

Testbed Komunikasi Terintegrasi untuk Angkutan Massal Cepat Surabaya

Tiffany Maliati Khumairoh Afandi, Dr. Ir. Achmad Affandi, DEA, dan Ir. Djoko Suprajitno R, MT
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Kampus ITS, Keputih-Sukolilo, Surabaya 60111 Indonesia
e-mail: affandi@ee.its.ac.id, djokosr@ee.its.ac.id, tiffany.maliati12@mhs.ee.its.ac.id

Abstrak—Pemanfaatan teknologi informasi sebagai sarana pengelolaan Kota Surabaya dapat meningkatkan keefektifan pelayanan publik. Salah satu caranya adalah dengan penggunaan *On-Board Unit* (OBU) pada Angkutan Massal Cepat yang akan diterapkan di Kota Surabaya. OBU adalah terminal elektronik yang dapat memberikan informasi mengenai identitas dan posisi AMC, dan menghubungkan komunikasi antara CC-ROOM dengan pengemudi. Tugas akhir ini bertujuan untuk merancang sebuah sistem yang digunakan pada *testbed* komunikasi yang terintegrasi antara OBU (*client*) dengan CC-ROOM (*server*).

Perancangan dan implementasi dilakukan dalam 2 tahap, yaitu membuat arsitektur jaringan dan aplikasi sistem *client-server* yang digunakan untuk sistem komunikasi terintegrasi AMC. Aplikasi pengiriman data dari *client* ke *server* dibuat dengan menggunakan *Borland Delphi 7*. Jaringan yang digunakan yaitu jaringan seluler dan jaringan *backbone*.

Pengujian sistem komunikasi terintegrasi AMC dilakukan dengan 2 macam pengujian, yaitu pengujian sistem *client-server* dan pengujian jaringan sistem yang terdiri dari *packet loss* dan *throughput*. Dari hasil pengujian *packet loss* dan *throughput* didapatkan rekomendasi ukuran data yang dapat diterapkan pada sistem komunikasi terintegrasi angkutan massal cepat Surabaya, yaitu sebesar 450 byte.

Kata kunci—*Intelligent Transportation System, Packet Loss, Testbed, Throughput*

I. PENDAHULUAN

KOTA Surabaya mengalami peningkatan jumlah kendaraan pribadi yang besar dari tahun ke tahun sehingga dapat menyebabkan kepadatan lalu lintas di ruas jalan raya Surabaya. Salah satu penyebabnya adalah kurangnya moda transportasi publik yang memadai. Badan Perencanaan Pembangunan Nasional (Bappenas) telah memperkirakan Angkutan Massal Cepat (AMC) Kota Surabaya beroperasi pada tahun 2017. Proyek AMC Kota Surabaya telah masuk dalam Rancangan Program Jangka Menengah Nasional (RPJMN) Bappenas tahun 2015-2019. Hal ini menunjukkan bahwa pemerintah Kota Surabaya telah berkomitmen untuk membangun sebuah Angkutan Massal Cepat (AMC) untuk mengurangi kemacetan lalu lintas di Kota Surabaya.

Angkutan Massal Cepat (AMC) memiliki sistem komunikasi terintegrasi dengan wujud pemasangan *On-Board Unit* (OBU) pada setiap armada dan sistem *Passenger Information Board* (PIB) pada halte. Data dari OBU akan melewati jaringan seluler sedangkan data dari PIB akan melewati jaringan *backbone*. Kemudian data tersebut akan diterima oleh *Control Center Room* (CC-ROOM). Untuk

menguji pengiriman data di perlukan suatu *testbed* terhadap *prototype* dari sistem tersebut.

Oleh karena itu perlu dibuat sebuah aplikasi yang mampu mensimulasikan *testbed* pada sejumlah titik OBU dan PIB. Dalam tugas akhir ini, aplikasi dibuat menggunakan *software Borland Delphi 7*. Aplikasi ini bertindak sebagai *client* yang memasukkan data untuk dikirim ke *server*. *Server* akan memberi balasan ke *client* dan menyimpan data dalam bentuk *database server*. Data yang disimpan oleh *server* antara lain identitas, lokasi, waktu kirim, *ip* pengirim, waktu terima, dan nomer paket. Data tersebut digunakan untuk mengetahui *packet loss* dan *throughput*.

II. DASAR TEORI

A. Angkutan Massal Cepat Surabaya

Angkutan massal cepat Surabaya biasa disebut sebagai Surotram dan Boyorail. Kedua jenis moda transportasi ini menghubungkan Surabaya bagian timur-barat dan utara-selatan. Surotram menghubungkan Surabaya bagian utara sampai Surabaya bagian selatan dengan total jarak $\pm 16,7$ Km dan melewati 26 titik halte. Daerah yang dilewati oleh Surotram yaitu Sonokembang, Bambu Runcing, Gub. Suryo, Tunjungan, Genteng, Siola, Baliwerti, Tugu Pahlawan, Veteran, Jembatan Merah, Rajawali, Indrapura, Kemayoran, Pasar Turi, Bubutan, Pasar Blauran, kedungdoro, Embong Malang, Tegalsari, Kombepol M Duryat, Panglima Sudirman, Pandegiling, Bintoro, Taman Bungkul, Bonbin, dan Joyoboyo. Boyorail menghubungkan Surabaya bagian timur sampai bagian barat dengan total jarak ± 23 Km dan melewati 24 titik halte. Daerah yang dilewati oleh Boyorail yaitu Kejawan, Mulyosari ITS, Gor Kertajaya, Dharmahasada Indah Timur, RS DR Sutomo, Stasiun Gubeng, Jl. Raya Gubeng, Irian Barat, Bung Tomo, Ngagel, Wonokromo, Joyoboyo, Adityawarman, Pakis, Dukuh Kupang, Bundaran Satelit, HR Muhammad, Simpan Darmo Permai, Lontar, Unesa, dan Lidah Kulon.

B. Sistem Client-Server

Client-server merupakan teknologi pendistribusian kerja aplikasi antara dua komputer atau lebih yang dihubungkan oleh jaringan komunikasi, dimana yang satu akan bertindak sebagai *server* atau pemberi layanan dan yang lainnya sebagai *client* atau peminta layanan. Baik *client* maupun *server* memiliki pemroses atau CPU sendiri sedangkan jaringan yang digunakan bisa berupa jaringan lokal (LAN) ataupun jaringan yang lebih luas (WAN) [1].

C. Borland Delphi 7

Delphi adalah sebuah bahasa pemrograman yang bersifat OOP (*Object Oriented Programming*), artinya sebuah program yang mempunyai objek-objek tertentu dalam pemrogramannya. OOP ini memiliki beberapa unsur yaitu: *Encapsulation* (pemodelan), *Inheritance* (penurunan), *Polymorphism* (polimorfisme). Selain itu, Delphi adalah sebuah program yang bersifat visual, artinya mempunyai tampilan grafik-grafik yang mudah dimengerti oleh pemula sekalipun (*Graphical User Interface*) [3].

D. Packet Loss

Packet loss merupakan banyaknya paket yang gagal mencapai tempat tujuan. Ketika packet loss besar maka dapat diketahui bahwa jaringan sedang sibuk atau terjadi overload Dinyatakan dalam persen (%) [4].

$$Packet\ loss\ rate = \left(\frac{total\ packet\ loss}{total\ packet\ sent} \right) * 100\% \quad (1)$$

E. Throughput

Throughput diartikan sebagai laju data aktual per satuan waktu (*bits per second*). Biasanya throughput selalu dikaitkan dengan bandwidth. Karena throughput memang bisa disebut sebagai bandwidth dalam kondisi yang sebenarnya. Bandwidth lebih bersifat tetap sementara throughput sifatnya dinamis tergantung trafik yang sedang terjadi.

Cara untuk menghitung throughput sebagai berikut:

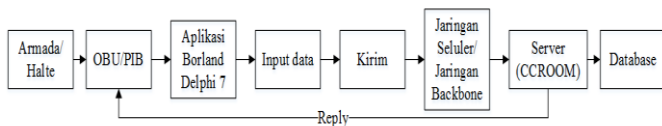
$$throughput = \frac{jumlah\ data\ yang\ diterima}{waktu\ pengiriman} \quad (2)$$

F. PHP MyAdmin

Salah satu perangkat lunak yang digunakan untuk mengelola database dalam MySQL adalah PHPMyAdmin. Dengan PHPMyAdmin kita dapat dengan mudah membuat tabel, mengisi data, dan banyak lagi hal lainnya yang tanpa harus hafal perintahnya, namun cukup dengan mengisi tabel-tabel yang telah tersedia [2].

III. METODOLOGI

Pada tugas akhir ini membahas perancangan testbed sistem client-server komunikasi terintegrasi Angkutan Massal Cepat Surabaya, dimana pembagiannya meliputi perancangan arsitektur jaringan, perancangan aplikasi, dan perancangan database. Dalam pembuatan sistem ini pertama kali dilakukan adalah mengetahui proses komunikasi terintegrasi AMC. Berikut ini adalah diagram alur sistem yang ditunjukkan dalam Gambar 1:



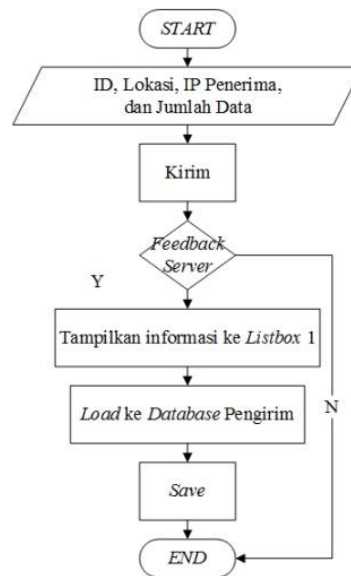
Gambar 1 Diagram blok sistem jaringan komunikasi AMC

On-Board Unit (OBU) berada pada tiap armada sedangkan Passenger Information Board (PIB) berada pada halte. Pada OBU dan PIB akan di install aplikasi yang telah dibuat menggunakan software Borland Delphi 7. Untuk mengirimkan

data ke server (CCROOM), client harus memasukkan beberapa data sesuai dengan kolom yang terdapat pada aplikasi. Kemudian data dikirim ke server. Jika pengiriman data menggunakan OBU, maka data dikirim melalui jaringan seluler. Jika pengiriman data menggunakan PIB, maka data dikirim melalui jaringan backbone. Setelah data diterima oleh server, server akan memberi balasan ke client. Selanjutnya data dimasukkan ke database untuk diolah.

A. Perancangan Aplikasi pada Borland Delphi 7

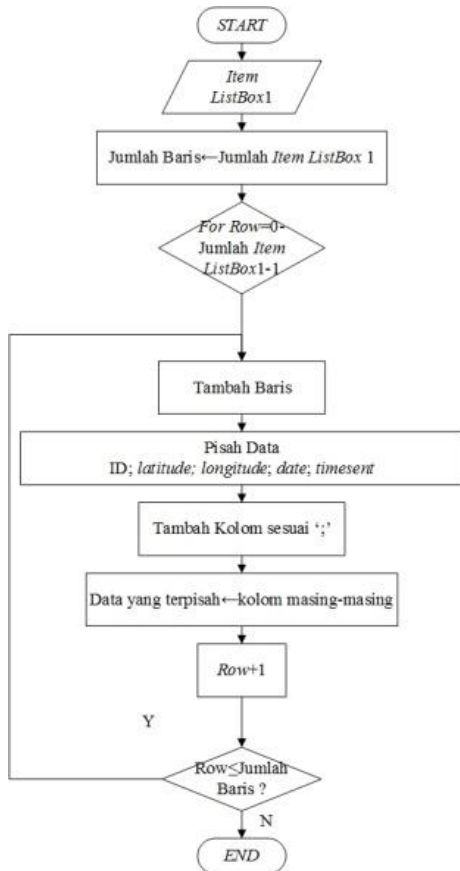
Aplikasi yang dibuat telah dirancang berfungsi sebagai client. Berikut flowchart aplikasi client Borland Delphi 7:



Gambar 2 Flowchart aplikasi Borland Delphi

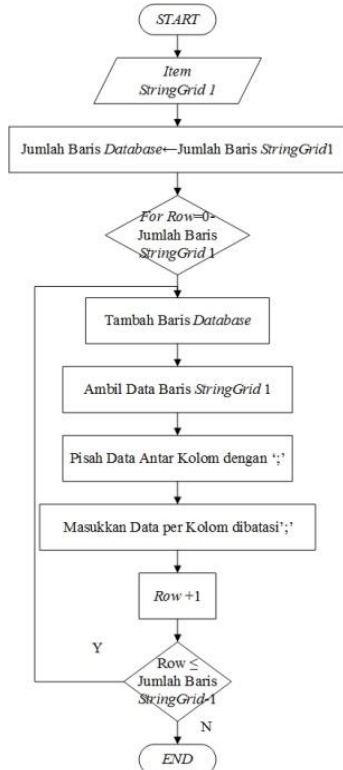
Dapat dilihat pada Gambar 2 mengenai alur aplikasi untuk uji coba. Client koneksi ke jaringan internet. Client memasukkan atau meng-inputkan data ID, Lokasi, IP penerima, dan jumlah data pada aplikasi Borland Delphi 7 yang telah dibuat. Kemudian data-data tersebut dikirim ke server. Server telah terhubung dengan database PHPMyAdmin. Client akan mendapat feedback (umpan balik) setelah data terkirim ke server. Feedback tersebut berupa informasi yang muncul pada ListBox 1. Informasi yang didapat yaitu balasan dari server berupa ID, lokasi, tanggal pengiriman, dan waktu pengiriman per paket data. Balasan dari server dapat di konversi ke database dengan format CSV (*Comma Separated Value*). Caranya adalah dengan menekan “Database Pengirim”, maka balasan dari server tersebut akan di muat terlebih dahulu dalam bentuk tabel yang ditampilkan pada StringGrid 1.

Adapun penjelasan proses load data ke StringGrid1 Pertama, data dari ListBox 1 diolah untuk ditampilkan dalam bentuk tabel pada StringGrid 1. Jumlah baris tabel StringGrid1 sebanyak jumlah item ListBox1 (data masuk). Pemisahan masukan data per kolom menggunakan “;” sebagai acuan. Tabel pada aplikasi ini terdapat 5 kolom, yaitu Identity, latitude, longitude, date, dan TimeSent. Jika data yang masuk sudah sesuai dengan baris dan kolom yang diinginkan, proses berakhir.



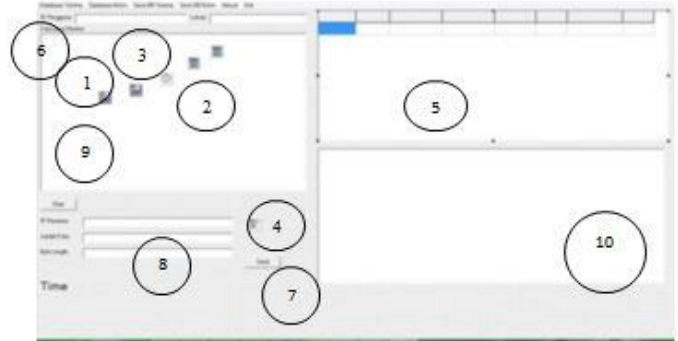
Gambar 3 Flowchart proses load ke dalam bentuk tabel

Kemudian tabel dapat disimpan ke memori komputer dengan menekan “Save DB Pengirim”. Alur pemrograman untuk menyimpan data ke format CSV dijelaskan pada Gambar 4:



Gambar 4 Flowchart proses save dalam format CSV

Data input merupakan item dari *StringGrid1*. Jumlah baris pada *StringGrid1* dijadikan acuan jumlah baris file CSV. Proses selanjutnya data pada setiap baris *StringGrid1* akan dikonversi menurut aturan CSV. Pemisah data antar kolom (*delimiter*) menggunakan “;”. Jumlah kolom pada tabel ada lima. Sehingga pada file CSV terdapat empat “;” untuk memisahkan data antar kolom. Pemisahan data pada file CSV menjadi “*Identity, latitude, longitude, date, TimeSent*”. Jika sudah sesuai data dari *StringGrid1* maka proses berakhir.



Gambar 5 Tampilan aplikasi untuk uji coba

Setelah pembuatan alur aplikasi, dibuatlah aplikasi pengiriman dan penerimaan data pada *Borland Delphi 7*. Tampilan (*interface*) aplikasi yang dibuat pada *Borland Delphi 7* dapat dilihat pada Gambar 5. Dibutuhkan beberapa komponen *form*, yaitu:

1. *UDP*, berfungsi untuk mengirim dan menerima data.
2. *Save Dialog*, berfungsi untuk menampilkan jendela *save database*.
3. *Timer*, berfungsi untuk membangkitkan waktu pada aplikasi.
4. *Menu files*, berfungsi untuk membuat desain menu pada aplikasi.
5. *StringGrid*, berfungsi untuk membuat tabel pada aplikasi
6. *ListBox*, berfungsi untuk menampilkan informasi yang dibutuhkan.
7. *Button*, berfungsi sebagai komponen tambahan yang membutuhkan tombol.
8. *Edit text*, berfungsi sebagai komponen tambahan untuk memasukkan atau mengeluarkan sebuah informasi.
9. *Label*, berfungsi sebagai pemberi keterangan yang dibutuhkan.
10. *Memo*, berfungsi untuk menampilkan informasi.

B. Perancangan Database PHPMYAdmin

Server yang digunakan untuk menerima data dalam tugas akhir ini adalah *server its2.lawanghosting.pw* dengan memanfaatkan *database PHPMYAdmin* pada *server*. Data yang masuk ke *server* akan disimpan dalam *database its2016a*. *Database* dibuat sesuai dengan format perancangan sistem komunikasi terintegrasi AMC. Tabel yang terdapat pada *database* yaitu tabel *udp_rxlog*. Berikut Gambar 6 yang merupakan tampilan struktur dari tabel *udp_rxlog*:

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Batasan	Ekstra	Tindakan
1	SerID	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	PeerIP	char(15)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Lainnya
3	PeerPort	smallint(6)			Ya	NULL		Ubah Hapus Lainnya
4	RxTStamp	char(19)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Lainnya
5	Rtimeserver	time			Ya	NULL		Ubah Hapus Lainnya
6	DataSize	smallint(6)			Ya	NULL		Ubah Hapus Lainnya
7	iduser	varchar(2000)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Lainnya
8	RxData	char(255)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Lainnya
9	timesent	varchar(20)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Lainnya
10	SvrTStamp	timestamp		on update CURRENT_TIMESTAMP	Tidak	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Ubah Hapus Lainnya

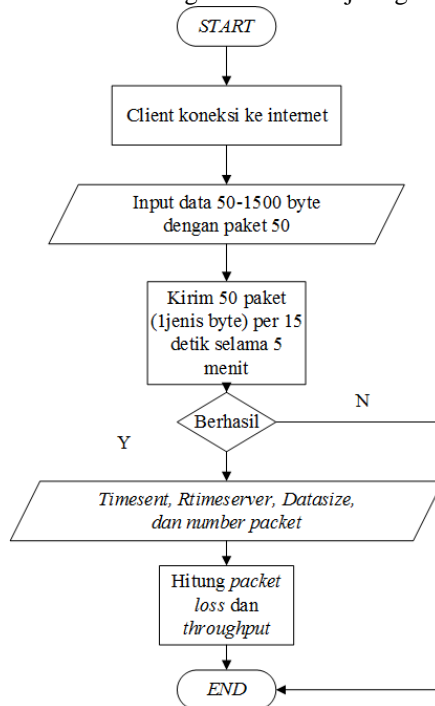
Gambar 6 Tampilan struktur dari tabel udp_rxlog

Gambar 6 adalah tampilan tabel udp_rxlog. Dalam tabel tersebut terdapat beberapa *field* antara lain:

1. *SerID*, berisi id ID urutan data yang masuk ke *server*
2. *PeerIP*, berisi IP *client* yang mengirimkan data.
3. *PeerPort*, berisi port berapa yang dilewati
4. *RxTStamp*, berisi waktu data diterima *database server*.
5. *Rtimeserver*, berisi waktu data diterima oleh *server*.
6. *Datasize*, berisi ukuran *byte* data yang diterima
7. *RxData*, berisi data yang diterima. Data-data yang diterima adalah ID *client*, *longitude*, dan *latitude*
8. *Timesent*, berisi waktu pengiriman oleh *client*
9. *SvrTStamp*, berisi waktu data masuk ke *database server*

C. Skenario Pengujian

Berikut ini Gambar 7 mengenai skenario jaringan:



Gambar 7 Skenario pengujian jaringan

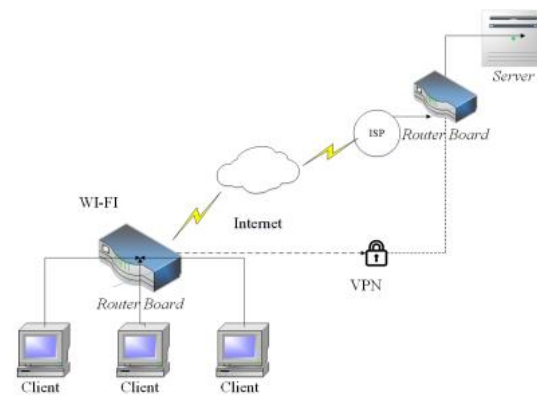
Pertama kali yang dilakukan adalah membuka aplikasi *testbed* komunikasi terintegrasi yang telah dibuat dengan menggunakan aplikasi *Borland Delphi 7* pada tiap laptop yang akan digunakan. Laptop yang digunakan berjumlah dua buah. Laptop-laptop tersebut sebagai *client* yang akan mengirim data ke *server* *its2.lawanghosting.pw* yang memiliki IP *Public*.

Pada pengujian ini, akan dilihat QoS jaringan yang telah dirancang. QoS jaringan yang ditinjau adalah *packet loss* dan

throughput. Langkah-langkah pengujian yang dilakukan yaitu aplikasi telah terkoneksi dengan jaringan *Wi-Fi* atau jaringan salah satu operator seluler. Kemudian mengatur besar *byte* data sekitar 50-1500 *byte* dan paket yang dikirim sebesar 50 paket yang di-*inputkan* oleh *client* pada aplikasi *testbed* komunikasi terintegrasi AMC. 50 paket ini adalah mewakili 50 OBU dan PIB. Pengaturan *byte* data tersebut bertujuan untuk mendapatkan rekomendasi berapa ukuran data terbaik yang dapat diterima oleh *server*, data tersebut dikirim melalui jaringan *backbone* dan jaringan seluler. *Range byte* 50-1500 *byte* tersebut diambil dari standar MTU (*Maximum Transmission Unit*), karena jika ukuran data yang dikirim lebih besar dibandingkan nilai MTU maka paket yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan akan terpecah menjadi beberapa fragmen yang akhirnya tidak jadi terkirim dengan benar. Selanjutnya, data dikirim ke *server*. Pengiriman data ke *server* per 50 paket satu jenis *byte* dengan jeda 15 detik sekali kirim dengan rentang waktu 5 menit. Jika berhasil maka didapatkan *output timesent*, *Rtimeserver*, *datasize*, dan *number packet* yang disimpan dalam *database*. *Timesent*, *Rtimeserver*, *datasize*, dan *number packet* digunakan untuk menghitung QoS, yaitu *packet loss* dan *throughput*. Apabila data tidak berhasil terkirim ke *server* maka proses diakhiri.

D. Konfigurasi Pengukuran Jaringan Wi-Fi

Arsitektur jaringan pengiriman data ini dibuat dengan menggunakan sistem *client-server*. Arsitektur dalam Gambar 8 merupakan arsitektur yang digunakan untuk melakukan *testbed* komunikasi sistem terintegrasi AMC melalui jaringan *Wi-Fi*.



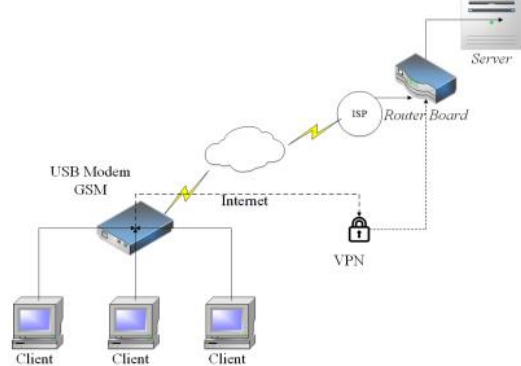
Gambar 8 Arsitektur pengukuran jaringan wi-fi

Dalam gambar tersebut terdapat beberapa *client* sebagai PIB pada halte yang terhubung dengan *access point* yang dianalogikan sebagai jaringan *backbone*. Kemudian data dikirim ke *server* yang memiliki IP *Public*. *Testbed* ini dilakukan dalam jaringan *Virtual Private Network (VPN)*.

E. Konfigurasi Pengukuran Jaringan Seluler

Arsitektur jaringan pengiriman data ini dibuat dengan menggunakan sistem *client-server*. Arsitektur dalam Gambar 9 merupakan arsitektur yang digunakan untuk melakukan *testbed* komunikasi sistem terintegrasi AMC melalui jaringan seluler.

Dalam Gambar 9 beberapa *client* sebagai OBU pada armada yang terhubung dengan jaringan operator seluler. Kemudian data dikirim ke *server* yang memiliki IP *Public*. *Testbed* ini dilakukan dalam jaringan *Virtual Private Network* (VPN).



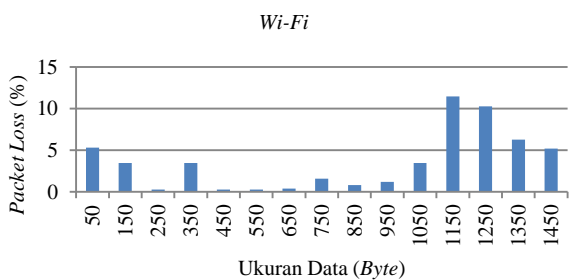
Gambar 9 Arsitektur pengukuran jaringan seluler

IV. IMPLEMENTASI DAN HASIL PENGUJIAN

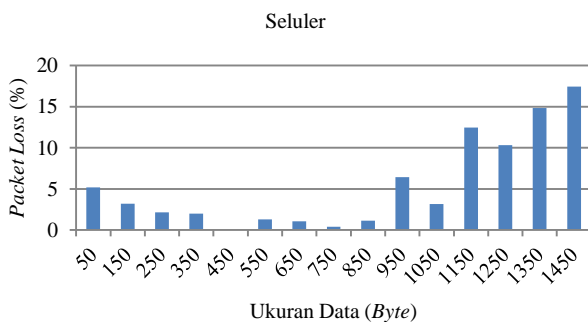
A. Pengujian Packet Loss

Pengujian *packet loss* digunakan untuk mengetahui paket data yang hilang sewaktu dilakukan pengiriman data dari *client* ke *server*. Pengujian ini dilakukan dengan mengirimkan data 50 paket secara terus-menerus selama 5 menit dengan jeda 15 detik per paket. Ukuran data yang dikirim ke *server* sebesar 50-1500 *byte*. Pengujian ini dilakukan dengan menggunakan dua laptop yang akan mengirim data secara bersamaan ke *server*. Satu laptop terkoneksi dengan jaringan *Wi-Fi* dan satu laptop terkoneksi dengan jaringan seluler.

Pengujian ini menghasilkan perbandingan antara ukuran data (*byte*) yang dikirim dengan *packet loss* ketika pengiriman data menggunakan jaringan *Wi-Fi* dan jaringan seluler. Berikut hasil dari uji coba:



Gambar 10 Grafik *packet loss* pada *Wi-Fi*



Gambar 11 Grafik *packet loss* pada seluler

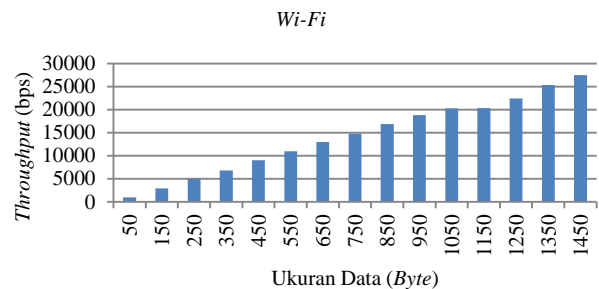
Pengujian ini menghasilkan nilai *packet loss* terbesar dan terkecil pada pengiriman data melalui jaringan *Wi-Fi* dan jaringan seluler. Pada jaringan *Wi-Fi*, *packet loss* terbesar dengan nilai sebesar 11.47% terletak pada ukuran data 1150 *byte*. *Packet loss* terkecil dengan nilai sebesar 0.267% pada ukuran data 250, 450, dan 550 *byte*. Pada jaringan seluler, *packet loss* terbesar dengan nilai sebesar 17.45% terletak pada ukuran data 1450 *byte*. Sedangkan untuk nilai *packet loss* terkecil ketika pengiriman data melalui jaringan seluler sebesar 0.13% terletak pada ukuran data 450 *byte*.

Berdasarkan grafik hasil pengujian, dapat dianalisis bahwa timbulnya *packet loss* disebabkan oleh kepadatan trafik yang menimbulkan antrian sehingga beberapa paket data tidak diterima oleh server. *Packet loss* seluler lebih besar daripada *packet loss Wi-Fi* karena kapasitas *bandwidth* seluler lebih kecil daripada kapasitas *bandwidth Wi-Fi*.

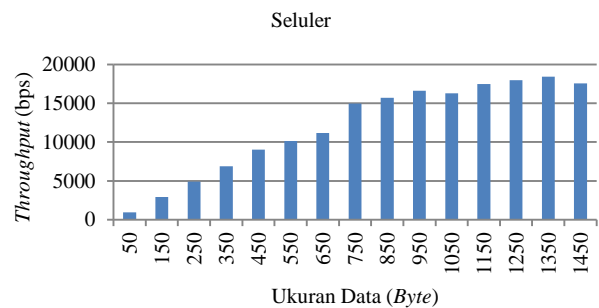
B. Pengujian Throughput

Perhitungan parameter ini akan digunakan untuk mendapatkan berapa *byte* atau jumlah panjang paket data yang optimal untuk dikirimkan pada *server*. Pengujian ini dilakukan dengan cara yang sama seperti pengujian *packet loss*, yaitu dengan mengirimkan data 50 paket secara terus menerus selama 5 menit dengan jeda 15 detik per 50 paket. Ukuran data yang dikirim ke *server* sebesar 50-1500 *byte*. Pengujian ini dilakukan dengan menggunakan dua laptop yang akan mengirim data secara bersamaan ke *server*. Satu laptop terkoneksi dengan jaringan *Wi-Fi* dan satu laptop terkoneksi dengan jaringan seluler.

Pengujian ini menghasilkan perbandingan antara ukuran data (*byte*) yang dikirim dengan *throughput* ketika pengiriman data menggunakan jaringan *Wi-Fi* dan jaringan seluler. Berikut hasil dari uji coba:



Gambar 12 *Throughput* pada *Wi-Fi*



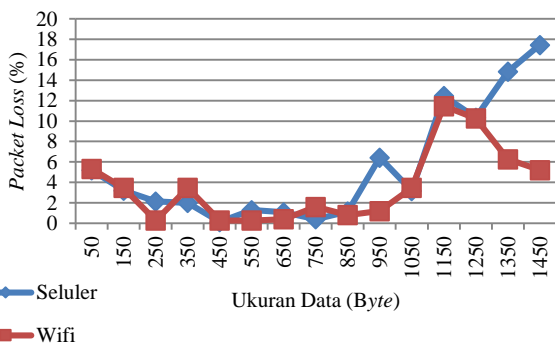
Gambar 13 *Throughput* pada seluler

Pengujian ini menghasilkan nilai *throughput* ketika pengiriman data melalui jaringan *Wi-Fi* dan jaringan seluler. Pada jaringan *Wi-Fi* didapatkan hasil *throughput* paling besar terletak pada ukuran data yang paling besar dan yang terkecil terletak pada ukuran data yang paling kecil. Ukuran data terbesar yaitu 1450 *byte* dengan nilai *throughput* sebesar 27507.44 bps dan ukuran data terkecil yaitu 50 *byte* dengan nilai *throughput* sebesar 962.0533bps. Pada jaringan seluler didapatkan hasil *throughput* paling besar terletak pada ukuran data sebesar 1350 *byte* dengan nilai *throughput* sebesar 18406.77bps. Sedangkan *throughput* terkecil ketika pengiriman data melalui jaringan seluler terletak pada ukuran data paling kecil. Ukuran data paling kecil yaitu 50 *byte* dengan nilai *throughput* sebesar 963.52bps.

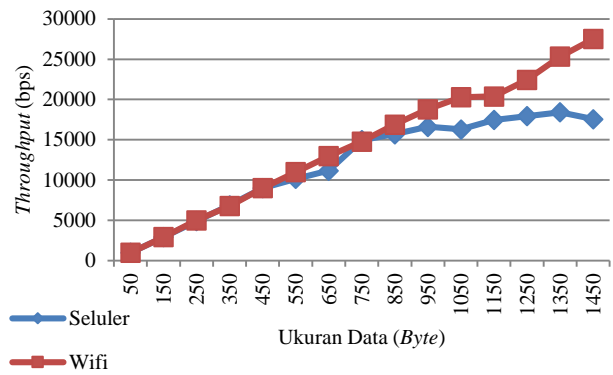
Dengan hasil pengujian tersebut, dapat dianalisis jika ukuran data semakin besar, maka nilai *throughput* yang dihasilkan semakin besar. Nilai *throughput* didapatkan dari pembagian antara jumlah ukuran data yang diterima dengan waktu pengiriman ke *server*. Namun pada jaringan seluler nilai *throughput* terbesar tidak terletak pada ukuran data tertinggi, hal ini terjadi karena pada 1450 *byte* banyak paket yang hilang.

C. Rekomendasi Ukuran Data

Berdasarkan hasil pengujian *packet loss* dan *throughput*, didapatkan rekomendasi ukuran data yang bisa digunakan untuk sistem komunikasi terintegrasi angkutan massal cepat Surabaya. Ukuran data yang sebaiknya digunakan adalah sebesar 450 *byte*. Pemilihan ukuran data sebesar 450 *byte* sebagai rekomendasi untuk sistem komunikasi terintegrasi angkutan massal cepat Surabaya dikarenakan ukuran data 450 *byte* memiliki nilai *packet loss* yang kecil, nilai *throughput* yang cukup bagus dan panjang data yang mencukupi (tidak terlalu pendek dan tidak terlalu panjang). Ukuran data 450 *byte* memiliki kualitas *packet loss perfect* berdasarkan standar versi TIPHON dan baik berdasarkan standar ITU-T G.114, baik melalui jaringan *wi-fi* maupun seluler. Nilai *packet loss* ukuran data 450 *byte* pada jaringan *wifi* sebesar 0.266667%, sedangkan pada jaringan seluler nilai *packet loss* nya sebesar 0.133333%. Nilai *throughput* yang dimiliki oleh ukuran data 450 *byte* sebesar 8992.347bps jika dikirim melalui jaringan *wi-fi*, sedangkan jika dikirim melalui jaringan seluler, nilai *throughput* nya sebesar 9004.373 bps. Ukuran data 450 *byte* yang tidak terlalu pendek dan tidak terlalu panjang sesuai dengan kebutuhan. Ukuran data 50, 150, 250, dan 350 memiliki panjang data kurang panjang. Agar lebih jelas, berikut grafik dan tabel hasil analisis:



Gambar 14 Perbandingan *Packet Loss* *Wi-Fi* dengan seluler



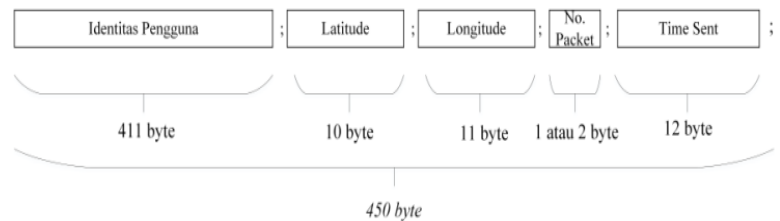
Gambar 15 Perbandingan *Throughput* *Wi-Fi* dengan seluler

Tabel 1 Kriteria Ukuran Data

Wifi	Ukuran Data (Byte)										
	450	550	650	750	850	950	1050	1150	1250	1350	1450
PL	√	√	√	√	√	√	√	x	x	x	x
Thr	√	√	√	√	√	√	√	√	√	√	√
Panjang	√	x	x	x	x	x	x	x	x	x	x

Seluler	Ukuran Data (Byte)										
	450	550	650	750	850	950	1050	1150	1250	1350	1450
PL	√	√	√	√	√	x	√	x	x	x	x
Thr	√	√	√	√	√	√	√	√	√	√	√
Panjang	√	x	x	x	x	x	x	x	x	x	x

Berikut isi data yang dapat diterapkan sesuai dengan rekomendasi:



Gambar 16 Format data pengguna

Ukuran data 450 *byte* dapat berisi identitas pengguna sebesar 411 *byte*. Dalam prakteknya nanti identitas pengguna dapat dipecah lagi menjadi beberapa bagian, seperti nomor ID, nama pengguna, dan klasifikasi pentarifan yang terdiri dari kelompok usia, kepala tiket, dan biaya yang harus dibayar. *Latitude* sebesar 10 *byte*. *Longitude* sebesar 11 *byte*. Nomor paket sebesar 1 atau 2 *byte*, *Time Sent* sebesar 12 *byte*. *Time sent* terdiri dari jam, menit, detik, dan mili detik (hh:mm:ss-*zz*). Pemisah tiap bagian data berupa titik koma (;) memiliki ukuran data sebesar 1 *byte*.

Rekomendasi ukuran data ini dapat berlaku pada kondisi seperti pada pengujian ini.

V. KESIMPULAN

Dari tugas akhir ini dapat ditarik kesimpulan:

1. Pengujian sistem *client-server* mengenai penerimaan data oleh *server*, penyimpanan data data dalam *database server*,

dan penyimpanan data balasan ke *client* telah sesuai rancangan. Sistem yang disimulasikan mampu memenuhi kebutuhan selama proses pengiriman dan penerimaan.

2. Pengujian *packet loss* terkecil pengiriman data melalui jaringan *wi-fi* terletak pada ukuran data 250, 450, dan 550 *byte* sebesar 0.26667% (*perfect*). Sedangkan pada jaringan seluler nilai *packet loss* terkecil terletak pada ukuran data 450 *byte* sebesar 0.13333% (*perfect*).
3. Nilai *throughput* berbanding lurus dengan ukuran data yang diterima.
4. Rekomendasi ukuran data yang dapat diterapkan pada sistem komunikasi terintegrasi angkutan massal cepat Surabaya, yaitu sebesar 450 *byte*. Rekomendasi ini berlaku pada kondisi seperti pengukuran ini.

DAFTAR PUSTAKA

- [1] Farhana, Nurul. "Pengembangan Aplikasi *Remote* Spesifikasi Dekstop Berbasis *Client Server*," Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2011.
- [2] Yuliantoro, Prasetyo. "Rancang Bangun Protokol *E-Ticketing*," Institut Teknologi Sepuluh Nopember, Surabaya, 2015.
- [3] Faesal, Andris. "Pemrograman *Client-server* dengan *Borland Delphi 7* dan *My SQL*," STMIK Bumigora, Mataram, 2012.
- [4] Sastriyana, Nym Yuni. "Perancangan dan Analisis Interkoneksi Jaringan pada *Electronic Toll Collection* Berbasis *Radio Frequency Identification* (RFID)," Institut Teknologi Sepuluh Nopember, Surabaya, 2010.