

BAB II

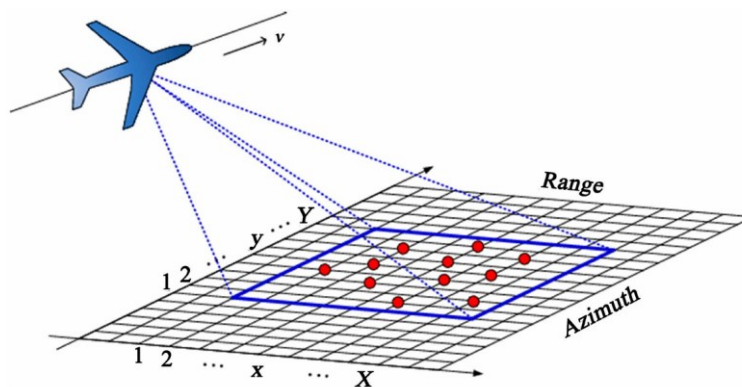
TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Citra Synthetic Aperture Radar

2.1.1.1 Pengertian dan Prinsip Kerja SAR

Synthetic Aperture Radar (SAR) adalah sistem penginderaan jauh aktif yang memanfaatkan gelombang mikro untuk menghasilkan citra permukaan bumi dalam berbagai kondisi pencahayaan dan cuaca (Villano, 2015). Tidak seperti sensor optik pasif yang bergantung pada sinar matahari, SAR bekerja dengan memancarkan pulsa gelombang elektromagnetik ke permukaan target dan kemudian merekam pantulan gelombangnya untuk membentuk citra dua dimensi berdasarkan waktu tunda dan intensitas pantulan sinyal (Chan, 2008). Teknologi ini menjadikan SAR sangat andal dalam kondisi ekstrem seperti malam hari, hujan, atau tutupan awan tebal.



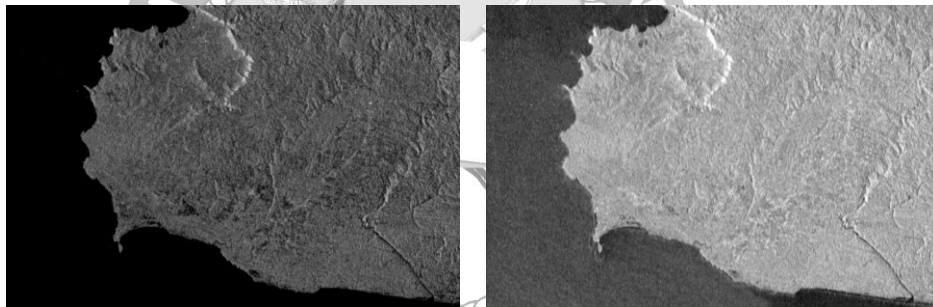
Gambar 2.1 Diagram sistem *Sentinel-1A*

(Voronin, 2022)

Seperti terlihat pada Gambar 2.1, sistem SAR satelit *Sentinel-1A* menggunakan antenna aktif berbasis C-band yang memungkinkan pencitraan area luas dengan resolusi tinggi, bahkan dalam mode operasional seperti Interferometric Wide Swath (IW) yang mengimplementasikan teknik TOPS (Terrain Observation with Progressive

Scans). Teknik ini memungkinkan pengambilan data dalam cakupan lebar tanpa kehilangan ketelitian spasial (Torres et al., 2017).

Polarimetri pada SAR mengacu pada arah polarisasi gelombang saat pemancaran dan penerimaan sinyal. *Sentinel-1A* menyediakan dua jenis polarimetri: VV (Vertical transmit – Vertical receive) dan VH (Vertical transmit – Horizontal receive) (Rao et al., 2017). Kanal VV lebih sensitif terhadap struktur vertikal seperti bangunan atau kendaraan, sementara kanal VH lebih efektif dalam mendeteksi vegetasi atau target tersembunyi di lingkungan alami (Dostálová et al., 2016; Diniz et al., 2022).



Gambar 2.2 Contoh hasil citra *Sentinel-1A* GRD Level-1 VH (Kiri) dan VV (Kanan) (Copernicus Browser, 2025)

Contoh visualisasi hasil polarimetri dapat dilihat pada Gambar 2.2, di mana kanal VV dan VH menunjukkan pola intensitas pantulan yang berbeda pada lokasi yang sama. Setiap piksel pada citra tersebut merepresentasikan energi pantulan radar dari permukaan tertentu, yang mengandung informasi amplitudo dan fasa yang sangat penting untuk analisis permukaan (Quentin et al., 2022).

Produk citra *Sentinel-1A* yang digunakan dalam penelitian ini adalah data Level-1 GRD (*Ground Range Detected*), yaitu citra hasil koreksi radiometrik dan geometrik dengan proyeksi langsung ke *permukaan* bumi (geocoding). Data ini disimpan dalam format .TIF dengan tipe UINT16, yang mampu merepresentasikan intensitas pantulan dalam rentang 0 hingga 65.535 tanpa kehilangan resolusi digital (Geudtner et al., 2018). Format ini

ideal untuk analisis ilmiah, tetapi menghasilkan ukuran file sangat besar—sering kali mencapai ratusan megabyte per frame, tergantung pada cakupan area dan resolusi spasialnya (Chen et al., 2015).

Karakteristik sinyal SAR yang koheren menghasilkan pola interferensi granular acak yang dikenal sebagai speckle. Speckle memang tampak menyerupai noise, namun sebenarnya mengandung informasi penting terkait kekasaran permukaan, komposisi objek, dan variasi lokal, sehingga tidak dapat dihilangkan secara sembarangan. (Zhang et al., 2007; El-Ashkar et al., 2021). Proses kompresi citra SAR memerlukan kehati-hatian agar struktur statistik dan spektral citra tetap terjaga, mengingat pentingnya menjaga keutuhan informasi yang terkandung di dalamnya.

Dengan ukuran file yang sangat besar dan kebutuhan akan kecepatan akses dalam sistem seperti UAV, satelit militer, dan pusat kendali taktis, kompresi citra SAR menjadi kebutuhan krusial. Tantangan ini tidak dapat diselesaikan dengan metode kompresi konvensional seperti JPEG atau PNG yang tidak memperhitungkan speckle dan struktur numerik SAR (Eichel & Ives, 1999). Oleh karena itu, dibutuhkan pendekatan khusus yang mempertahankan domain integer dan kompatibel dengan sistem real-time—yang menjadi fokus dalam penelitian ini.

2.1.1.2 Struktur dan Ukuran Data SAR

Citra *Synthetic Aperture Radar* (SAR) memiliki struktur data yang khas dan kompleks karena dihasilkan dari sinyal gelombang mikro yang dipantulkan dari permukaan bumi (Pachecho et al., 2019). Produk citra *Sentinel-1A* yang digunakan dalam penelitian ini merupakan data Level-1 GRD (*Ground Range Detected*), yaitu citra yang telah mengalami koreksi geometrik dan radiometrik, dan diproyeksikan ke koordinat geospasial permukaan bumi (Geudtner et al., 2018).

Data GRD ini disimpan dalam format geotiff (.TIF) dengan tipe data 16-bit unsigned integer (UINT16), yang berarti setiap piksel direpresentasikan oleh 2 byte data. Format ini mempertahankan informasi

amplitudo pantulan dengan akurasi tinggi, penting untuk menjaga kualitas analisis spektral dan spasial citra radar (Ives et al., 1998).

Setiap citra SAR *Sentinel-1A* disusun dalam bentuk matriks dua dimensi berukuran $M \times N$, di mana:

- M : jumlah baris piksel (resolusi spasial vertikal)
- N : jumlah kolom piksel (resolusi spasial horizontal)

Total ukuran data dalam byte (sebelum kompresi) untuk satu kanal, seperti VV atau VH, dapat dihitung menggunakan rumus berikut (Zou et al. 2012):

$$D = M \times N \times B \quad (2.1)$$

Dengan:

- B : jumlah byte per piksel (untuk UIN16, $B = 2$)

Misalnya, untuk citra *Sentinel-1A* beresolusi $M = 13,200$ dan $N = 18,000$, maka:

- $D = 13,200 \times 18,000 \times 2 = 475,200,000 \text{ byte} = 475.2 \text{ MB}$

Penggunaan dua kanal polarimetri (VV dan VH) akan menyebabkan ukuran file menjadi dua kali lebih besar:

$$D_{total} = 2 \times 475.2 \text{ MB} = 950.4 \text{ MB}$$

Ukuran file ini hanya untuk satu frame citra *Sentinel-1A*. Dalam praktiknya, satu jalur pengamatan dapat mencakup beberapa frame, sehingga kebutuhan penyimpanan dapat mencapai beberapa gigabyte, terutama untuk wilayah yang luas atau pengamatan waktu-majemuk (multitemporal).

Dalam konteks sistem operasi militer seperti UAV atau satelit taktis, kapasitas penyimpanan dan bandwidth transmisi sering kali terbatas. Oleh karena itu, ukuran file yang sangat besar ini menjadi hambatan utama dalam pemrosesan dan distribusi data SAR secara real-time (El-Ashkar et al., 2021; Wang et al., 2022). Permasalahan inilah yang melatarbelakangi perlunya metode kompresi yang dapat menurunkan ukuran data secara

signifikan, namun tetap menjaga struktur numerik asli dan kualitas informasi penting seperti speckle dan intensitas pantulan.

2.1.1.3 Karakteristik SVD Citra SAR

SVD merupakan teknik dekomposisi matriks yang memisahkan suatu citra menjadi komponen-komponen dominan berdasarkan tingkat kontribusi informasinya (Guaragnella & D'orazio, 2019). Dalam konteks citra SAR, citra dengan struktur yang sangat homogen, seperti permukaan laut terbuka, akan menghasilkan spektrum singular yang cepat menurun, yang menunjukkan bahwa mayoritas informasi terkonsentrasi pada sedikit komponen awal (Liu, 2014). Sebaliknya, citra dengan struktur kompleks seperti daratan urban atau vegetatif menghasilkan spektrum yang melandai, menandakan bahwa informasi tersebar lebih merata di banyak komponen (Park et al., 2012).

Penelitian oleh Ji & Zhang (2017) menunjukkan bahwa citra SAR dari wilayah daratan cenderung memiliki rank efektif yang lebih tinggi dibanding wilayah laut, karena terdapat banyak struktur lokal seperti bangunan, kontur vegetasi, atau elevasi medan yang kompleks. Temuan ini mendukung pentingnya pemrosesan berbasis rank seperti SVD dalam klasifikasi atau kompresi citra SAR.

2.1.1.4 Karakteristik Entropi Citra SAR

Entropi nol-order (H_0) mengukur tingkat keragaman simbol (nilai piksel) dalam suatu citra tanpa mempertimbangkan posisi atau keterkaitannya (Deng, 2022). Dalam konteks citra SAR, nilai H_0 dapat digunakan untuk menilai kompleksitas visual secara makroskopik, misalnya seberapa beragam nilai intensitas yang muncul dalam suatu segmen citra (Almeida et al., 2012). Citra laut cenderung memiliki sedikit variasi nilai karena permukaan air menghasilkan pantulan radar yang hampir seragam. Sebaliknya, citra daratan dengan bangunan, vegetasi, dan kontur yang kompleks menimbulkan nilai piksel yang lebih bervariasi (Shangguan et al., 2023).

Dari observasi visual terhadap dataset *Sentinel-1A*, dapat disimpulkan bahwa:

- Wilayah laut memiliki sebaran nilai yang sangat terkonsentrasi, dominan warna gelap-homogen, dengan sedikit fluktuasi akibat speckle.
- Wilayah daratan menunjukkan distribusi piksel yang sangat bervariasi, tergantung kerapatan struktur permukaan.
- Wilayah campuran seperti pantai atau delta menunjukkan zona berbeda dalam satu frame, dengan karakteristik citra laut dan darat muncul berdampingan.

2.1.1.5 Tantangan Kompresi SAR

Citra *Synthetic Aperture Radar* (SAR) memiliki tantangan unik dalam proses kompresi karena sifat sinyalnya yang koheren, kompleks, dan sangat bergantung pada struktur permukaan yang diamati (Maharjan & Li, 2023). Berbeda dengan citra optik, citra SAR memiliki struktur nilai piksel yang tidak dapat diprediksi hanya dengan melihat tekstur visualnya. Hal ini disebabkan oleh interaksi gelombang mikro dengan permukaan yang menghasilkan fenomena speckle, dan menghasilkan nilai piksel intensitas dalam rentang tinggi (0–65.535 pada format UINT16), tanpa ada perulangan pola yang jelas antar area (Doery, 2015).

Tantangan ini diperparah oleh perbedaan karakteristik spasial dan spektral antara berbagai jenis wilayah seperti daratan, lautan, dan campuran (Cassetti et al, 2022). Seperti diringkaskan dalam Tabel 2.1, masing-masing jenis citra memiliki ciri keterpusatan informasi dan keragaman simbol yang sangat bervariasi:

Tabel 2.1 Karakteristik Umum Citra SAR Berdasarkan Lingkungan

Jenis Wilayah	Ciri SVD	Ciri Entropi H_0	Karakter Umum Citra
Laut Terbuka	Spektrum tajam (low-rank)	Entropi rendah	Homogen, gelap, kontras rendah
Daratan Kompleks	Spektrum landai (high-rank)	Entropi tinggi	Tekstur rumit, detail banyak
Campuran (Pesisir)	Spektrum campuran	Entropi sedang	Blok beragam, area transisi

Dalam konteks penelitian ini, jenis wilayah tidak diproses secara berbeda dalam algoritma, tetapi digunakan sebagai pembanding untuk menguji ketahanan metode terhadap beragam karakteristik spasial. Oleh karena itu, metode kompresi yang digunakan harus bersifat adaptif, efisien, dan tetap mempertahankan kualitas informasi dalam kondisi struktur data yang sangat beragam ini.

2.1.2 Transformasi Preprocessing

2.1.2.1 Integer Wavelet Transform (IWT)

a. Wavelet Transform

Transformasi wavelet adalah metode dekomposisi sinyal yang menguraikan suatu data menjadi representasi dalam domain frekuensi dan waktu (atau spasial dalam kasus citra) (Das, 2012). Tidak seperti Transformasi Fourier yang bersifat global dan hanya memberikan informasi frekuensi, wavelet mampu menangkap karakter lokal dari sinyal, memungkinkan identifikasi fitur pada berbagai skala atau resolusi (Zhu, 2022). Hal ini menjadikan wavelet sangat cocok untuk pemrosesan citra, khususnya citra SAR, yang memiliki detail lokal sangat bervariasi dan tidak seragam.

Transformasi wavelet bekerja dengan memproyeksikan sinyal pada himpunan basis wavelet, yaitu fungsi-fungsi orthogonal atau ortonormal

yang bersifat lokal dan berskala (Ilyashenko et al., 2019). Suatu sinyal $f(t)$ dapat direpresentasikan dalam basis wavelet sebagai:

$$f(t) = \sum_{j,k} c_{j,k} \psi_{j,k}(t) \quad (2.2)$$

dengan:

- $\psi_{j,k}(t)$: fungsi wavelet hasil skala-translasi dari wavelet induk $\psi(t)$, didefinisikan sebagai:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad (2.3)$$

- j : indeks skala (resolusi)
- k : indeks translasi (posisi)
- $c_{j,k}$: koefisien wavelet, menunjukkan kontribusi dari basis $\psi_{j,k}$

Dalam ranah digital, implementasi yang paling umum adalah *Discrete Wavelet Transform* (DWT), yang secara efisien mendekomposisi sinyal atau citra menggunakan filter bank dua tahap (Chervyakov et al., 2018):

- Low-pass filter (L) untuk menghasilkan koefisien aproksimasi
- High-pass filter (H) untuk menghasilkan koefisien detail

Misalnya, untuk sinyal 1D diskrit $x[n]$, hasil transformasi wavelet adalah:

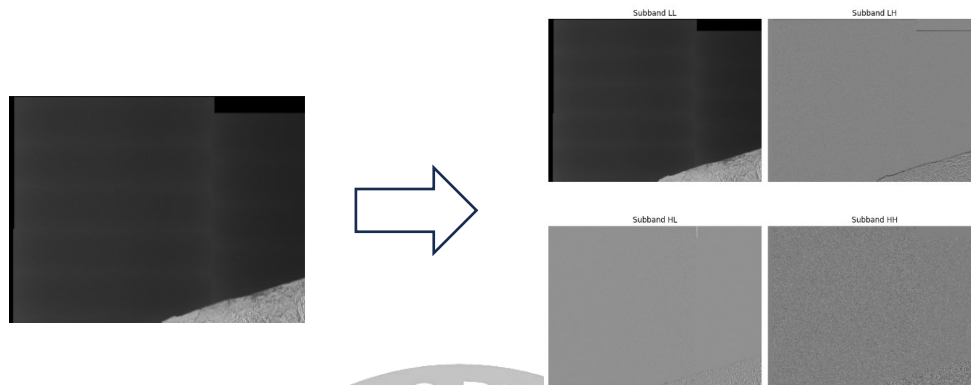
$$a[n] = \sum_k x[k] g[2n - k] \text{ (koefisien aproksimasi)}$$

$$d[n] = \sum_k x[k] h[2n - k] \text{ (koefisien detail)}$$

a dengan $g[n]$ dan $h[n]$ masing-masing merupakan filter low-pass dan high-pass.

Untuk citra 2D $I(x, y)$, transformasi ini diterapkan baris lalu kolom, sehingga menghasilkan empat subband:

- LL: aproksimasi
- LH: detail horizontal
- HL: detail vertikal
- HH: detail diagonal



Gambar 2.3 Skema 2D Wavelet Transform: LL, LH, HL, HH

Sebagaimana divisualisasikan pada Gambar 2.3, subband LL memuat informasi dominan dari citra, sedangkan LH, HL, dan HH memuat perubahan atau detail lokal.

Wavelet yang umum digunakan meliputi Haar, Daubechies, dan Biorthogonal, masing-masing dengan panjang filter dan karakteristik frekuensi yang berbeda. Namun, sebagian besar implementasi DWT bekerja di domain floating-point, yang memerlukan pembulatan saat konversi kembali ke representasi citra asli (Singh et al., 2011). Hal ini dapat mengakibatkan kehilangan presisi numerik, terutama pada citra SAR bertipe UINT16 yang sangat sensitif terhadap perubahan nilai.

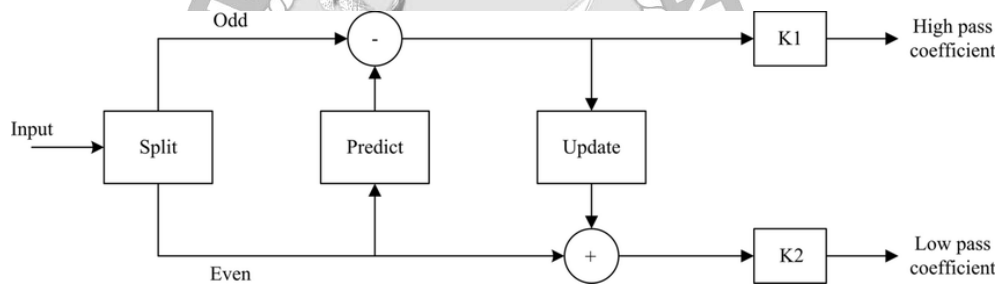
Untuk mengatasi keterbatasan ini, penelitian-penelitian terkini menyarankan penggunaan transformasi wavelet dalam domain bilangan bulat (integer) agar tetap reversibel dan lossless. Seperti yang dijelaskan oleh Cheng et al. (2019), transformasi wavelet integer menawarkan rekonstruksi yang tepat dan properti pelestarian integer yang penting untuk aplikasi seperti kompresi lossless pada citra penginderaan jarak jauh. Ini menjadi dasar pengembangan dari teknik *Lifting Scheme*, yang dibahas pada bagian selanjutnya.

b. Lifting Scheme

Lifting scheme adalah metode faktorisasi transformasi wavelet menjadi rangkaian operasi sederhana yang dapat diimplementasikan secara efisien dalam domain diskrit (Barina 2014). Tujuannya adalah menyusun transformasi wavelet tidak lagi sebagai proses konvolusi/filter

seperti pada DWT klasik, tetapi sebagai serangkaian langkah prediksi dan pembaruan (*predict-update steps*). Hal ini memungkinkan implementasi yang lebih cepat, lebih hemat memori, dan dapat dimodifikasi menjadi transformasi integer-to-integer, seperti pada *Integer Wavelet Transform* (IWT).

Konsep *lifting* pertama kali diperkenalkan oleh Sweldens (1996), dan dalam satu dekade terakhir masih menjadi dasar utama untuk transformasi wavelet lossless di bidang kompresi citra resolusi tinggi (Cheng et al., 2019; Tao et al., 2018).



Gambar 2.4 Diagram Alir Proses Lifting: Split → Predict → Update

Gambar 2.4 menunjukkan skema tiga tahap *lifting*:

1. *Split* (pemisahan): sinyal $x[n]$ dipisahkan menjadi:

- Sinyal genap: $x_e[n] = x[2n]$
- Sinyal ganjil: $x_o[n] = x[2n + 1]$

2. *Predict* step: sinyal ganjil diprediksi dari sinyal genap:

$$d[n] = x_o[n] - P(x_e[n]) \quad (2.4)$$

Dimana $P(\cdot)$ adalah fungsi prediksi berikut:

$$d[n] = x_o[n] - \left\lfloor \frac{x_e[n] + x_e[n + 1] + 1}{2} \right\rfloor \quad (2.5)$$

3. *Update* step: sinyal genap diperbarui menggunakan hasil prediksi untuk menjaga sifat konservatif energi (preservasi rata-rata lokal):

$$s[n] = x_e[n] + U(d[n]) \quad (2.6)$$

Dimana $U(\cdot)$:

$$s[n] = x_e[n] + \left\lfloor \frac{d[n-1] + d[n] + 2}{4} \right\rfloor \quad (2.7)$$

Hasil akhir dari *lifting* adalah dua set koefisien:

- $s[n]$: koefisien aproksimasi (low-pass)
- $d[n]$: koefisien detail (high-pass)

Transformasi ini reversibel bahkan dalam domain bilangan bulat jika dirancang tanpa operasi floating-point. Selain itu, *lifting* dapat dilakukan in-place, artinya koefisien dapat disimpan di array awal tanpa alokasi memori tambahan. Adapun keunggulan *Lifting* dibanding DWT klasik:

- Lebih cepat dan lebih efisien (memori rendah)
- Lebih mudah dimodifikasi untuk bekerja dalam domain integer
- Cocok untuk embedded system dan edge computing (misalnya UAV)

Cheng et al. (2019) menyatakan bahwa walaupun *Lifting* dapat mendukung implementasi di tempat yang efisien, rekonstruksi tanpa kehilangan penuh untuk sinyal masukan bilangan bulat, hanya mampu dicapai melalui Integer *Lifting* dan desain yang memperhatikan pembulatan. Dengan kata lain, agar transformasi wavelet benar-benar bekerja dalam integer-to-integer mapping tanpa kehilangan data, dibutuhkan pendekatan yang mengadaptasi *lifting* menjadi *Integer Wavelet Transform* (IWT).

2.1.2.2 Delta Encoding

Delta Encoding dalam konteks kompresi citra *Synthetic Aperture Radar* (SAR) digunakan sebagai transformasi awal yang bertujuan mengurangi pengulangan nilai piksel yang terjadi secara lokal (Al Qerom, 2025), khususnya pada bagian citra dengan perubahan intensitas yang minim. Teknik ini merepresentasikan data sebagai perbedaan antara satu nilai piksel dengan piksel sebelumnya, yang secara matematis dituliskan sebagai:

$$\delta_n = x_n - x_{n-1} \quad (2.8)$$

dengan x_n sebagai nilai intensitas piksel ke- n dan δ_n sebagai hasil *encoding*. Operasi sederhana ini memiliki kemampuan tinggi dalam mendeteksi pola linier atau semi-linier pada citra radar, terutama karena citra SAR cenderung memiliki struktur dengan fluktuasi lambat pada banyak bagian permukaannya, seperti laut atau dataran rendah (Xia et al., 2014). Transformasi ini dapat dibalik dengan mudah melalui penjumlahan kumulatif:

$$x_n = \delta_n + x_{n-1} \quad (2.9)$$

yang menjamin tidak adanya kehilangan informasi dalam proses *encoding* maupun *decoding*. Kelebihan lainnya adalah seluruh proses dapat dilakukan tanpa mengubah tipe data numerik dari citra asli. Transformasi ini sangat sesuai diterapkan pada data berformat UINT16, yang umum digunakan dalam penyimpanan citra SAR resolusi tinggi, mengingat seluruh operasinya berbasis bilangan bulat.

Efektivitas *Delta Encoding* dalam mengurangi redundansi secara matematis dapat dijelaskan dengan memperhatikan bahwa dalam citra SAR, apabila $x_n \approx x_{n-1}$, maka δ_n akan memiliki nilai mendekati nol. Contohnya, untuk vektor intensitas {1200,1201,1203,1203,1204}, maka hasil *Delta Encoding* menjadi {1200,1,2,0,1}. Jumlah nilai yang relatif kecil setelah *encoding* ini sangat signifikan karena pola data yang telah diratakan (*flattened*) dari matriks dua dimensi akan menghasilkan urutan data yang lebih mudah diprediksi dan lebih ringkas untuk disimpan atau dikirim (Fan, 2020). Pendekatan ini telah terbukti secara eksperimental meningkatkan kecepatan *encoding* hingga 8 kali dan *decoding* hingga 20 kali dibandingkan teknik konvensional seperti Xdelta atau Zdelta, sekaligus menjaga rasio kompresi tetap kompetitif (Xia et al., 2014).

Penelitian terbaru juga mendukung bahwa teknik ini tidak hanya unggul dalam efisiensi tetapi juga dalam ketangguhan operasionalnya. Dalam implementasi untuk sistem *backup* data skala besar, misalnya, *Delta*

Encoding mampu menghindari duplikasi data dalam sistem dengan tingkat fragmentasi tinggi berkat kemampuannya mengenali blok data serupa walaupun tidak identik secara *byte-to-byte* (Y. Zhang et al., 2024). Teknik ini dengan demikian relevan tidak hanya dalam penyimpanan data tetapi juga untuk transmisi data *real-time* dalam misi pengawasan militer menggunakan UAV atau satelit.

Representasi matematis dari proses ini juga memperlihatkan kelebihan logis dalam penyusutan ukuran data. Diferensiasi mampu memperkecil varian absolut antar elemen data, sehingga struktur hasil encoding menjadi lebih mudah untuk dikompresi secara lanjut (Fan et al., 2020). Metode ini tidak memanfaatkan korelasi spasial dua dimensi seperti halnya transformasi wavelet, namun tetap efektif dalam menangani redundansi satu dimensi tanpa menambah kompleksitas algoritma. Hal ini menjadikannya sangat ideal untuk diintegrasikan sebagai tahap awal dalam *pipeline* kompresi citra SAR multi-tahap yang menuntut kecepatan, efisiensi, dan kemampuan rekonstruksi tanpa cacat.

2.1.3 Regresi Linier

Salah satu tahap penting dalam metode ini adalah pengurangan redundansi pada subband HH dengan cara mengganti penyimpanan nilai aktual HH menjadi model prediksi berbasis regresi linier berganda (Fu et al., 2023). Pendekatan ini bertujuan untuk menyederhanakan representasi HH tanpa kehilangan struktur informasi utama, sehingga dapat menurunkan ukuran kompresi tanpa mengorbankan kualitas citra rekonstruksi secara signifikan.

Model regresi dibangun dengan memanfaatkan tiga variabel bebas, yaitu:

- HL : subband horizontal detail hasil IWT,
- LH : subband vertikal detail hasil IWT,
- $|HL - LH|$: selisih absolut antar dua subband tersebut,

serta ditambahkan satu konstanta bias untuk membentuk model linier empat parameter. Persamaan umum dari model regresi ini dapat dituliskan sebagai berikut:

$$\widehat{HH}(i,j) = \alpha \cdot HL(i,j) + \beta \cdot LH(i,j) + \gamma \cdot |HL(i,j) - LH(i,j)| + \delta \quad (2.10)$$

Keterangan:

- $\widehat{HH}(i,j)$: nilai prediksi dari subband HH pada posisi (i,j) ,
- α, β, γ : koefisien regresi yang dihitung dari data,
- δ : bias regresi,
- $HL(i,j)$ dan $LH(i,j)$: nilai pada subband HL dan LH pada posisi (i,j) .

Model regresi ini diterapkan per blok citra berukuran tertentu (misalnya 1024×1024 piksel), dan hanya parameter $\theta = [\alpha, \beta, \gamma, \delta]$ dari tiap blok yang disimpan, bukan nilai HH secara eksplisit. Dengan pendekatan ini, data subband HH dapat diprediksi ulang pada saat dekompresi dengan akurasi yang tinggi, serta mengurangi kebutuhan penyimpanan secara signifikan tanpa menurunkan kualitas visual secara berarti, seperti yang diindikasikan oleh nilai PSNR dan MSE dalam hasil evaluasi (Ji & Zhang, 2017).

2.1.4 Kompresi Berbasis Entropi Algoritma *Zstandard* (Zstd)

Zstandard (Zstd) adalah algoritma kompresi data *lossless* modern yang dirancang untuk mencapai rasio kompresi tinggi dengan kecepatan kompresi dan dekompresi yang sangat cepat. Algoritma ini dikembangkan oleh Yann Collet (2018) di Facebook dan menjadi populer karena fleksibilitas serta efisiensinya dalam berbagai konteks pemrosesan data, termasuk kompresi citra, *file* sistem, arsip, hingga transmisi data *real-time* (Huang et al., 2021). Dalam konteks pengolahan citra UINT16 seperti citra SAR, Zstd menjadi pilihan strategis karena mampu menangani distribusi data besar dengan nilai numerik tinggi dan pola berulang yang kompleks (Zhang et al., 2024).

Keunggulan utama Zstd terletak pada kombinasi kecepatan tinggi dan fleksibilitas tingkat kompresi yang dapat disesuaikan, tanpa

mengorbankan sifat *loss/less* dari data (Collet, 2018). Ini berarti seluruh data asli dapat dipulihkan secara identik setelah dekompresi. Fitur multilevel yang disediakan oleh Zstd—dengan kisaran level kompresi dari 1 hingga 22—memungkinkan pengguna memilih antara kecepatan pemrosesan atau rasio kompresi yang maksimal, tergantung pada konteks penggunaannya. Kecepatan dekompresi Zstd juga secara signifikan lebih unggul dibandingkan banyak algoritma lain seperti *Deflate* (yang digunakan dalam ZIP atau PNG), menjadikannya sangat efisien untuk sistem pemantauan *real-time* atau transmisi dari perangkat dengan sumber daya terbatas seperti UAV dan nanosatelit.

Dalam arsitektur internalnya, Zstd menggabungkan berbagai teknik optimasi seperti *entropy coding* berbasis *Finite State Entropy* (FSE), *Huffman Coding*, dan teknik pencocokan blok berulang yang cerdas. Efektivitas Zstd akan meningkat secara signifikan apabila data yang dikompresi telah melalui proses prapemrosesan seperti *Delta Encoding* atau transformasi *wavelet*, seperti IWT, karena pola-pola data menjadi lebih ringkas dan konsisten secara struktural (Xia et al., 2014).

Dengan struktur modular dan parameterisasi yang kuat, Zstd tidak hanya menjadi kompresor teknis, tetapi juga platform kompresi yang dapat disesuaikan dengan karakteristik khusus data input. Dalam aplikasi pertahanan dan keamanan yang menggunakan citra SAR 16-bit, penggunaan Zstd memungkinkan penyesuaian tingkat kompresi secara dinamis tanpa mengganggu integritas citra atau menambah beban komputasi yang signifikan. Fleksibilitas dan ketangguhan inilah yang menjadikan Zstd sangat cocok untuk integrasi dalam sistem kompresi citra berketelitian tinggi.

2.1.4.1 Block Splitting

Tahap pertama dalam proses kompresi Zstd disebut sebagai *Block Splitting*, yaitu proses segmentasi data menjadi blok-blok terpisah sebelum dilakukan pemrosesan lebih lanjut seperti pencarian pola atau pengkodean

entropi. Proses ini sangat penting dalam konteks kompresi citra SAR bertipe UIN16, karena ukuran data besar yang direpresentasikan oleh dua *byte* per piksel sering kali mengandung struktur berulang yang tidak tersebar merata di seluruh citra (Zhao, et al., 2021). Dengan membagi data menjadi blok-blok kecil, Zstd dapat memproses dan mengompresi setiap bagian secara lokal, menyesuaikan strategi kompresinya dengan karakteristik blok tersebut.

Block splitting dilakukan secara otomatis dan efisien dalam Zstd, biasanya menggunakan ukuran blok *default* sekitar 128 KB. Pemisahan ini tidak hanya dilakukan berdasarkan panjang *byte*, tetapi juga mempertimbangkan profil statistika lokal dalam data. Misalnya, jika suatu bagian dari citra memiliki variasi intensitas piksel yang rendah, maka blok tersebut dapat dikompresi lebih agresif dibandingkan blok lain dengan variasi kompleks. Secara konseptual, misalkan matriks citra memiliki dimensi $M \times N$, maka citra dapat diratakan menjadi vektor satu dimensi $X = [x_1, x_2, \dots, x_{MN}]$, yang kemudian dibagi menjadi blok-blok B_i dengan panjang tetap L , sehingga:

$$X = \bigcup_{i=1}^k B_i \quad (2.11)$$

dengan $B_i = [x_{(i-1)L+1}, x_{(i-1)L+2}, \dots, x_{iL}]$ dan $k = \lceil \frac{MN}{L} \rceil$. Pemrosesan kompresi dilakukan per blok, memungkinkan penggunaan teknik kompresi lokal seperti *match finding* dan *entropy coding* secara lebih presisi sesuai pola masing-masing blok.

Contoh penerapan *block splitting* pada citra SAR dapat dilihat pada pengolahan area laut dan area urban. Area laut cenderung memiliki nilai piksel yang seragam, menghasilkan blok-blok yang sangat cocok untuk kompresi dengan rasio tinggi. Sebaliknya, area urban dengan banyak tepi dan variasi permukaan menghasilkan blok-blok dengan kompleksitas lebih tinggi, yang akan diproses dengan strategi yang lebih konservatif. Strategi

ini mengurangi *noise* hasil kompresi dan mempertahankan kualitas citra pada bagian penting.

Penelitian terkini menegaskan bahwa *block-wise compression* tidak hanya meningkatkan efisiensi rasio kompresi, tetapi juga memungkinkan kompresi adaptif berbasis karakteristik lokal blok. (Zhao, et al., 2021) mengembangkan sistem kompresi adaptif yang memproses tiap blok berdasarkan kontennya, dan menunjukkan bahwa pendekatan ini menghasilkan kinerja *rate-distortion* (R-D) yang lebih baik dibanding metode global (Zhao, et al., 2021). Oleh karena itu, *block splitting* menjadi fondasi penting dalam *pipeline* kompresi Zstd, terutama untuk aplikasi citra UINT16 seperti SAR yang memiliki karakteristik spasial dan tekstural yang sangat variatif antar blok.

2.1.4.2 Match Finding

Tahap *match finding* dalam algoritma Zstd merupakan proses identifikasi pola berulang dalam data input dengan tujuan untuk menghindari penyimpanan ulang data yang identik (Collet & Kucherawy, 2018). Dalam konteks citra UINT16, seperti citra SAR, pola berulang sering terjadi pada area yang memiliki tekstur atau nilai intensitas seragam, sehingga proses ini sangat efektif dalam mengurangi redundansi struktural. Proses ini dilakukan dengan mencari urutan nilai piksel yang sudah muncul sebelumnya dan merepresentasikannya sebagai pasangan *offset* dan panjang (*offset-length pair*), bukan menyimpan ulang nilai aktual.

Secara matematis, diberikan sebuah urutan data $D = \{d_1, d_2, \dots, d_n\}$, maka untuk setiap posisi i , algoritma akan mencari posisi $j < i$ dan panjang l sedemikian sehingga:

$$d_i = d_j, d_{i+1} = d_{j+1}, \dots, d_{i+l-1} = d_{j+l-1} \quad (2.12)$$

dengan nilai l merupakan panjang maksimal dari subsekuen yang cocok. Data pada posisi i sampai $i + l - 1$ direpresentasikan sebagai token kompresi apabila kecocokan telah teridentifikasi sebelumnya:

$$T_i = (\text{offset} = i - j, \text{length} = l) \quad (2.13)$$

Contohnya, pada deret [34, 56, 78, 90, 34, 56, 78], subsekuen dari indeks 5 hingga 7 merupakan salinan dari indeks 1 hingga 3. Maka, nilai [34, 56, 78] kedua tidak akan disimpan sebagai literal, melainkan sebagai (offset = 4, length = 3).

Algoritma Zstd menggunakan *hash table* untuk mempercepat pencarian subsekuen yang cocok, di mana setiap nilai atau urutan nilai tertentu akan di-*hash* ke dalam indeks tabel *hash*. Hal ini memungkinkan proses pencarian dilakukan dalam waktu konstan rata-rata untuk setiap posisi dalam data. Parameter penting yang memengaruhi proses ini adalah *window size* atau ukuran jendela pencarian yang menentukan sejauh mana ke belakang algoritma akan mencari kecocokan. Semakin besar *window size*, semakin besar peluang untuk menemukan kecocokan panjang, tetapi juga semakin besar biaya komputasi dan penggunaan memori.

Level kompresi dalam Zstd secara langsung memengaruhi strategi pencocokan ini. Pada level rendah, algoritma menggunakan pendekatan *lazy match* dan menghentikan pencarian setelah menemukan kecocokan pertama, sedangkan pada level tinggi, algoritma menerapkan *greedy match* atau bahkan optimal parsing untuk menemukan kecocokan terpanjang. Strategi pencarian yang lebih dalam ini meningkatkan rasio kompresi tetapi juga memperbesar waktu komputasi karena lebih banyak kemungkinan diperiksa sebelum keputusan dibuat (Bringmann, et al., 2019).

2.1.4.3 Literal Coding

Literal coding adalah proses yang dilakukan dalam algoritma kompresi untuk menangani bagian-bagian dari data input yang tidak dapat direpresentasikan sebagai referensi dari pola sebelumnya. Dalam konteks algoritma Zstd, *literal coding* menjadi tahap penting dalam memastikan seluruh data dapat dikompresi secara efisien, termasuk simbol-simbol yang bersifat unik atau jarang muncul. Data literal dalam citra SAR UINT16 dapat mencakup nilai-nilai intensitas piksel yang tidak memiliki kecocokan dalam jendela pencarian *match finder*, sehingga tidak dapat dikodekan sebagai pasangan *offset* dan panjang.

Secara matematis, jika $D = \{d_1, d_2, \dots, d_n\}$ adalah deret nilai piksel, dan *match finder* gagal menemukan urutan $\{d_i, \dots, d_{i+k}\}$ yang cocok di posisi sebelumnya, maka deret tersebut akan diperlakukan sebagai literal dan dikodekan secara langsung. Proses ini mengelompokkan nilai-nilai literal ke dalam sebuah *buffer* literal $L = \{l_1, l_2, \dots, l_m\}$, di mana masing-masing nilai akan dikompresi selanjutnya menggunakan teknik *entropy coding*. Proses *literal coding* ini bukan hanya menyimpan data secara eksplisit, tetapi juga mempersiapkan struktur data agar optimal untuk tahap pengkodean probabilistik berikutnya.

Contohnya, untuk citra radar yang memiliki nilai intensitas unik di beberapa bagian, seperti:

$$[12000, 13500, 14000, 13000, 12500] \quad (2.14)$$

Urutan nilai tersebut akan dianggap sebagai literal dan dimasukkan ke dalam *buffer* literal jika tidak ditemukan kemunculan sebelumnya dalam jendela pencarian. Nilai-nilai ini kemudian diproses menggunakan histogram frekuensi untuk disiapkan dalam bentuk tabel probabilitas.

Menurut Collet & Kucherawy (2018), *literal coding* dalam Zstd memiliki fleksibilitas tinggi untuk beradaptasi terhadap data kompleks seperti UINT16 dengan histogram multi-puncak dan sebaran *non-uniform*, dan optimalisasi dilakukan melalui pemisahan *literal stream* dari token stream sebelum dilakukan *entropy encoding* (Collet & Kucherawy, 2018).

2.1.4.4 Entropy coding

Tahap berikutnya dalam algoritma kompresi Zstd adalah *Entropy coding*, yang bertujuan untuk mengubah data yang telah diproses sebelumnya menjadi representasi *bitstream* paling ringkas berdasarkan frekuensi kemunculan simbol. Dalam Zstd, metode utama yang digunakan untuk proses ini adalah *Finite State Entropy* (FSE) dan *Huffman Coding*, yang merupakan dua teknik yang mampu melakukan kompresi *loss/less* dengan efisiensi tinggi (Srimanth & Jatoth, 2024).

Proses *entropy encoding* bekerja dengan mengasumsikan bahwa simbol yang lebih sering muncul dapat direpresentasikan dengan lebih

sedikit bit, sedangkan simbol yang jarang muncul diberikan representasi bit yang lebih panjang. Representasi ini mengikuti prinsip dasar dari teori informasi yang mengukur jumlah informasi berdasarkan probabilitas kemunculan suatu simbol. Misalnya, jika sebuah simbol s memiliki probabilitas $P(s)$, maka secara teoritis panjang minimum representasi biner dari simbol tersebut adalah:

$$L(s) = -\log_2 P(s) \quad (2.15)$$

Namun, dalam praktiknya, pengkodean biner dilakukan dengan pembulatan nilai ini agar panjang kode dalam bit tetap *integer*, dan ini dilakukan dengan algoritma Huffman ataupun FSE secara efisien.

Dalam konteks FSE, data diproses melalui automata berbasis *state-machine* yang menyimpan keadaan saat ini dalam sebuah bilangan bulat tunggal, yang digunakan untuk menentukan bit keluaran berikutnya. Metode ini menggabungkan kecepatan *Huffman Coding* dengan efisiensi kompresi dari *arithmetic coding* (Duda, 2013). FSE menggunakan tabel *lookup* yang telah dipetakan sebelumnya berdasarkan histogram nilai input untuk mempercepat *encoding* dan *decoding*.

Misalnya, anggap kita memiliki distribusi simbol berikut untuk hasil *Delta Encoding*:

- Simbol 0: 50%
- Simbol 1: 30%
- Simbol 2: 20%

Maka kita dapat menetapkan kode Huffman sebagai:

- Simbol 0 $\rightarrow$ kode '0'
- Simbol 1 $\rightarrow$ kode '10'
- Simbol 2 $\rightarrow$ kode '11'

Dalam FSE, distribusi ini diubah menjadi tabel probabilitas dan *state transition* sehingga representasi *state machine* bisa digunakan untuk mengkodekan *bitstream* sesuai distribusi tersebut. Keuntungan utama dari FSE adalah tidak memerlukan pemisahan eksplisit antara simbol dan

panjang kodenya karena semua berada dalam alur otomatis berbasis tabel transisi, memungkinkan *decoding* sangat cepat dan kompak.

Selama proses *decoding*, algoritma membaca bit dari *bitstream* dan mengacu pada tabel inversi untuk menentukan simbol asli dan memperbarui state. Proses ini secara efisien memulihkan urutan simbol awal tanpa kehilangan informasi, menjadikan FSE sangat sesuai untuk aplikasi kompresi citra UINT16 dari SAR yang membutuhkan akurasi data absolut.

Implementasi *entropy encoding* seperti ini memberikan efisiensi sangat tinggi karena memanfaatkan keteraturan dalam distribusi data hasil transformasi atau *Delta Encoding*. Dalam studi terbaru, Srimanth dan Jatoth (2024) membuktikan bahwa penggunaan *entropy encoding* berbasis Huffman setelah transformasi seperti DCT atau *wavelet* secara signifikan meningkatkan rasio kompresi tanpa penurunan kualitas visual citra (Srimanth & Jatoth, 2024). Oleh karena itu, kombinasi *preprocessing* dan *entropy coding* seperti yang dilakukan Zstd menjadikannya metode unggulan dalam kompresi citra resolusi tinggi seperti SAR.

Setelah proses *entropy coding* selesai, representasi biner terkompresi dari blok data yang siap disimpan. Tahapan ini tidak melibatkan proses pengaturan teknis yang sangat presisi agar data dapat didekodekan kembali secara sempurna dan efisien. Proses ini dimulai dengan penggabungan hasil pengkodean dari setiap blok, baik itu hasil dari *Huffman Coding* maupun FSE, dan diikuti dengan penambahan *metadata* yang mendeskripsikan struktur blok dan tabel entropi yang digunakan (Konstantinov et al., 2019).

Bitstream yang dihasilkan Zstd memiliki struktur modular yang terdiri dari *header*, blok data terkompresi, dan bagian akhir yang dapat berisi checksum untuk verifikasi integritas. *Header* mencakup informasi penting seperti ukuran blok, tipe kompresi yang digunakan, serta offset dari tabel FSE atau Huffman yang diperlukan saat dekompresi. Hal ini memungkinkan sistem dekompresor untuk menafsirkan data secara langsung tanpa harus memproses keseluruhan *bitstream* sejak awal, mendukung kemampuan

decoding paralel dan *streaming* yang sangat berguna dalam sistem *real-time* seperti UAV atau sistem radar militer (Baroud, et al., 2017).

2.1.5 Evaluasi Efisiensi Kompresi

2.1.5.1 Singular Value Decomposition (SVD)

Citra SAR memiliki karakteristik kompleks yang khas, yaitu kombinasi antara struktur tekstur lokal yang kuat dan kehadiran *speckle* yang tampak acak. Berbagai studi menunjukkan bahwa, walaupun *speckle* bersifat seperti noise, citra SAR secara statistik tetap menyimpan informasi yang terstruktur dan dapat dianalisis untuk mengevaluasi distribusi spasialnya (Ren et al., 2015). Dalam konteks penelitian ini, pendekatan matematis yang digunakan untuk mengevaluasi keteraturan dan keterpusatan informasi dalam citra adalah melalui *Singular Value Decomposition* (SVD), yang dianalisis dalam bentuk logaritmik (log-SVD) guna mempermudah visualisasi dan interpretasi terhadap peluruhan nilai singular (Ren et al., 2015).

SVD merupakan suatu teknik dekomposisi matriks $A \in \mathbb{R}^{m \times n}$, yang memfaktorkan citra ke dalam tiga komponen utama:

$$A = U\Sigma V^T \quad (2.16)$$

dengan U dan V adalah matriks ortonormal, serta Σ adalah matriks diagonal yang mengandung nilai-nilai singular $\sigma_1, \sigma_2, \dots, \sigma_r$, diurutkan secara menurun. Nilai-nilai singular tersebut mencerminkan kontribusi energi dari masing-masing komponen terhadap representasi total citra. Citra dikatakan memiliki rank efektif rendah dan secara teoritis dapat dikompresi secara efisien apabila mayoritas energinya terkonsentrasi pada nilai singular awal.

Dalam implementasinya, analisis log-SVD diterapkan pada blok citra *Sentinel-1A* berukuran 256×256 piksel. Ukuran ini dipilih untuk menjamin keterwakilan struktur lokal sekaligus memastikan efisiensi perhitungan. Proses diawali dengan pembacaan citra .TIF format UINT16 menggunakan rasterio, diikuti dengan *padding* reflektif untuk menyesuaikan dimensi ke kelipatan dua. Setelah itu, citra dikenai transformasi IWT dua dimensi untuk menghasilkan empat *subband*: LL, LH, HL, dan HH. Subband

LL berfungsi menyimpan informasi frekuensi rendah atau struktur dominan citra, sedangkan *subband* lainnya berisi detail arah horizontal, vertikal, dan diagonal.

Setelah transformasi, masing-masing *subband* diratakan menjadi vektor satu dimensi dan dikenai proses *Delta Encoding* untuk menghitung perbedaan antar elemen berturut. *Delta Encoding* bertujuan mempersempit rentang distribusi nilai dan mengurangi korelasi spasial lokal sebelum kompresi. Untuk setiap *subband* dan hasil delta-nya, vektor dirapikan kembali menjadi matriks semu berukuran $\sqrt{N} \times \sqrt{N}$ agar dapat dihitung SVD-nya. Nilai singular dari masing-masing matriks kemudian dinormalisasi terhadap σ_1 dan ditampilkan dalam skala logaritmik:

$$\log_{10} \left(\frac{\sigma_i}{\sigma_1} + \epsilon \right) \quad (2.17)$$

dengan ϵ adalah konstanta kecil, biasanya 10^{-12} , untuk mencegah singularitas pada operasi log.

Hasil log-SVD divisualisasikan dalam satu grafik gabungan agar dapat dibandingkan antara citra asli, *subband* IWT, dan hasil *Delta Encoding*. Pola peluruhan yang tajam pada kurva log-SVD menandakan bahwa sebagian besar informasi terkandung dalam komponen singular awal. Fenomena ini diamati secara konsisten pada *subband* LL, baik sebelum maupun sesudah *Delta Encoding*, yang menunjukkan bahwa sebagian besar informasi struktural citra SAR tersimpan dalam dimensi rendah. Sebaliknya, *subband* HH dan HL menunjukkan kurva peluruhan yang lebih landai, menunjukkan sebaran energi yang lebih merata dan keterlibatan noise atau detail halus seperti *speckle*.

Penemuan menemukan bahwa citra radar memiliki struktur internal yang dapat direduksi secara signifikan tanpa kehilangan informasi esensial. Mereka juga menunjukkan bahwa spektrum singular citra radar cenderung menurun eksponensial pada subregion homogen, dan melandai pada wilayah heterogen atau bertekstur tinggi. Dalam konteks penelitian ini, hasil log-SVD memperkuat argumen bahwa transformasi dan *encoding* seperti

IWT dan delta dapat meningkatkan konsentrasi informasi pada komponen rank rendah, yang selanjutnya akan memperbesar efisiensi kompresi dengan algoritma seperti *Zstandard*.

Dengan demikian, pendekatan log-SVD dalam penelitian ini bukan hanya sekadar alat bantu visualisasi, tetapi merupakan indikator matematis dan struktural yang sah dalam menilai keterpusatan informasi dan potensi kompresibilitas citra SAR. Ia mampu menjelaskan secara objektif bahwa meskipun citra radar memiliki komponen noise seperti *speckle*, struktur dominan tetap hadir dan dapat ditangkap serta dikompres secara efektif melalui pendekatan transformasi spasial dan pengurangan entropi lokal.

2.1.5.2 Entropi Nol-Order (H_0)

Dalam penelitian ini, pengukuran kompleksitas informasi pada citra SAR dilakukan melalui perhitungan entropi nol-order atau Shannon entropy, yang merepresentasikan rata-rata jumlah bit yang dibutuhkan untuk menyandikan setiap simbol dalam data, tanpa mempertimbangkan hubungan antar simbol (Almeida et al., 2012). Entropi nol-order digunakan secara luas sebagai indikator awal tingkat kompresibilitas suatu sumber data diskrit, di mana semakin rendah nilai entropinya, semakin besar kemungkinan data tersebut dapat dikompresi secara efisien. Dalam konteks kompresi citra SAR yang berisi pola *speckle* dan struktur lokal yang khas, entropi nol-order digunakan untuk menilai seberapa terpusat atau tersebar distribusi nilai piksel, baik sebelum maupun sesudah transformasi.

Secara matematis, entropi nol-order H_0 dirumuskan sebagai:

$$H_0 = -\sum_{i=1}^n p(b_i) \log_2 p(b_i) \quad (2.18)$$

di mana H_0 menyatakan entropi dalam satuan bit per simbol, b_i adalah simbol ke- i dari himpunan simbol diskrit Σ , dan $p(b_i)$ merupakan probabilitas kemunculan simbol b_i yang dihitung sebagai rasio frekuensi kemunculan simbol terhadap total simbol N dalam domain data. Simbol dalam konteks ini adalah nilai piksel dalam representasi bilangan bulat (misalnya int16), baik pada citra asli, *subband* hasil IWT, maupun hasil

transformasi *Delta Encoding*. Untuk mencegah kesalahan numerik, simbol dengan probabilitas nol ($p(b_i) = 0$) diabaikan karena secara konvensional $\log_2(0)$ tidak terdefinisi.

Perhitungan dilakukan dengan menyusun histogram distribusi nilai simbol terlebih dahulu, kemudian menghitung proporsi relatif kemunculannya, sebelum diaplikasikan ke dalam rumus di atas. Proses ini dilakukan secara terpisah untuk masing-masing tahap pemrosesan, yakni citra asli, *subband* IWT (LL, LH, HL, HH), dan setelah *Delta Encoding*. Dengan cara ini, nilai H_0 dapat digunakan sebagai metrik kuantitatif untuk menilai perubahan kompleksitas informasi dan potensi kompresi di setiap tahap *pipeline*.

Metode ini didasarkan pada landasan teori informasi klasik yang masih digunakan secara luas dalam penelitian-penelitian kompresi citra modern. Referensi terkini yang mendukung penerapan entropi nol-order sebagai metrik dasar dalam sistem kompresi, termasuk untuk citra SAR dan citra bertipe *integer* lainnya, dapat ditemukan pada studi-studi mutakhir seperti Wang, et al., (2021) dan Zhao et al. (2023). Ketiganya menggunakan entropi nol-order untuk mengevaluasi *preprocessing* citra, validasi struktur redundansi, dan perancangan strategi *entropy coding*.

2.1.5.3 Mean Squared Error (MSE)

Mean Squared Error (MSE) merupakan salah satu metrik kuantitatif yang digunakan untuk mengukur perbedaan antara citra asli dengan citra hasil rekonstruksi setelah proses kompresi dan dekompresi. MSE menghitung nilai rata-rata dari kuadrat selisih antara pasangan piksel yang bersesuaian dalam dua citra berdimensi sama. Dalam konteks kompresi citra UINT16, khususnya citra SAR, MSE sangat penting untuk mengevaluasi sejauh mana informasi visual dan struktural tetap dipertahankan setelah data mengalami proses kompresi berbasis transformasi dan *encoding*, seperti IWT, *Delta Encoding*, dan Zstd (Zhong, et al., 2019).

Secara matematis, MSE didefinisikan dengan rumus sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i,j) - \hat{I}(i,j)]^2 \quad (2.19)$$

dengan $I(i,j)$ adalah nilai intensitas piksel asli pada posisi baris ke- i dan kolom ke- j , $\hat{I}(i,j)$ adalah nilai intensitas piksel hasil rekonstruksi, dan $M \times N$ adalah dimensi citra.

Nilai MSE yang kecil menunjukkan bahwa citra hasil kompresi sangat mendekati citra asli, yang berarti proses kompresi memiliki distorsi yang sangat rendah. Dalam pengolahan citra SAR, hal ini sangat krusial karena informasi visual seperti pola *speckle*, perubahan intensitas permukaan, dan detil kecil lainnya berpengaruh langsung terhadap hasil analisis strategis seperti klasifikasi medan, deteksi objek, atau interferometri (Wang, et al., 2022). Oleh karena itu, evaluasi performa kompresi menggunakan MSE tidak hanya bersifat teknis, tetapi juga aplikatif terhadap keputusan sistem dalam domain pertahanan dan penginderaan jauh.

Penggunaan MSE juga memungkinkan perbandingan langsung antara berbagai konfigurasi metode kompresi. Misalnya, perubahan level kompresi pada Zstd atau penghilangan salah satu *subband* dalam IWT dapat segera dievaluasi dampaknya terhadap kualitas citra melalui perubahan nilai MSE. Metrik ini bersifat deterministik dan konsisten, sehingga sangat cocok untuk digunakan dalam eksperimen kuantitatif dan sistem pengambilan keputusan otomatis yang memerlukan ambang batas toleransi terhadap kehilangan informasi.

Sebagai metrik error berbasis piksel, MSE bersifat objektif dan mudah diimplementasikan dalam sistem evaluasi otomatis, baik untuk proses kompresi batch dalam server maupun untuk transmisi citra *real-time* dari platform seperti UAV atau satelit. Evaluasi dengan MSE juga mendasari perhitungan metrik turunan lain seperti *Peak Signal-to-Noise Ratio* (PSNR), yang menggunakan nilai MSE sebagai variabel utama dalam pengukurannya (Sara, et al., 2019). Oleh karena itu, pemahaman dan

penerapan MSE dalam *pipeline* kompresi citra SAR merupakan bagian penting dalam menjaga keseimbangan antara efisiensi data dan kualitas informasi yang dipertahankan.

2.1.5.4 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) adalah metrik evaluasi kualitas citra yang diturunkan langsung dari nilai MSE, dan digunakan untuk mengukur tingkat degradasi yang terjadi akibat proses kompresi (Horé & Ziou, 2010). PSNR sangat penting dalam konteks pengolahan citra UINT16 seperti SAR, karena memberikan indikator numerik terhadap seberapa akurat citra hasil dekompresi dibandingkan dengan citra aslinya. Dalam sistem kompresi *near-lossless*, nilai PSNR yang tinggi menunjukkan bahwa proses kompresi berhasil mempertahankan struktur dan detail citra, bahkan ketika data telah melalui tahapan transformasi seperti *Integer Wavelet Transform* (IWT), *Delta Encoding*, dan *entropy coding* seperti *Zstandard* (Zstd).

Secara matematis, PSNR didefinisikan sebagai:

$$PSNR = 10 \log_2 \left(\frac{MAX_I^2}{MSE} \right) \quad (2.20)$$

di mana MAX_I adalah nilai maksimum dari intensitas piksel dalam citra. Untuk citra UINT16, nilai maksimum $MAX_I = 65535$, karena data disimpan dalam 16-bit unsigned *integer*. MSE dalam rumus ini adalah nilai MSE yang telah dihitung antara citra asli dan citra hasil rekonstruksi.

Nilai PSNR dinyatakan dalam satuan desibel (dB), dan semakin tinggi nilai PSNR, maka semakin kecil perbedaan antara citra asli dan hasil rekonstruksi. Dalam praktik pengolahan citra SAR, nilai PSNR di atas 40 dB dianggap sangat baik, karena perbedaan visual antar citra hampir tidak terlihat oleh mata manusia maupun sistem analitik berbasis tekstur atau statistik permukaan (Mitra et al., 2020).

Penggunaan PSNR sangat penting terutama dalam eksperimen pengujian performa kompresi karena memberikan metrik evaluatif yang

bersifat objektif dan langsung berkaitan dengan kualitas sinyal. Metrik ini juga memungkinkan perbandingan performa antar algoritma kompresi atau antar konfigurasi sistem seperti variasi level *Zstd*, jumlah *subband* yang dikompresi, atau efektivitas filter *speckle* sebelum transformasi. Dalam sistem otomatis berbasis UAV atau pengamatan militer *real-time*, PSNR digunakan sebagai bagian dari *pipeline* validasi untuk memastikan bahwa data yang diterima masih layak untuk analisis taktis atau klasifikasi objek permukaan (Ezuma, et al., 2020).

PSNR juga memiliki keunggulan dalam hal kesederhanaan implementasi, karena cukup menghitung MSE terlebih dahulu, kemudian diaplikasikan ke dalam rumus logaritmik. Ini menjadikannya efisien untuk diintegrasikan ke dalam sistem pengolahan citra *real-time* dan sangat berguna dalam komparasi antar metode kompresi *lossless*, *near-lossless*, dan *lossy*, tanpa memerlukan parameter tuning atau interpretasi kompleks. Dengan demikian, PSNR menjadi metrik standar dalam menilai kinerja algoritma kompresi citra SAR, memastikan bahwa efisiensi data tidak mengorbankan akurasi visual dan struktural yang krusial untuk aplikasi pertahanan dan pengawasan berbasis radar.

2.2 Hasil Penelitian Terdahulu

Beberapa penelitian sebelumnya tentang kompresi citra SAR yang menjadi acuan dalam skripsi ini disajikan dalam tabel 2.2.

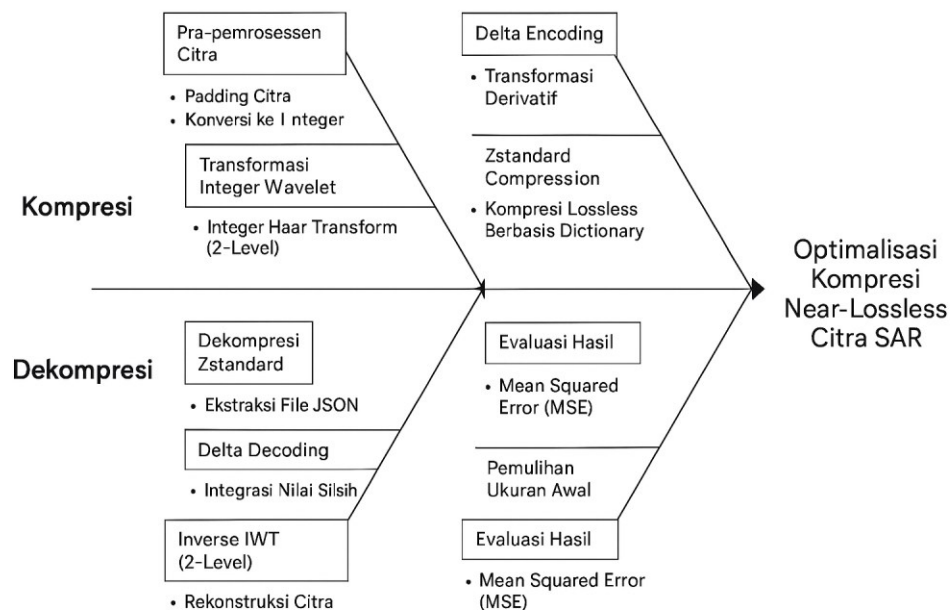
Tabel 2.2 Hasil Penelitian Terdahulu.

No.	Title	Result	Similiarity
1	<i>Compression of Medical Images Using Wavelet Transform and Metaheuristic Algorithm for Telemedicine Applications</i> (Shyamala & Geetha, 2022)	MGOA-IWT mempertahankan detail penting dan menunjukkan kompresi yang unggul di berbagai pencitraan medis.	Kompresi citra berbasis IWT

No.	Title	Result	Similiarity
2	<i>Image Compression and Encryption Based on Integer Wavelet Transform and Hybrid Hyperchaotic System</i> (Srinivas & Mayur, 2022)	IWT efektif meminimalkan data gambar dengan mengubah data gambar menjadi ringkas	Kompresi citra berbasis IWT
3	<i>Improved intensity rounding and division near lossless image compression algorithm using Delta Encoding</i> (Qerom, 2024)	<i>Delta Encoding</i> -RIFD, menghasilkan peningkatan rasio kompresi dan waktu pemrosesan sambil mempertahankan kualitas gambar yang tinggi	Penggunaan <i>Delta Encoding</i> untuk mengurangi intensitas gambar
4	<i>Efficient Container Image Updating in Low-bandwidth Networks with Delta Encoding</i> (Matsumoto, Kotani & Okabe, 2023)	Pengembangan DeltaMerging untuk pembuatan delta yang lebih optimal, sehingga menghasilkan pengurangan ukuran sebesar 5 hingga 40%	Pengembangan <i>Delta Encoding</i> untuk mengurangi kompleksitas data
5	<i>Zstandard Compression and the application/zstd Media Type</i> (Collet, Kucherawy, 2018)	Menggambarkan mekanisme dan mendaftarkan jenis media untuk konten yang dikompresi zstd.	Penggunaan Zstd untuk kompresi data
6	<i>Nonnegative and local linear regression for</i>	Paper ini mengimplementasikan	teknik regresi linier untuk

No.	Title	Result	Similiarity
	<i>classification in SAR imagery</i> (Dong et al., 2014)	regresi linier dengan batasan nonnegatif dan lokalitas untuk klasifikasi objek dalam citra SAR, memperlihatkan bahwa pendekatan regresi linier cukup efektif untuk memodelkan hubungan antar fitur citra.	mengonstruksi representasi citra SAR

2.3 Kerangka Pemikiran



Gambar 2.5 Diagram Tulang Ikan Kerangka Pemikiran

Ukuran data citra SAR berformat .TIF UINT16 sangat besar karena mencerminkan dinamika spektral dan spasial kompleks dengan representasi 16-bit. Permasalahan ini menjadi hambatan utama dalam transmisi dan analisis *real-time*, terutama dalam sistem pemantauan berbasis UAV, satelit militer, maupun sistem komando taktis berbasis bandwidth terbatas. Oleh karena itu, dibutuhkan metode kompresi yang

bersifat *near-lossless*, yaitu mampu mengurangi ukuran data secara signifikan tanpa kehilangan informasi penting secara analitik.

Penelitian ini menggunakan tiga jenis lokasi citra *Sentinel-1A* SAR, yaitu Jawa Barat, Laut Cina Selatan, dan Pangkal Pinang, yang masing-masing terdiri atas dua polarisasi radar: VH dan VV. Citra-citra ini diunduh dalam format .TIF UINT16 dengan resolusi penuh dari produk level-1 dan memiliki ukuran *file* yang besar. Untuk memudahkan proses pemrosesan di perangkat komputer dengan spesifikasi terbatas, setiap citra kemudian dibagi menjadi empat bagian: *split_1*, *split_2*, *split_3*, dan *split_4*. Pemisahan ini memungkinkan setiap bagian diproses secara efisien tanpa mengorbankan akurasi analisis.

Kerangka pemikiran dibagi menjadi dua alur besar, yaitu proses kompresi dan dekompresi, yang terintegrasi dalam satu sistem optimasi kompresi citra SAR.

Pada sisi kompresi, proses diawali dengan pra-pemrosesan citra, meliputi *padding* untuk memastikan ukuran habis dibagi 8 dan konversi nilai ke bilangan bulat (*integer*) agar kompatibel dengan transformasi *wavelet*. Selanjutnya, dilakukan IWT dua tingkat untuk mendekomposisi citra menjadi *subband* frekuensi rendah dan tinggi, di mana komponen LL (*Low-Low*) menyimpan informasi dominan.

Setelah transformasi *wavelet*, *Delta Encoding* digunakan untuk merepresentasikan perubahan nilai piksel antar elemen data dalam bentuk nilai selisih. Metode ini mengurangi variabilitas antar piksel, memfasilitasi pengkodean yang lebih efisien pada tahap berikutnya. Tahap akhir dari sisi kompresi adalah *Zstd*, yang bekerja berdasarkan teknik kompresi *lossless* berbasis kamus (*dictionary-based*) dan model *entropy coding* yang fleksibel, menghasilkan *bitstream* dengan efisiensi tinggi tanpa mengorbankan integritas data.

Dari sisi dekompresi, *bitstream* yang telah dikompresi mengalami proses dekompresi *Zstandard*, diikuti dengan ekstraksi informasi dari *file* JSON yang menyimpan blok delta yang telah dikodekan. Data yang

diperoleh kemudian diproses melalui *Delta decoding*, yakni rekonstruksi data asli dengan integrasi nilai selisih. Setelah itu, dilakukan proses *Inverse IWT* dua tingkat untuk merekonstruksi citra dari *subband* ke bentuk semula.

Akhir dari proses ini adalah evaluasi hasil, yang dilakukan dua kali: pertama untuk mengukur kualitas pasca kompresi, dan kedua setelah pemulihan ukuran asli citra. Evaluasi dilakukan menggunakan metrik kuantitatif MSE, untuk mengukur deviasi nilai piksel asli terhadap hasil rekonstruksi. Nilai MSE yang rendah menunjukkan bahwa sistem telah berhasil melakukan optimalisasi kompresi *near-lossless* terhadap citra SAR.

Seluruh tahapan ini dirancang untuk menghasilkan keseimbangan antara efisiensi ukuran dan ketelitian visual, dengan mempertahankan struktur *speckle* dan informasi radiometrik penting dalam citra radar. Dengan demikian, proses kompresi yang dikembangkan tidak hanya bersifat teknis, tetapi juga strategis dalam mendukung pengambilan keputusan berbasis citra SAR pada sistem pertahanan dan keamanan nasional.