

2. TEORI PENUNJANG

2.1 Tinjauan Pustaka

2.1.1 *Perceptual hashing*

Perceptual hashing, atau *hash* perseptual, adalah metode yang digunakan untuk menghasilkan representasi numerik yang unik untuk suatu objek berdasarkan karakteristik *perceptual*nya, seperti gambar, suara, atau video. *Perceptual hashing* digunakan dalam berbagai aplikasi, termasuk pengenalan konten, deteksi duplikat, pencarian konten, dan *watermarking digital* (*Perceptual hashing*, 2021). Metode ini bekerja dengan menerapkan transformasi matematis kompleks pada data asli untuk menghasilkan *hash* yang tahan terhadap perubahan kecil dalam data asli. *Perceptual hashing* dapat digunakan untuk membandingkan objek secara efisien, karena perbedaan kecil dalam data asli akan menghasilkan perbedaan yang signifikan dalam *hash*. Salah satu keuntungan utama dari *Perceptual hashing* adalah kemampuannya untuk mengidentifikasi objek yang serupa meskipun ada perbedaan yang signifikan dalam data asli mereka. Metode ini telah digunakan dalam berbagai aplikasi, termasuk pengenalan wajah, pencocokan sidik jari, dan pencarian gambar. Dalam beberapa tahun terakhir, *Perceptual hashing* telah menjadi topik penelitian yang semakin populer dalam bidang pengolahan citra dan multimedia. Berikut adalah langkah-langkah dari *perceptual hashing* (Hamadouche et al., 2021):

- **Resizing:** Gambar diubah ukurannya menjadi matriks 32×32 piksel.
- **Grayscale Conversion:** Gambar yang diperoleh kemudian diubah ke ruang warna skala abu-abu.
- **DCT:** Transformasi DCT (Discrete Cosine Transform) ukuran 32×32 dilakukan pada gambar skala abu-abu untuk mendapatkan matriks koefisien DCT 32×32 , di mana energi gambar akan terkumpul ke dalam beberapa koefisien DCT frekuensi rendah.
- **Vector Construction:** Vektor sepanjang 64 dibuat dengan menggabungkan koefisien DCT dari (1,1), yang sesuai dengan sudut kiri atas dari matriks ukuran 64, hingga koefisien (8,8), yang mewakili sudut kanan bawah.
- **Mean Computation:** Rata-rata dari array koefisien yang dihasilkan dihitung.
- **Comparison:** Perbandingan 64 koefisien DCT dengan nilai rata-rata dilakukan, di mana bit *hash* diatur ke 1 jika koefisien lebih besar dari nilai rata-rata, dan 0 sebaliknya.
- **Final Hash:** Akhirnya, *hash* biner 64 bit diperoleh.

2.1.2 *Digital art*

Seni *digital* atau biasa disebut *digital art* adalah sebuah bentuk seni yang menggunakan teknologi *digital* untuk menciptakan atau memanipulasi gambar, suara, atau media lain. Seni ini telah menjadi semakin populer dalam beberapa tahun terakhir karena ketersediaan alat dan platform *digital* yang telah mengubah penciptaan seni dan memberikan peluang yang luas bagi para seniman. *Digital art* juga dianggap sebagai cara untuk menyatukan kesenian tradisional dan modern. Namun, terdapat kesalahan umum yang sering dilakukan oleh pemula dalam *digital art*, seperti tidak memahami perangkat lunak, tidak menggunakan `\textit{layers}`, dan kurang berlatih. Untuk menjual *digital art*, seseorang dapat membuat *template* dengan Canva dan menjualnya di berbagai platform seperti WordPress, Shopify, dan Creative Market. Secara keseluruhan, *digital art* memiliki masa depan yang menjanjikan seiring dengan terus majunya teknologi dan semakin banyaknya minat masyarakat terhadap bentuk seni ini (Karya Seni *Digital* "Menjelajahi Dunia Karya Seni *Digital* Teknik, Sejarah, Dan Dampak", 2023).

2.1.3 Manipulasi Gambar

Manipulasi gambar adalah proses mengubah atau memanipulasi gambar *digital* untuk menciptakan efek visual yang diinginkan. Teknik ini dapat dilakukan dengan menggunakan perangkat lunak pengolah gambar seperti Adobe Photoshop, GIMP, atau CorelDRAW. Beberapa teknik manipulasi gambar yang umum digunakan antara lain:

- **Pengaturan warna dan kontras:** Mengubah tingkat kecerahan, kontras, dan keseimbangan warna untuk meningkatkan tampilan gambar.
- **Pemotongan dan pepadatan:** Menghapus bagian gambar yang tidak diinginkan atau menggabungkan beberapa gambar menjadi satu.
- **Pengaburan dan penajaman:** Membuat gambar terlihat lebih tajam atau lebih lembut dengan mengatur tingkat ketajaman dan mengaburkan detail yang tidak diinginkan.
- **Penghapusan objek:** Menghapus objek yang mengganggu dari gambar dengan menggunakan alat pemilihan dan alat pengeditan.
- **Penggabungan gambar:** Menggabungkan beberapa gambar menjadi satu untuk menciptakan efek komposit atau kolase.

- **Pemberian efek khusus:** Menambahkan efek visual seperti tekstur, bayangan, atau pencahayaan yang tidak ada dalam gambar asli.

Manipulasi gambar dapat digunakan dalam berbagai bidang, termasuk desain grafis, fotografi, dan seni *digital*. Dalam beberapa tahun terakhir, perkembangan teknologi dan perangkat lunak pengolah gambar telah memungkinkan para pengguna untuk melakukan manipulasi gambar dengan lebih mudah dan cepat (*Pedoman Penulisan*, 2019). Namun, penting untuk diingat bahwa manipulasi gambar juga dapat digunakan untuk tujuan yang tidak etis, seperti memalsukan informasi atau mengedit gambar dengan cara yang merugikan orang lain. Oleh karena itu, penggunaan teknik manipulasi gambar harus dilakukan dengan bijaksana dan bertanggung jawab.

2.1.4 Ambang Batas

Ambang batas adalah nilai batas yang digunakan untuk mengukur atau membatasi suatu kondisi, kuantitas, atau kualitas tertentu. Dalam konteks *perceptual hashing*, *threshold* adalah nilai batas yang digunakan untuk mengukur tingkat kesamaan antara dua *hash* yang mewakili gambar (*Perceptual hashing*, 2021). *Perceptual hashing* adalah teknik yang digunakan untuk mencari gambar yang identik atau sangat mirip dalam sejumlah besar foto. Beberapa hal yang perlu diperhatikan mengenai *threshold* dalam konteks ini antara lain: panjang *hash* yang dipilih harus dapat dibagi dengan 4, untuk gambar yang sangat mirip, *hash* yang lebih panjang akan memberikan akurasi yang lebih tinggi, jarak *Hamming* maksimum antara dua *hash*, ketika gambar benar-benar berbeda, sama dengan panjang *hash* pangkat 2, dan saat membandingkan *hash perceptual*, *hash-hash* harus memiliki panjang yang sama. Dalam konteks *perceptual hashing*, *threshold* digunakan untuk membatasi tingkat kesamaan antara dua *hash* yang mewakili gambar, sehingga dapat membantu dalam mencari gambar yang identik atau sangat mirip dalam sejumlah besar foto.

2.2 Tinjauan Studi

2.2.1 *Analysis of Perceptual hashing Algorithms in Image Manipulation Detection (Samanta & Jain, 2021)*

Dalam penelitian "*Analysis of Perceptual hashing Algorithms in Image Manipulation Detection*" yang dilakukan oleh Samanta dan Jain pada tahun 2021, terdapat beberapa masalah, metode, dan hasil yang dapat dirangkum. Penelitian ini bertujuan untuk menganalisis kinerja

algoritma *perceptual hashing* dalam mendeteksi manipulasi gambar, namun hasilnya menunjukkan kinerja yang kurang ideal dalam membedakan antara gambar yang dimanipulasi dengan sengaja dan gambar yang asli. Beberapa masalah yang dihadapi dalam penelitian ini antara lain:

- Keterbatasan algoritma *perceptual hashing* dalam mengatasi variasi manipulasi gambar yang kompleks dan canggih
- Ketidakmampuan algoritma *perceptual hashing* untuk menghasilkan representasi *hash* yang unik untuk setiap gambar
- Ketergantungan kinerja algoritma *perceptual hashing* pada kualitas gambar asli dan tingkat manipulasi yang dilakukan
- Tidak adanya metode yang efektif untuk menentukan ambang batas yang optimal dalam mendeteksi manipulasi gambar menggunakan algoritma *perceptual hashing*

Dalam penelitian ini, digunakan empat algoritma *perceptual hashing* yang berbeda, yaitu *Perceptual hashing (p-hash)*, *Wavelet hashing (w-hash)*, *Non-Negative Matrix Factorization (NMF)*, *Singular Value Decomposition (SVD)*. Berikut adalah Langkah-langkah proses *P-hash*:

- Transformasi DCT: Transformasi kosinus diskrit diterapkan pada gambar 32x32 piksel untuk mendapatkan matriks koefisien 32x32.
- Pemilihan Koefisien Frekuensi Rendah: Koefisien dari sudut kiri atas matriks, yaitu 8x8 koefisien frekuensi rendah untuk perhitungan *hash* akhir
- Kuantisasi Koefisien: Setiap koefisien dalam matriks 8x8 ini dikuantisasi dengan membandingkannya dengan nilai median
- *Hash* Akhir: Hasil dari proses ini adalah string *hash* biner sepanjang 64 bit.

Hasil evaluasi menunjukkan bahwa tidak ada algoritma yang secara konsisten unggul dalam mendeteksi manipulasi gambar, dan kinerja mereka sangat bergantung pada jenis manipulasi yang dilakukan, tetapi hasil menunjukkan bahwa *perceptual hashing* memiliki keunggulan. Dalam penelitian ini, juga ditemukan bahwa penggunaan ambang batas yang tepat dapat meningkatkan kinerja algoritma *perceptual hashing* dalam mendeteksi manipulasi gambar.

2.2.2 Overview of Perceptual hashing Technology (Overview of Perceptual hashing Technology Contents, 2022)

Penelitian "Overview of *Perceptual hashing* Technology" yang diselenggarakan oleh Ofcom mencakup sejumlah aspek penting. Masalah yang diidentifikasi dalam penelitian ini terkait dengan teknologi *hashing* berbasis persepsi. Penelitian ini memfokuskan perhatian pada evaluasi kinerja teknologi *hashing* persepsi, terutama dalam hal pengakurasiannya terhadap variasi perubahan citra dan kemampuannya untuk membedakan antara item yang serupa tetapi memiliki perubahan yang signifikan. Metode penelitian didokumentasikan dengan cermat, mencakup pendekatan penggunaan *Discrete Cosine Transform* (DCT) sebagai dasar dalam pembuatan *hash*. Hasil dari penelitian ini memberikan wawasan yang dalam tentang teknologi *hashing* berbasis persepsi dan penerapannya, terutama dalam hal pengenalan gambar, pencarian gambar serupa, dan keamanan *digital*. Penelitian ini memberikan kontribusi berharga dalam pemahaman teknologi *hashing* berbasis persepsi dan aplikasinya di berbagai bidang, termasuk perbandingan dengan *hashing* lainnya yaitu *average hashing*.

2.2.3 Comparison Detection Edge Lines Algoritma Canny dan Sobel (Resianta Perangin-angin & Julia, 2019)

Masalah yang dibahas di jurnal ini adalah perbandingan antara dua algoritma deteksi tepi, yaitu Algoritma Canny dan Sobel. Masalah yang diajukan dalam penelitian ini adalah menilai kinerja dan akurasi kedua algoritma tersebut dalam mengidentifikasi tepi pada citra *digital*. Metode penelitian melibatkan pengujian dan perbandingan kinerja Algoritma Canny dan Sobel pada berbagai jenis citra *digital*. Hasil penelitian ini memberikan pemahaman yang lebih dalam tentang efektivitas masing-masing algoritma dalam mendeteksi tepi dalam berbagai konteks dan dapat digunakan sebagai pedoman dalam pemilihan algoritma yang paling sesuai untuk aplikasi tertentu. Penelitian ini menyumbang pemahaman yang penting dalam pengolahan citra *digital* dan analisis tepi.

2.2.4 Penerapan Digital Watermark Sebagai Validasi Keabsahan Gambar Digital Dengan Skema Blind Watermark (Suheryadi, 2017)

Artikel ini membahas penggunaan transformasi *wavelet* dan penyisipan *threshold* untuk menyisipkan data pada gambar *digital*. Data disisipkan ke dalam bit-plane least significant bit (LSB) dari koefisien *wavelet* integer CDF frekuensi tinggi yang besarnya lebih kecil dari ambang batas yang telah ditentukan sebelumnya. Modifikasi histogram diterapkan sebagai

preprocessing untuk mencegah overflow/underflow. Evaluasi hasil percobaan menunjukkan bahwa skema ini lebih unggul dalam bentuk muatan yang lebih besar atau PSNR yang lebih tinggi. Artikel ini membahas konsep invisible *watermarking* dengan skema blind dan menggunakan secret key untuk memverifikasi karya *digital* berupa gambar. Tinjauan pustaka dalam artikel ini menjelaskan teknik penyisipan *watermark* secara *digital*, terutama dalam domain frekuensi dan spatio-domain. Metode penyisipan invisible *watermark* yang umum digunakan, seperti least-significant bit (LSB) dan histogram-based, juga dibahas. Selain itu, artikel ini membahas penggunaan transformasi *wavelet* dalam *digital watermarking* dan pentingnya penggunaan secret key dan public key untuk autentikasi dan verifikasi kepemilikan karya *digital* berupa gambar.

2.2.5 A Comparative study of *perceptual hashing* algorithms: Application on fingerprint images (Hamadouche et al., 2021)

Dalam studi ini, penulis membandingkan berbagai algoritma *hashing* perseptual dengan fokus pada penerapannya pada gambar sidik jari. *Hashing* perseptual adalah teknik penting dalam keamanan informasi dan forensik *digital* karena memungkinkan verifikasi integritas dan kesamaan gambar secara efisien. Studi ini menguji dan menganalisis kinerja beberapa algoritma *hashing* dalam konteks gambar sidik jari untuk menilai keandalan dan efektivitasnya.

Penulis melakukan perbandingan beberapa algoritma *hashing* perseptual yang mencakup P-Hash (*Perceptual Hash*), A-Hash (*Average Hash*), D-Hash (*Difference Hash*), dan R-Hash (*Radial Hash*). Setiap algoritma diuji dengan menggunakan kumpulan data gambar sidik jari dan dievaluasi berdasarkan kriteria seperti kecepatan komputasi, ketahanan terhadap manipulasi gambar, dan kemampuan membedakan gambar asli dari yang telah dimodifikasi. Berikut Langkah-langkah penting dari algoritma utama *perceptual hashing* di antara lain:

- Resizing: Gambar diubah ukurannya menjadi matriks 32×32 piksel.
- Grayscale Conversion: Gambar yang diperoleh kemudian diubah ke ruang warna skala abu-abu.
- DCT: Transformasi DCT (Discrete Cosine Transform) ukuran 32×32 dilakukan pada gambar skala abu-abu untuk mendapatkan matriks koefisien DCT 32×32 , di mana energi gambar akan terkumpul ke dalam beberapa koefisien DCT frekuensi rendah.

- **Vector Construction:** Vektor sepanjang 64 dibuat dengan menggabungkan koefisien DCT dari (1,1), yang sesuai dengan sudut kiri atas dari matriks ukuran 64, hingga koefisien (8,8), yang mewakili sudut kanan bawah.
- **Mean Computation:** Rata-rata dari array koefisien yang dihasilkan dihitung.
- **Comparison:** Perbandingan 64 koefisien DCT dengan nilai rata-rata dilakukan, di mana bit *hash* diatur ke 1 jika koefisien lebih besar dari nilai rata-rata, dan 0 sebaliknya.
- **Final Hash:** Akhirnya, *hash* biner 64 bit diperoleh.

Studi ini menemukan bahwa beberapa algoritma memiliki keunggulan yang berbeda-beda. *P-Hash*, misalnya, menunjukkan ketahanan yang baik terhadap manipulasi gambar kecil tetapi membutuhkan waktu komputasi yang lebih lama dibandingkan algoritma lain. *A-Hash* dan *D-Hash*, di sisi lain, lebih cepat tetapi kurang tahan terhadap perubahan kecil dalam gambar.

Hasil perbandingan menunjukkan bahwa tidak ada satu pun algoritma yang unggul dalam semua aspek. Pilihan algoritma terbaik sangat tergantung pada kebutuhan spesifik aplikasi, seperti kecepatan atau ketahanan terhadap manipulasi. Misalnya, untuk aplikasi yang membutuhkan deteksi cepat, *A-Hash* atau *D-Hash* mungkin lebih cocok, sementara untuk aplikasi yang memerlukan ketahanan yang lebih tinggi, *P-Hash* mungkin lebih baik.