Judul Proyek: Battleships: Engage in Explosive Rivalry (BEER)

Informasi Umum:

- **Mata Kuliah:** CITS3002 – Jaringan Komputer
- **Metode Pengumpulan:** via LMS
- **Bobot Nilai:** 30%
- **Format:** Kelompok berdua (jika tidak menemukan pasangan, boleh individu)
- **Output yang Dikumpulkan:**
    - Laporan (PDF)
    - Berkas kode (ZIP)
    - Video demo (maks. 10 menit, link ditulis di laporan)

Jika Anda mengerjakan sendiri, tetap dinilai seperti proyek kelompok.

Tujuan Proyek:

Anda ditugaskan oleh **Socket & Sunk** untuk membuat game Battleship berbasis jaringan bernama **BEER**. Tugas utama Anda adalah:

- Mengimplementasikan **server multiplayer**
- Menyinkronkan klien (pemain)
- Menangani status permainan dan komunikasi

Bahasa pemrograman bebas, tapi Python/C/C++ direkomendasikan.

Struktur Proyek (Berbasis Tier):

Tugas dibagi dalam 4 tingkat kesulitan (Tier 1–4). Semakin tinggi tier, semakin lengkap dan kuat solusi Anda.

Tier 1 – Game Dasar 2 Pemain dengan Konkurensi

T1.1 Masalah Konkurensi:

- Klien saat ini mengalami masalah seperti:
    - Prompt muncul terlambat
    - Pesan server muncul setelah input
    - Kesalahan tidak sesuai input
- **Solusi:** Gunakan threading di klien:
    - Thread 1: menerima dan menampilkan pesan server
    - Thread 2: menerima input dan mengirim perintah
    - Server: gunakan thread untuk setiap koneksi

T1.2 Server dan Dua Klien:

- Server menerima **dua koneksi**.
- Setelah dua pemain bergabung, permainan dimulai.

T1.3 Alur Game Dasar:

- Pemain menempatkan kapal → bergantian menembak
- Laporan: kena/miss, kapal tenggelam
- Permainan berakhir saat semua kapal satu pemain hancur

T1.4 Pertukaran Pesan Sederhana:

- Contoh pesan: FIRE B5, RESULT MISS
- Tidak perlu menangani input aneh pada tahap ini

T1.5 Tidak Perlu Tangani Diskoneksi:

- Jika klien keluar, game bisa diakhiri atau server ditutup

Tier 2 – Kenyamanan dan Skalabilitas

T2.1 Validasi Input:

- Tangani input tidak valid secara aman
- Tampilkan error message, bukan crash

T2.2 Game Berulang:

- Setelah satu game berakhir, server bisa memulai lagi tanpa restart

T2.3 Timeout:

- Jika pemain diam terlalu lama (mis. >30 detik), otomatis kalah atau gilirannya dilewati

T2.4 Diskoneksi:

- Deteksi jika socket klien tertutup
- Perlakukan sebagai "forfeit" atau tangani dengan elegan

T2.5 Klien Berlebih:

- Klien ke-3 bisa ditolak atau masuk ruang tunggu hingga game selesai

Tier 3 – Banyak Koneksi

T3.1 Banyak Klien:

- Server boleh menerima >2 klien
- 2 klien aktif bermain, sisanya jadi penonton

T3.2 Penonton:

- Terima update game (tembakan, hasil, pemenang)
- Perintah mereka diabaikan

T3.3 Reconnect:

- Pemain dapat masuk kembali dalam waktu tertentu (mis. 60 detik)
- Harus mempertahankan status permainan selama periode reconnect

T3.4 Transisi ke Match Berikutnya:

- Setelah game selesai, pilih dua pemain berikutnya dari antrean

Tier 4 – Fitur Lanjutan (Minimal 2 fitur, harus termasuk T4.1)

T4.1 Protokol Rendah Buatan Sendiri dengan Checksum:

- Buat format paket sendiri
- Sertakan header: sequence number, type, checksum, dll
- Verifikasi integritas data (mis. CRC-32)

T4.2 Kanal Pesan Chat (IM):

- Pemain & penonton dapat mengirim pesan teks
- Gunakan format protokol untuk ID pengirim

T4.3 Enkripsi:

- Implementasikan enkripsi simetris (mis. AES)
- Tentukan cara pertukaran kunci atau gunakan kunci tetap

T4.4 Analisis & Mitigasi Keamanan:

- Contoh serangan: session hijacking, replay
- Implementasikan mitigasi seperti session token, sequence number

Pengumpulan:

- Satu anggota grup mengumpulkan melalui LMS:
    - 12345678_23456789_BEER.pdf
    - 12345678_23456789_BEER.zip
- Laporan **wajib dalam PDF**, demo video < 10 menit
- Sertakan tautan video pada halaman awal laporan

Rubrik Penilaian:

- **Tier 1:** 40%
- **Tier 2:** 15%
- **Tier 3:** 15%
- **Tier 4:** 20%

- **Laporan & Demo:** 10%

Tiers harus dikerjakan berurutan agar mendapat nilai penuh. Fitur bisa parsial jika stabil.