

[Dashboard](#) / [My courses](#) / [ITB_IF2010_2_2526](#) / [Praktikum 1](#) / [Kelas: 4 Sekawan](#)

Started on Friday, 27 February 2026, 9:17 AM

State Finished

Completed on Friday, 27 February 2026, 10:56 AM

Time taken 1 hour 39 mins

Grade 300.00 out of 300.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Anda diminta untuk mendesign kelas untuk representasi data mahasiswa teknik informatika.

Setiap mahasiswa memiliki **nama**, **NIM**, **IPK**, dan **total SKS** yang sudah ditempuh. NIM diberikan secara otomatis oleh sistem dengan format "**135**" diikuti nomor urut pendaftaran yang dimulai dari **0**. Artinya, mahasiswa pertama yang terdaftar memiliki NIM "**1350**", mahasiswa kedua "**1351**", dan seterusnya. Penomoran ini bersifat **global** — tidak peduli siapa yang membuat objek, counter tetap berjalan.

Untuk mendaftarkan mahasiswa, hanya perlu data **nama** mahasiswa. Namun, tidak semua mahasiswa langsung memiliki nama saat didaftarkan. Jika nama belum diketahui, sistem mencatatnya sebagai "**NPC**" (Non-Player Character, istilah internal kampus untuk mahasiswa yang belum mengisi data). Data IPK dan total sks mahasiswa baru selalu mulai dari nol.

Setiap kali mahasiswa menyelesaikan suatu mata kuliah, nilai dan bobot SKS-nya dicatat. IPK dihitung secara kumulatif menggunakan rumus *weighted average*:

$$\text{IPK baru} = (\text{IPK lama} \times \text{SKS lama} + \text{nilai} \times \text{SKS baru}) / (\text{SKS lama} + \text{SKS baru})$$

Sistem juga harus bisa menampilkan informasi lengkap seorang mahasiswa kapan saja diperlukan.

Menampilkan informasi mahasiswa dengan format:

```
INFORMASI MAHASISWA
Nama: [name]
NIM: [nim]
IPK: [ipk] --> Tampilkan 2 desimal
SKS: [total_sks] --> Jangan lupa berikan newline juga
```

Hints

- **static** member adalah milik **kelas**, bukan milik objek. Semua objek berbagi nilai yang sama.
- Member **static** harus **didefinisikan di luar kelas** (contoh: `int Mahasiswa::counter = 0;`).
- Untuk menggabungkan string dengan integer, gunakan `to_string()`.
- `printf` tidak bisa langsung menerima `std::string`. Gunakan `.c_str()` untuk konversi.
- Format specifier: `%s` untuk string, `%d` untuk int, `%.2f` untuk float 2 desimal.
- Jika compile error karena vector ngga keimport, coba tambahin "using namespace std;"

Kata kunci (Gunakan ini untuk memilih nama kelas, tipe data, atribut)

name, ipk, total_sks, counter, Mahasiswa, string, nim, private, public, float, int, static, tambahNilai, info, nilai

Kumpulkan dalam Mahasiswa.cpp

C++14

 [Mahasiswa.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.00 sec, 3.03 MB
2	25	Accepted	0.00 sec, 3.06 MB
3	25	Accepted	0.00 sec, 3.04 MB
4	25	Accepted	0.00 sec, 3.11 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Diberikan file berikut yang berisi sebuah kelas bernama **Paper**. Kelas **Paper** menyimpan dua buah atribut integer dan menghasilkan keluaran khusus setiap dilakukan pemanggilan *constructor*, *copy constructor*, *copy assignment operator*, dan *destructor*.

[Paper.hpp](#)[Paper.cpp](#)

Perhatikan contoh program berikut:

```
#include "Paper.hpp"
int main() {
    Paper a('A');
    a.fold();
    a.fold();
    return 0;
}
```

Program di atas jika dijalankan dengan kelas **Paper** yang disediakan, akan mengeluarkan output:

```
ctor A
fold A(1)
fold A(2)
dtor A
```

Sebagai seorang mahasiswa yang telah memahami konsep dasar OOP dengan baik, anda ditugaskan untuk membuat program utama dengan memanfaatkan kelas Paper yang mengeluarkan output sebagai berikut:

```
ctor A
ctor B
ctor C
cctor C
fold A(1)
fold B(1)
fold C(1)
glue C
fold C(2)
dtor A
setName C => X
fold C(1)
glue C
fold C(2)
dtor C
dtor X
dtor B
```

Unggahlah program utama tersebut dengan nama file **main.cpp**

Hint : Untuk melakukan dealokasi secara dinamis manfaatkan pointer dan fungsi delete untuk dealokasi pointer pada C++

Contoh sintaks delete:

```
#include "Paper.hpp"
int main() {
    Paper *a = new Paper('A');
    delete a;
    return 0;
}
```

C++14

 [main.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	100	Accepted	0.00 sec, 2.89 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Pak Ganesh akan berjualan penganan-penganan. Untuk saat ini, dia akan menjual keik dan panekuk sebagai kudapan malam. Uang yang diperoleh dari menjual keik adalah 51, sedangkan uang yang diperoleh dari menjual panekuk adalah 37. N_rumah adalah variabel untuk menghitung jumlah kunjungan Pak Ganesh ke rumahnya. Jumlah uang dan n_rumah mula-mula adalah 0.

Terdapat beberapa operator yang perlu diimplementasikan:

1. **Operator +** : Pak Ganesh tentunya akan menambahkan stok penganan yang dia jual. Dia akan pulang ke rumah dan menambahkan stok keik dan panekuknya. Setiap mengunjungi rumah, n_rumah akan bertambah sebanyak 1.
2. **Operator -** : Pak Ganesh akan menjual penganan yang dia miliki kepada pembeli. Dari penjualan tersebut, dia akan meraih keuntungan. Jika permintaan pembeli melebihi stok, Pak Ganesh hanya akan menjual sesuai dengan stok yang dia miliki.
3. **Operator ^** : Pak Ganesh akan menyumbangkan penganan yang dia miliki masing-masing sebanyak n buah. Jika ternyata stok yang dia miliki, baik keik maupun panekuk kurang dari n , maka uang yang dia peroleh akan berkurang seturut dengan kekurangan dari sumbangan n buah tersebut. Asumsikan uang dapat negatif (berhutang). Misal, Pak Ganesh ingin menyumbang $n = 6$, akan tetapi stok Pak Ganesh yang dimiliki adalah 3 keik dan 8 panekuk, maka uang Pak Ganesh akan berkurang $(6-3) * 51$ dan stok Pak Ganesh sekarang menjadi 0 keik dan 2 panekuk.

Bantulah Pak Ganesh untuk mengimplementasikan header Penganan berikut.

Berikut file [Penganan.hpp](#)

Kumpulkan dengan nama file **Penganan.cpp**

C++14

 [Penganan.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.10 MB
2	10	Accepted	0.00 sec, 3.05 MB
3	10	Accepted	0.00 sec, 3.05 MB
4	10	Accepted	0.00 sec, 3.05 MB
5	10	Accepted	0.00 sec, 2.97 MB
6	10	Accepted	0.00 sec, 3.05 MB
7	10	Accepted	0.00 sec, 3.00 MB
8	10	Accepted	0.00 sec, 2.99 MB
9	10	Accepted	0.00 sec, 2.92 MB
10	10	Accepted	0.00 sec, 2.93 MB

[◀ Announcements](#)

Jump to...

[Post praktikum 1 ▶](#)