**LAPORAN UJIAN TENGAH SEMESTER**

**JARINGAN KOMPUTER**



**Dosen Penganpu:**

Taufiqotul Bariyah, S.Kom., M.IM., MCE

**Disusun Oleh:**

|  |  |
| --- | --- |
| Ari Setia Hinanda | ( 3012310005 ) |
| Muhammad Muqoffin Nuha | ( 3012310023 ) |

**IF-3B**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS INTERNASIONAL SEMEN INDONESIA**

**GRESIK**

**2024**

**Implementasi Aplikasi Chat Berbasis**

**Socket Programming Menggunakan Java**

1. **Pendahuluan**

* **Latar Belakang**

Komunikasi real-time melalui jaringan komputer adalah salah satu fondasi utama teknologi modern. Socket programming menggunakan Java memungkinkan pengembangan aplikasi yang efisien untuk komunikasi jaringan, termasuk aplikasi chat.

* **Tujuan**

Tujuan dibuatnya aplikasi ini adalah guna memenuhi penugasan Ujian Tengah Semester yaitu membuat aplikasi chat berbasis socket programming dengan kemampuan multi-client untuk mendukunng komunikasi pesan teks di dalam jaringan lokal (LAN).

1. **Arsitektur Sistem**

## **Batasan Aplikasi**

* **Server:**
  + Server berjalan pada alamat IP dan port yang ditentukan.
  + Server harus mampu menangani lebih dari satu client secara bersamaan menggunakan multithreading.
  + Ketika menerima pesan dari satu client, server harus mengirimkan pesan tersebut ke semua client yang terhubung.
  + Server harus mencatat setiap koneksi, pesan yang diterima, dan pesan yang dikirim pada konsol.
* **Client:**
  + Client terhubung ke server menggunakan alamat IP dan port yang ditentukan.
  + Client harus memiliki antarmuka pengguna sederhana dengan:
    - Kotak teks untuk menampilkan pesan yang diterima.
    - Kotak input untuk menulis pesan yang akan dikirim.
    - Tombol untuk mengirim pesan.
    - Ketika client mengirimkan pesan, pesan tersebut harus muncul di client lain yang terhubung.

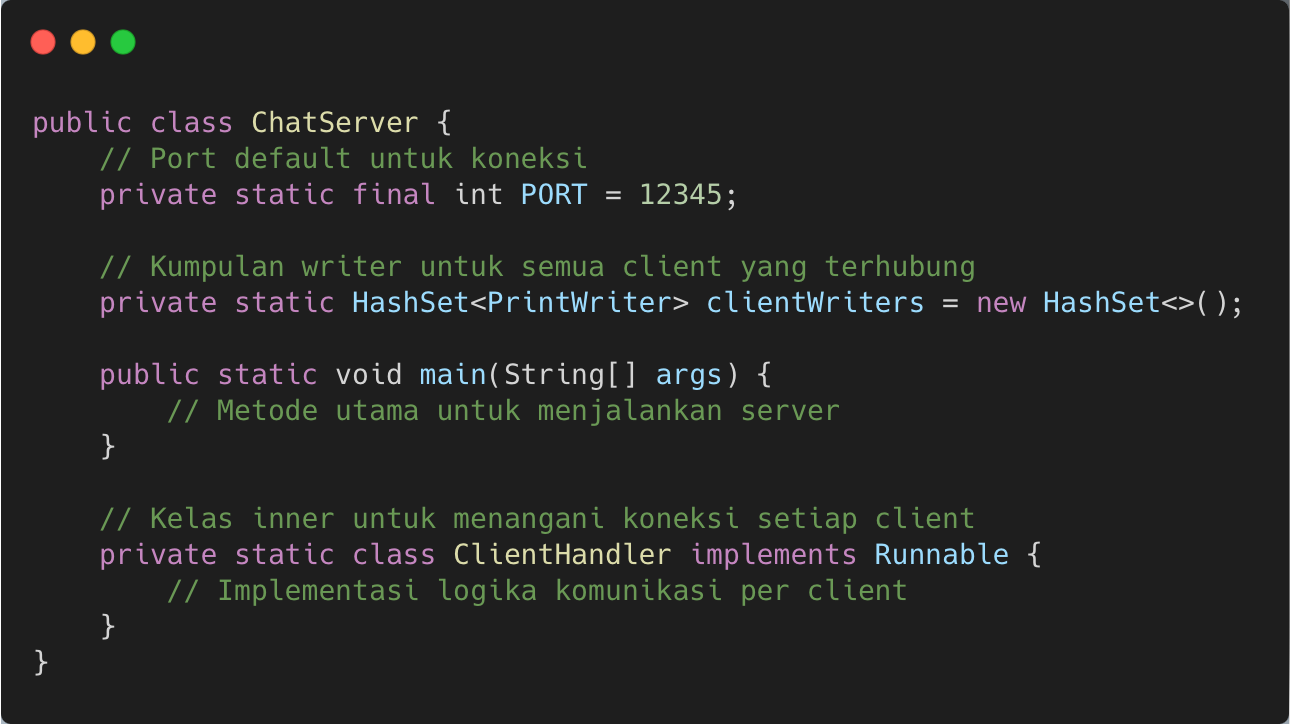
## **Komponen Utama**

* **ChatServer**: Mengelola koneksi dan distribusi pesan
* **ChatClient**: Antarmuka pengguna untuk mengirim dan menerima pesan

## **Teknologi Yang Digunakan**

* **Bahasa Pemrograman:** Java
* **Socket Programming:**java.net
* **Antarmuka Grafis:** Java Swing
* **Fitur Multi-threading:** Java Thread API

1. **Implementasi Teknis**
   1. **Server (ChatServer.java)**
      1. Umum:
      * Port default: 12345
      * Mendukung koneksi multi-client
      * Broadcast pesan ke semua client
      1. Struktur Kode Utama

****

* + 1. Multi-Threading
       1. ServerSocket Listener

****

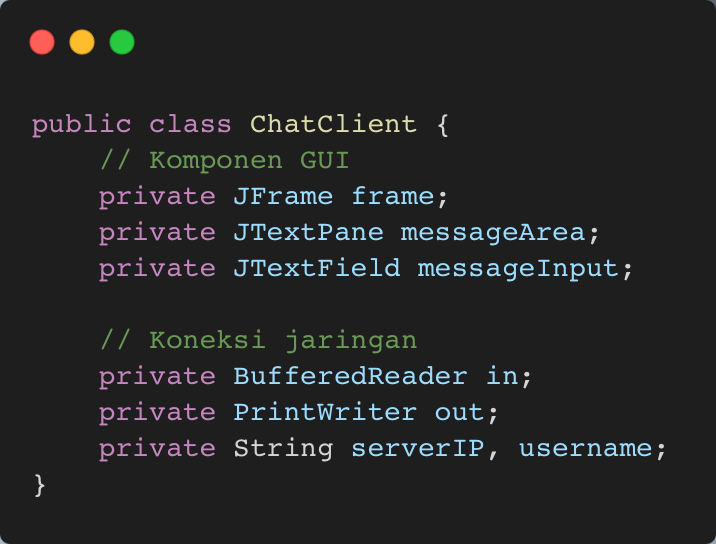
* + - * *serverSocket.accept()* memblokir dan menunggu koneksi baru
      * Setiap koneksi menghasilkan thread terpisah
      * Memungkinkan komunikasi paralel dengan multiple client
      1. Manajemen Koneksi Client

****

* + - 1. Mekanisme Broadcast

****

* Menggunakan *synchronized* untuk thread-safety
* Menjamin pesan dikirim ke semua client
  1. **Client (ChatClient.java**
     1. Umum:
     + Konfigurasi koneksi dinamis
     + Antarmuka grafis dengan format pesan berbeda
     + Manajemen koneksi dan error handlingclient
     1. Arsitektur Antarmuka Pengguna

****

* + 1. Konfigurasi Dinamis

****

* + 1. Koneksi dan Penerimaan Pesan

**A computer screen shot of a program code

Description automatically generated**

* + 1. Formatting Pesan

**A screen shot of a computer code

Description automatically generated**

1. **Pengujian**

* **Lingkungan Pengujian**
* **Hardware:** Laptop.
* **Software:** JDK 21, Visual Studio Code.
* **Jaringan:** LAN menggunakan koneksi Wi-Fi lokal.
  + - * **Running Server**

A screenshot of a computer

Description automatically generated

* + - * **Running Client**
* **Inisialisasi (IP Server, Port, Username) secara dinamis**

**A screenshot of a computer

Description automatically generated**

* **Tampilan Forum**

A screenshot of a chat client

Description automatically generated

1. **Kesimpulan**

* Aplikasi chat berbasis socket programming berhasil dibuat dan berfungsi sesuai spesifikasi.
* Server mendukung komunikasi multi-client dengan stabilitas yang baik.
* Antarmuka pengguna sederhana namun efektif untuk pengiriman dan penerimaan pesan.

1. **Lampiran**
2. **Persyaratan Sistem**
   1. Java Development Kit (JDK) 11 atau lebih tinggi
   2. Koneksi jaringan local/internet
3. **Konfigurasi Port**
   1. Default: 12345
   2. Dapat diubah di ChatServer.java dan ChatClient.java