

Penerapan Screen Mirroring Android Pada Projector Menggunakan Raspberry Pi

Dwijaya Kongky Salim, Justinus Andjawirawan, Lily Puspa Dewi
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
dksafu@yahoo.co.id, justin@petra.ac.id, lily@petra.ac.id

ABSTRAK

Dulu, cara untuk menggunakan *projector* hanya bisa dilakukan dengan menggunakan kabel VGA yang kemudian dihubungkan dengan laptop atau desktop pc yang memiliki VGA *connector*. ketika *projector* dan laptop atau desktop pc sudah terhubung, proyektor akan menduplikasi layar desktop pc atau laptop yang kemudian akan ditampilkan di layar proyektor. Pada skripsi ini dilakukan pembuatan aplikasi untuk memanfaatkan *framework* Gstreamer untuk membuat aplikasi yang dapat melakukan *screen mirroring* dan menggunakan Raspberry Pi untuk menampilkan hasil *mirroring* di *projector*, sehingga *user* dapat melakukan presentasi dengan menggunakan *smartphone* dan tidak memerlukan kabel. Hasil pengujian aplikasi menunjukkan bahwa terjadi delay dalam *mirroring* menyebabkan layar *smartphone* yang di *mirroring* dan ditampilkan di Raspberry Pi mengalami update frame yang sangat lambat. Hal ini disebabkan oleh hardware Raspberry Pi yang kurang mendukung Raspberry Pi untuk melakukan *mirroring* dengan lancar.

Kata Kunci: *Projector*, *screen mirroring*, Gstreamer, Raspberry Pi

ABSTRACT.

In the past, how to use a projector can only be done using a VGA cable which is then connected to a laptop or desktop pc that has a VGA connector, when the projector and laptop or desktop pc are connected, the projector will duplicate the desktop screen of the PC or laptop which will then be displayed on the screen projector In this thesis an application is made to utilize the Gstreamer framework to create applications that can perform screen mirroring and use the Raspberry Pi to display the results of mirroring on the projector, so that users can make presentations using a smartphone and do not need a cable. The application testing results show that delay in mirroring causes the smartphone screen mirrored and displayed on the Raspberry Pi to experience a very slow frame update, due to the Raspberry Pi hardware that doesn't support the Raspberry Pi to mirror smoothly.

Keywords: *Projector*, *screen mirroring*, Gstreamer, Raspberry Pi

1. PENDAHULUAN

Dulu, cara untuk menggunakan *projector* hanya bisa dilakukan dengan menggunakan kabel VGA yang kemudian dihubungkan dengan laptop atau desktop pc yang memiliki VGA *connector*, ketika *projector* dan laptop atau desktop pc sudah terhubung, proyektor akan menduplikasi layar desktop pc atau laptop yang kemudian akan ditampilkan di layar proyektor. Sekarang, *smartphone* mampu menggunakan *projector* untuk melakukan presentasi dengan menggunakan aplikasi dan tidak perlu menggunakan kabel, melainkan melalui *wireless support* yang

ditancapkan pada proyektor atau *projector* dengan *built-in wireless support*, tetapi kebanyakan aplikasi untuk melakukan presentasi di google play store tidak mampu untuk menduplikasi layar *smartphone* ke layar *projector*, hanya dapat melakukan hal yang terbatas, yaitu menampilkan gambar, slide, dan video. Ada pun aplikasi yang dapat melakukan duplikasi layar *smartphone* ke layar *projector* hanya bisa digunakan ke *projector* dari merek tertentu dan harganya mahal.

Untuk itu, diperlukan aplikasi yang dapat melakukan aplikasi yang dapat melakukan lebih dari aplikasi-aplikasi yang ada di Google *playstore*, yaitu menduplikasi layar *smartphone* ke layar *projector*, dapat menampilkan hal yang lain tidak hanya slide, gambar, atau video tetapi juga dapat menampilkan hal lain seperti aplikasi-aplikasi media sosial, atau game atau, dapat dipakai di *projector* mana pun yang tidak terpasang *wireless adapter*, dan harganya tidak terlalu mahal.

Terdapat penelitian sebelumnya dengan judul “*A Low-Cost Wireless Multi-Presentation on Single Screen in Classroom Using Raspberry Pi*” [2]. Isi paper tersebut menerapkan program pada Raspberry pi dan aplikasi android pada *smartphone* untuk melakukan presentasi secara *wireless*, dengan membuka file presentasi pada *smartphone* android kemudian menggunakan jaringan Wi-fi, *projector* akan menampilkan slide melalui Raspberry pi.

2. TINJAUAN PUSTAKA

2.1 Gstreamer

GStreamer adalah *framework* untuk membuat aplikasi *streaming* media. Desain mendasar berasal dari video *pipeline* di Oregon Graduate Institute, serta beberapa ide dari DirectShow.

Framework GStreamer memungkinkan untuk menulis semua jenis aplikasi multimedia *streaming*. *Framework* GStreamer dirancang untuk memudahkan menulis aplikasi yang menangani audio atau video atau keduanya. Tidak terbatas pada audio dan video, dan dapat memproses segala jenis aliran data. Desain *pipeline* dibuat untuk memiliki sedikit *overhead* di atas apa yang diterapkan oleh filter yang diaplikasikan. Hal ini membuat GStreamer menjadi *Framework* yang baik untuk merancang aplikasi audio kelas atas yang membuat permintaan latensi tinggi.

Salah satu penggunaan yang paling jelas dari GStreamer adalah menggunakannya untuk membangun pemutar media. GStreamer sudah menyertakan komponen untuk membangun pemutar media yang dapat mendukung berbagai format yang sangat beragam, termasuk MP3, Ogg / Vorbis, MPEG-1/2, AVI, *Quicktime*, *mod*, dan lainnya. GStreamer, bagaimanapun, jauh lebih dari sekadar pemutar media lain. Keuntungan utamanya adalah komponen yang dapat dipasangkan dapat dicampur dan dicocokkan menjadi arbitrary pipeline sehingga memungkinkan untuk menulis aplikasi penyuntingan video atau audio yang lengkap.

Framework ini didasarkan pada plugin yang akan menyediakan berbagai *codec* dan fungsi lainnya. Plugin dapat di *link* dan diatur dalam *pipeline*. *Pipeline* ini mendefinisikan aliran data. *Pipeline* juga dapat diedit dengan editor GUI dan disimpan sebagai XML sehingga *pipeline libraries* dapat dibuat dengan usaha minimum.

Fungsi inti GStreamer adalah menyediakan *Framework* untuk plugin, aliran data, dan penanganan / negosiasi jenis media. Ini juga menyediakan API untuk menulis aplikasi menggunakan berbagai plugin [1].

2.2 H.264

H.264 adalah metode dan format untuk kompresi video, proses mengkonversi video digital ke dalam format yang membutuhkan lebih sedikit kapasitas saat disimpan atau dikirim. Kompresi video atau pengkodean video adalah teknologi penting untuk aplikasi seperti televisi digital, DVD-Video, TV seluler, konferensi video dan streaming video internet. Standarisasi kompresi video memungkinkan produk dari produsen yang berbeda seperti *encoders*, *decoder* dan media penyimpanan untuk saling beroperasi. *Encoder* mengubah video menjadi format terkompresi dan *decoder* mengkonversi video terkompresi kembali menjadi format terkompresi.

Standar H.264 / AVC pertama kali diterbitkan pada tahun 2003, dengan beberapa revisi dan pembaruan diterbitkan sejak itu. Itu dibangun di atas konsep standar sebelumnya seperti MPEG-2 dan MPEG-4 Visual dan menawarkan potensi untuk efisiensi kompresi yang lebih baik, yaitu kualitas video terkompresi yang lebih baik, dan fleksibilitas yang lebih besar dalam mengompresi, mentransmisikan, dan menyimpan video.

H.264 / AVC menjelaskan seperangkat alat atau metode untuk kompresi video. Standar menentukan bagaimana video yang dikodekan dengan alat-alat ini harus diwakili dan diterjemahkan. *Encoder* video mungkin pilih alat mana yang digunakan dan cara menerapkannya pada urutan video saat ini, dengan beberapa kendala. *Decoder H.264-compliant* harus mampu menggunakan sub-set alat yang didefinisikan, dikenal sebagai *profil*.

Salah satu pendorong terpenting untuk standardisasi H.264 dan adopsi selanjutnya oleh industri adalah peningkatan kinerjanya dibandingkan dengan standar sebelumnya. *Benchmark* untuk aplikasi pasar massal seperti TV digital dan penyimpanan video konsumen pada DVD-Video adalah sebelumnya standar MPEG-2. H.264 menawarkan kinerja kompresi yang jauh lebih baik daripada MPEG-2 Visual. Dengan menggunakan H.264 dimungkinkan untuk mengompres video ke jumlah bit yang jauh lebih kecil daripada menggunakan MPEG-2, untuk resolusi video dan kualitas gambar yang sama. Ini berarti, semisalnya materi video yang lebih banyak dapat disimpan pada disk atau dikirim melalui siaran saluran dengan menggunakan format H.264[3].

2.3 Raspberry Pi

Raspberry Pi adalah komputer berukuran kecil yang memiliki ukuran seperti kartu atm yang dapat Anda hubungkan ke tv atau layar komputer dan keyboard. Perangkat ini adalah komputer kecil yang mumpuni, dapat digunakan untuk proyek elektronik dan dapat melakukan banyak hal layaknya PC *desktop* atau komputer Anda. Seperti menjalankan program perkantoran untuk membuat laporan, membuat dokumen, *browsing* internet bahkan memainkan permainan. Selain itu alat ini juga dapat memutar video beresolusi tinggi. Tujuan awal diproduksinya Raspberry pi adalah untuk digunakan oleh orang dewasa dan anak-anak di seluruh dunia untuk belajar pemrograman digital.

Selain untuk kegiatan edukasi, Raspberry pi dapat digunakan sebagai komputer desktop mini, *file server*, *download server*, *access point*, server dns, *multimedia player*, *home automation* dan lain sebagainya[4].

Nama Raspberry Pi, adalah kombinasi dari keinginan untuk menciptakan buah berbasis alternatif komputer (seperti Apple, BlackBerry, dan Aprikot) dan anggukan konsep awal yang sederhana komputer yang dapat diprogram menggunakan Python (disingkat menjadi Pi)[6].

2.4 Screen Mirroring

Screen Mirroring adalah teknologi *wireless* yang memungkinkan Anda mengganti media - atau mentransmisikannya - yang diputar di perangkat Android, Windows, atau Apple yang lebih kecil ke perangkat yang lebih besar, bukan untuk pengalaman menonton yang lebih baik [5].

3. ANALISIS DAN DESAIN SISTEM

3.1 Analisis Permasalahan

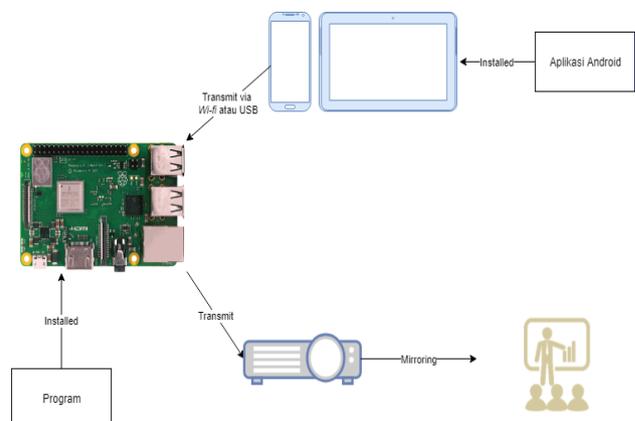
Presentasi dulu menggunakan *desktop* PC atau laptop dan memakai kabel VGA. Sekarang, *smartphone* juga dapat melakukan presentasi dan tidak memerlukan kabel lagi melainkan melalui *wi-fi*. Tetapi, hanya projector tertentu yang dapat melakukan presentasi melalui *wi-fi*, harganya mahal, kebanyakan belum mampu melakukan *screen mirroring* dan hanya bisa dijalankan menggunakan aplikasi yang dibuat oleh *vendor projector* tersebut.

Dalam aplikasi berbasis android ini, *user* dapat melakukan presentasi ke *projector* biasa secara *wireless*, menggunakan Raspberry Pi sebagai *wireless receiver*, dengan menghubungkan Raspberry Pi ke sebuah *projector* biasa, kemudian *user* menjalankan aplikasi android, agar *smartphone* dapat terhubung dan menampilkan layer *smartphone* ke Raspberry Pi, sehingga *user* dapat melakukan presentasi secara *wireless* tanpa menggunakan laptop dan kabel VGA.

3.2 Arsitektur Sistem

Dalam arsitektur ini memiliki komponen berupa *user* yang memakai *smartphone*, Raspberry Pi, dan *projector*. Pertama, di dalam komponen *user* terdapat aplikasi android yang digunakan untuk terhubung dengan Raspberry Pi melalui *wi-fi* atau USB. Kedua, Raspberry Pi akan melakukan *mirroring* pada layar *smartphone* yang terhubung dengan Raspberry Pi. Ketiga, *projector* akan menampilkan layar *smartphone*.

Pada Gambar 1 menjelaskan mengenai rancangan sistem arsitektur yang akan digunakan pada Raspberry Pi dan aplikasi *screen mirroring*.

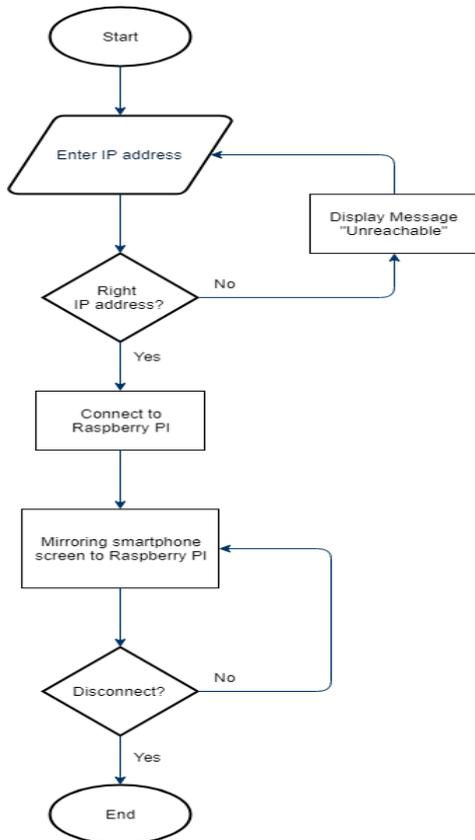


Gambar 1. Desain Arsitektur Sistem

3.3 Garis Besar Sistem

Pada aplikasi berbasis *android* dimulai dengan memasukkan IP address, dimana setelah itu menekan tombol connect untuk terhubung dengan Raspberry PI, setelah itu Raspberry PI akan melakukan *screen mirroring* dan menampilkan layar *smartphone* ke *projector*.

Pada gambar 2 berikut ini merupakan gambar *flowchart* yang menjelaskan menu awal pada aplikasi *android* yang akan dibuat.

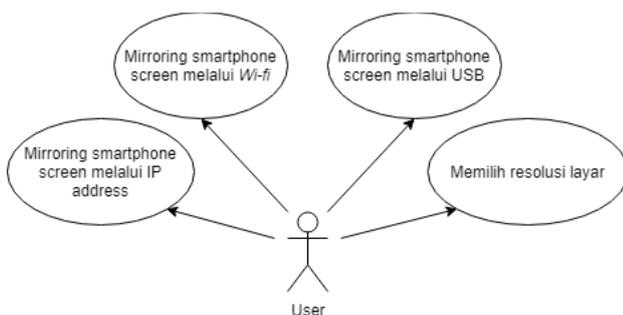


Gambar 2. Flowchart Menu Aplikasi Android

3.4 Use Case Diagram

Use case diagram User terdiri dari *mirroring smartphone screen* melalui IP address, *mirroring smartphone screen* melalui *Wi-fi*, *mirroring smartphone screen* melalui *USB*, dan memilih resolusi layar.

Pada Gambar 3 terdapat *use case diagram* mendeskripsikan aktor yang ada di dalam sistem yaitu *user*.



Gambar 3. Use Case Diagram User Aplikasi Android

4. IMPLEMENTASI SISTEM

4.1 Implementasi Aplikasi

Pembuatan aplikasi dalam skripsi ini membutuhkan beberapa software penunjang, sebagai berikut:

- Android Studio dengan versi 3.2.1 ,digunakan sebagai aplikasi Intergrated Development Enviroment, untuk memudahkan pembuatan aplikasi.
- Thorn merupakan tool yang yang digunakan sebagai editor untuk bahasa pemrograman python.

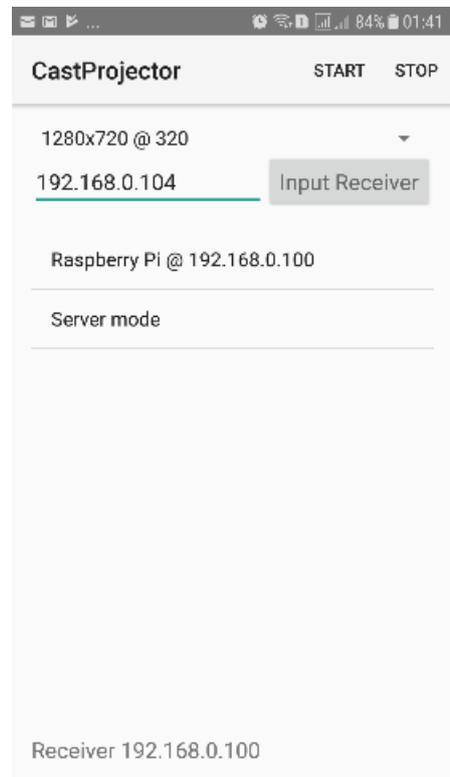
5. PENGUJIAN SISTEM

5.1. Pengujian Aplikasi

5.1.1. Tampilan Aplikasi

Tampilan aplikasi dimulai dengan tombol *Start* dan *Stop* di kanan atas aplikasi, kemudian di bawahnya terdapat resolusi untuk memilih resolusi antara 1280x720, atau 800x480. Di bawah pilihan resolusi terdapat list dari *network-network* yang dapat terhubung dengan Raspberry Pi, di bawah list *network* terdapat label untuk mengetahui *network* mana yang dipilih.

Tampilan aplikasi android terdapat pada Gambar 4.



Gambar 4 Tampilan Aplikasi Android

5.1.2. Pengujian Fitur Mirroring Melalui Wi-fi

User dapat menampilkan layar *smartphone* mereka ke dalam Raspberry Pi dengan memilih jenis resolusi yang ingin dipakai dan menghubungkan Raspberry Pi dengan menulis *IP address* atau memilih dari list yang tampil di aplikasi kemudian menekan tombol start untuk menjalankan *mirroring*.

Tampilan mirroring di Raspberry Pi melalui *Wi-fi* dengan mode portrait terdapat pada Gambar 5, dan tampilan mirroring di Raspberry Pi melalui *Wi-fi* dengan mode landscape terdapat pada Gambar 6.



Gambar 5. Mode Potrait



Gambar 6. Mode Landscape

5.1.3. Pengujian Fitur Mirroring Melalui Micro USB

User juga dapat menampilkan layar *smartphone* di projector dengan menggunakan *micro USB*, kemudian dengan memilih *USB mode* dan menekan tombol start untuk memulai *mirroring*.

Tampilan mirroring di Raspberry Pi melalui *Micro USB* dengan mode portrait terdapat pada Gambar 7, dan Tampilan mirroring di Raspberry Pi melalui *Micro USB* dengan mode landscape pada Gambar 8.



Gambar 7. Mode Potrait



Gambar 8. Mode Landscape

5.2. Hasil Pengujian Aplikasi

Hasil pengujian aplikasi menunjukkan bahwa terjadi delay dalam mirroring menyebabkan layar *smartphone* yang di mirroring dan ditampilkan di Raspberry Pi mengalami update *frame* yang sangat lambat. Contohnya ketika *user* mengeswipe *smartphone* mereka, atau membuka sebuah aplikasi, *update frame* di Raspberry Pi sangat lambat menyebabkan layar *smartphone* di Raspberry Pi baru melakukan swipe atau membuka aplikasi setelah 1 menit atau lebih dari itu.

5.2.1. Menghitung Delay

Percobaan ini dilakukan dengan melakukan perpindahan *frame* sebanyak 5 kali kemudian akan dihitung waktu yang dibutuhkan untuk mencapai dari 1 *frame* ke *frame* lainnya dengan *stopwatch* dan akan dihitung juga total waktu yang dibutuhkan untuk berpindah dari *frame* 1 ke *frame* 5. Hasil percobaan dapat dilihat pada Tabel 1, Tabel 2, Tabel 3, dan Tabel 4.

Tabel 1. 800x480 Samsung Galaxy J7 Prime Wi-fi

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Total
Percobaan 1	16.20	13.10	19.80	23.00	20.70	01:32.80
Percobaan 2	16.60	08.30	17.00	16.40	15.50	01:14.80
Percobaan 3	14.95	19.25	11.33	21.53	19.40	01:26.46

Tabel 2. 1280x720 Samsung Galaxy J7 Prime Wi-fi

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Total
Percobaan 1	29.70	23.20	32.60	33.90	31.80	02:31.20
Percobaan 2	32.40	18.30	30.80	36.70	36.50	02:36.20
Percobaan 3	37.63	27.01	17.49	32.40	30.75	02:25.28

Tabel 3. 800x480 Samsung Galaxy J7 Prime Micro USB

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Total
Percobaan 1	16.00	09.00	15.10	16.50	17.30	01:13.90
Percobaan 2	15.70	08.80	13.30	14.90	15.10	01:07.80
Percobaan 3	16.50	08.40	16.60	17.90	19.40	01:18.80

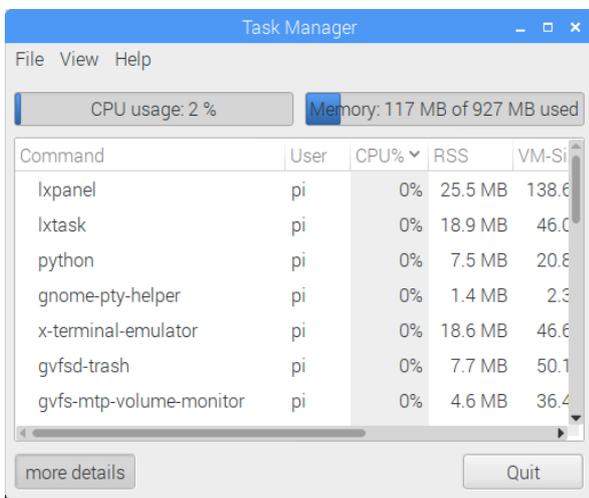
Tabel 4. 1280x720 Samsung Galaxy J7 Prime Micro USB

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Total
Percobaan 1	31.20	23.10	33.00	32.50	36.60	02:36.40
Percobaan 2	29.50	25.50	38.10	38.20	39.80	02:51.10
Percobaan 3	32.40	23.30	33.40	32.90	32.30	02:36.30

5.2.2. Penyebab Delay

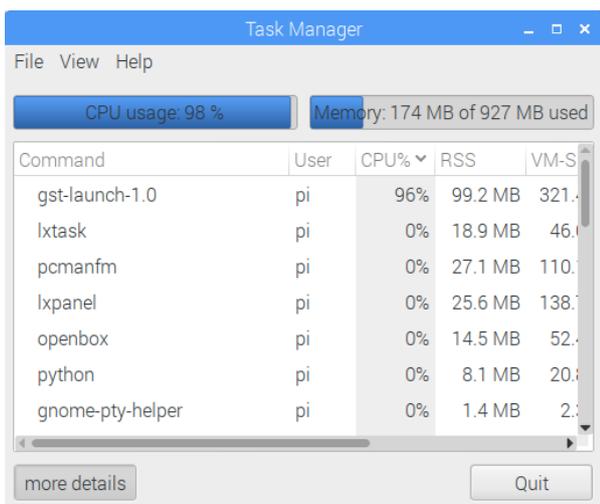
Untuk mengetahui penyebab delay, dilakukan perbandingan dengan membuka task manager Raspberry Pi untuk melihat CPU usage sebelum dan sesudah melakukan mirroring.

Pada Gambar 9. diperlihatkan CPU Usage dari Task Manager Raspberry Pi sebelum smartphone melakukan mirroring ke Raspberry Pi.



Gambar 9. CPU Usage Sebelum Mirroring

Dan pada gambar 10. adalah CPU Usage setelah mirroring



Gambar 10. CPU Usage Sesudah Mirroring

CPU usage menjadi penuh setelah melakukan mirroring, karena CPU usage penuh menyebabkan update frame menjadi lambat karena frame-frame yang dikirim dari smartphone ke raspberry pi tertahan menunggu diproses oleh CPU.

5.2.3. Pengujian di Laptop Asus Windows 10

Pengujian dilakukan di laptop Asus dengan spesifikasi Intel Core i5 6200U CPU 2.3 GHz, Ram 12 GB, dan GPU Nvidia Geforce 930 M. Pengujian dilakukan untuk menentukan apakah software atau hardware yang menyebabkan delay yang sangat lama di Raspberry Pi. Software yang dibuat untuk mirroring dapat diuji coba di Windows 10 karena framework Gstreamer tersedia untuk Windows user.

5.2.4. Hasil Uji Coba di Asus Windows 10

Hasil uji coba di Windows 10 memberitahu bahwa penyebab delay yang sangat lama pada saat Raspberry Pi disebabkan oleh hardware Raspberry Pi yang kurang mendukung Raspberry Pi untuk melakukan mirroring dengan lancar. Dapat disimpulkan bahwa penyebab delay yang lambat di Raspberry Pi karena hardware dikarenakan pada waktu dilakukan uji coba di laptop Asus Windows 10 dengan GPU yang lebih tinggi, software dapat melakukan mirroring dengan lancar dan hanya mengalami delay sekitar 1 hingga 5 detik

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari hasil perencanaan, analisis, dan desain pembuatan aplikasi sampai dengan penulisan buku ini, dapat diambil kesimpulan sebagai berikut:

- Mirroring tidak dapat berjalan dengan lancar di Raspberry Pi karena keterbatasan kemampuan hardware.
- Mirroring dapat berjalan dengan lancar dengan GPU yang lebih tinggi dari yang dimiliki oleh Raspberry Pi
- Aplikasi dapat dijalankan dengan lancar pada device diatas API 21. Untuk device dengan spesifikasi yang kurang mumpuni dibawah API 21 aplikasi tidak dapat dijalankan.

6.2. Saran

Selain kesimpulan, terdapat beberapa saran yang dapat digunakan dan berguna bagi pembaca untuk penelitian lebih lanjut, sebagai berikut:

- Kurangnya penjelasan tentang bagaimana cara menggunakan aplikasi.
- Perlu mencari lebih dalam penyebab delay
- Aplikasi dibuat untuk sistem operasi ios
- Tampilan dibuat lebih bagus agar terlihat menarik.

7. DAFTAR PUSTAKA

- [1]. Budi, Y. Rita, L. & Dewi, C.D. 2017. International Journal of Web-Based Learning and Teaching Technologies, Vol. 12 No. 3 A Low-Cost Wireless Multi-Presentation on Single Screen in Classroom Using Raspberry Pi. 23-33.
- [2]. Cox, Tim. 2014. Raspberry Pi Cookbook for Python Programmers. Birmingham: Packt Publishing Ltd.
- [3]. Gstreamer. 2018. What is Gstreamer. Retrieved from <https://gstreamer.freedesktop.org/documentation/application-development/introduction/gstreamer.html>
- [4]. Lifewire. 2018. What is screen mirroring. Retrieved from <https://www.lifewire.com/what-is-screen-mirroring-4154335>
- [5]. Ralali. 2017. Apa itu Raspberry pi? Dan apa perbedaan versi lama dengan yang baru. Retrieved <https://news.ralali.com/apa-itu-raspberry-pi/>
- [6]. Richardson, I.E. 2010. The H.264 Advanced Video Compression Standard. United Kingdom: John Wiley & Sons Ltd.

