



Penerapan Algoritma *Support Vector Machine* Dan *Scale Invariant Feature Transform* untuk Pengenalan Tulisan Tangan pada Karakter Hanacaraka Aksara Jawa

Rama Tri Agung
123180053

Penguji:

- Dessyanto Boedi Prasetyo, S.T., M.T.
- Mangaras Yanu Florestiyanto, S.T., M.Eng.
- Herry Sofyan, S. T., M. Kom.
- Wilis Kaswidjanti, S.Si., M.Kom.



Pembahasan

01

Pendahuluan

02

Metode Penelitian dan
Pengembangan
Sistem

03

Hasil dan Pembahasan

04

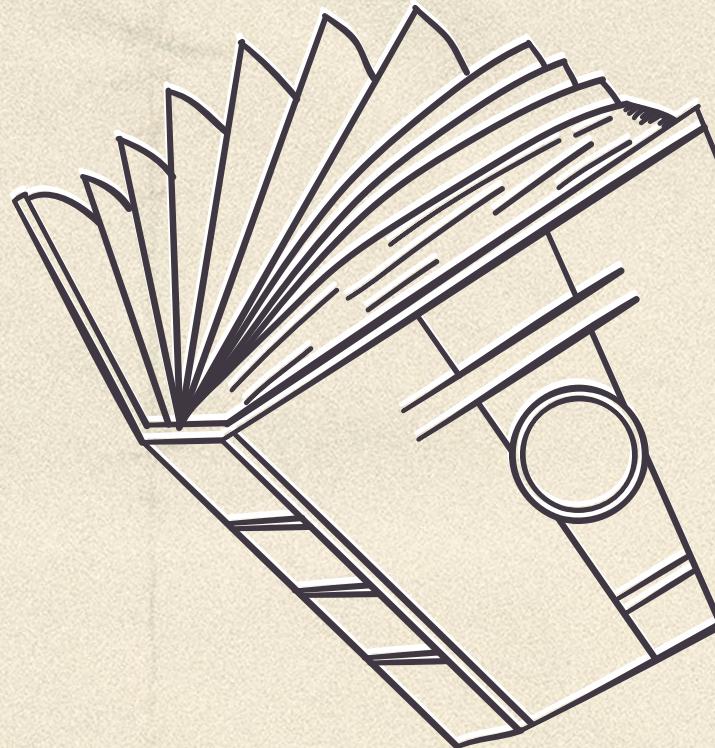
Kesimpulan dan
Saran

05

Hasil Revisi Seminar

06

Demo Program



01

Pendahuluan

Latar Belakang
Rumusan Masalah
Batasan Penelitian
Tujuan Penelitian
Manfaat Penelitian



Latar Belakang

- Aksara Jawa "ha-na-ca-ra-ka" merupakan salah satu warisan leluhur bangsa Indonesia (Sari et al., 2018).
- Ratusan bahasa daerah di Indonesia terancam punah. Salah satu bahasa daerah yang terancam adalah bahasa Jawa (Lorentius et al., 2019).
- Pengguna bahasa Jawa ini semakin berkurang jumlahnya dan hanya sedikit remaja yang mengenal aksara Jawa dengan jelas (Setiawan et al., 2019)
- Minat masyarakat terhadap aksara jawa juga sangat memprihatinkan (Lorentius et al., 2019).



Latar Belakang

- Melihat kondisi tersebut, bagaimana pentingnya nilai dan eksistensi budaya tersebut, maka perlu sebuah sistem yang dapat mengenali huruf-huruf Hanacaraka Aksara Jawa (Lorentius et al., 2019).
- Pengenalan tulisan tangan ini berguna untuk menunjang kelestarian Aksara Jawa sebagai alat atau perangkat lunak yang memiliki kemampuan untuk mengenali tulisan tangan karakter Aksara Jawa secara otomatis (Dewa et al., 2018).



Latar Belakang

- CNN memiliki performa klasifikasi yang sangat baik dalam bidang ini dengan tingkat akurasi yang dapat mencapai 94.57% (Wibowo et al., 2018), CNN juga mahir dalam menangani inputan yang bersifat noisy (Rajesh et al., 2016), namun akurasi yang tinggi pada CNN membutuhkan jumlah data training yang banyak (Wibowo et al., 2018)
- KNN adalah metode yang sederhana, efektif, mudah diterapkan, tidak parametrik dan memberikan tingkat kesalahan yang rendah dalam proses pelatihan (Thamilselvana & Sathiaseelan, 2015), kekurangannya pada metode KNN relatif memiliki hasil performa yang kurang baik daripada metode lainnya (Naufal et al., 2021)
- SVM merupakan metode paling efektif dalam klasifikasi, memiliki akurasi yang cukup tinggi (Thamilselvana & Sathiaseelan, 2015), tidak memiliki masalah dalam overfitting (Rajesh et al., 2016), dan tidak membutuhkan jumlah dataset yang sangat besar (Rismiyati et al., 2018).



Latar Belakang

- SVM dapat digunakan secara fleksibel tanpa membutuhkan dataset yang besar dan memiliki performa yang cukup baik, namun penelitian sebelumnya yang menggunakan metode tersebut belum dapat menyaingi akurasi dari metode CNN yang diatas 90% (Rismiyati et al., 2018) (Sari et al., 2018).
- Pada penilitian pengenalan tulisan tangan karakter lainnya (Thailand, Bangla dan Latin) telah mengusulkan metode yaitu menggunakan Scale Invariant Feature Transform Descriptor (SIFT Descriptor) yang berpengaruh dalam peningkatan akurasi klasifikasi secara signifikan menjadi diatas 95% dan mengungguli performa fitur ekstraksi Histograms of Oriented Gradients (HOG) (Surinta et al., 2015).



Rumusan Masalah



Menerapkan algoritma SVM
dalam klasifikasi



Menerapkan algoritma SIFT
sebagai ekstraksi fitur



Evaluasi performa akurasi
dalam klasifikasi



Batasan Masalah

- Klasifikasi dilakukan hanya pada 20 karakter Hanacaraka Aksara Jawa.
- Sumber data berasal dari dataset yang disediakan di www.kaggle.com oleh Phiard.
- Data diaugmentasi dengan tujuh variasi.
- Data yang digunakan dalam format gambar.
- Analisis dilakukan untuk melihat performa algoritma menggunakan akurasi klasifikasi.



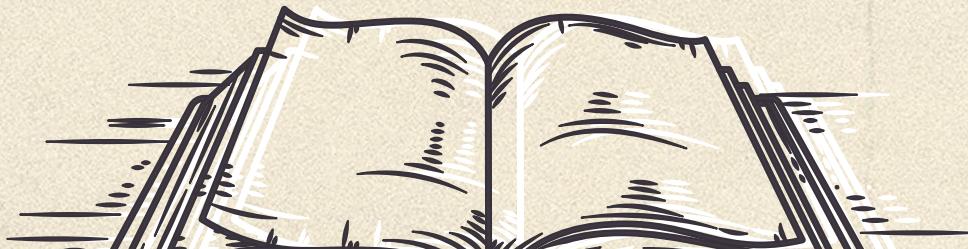
Tujuan Penelitian

Tujuan dari penelitian ini yaitu menerapkan algoritma SVM dengan bantuan SIFT sebagai ekstraksi fitur dalam melakukan klasifikasi tulisan tangan aksara jawa dan mengidentifikasi performa akurasi algoritma yang terbaik dalam melakukan klasifikasi.



Manfaat Penelitian

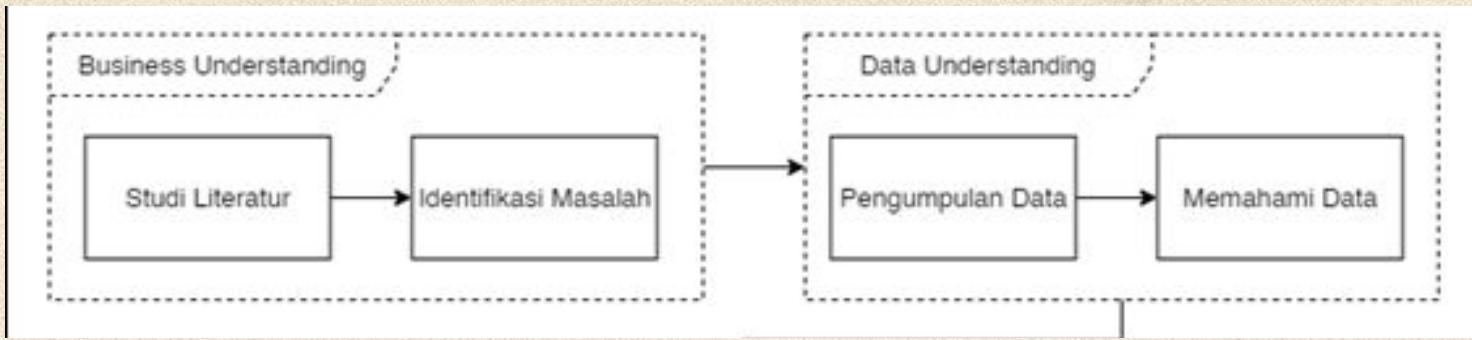
Hasil penelitian dapat dimanfaatkan dalam bantuan pembelajaran baik pada siswa di instansi pendidikan maupun orang lain secara individu dan membangun kembali budaya bahasa khas jawa dengan mengenal lebih mudah terhadap karakter-karakternya.

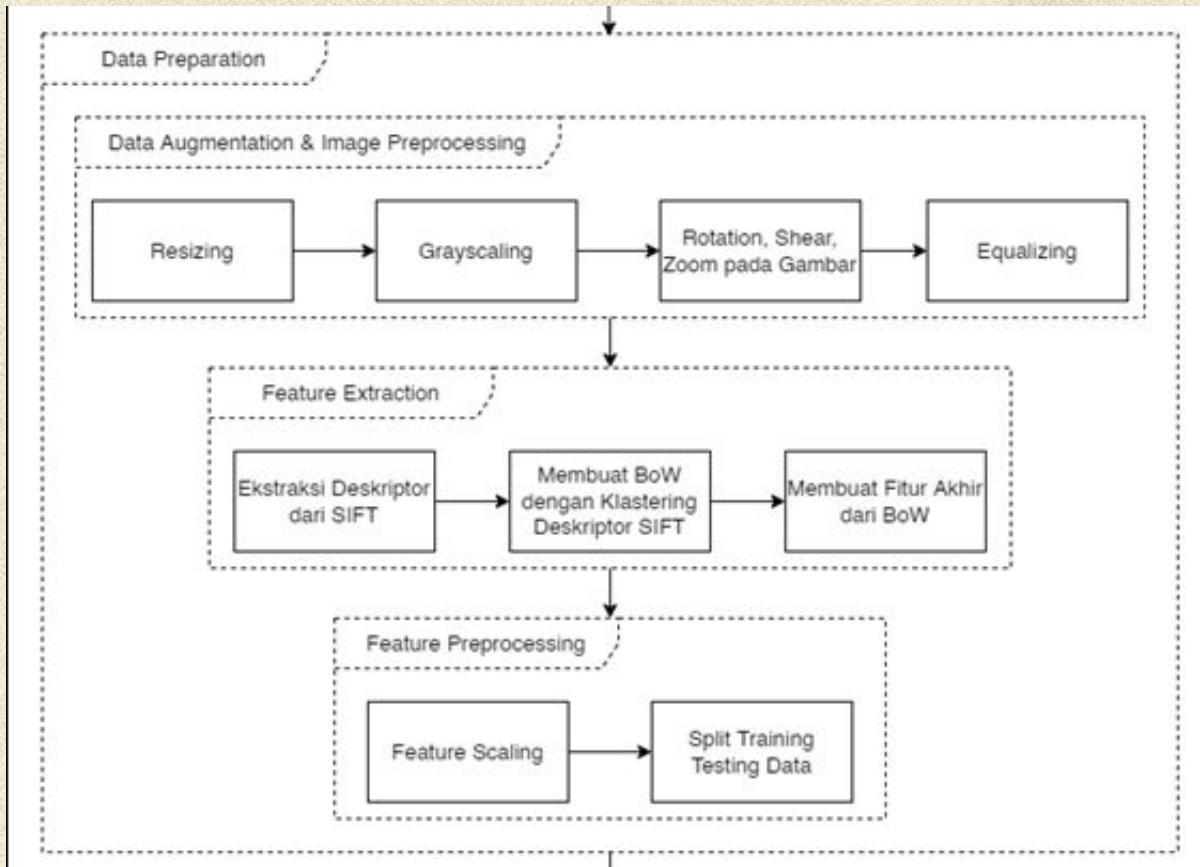


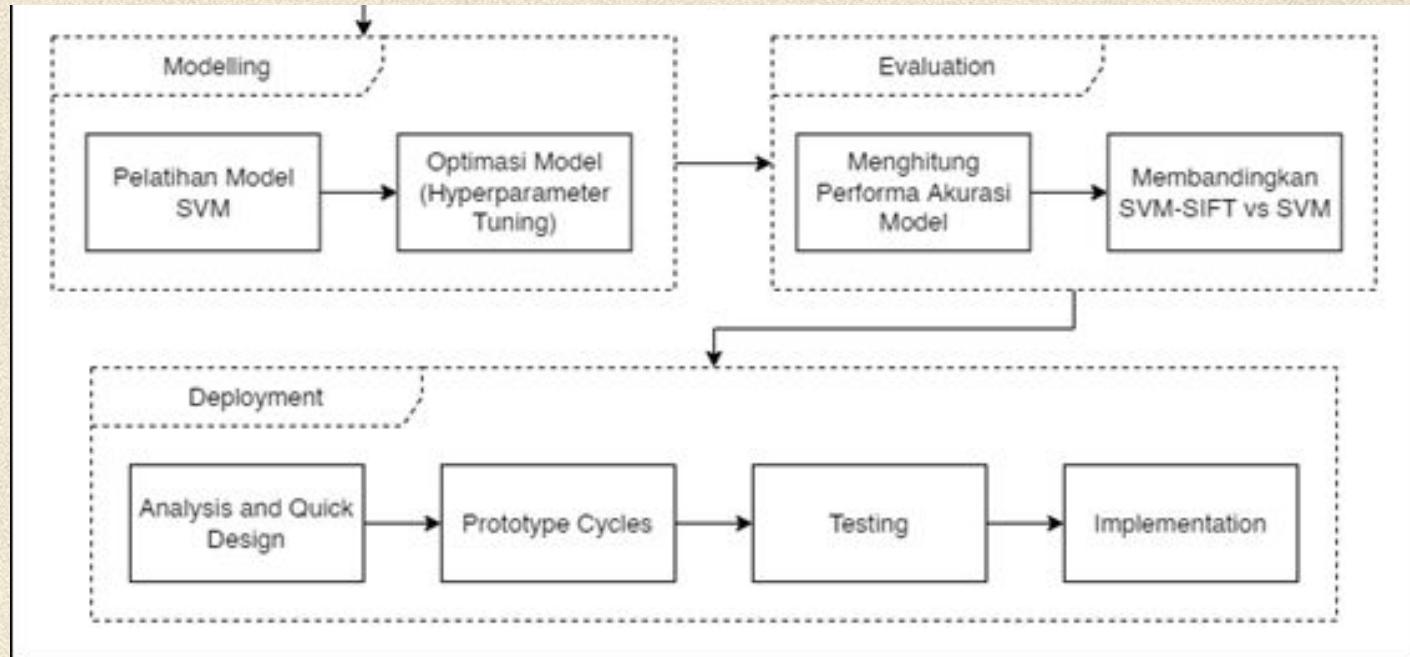


02

Metodologi Penelitian dan Pengembangan Sistem







03

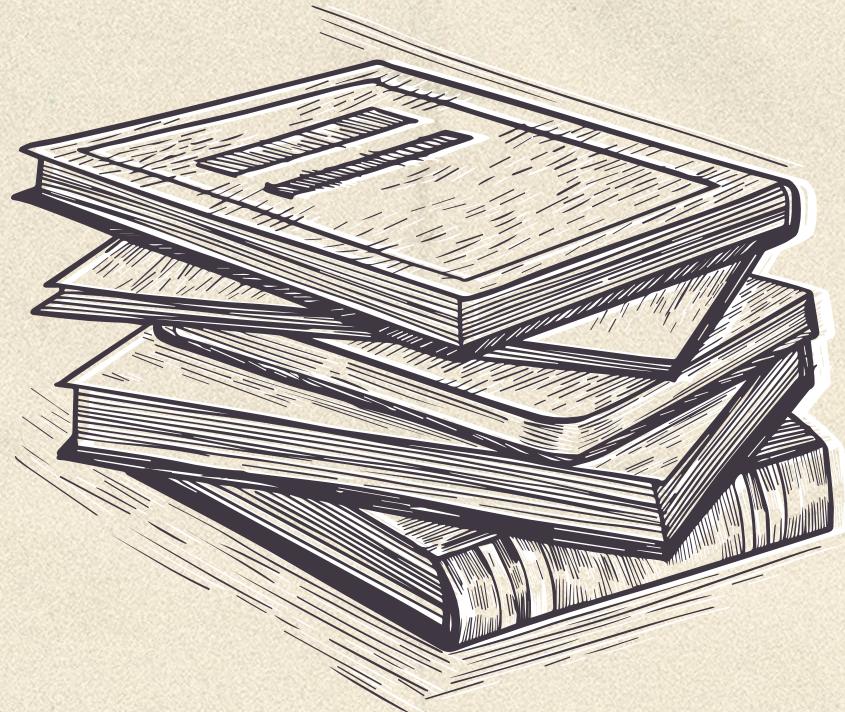
Hasil dan Pembahasan

SVM-SIFT

SVM tanpa SIFT

Perbandingan Performa Model

Pengujian Data Baru pada SVM-SIFT





- Bahasa Pemrograman Python
- Library Utama :
 - Scikit-learn
 - Opencv-python
 - Flask
- Platform Website

SVM with SIFT



SVM-SIFT

Pengujian model SVM-SIFT dilakukan dengan bermacam optimasi parameter diantaranya yaitu:

- *Size* (ukuran gambar)
 - [“128x128”, “160x160”, “192x192”, “224x224”]
- nilai K pada K-Means atau jumlah fitur
 - [180, 250, 500, 750, 1000]
- nilai C sebagai nilai parameter toleransi SVM
 - [1, 3, 6, 10, 15]
- nilai gamma (G) sebagai nilai parameter kernel RBF
 - Bergantung pada nilai Size dan nilai K, dengan perhitungan skala
`'scale = 1 / nfeatures * x.var()'`
Sehingga:
`[scale-(4%*scale), scale-(2%*scale), scale, scale+(2%*scale), scale+(4%*scale)]`

SVM-SIFT

Tabel 4.1 Hasil Pengujian Optimasi Parameter SVM-SIFT

#	Size	K	C	G	R_Train	R_Test
1	128	180	[1,3,6,10,15]	[0.24574 , 0.25086, 0.25598, 0.2611, 0.26622]	89.31%	92.51%
2		250	[1,3,6,10,15]	[0.20504 , 0.20931, 0.21358, 0.21785, 0.22212]	90.63%	92.97%
3		500	[1,3,6,10,15]	[0.13567 , 0.1385, 0.14133, 0.14416, 0.14699]	91.55%	95.01%
4		750	[1,3,6,10,15]	[0.10508 , 0.10727, 0.10946, 0.11165, 0.11384]	91.99%	94.10%
5		1000	[1,3,6,10,15]	[0.09142 , 0.09332, 0.09522, 0.09712, 0.09902]	91.27%	94.55%
6	160	180	[1,3,6,10,15]	[0.25495 , 0.26026, 0.26557, 0.27088, 0.27619]	89.07%	91.69%
7		250	[1,3,6,10,15]	[0.20114, 0.20533 , 0.20952, 0.21371, 0.2179]	89.95%	90.92%
8		500	[1,3,6,10,15]	[0.13022 , 0.13293, 0.13564, 0.13835, 0.14106]	90.63%	91.83%
9		750	[1,3,6,10,15]	[0.10009 , 0.10217, 0.10425, 0.10633, 0.10841]	90.79%	93.19%
10		1000	[1,3,6,10,15]	[0.08725 , 0.08907, 0.09089, 0.09271, 0.09453]	90.07%	92.97%
11	192	180	[1,3,6,10,15]	[0.26274, 0.26821 , 0.27368, 0.27915, 0.28462]	88.47%	90.92%
12		250	[1,3,6,10,15]	[0.20183 , 0.20603, 0.21023, 0.21443, 0.21863]	90.91%	91.60%
13		500	[1,3,6,10,15]	[0.12668, 0.12932, 0.13196 , 0.1346, 0.13724]	91.83%	93.42%
14		750	[1,3,6,10,15]	[0.10235 , 0.10448, 0.10661, 0.10874, 0.11087]	92.11%	94.55%
15		1000	[1,3,6,10,15]	[0.08487 , 0.08664, 0.08841, 0.09018, 0.09195]	92.67%	95.01%
16	224	180	[1,3,6, 10 ,15]	[0.23885, 0.24383 , 0.24881, 0.25379, 0.25877]	85.83%	90.47%
17		250	[1,3,6,10,15]	[0.19131 , 0.1953, 0.19929, 0.20328, 0.20727]	87.99%	90.92%
18		500	[1,3,6,10,15]	[0.11421, 0.11659 , 0.11897, 0.12135, 0.12373]	89.19%	93.19%
19		750	[1,3,6,10,15]	[0.09026 , 0.09214, 0.09402, 0.0959, 0.09778]	90.91%	92.51%
20		1000	[1,3,6,10,15]	[0.07609 , 0.07767, 0.07925, 0.08083, 0.08241]	90.47%	92.74%

Tabel 4.2 Rata-rata Akurasi Terhadap Parameter Size

Size	R_Train	R_Test
128	90.95%	93.83%
160	90.10%	92.12%
192	91.20%	93.10%
224	88.88%	91.97%

Tabel 4.3 Rata-rata Akurasi Terhadap Parameter K

K	R_Train	R_Test
180	88.17%	91.40%
250	89.87%	91.60%
500	90.80%	93.36%
750	91.45%	93.59%
1000	91.12%	93.82%

Parameter yang digunakan

- Size = 192x192
- K = 750
- C = 6
- G = 0.10235

Akurasi

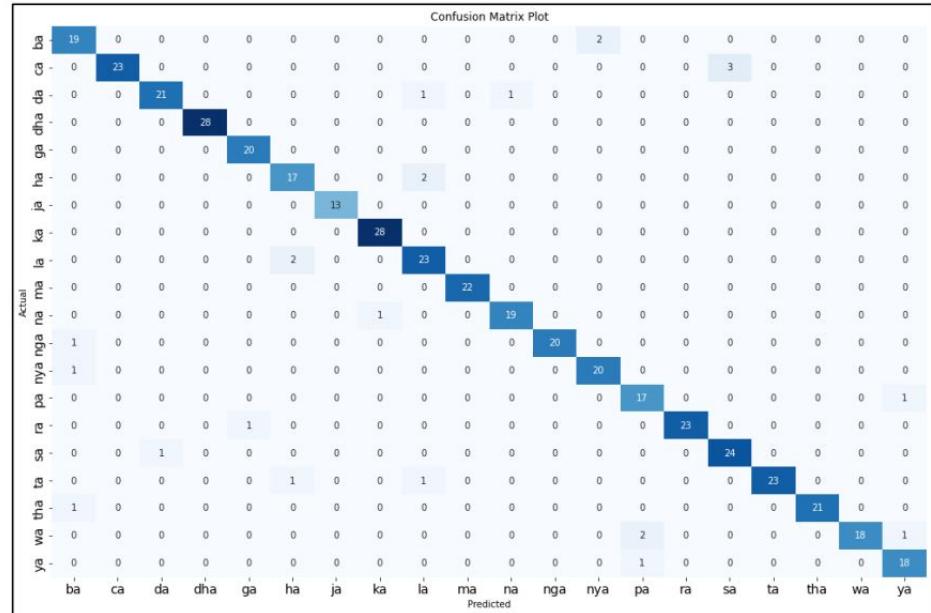
- Data Training = 92.11%
- Data Testing = 94.55%

SVM-SIFT (Skenario 14)

Tabel 4.4 Hasil Pelatihan SVM-SIFT Skenario 14

kernel	params	rank test score	mean test score	std test score
15_0.10235	{'C': 15, 'gamma': 0.10235}	1	0.921174	0.011467
10_0.10235	{'C': 10, 'gamma': 0.10235}	1	0.921174	0.011467
6_0.10235	{'C': 6, 'gamma': 0.10235}	1	0.921174	0.011467
6_0.10661	{'C': 6, 'gamma': 0.10661}	4	0.920774	0.011535
15_0.10661	{'C': 15, 'gamma': 0.10661}	4	0.920774	0.011535
15_0.10448	{'C': 15, 'gamma': 0.10448}	4	0.920774	0.011039
10_0.10661	{'C': 10, 'gamma': 0.10661}	4	0.920774	0.011535
10_0.10448	{'C': 10, 'gamma': 0.10448}	4	0.920774	0.011039
6_0.10448	{'C': 6, 'gamma': 0.10448}	4	0.920774	0.011039
10_0.10874	{'C': 10, 'gamma': 0.10874}	10	0.920773	0.011043
15_0.10874	{'C': 15, 'gamma': 0.10874}	10	0.920773	0.011043
6_0.10874	{'C': 6, 'gamma': 0.10874}	10	0.920773	0.011043
10_0.11087	{'C': 10, 'gamma': 0.11087}	13	0.919973	0.011771
6_0.11087	{'C': 6, 'gamma': 0.11087}	13	0.919973	0.011771
15_0.11087	{'C': 15, 'gamma': 0.11087}	13	0.919973	0.011771
3_0.10235	{'C': 3, 'gamma': 0.10235}	16	0.919574	0.012767
3_0.10661	{'C': 3, 'gamma': 0.10661}	17	0.919174	0.012904
3_0.10448	{'C': 3, 'gamma': 0.10448}	17	0.919174	0.012462
3_0.10874	{'C': 3, 'gamma': 0.10874}	19	0.918374	0.013077
3_0.11087	{'C': 3, 'gamma': 0.11087}	20	0.917574	0.013906
1_0.10235	{'C': 1, 'gamma': 0.10235}	21	0.907567	0.016443
1_0.10448	{'C': 1, 'gamma': 0.10448}	22	0.907167	0.016500
1_0.10874	{'C': 1, 'gamma': 0.10874}	23	0.906766	0.016648
1_0.10661	{'C': 1, 'gamma': 0.10661}	23	0.906766	0.016259
1_0.11087	{'C': 1, 'gamma': 0.11087}	25	0.905966	0.016524

SVM-SIFT (Skenario 14)



Gambar 4.11 Hasil Confusion Matrix SVM-SIFT Skenario 14

Classification report result:				
	precision	recall	f1-score	support
ba	0.86	0.90	0.88	21
ca	1.00	0.88	0.94	26
da	0.95	0.91	0.93	23
dha	1.00	1.00	1.00	28
ga	0.95	1.00	0.98	20
ha	0.85	0.89	0.87	19
ja	1.00	1.00	1.00	13
ka	0.97	1.00	0.98	28
la	0.85	0.92	0.88	25
ma	1.00	1.00	1.00	22
na	0.95	0.95	0.95	20
nga	1.00	0.95	0.98	21
nya	0.91	0.95	0.93	21
pa	0.85	0.94	0.89	18
ra	1.00	0.96	0.98	24
sa	0.89	0.96	0.92	25
ta	1.00	0.92	0.96	25
tha	1.00	0.95	0.98	22
wa	1.00	0.86	0.92	21
ya	0.90	0.95	0.92	19
accuracy			0.95	441
macro avg	0.95	0.95	0.95	441
weighted avg	0.95	0.95	0.95	441

Gambar 4.12 Hasil Classification Report SVM-SIFT Skenario 14

SVM without SIFT



SVM tanpa SIFT

Pengujian model SVM tanpa SIFT dilakukan dengan bermacam optimasi parameter diantaranya yaitu:

- *Size* (ukuran gambar)
 - 32x32
- nilai C sebagai nilai parameter toleransi SVM
 - [1, 3, 6, 10, 15]
- nilai gamma (G) sebagai nilai parameter kernel RBF
 - [0.02265, 0.02312, 0.02359, 0.02406, 0.02453]

SVM tanpa SIFT

Tabel 4.5 Hasil Pelatihan SVM tanpa SIFT

kernel	params	rank_test_score	mean_test_score	std_test_score
3_0.02312	{'C': 3, 'gamma': 0.02312}	1	0.511807	0.011146
3_0.02265	{'C': 3, 'gamma': 0.02265}	2	0.511407	0.011168
3_0.02406	{'C': 3, 'gamma': 0.02406}	3	0.510608	0.010277
3_0.02359	{'C': 3, 'gamma': 0.02359}	4	0.510207	0.010708
15_0.02265	{'C': 15, 'gamma': 0.02265}	5	0.509408	0.013446
10_0.02265	{'C': 10, 'gamma': 0.02265}	5	0.509408	0.013446
6_0.02265	{'C': 6, 'gamma': 0.02265}	5	0.509408	0.012712
6_0.02312	{'C': 6, 'gamma': 0.02312}	8	0.509407	0.013052
3_0.02453	{'C': 3, 'gamma': 0.02453}	9	0.509007	0.010085
10_0.02312	{'C': 10, 'gamma': 0.02312}	9	0.509007	0.012622
15_0.02312	{'C': 15, 'gamma': 0.02312}	9	0.509007	0.012622
6_0.02406	{'C': 6, 'gamma': 0.02406}	12	0.507407	0.014696
10_0.02406	{'C': 10, 'gamma': 0.02406}	13	0.507406	0.013294
10_0.02359	{'C': 10, 'gamma': 0.02359}	13	0.507406	0.012423
15_0.02359	{'C': 15, 'gamma': 0.02359}	13	0.507406	0.012423
15_0.02406	{'C': 15, 'gamma': 0.02406}	13	0.507406	0.013294

Tabel 4.6 Hasil Pelatihan SVM tanpa SIFT Lanjutan

kernel	params	rank_test_score	mean_test_score	std_test_score
6_0.02359	{'C': 6, 'gamma': 0.02359}	17	0.507006	0.013539
6_0.02453	{'C': 6, 'gamma': 0.02453}	18	0.507004	0.014284
10_0.02453	{'C': 10, 'gamma': 0.02453}	19	0.506604	0.014053
15_0.02453	{'C': 15, 'gamma': 0.02453}	19	0.506604	0.014053
1_0.02453	{'C': 1, 'gamma': 0.02453}	21	0.449780	0.006010
1_0.02312	{'C': 1, 'gamma': 0.02312}	22	0.447782	0.010193
1_0.02406	{'C': 1, 'gamma': 0.02406}	23	0.447780	0.006575
1_0.02359	{'C': 1, 'gamma': 0.02359}	24	0.447381	0.007565
1_0.02265	{'C': 1, 'gamma': 0.02265}	25	0.445781	0.009792

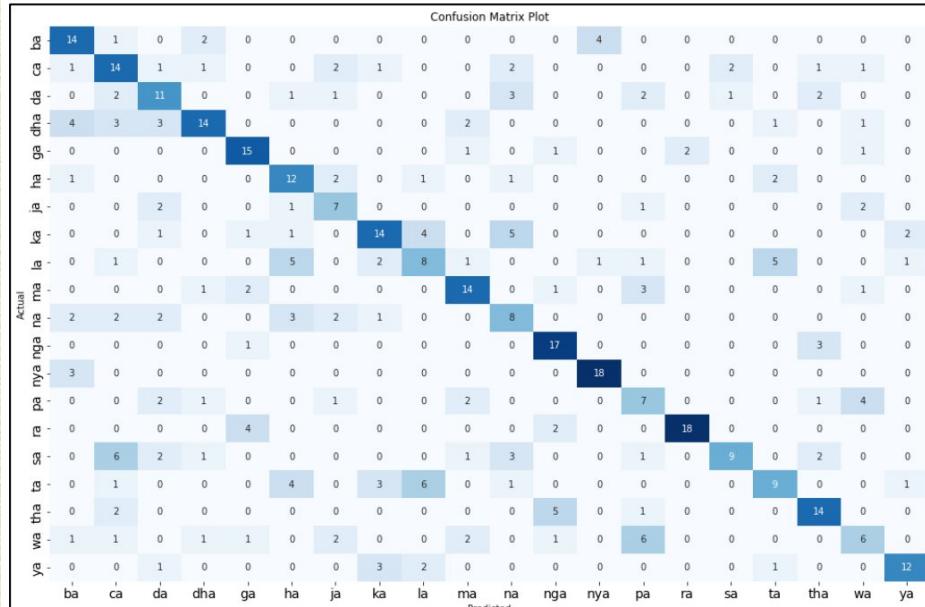
Parameter yang digunakan

- Size = 32x32
- C = 3
- G = 0.02312

Akurasi

- Data Training = 51.18%

SVM tanpa SIFT



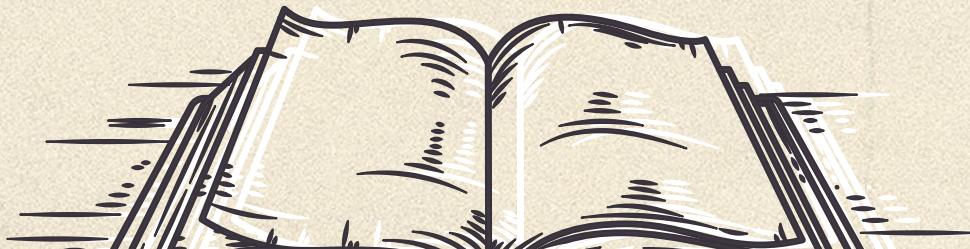
Gambar 4.13 Hasil Confusion Matrix SVM tanpa SIFT

Akurasi pada data testing = 54.64%

classification report result:				
	precision	recall	f1-score	support
ba	0.54	0.67	0.60	21
ca	0.42	0.54	0.47	26
da	0.44	0.48	0.46	23
dha	0.67	0.50	0.57	28
ga	0.62	0.75	0.68	20
ha	0.44	0.63	0.52	19
ja	0.41	0.54	0.47	13
ka	0.58	0.50	0.54	28
la	0.38	0.32	0.35	25
ma	0.61	0.64	0.62	22
na	0.35	0.40	0.37	20
nga	0.63	0.81	0.71	21
nya	0.78	0.86	0.82	21
pa	0.32	0.39	0.35	18
ra	0.90	0.75	0.82	24
sa	0.75	0.36	0.49	25
ta	0.50	0.36	0.42	25
tha	0.61	0.64	0.62	22
wa	0.38	0.29	0.32	21
ya	0.75	0.63	0.69	19
accuracy			0.55	441
macro avg	0.55	0.55	0.54	441
weighted avg	0.56	0.55	0.54	441

Gambar 4.14 Hasil Classification Report SVM tanpa SIFT

Comparison



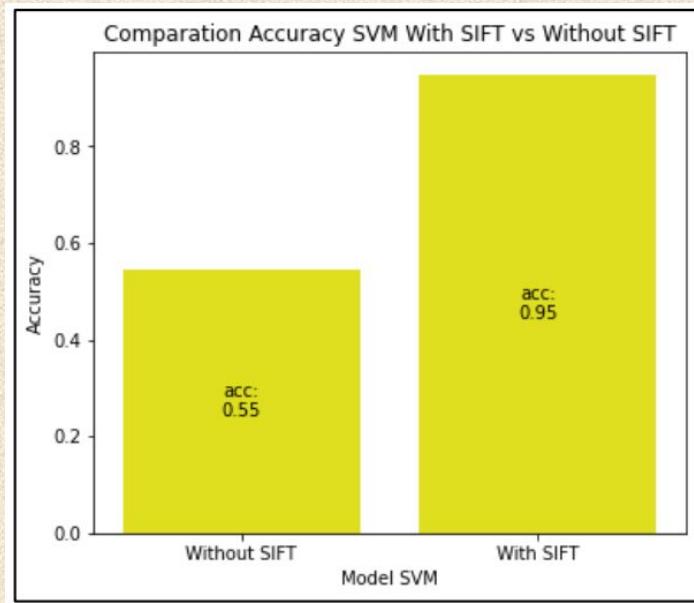
Comparison

SVM tanpa SIFT:

- Data Training = 51.18%
- Data Testing = 54.64%

SVM-SIFT:

- Data Training = 92.11%
- Data Testing = 94.55%



Comparison

Tabel 4.9 Perbandingan Akurasi dengan Penelitian Lain

No	Penulis	Metode	Ekstraksi Fitur	Akurasi
1	Rismiyati et al. (2017)	CNN	-	70.22%
2	Rismiyati et al. (2018)	SVM	<i>Zone Based</i>	81.98%
3	Widiarti & Wastu (2009)	HMM	<i>Horizontal & Vertical Vector</i>	85.7%
4	Sari et al. (2018)	KNN	<i>Roundness & Eccentricity</i>	87.5%
5	Rismiyati et al. (2018)	SVM	HOG	88.45%
6	Dewa et al. (2018)	CNN	-	89%
7	Yulianti et al. (2019)	SVM	<i>Moment Invariant</i>	92.57%
8	Wibowo et al. (2018)	CNN	-	94.57%
9	Rasyidi et al. (2021)	<i>Random Forest</i>	-	97.77%
10	Susanto et al. (2021)	KNN	HOG	98.5%
11	Penelitian ini	SVM	SIFT	94.55%

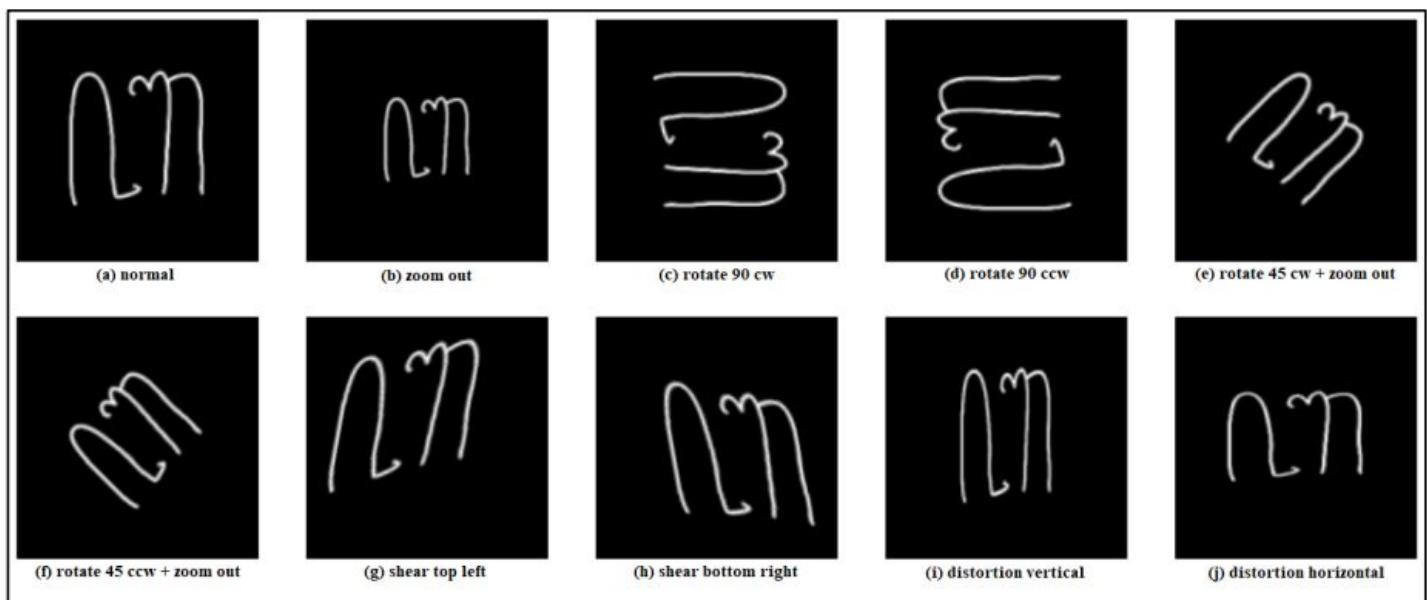
Testing New Data



Testing New Data

- Menggunakan 10 variasi skala data yang berbeda:
 - a. Gambar normal
 - b. Pengecilan gambar
 - c. Rotasi 90 derajat searah jarum jam
 - d. Rotasi 90 derajat berlawanan arah jarum jam
 - e. Rotasi 45 derajat searah jarum jam dan pengecilan gambar
 - f. Rotasi 45 derajat berlawanan arah jarum jam dan pengecilan gambar
 - g. Shear gambar kearah atas kiri
 - h. Shear gambar kearah bawah bawah
 - i. Distorsi secara vertikal
 - j. Distorsi secara horizontal
- Mengambil satu data secara acak setiap kelas
- Jumlah data = $20 * 10 = 200$ data

Testing New Data



Gambar 4.16 Contoh Variasi Sampel Data Karakter ‘Ba’

Testing New Data

label	var_a	var_b	var_c	var_d	var_e	var_f	var_g	var_h	var_i	var_j	total
ba	ba	ha	9								
ca	ca	sa	9								
da	da	10									
dha	dha	ma	9								
ga	ga	10									
ha	ha	la	ha	ha	9						
ja	ja	ja	ja	ja	ja	ja	pa	ja	ja	ja	9
ka	ka	10									
la	la	la	la	la	la	la	ha	la	ga	ha	7
ma	ma	10									
na	na	da	na	9							
nga	nga	10									
nya	nya	ba	nya	9							
pa	pa	wa	9								
ra	ra	10									
sa	sa	10									
ta	ta	10									
tha	tha	10									
wa	wa	wa	wa	wa	wa	wa	dha	wa	wa	wa	9
ya	ya	nya	9								
total	20	20	20	20	20	20	17	19	17	14	187

Testing New Data

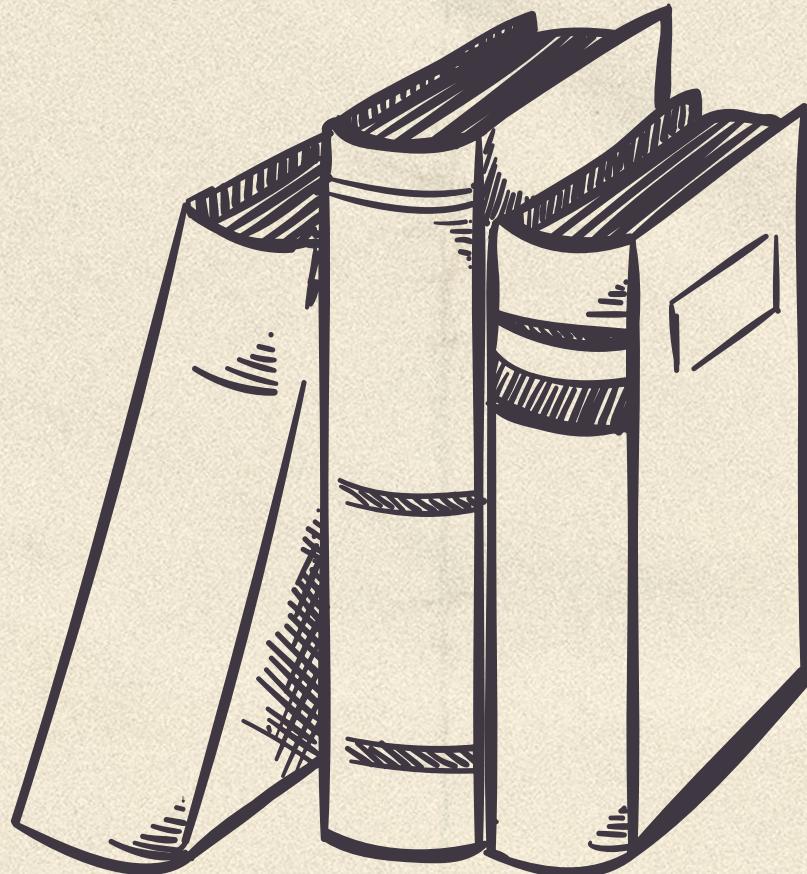
Classification report result:				
	precision	recall	f1-score	support
ba	0.90	0.90	0.90	10
ca	1.00	0.90	0.95	10
da	0.91	1.00	0.95	10
dha	0.90	0.90	0.90	10
ga	0.91	1.00	0.95	10
ha	0.75	0.90	0.82	10
ja	1.00	0.90	0.95	10
ka	1.00	1.00	1.00	10
la	0.88	0.70	0.78	10
ma	0.91	1.00	0.95	10
na	1.00	0.90	0.95	10
nga	1.00	1.00	1.00	10
nya	0.90	0.90	0.90	10
pa	0.90	0.90	0.90	10
ra	1.00	1.00	1.00	10
sa	0.91	1.00	0.95	10
ta	1.00	1.00	1.00	10
tha	1.00	1.00	1.00	10
wa	0.90	0.90	0.90	10
ya	1.00	0.90	0.95	10
accuracy			0.94	200
macro avg	0.94	0.93	0.93	200
weighted avg	0.94	0.94	0.93	200

Akurasi yang diperoleh **93.5%**

Gambar 4.17 Hasil *Classification Report* SVM-SIFT pada Sampel Data

04

Kesimpulan dan Saran



Kesimpulan



Kesimpulan

- Pengenalan 20 karakter hanacaraka aksara jawa menggunakan metode SVM dan SIFT sebagai ekstraksi fitur berhasil diterapkan dengan jumlah data 2940 yang telah dilakukan augmentasi (rotasi, pengecilan gambar, shear) dan telah dilakukan image preprocessing (resize, grayscale, equalization histogram).
- Metode SIFT berhasil bekerja dengan melakukan ekstraksi fitur dari data gambar, kemudian fitur keypoint yang penting pada gambar dipilih dan dimasukkan ke dalam BoW dengan menggunakan metode K-Means. BoW tersebut digunakan untuk membuat fitur akhir dari SIFT agar menyesuaikan ukuran yang dibutuhkan oleh model.
- Metode SVM berhasil bekerja menggunakan kernel RBF. Data yang dimasukkan ke dalam model dipecah menjadi data latih dan data uji dengan rasio 85:15% sehingga jumlah data latih sebesar 2499 dan data uji sebesar 441.



Kesimpulan

- Evaluasi model pada penelitian ini dilakukan dengan mengidentifikasi performa akurasi dari model SVM-SIFT yang telah dibangun. Dalam prosesnya, dalam meningkatkan akurasi dilakukan parameter tuning dari beberapa parameter yaitu ukuran gambar, nilai K pada K-Means, nilai C pada SVM, dan nilai G pada RBF. Terdapat 20 skenario yang dilakukan dalam parameter tuning, hasil performa terbaik diperoleh pada skenario 14 dimana ukuran gambar = 192, K = 750, C = 6, dan G = 0.10235. Skenario tersebut memperoleh tingkat akurasi yang tinggi yaitu 92.11% pada data latih dan 94.55% pada data uji. Performa akurasi yang diperoleh pada model SVM-SIFT sangat tinggi dibandingkan dengan performa akurasi yang diperoleh pada model SVM tanpa SIFT yaitu hanya dapat mencapai akurasi 54.64% pada data uji.
- Pengujian lainnya dengan pendekatan yang berbeda pada model SVM-SIFT berfokus pada performa metode SIFT dilakukan dengan sampel data berupa 10 varian cukup ekstrim dari rotasi, skala, shear, dan distorsi pada setiap label. Pengujian ini memberikan hasil bahwa kemampuan ekstraksi fitur SIFT sangat baik tanpa ada kesalahan pada varian rotasi dan skala, sedangkan metode SIFT melemah ketika dihadapi dengan data yang memiliki perubahan bentuk dari gambar yang pernah ditemui seperti pada varian shear dan distorsi.



Saran



Saran

- Pada proses tahapan image preprocessing dapat menggunakan metode lainnya untuk dapat meningkatkan kualitas dari data gambar.
- Dalam penelitian lebih lanjut dapat melakukan segmentasi huruf hanacaraka dari serangkaian kata ataupun kalimat.
- Dalam hal evaluasi penelitian juga dapat diperluas lebih lanjut dengan memperhatikan bagaimana performa selain akurasi seperti kompleksitas waktu maupun komputasi





05

Hasil Revisi Seminar

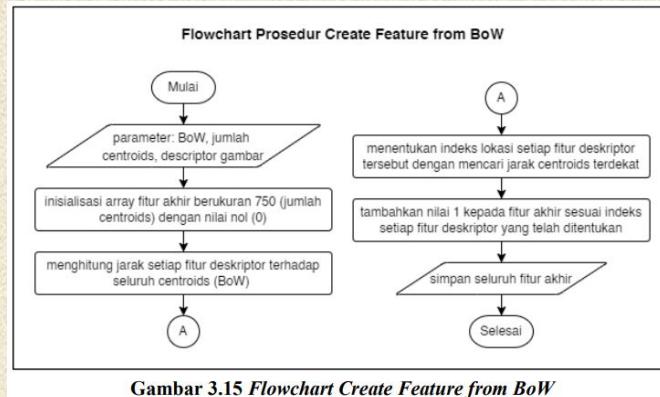
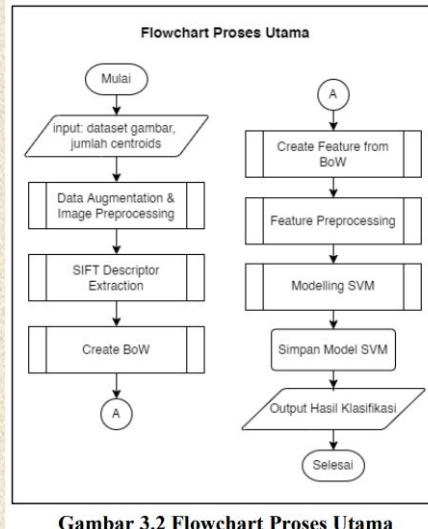
Revisi 1

Catatan Penguji

Memperbaiki flowchart pada bab 3 sehingga flowchart tidak terpotong dengan blok 'selesai' karena masih terdapat proses selanjutnya.

Perbaikan

Perbaikan telah dilakukan dengan mengubah semua flowchart proses sebagai fungsi/prosedur yang memiliki nilai kembalian. Dan membuat flowchart proses umum di bagian awal bab 3 yang berisi pemanggilan fungsi pada tiap proses



Gambar 3.15 Flowchart Create Feature from BoW

Revisi 2

Catatan Penguji

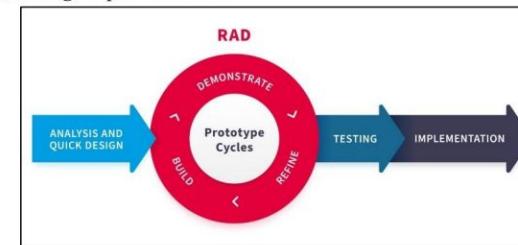
Memberikan saran terhadap metode pengembangan aplikasi yang dilakukan agar lebih masuk akal dengan kasus pengembangan yang dilakukan secara real, sehingga pilihlah metode pengembangan aplikasi yang sederhana saja.

Perbaikan

Perbaikan dilakukan dengan mengubah metode Agile menjadi metode Rapid Application Development (RAD)

6. Deployment

Melakukan pengembangan sistem aplikasi berbasis website dari hasil model yang telah selesai dilatih dan diuji dengan baik. *Deployment* ini fase terakhir yang berguna untuk menyediakan interaksi pengguna terhadap aplikasi sehingga dapat dioperasikan secara luas atau global. Tahap ini juga disebut dengan proses pengembangan perangkat lunak. Proses pengembangan ini akan menggunakan metode *Rapid Application Development* (RAD) yang merupakan salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat (Noertjahyana, 2004). RAD dapat dijadikan acuan untuk mengembangkan suatu sistem informasi yang unggul dalam hal kecepatan, ketepatan dan biaya yang lebih rendah (Hidayat & Hati, 2021). Beberapa proses yang ada di dalam metode RAD diantaranya yaitu *analysis and quick design, prototype cycles (build, demonstrate, refine), testing, implementation*.



Gambar 1.2 Metode Pengembangan Perangkat Lunak RAD

Revisi 3

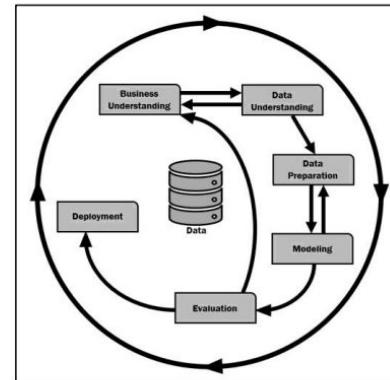
Catatan Penguji

Jelaskan terlebih dahulu singkatan dari CRISP-DM sebelum menuliskannya.

Perbaikan

Perbaikan dilakukan dengan menjelaskan apa itu CRISP-DM di bagian awal subbab 1.6.

Tahapan-tahapan pada penelitian ini menerapkan metode proses *Cross Industry Standard Process for Data Mining* (CRISP-DM) merujuk pada penelitian oleh Rasyidi, et al. (2021), Dowa, et al. (2018), dan Schröera, et al. (2021) yang telah dimodifikasi untuk menyesuaikan kebutuhan penelitian. Metode ini adalah suatu standarisasi pemrosesan *data mining* yang telah dikembangkan dimana data yang ada akan melewati setiap fase terstruktur dan terdefinisi dengan jelas dan efisien (Hasanah et al., 2021). Berikut adalah proses CRISP-DM yang diterapkan pada penelitian ini



Gambar 1.1 *CRISP-DM Data Science Process*

Revisi 4

Catatan Penguji

Pada bab 2 tidak perlu membuat subbab baru dengan nama dasar teori dan studi literatur, langsung digabung saja.

Perbaikan

Perbaikan dilakukan dengan menghapus subbab 2.1. dan 2.2. kemudian level subbab di bawahnya akan dinaikkan.

BAB II TINJAUAN LITERATUR	5
2.1. Hanacaraka Aksara Jawa	5
2.2. Pengenalan Karakter Tulisan Tangan	5
2.3. <i>Machine Learning</i>	5
2.4. Augmentasi Data Gambar	6
2.5. Pengolahan Citra	7
2.6. <i>Scale Invariant Feature Transform</i>	9
2.7. <i>K-Means Clustering</i>	12
2.8. <i>Bag of Words</i>	13
2.9. <i>Feature Scaling</i>	13
2.10. <i>Support Vector Machine</i>	14
2.11. Optimasi Parameter Model	16
2.12. Evaluasi	17
2.13. Penelitian Sebelumnya	17

Revisi 5

Catatan Pengudi

Tambahkan pada tinjauan literatur jika ada penelitian yang hanya menggunakan metode SVM pada objek karakter hanacaraka aksara jawa.

Perbaikan

Tidak menemukan penelitian terkait yang hanya menggunakan metode SVM saja.



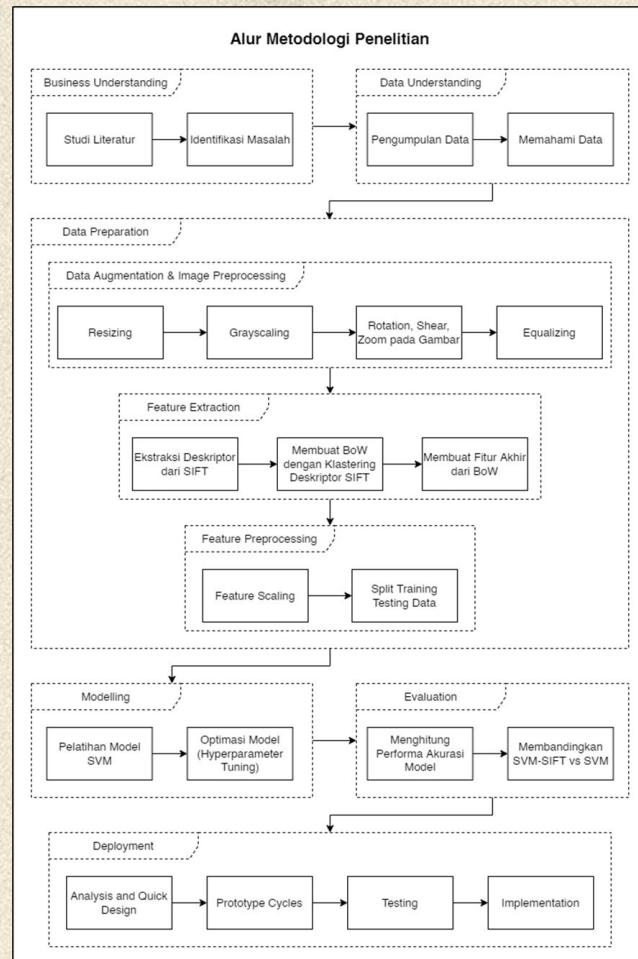
Revisi 6

Catatan Penguji

Mengubah penamaan ‘preprocessing’ pada diagram alur metodologi penelitian.

Perbaikan

Perbaikan dilakukan dengan mengganti judul ‘preprocessing’ menjadi ‘feature preprocessing’ pada diagram alur metodologi penelitian.



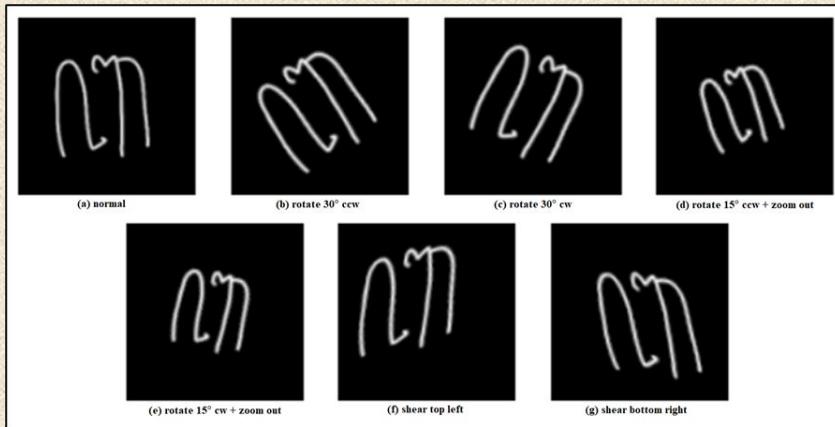
Revisi 7

Catatan Penguji

Pada bab 3 bagian judul data augmentasi dan image preprocessing sebaiknya digabung saja menyesuaikan flowchart, kemudian gambar hasil proses tersebut dijelaskan lebih rinci pada tiap gambar.

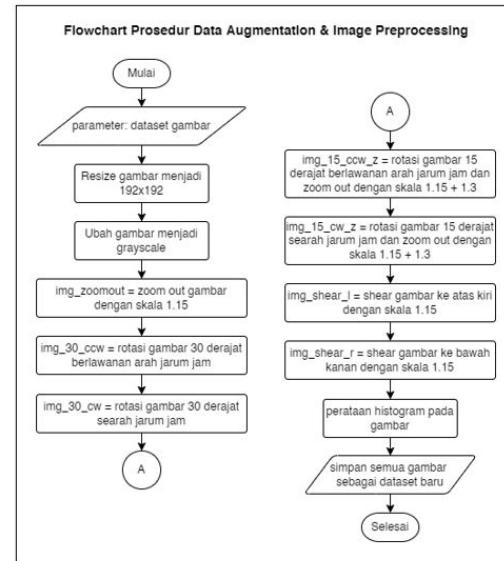
Perbaikan

Perbaikan dilakukan dengan menggabungkan judul data augmentasi dan image preprocessing, kemudian menambahkan penjelasan pada gambar dari hasil proses.



3.3.1. Data Augmentation & Image Preprocessing

Proses pada tahapan ini secara ringkas dapat dilihat pada gambar *flowchart* dibawah berikut ini.



Gambar 3.4 Flowchart Image Augmentation & Preprocessing Process

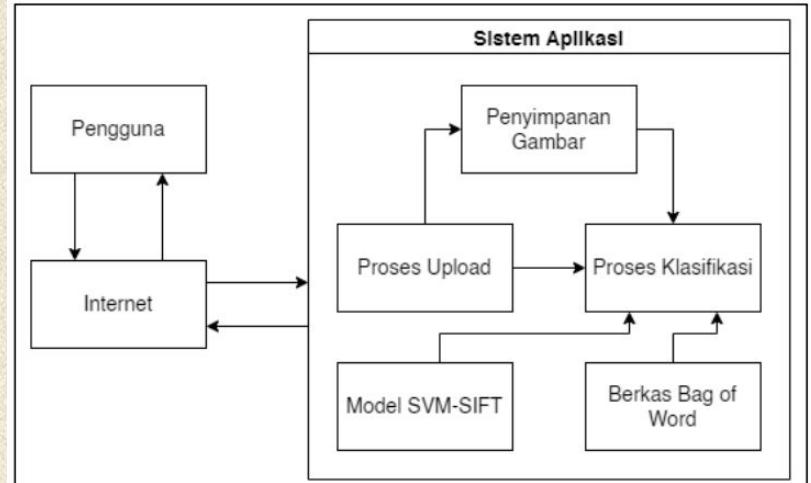
Revisi 8

Catatan Penguji

Pada bagian design subbab 3.6.2 perlu dijelaskan lebih rinci terhadap arsitektur yang dibangun seperti penambahan penggunaan internet dan penjelasan alur data yang ada pada platform github.

Perbaikan

Perbaikan dilakukan dengan menambahkan rincian arsitektur aplikasi.



Gambar 3.18 Perancangan Arsitektur Sistem

06

Demo Program



Matur Nuwun

CREDITS:

This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik

