

# RANCANGAN APLIKASI TUGAS BESAR

## MATA KULIAH PEMROGRAMAN JARINGAN

### 1. HALAMAN JUDUL

Judul Aplikasi:

Ping-Pong Based on CLI (ASCII)

(Implementasi Host-Client Architecture pada Permainan Multiplayer Real-Time Berbasis TCP Socket)

Disusun Oleh:

Nama Mahasiswa : Raditya Rizki Raharja

NIM : 714240041

Kelas : 2A

### 2. DESKRIPSI APLIKASI

#### A. Gambaran Umum

**Terminal Pong** adalah versi fungsional dasar dari permainan ping-pong multiplayer yang berjalan di Terminal/CLI. Aplikasi ini dirancang dengan arsitektur **Host-Client**, di mana satu perangkat bertindak sebagai Server sekaligus Pemain 1 (Host), dan perangkat lain bertindak sebagai Pemain 2 (Guest).

Tujuan utama versi MVP ini adalah membuktikan keberhasilan pertukaran paket data koordinat secara *real-time* melalui protokol TCP tanpa kompleksitas fitur tambahan seperti *auto-discovery*.

#### B. Perubahan Arsitektur (2 Devices)

Berbeda dengan desain server terdedikasi, sistem ini mengoptimalkan penggunaan perangkat keras:

1. **Device A (Host):** Menjalankan skrip Server (Game Logic) di latar belakang dan skrip Client (Player 1) secara bersamaan.
2. **Device B (Guest):** Hanya menjalankan skrip Client (Player 2).

#### C. Fitur Utama

1. Host IP Display:  
Saat Server dinyalakan di Device A, sistem secara otomatis mendeteksi IP Address lokal (LAN) dan menampilkannya di layar (misal: HOST IP: 192.168.1.5). Ini memudahkan Guest untuk mengetahui alamat tujuan.
2. Manual IP Connection:

Client di Device B memiliki fitur input sederhana untuk memasukkan IP Host yang ditampilkan tersebut.

3. Unified TCP Stream:

Menggunakan satu kanal TCP yang stabil untuk sinkronisasi posisi bola, paddle, dan skor.

4. TUI Sederhana (ASCII):

Visualisasi menggunakan karakter ASCII standar (|, O, :) untuk menjamin kompatibilitas penuh antara Host (PC) dan Guest (PC atau Android/Termux).

### 3. WORKFLOW APLIKASI

Berikut adalah alur kerja sistem Host-Client:

#### Tahap 1: Inisialisasi Host (Device 1)

1. Mahasiswa menjalankan server.py di Terminal 1.
2. Server melakukan *binding* ke 0.0.0.0 dan mendeteksi IP LAN aktif.
3. **Output:** Server menampilkan pesan besar: "**SERVER READY. HOST IP: 192.168.1.10**".
4. Mahasiswa membuka Terminal 2 (di Device yang sama) dan menjalankan client.py.
5. Client 1 otomatis terhubung ke localhost (127.0.0.1) dan masuk status "Waiting for Opponent".

#### Tahap 2: Koneksi Guest (Device 2)

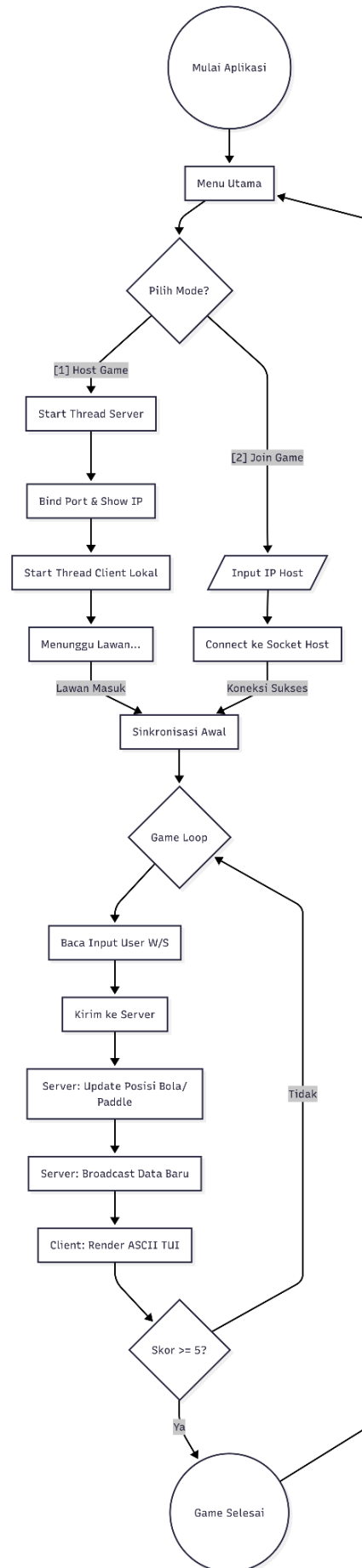
1. Mahasiswa menjalankan client.py di Device 2 (Laptop lain atau HP Termux).
2. Aplikasi meminta input: Masukkan IP Host: .
3. User mengetik IP yang tertera di layar Device 1 (misal: 192.168.1.10).
4. Koneksi TCP terbentuk.

#### Tahap 3: Permainan (Synchronization Loop)

1. Server mendeteksi 2 klien terhubung -> Mengirim sinyal START.
2. **Host (P1)** mengendalikan paddle kiri dengan tombol 'W/S'. Input dikirim via *loopback* ke Server.
3. **Guest (P2)** mengendalikan paddle kanan dengan tombol 'W/S'. Input dikirim via Wi-Fi ke Server.
4. Server memproses fisika bola dan mem-*broadcast* posisi terbaru ke Host dan Guest.

#### Tahap 4: Terminasi

1. Jika skor mencapai 5, Server mengirim paket GAMEOVER.
2. Koneksi ditutup, aplikasi Client kembali ke terminal.



## 4. PENUTUP

### A. Kesimpulan

Rancangan MVP dengan model Host-Client ini merupakan solusi paling efisien untuk mendemonstrasikan kemampuan pemrograman jaringan dalam lingkup Tugas Besar. Dengan menghilangkan ketergantungan pada perangkat ketiga (Dedicated Server) dan fitur kompleks, pengembangan dapat difokuskan pada kestabilan koneksi TCP dan sinkronisasi data permainan. Arsitektur ini membuktikan bahwa komunikasi *real-time* dapat berjalan lancar antar perangkat dengan konfigurasi jaringan manual yang sederhana namun handal.

### B. Rencana Pengembangan (Future Development)

Meskipun versi MVP ini berfokus pada fungsionalitas dasar, rancangan sistem telah disiapkan untuk mendukung pengembangan lebih lanjut guna meningkatkan pengalaman pengguna dan performa jaringan. Berikut adalah fitur-fitur yang direncanakan untuk implementasi masa depan:

1. Zero-Config Auto Discovery (UDP Broadcast):  
Menggantikan input IP manual dengan mekanisme UDP Broadcast. Klien akan secara otomatis memindai jaringan lokal untuk menemukan Server yang aktif, meningkatkan User Experience (UX) setara dengan aplikasi komersial.
2. Hybrid Protocol Implementation (TCP + UDP):  
Mengimplementasikan arsitektur dual-socket di mana TCP digunakan untuk data kritis (skor, status game) demi reliabilitas, sementara UDP digunakan untuk sinkronisasi posisi koordinat bola dan paddle demi latensi yang lebih rendah (high-frequency updates).
3. Modern TUI & Cross-Platform UX:  
Peningkatan visual menggunakan karakter Unicode Block untuk tampilan yang lebih solid dan estetik, serta implementasi skema kontrol adaptif yang membedakan input keyboard fisik (PC) dengan kontrol sentuh (Termux/Mobile).
4. Lobby Management System:  
Penambahan fitur ruang tunggu (waiting room) yang memungkinkan pemain melakukan sinkronisasi status "Ready" sebelum permainan dimulai, serta fitur obrolan (chat) sederhana berbasis teks di dalam lobi.