

Sistem Deteksi Nominal Mata Uang Kertas dengan Metode Template Matching

Lailatul Fadilah¹, Adisa Dwi Wanti², Salma Zulfatul Latifah³
Program Studi Teknik Informatika,
Universitas Islam Negeri Maulana
Malik Ibrahim, Indonesia

¹19650032@student.uin-malang.ac.id;
²19650037@student.uin-malang.ac.id;
³19650038@student.uin-malang.ac.id

Abstrak

Uang kertas merupakan salah satu alat pembayaran sah yang terbuat dari kertas dengan gambar dan cap tertentu yang dikeluarkan oleh bank sentral/Bank Indonesia. Di Indonesia sendiri, Bank Indonesia mengeluarkan berbagai jenis dan model uang kertas, hal tersebut dimaksudkan agar kegiatan transaksi menjadi lebih mudah. Namun, di beberapa kasus, banyaknya jenis dan model uang kertas justru membuat suatu kegiatan menjadi terhambat. Salah satunya adalah kegiatan di kasir toko. Oleh karena itu, paper ini dibuat dengan tujuan memudahkan manusia dalam menjalankan aktivitas pekerjaannya seperti menjadi kasir dalam suatu toko.

Sistem Deteksi Nominal Mata Uang Kertas ini menggunakan metode Template Matching dalam implementasinya. Metode Template Matching adalah salah satu teknik image processing dimana citra dari sumber atau input akan dibandingkan dengan citra dari template untuk menemukan bagian yang sesuai dengan citra template. Dengan menggunakan metode Template Matching, maka akan diperoleh output berupa hasil deteksi nominal uang sesuai dengan gambar yang telah diinputkan.

1. Pendahuluan

Uang merupakan alat pembayaran yang sah dalam sebuah transaksi dan berlaku dalam

wilayah tertentu dan memiliki satuan hitung tertentu. Dalam kehidupan sehari-hari, uang memiliki beberapa fungsi untuk mempermudah kegiatan masyarakat sehari-hari, di antaranya adalah uang berfungsi sebagai alat penukar, sebagai alat pengukur nilai, sebagai alat penimbun kekayaan, dan sebagai komoditi bisnis (N.Lubis,-). Di dalam sejarah perkembangan uang, terdapat berbagai jenis uang yang beredar di masyarakat, jenis-jenis uang yang paling sering dijumpai antara lain uang kertas, uang logam, dan uang giral. Uang kertas merupakan salah satu alat pembayaran sah yang terbuat dari kertas dengan gambar dan cap tertentu yang dikeluarkan oleh bank sentral/Bank Indonesia. Uang kertas menjadi salah satu jenis uang yang paling banyak digunakan, hal tersebut karena jenis mata uang kertas ini lebih ringan dan mudah dibawa, sehingga lebih praktis. Di Indonesia sendiri, Bank Indonesia mengeluarkan berbagai jenis dan model uang kertas, hal tersebut dimaksudkan agar kegiatan atau proses transaksi menjadi lebih mudah. (Dwiki Aditama dkk, 2021).

Pada era saat ini, kegiatan atau aktivitas manusia sangat bergantung pada perkembangan teknologi. Dengan perkembangan teknologi yang semakin maju, maka aktivitas atau kegiatan manusia juga akan semakin mudah. Saat ini, berbagai jenis sistem automasi sudah banyak hadir dan menawarkan kemudahan untuk berbagai pekerjaan (Julian Fuad Fauzi dkk, 2018). Contoh sistem yang dapat mempermudah pekerjaan manusia adalah sistem deteksi nominal mata uang kertas dengan metode template matching. Banyak kondisi di lingkungan masyarakat yang membutuhkan pembacaan nominal uang kertas untuk memudahkan suatu kegiatan. Misalnya adalah deteksi nominal uang kertas di kasir toko. Sebelumnya, orang-orang harus secara manual melihat uang kertas untuk mengetahui nominalnya.

Nominal uang terletak di bagian pinggir uang (edges). Proses tersebut tergolong mudah untuk dilakukan, akan tetapi lebih mudah lagi jika prosesnya dibuat otomatis dengan bantuan program. Karena itulah project ini dibuat untuk menghasilkan prototype program sederhana yang dapat mendeteksi nominal uang dengan mengimplementasikan prinsip-prinsip keilmuan komputer vision.

Oleh karena itu, paper ini dibuat dengan tujuan untuk memudahkan manusia dalam menjalankan aktivitas pekerjaannya seperti menjadi kasir dalam suatu toko. Sistem ini dapat dikembangkan lagi untuk mendukung kemajuan teknologi dalam dunia penjualan makanan ringan, minuman dan beberapa barang lainnya yang dalam proses penjualannya tidak membutuhkan seorang karyawan dalam melayani konsumen. Sehingga pembayaran dapat dikelola oleh mesin, termasuk dalam hal membaca nominal pada uang kertas. Sehingga, paper ini bermanfaat bagi pembaca untuk bisa mendapatkan wawasan baru mengenai sistem deteksi nominal uang kertas dengan metode template matching. Untuk dapat mendeteksi nominal mata uang kertas, digunakan metode template matching. Template Matching merupakan salah satu teknik image processing dimana citra dari sumber atau input akan dibandingkan dengan citra dari template untuk menemukan bagian yang sesuai dengan citra template. Input yang digunakan dalam sistem ini adalah berupa gambar beberapa pecahan uang kertas yang nantinya akan dideteksi dan juga gambar template yang nantinya akan digunakan sebagai dataset untuk mendeteksi nominal uang kertas tersebut. Sedangkan output yang akan dihasilkan dari sistem ini adalah hasil deteksi nominal uang sesuai dengan gambar yang telah diinputkan. Misalkan gambar uang yang diinputkan adalah nominal 100.000, maka sistem akan mendeteksi dan mengeluarkan input

berupa nominal "Seratus ribu rupiah". Begitu pula dengan nominal-nominal yang lain.

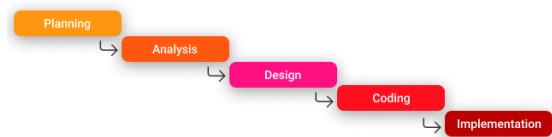
2. Ketersediaan Data

Dalam melakukan penelitian, langkah pertama yang perlu dilakukan adalah pengumpulan data. Data yang digunakan dalam penelitian ini berupa gambar nominal mata uang kertas rupiah bernilai 5000, 20000, dan 100000 pada mata uang kertas rupiah keluaran tahun 2016. Gambar tersebut akan digunakan sebagai template pada sistem untuk dilakukan perhitungan tingkat kecocokan dengan gambar uang yang akan dideteksi nominalnya.

Untuk mendapatkan informasi yang lebih akurat diperlukan sejumlah referensi untuk menunjang penulisan paper penelitian ini. Referensi yang digunakan berupa jurnal, artikel, dan tugas akhir yang diperoleh dari internet.

3. Desain System

Dalam penelitian ini, model penelitian yang digunakan untuk pengembangan sistem deteksi nominal pada mata uang kertas adalah SDLC (*System Development Life Cycle*). Gambaran singkat mengenai metode waterfall dapat dilihat pada gambar 1.



Gambar 1. Metode Waterfall

3.1 Perencanaan (Planning)

Alur kerja pada sistem setelah melakukan pengumpulan training berupa mata uang kertas rupiah adalah melakukan ekstraksi fitur edge pada gambar template menggunakan metode canny. Selanjutnya adalah tahap *preprocessing* pada gambar inputan. *Preprocessing* gambar berupa mengkonversi citra warna gambar inputan ke grayscale dan scaling gambar. Kegunaan scaling gambar pada tahap ini untuk mengantisipasi ukuran gambar

inputan tidak sama rata, sehingga perlu dilakukan *scaling* untuk menyamakan ukuran gambar yang akan dilakukan *template matching*. Hal ini bertujuan untuk mempermudah sistem dalam melakukan tahap selanjutnya. Setelah mendapat hasil dari tahap sebelumnya, akan dilakukan edge detection pada gambar inputan. Jika gambar input dapat dilakukan edge detection maka proses selanjutnya adalah *template matching*. *Template matching* merupakan proses perhitungan tingkat kemiripan citra pada gambar training dengan citra gambar template (Rahmad dkk, -). Tahap yang dilakukan untuk mengantisipasi tidak terbacanya max value pada saat proses *template matching* adalah dengan *thresholding*. *Thresholding* merupakan proses pemisahan *background* dan *foreground* pada gambar dengan cara melihat perbedaan intensitas warnanya. Apabila nilai intensitas pixel pada gambar lebih tinggi dari intensity threshold maka pixel diubah menjadi warna putih. Namun jika sebaliknya, maka pixel akan diubah menjadi warna hitam (Rahmad dkk, -). Dari hasil *thresholding* akan didapatkan sebuah nilai. Nilai tersebut akan dijadikan sebagai penentu output yang dikeluarkan sistem. Output yang dihasilkan sistem dari proses *template matching* dan *thresholding* berupa keterangan tulisan berupa keberhasilan deteksi nominal mata uang.

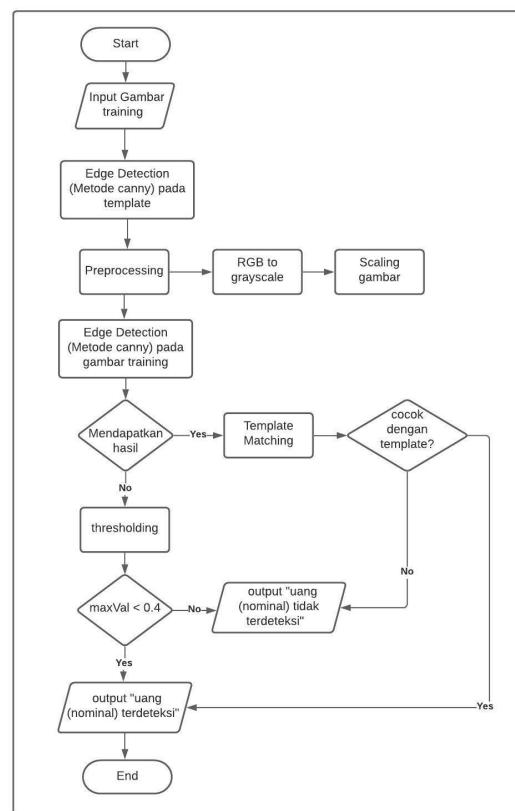
3.2 Analisis

Proses pengumpulan daftar kebutuhan sistem dilakukan secara intensif agar kebutuhan sistem dapat dipahami dengan mudah. *Requirement* yang dibutuhkan untuk mendukung perancangan sistem pada penelitian ini antara lain spesifikasi perangkat komputer/laptop, software, dataset dan bahasa pemrograman. Pemenuhan kebutuhan sistem dapat dilakukan secara berkala dan dilakukan dengan semaksimal mungkin agar

keberhasilan sistem dapat terwujud.

Spesifikasi perangkat yang digunakan berupa laptop ASUS FX505GD dengan memiliki 8 RAM dan menggunakan prosesor intel core i5 gen 8. Bahasa pemrograman yang digunakan pada sistem ini adalah phyton dengan menggunakan beberapa library yang disediakan didalamnya. Implementasi programnya akan dijalankan pada IDE pycharm. Pycharm merupakan IDE yang digunakan dalam bahasa pemrograman phyton. Beberapa library yang digunakan dalam sistem ini diantaranya cv2, numpy, glob. Terdapat satu class yang digunakan sebagai library dalam sistem ini, yaitu class imutils dengan ekstensi phyton.

3.3 Perancangan (Design)



Gambar 2. Desain Flowchart Sistem

Pada gambar 2 mengenai flowchart sistem dijelaskan bahwa tahap pertama yang dilakukan sistem adalah memasukkan gambar

input. Gambar tersebut akan dibaca oleh sistem kemudian akan dilakukan tahap *preprocessing* dimana terjadi proses perubahan citra warna gambar menjadi *grayscale* dan proses *scaling* gambar menggunakan metode canny. Namun sebelum melakukan *preprocessing* pada gambar training, sistem akan melakukan ekstraksi fitur edge (tepi) pada gambar template. Tahap selanjutnya setelah *preprocessing* adalah *edge detection* pada gambar input. Proses *edge detection* dalam sistem ini menggunakan metode canny. Hasil yang didapat pada proses *preprocessing* dan *edge detection* digunakan untuk melakukan *template matching*. Jika citra pada gambar training memiliki kecocokan dengan citra pada gambar template maka output yang dihasilkan berupa tulisan "Uang (nominal) terdeteksi". Jika tidak maka output yang dihasilkan berupa keterangan "uang (nominal) tidak terdeteksi". Untuk mengantisipasi nilai max tidak terinisialisasi pada proses *template matching*, maka dilakukan thresholding. Jika nilai max lebih dari 0.4 maka gambar diasumsikan memiliki kemiripan dengan citra gambar.

3.3 Coding

Perancangan kode program pada tahap ini menggunakan pseudocode. Penggunaan pseudo code pada tahap ini dikarenakan dapat mempermudah dalam memahami cara penyelesaian masalah. Selain itu dapat juga mempermudah manusia dalam membaca suatu algoritma atau bahasa pemrograman.

Berikut rancangan pseudocode pada sistem pendekripsi nilai mata uang pada mata uang kertas rupiah.

Deklarasi :

```
Template_data, image, imageP,
imageS, gray, imageP1, blob;
scale, resized, r, edged,
result, float;

found, rep, string;Files1,
file;
```

```
Result, template_matching;

Deskripsi :

Template_data <- []
Files <-
glob.glob('nm_folder/*.extensi
on')

FOR (myfile <- files1)
Image <- read(myfile)
Image <- changeColor(image,
color)
Image <- resize(image, width,
height)
template_data.append(image)
END FOR

FOR (tmp <- template_data)
(height, width) <-
tmp.shape[:2]
show("Template", tmp)
END FOR

FOR (imageP <-
glob.glob('nm_folder/*.extensi
on'))
imageS <- read(imageP)
gray <- changeColor(imageS,
color)
Found <- null

FOR (Scale <- linspace(0.2,
1.0, 20) [::-1])
Resized <- resize(gray, width =
int(gray.shape[1] * scale))
R <- gray.shape[1] /
float(resized.shape[1])

IF => resized.shape[0] <
height OR resized.shape[1] <
width THEN
BREAK
END IF

Edged <- resize(50, 200)
Result <- matchTemplate(edged,
tmp) #melakukan match
templating gambar template
dengan edge gambar uang

IF => Found is None OR
maximalValue > found[0] THEN
Found <- (maxValue, maxLoc, r)
IF => maximalValue >= 0.4 THEN
Rep <- "uang"
show("image", imageS)
```

```

Output ("Uang", keterangan
uang, "Terdeteksi")
END IF
END IF
END FOR
END FOR
END

```

3.4 Implementasi

1) Bahasa Pemrograman dan IDE yang Digunakan

Bahasa pemrograman yang digunakan pada sistem ini adalah phyton dengan menggunakan beberapa library yang disediakan didalamnya. Implementasi programnya akan dijalankan pada IDE pycharm.

2) Input

Input yang digunakan dalam sistem ini adalah berupa gambar beberapa pecahan uang kertas yang nantinya akan dideteksi dan juga gambar template yang nantinya akan digunakan sebagai dataset untuk mendeteksi nominal uang kertas tersebut. Gambar uang yang digunakan sebagai inputan hanya satu sisi saja (bagian depan). Contoh :



Gambar 3. Contoh template mata uang

Uang kertas yang digunakan dalam penelitian ini adalah uang dengan nominal 5000, 20000 dan 100000 rupiah.

3) Feature Extraction

Pada bagian feature, dilakukan edge detection.



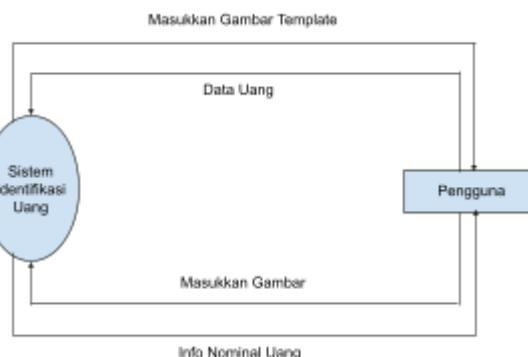
Gambar 4. Feature yang Diekstrak pada Tahap Feature Extraction

Dimana feature yang diekstrak dari gambar adalah bagian edge uang yang bertuliskan nominal. Ilustrasinya ditunjukkan oleh gambar 4 di atas.

4) Rencana Output

Output yang akan dihasilkan dari sistem ini adalah hasil deteksi nominal uang sesuai dengan gambar yang telah diinputkan. Misalkan gambar uang yang diinputkan adalah nominal 100.000, maka sistem akan mendeteksi dan mengeluarkan input berupa nominal "Seratus ribu rupiah". Begitu pula dengan nominal-nominal yang lain.

4) Desain Proses



Gambar 5. Desain proses sistem

User memasukkan gambar kedalam sistem identifikasi uang, yang kemudian akan diolah dan memberikan data uang berupa data RGB kepada user. Kemudian user memasukkan gambar yang berupa template ke sistem, memberikan info kepada user berupa hasilnya.

4. Metode Template Matching

Template Matching merupakan salah satu teknik image processing dimana citra dari sumber atau input akan dibandingkan dengan citra dari template untuk menemukan bagian yang sesuai dengan citra template. Gambar dari template biasanya berukuran lebih kecil daripada gambar inputan. Pada penggunaannya, metode template

matching biasanya digunakan untuk proses deteksi karakter, angka, benda-benda kecil, maupun hal-hal sederhana lainnya (Rahmad dkk,-).

Dalam implementasinya, metode template matching memiliki beberapa kelebihan dan kekurangan. Kelebihan dari metode template matching antara lain : template matching lebih unggul dalam proses deteksi bentuk, huruf, atau bentuk visual lainnya yang memerlukan kontak dengan bentuk-bentuk internal. Selain itu, output yang dihasilkan dari metode template matching relatif akurat, karena proses deteksi kesalahan dilakukan hingga ukuran piksel. Sedangkan untuk kelemahannya, metode template matching cenderung tidak akan mengenali objek dengan perbandingan eksternal objek dengan internal objek 1:1. Untuk itu, diperlukan banyak template yang spesifik agar objek bisa dikenali atau dideteksi (Zaenudin dkk, 2017).

5. Hasil Implementasi

Pada poin ini, disampaikan hasil penelitian berupa proses implementasi program yang telah dicapai yang dibantu dengan ilustrasi berupa tabel, grafik, hasil screenshot, dan bentuk ilustrasi lain. Penyampaian hasil akan disajikan dalam beberapa poin subbab.

Program diimplementasikan menggunakan perangkat keras dengan CPU Intel Core i-5, RAM 8GB. Sedangkan perangkat lunak yang digunakan adalah sistem operasi windows 10 64-bit, bahasa pemrograman python, dan IDE python pycharm community.

5.1 Input Dataset

Dataset yang digunakan sebagai data input program ada dua macam yaitu dataset template dan dataset test. Keduanya berupa gambar uang kertas rupiah.



Gambar 6. Dataset Template

Dataset template ditunjukkan oleh gambar 6 berupa gambar uang kertas bagian nominal yang telah didefinisikan warnanya menjadi kontras terang dan gelap.



Gambar 7. Dataset Test

Sedangkan data test ditunjukkan oleh gambar 7 merupakan gambar uang kertas rupiah baik dari sisi depan maupun belakang. Kedua dataset ini yang diinputkan ke dalam sistem yaitu dengan memasukkannya ke dalam folder terlebih dahulu. Kemudian folder tersebut dimasukkan ke dalam folder project dan diinisiasi ke dalam sistem menggunakan source code.

```
template_data=[]
files1= glob.glob('template/*.png')

for myfile in files1:
    image = cv2.imread(myfile)
```

Gambar 8. Proses Inisiasi Data Template

Gambar 8 di atas menunjukkan proses inisiasi folder dataset template ke dalam sistem.

```
for imageP in glob.glob('uang/*.png'):
    imageS = cv2.imread(imageP)
```

Gambar 9. Proses Inisiasi Folder Dataset Testing

Sedangkan gambar 9 menunjukkan proses inisiasi folder dataset testing ke dalam sistem.

5.2 Konversi Gambar RGB ke Grayscale

Dari gambar uang yang telah diinputkan ke dalam sistem, gambar uang kemudian dikonversi menjadi grayscale. Tujuan dari konversi ini adalah untuk memudahkan proses komputasi gambar (Dwiki Aditama dkk, 2021). Konversi RGB ke grayscale ini tidak hanya dilakukan kepada gambar uang dalam dataset test tetapi juga data template.

```
template_data=[]
files1= glob.glob('template/*.png')

for myfile in files1:
    image = cv2.imread(myfile)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = cv2.Canny(image, 50, 200)
    template_data.append(image)
```

Gambar 10. Proses Grayscaling Dataset Template

```
for imageP in glob.glob('uang/*.png'):
    imageS = cv2.imread(imageP)
    gray = cv2.cvtColor(imageS, cv2.COLOR_BGR2GRAY)
    found = None
```

Gambar 11. Proses Grayscaling Dataset Testing

Gambar 10 dan 11 menunjukkan konversi grayscale. Setelah gambar template diubah ke grayscale gambar di resize dengan ukuran 50 x 200 pixels, kemudian dimasukkan ke dalam list array `template_data` [].

```
image = cv2.Canny(image, 50, 200)
template_data.append(image)
```

Gambar 12. Proses Memasukkan Gambar ke arraylist `template_data`

Selain gambar template, gambar testing juga di load ke

dalam sistem dan dikonversi ke grayscale.

5.3 Scaling Gambar

Setelah dataset di grayscale, langkah implementasi selanjutnya adalah scaling gambar uang.

```
for scale in np.linspace(0.2, 1.0, 20)[::-1]:
    # Merescale gambar
    resized = imutils.resize(gray, width =
int(gray.shape[1] * scale))
    r = gray.shape[1] / float(resized.shape[1])
```

Gambar 13. Proses Scaling Gambar

Proses scaling diawali dengan membuat array dengan jarak yang sama (`np.linspace(0.2, 1.0, 20) [::-1]`). Kemudian gambar di rescale sesuai dengan skala dan rasio yang diberikan.

Setelah rescaling gambar, kemudian gambar di cek apakah size nya lebih kecil dari gambar template.

```
if resized.shape[0] < tH or resized.shape[1] < tW:
    break
```

Gambar 14. Proses Cek Size Gambar

Jika gambar yang telah di scaling lebih kecil size nya daripada gambar template maka break loop.

5.4 Feature Extraction (Edge Detection) dan Template Machine

Edge detection merupakan jenis feature extraction yang dilakukan dalam project ini. Feature yang diekstrak adalah bagian edge gambar uang dimana ada gambar nominal uang.

Gambar yang di proses dalam feature extraction adalah gambar uang yang telah melalui proses grayscale dan resized sehingga warna dan ukurannya seragam. Setelah di resized, edge detection pada gambar dilakukan menggunakan metode *Canny*.

Template matching disebut juga feature matching adalah sebuah metode yang melibatkan pendekripsi feature yang memiliki pendekripsi sama atau mirip (Dwiki Aditama dkk, 2021).

Template matching dilakukan antara dua data yaitu data gambar template dan data gambar uang.

```
edged = cv2.Canny(resized, 50, 200)
result = cv2.matchTemplate(edged, tmp,
cv2.TM_CCOEFF_NORMED)(_, maxVal, _, maxLoc) =
cv2.minMaxLoc(result)
```

Gambar 15. Proses Edge Detection

5.5 Thresholding

Thresholding merupakan proses pemisahan *background* dan *foreground* pada gambar dengan cara melihat perbedaan intensitas warnanya.

```
if found is None or maxVal > found[0]:
    found = (maxVal, maxLoc, r)
```

Gambar 16. Proses Template Machine

Yang pertama dilakukan adalah melihat apakah template telah ditemukan. Jika belum (variable 'found' is None), maka inisialisasi `found = (maxVal, maxLoc, r)`.

Selanjutnya adalah melihat apakah max Value nya lebih dari sama dengan 0.4. Dengan anggapan, jika kurang dari 0.4 maka gambar akan di *destruct*.

```
if maxVal >= 0.4:
    rep = "uang"
    imagePl = imageP
    imagePl = imagePl.replace(rep, "")
    imagePl = imagePl.replace("\\", "")
    imagePl = imagePl.replace(".png", "")
    print("Uang",imagePl,"Terdeteksi")
```

Gambar 17. Melakukan Thresholding

Jika max Value lebih dari sama dengan 0.4, maka dilakukan thresholding terhadap gambar. Langkah thresholding ini adalah berfungsi untuk mencocokkan gambar dengan template.

Kemudian hasil yang di dapat, di print dengan format "Uang (nominal) Terdeteksi oleh Sistem"

6. Hasil Eksperimen

Eksperimen dilakukan dengan melakukan running program terhadap sistem yang telah dibangun. Indikator keberhasilan eksperimen adalah sebagai berikut.

1. Sistem menampilkan gambar template yang telah melalui proses grayscale.
2. Sistem menampilkan gambar testing (uang kertas) yang *matched* / sesuai dengan gambar template yang muncul.
3. Sistem menampilkan nominal gambar uang yang dimunculkan dalam format tulisan / kalimat.

Setelah memperhatikan indikator tersebut, berikut adalah hasil eksperimen yang dapat.

Experimen Pertama

Pada eksperimen pertama dilakukan deteksi nominal pada uang kertas Rp 100.000



Gambar 18. Output Gambar Template yang Muncul



Gambar 19. Output Gambar Uang yang Muncul

Uang 100000 Terdeteksi

Gambar 20. Output Keterangan Nominal Uang yang Muncul

Experimen Kedua

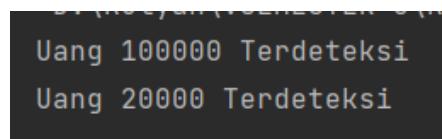
Pada eksperimen kedua dilakukan deteksi nominal pada uang kertas Rp 20.000



Gambar 21. Output Gambar Template yang Muncul



Gambar 22. Output Gambar Uang yang Muncul



Gambar 23. Output Keterangan Nominal Uang yang Muncul

Experimen Ketiga

Pada eksperimen ketiga dilakukan deteksi nominal pada uang kertas Rp 5000



Gambar 24. Output Gambar Template yang Muncul



Gambar 25. Output Gambar Uang yang Muncul

Uang 100000 Terdeteksi
Uang 20000 Terdeteksi
Uang 5000 Terdeteksi

Gambar 26. Output Keterangan Nominal Uang yang Muncul

7. Kesimpulan

Berdasarkan hasil implementasi dan eksperimen yang telah didapatkan, maka dapat disimpulkan bahwa metode template matching dengan feature extraction berjenis edge detection dapat memberikan keluaran yang tepat dan efektif untuk mendeteksi nominal uang kertas.

Untuk mendeteksi nominal uang kertas menggunakan template matching dilakukan bertahap yaitu mulai dari menyiapkan dataset template dan dataset testing (gambar uang). Kemudian melatih data template dan mengenalkan data testing. Selanjutnya disusul proses Grayscale, Scaling / Resizing, Feature Extraction (dalam hal ini Edge Detection), Template Matching.

Dengan menggunakan metode Template Matching, maka akan diperoleh output berupa hasil deteksi nominal uang sesuai dengan gambar yang telah diinputkan.

8. Saran

Untuk pengembangan selanjutnya, output dari sistem bisa dimodifikasi ke bentuk suara sehingga bisa mencapai sasaran kebermanfaatan yang lebih luas lagi. Selain itu, penggunaan gambar dengan kualitas resolusi yang lebih tinggi juga disarankan agar pendekripsi objek dapat memberikan hasil yang lebih akurat.

Daftar Pustaka

Aditama S., Hilal, M.,dkk. (2021). PERBANDINGAN KOMBINASI METODE TEMPLATE MATCHING DAN ALGORITMA FEATURE MATCHING PADA PENGENALAN MATA UANG INDIA. XVI, 47-54.

- Fauzi, J. F., Tolle, H., & Dewi, R. K. (2018). Implementasi Metode RGB To HSV pada Aplikasi Pengenalan Mata Uang Kertas Berbasis Android untuk Tuna Netra. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(6), 2319-2325.
- Lubis, N. (n.d.). Pengertian, Fungsi, Jenis dan Nilai Uang. 1-40.
<http://repository.ut.ac.id/3827/1/ADBI4331-M1.pdf>
- Rahmad, C., Rismanto, R., & Pranata, F. D. (n.d.). Pengenalan Nilai Mata Uang Kertas Untuk Tunanetra Menggunakan Metode Template Matching Correlation Berbasis Android. 73-80.
- Zaenudin, C. A., Novamizanti, L., & Saidah, S. (2017). Perancangan Alat Identifikasi Keaslian Dan Nominal Mata Uang Kertas Real Time Untuk Penyandang Tunanetra Berbasis Template Matching Dengan Raspberry Pi