

Template

Menggunakan template programmer dapat menerapkan konsep generic class. Pada modul kali ini akan ditunjukkan contoh membuat generic function menggunakan template. Sebuah generic function dapat menggunakan parameter dengan tipe data yang berbeda-beda.

baca lebih lanjut di chapter 13 ebook OBJECT-ORIENTED PROGRAMMING C++ SIMPLIFIED

Source code 1

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

template <class FUNC>
void show(FUNC par){
    cout << "isi parameter : "<< par << endl;
}

int main(){
    int x = 234;
    float y = 34.56f;
    double d = 3.444456;
    char ch = 'P';
    string s = "Template";
    show(x);
    show(y);
    show(d);
    show(ch);
    show(s);
    return 0;
}
```

Source code 2

```
#include <iostream>
using namespace std;
#define S 5

template <class T>
T hitungJmlElemenMatrix(T arr[]) {
    T sum = arr[0];
    for(int i=1; i<S; i++) {
        sum = sum + arr[i];
    }
    return sum;
}

int main() {
    int arr[S] = {1, 2, 3, 4, 5};
    cout << hitungJmlElemenMatrix(arr) << endl;
    double arr2[S] = {1.5, 2.1, 3.2, 4.0, 5.3};
    cout << hitungJmlElemenMatrix(arr2) << endl;
    return 0;
}
```

1. Jelaskan apa yang dilakukan pada Source code 1 dan Source code 2! Lampirkan output dari kedua source code tersebut!
2. Apa manfaat yang bisa didapatkan dengan membuat generic function menggunakan template?