

PEMBUATAN SYSTEM REMOTE DESKTOP DARI MOBILE KE PC TANPA HUBUNGAN END TO END

Esther Irawati Setiawan, Stefanus Christofan

Teknik Informatika

Sekolah Tinggi Teknik Surabaya

esther@stts.edu, sonicver2@gmail.com

ABSTRAK

Mobile device smart phone, yang dapat melakukan berbagai hal, seperti Android merupakan suatu gadget yang umum dipakai oleh masyarakat sekarang. Namun terdapat keterbatasan fungsionalitas yang dapat dilakukan pada Desktop PC dan tidak dapat dilakukan oleh sebuah smart phone. Ada pula saat di mana pengguna komputer membutuhkan akses pada Desktop PC-nya, tetapi pengguna tersebut tidak berada pada tempat PC tersebut. Internet dan teknologi Remote Desktop dapat membantu menyelesaikan permasalahan tersebut, hanya saja Remote Desktop pada umumnya membutuhkan Desktop PC/Laptop untuk mengakses komputer yang dibutuhkan secara remote. Permasalahan lain yang dihadapi oleh aplikasi Remote Desktop yang sudah ada adalah keterbatasan dalam akses komputer tujuan, di mana komputer tujuan harus berada pada satu jaringan yang sama atau membutuhkan adanya IP publik pada komputer tersebut.

Aplikasi remote desktop dikembangkan pada mobile device berbasis Android, dimana dengan aplikasi ini pengguna dapat mengakses komputernya secara remote dimanapun dan kapanpun menggunakan handphone Android. Fitur-fitur utama dari aplikasi remote desktop ini adalah connect and login untuk pembukaan koneksi, interact live untuk mode remote dengan kontrol penuh, view live untuk mode remote tanpa kontrol, view snapshot untuk melihat satu screenshot saja, shortcut command untuk menjalankan perintah shortcut yang sudah diatur, dan control processes untuk melihat task manager.

Aplikasi remote desktop ini memiliki 3 komponen utama, yaitu aplikasi mobile, aplikasi web, dan aplikasi desktop. Aplikasi mobile adalah komponen yang menampilkan gambar dan menerima inputan pengguna, akan diletakkan pada handphone Android. Aplikasi desktop adalah komponen yang melakukan capture screenshot dan mengirimkannya untuk ditampilkan pada aplikasi mobile, sekaligus menerima input perintah untuk dijalankan, akan diletakkan pada komputer tujuan remote. Aplikasi web adalah komponen perantara yang berfungsi sebagai tempat penyimpanan sementara dari data yang dikirimkan oleh aplikasi desktop dan aplikasi mobile.

Kata Kunci: komputasi bergerak, remote desktop, Android

ABSTRACT

Smart Phone mobile device, that can do so many things, like Android is a device that is commonly used by people nowadays, but there are several things that PC can do and a smart phone cannot do. There is also a time where computer user needs access to

a Desktop PC, but the user is not there at the computer's place. Internet and remote desktop technology can help solve the problem, but remote desktop commonly need another PC/Laptop to access the target computer remotely. Another problem comes from existing remote desktop application is a limitation in target computer access, the target computer needs to be in the same network or needs a public IP in that computer.

With this remote desktop application, user can access their computer remotely from anywhere using Android cell phone. The main features of this application are connect and login for opening connection, interact live for remote mode with full control, view live for remote mode with no control, view snapshot for viewing 1 screenshot only, shortcut command for executing preset shortcut commands, and control processes for viewing task manager.

This remote desktop application has three main components, mobile application, web application, and desktop application. Mobile application –a component that shows image and receive user's inputs– will be installed in Android cell phone. Desktop application –a component that captures screenshot and send it to show in mobile application, and receives command inputs to be executed– will be installed in target computer. Web application is a mediation component that functions as a temporary storage of the sent data from desktop application and mobile applications.

Key Words: mobile computing, remote desktop, Android

1. PENDAHULUAN

Android merupakan OS bagi smart phone yang sudah memiliki berbagai fitur-fitur umum yang dimiliki smart phone lainnya. Seperti yang dikatakan sebelumnya, fitur umum smart phone sudah dapat melakukan hampir semua hal yang dapat dilakukan oleh Desktop PC biasa. Tetapi tetap saja, ada hal yang dapat dilakukan pada Desktop PC dan tidak dapat dilakukan oleh sebuah smart phone.

Ada saat di mana seorang pengguna komputer membutuhkan akses pada komputernya, yang merupakan Desktop PC, tetapi pengguna tersebut tidak berada pada tempat PC tersebut diletakkan. Internet dan teknologi Remote Desktop dapat membantu menyelesaikan permasalahan tersebut, hanya saja Remote Desktop pada umumnya membutuhkan Desktop PC/Laptop untuk mengakses komputer yang dibutuhkan secara remote. Tidak semua orang bisa membawa laptop setiap saat, tetapi hampir semua orang selalu membawa ponsel (terutama smart phone) di mana pun dan kapan pun.

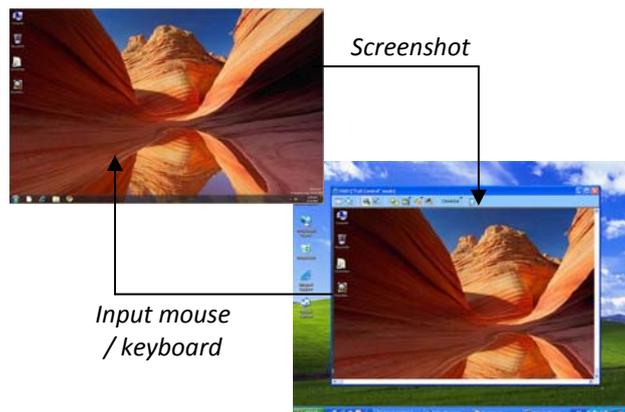
Permasalahan lain yang dihadapi oleh aplikasi Remote Desktop yang sudah ada adalah keterbatasan dalam akses komputer tujuan. Aplikasi remote desktop pada umumnya membutuhkan komputer tujuan untuk berada pada satu jaringan yang sama dan membutuhkan penggunanya mengetahui alamat IP komputer tujuan. Jika komputer tujuan ingin diraih lewat koneksi internet, maka komputer tujuan harus memiliki sebuah IP public. Tidak semua komputer bisa memiliki IP public, apalagi komputer yang berada di dalam suatu jaringan komputer lain.

Dari permasalahan ini, maka dibuatlah sebuah aplikasi Remote Desktop untuk Android sebagai OS sebuah Smart Phone. Aplikasi ini memberikan kesempatan pada penggunanya untuk mengakses komputer pribadinya (Desktop PC/Laptop) di mana pun menggunakan jaringan Internet selama komputer tujuannya juga terhubung dengan internet. Pada umumnya smart phone dapat mengakses internet lewat 3G atau GPRS, dan lewat internet, ponsel tersebut dapat menghubungi komputer tujuan. Aplikasi ini memberikan penggunanya, kemampuan untuk mengakses komputer tujuannya,

sehingga pengguna dapat melakukan segala hal yang dapat dilakukan komputernya, tanpa mengenal batas yang dimiliki oleh ponselnya. Aplikasi ini juga menghapus keterbatasan IP publik yang dihadapi oleh pengguna aplikasi Remote Desktop lainnya.

2. TINJAUAN PUSTAKA

Remote desktop adalah suatu aplikasi yang dapat mengontrol sebuah komputer yang berada di tempat lain dari komputer lainnya. Aplikasi remote desktop ini biasanya meliputi tampilan grafis dari komputer tujuan dan input keyboard atau mouse untuk komputer tujuan.



Gambar 1. Ilustrasi Aplikasi Remote Desktop

Aplikasi ini biasa digunakan oleh orang yang sering berada di luar jangkauan komputer tetapi masih perlu banyak mengakses komputernya, seperti seorang administrator server / database. Remote desktop diawali dengan membuat sebuah koneksi remote antar komputer, kemudian setelah koneksi tercipta, kedua computer akan mempertahankan koneksi dengan mengirimkan screenshot dari komputer tujuan ke komputer pengontrol dan mengirimkan input mouse atau keyboard dari komputer pengontrol ke komputer tujuan. Koneksi remote desktop diilustrasikan pada gambar 1. Screenshot atas pada gambar 1 adalah komputer tujuan, sedangkan screenshot bawah pada gambar 1 adalah komputer yang mengontrol. Dapat dilihat bahwa screenshot pada komputer tujuan akan tampil pada komputer pengontrol di dalam window aplikasi remote desktop tersebut.

Robot merupakan salah satu class yang tergabung dalam Java SE, tepatnya di dalam package java.awt. Class ini berfungsi untuk membuat dan mensimulasikan sebuah input untuk sistem, yaitu inputan mouse dan keyboard. Selain pembentukan input, class Robot juga menyediakan method untuk mengambil screenshot dari komputer tempat aplikasi tersebut aktif. Class Robot ini merupakan class inti dari pengembangan aplikasi remote desktop ini, di mana untuk proses seperti pengendalian input dan pengambilan screenshot ditangani oleh class ini.

Method yang disediakan oleh class Robot ini sifatnya sangat mendasar dan detail. Method mousePress pada class Robot bukanlah method yang digunakan untuk membuat satu klik, tetapi membuat sebuah simulasi mouse yang ditekan terus menerus sampai ada pelepasan mouse yang dilakukan oleh method mouseRelease. Sama halnya seperti pada penekanan tombol keyboard, penggunaan class Robot harus diatur sedemikian rupa sehingga terjadi simulasi penekanan tombol keyboard yang benar.

Method terpenting yang disediakan class Robot, selain untuk pembentukan input, adalah method `createScreenCapture`. Method ini akan membuat sebuah gambar screenshot dari layar tepat saat method ini dipanggil. Gambar screenshot yang dihasilkan oleh method ini berbentuk instance dari `java.awt.image.BufferedImage`. Instance dari `BufferedImage` ini yang akan diproses lebih lanjut untuk keperluan aplikasi. Pemakaian method ini sendiri cukup mudah, hanya dengan menyebutkan posisi layar yang akan diambil maka method ini akan langsung mengembalikan screenshot layar yang disebutkan.

Aplikasi Android dibuat dengan bahasa pemrograman Java. Kode Java yang sudah dikompilasi, beserta segala data dan file yang diperlukan oleh aplikasi, akan dibungkus menjadi satu *Android Package*, sebuah file arsip yang memiliki *extension* `.apk`. File inilah yang akan didistribusikan dan dipasang pada berbagai peralatan Android, dan satu file `.apk` inilah yang disebut satu aplikasi. Aplikasi Android memiliki beberapa komponen utama. Berikut ini adalah daftar komponen utama yang digunakan:

- **Activity**: komponen utama Android yang berfungsi untuk membuat *user interface* di layar Android. Peran Activity di sini mirip dengan `JFrame` pada Java.
- **Service**: komponen utama Android yang berfungsi untuk melaksanakan proses di balik layar tanpa memberikan tampilan visual. Service biasa digunakan untuk melaksanakan proses yang cukup berat dan dalam waktu yang tidak tentu.
- **Broadcast Receiver**: komponen utama Android yang berfungsi menerima “siaran” dari aplikasi tertentu. Siaran bisa berasal dari Activity, Service, atau bahkan aplikasi lainnya. Komponen Broadcast Receiver akan menerima suatu siaran, jika ada, dan akan langsung melaksanakan suatu proses yang telah ditentukan oleh pengembang setelahnya. Komponen ini biasa dipakai jika diperlukan adanya komunikasi antara Service dengan Activity.

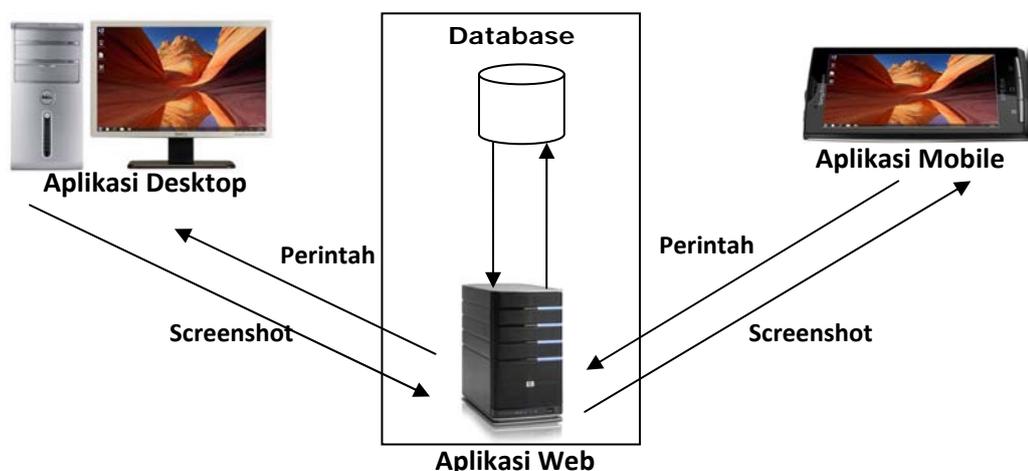
Dalam Android, cara menjalankan Activity, Service, atau Broadcast Receiver tidak sama seperti memulai Thread atau membuka `JFrame` pada Java. Ketiga komponen tersebut harus dijalankan dengan Intent.

HTTP Web Request adalah teknik untuk menghubungi suatu server yang melayani web request melalui protokol HTTP (HyperText Transfer Protocol). HTTP web request dimanfaatkan untuk dijadikan metode komunikasi data, antara aplikasi Android dengan server dan aplikasi desktop dengan server. Aplikasi yang melakukan komunikasi menggunakan protokol HTTP tidak memerlukan aplikasi pengirimnya untuk menyebutkan alamat Internet Protocol (IP) asal, dan respon dari aplikasi tujuan bisa sampai kembali ke aplikasi pengirim tanpa membutuhkan IP dan bisa melewati batasan IP public dan private. Karena itu HTTP web request digunakan agar aplikasi dapat berjalan tanpa menyebutkan alamat IP.

3. DESKRIPSI SISTEM

Aplikasi ini merupakan sebuah sistem yang terdiri dari 3 komponen utama: aplikasi mobile pada Android, aplikasi desktop pada komputer yang dituju, dan aplikasi web pada komputer server sebagai perantara. Tentu saja ketiga komponen ini harus terhubung ke internet dengan dukungan koneksi yang cukup baik, agar aplikasi bisa berjalan dengan lancar. Aplikasi mobile ini akan menampilkan gambar *screenshot* layar yang menunjukkan kondisi layar komputer yang sedang dituju. Aplikasi desktop berperan menyediakan screenshot gambar layar untuk dikirimkan ke aplikasi mobile yang membutuhkan melalui server, serta mengerjakan perintah masukan user yang

berasal dari aplikasi mobile. Aplikasi web pada server berfungsi sebagai perantara komunikasi antara aplikasi mobile dengan aplikasi desktop.



Gambar 2. Arsitektur Umum Aplikasi Remote Desktop

Pada dasarnya, sistem ini memiliki 2 data penting yang harus dikirimkan, yaitu data screenshot layar yang dihasilkan oleh Robot di aplikasi desktop dan dikirim ke aplikasi mobile untuk ditampilkan, dan data perintah yang dihasilkan oleh handler di aplikasi mobile dan dikirim ke aplikasi desktop untuk dilaksanakan.

Aplikasi desktop berfungsi untuk mengambil data screenshot dan mengirimkannya, melakukan request ke server untuk meminta data perintah dan akan menjalankan perintah tersebut. Aplikasi mobile berfungsi untuk melakukan request data layar ke server dan menampilkan data layar, menangkap inputan user yang berupa sentuhan layar dan ketikan keyboard dan mengirimkannya ke server. Aplikasi web berfungsi menyimpan data dari satu bagian agar dapat diambil oleh bagian yang lain, dan mengambil data satu bagian yang telah disimpan oleh bagian yang lain.

Aplikasi desktop tersusun dari 3 thread utama, thread screenshot, thread pengambilan perintah, dan thread pelaksana perintah. Ketiga thread ini dibentuk agar aplikasi desktop dapat melakukan 3 proses yang berbeda secara *multithreading*, sehingga 3 proses itu dapat berjalan sendiri tanpa menunggu satu sama lain. Thread screenshot berfungsi untuk mengambil data screenshot dan mengirimkannya. Thread pengambilan perintah berfungsi untuk melakukan request ke server untuk meminta data perintah, menterjemahkan perintah yang masih berbentuk XML menjadi objek Java, dan menambahkan perintah tersebut ke dalam queue perintah milik Thread pelaksana perintah. Thread pelaksana perintah akan mengambil perintah yang berada di queue satu demi satu dan akan menjalankan perintah tersebut.

Aplikasi mobile memiliki 2 komponen utama yaitu Service dan Activity. Service berfungsi untuk melakukan proses request data layar ke server, menterjemahkan data layar, dan menyiarkan datanya agar diterima oleh broadcast receiver yang dipasang di Activity. Service memiliki thread tersendiri, sehingga proses panjang yang dilakukan Service tidak akan menghambat jalannya Activity atau aplikasi lain. Activity berfungsi untuk memberikan tampilan kepada pengguna, di mana Activity ini akan menampilkan data layar yang disiarkan oleh Service. Siaran Service dapat ditangkap oleh Activity dengan bantuan Broadcast Receiver yang telah didaftarkan sebelumnya. Selain itu, Activity ini juga berperan menangkap inputan user yang berupa sentuhan layar dan

ketikan keyboard, lalu membuat request berdasarkan inputan tersebut, dan mengirimkannya ke server.

Aplikasi web memiliki satu Servlet, tetapi satu Servlet ini terbagi menjadi 2 bagian secara arsitektur, yaitu bagian penanganan request dari aplikasi desktop dan bagian penanganan request dari aplikasi mobile. Kedua bagian ini sebenarnya memiliki 2 fungsi yang sama, yaitu menyimpan data dari satu bagian agar dapat diambil oleh bagian yang lain, dan mengambil data satu bagian yang telah disimpan oleh bagian yang lain.

Data screenshot layar berasal dari komponen Robot yang dimiliki oleh aplikasi desktop. Screenshot layar ini akan diproses untuk dicari bagian mana saja yang berubah sehingga gambar yang dikirim adalah bagian layar yang berubah saja. Data layar yang telah terproses akan disusun menjadi sebuah POST request yang memuat data byte dari data layar tersebut, kemudian POST request tersebut dapat dikirimkan ke aplikasi web melalui metode web request POST. Sesampainya di server, data POST akan di-*parse* dan diambil data bytenya, kemudian diproses menjadi BLOB untuk disimpan ke dalam database. Peran aplikasi desktop dalam aliran data layar telah berakhir sampai di sini. Data layar tersebut harus dipertahankan dalam database agar aplikasi server dapat memberikan gambar itu jika sewaktu-waktu diminta oleh aplikasi mobile.

Aplikasi mobile akan melakukan web request untuk meminta data layar melalui komponen Service. Server yang menerima request tersebut akan menyusun sebuah web response yang berisi data layar dalam bentuk byte. Service yang menerima respon tersebut harus membaca respon dan mengambil data byte yang tercantum di dalamnya. Kemudian data byte tersebut dapat dikirimkan ke Activity melalui siaran yang dilakukan oleh Service dan penerimaan yang dilakukan oleh Broadcast Receiver. Siaran yang diterima oleh Broadcast Receiver akan dikonversi menjadi Bitmap Android (`android.graphics.Bitmap`) dan akan ditampilkan di layar.

Data perintah berasal dari penanganan input user yang berada di Activity Android. Sentuhan layar dan/atau ketikan keyboard yang diterima oleh Activity akan diproses dan akan dibuatkan web request berdasarkan inputan tersebut. Lalu web request yang berisi data perintah akan dikirimkan ke server, sehingga server dapat menyimpannya ke dalam database agar aplikasi server dapat memberikan data perintah tersebut jika sewaktu-waktu diminta oleh aplikasi desktop. Bentuk data perintah ini lebih sederhana dari data layar karena data perintah ini hanya terdiri dari data teks dan/atau angka. Web request yang disusun oleh Activity aplikasi mobile adalah web request metode GET, di mana data perintah disertakan pada query string di URL metode GET.

Aplikasi desktop, tepatnya pada thread pengambilan perintah, akan melakukan request pada server untuk meminta data perintah. Server yang menerima request tersebut akan menyusun sebuah respon yang berbentuk XML dan memuat data perintah yang ada bagi aplikasi desktop tersebut. XML yang telah dikirimkan akan diterima oleh thread pengambilan perintah pada aplikasi desktop, dan XML tersebut akan diparse dan diubah menjadi bentuk objek Java. Selanjutnya, objek Java tersebut akan dipindahkan ke thread pelaksana perintah dan dimasukkan ke dalam queue perintah milik thread tersebut. Perintah akan berada dalam antrian queue perintah, di mana thread pelaksana perintah akan mengambil perintah yang berada di dalam queue perintah dan menjalankannya satu per satu.

Sistem aplikasi remote desktop ini memiliki beberapa syarat dan beberapa batasan penting agar sistem berjalan dengan baik. Adapun syarat utama yang menentukan jalannya sistem ini adalah:

- Koneksi internet pada komputer untuk mendukung koneksi aplikasi desktop.
- Koneksi internet pada handphone Android untuk mendukung koneksi aplikasi mobile. Penggunaan koneksi 3G sangat dianjurkan.
- Server dengan fasilitas Apache Tomcat 6.0 dan MySQL yang memiliki IP publik dan uptime 99%.
- Handphone dengan OS Android minimal versi 1.6.
- Komputer PC desktop dengan Java Virtual Machine, dan mendukung jalannya class Robot dan Tray Icon.
- Komputer PC desktop harus berada dalam keadaan nyala dan aplikasi desktop harus tetap dijalankan dalam kondisi telah terhubung.

Semua syarat ini harus terpenuhi untuk satu koneksi, kecuali syarat server, agar satu koneksi remote dapat berjalan dengan baik. Jika salah satu syarat ini tidak terpenuhi, maka koneksi remote tidak dapat berjalan.

Sistem aplikasi remote desktop yang sudah terpenuhi syaratnya dapat berjalan dengan baik. Tetapi dalam pemakaian aplikasi ini, masih ada beberapa batasan yang perlu diperhatikan. Adapun batasan utama dari sistem ini adalah:

- Hasil pengiriman data layar yang diberikan tidaklah real time dan memiliki banyak delay. Hal ini disebabkan karena metode yang digunakan untuk mengirim data adalah metode pull.
- Hasil pengiriman data perintah juga tidak real time dan memiliki banyak delay. Hal ini disebabkan oleh hal yang sama dengan delay pengiriman layar.
- Aplikasi remote ini hanya dapat dipakai untuk keperluan mendasar yang tidak melibatkan banyak input mouse dan keyboard.
- Aplikasi remote tidak dapat melakukan input keyboard yang rumit, seperti penekanan tombol bersama, atau tombol-tombol yang tidak mendasar.
- Aplikasi remote tidak dapat melakukan operasi mouse yang rumit, seperti penekanan lebih dari satu tombol, *drag and drop*, dan tidak ada model *hover*.
- Aplikasi remote dapat dipakai secara multiuser, tetapi tidak memiliki fungsi registrasi. Pengguna dapat menggunakan Computer ID apapun selama tidak kembar dengan Computer ID milik pengguna lainnya.

Poin kelima pada batasan sistem disebabkan karena mode input dari aplikasi mobile adalah layar sentuh. Layar sentuh akan menerima input drag and drop sebagai mode *pan and scan* (menggeser layar untuk melihat gambar yang ukurannya lebih besar dari layar). Selain itu, layar sentuh tidak memiliki bentuk input seperti pergerakan mouse, di mana layar sentuh hanya memiliki input sentuhan pada titik tertentu, sehingga tidak ada pergerakan mouse yang menyebabkan adanya model *hover*.

4. UJI COBA

Uji coba performa ini dilakukan terutama untuk mengetahui kecepatan sebuah koneksi aplikasi. Uji coba performa ini bertujuan untuk mengukur performa dari pengiriman screenshot. Performa pengiriman screenshot diukur dari waktu yang dibutuhkan untuk melakukan pengiriman screenshot dari aplikasi desktop ke aplikasi web dan dari aplikasi web ke aplikasi mobile. Perhitungan waktu akan dilakukan dengan menyelipkan sebuah kode program yang mencatat waktu awal proses dan akhir

proses, kemudian menghasilkan selisih dari kedua catatan waktu tersebut pada log program. Berikut ini akan dijelaskan langkah-langkah untuk melakukan pengukuran waktu perbandingan screenshot dan pengiriman screenshot dari aplikasi desktop ke aplikasi web:

1. Program diatur untuk melakukan pencatatan waktu pada beberapa titik, yaitu titik awal pengambilan screenshot, titik akhir perbandingan screenshot tepat sebelum dikirim, titik awal pembuatan web request, dan titik akhir setelah web request selesai dilaksanakan.
2. Program diatur untuk menghitung selisih waktu untuk keempat titik tersebut. Selisih yang dihitung adalah selisih antara awal pengambilan screenshot dan akhir perbandingan screenshot, serta selisih antara awal pembuatan web request, dan akhir pelaksanaan web request. Setelah itu, kedua selisih ini akan ditampilkan pada console / log.
3. Server yang menerima akan disiapkan terlebih dahulu, sehingga lingkungan pengiriman ke internet akan tercipta.
4. Program akan dijalankan pada layar yang berubah saja, di mana ukuran layar yang berubah akan diatur sedemikian rupa untuk beberapa percobaan.
5. Komputer diatur untuk tidak melakukan hal apa-apa selain menjalankan video. Penggunaan video ditujukan untuk membuat area layar yang selalu berubah secara otomatis dan untuk memudahkan manipulasi ukuran area layar yang selalu berubah.
6. Percobaan akan dilakukan pada 5 ukuran yang berbeda, yaitu tidak ada perubahan sama sekali, perubahan ukuran kecil, perubahan ukuran sedang, perubahan ukuran besar, dan perubahan ukuran fullscreen.
7. Video akan dijalankan dan diatur sesuai ukuran yang dibutuhkan. Saat video berjalan, aplikasi akan dijalankan dan pencatatan waktu akan dilakukan dengan melihat console / log hasil.
8. Setiap ukuran akan dilakukan masing-masing 10 kali pencatatan waktu. Hasil yang diperoleh adalah 10 catatan waktu perbandingan dan waktu pengiriman untuk setiap ukuran. Total hasil yang diperoleh adalah 50 catatan waktu.

Pada Tabel 1, dapat dilihat bahwa ukuran layar yang berubah sangat mempengaruhi waktu pengiriman dan perbandingan screenshot. Ukuran layar yang berubah berbanding lurus dengan waktu pengiriman, tetapi berbanding terbalik dengan waktu perbandingan layar, kecuali saat tidak ada perubahan.

Tabel 1. Rangkuman Hasil Uji Coba Performa Pengiriman Screenshot dari Desktop

No.	Ukuran Layar yang Berubah	Waktu Perbandingan	Waktu Pengiriman
1.	Tidak ada (0px)	119 ms	0 ms
2.	Kecil (273x260px)	139,1 ms	1471,7 ms
3.	Sedang (792x516px)	106,2 ms	2408 ms
4.	Besar (1196x906px)	37,6 ms	3962 ms
5.	Fullscreen (1280x1024px)	0 ms	4529 ms

Uji coba penerimaan screenshot pada aplikasi mobile dari aplikasi web, hanya terdapat 1 jenis pengukuran, yaitu pengukuran waktu pelaksanaan request screenshot dari aplikasi mobile ke aplikasi web serta respon baliknya. Pengukuran waktu ini

memiliki langkah-langkah yang hampir sama dengan langkah pengukuran waktu pengiriman screenshot pada aplikasi desktop.

Tabel 2. Rangkuman Hasil Uji Coba Performa Penerimaan Screenshot pada Mobile

No.	Ukuran Layar yang Berubah	Waktu Pengiriman
1.	Tidak ada (0px)	0 ms
2.	Kecil (273x260px)	16096,7 ms
3.	Sedang (792x516px)	26337,5 ms
4.	Besar (1196x906px)	43340 ms
5.	Fullscreen (1280x1024px)	49453,6 ms

Dapat dilihat bahwa hasil waktu pengiriman masih berbanding lurus dengan ukuran layar yang berubah. Tetapi koneksi mobile di Indonesia masih sangat rendah, sehingga waktu yang dihasilkan sangat lama, dan terdapat suatu delay yang sangat panjang. Sangat direkomendasikan jika aplikasi mobile dijalankan pada area WiFi yang sudah didukung koneksi internet yang sangat cepat dan stabil.

5. KESIMPULAN

Dalam pembuatan aplikasi desktop ini ditemukan adanya hal-hal baru dan hambatan-hambatan tertentu. Dari pembuatan aplikasi desktop ini, dapat ditarik beberapa kesimpulan, antara lain:

- Aplikasi remote desktop pada Android dapat dibuat dengan bantuan server sebagai perantara. Server ini menyediakan sebuah *handler* HTTP web request yang sifatnya mirip seperti REST web request.
- Metode komunikasi yang digunakan adalah metode pulling, dari aplikasi mobile ke aplikasi web untuk mengambil screenshot, dan dari aplikasi desktop ke aplikasi web untuk mengambil perintah.
- Dengan adanya server sebagai perantara dan penggunaan metode pulling, komputer tujuan tidak membutuhkan IP publik untuk dapat diakses secara remote. Hal ini disebabkan karena aplikasi desktop dan aplikasi mobile tidak terhubung secara langsung melainkan terhubung secara tidak langsung lewat aplikasi web dan web request.

Penggunaan aplikasi web dan metode pulling menyebabkan adanya delay tambahan, sehingga aplikasi ini tidak dapat menghasilkan suatu bentuk komunikasi yang real-time. Koneksi yang tidak real-time ini menyebabkan munculnya berbagai batasan aktivitas yang dapat dilakukan oleh aplikasi ini.

6. DAFTAR PUSTAKA

- Becker, B.W. Java: Learning to Program with Robots. Course Technology. 2009.
 Burnette, Ed. Introduction to Android Development. For Dummies. 2009.
 DiMarzio, J.F. Android A Programmer's Guide. McGraw-Hill. 2008.
 Haseman, Chris. Android Essentials. Apress. 2008.
 Mikkonen, Tommi. Programming Mobile Devices An Introduction for Practitioners. Wiley. 2007.
 Rogers, Rick. Android Application Development. O'Reilly. 2009.