MEANS (Media Informasi Analisa dan Sistem)
p-ISSN: 2548-6985, e-ISSN:2599-3089

Konsep Komunikasi Data Antar Sistem Informasi Menggunakan Transmisi Kontrol Protokol (TCP)

[1] Joko Triyono, [2] Harmastuti , [3]Robertus Tefa
[1] Universitas AKPRIND Indonesia, Jl. Kalisahak Komplek Balapan no 28 Yogyakarta
E-Mail: jack@akprind.ac.id[1]

[2] Harmastuti
[1] Universitas AKPRIND Indonesia, Jl. Kalisahak Komplek Balapan no 28 Yogyakarta
E-Mail: harmastuti@akprind.ac.id[2]

[3]Robertus Tefa
[1] Universitas AKPRIND Indonesia, Jl. Kalisahak Komplek Balapan no 28 Yogyakarta
E-Mail: robertustefa1@gmail.com[3]

Abstract

Data communication between information systems is an important component in modern computer networks. The Input Process Output Cycle in an information system is something that will always happen and the output of an information system can become input for another information system, which is almost certain to happen. Current conditions cannot be denied that many information systems have been built and will always experience development. On the one hand, it is not easy to migrate to a new system, because in reality the old system is still stable. This research discusses the concept of data communication between information systems using Transmission Control Protocol (TCP). TCP is a protocol widely used in computer networks to ensure reliable data transmission between devices. From the research results, it was found that the TCP concept makes it possible to overcome data communication problems as expected.

Keywords: Communication, Data, Input, Output, TCP.

PENDAHULUAN

Komunikasi data antar sistem informasi merupakan komponen penting dalam jaringan komputer modern. Untuk memastikan bahwa data dikirim dan diterima dengan andal, protokol komunikasi seperti TCP digunakan secara luas. TCP adalah protokol koneksi yang menyediakan transmisi data yang dapat diandalkan dan berurutan. Tidak bisa dipungkiri saat ini telah banyak sistem informasi maupun aplikasi yang berdiri sendiri dan tidak saling terhubung satu sama lain, walaupun sebetulnya banyak beberapa informasi akan saling mendukung untuk meningkatkan mutu informasi yang dihasilkan. Bahkan tidak jarang sebuah output dari sebuah sistem informasi akan menjadi input untuk sistem informasi yang lain. Berdasarkan masalah-masalah tersebut maka dibutuhkan sebuah metode untuk membangun konsep komunikasi data yang bisa memberikan solusi. Yang akhirnya pembahasan ini akan memberikan pemahaman mendalam tentang cara kerja TCP dalam komunikasi data antar sistem informasi dan untuk

mengidentifikasi keuntungan dan tantangan yang terkait dengan penggunaannya serta implementasi dalam yang mungkin bisa diterapkan.

Beberapa pustaka yang menguatkan penelitian ini adalah tentang "Security Framework for Distributed Database System" [1], juga tentang implementasi sistem terdistribusi pada replikasi dan web service [1], tentang model kontrol transaksi rdbms [1], tentang model aplikasi terdistribusi menggunakan FTP [1].

Beberapa buku yang membahas tentang TCP [1] Arsitektur dan Mekanisme Kerja TCP, Model OSI dan TCP/IP dimana TCP bekerja pada lapisan transport dalam model OSI dan berfungsi di atas lapisan Internet dalam model TCP/IP. Bagian ini akan membahas peran TCP dalam model tersebut. Proses Establishment dan Termination terdiri dari dua model yaitu (A) Three-Way Handshake

SYN: Klien mengirim segmen SYN untuk memulai koneksi.
SYN-ACK: Server merespons dengan segmen SYN-ACK.
ACK: Klien mengirim segmen ACK untuk menyelesaikan pembentukan koneksi.

(B) Termination

[1]FIN: Salah satu pihak mengirim segmen FIN.
[2]FIN-ACK: Pihak lain mengakui FIN dan mengirim segmen FIN sendiri.
[3]ACK: Koneksi ditutup setelah segmen ACK diterima.

Juga tentang Pengendalian Aliran dan Kesalahan meliputi:

a. Sliding Window: Mengelola ukuran jendela pengiriman untuk kontrol aliran.
b. Retransmission: Mengirim ulang segmen yang hilang atau rusak.
c. Checksum: Memverifikasi integritas data.

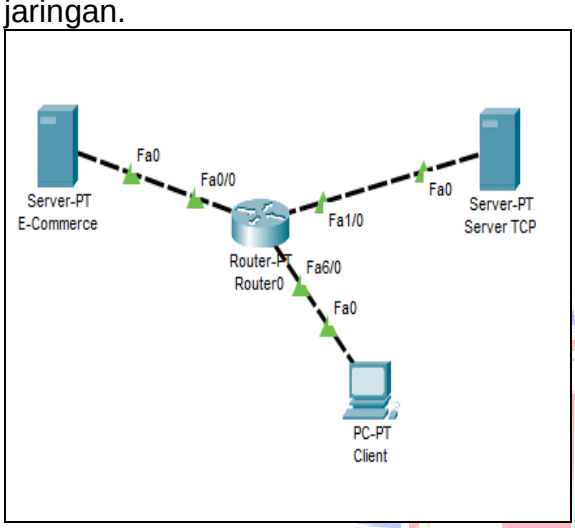
Beberapa Keuntungan Penggunaan TCP [1] seperti Reliabilitas dimana TCP menyediakan pengiriman data yang andal dengan mekanisme pengendalian kesalahan dan retransmisi. Kontrol Aliran dan Kepadatan dimana Mekanisme kontrol aliran dan kepadatan mencegah overload jaringan dan mengatur pengiriman data sesuai kemampuan penerima. Serta Urutan Data dimana TCP menjamin bahwa data diterima dalam urutan yang benar, sehingga aplikasi dapat memproses data secara konsisten. Juga bagaimana bentuk tantangan dalam Implementasi TCP [1] seperti Latensi dan Overhead Proses three-way handshake dan retransmisi dapat menambah latensi dan overhead dalam komunikasi data. Juga Skalabilitas TCP mungkin menghadapi masalah skalabilitas dalam jaringan dengan jumlah pengguna yang sangat b2esar atau dalam lingkungan yang sangat dinamis. Serta Keamanan Meskipun TCP menyediakan komunikasi yang andal, ia rentan terhadap serangan seperti TCP SYN Flooding yang dapat menyebabkan Denial of Service (DoS). Koneksi Program Java dengan Database [1] di Buku Mobul Praktikum dengan memanfaatkan JDBC. Juga tentang Java Network programming [1] serta beberapa referensi online [1], juga tentang TCP/IP Socket in Java [1].

METODE PENELITIAN

Metode penelitian dilakukan dalam skala laboratorium, dimana penelitian ini dilakukan di laboratorium komputer III Jurusan Rekayasa Sistem Komputer Universitas AKPRIND Yogyakarta. Data dan aplikasi yang digunakan dalam penelitian ini adalah data simulasi terdiri (1) sebuah sistem informasi berbasis web yang berjalan di sebuah server web sistem bisnis e-commerce dengan fokus terhadap

MEANS (Media Informasi Analisa dan Sistem)
p-ISSN: 2548-6985, e-ISSN:2599-3089

data produk sebagai pokok penelitian dan memiliki database sebagai sumber data dan ditanam sebuah aplikasi client berbasis java sebagai input dan output yang akan menghubungi sebuah server menggunakan protokol TCP, (2) Server TCP, dimana server tersebut akan mengatur komunikasi antar klien tersebut dan (3) Client TCP yang akan digunakan untuk melakukan simulasi pencarian data pada sistem informasi. Gambar 1 menunjukkan arsitektur jaringan.



Gambar 1. Arsitektur Jaringan

Server TCP, akan menjalankan sebuah script server berbasis TCP , dimana server ini akan mengelola semua server lain dalam melakukan komunikasi datanya.

Server Ecommerce, akan menjalankan aplikasi ecommerce terkait penjualan secara penuh, server ini juga menjalankan script client TCP dengan nama account "**Toko**" yang akan dihubungkan ke Server TCP.

Server Client, akan menjalankan menjalankan script Client TCP yang akan menghubungkan ke server Ecommerce melalui server TCP untuk meminta informasi data produk dengan format tertentu.

HASIL DAN PEMBAHASAN

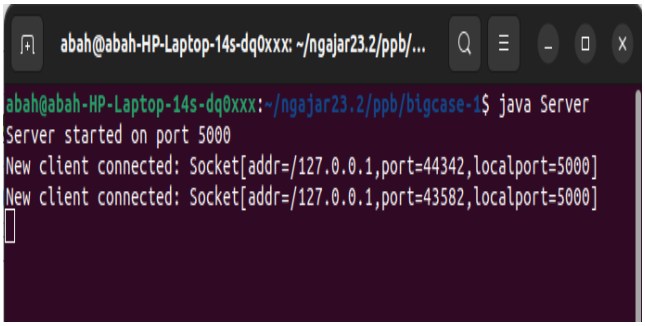
Server TCP

Server TCP bertindak sebagai pengontrol dan pengatur komunikasi antar client, server ini menerima request dari client untuk bergabung/join dan juga meneruskan permintaan komunikasi dari client ke klient lain. Berikut Script dari Server TCP

```
import java.io.*;
import java.net.*;
import java.util.*;
public class Server {
    public ServerSocket server;
    public HashMap<String, Socket> clientList =
new HashMap<>();
    public static void main(String[] args) {
        Server server = new Server();
        server.startServer(5000);
    }
    public void startServer(int port) {
        try {
            server = new ServerSocket(port);
            System.out.println("Server started on
port " + port);

            while (true) {
                Socket client = server.accept();
                System.out.println("New client
connected: " + client);
                Thread clientThread = new
Thread(new ClientHandler(client));
                clientThread.start();
            }
        } catch (IOException e) {
            System.out.println("Error occurred
while starting the server: " + e.getMessage());
        }
    }
    class ClientHandler implements Runnable {
        private Socket clientSocket;
        private DataInputStream inputStream;
        private DataOutputStream outputStream;
        public ClientHandler(Socket socket) {
            this.clientSocket = socket;
            try {
                inputStream = new
DataInputStream(socket.getInputStream());
                outputStream = new
DataOutputStream(socket.getOutputStream());
            } catch (IOException e) {
                System.out.println("Error setting up
input/output streams for client: " +
e.getMessage());
            }
```

```
        }
        @Override
        public void run() {
            try {
                String username = inputStream.readUTF();
                clientList.put(username, clientSocket);

outputStream.writeUTF("#accepted");
                while (true) {
                    String message = inputStream.readUTF();
                    int index = message.indexOf("@");
                    if (index != -1) {
                        String recipient = message.substring(index + 1);
                        Socket recipientSocket = clientList.get(recipient);
                        if (recipientSocket != null) {
                            DataOutputStream recipientStream = new DataOutputStream(recipientSocket.getOutputStream());

recipientStream.writeUTF(username + ": " + message.substring(0, index));
                        } else {

outputStream.writeUTF("Message from Server: No such username found");
                        }
                    } else {

outputStream.writeUTF("Message from Server: Message should include recipient username preceded by @");
                    }
                }
            } catch (IOException e) {
                System.out.println("Error occurred while handling client connection: " + e.getMessage());
            }
        }
    }
}
```

Tampilan server FTP saat running terlihat pada gambar 2. Pada gambar tersebut terdapat 2 clinet yang sudah terhubung ke server .



Gambar 2. Server TCP

Server E-Commerce

Dalam server e-commerce hanya akan dilihat sebuah tabel yang menggambarkan kondisi produk terkini dengan nama tabel barang, seperti terlihat pada script berikut.

```
MariaDB [jualan]> select * from barang;
+----------+--------------+--------+--------+
| idbarang | namabarang   | harga  | jumlah |
+----------+--------------+--------+--------+
|        3 | baju         | 150000 |     99 |
|        4 | celana       | 160000 |     88 |
|        5 | celana pendek |  50000 |     99 |
|        6 | baju batik   | 200000 |     88 |
|        7 | Kaos         | 150000 |      5 |
+----------+--------------+--------+--------+
5 rows in set (0,001 sec)

MariaDB [jualan]>
```

Data hasil penelitian dibuat dalam bentuk Tabel atau Gambar. Sebelum menyajikan Tabel atau Gambar, harus terlebih dahulu membuat narasi sebagai pengantar Tabel atau Gambar (Tidak boleh langsung Tabel atau Gambar tanpa narasi). Setiap Tabel atau Gambar harus ada petunjuk sebelumnya.

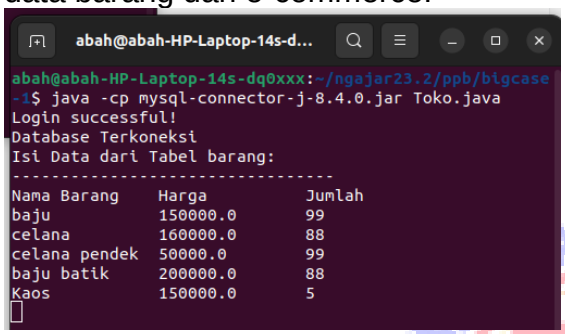
Client e-commerce dalam penelitian ini di beri nama account “Toko” dengan script sebagai berikut:

```
import java.io.*;
import java.net.*;
import java.sql.*;
public class Toko {
    public static void main(String[] args) {
        new Toko().startToko("127.0.0.1", 5000);
    }
public void startToko(String serverAddress, int serverPort) {
```

```
 try (Socket socket = new
Socket(serverAddress, serverPort);
 DataInputStream inputStream = new
DataInputStream(socket.getInputStream());
 DataOutputStream outputStream = new
DataOutputStream(socket.getOutputStream());
 BufferedReader consoleInput = new
BufferedReader(new
InputStreamReader(System.in))) {
 String username = "Toko"; // Set username to
"Toko"
 outputStream.writeUTF(username);
 String response = inputStream.readUTF();
 if (response.equals("#accepted")) {
   System.out.println("Login successful!");
   System.out.println("Database Terkoneksi");
   // Setup database connection
   String databaseURL =
"jdbc:mysql://localhost:3306/jualan";
   String databaseUsername = "ecomer";
   String databasePassword = "zxasa";
   try (Connection connection =
DriverManager.getConnection(databaseURL,
databaseUsername, databasePassword);
   Statement statement =
connection.createStatement()) {
   // Query untuk mengambil data dari tabel
tb_barang
   String query = "SELECT namabarang,
harga, jumlah FROM barang";
   try (ResultSet resultSet =
statement.executeQuery(query)) {
    System.out.println("Isi Data dari Tabel
barang:");
    System.out.println("----------------------------
-");
    String h1="Nama Barang";
    String h2="Harga      ";
    String h3="Jumlah     ";
    int     columnWidth=15;
    System.out.println(String.format("%-" +
columnWidth + "s%-" + columnWidth + "s%-" +
columnWidth + "s", h1, h2, h3));
    while (resultSet.next()) {
     String nama =
resultSet.getString("namabarang");
     double harga =
resultSet.getDouble("harga");
     int jumlah = resultSet.getInt("jumlah");
     // Print data
     System.out.println(String.format("%-" +
columnWidth + "s%-" + columnWidth + "s%-" +
columnWidth + "s", nama, harga, jumlah));
     }} catch (SQLException e)
{System.out.println("Error executing query: " +
e.getMessage());}
Thread readThread = new Thread(() -> {
try {
  while (true) {
    String message = inputStream.readUTF();
    System.out.println(message);
    // Check if the message contains '#'
    if (message.contains("#")) {
      // Find the first word after '#'
      int index = message.indexOf("#");
      if (index != -1 && index + 1 <
message.length()) {
        int nextSpaceIndex =
message.indexOf(" ", index);
        String firstWordAfterHash =
message.substring(index + 1, nextSpaceIndex
!= -1 ? nextSpaceIndex : message.length());

        // Query untuk mencari barang
berdasarkan nama
        String queryBarang = "SELECT
namabarang, harga, jumlah FROM barang
WHERE namabarang LIKE ?";
        try (PreparedStatement
preparedStatement =
connection.prepareStatement(queryBarang)) {
        preparedStatement.setString(1, "%" +
firstWordAfterHash + "%");
        ResultSet resultSetBarang =
preparedStatement.executeQuery();
        // Jika nama barang ditemukan,
tampilkan harga dan jumlah
        while (resultSetBarang.next()) {
        String namabarang =
resultSetBarang.getString("namabarang");
        double harga =
resultSetBarang.getDouble("harga");
        int jumlah =
resultSetBarang.getInt("jumlah");
        outputStream.writeUTF("\n Barang '" +
namabarang + "' ditemukan. Harga: " + harga
+ ", Jumlah: " + jumlah + ". Terima kasih!
@Client");
}} catch (SQLException e) {
outputStream.writeUTF("Automated Reply:
Terjadi kesalahan dalam mencari barang.
Silakan coba lagi nanti. Terima kasih!
@Client");
System.out.println("Error executing query: " +
e.getMessage());
}}}}} catch (IOException e) {
 e.printStackTrace();
}});
 readThread.start();
 while (true) {
  String message = consoleInput.readLine();
  if (!message.contains("@")) {
    System.out.println(String.format("Message
should include recipient username preceded
by @ %s",message));
    continue; // Skip sending message
  }
  outputStream.writeUTF(message);
  }} catch (SQLException e) {
```

```
   System.out.println("Database connection
error: " + e.getMessage());
 }} else {
   System.out.println("Login failed: " +
response);
 }} catch (IOException e) {
  System.out.println("Error occurred while
connecting to server: " + e.getMessage());
}}}
```

Tampilan Client Toko saat berjalan seperti pada gambar 3. Gambar ini menunjukkan data barang yang ada dan ditampilkan saat client Toko dijalan untuk membuktikan bahwa client Toko bisa mengakses data barang dari e-commerce.



Gambar 3. Tampilan Client Toko saat running awal

Client

Client sebagai end user dalam penelitian ini, akan melakukan melakukan koneksi ke Server TCP dan juga akan melakukan komunikasi dengan Client Toko untuk mendapatkan informasi tentang barang yang akan dicari. Berikut Script dari Client.

```
import java.io.*;
import java.net.*;
public class Client {
 public static void main(String[] args) {
   new Client().startClient("127.0.0.1", 5000);
 }
public void startClient(String serverAddress,
int serverPort) {
  try (Socket socket = new
Socket(serverAddress, serverPort);
  DataInputStream inputStream = new
DataInputStream(socket.getInputStream());
  DataOutputStream outputStream = new
DataOutputStream(socket.getOutputStream());
  BufferedReader consoleInput = new
BufferedReader(new
InputStreamReader(System.in))) {
  String username = "Client"; // Set username
to "Client"
  outputStream.writeUTF(username);
  String response = inputStream.readUTF();
  if (response.equals("#accepted")) {
     System.out.println("Login successful!");
Thread readThread = new Thread(() -> {
  try {
  while (true) {
    String message = inputStream.readUTF();
    System.out.println(message);
  }
  } catch (IOException e) {
e.printStackTrace();}
});
readThread.start();
while (true) {
 String message = consoleInput.readLine();
 if (!message.contains("@")) {
   System.out.println("Message should include
recipient username preceded by @");
   continue; // Skip sending message
 }
 outputStream.writeUTF(message);
 }} else {
  System.out.println("Login failed: " +
response);
  }
} catch (IOException e) {
  System.out.println("Error occurred while
connecting to server: " + e.getMessage());
  }}}
```

Client ini dalam penelitian di sederhanakan dengan nama account Client, Gambar 4 menunjukkan saat client di jalankan. Gambar 4 menunjukkan bahwa client sukses join ke server TCP.

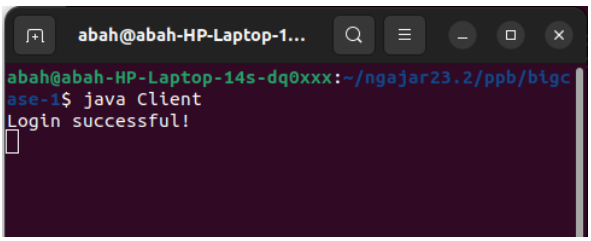
Untuk mengakses client Toko dan mencari informasi tentang produk yang diingingkan menggunakan format sebagai berikut:

Cari #[produk yang dicari] @Toko

Dengan keterangan:

- Cari sebagai kata kunci bahwa akan dilakukan proses pencarian
- # sebagai simbol bahwa data setelah simbol ini adalah produk yang di cari

- [Produk yang dicari] sebagai key pencarian
- @Toko adalah client Toko yang telah running.



Gambar 4. Client saat running

Dalam pengujian dilakukan proses pencarian terhadap data "Celana" seperti paga gambar 5 dengan perintah

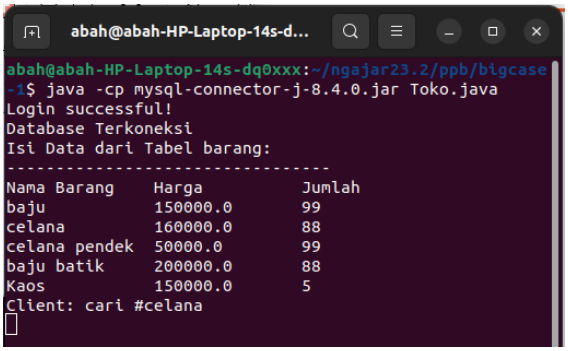
Cari #celana @Toko

Yang artinya client akan mencari data dengan kunci "celana" pada toko, dan hasil yang diperoleh adalah 2 data yaitu "celana" dan "celana pendek". Data ini diperoleh dari @Toko dimana script pada client Toko melakukan pencarian sebagai berikut:

```
String queryBarang = "SELECT namabarang,
harga, jumlah FROM barang WHERE
namabarang LIKE ?";
try (PreparedStatement preparedStatement =
connection.prepareStatement(queryBarang)) {
   preparedStatement.setString(1, "%" +
firstWordAfterHash + "%");
```

Sehingga semua data namabarang yang mengandung kata "celana" akan ditampilkan. Sedangkan pada Client toko akan ditampilkan respon seperti pada gambar 6 dimana Client Toko akan mendapatkan inputstream *Client: cari #celana.*

Gambar 5. Client Melakukan Pencarian Data



Gambar 6, Tampilan Respon Client Toko saat ada pencarian

KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa: (1) Konsep ini membuktikan bahwa komunikasi antar sistem informasi bisa dilakukan menggunakan transmisi protokol protokol (TCP) dengan baik. (2) Komunikasi ini langsung mengakses ke rdbms dari sistem informasi yang ada. (3) Model komunikasi ini bisa digunakan untuk mengelola sistem informasi yang sudah ada dan tersebar dalam berbagai versi dan bentuk, tanpa harus mengganti dan membangun sistem informasi baru untuk menghadapi kebutuhan dan tantangan baru.

UCAPAN TERIMAKASIH

Ucapan terimakasih kepada Universitas AKPRIND Indonesia yang telah memberikan fasilitas laboratorium komputer untuk mengerjakan penelitian ini.

DAFTAR PUSTAKA

[1]. D. Khamis and S. Subair, "Security Framework for Distributed Database System," Journal of Data Analysis and Information Processing, vol. 7, pp. 1-13, 2019.

[2]. J. Triyono, P. Nadira and C. A. Subhkan, "IMPLEMENTASI SISTEM TERDISTRIBUSI MENGGUNAKAN REPLIKASI DATABASE DAN WEB

SERVICE," in Prosiding Seminar Nasional Multidisiplin Ilmu, Yogyakarta, 2021.

[3]. J. Triyono, P. Haryani and A. Padmanaba, "Model Kontrol Transaksi RDBMS Menggunakan Trigger dan Waktu Server," Jurnal Teknologi, vol. 12, no. 1, pp. 80-86, 2019.

[4]. J. Triyono, E. Fatkhiyah, H. I. Ramadhan and N. I. Yatim Fadlan, "Model Aplikasi Terdistribusi Dengan Menerapkan Sinkronisasi Data Searah Melalui FTP Server Dengan Format CSV Studi Kasus Warung Snack KWT Kemuning," in Prosiding Seminar Nasional Teknologi Informasi dan Bisnis, Surakarta, 2022.

[5]. W. Steven, TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994.

[6]. D. Comer, Internetworking with TCP/IP Volume one, Prentice Hall, 2000.

A. &. W. D. Tanenbaum, Computer Networks, Prentice Hall, 2010.

[7]. Hamzah, E. Susanti and S. , MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK LANJUT, Yogyakarta: AKPRIND PRESS, 2018.

[8]. E. R. Harold, Java Network Programming Fourth Edition, -: O'Reilly Media; Fourth Edition (November 12, 2013), 2013.

[9]. "Java Documentation," 01 05 2024. [Online]. Available: https://docs.oracle.com/javase/tutorial/networking/index.html.

[10]. M. J. D. Kenneth L. Calvert, TCP/IP Sockets in Java: Practical Guide for Programmers, Morgan Kaufmann Publishers, 2002.