

Pengenalan Citra Tanda Tangan *Off-line* dengan Pemanfaatan Ciri *Centroid Distance Function*

Rizka Husnun Zakiyyah¹, Agus Wahyu Widodo², Fitri Utaminingrum³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹rizkazakiyyah15@gmail.com, ²a_wahyu_w@ub.ac.id, ³f3_ningrum@ub.ac.id

Abstrak

Tanda tangan merupakan salah satu bukti sah kepemilikan dokumen dan transaksi yang menyimpan data terpenting dari setiap individu dan dijadikan sebagai alat verifikasi identitas. Namun, pengecekan keaslian tanda tangan masih dilakukan secara manual dan mengandung banyak risiko dalam proses pengecekannya. Untuk menyelesaikan masalah tersebut, dilakukan pengenalan citra tanda tangan dengan pemanfaatan ciri *centroid distance function*. Proses pengenalan citra tanda tangan diawali dengan melakukan *preprocessing*, yaitu: *binerisasi*, *filtering*, *cropping*, *resizing*, dan *thinning*. Selanjutnya dilakukan penelusuran piksel untuk menyimpan semua posisi piksel yang mengandung *foreground* dan dicari piksel tengah dari citra. Semua piksel yang tersimpan akan dihitung *centroid distance function*-nya dan kemudian akan dikelompokkan sesuai dengan banyak segmen yang membagi citra sehingga setiap kelompok memiliki jumlah data yang sama. Setiap kelompok akan dihitung nilai rata-rata *centroid distance function*. Fitur hasil ekstraksi ciri akan diproses dengan *k-nearest neighbor* sehingga menghasilkan hasil klasifikasi. Pada penelitian yang telah dilakukan, didapatkan akurasi tertinggi untuk data 20 kelas mencapai 88.5% yang diperoleh dari jumlah fitur sebanyak 20 fitur dan $k=1$ dengan data latih sebanyak 10 dan 14 data tiap kelas. Akurasi tertinggi untuk data 50 kelas mencapai 67.4% dengan menggunakan banyak fitur sebanyak 15 fitur dan $k=3$ dengan data latih sebanyak 10 dan 14 data tiap kelas.

Kata kunci: Tanda Tangan, Citra, Preprocessing, Centroid Distance Function, K-Nearest Neighbor

Abstract

A person's signature is one of the most valid proof that shows ownership of documents and transactions that contain their most important data. However, the process of analyzing its authenticity is still done manually. To resolve this problem, an image recognition system for signature will be developed by applying characteristic centroid distance function. This Image recognition process begins with preprocessing, such as binerisasi, filtering, cropping, resizing, and thinning. Next the position of pixels will be searched to store all the foreground pixels and centroid pixels of the image. All pixels stored distance will be calculated using centroid function and grouped according to the amount of features that were selected so that each group has the same amount of data. The average of centroid distance function will be counted on every group so that each group will generate one feature. The results of feature extraction will be processed with the k-nearest neighbor classification method. On the research that has been done the highest accuracy obtained from extraction characteristics of centroid distance function uses 20 class is 88.5% obtained from 20 features and $k=1$ with the amount of 10 and 14 training data for each class. The highest accuracy to 50 class is 67.4% obtained from 15 features and $k=3$ with 10 and 14 training data for each class.

Keywords: Signature, Image, Preprocessing, Centroid Distance Function, K-Nearest Neighbor

1. PENDAHULUAN

Tanda tangan merupakan salah satu tanda bukti sah dari kepemilikan dokumen dan transaksi yang menyimpan data penting dari setiap individu dan dijadikan sebagai alat

verifikasi identitas seseorang. Oleh karena itu diperlukan suatu proses pengenalan tanda tangan yang benar agar tidak terjadi kesalahan dalam verifikasi tanda tangan. Namun saat ini pengecekan keaslian tanda tangan masih dilakukan secara manual yaitu dengan disediakan seseorang sebagai pengecek keaslian

tanda tangan. Hal ini mengandung banyak risiko, seperti pengecek bisa saja melakukan kesalahan dalam proses pengecekan tanda tangan. Selain itu, pengenalan tanda tangan dengan jumlah yang banyak dalam waktu bersamaan memiliki peluang cukup besar untuk terjadi kesalahan. Oleh karena itu, diperlukan suatu sistem yang dapat mempermudah dan mengatasi masalah kesalahan pengecekan tanda tangan.

Pada pemrosesan citra berbasis ciri, citra dapat dikatakan serupa apabila memiliki ciri karakter yang sama antara satu tanda tangan dengan tanda tangan yang lain. Ciri yang dimaksud adalah ciri arah dan ciri bentuk, sehingga setiap tanda tangan terdiri dari beragam lengkung (*curve*) dan garis yang memiliki orientasi atau arah (Cheriet, et al., 2007). Arah goresan pada tanda tangan berperan penting dalam membedakan tanda tangan atau karakter orang tertentu dengan orang lain. Oleh karena itu dalam penelitian ini, ciri arah dan bentuk menjadi ciri yang akan diekstraksi dari citra tanda tangan dan dipakai dalam klasifikasi tanda tangan.

Ciri *Centroid Distance Function* (CDF) adalah salah satu *descriptor* ciri untuk representasi bentuk yang digunakan pada pengolahan citra. Penelitian ekstraksi ciri *Centroid Distance Function* pernah digunakan untuk mengenali objek citra dengan *contour* (Wang, et al., 2009) dan menghasilkan hasil optimal ketika banyak titik tepi yang digunakan lebih dari 90. Penelitian lain menggunakan *Centroid Distance Function* juga pernah dilakukan oleh Bhonsle & Klinzmann untuk mengelompokkan objek citra sel kanker (2011). Hasil penelitian ini menunjukkan bahwa dengan nilai $k=2$ dan $k=3$ untuk *K-means clustering*, objek dengan bentuk yang mirip dapat dikelompokkan menjadi kelompok yang sama. Kedua penelitian tersebut menunjukkan bahwa CDF berhasil melakukan ekstraksi pada objek citra, sehingga *Centroid Distance Function* akan dicoba untuk melakukan ekstraksi pada citra tanda tangan.

Penelitian tentang tanda tangan pernah dilakukan oleh Jariah, et al pada objek citra tanda tangan dengan menggunakan *Invariant Moment Radial Basis Function* (2011). Penelitian ini menghasilkan 7 fitur dengan akurasi mencapai 90% dan terjadi eror sebesar 20%. Penelitian dengan menggunakan objek yang sama juga pernah dilakukan oleh Widodo & Harjoko. Penelitian ini menggunakan ekstraksi ciri *Histogram of oriented gradient* (HoG) dan

Histogram of Curvatur (HoC) untuk melakukan verifikasi tanda tangan (2015). Hasil penelitian ini menunjukkan nilai persentase *False Rejection Rate* (FRR) sebesar 4% dan *False Acceptance Rate* (FAR) mencapai 57%. Yadav & Tyagi pernah melakukan penelitian tentang tanda tangan dengan menggunakan *Invariant Moment* dan *Zernicke Moment* (2015). Hasil penelitian tersebut menunjukkan nilai FAR terkecil sebesar 1,8% dengan menggunakan *Generic Algorit* untuk klasifikasi.

Metode lain juga perlu digunakan untuk melakukan proses klasifikasi dari data fitur yang dihasilkan dari proses ekstraksi fitur *centroid distance function*, yaitu *K-Nearest Neighbor*. Penelitian tentang *k-nearest neighbor* yang dilakukan oleh Hassanat, et al. menunjukkan nilai akurasi mencapai 85% untuk klasifikasi 28 *dataset* (2014). Dalam penelitian ini, terdapat sepuluh tipe klasifikasi untuk membandingkan hasil performa dan menunjukkan bahwa semakin besar nilai k , maka nilai akurasi tidak semakin besar. Nilai akurasi yang paling besar ditunjukkan dengan nilai $k=5$ dan $k=7$. Penelitian mengenai *k-nearest neighbor* juga pernah dilakukan dengan menggunakan data sebanyak 200 rekam data (Alkhatib, et al., 2013). Penelitian tersebut menggunakan nilai $k=5$ dan menunjukkan hasil yang stabil serta *rasio eror* yang kecil. Hal ini menunjukkan bahwa metode *K-Nearest Neighbor* merupakan metode yang cukup handal untuk digunakan sebagai metode klasifikasi.

2. STUDI PUSTAKA

2.1 Citra Tanda Tangan

Tanda tangan yang dimiliki oleh setiap orang mempunyai keunikan, meski tidak jarang tanda tangan tersebut memiliki kemiripan antara satu orang dengan orang yang lain. Hal ini dikarenakan setiap tanda tangan memiliki bentuk, penekanan, dan lekukan, serta arah yang berbeda-beda dalam setiap tanda tangan. Tanda tangan yang dimiliki oleh seseorang dapat memiliki informasi yang mewakili data yang terdapat pada individu, seperti nama, alamat, dan tanggal lahir sehingga banyak dokumen-dokumen menggunakan tanda tangan sebagai pembuktian. Pihak yang menandatangani, mengetahui, dan menyetujui seluruh isi dokumen dinyatakan dengan keberadaan tanda tangan dalam sebuah dokumen (Ardiansyah, 2015).

Dalam penelitian ini, citra tanda tangan yang digunakan sebanyak 20 jenis tanda tangan dari 20 orang yang berbeda dengan melakukan 24 kali tanda tangan. Dengan adanya variasi untuk setiap tanda tangan, hasil klasifikasi akan lebih akurat karena segala kondisi data uji yang diberikan bisa diproses dan diklasifikasikan.

2.2 Preprocessing Citra

Preprocessing citra adalah sebuah proses awal dalam pengolahan citra yang harus dilakukan sebelum melakukan proses ekstraksi fitur. Pada *preprocessing* citra ini, dilakukan beberapa proses perbaikan citra antara lain, *binarization* (mengubah citra menjadi citra biner), *denoising* (menghilangkan *noise* yang ada di citra), dan normalisasi.

2.2.1. Binerisasi

Binerisasi adalah proses mengubah citra input (citra *RGB* atau citra *Grayscale*) menjadi citra biner atau citra yang hanya terdiri dari warna hitam dan putih (0 dan 1). Proses merubah citra *grayscale* menjadi citra biner adalah dengan membandingkan nilai *threshold* yang telah ditentukan dengan nilai di tiap piksel pada citra. *Threshold* ini berfungsi sebagai nilai pembatas yang akan membedakan nilai pada piksel menjadi bernilai hitam (*foreground*) atau bernilai putih (*background*). Jika nilai di suatu piksel lebih kecil dibandingkan dengan nilai *threshold*, maka nilai piksel nya menjadi hitam dan jika nilai suatu piksel lebih besar dibandingkan dengan nilai *threshold*, maka nilai piksel akan menjadi putih. Hal ini dapat dituliskan dalam bentuk matematika:

$$\begin{aligned} t(x,y) \geq \text{threshold} & \text{ maka } t(x,y) = 0\text{xFFFF} & (1) \\ t(x,y) < \text{threshold} & \text{ maka } t(x,y) = 0\text{xFF000000} & (2) \end{aligned}$$

Penentuan nilai *threshold* dilakukan dengan menggunakan metode *otsu thresholding*. Metode *otsu thresholding* adalah menghitung nilai ambang pada citra input secara otomatis sehingga dapat memisahkan objek dengan latar belakang (Putra, 2010). Untuk mendapatkan nilai *threshold*, dilakukan perhitungan *variance* dengan menjumlahkan hasil perkalian bobot dengan varian pada setiap *background* dan *foreground* yang ditunjukkan pada Persamaan 3.

$$\sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2 \quad (3)$$

σ_w^2 adalah *variance* atau hasil *threshold*, W_b adalah bobot untuk *background*, σ_b^2 adalah varian

untuk *background*, W_f adalah bobot untuk *foreground*, dan σ_f^2 adalah varian untuk *foreground*. Perhitungan bobot dan varian untuk *background* dan *foreground* ditunjukkan pada Persamaan 4, 5, 6, dan 7.

$$W_b = \frac{\sum_{i=1}^b n_{bi}}{N} \quad (4)$$

$$\sigma_b^2 = \frac{\sum_{i=1}^b ((i - \mu_b)^2 \cdot n_{bi})}{\sum_{i=1}^b n_{bi}} \quad (5)$$

$$W_f = \frac{\sum_{i=1}^f n_{fi}}{N} \quad (6)$$

$$\sigma_f^2 = \frac{\sum_{i=1}^f ((i - \mu_f)^2 \cdot n_{fi})}{\sum_{i=1}^f n_{fi}} \quad (7)$$

W_b adalah bobot untuk *background*, W_f adalah bobot untuk *foreground*, σ_b^2 adalah varian untuk *background*, σ_f^2 adalah varian untuk *foreground*, n_{bi} adalah frekuensi ke i pada *background*, n_{fi} adalah frekuensi ke i pada *foreground*, N adalah total keseluruhan frekuensi yang ada pada citra. Nilai μ_b dan μ_f didapatkan dari Persamaan 8 dan 9.

$$\mu_b = \frac{\sum_{i=1}^b i \cdot n_{bi}}{\sum_{i=1}^b n_{bi}} \quad (8)$$

$$\mu_f = \frac{\sum_{i=1}^f i \cdot n_{fi}}{\sum_{i=1}^f n_{fi}} \quad (9)$$

μ_b adalah rata-rata *background* dan μ_f adalah rata-rata *foreground*.

2.2.2. Denoising

Proses *denoising* adalah proses untuk menghaluskan dan menghilangkan *noise* yang ada pada citra sehingga kualitas citra bisa lebih baik dan bisa lebih jelas. Metode *denoising* yang digunakan adalah *median filter* yang merupakan *filter non-linier* yang dihitung dengan mengurutkan nilai intensitas sekelompok piksel, kemudian menggantikan nilai piksel yang diproses menjadi nilai tertentu (Syarifuddin, 2015). *Median filter* mengganti nilai piksel target dengan nilai tengah atau median piksel tetangga. Pertama, nilai piksel tetangga akan diurutkan dari nilai terkecil ke nilai terbesar dan kemudian nilai piksel target akan diganti dengan nilai tengah dari piksel yang telah diurutkan (Boateng, et al., 2012). Hal ini dapat dituliskan dalam bentuk matematika:

$$g(x,y) = \text{Median}\{f(s,t)\}, (s,t) \in S_{xy} \quad (10)$$

$g(x,y)$ merupakan citra yang dihasilkan dari *denoising* dan S_{xy} himpunan koordinat dalam *window*. *Window* adalah piksel tetangga yang

mengelilingi sebuah piksel tertentu dan berpusat pada piksel target.

2.2.3. Normalisasi

Normalisasi adalah proses penyamaan objek citra tanda tangan dari segi ukuran dan dimensi sehingga memiliki luasan citra yang sama. Pada tahap normalisasi dilakukan beberapa proses yang terdiri dari *cropping*, *resize* dan *thinning*.

Proses *cropping* dilakukan dengan terlebih dahulu mencari piksel terujung yang mengandung warna hitam yang berada di ujung atas, ujung bawah, ujung kanan, dan ujung kiri dari sebuah citra. Piksel-piksel inilah yang nantinya akan dijadikan batas-batas pemotongan citra. Langkah berikutnya adalah proses *resize* untuk menyamakan ukuran citra hasil *cropping* yang cenderung menghasilkan citra dengan ukuran yang berbeda antara citra satu dengan citra lain. Pada proses ini, ukuran semua citra akan diubah menjadi 100x100 piksel, sehingga semua citra memiliki ukuran dan dimensi yang sama.

Thinning adalah proses pengurangan ketebalan sebuah objek pada citra digital sehingga objek tersebut menjadi ukuran yang minimum sehingga mencapai 1 piksel saja. Proses *thinning* diperlukan untuk mempermudah proses penelusuran piksel yang mengandung warna hitam (*foreground*) yang akan disimpan.

2.3 Centroid Distance Function

Centroid Distance Function adalah salah satu cara untuk melakukan ekstraksi fitur bentuk dari sebuah citra digital yang berfungsi untuk mengekspresikan jarak titik batas luar dari titik pusat atau *centroid*. Metode ini menghitung nilai jarak dari titik-titik yang telah disegmentasi pada garis batas luar sebuah citra dengan Persamaan,

$$r(t) = \sqrt{[(x(t) - g_x)^2 + (y(t) - g_y)^2]} \quad (11)$$

$r(t)$ adalah *centroid distance* yang juga disebut shape signature dan $t = 0, 1, 2, 3, \dots, N-1$. g_x dan g_y adalah *centroid* koordinat x dan y pada *centroid* citra tanda tangan. $x(t)$ dan $y(t)$ adalah koordinat x dan y pada titik segmentasi ke- t .

$$g_x = \frac{\text{Width}}{2} \quad (12)$$

$$g_y = \frac{\text{Height}}{2} \quad (13)$$

Cara untuk menentukan titik segmentasi pada suatu citra adalah dengan menelusuri semua piksel pada citra dan menyimpan piksel-

piksel yang mengandung warna hitam. Jumlah semua piksel yang disimpan akan dikelompokkan menjadi beberapa kelompok sesuai dengan banyaknya fitur yang ingin dihasilkan sehingga setiap kelompok memiliki jumlah piksel yang sama. Selanjutnya, semua piksel akan dihitung jaraknya dengan *centroid* (titik tengah) dari citra dan dicari nilai rata-rata hasil jarak dari setiap kelompok.

2.4 K-Nearest Neighbor

K-Nearest Neighbor (K-NN) adalah salah satu metode pengenalan pola yang sering digunakan untuk proses klasifikasi data. Klasifikasi *K-NN* mengkategorikan sebuah sampel data tidak berlabel dengan menggunakan label mayoritas dari sampel data tetangga terdekat dalam data training (Hassanat, et al., 2014).

Langkah pertama yang dilakukan adalah menentukan nilai k atau menentukan banyaknya tetangga. Hal ini berpengaruh pada hasil klasifikasi dan keakuratan klasifikasi. Selanjutnya adalah proses mengukur jarak kedekatan. Terdapat beberapa cara yang bisa dilakukan untuk mengukur jarak kedekatan antara data baru (data uji) dan data lama (data latih), yaitu dengan menggunakan rumus *Euclidean distance*, Persamaannya sebagai berikut:

$$\text{Eucli} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \quad (14)$$

a_1 adalah nilai data a pada fitur 1, b_1 adalah nilai data b pada fitur 2, dan seterusnya hingga data a pada fitur ke- n dan data b pada fitur ke- n . Semakin besar nilai *Euclidean distance*, maka dua data tersebut semakin tidak memiliki kemiripan atau kedekatan. Namun sebaliknya semakin kecil nilai *Euclidean distance*, maka semakin besar jarak kedekatan dua data tersebut.

Ketika proses perhitungan *Euclidean distance* selesai dilakukan, maka nilai-nilai *Euclidean distance* diurutkan dari nilai terkecil ke nilai terbesar. Nilai *Euclidean distance* yang kecil menandakan bahwa dua data tersebut memiliki kedekatan yang besar. Setelah diurutkan, data akan diambil sebanyak k data dari urutan nilai *Euclidean* terkecil untuk dilakukan proses penentuan kelas data uji berdasarkan banyaknya kelas dari data latih.

3. HASIL DAN PEMBAHASAN

3.1 Hasil program

Dalam penelitian ini, citra tanda tangan yang digunakan sebanyak 20 jenis tanda tangan dari 20 orang yang berbeda dengan melakukan 24 kali tanda tangan. Penelitian ini juga menggunakan tanda tangan sebanyak 50 kelas yang digunakan sebagai perbandingan dengan data 20 kelas.

Citra data *training* yang telah dilakukan proses preprocessing hingga menghasilkan citra dengan ukuran 100x100, akan diproses untuk menghasilkan ekstraksi fitur dengan menggunakan ekstraksi ciri *Centroid Distance Function*. Fitur yang dihasilkan terdiri dari nilai rata-rata *Centroid distance function*.

Pengenalan Citra Tanda Tangan dengan Pemanfaatan Ciri Centroid Distance Function

Training Testing

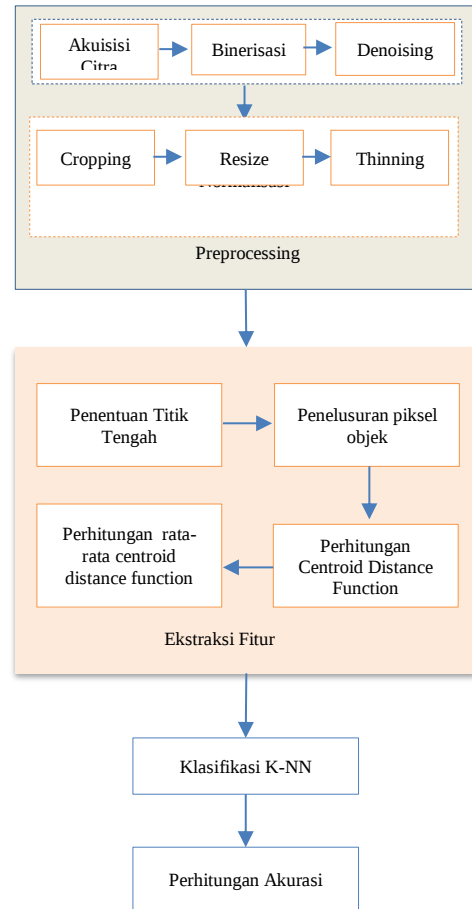
Banyak Fitur: 100

Mulai Training

No	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6	Fitur 7	Fitur 8	Fitur 9	Fitur 10	Kelas
1	45.43	113.004	20.968	82.164	12.805	82.449	20.787	53.871	39.769	82.139	002
2	50.028	119.737	23.882	80.597	16.489	80.494	16.34	33.271	29.686	104.541	002
3	50.063	83.583	30.134	37.876	18.87	132.413	15.945	46.962	31.666	116.332	002
4	46.948	103.008	29.606	87.509	17.622	46.396	17.679	66.729	26.034	109.895	002
5	52.256	113.665	29.418	64.275	17.006	107.413	12.704	52.218	27.764	196.763	002
6	45.143	83.256	28.684	149.975	18.672	147.802	19.745	96.547	36.076	210.277	003
7	43.773	105.362	19.773	100.406	18.466	139.524	21.91	89.341	37.881	187.719	003
8	45.236	86.995	25.029	212.731	19.079	185.617	21.068	95.485	35.815	219.53	003
9	45.616	97.283	24.694	166.827	19.362	155.352	22.511	114.447	40.853	103.601	003
10	42.533	97.073	20.975	166.138	19.467	132.461	22.677	88.899	40.834	141.62	003
11	40.189	82.536	22.335	82.184	12.829	92.639	30.049	126.341	47.073	103.184	005
12	42.881	47.041	28.629	73.995	15.412	97.784	24.751	134.263	44.086	147.734	005
13	38.434	69.379	20.708	67.281	14.432	106.403	27.255	195.422	47.149	120.267	005
14	44.016	59.813	28.827	87.511	16.039	61.836	25.971	136.883	45.058	198.196	005
15	43.117	58.369	27.695	100.369	16.305	113.249	23.725	119.108	46.474	106.91	005
16	50.153	172.994	28.808	167.139	13.284	45.948	20.668	39.872	35.665	44.58	007
17	52.224	102.96	21.331	53.328	14.415	32.427	25.563	68.977	35.649	83.635	007
18	51.014	124.056	22.067	77.58	12.275	24.108	25.44	48.127	37.651	68.646	007
19	48.857	115.882	19.402	81.353	12.566	45.884	25.911	62.009	35.958	70.608	007

Gambar 1. Hasil Training

Pada Gambar 1, setiap citra akan menghasilkan nilai fitur sesuai banyak fitur yang dipilih. Setelah setiap citra dilakukan *preprocessing*, maka setiap citra akan dicari *centroid*-nya dan dilakukan penelusuran piksel. Setiap piksel akan disimpan dan dihitung *centroid distance function*-nya. Nilai hasil *centroid distance function* tersebut akan dikelompokkan berdasarkan banyak fitur yang dipilih dan kemudian akan dihitung rata-rata dari setiap kelompok. Hasil rata-rata inilah yang dijadikan sebagai nilai fitur hasil ekstraksi ciri.



Gambar 6. Proses Algoritma

Pengenalan Citra Tanda Tangan dengan Pemanfaatan Ciri Centroid Distance Function

Training Testing

Citra Input: 1

Mulai Klasifikasi

Hasil Klasifikasi

No	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6	Fitur 7	Fitur 8	Fitur 9	Fitur 10	Kelas
1	47	103	26	62	16	62	20	106	38	149	253
2	43	129	22	139	11	74	21	56	45	104	002
3	44.54	140	23	60	18	85	24	107	37	120	253
4	42	106	24	68	13	41	20	75.81	40	118	002
5	44	93	22	53	14	61	18	50	44	139	002
6	45	142	21	53	13	47	23	89.6	41	116	002
7	42	106	24	69	13	41	20	75.81	40	118	002
8	43	94	19	74	14	84	22	57.64	42	131	002
9	43	145	19	72	16	42.69	25	86	43	98	002
10	42	136	22	84	13	64	22	35	44	91	002
11	45	134	25	106	23	171	21	125	38.24	151	115
12	40.15	149	21	108	17	136	19	87.19	34	192	003
13	41	133	22	170	18	160	23	113	38.24	189.1	003
14	40.34	93	21	234	20	174	24	102	40	181	115
15	40.15	149	21	168	17	139	19	87.19	34	192	003
16	45	94	25	184	19	206	22	105	39	198	003
17	41	180	24	177	19	185	23	96	39	195	003
18	44.68	74	25	156	18	178	21	110	40	218	003
19	48	77	28	160	19	194	22	151	35	203	003
20	43	187	23	157	20	144	21	77	44	119	043
21	42	60	27	107	16	125	25	133	45	106	005
22	44	46	28	96	16	114	26	165	49.5	106	005
23	45	66	26	95	20.57	162	36	114	51	64	236
24	42	57.79	28	92	16	117	26	128	49	107	005
25	44	58	27	69	16	106	30	128	48	90	005
26	43	50	27	82	18	122	31.87	141	48	68	005
27	45	56	27	88	15	79	28	145	45	127	005

Akurasi: 76.5

Gambar 2. Hasil ekstraksi fitur *Centroid Distance Function*

Pada Gambar 2, ditunjukkan hasil ekstraksi ciri *centroid distance function* untuk data *testing*. Setelah semua citra menghasilkan nilai fitur, fitur tersebut akan digunakan untuk proses klasifikasi menggunakan metode *K-Nearest Neighbor*. Nilai *k* didapatkan dari masukan *user*.

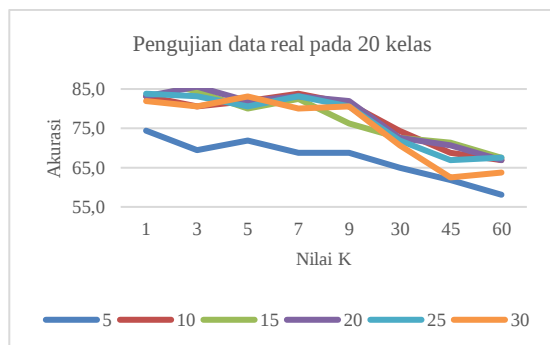
3.2 Pengujian

3.2.1 Pengujian data 20 kelas

Pengujian ini menggunakan data sebanyak 480 citra tanda tangan dengan 320 citra tanda tangan sebagai data latih dan 160 data tanda tangan sebagai data uji. Pengujian *fitting* dilakukan dengan menguji kembali data latih sebagai data uji. Fitur yang digunakan dalam pengujian ini hanya menggunakan rata-rata *centroid distance function*. Hasil pengujian *fitting* ditunjukkan pada Tabel 1.

Tabel 1. Pengujian Data Real

fitur	nilai K							
	1	3	5	7	9	30	45	60
5	74.4	69.4	71.9	68.8	68.8	65.0	61.9	58.1
10	83.1	80.6	81.9	83.8	81.3	74.4	68.8	66.9
15	87.5	84.4	80.0	82.5	76.3	72.5	71.3	67.5
20	83.1	85.6	81.9	83.1	81.9	72.5	70.6	66.9
25	83.8	83.1	80.6	83.1	80.6	71.9	66.9	67.5
30	81.9	80.6	83.1	80.0	80.6	70.6	62.5	63.7



Grafik 1. Pengujian Data Real pada 20 Kelas

Pada Grafik 1 terlihat bahwa akurasi untuk pengujian data *real* mencapai angka tertinggi pada jumlah fitur 15 dengan nilai $k=1$ mencapai 87.5%. Hasil pengujian data *real* menunjukkan nilai yang fluktuatif namun cenderung menurun ketika nilai k bertambah.

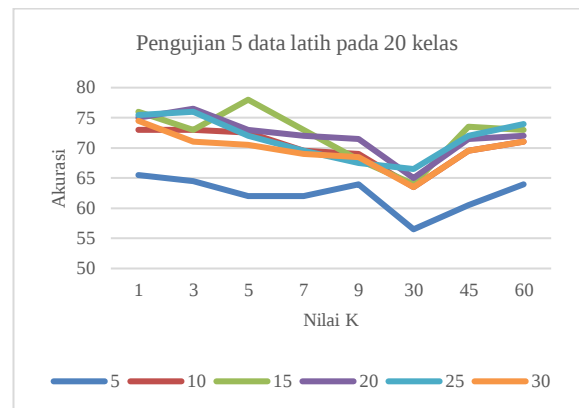
Pengujian data *random* 10 data uji dengan 5 data latih untuk dataset 20 kelas ditunjukkan pada tabel 2.

Tabel 2. Pengujian 5 Data Latih pada 20 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	65.5	64.5	62	62	64	56.5	60.5	64
10	73	73	72.5	69.5	69	63.5	69.5	71
15	76	73	78	73	68	64	73.5	73
20	75	76.5	73	72	71.5	65	71.5	72
25	75.5	76	72	69.5	67.5	66.5	72	74
30	74.5	71	70.5	69	68.5	63.5	69.5	71

Pada Tabel 2, terlihat bahwa nilai akurasi terbesar 78% yang diperoleh dari jumlah fitur 15 dan nilai $k=5$, sedangkan untuk akurasi terkecil 56.5% yang diperoleh dari jumlah fitur sebesar 5 dengan nilai $k=30$.

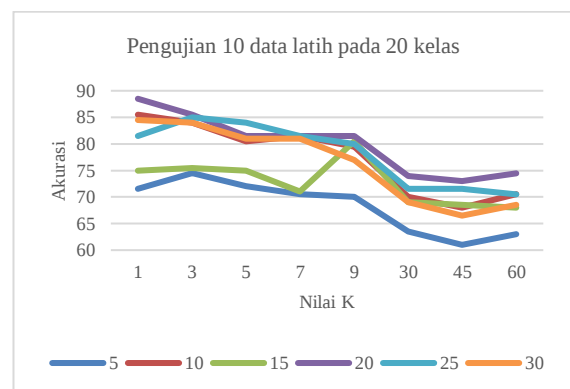
Pengujian data *random* 10 data uji dengan 10 data latih untuk dataset 20 kelas ditunjukkan pada Tabel 3.



Grafik 2. Pengujian 5 Data Latih pada 20 Kelas

Tabel 3. Pengujian 10 Data Latih pada 20 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	71.5	74.5	72	70.5	70	63.5	61	63
10	85.5	84	80.5	81.5	79.5	70	68	70.5
15	75	75.5	75	71	80.5	69	68.5	68
20	88.5	85.5	81.5	81.5	81.5	74	73	74.5
25	81.5	85	84	81.5	80	71.5	71.5	70.5
30	84.5	84	81	81	77	69	66.5	68.5



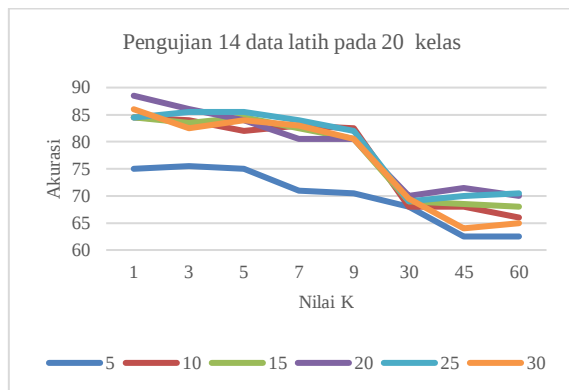
Grafik 3. Pengujian 10 Data Latih pada 20 Kelas

Pada tabel 3, terlihat bahwa akurasi tertinggi sebesar 88.5% didapatkan dari jumlah fitur sebanyak 20 dengan nilai $k=1$, sedangkan akurasi terendah sebesar 61% diperoleh dari jumlah fitur sebanyak 5 dengan nilai $k=45$.

Pengujian data *random* 10 data uji dengan 14 data latih untuk dataset 20 kelas ditunjukkan pada Tabel 4.

Tabel 4. Pengujian 14 Data Latih pada 20 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	75	75.5	75	71	70.5	68	62.5	62.5
10	84.5	84	82	83	82.5	68	68	66
15	84.5	83.5	84.5	82.5	80.5	69	68.5	68
20	88.5	86	84	80.5	80.5	70	71.5	70
25	84.5	85.5	85.5	84	82	69	70	70.5
30	86	82.5	84	83	80.5	69.5	64	65



Grafik 4. Pengujian 14 Data Latih pada 20 Kelas

Pada Tabel 4, terlihat bahwa akurasi tertinggi sebesar 88.5% yang diperoleh dari jumlah fitur sebanyak 20 dan nilai $k=1$, sedangkan akurasi terkecil 62.5% diperoleh dari jumlah fitur sebanyak 5 dengan nilai $k=60$. Pada Grafik 8 secara umum terlihat bahwa nilai akurasi mengalami penurunan ketika k bertambah.

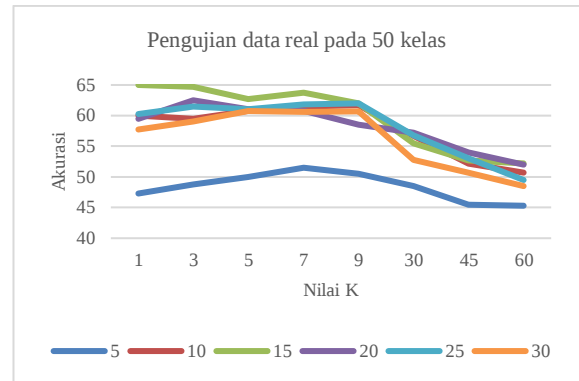
3.2.2 Pengujian data 50 kelas

Pengujian data 50 kelas dilakukan dengan menggunakan 1200 data yang terdiri dari 24 tanda tangan untuk setiap kelas. Dari 1200 data tanda tangan, 800 data digunakan sebagai data latih dan 400 data sebagai data uji. Fitur yang digunakan dalam pengujian ini merupakan hasil rata-rata *centroid distance function*.

Pengujian data *real* menggunakan 50 kelas ditunjukkan pada Tabel 5.

Tabel 5. Pengujian Data *Real*

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	47.3	48.8	50	51.5	50.5	48.5	45.5	45.3
10	60	59.5	60.75	61.5	61.7	56.7	52.2	50.7
15	65	64.7	62.7	63.7	62	55.5	52.7	52.2
20	59.5	62.5	61	60.8	58.5	57.2	54	52
25	60.25	61.5	61	61.8	62	56.8	53	49.5
30	57.75	59	60.75	60.6	60.7	52.8	50.7	48.5

Grafik 5. Pengujian Data *Real* pada 50 Kelas

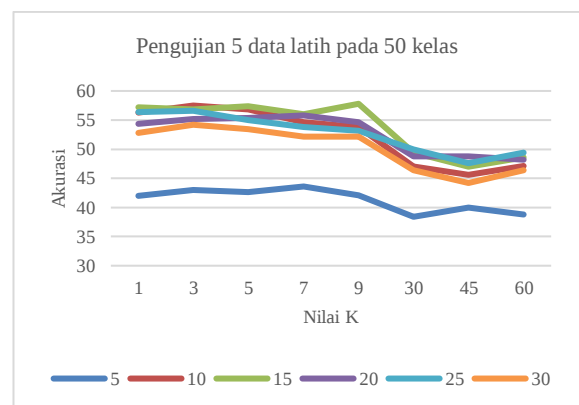
Berdasarkan Tabel 5, terlihat bahwa akurasi untuk pengujian data *real* mencapai angka tertinggi pada jumlah fitur 15 dengan nilai $k=1$ sebesar 65%. Sedangkan untuk akurasi terkecil 45.3% ada pada jumlah fitur sebesar 5 dengan nilai $k=60$.

Pengujian data *random* dilakukan dengan memilih 10 data dari 24 data tanda tangan di tiap kelas secara acak dan data lain yang tidak terpilih akan dibagi menjadi 5 data, 10 data, dan 14 data untuk setiap kelas sebagai data latih.

Pengujian data uji asli untuk data *random* dilakukan dengan menggunakan 10 data yang dipilih secara acak pada setiap kelas. Pengujian data asli dengan menggunakan 5 data latih untuk setiap kelas ditunjukkan pada Tabel 6.

Tabel 6. Pengujian 5 Data Latih pada 50 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	42	43	42.6	43.6	42.1	38.4	40	38.8
10	56.3	57.5	56.8	54.6	53.8	47	45.6	47.1
15	57.2	56.8	57.4	56	57.8	49.4	47	48.6
20	54.4	55.2	55.4	55.8	54.6	48.8	48.8	48.2
25	56.4	56.6	55	53.8	53.2	50	47.6	49.4
30	52.8	54.2	53.4	52.2	52.2	46.4	44.2	46.4



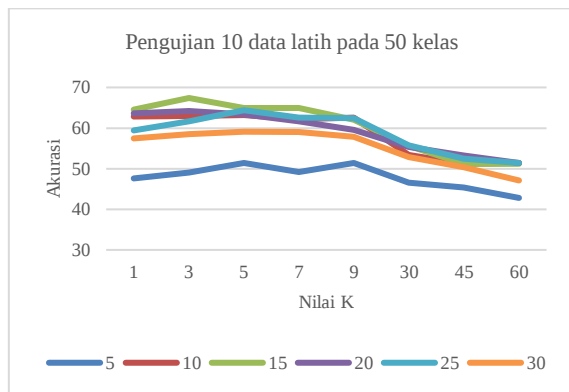
Grafik 6. Pengujian 5 Data Latih pada 50 Kelas

Pada Tabel 6, nilai akurasi tertinggi sebesar 57.8% diperoleh dari jumlah fitur sebanyak 15 dengan $k=1$, sedangkan untuk akurasi terendah sebesar 38.44% diperoleh dari jumlah fitur 5 dengan $k=30$.

Pengujian data asli dengan menggunakan 10 data latih untuk tiap kelas ditunjukkan pada Tabel 7.

Tabel 7. Pengujian 10 Data Latih pada 50 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	47.6	49	51.4	49.2	51.4	46.6	45.4	42.8
10	62.8	63	63.2	62.2	62.6	53.4	51.2	51.4
15	64.6	67.4	65	65	62	55.6	51.2	51.2
20	63.6	64.2	63.4	61.6	59.6	55.4	53.2	51.4
25	59.4	61.6	64.4	62.6	62.4	55.8	52.4	51.4
30	57.4	58.5	59.1	59	57.9	52.8	50.4	47.1



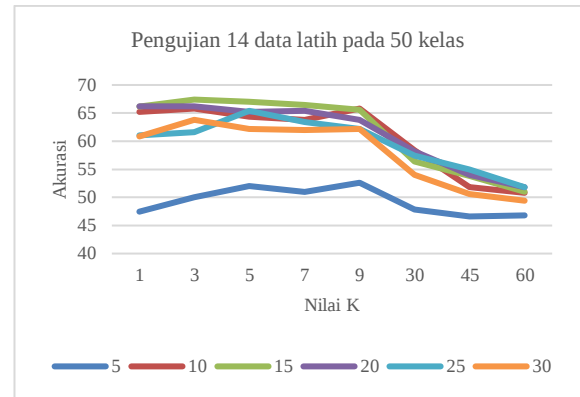
Grafik 7. Pengujian 10 Data Latih pada 50 Kelas

Pada Tabel 7, nilai akurasi tertinggi sebesar 67.6% diperoleh dari jumlah fitur sebanyak 15 dengan $k=3$, sedangkan untuk nilai akurasi terkecil sebesar 42.8% yang diperoleh dari jumlah fitur sebanyak 5 dengan $k=60$.

Pengujian data asli dengan menggunakan 14 data latih untuk tiap kelas ditunjukkan pada Tabel 8.

Tabel 8. Pengujian 14 Data Latih pada 50 Kelas

Fitur	Nilai K							
	1	3	5	7	9	30	45	60
5	47.5	50	52	51	52.6	47.8	46.6	46.8
10	65.2	65.8	64.4	63.8	65.8	58.4	51.8	50.8
15	66.2	67.4	67	66.4	65.6	56.4	53.8	51
20	66.2	66.2	65.2	65.4	63.8	58.2	54	51.8
25	61	61.6	65.4	63.4	62.2	57.4	55	51.8
30	60.8	63.8	62.2	62	62.2	54	50.6	49.4



Grafik 8. Pengujian 14 Data Latih pada 50 Kelas

Pada Tabel 8, nilai akurasi tertinggi sebesar 67.4% yang diperoleh dari jumlah fitur sebanyak 20 dengan $k=3$, sedangkan untuk nilai akurasi terendah sebesar 46.8% diperoleh dari jumlah fitur sebanyak 15 dengan $k=5$.

3.3 Analisis

Berdasarkan pengujian yang telah dilakukan, terlihat bahwa jumlah kelas yang digunakan dalam pengujian berpengaruh terhadap akurasi yang dihasilkan. Jumlah Kelas yang lebih banyak cenderung menghasilkan nilai akurasi yang lebih rendah dibandingkan dengan jumlah kelas yang lebih sedikit. Hal ini ditunjukkan pada Tabel 1, Tabel 2, Tabel 3, Tabel 4, Tabel 5, Tabel 6, Tabel 7, dan Tabel 8. Nilai akurasi tertinggi untuk data 20 kelas mencapai 88.5% yang ditunjukkan pada Tabel 3 dan Tabel 4, sedangkan akurasi tertinggi untuk data 50 kelas hanya mencapai 67.4% pada Tabel 7 dan Tabel 8. Kondisi tersebut menunjukkan bahwa ekstraksi *centroid distance function* untuk data 20 kelas menghasilkan akurasi yang lebih besar dibandingkan dengan data 50 kelas. Hal ini bisa terjadi dikarenakan tanda tangan yang digunakan pada pengujian 50 kelas memiliki kemiripan yang cukup tinggi dengan tanda tangan pada 20 kelas.

Berdasarkan pengujian yang telah dilakukan (20 kelas dan 50 kelas), menunjukkan bahwa 20 fitur merupakan fitur yang optimal. Hal ini ditunjukkan pada Tabel 3 dan Tabel 4 dimana nilai akurasi tertinggi (88.5%) diperoleh dari jumlah fitur sebanyak 20 fitur dengan $k=1$. Sehingga dapat disimpulkan bahwa fitur yang optimal adalah sebanyak 20.

Berdasarkan pengujian dengan variasi jumlah data latih, terlihat bahwa data latih dengan jumlah 10 dan 14 pada tiap kelas menghasilkan nilai akurasi yang lebih tinggi

dibandingkan dengan jumlah data latih yang lain. Hal ini ditunjukkan pada Tabel 3 dan Tabel 4 dimana kedua tabel tersebut merupakan hasil pengujian dengan menggunakan data latih masing-masing 10 data latih dan 14 data latih yang diuji dengan data 20 kelas. Kedua tabel tersebut menghasilkan akurasi tertinggi dengan nilai yang sama yaitu 88.5% yang diperoleh dengan menggunakan jumlah fitur (tanpa varian) sebanyak 20 fitur dan $k=1$. Jika dibandingkan dengan hasil pengujian jumlah data latih yang lain, akurasi tertinggi yang bisa didapatkan untuk 5 data latih hanya 78% dan untuk 16 data latih sebanyak 87.5%. Hal ini menunjukkan bahwa 5 data latih dan 16 data latih menghasilkan nilai akurasi yang lebih rendah dibandingkan dengan 10 data latih dan 14 data latih. Sehingga dapat disimpulkan bahwa 10 data latih dan 14 data latih merupakan jumlah data latih yang optimal.

Berdasarkan pengujian yang telah dilakukan, menunjukkan bahwa terjadi kecenderungan penurunan nilai akurasi ketika jumlah ketetangaan pada *K-Nearest Neighbor* bertambah banyak. Penurunan ini terjadi secara signifikan ketika $k=30$ dan $k=45$ untuk semua jumlah fitur. Kondisi tersebut ditunjukkan pada Tabel 1, Tabel 3, Tabel 4, Tabel 5, Tabel 7, dan Tabel 8. 6 Tabel tersebut merupakan hasil untuk pengujian data real, pengujian 10 data latih, dan pengujian 14 data latih menggunakan data 20 kelas dan data 50 kelas. Namun, Kondisi yang berbeda terjadi pada Tabel 2, dan Tabel 6, dimana penurunan akurasi berada pada titik terendah ketika $k=30$ dan mengalami peningkatan pada $k>30$. 4 tabel tersebut merupakan hasil untuk pengujian 5 data latih pada data 20 kelas dan 50 kelas. Berdasarkan pengujian, nilai akurasi tertinggi sebesar 88.5% diperoleh dari jumlah fitur sebanyak 20 dan $k=1$ yang ditunjukkan pada Tabel 3 dan 4. Hal ini menunjukkan bahwa $k=1$ merupakan banyak ketetangaan yang optimal.

4. KESIMPULAN

Untuk mengimplementasikan pengenalan citra tanda tangan *off-line* dengan pemanfaatan ciri *centroid distance function* yang harus dilakukan adalah melakukan *preprocessing* citra dengan mengubah citra menjadi citra *biner*, menghilangkan *noise* dengan *median filter*, *cropping*, *resize*, dan *thinning*. Selanjutnya mencari posisi *centroid* citra dan melakukan penelusuran piksel yang mengandung warna

hitam serta menyimpan posisi dari piksel tersebut. Piksel yang telah disimpan dihitung jaraknya terhadap *centroid* citra (*centroid distance function*). Nilai *centroid distance function* akan dikelompokkan menjadi beberapa kelompok sesuai dengan jumlah segmen sehingga setiap kelompok memiliki jumlah data yang sama. Di setiap kelompok dihitung rata-rata *centroid distance* sehingga satu kelompok menghasilkan sebuah nilai yang dijadikan sebagai fitur hasil ekstraksi. Citra yang sudah menghasilkan fitur akan diklasifikasikan menggunakan metode *k-nearest neighbor*.

Berdasarkan hasil pengujian, akurasi tertinggi diperoleh dengan menggunakan jumlah fitur sebanyak 20 fitur yang terdiri dari rata-rata *centroid distance function* saja. Berdasarkan hasil pengujian menggunakan 20 kelas dan 50 kelas, dapat disimpulkan bahwa 20 kelas menghasilkan akurasi lebih tinggi dibandingkan dengan 50 kelas. Hasil pengujian juga menunjukkan bahwa nilai k yang paling optimal untuk menghasilkan akurasi tertinggi adalah $k=1$. Pengujian yang dilakukan dengan menggunakan 3 kelompok data latih (5 data latih, 10 data latih, dan 14 data latih) menunjukkan bahwa 10 data latih dan 14 data latih merupakan jumlah data latih yang optimal.

5. REFERENCE

- Alkhatib, K., Najadat, H., Hmeidi, I. & Shatnawi, M. K. A., 2013. Stock Price Prediction Using K-Nearest Neighbor(K-NN) Algoritma. *Business, Humanities and Technology*, 3(3), pp. 32-44.
- Ardiansyah, R. F., 2015. *Pengenalan Pola Tanda Tangan dengan menggunakan Metode Principal Component Analysis (PCA)*, Semarang: s.n.
- Bhonsle, S. & Klinzmann, A., 2011. *Centroid Distance Function and the Fourier Descriptor with Applications to Cancer Cell Clustering*, s.l.: s.n.
- Boateng, K. O., Asuban, B. W. & Laar, S. D., 2012. Improving the Effectiveness of the Median Filter. *Electronics and Communication Engineering*, 5(1), pp. 85-97.
- Cheriet, M., Kharm, N., Liu, C. L. & Sueh, C. Y., 2007. *Character ecognition Systems - A guide for Student and Practitioners*. New Jersey - USA, Wiley Interscience.
- Hassanat, A. B., Abbadi, M. A. & Altarawneh, G. A., 2014. Solving the problem of the K

- parameter in the K-NN Classifier using an Ensemble learning Approach. *Computer Science and Informatic*, 12(8), pp. 33-39.
- Putra, D., 2010. *Pengolahan Citra Digital*. 1st penyunt. Yogyakarta: C.V ANDI OFFSET.
- Syarifuddin, S. N., 2015. *Analaisis Filtering citra dngan metode mean filter dan median filter*, indoneia: s.n.
- Widodo, A. W. & Harjoko, A., 2015. Sistem Verifikasi Tanda Tangan Off-line berdasarkan Ciri Histrogram of Oriented (HOG) dan Histogram of Curvature (HoC). *Jurnal Teknologi Informasi dan Ilmu Komputer*, 2(1), pp. 1-10.