Deteksi Manipulasi Foto Profil KTP Menggunakan Perceptual Hashing

Willhelmina Rachel Silalahi -18222049
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 18222049@std.stei.itb.ac.id

***Abstrak*—**Makalah ini membahas implementasi perceptual hashing untuk mendeteksi manipulasi pada foto profil KTP yang diunggah ke sistem verifikasi identitas. Masalah utama yang dikaji adalah keterbatasan cryptographic hash seperti SHA-256 yang hanya membandingkan kesamaan byte file sehingga sangat sensitif terhadap kompresi, resize, dan perubahan metadata, tetapi tidak memberikan ukuran kemiripan visual. Sistem yang dirancang menggunakan area crop foto profil, konversi grayscale, pembentukan hash 64-bit dengan difference hash (dHash) dan perceptual hash (pHash), serta pengambilan keputusan berdasarkan jarak Hamming. Eksperimen dilakukan menggunakan citra KTP simulasi dengan beberapa skenario perubahan, meliputi kompresi JPEG, peningkatan kecerahan, peningkatan kontras, resize, crop kecil, blur ringan, rotasi, penggantian foto, dan perubahan teks di luar area foto. Hasil menunjukkan bahwa SHA-256 selalu berubah untuk manipulasi byte sekecil apa pun, sedangkan dHash dan pHash lebih stabil terhadap perubahan visual ringan. Manipulasi signifikan seperti rotasi dan penggantian foto menghasilkan jarak Hamming lebih tinggi dari threshold. Dengan demikian, pHash dan dHash dapat digunakan sebagai komponen awal deteksi manipulasi foto profil, namun tetap membutuhkan validasi tambahan untuk kasus adversarial dan data nyata.

***Kata kunci*—**KTP digital; perceptual hashing; pHash; dHash; SHA-256; jarak Hamming; deteksi manipulasi citra.

I. PENDAHULUAN

Penggunaan Kartu Tanda Penduduk (KTP) dalam layanan digital meningkat seiring berkembangnya proses pendaftaran akun, pembukaan rekening, peminjaman daring, dan berbagai layanan administrasi elektronik. Pada proses tersebut, pengguna sering diminta mengunggah foto KTP atau citra dokumen identitas. Salah satu risiko yang muncul adalah manipulasi foto profil pada KTP, misalnya mengganti foto pemilik dokumen, mengubah sebagian area foto, atau melakukan penyuntingan yang bertujuan mengelabui sistem verifikasi awal.

Pendekatan paling langsung untuk memeriksa perubahan file adalah cryptographic hash. SHA-256, misalnya, menghasilkan message digest dengan sifat avalanche, sehingga perubahan kecil pada byte input akan menghasilkan digest yang sangat berbeda. Sifat ini berguna untuk integritas file secara eksak, tetapi tidak selalu sesuai untuk kasus citra digital yang mengalami transformasi wajar seperti kompresi JPEG, resize, perubahan brightness, atau perubahan metadata. Dua gambar yang secara visual sama dapat memiliki nilai SHA-256 berbeda total karena representasi byte-nya berubah [1].

Masalah tersebut mendorong penggunaan perceptual hashing. Berbeda dengan cryptographic hash, perceptual hash dirancang agar citra yang tampak mirip menghasilkan nilai hash yang juga mirip. Dengan demikian, kesamaan tidak ditentukan oleh equality byte, melainkan oleh jarak Hamming antar-hash. Algoritma seperti dHash dan pHash banyak digunakan untuk near-duplicate image detection, image retrieval, dan pemeriksaan kemiripan citra [2], [3].

Makalah ini mengusulkan dan mengimplementasikan sistem deteksi manipulasi foto profil KTP berbasis dHash dan pHash. Fokus sistem bukan autentikasi legal dokumen KTP, melainkan deteksi awal apakah area foto profil pada KTP yang diunggah masih konsisten secara visual terhadap gambar referensi. Kontribusi makalah ini adalah:

- merancang pipeline deteksi berbasis crop area foto profil KTP;
- mengimplementasikan SHA-256, dHash, pHash, dan jarak Hamming dalam program Python;
- menguji sistem pada beberapa skenario manipulasi citra;
- membandingkan sensitivitas SHA-256 dengan toleransi visual dHash dan pHash;
- menganalisis batasan sistem untuk penggunaan pada data identitas digital.

II. DASAR TEORI

A. Cryptographic Hash SHA-256

Cryptographic hash function memetakan pesan berukuran sembarang ke digest berukuran tetap. SHA-256 adalah bagian dari keluarga Secure Hash Algorithm yang distandardisasi dalam FIPS 180-4. Fungsi ini dirancang untuk kebutuhan integritas dan keamanan kriptografi, antara lain preimage resistance, second preimage resistance, dan collision resistance pada tingkat keamanan tertentu [1].

Dalam konteks gambar KTP, SHA-256 dapat dipakai untuk memastikan apakah dua file identik secara byte-level. Namun, perubahan format penyimpanan, metadata, kompresi ulang, atau resize akan mengubah byte file, sehingga digest menjadi berbeda meskipun gambar masih tampak sama bagi manusia. Oleh karena itu, SHA-256 lebih tepat diposisikan sebagai baseline integritas eksak, bukan sebagai alat ukur kemiripan visual.

B. Perceptual Hashing

Perceptual image hashing menghasilkan sidik jari digital berdasarkan karakteristik visual citra. Tujuannya berbeda dari cryptographic hash. Pada perceptual hashing, kemiripan visual diharapkan menghasilkan hash yang

memiliki perbedaan bit rendah. Zauner menjelaskan bahwa perceptual image hash dapat digunakan untuk identifikasi dan verifikasi integritas citra berdasarkan persepsi, dengan menyeimbangkan *robustness* terhadap transformasi ringan dan discriminability terhadap citra berbeda [2].

Perceptual hashing tidak memiliki sifat keamanan yang sama dengan cryptographic hash. Nilai hash 64-bit pada dHash atau pHash tidak dimaksudkan untuk tahan terhadap serangan kriptografis. Fungsi ini lebih tepat digunakan sebagai fitur visual untuk deteksi near-duplicate atau perubahan signifikan pada konten gambar.

C. Difference Hash

Difference hash atau dHash membentuk hash berdasarkan pola gradien horizontal piksel. Citra diubah menjadi grayscale, diresize menjadi ukuran 9x8, kemudian setiap piksel dibandingkan dengan piksel di sebelah kanannya. Jika piksel kanan lebih terang, bit diisi 1; jika tidak, bit diisi 0. Proses ini menghasilkan 64 bit untuk hash_size 8. Karena hanya mempertimbangkan arah perubahan intensitas, dHash relatif cepat dan cukup stabil terhadap beberapa perubahan ringan [3].

D. Perceptual Hash Berbasis DCT

pHash pada makalah ini menggunakan Discrete Cosine Transform (DCT). Citra diubah menjadi grayscale dan di *resize* menjadi 32x32, kemudian dilakukan DCT dua dimensi. Komponen frekuensi rendah berukuran 8x8 diambil karena merepresentasikan struktur global citra. Nilai koefisien dibandingkan dengan median untuk menghasilkan bit hash. Pendekatan berbasis DCT lazim digunakan karena perubahan ringan pada kompresi dan brightness tidak selalu mengubah struktur frekuensi rendah secara drastis [2], [5].

E. Jarak Hamming

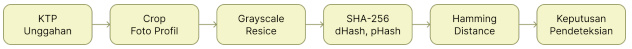
Jarak Hamming digunakan untuk mengukur jumlah bit yang berbeda antara dua hash. Untuk dua hash biner h1 dan h2, jarak Hamming dapat dihitung dengan operasi XOR dan popcount:

$$D(h1,h2) = \text{popcount}(h1 \text{ XOR } h2)$$

Semakin kecil jarak Hamming, semakin dekat representasi visual kedua gambar. Pada eksperimen ini digunakan threshold awal 10 dari 64 bit. Nilai tersebut dipilih sebagai parameter eksperimen sederhana, bukan sebagai standar universal.

III. RANCANGAN SISTEM

Sistem dirancang untuk menerima dua gambar: gambar KTP referensi dan gambar KTP uji. Karena objek pemeriksaan adalah foto profil, sistem menyediakan opsi crop dalam format x,y,w,h. Dengan crop, perubahan pada teks atau area lain di luar foto tidak memengaruhi hasil perceptual hash foto profil. Hal ini penting karena pada kasus verifikasi identitas, sistem dapat diarahkan untuk memeriksa bagian yang paling relevan terhadap manipulasi wajah atau foto pemilik dokumen.



Gambar 1. Alur sistem deteksi manipulasi foto profil KTP menggunakan SHA-256, dHash, pHash, dan jarak Hamming.

Tahap pertama adalah praproses. Gambar dibaca dalam format RGB, kemudian jika parameter crop tersedia, hanya area foto profil yang diambil. Untuk dHash dan pHash, area tersebut dikonversi ke grayscale. dHash menggunakan ukuran 9x8, sedangkan pHash menggunakan ukuran 32x32 untuk memperoleh komponen frekuensi rendah. SHA-256 tidak menggunakan praproses visual karena digest dihitung langsung dari byte file asli.

Tahap kedua adalah pembentukan hash. SHA-256 dihitung dengan membaca file dalam blok byte. dHash menghitung perbandingan piksel horizontal. pHash membentuk matriks DCT, mengambil blok 8x8 frekuensi rendah, lalu membandingkan nilai koefisien terhadap median. Tahap ketiga adalah evaluasi: SHA-256 dibandingkan secara equality, sedangkan dHash dan pHash dibandingkan dengan jarak Hamming. Jika jarak Hamming lebih kecil atau sama dengan threshold, gambar diklasifikasikan sebagai mirip. Jika melebihi threshold, gambar diklasifikasikan sebagai berbeda atau terindikasi manipulasi.

Rancangan ini sengaja dibuat modular agar setiap algoritma dapat diuji secara terpisah. Penggunaan threshold juga dibuat eksplisit agar peneliti dapat melakukan kalibrasi berdasarkan karakteristik dataset dan toleransi risiko aplikasi. Untuk penggunaan produksi, threshold seharusnya diperoleh dari validasi data nyata, bukan hanya dari asumsi.

IV. IMPLEMENTASI

Implementasi dibuat menggunakan Python dengan pustaka Pillow, NumPy, dan Pandas. Pillow digunakan untuk membaca, mengubah ukuran, melakukan crop, dan membuat sampel manipulasi. NumPy digunakan untuk operasi matriks, pembentukan DCT, perbandingan piksel, dan operasi numerik. Pandas digunakan untuk menyimpan hasil eksperimen batch ke CSV. Program tidak menggunakan pustaka imagehash siap pakai agar algoritma dHash dan pHash dapat dijelaskan sebagai kontribusi implementasi mandiri.

A. Struktur Program

Program utama terdiri dari fungsi load_image_grayscale, sha256_file, dhash, phash, hamming_distance, compare_images, create_manipulated_samples, dan batch_compare. Fungsi compare_images menjadi penghubung utama karena memanggil ketiga metode hash dan mengembalikan hasil dalam struktur dictionary. Fungsi batch_compare dipakai untuk eksperimen banyak gambar sehingga keluaran dapat dianalisis dalam tabel.

B. Cuplikan Implementasi dHash

```python
def dhash(path: str, crop: CropBox = None, hash_size: int = 8) -> str:
    img = buka_gambar(path, crop=crop, ukuran=(hash_size + 1, hash_size))
    piksel = np.array(img, dtype=np.int16)

    beda = piksel[:, 1:] > piksel[:, :-1]
    return bits_ke_hex(beda)
```

Gambar 2. Cuplikan Implementasi dHash

Cuplikan tersebut menunjukkan bahwa dHash tidak membandingkan piksel absolut, tetapi membandingkan arah perubahan intensitas. Hal ini membuatnya lebih toleran terhadap perubahan global tertentu, misalnya kenaikan brightness yang memengaruhi sebagian besar piksel secara relatif seragam.

C. Cuplikan Implementasi pHash

```python
def phash(path: str, crop: CropBox = None, hash_size: int = 8) -> str:
    ukuran = hash_size * 4

    img = buka_gambar(path, crop=crop, ukuran=(ukuran, ukuran))
    piksel = np.array(img, dtype=np.float64)

    matriks_dct = buat_matriks_dct(ukuran)
    hasil_dct = matriks_dct @ piksel @ matriks_dct.T

    frekuensi_rendah = hasil_dct[:hash_size, :hash_size]
    nilai = frekuensi_rendah.flatten()

    median = np.median(nilai[1:])
    bit_hash = frekuensi_rendah > median

    return bits_ke_hex(bit_hash)
```

Gambar 3. Cuplikan Implementasi pHash

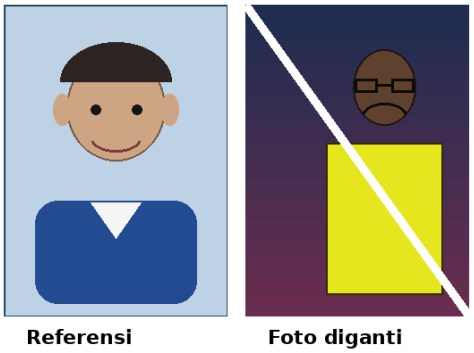
Pada pHash, koefisien DC diabaikan ketika median dihitung karena koefisien tersebut merepresentasikan intensitas rata-rata global. Pengabaian ini membantu mengurangi pengaruh perubahan kecerahan umum terhadap pembentukan hash.

D. Etika dan Keamanan Data

Karena KTP mengandung data pribadi, implementasi ini harus digunakan hanya pada data milik sendiri, data dummy, atau data yang memiliki persetujuan eksplisit. Makalah ini menggunakan citra KTP simulasi, bukan KTP asli. Sistem juga tidak menyimpan NIK atau identitas asli. Dalam penerapan nyata, data KTP harus diproses dengan prinsip minimisasi data, enkripsi penyimpanan, pembatasan akses, dan retensi sesingkat mungkin.

V. SKENARIO EKSPERIMEN

Eksperimen dilakukan menggunakan satu citra KTP simulasi sebagai referensi. Area foto profil berada pada koordinat crop x=65, y=170, w=250, h=350. Dari gambar referensi dibuat sembilan skenario uji: identik, kompresi JPEG kualitas 50, brightness 125%, contrast 130%, resize turun-naik, crop kecil, blur ringan, rotasi 2 derajat, penggantian foto profil, serta perubahan teks di luar area foto. Skenario terakhir dimasukkan untuk menguji apakah crop foto profil mencegah perubahan teks di luar foto memengaruhi hash visual.



Gambar 4. Contoh area crop foto profil pada KTP simulasi: referensi dan skenario foto diganti.

Setiap gambar uji dibandingkan dengan gambar referensi menggunakan tiga metode. Untuk SHA-256, hasil dibandingkan sebagai boolean sama atau tidak sama. Untuk dHash dan pHash, sistem menghitung jarak Hamming 64-bit. Klasifikasi dibuat dengan aturan sederhana: jarak 0 dianggap identik atau sangat mirip, jarak 1 sampai 10 dianggap mirip atau perubahan visual ringan, sedangkan jarak lebih dari 10 dianggap berbeda atau terindikasi manipulasi.

Eksperimen ini tidak dimaksudkan sebagai benchmark komprehensif. Tujuannya adalah menunjukkan perbedaan perilaku antara hash kriptografis dan perceptual hash pada masalah manipulasi citra. Untuk evaluasi produksi, dibutuhkan dataset yang lebih besar, variasi scanner/kamera, kualitas cahaya berbeda, dan contoh manipulasi yang lebih realistis.

VI. HASIL DAN PEMBAHASAN

Eksperimen dilakukan dengan membandingkan gambar referensi dengan sepuluh skenario gambar uji. Setiap gambar diuji menggunakan SHA-256, dHash, dan pHash. SHA-256 digunakan untuk melihat apakah file masih sama secara byte-level, sedangkan dHash dan pHash digunakan untuk mengukur kemiripan visual berdasarkan jarak Hamming. Pada eksperimen ini digunakan threshold jarak Hamming sebesar 10. Artinya, nilai jarak lebih dari 10 dikategorikan sebagai terindikasi berbeda.

Skenario	SHA-256 sama	dHash	Status dHash	pHash	Status pHash
Asli copy	True	0	Identik	0	Identik
Kompresi JPEG Q50	False	0	Identik	0	Identik
Brightness 125%	False	1	Mirip	0	Identik
Contrast 130%	False	2	Mirip	4	Mirip
Resize turun-naik	False	1	Mirip	0	Identik
Crop kecil	False	15	Terindikasi berbeda	14	Terindikasi berbeda
Blur ringan	False	2	Mirip	0	Identik
Rotasi 2 derajat	False	9	Mirip	16	Terindikasi berbeda
Foto diganti simulasi	False	39	Terindikasi berbeda	32	Terindikasi berbeda
Teks luar foto	False	0	Identik	0	Identik

Tabel I. Hasil eksperimen pada area crop foto profil dengan threshold jarak Hamming = 10.

Berdasarkan Tabel I, SHA-256 hanya menghasilkan nilai sama pada skenario **asli copy**. Pada skenario tersebut, file uji merupakan salinan langsung dari file referensi sehingga tidak ada perubahan byte. Namun, pada skenario lain, SHA-256 selalu bernilai **False**. Hal ini terjadi meskipun perubahan yang dilakukan tergolong ringan, seperti kompresi JPEG, perubahan brightness, perubahan contrast, resize, blur, maupun rotasi kecil. Hasil ini menunjukkan bahwa SHA-256 sangat sensitif terhadap perubahan file. Dengan demikian, SHA-256 cocok digunakan untuk memeriksa kesamaan file secara

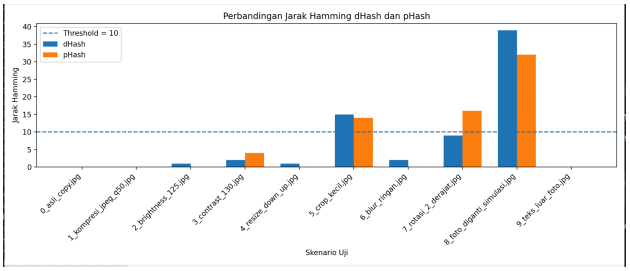
eksak, tetapi kurang sesuai untuk mengukur kemiripan visual gambar.

Pada skenario perubahan ringan, dHash dan pHash menunjukkan hasil yang lebih stabil. Kompresi JPEG Q50 menghasilkan jarak dHash 0 dan pHash 0, sehingga keduanya masih menganggap gambar identik. Brightness 125% menghasilkan jarak dHash 1 dan pHash 0. Resize turun-naik juga hanya menghasilkan jarak dHash 1 dan pHash 0. Sementara itu, blur ringan menghasilkan jarak dHash 2 dan pHash 0. Hasil ini menunjukkan bahwa perceptual hashing masih dapat mengenali gambar sebagai gambar yang sama atau sangat mirip meskipun representasi file-nya sudah berubah.

Skenario crop kecil menghasilkan jarak dHash 15 dan pHash 14. Kedua nilai tersebut sudah melewati threshold 10, sehingga dikategorikan sebagai terindikasi berbeda. Hal ini menunjukkan bahwa pemotongan kecil pada area foto dapat memengaruhi struktur visual secara cukup signifikan. Pada konteks foto profil KTP, crop kecil dapat mengubah posisi wajah, tepi foto, atau komposisi objek, sehingga hash visual yang dihasilkan menjadi lebih jauh dari gambar referensi.

Skenario rotasi 2 derajat menunjukkan hasil yang berbeda antara dHash dan pHash. dHash menghasilkan jarak 9 sehingga masih dikategorikan mirip, sedangkan pHash menghasilkan jarak 16 sehingga dikategorikan terindikasi berbeda. Perbedaan ini menunjukkan bahwa pHash dalam eksperimen ini lebih sensitif terhadap perubahan orientasi gambar. Rotasi kecil dapat mengubah susunan frekuensi rendah pada citra sehingga nilai pHash berubah lebih besar dibandingkan dHash. Temuan ini menunjukkan bahwa pada sistem nyata, proses normalisasi orientasi atau koreksi kemiringan gambar perlu dilakukan sebelum hashing.

Skenario paling penting dalam eksperimen ini adalah foto diganti simulasi. Pada skenario tersebut, dHash menghasilkan jarak 39 dan pHash menghasilkan jarak 32. Kedua nilai tersebut jauh di atas threshold 10, sehingga sistem berhasil mengklasifikasikannya sebagai terindikasi berbeda. Hasil ini menunjukkan bahwa dHash dan pHash mampu mendeteksi perubahan visual yang signifikan pada area foto profil. Dengan demikian, perceptual hashing dapat digunakan sebagai metode awal untuk menandai kemungkinan manipulasi foto pada KTP.



Gambar 5. Perbandingan jarak Hamming dHash dan pHash untuk setiap skenario uji.

Sementara itu, skenario teks luar foto menghasilkan jarak dHash 0 dan pHash 0. Hasil ini menunjukkan bahwa perubahan pada area di luar foto profil tidak memengaruhi hasil perceptual hashing. Hal ini terjadi karena eksperimen menggunakan crop pada area foto profil, sehingga bagian teks di luar area foto tidak ikut dihitung. Hasil ini penting karena tujuan sistem memang bukan memeriksa seluruh isi KTP, melainkan fokus pada perubahan foto profil.

Secara umum, hasil eksperimen menunjukkan perbedaan fungsi antara SHA-256 dan perceptual hashing. SHA-256 sangat baik untuk mendeteksi apakah file benar-benar identik, tetapi tidak dapat membedakan apakah perubahan tersebut hanya perubahan teknis ringan atau perubahan visual penting. Sebaliknya, dHash dan pHash lebih sesuai untuk melihat kemiripan visual. Perubahan ringan seperti kompresi, brightness, resize, dan blur masih dianggap mirip, sedangkan perubahan besar seperti crop kecil dan penggantian foto menghasilkan jarak Hamming tinggi.

Dari hasil tersebut, dapat disimpulkan bahwa penggunaan dHash dan pHash lebih relevan untuk deteksi awal manipulasi foto profil KTP dibandingkan SHA-256 saja. Namun, threshold tetap perlu diperhatikan. Threshold 10 pada eksperimen ini cukup mampu membedakan perubahan ringan dan perubahan signifikan, tetapi belum tentu ideal untuk semua jenis gambar. Pada penerapan nyata, threshold sebaiknya dikalibrasi menggunakan dataset yang lebih besar agar sistem tidak terlalu sensitif dan tidak terlalu longgar dalam mendeteksi manipulasi.

VII. ANALISIS KEAMANAN DAN BATASAN

Sistem yang dibuat pada makalah ini berfungsi sebagai metode deteksi awal terhadap kemungkinan manipulasi foto profil pada KTP. Walaupun hasil eksperimen menunjukkan bahwa dHash dan pHash dapat membedakan perubahan ringan dan perubahan signifikan, sistem ini tidak dapat dianggap sebagai mekanisme keamanan kriptografis penuh. dHash dan pHash tidak dirancang untuk memiliki sifat seperti collision resistance, preimage resistance, atau second preimage resistance sebagaimana SHA-256. Dengan demikian, perceptual hashing lebih tepat digunakan sebagai alat bantu analisis kemiripan visual, bukan sebagai bukti mutlak keaslian dokumen.

Dari sisi keamanan, kelemahan utama perceptual hashing adalah kemungkinan adanya citra manipulasi yang tetap menghasilkan hash mirip dengan citra referensi. Penyerang yang mengetahui cara kerja algoritma dapat mencoba melakukan perubahan secara halus agar jarak Hamming tetap berada di bawah threshold. Misalnya, manipulasi dilakukan dengan mempertahankan pola pencahayaan, bentuk umum wajah, atau struktur frekuensi rendah citra. Oleh karena itu, hasil "mirip" pada dHash atau pHash tidak selalu berarti gambar benar-benar asli. Hasil tersebut hanya menunjukkan bahwa gambar masih memiliki kemiripan visual berdasarkan metode hash yang digunakan.

Batasan berikutnya terletak pada penggunaan area crop. Dalam eksperimen ini, area foto profil dipotong menggunakan koordinat tertentu agar sistem hanya membandingkan bagian foto, bukan seluruh isi KTP. Pendekatan ini cukup efektif karena perubahan teks di luar area foto menghasilkan jarak dHash dan pHash sebesar 0. Namun, crop manual juga memiliki kelemahan. Jika posisi foto pada gambar KTP bergeser, miring, terlalu jauh, atau terpotong saat pengambilan gambar, area crop dapat tidak tepat. Akibatnya, sistem dapat menghasilkan jarak Hamming yang tinggi walaupun foto

sebenarnya tidak dimanipulasi. Untuk penerapan yang lebih nyata, proses crop sebaiknya dilakukan secara otomatis dengan deteksi layout KTP atau deteksi area foto.

Batasan lain terlihat pada skenario rotasi. Pada eksperimen, rotasi 2 derajat menghasilkan dHash distance 9 dan pHash distance 16. dHash masih menganggap gambar mirip, sedangkan pHash mengindikasikan perbedaan. Hasil ini menunjukkan bahwa perubahan orientasi kecil dapat memengaruhi nilai hash, terutama pHash. Dalam kondisi nyata, gambar KTP sering diambil menggunakan kamera ponsel dengan posisi miring, tidak sejajar, atau mengalami distorsi perspektif. Karena itu, sistem perlu dilengkapi dengan tahap normalisasi gambar, seperti koreksi kemiringan, koreksi perspektif, dan penyesuaian posisi sebelum proses hashing dilakukan.

Selain itu, eksperimen pada makalah ini masih menggunakan data simulasi dan jumlah skenario yang terbatas. Data yang digunakan belum mencakup variasi kondisi nyata seperti pencahayaan buruk, pantulan cahaya pada permukaan kartu, blur akibat gerakan tangan, noise kamera, kualitas kamera berbeda, serta variasi hasil scan. Oleh karena itu, hasil eksperimen ini belum dapat digunakan untuk menyimpulkan performa sistem pada semua kondisi unggahan KTP. Eksperimen ini lebih tepat dipahami sebagai pembuktian konsep bahwa perceptual hashing dapat digunakan untuk membedakan perubahan visual ringan dan perubahan visual signifikan pada area foto profil.

Threshold juga menjadi faktor penting yang membatasi sistem. Pada eksperimen ini digunakan threshold jarak Hamming sebesar 10. Nilai tersebut cukup untuk membedakan beberapa perubahan ringan dari perubahan besar seperti crop kecil dan foto diganti simulasi. Namun, threshold ini belum tentu cocok untuk semua dataset. Jika threshold terlalu rendah, gambar yang sebenarnya masih sah dapat dianggap sebagai manipulasi. Sebaliknya, jika threshold terlalu tinggi, manipulasi halus dapat lolos sebagai gambar yang mirip. Karena itu, threshold sebaiknya ditentukan melalui pengujian dataset yang lebih besar, bukan hanya berdasarkan satu contoh citra.

Sistem ini juga belum melakukan verifikasi identitas wajah. dHash dan pHash hanya membandingkan pola visual gambar, bukan mengenali apakah wajah pada foto benar-benar milik orang yang sama. Pada kasus tertentu, dua foto yang memiliki komposisi, warna, dan pencahayaan mirip dapat menghasilkan jarak hash yang tidak terlalu besar, walaupun identitas orangnya berbeda. Untuk penggunaan pada sistem verifikasi identitas, perceptual hashing sebaiknya digabungkan dengan metode lain seperti face detection, face embedding, OCR, validasi data kependudukan, atau tanda tangan digital dari penerbit dokumen.

Dengan mempertimbangkan batasan-batasan tersebut, sistem yang dibuat dalam makalah ini lebih tepat diposisikan sebagai komponen pendeteksi awal. Sistem dapat membantu menandai gambar yang mencurigakan berdasarkan perubahan visual pada area foto profil, tetapi keputusan akhir tetap memerlukan pemeriksaan tambahan. Perceptual hashing dapat menjadi lapisan awal yang ringan dan mudah diimplementasikan, namun tidak boleh digunakan sebagai satu-satunya dasar untuk menyatakan bahwa sebuah KTP asli atau palsu.

VIII. REKOMENDASI PENGEMBANGAN SISTEM

Berdasarkan hasil eksperimen, sistem yang lebih siap digunakan perlu menambahkan normalisasi geometri sebelum hash dihitung. Normalisasi tersebut dapat berupa deteksi tepi kartu, koreksi perspektif, deskew, dan penentuan ulang posisi foto profil secara otomatis. Tanpa tahap ini, gambar yang sah tetapi sedikit miring dapat menghasilkan jarak Hamming tinggi, sebagaimana terlihat pada skenario rotasi 2 derajat.

Pengembangan kedua adalah kalibrasi threshold. Pada makalah ini threshold 10 digunakan sebagai nilai awal agar eksperimen mudah direproduksi. Dalam sistem nyata, threshold harus dicari dari distribusi jarak Hamming pada pasangan gambar asli-asli dan asli-manipulasi. Dari distribusi tersebut, pengembang dapat memilih titik operasi sesuai kebutuhan: lebih ketat untuk menekan false negative atau lebih longgar untuk menekan false positive.

Pengembangan ketiga adalah penggabungan beberapa fitur. dHash dan pHash dapat dipakai bersama dengan fitur lain seperti histogram warna, structural similarity index, deteksi wajah, face embedding, dan analisis konsistensi OCR. Pendekatan multi-fitur lebih kuat karena manipulasi yang lolos pada satu fitur masih dapat terdeteksi oleh fitur lain. Misalnya, penggantian foto dengan wajah berbeda sebaiknya tidak hanya dilihat dari hash visual, tetapi juga dari kecocokan embedding wajah terhadap foto selfie pembanding.

Pengembangan keempat adalah pembuatan audit trail. Ketika sistem memberi label terindikasi manipulasi, sistem sebaiknya menyimpan nilai hash, jarak Hamming, parameter crop, waktu pemeriksaan, dan versi algoritma. Audit trail ini membantu proses forensik dan evaluasi ulang ketika threshold atau algoritma diperbarui. Namun, penyimpanan audit harus tetap mengikuti prinsip minimisasi data karena KTP merupakan dokumen identitas pribadi.

Dari sisi keamanan, perceptual hash dapat disimpan bersama cryptographic hash. SHA-256 tetap berguna untuk memastikan apakah file yang sudah pernah diverifikasi berubah secara byte-level, sedangkan pHash dan dHash berguna untuk mengukur kemiripan visual. Kombinasi keduanya memberi dua jenis sinyal: integritas eksak dan konsistensi visual. Pemisahan fungsi ini penting agar sistem tidak menggunakan SHA-256 untuk tujuan yang bukan desain utamanya.

IX. KESIMPULAN

Makalah ini telah merancang dan mengimplementasikan sistem deteksi manipulasi foto profil KTP menggunakan pHash dan dHash, serta membandingkannya dengan SHA-256. Hasil eksperimen menunjukkan bahwa SHA-256 efektif untuk memeriksa kesamaan file secara eksak, tetapi terlalu sensitif untuk perbandingan visual. Sebaliknya, dHash dan pHash dapat mempertahankan jarak Hamming rendah pada perubahan ringan seperti kompresi, brightness, contrast, resize, dan blur, tetapi menghasilkan jarak tinggi pada manipulasi signifikan seperti penggantian foto dan rotasi tanpa normalisasi.

Secara praktis, pHash dan dHash dapat digunakan sebagai komponen awal untuk menandai unggahan KTP yang mencurigakan. Namun, sistem harus dilengkapi

dengan normalisasi geometri, kalibrasi threshold berbasis dataset, perlindungan data pribadi, dan mekanisme verifikasi tambahan. Untuk penelitian lanjutan, eksperimen dapat diperluas dengan dataset KTP asli yang dianonimkan, variasi perangkat kamera, manipulasi wajah yang lebih halus, dan evaluasi metrik seperti false positive rate, false negative rate, precision, recall, dan F1-score.

APENDIKS A. PERINTAH EKSPERIMEN

Perintah berikut digunakan untuk menjalankan eksperimen pada program deteksi manipulasi foto profil KTP. Program dijalankan melalui terminal dari direktori utama project.

```
# Instalasi dependensi
pip install -r requirements.txt
# Perbandingan satu gambar
python src/ktp_hash_detector.py --reference
data/reference/ktp_asli.jpg --query
data/query/ktp_uji.jpg --crop 421,86,140,184
# Membuat seluruh sampel uji
python src/ktp_hash_detector.py --make-samples
data/reference/ktp_asli.jpg --sample-output-dir
data/samples --crop 421,86,140,184
# Eksperimen batch dan ekspor hasil ke CSV
python src/ktp_hash_detector.py --reference
data/reference/ktp_asli.jpg --batch-folder data/samples
--crop 421,86,140,184 --output-csv
outputs/hasil_pengujian.csv
```

Koordinat crop 421,86,140,184 digunakan untuk mengambil area foto profil pada citra KTP simulasi. Format crop yang digunakan adalah x,y,w,h, dengan x dan y sebagai posisi awal area crop, sedangkan w dan h sebagai lebar dan tinggi area foto. File hasil eksperimen disimpan dalam bentuk CSV pada direktori outputs/hasil_pengujian.csv.

APENDIKS B. STRUKTUR DIREKTORI PROGRAM

```
ktp_hash_detector_simple/
├── README.md
├── requirements.txt
├── src/
│   └── ktp_hash_detector.py
├── data/
│   ├── reference/
│   ├── query/
│   └── samples/
└── outputs/
```

Struktur tersebut memisahkan kode sumber, data referensi, data uji, dan keluaran eksperimen. Pemisahan ini mempermudah replikasi hasil, pemeriksaan ulang CSV, serta penggantian dataset tanpa mengubah kode utama.

APENDIKS D. CONTOH FORMAT KELUARAN CSV

Hasil Batch :

file_uji	sha256_sama	jarak_dhash	status_dhash	jarak_phash	status_phash
0_asli_copy.jpg	True	0	identik	0	identik
1_kompresi_jpeg_q50.jpg	False	0	identik	0	identik
2_brightness_125.jpg	False	1	mirip	0	identik
3_contrast_130.jpg	False	2	mirip	4	mirip
4_resize_down_up.jpg	False	1	mirip	0	identik
5_crop_kecil.jpg	False	15	terindikasi berbeda	14	terindikasi berbeda
6_blur_ringan.jpg	False	2	mirip	0	identik
7_rotasi_2_derajat.jpg	False	9	mirip	16	terindikasi berbeda
8_foto_diganti_simulasi.jpg	False	39	terindikasi berbeda	32	terindikasi berbeda
9_teks_luar_foto.jpg	False	0	identik	0	identik

CSV tersebut dapat digunakan untuk membuat grafik, mengevaluasi threshold, atau membandingkan algoritma lain pada eksperimen berikutnya. Format keluaran yang sederhana juga memudahkan proses replikasi oleh pembaca.

REFERENCES

[1] National Institute of Standards and Technology, Secure Hash Standard (SHS), FIPS PUB 180-4, Aug. 2015.

[2] C. Zauner, Implementation and Benchmarking of Perceptual Image Hash Functions, M.S. thesis, Upper Austria University of Applied Sciences, Hagenberg, Austria, 2010.

[3] N. Krawetz, “Kind of Like That,” The Hacker Factor Blog, Jan. 2013.

[4] J. Buchner, ImageHash: A Python perceptual image hashing module, GitHub repository, accessed 2026.

[5] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. New York, NY, USA: Pearson, 2018.

[6] S. McKeown and W. J. Buchanan, “Hamming distributions of popular perceptual hashing techniques,” Forensic Science International: Digital Investigation, vol. 44, 2023.

[7] J. Fridrich, Digital Image Forensics. IEEE Signal Processing Magazine, vol. 26, no. 2, pp. 26-37, 2009.

[8] Python Software Foundation, hashlib - Secure hashes and message digests, Python documentation, accessed 2026.

[9] Pillow contributors, Pillow Image processing library documentation, accessed 2026.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.
Bandung, 19 Juni 2026

Ttd.

Willhelmina Rachel Silalahi

Willhelmina Rachel Silalahi