STUDI MENGENAI PENYEMBUNYIAN DATA UNTUK PENGKOREKSIAN ERROR PADA DATA VIDEO

Eriek Rahman Syah Putra – NIM : 13503032

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13032@students.if.itb.ac.id

Abstrak

Transmisi data video melalui saluran nirkabel yang rentan terhadap *noisy* memungkinkan terjadinya *error* pada data video, di mana dapat menurunkan kualitas gambar dan membuat proses penyembunyian *error* menjadi hal yang sia-sia. Terdapat beberapa teknik penyembunyian *error* dengan membuat estimasi bagian yang hilang dari data video yang ada. Untuk memecahkan masalah ini digunakan penyembunyian data yang dirasakan dapat menjadi alternatif dalam memprediksi data yang hilang dan memberikan informasi cadangan mengenai video kepada *receiver* selama tidak mengubah sintaks *bit-stream* yang ditransmisikan. Dengan demikian teknik seperti ini dapat memperbaiki kualitas gambar tanpa penggunaan saluran ekstra yang signifikan.

Data yang disembunyikan merupakan informasi yang tidak kelihatan dan ter-*embed* untuk mendeteksi, resinkronisasi, dan rekonstruksi *error* dan data yang hilang. Data yang disembunyikan dimasukkan ke dalam data-data video yang bersangkutan pada bagian *encoder*, dan ekstraksi dilakukan dari video pada bagian *decoder* untuk digunakan dalam penyembunyian *error*. *Error-error* yang terdapat pada *frame inter-coded* pada dasarnya dipulihkan dengan cara menyembunyikan informasi vektor gerakan (*motion*) bersamaan dengan *checksum* ke dalam frame-frame selanjutnya. Teknik ini mencoba untuk mengembalikan data yang hilang baik dengan interaksi antara *encoder* dan *decoder*, dengan operasi *post-processing* pada *decoder*, atau dengan mengurangi beberapa redundansi yang signifikan pada bagian *encoder* untuk meminimalisasi rekonstruksi *error*.

***Kata kunci**: Penyembunyian Data, Error Concealment, Sinkronisasi, Pendeteksian Error, H.263+*

1. Pendahuluan

Sistem nirkabel pada generasi mendatang menjanjikan akan adanya *bit rate* yang lebih cepat lagi di mana dapat mengakomodasi pentransmisian video melalui perangkat nirkabel. Tetapi, transmisi nirkabel selalu dipengaruhi beberapa faktor, terutama oleh gangguan keadaan sekeliling berupa situasi atmosferik pada saat itu atau mungkin adanya interferensi dengan sistem elektronik yang lain. Selain itu, pentransmisian sinyal video melalui saluran nirkabel dapat menyebabkan beberapa *error* yang tidak dapat dihindari, di mana secara langsung dapat menurunkan kualitas visual. Oleh karena itu, untuk mengatasi *error-error* yang timbul, dibutuhkan suatu metode atau teknik penyembunyian *error*.

Teknik penyembunyian *error* ini mencoba untuk me*recover* data yang hilang melalui interaksi antara *encoder* dan *decoder*, sebagai sebuah sinyal *re-send*, atau operasi *post-processing* pada bagian *decoder*, atau mengurangi beberapa data redundan pada bagian *encoder* untuk meminimalisir perbaikan error.

Pada teknik ini, *encoder* dan *decoder* saling bekerja sama, jika terdapat saluran penghubung dari *decoder* ke *encoder*. Berdasarkan informasi *feedback*, parameter pengkodean, jumlah bit *Forward Error Correction* (*FEC*), dan *bandwith* retransmisi dapat diubah. Akan tetapi, retransmisi menimbulkan keterlambatan pendekodean, di mana sama sekali tidak diinginkan terjadi di dalam beberapa sistem yang *real-time*. Teknik penyembunyian *error post-processing* menggunakan korelasi antara blok yang rusak dan blok-blok tetangganya pada *frame* yang sama dan *frame* sebelumnya. Teknik ini didasarkan pada perbedaan yang kecil dari nilai intensitas secara spasial. Tetapi pada daerah yang memiliki sisi yang tajam,

rekonstruksi macam ini tidak dapat diaplikasikan dengan operasi *post-processing*.

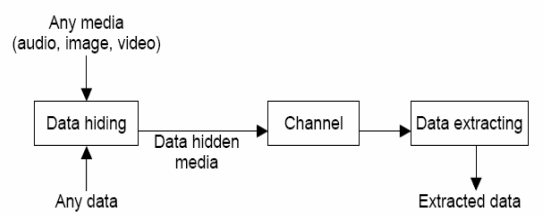
Teknik penyembunyian *error* pada kelompok ketiga menggunakan data redundan pada *bit stream*, yang ditambahkan pada sisi *encoder* setelah pengkodean *source*. *Source* video dapat dikodekan di dalam *layer* atau di dalam *multiple description* selama pengkodean *source* dan beberapa nilai FEC dapat diaplikasikan. Kelemahan utama dari metode ini yaitu meningkatkan *overhead* transmisi.

Semua pendekatan di atas dapat digabung secara komprehensif dengan menyembunyikan beberapa informasi yang *imperceptible* yang akan berguna pada saat penyembunyian *error*. Selama pengkodean *source*, beberapa informasi mengenai video dapat di-*embed* ke dalam bagian-bagian tertentu dari video itu sendiri dan *decoder* dapat menggunakan informasi tersembunyi ini dalam penymbunyian *error*. Dengan cara ini, informasi yang tersembunyi tidak hanya ditransmisikan melalui saluran rahasia dari *encoder* ke *decoder* yaitu dengan mengirim kembali beberapa informasi yang hilang, tetapi juga mengurangi beban pada *post-processing*.

Penyembunyian data ke dalam sebuah video dapat membuat degradasi kualitas visual dan menyebabkan peningkatan yang relatif kecil pada *coding bit rate*. Pendekatan radikal ini, yang menggunakan data tersembunyi dalam menyembunyikan *error*, adalah hasil dari proses steganografi, sebuah teknik baru untuk membuat modifikasi yang tidak tampak pada sebuah media, yang kebanyakan digunakan untuk perlindungan *copyright* dan aplikasi *security-based* lainnya. Hal ini ditegaskan bahwa informasi yang disembunyikan dapat ditransmisikan tanpa sebuah *overhead bit-rate* yang signifikan dari standar kompresi yang digunakan. *Receiver* standar yang tidak pernah tahu akan informasi yang disembunyikan itu tidak akan pernah mendapat pengaruh dan mendekodekan *bit-stream* tanpa masalah. Tanpa menggunakan pendekatan penyembunyian data ini, pentransmisian informasi yang ekivalen di dalam *bit-stream* membutuhkan *bit rate* yang ekstra, sama dengan beberapa modifikasi di semua *receiver*.

2. Teknik Penyembunyian Data Dasar

Kata "steganografi", yang merupakan nama umum untuk teknik data hiding (penyembunyian data), adalah sebuah teknik dalam membuat modifikasi yang *imperceptible* pada sebuah media apa pun seperti media tekstual, audio, gambar, dan video. Perkembangan teknologi dalam multimedia dijital telah membuat permasalahan penyembunyian informasi semakin populer. Skema umum dari steganografi ditunjukkan pada gambar di bawah. Data jenis apa pun dapat dimasukkan ke dalam sebuah media dengan teknik steganografi. Media yang dimasukkan data tersebut kemudian ditransmisikan ke seorang penerima pesan melalui saluran tertentu. Selama transmisi, media yang bersangkutan mungkin saja mengalami beberapa kerusakan karena adanya *noise* di sekitar saluran transmisi. Oleh karena itu, dibutuhkan sebuah teknik steganografi supaya data yang disembunyikan tetap eksis.



Gambar 1. Skema umum steganografi

Teknik steganografi mempunyai 4 macam *constraint* yang fundamental yaitu *imperceptibility* (kemampuan untuk tidak dapat dilihat), *capacity*, *robustness*, dan *security*. Hal ini tidaklah mungkin untuk memenuhi semua *constraint* tersebut dalam satu teknik. Tetapi, berdasarkan pada kebutuhan beberapa aplikasi secara khusus, dapat dibangun sebuah *trade-off* untuk membangun teknik steganografi yang memuaskan.

Imperceptibility mengacu pada perbedaan perceptual antara data original dan data yang telah dibubuhi data tambahan (*marked*). Pemberi pesan rahasia menginginkan supaya pemilik sebuah gambar tidak akan sadar mengenai degradasi visual yang ada pada gambar miliknya karena adanya penyembunyian data rahasia yang ditujukan kepada orang lain. Oleh karena itu, data yang disembunyikan seharusnya menghasilkan distorsi yang minimum pada data atau media inangnya.

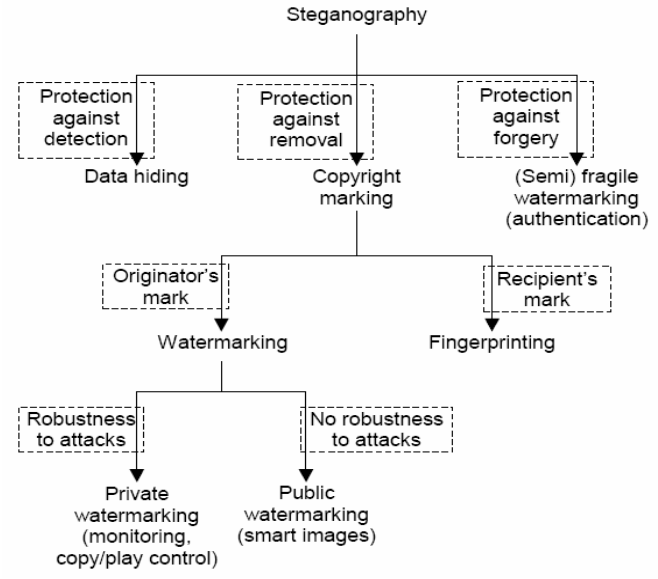
Capacity adalah property lain di dalam steganografi. Jumlah data yang sengaja dimasukkan ke dalam media seharusnya dapat semaksimum mungkin. Tetapi, dengan menambah sejumlah data yang dimasukkan ke dalam media akan membuat data yang

disembunyikan makin terlihat dan membuat degradasi pada kualitas visual media. Oleh karena itu, *imperceptible* seharusnya juga dipertimbangkan ketika kapasitas sistem diuji coba.

Robustness pada teknik steganografi pastinya sangat diharapkan ketika informasi yang disembunyikan dapat melewati beberapa operasi pemrosesan sinyal seperti *filtering* dan kompresi, atau beberapa distorsi geometrikal seperti rotasi, translasi, dilatasi, dan lain sebagainya. Jika data yang disembunyikan masih bisa dideteksi setelah dilakukannya operasi dan distorsi tersebut, maka sistem bisa dinyatakan *robust*. Namun, teknik ini tidak serta merta bertahan pada semua operasi.

Sisi *security* dari teknik steganografi diacu sebagai resistansi terhadap serangan musuh. Pendeteksian informasi yang tersembunyi pada media seharusnya tidak memungkinkan. Serangan ini mencoba untuk mengubah data yang disembunyikan dari media inangnya. Jenis lain dari serangan meliputi pendeteksian tanda tersembunyi atau tanda lain yang sengaja disembunyikan pada *cover* media. Hal ini seharusnya patut diingat bahwa serangan ini didesain untuk menjaga kualitas visual dari media berada pada level yang masih diterima. Namun, sama halnya dengan *robustness*, kriteria *security* juga *dependent* dalam aplikasinya.

Berdasarkan pada kriteria yang dibahas di atas, maka didapatkan sebuah klasifikasi teknik steganografi seperti pada gambar di bawah. Steganografi dapat dibagi menjadi 3 kelompok: penyembunyian data (*data hiding*), *copyright marking*, dan *semifragile watermarking.* Penyembunyian data pada pokoknya digunakan untuk penyembunyian *error* dengan mentransportasikan beberapa data *error* yang disembunyikan dari *encoder* ke *decoder*. Hal ini tidak sepenuhnya *robust* tetapi merupakan teknik steganografi yang aman, yang memberikan perlindungan dari pendeteksian dini.



Gambar 2. Jenis teknik steganografi

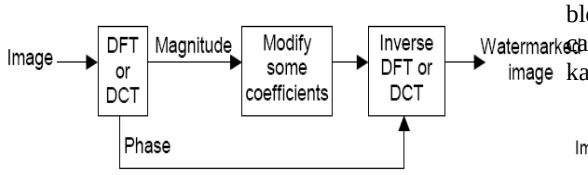
Teknik steganografi mengambil nama-nama yang berbeda sesuai dengan aplikasi dan *property*nya. Kebanyakan dari mereka didasarkan pada substitusi redundan atau bagian yang tidak penting dari sinyal. Beberapa teknik steganografi dapat dirinci sebagai modulasi *low-bit*, modulasi *spread spectrum*, modulasi *quantization index*, dan metode statistikal.

Bidang *Least Significant Bit* (LSB) dari piksel gambar disubstitusi dengan bit-bit pesan untuk modulasi *low-bit*. Penerima pesan mengekstrak bit pesan yang tersembunyi jika si penerima pesan itu tahu piksel mana saja yang mengalami perubahan. Selama gambar tersebut didistorsikan sedikit sekali di dalam proses ini, kapasitas yang dimasukkan menjadi banyak. Namun, data yang disembunyikan mudah terkena serangan. Bahkan perubahan yang kecil pada gambar akibat dari operasi pemrosesan sinyal dapat mendistorsikan data yang disembunyikan. Beberapa jenis dari teknik ini meliputi proses pengacakan urutan dari bit-bit pesan yang disembunyikan atau membagi-bagi gambar menjadi beberapa bagian dan meng-*embed* satu bit pesan ke dalam satu bagian.

Teknik *spread spectrum* meng*embed* pesan di dalam sebuah domain transformasi, seperti *Discrete Fourier Transform* (*DFT*), *Discrete Cosine Transform* (*DCT*), atau *Discrete Wavelet Transform* (*DWT*). Memodifikasi koefisien transformasi memberikan sisi *robustness* yang lebih baik dari kompresi,

cropping, atau beberapa pemrosesan gambar, daripada substitusi LSB yang dibutuhkan pada *watermarking*.

Sebagai contoh, sebuah teknik *spread spectrum* meng-*embed* sebuah *watermark* ke dalam kefisien domain frekuensi dari sebuah gambar sedemikian sehingga beberapa koefisien diubah dengan menambahkannya dengan beberapa angka acak antara 0 dan 1, seperti yang diilustrasikan pada gambar di bawah. Secara visual, koefisien yang signifikan dipilih untuk dimodifikasi, yang membuat *watermark* menjadi lebih *robust* terhadap serangan pada saat gambar yang ber*watermark* secara *perceptual* masih sama dengan aslinya. *Watermark* yang di-*embed* dengan teknik ini bertahan pada operasi perbesaran, distorsi, *cropping*, *print-scan process*, serangan yang berturut-turut, dan serangan *collusion*.



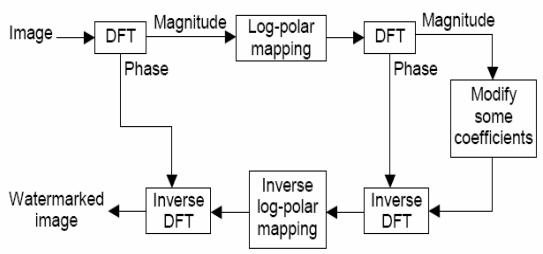
Gambar 3. Watermarking spread-spectrum

Sebuah *trade-off* eksis di antara sisi *robustness* dan *imperceptibility* di dalam teknik domain transformasi. Bila sejumlah data yang disembunyikan bertambah, *watermark* akan menjadi lebih *robust* akan tetapi dapat kembali terlihat oleh orang lain. Untuk meng-*embed* data sebanyak mungkin tanpa menurunkan *imperceptibility*, dikembangkanlah *Human Visual System* (*HVS*) yang berdasarkan pada metode *watemarking*. *Watermark* dimasukkan ke dalam koefisien DWT dari sebuah gambar sesuai dengan batas ambang yang ditentukan dari beberapa pengujian. Pada teknik *watermarking* yang lain, transformasi Fourier-Mellin diaplikasikan pada media gambar dan watermark di-*embed* pada domain ini seperti yang ditunjukkan pada gambar di bawah untuk memberikan sebuah *watermarking invariant* rotasi, perbesaran, dan translasi. Besar DFT tidak dipengaruhi dari translasi sehingga domain DFT memberikan *robustness* pada translasi.

Pada awalnya, efek translasi dieliminasi dengan menggunakan besar DFT dan kemudian koordinat Cartesian diubah menjadi koordinat Log-polar dalam rangka untuk mengubah efek rotasi dan skala pada translasi, dan yang terakhir yaitu dengan menggunakan kembali besar DFT, dan semua efek ini dapat segera dieliminasi. Oleh karena itu, *watermark* hasil final ini menjadi *robust* terhadap RST.

Banyak dari standar kompresi mengkuantisasi data sumber untuk efisiensi pengkodean yang lebih baik. Langkah kuantisasi ini dapat digunakan untuk meng-*embed* data. Berdasarkan pada nilai data yang disembunyikan, *quantizer* yang berbeda digunakan di dalam mengkuantisasi data sumber. Pada bagian *receiver*, data yang disembunyikan dapat diekstrak dengan menentukan *quantizer* mana yang digunakan. Salah satu pendekatan yang paling terkenal yang menggunakan metode ini yaitu *Quantization Index Modulation* (*QIM*).

Pada steganografi statistik, beberapa karakter statistik dari sebuah gambar diubah untuk memasukkan bit data. Teknik ini mempunyai kapasitas yang kurang. Tetapi sebuah gambar dapat dibagi menjadi blok-blok dan setiap blok, *property* statistik dimodifikasi dalam cara yang berbeda untuk meningkatkan kapasitas.



Gambar 4. Watermarking invariant rotasi, perbesaran, dan translasi

Pada makalah ini, *signaling* "genap-ganjil" koefisien DCT digunakan untuk menyembunyikan data. LSB dari koefisien DCT disubstitusi dengan bit-bit data yang di-*embed*. Selama data disisipkan di dalan domain yang terkompresi, metode yang dipilih akan menjadi optimal, dalam arti lain yaitu metode ini tidak membutuhkan sisi robustness apa pun dan memberikan performansi yang terbaik di dalam hal kapasitas yang bisa dimanfaatkan. Akan tetapi, bila standar pentransmisian sebelumnya tidak ditentukan, informasi ini harus dimasukkan ke dalam intensitas gambar dalam keadaan yang lebih *robust*. Pada kasus tersebut, jumlah bit untuk menyembunyikan data akan menjadi berkurang, dan dengan demikian, jumlah total bit yang dibutuhkan dapat menjadi faktor yang kritis.

3. Teknik Penyembunyian Error pada bagian *Decoder*

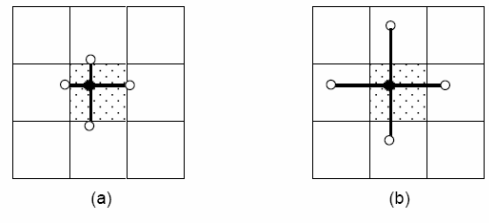
Teknik penyembunyian error dibagi menjadi tiga kategori yaitu dengan *re-sending*, *post-processing*, dan penambahan redundansi. Tetapi pada bagian ini akan dibahas mengenai penyembunyian *error* pada *decoder* seperti operasi *post-processing* untuk mengembalikan informasi yang hilang pada bagian *receiver*.

Standar pengkodean populer seperti MPEG, ITU-T, dsb. mengkompresi *frame-frame* pada video *digital* dengan dua cara, yaitu secara intra dan inter. Pengkodean intra sangat mirip dengan pengkodean fragmen gambar seperti JPEG yang menggunakan hubungan spasial antar nilai piksel pada gambar. Sedangkan pengkodean inter, menggunakan hubungan temporal antar *frame-frame* video yang digunakan, di samping hubungan spasial. Hampir semua teknik penyembunyian error berusaha untuk memperbaiki bagian dari gambar atau video yang rusak, karena pada umumnya menggunakan standar pengkodean yang *block-based* dan *error* bit yang mungkin terjadi biasanya menyebabkan *error* pada blok atau bagian dari blok.

3.1 Penyembunyian Error Intra-frame

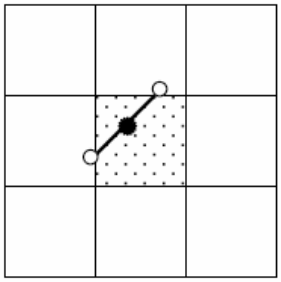
Intensitas pada sebuah gambar, dari satu piksel ke tetangga-tetangganya, memiliki deviasi atau perbedaan yang tidak mencolok menjadi suatu gambar keseluruhan, kecuali pada daerah yang memiliki sisi yang tajam. Teknik ini, yang merekonstruksi blok fragmen gambar atau rangkaian video *intra-frame* yang hilang, menggunakan *property smoothness* di dalam sebuah gambar dan data tetangga secara spasial. Teknik yang paling dasar dan paling populer dikenal sebagai interpolasi bilinier dalam domain spasial. Tiap-tiap piksel di dalam blok yang rusak diinterpolasi menggunakan empat piksel terdekat yang mengelilinginya dalam empat arah. Bobot interpolasi dihitung berdasarkan jarak antara piksel yang rusak dan piksel yang benar. Semakin kecil jaraknya, semakin besar bobotnya (Gambar a).

Interpolasi pada domain frekuensi juga bisa menggunakan dengan cara yang sama. Tiap koefisien DCT dari blok yang rusak diinterpolasi dari koefisien empat blok tetangga yang berhubungan dengan bobot interpolasinya semua sama (Gambar b)



Gambar 5. Interpolasi blok yang rusak dalam domain (a) spasial dan, (b) frekuensi

Oleh karena beberapa bagian dari gambar mengandung perbedaan yang tinggi, interpolasi bilinier yang sederhana tidak dapat memenuhi hasil pada semua kasus. Blok-blok dapat memiliki struktur geometri yang kompleks. Beberapa teknik dicoba untuk memperkirakan struktur geometri lokal di sekeliling blok yang hilang atau rusak. Skema interpolasi spatial directional diajukan sebagai cara interpolasi yang lebih baik. Piksel yang berhubungan direkonstruksi melalui interpolasi dua piksel terdekat dari blok tetangga yang dengan benar diterima dalam arah yang diprediksi. Dua lapisan piksel terdekat di sekeliling blok yang hilang/rusak digunakan untuk mengestimasi *edge passing* melalui blok yang hilang/rusak tersebut dan interpolasi dilakukan sepanjang arah sisi (Gambar c). Pada kasus ini, bobot interpolasi sebanding dengan jarak dari piksel yang rusak dengan yang benar (eksis/tidak rusak). Pada pendekatan yang sama, transformasi Hough digunakan untuk menentukan arah terbaik untuk interpolasi. Pada pendekatan ini, diasumsikan bahwa struktur visual di sekeliling blok yang hilang/rusak juga direpresentasikan di dalamnya dengan cara yang sama.



Gambar 6. Interpolasi di sepanjang arah domain spasial

3.1.1 Penggunaan Penyembunyian Data untuk Penyembunyian Error

Metode yang paling akhir dilakukan mencoba untuk mengestimasi blok yang hilang dari data-data yang diterima secara benar. Pokok pikiran dari pendekatan ini adalah untuk menyisipkan beberapa data tersembunyi yang berhubungan dengan sebuah blok di dalam tetangga-tetangganya, sehingga informasinya dapat digunakan di dalam *receiver* untuk penyembunyian *error* yang lebih baik atau *robust*. Selain itu, metode ini juga memberikan *data forehand* untuk merekonstruksi blok yang rusak atau hilang dan meningkatkan kualitas rekonstruksi.

Di dalam metode ini, sisi dari blok yang hilang diestimasikan (dihampirkan) untuk interpolasi sepanjang arah sisi. Arah sisi setiap blok ditentukan pada *encoder* dan disembunyikan ke dalam blok tetangganya. Oleh karena itu, tidak diperlukan untuk mengestimasi sisi dari blok yang hilang pada decoder, karena informasi sisi pada blok yang hilang dikirim ke *decoder* dengan menggunakan saluran tersembunyi di dalam gambar. Jelas saja, informasi utama mengenai arah sisi ini menghasilkan kualitas rekonstruksi yang lebih memuaskan.

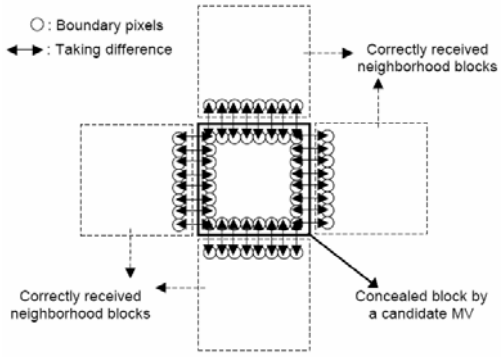
Di samping menginterpolasi koefisien domain frekuensi dari blok *receiver*, pada pendekatan yang sama dalam menyembunyikan arah sisi, koefisien dengan kualitas rendah tiap-tiap blok dapat juga di-*embed* ke dalam blok tetangga lainnya. Tetapi pendekatan ini membutuhkan sejumlah bit untuk menyembunyikan koefisien sebagai usaha perbaikan (penemuan kembali data yang hilang), bahkan kuantisasi secara kasar juga diaplikasikan pada koefisien untuk disembunyikan. Dengan demikian, kualitas visual video dikurangi sedapat mungkin untuk menyembunyikan data sebelum adanya proses pengiriman video, ketika mencoba untuk merekonstruksi ulang blok-blok yang hilang dari koefisien-koefisien yang tersembunyi setelah melewatkan video melalui saluran yang *noise*.

Kehilangan sinkronisasi adalah masalah utama lain dalam kasus *bit-error* pada saluran *noisy*. *Decoder* tidak dapat mengetahui error yang terjadi dan melanjutkan mendecode bit stream di mana membuat *error* memperbanyak diri ke blok-blok tetangganya. Menyembunyikan sebuah data resinkronisasi, seperti panjang bit blok dari blok tetangganya, yang menggunakan beberapa metode penyembunyian data dapat menjadi pendekatan lain dalam masalah penyembunyian *error*. Men*decode* bit dengan jumlah sedikit atau lebih banyak untuk sebuah blok dihindari di dalam cara ini.

3.2 Penyembunyian Error Inter-frame

Pada kasus transmisi video, perbaikan blok-blok yang hilang adalah masalah yang sederhana dibandingkan dengan transmisi gambar, karena hubungan temporal dapat juga digunakan sebagai tambahan dari hubungan spasial. Teknik pada kategori ini menggunakan hubungan temporal sebagus dengan *property* spasial *smoothness* untuk mendapatkan hasil yang lebih baik. Dengan semakin tingginya nilai *smoothness* antar *frame* video, semakin lebih baik penyembunyian *error*nya.

Pengkodean *inter-frame* berdasarkan pada penemuan informasi *motion* dari tiap blok antar frame dan mengkodekan *motion vector* (MV) untuk tiap blok. Intinya, pendekatan penyembunyian *error* pada kategori ini mencoba untuk me*recover* informasi *motion* dari blok yang rusak melalui blok tetangganya. Untuk mengestimasi MV yang hilang, terdapat beberapa pendekatan seperti mengasumsikan dengan nilai nol untuk MV yang hilang, atau menempatkan MV dari blok yang bersangkutan pada lokasi yang sama di *frame* sebelumnya dari MV yang hilang tersebut, atau dengan menggunakan nilai rata-rata/ median MV dari blok yang berdekatan secara spasial sebagai suatu estimasi MV yang hilang. Sesuai dengan metode di atas, terdapat lebih dari satu kandidat untuk menggantikan MV yang hilang. Pada kasus ini, beberapa macam mekanisme pemilihan dibutuhkan untuk rekonstruksi yang terbaik. *Boundary Matching Algorithm* (*BMA*) dapat memberikan sebuah pengukuran dalam memilih MV dari himpunan MV kandidat. Pada metode BMA, sebuah MV dipilih bila menghasilkan distorsi minimum, seperti yang digambarkan pada gambar di bawah



Gambar 7. Perhitungan *side match distortion* pada BMA

Distorsi tersebut ditentukan sebagai penjumlahan dari nilai absolut diferensiasi antara intensitas tetangga terdekat dengan perbatasan blok yang disembunyikan dan blok-blok tetangganya.

BMA digunakan untuk rekonstruksi blok yang hilang dengan berbagai bentuk. BMA sendiri juga digunakan untuk menggantikan blok yang hilang dengan pola yang sesuai melalui proses pencarian pada *frame-frame* sebelumnya. Kemudian, proses *warping* (pembengkokan) dilakukan untuk bisa menyesuaikan blok yang bersangkutan dengan tempat di sekelilingnya dan untuk mengurangi artifak-artifak yang dihasilkan dari pergerakan, rotasi, atau deformasi yang cepat. Dengan teknik ini, perubahan lebih dari satu blok secara berturut-turut bisa dikatakan tidak wajar. Untuk mencegah efek *blocky* ini, diimplementasikan algoritma penyembunyian error iteratif berdasarkan BMA. Pada mulanya, BMA konvensional diterapkan pada blok yang rusak pada satu bagian. Bila tidak ada tetangga kiri dari blok yang bersangkutan, rekonstruksi tidak mungkin dapat memenuhi. Kemudian, BMA diimplementasikan lagi, tetapi untuk kasus ini, tetangga kanan dari blok yang rusak penuh dengan blok-blok rekonstruksi dari tahap pertama. Iterasi dihentikan sampai total error yang terjadi seminim mungkin.

Pada pendekatan lain menggunakan tetangga atas dan bawahnya untuk merekonstruksi blok yang hilang. Daerah atas dan bawah dari blok tersebut dicari pada *frame-frame* sebelumnya untuk mendapatkan blok yang pas (sesuai). Pada awalnya, setengah bagian atas dari blok yang rusak direkonstruksi setelah pencarian di dalam frame-frame sebelumnya untuk menyesuaikan pola dari bagian atas blok tetangga. Selanjutnya, setengah blok yang rusak yang lain direkonstruksi dengan cara yang sama, yaitu juga mempertimbangkan setengah bagian atas yang direkonstruksi pada tahap pertama.

Di samping menggunakan blok untuk proses pencarian pola yang pas pada frame sebelumnya, piksel-piksel dari dua garis dipilih di sekeliling blok yang hilang. Algoritma yang diterapkan menggunakan beberapa bobot selama pencarian sesuai dengan jenis data tetangga yang ada, di mana terdapat kemungkinan bisa hilang, disembunyikan, atau diterima secara benar. Pendekatan yang lain untuk mengukur MV dari blok yang hilang yaitu dengan menggunakan *motion field interpolation* (*MFI*) yang mana MV-nya ditentukan untuk semua *point* di dalam blok yang hilang. MV yang hilang didapatkan dengan cara interpolasi dari informasi *motion* yang ada pada sejumlah *control points*.

Pada tiap piksel dari blok yang rusak, kandidat MV ditemukan dengan menggunakan MFI bilinier yang menyertakan empat tetangga MV. Kandidat MV yang lain ditemukan dengan menggunakan BMA. Dua himpunan vektor yang dihasilkan dari tiap metode ini dikombinasikan baik dengan merata-ratakan atau dengan memberi bobot untuk menempatkan MV pada blok yang hilang, hingga akhirnya blok yang hilang tersebut dapat dikembalikan / diperbaiki.

Bila teknik MFI untuk penyembunyian *error* diaplikasikan pada *codec multi-reference* yang menggunakan lebih dari satu frame referensi untuk estimasi dan kompensasi *motion*, maka paling banyak rekonstruksi empat kandidat dapat diperoleh dari empat MV tetangga yang berhubungan. Keempat kandidat dipilih oleh BMA atau dengan merata-ratakan bobotnya.

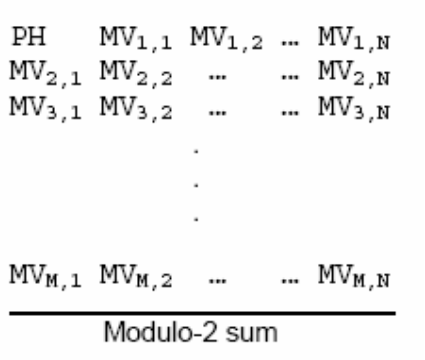
Video tidak selalu mengandung *motion* yang *translational*. *Motion* yang kompleks seperti gerak rotasi, perbesaran, atau pengurangan, seharusnya juga dipertimbangkan pada saat mengestimasi motion yang sebenarnya. Transformasi *affine*, yang merupakan sebuah transformasi koordinat, dapat digunakan untuk memodelkan motion yang kompleks ini. Transformasi ini diaplikasikan untuk mengestimasi parameter *motion* dari blok yang hilang menggunakan data blok tetangga yang diterima secara benar.

3.2.1 Penggunaan Penyembunyian Data untuk Penyembunyian Error

Menyembunyikan informasi dalam rangka menyembunyikan *error* dapat juga diaplikasikan pada *inter-frame* video yang terkompresi. *Frame-frame inter-coded* dapat dimodifikasi untuk menjadi lebih *robust* dengan pertolongan data yang sengaja disembunyikan. Oleh karena MV sangat penting di dalam pengkodean *inter-frame*, secara umum informasi MV disembunyikan untuk mengurangi *error* ppada *inter-frame*. Terdapat sejumlah metode di dalam beberapa literatur yang dikhususkan untuk menangani penyembunyian *error* di dalam *inter-frame*.

Picture Header (*PH*) dan MV dari sebuah *frame* dilindungi oleh beberapa *parity bit*. MV yang dikodekan untuk setiap blok disusun baris per baris seperti yang terlihat pada

gambar di bawah. Setelah itu, blok-blok tersebut dikenakan operasi modulo-2 dan *parity bit* yang dihasilkan disembunyikan ke dalam frame berikutnya. Data di-*embed* dengan memodifikasi vektor *motion* dari *frame* berikutnya sebesar setengah piksel, misalnya dalam kasus data di-*embed* ke dalam MV.



Gambar 8. Mendapatkan parity bit

Dengan mempertimbangkan paket-paket yang hilang misalnya karena adanya *error* yang berlebihan, pendekatan di atas diperluas ke arah frame dengan menghitung *parity bit frame* per *frame* terhadap sebuah kelompok *frame* dan menyembunyikan *parity bit* ke dalam *frame* yang mengikuti kelompok *frame* tersebut. Tetapi, pada pendekatan ini, data di-*embed* ke dalam koefisien DCT yang terkompensasi *motion*. Sebagai tambahan, skema *block-shuffling* dapat dipakai untuk mengisolasi blok-blok yang salah sedemikian sehingga blok-blok yang ditransmisikan secara benar dapat mengelilingi blok-blok yang rusak.

Bersamaan dengan algoritma penyembunyian error ini, beberapa skema pendeteksian *error* yang bertumpu pada penyembunyian data, juga dapat diaplikasikan untuk kepentingan pentransmisian video. salah satu metode yang dipakai, menyembunyikan kode *parity check* satu frame MB ke dalam MV dan koefisien DCT residual pada *frame* selanjutnya. Akhir-akhir ini, sebuah algoritma baru memodifikasi koefisien DCT untuk mendeteksi *error* di dalam *bit-stream*. Pada metode ini, koefisien-koefisien yang selokasi dengan indeks genap dalam keadaan *reordering* yang zig zag dipaksa supaya bernilai genap, begitu juga sebaliknya.

4. Implementasi Sistem Terbaru dalam Transmisi Video yang Handal

Di dalam implementasi ini, *frame-frame* dari pengkodean intra dan inter dianggap terpisah di dalam sudut pandang penyembunyian *error* ini. Untuk menyembunyikan beberapa *error* di dalam *intra-coded frame*, maka digunakan data arah sisi dari MB dan nilai panjang bit MB yang dikodekan. Di sisi lain, bit-bit vektor motion yang dikodekan juga digunakan dalam penyembunyian *error* di dalam *inter-coded frame*. Semua data-data tersebut di-*embed* ke dalam video pada bagian *encoder* kemudian diekstraksi pada bagian *decoder*nya sebagai sebuah data penunjang dalam mengurangi *error*. Penyembunyian data dilakukan dengan signaling "genap-ganjil" sederhana pada koefisien DCT. Untuk menyembunyikan "0" ke dalam koefisien, bit tersebut dipaksakan untuk menjadi ganjil, sedangkan untuk menyembunyikan "1" dipaksakan supaya menjadi ganjil. Data tersebut di-*embed* ke dalam bidang LSB dari koefisien DCT dengan cara demikian. Hanya koefisien-koefisien yang tidak nol yang dimodifikasi sehingga kecepatan pengkodean panjangnya tidak bergerak meningkat. Jika semua LSB dari koefisien yang tidak nol dialokasikan untuk penyembunyian data dan masih terdapat data yang di-*embed*, maka bidang LSB yang lebih tinggi (kedua, ketiga, atau keempat) dipakai. Jika keempat bidang LSB tidak cukup untuk menyembunyikan data maka data tidak di-*embed* ke dalam blok tersebut.

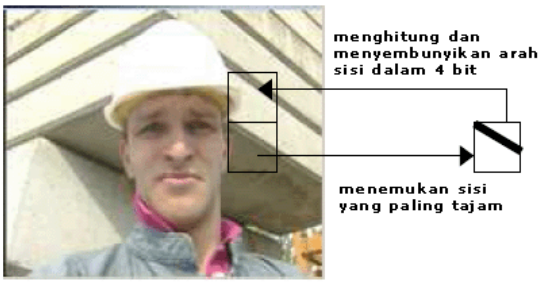
4.1 Penyembunyian Error Intra-frame

Untuk memperoleh sebuah *recovery error* yang sukses, lokasi yang pasti dari error tersebut contohnya blok yang rusak, seharusnya dideteksi sebagai langkah pertama yang harus dilakukan. Setelah mendeteksi blok yang rusak, proses sinkronisasi harus dibangun ulang supaya mencegah penyebaran error ke blok yang lain. Langkah terakhir yaitu rekonstruksi intensitas blok yang rusak untuk recovery error akhir. Oleh karena itu, tiga pendekatan pokok dalam *recovery error* yang sukses yaitu pendeteksian error, resinkronisasi, dan rekonstruksi dari blok yang rusak.

Selain itu, informasi mengenai arah sisi dan data *bit-length* keduanya dibutuhkan juga untuk proses *recovery frame intra-coded* pada saat ketiga pendekatan di atas dilakukan. Pada saat nilai *bit-length* dibutuhkan untuk sinkronisasi, informasi arah sisi harus sesuai untuk rekonstruksi blok yang rusak. Akhirnya, dua data ini dapat digunakan bersama-sama

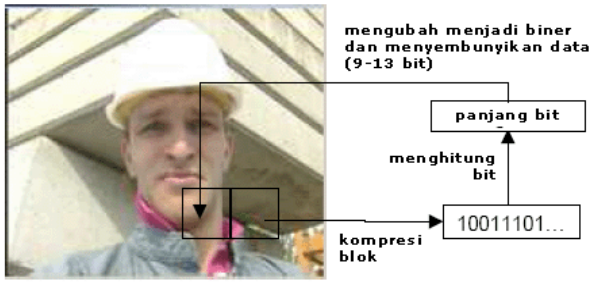
untuk mendeteksi *error* pada bit. Namun, rekonstruksi semua blok yang rusak tidak selalu memberikan hasil yang menjanjikan. di mana dapat menyebabkan penyembunyian data yang berlebihan atau pada data yang tidak seharusnya. Untuk mengantisipasi terjadinya kondisi yang demikian, sebuah *overconcealment parity* berukuran 2-bit yang didapatkan dari koefisien DCT blok, digunakan untuk mengiringi informasi arah sisi. Tetapi, data yang disembunyikan tidak dapat mendeteksi semua *error* pada bit. Untuk memberikan kapabilitas pendeteksian yang handal, *single parity bit* digunakan dalam sinkronisasi data.

Dalam meng-*embed* arah sisi, blok pertama kali diklasifikasikan sebagai sebuah blok sisi dengan mengaplikasikan algoritma pendeteksian sisi. Pada tiap piksel di dalam blok, besar vektor gradien dan sudut vektor gradiennya dihitung dengan menggunakan operator gradien Robert. Sudut piksel yang besar gradiennya di atas ambang, dikuantisasi ke dalam 16 arah yang jaraknya sama dan besar gradien yang arahnya sama dijumlahkan. Arah yang jumlah besar gradiennya paling besar diseleksi sebagai arah sisi *single final* dari blok keseluruhan. Yang pasti, sebuah bit pesan single seharusnya juga disembunyikan untuk menyatakan jenis dari blok, misalnya sebuah sisi atau blok. Oleh karena itu, pendekatan ini membutuhkan hanya 5 bit per blok untuk meng-*embed* informasi arah sisi ke dalam koefisien DCT.



Gambar 9. Menyembunyikan data arah sisi untuk penyembunyian error intra-frame

Untuk menyembunyikan data panjang bit, jumlah bit yang dibutuhkan untuk mengkodekan blok ditentukan dan nilai ini di-*embed* ke dalam koefisien DCT dari blok sebelumnya setelah dikonversi ke dalam representasi biner selama mengkodekan. Jumlah bit yang digunakan dalam representasi ini seharusnya dihitung sebelumnya dengan mempertimbangkan panjang bit maksimum blok. Metode ini membutuhkan 9 sampai 13 bit yang bergantung pada *bit rate* dari video *encoder* yang digunakan. Dengan melihat pada parameter kuantisasi, *decoder* dapat menentukan jumlah bit yang digunakan untuk data panjang bit.



Gambar 10. Menyembunyikan nilai panjang bit untuk penyembunyian error intra-frame

4.1.1 Pendeteksian error

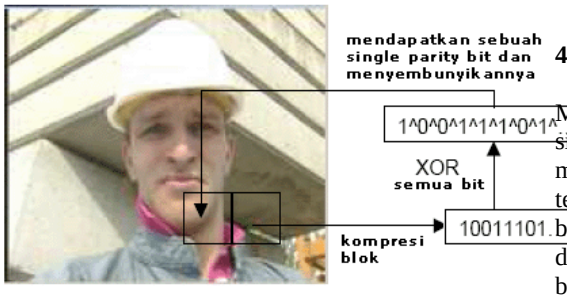
Informasi sisi arah dan data panjang bit keduanya digunakan untuk pendeteksian *error*. *Decoder* H.263+ sendiri dapat juga mendeteksi adanya *error* selama dekompresi MB, jika menemui sebuah *codeword* yang tidak sesuai dengan *entry* apa pun di dalam tabel Huffman. Bila *decoder* selesai men*decode* MB tanpa pendeteksian *error*, maka total jumlah bit yang dibaca dari *bit-stream* untuk pendekodean blok tersebut dan nilai panjang bit yang disembunyikan pada blok sebelumnya dibandingkan.

Decoder H.263+ tidak dapat mendeteksi semua error, Biasanya *decoder* ini tidak mendeteksi *error* pada *codeword* dan membandingkan *codeword* dengan *entry* yang salah pada tabel Huffman. Di dalam beberapa kondisi, beberapa bit didekodekan untuk merekonstruksi MB, dan dalam beberapa kondisi juga, *decoder* memperbanyak diri ke blok selanjutnya tanpa antisipasi awal. Supaya *decoder* tidak memperbanyak diri ke blok selanjutnya karena sebuah *error* yang tidak dapat dideteksi, data panjang bit yang disembunyikan diperiksa secara berulang kali dengan jumlah bit dibaca dari *bit-stream*. Bila data panjang bit yang di-*embed* di dalam blok sebelumnya tidak ada, maka arah sisi blok yang didekodekan dihitung sekali lagi pada *decoder* dan dibandingkan dengan informasi arah sisi yang disembunyikan di dalam blok yang lebih tinggi sebagai cara kedua.

4.1.1 Penggunaan *parity*

Meskipun memeriksa panjang bit blok dengan niai panjang bit yang tersembunyi pada blok sebelumnya dirasa cukup untuk mendeteksi *error*. hal ini belum tentu dapat menentukan jenis error. Jenis-jenis *error* ini kemungkinan besar dapat merusak informasi tersembunyi pada blok yang bersangkutan, terlebih lagi bila kerusakan secara visual yang terjadi sangat kecil.

Dalam kasus ini, digunakanlah *single parity bit* dari *macroblck bit-stream*. *Parity bit* ini didapatkan dengan meng-XOR-kan semua bit dari *macroblock* yang dikodekan dan disembunyikan ke dalam koefisien DCT blok sebelumnya sebagai sebuah informasi tersembunyi ekstra. Bila pemeriksaan nilai panjang bit dan *parity bit* tidak memberitahukan informasi *error*, maka dapat dipastikan bahwa tidak ada *error* di dalam *macroblock* yang didekodekan dan data yang disembunyikan di dalamnya masih terjaga dengan baik. Pastinya, hal ini berdasarkan pada asumsi yang menyatakan bahwa terdapat *single bit error* di dalam *bit-stream* blok yang bersangkutan. Asumsi ini dapat ditoleransi secara praktis, tidak hanya untuk *binary simmetric*, tetapi juga untuk *fading channel*.

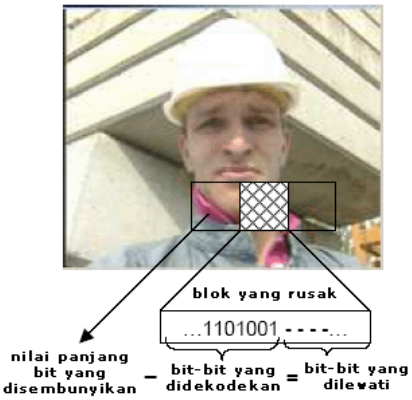


Gambar 11. Menyembunyikan single parity bit untuk penyembunyian error intra-frame

4.1.2 Resinkronisasi

Oleh karena koefisien dikodekan dalam bentuk variabel panjang, maka *error* dapat dengan mudah mengubah panjang bit MB. Situasi seperti ini menyebabkan hilangnya sinkronisasi pada *bit-stream* yang didekodekan. Kita dapat menyusun kembali *decoder*nya dengan menggunakan info panjang bit yang tersembunyi, yang mana diekstraksi dari blok sebelumnya. Setelah adanya pendeteksian *error*, untuk meresinkronisasi *decoder*, sistem tidak diperbolehkan untuk men*decode* bit lebih dari jumlah yang diperintahkan sebagai nilai yang disembunyikan di dalam blok sebelumnya.

Pada saat resinkronisasi, dipastikan bahwa jumlah bit yang didekodekan lebih sedikit atau sama dengan nilai yang disembunyikan. Perbedaan antara jumlah bit yag disembunyikan dan yang didekodekan dikalkulasi dan *decoder* men*skip* jumlah bit yang dikalkulasi untuk memulai mendekode dari blok selanjutnya yang tidak rusak. Dengan cara ini, tanpa menggunakan *header macroblock* pun, sistem bisa mensinkronkan diri pada permulaan tiap-tiap *macroblock*.



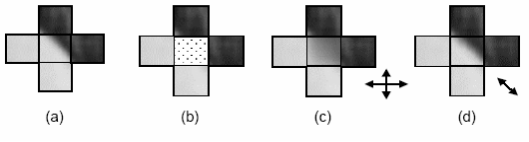
Gambar 12. Resinkronisasi decoder

4.1.3 Rekonstruksi

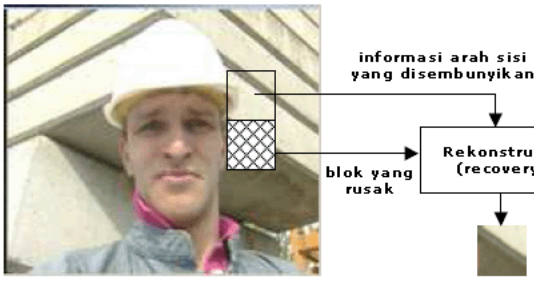
Mekanisme dari interpolasi di sepanjang arah sisi diilustrasikan pada (gambar a), blok tengah memiliki sisi yang tajam dengan empat blok tetangga, kemudian blok ini dirusak (gambar b). Setelah itu, blok yang dirusakkan tersebut direkonstruksi dengan teknik interpolasi bilinier (gambar c) dan teknik interpolasi sisi secara langsung (gambar d). Hasil memperlihatkan adanya superioritas dari interpolasi *edge-based*, terutama untuk blok-blok dengan sisi mayor.

Segera setelah error dideteki dan sinkronisasi didapatkan, langkah terakhir adalah me*recover single block* di mana *error* tersebut terjadi, sehingga informasi arah sisi diekstrak dari blok-blok pada lapisan atas ke setiap blok yang ada. Bit pertama yang disembunyikan yang mengindikasikan jenis dari blok yang hilang, dites untuk memeriksa apakan blok ini adalah sisi atau blok yang *smooth*. Bila yang diperiksa adalah blok sisi maka selanjutnya diinterpolasi dari dua blok tetangganya di sepanjang arah sisi (lihat gambar). Kalau tidak, untuk blok

yang *smooth*, maka diaplikasikan teknik interpolasi bilinier.



Gambar 13. Interpolasi di sepanjang arah sisi: (a) blok sisi dengan 4 blok tetangga, (b) blok yang dirusak, (c) hasil setelah interplasi bilinier, (d) hasil setelah interpolasi di sepanjang arah sisi

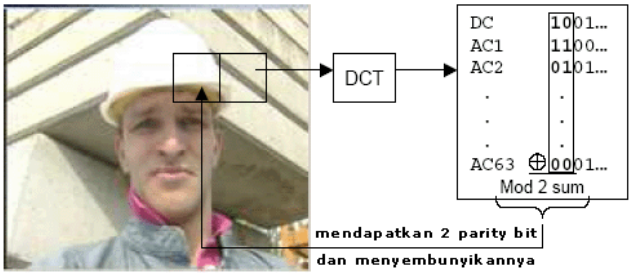


Gambar 14. Rekonstruksi blok yang rusak di dalam error intra-frame

4.1.3.1 *Overconcealment*

Setelah pendeteksian error, masalah lain yang penting yaitu mengukur kerusakan visual pada blok sebelum *recovery*, walaupun kerusakan visual yang terjadi terbilang kecil dan tidak dideteksi oleh *codec*, tetapi dapat dideteksi oleh sistem ini. Teknik *recovery edge-direction based* mencoba untuk merekonstruksi blok, yang memiliki degradasi visual yang sangan kecil, yang membuang semua informasi yang ada di dalamnya. Metode seperti itu untuk kasus ini tidak *preferable*, mengingat kapabilitas interpolasinya yang terbatas.

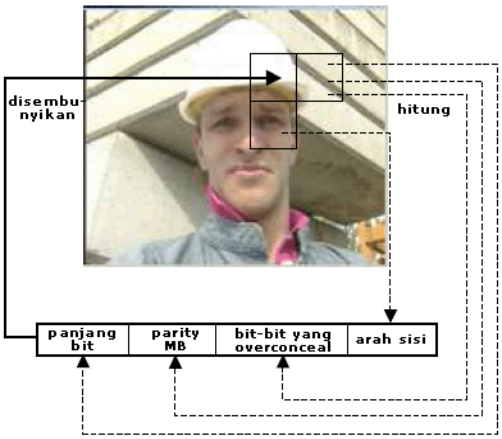
Dapat dipastikan, kualitas rekonstruksi biasanya berubah menjadi lebih jelek daripada blok *erroneus* yang ada. Situasi ini didefinisikan sebagai *overconcealment* dan dapat dihindari dengan menggunakan jumlah modula-2 dari 2 *most significant bits* (MSB) koefisien blok. Diasumsikan bahwa di dalam kasus *error* visual, MSB berukuran 2 bit diubah dan dapat dideteksi oleh 2 bit *parity* yang disembunyikan di dalam blok sebelumnya sehingga penyembunyian *error* dapat diaplikasikan (lihat gambar).



Gambar 15. Mendapatkan dan menyembunyikan *parity bit* untuk *overconcealment*

4.1.4 Sistem keseluruhan

Semua informasi yang dibutuhkan untuk penyembunyian *error intra-frame* dapat dilihat pada gambar di bawah. Pada saat panjang bit, *parity* blok, dan bit-bit *overconcealment* pada tiap blok disembunyikan ke dalam blok sebelumnya yaitu pada blok sebelah kiri dari blok yang bersangkutan, data arah sisi di-*embed* ke dalam blok di atasnya. Semua data tersebut digabung dan didapatkan *bit-stream* yang pendek. Setelah itu, *bit-stream* disembunyikan ke dalam blok tetangganya.

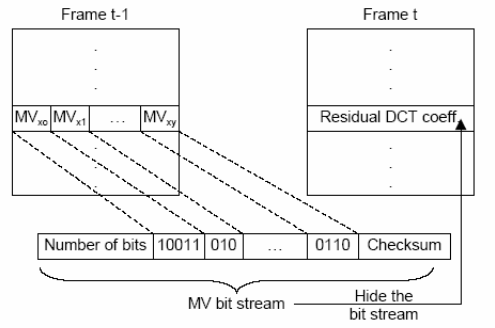


Gambar 16. Mendapatkan dan menyembunyikan *bit-bit* yang dibutuhkan untuk *overconcealment*

4.2 Penyembunyian Error Inter-Frame

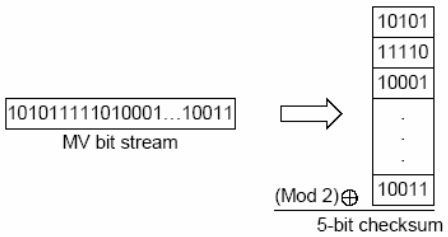
Metode-metode *recovery error inter-frame* difokuskan pada *recovery* informasi *motion* dari blok yang hilang untuk teknik penyembunyian yang lebih baik. Informasi

yang disembunyikan adalah vektor *motion*. Selebihnya, *checksum* digunakan untuk mendeteksi *error* pada data yang disembunyikan. Pada pendekatan ini, bit MV Huffman dan mode pengkodean pada blok-blok yang sebaris digabung dan didapatkan sebuah bit-stream. Lebih dari 9 bit ditambahkan di awal *bit stream* MV untuk mentransmisikan sejumlah bit di dalam bit stream ke *decoder*, yang mana MV dikodekan dalam bentuk variabel panjang. Oleh karena kemampuan pendeteksian *error* dari *decoder* H.263+ dibatasi, sebuah *checksum* 5-bit juga ditambahkan di akhir *bit stream* untuk kepentingan pendeteksian *error* pada *bit sream* MV. *Bit stream* ini didapat dari setiap baris MB di dalam *inter-frame* dan dimasukkan ke dalam koefisien DCT residual dari baris MB yang bersangkutan pada inter-frame selanjutnya, seperti digambarkan berikut ini:



Gambar 17. Menyembunyikan bit-bit MV untuk penyembunyian error inter-frame

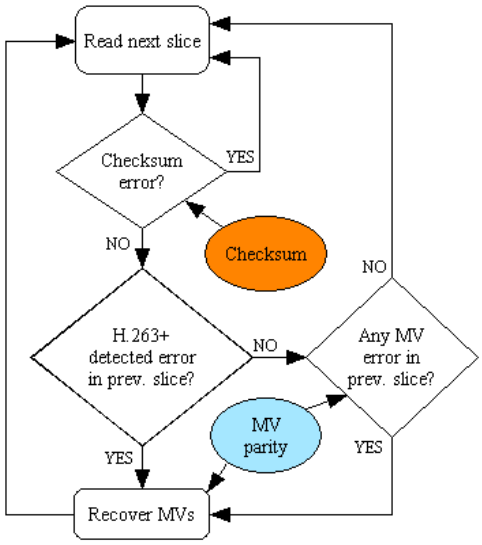
Setelah *decoder* mendeteksi *error*, *frame* berikutnya didekodekan untuk mendapatkan data MV yang tersembunyi pada *frame* pada saat itu. Dengan bit-bit *checksum*, *reliability* dari data yang disembunyikan diperiksa dan blok-blok yang rusak direkonstruksi dengan informasi MV. Pada beberapa kasus, hanya koefisien DCT residual yang dipengaruhi oleh sebuah kesalahan bit dan *decoder* tidak kehilangan sinkronisasinya, dalam arti lain hanya blok tersebut yang mengalami *corrupt*. Pada umumnya, *decoder* H.263+ tidak dapat mendeteksi *error* yang demikian. Tetapi, data yang disembunyikan di dalam koefisien DCT ini mungkin saja rusak. Dengan situasi seperti ini, *checksum* 5-bit digunakan untuk menegaskan *reliability* dari data yang disembunyikan. Untuk mendapatkan bit-bit checksum ini, bit stream MV pada mulanya dibagi menjadi blok-blok berukuran 5-bit. Setelah menyusun blok-blok tersebut secara vertikal, mereka dijumlahkan dengan modula-2.



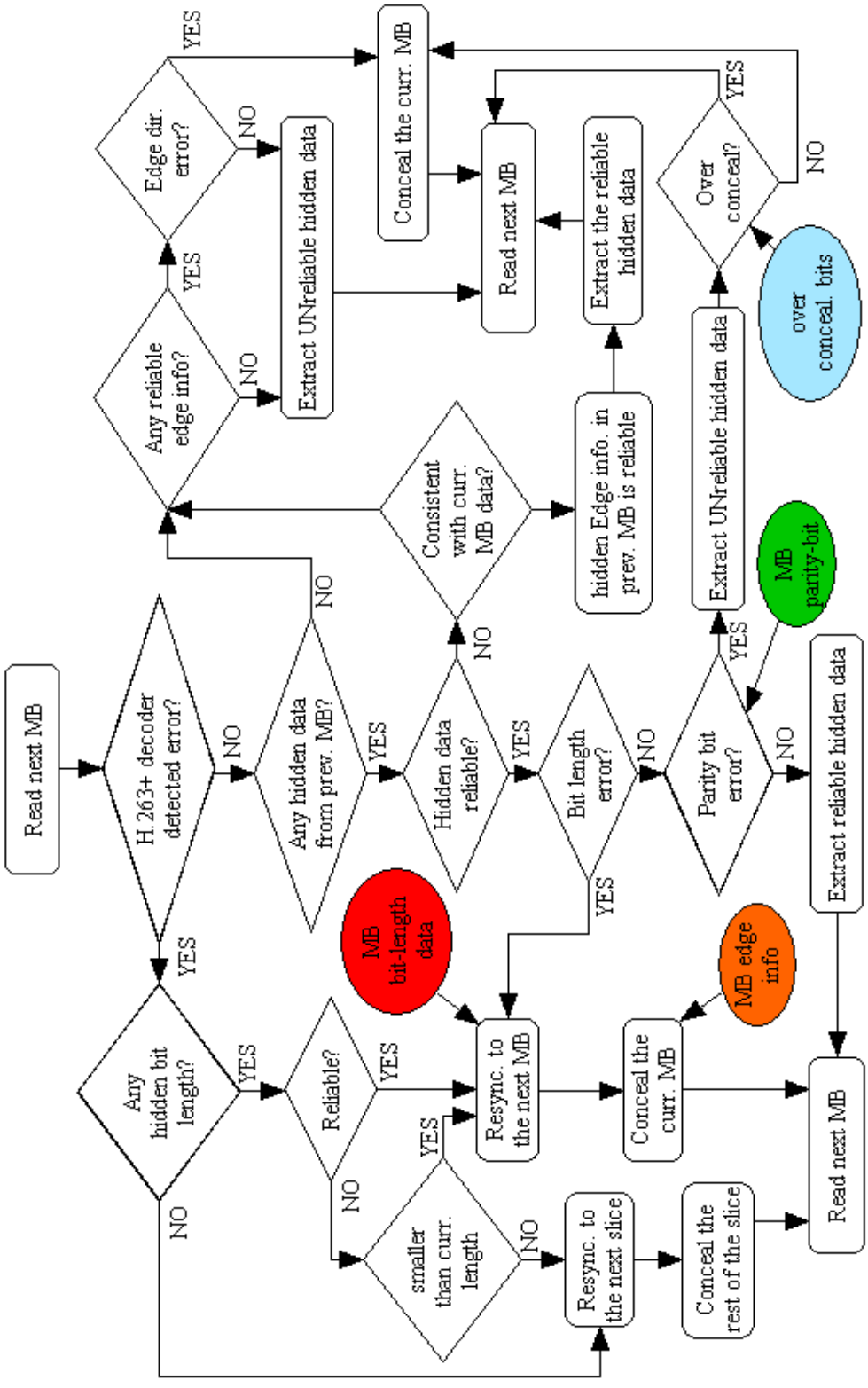
Gambar 18. Mendapatkan *bit-bit checksum*

4.3. Algoritma

Overview dari diagram blok algoritma ditunjukkan pada gambar di bawah. Pada kedua versi ini, terdapat tahap pendeteksian *error* yang berurutan. Mekanisme pendeteksian *error* internal dari H.263+ digunakan di dalam versi pengkodean intra maupun inter. Supaya error tidak kelihatan oleh *codec*, pengujian yang pertama dilakukan untuk *intra-frame* yaitu sinkronisasi dan *parity check*, sedangkan untuk *inter-frame* yaitu mengontrol informasi *checksum*. Sebagai tambahan, pengkodean intra juga memeriksa *overconcealment* sebelum memutuskan untuk melakukan beberapa rekonstruksi.



Gambar 19. Diagram blok sistem penyembunyian error inter-frame



Gambar 20. Diagram blok sistem penyembunyian error intra-frame

5. Kesimpulan

Metode penyembunyian *error* video baru, yang melakukan pendeteksian, sinkronisasi, dan rekonstruksi menggunakan data yang tersembunyi, mengkombinasikan sejumlah metode-metode yang diaplikasikan sebelumnya ditujukan untuk mendapatkan kualitas rekonstruksi yang lebih baik. Dengan penggabungan metode ini, beberapa metode yang baru juga digunakan untuk memperbaiki efisiensi metode sebelumnya.

Intensitas blok yang rusak di dalam *intra-frame* diperbaiki dengan interpolasi berbasis sisi dari blok-blok tetangganya sebagai metode *post-processing*. Informasi arah sisi dari blok yang rusak ditransmisikan ke *decoder* dengan menyembunyikannya melalui koefisien DCT blok tetangga. Dapat dipastikan pula bahwa semua blok tidak mempunyai karakteristik yang sama dari sudut pandang rekonstruksi. Meskipun interpolasi arah sisi unggul daripada interplasi bilinier konvensional, dapat ditunjukkan bahwa blok-blok tanpa single edge utama (seperti area yang bertekstur kasar), tidak dapt diinterpolasi dengan sukses melalui interpolasi berbasis sisi.

Beberapa *error* tidak menyebabkan degradasi visual yang besar pada blok tertentu dan karena skema interpolasi pada situasi seperti ini tidak akan dapat mendapatkan rekonstruksi visual yang lebih baik, maka dibutuhkan nilai kerusakan visual dari blok yang bersangkutan sebelum mereknstruksi blok yang rusak. Permasalahan ini dapat dipecahkan dengan menggunakan *parity* 2-bit, yang didapat dari MSB kopefisien DCT yang terkuantisasi. Walaupun performansi bit-bit *overconcealment* dapat memuaskan pada video yang mempunyai komponen frekuensi yang tinggi dan pergerakan yang cepat, tetapi hal itu tidak dapat berjalan dengan baik.

Hilangnya sinkronisasi menimbulkan masalah lain di dalam penyembunyian *error intra-frame*. Selama struktur *header* MB di dalam H.263+ tidak memberikan sebuah sinkronisasi yang mengacu pada *decoder*, koefisien-koefisien mulai didekodekan dengan salah di mana *decoder* dapat kehilangan *starting point* dari blok rusak berikutnya. Menginformasikan *decoder* mengenai panjang bit tiap blok dilakukan dengan menyembunyikan nilai panjang bit tiap blok ke dalam blok tetangganya.

DAFTAR PUSTAKA

[1] Yilmaz, Ayhan, Aydin Alatan. *Error Concealment of Video Sequences By Data Hiding.*

[2] Robie, David (2002) Russel Mersereau. *Video Error Correction Using Steganography*.

[3] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[4] Benjamin W. Wah, Xiao Su, and Dong Lin (2000), *"A survey of error-concealment schemes for real-time audio and video transmissions over the internet,"* Proceedings of the IEEE International Symposium on Multimedia Software Engineering.

[5] Shahram Shirani, Faouzi Kossentini, and Rabab Ward (1999), *"Error concealment methods, A comparative study,"* Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering.

[6] David Kwon and Peter Driessen, *"Error concealment techniques for H.263 video transmission,"* Proceedings of the 1999 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM '99).

[7] Stefan Katzenbeisser and Fabien A. P. Petitcolas (2000), *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Inc., USA.

[8] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom (2002), *Digital Watermarking*, Academic Press, USA.

[9] Mahalingam Ramkumar, Ali N. Akansu, and A. Aydın Alatan (1999), *"On the choice of transforms for data hiding in compressed video,"* Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing.

[10] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal Shamoon (1997), *"Secure spread spectrum watermarking for multimedia,"* IEEE Transactions on Image Processing, vol. 6.

[11] Joseph J.K. Ó Ruanaidh and Thierry Pun (1998), *"Rotation, scale and translation invariant spread spectrum digital image watermarking,"* Signal Processing, Elsevier, vol. 66, no. 3

[12] Min Wu and Bede Liu (1998), *"Watermarking for image authentication,"* Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP '98), vol. 2.

[13] International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), (1998), *"Draft text of recommendation H.263 version 2 ("H.263+") for decision"*

[14] Wenjun Zeng and Bede Liu (1999), *"Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding,"* IEEE Transactions on Circuits and Systems for Video Technology, vol. 9.

[15] David L. Robie and Russell M. Mersereau (2000), *"The use of Hough transforms in spatial error concealment,"* Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00), vol. 4.