BAB II
TEORI PENUNJANG

Pada bab berikut akan membahas beberapa teori penunjang serta perangkat dan juga alat yang digunakan untuk Kompresi Video Kamera Keamanan Berbasis Raspberry Pi. Pembahasan meliputi Raspberry Pi, Kompresi, H.264 CODEC, *Video Encoder*, dan Pengolahan Citra.

2.1 Raspberry Pi

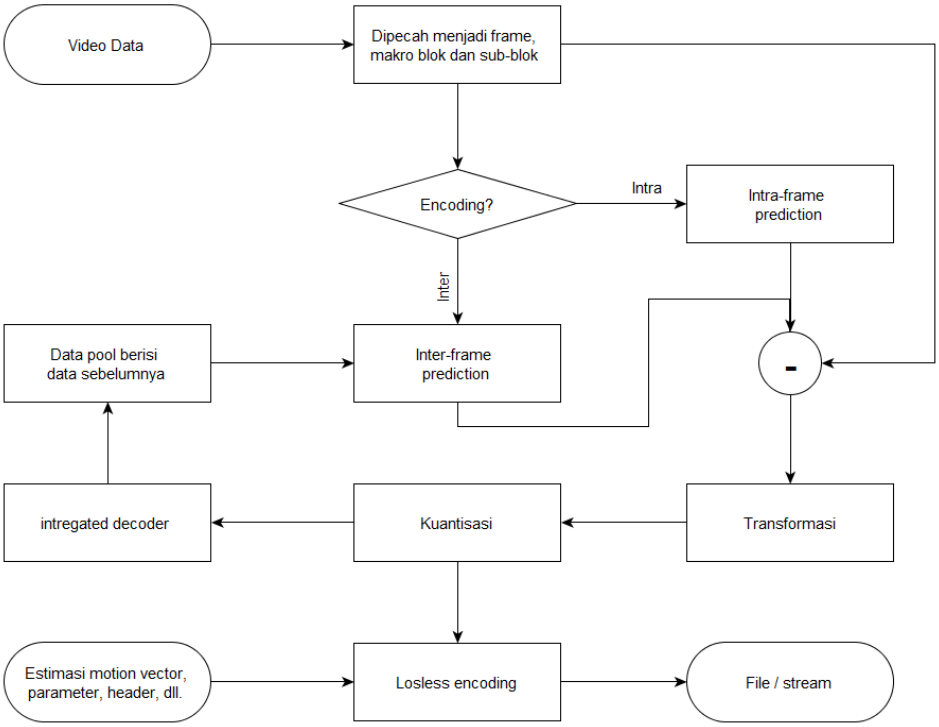
Raspberry Pi adalah sebuah mikro komputer dimana Raspberry Pi ini sudah memiliki prosesor, RAM dan berbagai port. Raspberry biasanya digunakan untuk mengganti sebuah komputer yang memakan daya lebih besar yang biasanya digunakan untuk media server, NAS (*Network Attached Storage*), server hosting website, dll. Raspberry Pi menggunakan bahasa pemrograman Python. Sistem operasi Raspberry Pi yang paling umum digunakan adalah Raspbian. Raspbian merupakan sistem operasi yang berbasis Debian (*based on debian*). [1]. Pada gambar 2.1 merupakan tampilan berserta port-port yang ada pada Raspberry Pi 3 tipe B.

Gambar 2.1 Raspberry Pi 3 type B

2.2 H.264 CODEC

H.264 CODEC 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. Sebagian besar elemen fungsional dasar seperti prediksi, transformasi, kuantisasi, dan pengkodean entropi hadir dalam standar sebelumnya (MPEG-1, MPEG-2, MPEG-4, H.261, H.263). H.264 CODEC merupakan salah satu standar yang paling umum digunakan pada perekaman, kompresi, dan distribusi konten video. [2].

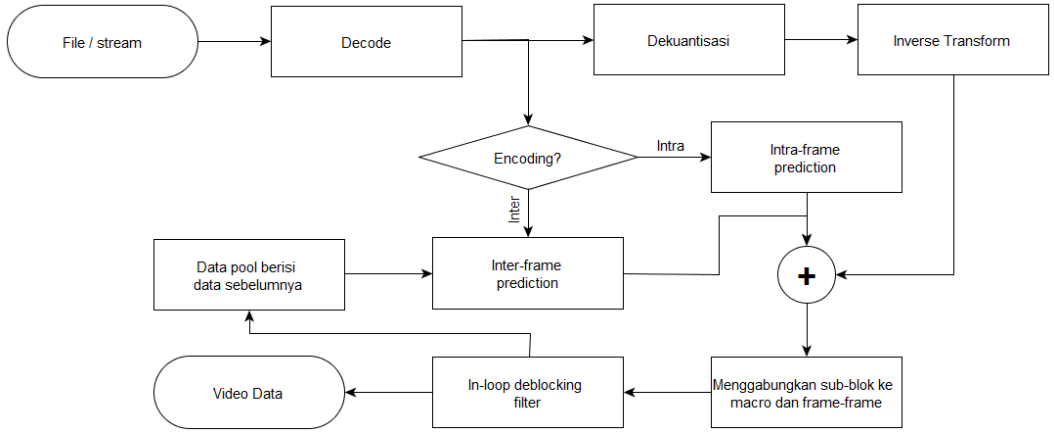


Gambar 2.2 Diagram Alir H.264 Encoder

Dari gambar 2.2 dapat dijelaskan bahwa data video pertama kali dibedah menjadi frame, makro blok dan sub-blok. Prediksi dibuat baik oleh *inter-frame prediction* atau *intra-frame prediction* untuk setiap blok. Data prediksi dikurangi dari data asli untuk mendapatkan data residual. Data ini kemudian diubah (*transform*), dikuantisasi dan dikodekan. Selain itu, ada decoder terintegrasi. Decoder terintegrasi ini digunakan untuk memastikan bahwa baik encoder dan decoder memiliki data yang sama untuk prediksi antar frame yang digunakan untuk kompresi. Dengan dilengkapi oleh parameter dan header, proses encode (*lossless*) akan menghasilkan file video atau aliran video (*stream*).[3].

Dari gambar 2.3 dapat dijelaskan bahwa data video yang telah dikodekan (*encode*) pertama kali didekodekan menggunakan teknik lossless *decoding* yang

tepat. Data yang diperoleh lalu dilakukan dekuantisasi dan transformasi terbalik (*inverse transform*) untuk mendapatkan data residual.



Gambar 2.3 Diagram Alir H.264 Decoder

Berdasarkan parameter dalam data, prediksi ditambahkan ke data residual untuk mendapatkan satu blok 4x4 yang telah diterjemahkan. Beberapa sub-blok kemudian digabungkan menjadi satu *macroblock*, yang kemudian digabungkan bersama menjadi slice dan akhirnya menjadi frame. *In-loop deblocking filte*r diterapkan pada masing-masing dari 4x4 border, untuk mendapatkan frame video terakhir. Tergantung pada jenis frame, data disimpan untuk *inter-frame prediction* selanjutnya.[3].

2.3 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 gambar dan video telah menjadi bidang penelitian dan pengembangan yang sangat aktif selama lebih dari 20 tahun dan telah banyak sistem dan algoritma yang berbeda untuk kompresi dan dekompresi yang telah diusulkan dan dikembangkan. [2].

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. [2].

Kompresi data dicapai dengan menghapus redundansi, yaitu komponen yang tidak diperlukan untuk reproduksi data yang sebenarnya. Banyak jenis data mengandung redundansi statistik dan dapat dikompresi menggunakan kompresi lossless, sehingga data yang direkonstruksi pada output decoder adalah salinan sempurna dari data asli. [2].

Lossless compression didasarkan pada ide membuat file menjadi bentuk yang lebih kecil untuk transmisi atau penyimpanan dan kemudian direpresentasikan menjadi file baru sehingga dapat digunakan kembali. *Lossy compression* bekerja sangat berbeda, teknik ini didasarkan pada prinsip menghilangkan redundansi, elemen gambar atau urutan video yang dapat dihapus dan dapat mempengaruhi persepsi tentang kualitas visual. Jenis kompresi ini banyak digunakan untuk mengurangi ukuran file gambar bitmap, yang cenderung cukup besar. Dalam sistem *Lossy compression*, data yang telah dikompres tidak identik dengan sumber data dan rasio kompresi yang lebih tinggi dapat dilakukan dengan mengorbankan kualitas visual. [2].

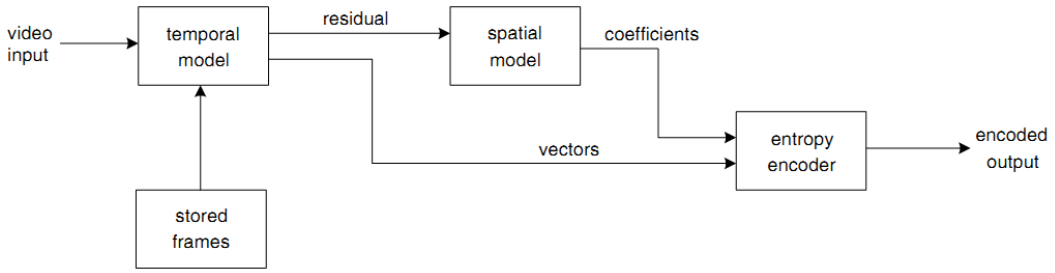
2.3.1 Transcoding

Transcoding pada video merupakan proses mengubah sinyal video terkompresi untuk menyesuaikan karakteristik video seperti bitrate video, resolusi video, atau codec video, sehingga dapat memenuhi spesifikasi saluran komunikasi dan perangkat *endpoint*. Sinyal video dikompresi dan didekompresi dengan teknik pengkodean video tertentu. Pada prosesnya, kompresi sering dilambangkan sebagai “encoder” dan dekompresi sebagai “DECoder”, yang secara kolektif membentuk istilah CODEC. Oleh karena itu, CODEC adalah kumpulan metode yang digunakan untuk melakukan kompresi dan mendekompresi video digital. [4].

2.4 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. [5].

Dalam MPEG-4 Visual dan H.264, prediksi terbentuk dari satu atau lebih frame sebelumnya atau frame yang akan datang dan diperbaiki dengan mengkompensasi perbedaan antara frame (*motion compensated prediction*). Output dari model temporal adalah kerangka residual (dibuat dengan mengurangi prediksi dari frame aktual) dan satu set parameter model, biasanya satu set vektor gerak yang menggambarkan bagaimana gerakan itu dikompensasikan. [5]. Berikut adalah diagram blok video encoder yang menggambarkan alur proses kompresi serta menggambarkan tiga unit fungsional utama pada video encoder yaitu *prediction model (temporal model), spatial model* dan *entropy encoder*:



Gambar 2.4 Diagram Blok Video Encoder

Input yang masuk ke *prediction model* merupakan urutan video 'mentah' yang tidak terkompresi. *Prediction model* mencoba mengurangi redundansi dengan mengeksploitasi kemiripan antara frame video sekuential, biasanya dengan membangun prediksi frame video saat ini atau blok data video. Dalam H.264 / AVC, prediksi dibentuk dari data dalam frame saat ini atau dalam satu atau lebih frame sebelumnya dan / atau yang akan datang. Hal tersebut dibangun oleh ekstrapolasi spasial dari sampel gambar yang bersebelahan, prediksi intra,

atau dengan mengkompensasi perbedaan antara frame, prediksi inter atau gerak kompensasi. Output dari model prediksi adalah frame residual, dibuat dengan mengurangi prediksi dari frame aktual, dan satu set parameter model yang menunjukkan tipe prediksi intra atau menggambarkan bagaimana gerakan dikompensasi.[5].

2.4.1 Prediction Model

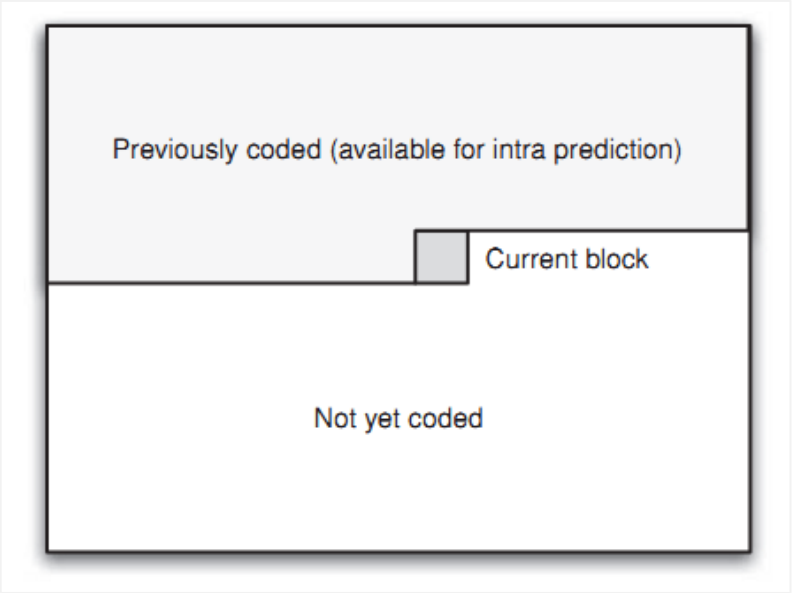
Data yang akan diproses adalah sekumpulan sampel citra atau gambar (*image*) dalam suatu *frame* atau sebuah *field* yang ada pada saat itu juga. Tujuan dari *prediction model* adalah untuk mengurangi redundansi dengan membentuk prediksi data dan mengurangi prediksi ini dari data yang ada pada saat ini. Prediksi dapat dibentuk dari frame yang dikodekan sebelumnya (prediksi temporal) atau dari sampel gambar yang dikodekan sebelumnya dalam bingkai yang sama (prediksi spasial). Output dari proses ini adalah satu set sampel residual atau perbedaan sampel dan semakin akurat proses prediksi, semakin sedikit energi yang terkandung pada bagian residual (residu). Residu dikodekan dan dikirim ke decoder yang menciptakan kembali prediksi yang sama sehingga dapat menambahkan residu yang telah didekodekan dan merekonstruksi frame saat ini. Agar decoder dapat membuat prediksi yang sama, penting untuk sebuah encoder membentuk prediksi dengan hanya menggunakan data yang tersedia pada decoder, yaitu data yang sebelumnya telah dikodekan dan ditransmisikan. [5].

2.4.1.1 Temporal prediction

Frame yang telah diprediksi dibuat dari satu atau lebih frame sebelumnya atau yang akan datang yang dikenal sebagai *reference frames* (referensi bingkai). Keakuratan prediksi biasanya dapat ditingkatkan dengan mengkompensasi gerakan antara referensi frame (satu atau lebih referensi) dan frame saat ini. [5].

2.4.1.2 Spatial model: intra prediction

Prediksi untuk blok sampel gambar saat ini dibuat dari sampel sebelumnya yang telah dikodekan dalam bingkai yang sama. Gambar 2.5 menunjukkan blok yang akan diprediksi, blok terkini dan blok yang belum belum dikode.



Gambar 2.5 Blok Sampel Pada *Intra Prediction*

Dengan asumsi bahwa blok sampel gambar dikodekan dalam urutan raster-scan, yang tidak selalu terjadi, blok bagian atas tersedia untuk prediksi intra. Blok-blok ini telah dikodekan dan ditempatkan di bitstream output. Ketika decoder memproses blok saat ini, blok bagian atas telah di decode dan selanjutnya dapat digunakan untuk menciptakan ulang (*re-create*) prediksi. [5].

2.4.2 Encoder entropi

Encoder entropi mengkonversi serangkaian simbol yang merepresentasi suatu elemen dari urutan video ke dalam bitstream terkompresi yang cocok untuk transmisi atau penyimpanan. Simbol masukan dapat mencakup koefisien transformasi terkuantifikasi, run-level atau zerotree yang, vektor gerakan dengan integer atau resolusi sub-pixel, kode penanda yang menunjukkan titik sinkronisasi ulang dalam urutan, header macroblock, header gambar, urutan header dll dan informasi tambahan, informasi sisi yang tidak penting untuk proses decoding yang tepat. [5].

2.4.3 FFmpeg

FFmpeg (*Fast Forward Motion Pictures Expert Group*) 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 mltimedia 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.[6].

2.5 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). Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila:

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

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. [7].