BAB IV
PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai pengujian dari perangkat presentasi pada proyektor digital untuk *smartphone* Android dengan antarmuka *Video Graphics Array* (VGA) berikut analisisnya. Pengujian dilakukan untuk mengetahui sejauh mana kinerja perangkat yang dirancang dapat memenuhi spesifikasi yang telah ditentukan.

4.1. Pengujian Modul *VGA Encoder* pada *VGA Adapter*

Pengujian ini dilakukan untuk mengetahui kemampuan modul *VGA Encoder* dalam menampilkan sebuah gambar pada resolusi VGA (800 x 600 piksel). Metode pengujian dilakukan dengan membuat sebuah program pada modul mikroprosesor OK6410-B yang berfungsi untuk mengambil nilai-nilai RGB pada sebuah gambar dan menampilkannya pada resolusi VGA. Masukan berupa lima gambar digital dengan ekstensi **.bmp* (*bitmap*) berdimensi 800 x 600 piksel yang tersimpan dalam memori modul mikroprosesor. Kemudian modul *VGA Encoder* dihubungkan dengan sebuah monitor lewat kabel VGA. Diagram alir dari metode pengujian modul *VGA Encoder* ditunjukkan pada Gambar 4.1.

Gambar 4.1. Diagram alir pengujian modul *VGA Encoder*.

Program yang digunakan untuk pengujian menggunakan pustaka *QDBMP* yang memiliki fungsi-fungsi untuk keperluan *encoding* dan *decoding* gambar *bitmap*. Pustaka ini digunakan untuk mengambil nilai-nilai RGB dari sebuah gambar *bitmap*. Nilai-nilai RGB tersebut selanjutnya ditampilkan pada keluaran VGA. Gambar 4.2 menunjukkan contoh pengujian untuk modul *VGA Encoder*.

Gambar 4.2. Contoh pengujian modul *VGA Encoder*.

Tabel 4.1 menunjukkan hasil pengujian modul *VGA Encoder* dengan lima gambar *bitmap* sebagai gambar uji. Pada setiap gambar uji dilakukan pencatatan waktu untuk proses menampilkan gambar pada keluaran VGA dengan resolusi 800 x 600 piksel.

Tabel 4.1. Hasil pengujian modul *VGA Encoder*.

Nomor Gambar Uji	Kesesuaian Gambar	Keluaran VGA (detik)
1	✓	0.02
2	✓	0.02
3	✓	0.03
4	✓	0.03
5	✓	0.02

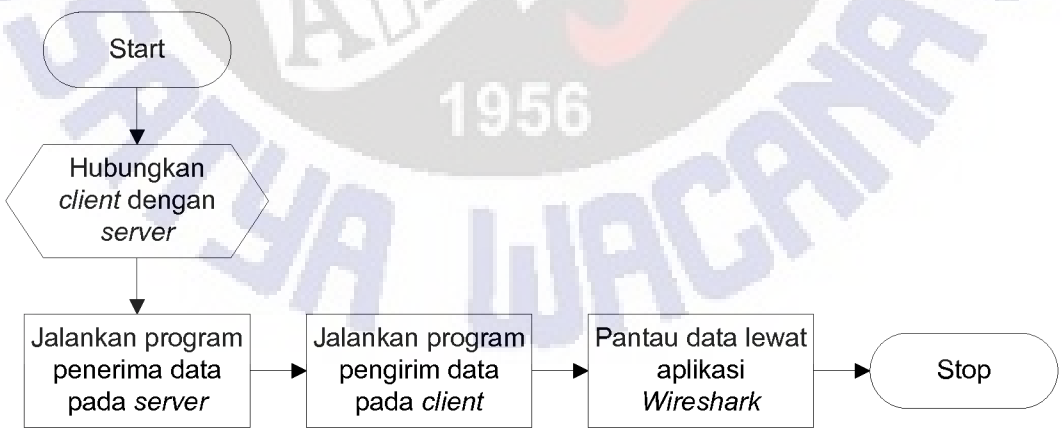
Menurut hasil pengujian sesuai pada Tabel 4.1, modul *VGA Encoder* dapat menampilkan gambar dalam waktu yang relatif sama yaitu 0.02 sampai 0.03 detik. Dapat dilihat juga bahwa gambar yang ditampilkan oleh *VGA Encoder* sesuai dengan gambar uji. Hanya saja gambar yang ditampilkan oleh *VGA Encoder* terkesan kurang cerah. Hal ini

disebabkan setiap gambar yang ditampilkan oleh *VGA Encoder* hanya memiliki kedalaman warna 16-bit, lebih rendah dari gambar asli yang memiliki kedalaman warna 24-bit, sehingga sedikit mengurangi kualitas warna pada tampilan gambar.

4.2. Pengujian Komunikasi Data TCP dengan *VGA Adapter*

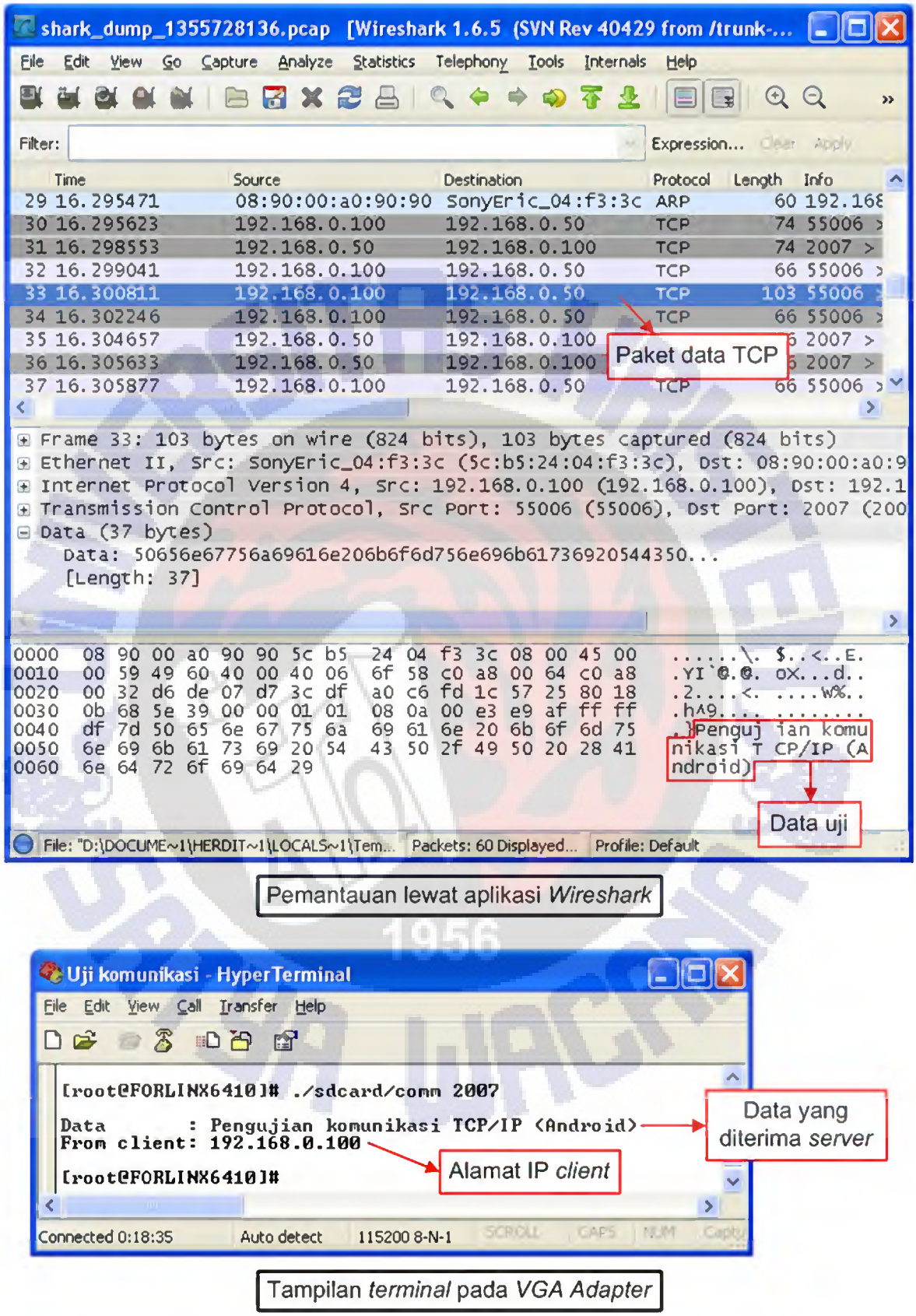
Pengujian ini bertujuan untuk menguji komunikasi data antara *client* (Android, Windows, Linux) dengan *server* (*VGA Adapter*) menggunakan protokol TCP. Baik *client* dan *server* menggunakan teknik pemrograman *socket* untuk komunikasi data dalam jaringan. Sebelum komunikasi dapat dilakukan, *server* harus membuka *socket* miliknya melalui sebuah *port* yang disepakati bersama dengan *client*. Nomor *port* yang digunakan yaitu *port* 2007.

Metode pengujian dilakukan dengan membuat sebuah program sederhana pada *client* dan *server*. Program pada *client* digunakan untuk mengirim sebuah *string* sebagai data uji ke *server*. Sedangkan program pada *server* berfungsi untuk menerima dan menampilkan *string* yang dikirim oleh *client*. Untuk memantau pengiriman data uji dari *client* ke *server* digunakan aplikasi *Wireshark*. Aplikasi tersebut dijalankan pada sebuah komputer yang dihubungkan ke jaringan *Wi-Fi* milik *server* (*VGA Adapter*). Gambar 4.3 merupakan diagram alir metode pengujian komunikasi data TCP antara *client* (Android, Windows, Linux) dan *server* (*VGA Adapter*).

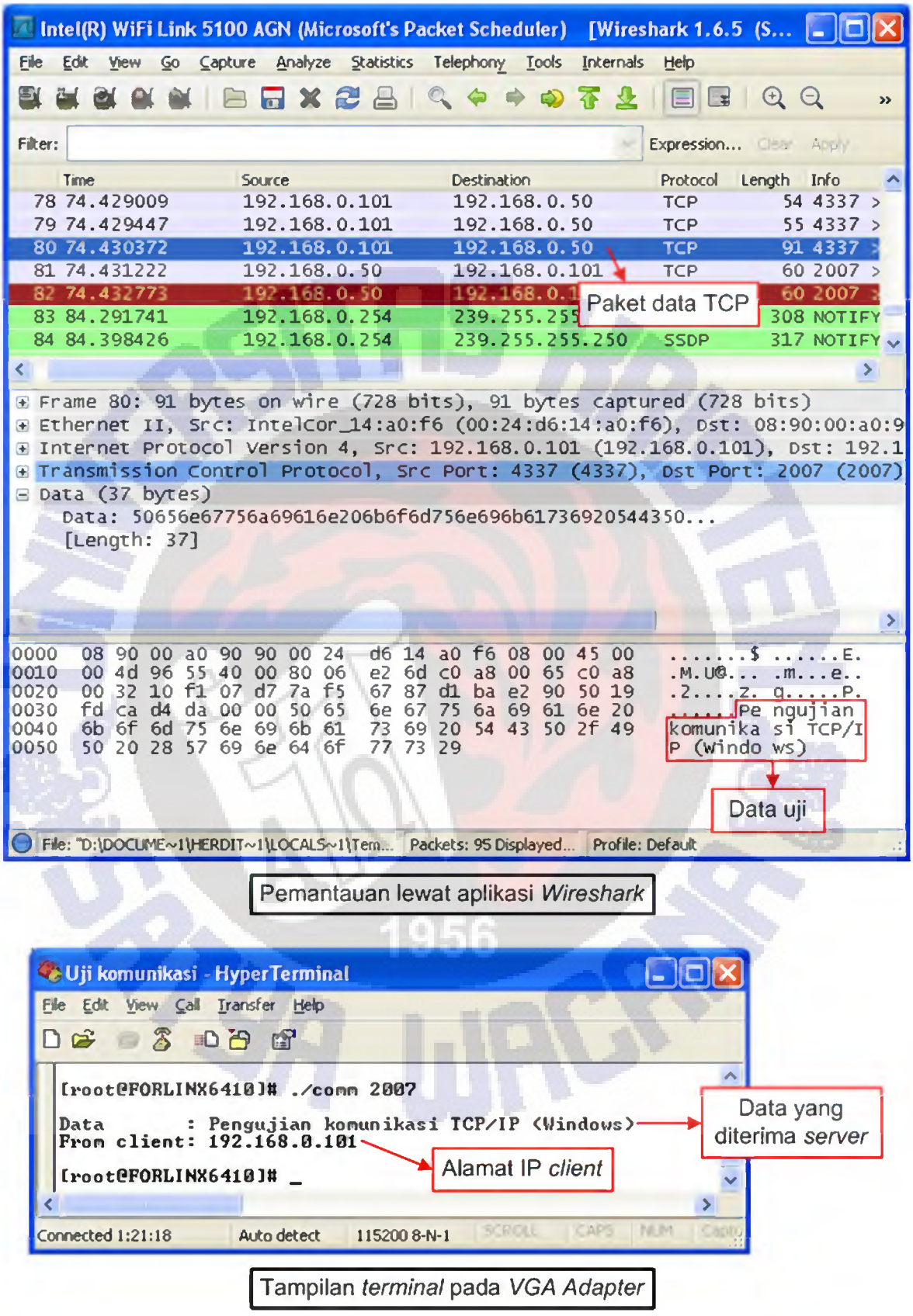


Gambar 4.3. Diagram alir metode pengujian komunikasi data TCP.

Gambar 4.4 menunjukkan hasil pengujian komunikasi data TCP dengan *smartphone* Android sebagai *client*. Sedangkan Gambar 4.5 merupakan hasil pengujian komunikasi data TCP dengan *client* yaitu komputer *desktop* (Windows XP).



Gambar 4.4. Hasil pengujian komunikasi data TCP pada *smartphone* Android.



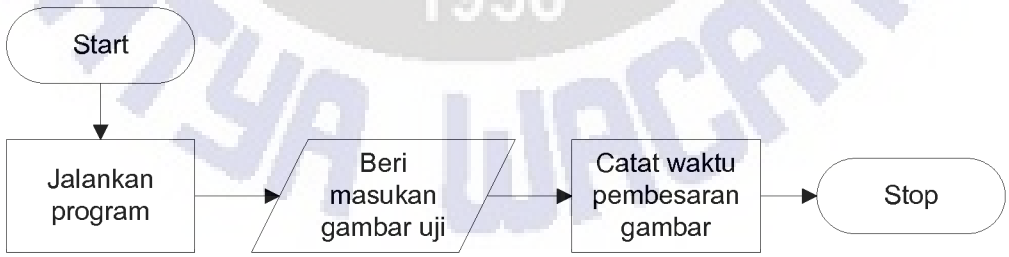
Gambar 4.5. Hasil pengujian komunikasi data TCP pada komputer Windows XP.

Hasil pengujian pada Gambar 4.4 dan Gambar 4.5 menunjukkan bahwa *server* dapat menerima data yang dikirim oleh *client* dengan benar. Hal tersebut dibuktikan dengan data yang ditampilkan oleh *server* sesuai dengan data yang dikirim oleh *client*, dan data tersebut juga dipantau lewat *Wireshark* untuk memastikan tidak terjadi kesalahan saat pengiriman data.

4.3. Pengujian *Scaling* Gambar dengan *Bilinear Interpolation* pada *VGA Adapter*

Pengujian ini dilakukan untuk mengetahui kemampuan modul mikroprosesor OK6410-B pada *VGA Adapter* dalam mengimplementasikan teknik *bilinear interpolation* untuk *scaling* atau pembesaran gambar. Pembesaran gambar ini dilakukan karena data *screen capture* dari aplikasi *mobile* Android memiliki dimensi yang lebih kecil dari 800 x 600 piksel, sehingga sebelum dapat ditampilkan pada keluaran VGA data *screen capture* harus diperbesar terlebih dahulu.

Metode pengujian dilakukan dengan membuat sebuah program pada modul mikroprosesor OK6410-B yang berfungsi untuk memperbesar gambar uji dengan dimensi 480 x 360 dan 640 x 480 piksel menjadi 800 x 600 piksel. Masing-masing terdiri dari 10 gambar uji. Setiap proses pembesaran pada gambar uji akan dicatat waktunya. Dimensi gambar uji ditentukan berdasarkan orientasi layar dengan *aspect ratio* 4:3 relatif terhadap dimensi layar *smartphone* Android yang digunakan dalam pengujian ini. Gambar 4.6 menunjukkan diagram alir untuk pengujian pembesaran gambar dengan *bilinear interpolation*.

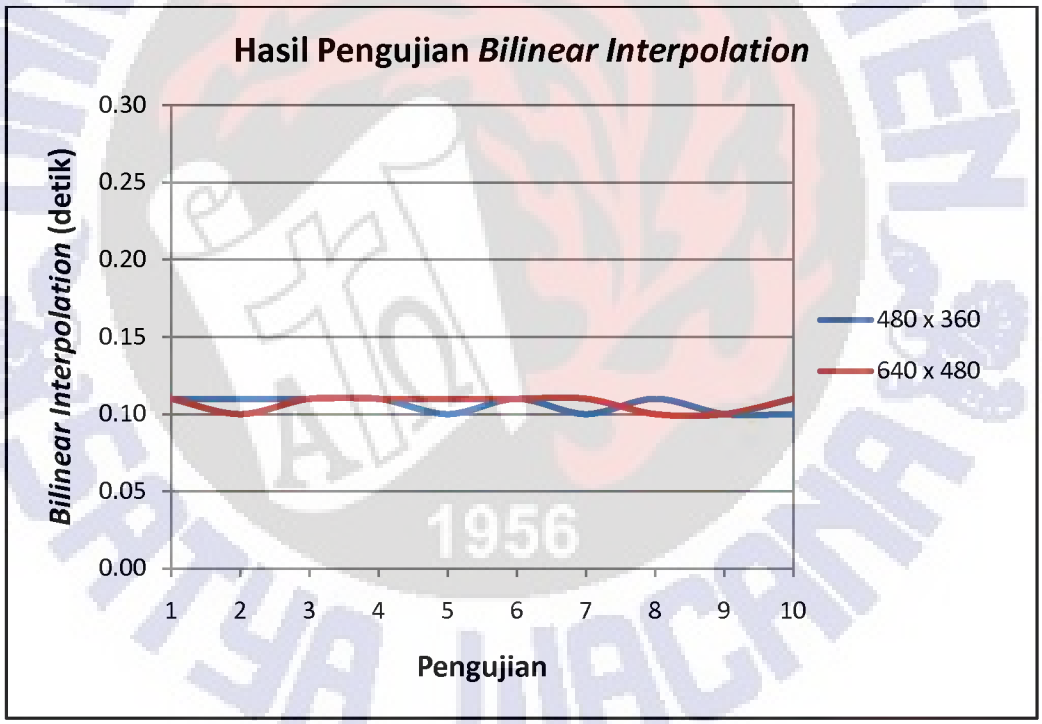


Gambar 4.6. Diagram alir pengujian *bilinear interpolation.*

Tabel 4.2 menunjukkan hasil pengujian pembesaran gambar dengan *bilinear interpolation* pada 10 gambar berdimensi 480 x 360 piksel dan 10 gambar berdimensi 640 x 480 piksel. Hasil pengujian juga ditampilkan dalam sebuah grafik pada Gambar 4.7.

Tabel 4.2. Hasil pengujian *bilinear interpolation.*

Pengujian	*Bilinear Interpolation* (detik)	
	480 x 360 piksel	640 x 480 piksel
1	0.110	0.110
2	0.110	0.100
3	0.110	0.110
4	0.110	0.110
5	0.100	0.110
6	0.110	0.110
7	0.100	0.110
8	0.110	0.100
9	0.100	0.100
10	0.100	0.110
Rata-Rata	**0.106**	**0.107**



Gambar 4.7. Grafik hasil pengujian *bilinear interpolation.*

Dari hasil pengujian dapat diketahui waktu yang diperlukan untuk pembesaran gambar dengan *bilinear interpolation* baik pada gambar berdimensi 480 x 360 dan 640 x 480 piksel relatif sama, yaitu 0.100 sampai 0.110 detik. Hal ini disebabkan faktor pembesaran atau perbandingan antara dimensi awal dengan dimensi setelah pembesaran tidak terlalu

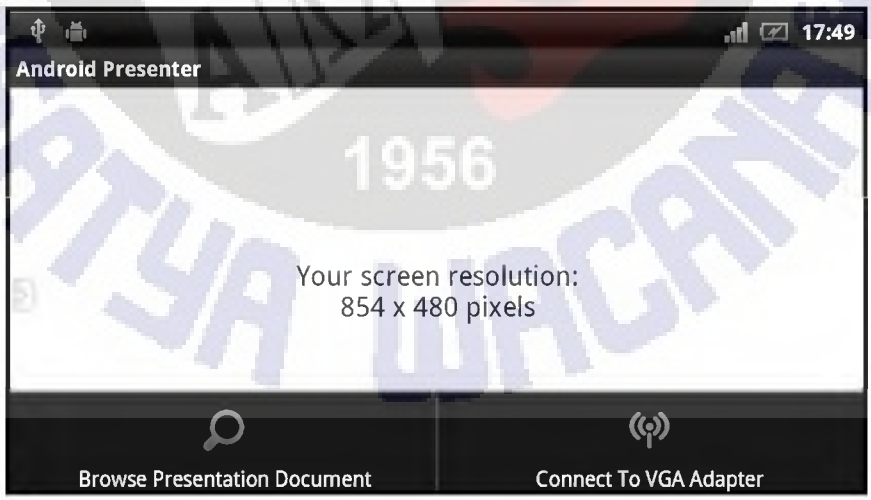
signifikan, yaitu 1.67 untuk dimensi 480 x 360 piksel dan 1.25 untuk dimensi 640 x 480 piksel sehingga didapatkan waktu pembesaran yang relatif sama.

4.4. Pengujian Aplikasi *Mobile* Android

Pengujian ini dilakukan untuk menguji seluruh fungsi aplikasi *mobile* Android yang meliputi *user interface, file explorer, screen capture, socket client* serta performa aplikasi *mobile* Android yaitu pengujian *frame rate* pengiriman data dan pengujian durasi presentasi. Pengujian ini dilakukan pada *smartphone* Sony Ericsson Xperia Neo-V dengan sistem operasi Android versi 2.3.4 dan dimensi layar 480 x 854 piksel.

4.4.1. Pengujian *User Interface*

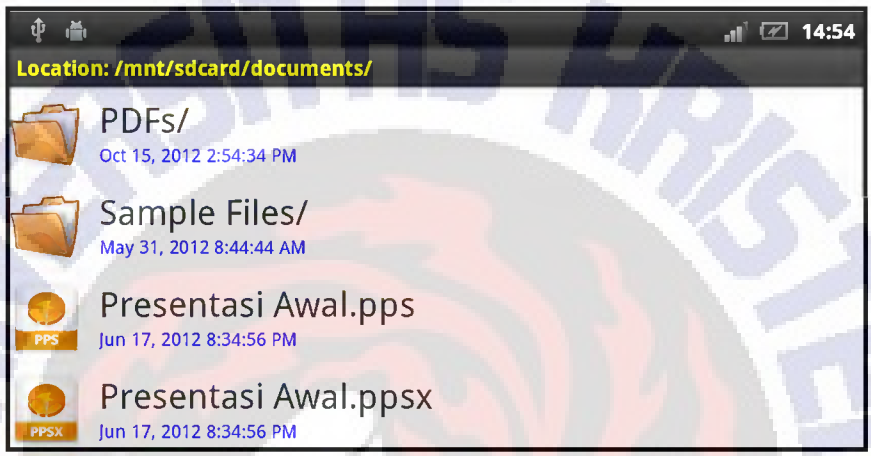
Pengujian *user interface* dilakukan untuk mengetahui apakah fungsi-fungsi yang dibutuhkan oleh pengguna untuk menggunakan perangkat presentasi ini berfungsi dengan baik atau tidak. *User interface* dirancang dalam sebuah *Activity* yang memiliki sebuah menu dengan dua buah pilihan, yaitu pilihan untuk masuk ke *file explorer* untuk mencari dan membuka dokumen presentasi serta pilihan untuk melakukan koneksi *Wi-Fi* dengan *VGA Adapter*. Tampilan *user interface* beserta pilihan menunya ditunjukkan pada Gambar 4.8.



Gambar 4.8. Tampilan *user interface*.

4.4.1.1. Pengujian *File Explorer*

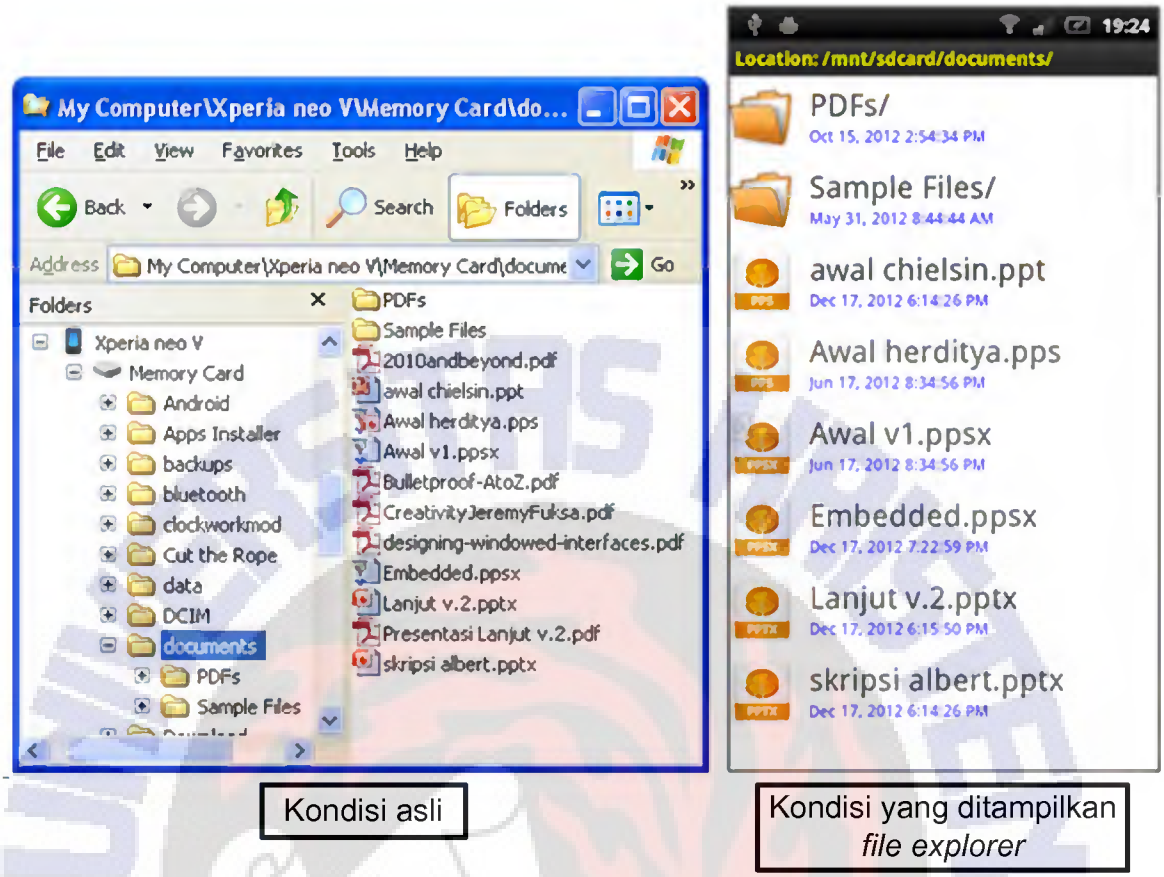
File explorer merupakan tampilan yang berisi daftar *folder* dan konten-konten yang ada di dalam kartu memori *smartphone* Android seperti pada Gambar 4.9. *File explorer* ini dirancang untuk memudahkan pengguna dalam mencari dan membuka dokumen presentasi. Konten yang dapat ditampilkan yaitu dokumen presentasi dengan ekstensi **.ppt, *.pptx, *.pps, *.ppsx*.



Gambar 4.9. Tampilan *file explorer*.

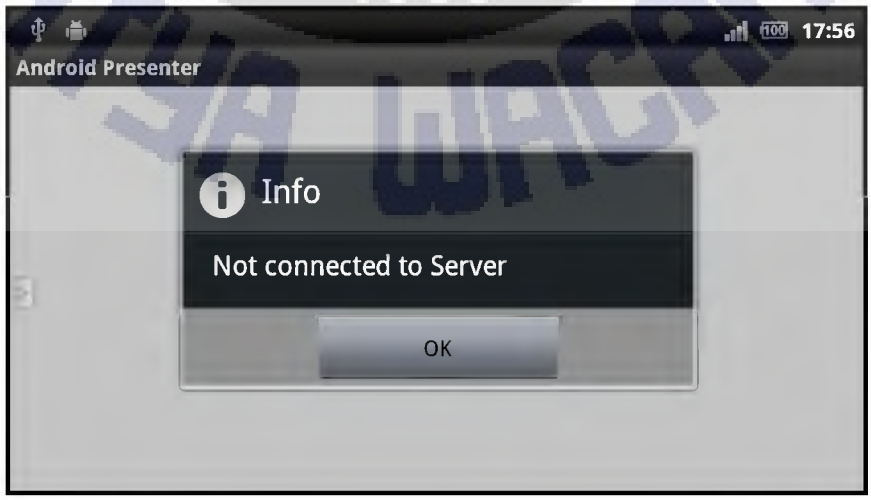
Setiap *folder* ditampilkan dengan sebuah ikon khusus yang melambangkan bentuk *folder* serta memiliki dua informasi yaitu nama dan waktu terakhir *folder* dimodifikasi. Nama *folder* ditampilkan dalam format `Nama_Folder/`. Sedangkan untuk dokumen presentasi ditampilkan dengan ikon khusus yang melambangkan jenis dokumen presentasi berdasarkan ekstensinya, informasi nama dengan format `Nama_Dokumen.ekstensi` dan informasi waktu terakhir dokumen dimodifikasi.

Sesuai dengan spesifikasi yang telah ditentukan, dokumen presentasi yang didukung oleh aplikasi *mobile* Android adalah dokumen dari *Microsoft PowerPoint* dengan ekstensi **.ppt, *.pptx, *.pps, *.ppsx*. Oleh karena itu, dokumen dengan ekstensi lainnya tidak akan ditampilkan pada *file explorer*. Gambar 4.10 memperlihatkan kondisi asli yang menampilkan seluruh konten yang ada di dalam sebuah direktori atau *folder* di dalam kartu memori *smartphone* Android dan kondisi yang ditampilkan oleh *file explorer*.



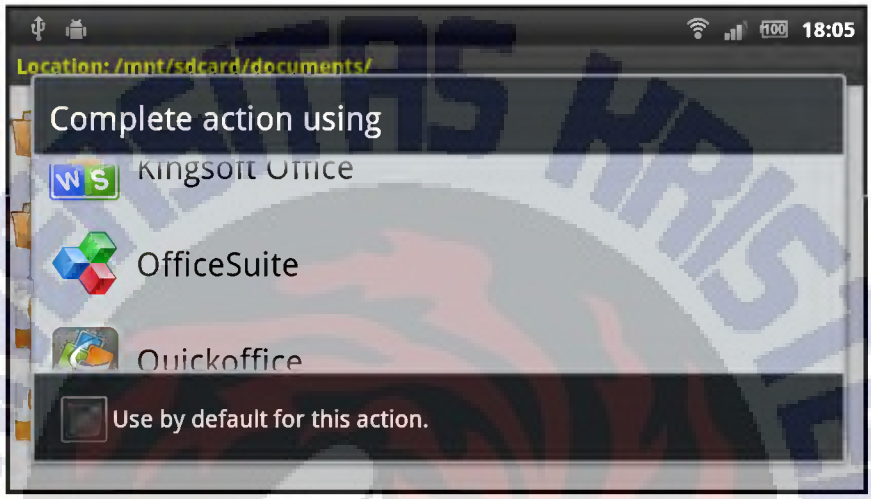
Gambar 4.10. Perbandingan kondisi asli dan kondisi yang ditampilkan *file explorer*.

Untuk dapat masuk ke *file explorer* ini, pengguna harus menghubungkan *smartphone* Android miliknya dengan perangkat *VGA Adapter* terlebih dahulu. Jika belum terhubung *user interface* akan menampilkan dialog peringatan seperti pada Gambar 4.11.



Gambar 4.11. Peringatan belum terhubung ke *VGA Adapter*.

Jika pengguna memilih sebuah dokumen presentasi, aplikasi akan memanggil aplikasi *office* pihak ketiga yang terpasang di dalam sistem untuk membantu menampilkan *slide-slide* presentasi. Jika di dalam sistem operasi Android terdapat lebih dari satu aplikasi *office*, *user interface* akan memberikan pilihan aplikasi mana yang ingin digunakan untuk menampilkan dokumen presentasi seperti terlihat pada Gambar 4.12.

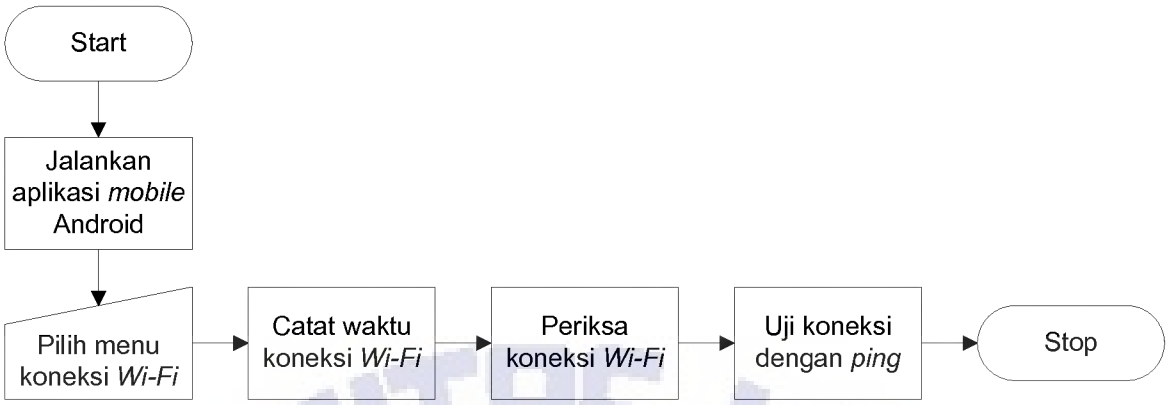


Gambar 4.12. Pilihan aplikasi *office* yang dapat dipilih pengguna.

4.4.1.2. Pengujian Koneksi *Wi-Fi* dengan *VGA Adapter*

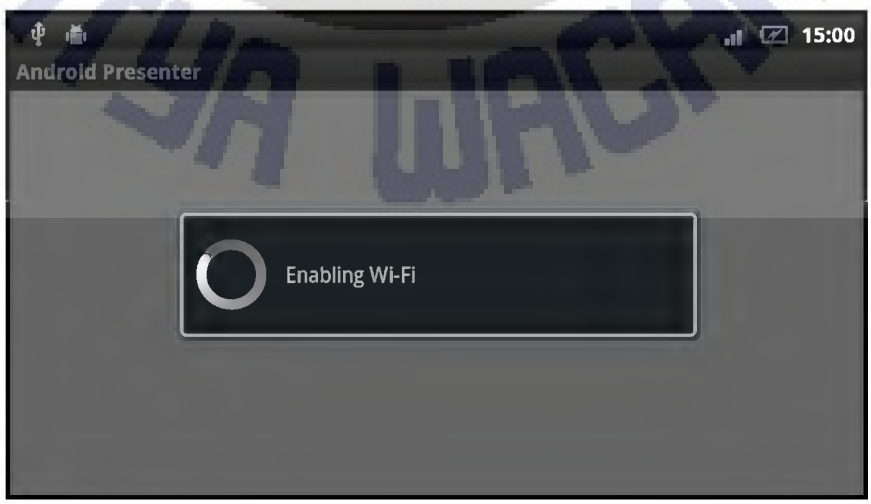
Pengujian ini digunakan untuk mengetahui kemampuan aplikasi *mobile* pada *smartphone* Android dalam melakukan koneksi ke jaringan *Wi-Fi* milik perangkat *VGA Adapter*. Aplikasi *mobile* Android harus mampu melakukan koneksi ke jaringan *Wi-Fi* milik *VGA Adapter* secara otomatis karena pengguna tidak perlu mengetahui SSID dan *password* untuk masuk ke jaringan. Kedua parameter tersebut sudah dimiliki oleh aplikasi *mobile* Android dan proses masuk ke jaringan dilakukan tanpa ada interaksi dari pengguna, dengan terlebih dahulu menyalakan perangkat *Wi-Fi* internal pada *smartphone* Android.

Metode pengujian dilakukan dengan menjalankan *user interface* dari aplikasi *mobile* Android, kemudian memilih menu untuk koneksi *Wi-Fi* dan setiap koneksi yang berhasil akan dicatat waktunya. Untuk memeriksa apakah perangkat Android sudah terkoneksi dengan jaringan *Wi-Fi* milik *VGA Adapter* dapat menggunakan menu *Settings* → *Wireless & networks* → *Wi-Fi settings* → *Wi-Fi networks* pada sistem operasi Android. Langkah terakhir adalah menguji koneksi dengan melakukan *ping* ke *host* lewat aplikasi *Terminal Emulator*. Gambar 4.13 merupakan diagram alir untuk metode pengujian koneksi *smartphone* Android dengan *VGA Adapter*.

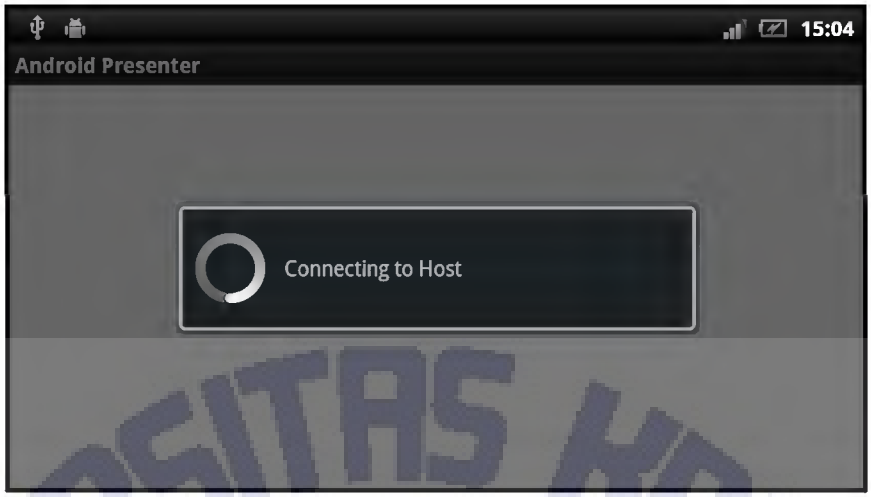


Gambar 4.13. Diagram alir pengujian koneksi *Wi-Fi* pada *smartphone* Android.

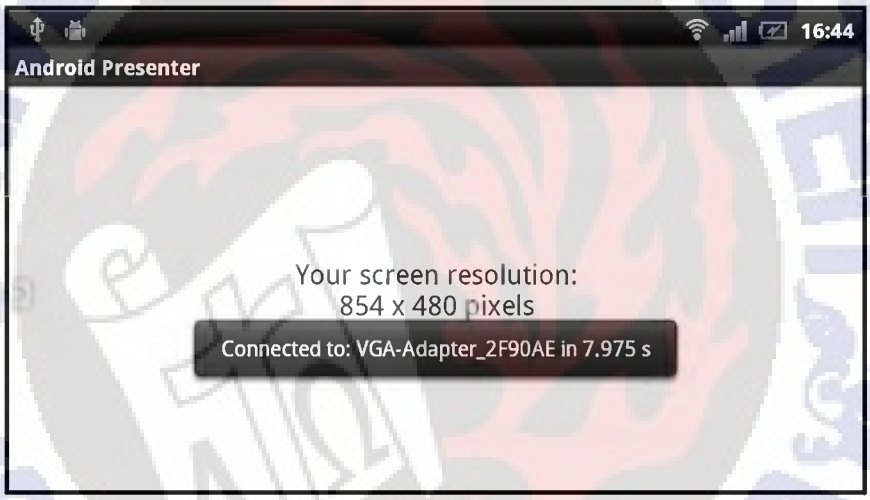
Saat menu koneksi *Wi-Fi* dipilih maka aplikasi akan menyalakan perangkat *Wi-Fi* internal Android terlebih dulu (Gambar 4.14), kemudian aplikasi akan mencari *Service Set Identifier* (SSID) atau nama jaringan *Wi-Fi* dari *VGA Adapter* (Gambar 4.15). SSID *VGA Adapter* memiliki format "`VGA-Adapter_xxxxxx`" dengan "`xxxxxx`" merupakan enam digit terakhir alamat *Media Access Control* (MAC) *Wi-Fi Access Point* pada *VGA Adapter*. *Password* untuk masuk ke jaringan *Wi-Fi* yaitu "`herditya`" disediakan dan dimasukkan secara otomatis oleh aplikasi *mobile*. Gambar 4.16 menunjukkan jika koneksi *Wi-Fi* berhasil dilakukan dan Gambar 4.17 menunjukkan informasi detail koneksi tersebut pada sistem operasi Android. Gambar 4.18 menunjukkan hasil uji koneksi *ping* lewat *Terminal Emulator* sehingga dapat diketahui bahwa *smartphone* Android sebagai *client* berada dalam satu jaringan dengan *VGA Adapter* sebagai *server* dan sudah terkoneksi dengan baik.



Gambar 4.14. Tampilan saat proses menyalakan *Wi-Fi* pada *smartphone* Android.

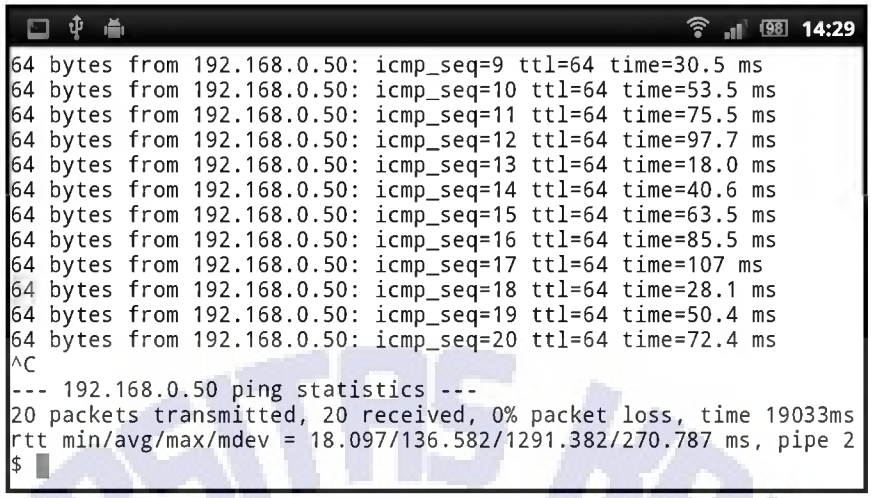


Gambar 4.15. Tampilan saat koneksi ke *host* (*VGA Adapter*).



Gambar 4.16. Koneksi *Wi-Fi* berhasil dilakukan.

Gambar 4.17. Informasi koneksi *Wi-Fi* pada sistem operasi Android.



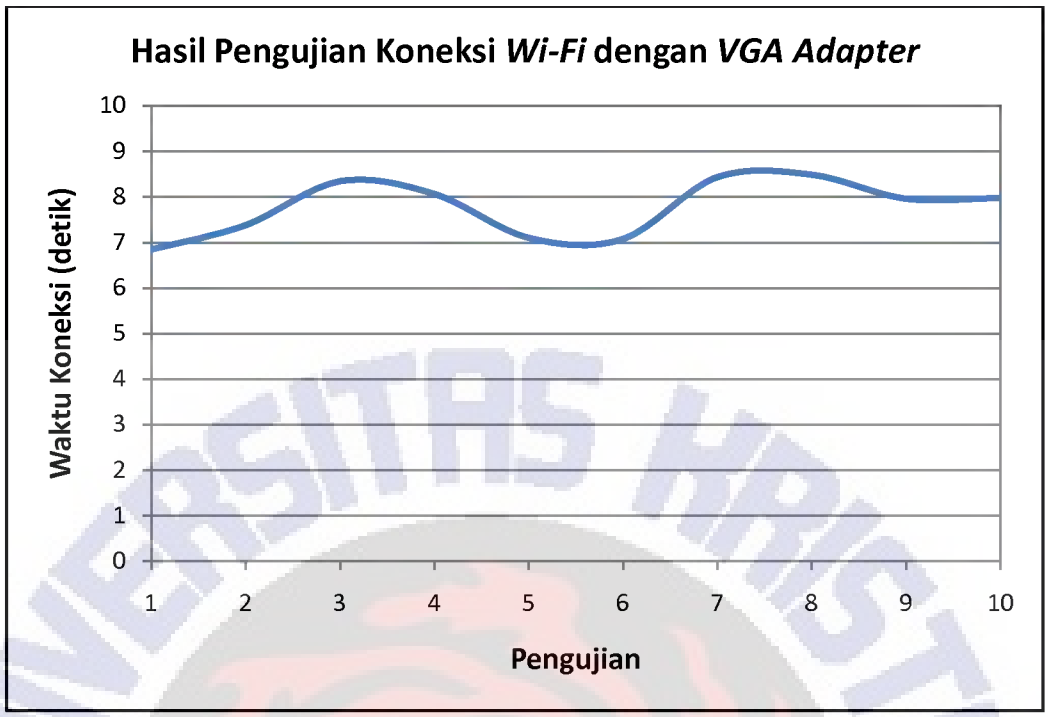
Gambar 4.18. Uji koneksi *Wi-Fi* menggunakan aplikasi *Terminal Emulator*.

Tabel 4.3 menunjukkan lamanya waktu yang diperlukan oleh aplikasi *mobile* Android untuk melakukan koneksi *Wi-Fi* dengan *VGA Adapter*, termasuk di dalamnya waktu yang diperlukan untuk menyalakan perangkat *Wi-Fi* internal *smartphone*. Waktu koneksi ini dihitung mulai saat menu koneksi *Wi-Fi* pada *user interface* dipilih sampai koneksi dengan *VGA Adapter* berhasil dilakukan. Pencatatan waktu dilakukan dalam 10 kali pengujian.

Tabel 4.3. Waktu koneksi aplikasi *mobile* Android dengan *VGA Adapter*.

Pengujian	Waktu Koneksi (detik)
1	6.845
2	7.380
3	8.343
4	8.067
5	7.101
6	7.073
7	8.434
8	8.491
9	7.961
10	7.975
Rata-Rata	**7.767**

Hasil pengujian pada Tabel 4.3 disajikan dalam sebuah grafik seperti pada Gambar 4.19.

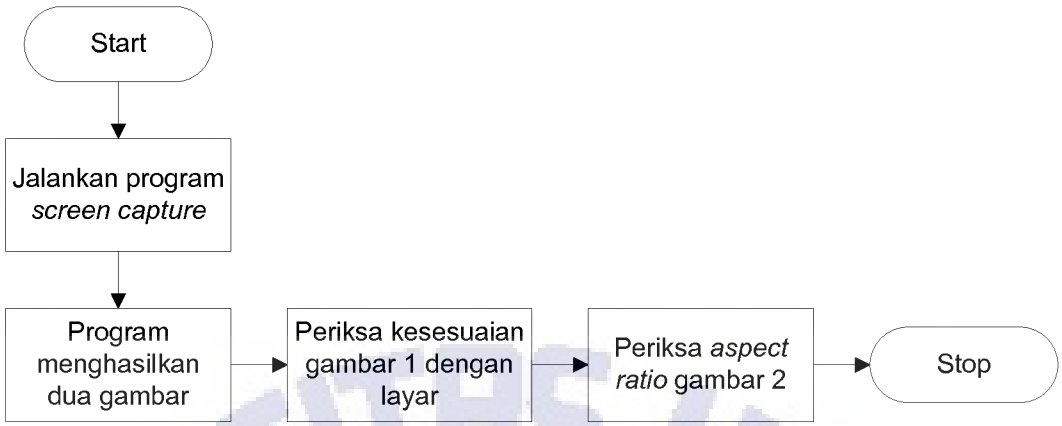


Gambar 4.19. Grafik hasil pengujian koneksi *Wi-Fi* dengan *VGA Adapter*.

Dari hasil pengujian diperoleh waktu rata-rata yang diperlukan oleh aplikasi *mobile* untuk melakukan koneksi dengan *VGA Adapter* yaitu 7.767 detik. Waktu tersebut dapat bervariasi tergantung pada spesifikasi *smartphone* yang digunakan serta kondisi jaringan *Wi-Fi* dari *VGA Adapter*, termasuk jarak antara *smartphone* Android dengan *VGA Adapter*.

4.4.2. Pengujian *Screen Capture*

Pengujian ini dilakukan untuk mengetahui kemampuan aplikasi *mobile* Android dalam melakukan fungsi *screen capture* yang merupakan fungsi utama dalam perancangan skripsi ini. Metode pengujian dilakukan dengan membuat sebuah program sederhana yang berfungsi melakukan *capturing* terhadap layar *smartphone*. Hasil *screen capture* disimpan dalam dua buah gambar *bitmap*. Gambar pertama (`satu.bmp`) merupakan hasil *capturing* asli dengan dimensi gambar sesuai dengan dimensi layar. Gambar kedua (`dua.bmp`) merupakan hasil dari transformasi seperti yang telah dijelaskan pada Bab III (*cropping, flipping, rotating, color rearrangements*) dengan dimensi gambar memiliki *aspect ratio* 4:3 relatif terhadap dimensi layar. Diagram alir untuk metode pengujian *screen capture* dapat dilihat pada Gambar 4.20.

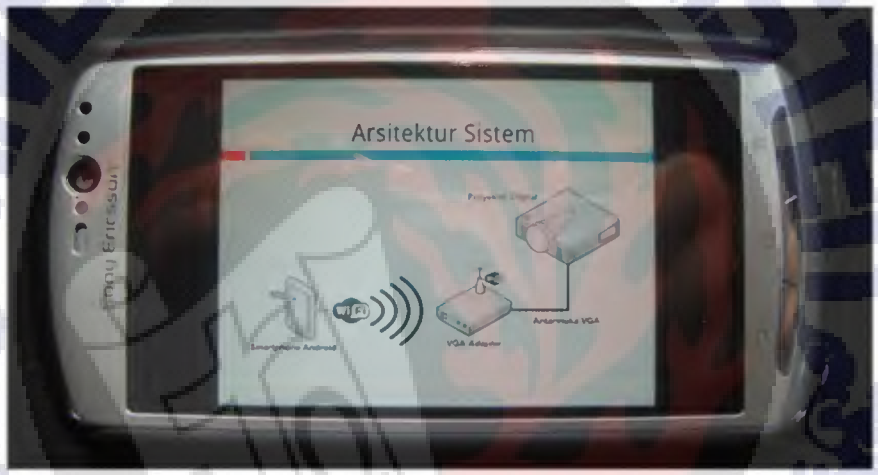


Gambar 4.20. Diagram alir pengujian *screen capture*.

Kode sumber program untuk pengujian *screen capture* dibuat menggunakan bahasa *native* C dan dikompilasi menggunakan NDK Android. Kode sumber tersebut dikompilasi menjadi sebuah berkas *executable* dan dijalankan lewat *command shell* Android. Pada kode sumber menggunakan pustaka *QDBMP* untuk keperluan *encoding* gambar *bitmap*.

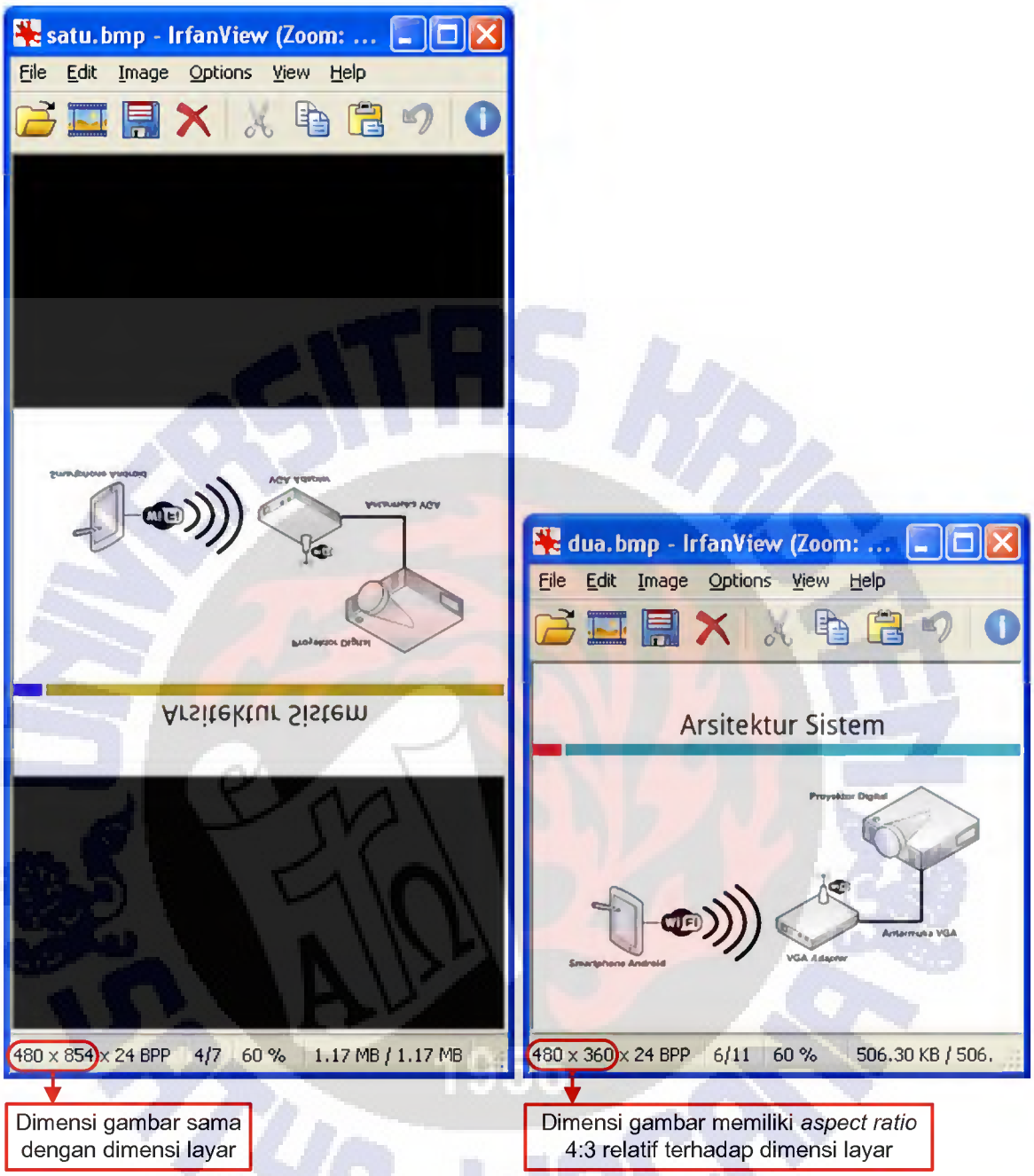
Pengujian ini dilakukan sebanyak dua kali, masing-masing untuk setiap orientasi layar *smartphone* yaitu p*ortrait* dan *landscape*. Hal ini perlu dilakukan karena *screen capture* memiliki proses dan perhitungan yang berbeda untuk masing-masing orientasi layar. Gambar 4.21 menunjukkan kondisi layar aktual pada masing-masing orientasi layar saat pengujian dilakukan. Gambar 4.22 merupakan hasil pengujian *screen capture* pada orientasi layar *portrait*, sedangkan Gambar 4.23 merupakan hasil pengujian *screen capture* pada orientasi layar *landscape*.

Orientasi layar *portrait*

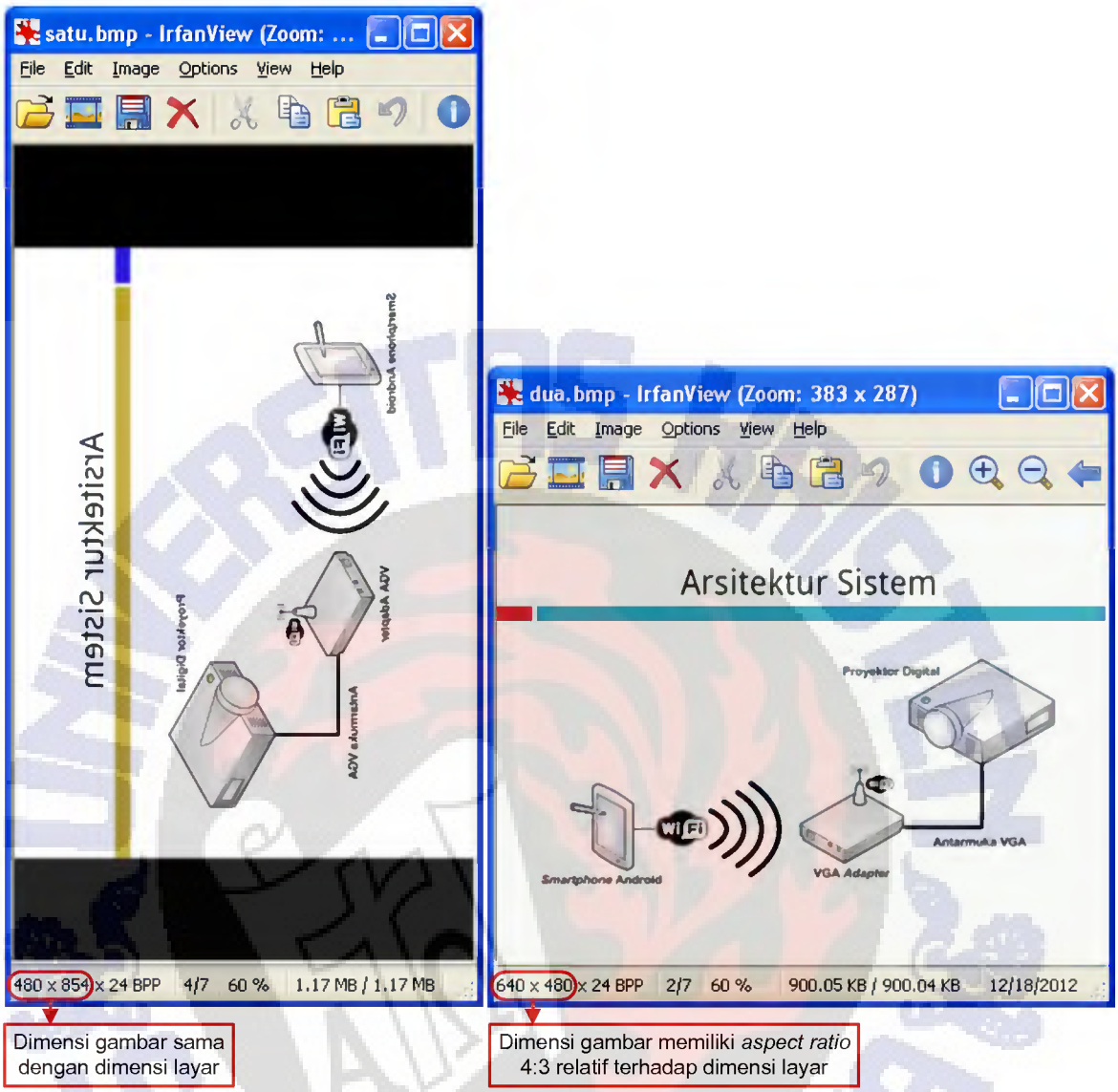


Orientasi layar *landscape*

Gambar 4.21. Kondisi layar aktual pada masing-masing orientasi layar.



Gambar 4.22. Hasil pengujian *screen capture* pada orientasi layar *portrait.*



Gambar 4.23. Hasil pengujian *screen capture* pada orientasi layar *landscape*.

Dari hasil pengujian dapat dilihat bahwa antara kondisi layar aktual dengan gambar hasil *screen capture* yang asli memiliki tampilan yang sedikit berbeda. Oleh karena itu diperlukan beberapa transformasi agar hasil *screen capture* memiliki tampilan yang sama dengan kondisi layar aktual. Selain untuk mendapatkan tampilan yang sama, transformasi tersebut juga bertujuan untuk menghasilkan keluaran berupa data gambar dengan *aspect ratio* 4:3 relatif terhadap dimensi layar.

4.4.3. Pengujian *Socket Client*

Pengujian ini dilakukan untuk mengetahui kemampuan aplikasi *mobile* Android sebagai *client* dalam mengirimkan data *screen capture* ke *VGA Adapter* sebagai *server*. Pengujian ini perlu dilakukan untuk memastikan bahwa aplikasi *mobile* Android dapat melakukan pengiriman data *screen capture* dengan benar dan lengkap.

Metode pengujian dilakukan dengan membuat dua program sederhana yang masing-masing bertindak sebagai *client* dan *server*. Program pada *client* berfungsi untuk men-*decode* sebuah gambar *bitmap* menjadi *array* RGB untuk dikirimkan ke *server*. Sedangkan program pada *server* digunakan untuk menerima *array* RGB dari *client* dan di-*encode* menjadi sebuah gambar *bitmap* kembali. Program pada *client* dan *server* keduanya menggunakan pustaka *QDBMP* untuk *encoding-decoding* gambar *bitmap* dan menggunakan pemrograman *socket* untuk menangani transmisi data.

Pengujian dilakukan pada 20 gambar *bitmap* yang terdiri dari 10 gambar dengan dimensi 480 x 360 piksel dan 10 gambar dengan dimensi 640 x 480 piksel. Dimensi gambar uji tersebut ditentukan berdasarkan orientasi layar dan dimensi dengan *aspect ratio* 4:3 relatif terhadap dimensi layar *smartphone* Android yang digunakan dalam pengujian ini. Selanjutnya dilakukan pencatatan waktu terhadap proses-proses yang dilakukan oleh *client* (*decoding* gambar dan pengiriman data RGB) dan *server* (penerimaan data RGB dan *encoding* gambar). Langkah terakhir adalah memastikan bahwa gambar yang dihasilkan oleh *server* sama dengan gambar yang dikirim *client*. Gambar 4.24 menunjukkan diagram alir untuk pengujian *socket client*.

Gambar 4.24. Diagram alir pengujian *socket client*.

Hasil pengujian *socket client* pada 10 gambar uji dengan dimensi 480 x 360 piksel ditunjukkan pada Tabel 4.4, sedangkan hasil pengujian untuk 10 gambar uji dengan dimensi 640 x 480 piksel ditunjukkan pada Tabel 4.5.

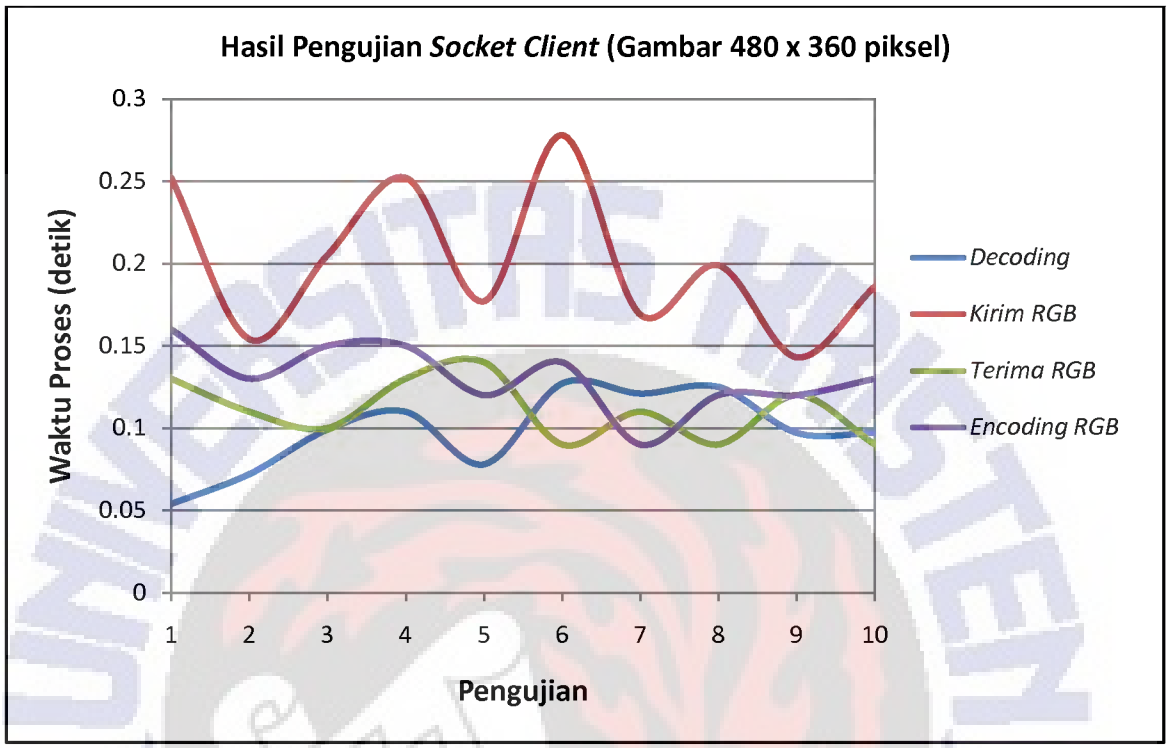
Tabel 4.4. Hasil pengujian *socket client* pada gambar berdimensi 480 x 360 piksel.

Nomor Gambar	Dimensi Gambar (piksel)	Kesesuaian Gambar	Waktu Proses (detik)			
			Client (*Smartphone* Android)		*Server* (*VGA Adapter*)	
			Decoding	Kirim RGB	Terima RGB	*Encoding*
1	480 x 360	✓	0.054	0.252	0.130	0.160
2	480 x 360	✓	0.072	0.154	0.110	0.130
3	480 x 360	✓	0.099	0.205	0.100	0.150
4	480 x 360	✓	0.110	0.252	0.130	0.150
5	480 x 360	✓	0.078	0.177	0.140	0.120
6	480 x 360	✓	0.127	0.278	0.090	0.140
7	480 x 360	✓	0.121	0.169	0.110	0.090
8	480 x 360	✓	0.125	0.199	0.090	0.120
9	480 x 360	✓	0.097	0.143	0.120	0.120
10	480 x 360	✓	0.097	0.186	0.090	0.130
Waktu Rata-Rata			**0.098**	**0.202**	**0.111**	**0.131**

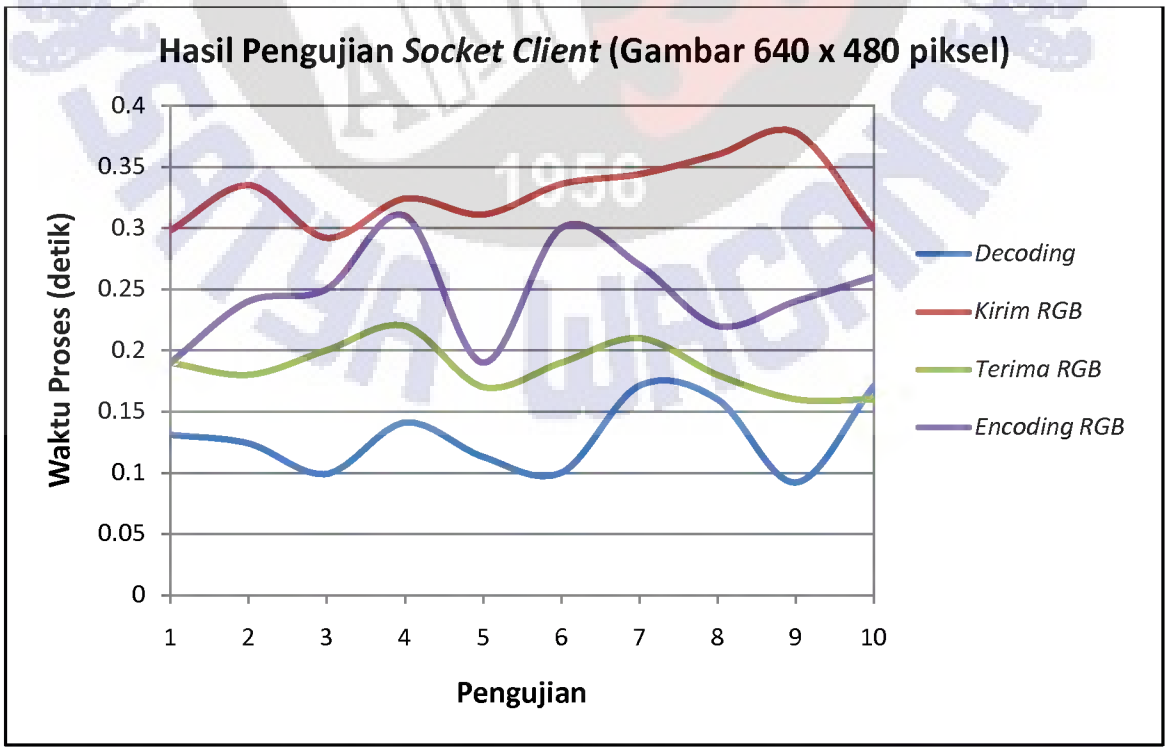
Tabel 4.5. Hasil pengujian *socket client* pada gambar berdimensi 640 x 480 piksel.

Nomor Gambar	Dimensi Gambar (piksel)	Kesesuaian Gambar	Waktu Proses (detik)			
			Client (*Smartphone* Android)		*Server* (*VGA Adapter*)	
			Decoding	Kirim RGB	Terima RGB	*Encoding*
1	640 x 480	✓	0.131	0.298	0.190	0.190
2	640 x 480	✓	0.124	0.335	0.180	0.240
3	640 x 480	✓	0.099	0.292	0.200	0.250
4	640 x 480	✓	0.141	0.324	0.220	0.310
5	640 x 480	✓	0.113	0.311	0.170	0.190
6	640 x 480	✓	0.100	0.336	0.190	0.300
7	640 x 480	✓	0.171	0.344	0.210	0.270
8	640 x 480	✓	0.160	0.360	0.180	0.220
9	640 x 480	✓	0.092	0.378	0.160	0.240
10	640 x 480	✓	0.171	0.299	0.160	0.260
Waktu Rata-Rata			**0.130**	**0.328**	**0.186**	**0.247**

Hasil pengujian pada Tabel 4.4 dan Tabel 4.5 masing-masing disajikan dalam bentuk grafik seperti pada Gambar 4.25 dan Gambar 4.26.



Gambar 4.25. Grafik hasil pengujian *socket client* (gambar 480 x 360 piksel).



Gambar 4.26. Grafik hasil pengujian *socket client* (gambar 640 x 480 piksel).

Menurut hasil pengujian, dapat diketahui bahwa aplikasi *mobile* Android dapat mengirimkan gambar ke *VGA Adapter* dengan benar dan lengkap, antara gambar yang dikirim *client* dengan gambar yang dihasilkan *server* tidak ada perbedaan. Lewat pengujian ini juga dapat diketahui bahwa proses *encoding-decoding* gambar *bitmap* cukup memakan waktu saat dijalankan, sehingga pada perancangan skripsi (Bab III) hasil *screen capture* tidak disimpan sebagai gambar *bitmap* melainkan disimpan sebagai *array* RGB. Hal ini bertujuan untuk menghindari proses *encoding-decoding* gambar bitmap yang dapat menyebabkan penurunan *frame rate* pengiriman gambar dari aplikasi *mobile* Android ke *VGA Adapter*.

4.4.4. Pengujian *Frame Rate*

Pengujian *frame rate* digunakan untuk mengetahui banyaknya *frame* yang dapat ditampilkan *VGA Adapter* pada keluaran VGA dalam satu detik. Satuan yang biasa digunakan adalah *frame per second* atau FPS. *Frame* yang dimaksud disini adalah sebuah data *screen capture* dari aplikasi *mobile* Android yang ditampilkan oleh *VGA Adapter* pada keluaran VGA. Hasil pengujian ini akan memperlihatkan seberapa responsif *VGA Adapter* dalam menampilkan perubahan yang terjadi pada tampilan layar *client*.

Metode pengujian dilakukan dengan menambahkan beberapa baris program pada perangkat lunak *VGA Adapter* untuk menghitung selisih waktu antar tiap *frame* yang ditampilkan. Nilai selisih waktu munculnya tiap *frame* ini dicatat kemudian dirata-rata setelah didapatkan 10 nilai, dan perhitungan nilai rata-rata ini dilakukan sebanyak 10 kali. Kode 4.1 menunjukkan potongan kode untuk menghitung waktu eksekusi program yang digunakan untuk menghitung selisih waktu munculnya tiap *frame*.

```
#include <sys/time.h>
...
struct timeval tv1, tv2;
gettimeofday(&tv1, NULL );
/*
     do some stuff here
*/
gettimeofday(&tv2, NULL );
printf("%f\n",(double) (tv2.tv_usec - tv1.tv_usec) / 1000000 +
               (double) (tv2.tv_sec - tv1.tv_sec));
...
```

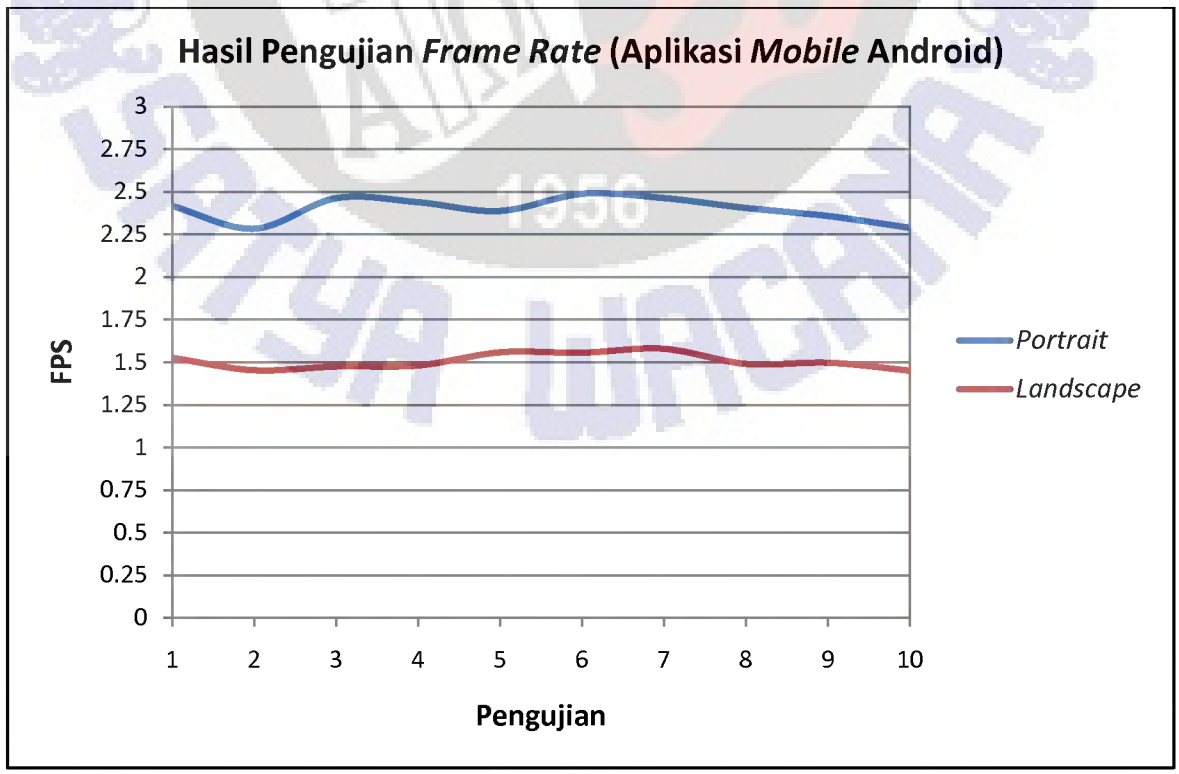
Kode 4.1. Potongan kode untuk menghitung waktu eksekusi program.

Hasil pengujian *frame rate* untuk data *screen capture* dari aplikasi *mobile* Android ditunjukkan pada Tabel 4.6.

Tabel 4.6. Hasil pengujian *frame rate* pada aplikasi *mobile* Android.

Pengujian	*Frame Rate* (FPS – *frame per second*)	
	Portrait (480 x 360)	*Landscape* (640 x 480)
1	2.415	1.525
2	2.283	1.453
3	2.463	1.475
4	2.439	1.481
5	2.387	1.558
6	2.488	1.555
7	2.463	1.579
8	2.404	1.490
9	2.358	1.497
10	2.288	1.451
Rata-Rata	**2.399**	**1.506**

Hasil pengujian pada Tabel 4.6 disajikan dalam bentuk grafik seperti pada Gambar 4.27.



Gambar 4.27. Grafik hasil pengujian *frame rate* untuk aplikasi *mobile* Android.

Dari hasil pengujian yang didapat sesuai dengan Tabel 4.6 dan grafik pada Gambar 4.27, *frame rate* yang dicapai pada orientasi layar *portrait* yaitu 2.2 sampai 2.5 FPS, sedangkan untuk orientasi layar *landscape* yaitu 1.4 sampai 1.6 FPS. Artinya dalam satu detik minimal ada satu *frame* yang ditampilkan oleh *VGA Adapter*. *Frame rate* ini dirasa sudah cukup untuk melakukan presentasi dengan *smartphone* Android, dengan asumsi bahwa aplikasi-aplikasi *office* untuk *smartphone* Android pada saat skripsi ini dibuat tidak bisa menampilkan animasi atau transisi objek yang biasa terdapat pada dokumen presentasi yang dibuat dengan *Microsoft PowerPoint*.

Faktor yang paling mempengaruhi nilai *frame rate* adalah besarnya *latency* saat pengiriman data *screen capture* menggunakan protokol TCP. Dari Tabel 4.4 dan Tabel 4.5 yang berisi hasil pengujian *socket client*, dapat diketahui jika *latency* yang paling besar saat transmisi data ada pada *smartphone* Android (kolom kirim RGB dan terima RGB). Hal ini disebabkan karena Android merupakan sistem operasi yang kompleks dan memiliki banyak *task* yang harus dikerjakan dalam satu waktu. Setiap *task* memiliki prioritas masing-masing dan saling berbagi sumber daya yang jumlahnya terbatas. CPU dan RAM adalah sumber daya yang digunakan pada semua *task* dalam sebuah sistem operasi. Keterbatasan sumber daya inilah yang dapat menyebabkan sebuah *task* membutuhkan waktu eksekusi yang lebih panjang daripada waktu eksekusi normal.

4.4.5. Pengujian Durasi Presentasi

Pengujian ini dilakukan untuk mengetahui durasi atau berapa lama presentasi dapat dilakukan dengan perangkat presentasi yang dirancang dalam skripsi ini. Sesuai dengan spesifikasi yang telah ditentukan, presentasi harus bisa dilakukan selama minimal dua jam menggunakan *smartphone* Android dan perangkat *VGA Adapter*.

Metode pengujian dilakukan dengan cara menghubungkan *smartphone* Android dengan *VGA Adapter*, kemudian menjalankan aplikasi *Stopwatch* pada *smartphone* Android. Setelah aplikasi *Stopwatch* berjalan, aplikasi *mobile* Android akan meng-*capture* tampilan aplikasi *Stopwatch* dan mengirimkannya ke *VGA Adapter* untuk selanjutnya ditampilkan pada keluaran VGA. Dengan demikian dapat diketahui berapa lama sistem presentasi sudah berjalan. Langkah pengujian ini ditunjukkan pada Gambar 4.28.

Gambar 4.28. Pengujian durasi presentasi dengan aplikasi *Stopwatch*.

Langkah selanjutnya yaitu memeriksa lamanya waktu *service* dari aplikasi *mobile* Android yang dijalankan dalam sistem operasi Android. Informasi dari seluruh *service* yang berjalan dalam sistem operasi Android dapat dilihat pada menu *Settings* → *Applications* → *Running services* seperti pada Gambar 4.29.

Gambar 4.29. Penunjukan durasi *service* dari aplikasi *mobile* Android.

Dari pengujian ini dapat diketahui bahwa sistem presentasi dapat dijalankan selama dua jam. Artinya proses-proses yang dikerjakan mulai dari *screen capture* pada *smartphone* Android sampai proses menampilkannya pada keluaran VGA tidak mengalami hambatan selama dua jam.

4.5. Pengujian Aplikasi *Desktop*

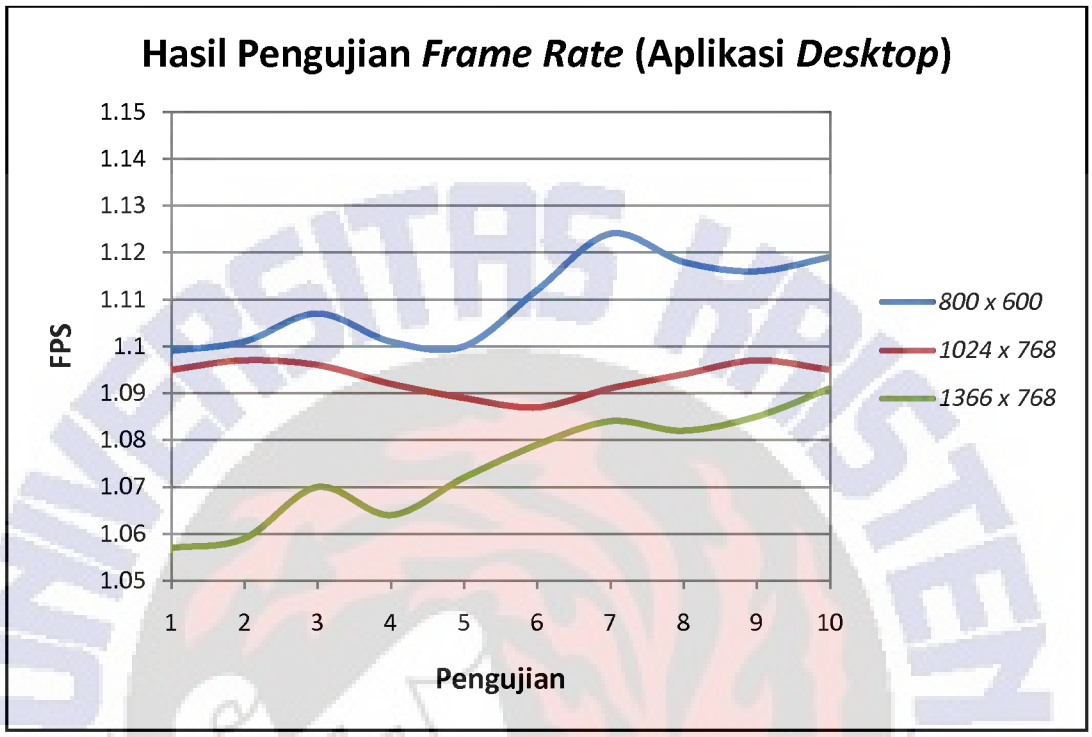
Aplikasi *desktop* digunakan untuk menguji apakah sistem presentasi yang dirancang pada skripsi ini dapat dijalankan juga pada komputer atau *notebook*. Pada dasarnya aplikasi *desktop* memiliki fungsi serupa dengan aplikasi *mobile* Android. Pengujian yang dilakukan pada aplikasi *desktop* hanya pengujian *frame rate* yang bertujuan untuk membandingkan *frame rate* yang dicapai oleh aplikasi *desktop* dengan *frame rate* yang dicapai oleh aplikasi *mobile* Android.

Metode pengujian *frame rate* aplikasi *desktop* sama seperti pengujian *frame rate* pada aplikasi *mobile* Android. Setiap *frame* yang ditampilkan *VGA Adapter* dihitung selisih waktunya dengan *frame* sebelumnya. Tiap didapat 10 nilai selisih waktu kemudian dihitung rata-ratanya, dan dilakukan sampai didapat 10 nilai rata-rata selisih waktu antar *frame*. Pengujian ini dilakukan pada sebuah *notebook* dengan tiga resolusi berbeda, yaitu 1366 x 768 piksel, 1024 x 768 piksel dan 800 x 600 piksel. Hal tersebut bertujuan untuk mengetahui pengaruh dari besar kecilnya resolusi terhadap nilai *frame rate* yang dicapai. Hasil pengujian *frame rate* aplikasi *desktop* ditunjukkan pada Tabel 4.7.

Tabel 4.7. Hasil pengujian *frame rate* pada aplikasi *desktop*.

Pengujian	*Frame Rate* (FPS – *frame per second*)		
	800 x 600	1024 x 768	1366 x 768
1	1.099	1.095	1.057
2	1.101	1.097	1.059
3	1.107	1.096	1.070
4	1.101	1.092	1.064
5	1.100	1.089	1.072
6	1.112	1.087	1.079
7	1.124	1.091	1.084
8	1.118	1.094	1.082
9	1.116	1.097	1.085
10	1.119	1.095	1.091
Rata-Rata	**1.109**	**1.093**	**1.074**

Hasil pengujian pada Tabel 4.7 disajikan dalam bentuk grafik seperti pada Gambar 4.30.



Gambar 4.30. Grafik hasil pengujian *frame rate* aplikasi *desktop*.

Dari hasil pengujian aplikasi *desktop*, dapat diketahui jika besar kecilnya resolusi ikut mempengaruhi *frame rate* yang dicapai. Sesuai pada diagram alir aplikasi *desktop* pada Gambar 3.13 (Bab III), jika data *screen capture* memiliki dimensi lebih dari 800 x 600 piksel maka data *screen capture* diperkecil (*scaling*) terlebih dahulu menjadi 800 x 600 piksel menggunakan teknik *bilinear interpolation*. Waktu yang dibutuhkan untuk proses *scaling* tersebut yang menyebabkan penurunan pada *frame rate* yang dicapai oleh aplikasi *desktop*.

Selain karena proses *scaling*, proses lain yang turut menyebabkan penurunan *frame rate* adalah proses *cropping* untuk resolusi dengan *ascpect ratio* lebih besar dari 4:3. Pada grafik di atas terlihat *frame rate* untuk resolusi 1366 x 768 piksel lebih rendah daripada resolusi 1024 x 768 piksel. Hal ini disebabkan karena resolusi 1366 x 768 piksel memiliki *aspect ratio* 16:9, sehingga sebelum diperkecil data *screen capture* harus di-*crop* terlebih dahulu untuk mendapatkan data gambar dengan *aspect ratio* 4:3.