

Implementasi Live Audio Streaming Menggunakan Raspberry Pi

By Hudaya hudaya

Implementasi Live Audio Streaming Menggunakan Raspberry Pi

Hudaya Program Studi D3 Teknik Komputer Fakultas Ilmu Terapan Bandung, Indonesia Mulki.Hudaya@gmail.com	Gita Indah Hapsari Program Studi D3 Teknik Komputer Fakultas Ilmu Terapan Bandung, Indonesia gitaindahhapsari@tass.telkomuniversity .ac.id
Giva Andriana Mutiara Program Studi D3 Teknik Komputer Fakultas Ilmu Terapan Bandung, Indonesia giva.andriana@tass.telkomuniversity .ac.id	

Abstrak— *Live audio streaming* adalah suatu teknologi yang digunakan untuk memainkan dan memberikan akses melihat *file audio* secara langsung (*real-time*) dari sebuah *server* tanpa harus melalui proses *download*. Keterbatasan ruang server menjadi dasar munculnya suatu ide untuk merancang sebuah *server* dengan komponen perangkat keras yang bersifat *portable*. Dalam penelitian ini dirancang suatu sistem server untuk implementasi *live audio streaming* dengan menggunakan Raspberry Pi sebagai *server*. Perancangan dilakukan dengan melakukan instalasi aplikasi yang digunakan untuk menunjang *live audio streaming server*, *mp3 streaming* dan *user feedback* berupa *shoutbox* serta konfigurasi. Aplikasi ini diimplementasikan dalam bentuk aplikasi *web* untuk diakses oleh pengguna. Perubahan tampilan *default* halaman *web* dilakukan untuk menyesuaikan dengan kebutuhan aplikasi dan kenyamanan pengguna dalam mengaksesnya. Dari hasil pengujian diperoleh bahwa sistem pada raspberry pi dapat digunakan sebagai *server* untuk aplikasi *live audio streaming* dengan delay sekitar 2 detik.

Kata Kunci—*raspberry pi; live audio streaming; aplikasi web.*

Abstract— *Live audio streaming* is a technology used to play or provide access to view the audio file directly (*real-time*) from a server without having to go through the download process. Limitations of server space became an idea of designing a server with a smaller hardware components. This study designed a server system for the implementation of live audio streaming using the Raspberry Pi as a server. The design was done by installations of applications that are used to support server streaming live audio, MP3 streaming and user feedback in the form shoutbox and its configurations. This application is implemented in the web application to be accessed by the user. Change the default view web pages was made to suit the needs of the application, user's convenience and ease of access . The test results showed that the

raspberry pi system can be used as a server for live audio streaming applications with a delay of about 2 seconds

Keywords—*raspberry pi; live audio streaming; web application*

I. PENDAHULUAN

Live Audio Streaming merupakan aplikasi yang memberikan fasilitas audio berupa *live talk show, concert*, perekaman suara dan lain sebagainya. Keterbatasan ketersediaan ruang untuk perancangan suatu sistem *server* menjadi kendala bagi ruang *server* saat ini. Keterbatasan ini kemudian memunculkan suatu ide yaitu merancang sebuah server dengan komponen-komponen perangkat keras yang berukuran jauh lebih kecil dibandingkan dengan komponen-komponen yang biasa digunakan dalam membangun sebuah *server*.

Raspberry Pi adalah salah satu komponen dengan ukuran yang sangat kecil dengan kualitas yang hampir sama dan fitur tidak kalah dengan komponen sejenisnya. Raspberry Pi adalah komponen *mother board* mini berukuran sebesar kartu kredit dengan sistem operasi Raspbian yang berbasis Debian GNU atau Linux sehingga tidaklah sulit dalam mengoperasikannya.

Berdasarkan hal tersebut maka dibangunlah sebuah sistem *server* untuk *live audio streaming* dengan menggunakan raspberry Pi sebagai *server*. Tujuan dari penelitian ini adalah membangun sebuah sistem *server* berbasis Raspberry Pi untuk implementasi *live audio streaming* dalam format .mp3 dengan dilengkapi pembuatan aplikasi antar muka pengguna dengan fungsi dasar *broadcast* dan *user feedback*.

II. TINJAUAN PUSTAKA

A. Live Audio Streaming

Live audio streaming adalah suatu teknologi yang digunakan untuk memainkan atau memberikan akses untuk melihat *file*

audio secara langsung (*real-time*) dari sebuah *server* tanpa harus melalui proses *download*.

B. Raspberry Pi

Dunia Raspberry Pi berkembang dengan pesat dengan berbagai varian papan antar muka [1]. *Raspberry pi* (Gambar 1) adalah *single board circuit* yang memiliki ukuran sebesar kartu kredit. *Raspberry pi* merupakan komputer mini berkemampuan lengkap dengan sistem operasi linux. Selain Linux yang merupakan induk sistem operasinya, Raspberry Pi memiliki tujuh sistem operasi pilihan yang dapat diinstall pada raspberry pi yaitu *Bodhi*, *GeeXbox*, *Pidora*, *Raspbian*, *Raspbmc*, *RaspyFi*, *RISC OS Open*. [2]



Gambar 1. Raspberry Pi

C. Broadcast

Broadcast adalah suatu metode pengiriman sinyal ke berbagai lokasi secara bersamaan baik melalui satelit, radio, televisi, komunikasi data pada jaringan dan lain sebagainya tanpa harus melakukan pemeriksaan terhadap kesiapan di lokasi tujuan tersebut atau tanpa memperhatikan data yang dikirimkan tersebut sampai atau tidak. *Broadcast* dapat juga didefinisikan sebagai layanan *server* ke *client* dengan menyebarkan data kepada beberapa *client* sekaligus secara paralel dengan akses yang cukup cepat dari sumber *video* atau *audio*. Contoh penggunaan sistem ini adalah siaran televisi dan radio. [3]

D. Streaming

Streaming merupakan istilah sebuah *file video* maupun *audio* yang dapat dimainkan tanpa terlebih dahulu dilakukan pengunduhan untuk file tersebut. *Streaming* juga dapat diartikan teknik yang digunakan untuk melakukan transfer data sehingga dapat diproses secara tetap dan berlanjut [3]. Teknologi *streaming* berkembang sesuai dengan perkembangan internet. Saat ini kebanyakan user internet masih belum memiliki koneksi *broadband* untuk mengunduh file multimedia berukuran besar dengan cepat.

Streaming identik dengan waktu nyata (*realtime*). Namun tidak dapat dipungkiri setiap media *streaming* memiliki kendala waktu tunda (*delay*). Waktu tunda adalah jumlah waktu yang tertinggal dengan waktu *real* atau waktu nyata. Waktu tunda terjadi pada proses *streaming* tersebut dikarenakan *video streaming* merupakan metode pengiriman data berupa audio atau video sehingga terdapat proses-proses tertentu seperti

proses *encoding* yang akhirnya menghasilkan waktu tunda. Walaupun demikian, waktu tunda pada *video streaming* tidak memakan waktu yang lama, sehingga pemanfaatannya lebih banyak digunakan dibandingkan dengan metode *transfer audio* ataupun *video via on demand* dan *via download*.

Adapun konsep utama dari proses penerimaan aliran data yaitu [3]:

1. Proses Unduh (*Download*)

Pada penerimaan dengan cara pengunduhan, akses file dilakukan dengan cara melakukan *download* terlebih dahulu suatu *file* dari *server*. Penggunaan ini mengharuskan keseluruhan file diterima secara lengkap oleh *client*.

2. *Streaming*

Pada penerimaan *video* secara *streaming*, pengguna dapat melihat atau mengakses suatu *file* multimedia hampir bersamaan ketika file tersebut mulai diterima. Penggunaan cara ini mengharuskan pengiriman suatu *file* multimedia ke pengguna secara konstan.

3. *Progressive download*

Progressive download adalah suatu metode *hybrid* yang merupakan hasil penggabungan antara metode pengunduhan dan *streaming*. *File* yang sedang diakses dapat diterima dengan cara diunduh menggunakan aplikasi *player* yang ada pada pengguna.

Adapun metode transmisi *streaming* yaitu [3] :

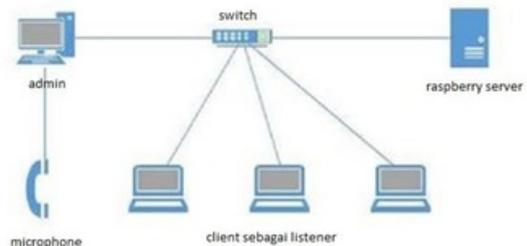
1. Transmisi *unicast*

Transmisi *unicast* adalah metode koneksi secara langsung antara pengguna dan *server*. Transmisi *unicast* bersifat *end to end*, setiap *client* mendapatkan stream data yang berbeda. Meskipun data atau file yang dialirkan sama, namun data yang dialirkan melalui jaringan akan disalin dan diterima oleh setiap *client* berbeda, sehingga menguras *bandwidth*.

2. Transmisi *multicast*

Transmisi *multicast* adalah metode distribusi dari satu sumber untuk banyak pengguna, atau dari satu grup untuk banyak pengguna. Tidak ada koneksi langsung antara pengguna dan *server*. Metode transmisi *multicast* sesuai diterapkan pada proses *streaming* karena *bandwidth* yang diproses sama, setiap user mengakses suatu *file* secara bersamaan dari *server*.

E. Pemodelan Live Audio Streaming



Gambar 2. Konsep sistem live audio streaming


```

GNU nano 2.2.6      File: /etc/bind/db.192
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@        IN      SOA      ns.radioku.com. root.radioku.com. (
; Serial
; Refresh
; Retry
; Expire
; Negative Cache TTL
)

@        IN      NS       ns.
1        IN      PTR      ns.radioku.com.
1        IN      PTR      www.radioku.com.

```

Gambar 6 Zone Reverse

- Memodifikasi *named.conf* untuk *zone forward* dan *reverse* dengan perintah “nano /etc/bind/named.conf.default.zones”
- Selanjutnya tambahkan *name server* 192.168.1.1 dengan perintah “nano /etc/resolv.conf”
- Karena raspberry tidak bisa mengecek keberhasilan *dns*, maka lakukan pemasangan *dnstools* dengan perintah “apt-get install dnstools”.
- Melakukan pemeriksaan konfigurasinya sudah berjalan atau tidak dengan perintah “named-checkconf”, “named-checkzone db.radioku db.192”
- *Restart dns server* dengan perintah “/etc/init.d/bind9 restart”
- Melakukan pemeriksaan *name server* dengan perintah “nslookup www.radioku.com” maka akan tampil seperti pada gambar 7.

```

pi@raspberrypi ~
root@raspberrypi:/home/pi# nslookup www.radioku.com
Server:      192.168.1.1
Address:     192.168.1.1#53

Name:   www.radioku.com
Address: 192.168.1.1

root@raspberrypi:/home/pi#

```

Gambar 7 Pemeriksaan dengan Nslookup

3. Melakukan konfigurasi aplikasi untuk *Live Audio streaming*. Untuk aplikasi *live audio streaming* ini, membutuhkan aplikasi *Icecast2* pada *raspberry pi* dan SAM Broadcast pada laptop admin. Konfigurasi dilakukan dengan tahapan sebagai berikut:
 - Memasang *Icecast2* di *raspberry* dengan perintah “apt-get install icecast2”.
 - Melakukan perubahan *user*, *password* dan *port* nya dengan perintah “nano /etc/icecast2/icecast.xml” seperti pada gambar 8.

```

GNU nano 2.2.6      File: /etc/icecast2/icecast.xml
</limits>
<authentication>
  <!-- Sources log in with username 'source' -->
  <source-password>hdaya</source-password>
  <!-- Relays log in username 'relay' -->
  <relay-password>hdaya</relay-password>

  <!-- Admin logs in with the username given below -->
  <admin-user>admin</admin-user>
  <admin-password>hdaya</admin-password>
</authentication>

<!-- set the mountpoint for a shoutcast source to use, the default if not
specified is /stream but you can change it here if an alternative is
wanted or an extension is required
<shoutcast-mount>/live.nav</shoutcast-mount>
-->

<!-- Uncomment this if you want directory listings -->
<!--
<directory>
  <yp-url-timeout>15</yp-url-timeout>
  <yp-url>http://dir.xiph.org/cgi-bin/yp.cgi</yp-url>
</directory>
-->

<!-- This is the hostname other people will use to connect to your server.
It affects mainly the urls generated by Icecast for playlists and yp
listings. -->
<hostname>hdaya</hostname>

<!-- You may have multiple <listeners> elements -->
<listen-socket>
  <port>8000</port>
  <!-- <bind-address>127.0.0.1</bind-address> -->
  <!-- <shoutcast-mount>/stream</shoutcast-mount> -->
</listen-socket>

```

Gambar 8 icecast.xml

- Melakukan perubahan file *icecast2* yang berada di /etc/default/icecast2 yang sebelumnya *false* menjadi *true*.
- Kemudian beralih ke komputer yang akan dipakai untuk melakukan *broadcast* suara (komputer admin).
- Setelah menginstal aplikasi SAM Broadcaster dan *database firebird*, pilih *desktop B* pada SAM Broadcaster seperti pada gambar 9.

Pada aplikasi SAM Broadcast dilakukan beberapa *setting* yaitu pemilihan format suara, *bit rate*, dan *server* yang digunakan dengan mengubah *ip server*, *server port* dan *password* sesuai dengan konfigurasi pada raspberry pi. Pada tahap ini maka admin bisa melakukan *broadcast audio* ke *client*.



Gambar 9 SAM Broadcast.xml

4. Melakukan konfigurasi *shoutbox*.

Pembuatan *Shoutbox* di *web* menggunakan *mysql-server*, *mysql-client*, *php5*, *phpmyadmin*, yaitu dengan melakukan pembuatan basis data menggunakan *mysql*. Pembuatan basis data dapat juga dilakukan dengan menggunakan *phpmyadmin*.

5. Melakukan konfigurasi *mp3 streaming*.

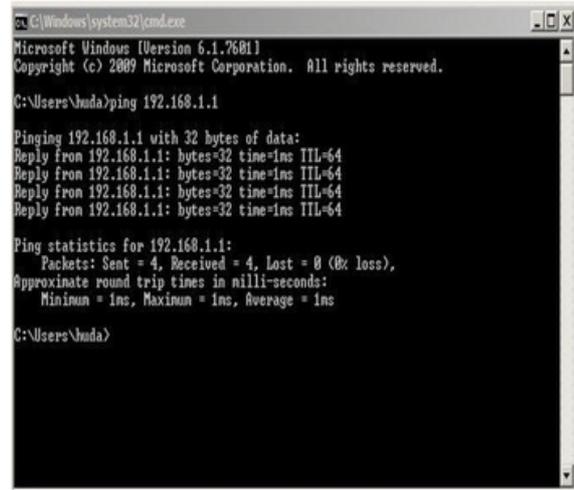
Mp3 streaming dibuat menggunakan aplikasi *gnump3d* dengan melakukan konfigurasi pengaturan *server mp3 streaming* pada file *gnump3d.conf*. Konfigurasi dilakukan terhadap port dan folder *mp3* dan penggunaanya

Pengujian yang dilakukan meliputi :

1. Dilakukan proses ping antar jaringan internal sehingga terhubung satu dengan yang lain.
2. *Client* melakukan akses *live audio streaming*.
3. *Client* melakukan akses *shoutbox*.
4. *Client* melakukan akses *mp3 streaming*.

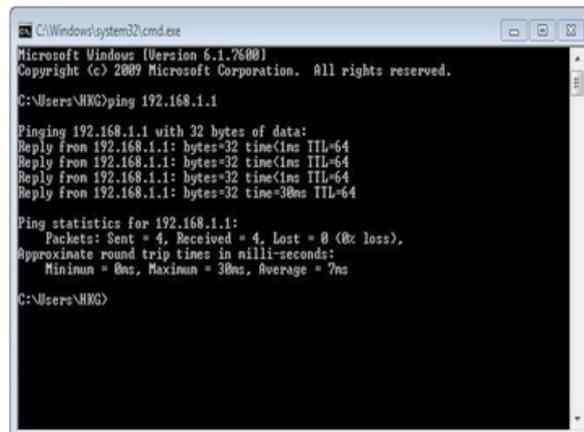
Adapun pengujian dilakukan dengan langkah sebagai berikut.

1. Pengujian pertama adalah dengan melakukan proses *Ping* antar jaringan internal yaitu :
 - a. Pengujian proses Admin ke *server* seperti pada gambar 10. Pada gambar tersebut terlihat proses tersebut berhasil dilakukan.
 - b. Kemudian pengujian dilanjutkan dengan melakukan proses *Ping* dari *Client* ke *Server* seperti pada gambar 10.



Gambar 10 Proses Ping Admin - Server Raspberry pi

2. Pengujian selanjutnya adalah ketiga *client* melakukan akses *live audio streaming* melalui *web server*. Pada tahap ini *client* berhasil melakukan akses dan tampilan *web* halaman utama oleh *client* seperti pada gambar 11. Hasil pengujian diperoleh kedua *client* berhasil melakukan akses dengan tampilan akses ditunjukkan pada gambar 11,12,13.



Gambar 11 Ping Client - Server Raspberry pi



Gambar 12 Tampilan Halaman Utama



Gambar 13 Tampilan Live Audio Streaming client 3

Pengujian terhadap waktu akses dilakukan untuk mengetahui kecepatan delay waktu akses ketiga client tersebut. Hasil pengujian diperlihatkan pada tabel 1.

TABLE 1 DELAY AKSES CLIENT PADA MENU AUDIO STREAMING

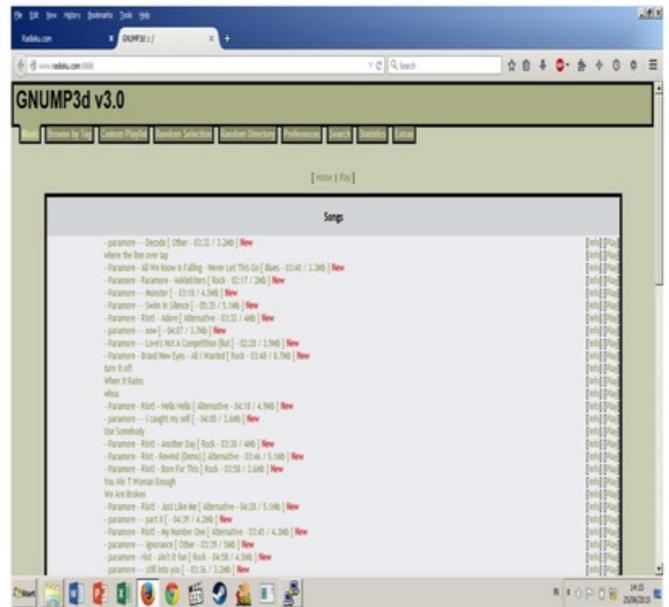
Client	Delay
1	2 detik
2	2,25 detik
3	2,5 detik

3. Pengujian selanjutnya adalah *client* melakukan akses *web shoutbox*. Pada tahap ini *client* berhasil melakukan akses *web shout box* seperti yang ditunjukkan pada gambar 14.



Gambar 14 Tampilan Shoutbox

4. Pengujian pengaksesan *mp3 streaming* oleh *client* dilakukan dengan cara mengakses audio dengan format *.mp3*. Gambar 8 menunjukkan tampilan dari *Mp3streaming*.



Gambar 15 Tampilan Mp3 Streaming

V. KESIMPULAN

1. Raspberry Pi dapat diimplementasikan sebagai server untuk *Live Audio Streaming*, *user feedback* dan *mp3 streaming*.
2. Aplikasi web untuk *live audio streaming* dengan server Raspberry Pi dapat berjalan dan berhasil diimplementasikan namun masih terdapat delay sekitar 2 detik untuk *audio* setelah *broadcast* oleh *admin* kepada *user*.

VI. REFERENCES

- [1] S. Monk, Raspberry Pi CookBook, O'Reilly Media, 2013.
- [2] S. Nazarko, Raspberry Pi Media Center, Birmingham: Packt Publishing Ltd, 2013.
- [3] D. Austerberry, The Technology of Video and Audio Streaming, San Francisco: Focal Press, 2004.
- [4] K. Yaghmour, Building Embedded Linux System, Newyork: O'Reilly Media Inc, 2008.
- [5] J. W. E. Shotts, The Linux Command Line: A Complete Introduction, San Francisco: William Pollock, 2012.

Implementasi Live Audio Streaming Menggunakan Raspberry Pi

ORIGINALITY REPORT

16%

SIMILARITY INDEX

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

★cintiaelind.blogspot.com

Internet

4%

EXCLUDE QUOTES ON

EXCLUDE MATCHES OFF

EXCLUDE BIBLIOGRAPHY ON