

Pedoman Umum Pembuatan **Source Code** untuk Bahasa C++

Penamaan

Hindari singkatan dalam penamaan apapun kecuali untuk singkatan yang sudah umum. Gunakan nama yang deskriptif dan bermakna.

Nama *File*

1. Seluruh nama file harus dalam *lowercase*. Antar kata dipisahkan dengan *underscore* (_).
2. File *header* diakhiri dengan ekstensi *.h*.
3. File *source* diakhiri dengan ekstensi *.cpp*.

Nama *Type*

Penamaan *type* menggunakan *camel case*, dengan huruf kapital untuk setiap awal kata.

Contoh:

```
// Kelas
class Matriks { ...
class TabelInteger { ...

// Typedef
typedef int NilaiSiswa;
```

Penamaan Variabel

Seluruh variabel, termasuk parameter pada fungsi dan *data member* menggunakan *lowercase*. Antar kata dipisahkan dengan *underscore*.

Contoh:

```
int kolom;
int current_value;
```

Penamaan Fungsi

Penamaan fungsi menggunakan *camel case*, dengan huruf kapital di awal setiap kata. Akronim lebih baik hanya menggunakan huruf kapital di awal kata.

Contoh:

```
AddValue()
CheckDns()
Delete()
```

Komentar

1. Gunakan format komentar yang kompatibel dengan Doxygen. Daftar perintah dapat diakses pada halaman <http://www.stack.nl/~dimitri/doxygen/manual/index.html>
2. Gunakan `@brief` untuk memberi deskripsi singkat. Seluruh parameter harus didokumentasikan menggunakan `@param`.

Contoh:

```
/** @brief Menjumlahkan objek dengan pecahan masukan.  
 * Hasil penjumlahan dinormalisasi.  
 *  
 * @param pecahan Pecahan yang akan dijumlahkan.  
 */  
void AddPecahan(Pecahan pecahan);
```

3. Sebisa mungkin tulislah kode yang jelas dan tidak memerlukan komentar tambahan di dalam *body*, sehingga komentar hanya ditulis pada definisi method / fungsi.

Format

Umum

1. Panjang baris maksimal 80 karakter.
2. Gunakan indentasi 2 spasi, dan jangan gunakan tab.
3. Sebaiknya gunakan format Unix untuk *End of line*. Silakan atur *editor* yang Anda gunakan.

Whitespace

1. Jangan ada *whitespace* (spasi, tab). pada akhir baris.
2. Fungsi tidak perlu dipisahkan oleh baris kosong.
3. *Assignment operator* (=) dan *operand* selalu dipisahkan spasi. Contohnya, `x = 0;`
4. Tidak ada spasi antara *unary operator* dan *operand*. Contohnya, `x++;`
5. Operator lain biasanya diselingi spasi, namun spasi dapat dihilangkan untuk mempermudah pembacaan. Contohnya `v = w*x + y/z;`
6. Terdapat spasi setelah *keyword* untuk *conditional* dan *loop*.

Contoh:

```
if (found) {  
    DoSomething();  
} else {  
    for (int i = 0; i < 10; i++) {
```

```
        DoSomething();  
    }  
}
```

Deklarasi dan Definisi Fungsi

1. *Return type*, nama fungsi, dan parameter fungsi ditulis dalam satu baris jika cukup. Jika tidak, parameter fungsi dipisah dalam beberapa baris, sejajar dengan parameter pertama.
- 2.
- 3.
4. lalu berada pada akhir dari baris deklarasi fungsi, tidak pada baris baru.
- 5.
- 6.

Contoh:

```
bool Matriks::IsEmpty() {  
    ...  
}  
void NamaKelas::NamaFungsiYangPanjang(int a,  
                                         int b,  
                                         char c) {  
    DoSomething();  
    ...  
}
```

Header

Selalu gunakan *include guard* dengan format <FILE>_H, dengan FILE merupakan nama file tanpa ekstensi.

Contoh:

```
#ifndef FILE_H  
#define FILE_H  
.  
.  
.  
#endif
```