ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO) DENGAN METODE *TEMPLATE MATCHING* DAN *K-NEAREST NEIGHBORS* (KNN)

Andika Dicky Saputra[1], Jayanta[2], Ing. Artambo B. Pangaribuan[3]
Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
andikadiicky@gmail.com[1], jayanta@upnvj.ac.id[2], artambo@upnvj.ac.id[3]

Abstrak. Untuk membentuk sebuah hubungan interaksi yang baik antar individu dibutuhkan suatu komunikasi yang baik yaitu menggunakan bahasa. Berbeda dengan penderita tuna rungu yang berkomunikasi menggunakan bahasa isyarat. Untuk berkomunikasi dengan orang normal yang tidak memahami bahasa isyarat membutuhkan perantara untuk menerjemahkan bahasa isyarat. Penelitian ini bertujuan untuk mendeteksi abjad isyarat statis dan dinamis lalu mengklasifikasikannya sehingga output nya adalah sebuah teks yang dapat dipahami semua orang. Penelitian ini menggunakan metode template matching dan algoritma KNN. Data yang digunakan merupakan hasil ekstraksi keyframe video yang terdiri dari 136 gambar template dan 17 gambar uji. Data tersebut kemudian dilakukan uji kecocokan menggunakan metode template matching dan tahap akhir klasifikasi menggunakan KNN. Pada tahap uji kecocokan mendapatkan hasil sebesar 85.04% untuk abjad isyarat statis, sedangkan abjad isyarat dinamis sebesar 84.65%. Untuk klasifikasi KNN didapatkan akurasi sebesar 96.52%, sehingga penelitian ini berhasil melakukan klasifikasi abjad isyarat statis dan dinamis.

Kata kunci: Komunikasi, Bahasa Isyarat Indonesia (BISINDO), *Template Matching*, *K-Nearest Neighbors*.

1 Pendahuluan

Komunikasi ialah sebuah hal yang amat mendasar dalam menjalani kegiatan sehari-hari. Karena manusia sejatinya adalah makhluk yang membutuhkan keberadaan manusia lain dan melakukan interaksi sosial, dan dalam kehidupan sosial tersebut dibutuhkan suatu komunikasi. Untuk membentuk sebuah hubungan interaksi yang baik antar individu maka dibutuhkan suatu komunikasi yang baik pula yaitu dengan menggunakan bahasa. Dalam kehidupan manusia tidak semuanya memiliki kesanggupan untuk melakukan komunikasi, contohnya teman tuna rungu, mereka yang mengidap keterbatasan tersebut memerlukan upaya yang lebih untuk dapat berkomunikasi dengan sesama mereka atau dengan manusia normal lainnya, salah satu caranya dengan menggunakan bahasa non-verbal yakni bahasa isyarat.

Untuk berkomunikasi dengan orang normal yang tidak mengetahui gerakan abjad isyarat biasanya membutuhkan seorang perantara untuk memberitahukan gerakan abjad isyarat yang dimaksudkan. Oleh sebab itu, untuk mempermudah berkomunikasi antara orang normal dan teman tuna rungu diperlukan sebuah program yang dapat menerjemahkan gerakan abjad isyarat statis dan gerakan abjad isyarat dinamis lalu mengklasifikannya sehingga didapatkan output berupa text yang dapat dipahami oleh orang pada umumnya.

Dalam penelitian ini akan membahas mengenai klasifikasi abjad Bahasa isyarat statis dan dinamis, sebelum dilakukan tahap pengujian, terdapat tahap akuisisi data yang menggunakan input video dan diproses dengan ekstraksi keyframe video untuk mendapatkan frame gambar. Setelah itu akan melewati tahap pra-proses yaitu

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

grayscale, *noise removal*, *resize* dan *edge detection*. Kemudian akan dilanjutkan tahap uji kecocokan menggunakan metode template matching lalu hasil dari uji kecocokan tersebut akan digunakan dalam klasifikasi KNN, sehingga hasil akhir yang didapat adalah klasifikasi gerakan abjad isyarat statis dan dinamis.

1.1 **Tujuan Penelitian**

Mengimplementasikan metode *template matching* dan algoritma KNN untuk melakukan klasifikasi abjad isyarat statis dan dinamis.

1.2 **Kegunaan Penelitian**

Menghasilkan sebuah program dengan akurasi tinggi yang dapat melakukan klasifikasi abjad isyarat statis dan dinamis.

2 Landasan Teori

Bahasa isyarat merupakan istilah umum yang mengacu pada gestural atau bahasa visual yang menggunakan gerakan-gerakan jari tangan, tangan, lengan, serta gerakan mata, wajah, kepala dan badan [1].

BISINDO merupakan bahasa komunikasi yang secara alami dilakukan antara penderita tuna rungu dengan lingkungannya oleh sebab itu menimbulkan adanya istilah Bahasa Isyarat Indonesia (BISINDO). Sedangkan Sistem Isyarat Bahasa Indonesia (SIBI) adalah salah satu komunikasi bahasa isyarat yang dimiliki oleh negara Indonesia. SIBI dibangun dengan mengadopsi dari bahasa isyarat *American Sign Language* (ASL) yang dimiliki oleh negara Amerika [2].

Edge detection merupakan sebuah metode dalam *image processing* untuk mencari tepi objek dalam sebuah gambar. Cara kerja nya adalah dengan mendeteksi diskontinuitas dalam kecerahan. Deteksi tepi ini digunakan untuk segmentasi gambar dan ekstraksi data di berbagai bidang seperti *image processing*, *computer vision* dan *machine vision* [3].

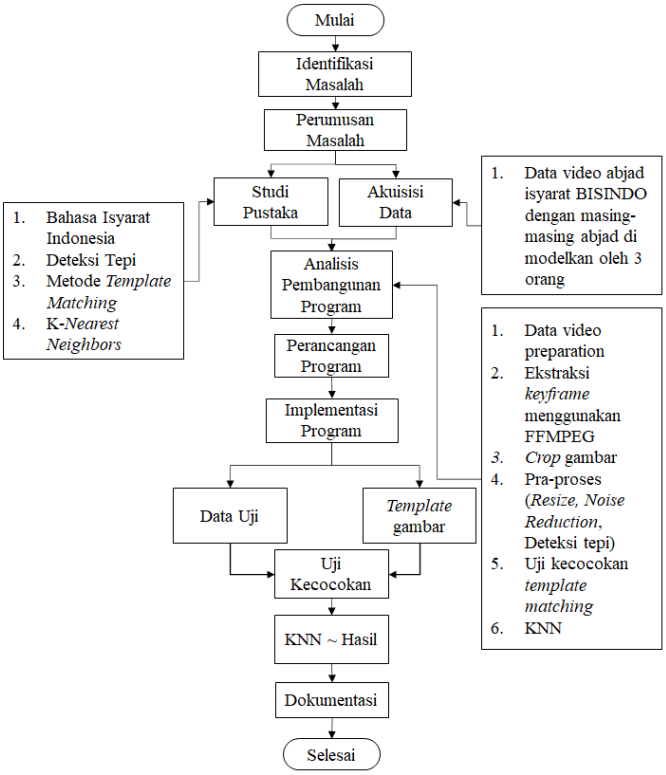
Template matching merupakan salah satu teknik dalam image processing yang memiliki proses yaitu membandingkan ciri yang ada pada citra dengan ciri yang terdapat pada template [4]. Dalam penelitian ini *template matching* digunakan untuk melakukan uji kecocokan antara gambar uji dengan gambar *template*.

Algoritma *K-Nearest Neighbors* (KNN) merupakan sebuah metode untuk setiap piksel yang belum terklasifikasi, dimana K piksel terdekat dicari di daerah-darah sampel. Kemudian piksel tersebut dimasukkan ke kelas mayoritas dari K piksel terdekat. Jauh dekatnya jarak tersebut dapat dihitung berdasarkan *euclidean distance* [5].

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

3 Metodologi Penelitian

Penelitian ini memiliki beberapa tahapan yang dimulai dari merumuskan masalah, studi pustaka, akuisisi data, pra-proses kemudian melakukan pencocokan gambar dengan menggunakan metode *template matching*, lalu hasil dari uji kecocokan *template matching* akan digunakan untuk melakukan perhitungan jarak dan dari hasil perhitungan jarak tersebut akan dilakukan klasifikasi berdasarkan nilai K yang sudah ditentukan yaitu 1, 3, 5, 7 dan 9. Dalam penelitian yang dilakukan oleh penulis, program dirancang supaya dapat mengklasifikasikan gerakan abjad bahasa isyarat indonesia. Abjad yang digunakan dalam penelitian ini hanya A, D, E, I, J, M, N, O, R, U, dan Y. Abjad yang dipilih merupakan abjad yang termasuk dalam huruf vokal dan konsonan. Dalam abjad-abjad tersebut juga terdapat abjad isyarat statis dan dinamis. Dibawah ini merupakan gambar dari alur penelitian:



Gambar. 1. Alur penelitian

3.1 Studi Pustaka

Dalam tahap ini peneliti melakukan pencarian informasi yang berguna dalam mendukung kegiatan yang berkaitan dengan bahasa isyarat, metode *template matching*, algoritma KNN dan berbagai informasi lain yang berhubungan dengan kegiatan ini, sumber pembelajaran ini didapatkan dari jurnal penelitian, website dan buku.

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

3.2 **Pengumpulan Data**

Dalam tahap ini peneliti melakukan pengumpulan data yang akan digunakan dalam penelitian, data yang dikumpulkan berupa 99 video gerakan abjad isyarat yang diperankan oleh 3 orang model. Dari pengumpulan video tersebut akan dilanjutkan dengan ekstraksi keyframe video dan penyortiran data gambar sehingga jumlah data gambar akhir berjumlah 153 gambar.

3.3 **Perancangan**

Setelah hal-hal seperti input, output, program, prosedur, software, dan hardware sudah terkumpul dengan baik, sistem kemudian akan melakukan pengolahan, dimulai dengan ekstraksi keyframe video. Setelah didapatkan hasil dari ekstraksi keyframe selanjutnya melalui tahap pra-proses, kemudian citra yang telah melalui tahap pra-proses akan dilakukan pengujian menggunakan metode *template matching* dan algoritma KNN apakah citra hasil ekstraksi keyframe video tersebut sudah sesuai dan dapat diklasifikasi sebagai salah satu abjad isyarat.

4. Hasil dan Pembahasan

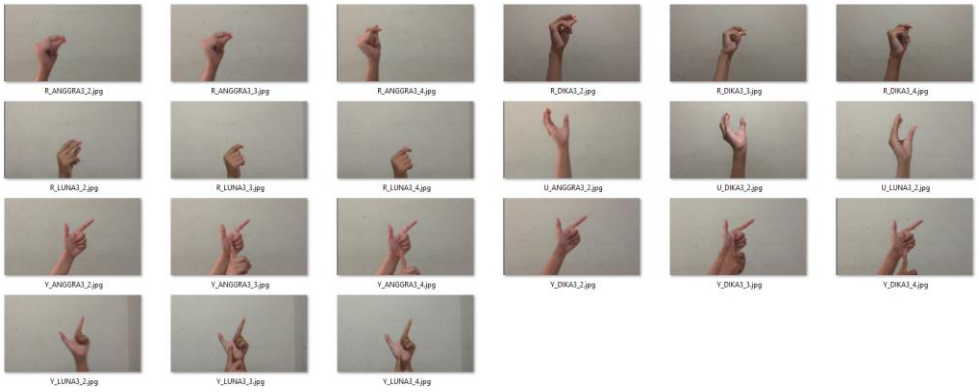
Hasil dari pengumpulan data berupa video gerakan abjad isyarat akan dilanjutkan dengan melakukan beberapa tahapan antara lain ekstraksi *keyframe video*, pra-proses, uji kecocokan *template matching* dan terakhir klasifikasi menggunakan algoritma KNN.

4.1 **Pengumpulan Data File**

Pada tahapan ini penulis akan memberikan sampel tampilan gerakan abjad isyarat. Berikut ini adalah contoh tampilan model dari tiap abjad isyarat. Masing-masing abjad isyarat diperankan oleh 3 orang model, sehingga total data video yang dikumpulkan sebanyak 99 video file.

Gambar. 2. Tampilan Data Video

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020



Gambar. 3. Tampilan Data Video

4.2 **Ekstraksi Keyframe Video**

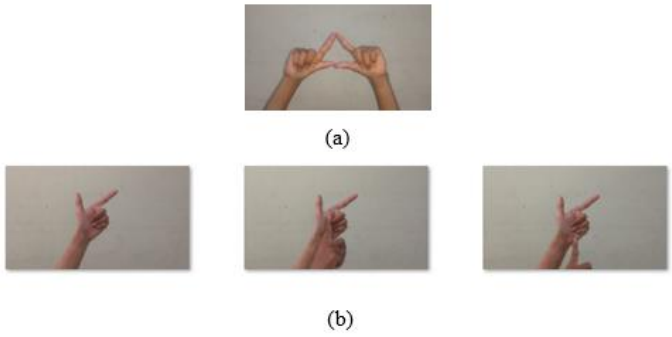
Karena video merupakan kumpulan gambar yang dirangkai dalam suatu waktu. Gambar-gambar tersebut dinamakan frame dan *frame-frame* tersebut dimainkan dengan kecepatan tinggi dan tetap, sehingga menciptakan sebuah ilusi gerak. Program yang digunakan dalam ekstraksi *keyframe* ini menggunakan *python package* bernama Video-kf. Video-kf dapat dijalankan baik dari *command-line*, atau dari dalam *python* dengan mengimpornya. Video-kf akan melakukan ekstraksi *keyframe* berdasarkan metode yang dipilih, penulis memilih menggunakan metode *flow*. Metode ini akan mengembalikan *frame* yang paling stabil berdasarkan *frame-frame* sebelumnya dari setiap *shot-sequence*. Oleh karena itu penulis dalam tahapan ini melakukan ekstraksi *keyframe* untuk mendapatkan *frame* dari video dengan jumlah yang lebih sedikit dari jumlah sebenarnya agar mempercepat proses pengolahan citra. Gambar dibawah ini merupakan contoh hasil ekstraksi video *keyframe* pada gerakan abjad A dan Y.

Gambar. 4. Data Ekstraksi Video *Keyframe* Abjad A (Abjad Statis)

Gambar. 5. Data Ekstraksi Video *Keyframe* Abjad Y (Abjad Dinamis)

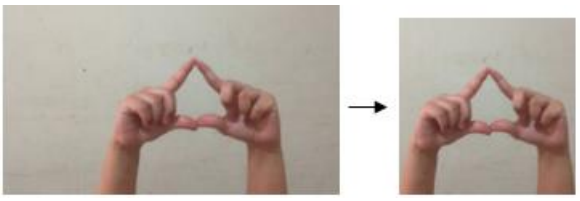
ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

Pada kedua gambar diatas merupakan contoh tampilan data video setelah dilakukan ekstraksi *keyframe*. Hasil dari masing-masing ekstraksi video *keyframe* menghasilkan 4 gambar. Sehingga total gambar yang dihasilkan dari ekstraksi video *keyframe* sebanyak 396 gambar. Namun dari semua gambar yang dihasilkan pada ekstraksi *keyframe* terdapat gambar yang tidak memiliki objek tangan sehingga gambar tersebut tidak akan digunakan, karena dianggap tidak memiliki arti apa-apa. Oleh karena itu penulis melakukan penyortiran citra hasil ekstraksi *keyframe* dan didapatkan total gambar sebanyak 153.



Gambar. 6. (a) Hasil Penyortiran Gambar Abjad Statis; (b) Hasil Penyortiran Gambar Abjad Dinamis

Dapat dilihat pada Gambar 6 yang sebelumnya pada hasil ekstraksi *keyframe* terdapat gambar yang hanya background putih saja dan gambar dengan objek tangan yang terdapat goncangan dihilangkan dan menyisakan gambar dengan kualitas baik untuk selanjutnya dilakukan pra-proses. Karena gambar yang dihasilkan setelah ekstraksi *keyframe* masih berdimensi 1920x1080 maka sebelum dilakukan pengujian, gambar-gambar hasil ekstraksi *keyframe* akan dilakukan *cropping* ke dalam ukuran 1080x1080 terlebih dahulu agar objek tangan menjadi lebih jelas terlihat dengan menggunakan bantuan software *Adobe Photoshop CC* 2017. Berikut ini adalah contoh tampilan sebelum dan sesudah *cropping*.



Gambar. 7. Contoh Hasil *Cropping* Gambar

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

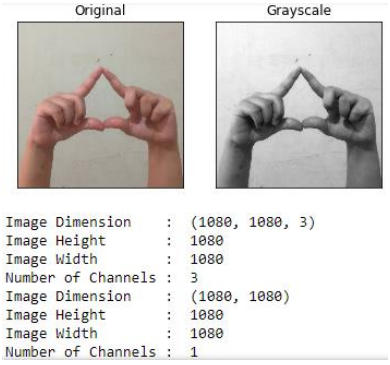
4.3 **Pra-proses**

Dalam praktiknya, metode yang digunakan untuk pra-proses ada berbagai macam, disesuaikan dengan kebutuhan penulis diantaranya yaitu:

1. *Grayscale* untuk mengubah channel warna RGB menjadi grayscale
2. *Resize* merupakan tahapan untuk memperkecil atau memperbesar ukuran citra.
3. *Noise Reduction* merupakan tahapan untuk mengurangi noise pada citra sebelum dilakukan deteksi tepi.
4. Deteksi tepi (metode *Canny*) merupakan sebuah tahapan untuk mendapatkan tampak garis batas suatu objek dalam citra, dalam penelitian ini objek tersebut adalah tangan.

4.3.1 ***Grayscale***

Setelah data gambar pada folder *template* telah berhasil terbaca secara keseluruhan, tahap selanjutnya adalah melakukan pra-proses dengan tahapan awal yaitu perubahan ruang warna rgb menjadi *grayscale*.



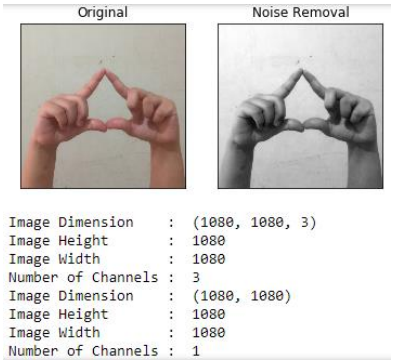
Gambar. 8. Hasil Grayscale Gambar

Gambar diatas merupakan hasil setelah dilakukan konversi gambar berwarna menjadi *grayscale*, dapat dilihat bahwa inputan gambar memiliki *size* sebesar 1080x1080 dan memiliki channel warna senilai 3 yaitu berwarna. Sedangkan setelah dilakukan konversi menjadi *grayscale* size gambar masih tetap sama namun *channel* warna sudah berubah menjadi 1 yang berarti gambar tersebut telah berhasil diubah menjadi *grayscale*.

4.3.2 ***Noise Removal***

Setelah dilakukan perubahan gambar berwarna menjadi *grayscale*, tahap selanjutnya adalah melakukan *noise removal*, hal ini dilakukan untuk mendapatkan hasil yang baik untuk kemudian dilakukan deteksi tepi dan uji kecocokan *template matching*.

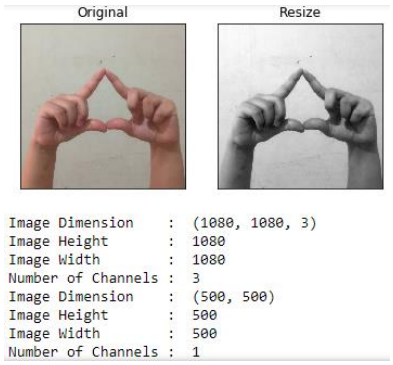
ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020



Gambar. 9. Hasil *Noise Removal* Gambar

4.3.3 ***Resize***

Setelah dilakukan *noise removal* yang ditunjukkan pada Gambar 9, pada tahapan ini gambar akan dilakukan *resizing*. *Resize* yang dilakukan adalah memperkecil ukuran gambar, hal ini dilakukan agar mempercepat proses pengolahan citra.



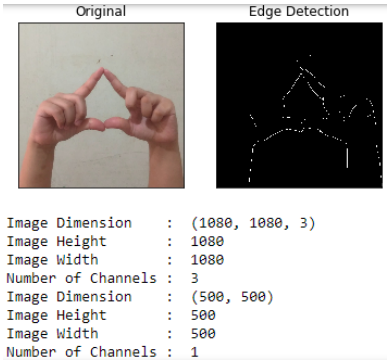
Gambar. 10. Hasil *Resize* Gambar

Dapat dilihat pada gambar diatas bahwa proses *resize* telah berhasil dilakukan, pada gambar original memiliki ukuran sebesar 1080x1080 sedangkan pada gambar setelah di *resize* memiliki ukuran sebesar 500x500.

4.3.4 ***Edge Detection***

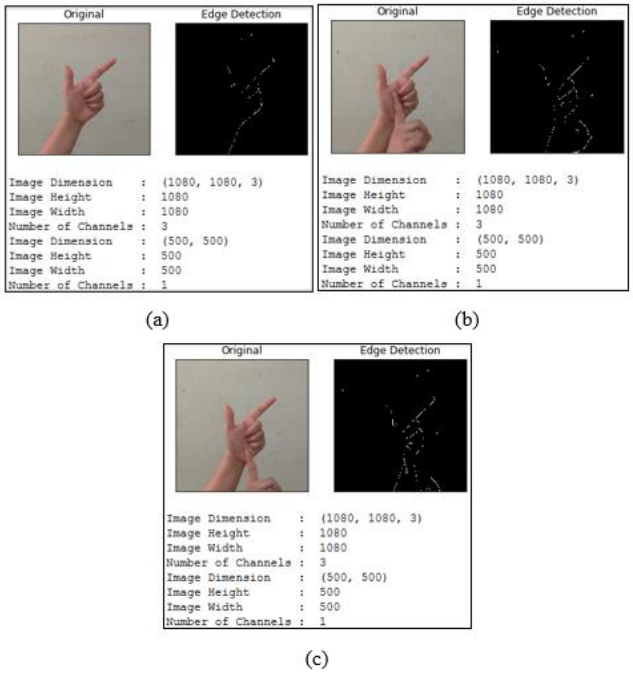
Setelah proses *resize* berhasil dilakukan, tahap selanjutnya adalah melakukan deteksi tepi. Deteksi tepi merupakan salah satu operasi dasar ketika melakukan pemrosesan gambar, tahap ini membantu menyeleksi objek tangan dari *background* yang sedang memperagakan gerakan abjad isyarat. Dibawah ini merupakan hasil dari deteksi tepi menggunakan fungsi “cv2.Canny”.

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020



Gambar. 11. Hasil *Edge Detection* Gambar

Karena tahapan *edge detection* diatas hanya menyontohkan hasil untuk abjad isyarat statis A, oleh karena itu akan disajikan contoh hasil dari *edge detection* untuk abjad isyarat dinamis (abjad Y).



Gambar. 12. (a) Hasil *edge detection* Y Fase 1, (b) Hasil *edge detection* Y Fase 2, (c) Hasil *edge detection* Y Fase 3

Gambar diatas merupakan contoh tampilan dari hasil *edge detection* untuk abjad dinamis, dari serangkaian proses yang sudah dijalankan sebelumnya, lalu akan dilakukan tahap uji kecocokan menggunakan *template matching*.

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

4.4 ***Template matching***

Setelah gambar uji melewati tahapan pra-proses, gambar tersebut selanjutnya akan dilakukan uji kecocokan menggunakan *template matching*, tahapan ini akan dilakukan selama 17 kali. Karena gambar yang akan diuji juga berjumlah 17 gambar. Dari hasil uji kecocokan gambar "A_ANGGRA3_3.jpg" dengan *template matching* menggunakan metode "cv2.TM_COEFF_NORMED" didapatkan persentase kecocokan gambar yaitu 85.76%.

Table 1. Hasil Uji Kecocokan *Template matching* Abjad Statis

Abjad Statis	Data Uji	Hasil Uji Kecocokan	Kelas Sebenarnya	Persentase Kecocokan
A	A_ANGGRA3_3.jpg	A	A	85.76%
D	D_ANGGRA3_3.jpg	D	D	84.52%
E	E_ANGGRA3_2.jpg	E	E	82.06%
I	I_ANGGRA3_3.jpg	-	I	-
M	M_ANGGRA3_3.jpg	-	M	-
N	N_ANGGRA3_3.jpg	N	N	84.08%
O	O_ANGGRA3_2.jpg	O	O	92.06%
U	U_ANGGRA3_2.jpg	U	U	81.78%
Rata-rata Uji Kecocokan Abjad Statis				85.04%

Tabel 1 merupakan hasil dari uji kecocokan gambar menggunakan *template matching*, dapat dilihat pada tabel diatas metode *template matching* sudah cukup baik dalam membandingkan gambar uji dengan gambar template yang sudah disediakan penulis. Dari tabel diatas dapat dilihat persentase uji kecocokan keseluruhan abjad statis sebesar 85.04%. Namun terdapat abjad yang tidak terdeteksi sebagai abjad apapun yaitu abjad I dan M, hal tersebut bisa dikarenakan posisi tangan yang sedang memperagakan gerakan abjad isyarat belum terbentuk dengan sempurna.

Table 2. Hasil Uji Kecocokan *Template matching* Abjad Dinamis

Abjad Dinamis	Data Uji	Hasil Uji Kecocokan	Kelas Sebenarnya	Persentase Kecocokan
J Fase 1	J_ANGGRA3_2.jpg	J Fase 1	J Fase 1	84.31%
J Fase 2	J_ANGGRA3_3.jpg	J Fase 2	J Fase 2	80.60%
J Fase 3	J_ANGGRA3_4.jpg	J Fase 3	J Fase 3	82.59%

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

R Fase 1	R_ANGGRA3_2.jpg	Y Fase 1	R Fase 1	-
R Fase 2	R_ANGGRA3_3.jpg	R Fase 2	R Fase 2	91.52%
R Fase 3	R_ANGGRA3_4.jpg	*no match*	R Fase 3	-
Y Fase 1	Y_ANGGRA3_2.jpg	*Y Fase 3*	Y Fase 1	-
Y Fase 2	Y_ANGGRA3_3.jpg	*Y Fase 3*	Y Fase 2	-
Y Fase 3	Y_ANGGRA3_4.jpg	*Y Fase 3*	Y Fase 3	84.25%
	Rata-rata Uji Kecocokan Abjad Dinamis			84.65%

Table 2 merupakan hasil dari uji kecocokan gambar menggunakan *template matching*, dapat dilihat pada tabel diatas metode *template matching* sudah cukup baik dalam membandingkan gambar uji dengan gambar yang sudah disediakan oleh peneliti, dari hasil uji kecocokan abjad dinamis diatas menghasil persentase kecocokan sebesar 84.65%. Karena abjad dinamis merupakan abjad yang yang dilakukan dengan gerakan oleh karena itu peneliti membagi abjad dinamis ini menjadi 3 fase gerakan, dan dalam pengujian kecocokan *template matching* ini terdapat beberapa fase gerakan yang tidak terdeteksi sebagai fase gerakan abjad dinamis yaitu R Fase 1 yang terdeteksi sebagai Y Fase 1, R Fase 3 tidak terdeteksi sebagai fase gerakan abjad dinamis apapun, Y Fase 1 terdeteksi sebagai Y Fase 3, dan Y Fase 2 terdeteksi sebagai Y Fase 3. Dari hasil uji kecocokan *template matching* didapatkan 11 abjad yang terdeteksi cocok dengan template dari jumlah keseluruhan gambar uji sebanyak 17.

4.5 **Klasifikasi KNN**

Setelah didapatkan persentase kecocokan gambar dari *template matching*, tahap selanjutnya adalah menghitung jarak.

```
#A.Hitung Selisih / Bobot
DISTANCE=list()
Target=100
for i in range(0, index):
    fototraining=PATH[i]
    bobot=MBobot[i]
    r=Target-bobot
    print(fototraining+' =>' +str(Target) +' - ' +str(bobot+0) +' = '+str(r))
    DISTANCE.append(r)
```

output yang dihasilkan dari kode tersebut adalah sebagai berikut:

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

```
1.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA1_3.jpg =>100 - [[90.755135]] = [[9.244865]]
2.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA2_3.jpg =>100 - [[85.76239]] = [[14.23761]]
3.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA1_3.jpg =>100 - [[58.17752]] = [[41.82248]]
4.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA2_3.jpg =>100 - [[66.14273]] = [[33.85727]]
5.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA3_3.jpg =>100 - [[68.627304]] = [[31.372696]]
6.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA1_3.jpg =>100 - [[64.6167]] = [[35.3833]]
7.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA2_3.jpg =>100 - [[66.93967]] = [[33.060333]]
8.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA3_3.jpg =>100 - [[62.43614]] = [[37.56386]]
9.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_ANGGRA1_3.jpg =>100 - [[46.280445]] = [[53.719555]]
10. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_ANGGRA2_3.jpg =>100 - [[52.56423]] = [[47.43577]]
11. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA1_3.jpg =>100 - [[64.19384]] = [[35.80616]]
12. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA2_3.jpg =>100 - [[60.797565]] = [[39.202435]]
13. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA3_3.jpg =>100 - [[46.797436]] = [[53.202564]]
14. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA1_3.jpg =>100 - [[54.20435]] = [[45.79565]]
15. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA2_3.jpg =>100 - [[52.682854]] = [[47.317146]]
    ----------------------------------------------------------------------------------------------------

136. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\YFase3_LUNA3_4.jpg =>100 - [[27.311266]] = [[72.688736]]
```

Kode diatas akan melakukan selisih dari list() hasil uji kecocokan *template matching*, hasil selisih akan dicetak menggunakan fungsi print(fototraining) dan akan ditambahkan kedalam sebuah array (r).

```
#B.Cetak Sebelum Diurutkan KNN
for i in range(0, index):
    fototraining=PATH[i]
    r=DISTANCE[i]
    init=TITLE[i]
    print(fototraining+ ' => ' +str(r) +' -> ' +str(init))
```

Kode diatas hanya mencetak hasil dari perhitungan selisih jarak, *output* dari kode diatas adalah sebagai berikut:

```
1.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA1_3.jpg => [[9.244865]] -> A
2.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA2_3.jpg => [[14.23761]] -> A
3.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA1_3.jpg => [[41.82248]] -> A
4.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA2_3.jpg => [[33.85727]] -> A
5.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA3_3.jpg => [[31.372696]] -> A
6.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA1_3.jpg => [[35.3833]] -> A
7.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA2_3.jpg => [[33.060333]] -> A
8.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA3_3.jpg => [[37.56386]] -> A
9.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_ANGGRA1_3.jpg => [[53.719555]] -> D
10. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_ANGGRA2_3.jpg => [[47.43577]] -> D
11. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA1_3.jpg => [[35.80616]] -> D
12. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA2_3.jpg => [[39.202435]] -> D
13. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA3_3.jpg => [[53.202564]] -> D
14. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA1_3.jpg => [[45.79565]] -> D
15. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA2_3.jpg => [[47.317146]] -> D
    ----------------------------------------------------------------------------------------------------

136. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\YFase3_LUNA3_4.jpg => [[72.688736]] -> YFase3
```

Dapat dilihat pada kode diatas menggunakan input dari hasil kode sebelumnya, yang membedakannya adalah pada kode ini hanya mencetak hasil selisih nya saja, namun hasil selisih tersebut masih belum berurutan dari yang terkecil ke terbesar. Sorting akan dilakukan menggunakan kode dibawah ini.

```
#Sorting hasil menggunakan Bubblesort
```

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

```
for i in range(len(DISTANCE)):
    for j in range(len(DISTANCE) - 1):
        if DISTANCE[j] > DISTANCE[j+1]:
            DISTANCE[j], DISTANCE[j+1] = DISTANCE[j+1], DISTANCE[j]
            MBobot[j], MBobot[j+1] = MBobot[j+1], MBobot[j]
            TITLE[j], TITLE[j+1] = TITLE[j+1], TITLE[j]
            PATH[j], PATH[j+1] = PATH[j+1], PATH[j]
#C.Cetak Setelah Diurutkan KNN
for i in range(0, index):
    fototraining=PATH[i]
    r=DISTANCE[i]
    init=TITLE[i]
    print(fototraining+ ' => ' +str(r) +' -> ' +str(init))
```

Kode diatas akan melakukan sorting menggunakan bubblesort yaitu pengurutan dari angka terkecil ke terbesar, hasil dari menjalankan kode diatas adalah sebagai berikut:

```
1.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA1_3.jpg => [[9.244865]] -> A
2.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_ANGGRA2_3.jpg => [[14.23761]] -> A
3.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA3_3.jpg => [[31.372696]] -> A
4.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA2_3.jpg => [[33.060333]] -> A
5.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA2_3.jpg => [[33.85727]] -> A
6.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA1_3.jpg => [[35.3833]] -> A
7.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA1_3.jpg => [[35.80616]] -> D
8.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_LUNA3_3.jpg => [[37.56386]] -> A
9.  C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_DIKA2_3.jpg => [[39.202435]] -> D
10. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\A_DIKA1_3.jpg => [[41.82248]] -> A
11. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\E_LUNA3_2.jpg => [[44.06612]] -> E
12. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\E_LUNA2_2.jpg => [[45.40384]] -> E
13. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA1_3.jpg => [[45.79565]] -> D
14. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA3_3.jpg => [[47.047405]] -> D
15. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\D_LUNA2_3.jpg => [[47.317146]] -> D
    ---------------------------------------------------------------------------------

136. C:\Users\Andika\DATA_SKRIPSI\Sorted_Data\DATA_TRAIN\O_LUNA2_2.jpg => [[91.43446]] -> O
```

Dapat dilihat pada output diatas, hasil pengurutan telah berhasil dilakukan, hasil dari pengurutan dari terkecil ke terbesar ini yang akan digunakan dalam klasifikasi KNN. *Output* dari klasifikasi KNN untuk sampel uji gambar “A_ANGGRA3_3.jpg” adalah sebagai berikut:

Hasil KNN Terdefinisi Sebagai Huruf A Dengan Jumlah 7 Gambar Yang Sesuai Dengan Template Gambar

ISBN 978-623-93343-1-4
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 14 Agustus 2020

Pada proses klasifikasi dengan KNN ini penulis akan menggunakan 5 nilai K yaitu 1, 3, 5, 7 dan 9. Kemudian akan ditarik sebuah kesimpulan, manakah nilai K yang menghasilkan *output* terbaik. Dan dari serangkaian tahapan yang sudah dilakukan sebelumnya sampai dengan pengujian klasifikasi abjad isyarat Indonesia menggunakan metode *template matching* dan KNN berikut ini adalah hasil persentase akurasi yang didapatkan untuk masing-masing nilai K:

1. Dari hasil klasifikasi KNN dengan nilai K=1 didapatkan hasil akurasi sebesar 96.52%.
2. Dari hasil klasifikasi KNN dengan nilai K=3 didapatkan hasil akurasi sebesar 95.15%.
3. Dari hasil klasifikasi KNN dengan nilai K=5 didapatkan hasil akurasi sebesar 93.58%.
4. Dari hasil klasifikasi KNN dengan nilai K=7 didapatkan hasil akurasi sebesar 93.01%.
5. Dari hasil klasifikasi KNN dengan nilai K=9 didapatkan hasil akurasi sebesar 92.49%.

5. Penutup

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, penggunaan metode template matching sangat efektif dalam mengklasifikasi dan mendeteksi gerakan abjad isyarat sehingga dapat dikonversi menjadi sebuah teks dengan baik yang dapat membatu orang normal untuk mengenali gerakan abjad isyarat. Hasil yang didapatkan dari uji kecocokan template matching yaitu 11 gambar yang cocok dengan gambar template dari total 17 gambar yang diujikan. Dengan persentase kecocokan gambar sebesar 85.04% untuk abjad isyarat statis dan 84.65% untuk abjad isyarat dinamis. Sedangkan hasil yang didapatkan dari tahap klasifikasi KNN sebesar 96.52%.

5.2 Saran

Berdasarkan kesimpulan diatas, kelemahan yang dimiliki oleh program klasifikasi abjad isyarat Indonesia ini adalah kurang mudahnya dalam melakukan uji gambar, karena pengujian harus dilakukan secara satu persatu sehingga saran yang dapat diberikan guna memperbaiki kekurangan yang ada dalam penelitian ini yaitu penelitian selanjutnya dapat mengembangkan model klasifikasi abjad BISINDO dengan tampilan GUI sehingga memudahkan penggunaan, serta pengguna *hardware* tambahan seperti *kinect* untuk merekam gerakan abjad dinamis.

Referensi

[1] C. R. Reynolds and E. Fletcher-Janzen, *A Reference for the Education of the Handicapped and Other Exceptional Children and Adults*. 2004.

[2] A. Muhammad, “Sistem Isyarat Bahasa Indonesia (SIBI) atau Bahasa Isyarat Indonesia (BISINDO)?,” *Young On Top*, 2019. [Online]. Available: https://www.youngontop.com/read/20433/sistem-isyarat-bahasa-indonesia-sibi-atau-bahasa-isyarat-indonesia-bisindo/. [Accessed: 16-Jul-2020].

[3] MathWorks, “Edge Detection Methods for Finding Object Boundaries in Images,” 2020. [Online]. Available: https://www.mathworks.com/discovery/edge-detection.html#:~:text=Edge detection is an image,computer vision%2C and machine vision. [Accessed: 19-Jun-2020].

[4] Putra and Darma, *Pengolahan Citra Digital*. Yogyakarta: Andi, 2010.

[5] O. Widiarsana, N. . Putra, P. G. . Budiyasa, A. N. . Bismantara, and S. . Mahajaya, “Data Mining: Metode Clasification K-Nearest Neighbor (KNN),” *Bali Progr. Stud. Teknol. Inf. Univ. Udayana*, 2011.