

[Dashboard](#) / [My courses](#) / [ITB IF2010 2 2425](#) / [\[CPP\] Praktikum 3](#) / [Praktikum 3 \(Latihan\)](#).

| | |
|--------------|----------------------------------|
| Started on | Thursday, 17 April 2025, 3:55 AM |
| State | Finished |
| Completed on | Thursday, 17 April 2025, 9:09 AM |
| Time taken | 5 hours 13 mins |
| Grade | 290.00 out of 300.00 (97%) |

Question **1**

Partially correct

Mark 90.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Gudang

Di kota **Wonderland**, manajemen logistik adalah urusan yang sangat penting. Kota ini punya satu gudang pusat yang menangani **penyimpanan barang-barang penting**, mulai dari **barang elektronik** hingga **bahan makanan**. Gudang ini adalah jantung distribusi kota.

Pengelola gudang, **Pak Bram**, dikenal sangat disiplin dan rapi. Setiap kali ada barang masuk, ia akan memeriksa:

- Apakah barangnya **masih layak simpan** (tidak kedaluwarsa)?
- Apakah gudang punya **kapasitas cukup**?
- Apakah ada cukup **uang** untuk memproses barang masuk
- Dan tentu saja—apakah ada cukup **tenaga kerja** untuk menangani barang tersebut?

Namun, seiring berkembangnya kota dan banyaknya barang yang harus ditangani, Pak Bram tidak bisa terus-terusan memproses semuanya secara manual. Ia membutuhkan sebuah **sistem cerdas** untuk:

- Menyimpan barang berdasarkan jenisnya.
- Mendeteksi dan memberi **peringatan otomatis** saat kondisi tidak terpenuhi.
- Memberi tahu status gudang: kapasitas, uang, tenaga kerja, dan barang yang disimpan.

Berikut adalah file header yang diberikan:

- [Exception.hpp](#)
- [Barang.hpp](#)
- [BarangMakanan.hpp](#)
- [BarangElektronik.hpp](#)
- [Gudang.hpp](#)

Anda perlu membuat **Barang.cpp**, **BarangMakanan.cpp**, **BarangElektronik.cpp**, dan **Gudang.cpp** sesuai spesifikasi yang diberikan. Kumpulkan semua dalam 1 zip file dengan nama **submission.zip**

Catatan:

- Semua output diakhiri dengan newline

C++14

 [submission.zip](#)

Score: 90

Blackbox

Score: 90

Verdict: Wrong answer

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 10 | Accepted | 0.00 sec, 3.07 MB |
| 2 | 10 | Accepted | 0.00 sec, 3.09 MB |
| 3 | 10 | Accepted | 0.00 sec, 3.08 MB |
| 4 | 10 | Accepted | 0.00 sec, 2.89 MB |
| 5 | 10 | Accepted | 0.00 sec, 3.10 MB |

| No | Score | Verdict | Description |
|----|-------|--------------|-------------------|
| 6 | 10 | Accepted | 0.00 sec, 3.05 MB |
| 7 | 10 | Accepted | 0.00 sec, 2.97 MB |
| 8 | 0 | Wrong answer | 0.00 sec, 3.01 MB |
| 9 | 10 | Accepted | 0.00 sec, 3.00 MB |
| 10 | 10 | Accepted | 0.00 sec, 3.00 MB |

Question **2**

Correct

Mark 100.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Playlist

Nada, seorang mahasiswa ITB yang sangat mencintai musik, merasa frustrasi dengan aplikasi pemutar musik yang terlalu rumit. Ia ingin sebuah sistem pemutaran musik yang sederhana namun powerful, yang bisa mengelola lagu-lagu favoritnya dengan efisien.

Sistem yang diinginkan Nada harus memiliki komponen utama

1. Manajemen Playlist (Playlist)

- Dapat membuat playlist dengan nama tertentu
- Menambah dan menghapus lagu dari playlist
- Menampilkan daftar lagu dalam playlist dengan format yang terstruktur
- Mencegah duplikasi lagu dalam playlist yang sama

2.Sistem Pemutaran (Playback)

- Mengatur antrian pemutaran lagu
- Dapat menambahkan lagu individual atau seluruh playlist ke antrian
- Memiliki kontrol pemutaran: next, skip, play all
- Dapat membersihkan antrian pemutaran

Berikut adalah file header yang diberikan:

- [Song.hpp](#)
- [Playlist.hpp](#)
- [Playback.hpp](#)

Anda perlu mengimplementasikan **Playlist.cpp** dan **Playback.cpp** dan compress ke dalam **submission2.zip** sesuai dengan spesifikasi yang diberikan dalam header files.

Catatan Penting:

- Semua output harus diakhiri dengan newline
- Format output harus sesuai dengan spesifikasi yang diberikan dalam komentar header files
- Untuk throw out_of_range, gunakan code **throw out_of_range("Index out of range");**

C++14

 [submission2.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 15 | Accepted | 0.00 sec, 2.96 MB |
| 2 | 15 | Accepted | 0.00 sec, 2.89 MB |
| 3 | 15 | Accepted | 0.00 sec, 2.98 MB |
| 4 | 15 | Accepted | 0.00 sec, 3.10 MB |

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 5 | 20 | Accepted | 0.00 sec, 2.96 MB |
| 6 | 20 | Accepted | 0.00 sec, 3.01 MB |

Question **3**

Correct

Mark 100.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Gradebook

Pak Subianto, seorang dosen di Universitas Indonesia Emas, membutuhkan sistem untuk mengelola nilai mahasiswanya yang semakin banyak. Untuk efisiensi, ia ingin membuat suatu sistem Gradebook.

Sistem Gradebook yang dibutuhkan harus dapat:

- Menambah/update nilai mahasiswa
- Menghapus data mahasiswa
- Mencari nilai mahasiswa
- Menampilkan daftar nilai (biasa dan ranking)
- Mencari mahasiswa dengan nilai di atas threshold
- Menghitung rata-rata kelas
- Mengetahui jumlah mahasiswa

Implementasikan fungsi-fungsi dalam [Gradebook.hpp](#) ke dalam file Gradebook.cpp sesuai spesifikasi yang diberikan. Kumpulkan Gradebook.cpp

Catatan:

- Semua output diakhiri newline
- Gunakan STL map, vector, sort, pair

C++14

 [Gradebook.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 25 | Accepted | 0.00 sec, 3.09 MB |
| 2 | 25 | Accepted | 0.00 sec, 3.09 MB |
| 3 | 25 | Accepted | 0.00 sec, 3.14 MB |
| 4 | 25 | Accepted | 0.00 sec, 3.09 MB |

◀ [Praktikum 3](#)

Jump to...

[Link Slides Praktikum 3](#) ▶