

## PERBANDINGAN KOMBINASI METODE TEMPLATE MATCHING DAN ALGORITMA FEATURE MATCHING PADA PENGENALAN MATA UANG INDIA

Dwiki Aditama S., Fahmi Nugroho A., M. Rifan D., M. Hilal, M. Atay N.N., Fetty Tri Anggraeny

Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur  
Email: 18081010064@student.upnjatim.ac.id

**Abstrak.** *Pengolahan Cira Digital (Digital Image Processing) adalah sebuah ilmu yang mempelajari tentang teknik-teknik dalam mengolah citra. Ada beberapa metode dalam pengolahan citra digital salah satunya adalah template matching dan feature matching. Pada penelitian ini dalam template matching dilakukan perbandingan terhadap dua metode yaitu CV TM SQDIFF dengan CV TM CCOEFF, sedangkan pada feature matching dilakukam perbandingan terhadap dua algoritma yaitu ORB (Oriented FAST and Rotated BRIEF) dengan SIFT (Scale-Invariant Feature Transform), pada feature matching hal pertama yang harus dilakukan ialah ekstraksi fitur, ekstraksi fitur memiliki banyak manfaat salah satunya adalah mendeteksi fitur-fitur pada benda. Pada penelitian ini ekstraksi fitur digunakan untuk mengenali fitur-fitur pada mata uang negara India yang digunakan untuk mendeteksi nominal pada mata uang kertas yang diinputkan dan juga akan dilakukan perbandingan dari keempat algoritma diatas. Pada tahap pembuatan system pengenalan nominal pada mata uang terdapat beberapa tahapan proses yaitu pengumpulan data, perancangan system, training dan testing. Hasil yang didapatkan dalam penelitian ini menggunakan algoritma CV TM SQDIFF dikombinasi dengan ORB (Oriented FAST and Rotated BRIEF) didapatkan akurasi sebesar 80%, ketika menggunakan algoritma CV TM SQDIFF dengan SIFT (Scale-Invariant Feature Transform) didapatkan akurasi sebesar 94,28%, ketika menggunakan algoritma CV TM CCOEFF dengan ORB (Oriented FAST and Rotated BRIEF) didapatkan hasil akurasi 74%, dan ketika menggunakan algoritma CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform) didapatkan hasil 97%. Sehingga dapat disimpulkan bahwa system pengenalan mata uang menggunakan algoritma CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform) dapat mendeteksi nominal uang dengan baik.*

**Kata Kunci:** *pengolahan citra digital, template matching, feature matching, open cv, pengenalan mata uang.*

Uang kertas merupakan salah satu alat pembayaran yang sering digunakan oleh manusia hampir di seluruh penjuru dunia. Pasalnya, uang adalah alat untuk transaksi dalam jual beli barang dan jasa [1]. Di negara India banknya mengeluarkan berbagai jenis dan model uang kertas, sehingga membuat masyarakat di India dalam melakukan proses transaksi menggunakan uang kertas dengan beragam model dan nominal.

Pada era saat ini perkembangan teknologi memiliki pengaruh besar terhadap kemudahan aktivitas manusia. Berbagai jenis sistem automasi sudah banyak dibuat dan telah banyak memberikan kemudahan dan dapat membantu pekerjaan [2]. Contoh kemudahan yang dirasakan manusia adalah mesin pendeteksi nominal mata uang. Pengenalan nilai atau nominal dari mata uang kertas tidak kalah pentingnya dengan pengenalan objek lain [3].

Nilai mata uang kertas di India sangat beragam dan berbeda-beda. Seperti, ukuran yang bervariasi, warna yang berbeda, dan pola yang beragam sehingga masyarakat India harus

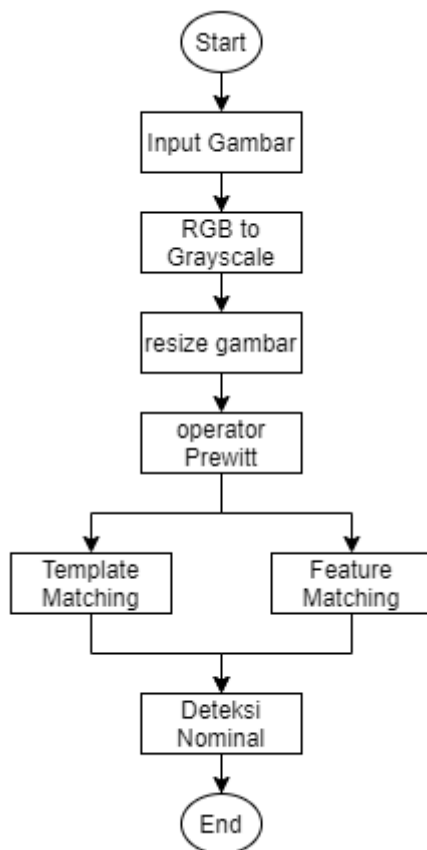
mengingat tampilan dari setiap uang kertas tersebut. Hal ini dapat menyebabkan pengenalan uang kertas yang salah sehingga orang membutuhkan sistem yang efisien dan tepat untuk membantu pekerjaan mereka [4].

Tujuan dari sistem pengenalan uang kertas adalah mengenali nilai uang kertas secara otomatis dengan menggunakan berbagai teknik dan metode salah satunya adalah dengan *template matching* dan *feature matching*. Sistem pengenalan mata uang kertas juga dikembangkan berdasarkan pengolahan citra digital. Pengolahan Citra Digital (*Digital Image Processing*) merupakan salah satu disiplin ilmu yang mempelajari bagaimana citra dapat dibentuk, dilakukan pengolahan, dan dianalisis oleh komputer sehingga menghasilkan informasi yang dapat dipahami oleh manusia [5]. *Input* pada pengolahan citra adalah citra, sedangkan *output* yang dihasilkan adalah citra hasil pengolahan [6]. Tahapan-tahapan tersebut diterapkan pada sistem ini dengan pra-pengolahan (*preprocessing*) menggunakan

*grayscale*, *resizing*, dan perbaikan citra yang di-*input*-kan.

## I. Metodologi

Metodologi penelitian yang dilakukan sebagaimana ditunjukkan pada gambar 1, dimulai dari peng-*input*-an gambar yang akan diuji kemudian dilakukan *pre-processing* terhadap gambar *input*. *Pre-processing* yang dilakukan adalah merubah citra uang dari RGB menjadi *grayscale* kemudian mengubah ukuran sesuai dengan yang ditetapkan dan dilakukan konvolusi yaitu menggunakan operator Prewitt. Setelah itu gambar *input* dilakukan proses *template matching* dan *feature matching* untuk mencocokkan dengan dataset yang ada. Hasil dari *template matching* dan *feature matching* digunakan untuk menentukan nominal dari gambar mata uang yang di *input*-kan.



Gambar 1. Metodologi Penelitian

### Konversi citra RGB ke *Grayscale*

Pada tahap *Preprocessing Image* yang pertama yaitu perubahan citra uang kertas dari citra RGB menjadi citra *Grayscale*. Citra *grayscale* dapat dihasilkan dari citra RGB

dimana ketiga komponen warna citra RGB dikalikan dengan suatu koefisien yang jumlahnya satu. Citra *grayscale* dihitung menggunakan Persamaan 1.

$$Y = (a \times R) + (b \times G) + (c \times B) \quad (1)$$

Y merupakan citra *grayscale* dan a, b, c adalah variabel koefisien dengan nilai  $a + b + c = 1$ . Dalam komputasi, meskipun *grayscale* dapat dihitung menggunakan bilangan rasional, piksel citra tetap disimpan dalam bentuk biner dan sudah dikuantisasi. Hal ini disebabkan citra *grayscale* memiliki struktur yang lebih sederhana dibanding citra warna RGB sehingga komputasi dan pengolahan citra dapat dilakukan lebih cepat dan efisien. Intensitas keabuan biasanya disimpan sebagai data citra 8 bit per sampel piksel atau 256 intensitas warna gray dari nilai 0 (hitam) dan 255 (putih)[7].

### Deteksi Tepi Prewitt

*Edge detection* atau disebut deteksi tepi yaitu merupakan suatu proses yang dilakukan pada citra yang memberikan hasil berupa tepian dari objek-objek yang ada pada gambar. Pada deteksi tepi dari sebuah citra suatu titik (x,y) dapat dikatakan menjadi tepi dari suatu citra jika titik-titik tersebut memiliki perbedaan yang tinggi atau signifikan terhadap tetangga. Fungsi dari deteksi tepi yaitu memperoleh tepi dari setiap objek pada citra. Proses pada deteksi tepi merupakan hasil dari pemanfaatan perubahan nilai intensitas yang signifikan pada batas dua area. Jika citra tersebut memiliki ketajaman dan kejelasan yang baik maka letak tepi dapat lebih mudah ditentukan, tetapi jika citra tidak jelas dan banyak terdapat *noise* dan gangguan maka letak tepi sulit untuk ditentukan[8]. deteksi tepi yang digunakan pada penelitian ini adalah menggunakan metode Prewitt.

Prewitt merupakan metode atau operator yang menggunakan matriks *neighbor* dengan ukuran 3x3 dengan titik yang sedang diperiksa sebagai titik tengah matriks. Prewitt diterapkan pada dua buah matriks mask yaitu matriks yang memiliki n x n ukuran yang sama dengan matriks *neighbor*[9]. Mask terbagi menjadi dua yaitu mask horizontal yang berfungsi untuk menghitung selisih antara titik pada sisi horizontal, mask vertikal memiliki fungsi menghitung selisih antara titik pada sisi vertikal. Metode Prewitt adalah hasil pengembangan dari metode Robert yaitu menggunakan High pass Filter (HPF) yang

diberi satu angka nol sebagai penyangga[10] . Gambar 2 merupakan matriks horisontal dan vertikal pada metode Prewitt.

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Gambar 2. Matriks Prewitt

Perhitungan yang dilakukan pada metode Prewitt yaitu dengan melakukan perkalian matriks antara matriks *neighbor* dengan matriks mask horisontal, selanjutnya melakukan perkalian matriks antara matriks *neighbor* dengan matriks mask vertikal. Hasil perkalian tersebut dijumlahkan yang memiliki hasil berupa penelusuran secara horisontal dan vertikal ( $G[fx,y]$ ).

### Template Matching

*Template Matching* merupakan salah satu teknik pengolahan citra yaitu membanding citra sumber (*input-an*) dengan citra template untuk menemukan bagian yang cocok dengan citra template. Template Matcing sendiri merupakan proses mencari suatu objek pada keseluruhan objek yang berada dalam suatu citra yang di-*input*-kan[11]. Pada *template matching* dilakukan perbandingan terhadap dua metode yaitu CV TM SQDIFF dengan CV TM CCOEFF. CV TM SQDIFF yaitu metode yang melakukan perhitungan selisih kuadrat antara template dan gambar masukan, nilai kecocokan terbaik adalah 0, semakin buruk kecocokannya, semakin besar nilai kecocokannya. yang dapat dihitung pada Persamaan 2.

$$R_{sdiff} = \sum_{ij} [t(i,j) - f(x+i, y+j)]^2 \quad (2)$$

CV TM CCOEFF yaitu metode yang melakukan perhitungan korelasi co-efisien matching dan membuat korelasi antara template yang dikurangi dari rata-ratanya dan gambar sumber dikurangi dari rata-ratanya di setiap posisi, dengan mempertimbangkan ukuran template. Dalam menghitung koefisien korelasi digunakan persamaan 3, jika koefisien korelasi bernilai 1 berarti kecocokan sempurna, jika bernilai -1 berarti kecocokan sangat buruk[12].

$$R_{ccoeff} = \sum_{ij} [(t(i,j) - \bar{t}) * (f(x+i, y+j) - \bar{f})]^2 \quad (3)$$

### Feature Matching

*Feature matching* merupakan sebuah metode yang melibatkan pendeteksian feature yang memiliki descriptor yang sama atau mirip. Fitur citra adalah potongan atau bagian kecil dari gambar yang berguna dalam menghitung kemiripan antara dua citra[13]. Feature sendiri merupakan karakteristik unik dari suatu gambar. Deteksi feature merupakan langkah pertama yang harus dilakukan dalam proses pengenalan, karena feature yang ditemukan memiliki peran penting dalam proses pencocokkan dengan gambar template. Setelah itu melakukan feature description yaitu memberi deskripsi atas area sekitarnya pada setiap keypoint yang ditemukan. Algoritma yang dapat memenuhi hal tersebut adalah ORB, SURF, dan SIFT [14]. Pada *feature matching* dilakukan perbandingan terhadap dua algoritma yaitu ORB (Oriented FAST and Rotated BRIEF) dengan SIFT (Scale-Invariant Feature Transform).

ORB adalah perpaduan dari detektor titik kunci FAST dan deskriptor BRIEF dengan beberapa modifikasi. FAST digunakan untuk menentukan poin kunci. Untuk menemukan poin N teratas maka diterapkan ukuran sudut Harris. FAST merupakan varian rotasi sehingga tidak melakukan perhitungan orientasi. FAST menghitung intensitas pusat massa patch menggunakan sudut yang terletak di tengah. Arah vektor dari titik sudut tersebut ke pusat massa memberikan orientasi. Untuk meningkatkan invarian rotasi dilakukan perhitungan momen. Deskriptor BRIEF tidak dapat bekerja dengan baik jika terdapat rotasi. Dalam ORB, untuk menghitung matriks rotasi digunakan orientasi tambahan yang kemudian deskriptor BRIEF diarahkan sesuai dengan orientasi [15].

Pada ORB metode yang digunakan adalah FAST-9 dalam mendeteksi feature, yaitu kandidat keypoint dibandingkan dengan pixel lingkaran disekitarnya dengan radius sebesar 9 pixel. Setiap keypoint dihitung bobotnya pada persamaan 4[16].

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (4)$$

SIFT merupakan algoritma yang diusulkan oleh Lowe yang bertujuan untuk memecahkan rotasi gambar, transformasi affine, intensitas, dan perubahan sudut pandang dalam fitur yang cocok. Dalam algoritma SIFT terdapat 4 langkah dasar. Pertama yaitu

memperkirakan ekstrema ruang skala menggunakan Difference of Gaussian (DoG). Pada tahap ini gambar diperkecil sejumlah oktaf dimana setiap oktaf akan dilakukan Gaussian blur dengan tingkat level Difference of Gaussian (DoG) yang didapat dari selisih gambar setiap oktaf. Kedua, lokalisasi titik kunci yaitu pada setiap kandidat titik kunci dilokalkan dan disempurnakan dengan cara menghilangkan titik kontras rendah. Setiap kandidat titik kunci dicari maksimal dan minimalnya menggunakan perbandingan 3x3x3 tetangga pada level yang sama, level dibawah dan diatasnya. Ketiga, penetapan orientasi kanonik titik kunci yang mengacu pada gradien gambar lokal. Pada tahap ini dilakukan filterisasi pada titik kunci menggunakan kontras rendah dan *edge* dengan menggunakan matriks Hessian dari titik kunci dan nilai eigenvalue-nya dari keduanya dapat diambil menggunakan persamaan 5.

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (5)$$

Terakhir generator descriptor pada titik kunci yang digunakan untuk menghitung deskriptor gambar lokal pada setiap titik kunci berdasarkan besaran dan orientasi gradien gambar. Dari 16x16 piksel yang dibagi menjadi 4x4 untuk dibentuk histogram yang kemudian menghasilkan descriptor sebesar 128 dimensi[17].

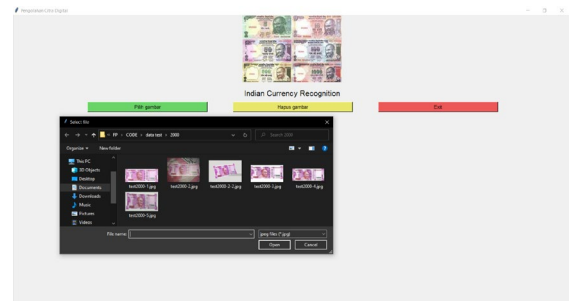
## II. Hasil dan Pembahasan

Pada hasil dan pembahasan memuat jawaban atas pertanyaan-pertanyaan penelitian. Hasil yang disajikan adalah 'hasil bersih'. Penyampaian hasil penelitian dapat dibantu dengan pemakaian tabel, grafik, atau bentuk/format komunikasi yang lain. Grafik dan tabel harus dibahas dalam tubuh artikel. Jika penyajian hasil cukup panjang, dapat dibagi dalam beberapa sub bagian.

Pengujian yang dilakukan menggunakan perangkat keras dengan CPU AMD Ryzen 3550H @2.1GHz – 3.7GHz, GPU NVIDIA GTX 1650 4GB, RAM DDR4 8 GB. Kemudian untuk perangkat lunak menggunakan sistem operasi Windows 10 64-bit, Python versi 3.8.10, Jupyter Notebook versi 6.4.0. Hasil Penelitian yang dilakukan adalah sebagai berikut.

## Input Gambar

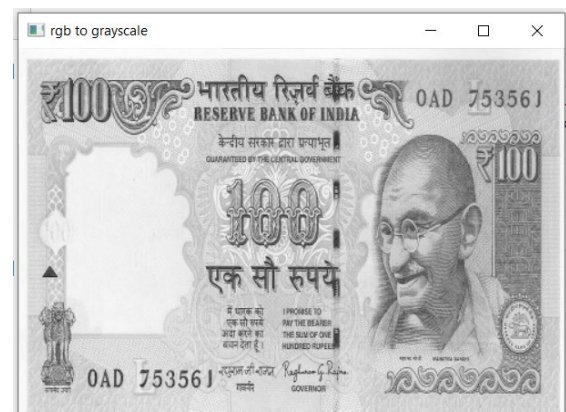
Proses memasukkan gambar kedalam sistem yaitu menggunakan folder yang sudah di inisiasi, sehingga ketika tombol pilih gambar ditekan maka otomatis folder terbuka dan menampilkan gambar yang akan dipilih. Gambar 3 merupakan tampilan ketika proses memilih dan memasukkan gambar ke dalam sistem.



Gambar 3. Proses *Input* Gambar

## Konversi RGB menjadi Grayscale

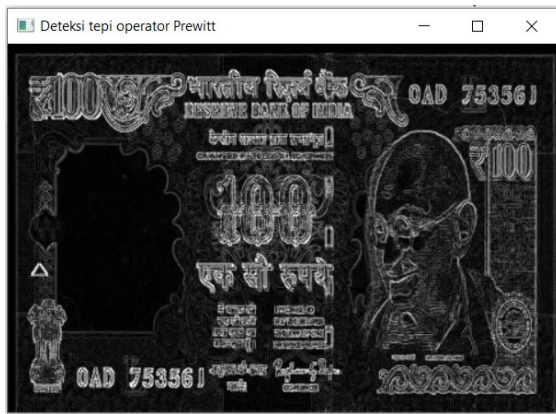
Dari hasil gambar yang telah dimasukkan ke sistem pengenalan, gambar akan dikonversi menjadi *grayscale* dengan tujuan memudahkan dan meringankan komputasi. Gambar 4 merupakan hasil dari konversi RGB menjadi *grayscale*.



Gambar 4. Hasil konversi RGB ke *Grayscale*

## Operator Prewitt

Dari hasil gambar *grayscale* yang telah diresize dilanjutkan ketahap konvolusi yaitu deteksi tepi menggunakan operator Prewitt. Gambar 5 merupakan hasil dari deteksi tepi menggunakan operator Prewitt.



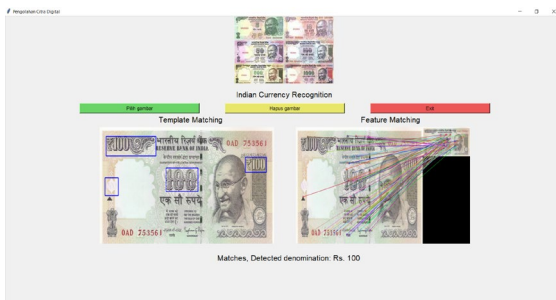
Gambar 5. Hasil Deteksi Tepi Prewitt

### Template matching dan Feature matching

Pada tahap ini proses dilakukan secara bersamaan menggunakan kombinasi metode dan algoritma sebagai berikut.

### CV TM SQDIFF dengan ORB (Oriented FAST and Rotated BRIEF)

Tahap ini setelah gambar dilakukan deteksi tepi menggunakan operator Prewitt dilanjutkan pada proses *template matching* dan *feature matching* menggunakan CV TM SQDIFF dengan ORB (Oriented FAST and Rotated BRIEF). Setelah proses selesai gambar dikembalikan menjadi citra RGB dan ditampilkan hasil prosesnya. Gambar 6 merupakan hasil proses pada tahap ini.



Gambar 6. Hasil Proses CV TM SQDIFF dengan ORB

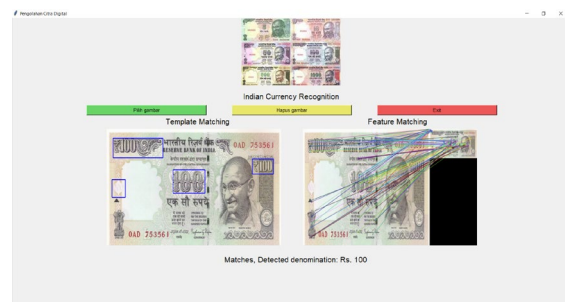
Tabel 1. Hasil Pengujian CV TM SQDIFF dengan ORB

No	Nominal yang diuji	Salah	Benar	Persentase
1.	10	0	5	100%
2.	20	1	4	80%
3.	50	1	4	80%
4.	100	1	4	80%
5.	200	1	4	80%
6.	500	2	3	60%
7.	2000	1	4	80%
Total				80%

Pada Tabel 1 merupakan hasil pengujian menggunakan metode *template matching* CV TM SQDIFF dengan algoritma ORB pada *feature matching* yang menghasilkan nilai rata-rata keberhasilan sebesar 80% pada akurasi pendeteksian nominal uang kertas. Hasilnya cukup baik walaupun ada sedikit kegagalan deteksi hampir disetiap jenis mata uang.

### CV TM SQDIFF dengan SIFT (Scale-Invariant Feature Transform)

Tahap ini setelah gambar dilakukan deteksi tepi menggunakan operator Prewitt dilanjutkan pada proses *template matching* dan *feature matching* menggunakan CV TM SQDIFF dengan SIFT (*Scale-Invariant Feature Transform*). Setelah proses selesai gambar dikembalikan menjadi citra RGB dan ditampilkan hasil prosesnya. Gambar 7 merupakan hasil proses pada tahap ini.



Gambar 7. Hasil Proses CV TM SQDIFF dengan SIFT

Tabel 2. Hasil Pengujian CV TM SQDIFF dengan SIFT

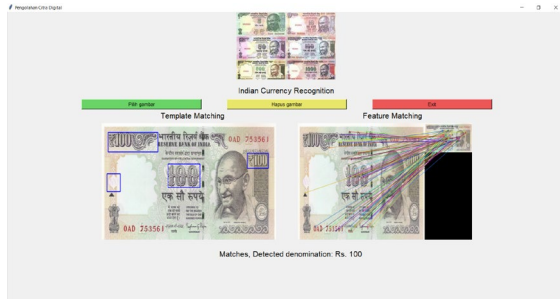
No	Nominal yang diuji	Salah	Benar	Persentase
1.	10	0	5	100%
2.	20	0	5	100%
3.	50	0	5	100%
4.	100	1	4	80%
5.	200	1	4	80%
6.	500	0	5	100%
7.	2000	0	5	100%
Total				94.28%

Pada Tabel 2 merupakan hasil pengujian menggunakan metode *template matching* CV TM SQDIFF dengan algoritma SIFT pada *feature matching* yang menghasilkan nilai rata-rata keberhasilan sebesar 94.28% pada akurasi pendeteksian nominal uang kertas. Hasilnya menunjukkan peningkatan dibanding menggunakan algoritma ORB pada *feature matching*nya.



**CV TM CCOEFF dengan ORB (Oriented FAST and Rotated BRIEF)**

Tahap ini setelah gambar dilakukan deteksi tepi menggunakan operator Prewitt dilanjutkan pada proses *template matching* dan *feature matching* menggunakan CV TM CCOEFF dengan ORB (Oriented FAST and Rotated BRIEF). Setelah proses selesai gambar dikembalikan menjadi citra RGB dan ditampilkan hasil rosesnya. Gambar 8 merupakan hasil proses pada tahap ini.



Gambar 8. Hasil Proses CV TM CCOEFF dengan ORB

Tabel 3. Hasil Pengujian CV TM CCOEFF dengan ORB

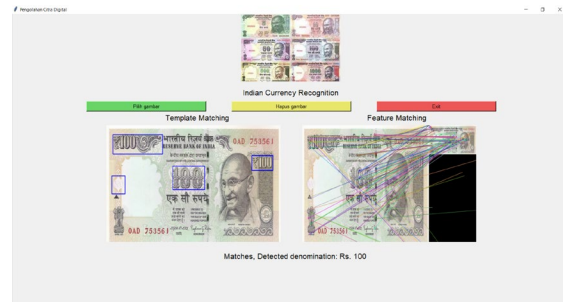
No	Nominal yang diuji	Salah	Benar	Persentase
1.	10	0	5	100%
2.	20	2	3	60%
3.	50	1	4	80%
4.	100	1	4	80%
5.	200	2	3	60%
6.	500	1	4	80%
7.	2000	2	3	60%
Total				74%

Pada Tabel 3 merupakan hasil pengujian menggunakan metode *template matching* CV TM CCOEFF dengan algoritma ORB pada *feature matching* yang menghasilkan nilai rata-rata keberhasilan sebesar 74% pada akurasi pendeteksian nominal uang kertas. Hasilnya menurun drastis dibandingkan dengan menggunakan CV TM SQDIFF dengan SIFT maupun CV TM SQDIFF dengan ORB.

**CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform)**

Tahap ini setelah gambar dilakukan deteksi tepi menggunakan operator Prewitt dilanjutkan pada proses *template matching* dan *feature matching* menggunakan CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform). Setelah proses selesai gambar

dikembalikan menjadi citra RGB dan ditampilkan hasil rosesnya. Gambar 9 merupakan hasil proses pada tahap ini.



Gambar 9. Hasil Proses CV TM CCOEFF dengan SIFT

Tabel 4. Hasil Pengujian CV TM CCOEFF dengan SIFT

No	Nominal yang diuji	Salah	Benar	Persentase
1.	10	0	5	100%
2.	20	0	5	100%
3.	50	0	5	100%
4.	100	0	5	100%
5.	200	0	5	100%
6.	500	0	5	100%
7.	2000	1	4	80%
Total				97%

Pada Tabel 4 merupakan hasil pengujian menggunakan metode *template matching* CV TM CCOEFF dengan algoritma SIFT pada *feature matching* yang menghasilkan nilai rata-rata keberhasilan sebesar 97% pada akurasi pendeteksian nominal uang kertas. Hasilnya menjadi yang paling tinggi dibanding ketiga percobaan dengan metode dan algoritma yang berbeda.

Dari hasil seluruh pengujian dilakukan perhitungan persentase keberhasilan deteksi nominal, tujuannya adalah untuk mencari dan membandingkan hasil yang paling optimal.

Tabel 5. Perbandingan Setiap Percobaan

No	Jenis Percobaan	Persentase
1.	CV TM SQDIFF dengan ORB (Oriented FAST and Rotated BRIEF)	80%
2.	CV TM SQDIFF dengan SIFT (Scale-Invariant Feature Transform)	94.28%
3.	CV TM CCOEFF dengan ORB (Oriented FAST and Rotated BRIEF)	74%

- |   |     |
|---|-----|
| 4. CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform) | 97% |
|---|-----|

Tabel 5 merupakan tabel perbandingan keempat percobaan. Pada tabel tersebut memperlihatkan persentase rata-rata keberhasilan ke empat percobaan. Dari ke empat percobaan yang menunjukkan hasil terbaik adalah menggunakan CV TM CCOEFF dengan SIFT (Scale-Invariant Feature Transform) dengan nilai 97% dan yang terburuk dengan nilai 74% menggunakan CV TM CCOEFF dengan ORB (Oriented FAST and Rotated BRIEF).

### III. Kesimpulan

Kesimpulan menyajikan ringkasan dari uraian mengenai hasil penelitian dan pembahasan. Dari kedua hal ini dikembangkan pokok-pokok pikiran (baru) yang merupakan esensi dari temuan penelitian.

Pada penelitian ini dilakukan perbandingan template dan *feature matching* dengan metode dan algoritma yang berbeda-beda pada proses pengenalan nominal mata uang. Hasil perbandingan menunjukkan hasil yang berbeda-beda disetiap metode dan algoritma. Algoritma SIFT pada *feature matching* menunjukkan hasil yang lebih baik dibandingkan dengan ORB. Dan metode CV TM SQDIFF cenderung lebih kecil selisihnya dibanding CV TM CCOEFF jika di gabung dengan algoritma SIFT dan ORB. Kendala yang dihadapi pada penelitian ini adalah pada template matching, jika *input* gambar yang akan diujikan tidak sama dengan *dataset template* yang ada maka ROI yang dideteksi selalu tidak sesuai. Hal ini dikarenakan cara kerja dari *template matching* itu sendiri yaitu hanya membandingkan dua gambar yaitu gambar masukan dengan gambar template yang ada pada dataset tanpa adanya ekstraksi apapun. Jika gambar yang tepat tidak ditemukan maka akan bergeser ke gambar yang cenderung memiliki kemiripan sehingga menimbulkan ketidakvalidan hasil. Dikarenakan kendala tersebut deteksi nominal lebih banyak mengandalkan pada *feature matching*. Untuk penelitian selanjutnya, *dataset template* dapat diperbaiki lagi dengan data yang mudah digunakan untuk menjadi acuan bagi setiap mata uang yang akan dilakukan pengenalan. Untuk metode atau algoritma *template matching* dan *feature matching* dapat dilakukan

perbaikan atau perubahan menggunakan metode atau algoritma yang lebih baik lagi untuk meningkatkan akurasi pengenalan nominal mata uang.

### IV. Daftar Pustaka

- [1] J. F. Fauzi, H. Tolle, & R. K. Dewi. (2018). Implementasi Metode RGB To HSV pada Aplikasi Pengenalan Mata Uang Kertas Berbasis Android untuk Tuna Netra. *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, 2(6), 2319–2325.
- [2] J. Fathani, U. Sunarya, & I. N. A. Ramatryana. (2014). Aplikasi Identifikasi Dan Konversi Mata Uang Kertas Asing Terhadap Rupiah Dengan Metoda Local Binary Pattern ( Lbp ) Berbasis Android. *e-Proceeding Eng*, 1( 1), 363–371.
- [3] M. Sarfraz. (2015). An Intelligent Paper Currency Recognition System. *Procedia Comput. Sci.*, no. *International Conference on Communication, Management and Information Technology*, 65, 538– 545.
- [4] C. Page & S. H. M. G. (2018). Paper Currency Detection based Image Processing Techniques: A review paper. *J. Al-Qadisyah Comput. Sci. Math*, 10( 1), 1–8.
- [5] Rahmad Cahya, Rismanto Ridwan, & Pranata Febrita Dian. (2019). Pengenalan Nilai Mata Uang Kertas Untuk Tunanetra Menggunakan Metode *Template Matching* Correlation Berbasis Android. *Seminar Informatika Aplikatif Polinema.*, 73-80.
- [6] Wahyono, Eko S, & Ernastuti. (2009) Identifikasi Nomor Polisi Menggunakan Metode Jaringan Syaraf Buatan Learning Vector Quantization. *Jurusan Teknik Informatika Universitas Gunadarma*.
- [7] Tim Laboratorium Pengolahan Sinyal Digital. (2009). Modul Praktikum Pengolahan Sinyal Digital Fakultas Elektro dan Komunikasi. Bandung : Universitas Telkom.
- [8] S. Sukatmi. (2017). Perbandingan Deteksi Tepi Citra Digital dengan Menggunakan Metode Prewitt, Sobel dan Canny. *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, 1(1), 1–4.
- [9] I. Riadi, A. Yudhana, & W. Y. Sulisty. (2020) Analisis Perbandingan Nilai Kualitas Citra pada Metode Deteksi Tepi. *Rekayasa Sist. dan Teknol. Inf.*, 4(2), 345–351.

- [10] B. Niam & Q. Qirom. (2019). Deteksi Tulang Retak Dengan Metode Deteksi Tepi Prewitt. *Power Elektron. J. Orang Elektro*, 8(2), 25–28.
- [11] Putri Mentari Adhatil, Hendrick, Erlina Tati, Derisma. (2015). Rancang Bangun Alat Deteksi Uang Kertas Palsu Dengan Metode *Template Matching* Menggunakan Raspberry Pi. *Seminar Nasional Sains dan Teknologi 2015 Fakultas Teknik Universitas Muhammadiyah Jakarta*, 1-9.
- [12] M. Marengoni & D. Stringhini. (2011). High level computer vision using OpenCV. Proc. - 24th SIBGRAPI Conf. Graph. Patterns, Images Tutorials, SIBGRAPI-T 2011 (pp. 11–24)
- [13] J. Gao & H. Zhu. (2016). Moving Object Detection for Driving Assistance System. *Internet and Distributed Computing Systems*, 446–457,
- [14] G. Khoharja et al. (2017). Aplikasi Deteksi Nilai Uang pada Mata Uang Indonesia dengan Metode *Feature Matching*. *Jurnal INFRA*, 2–6.
- [15] D. Mistry & A. Banerjee. (2017). Comparison of Feature Detection and Matching Approaches: SIFT and SURF *GRD Journals- Glob. Res. Dev. J. Eng.*, 2(3), 7–13..
- [16] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. Proceedings of the 2011 International Conference on Computer Vision (pp. 2564-2571). Barcelona: IEEE
- [17] V. Faturhman, J. Kusnaendar, & Y. Wihardi. (2020). Perbandingan Algoritma Deteksi Fitur SIFT , SURF dan ORB dalam Proses Deteksi Objek Pada Video CCTV. *JATIKOM: Jurnal Aplikasi dan Teori Ilmu Komputer*, 3(2), 70–77.