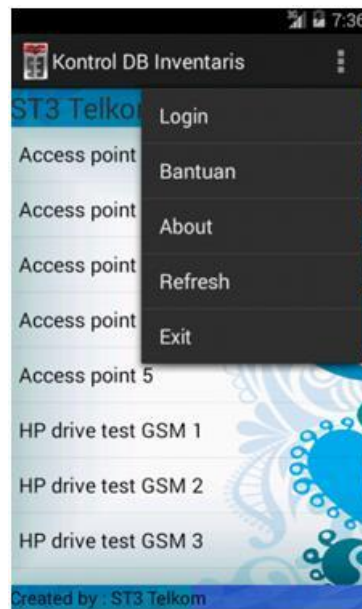


BAB IV

HASIL DAN PEMBAHASAN APLIKASI

4.1 HASIL PERANCANGAN APLIKASI

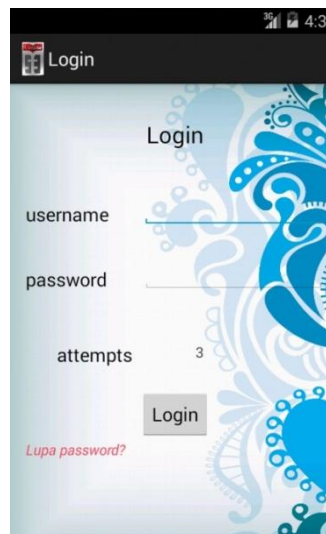
4.1.1 Tampilan *List* Barang untuk *User*



Gambar 4.1 Tampilan *list* barang untuk *user*

Gambar 4.1 menunjukkan tampilan pertama dari aplikasi pengontrolan data inventaris barang dimana dalam tampilan pertama merupakan daftar *list* dari nama barang inventaris yang terdapat di *database server*, daftar yang ditampilkan sama dengan data yang ada di *database*, namun disini hanya di tampilkan nama dari barang inventaris. Dalam tampilan ini terdapat submenu yang terdapat dipojok kanan atas dimana apabila di klik akan muncul menu yaitu *Login*, *Refresh*, *About*, *Bantuan*, dan *exit*. Setiap menu mempunyai perintahnya masing – masing, seperti menu *login* digunakan untuk masuk sebagai admin yang mempunyai hak akses untuk menambah, mengedit dan manghapus *database*.

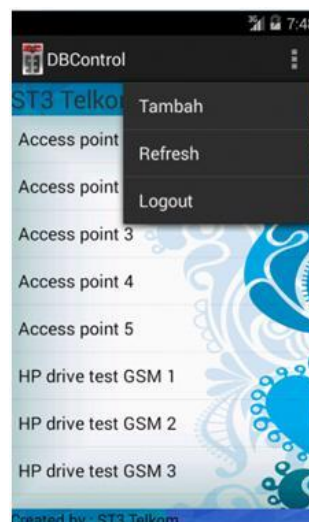
4.1.2 Tampilan *Login* Admin



Gambar 4.2 Tampilan *Login*

Pada gambar 4.2 menunjukkan tampilan dari menu *login*, dimana menu *login* berfungsi sebagai *interface* untuk masuk sebagai admin ke aplikasi yang mempunyai hak akses untuk menambah, mengedit dan menghapus data di *database server*. Dalam *layout* ini terdapat *textfield* yang harus diisi yaitu *username* dan *password*. Untuk *username* dan *password* tidak bisa diganti langsung diaplikasi, tetapi harus menggantinya dari *source code* dalam pembuatan aplikasi tersebut.

4.1.3 Tampilan *List Barang* untuk Admin



Gambar 4.3 Tampilan *list* barang untuk admin

Gambar 4.3 menunjukkan tampilan untuk admin dalam melihat *list* nama barang dari *database server*, tampilan ini sama dengan tampilan untuk *user* umum seperti gambar 4.1 namun disini terdapat fitur untuk menambah, mengedit dan menghapus data.

4.1.4 Tampilan Edit Data



Gambar 4.4 tampilan edit data

Gambar 4.4 menampilkan tampilan dari *layout* edit data yang berfungsi untuk meng-*update* data, dimana data yang sudah ada di-*update* sesuai kebutuhan. Tampilan ini akan muncul ketika kita mengklik salah satu nama dari *list* nama barang yang terdapat di *layout* gambar 4.3 . *Layout* ini berisikan data lengkap dari dari salah satu barang yang dipilih, disini admin bisa mengganti data tersebut dan menyimpannya kembali.

4.1.5 Tampilan Tambah Data



The screenshot shows a mobile application interface for adding data. The title bar at the top is black with a white icon and the text "Tambah Data". The background is light blue with a decorative pattern on the right side. The form consists of several text input fields stacked vertically, each with a label to its left: "Serial Number", "Uraian Peralatan", "Merk/Type", "Jumlah", "Kondisi", and "Ket". At the bottom of the form is a grey button labeled "Simpan". The status bar at the top right shows the time as 7:48.

Gambar 4.5 Tampilan tambah data

Seperti pada gambar 4.5 *layout* tambah data hampir sama dengan *layout* edit data, tetapi *layout* ini tidak menampilkan data lengkap dari barang yang dipilih, melainkan *form* pengisian untuk menambah data, yang didalamnya terdapat *text field* untuk nomer serial, nama barang, merk, jumlah, kondisi dan keterangan dan dapat disimpan ke *database server*.

4.1.6 Tampilan Info untuk User

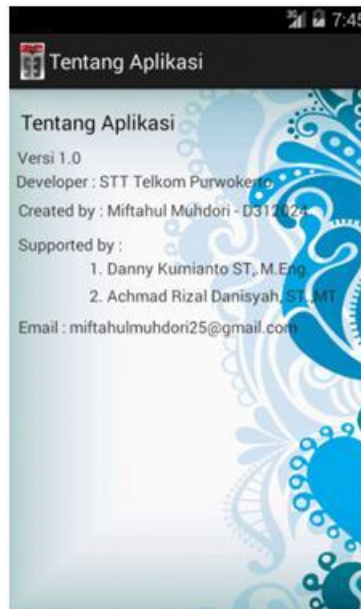


The screenshot shows a mobile application interface for displaying item information. The title bar at the top is black with a white icon and the text "Info Barang". The background is light blue with a decorative pattern on the right side. The form displays the following information: "Serial Number" (13BC6703494), "Uraian Peralatan" (Access point 5), "Merk/Type" (TP-Link (versi 3) Model NO: TL WA901ND), "Jumlah" (1), "Kondisi" (Baik), and "Ket" (null). At the bottom of the form is a grey button labeled "Kembali". The status bar at the top right shows the time as 7:36.

Gambar 4.5 Tampilan Info barang

Gambar 4.5 menunjukkan tampilan dari menu info yang di pilih dari salah satu barang di *list* nama barang pada *layout* di gambar 4.1 , di *layout* ini ditampilkan informasi lengkap dari nama barang yang dipilih.

4.1.7 Tampilan Tentang Aplikasi



Gambar 4.7 Tampilan Tentang aplikasi

Seperti pada gambar 4.7 tampilan ini merupakan informasi mengenai aplikasi yang dibuat.

4.1.8 Tampilan Bantuan



Gambar 4.8 Tampilan bantuan

Gambar 4.8 menampilkan tampilan dari *layout* bantuan yang berisikan tentang petunjuk untuk menggunakan aplikasi.

4.1.9 Tampilan Fasilitas untuk Lupa *Password*



Gambar 4.9 Tampilan fasilitas untuk lupa *password*

Gambar 4.9 menampilkan *layout* yang berfungsi untuk membantu admin ketika admin lupa dengan *password* yang digunakan, dimana dalam *layout* ini terdapat pertanyaan yang harus diisi oleh admin. Jika jawabannya benar, maka akan masuk ke *layout* yang menampilkan *username* dan *password* yang digunakan dalam aplikasi.

4.1.10 Tampilan Informasi *Username* dan *Password*



Gambar 4.10 Tampilan informasi *username* dan *password*

Gambar 4.10 merupakan *layout* yang menampilkan informasi *username* dan *password* yang digunakan dalam aplikasi pengontrolan data inventaris barang.

Tabel 4.1 dan tabel 4.2 menunjukkan hasil pengujian unit aplikasi yang dibuat pada Tugas Akhir ini.

Tabel 4.1 Hasil Pengujian Unit

No	Modul	Unit	Fungsi	Hasil
1	Layout utama	- class inisialisasi <i>layout</i> - class pengambilan <i>database</i> - class info - class bantuan - class login - class about - class refresh - class exit	Sebagai navigasi utama untuk mengarahkan <i>user</i> umum dalam mendapatkan informasi	OK
2	Layout info	- class inisialisasi <i>layout</i> - class informasi - class pengambilan <i>database</i>	Menampilkan informasi barang yang dipilih	OK
3	Layout bantuan	- class inisialisasi <i>layout</i> - class informasi	Menampilkan petunjuk penggunaan aplikasi	OK
4	Layout Login	- class inisialisasi <i>layout</i> - class informasi	Menampilkan kotak isian <i>username</i> dan <i>password</i> untuk login	OK
5	Layout about	- class inisialisasi <i>layout</i> - class informasi	Menampilkan informasi tentang	OK

			aplikasi dan pembuatnya	
6	<i>Layout list</i>	<ul style="list-style-type: none"> - <i>class</i> inialisasi <i>layout</i> -<i>class</i> pengambilan <i>database</i> - <i>class</i> tambah data - <i>class</i> edit data - <i>class delete</i> data - <i>class about</i> -<i>class refresh</i> 	Sebagai navigasi utama untuk mengarahkan admin dalam mendapatkan informasi	OK
7	<i>Layout</i> tambah data	<ul style="list-style-type: none"> - <i>class</i> inialisasi <i>layout</i> - <i>class</i> simpan -<i>class</i> pengontrolan <i>database</i> 	Sebagai <i>form</i> pengisian data baru	OK
8	<i>Layout</i> edit data	<ul style="list-style-type: none"> - <i>class</i> inialisasi <i>layout</i> - <i>class</i> simpan -<i>class</i> pengambilan <i>database</i> -<i>class</i> pengontrolan <i>database</i> 	Sebagai <i>form</i> pengeditan data yang sudah ada	OK

Tabel 4.2 Hasil Pengujian Fungsional

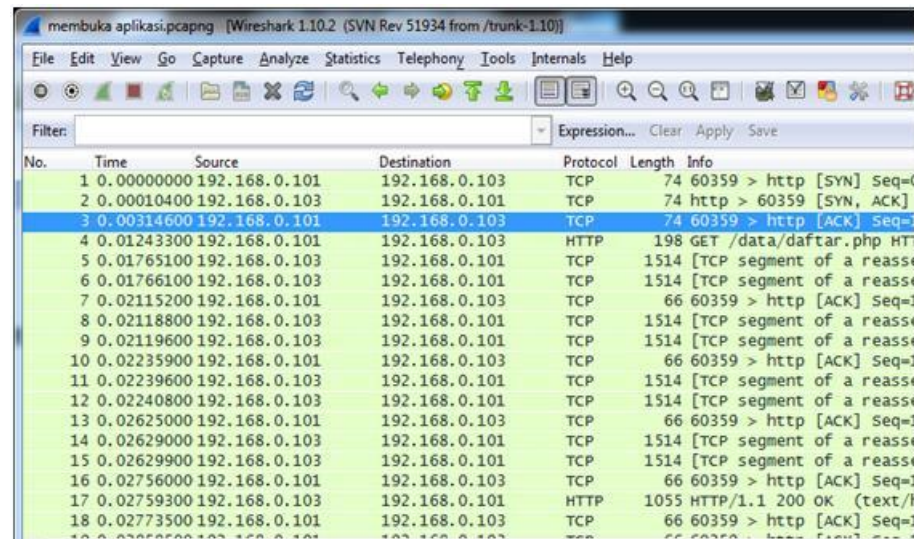
No	Kebutuhan	Kesesuaian
1	Menu pilihan	Sesuai
2	<i>Button</i>	Sesuai
3	Koneksi jaringan	Sesuai

4.2 PENGIRIMAN PAKET DATA

Pengiriman paket data dapat dilihat dengan menggunakan *software* Wireshark, Wireshark adalah *tool* yang ditunjukkan untuk penganalisisan paket data jaringan. Wireshark melakukan pengawasan paket secara waktu nyata (*real time*). Wireshark

dapat *capture* jalannya pengiriman data dari *server* ke *client* seperti pada aplikasi pengontrolan data inventaris yang melakukan komunikasi data dengan *database server*, dimana untuk masing – masing *layout* mempunyai paket data yang berbeda, seperti berikut.

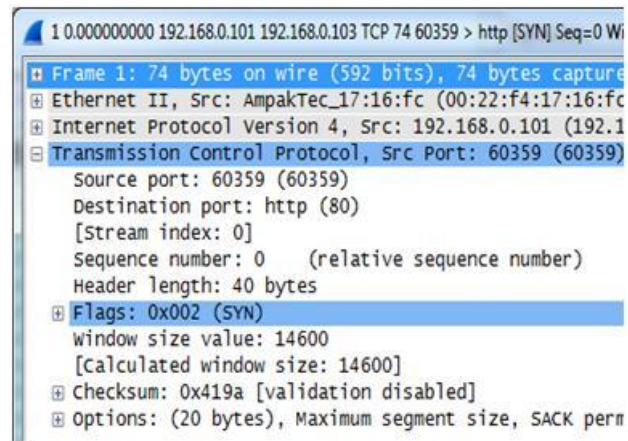
4.2.1 *MainActivity.java*



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.0.101	192.168.0.103	TCP	74	60359 > http [SYN] Seq=
2	0.00010400	192.168.0.103	192.168.0.101	TCP	74	http > 60359 [SYN, ACK]
3	0.00314600	192.168.0.101	192.168.0.103	TCP	74	60359 > http [ACK] Seq=
4	0.01243300	192.168.0.101	192.168.0.103	HTTP	198	GET /data/daftar.php HT
5	0.01765100	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
6	0.01766100	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
7	0.02115200	192.168.0.101	192.168.0.103	TCP	66	60359 > http [ACK] Seq=
8	0.02118800	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
9	0.02119600	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
10	0.02235900	192.168.0.101	192.168.0.103	TCP	66	60359 > http [ACK] Seq=
11	0.02239600	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
12	0.02240800	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
13	0.02625000	192.168.0.101	192.168.0.103	TCP	66	60359 > http [ACK] Seq=
14	0.02629000	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
15	0.02629900	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
16	0.02756000	192.168.0.101	192.168.0.103	TCP	66	60359 > http [ACK] Seq=
17	0.02759300	192.168.0.103	192.168.0.101	HTTP	1055	HTTP/1.1 200 OK (text/p
18	0.02773500	192.168.0.101	192.168.0.103	TCP	66	60359 > http [ACK] Seq=

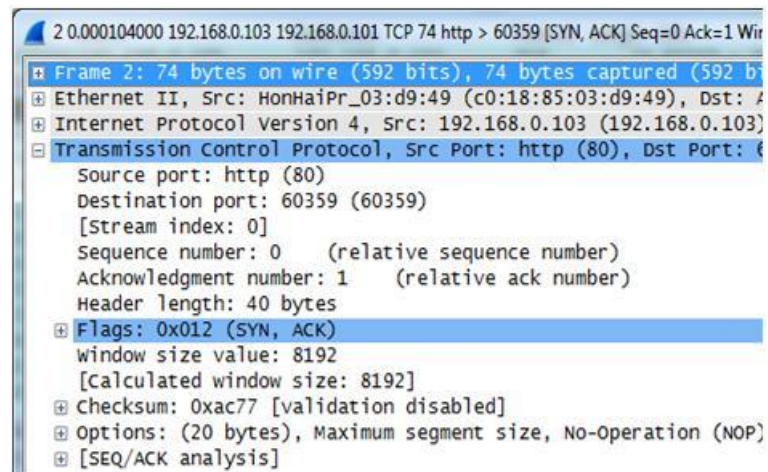
Gambar 4.11 Paket data di *interface list* barang untuk *user*

Pada gambar 4.11 merupakan paket data keseluruhan pada saat aplikasi pertama kali dibuka, dimana memuat *layout main.xml* dan *MainActivity.java*, pada paket data yang pertama IP sumber (192.168.0.101) merupakan IP dari aplikasi yang berkomunikasi dengan IP tujuan yaitu 192.168.0.103 yang merupakan IP dari *database server* dengan menggunakan *protocol* TCP karena data yang dikirimkan harus benar – benar sampai seutuhnya dimana protokol TCP bekerja di *layers transport*. Paket data pertama bertugas untuk mensinkronkan antara sumber dengan tujuan, yang nantinya akan di balas dengan umpan balik dari tujuan. Rincian untuk paket data pertama dapat dilihat pada gambar 4.12



Gambar 4.12 detail isi paket data 1

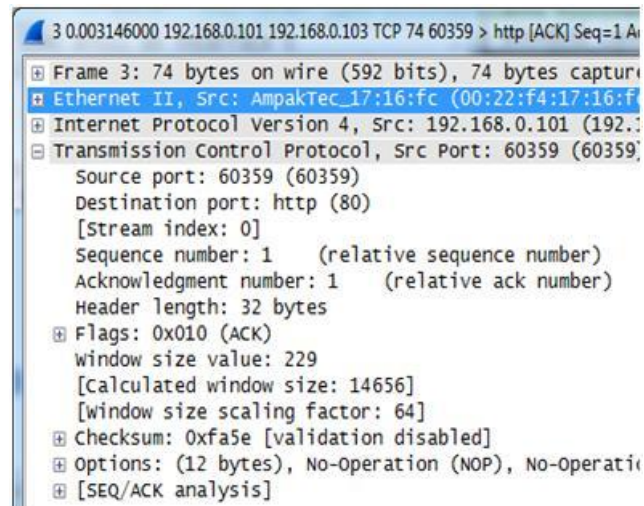
Pada gambar 4.12 terlihat *sourceport* menggunakan *port* 60359 yang merupakan *port* dari aplikasi, sedangkan untuk *port* tujuan adalah 80 (http) yang merupakan *port* dari *database* sumber (mySql), dengan *sequence number* 0 dan panjang header 40 *bytes*. Untuk paket data kedua, terjadi proses pemberitahuan (*acknowledgement*) yang memberitahukan bahwa aplikasi dan *database* sudah sinkron, lebih jelasnya terdapat di gambar 4.13



Gambar 4.13 detail paket data 2

Pada gambar 4.13 *port* sumber yang digunakan adalah 80 yaitu *port* dari *database server* dan *port* tujuan adalah 60359, perbedaan dengan paket data pertama adalah pada paket data kedua terdapat *ACK number* yaitu 1, yang menandakan bahwa sumber mengirimkan *ACK/pemberitahuan* ke aplikasi. Sedangkan untuk paket data yang ketiga, aplikasi akan merespon

ACK dari *database* sumber dengan mengirimkan *ACK* bahwa aplikasi juga sudah siap untuk melakukan pengiriman data, untuk lebih jelasnya lihat gambar 4.14



Gambar 4.14 detail paket data 3

Pada gambar 4.14 dapat dilihat bahwa yang melakukan pengiriman *ACK* dengan *ACK* number 1 adalah aplikasi dengan melihat *port* sumber yaitu 60359 yang merupakan *port* dari aplikasi. Untuk paket data yang keempat menggunakan protokol http yang digunakan untuk mengakses *database* yang menggunakan *port* 80 (http) seperti terlihat pada gambar 4.15

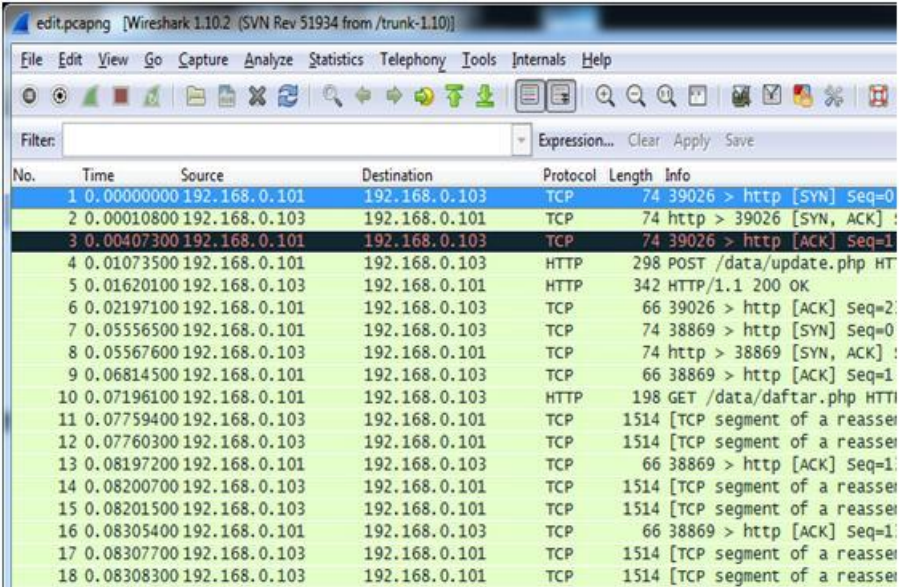


Gambar 4.15 detail paket data untuk protokol http

Pada gambar 4.15 , paket data nomer 4 dengan *source port* 60359 (apliaksi) meminta data yang beralamat alamat

`http://192.168.0.103/data/daftar.php` dengan menggunakan protokol HTTP yang bekerja pada *layer application* yang akan memberikan alamat khusus berbasis *world wide web* yang berfungsi menghubungkan aplikasi ke jaringan dengan memanfaatkan perintah yang merupakan *source code* .php yang berfungsi untuk mengubah data dari bahasa SQL ke bahasa java dan menampilkannya dalam *format Array*. Setelah itu komunikasi dilakukan dengan menggunakan protokol TCP untuk mengirimkan data dari *database*, dan ketika data berhasil dikirimkan *database server* akan memberikan pemberitahuan “OK” dengan menggunakan protokol http.

4.2.2 *EditActivity.java*



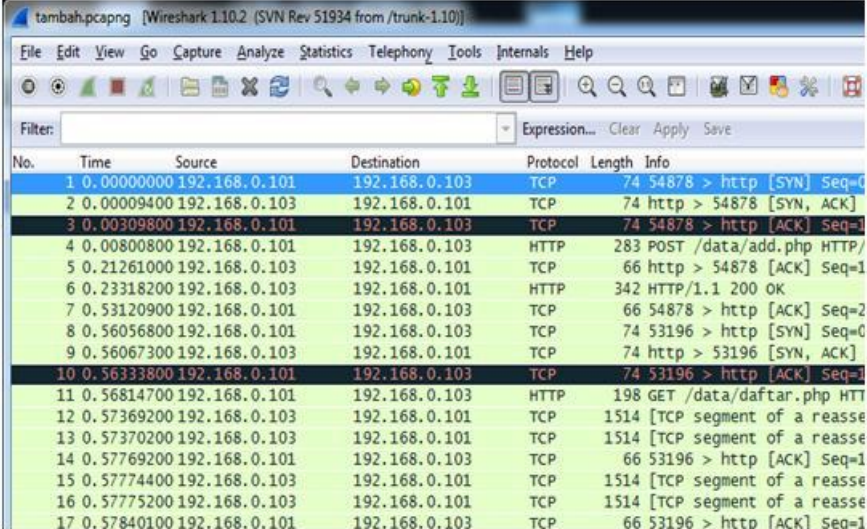
No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.0.101	192.168.0.103	TCP	74	39026 > http [SYN] Seq=0
2	0.00010800	192.168.0.103	192.168.0.101	TCP	74	http > 39026 [SYN, ACK] :
3	0.00407300	192.168.0.101	192.168.0.103	TCP	74	39026 > http [ACK] Seq=1
4	0.01073500	192.168.0.101	192.168.0.103	HTTP	298	POST /data/update.php HT
5	0.01620100	192.168.0.103	192.168.0.101	HTTP	342	HTTP/1.1 200 OK
6	0.02197100	192.168.0.101	192.168.0.103	TCP	66	39026 > http [ACK] Seq=2
7	0.05556500	192.168.0.101	192.168.0.103	TCP	74	38869 > http [SYN] Seq=0
8	0.05567600	192.168.0.103	192.168.0.101	TCP	74	http > 38869 [SYN, ACK] :
9	0.06814500	192.168.0.101	192.168.0.103	TCP	66	38869 > http [ACK] Seq=1
10	0.07196100	192.168.0.101	192.168.0.103	HTTP	198	GET /data/daftar.php HTTI
11	0.07759400	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser
12	0.07760300	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser
13	0.08197200	192.168.0.101	192.168.0.103	TCP	66	38869 > http [ACK] Seq=1
14	0.08200700	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser
15	0.08201500	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser
16	0.08305400	192.168.0.101	192.168.0.103	TCP	66	38869 > http [ACK] Seq=1
17	0.08307700	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser
18	0.08308300	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasser

Gambar 4.16 pengiriman paket data di *interface* edit data

Gambar 4.16 merupakan kumpulan paket data yang dihasilkan oleh aplikasi ketika melakukan pengeditan data pada saat tombol simpan di tekan, dimana pada paket nomer 1 aplikasi akan mengirimkan sinkronisasi ke *database server* yang selanjutnya akan di balas oleh *database server* disertai dengan *ACK* pada paket 2, dan dipaket 3 aplikasi akan membalas *ACK* tersebut dengan mengirimkan *ACK* ke *database server* yang menandakan keduanya siap bertukar data, kemudian pada paket nomer 4 dengan menggunakan protokol http aplikasi meminta perintah dengan alamat `http://192.168.0.103/data/edit.php` yang berfungsi untuk menyimpan hasil

editan ke *database server*. Setelah berhasil *database server* akan memberitahukan dengan menjawab “OK” yang artinya data berhasil disimpan, setelah itu akan kembali ke *layout main.xml* yang paket datanya kembali bekerja seperti pada *point 1 (MainActivity.java)*.

4.2.3 AddActivity.java

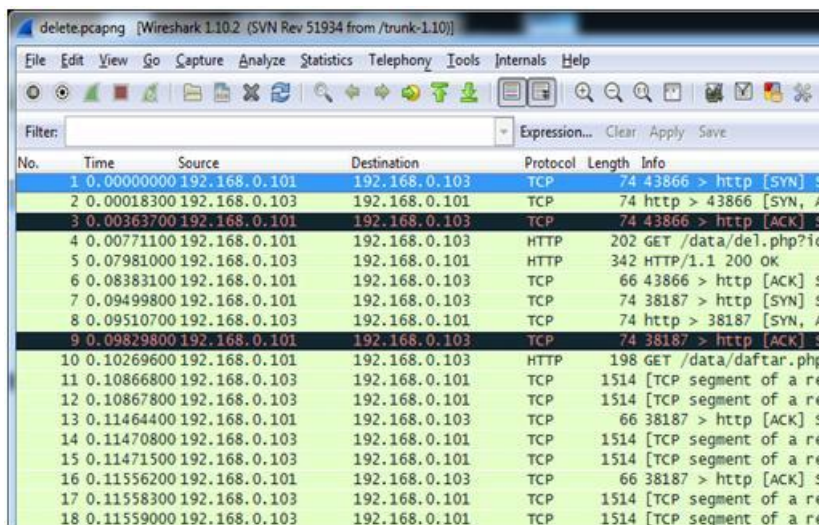


No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.0.101	192.168.0.103	TCP	74	54878 > http [SYN] Seq=0
2	0.00009400	192.168.0.103	192.168.0.101	TCP	74	http > 54878 [SYN, ACK]
3	0.00309800	192.168.0.101	192.168.0.103	TCP	74	54878 > http [ACK] Seq=1
4	0.00800800	192.168.0.101	192.168.0.103	HTTP	283	POST /data/add.php HTTP/
5	0.21261000	192.168.0.103	192.168.0.101	TCP	66	http > 54878 [ACK] Seq=1
6	0.23318200	192.168.0.103	192.168.0.101	HTTP	342	HTTP/1.1 200 OK
7	0.53120900	192.168.0.101	192.168.0.103	TCP	66	54878 > http [ACK] Seq=2
8	0.56056800	192.168.0.101	192.168.0.103	TCP	74	53196 > http [SYN] Seq=0
9	0.56067300	192.168.0.103	192.168.0.101	TCP	74	http > 53196 [SYN, ACK]
10	0.56333800	192.168.0.101	192.168.0.103	TCP	74	53196 > http [ACK] Seq=1
11	0.56814700	192.168.0.101	192.168.0.103	HTTP	198	GET /data/daftar.php HTT
12	0.57369200	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
13	0.57370200	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
14	0.57769200	192.168.0.101	192.168.0.103	TCP	66	53196 > http [ACK] Seq=1
15	0.57774400	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
16	0.57775200	192.168.0.103	192.168.0.101	TCP	1514	[TCP segment of a reasse
17	0.57840100	192.168.0.101	192.168.0.103	TCP	66	53196 > http [ACK] Seq=1

Gambar 4.17 pengiriman paket data di *interface* tambah data

Gambar 4.17 merupakan kumpulan paket data yang dihasilkan oleh aplikasi ketika melakukan penambahan data pada saat tombol simpan di tekan, dimana pada paket nomor 1 aplikasi akan mengirimkan sinkronisasi ke *database server* yang selanjutnya akan di balas oleh *database server* disertai dengan *ACK* pada paket 2, dan dipaket 3 aplikasi akan membalas *ACK* tersebut dengan mengirimkan *ACK* ke *database server* yang menandakan keduanya siap bertukar data, kemudian pada paket nomor 4 dengan menggunakan protokol http aplikasi meminta perintah dengan alamat `http://192.168.0.103/data/add.php` yang berfungsi untuk menyimpan data inventaris baru ke *database server*. Setelah berhasil *database server* akan memberitahukan dengan menjawab “OK” yang artinya data berhasil disimpan, setelah itu akan kembali ke *layout main.xml* yang paket datanya kembali bekerja seperti pada *point 1 (MainActivity.java)*.

4.2.4 Delete



Gambar 4.18 pengiriman paket data untuk perintah delete

Perintah *delete* pada aplikasi mempunyai pengiriman paket seperti pada gambar 4.18, dimana pada paket nomor 1 aplikasi akan mengirimkan sinkronisasi ke *database server* yang selanjutnya dibaloes oleh *database server* disertai dengan *ACK* pada paket nomor dua, pada paket nomor 3 aplikasi akan membalas *ACK* tersebut dengan mengirimkan *ACK* ke *database server* yang menandakan bahwa keduanya siap untuk bertukar data, dipaket nomor 4 protokol yang digunakan adalah http dimana aplikasi meminta perintah dengan alamat `http://192.168.0.103/data/delete.php` yang berfungsi untuk menghapus data yang telah dipilih sebelumnya.

Selanjutnya pengujian alat dalam hal estimasi waktu, dalam pengujian ini diambil sampel sebanyak lima barang seperti pada tabel 4.3, dimana dilakukan penambahan data, pengupdatean data, dan penghapusan data secara manual dan kemudian dilakukan dengan menggunakan aplikasi dengan perolehan waktu rata - rata seperti pada tabel 4.4.

Tabel 4.3 Sampel barang

No	Nama	Serial Number	Merk/Type	Jumlah	Kondisi	Ket
1	Access point	13BC67034 94	TP-Link (versi 3) Model NO:TL-	1	Baik	

			WA901ND			
2	HP drive test GSM	CB5AOBF MOV	Sony Ericsson K800i	1	Baik	
3	USB to Serial		ATEN Model : UC-232A	5	Baik	
4	Switch catalys	FOC111O W19L	CISCO System Catalys 2950 series	1	Baik	
5	Router	JPE08321O RS	Cisco system (2621) Cisco 2600 series	1	Baik	

Tabel 4.4 Perbandingan Waktu Kontrol data inventaris

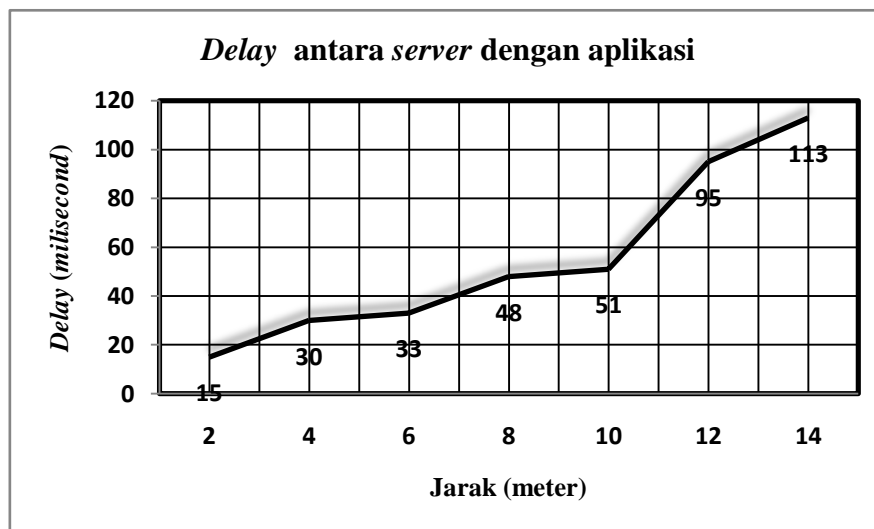
No	Uraian Pekerjaan	Waktu Yang Dibutuhkan Menyelesaikan Pekerjaan	
		Aplikasi	Manual
1	Menambah data baru	00:04:36 detik	00:13:43 detik
2	Mengupdate data	00:03:00 detik	00:08:04 detik
3	Menghapus data	00:00:09 detik	00:00:17 detik
	Rata - rata	00:02:35 detik	00:07:21 detik

Pada tabel 4.4 perolehan waktu rata – rata yang dibutuhkan oleh aplikasi untuk menyelesaikan ketiga pekerjaan yaitu selama 2 menit 35 detik atau 155 detik sedangkan perolehan waktu yang dibutuhkan untuk menyelesaikan ketiga pekerjaan secara manual membutuhkan waktu 7 menit 21 detik atau 441 detik.

Selain pengujian estimasi waktu, dilakukan juga pengujian *delay* dalam komunikasi data antara *server* dan aplikasi dengan menggunakan *software* wireshark yang membandingkan waktu pengiriman dan penerimaan data, pengujian ini dilakukan untuk mengetahui perubahan *delay* yang terjadi antara aplikasi dengan *server* , dimana pengujian ini dilakukan dengan mengubah jarak antara *handphone* dengan *access point* seperti terlihat pada tabel 4.5.

Tabel 4.5 *Delay* berdasarkan jarak antara *access point* dengan *handphone*

No	Jarak (meter)	<i>Delay</i> (milisecond)	Perubahan <i>Delay</i> (milisecond)
1	2	15	0
2	4	30	15
3	6	33	3
4	8	48	15
5	10	51	3
6	12	95	44
7	14	113	18
Rata – rata perubahan <i>delay</i>			14

Gambar 4.19 Grafik *delay* komunikasi data antara *server* dengan aplikasi

Gambar 4.19 menunjukkan bahwa semakin jauh jarak dari aplikasi dengan *server* akan menghasilkan *delay* yang semakin besar, hal ini dikarenakan jarak yang ditempuh oleh data yang dikirimkan dari *server* ke aplikasi atau dari aplikasi ke *server* semakin panjang yang mempengaruhi besar daya sinyal *Wi-Fi* yang diterima oleh *handphone*, dimana daya sinyal *Wi-Fi* semakin kecil ketika jarak antara *handphone* dengan *access point* semakin jauh. Selain itu adanya penambahan *delay* waktu dalam pengiriman data yang hilang atau rusak di tengah jalan.