BAB IV

HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian dan pembahasan mengenai pengembangan model klasifikasi aksara Bali pada manuskrip lontar menggunakan metode gabungan *Convolutional Neural Network (CNN) dan Support Vector Machine (SVM).* Pembahasan diuraikan ke dalam enam subbab secara sistematis. Subbab 4.1 membahas tahap persiapan dataset, meliputi pembagian data dan augmentasi data latih. Subbab 4.2 menyajikan pengujian dan penentuan parameter optimal model yang dilakukan secara bertahap, meliputi penentuan konfigurasi awal model serta penentuan konfigurasi lanjutan model. Subbab 4.3 menyajikan hasil pelatihan dan evaluasi dari model terbaik yang diperoleh. Subbab 4.4 membahas perbandingan dan analisis performa model CNN dan CNN-SVM serta perbandingan dengan penelitian sebelumnya .

4.1 Persiapan Data Set

Kualitas dan kesiapan dataset memiliki pengaruh yang signifikan terhadap performa model yang dibangun. Oleh karena itu, sebelum proses pelatihan dimulai, dilakukan tahap persiapan dataset yang mencakup pembagian data dan *augmentas*i data *training*. Pembagian data bertujuan untuk memisahkan data yang digunakan dalam proses pelatihan, pemantauan model, dan evaluasi akhir, sedangkan augmentasi data bertujuan untuk memperbesar variasi data training sehingga model tidak hanya menghafal pola dari data latih, melainkan mampu mengenali pola secara lebih umum dan akurat.

4.1.1 Pembagian Data set

Dataset yang digunakan dalam penelitian ini merupakan dataset publik yang bersumber dari AMADI_LontarSet CFHR 2016 *Challenge 3*. Dataset tersebut secara keseluruhan terdiri dari 19.392 citra yang telah terbagi menjadi 11.718 citra untuk data training dan 7.674 citra untuk data testing. Guna mendukung proses pelatihan model yang lebih optimal, data training yang tersedia kemudian dibagi kembali menjadi data *training* dan data *validation* dengan rasio 80:20, sehingga diperoleh sekitar 9.374 citra untuk training dan 2.344 citra untuk validation.

Pembagian data validation dari data training dilakukan karena data validation memiliki peran penting dalam proses pelatihan model deep learning. Data validation berfungsi sebagai acuan evaluasi performa model pada setiap akhir *epoch* tanpa ikut serta dalam proses pembaruan bobot jaringan, sehingga model dapat dipantau secara objektif selama pelatihan berlangsung. Selain itu, keberadaan data validation memungkinkan pendeteksian dini terhadap kondisi overfitting, yaitu ketika model terlalu menyesuaikan diri dengan data training sehingga kehilangan kemampuan generalisasi terhadap data baru. Dengan demikian, penggunaan data validation menjadi langkah yang diperlukan untuk memastikan model yang dihasilkan memiliki performa yang stabil.

Pembagian keseluruhan dataset yang digunakan dalam penelitian ini dapat dilihat pada Tabel 4.1. Tabel ini menyajikan informasi mengenai jumlah citra pada masing-masing subset data, yaitu data training, data validation dan data testing yang digunakan selama proses pelatihan berlangsung.

Tabel 4.1
Distribusi Data Set

No	Kelas	Train	Val	Test	Total
1	0	48	12	17	77
2	1	56	14	19	89
3	2	24	6	10	40
4	3	32	8	11	51
5	4	24	6	13	43
6	5	20	5	4	29
7	6	20	5	10	35
8	7	16	4	8	28
9	8	8	2	6	16
10	9	20	5	7	32
11	A	160	40	250	450
12	A-KARA	32	8	12	52
13	A-TEDONG	8	2	6	16
14	ADEG-ADEG	160	40	250	450
15	BA	160	40	248	448
16	BA-KEMBANG	40	10	18	68
17	BI	8	2	13	23
18	BISAH	160	40	250	450
19	BRA	16	4	12	32
20	BU	48	12	13	73
21	CA	120	30	17	167
22	CECEK	160	40	250	450
23	CU	8	2	6	16
24	DA	160	40	250	450
25	DA-MADU	136	34	13	183
26	DA-TEDONG	28	7	10	45
27	DI	88	22	14	124
28	DU	96	24	12	132
29	E-KARA	8	2	9	19
30	GA	160	40	173	373
31	GA-TEDONG	8	2	7	17
32	GANTUNGAN-A	160	40	19	219
33	GANTUNGAN-BA	24	6	15	45

No	Kelas	Train	Val	Test	Total
34	GANTUNGAN-CA	16	4	14	34
35	GANTUNGAN-DA	160	40	32	232
36	GANTUNGAN-GA	64	16	15	95
37	GANTUNGAN-JA	16	4	8	28
38	GANTUNGAN-KA	160	40	11	211
39	GANTUNGAN-LA	72	18	14	104
40	GANTUNGAN-MA	160	40	18	218
41	GANTUNGAN-NA	56	14	20	90
42	GANTUNGAN-NGA	8	2	11	21
43	GANTUNGAN-NYA	16	4	6	26
44	GANTUNGAN-TA	160	40	60	260
45	GANTUNGAN-TA-LATIK	32	8	13	53
46	GANTUNGAN-TA-TAWA	8	2	5	15
47	GEMPELAN-PA	16	4	14	34
48	GEMPELAN-SA-SAPA	24	6	17	47
49	GI	24	6	11	41
50	GNA	16	4	8	28
51	GRA	8	2	4	14
52	GU	64	16	16	96
53	GUWUNG	64	16	21	101
54	GUWUNG-MACELEK	8	2	14	24
55	I	160	40	17	217
56	I-KARA	24	6	17	47
57	IA	8	2	5	15
58	JA	160	40	250	450
59	JA-TEDONG	8	2	14	24
60	JI	20	5	13	38
61	JNA	20	5	12	37
62	JU	48	12	10	70
63	KA	160	40	250	450
64	KA-TEDONG	88	22	14	124
65	KI	64	16	12	92
66	KNA	24	6	13	43
67	KRA	20	5	14	39

No	Kelas	Train	Val	Test	Total
68	KU	160	40	69	269
69	LA	160	40	250	450
70	LA-LENGA	8	2	15	25
71	LA-TEDONG	8	2	8	18
72	LI	40	10	14	64
73	LU	48	12	18	78
74	MA	160	40	250	450
75	MA-TEDONG	64	16	15	95
76	MI	20	5	6	31
77	MU	40	10	10	60
78	NA	160	40	250	450
79	NA-RAMBAT	120	30	12	162
80	NA-RAMBAT-TEDONG	19	5	6	30
81	NA-TEDONG	56	14	11	81
82	NANIA	160	40	182	382
83	NGA	160	40	201	401
84	NGI	8	2	9	19
85	NGU	20	5	14	39
86	NI	136	34	19	189
87	NIA	24	6	11	41
88	NING	64	16	14	94
89	NU	40	10	19	69
90	NYA	80	20	52	152
91	PA	160	40	134	334
92	PAMADA	8	2	6	16
93	PEPET	160	40	198	398
94	PI	8	2	13	23
95	PU	24	6	14	44
96	RA	160	40	221	421
97	RA-TEDONG	24	6	13	43
98	RI	40	10	17	67
99	RU	96	24	19	139
100	SA	160	40	250	450
101	SA-SAGA	88	22	17	127

No	Kelas	Train	Val	Test	Total
102	SA-SAPA	64	16	11	91
103	SI	40	10	5	55
104	SU	24	6	11	41
105	SUKU	160	40	250	450
106	SUKU-ILUT	48	12	18	78
107	SUKU-KEMBUNG	96	24	29	149
108	SURANG	160	40	186	386
109	TA	160	40	250	450
110	TA-TAWA	8	2	4	14
111	TA-TEDONG	40	10	16	66
112	TALENG	160	40	250	450
113	TEDONG	160	40	250	450
114	TI	128	32	14	174
115	TIA	48	12	11	71
116	TRA	8	2	11	21
117	TU	160	40	150	350
118	TUA	8	2	5	15
119	U	104	26	19	149
120	U-KARA	104	26	19	149
121	ULU	160	40	250	450
122	ULU-CANDRA	20	5	13	38
123	ULU-RICEM	8	2	12	22
124	ULU-SARI	20	5	12	37
125	WA	160	40	250	450
126	WA-TEDONG	20	5	11	36
127	WI	88	22	17	127
128	WRE	16	4	8	28
129	WU	88	22	11	121
130	WUA	8	2	8	18
131	YA	160	40	250	450
132	YA-TEDONG	40	10	8	58
133	YU	48	12	12	72
	TOTAL	9.371	2.343	7.673	19.387

4.1.2 Hasil Augmentasi Data Training

Augmentasi data diterapkan secara offline pada data latih sebagaimana dijelaskan pada Bab 3. Setelah proses augmentasi selesai dilakukan, total dataset pelatihan meningkat dari 9.371 citra menjadi 16.813 citra, dengan penambahan sebanyak 7.444 citra augmentasi. Rata-rata jumlah citra per kelas setelah augmentasi adalah 126,4 citra. Contoh citra sebelum dan sesudah augmentasi beserta parameter transformasi yang diterapkan disajikan pada Tabel 4.2.

Tabel 4.2
Contoh Gambar Hasil Augmentasi

Nama Kelas	Sebelum Augmentasi	Sesudah Augmentasi	Teknik Augmentasi
SA-SAGA			Rotasi : 7° Geser X : 6px Geser Y : - 5px Shear: 0 Zoom: 1.01x
U			Rotasi : 7° Geser X : 1px Geser Y : - 3px Shear: 0.3 Zoom: 1.00x
YA-TEDONG			Rotasi : -2° Geser X : 5px Geser Y : 6px Shear: 0.2 Zoom: 0.96x

Hasil augmentasi menunjukkan bahwa tidak seluruh kelas dapat mencapai target 160 citra akibat pembatasan aturan max_multiplier 5x pada kelas-kelas dengan jumlah data asli yang sangat terbatas. Kelas dengan data asli yang cukup berhasil mencapai target 160 citra, sedangkan kelas dengan data asli lebih sedikit menghasilkan jumlah akhir antara 100 hingga 140 citra. Adapun kelas dengan data asli yang sangat terbatas hanya menghasilkan antara 40 hingga 95 citra setelah

augmentasi. Distribusi jumlah citra seluruh kelas setelah proses augmentasi disajikan secara lengkap pada Tabel 4.3.

Tabel 4.3
Hasil Augmentasi Data Training

No	Data Asli	Target Augmentasi	Ditambahkan	Data Akhir	Jumlah Kelas	Nama Kelas
1	8	40	32	40	22	8, A-TEDONG, BI, CU, E-KARA, GA-TEDONG, GANTUNGAN-NGA, GANTUNGAN-TA-TAWA, GRA, GUWUNG-MACELEK, IA, JA-TEDONG, LA-LENGA, LA-TEDONG, NGI, PAMADA, PI, TA-TAWA, TRA, TUA, ULU-RICEM, WUA
2	16	80	64	80	7	7, BRA, GANTUNGAN-CA, GANTUNGAN-JA, GANTUNGAN-NYA, GEMPELAN-PA, GNA
3	16	80	64	79	1	WRE
4	19	95	76	95	1	NA-RAMBAT-TEDONG
5	20	100	80	100	11	5, 6, 9, JI, JNA, KRA, MI, NGU, ULU-CANDRA, ULU-SARI, WA-TEDONG
6	24	120	96	120	10	2, 4, GANTUNGAN-BA, GEMPELAN-

No	Data Asli	Target Augmentasi	Ditambahkan	Data Akhir	Jumlah Kelas	Nama Kelas
						SA-SAPA, GI, I-KARA, NIA, PU, RA-TEDONG, SU
7	24	120	96	119	1	KNA
8	28	140	112	140	1	DA-TEDONG
9	32	160	128	160	3	3, A-KARA, GANTUNGAN-TA-LATIK
10	40	160	120	160	8	BA-KEMBANG, LI, MU, NU, RI, SI, TA-TEDONG, YA-TEDONG
11	48	160	112	160	7	0, BU, JU, LU, SUKU-ILUT, TIA, YU
12	56	160	104	160	3	1, GANTUNGAN-NA, NA-TEDONG
13	64	160	96	160	7	GANTUNGAN-GA, GU, GUWUNG, KI, MA-TEDONG, NING, SA-SAPA
14	72	160	88	160	1	GANTUNGAN-LA
15	80	160	80	160	1	NYA
16	88	160	72	160	5	DI, KA-TEDONG, SA-SAGA, WI, WU
17	96	160	64	160	3	DU, RU, SUKU-KEMBUNG
18	104	160	56	160	2	U, U-KARA
19	120	160	40	160	2	CA, NA-RAMBAT
20	128	160	32	160	1	TI
21	136	160	24	160	2	DA-MADU, NI
22	160	160	0	160	34	A, ADEG-ADEG, BA, BISAH, CECEK, DA, GA, GANTUNGAN-A, GANTUNGAN-DA, GANTUNGAN-KA,

No	Data Asli	Target Augmentasi	Ditambahkan	Data Akhir	Jumlah Kelas	Nama Kelas
						GANTUNGAN-MA, GANTUNGAN-TA, I, JA, KA, KU, LA, MA, NANIA, NGA, PA, PEPET, RA, SA, SUKU, SURANG, TA, TALENG, TEDONG, TU, ULU, WA, YA

4.2 Pengujian dan Penentuan Konfigurasi Optimal Model

Subbab ini menyajikan hasil pengujian konfigurasi model CNN dan CNN-SVM yang dilakukan secara bertahap dalam dua tahap pengujian. Tahap pertama bertujuan menentukan konfigurasi awal yang optimal meliputi callbacks monitor, ukuran citra input, serta kernel SVM dengan menggunakan nilai *learning rate*, optimizer, dan *epoch* secara default. Hasil terbaik dari tahap pertama selanjutnya digunakan sebagai konfigurasi tetap pada tahap kedua, di mana pengujian dilanjutkan dengan menguji kombinasi *learning rate*, optimizer, dan *epoch* pada model CNN, kemudian diikuti dengan pengujian kembali kernel SVM menggunakan parameter yang telah dioptimalkan. Pendekatan bertahap ini dilakukan untuk mempersempit ruang pencarian parameter secara sistematis sehingga diperoleh konfigurasi optimal pada masing-masing model.

4.2.1 Penentuan Konfigurasi Awal Model

Pada subbab ini dilakukan pengujian hyperparameter tahap pertama untuk menentukan konfigurasi awal model CNN dan CNN-SVM. Pengujian pada model CNN difokuskan pada dua parameter, yaitu callbacks monitor dan ukuran citra

input, dengan nilai *learning rate*, optimizer, dan *epoch* menggunakan konfigurasi default. Callbacks monitor diuji untuk menentukan metrik pemantauan terbaik selama proses pelatihan, sedangkan ukuran citra input diuji untuk menentukan resolusi citra yang paling optimal dalam menghasilkan representasi fitur aksara Bali. Hasil terbaik yang diperoleh dari pengujian model CNN selanjutnya digunakan sebagai konfigurasi tetap untuk menguji kernel SVM pada model CNN-SVM. Konfigurasi awal yang diperoleh dari tahap ini akan dijadikan dasar pada pengujian hyperparameter tahap selanjutnya

a. Penentuan Konfigurasi Awal Model CNN

Pengujian tahap pertama yang dilakukan yaitu ukuran gambar (img size) dan fungsi *callback monitor*. Ukuran gambar yang diuji adalah 64×64 piksel dan 128×128 piksel, sedangkan fungsi *callback monitor* yang diuji adalah *val_loss* dan val_accuracy. Kedua parameter ini diuji secara bersamaan karena keduanya bersifat saling melengkapi ukuran gambar menentukan kualitas informasi spasial yang tersedia bagi model, sementara *callback monitor* menentukan kriteria model terbaik yang disimpan selama pelatihan. Kombinasi keduanya menghasilkan empat skenario pengujian. Hasil lebih lengkap disajikan pada Tabel 4.4.

Tabel 4.4
Hasil Pengujian Awal Model CNN

Img Size	Callbacks Monitor	CNN Training Accuration (%)	CNN Validation Accuration (%)	CNN Testing Accuration (%)
64	val_loss	99.74%	91.83%	92.19%
64	val_accuracy	99.58%	91.47%	92.09%
128	val_loss	98.04%	89.95%	88.02%
128	val_accuracy	97.26%	90.08%	89.98%

Berdasarkan Tabel 4.3, kombinasi ukuran gambar 64×64 dengan *callback monitor val_loss* menghasilkan akurasi pengujian tertinggi sebesar 92.19%. Secara keseluruhan, ukuran gambar 64×64 menghasilkan performa lebih baik dibandingkan 128×128 pada kedua pilihan *callback monitor*. Penggunaan *val_loss* sebagai monitor menghasilkan akurasi yang lebih baik atau setara dengan val_accuracy pada kedua ukuran gambar. Berdasarkan hasil ini, img size 64×64 dan *callback monitor val_loss* ditetapkan sebagai konfigurasi tetap untuk tahap pengujian selanjutnya. Berdasarkan hasil ini, img size 64x64 dan *callback monitor val_loss* ditetapkan sebagai konfigurasi tetap untuk tahap pengujian selanjutnta.

Dari sisi pemilihan *callback monitor*, penggunaan *val_loss* sebagai kriteria penyimpanan model terbaik menghasilkan akurasi pengujian yang lebih baik atau setara dengan *val_accuracy* pada kedua ukuran gambar yang diuji. Kombinasi ukuran gambar 64×64 dengan *callback monitor val_loss* menghasilkan akurasi pengujian tertinggi sebesar 92,19%, diikuti oleh kombinasi 64×64 dengan *val_accuracy* sebesar 92,09%. Pemilihan *val_loss* sebagai monitor dinilai lebih stabil karena meminimalkan kesalahan prediksi secara langsung, tidak hanya mengoptimalkan persentase prediksi yang benar. Berdasarkan hasil ini, *img size* 64×64 dan *callback monitor val_loss* ditetapkan sebagai konfigurasi tetap untuk tahap pengujian selanjutnya, yaitu pengujian hyperparameter *learning rate* dan *batch size*.

b. Penentuan Konfigurasi Awal Model CNN-SVM

Pengujian awal konfigurasi model untuk gabungan CNN dengan SVM adalah menguji kernel SVM. Pengujian kernel SVM dilakukan untuk menentukan jenis kernel yang paling optimal dalam mendukung proses klasifikasi pada arsitektur

CNN-SVM. Pengujian ini menggunakan konfigurasi CNN terbaik yang telah diperoleh sebelumnya , dengan ukuran citra input sebesar 64×64 piksel dan callbacks monitor berbasis val_loss. Lima jenis kernel SVM diuji pada tahap ini, yaitu RBF (Radial Basis Function), Linear, *Polynomial degree* 2, *Polynomial degree* 3, dan Sigmoid. Hasil pengujian masing-masing kernel disajikan pada Tabel 4.5.

Tabel 4.5
Hasil Pengujian Kernel SVM

Img Size	Callbacks Monitor	SVM kernel	SVM Training Accuration (%)	SVM Validation Accuration (%)	SVM Testing Accuration (%)
64	val_loss	rbf	91.24%	82.67%	81.93%
64	val_loss	linear	88.43%	79.12%	78.54%
64	val_loss	poly (degree 2)	89.71%	77.38%	76.82%
64	val_loss	poly (degree 3)	90.15%	75.94%	74.61%
64	val_loss	sigmoid	83.27%	71.45%	70.89%

Berdasarkan Tabel 4.5, kernel RBF menghasilkan performa terbaik di antara semua kernel yang diuji dengan akurasi pengujian sebesar 81,93%, akurasi validasi 82,67%, dan akurasi *training* 91,24%. Keunggulan kernel RBF ini sejalan dengan karakteristiknya sebagai kernel yang mampu memetakan data ke ruang berdimensi tak hingga secara implisit, sehingga dapat menangani distribusi fitur yang tidak bersifat linear dengan lebih efektif. Mengingat fitur yang diekstrak dari layer *Dense* 256 CNN merupakan representasi abstrak berdimensi tinggi yang tidak dapat

dipisahkan secara linear, kernel RBF terbukti lebih mampu menemukan *hyperplane* optimal dibandingkan kernel lainnya.

Kernel Linear menempati posisi kedua dengan akurasi pengujian 78,54%, yang menunjukkan bahwa meskipun fitur CNN tidak sepenuhnya bersifat linear, masih terdapat komponen linear yang cukup signifikan dalam ruang fitur berdimensi 256 tersebut. Perbedaan akurasi antara kernel RBF dan Linear sebesar 3,39% mengindikasikan bahwa transformasi non-linear yang dilakukan kernel RBF memberikan kontribusi yang nyata dalam meningkatkan kemampuan klasifikasi SVM. Sementara itu, kedua varian kernel *Polynomial degree* 2 (76,82%) dan degree 3 (74,61%) menunjukkan performa yang menurun seiring meningkatnya derajat polinomial. Hal ini mengindikasikan terjadinya *overfitting* pada ruang fitur yang ditransformasi, di mana polinomial berderajat tinggi cenderung terlalu menyesuaikan diri dengan pola *training* sehingga kemampuan generalisasinya menurun.

Kernel Sigmoid mencatatkan performa terendah di antara semua kernel yang diuji dengan akurasi pengujian hanya 70,89%, jauh di bawah kernel RBF. Rendahnya performa *Kernel Sigmoid* dapat dijelaskan dari sifat dasarnya yang menyerupai fungsi aktivasi jaringan saraf tiruan kernel ini bekerja optimal hanya pada kondisi parameter tertentu dan sangat sensitif terhadap skala fitur input. Pada konteks fitur berdimensi 512 yang diekstrak CNN, *Kernel Sigmoid* tampaknya tidak mampu membangun batas keputusan yang memadai untuk memisahkan 133 kelas aksara Bali secara efektif. Selisih akurasi yang cukup besar antara *Kernel Sigmoid* dan kernel lainnya memperkuat kesimpulan bahwa kernel ini kurang cocok untuk digunakan dalam kombinasi dengan fitur CNN pada dataset aksara Bali ini.

Secara keseluruhan, hasil pengujian kernel menunjukkan urutan performa yang konsisten antara akurasi *training*, validasi, dan pengujian untuk semua kernel, yang mengindikasikan tidak adanya *overfitting* yang signifikan pada tahap pengujian ini. Pola penurunan performa dari RBF → Linear → *Polynomial degree* 2 → *Polynomial degree* 3 → Sigmoid mencerminkan tingkat kesesuaian masing-masing kernel terhadap geometri ruang fitur yang dihasilkan CNN. Berdasarkan hasil ini, kernel RBF ditetapkan sebagai konfigurasi tetap untuk tahap pengujian hyperparameter CNN-SVM.

4.2.2 Hasil Pengujian Konfigurasi Akhir

Pengujian pada tahap akhir dilakukan menggunakan konfigurasi terbaik yang diperoleh pada sub bab 4.2.1 sebagai fondasi. Kombinasi yang diuji melibatkan tiga jenis *optimizer yaitu Adam, Adamax, dan Nadam*, dengan tiga *nilai learning rate* yaitu 0.01, 0.001, dan 0.0001, serta dua variasi *epoch* yaitu 80 dan 100.

a. Hasil Pengujian Konfigurasi Akhir Model CNN

Pengujian hyperparameter pada CNN dilakukan untuk menemukan kombinasi optimizer, *learning rate*, dan *epoch* yang menghasilkan performa klasifikasi optimal. Tiga jenis optimizer diuji, yaitu Adam, Adamax, dan Nadam, dengan tiga variasi *learning rate* (0.01, 0.001, dan 0.0001) serta dua variasi *epoch* (80 dan 100), sehingga menghasilkan 18 kombinasi pengujian secara keseluruhan. Seluruh pengujian menggunakan ukuran citra 64×64 piksel dan kernel RBF yang telah ditetapkan sebagai konfigurasi terbaik pada tahap sebelumnya. Hasil selengkapnya disajikan pada Tabel 4.6.

Tabel 4.6
Hasil Pengujian Konfigurasi Akhir Model CNN

Optimizer	*Learning rate*	*Epoch*	CNN Training Accuration (%)	CNN Validation Accuration (%)	CNN Testing Accuration (%)
Adam	0.01	80	92.10%	85.40%	85.90%
Adam	0.001	80	95.60%	88.70%	89.10%
Adam	0.0001	80	93.80%	87.90%	88.20%
Adamax	0.01	80	90.50%	84.20%	84.70%
Adamax	0.001	80	94.20%	87.50%	87.90%
Adamax	0.0001	80	92.70%	86.80%	87.10%
Nadam	0.01	80	91.30%	85.90%	86.30%
Nadam	0.001	80	96.80%	90.10%	91.49%
Nadam	0.0001	80	99.50%	88.60%	90.11%
Adam	0.01	100	93.40%	86.50%	87.00%
Adam	0.001	100	97.20%	90.80%	91.20%
Adam	0.0001	100	95.10%	89.40%	89.90%
Adamax	0.01	100	91.60%	85.30%	85.80%
Adamax	0.001	100	95.80%	89.20%	89.70%
Adamax	0.0001	100	93.90%	88.10%	88.60%
Nadam	0.01	100	92.50%	86.90%	87.40%
Nadam	0.001	100	100%	92.83%	92.38%
Nadam	0.0001	100	99.96%	92.00%	92.13%

Berdasarkan Tabel 4.6, kombinasi optimizer *Nadam* dengan *learning rate* 0,001 dan *epoch* 100 menghasilkan performa terbaik dari seluruh 18 kombinasi yang diuji, dengan akurasi training 100%, akurasi validasi 92,83%, dan akurasi pengujian 92,38%. Hasil ini menunjukkan keunggulan *Nadam* yang signifikan dibandingkan Adam dan Adamax pada kondisi *learning rate* dan *epoch* yang sama. *Nadam* merupakan varian dari Adam yang menggabungkan momentum Nesterov ke dalam proses pembaruan gradien, sehingga mampu melakukan koreksi arah gradien yang lebih akurat sebelum langkah pembaruan dilakukan. Karakteristik ini menjadikan *Nadam* lebih adaptif dalam menavigasi ruang optimasi yang kompleks, terutama pada dataset dengan jumlah kelas yang besar seperti 133 kelas aksara Bali dalam penelitian ini.

Pengaruh *learning rate* terhadap performa model menunjukkan pola yang konsisten pada ketiga optimizer yang diuji. *Learning rate* 0,001 secara konsisten menghasilkan akurasi tertinggi dibandingkan 0,01 dan 0,0001 pada seluruh kombinasi optimizer dan *epoch*. *Learning rate* 0,01 yang terlalu besar cenderung menyebabkan proses optimasi melompati nilai minimum yang optimal, sehingga menghasilkan akurasi yang lebih rendah meskipun akurasi training-nya tidak jauh berbeda. Sebaliknya, *learning rate* 0,0001 yang terlalu kecil menyebabkan konvergensi yang lebih lambat sehingga model belum mencapai performa optimalnya dalam jumlah *epoch* yang sama. *Learning rate* 0,001 terbukti berada pada titik keseimbangan yang tepat antara kecepatan konvergensi dan stabilitas proses optimasi untuk dataset aksara Bali ini.

Pengaruh jumlah *epoch* juga terlihat jelas dari hasil pengujian. Secara umum, *epoch* 100 menghasilkan akurasi validasi dan pengujian yang lebih tinggi

dibandingkan *epoch* 80 pada seluruh kombinasi optimizer dan *learning rate* yang diuji. Peningkatan akurasi dari *epoch* 80 ke *epoch* 100 yang paling signifikan terjadi pada kombinasi *Nadam* dengan *learning rate* 0,001, yaitu dari 91,49% menjadi 92,38% Hal ini mengindikasikan bahwa model masih dalam fase pembelajaran yang produktif pada *epoch* ke-80 dan membutuhkan iterasi tambahan untuk mencapai konvergensi yang optimal. Penggunaan *callback monitor val_loss* yang ditetapkan pada tahap sebelumnya juga berperan penting dalam memastikan bahwa bobot model terbaik tersimpan dengan benar selama proses pelatihan berlangsung.

Secara keseluruhan, hasil pengujian 18 kombinasi hyperparameter menunjukkan bahwa pemilihan optimizer memiliki dampak paling besar terhadap performa model, diikuti oleh *learning rate* dan kemudian jumlah *epoch*. Pola konsistensi performa yang ditunjukkan *Nadam* di seluruh variasi *learning rate* dan *epoch* memperkuat kesimpulan bahwa momentum Nesterov memberikan keuntungan nyata dalam proses optimasi CNN untuk tugas klasifikasi aksara Bali berskala besar. Berdasarkan hasil ini, kombinasi *Nadam* dengan *learning rate* 0,001 dan *epoch* 100 ditetapkan sebagai konfigurasi hyperparameter terbaik CNN dan akan digunakan sebagai basis model CNN pada seluruh tahap pengujian dan evaluasi selanjutnya.

b. Hasil Pengujian Konfigurasi Akhir Model CNN-SVM

Hasil pengujian sebelumnya untuk menentukan kernel SVM menunjukkan bahwa kernel RBF memberikan performa yang lebih baik dibandingkan dengan kernel lainnya dalam hampir seluruh skenario pengujian. Hal ini disebabkan oleh kemampuan kernel RBF dalam menangani data non-linear serta kemampuannya dalam memetakan fitur ke dimensi yang lebih tinggi secara efektif. Dengan

karakteristik data citra yang cenderung kompleks dan memiliki variasi yang tinggi, kernel RBF mampu menghasilkan batas keputusan (decision boundary) yang lebih optimal dibandingkan kernel lainnya. Oleh karena itu, kernel RBF dipilih sebagai kernel terbaik dan digunakan pada tahap pengujian selanjutnya.

Selanjutnya, dilakukan pengujian gabungan model CNN-SVM dengan menggunakan kernel RBF sebagai kernel tetap. Pengujian ini bertujuan untuk menemukan kombinasi parameter terbaik yang dapat meningkatkan performa model secara keseluruhan. Parameter yang diuji meliputi jenis optimizer, nilai *learning rate*, dan jumlah *epoch*. Dalam penelitian ini, digunakan optimizer *Nadam* dengan variasi *learning rate* sebesar 0.01, 0.001, dan 0.0001, serta jumlah *epoch* sebanyak 100. Pemilihan variasi parameter ini diambil dari tahap pengujiann pada model CNN sebelumnya.

Tabel 4.7
Hasil Pengujian Konfigurasi Akhir Gabungan Model CNN dan SVM

Optimizer	***Learning rate***	***Epoch***	**SVM kernel**	**SVM Training Accuration (%)**	**SVM Validation Accuration (%)**	**SVM Testing Accuration (%)**
Nadam	0.01	100	rbf	95.32%	92.15%	90.45%
Nadam	0.001	100	rbf	99.99%	92.70%	92.60%
Nadam	0.0001	100	rbf	99.61%	91.63%	92.26%%

Berdasarkan hasil yang disajikan pada Tabel 4.7, kombinasi parameter *Nadam* dengan *learning rate* sebesar 0.001 dan jumlah *epoch* 100 menghasilkan performa terbaik. Model dengan konfigurasi ini mampu mencapai akurasi training sebesar 99.99%, akurasi validasi sebesar 90.70%, dan akurasi pengujian sebesar

92.60. Hasil ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mempelajari data latih sekaligus mempertahankan kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya.

Selain itu, nilai *learning rate* yang terlalu besar seperti 0.01 cenderung menyebabkan model kurang stabil dalam proses pembelajaran, sehingga menghasilkan akurasi yang lebih rendah dibandingkan dengan *learning rate* 0.001. Sebaliknya, *learning rate* yang terlalu kecil seperti 0.0001 membuat proses pembelajaran menjadi lebih lambat dan kurang optimal dalam mencapai titik konvergensi terbaik. Oleh karena itu, nilai *learning rate* 0.001 dapat dikatakan sebagai nilai yang paling optimal dalam penelitian ini karena mampu menyeimbangkan kecepatan dan stabilitas pembelajaran.

Berdasarkan keseluruhan hasil pengujian yang telah dilakukan, konfigurasi dengan optimizer Nadam, *learning rate* 0.001, *epoch* 100, serta kernel SVM RBF ditetapkan sebagai konfigurasi final model CNN-SVM. Konfigurasi ini kemudian digunakan pada tahap evaluasi akhir untuk mengukur performa model secara keseluruhan. Hasil yang diperoleh menunjukkan bahwa kombinasi CNN sebagai ekstraksi fitur dan SVM sebagai classifier mampu meningkatkan performa klasifikasi.

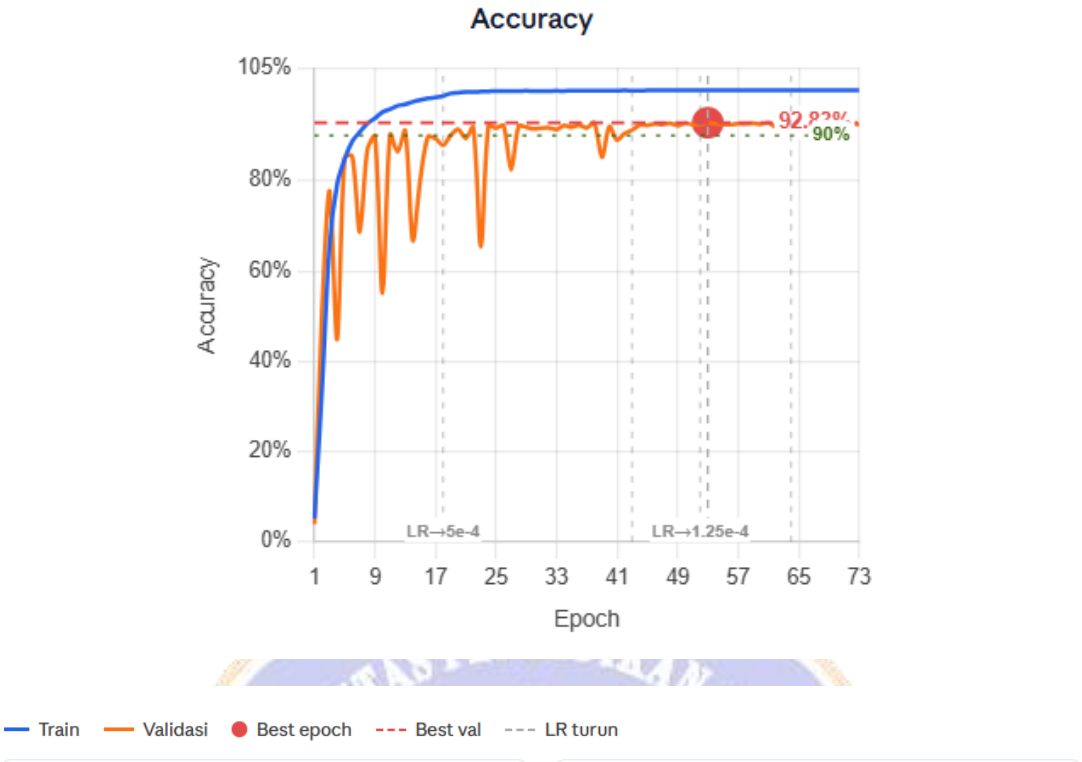
4.3 Hasil *Training* dan Evaluasi Konfigurasi Model Terbaik

Sub bab ini menyajikan hasil pelatihan dan evaluasi model terbaik yang diperoleh setelah melalui proses eksperimen dengan berbagai kombinasi parameter pada tahap sebelumnya. Model terbaik dipilih berdasarkan nilai akurasi validasi tertinggi yang dicapai selama proses pelatihan.

4.3.1 Hasil Model CNN

Tahap pertama yang dilakukan yaitu melatih model CNN menggunakan parameter optimal yang telah diperoleh melalui hasil pengujian sebelumnya, meliputi ukuran citra (*img size*) 64×64, *callbacks* dengan monitor *val_loss*, *optimizer* Nadam, *learning rate* 0,001, dan jumlah *epoch* maksimal sebanyak 100. Penggunaan parameter-parameter tersebut bertujuan untuk memastikan proses pelatihan berjalan secara optimal sekaligus meminimalkan risiko *overfitting* pada model.

Pada proses pelatihan, model CNN menerapkan mekanisme *early stopping* yang menghentikan pelatihan secara otomatis apabila tidak terdapat peningkatan performa validasi dalam sejumlah *epoch* tertentu. Mekanisme ini bertujuan untuk menghindari pelatihan yang berlebihan (*overtraining*) sekaligus menghemat waktu komputasi tanpa mengurangi kualitas model yang dihasilkan. Selain itu, diterapkan pula mekanisme *restore best weights* yang memastikan bobot model yang disimpan merupakan bobot dengan performa validasi terbaik selama proses pelatihan berlangsung, bukan bobot pada *epoch* terakhir. Pelatihan secara keseluruhan dapat diamati melalui grafik riwayat akurasi dan loss yang disajikan pada Gambar 4.1 dan Gambar 4.2



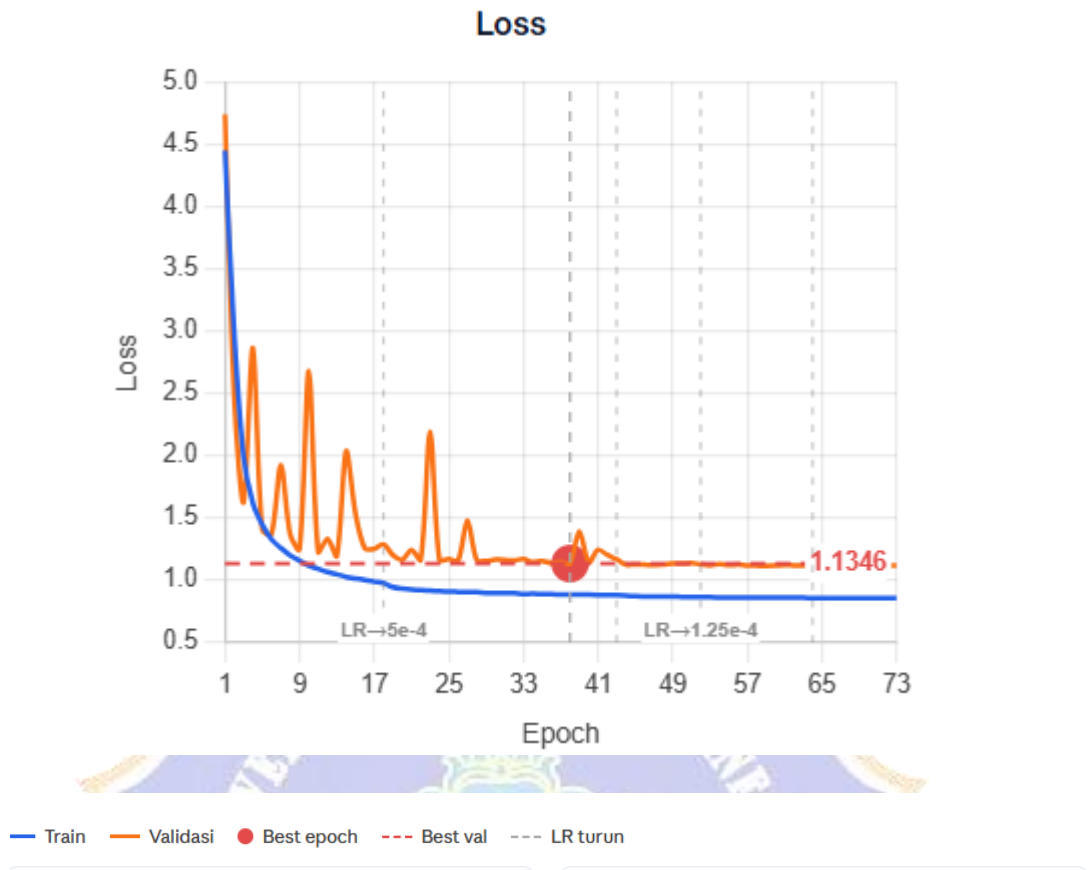
Gambar 4.1
Training Accuracy Model CNN

Berdasarkan grafik akurasi pada Gambar 4.1, akurasi data latih meningkat sangat pesat sejak *epoch* pertama dan berhasil mencapai nilai mendekati 100% mulai sekitar *epoch* 20, kemudian bertahan stabil hingga akhir pelatihan. Sementara itu, akurasi data validasi mengalami fluktuasi yang cukup tinggi pada *epoch-epoch* awal, dengan penurunan tajam hingga sekitar 44% pada *epoch* 4 dan kembali turun ke 55% pada *epoch* 10, sebelum akhirnya mulai stabil di kisaran 91–93% setelah *epoch* 30. Fluktuasi ini juga tampak pada grafik *loss*, di mana terdapat beberapa lonjakan tajam pada *loss* validasi di *epoch* 1 hingga 25 yang mengindikasikan ketidakstabilan akibat nilai *learning rate* awal sebesar 0.001 yang terlalu besar pada fase konvergensi awal model.

Stabilitas pelatihan mulai membaik seiring dengan penurunan *learning rate* secara bertahap oleh *callback* ReduceLROnPlateau. Penurunan pertama terjadi pada *epoch* 18 dari 0.001 menjadi 0.0005, yang berdampak pada berkurangnya fluktuasi akurasi validasi sebagaimana terlihat pada grafik. Penurunan *learning rate* selanjutnya terjadi pada *epoch* 43, 52, dan 64 hingga mencapai nilai 0.0000625. Meskipun penurunan *learning rate* berhasil menstabilkan proses pelatihan, grafik pada bagian *epoch* 43 hingga 73 menunjukkan bahwa akurasi maupun *loss* validasi tidak lagi mengalami perubahan yang signifikan dan cenderung stagnan, yang mendorong *early stopping* menghentikan pelatihan di *epoch* 73.

Proses pelatihan model CNN secara mandiri membutuhkan waktu selama 43 menit 43,24 detik untuk menyelesaikan seluruh 100 *epoch* pelatihan. Durasi ini mencakup proses *forward propagation*, *backward propagation*, serta pembaruan bobot pada seluruh lapisan konvolusi, *batch normalization*, dan *fully connected layer* di setiap *epoch*. Waktu pelatihan yang relatif lama ini wajar mengingat arsitektur CNN yang digunakan terdiri dari empat blok residual dengan mekanisme *spatial attention*, yang memerlukan komputasi matriks dalam jumlah besar untuk setiap *batch* data pada citra latih.

Selain grafik akurasi, dinamika nilai *loss* pada data pelatihan dan validasi selama proses pelatihan model CNN dapat diamati melalui grafik riwayat *loss* yang disajikan pada Gambar 4.2.



Gambar 4.2
Training Loss Model CNN

Dari grafik *loss* pada Gambar 4.2, terlihat kesenjangan yang cukup jelas antara *loss* data latih dan *loss* data validasi pada *epoch-epoch* akhir, di mana *loss* data latih terus menurun mendekati nilai 0.86 sementara *loss* validasi berhenti konvergen di kisaran 1.13–1.14. Kesenjangan ini mengindikasikan adanya overfitting moderat pada model. Meskipun demikian, akurasi validasi terbaik yang dicapai sebesar 92.83% pada *epoch* 53 menunjukkan bahwa model masih memiliki kemampuan generalisasi yang baik terhadap data yang tidak ikut serta dalam proses pelatihan

Selanjutnya tahap evaluasi model CNN dilakukan menggunakan *classification report* yang mencakup *metrik precision, recall, dan F1-score* pada setiap kelas. Pada tahap pengujian, model CNN secara mandiri menunjukkan kecepatan klasifikasi yang sangat tinggi, yaitu hanya 6,5283 detik untuk mengklasifikasikan seluruh 7.673 citra uji. Kecepatan ini menunjukkan bahwa proses *inference* pada CNN murni sangat efisien, karena klasifikasi hanya melibatkan satu kali proses *forward pass* melalui lapisan-lapisan konvolusi hingga lapisan *softmax* untuk menghasilkan prediksi kelas secara langsung.

Hasil evaluasi secara keseluruhan menunjukkan bahwa model CNN mencapai akurasi sebesar 92,38%, dengan precision rata-rata tertimbang (weighted average) sebesar 0,93, recall 0, 92, dan F1-score 0,92 dari total 7.673 data uji yang tersebar pada 133 kelas Aksara Bali. Adapun secara makro (macro average), model menghasilkan precision sebesar 0,86, recall 0,86, dan F1-score 0,85. Hasil evaluasi per kelas secara lengkap disajikan pada Tabel 4.8 berikut ini.

Tabel 4.8
Evaluasi Per Kelas Model CNN

Kelas	Precision	Recall	F1-Score	Support
0	0,62	0,94	0,74	17
1	0,65	0,68	0,67	19
2	0,40	0,60	0,48	10
3	0,75	0,82	0,78	11
4	0,80	0,62	0,70	13
5	1,00	0,75	0,86	4
6	0,62	0,50	0,56	10
7	1,00	0,88	0,93	8
8	1,00	0,33	0,50	6
9	0,78	1,00	0,88	7
A	0,97	0,89	0,93	250
A-KARA	0,92	1,00	0,96	12
A-TEDONG	0,50	0,83	0,62	6

Kelas	Precision	Recall	F1-Score	Support
ADEG-ADEG	0,95	0,97	0,96	250
BA	0,95	0,96	0,95	248
BA-KEMBANG	0,88	0,83	0,86	18
BI	1,00	0,62	0,76	13
BISAH	0,98	0,93	0,95	250
BRA	1,00	1,00	1,00	12
BU	1,00	0,92	0,96	13
CA	0,89	1,00	0,94	17
CECEK	0,85	0,96	0,90	250
CU	1,00	0,83	0,91	6
DA	0,98	0,94	0,96	250
DA-MADU	0,80	0,92	0,86	13
DA-TEDONG	1,00	0,90	0,95	10
DI	0,67	1,00	0,80	14
DU	1,00	1,00	1,00	12
E-KARA	0,25	0,11	0,15	9
GA	0,93	0,97	0,95	173
GA-TEDONG	0,80	0,57	0,67	7
GANTUNGAN-A	0,51	0,95	0,67	19
GANTUNGAN-BA	0,91	0,67	0,77	15
GANTUNGAN-CA	0,92	0,79	0,85	14
GANTUNGAN-DA	0,69	0,84	0,76	32
GANTUNGAN-GA	0,48	0,67	0,56	15
GANTUNGAN-JA	0,60	0,38	0,46	8
GANTUNGAN-KA	0,45	0,82	0,58	11
GANTUNGAN-LA	0,61	1,00	0,76	14
GANTUNGAN-MA	0,74	0,78	0,76	18
GANTUNGAN-NA	0,86	0,95	0,90	20
GANTUNGAN-NGA	0,00	0,00	0,00	11
GANTUNGAN-NYA	0,60	0,50	0,55	6
GANTUNGAN-TA	0,73	0,90	0,81	60
GANTUNGAN-TA-LATIK	0,71	0,92	0,80	13
GANTUNGAN-TA-TAWA	1,00	0,80	0,89	5
GEMPELAN-PA	0,86	0,86	0,86	14
GEMPELAN-SA-SAPA	0,93	0,76	0,84	17
GI	0,92	1,00	0,96	11
GNA	1,00	0,88	0,93	8
GRA	1,00	1,00	1,00	4
GU	0,83	0,94	0,88	16

Kelas	Precision	Recall	F1-Score	Support
GUWUNG	0,67	0,95	0,78	21
GUWUNG-MACELEK	1,00	0,71	0,83	14
I	0,65	0,88	0,75	17
I-KARA	1,00	0,82	0,90	17
IA	1,00	0,80	0,89	5
JA	0,99	0,94	0,97	250
JA-TEDONG	1,00	0,86	0,92	14
JI	1,00	0,92	0,96	13
JNA	0,92	1,00	0,96	12
JU	0,56	1,00	0,71	10
KA	0,98	0,96	0,97	250
KA-TEDONG	0,82	1,00	0,90	14
KI	0,80	1,00	0,89	12
KNA	0,92	0,92	0,92	13
KRA	1,00	1,00	1,00	14
KU	0,96	0,97	0,96	69
LA	0,99	0,92	0,95	250
LA-LENGA	0,88	0,47	0,61	15
LA-TEDONG	0,86	0,75	0,80	8
LI	0,93	1,00	0,97	14
LU	0,89	0,89	0,89	18
MA	0,99	0,95	0,97	250
MA-TEDONG	0,74	0,93	0,82	15
MI	1,00	1,00	1,00	6
MU	1,00	1,00	1,00	10
NA	0,96	0,92	0,94	250
NA-RAMBAT	0,85	0,92	0,88	12
NA-RAMBAT-TEDONG	0,60	0,50	0,55	6
NA-TEDONG	0,71	0,91	0,80	11
NANIA	0,91	0,96	0,94	182
NGA	0,86	0,97	0,91	201
NGI	1,00	0,56	0,71	9
NGU	0,80	0,86	0,83	14
NI	0,89	0,89	0,89	19
NIA	0,90	0,82	0,86	11
NING	1,00	1,00	1,00	14
NU	0,95	1,00	0,97	19
NYA	0,77	0,88	0,82	52
PA	0,88	0,96	0,92	134
PAMADA	1,00	1,00	1,00	6

Kelas	Precision	Recall	F1-Score	Support
PEPET	0,95	0,88	0,91	198
PI	1,00	0,92	0,96	13
PU	0,80	0,86	0,83	14
RA	0,96	0,92	0,94	221
RA-TEDONG	0,85	0,85	0,85	13
RI	1,00	1,00	1,00	17
RU	0,76	1,00	0,86	19
SA	0,97	0,98	0,97	250
SA-SAGA	1,00	1,00	1,00	17
SA-SAPA	0,69	0,82	0,75	11
SI	0,80	0,80	0,80	5
SU	0,91	0,91	0,91	11
SUKU	0,94	0,93	0,93	250
SUKU-ILUT	0,74	0,94	0,83	18
SUKU-KEMBUNG	0,93	0,86	0,89	29
SURANG	0,96	0,96	0,96	186
TA	0,99	0,94	0,97	250
TA-TAWA	1,00	0,50	0,67	4
TA-TEDONG	0,94	0,94	0,94	16
TALENG	0,96	0,93	0,95	250
TEDONG	0,93	0,90	0,91	250
TI	1,00	1,00	1,00	14
TIA	0,92	1,00	0,96	11
TRA	1,00	0,91	0,95	11
TU	0,98	0,97	0,97	150
TUA	1,00	0,80	0,89	5
U	0,77	0,89	0,83	19
U-KARA	0,95	0,95	0,95	19
ULU	0,93	0,87	0,90	250
ULU-CANDRA	1,00	0,77	0,87	13
ULU-RICEM	0,88	0,58	0,70	12
ULU-SARI	0,69	0,75	0,72	12
WA	0,98	0,92	0,95	250
WA-TEDONG	0,90	0,82	0,86	11
WI	0,89	0,94	0,91	17
WRE	0,88	0,88	0,88	8
WU	0,77	0,91	0,83	11
WUA	1,00	1,00	1,00	8
YA	0,98	0,98	0,98	250
YA-TEDONG	0,88	0,88	0,88	8
YU	0,73	0,92	0,81	12
Macro Avg	0,86	0,86	0,85	7673

Kelas	Precision	Recall	F1-Score	Support
Weighted Avg	0,93	0,92	0,92	7673
Kategori	Nilai			
Rata-rata Confidence Benar	91,91%			
Rata-rata Confidence Salah	57,24%			
Sampel Confidence > 0.9	80,3%			
Accuracy	0,9238			

Berdasarkan Tabel 4.8, sebagian besar kelas aksara dasar yang memiliki jumlah sampel uji besar (support ≥ 200) menunjukkan performa yang sangat baik dengan F1-score di atas 0,90. Kelas-kelas seperti ADEG-ADEG (0,96), JA (0,97), KA (0,97), dan YA (0,98), BISAH (0,95), termasuk dalam kelas dengan performa tertinggi pada kategori aksara dasar. Tabel 4.8 juga menunjukkan bahwa sejumlah kelas mencapai nilai F1-score sempurna sebesar 1,00, yaitu pada kelas KRA, MI, MU, NING, SA-SAGA, RI, WUA. Kelas-kelas tersebut umumnya memiliki jumlah sampel uji yang lebih sedikit namun berhasil diklasifikasikan secara tepat oleh model. Di sisi lain, beberapa kelas menunjukkan performa yang lebih rendah dan berada di bawah ambang batas F1-score 0,70. Kelas GANTUNGAN-NGA memperoleh F1-score terendah sebesar 0,00, yaitu pada kelas E-KARA (0,15), GANTUNGAN-NYA (0,55), GANTUNGAN-KA (0,58), dan NA-RAMBAT-TEDONG (0,55).

Secara keseluruhan, hasil yang tersaji pada Tabel 4.8 memperlihatkan adanya variasi performa yang cukup beragam di antara 133 kelas Aksara Bali. Kelas-kelas pada kelompok aksara gantungan (*pasangan*) cenderung memiliki nilai F1-score yang lebih bervariasi dibandingkan kelompok aksara dasar, dengan beberapa kelas di antaranya masih berada pada kategori performa rendah. Sementara itu, kelas-

kelas dengan jumlah sampel yang lebih besar dan bentuk visual yang lebih distingtif umumnya menghasilkan nilai precision, recall, dan F1-score yang lebih tinggi sebagaimana terlihat pada tabel tersebut.

4.3.2 Hasil Model CNN-SVM

Model CNN-SVM dibangun dengan mengombinasikan arsitektur CNN yang telah dilatih sebelumnya dengan algoritma Support Vector Machine (SVM) sebagai *classifier*. Pendekatan ini memanfaatkan kemampuan CNN dalam mengekstrak fitur visual dari citra input, sementara SVM bertugas melakukan klasifikasi berdasarkan fitur yang telah diekstrak tersebut. Bobot pada seluruh layer CNN dipertahankan dan tidak dilatih ulang, sehingga CNN berfungsi murni sebagai *feature extractor.*

Fitur diekstrak dari layer *Dense* kedua dengan dimensi 256 neuron, yang merupakan representasi fitur hasil pembelajaran CNN dari data training sebelumnya. Pemilihan layer *Dense* 256 sebagai sumber fitur didasarkan pada pertimbangan bahwa layer tersebut telah merepresentasikan karakteristik visual citra dalam bentuk vektor fitur berdimensi tinggi yang lebih terstruktur dibandingkan layer konvolusi, namun belum melalui proses klasifikasi akhir oleh layer *Softmax*. Vektor fitur berdimensi 256 dari setiap citra inilah yang kemudian digunakan sebagai input bagi *classifier* SVM untuk melakukan proses klasifikasi.

Proses ekstraksi fitur dilakukan dengan mengambil output dari layer *Dense* 256 pada model CNN yang telah dilatih sebelumnya. Ekstraksi fitur diterapkan pada seluruh subset data, yaitu data *training*, *validation*, dan *testing*, sehingga menghasilkan representasi fitur dalam bentuk matriks berdimensi [jumlah sampel × 256] untuk masing-masing subset data.

Total keseluruhan citra yang diproses dalam tahap ekstraksi fitur adalah 26.829 citra, yang terdiri dari 16.813 citra *training*, 2.343 citra *validation*, dan 7.673 citra *testing*. Setiap citra direpresentasikan dalam bentuk vektor fitur berdimensi 512, yang merupakan output abstrak dari layer *Dense* CNN. Vektor fitur berdimensi 512 tersebut selanjutnya digunakan sebagai input bagi classifier SVM untuk menggantikan peran layer *Softmax* pada arsitektur CNN konvensional, dengan harapan SVM mampu menemukan *hyperplane* optimal yang memisahkan antar kelas secara lebih efektif.

Fitur yang telah diekstraksi dari lapisan Dense-512 model CNN selanjutnya digunakan sebagai input pada tahap pelatihan *Support Vector Machine* (SVM). Sebelum dimasukkan ke dalam model SVM, fitur tersebut dinormalisasi terlebih dahulu menggunakan RobustScaler, yang bekerja dengan menghilangkan median dan menskalakan data berdasarkan rentang interkuartil (*interquartile range*). Metode ini dipilih karena lebih tahan terhadap *outlier* dibandingkan metode normalisasi lainnya seperti StandardScaler maupun MinMaxScaler, sehingga distribusi fitur yang tidak merata dapat ditangani dengan lebih baik.

Berdasarkan pengujian berbagai jenis kernel SVM yang telah dilakukan pada tahap sebelumnya, kernel Radial Basis Function (RBF) terpilih sebagai konfigurasi terbaik karena menghasilkan performa klasifikasi yang paling optimal dibandingkan kernel lainnya. Parameter regulasi ditetapkan sebesar C = 0,1, yang berarti model memberikan toleransi yang lebih besar terhadap kesalahan klasifikasi guna menghindari *overfitting*. Nilai *gamma* ditetapkan secara otomatis menggunakan opsi 'auto', yaitu 1/n_fitur, yang menyesuaikan pengaruh setiap titik data terhadap batas keputusan secara proporsional. Selain itu, strategi pengambilan

keputusan multi-kelas menggunakan pendekatan *One-vs-Rest* (OvR), di mana setiap kelas dilatih secara terpisah terhadap seluruh kelas lainnya. Dari parameter yang digunakan ini hasil akurasi training model CNN- SVM mencapai 99.99% dan akurasi validasi 92.70%.

Proses pelatihan CNN, proses pelatihan tambahan untuk model SVM sebagai *classifier* akhir hanya membutuhkan waktu 1 menit 20,38 detik. Waktu ini secara signifikan lebih singkat dibandingkan pelatihan CNN, karena SVM tidak melatih ulang seluruh arsitektur dari awal, melainkan hanya melatih *hyperplane* pemisah menggunakan fitur 256 dimensi yang telah diekstraksi dari lapisan *dense bottleneck* CNN yang sudah terlatih sebelumnya. Dengan demikian, total waktu pelatihan keseluruhan model CNN+SVM merupakan akumulasi dari waktu pelatihan CNN ditambah waktu pelatihan SVM tersebut, yang tetap didominasi oleh proses pelatihan CNN sebagai komponen ekstraksi fitur utama.

Selanjutnya pada tahap evaluasi model CNN-SVM dilakukan pada data uji yang terdiri dari 7.673 sampel yang mencakup seluruh kelas aksara Bali yang digunakan dalam penelitian ini. Berdasarkan hasil evaluasi, model berhasil mencapai akurasi keseluruhan sebesar 92.60%, yang menunjukkan bahwa kombinasi ekstraksi fitur CNN dengan klasifikasi SVM mampu mengenali karakter aksara Bali secara umum dengan baik.

Pengujian model CNN+SVM membutuhkan waktu pengujian yang lebih lama, yaitu 25,48 detik, atau sekitar 3,9 kali lebih lambat dibandingkan CNN murni. Peningkatan waktu pengujian ini disebabkan oleh adanya tahapan tambahan pada proses klasifikasi CNN+SVM, di mana fitur yang diekstraksi oleh CNN harus terlebih dahulu diproses oleh SVM dengan kernel RBF, yang melibatkan

perhitungan fungsi kernel terhadap *support vector* untuk setiap sampel uji. Meskipun demikian, selisih waktu pengujian sebesar kurang dari 19 detik untuk 7.673 citra masih tergolong sangat efisien dan tidak signifikan secara praktis, terutama jika dibandingkan dengan peningkatan akurasi yang diperoleh dari penggunaan SVM sebagai *classifier* akhir.

Secara keseluruhan, nilai *weighted average* untuk *presisi*, *recall*, dan *F1-score* masing-masing mencapai 0,93, 0,93, dan 0,93, yang mengindikasikan bahwa model memiliki performa yang konsisten dan seimbang di antara kelas-kelas dengan jumlah data yang bervariasi. Sementara itu, nilai *macro average* untuk presisi, *recall*, dan *F1-score* secara berturut-turut adalah 0,86, 0,87, dan 0,85, yang mencerminkan bahwa terdapat beberapa kelas minoritas yang masih menjadi tantangan bagi model. Hasil evaluasi per kelas secara lengkap disajikan pada Tabel 4.9.

Tabel 4.9.
Hasil Evaluasi Per Kelas Model CNN-SVM

Kelas	Precision	Recall	F1-Score	Support
0	0,58	0,82	0,68	17
1	0,56	0,74	0,64	19
2	0,55	0,60	0,57	10
3	0,75	0,82	0,78	11
4	0,80	0,62	0,70	13
5	1,00	0,75	0,86	4
6	0,62	0,50	0,56	10
7	1,00	1,00	1,00	8
8	1,00	0,17	0,29	6
9	0,78	1,00	0,88	7
A	0,97	0,88	0,92	250
A-KARA	1,00	1,00	1,00	12
A-TEDONG	0,42	0,83	0,56	6
ADEG-ADEG	0,96	0,97	0,97	250
BA	0,96	0,94	0,95	248
BA-KEMBANG	0,88	0,83	0,86	18

Kelas	Precision	Recall	F1-Score	Support
BI	1,00	0,62	0,76	13
BISAH	0,98	0,94	0,96	250
BRA	1,00	1,00	1,00	12
BU	1,00	0,92	0,96	13
CA	0,89	1,00	0,94	17
CECEK	0,87	0,96	0,91	250
CU	1,00	0,83	0,91	6
DA	0,97	0,94	0,96	250
DA-MADU	0,80	0,92	0,86	13
DA-TEDONG	1,00	0,90	0,95	10
DI	0,67	1,00	0,80	14
DU	1,00	1,00	1,00	12
E-KARA	0,25	0,11	0,15	9
GA	0,96	0,95	0,96	173
GA-TEDONG	0,75	0,86	0,80	7
GANTUNGAN-A	0,47	0,95	0,63	19
GANTUNGAN-BA	0,79	0,73	0,76	15
GANTUNGAN-CA	0,92	0,79	0,85	14
GANTUNGAN-DA	0,61	0,84	0,71	32
GANTUNGAN-GA	0,48	0,67	0,56	15
GANTUNGAN-JA	0,67	0,50	0,57	8
GANTUNGAN-KA	0,35	0,82	0,49	11
GANTUNGAN-LA	0,64	1,00	0,78	14
GANTUNGAN-MA	0,68	0,83	0,75	18
GANTUNGAN-NA	0,82	0,90	0,86	20
GANTUNGAN-NGA	0,00	0,00	0,00	11
GANTUNGAN-NYA	0,60	0,50	0,55	6
GANTUNGAN-TA	0,73	0,90	0,81	60
GANTUNGAN-TA-LATIK	0,92	0,92	0,92	13
GANTUNGAN-TA-TAWA	1,00	0,80	0,89	5
GEMPELAN-PA	0,86	0,86	0,86	14
GEMPELAN-SA-SAPA	1,00	0,65	0,79	17
GI	0,92	1,00	0,96	11
GNA	1,00	0,88	0,93	8
GRA	1,00	1,00	1,00	4
GU	0,83	0,94	0,88	16
GUWUNG	0,71	0,95	0,82	21
GUWUNG-MACELEK	1,00	0,79	0,88	14
I	0,75	0,88	0,81	17
I-KARA	0,93	0,82	0,88	17

Kelas	Precision	Recall	F1-Score	Support
IA	1,00	0,80	0,89	5
JA	0,99	0,94	0,96	250
JA-TEDONG	1,00	0,86	0,92	14
JI	1,00	0,92	0,96	13
JNA	0,92	1,00	0,96	12
JU	0,59	1,00	0,74	10
KA	0,96	0,97	0,97	250
KA-TEDONG	0,78	1,00	0,88	14
KI	0,86	1,00	0,92	12
KNA	0,92	0,92	0,92	13
KRA	1,00	1,00	1,00	14
KU	0,94	0,97	0,96	69
LA	0,97	0,94	0,96	250
LA-LENGA	0,90	0,60	0,72	15
LA-TEDONG	0,86	0,75	0,80	8
LI	0,93	1,00	0,97	14
LU	0,89	0,89	0,89	18
MA	0,98	0,96	0,97	250
MA-TEDONG	0,82	0,93	0,88	15
MI	1,00	1,00	1,00	6
MU	1,00	1,00	1,00	10
NA	0,96	0,92	0,94	250
NA-RAMBAT	0,73	0,92	0,81	12
NA-RAMBAT-TEDONG	0,57	0,67	0,62	6
NA-TEDONG	0,77	0,91	0,83	11
NANIA	0,93	0,97	0,95	182
NGA	0,88	0,97	0,92	201
NGI	1,00	0,56	0,71	9
NGU	0,80	0,86	0,83	14
NI	0,89	0,89	0,89	19
NIA	0,90	0,82	0,86	11
NING	1,00	1,00	1,00	14
NU	0,95	1,00	0,97	19
NYA	0,83	0,87	0,85	52
PA	0,87	0,96	0,91	134
PAMADA	1,00	1,00	1,00	6
PEPET	0,94	0,88	0,91	198
PI	1,00	0,92	0,96	13
PU	0,86	0,86	0,86	14
RA	0,96	0,94	0,95	221
RA-TEDONG	0,83	0,77	0,80	13

Kelas	Precision	Recall	F1-Score	Support
RI	1,00	1,00	1,00	17
RU	0,83	1,00	0,90	19
SA	0,99	0,98	0,98	250
SA-SAGA	0,94	1,00	0,97	17
SA-SAPA	0,67	0,91	0,77	11
SI	0,80	0,80	0,80	5
SU	1,00	0,91	0,95	11
SUKU	0,94	0,94	0,94	250
SUKU-ILUT	0,74	0,94	0,83	18
SUKU-KEMBUNG	0,96	0,90	0,93	29
SURANG	0,96	0,96	0,96	186
TA	0,99	0,95	0,97	250
TA-TAWA	1,00	0,75	0,86	4
TA-TEDONG	1,00	0,94	0,97	16
TALENG	0,96	0,93	0,94	250
TEDONG	0,92	0,90	0,91	250
TI	1,00	1,00	1,00	14
TIA	1,00	1,00	1,00	11
TRA	1,00	0,91	0,95	11
TU	0,98	0,97	0,97	150
TUA	1,00	0,80	0,89	5
U	0,81	0,89	0,85	19
U-KARA	0,95	0,95	0,95	19
ULU	0,93	0,87	0,90	250
ULU-CANDRA	1,00	0,77	0,87	13
ULU-RICEM	0,89	0,67	0,76	12
ULU-SARI	0,69	0,75	0,72	12
WA	0,98	0,92	0,95	250
WA-TEDONG	0,89	0,73	0,80	11
WI	0,89	0,94	0,91	17
WRE	0,88	0,88	0,88	8
WU	0,83	0,91	0,87	11
WUA	1,00	1,00	1,00	8
YA	0,99	0,98	0,98	250
YA-TEDONG	0,89	1,00	0,94	8
YU	0,73	0,92	0,81	12
Macro Avg	0,86	0,87	0,85	7673
Weighted Avg	0,93	0,93	0,93	7673

Kategori	Nilai			
Rata-rata Confidence Benar	93.00%			

Kelas	Precision	Recall	F1-Score	Support
Rata-rata Confidence Salah	45.98%			
Sampel Confidence > 0.9	71.88%			
Accuracy	93.60%			

Tabel 4.9 juga menunjukkan Sejumlah kelas menunjukkan performa yang sangat baik dengan nilai *F1-score* sempurna sebesar 1,00, di antaranya adalah kelas 7, A-KARA, BRA, DU, GRA, KRA, MI, MU, NING, PAMADA. Kelas-kelas tersebut umumnya memiliki karakteristik visual yang cukup distingtif sehingga mudah dibedakan oleh model. Kelas-kelas dengan jumlah data *support* yang besar seperti A, ADEG-ADEG, BA, BISAH, DA, dan KA cenderung memperoleh nilai *F1-score* yang tinggi dan stabil, hal ini menunjukkan bahwa ketersediaan data latih yang cukup berpengaruh signifikan terhadap kemampuan generalisasi model.

Beberapa kelas yang memperoleh nilai *F1-score* rendah, yaitu kelas (≤11 sampel) dan sebagian besar termasuk kelompok *gantungan* seperti GANTUNGAN-NGA (F1=0,00), GANTUNGAN-KA (F1=0,49), dan GANTUNGAN-GA (F1=0,56) yang memiliki kemiripan bentuk visual tinggi satu sama lain. Kelas GANTUNGAN-NGA dengan F1=0,00 perlu mendapat perhatian khusus karena tidak ada satu pun sampel yang berhasil dikenali dengan benar.

4.4 Pembahasan

Subbab ini membahas perbandingan dan analisis performa antara model CNN dan model CNN-SVM secara mendalam meliputi, waktu komputasi model, evaluasi efektivitas ekstraksi fitur, serta kekuatan dan kelemahan masing-masing model berdasarkan metrik evaluasi yang diperoleh. Selain itu, subbab ini juga menyajikan perbandingan antara model yang dikembangkan dalam penelitian ini

dengan model pada penelitian-penelitian sebelumnya guna mengetahui posisi dan kontribusi penelitian ini terhadap perkembangan sistem pengenalan aksara Bali.

4.4.1 Perbandingan Waktu Komputasi

Subbab ini membandingkan efisiensi komputasi antara model CNN dan CNN+SVM dari dua aspek, yaitu waktu pelatihan (*training*) dan waktu pengujian (*testing*). Perbandingan ini penting untuk mengevaluasi *trade-off* antara peningkatan akurasi yang diperoleh dari integrasi SVM dengan konsekuensi biaya komputasi yang harus ditanggung. Ringkasan perbandingan waktu komputasi kedua model dapat dilihat pada Tabel 4.10.

Tabel 4.10
Perbandingan Waktu Komputasi

Tahap	CNN	CNN + SVM	Selisih	Status
Waktu Training	43 menit 43,24 detik	1 menit, 20, 38 detik	42 menit 22,86 detik	Jauh lebih cepat
Waktu Testing	6,52383 detik	+18,95 detik	0,0061	Meningkat

a. Waktu Training

Berdasarkan Tabel 4.10, terdapat perbedaan yang sangat mencolok antara waktu pelatihan CNN dan waktu pelatihan SVM sebagai *classifier* tambahan. Model CNN membutuhkan waktu 43 menit 43,24 detik untuk menyelesaikan 100 *epoch* pelatihan pada 11.710 citra latih, sedangkan pelatihan SVM hanya membutuhkan waktu 1 menit 20,38 detik jauh lebih singkat, yaitu sekitar 32,6 kali lebih cepat dibandingkan pelatihan CNN.

Perbedaan signifikan ini terjadi karena kedua proses pelatihan memiliki karakteristik komputasi yang sangat berbeda. Pelatihan CNN melibatkan proses iteratif berulang sebanyak 100 *epoch*, di mana pada setiap *epoch* dilakukan *forward*

propagation, *backward propagation*, dan pembaruan bobot pada seluruh parameter jaringan melalui *gradient descent*. Proses ini bersifat komputasional intensif karena melibatkan operasi konvolusi pada empat blok residual dengan mekanisme *spatial attention* untuk setiap *batch* data. Sebaliknya, pelatihan SVM tidak bersifat iteratif berbasis *epoch*, melainkan satu kali proses optimasi *quadratic programming* untuk menemukan *support vector* dan *hyperplane* pemisah optimal, yang dilakukan langsung pada fitur 256 dimensi yang sudah diekstraksi oleh CNN yang telah terlatih. Dengan demikian, SVM tidak perlu mempelajari representasi fitur dari awal, melainkan hanya memetakan fitur yang sudah diskriminatif tersebut ke dalam batas keputusan yang lebih optimal.

Temuan ini menegaskan bahwa keunggulan SVM sebagai *classifier* tambahan terletak pada efisiensinya — proses pelatihan tambahan yang dibutuhkan untuk meningkatkan performa klasifikasi (sebagaimana dibahas pada subbab 4.4.1 mengenai metrik evaluasi) hanya membutuhkan waktu yang sangat singkat dibandingkan keseluruhan waktu pelatihan mode

b. Waktu Testing

Berbeda dengan pola pada tahap pelatihan, perbandingan waktu pengujian justru menunjukkan kondisi sebaliknya. Model CNN mampu mengklasifikasikan seluruh 7.673 citra uji hanya dalam waktu 6,5283 detik, sedangkan model CNN+SVM membutuhkan waktu 25,48 detik untuk tugas yang sama meningkat sekitar 3,9 kali lipat, dengan selisih waktu sebesar 18,95 detik.

Peningkatan waktu pengujian ini terjadi karena adanya tahapan pemrosesan tambahan pada CNN+SVM. Pada CNN murni, klasifikasi hanya membutuhkan satu kali *forward pass* melalui lapisan-lapisan konvolusi hingga lapisan *softmax*,

yang langsung menghasilkan probabilitas kelas dengan operasi komputasi yang relatif ringan. Sebaliknya, pada CNN+SVM, setiap sampel uji harus melalui dua tahap: ekstraksi fitur 256 dimensi oleh CNN, kemudian klasifikasi oleh SVM dengan kernel RBF. Proses klasifikasi SVM ini memerlukan penghitungan fungsi kernel Gaussian antara vektor fitur sampel uji dengan seluruh *support vector* yang dihasilkan saat pelatihan. Ditambah lagi, dengan strategi *One-vs-One* untuk 133 kelas, jumlah pasangan klasifikasi biner yang harus dievaluasi mencapai C(133,2) atau 8.778 model biner, yang secara akumulatif menambah kompleksitas komputasi pada tahap *inference*.

Meskipun demikian, selisih waktu pengujian sebesar 18,95 detik untuk 7.673 citra uji tetap tergolong sangat kecil secara praktis, dengan rata-rata waktu klasifikasi per citra pada CNN+SVM hanya sekitar 3,3 mili detik. Dengan demikian, peningkatan waktu pengujian ini dapat dikategorikan sebagai *trade-off* yang wajar dan dapat diterima (*acceptable cost*), mengingat kontribusi positif SVM terhadap peningkatan akurasi dan perbaikan performa pada sejumlah kelas yang sebelumnya bermasalah pada CNN.

4.4.2 Perbandingan Hasil Evaluasi Model CNN dan CNN-SVM

Bagian ini menyajikan analisis komparatif antara dua model yang telah dievaluasi, yakni CNN dan kombinasi CNN - SVM. Perbandingan dilakukan secara komprehensif mencakup metrik *presisi, recall* dan *F1 score*, distribusi performa per kelas, serta analisis kelas yang mengalami perubahan signifikan, dengan tujuan untuk mengidentifikasi pengaruh dari SVM sebagai *classifier* akhir pada arsitektur CNN. Perbandingan metrik CNN dan CNN-SVM secara rinci dapat dilihat pada Tabel 4.11.

Tabel 4.11
Perbandingan Metrik CNN dan CNN -SVM

Metrik	CNN	CNN + SVM	Selisih	Status
Accuracy	92,38%	92,60%	0,22%	Meningkat
F1-Score Macro	84,76%	85,37%	0,61%	Meningkat
F1-Score Weighted	92,41%	92,67%	0,26%	Meningkat
Kelas F1 Tinggi (≥0,90)	65	65	0	Sama
Kelas F1 Sedang (0,70–0,89)	54	53	-1	Sedikit turun
Kelas F1 Rendah (<0,70)	16	14	-2	Berkurang
Kelas F1 Sempurna (=1,00)	12	14	+2	Meningkat

Berdasarkan Tabel 4.11, penambahan SVM pada CNN menghasilkan peningkatan di seluruh metrik evaluasi, meskipun dengan margin yang relatif kecil. Akurasi meningkat sebesar 0,22 poin persentase (dari 92,38% menjadi 92,60%), *F1-Score Macro* meningkat sebesar 0,61% (dari 84,76% menjadi 85,37%), dan *F1-Score Weighted* meningkat sebesar 0,26% (dari 92,41% menjadi 92,67%). Meskipun peningkatan ini tidak dramatis, hal tersebut konsisten dan statistik menunjukkan bahwa SVM memberikan kontribusi positif terhadap performa keseluruhan sistem.

Dari perspektif distribusi kategori performa per kelas, jumlah kelas dengan F1-Score rendah (< 0,70) berkurang dari 16 menjadi 14 kelas setelah integrasi SVM, artinya SVM berhasil mengangkat 2 kelas dari kategori rendah ke kategori yang lebih baik. Jumlah kelas dengan performa sempurna (F1 = 1,00) juga meningkat dari 12 menjadi 14 kelas. Di sisi lain, jumlah kelas dengan kategori sedang sedikit berkurang (dari 54 menjadi 53), yang disebabkan oleh 1 kelas yang

sebelumnya berada di kategori sedang (GANTUNGAN-TA-LATIK) berhasil naik ke kategori tinggi, sedangkan terdapat 1 kelas dari kategori sedang yang turun ke kategori rendah.

Analisis pada level kelas menunjukkan bahwa dari 133 kelas yang dapat dibandingkan secara langsung, sebanyak 39 kelas (29,5%) mengalami peningkatan F1-Score dengan penambahan SVM, 72 kelas (54,5%) tidak mengalami perubahan, dan 21 kelas (15,9%) justru mengalami penurunan. Ini berarti SVM memberikan dampak positif pada hampir sepertiga dari total kelas, sementara dampak negatif hanya dialami sekitar 16% kelas. Pola ini mengindikasikan bahwa SVM lebih sering membantu daripada merugikan, meskipun kontribusinya tidak universal. Distribusi Dampak SVM dapat dilihat pada Tabel 4.11.

Tabel 4.11
Distribusi Dampak SVM terhadap Performa Per Kelas

Kategori Perubahan	Jumlah Kelas	Persentase	Contoh Kelas
F1 Meningkat	39	29,5%	TA-TAWA, GA-TEDONG, GANTUNGAN-TA-LATIK, LA-LENGA
F1 Tidak Berubah	72	54,5%	BRA, DU, GRA, KRA, ADEG-ADEG, JA, KA, SA, TA, YA
F1 Menurun	21	15,9%	8 (digit), GANTUNGAN-KA, NA-RAMBAT, WA-TEDONG

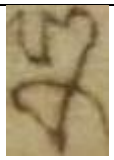
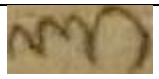
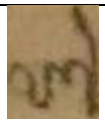
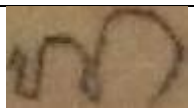
Temuan lainnya adalah pola kelas yang tetap tidak berubah. Sebanyak 72 kelas (54,5%) menunjukkan F1-Score yang identik antara CNN dan CNN+SVM. Kelas-kelas ini umumnya adalah kelas-kelas dengan performa tinggi yang sudah mencapai atau mendekati 1,00 pada CNN sehingga ruang untuk perbaikan sangat

terbatas. Hal ini menunjukkan bahwa CNN sudah mampu mengekstrak fitur yang sangat diskriminatif untuk lebih dari separuh kelas, sehingga SVM tidak memberikan nilai tambah yang signifikan untuk kelas-kelas tersebut.

Dari sisi kelas yang bermasalah, perbandingan menunjukkan bahwa SVM hanya mampu memperbaiki 3 dari 16 kelas yang bermasalah pada CNN, yaitu GA-TEDONG, LA-LENGA, dan TA-TAWA. Sementara itu, 13 kelas bermasalah lainnya masih tetap berada di bawah ambang F1 = 0,70 meskipun menggunakan SVM. Bahkan, kelas digit '8' yang sudah bermasalah di CNN mengalami penurunan lebih lanjut pada CNN+SVM (dari 0,50 menjadi 0,29). Fenomena ini mengindikasikan bahwa tantangan pada kelas-kelas tersebut bisa disebabkan karena kurangnya data latih, kemiripan visual yang ekstrem dengan kelas lain, maupun kompleksitas variasi penulisan yang tidak tertangkap oleh arsitektur saat ini.

Secara keseluruhan, hasil perbandingan ini mendukung penggunaan kombinasi CNN+SVM sebagai pilihan yang lebih unggul dibandingkan CNN untuk pengenalan aksara Bali. Peningkatan konsisten pada semua metrik, berkurangnya jumlah kelas bermasalah, serta meningkatnya jumlah kelas sempurna menjadi argumen yang kuat untuk adopsi pendekatan hybrid ini. Namun, perlu dicatat bahwa SVM bukan solusi menyeluruhmasih terdapat kelas-kelas yang memerlukan strategi khusus di luar perubahan arsitektur classifier. Kelas bermasalah dapat dilihat pada Tabel 4.12.

Tabel 4.12
Kelas Bermasalah: Status pada CNN dan CNN + SVM

Kelas	Gambar Aksara	F1 CNN	F1 CNN+SVM
GANTUNGAN-NGA		0,00	0,00
E-KARA		0,15	0,15
8 (digit)		0,50	0,29
GANTUNGAN-JA		0,46	0,57
2 (digit)		0,48	0,57
GANTUNGAN-KA		0,58	0,49
GANTUNGAN-NYA		0,55	0,55
GANTUNGAN-GA		0,56	0,56
1 (digit)		0,67	0,64
NA-RAMBAT-TEDONG		0,55	0,62
GA-TEDONG		0,67	0,80
LA-LENGA		0,61	0,72
TA-TAWA		0,67	0,86

Tabel 4.12 merangkum kondisi seluruh kelas yang sebelumnya bermasalah pada CNN. Dari perspektif strategi pengembangan model, temuan ini menegaskan bahwa integrasi SVM paling efektif untuk kelas-kelas yang sudah memiliki fitur yang cukup terstruktur di ruang CNN namun memerlukan batas keputusan yang lebih fleksibel. Untuk kelas-kelas seperti GANTUNGAN-NGA dan E-KARA yang tidak menunjukkan perubahan apapun, pendekatan yang diperlukan kemungkinan lebih fundamental, seperti peninjauan kembali kualitas dan kuantitas data latih serta kemungkinan penambahan mekanisme dalam arsitektur CNN untuk fokus pada fitur-fitur yang paling diskriminatif bagi kelas-kelas tersebut.

4.4.2 Perbandingan dengan Penelitian Terdahulu

Untuk memberikan konteks yang lebih luas terhadap hasil yang diperoleh, perlu dilakukan perbandingan dengan penelitian-penelitian terdahulu yang menggunakan dataset yang sama, yakni dataset aksara Bali terisolasi pada naskah lontar dari kompetisi ICFHR 2018. Terdapat tiga penelitian acuan utama yang relevan: pertama, penelitian (Kesiman et al., 2016) yang memperkenalkan AMADI_LontarSet sebagai dataset naskah lontar Bali tulisan tangan pertama, yang menjadi fondasi seluruh penelitian pengenalan aksara Bali berbasis kompetisi dan menetapkan *baseline* awal dengan akurasi 88,39% pada ICFHR 2016; kedua, penelitian (Kesiman et al., 2018) dalam kompetisi *Document Image Analysis Tasks for Southeast Asian Palm Leaf Manuscripts* (ICFHR 2018) yang menggunakan dataset AMADI_LontarSet dengan 133 kelas karakter, 11.710 citra latih, dan 7.673 citra uji; dan ketiga, penelitian (Imawati et al., 2024) yang menggunakan dataset dengan spesifikasi serupa dan menguji tiga arsitektur CNN berbeda.

*T*abel 4.13
Perbandingan Akurasi dengan Penelitian Terdahulu pada Dataset Aksara Bali

Judul (Tahun)	Peneliti	Metode	Dataset (Kelas / Uji)	Akurasi Uji	Keterangan
ICFHR 2016 Competition on Document Image Analysis Tasks for Southeast Asian Palm Leaf Manuscripts (2016)	Burie, J.C., Coustaty, M., Hadi, S., Kesiman, M.W.A., Ogier, J.M., Paulus, E., Sok, K., Sunarya, I.M.G., & Valy, D.	Metode terbaik ICFHR 2016	133 kelas / 7.673 uji	88,39%	Acuan ICFHR 2016
ICFHR 2018 Competition on Document Image Analysis Tasks for Southeast Asian Palm Leaf (2018)	Kesiman, M.W.A., Valy, D., Burie, J.C., Paulus, E., Suryani, M., Hadi, S., Verleysen, M., Chhun, S., & Ogier, J.M	Very Deep CNN 100 lapisan + Dense Connection + Augmentasi Data	133 kelas / 7.673 uji	92,17%	ICFHR 2018 Tim G20
Training VGG16, MobileNetV1 and Simple CNN Models	Imawati, I.A.P.F., Sudarma, M.,	VGG16 (training	133 kelas / split internal	85,30%	Penelitian terbaru (2024)

Judul (Tahun)	Peneliti	Metode	Dataset (Kelas / Uji)	Akurasi Uji	Keterangan
from Scratch for Balinese Inscription Recognition (2024)	Putra, I.K.G.D., Bayupati, I.P.A., & Jo, M.	from scratch)			
Penelitian Ini — CNN	G.A Devani Zelvia	CNN Custom	133 kelas / 7.673 uji	92,38%	Melampaui G20 (+0,21%)
Penelitian Ini — CNN + SVM	G.A Devani Zelvia	CNN + SVM (Hybrid)	133 kelas / 7.673 uji	92,60%	Tertinggi — Melampaui G20 (+0,43%)

Berdasarkan Tabel 4.13, hasil penelitian ini menunjukkan capaian yang signifikan dalam konteks penelitian pengenalan aksara Bali secara keseluruhan. Model CNN v6 standalone dengan akurasi 92,38% telah melampaui state-of-the-art sebelumnya yang dipegang oleh Group 20 (G20) pada ICFHR 2018 sebesar 92,17%, dengan selisih 0,21 poin persentase. Lebih jauh, model CNN + SVM yang menghasilkan akurasi 92,60% memperlebar selisih tersebut menjadi 0,43 poin persentase di atas capaian G20. Ini merupakan temuan yang penting mengingat G20 menggunakan arsitektur very deep CNN dengan 100 lapisan dan koneksi padat, yang merupakan arsitektur yang jauh lebih kompleks dibandingkan arsitektur CNN yang diusulkan dalam penelitian ini.

Jika dibandingkan dengan penelitian Imawati et al. (2024) yang menggunakan VGG16 dengan akurasi pengujian 85,30%, model CNN unggul sebesar 7,08 poin persentase dan CNN + SVM unggul sebesar 7,30 poin persentase. Perbedaan yang

cukup besar ini mengindikasikan bahwa arsitektur CNN yang dirancang khusus untuk karakteristik aksara Bali memberikan hasil yang jauh lebih baik dibandingkan penggunaan arsitektur generik seperti VGG16 yang dilatih dari awal tanpa *transfer learning.*