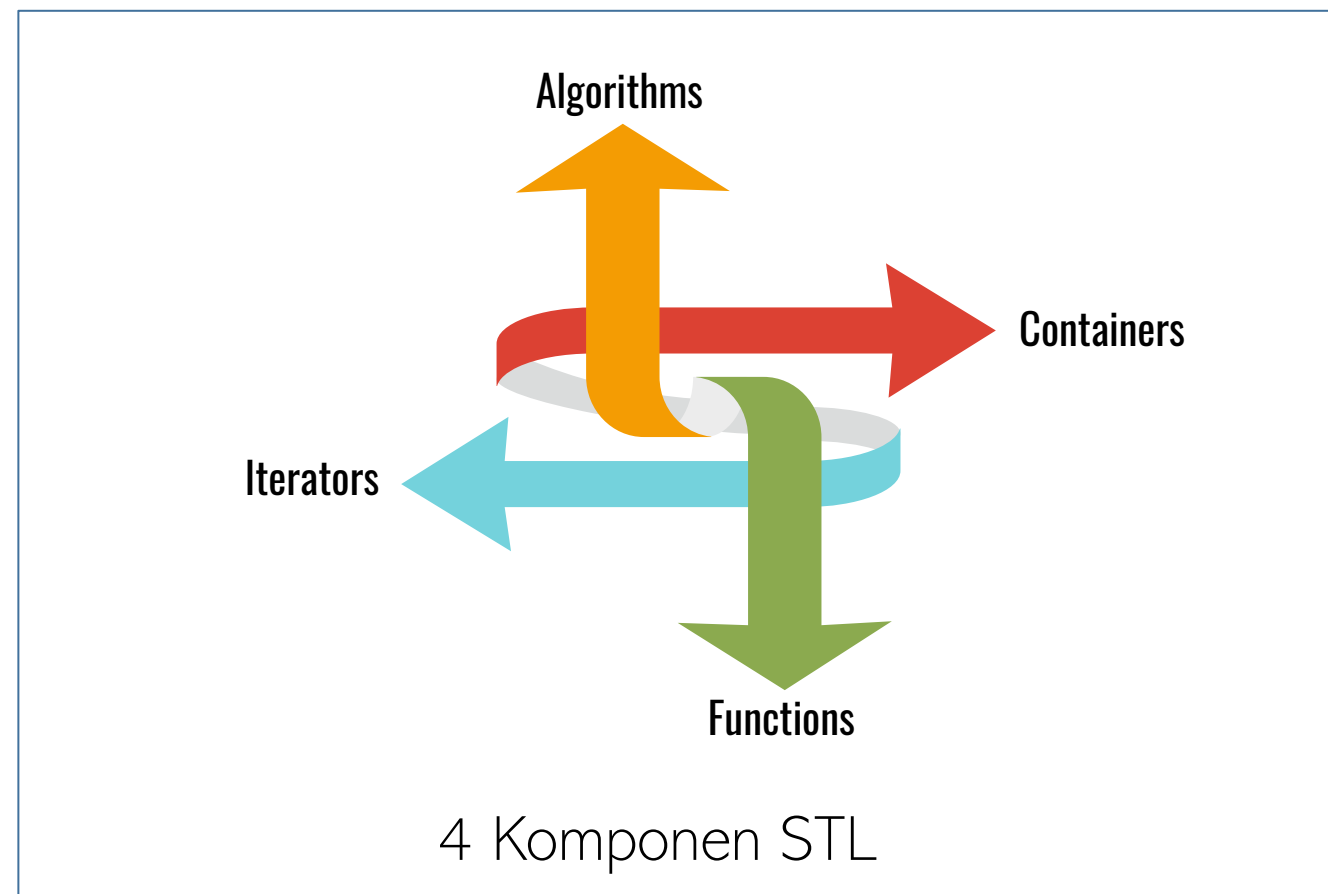
`maks(5, 10) → 10`

`maks('A', 'B') → 'A'`

Standard Template Library (STL)

Salah satu referensi → <https://www.cplusplus.com/reference/stl/>

STL adalah sekumpulan *template class* yang berisi fungsi dan struktur data untuk pemrograman generik seperti list, stack, queue, etc.



STL :: Algorithm

Komponen ini berupa **header** yang mendefinisikan sekumpulan fungsi yang dirancang khusus untuk melakukan algoritme tertentu, misalnya: sort, search, manipulasi array, dsb. Berikut salah satu contoh:

```
#include <algorithm>
#include <iostream>
int main()
{
    int a[] = { 1, 5, 8, 9, 6, 7, 3, 4, 2, 0 };
    int size = sizeof(a) / sizeof(a[0]);
    sort(a, a + size);
    ...
    return 0;
}
```

header file

panggil fungsi sort()

STL :: Container

Container (juga container class) menyimpan objek dan data. Berisi implementasi dari struktur data, seperti set, list, vector, map, stack, queue, etc. Kita akan lebih banyak membahas komponen ini.

STL :: Iterator

Iterator digunakan untuk menunjuk (sebagai pointer) ke alamat memori dari STL Container.

Operasi iterator: `begin()`, `end()`, `advance()`, `next()`, `prev()`

STL :: Function

STL juga menyediakan class yang meng-overload fungsi. Instance dari class ini disebut sebagai objek fungsi atau functors (bukan fungsi). Contoh kegunaan: program berikut menambah 1 ke setiap elemen array.

```
int increment(int x) { return (x+1); }

int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr)/sizeof(arr[0]);

    transform(arr, arr+n, arr, increment);
    // ...

    return 0;
}
```

`transform(inputBegin, inputEnd, OutputBegin, unary_operation)`

STL :: set

Himpunan nilai atau objek unik dengan tipe data tertentu. Nilai disimpan dalam urutan tertentu.

STL juga menyediakan class yang meng-overload fungsi. Instance dari class ini disebut sebagai objek fungsi atau functors (bukan fungsi). Contoh kegunaan: program berikut menambah 1 ke setiap elemen array.

Sintaks:

```
#include <set>
set<datatype> setname;
```

Contoh:

```
set<int> s1; // empty set
set<int> s2 = {6, 10, 5, 1}; // disimpan {1, 5, 6, 10}
```

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

int main()
{
    set<int> s1;
    s1.insert(40); s1.insert(30);
    s1.insert(60); s1.insert(20); s1.insert(40);

    set<int>::iterator p;
    for(p=s1.begin(); p!=s1.end(); ++p)
        cout << *p << " ";
    cout << endl;
    return 0;
}
```

Output:

20 30 40 60

Bagaimana jika ingin urutan *descending*?

Gunakan unary operation function:

```
set<int, greater<int> > s1;
```

STL :: vector

Himpunan nilai atau objek dengan tipe data tertentu. Sama dengan array dinamis.

STL juga menyediakan class yang meng-overload fungsi. Instance dari class ini disebut sebagai objek fungsi atau functors (bukan fungsi). Contoh kegunaan: program berikut menambah 1 ke setiap elemen array.

Sintaks:

```
#include <vector>
vector<datatype> vectorname;
```

Contoh:

```
vector<int> v;                // empty vector
v.assign(7, 100);             // 7 elemen bernilai 100
v.push_back(15);              // menambah elemen bernilai 15
cout << v.size();             // 8

// print semua elemen
vector<int>::iterator it;     // create iterator
for(it=v.begin(); it!=v.end(); ++it)
    cout << *it;
```

Vector of Class

Bisa dilakukan karena class merupakan ADT. Hal yang sama berlaku pada STL lainnya.

Contoh kasus: class People yang merupakan objek dari n Person.

```
class People
{
    Person orang[SIZE];
    int populasi;
public:
    People() { populasi=0; }
    void add(Person p)
        { orang[populasi++]=p; }
    void print();
    double rataanTinggi();
    int lebihTinggi();
};
```

```
class People
{
    vector<Person> orang;
    int populasi;
public:
    People() { populasi=0; }
    void add(Person p)
        { orang.push_back(p); populasi=orang.size(); }
    void print() {
        vector<Person>::iterator p;
        for(p=orang.begin(); p!=orang.end(); ++p)
        {
            cout << p->getNama()
                << " " << p->getUsia() << endl;
        }
    }
    double rataanTinggi();           // bagaimana ?
    int lebihTinggi();              // bagaimana ?
};
```

STL :: pair

Merupakan salah satu utility dalam STL C++.

Sintaks:

```
#include <utility>
pair<datatype,datatype> pairname;
```

Contoh:

```
pair<int,int> p1;

p1.first = 100;  p1.second = 76;

pair<string, Person> p2;           // Person adalah class
                                   // Misalnya string untuk menyimpan NIK
                                   // sebagai primary key
```

Latihan

Gunakan konsep OOP

Deskripsi

Buat program membaca beberapa bilangan bulat, dan menghapus beberapa bilangan pada posisi tertentu.

Format Masukan

Bagian pertama berisi beberapa bilangan bulat yang diakhiri dengan nilai -9 (sentinel). Bagian kedua adalah beberapa bilangan terurut dari kecil ke besar yang menunjukkan elemen beberapa dari bilangan masukan yang dihapus. Diakhiri dengan -9 (sentinel).

Format Keluaran

Baris pertama adalah dua bilangan yang menunjukkan banyaknya bilangan awal dan banyaknya bilangan setelah dihapus. Baris kedua adalah dua nilai rata-rata yang dituliskan dalam dua digit di belakang tanda desimal dari bilangan-bilangan awal dan bilangan-bilangan setelah dihapus. Jika data dalam array kosong, maka nilai rata-rata dituliskan -9.99.

Contoh Input

```
10 20 30 40 50 -9 1 3 4 -9
```

Contoh Output

```
5 2  
30.00 35.00
```