

# Implementasi Socket Tcp/Ip Untuk Mengirim Dan Memasukan File Text Kedalam Database

Iedam Fardian Anshori

E-mail : iedam.iao@bsi.ac.id

## Abstrak

Aplikasi *socket* umumnya terdiri dari dua kategori berdasarkan pengiriman datanya yaitu: Data gram *socket* (menggunakan UDP) yaitu *stream socket* (menggunakan TCP Penggunaan *socket programming* memungkinkan adanya komunikasi antara *client* dan *server*. Salah satu contoh sederhana penggunaa *socket programming* adalah pembuatan program untuk mengirim dan memasukan *file*. Program tersebut sebenarnya merupakan bentuk aplikasi berupa komunikasi antara *client* dan *server*. Ketika seorang *client* melakukan koneksi ke *server*, program akan membuka koneksi ke *port* yang diberikan, sehingga *server* perlu membuka *socket* pada *port* tersebut dan "mendengarkan" koneksi yang datang. *Socket* sendiri merupakan gabungan antara *host-address* dan *port address*. Dalam hal ini *socket* digunakan untuk komunikasi antara *client* dan *server*. Berdasarkan penelitian, maka dapat disimpulkan bahwa penelitian TCP/IP untuk membuat sistem *database* pengiriman dan *input file*, dapat disimpulkan, Melalui jaringan TCP/IP dengan menggunakan pemrograman Java. dan menggunakan *socket*, *transfer* data bisa ditransmisikan antara program aplikasi dan mengirimkan *file* ataupun *input file*. Dari hasil pengujian yang sudah dilakukan dengan menggunakan program, memberikan kesimpulan bahwa TCP/IP layak sebagai sarana pengiriman data ke aplikasi lain.

**Kata Kunci :** *Socket, client, server, program, port, database, TCP/IP*

## 1. Pendahuluan

Teknologi komputer saat ini berkembang sangat cepat, baik dari sisi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang ada. Karena begitu cepatnya perkembangan teknologi ini, manusia dapat mengakses segala jenis informasi dengan mudah dan cepat meskipun jarak yang berjauhan sehingga dapat meningkatkan efisiensi dan efektifitas dalam melakukan pekerjaan (Raven et al. 2015)

Aplikasi *socket* umumnya terdiri dari dua kategori berdasarkan pengiriman datanya yaitu: *Datagram socket* (menggunakan UDP) yaitu *stream socket* (menggunakan TCP). Walaupun sama-sama berfungsi sebagai protokol pertukaran data, terdapat perlakuan yang berbeda antara UDP dan TCP. UDP tidak memerlukan proses koneksi terlebih dahulu untuk dapat mengirimkan data, paket-paket data yang dikirimkan UDP bisa jadi melalui rute yang berbeda-beda, sehingga hasil yang diterima bisa jadi tidak berurutan. Contohnya jika aplikasi *socket* pengirim mengirimkan berturut-turut pesan 1, pesan 2, dan pesan 3, maka aplikasi *socket* penerima belum tentu mendapatkan pesan yang berurutan dimulai dari pesan 1, pesan 2, dan

terakhir pesan 3. Bisa saja pesan 2 terlebih dulu diterima, menyusul pesan-pesan yang lain, atau berbagai kemungkinan lainnya. Bahkan, dapat terjadi pesan yang dikirimkan tidak sampai ke penerima karena kegagalan pengiriman paket data. Tidak demikian halnya dengan *stream socket* yang menggunakan TCP. Jenis ini mengharuskan terjadinya koneksi terlebih dahulu, kemudian mengirimkan paket-paket data secara berurutan, penerima juga dijamin akan menerima data dengan urutan yang benar, dimulai dari data pertama yang dikirimkan hingga data terakhir. TCP dapat menangani data yang hilang, rusak, terpecah, ataupun terduplikasi. (Listiyono et al. 2009)

Tujuan dari penelitian ini adalah bagaimana mengontrol data *file text* dari jaringan *TCP IP* jarak jauh. Dalam hal ini, kami menggunakan sebuah program yang telah dibangun. Untuk mengatasi masalah ini, kita membutuhkan sebuah program kontrol yang dapat mentransfer data melalui jaringan *TCP/IP*. Untuk mengatasi transfer data ini, kita dapat menggunakan beberapa alat pemrograman untuk membangun program pengendalian. Salah satu yang paling signifikan dari bahasa pemrograman *Java*.

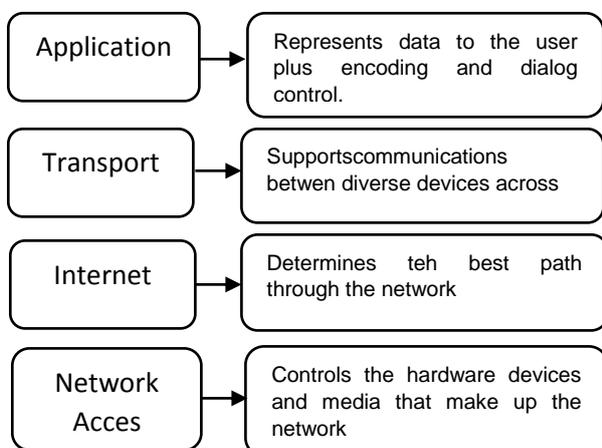
Java memiliki kemampuan untuk berpindah dengan mudah dari satu sistem ke sistem lain dan memiliki pemrograman platform-independen. Dengan menggunakan *socket Java*, kita dapat membangun aplikasi program dalam jaringan *TCP/IP*. Dalam tulisan ini, kami mengusulkan kontrol program untuk mengatur konfigurasi data *file text* melalui jaringan *TCP/IP* dan kami mengadopsi modus *client-server* untuk mengatur saluran transmisi permintaan dan pesan respon antara perangkat. (Sugiarto 2015)

## 2. Metode Penelitian

*TCP/IP (Transmission Control Protocol/Internet Protocol)* adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam suatu jaringan. Prinsip pembagian lapisan pada *TCP/IP* menjadi protokol komunikasi data yang fleksibel dan dapat diterapkan dengan mudah di setiap jenis komputer dan antar-muka jaringan. Oleh karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. (Wardoyo, Ryadi, and Fahrizal 2014)

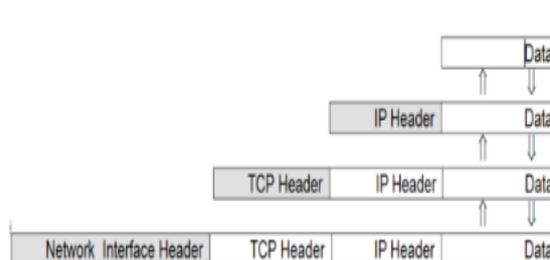
Seperti telah disebutkan sebelumnya, *TCP/IP* berisi kumpulan dari protokol-protokol yang melakukan fungsinya masing-masing secara spesifik. Protokol-protokol ini dikumpulkan berdasarkan fungsinya dalam lapisan-lapisan tertentu. *TCP/IP* memiliki 4 lapisan yang antara satu dengan lainnya memiliki protokol dengan fungsi yang saling melengkapi satu sama lain. Lapisan-lapisan tersebut adalah. (Karim et al. 2017)

### TCP/IP Model



Gambar.1. Layer *TCP/IP*

Jika suatu protokol menerima data dari protokol lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut, informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol lain pada layer di bawahnya. Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada layer di bawahnya. Jika data ini dianggap valid, protokol akan melepas informasi tambahan tersebut untuk kemudian meneruskan data itu ke protokol lain yang berada pada layer di atasnya. (Karim et al. 2017)



Gambar .2. Hubungan Layer *TCP/IP*

### Kerangka Berfikir

Kerangka berfikir dalam metodologi perancangan ini adalah dimulai dengan identifikasi masalah dan perumusan masalah yang dapat dibuat sistem, kemudian menentukan basis pengetahuan dan jenis inferensi yang akan digunakan sehingga dapat di implementasikan pada aplikasi sistem yang dibuat.

### Metode pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan studi pustaka. Teknik pengumpulan data dengan mengadakan studi penelaah terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang ada hubungannya dengan masalah yang dipecahkan. Teknik ini digunakan untuk memperoleh dasar-dasar dan pendapat secara tertulis yang dilakukan dengan cara mempelajari berbagai literatur yang berhubungan dengan masalah yang diteliti. Hal ini juga dilakukan untuk mendapatkan data sekunder yang akan digunakan sebagai landasan perbandingan antara teori dengan prakteknya di lapangan. Data sekunder melalui metode ini diperoleh dengan browsing di internet, membaca berbagai literatur, hasil kajian dari peneliti

terdahulu, catatan perkuliahan, serta sumber-sumber lain yang relevan. (Nazir 2013)

### Protokol TCP (Transmission Control Protocol)

Di dalam penggunaan internet dan secara general jaringan TCP/IP, pengomunikasian setiap aplikasi dengan menggunakan protokol pendukung. Protokol ini bagian di dalam layer transport (transport layer) pada standar OSI yaitu bagian yang memberikan efisiensi dan jaminan komunikasi end-to-end. (Tanenbaum, Hall, and Mercedes n.d.)

TCP adalah protokol yang dapat dipercaya dan dirancang untuk menyediakan alur data pada jaringan internet yang secara umum diketahui dengan kondisi tidak dapat dipercaya serta dirancang untuk beradaptasi dengan peralatan jaringan terhadap berbagai macam permasalahannya. Dirancangnya protokol ini untuk dapat dipercaya maka TCP bersifat *connection oriented* dalam mengirimkan data. TCP menjamin data yang terpercaya dengan menggunakan ARQ (*Automatic Repeat Request*). ARQ akan mentransmisikan secara otomatis berdasarkan informasi gagal diterimanya data ACK (*Acknowledgement*) dari penerima data. Untuk menjamin kontrol efektif terhadap hambatan maka dilakukan dengan cara mengestimasi delay dari transmisi round trip time secara akurat, sehingga dengan mempergunakan informasi balasan dari jaringan tersebut maka dapat mendeteksi sebuah kemacetan jaringan dan menyelesaikannya. Penjelasan TCP dapat ditemui pada RFC 793, 1122 dan 1323.

TCP memiliki tujuh fitur utama yaitu sebagai berikut:

1. *Connection oriented*, aplikasi meminta koneksi dan menggunakannya dalam transfer data.
2. *Point-to-point communication*, setiap koneksi TCP memiliki pasti dua titik.
3. *Reliability*, TCP menjamin bagi data yang dikirimkan dalam koneksi dapat terkirim dengan pasti tanpa ada yang hilang atau double.
4. *Full-duplex connection*, koneksi TCP memperbolehkan data untuk berkoneksi dari salah satu titik koneksi setiap saat.
5. *Stream interface*, TCP memperbolehkan aplikasi untuk mengirimkan koneksi yang berkesinambungan.
6. *Reliable startup*, membutuhkan persetujuan dari kedua aplikasi untuk melakukan koneksi baru.
7. *Graceful shutdown*, aplikasi dapat membuka aplikasi, mengirim data dan

menutup koneksi serta menjamin bahwa data sampai sebelum koneksi terputus. (Tanenbaum et al. n.d.)

### IP address

*IP address* adalah metode pengalamatan pada jaringan komputer dengan memberikan sederet angka pada komputer (*host*), *router* atau peralatan jaringan lainnya. *IP address* sebenarnya bukan diberikan kepada komputer (*host*) atau *router*, melainkan pada *interface* jaringan dari *host/router* tersebut. IP (*Internet protocol*) sendiri di desain untuk interkoneksi sistem komunikasi komputer pada jaringan paket *switched*. Pada jaringan TCP/IP, sebuah komputer diidentifikasi dengan alamat IP.

Tiap-tiap komputer memiliki alamat IP yang unik, masing-masing berbeda satu sama lainnya. Hal ini dilakukan untuk mencegah kesalahan pada transfer data. Terakhir, protokol data akses berhubungan langsung dengan media fisik. Secara umum protokol ini bertugas untuk menangani pendeteksian kesalahan pada saat transfer data, namun untuk komunikasi datanya, IP mengimplementasikan dua fungsi dasar yaitu *addressing* dan fragmentasi.

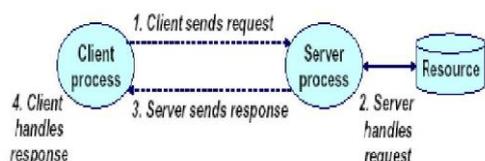
### Port

Salah satu elemen penting yang digunakan dalam aplikasi *socket* adalah *port*. *Port* merupakan sebuah koneksi data *virtual* yang digunakan aplikasi untuk bertukar data secara langsung. Terdapat banyak *port* di dalam sebuah sistem komputer dengan fungsinya masing-masing. Sebagai contoh, dalam mengirim *e-mail* digunakan *service SMTP* yang umumnya menggunakan port 25. Sementara *service POP3* untuk menerima *e-mail* menggunakan port 110, port 80 digunakan untuk *HTTP*, port 443 digunakan untuk *HTTPS*, dan seterusnya.

Nomor-nomor *port* dikategorikan dalam tiga jenis sebagai berikut: **well-known** ports, yaitu merupakan port yang telah digunakan secara internal oleh sistem *Windows*, misalnya port untuk koneksi *Internet*, *service FTP*, dan seterusnya. *Port* yang telah digunakan ini adalah port 0 sampai dengan port 1023. Yang kedua adalah *registered ports*, yaitu port yang dapat digunakan dalam aplikasi yang kita buat, *range*-nya adalah port 1024 hingga port 49151. Ketiga adalah *dynamic/private ports* yaitu port 49152 sampai dengan port 65535.

### Client-server

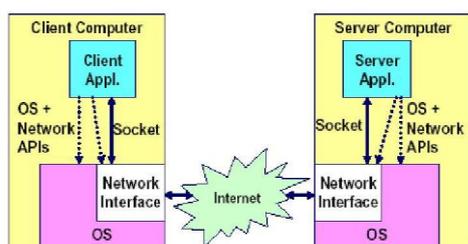
Setiap aplikasi di jaringan, transaksinya didasarkan pada konsep client-server. Sebuah server dan sebuah client atau beberapa client yang meminta/request pelayanan ke server. Fungsi server sebagai pengatur resource yang ada, yang menyediakan pelayanan dengan memanfaatkan resource yang untuk kebutuhan client. Proses ini (proses client-server) dijalankan pada satu komputer berfungsi sebagai server dan sebuah atau beberapa komputer berfungsi sebagai client. Hubungan tersebut dapat digambarkan sebagai berikut :



Gambar 3 Transaksi *Client – server*

Aplikasi *client-server* menggunakan protokol tranport untuk saling berinteraksi. Ketika proses interaksi terjadi, suatu aplikasi harus memberikan informasiinformasi secara detail tentang : informasi tentang apakah dia *client* atau *server*, pengirim memberikan informasi tentang data yang dikirim, penerima memberikan informasi tentang dimana data diletakkan, dll.

Antarmuka (*interface*) antara program aplikasi dengan protokol komunikasi pada suatu sistem operasi disebut Application Program *Interface* (API). API didefinisikan sebagai suatu kumpulan instruksi yang mendukung proses interaksi antara suatu perangkat lunak dengan suatu protokol yang digunakan.



Gambar 4 Aplikasi Jaringan

Pada saat suatu aplikasi berkomunikasi, awalnya aplikasi membuat socket baru, maka pada aplikasi tersebut akan diberikan nomer yang digunakan sebagai referensi socket. Jika ada suatu sistem yang menggunakan nomer referensi socket tersebut, maka akan terjalin suatu jaringan komunikasi antar komputer sebaik transfer data lokal.

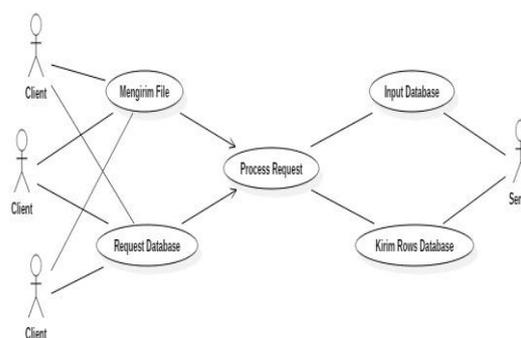
Untuk berkomunikasi antara server dan client harus tahu alamat kelas IP server apakah A, B atau C, seperti contoh di atas dengan menggunakan kelas C, yaitu

192.168.1.1 untuk server dan 192.168.1.2 untuk client, begitu juga nomor port server yang dituju haruslah sama. Disini memakai port 12345.

### 3. Hasil Dan Pembahasan

#### 3.1 UML (Unified Modeling Language)

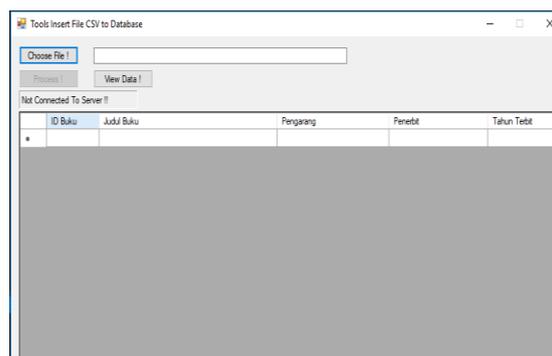
Pada Diagram UML yang meliputi usecase diagram dan activity diagram yang menggambarkan apa saja yang dapat dilakukan oleh *server* dan *client*.



Gambar 5. Use Case Server dan User

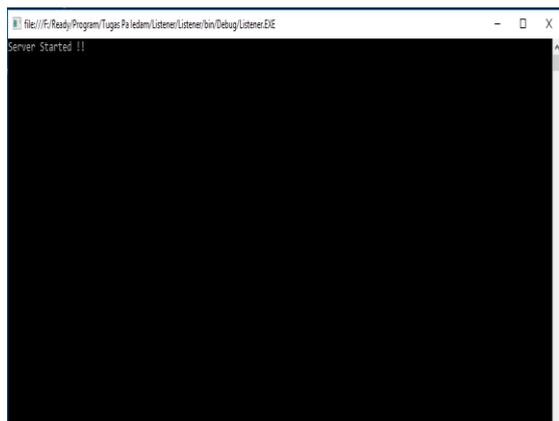
#### 3.2 Penjelasan Program

##### Untuk client



Gambar 6 Tampilan Client

##### Untuk Server



Gambar 7 Tampilan Server

#### 4. Kesimpulan

Berdasarkan penelitian *TCP/IP* untuk membuat sistem *database* pengiriman dan *input file*, dapat disimpulkan:

- Melalui jaringan *TCP / IP* dengan menggunakan pemrograman *Java*. dan menggunakan *socket*, *transfer* data bisa ditransmisikan antara program aplikasi dan mengirimkan *file* ataupun *input file*.
- Dari hasil pengujian yang sudah di lakukan dengan menggunakan program, memberikan kesimpulan bahwa *TCP/IP* layak sebagai sarana pengiriman data ke aplikasi lain.

#### Referensi

Karim, Abdul, Tatang Hidayat Pohan, Awaludin Hasibuan, Elvitrianim Purba, and Sri Trianovie. 2017. "SISTEM PAKAR DIAGNOSA KEGAGALAN KONEKSI *TCP / IP* PADA JARINGAN." 1:138–44.

Listiyono, Hersatoto, Fakultas Teknologi Informasi, Universitas Stikubank Semarang, Internet Protocol, and Local Port. 2009. "PROGRAM SOCKET UNTUK MENGIRIM FILE DENGAN VISUAL BASIC PADA SISTEM OPERASI WINDOWS." 1(1):1–5.

Nazir, Moh. 2013. *No Title*. edited by Metode Penelitian. Ghalia Indonesia.

Raven, Yones, Teady Matius, Surya Mulyana, Program Studi, Teknik Informatika, and Universitas Bunda Mulia. 2015. "IMPLEMENTASI *TCP / IP* UNTUK MEMBUAT SERVER DATABASE ACCESS Yones Raven , Teady Matius Surya Mulyana." 11:44–50.

Sugiarto, Bambang. 2015. "A Remote Control System for Channel Configuration on Weather Station Over *TCP / IP* Network." 2015(Icamimia):76–80.

Tanenbaum, Andrew S., Prentice Hall, and P. T. R. Mercedes. n.d. *Andrew S. Tanenbaum*.

Wardoyo, Siswo, Taufik Ryadi, and Rian Fahrizal. 2014. "Vol : 3 No . 2 September 2014 ISSN : 2302 - 2949 ANALISIS PERFORMA FILE TRANSPORT PROTOCOL PADA PERBANDINGAN METODE *IPv4 MURNI* , *IPv6 MURNI* DAN *TUNNELING 6to4* BERBASIS ROUTER MIKROTIK Jurnal Nasional Teknik Elektro Jurnal Nasional Teknik Elektro." (2):106–17.