

KOMPRESI VIDEO KAMERA KEAMANAN BERBASIS RASPBERRY PI

M. H. Sita¹, S. I. Lestaringati²

^{1,2}Teknik Komputer Universitas Komputer Indonesia, Bandung
¹hillmansita10@gmail.com, ²susmini.indriani@email.unikom.ac.id

ABSTRAK

Perangkat keras media penyimpanan merupakan salah satu perangkat yang paling penting pada sebuah sistem atau alat untuk menyimpan kumpulan data dan informasi, salah satunya media penyimpanan pada sistem kamera keamanan. Pada umumnya setelah melakukan proses perekaman, sistem kamera keamanan langsung menyimpan hasil rekaman ke sebuah media penyimpanan. Hasil rekaman yang diperoleh pada umumnya berupa video mentah (raw video) dengan ukuran file yang cukup besar. Maka dibangunlah sistem kompresi video pada sebuah kamera keamanan yang dapat melakukan kompresi dan konversi video. Proses kompresi yang dilakukan meliputi pemadatan hingga melakukan konversi pada video mentah (transcode), misalnya pada video mentah dengan format h264 yang dikompresi dan dikonversi menjadi video MKV (Matroska Video). Dengan diterapkannya sistem kompresi pada kamera keamanan berbasis Raspberry Pi, diharapkan dapat mengurangi biaya untuk perangkat keras media penyimpanan. Dari hasil pengujian yang dilakukan, sistem kompresi video telah berhasil melakukan kompresi pada sebuah video dengan nilai kompresi lebih dari 50%.

Kata Kunci : H264, Kompresi, Konversi.

ABSTRACT

Storage media hardware is one of the most important devices in a system or device for storing data and informations, for example is a storage on a security camera system. Usually after doing the recording process, the security camera system immediately saves the recorded video files to a storage media. The recorded videos are usually in the form of raw video with a fairly large file size. Therefore, a video compression system is built on a security camera that can do video compression and conversion. The compression process involves file compacting and also converting on raw video (transcode), for example on h264 raw video compressed and converted into MKV (Matroska Video). By applying this compression method to the device, it allows to make storage device is not filled quickly and also can reduce the cost of storage device hardware. From the test results performed, video compression system has managed to compress a video with a compression value more than 50%.

Keywords: H264, Compression, Conversion.

I. PENDAHULUAN

Jaringan bandwidth yang terus meningkat, koneksi internet yang semakin cepat dan semakin besarnya kapasitas pada media penyimpanan tidak serta merta membuat proses kompresi pada file multimedia tidak diperlukan. Kompresi adalah proses mengecilkan ukuran sebuah data dengan cara mengurangi jumlah bit yang tidak diperlukan dan kemudian dipresintasikan menjadi file yang baru. Proses kompresi video dilakukan oleh sebuah program menggunakan fungsi atau algoritma tertentu untuk menentukan bagaimana file tersebut akan dikecilkan. Metode kompresi telah dioptimalkan pada penyimpanan cadangan, namun baru-baru ini seringkali digunakan untuk mengurangi ukuran berlebih pada penyimpanan primer.

Dengan adanya fitur kompresi video pada sebuah transmisi atau sebuah alat, ukuran file dapat

dipadatkan hingga 50% dari ukuran asalnya. Proses kompresi yang dilakukan meliputi pemadatan hingga melakukan konversi pada file mentah (raw file), misalnya pada video H.264 yang dikompresikan menjadi video MKV (matroska video). Metode kompresi yang akan digunakan adalah Lossless Compression, yaitu mengeliminasi bit berlebih atau bit yang tak terpakai dimana saat bit-bit tersebut dihapus, file yang telah dikompresi terlihat tidak jauh berbeda dengan file asalnya.

II. TEORI PENUNJANG

A. Raspberry Pi

Raspberry Pi adalah sebuah mikro komputer dimana Raspberry Pi ini sudah memiliki prosesor, RAM dan berbagai port. Raspberry Pi menggunakan bahasa pemrograman Python. Sistem operasi Raspberry Pi yang paling umum digunakan adalah

Raspbian. Raspbian merupakan system operasi yang berbasis Debian.[3].

B. H.264 CODEC

H.264 dikenal juga sebagai MPEG-4 AVC (Advanced Video Coding) merupakan salah satu standar untuk representasi kode informasi visual. Sama dengan standar pengkodean sebelumnya, H.264 tidak secara tegas mendefinisikan CODEC (enCOder/DECOder) tetapi lebih pada sintaks dari bitstream video yang dikodekan secara bersamaan dengan metode decoding bitstream. [1].

C. Kompresi

Kompresi merupakan proses mengecilkan ukuran sebuah data dengan cara mengurangi jumlah bit yang tidak diperlukan dan kemudian dipresintasikan menjadi file yang baru. Proses kompresi video dilakukan oleh sebuah program menggunakan fungsi atau algoritma tertentu untuk menentukan bagaimana file tersebut akan dikecilkan. Kompresi video memiliki dua manfaat penting. Pertama, memungkinkan video untuk digunakan di lingkungan transmisi dan penyimpanan yang tidak mendukung video yang tidak dikompres (raw file). Kedua, kompresi video memungkinkan penggunaan sumber daya transmisi dan penyimpanan lebih efisien, dengan kata lain proses kompresi membuat media penyimpanan tidak cepat penuh. [1].

D. Video Encoder

Video encoder bertugas untuk mengkodekan video atau urutan frame pada video ke dalam bentuk terkompresi dan menerjemahkannya untuk menghasilkan salinan atau file baru. Sebuah video encoder terdiri dari tiga unit fungsional utama: temporal model, spatial model dan encoder entropi. Masukan ke temporal model adalah urutan video yang tidak dikompres. Model temporal mencoba mengurangi redundansi temporal dengan memanfaatkan kemiripan frame video sebelumnya, biasanya dilakukan dengan membuat prediksi frame video pada waktu tertentu. [2].

- *FFmpeg*

Fast Forward Motion Pictures Expert Group (FFmpeg) merupakan kerangka multimedia terkemuka yang mampu melakukan decode, encode, transcode, mux, demux, stream, filter dan melakukan pemutaran video dan audio dalam berbagai format seperti, Motion Pictures Expert Group (MPEG), H.264, dan Audio Video Interleave (AVI). FFmpeg mendukung format-format multimedia terdahulu hingga format-format multimedia terbaru yang dipakai saat ini. FFmpeg sendiri juga sangat portable dan dapat melakukan beberapa pekerjaan seperti

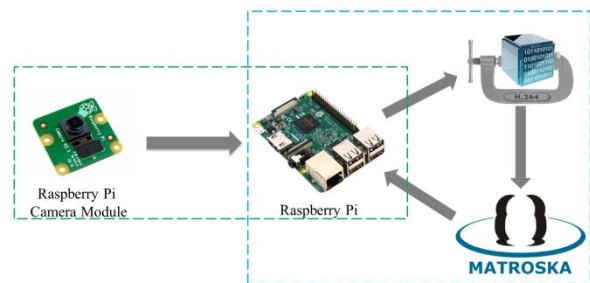
mengkompilasi, menjalankan, dan telah melewati infrastruktur pengujian FATE (FFmpeg Automated Testing Environment) pada sistem operasi Linux, Mac OS X, Microsoft Windows, BSD, Solaris, dll.[4].

E. Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, untuk menjadikan citra yang kualitasnya lebih baik. pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas yang berbeda daripada citra masukan. [5].

III. PERANCANGAN

Pada gambar 1 menjelaskan gambaran umum sistem, bahwa Pi Camera berfungsi sebagai input yang menghasilkan *raw video* (video mentah). Kemudian video tersebut diproses oleh Raspberry Pi 3. Selanjutnya, data citra berupa video mentah akan di kompresikan dan dikonversikan dari H.264 ke format MKV yang kemudian disimpan kembali ke penyimpanan Raspberry Pi.

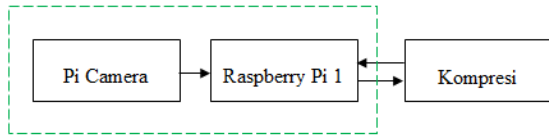


Gambar 1 Gambaran Umum Sistem

Keterangan gambaran umum dari sistem kompresi video:

1. Raspberry Pi Camera Module v2: Sebagai input atau alat untuk memperoleh sumber data, dimana data tersebut merupakan video mentah yang belum diolah.
2. Raspberry Pi 3 Type B: Sebagai pusat pengolah dan pemroses data, dimana data yang masuk dari Pi Camera akan diolah untuk dilakukan proses kompresi.
3. H.264: Merupakan salah satu standar untuk representasi kode informasi visual, dimana video mentah yang dihasilkan Pi Camera direpresentasikan oleh H.264 CODEC.

4. MKV (Matroska Video): Hasil / output dari proses kompresi dan konversi, video mentah yang telah dikompresi dan dikonversi dari H.264 video menjadi Matroska Video kemudian disimpan ke media penyimpanan Raspberry Pi.



Gambar 2 Diagram Blok Sistem

Dari Gambar 2 di atas, pada saat Pi Camera menghasilkan sebuah video mentah, video tersebut disimpan terlebih dahulu di memori kemudian dilakukan proses kompresi serta merubah formatnya dari H.264 menjadi MKV.

A. Analisis Kebutuhan Sistem

Analisis perangkat keras bertujuan untuk memudahkan proses perancangan dan implementasi dalam pembangunan system Kompresi Video Kamera Keamanan Berbasis Raspberry Pi. Adapun kebutuhan perangkat keras yang dibutuhkan oleh pengguna untuk aplikasi yang dibangun dapat dilihat pada tabel dibawah ini.

Tabel 1 Kebutuhan Sistem

| No | Nama | Spesifikasi |
|----|-----------------------|---|
| 1 | Raspberry Pi 3 Type B | <ul style="list-style-type: none"> • Quad Core 1.2GHz Broadcom BCM2837 64bit CPU • 1GB RAM • BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board • 40-pin extended GPIO • 4 USB 2 ports • 4 Pole stereo output and composite video port • Full size HDMI • CSI camera port for connecting a Raspberry Pi camera • DSI display port for connecting a Raspberry Pi touchscreen display • Micro SD port for loading your operating system and storing data • Upgraded switched Micro USB power source up to |

| | | |
|---|-------------------------------|---|
| | | 2.5A |
| 2 | Raspberry Pi Camera Module v2 | <ul style="list-style-type: none"> • Image Sensor: Sony IMX219 • Optical Size: 1/4" • Still Resolution: 8 Megapixels • Maximum Photo Resolution: 3280 x 2464 pixel • Supported Video Resolution: 1080p30, 720p60 and 640x480p90 • Physical Dimensions: 25mm x 23mm x 9mm • Interface: CSI connector 15cm ribbon cable included • Supported OS: Raspbian |
| 3 | Raspbian Stretch | Raspbian Stretch |
| 4 | Python | Python 2.4.7 |

B. Instalasi Codec

Untuk melakukan proses kompresi pada video dibutuhkan *tools* dan *codec library* dan dimana codec tersebut harus diterapkan terlebih dahulu pada Raspberry Pi. Instalasi codec meliputi YASM Assembler, x264-codec, libvpx serta codec FFmpeg. Langkah pertama yang dilakukan untuk melakukan instalasi codec yaitu membuat direktori file untuk menyimpan library codec serta menginstall *build tools* seperti *command line* yang diketik melalui terminal dibawah ini.

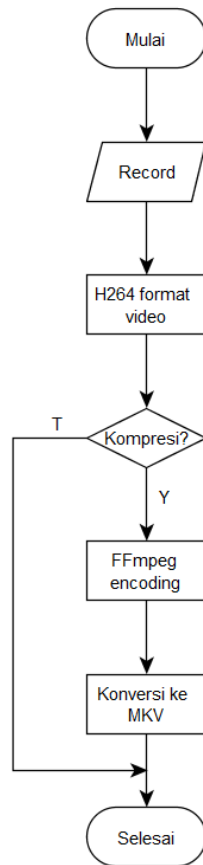
```

cd ~
mkdir ~/FFmpeg_sources
mkdir ~/FFmpeg_build
#buildtools
Sudo apt-get -y install autoconf automake build-essential libass-dev libfreetype6-dev \
  libssl1.2-dev libtheora-dev libtool libva-dev \
  libvdpau-dev libvorbis-dev libxcb1-dev libxcb-shm0-dev \
  libxcb-xfixes0-dev pkg-config texinfo zlib1g-dev
  
```

C. Algoritma Utama Sistem

Pada Gambar 3 menunjukkan algoritma utama sistem berupa diagram alir sistem encoding dan

decoding, berikut merupakan gambar dari diagram alir:



Gambar 3 Algoritma Utama Sistem

IV. PENGUJIAN

A. Pengujian Modul Kamera

Pengujian modul kamera dilakukan dengan cara pengambilan gambar (capturing) yang dapat di eksekusi pada terminal atau pada program Python. Hasil dari pengambilan gambar yang dilakukan kamera dapat dilihat pada gambar 4



Gambar 4 Hasil pengambilan gambar modul kamera

B. Pengujian Proses Transcoding

Pada tahap ini akan dilakukan transcoding dengan menggunakan encoder ffmpeg. Pada proses ini video akan di kompres dengan metode lossless compression menggunakan x264 codec. Faktor penting dalam melakukan transcoding salah satunya adalah penentuan preset. Preset merupakan opsi yang mengoptimalkan efisiensi kompresi (kualitas per bit) yang bergantung pada kecepatan proses encoding (frame yang dikodekan per detik). Semakin cepat opsi preset yang digunakan, semakin banyak frame yang dikodekan dalam satu detik dan waktu yang dibutuhkan untuk encoding semakin singkat. Sebaliknya jika semakin lambat opsi preset, maka semakin sedikit frame yang dikodekan dan waktu yang dibutuhkan semakin lama. Berikut adalah dokumentasi saat pengujian beserta kinerja cpu pada modul Raspberry Pi.

C. Pengujian Hasil Transcode

Setelah mendapatkan hasil encode dari pengujian transcoding, selanjutnya akan dilakukan pengujian perbandingan hasil dengan menyertakan metadata dan screenshot di durasi tertentu pada video.

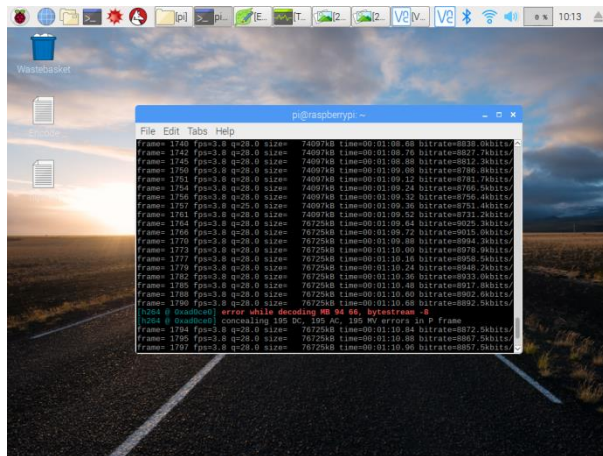
1. Percobaan Video Resolusi 1920x1080p

Tabel 3 Percobaan Video Resolusi 1920x1080p

| No | Jenis / Preset | Durasi | Ukuran (MB) | Waktu Encode | Ket. |
|----|----------------|--------|-------------|--------------|------|
| 1 | Raw | 05:52 | 595.6 | - | - |
| | Ultrafast | | 542.3 | 14:10 | ✓ |
| | Medium | | 413 | 01:56:38 | ✓ |
| 2 | Raw | 01:11 | 121.4 | - | - |
| | Ultrafast | | 110.1 | 02:25 | ✓ |
| | Superfast | | 102.4 | 04:40 | ✓ |

| | | | |
|---------|------|----------|---|
| Fast* | 91.2 | 07:51 | ✓ |
| Medium* | 84.5 | 20:59 | ✓ |
| Slow | 84.2 | 30:52:00 | ± |

✓ = encode berhasil ; ± = file sudah tidak dapat diperkecil ; * = percobaan diulang



Gambar 5 Notifikasi error pada percobaan kedua

Pada percobaan ini, dilakukan beberapa kali pengujian transcoding dengan preset berbeda karena terdapat notifikasi error di akhir proses. Notifikasi error selalu muncul pada percobaan kedua. Untuk preset yang lebih lambat dari preset “fast” ukuran file mkv selalu dikisaran ±80MB dan ukuran filenya tidak bisa lagi untuk dikecilkan.



Gambar 6 Perbandingan kualitas video antar preset

2. Percobaan Video Resolusi 1280x720p

Tabel 4 Percobaan Video Resolusi 1280x720p

| No | Jenis / Preset | Durasi | Ukuran (MB) | Waktu Encode | Ket. |
|----|----------------|--------|-------------|--------------|------|
| 1 | Raw | 05:52 | 324.6 | - | - |
| | Ultrafast | | 112.7 | 05:35 | ✓ |
| | Medium | | 35.7 | 22:29 | ✓ |
| | Slow | | 35.5 | 45:03:00 | ± |
| 2 | Raw | 01:10 | 86.1 | - | - |
| | Ultrafast | | 35.2 | 01:11 | ✓ |
| | Medium | | 11.6 | 05:41 | ✓ |
| | Slow | | 10.7 | 08:15 | ± |

✓ = encode berhasil ; ± = file sudah tidak dapat diperkecil ; * = percobaan diulang



Gambar 7 Perbandingan kualitas video antar preset

D. Analisis

Berdasarkan hasil pengujian yang telah dilakukan, sistem dapat melakukan transcoding dan memenuhi tujuan dari dibuatnya sistem kompresi video. Notifikasi error yang muncul pada percobaan kedua disebabkan karena terdapat glitch pada durasi 1:09-1:10 yang menyebabkan file tidak dapat diperkecil kembali setelah preset medium. Berikut merupakan data efisiensi dari proses kompresi yang telah didapatkan dari proses pengujian.

Tabel 5 Analisis efisiensi dari proses kompresi

| No | Preset | Ukuran Asal > setelah dikompres (MB) | Efisiensi (%) |
|----|-----------|--------------------------------------|---------------|
| 1 | Ultrafast | 595.6 > 542.3 | 9 |
| | Medium | 595.6 > 413 | 30 |
| 2 | Ultrafast | 121.4 > 110.1 | 9 |

| | | | |
|---|-----------|---------------|----|
| | Superfast | 121.4 > 102.4 | 15 |
| | Fast | 121.4 > 91.2 | 24 |
| | Medium | 121.4 > 84.5 | 30 |
| | Slow | 121.4 > 84.2 | 30 |
| 3 | Ultrafast | 324.6 > 112.7 | 65 |
| | Medium | 324.6 > 35.7 | 89 |
| | Slow | 324.6 > 35.5 | 89 |
| 4 | Ultrafast | 86.1 > 35.2 | 59 |
| | Medium | 86.1 > 11.6 | 86 |
| | Slow | 86.1 > 10.7 | 87 |

Pada percobaan pengambilan video dan melihat perbandingan kualitas video seperti gambar 8, dapat dilihat bahwa video 1280x720 pixel lebih baik ketimbang resolusi default yaitu 1920x1080 pixel. Presentase efisiensi pada proses kompresi resolusi 720p lebih tinggi dibandingkan resolusi default 1080p. Pada proses kompresi video dengan resolusi 720p, video h264 dapat dikompres dengan tingkat efisiensi hingga 89%. Sedangkan video dengan resolusi 1080p hanya dapat dikompres hingga 30%.



Gambar 8 Perbandingan kualitas 1080p dan 720p

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pengujian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

Berdasarkan pengujian dari semua sampel, sistem dapat melakukan proses transcoding mulai dari melakukan kompresi hingga melakukan konversi format data.

1. Pada pengujian video 1080p, presentase efisiensi dari proses kompresi yang dilakukan sistem adalah 9% hingga 30%.
2. Pada pengujian kedua, terdapat notifikasi error yang disebabkan oleh adanya glitch dari video mentah yang di encode.
3. Pada pengujian video 720p, presentase efisiensi dari proses kompresi yang dilakukan sistem adalah 59% hingga 89%.
4. Berdasarkan pengujian perbandingan hasil, kualitas video mentah hampir sama dengan kualitas video yang telah di encode.
5. Selama proses transcoding, tingkat kesibukkan CPU selalu diatas 90%.

B. Saran

Adapun saran untuk dilakukan pengembangan pada sistem ini seperti mengubah format hasil transcoding ke format yang lain supaya dapat digunakan untuk fungsi stream / live feed seperti format mp4.

DAFTAR PUSTAKA

- [1] Richardson, Iain E.G (2003): H.264 and MPEG-4 Video Compression. UK: Wiley.
- [2] Richardson, Iain E.G (2010): H.264 The Advanced Video Compression Standard. UK: Wiley.
- [3] M. Richardson and S. Wallace (2012): Getting Started with Raspberry Pi. USA: O'Reilly Media.
- [4] About FFMPEG (Online), diakses pada tanggal 10 Maret 2018 dari world wide web: <https://www.ffmpeg.org/about.html>
- [5] Rinaldi Munir. (2004): Pengolahan Citra Digital. Bandung : Informatika