PERANCANGAN RELIABILITAS SISTEM TRANSMISI DATA PADA PROTOKOL UDP (*USER DATAGRAM PROTOCOL*)

Wiwin Sulistyo
Program Studi Teknik Informatika
Fakultas Teknologi Informasi Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga, Jawa Tengah 50711
e-mail : wien_soelistyo@yahoo.com

Abstrak
Jaminan pengiriman data melalui jaringan komputer menjadi sesuatu yang sangat penting .Salah satu jaminan yang dibutuhkan olrh user (pemakai) ádalah bahwa data yang dikirimkan ke komputer tujuan sampai dengan baik. Selain itu, terdapat fasilitas yang memberikan keterangan terhadap status pengiriman data kepada user terutama bila pengiriman gagal dilakukan. Sehingga dibutuhkan mekanisme yang berfungsi untuk melakukan kontrol pada saat proses pengiriman data. Protokol UDP merupakan protokol yang bersifat connectionless dan unreliable dalam proses pengiriman data.Dengan menggunakan metode command/request dapat dilakukan perancangan sistem reliabilitas transmisi data pada protokol udp (user datagram protocol).

Keyword *: user datagram protocol (UDP), metode command/request, reliabilitas*

1. PENDAHULUAN

Proses pengiriman data pada jaringan komputer selalu membutuhkan protokol transportasi yang berfungi mengatur pengiriman data antara pengirim dan penerima. Protokol transport tersebut berada pada *layer* ke-4 pada model OSI atau *layer* ke-3 pada model TCP/IP. Terdapat dua protocol utama pada *layer* transportasi yaitu *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP), dimana masing-masing memiliki cara dan karakteristik tersendiri dalam pengiriman data.

Pada protokol TCP, pengiriman data dilakukan dengan berbasiskan *connection oriented* (CNAP, 2003,2006). Dimana proses pengiriman data akan difasilitasi dengan *error control* dan retransmission, sehingga terdapat jaminan bahwa data yang dikirimkan sampai ketujuan dengan baik. Lain halnya dengan protocol UDP, dimana pengiriman data tidak difasilitasi dengan *error control* seperti yang ada pada protocol TCP. Sehingga pengiriman data dengan menggunakan protokol transportasi UDP tidak ada jaminan atau tidak menyediakan mekanisme yang ada pada dirinya bahwa data telah sampai ditujuan dengan baik, sehingga protocol tersebut sering disebut dengan *connectionless oriented* (CNAP, 2003,2006).

Meskipun demikian, protokol UDP memiliki keunggulan dalam hal kecepatan pengiriman data. Sehingga menjadi sesuatu yang baik bila protokol UDP tersebut memiliki kemampuan untuk mengontrol proses pengiriman data sendiri, sehingga kehilangan sebagian paket data dapat diatasi pada saat pegiriman data. Oleh sebab itu, pada penelitian ini difokuskan pada bagaimanakah membangun reliabilitas pengiriman data pada protokol UDP dengan menggunakan metode *command/request*, sehingga protokol tersebut memiliki kemampuan sendiri untuk mengontrol pengiriman paket-paket data pada jaringan komputer.

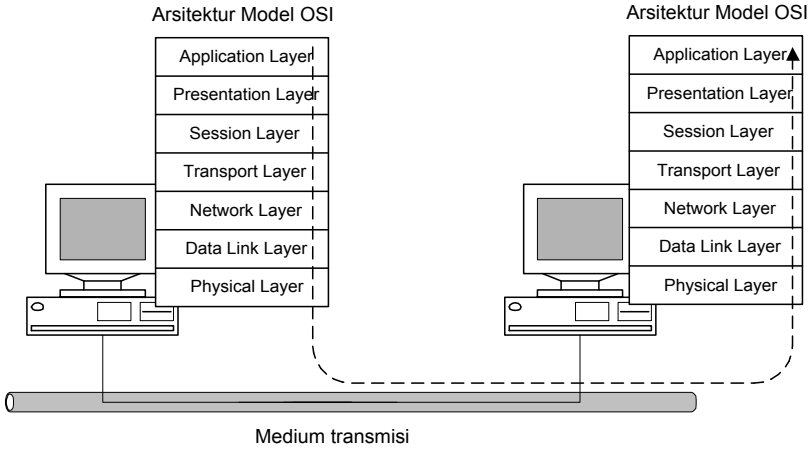
2. TINJAUAN PUSTAKA

Arsitektur OSI

Model OSI (*Open System Interconnection*) dikembangkan oleh ISO (*International Organization for Standardization*) sebagai model untuk arsitektur komunikasi komputer, serta sebagai kerangka kerja bagi pengembangan standar-standar protokol (William Stallings, 2000). Model OSI terdiri dari tujuh lapisan, yang masing-masing lapisan memiliki fungsi sendiri-sendiri (seperti dijelaskan pada tabel 1). Dalam proses transmisi data antara sumber ke tujuan dapat digambarkan dalam proses koneksi dari lapisan aplikasi sampai dengan lapisan fisik dan ketika sampai ketujuan maka akan terjadi aliran proses dari lapisan fisik sampai dengan lapisan aplikasi (seperti pada gambar 1).

Tabel 1. Lapisan dan fungsi masing-masing lapisan pada model OSI

Lap. Aplikasi	Aplikasi dan User Interface
Lap. Presentasi	• Presentasi data : HTML, ASN1, JPEG, MPEG • Proses Enkripsi
Lap. Session	• Memelihara koneksi untuk aplikasi yang terpisah
Lap. Transport	• Menentukan penyampaian data yang *realible / unreliable* • Proses koreksi error sebelum retransmit
Lap. Network	• Menyediakan *logical Addressing* • Menetapkan lintasan (*routing*)
Lap. Data Link	• Assembly/disassembly paket ke *frame*/*frame* ke paket • Mengatur akses kejaringan (Media) dengan mekanisme pengalamatan MAC • Menyediakan deteksi error
Lap. Fisik	• Bertanggungjawab menyampaikan aliran bit dari piranti ke piranti



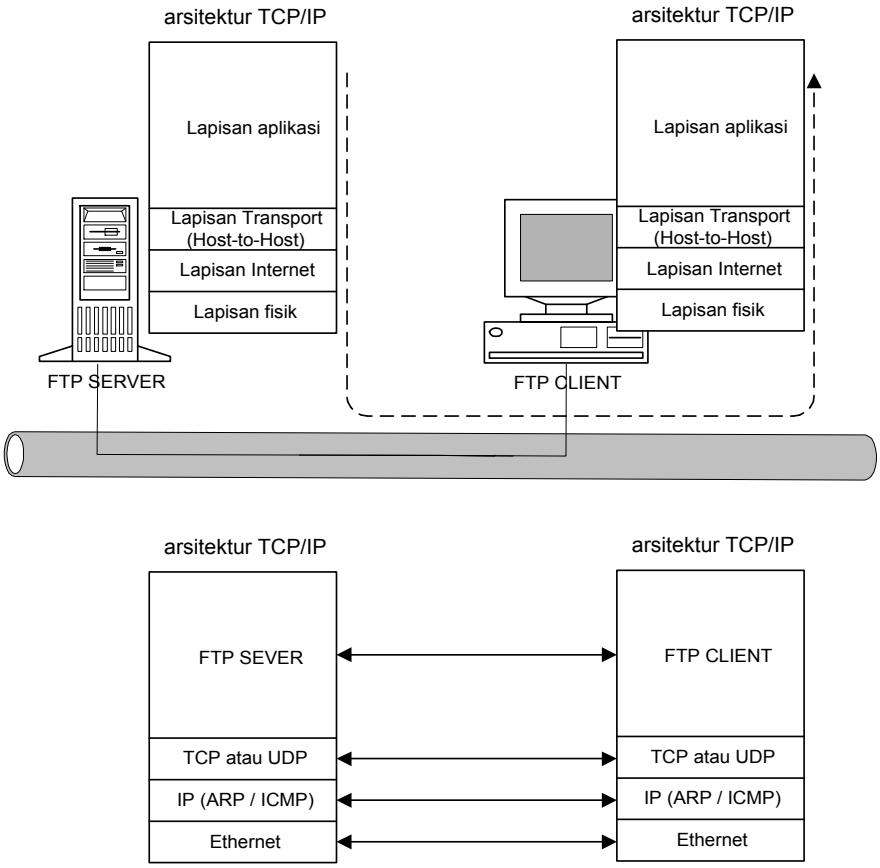
Gambar 1. Model komunikasi pada arsitektur model OSI

Arsitektur Protokol TCP/IP

TCP/IP merupakan arsitetur standar yang banyak digunakan dan OSI telah menjadi model standar untuk mengklasifikasikan fungsi-fungsi komunikasi. TCP/IP merupakan hasil dari pengembangan dari riset protokol yang dilakukan atas jaringan paket *switching*, ARPANET, dan didanai oleh DARPA (*Defense Advanced Rasearch Project Agency*) dan secara umum dikenal sebagai TCP/IP *protocol suite* (William Stallings, 2000) (David Groth, 2001). Protocol suite ini terdiri dari protokol-protokol atau jajaran-jajaran protokol yang bekerja bersama-sama dalam bagiannya masing-masing (fungsi setiap lapis seperti pada tabel 2). Dalam proses transmisi data antara sumber ke tujuan dapat digambarkan dalam proses koneksi antar aplikasi yang masing masing menggunakan protocol TCP/IP (seperti pada gambar 2).

Tabel 2. Fungsi masing-masig lapisan pada protokol TCP/IP

Lap. Aplikasi	Aplikasi, Proses dan User Interface
Lap. Transport	• Menentukan penyampaian data yang realible / unreliable • Proses koreksi error sebelum retransmit
Lap. Internet	• Menyediakan logical Addressing • Menetapkan lintasan (routing)
Lap. Fisik (Network)	• Assembly/disassembly paket ke frame/frame ke paket • Mengatur akses kejaringan (Media) dengan mekanisme pengalamatan MAC • Menyediakan deteksi error • Bertanggungjawab menyampaikan aliran bit dari piranti ke piranti



Gambar 2. Koneksi antar Aplikasi pada TCP/IP

User Datagram Protocol (UDP)

UDP, singkatan dari ***User Datagram Protocol***, adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP. Protokol ini didefinisikan dalam *RFC 768*. UDP memiliki karakteristik-karakteristik berikut (wikipedia, 2009):

- *Connectionless* (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berukar informasi.
- *Unreliable* (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan acknowledgment. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.

- UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. *Header* UDP berisi *field* Source Process Identification dan Destination Process Identification.
- UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP.

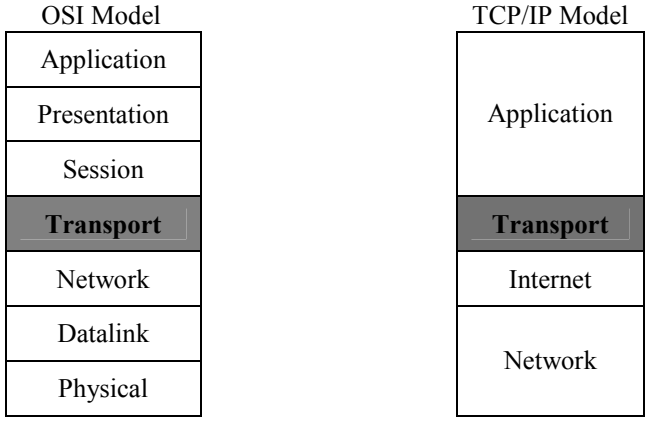
UDP tidak menyediakan layanan-layanan antar-host berikut:

- UDP tidak menyediakan mekanisme penyanggaan (*buffering*) dari data yang masuk ataupun data yang keluar. Tugas *buffering* merupakan tugas yang harus diimplementasikan oleh protokol lapisan aplikasi yang berjalan di atas UDP .
- UDP tidak menyediakan mekanisme segmentasi data yang besar ke dalam segmen-segmen data, seperti yang terjadi dalam protokol TCP. Karena itulah, protokol lapisan aplikasi yang berjalan di atas UDP harus mengirimkan data yang berukuran kecil (tidak lebih besar dari nilai Maximum Transfer Unit/MTU) yang dimiliki oleh sebuah antarmuka di mana data tersebut dikirim. Karena, jika ukuran paket data yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan bisa saja terpecah menjadi beberapa fragmen yang akhirnya tidak jadi terkirim dengan benar.
- UDP tidak menyediakan mekanisme *flow-control*, seperti yang dimiliki oleh TCP.

3. METODE PENELITIAN

Pengiriman Data Pada Protokol UDP Dengan Metode *command/request*

Pada prinsipnya pengiriman data pada jaringan komputer membutuhkan protokol transportasi. Protokol transportasi dalam proses pengiriman data memiliki peranan, antara lain: membangun komunikasi antara aplikasi sumber dan tujuan, segementasi dan *reasembling* data serta mengidentifikasi komunikasi pada aplikasi yang berbeda. Kalau dilihat posisi transport layer pada Model OSI atau TCP/IP adalah seperti pada gambar berikut ini.



Gambar 3. Posisi *Transport Layer* pada Model OSI dan TCP/IP

UDP diciptakan untuk menangani permasalahan batasan pesan pada protokol TCP. Hal ini dikarenakan UDP merupakan protokol transport yang dirancang bukan untuk difokuskan pada masalah reliabilitas transportasi data, sehingga tidak membutuhkan *buffer* lokal untuk menyimpan data yang dikirimkan maupun yang dipesan. Dimana setiap pesan akan dikirimkan dalam sebuah paket tunggal dari sebuah program aplikasi. Begitu juga, pada setiap pesan yang diterima dari jaringan akan diterusakan ke program aplikasi sebagai pesan tunggal (*single message*). Selain itu tidak terdapat respon dari computer *client* yang mengisyaratkan bahwa data telah diterima dengan baik. Sehingga kehilangan data pada saat pengiriman menjadi tidak dapat dideteksi oleh pengirim.
Oleh sebab itu dibutuhkan pengembangan pada protokol UDP untuk melengkapi kelemahan-kelemahan yang dimiliki. Salah satunya adalah dengan memanfaatkan metode *command/response* antara *server* (*sender*) dengan *client* (penerima). Secara umum penerapkan metode *command/response* untuk pengiriman data dengan UDP melalui 4 tahap (Richard Blum, 2003):

1. Mengirimkan data ke *remote device*.
2. Mulai aktifkan *timer*, set dalam periode waktu tertentu.
3. Menunggu respon dari *remote device*. Ketika respon sampai, *stop timer* dan lanjutkan dengan program berikutnya.
4. Jika *timer* telah lewat atau *expire* sebelum terjadi respon dari *remote device*, maka kembali ke tahap 1. Setelah terjadi perulangan dalam pada langkah 1 dalam jumlah tertentu tanpa ada respon, maka diasumsikan bahwa telah terjadi kegagalan komunikasi dengan remote host.

Dengan tahapan proses diatas, maka dapat dirancang proses pengiriman data dengan protokol UDP dengan memperhatikan aspek reliabilitas yang sebelumnya tidak terdapat pada protokol UDP itu sendiri.

4. HASIL DAN PEMBAHASAN

Perancangan Sistem

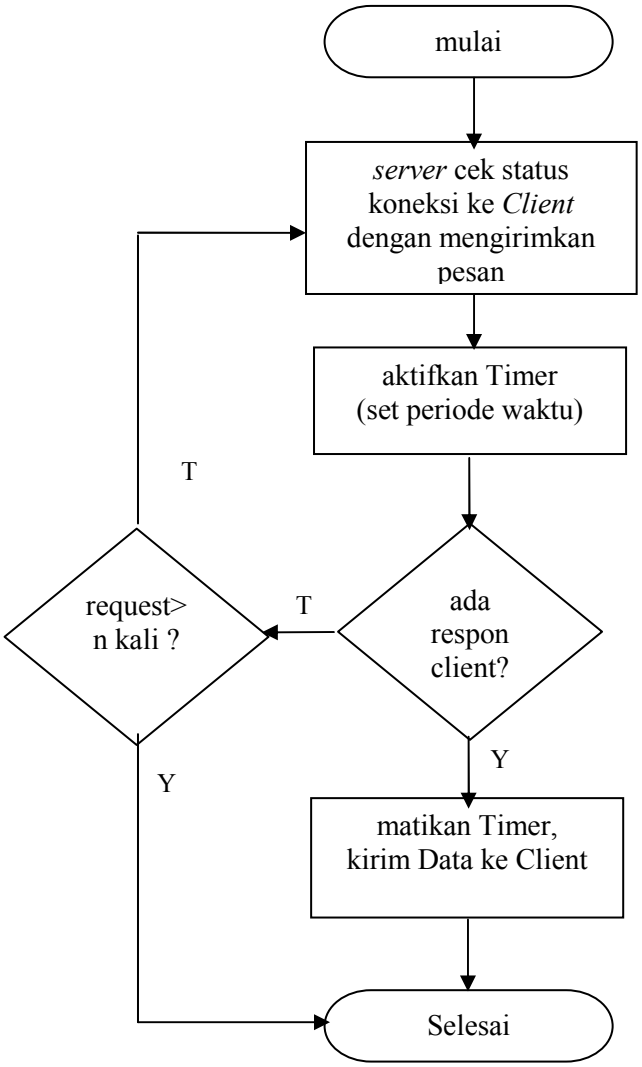
Dengan memanfaatkan metode *command/response* antara *server* (*sender*) dengan *client* (penerima), maka dapat dirancang proses pengiriman data dengan protokol UDP antara *server* dan *client* yang memperhatikan aspek reliabilitas. Sehingga terdapat tahapan proses atau prosedur komunikasi, seperti berikut ini :

1. *Sender* (*server*) mengirimakan pesan untuk memeriksa status koneksi *receiver* (*client*) dan permintaan pengiriman data, sambil mengaktifkan *timer*.
2. Jika tidak ada respon dari *client* sampai periode *timer* habis, ulangi tahap 1, dan bila tetap tidak ada respon maka diasumsikan *client* tidak aktif atau terjadi kegagalan koneksi.
3. Jika *client* berhasil menerima pesan dari server maka akan memberikan respon positif ke *server*, sekaligus mengaktifkan *timer* dengan periode waktu tertentu.
4. *Server* menangkap respon dari *client*, selanjutnya melakukan pengiriman data.
5. Jika *client* tidak menerima data sampai periode waktu tertentu (sesuai *timer*), maka *client* akan mengirim pesan ke server untuk meminta pengiriman data, sambil mengaktifkan *timer* kembali, dan kembali pada tahap ke-4. Bila tidak ada respon dari *server* maka diasumsikan terjadi kegagalan koneksi.

Dalam implementasinya maka dapat dirancang sebuah proses pengiriman data antara server (pengirim) dan client (penerima) yang masing-masing seperti dijelaskan pada gambar 4, 5, 6.

Sisi *Server*

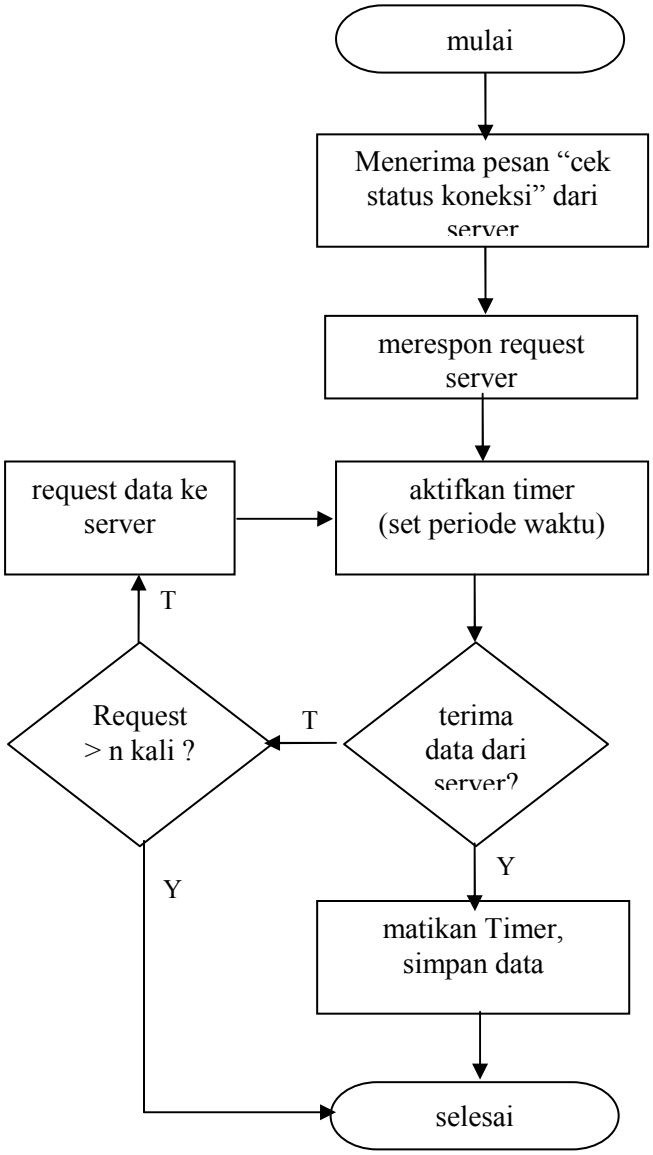
Pada sisi *server* terdapat proses yang pertama, dimana *server* akan melakukan pemeriksaan status koneksi ke *client* sebelum melakukan pengiriman data. Status koneksi ke client akan ditandai dengan respon dari *client* sebelum periode waktu/timer habis. Bila sampai batas waktu tidak terjadi respon dari *client*, maka *server* akan melakukan pemeriksaan status koneksi lagi ke *client* dan memulai *set timer* dari awal dan bila tidak terdapat jawaban dari *client* sampai dengan batas waktu habis, maka stelah dilakukan pengulangan sampai N kali dan tidak ada respon dari *client* maka diasumsikan bahwa telah terjadi kegagalan koneksi (tidak ada koneksi). Tetapi bila status koneksi positif (menerima respon dari *client*), maka selanjutnya server akan melakukan pengiriman data ke *client* sambil mematikan *timer*. Proses pada sisi *server* tersebut dapat dlihat seperti pada gambar 4.



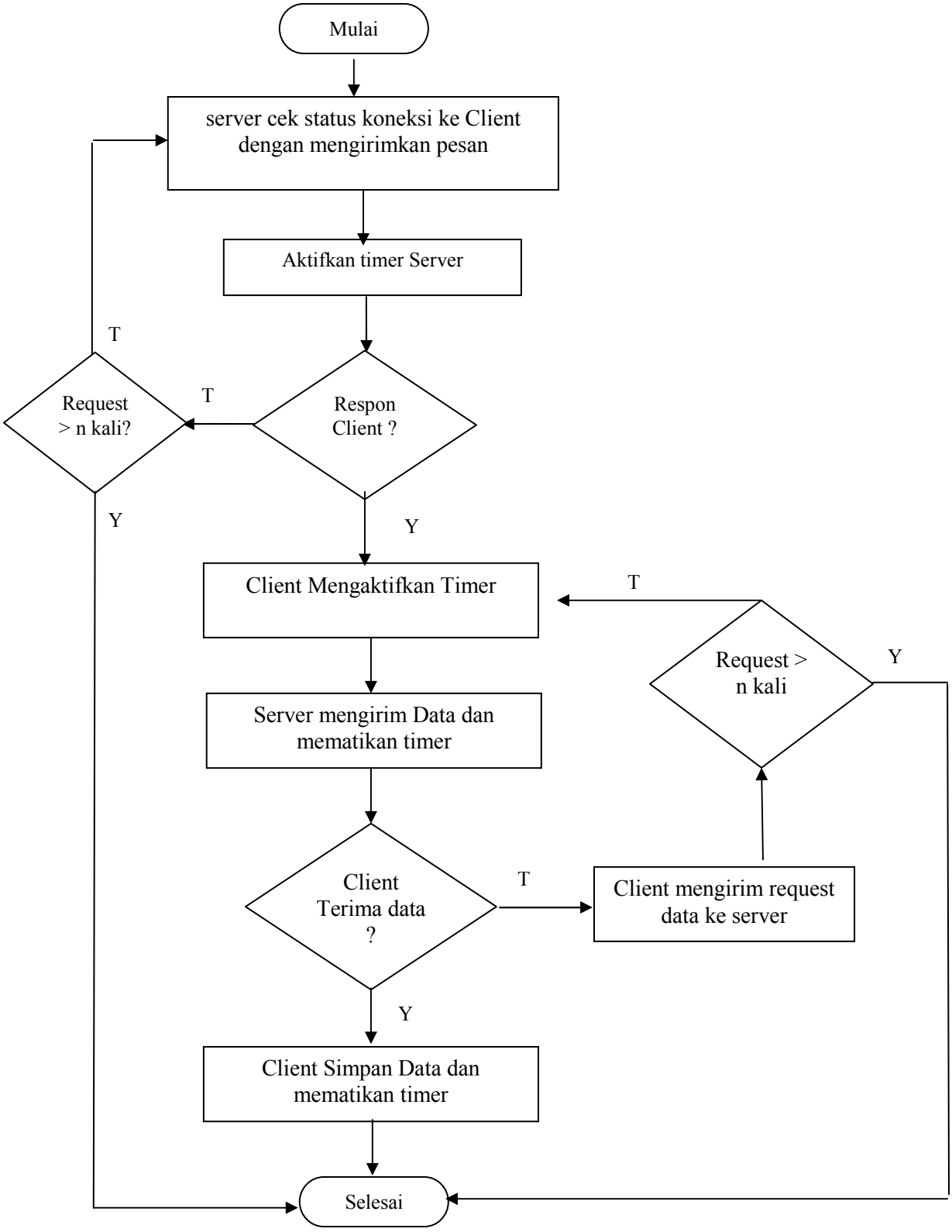
Gambar 4. Proses disisi *server*

Sisi *Client*

Dari sisi *client* akan terjadi proses setelah menerima pesan cek status dari *server* yang akan direspon oleh client yang menyatakan bahwa *client* siap untuk menerima data dari *server*. Selanjutya *client* akan menunggu transmisi data dari *server* sambil mengaktifkan timer untuk periode waktu tunggu pengiriman data dari *server*. Bila sampai batas periode waktu habis data belum diterima oleh *client* maka diasumsikan terjadi kegagalan transmisi data dari *server* sehingga *client* akan melakukan permintaan ulang pengiriman data ke server sampai sejumlah N kali, sampai akhirnya *client* mengakhiri prosesnya dengan mengasumsikan bahwa koneksi ke server gagal. Tetapi bila *client* telah menerima data sebelum batas periode waktu habis, maka selanjutnya client akan menyimpan data tersebut. Proses pada sisi *client* tersebut dapat dlihat seperti pada gambar 5. Secara lengkap pengiriman data antara *server* dengan client, baik dari sisi *server* maupun dari sisi *client* dapat dilihat pada gambar 6.



Gambar 5. Tahapan proses disisi client



Gambar 6. Tahapan proses transmisi data secara menyeluruh

5. KESIMPULAN

Dengan menggunakan metode *command/request* maka telah berhasil dibangun perancangan dalam bentuk tahapan proses transmisi data antara *server* dengan *client*, maka aspek reliabilitas pengiriman data dengan protokol UDP terletak pada sisi server sebagai pengirim data (*sender*) dan *client* sebagai *receiver* (penerima) data. Pada sisi *server* akan melakukan cek status

koneksi dengan *client* sebelum melakukan pengiriman data. Sedangkan dari sisi *client*, setelah menerima pesan cek status koneksi dari *server*, maka client akan memberikan respon ke *server* dan menunggu pengiriman data dari *server*. Bila data sampai batas periode waktu tidak datang maka *client* akan berinisiatif untuk melakukan permintaan ulang pengiriman data ke *server*. Dilain pihak, setelah *server* melakukan pengiriman data ke *client* dan tidak ada permintaan ulang data oleh *client*, maka diasumsikan data telah diterima dengan baik oleh *client*. Tetapi perlu juga diperhatikan disini bahwa proses reliabilitas yang diterapkan pada perancangan ini, akan membutuhkan konsumsi waktu pengiriman data dengan protokol UDP lebih besar. Oleh sebab itu, perlu dilakukan tahap implementasi dari perancangan yang telah dibuat untuk menganalisa aspek efektifitas maupun efisiensi dari perancangan tersebut.

6. DAFTAR PUSTAKA

Cisco Network Academy Program, 2008, *ver.*40, *Cisco Esploration 4.0*, Cisco Networking Academy Program (CNAP).

Cisco Network Academy Program, 2006, *ver.* 3.1, *Cisco Fundamental 3.1*, Cisco Networking Academy Program(CNAP).

David Groth, 2001, *Network+ Study Guide*, Sybex.

Richard Blum, 2003, *C# Network Programming*, Sybex.

William Stallings, 2000, *Data and Computer Communication,* Prentice-Hall, Inc.

Wikipedia online http://www.wikipedia.com, 2009.