



NAMA : LASMARIA KATRINA ANGELINA
NAINGGOLAN

NIM : 201402110

MATKUL : PEMROGRAMAN BERORIENTASI OBJEK

OVERLOADING

```
1) #include <iostream>
2)
3) using namespace std;
4)
5) int kali(int var4, int var5) {
6)     return var4 * var5;
7) }
8)
9) int kali(int var8, int var7, int var6) {
10)     return var8 * var7 * var6;
11) }
12)
13) string kali(string var4, string var5) {
14)     return var4 + " Kali " + var5;
15) }
16)
17) int main()
18) {
19)     cout << kali(4, 5) << endl;
20)     cout << kali(8, 7, 6) << endl;
21)     cout << kali("empat", "lima") << endl;
22)
23)     return 0;
24) }
```

- Dalam kode program ini saya membuat 3 buah function dengan nama **kali()**. Ketiga function dibedakan berdasarkan jumlah parameter seperti function **kali()** pertama dan kedua, serta dengan jenis tipe data parameter seperti contoh function **kali()** ketiga.
- Pada saat di panggil, kita harus menulis function **kali()** dengan jumlah dan tipe data argument yang sesuai.
- Di baris 19 terdapat pemanggilan fungsi **kali(4, 5)**. Bisa kita lihat kalau argumen pemanggilan fungsi ini ada 2 dan keduanya bertipe integer. Maka akan cocok dengan definisi fungsi **kali()** di baris 5 – 7.
- Di baris 20 terdapat pemanggilan fungsi **kali(8, 7, 6)**. Kali ini argumen pemanggilan fungsi kali() ada 3 dan ketiganya bertipe integer. Maka ini akan cocok dengan definisi fungsi **kali()** di baris 9 – 11.
- Terakhir di baris 21 terdapat pemanggilan fungsi **kali("empat", "lima")**. Sekarang kedua argument bertipe string yang akan cocok dengan definisi fungsi **kali()** di baris 13– 15.
- Inilah maksud dari function overloading, dimana kita membuat beberapa fungsi dengan nama yang sama.