

Sistem Deteksi Dengan Algoritma *Deep Learning Convolutional Neural Network* Pada Sibi (Sistem Isyarat Bahasa Indonesia)

Ilham Rizaldy¹⁾ Heri Pratikno²⁾ Musayyanah³⁾

Program Studi/Jurusan Teknik Komputer
 Universitas Dinamika.

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1) ilhamrizaldy0729@gmail.com 2) heri@dinamika.ac.id 3) musayyanah@dinamika.ac.id

Abstrak: Penyandang tunarungu adalah mereka yang mengalami gangguan dalam pendengarannya, sehingga untuk berkomunikasi mereka menggunakan bahasa isyarat. Di Indonesia, terdapat 2 bahasa isyarat yang digunakan, yaitu BISINDO (Bahasa Isyarat Indonesia) yaitu bahasa isyarat yang lahir dari golongan tunarungu itu sendiri, lalu SIBI (Sistem Isyarat Bahasa Indonesia) yang diresmikan pemerintah pada tahun 1994. Namun masih banyak yang kurang memahami bahasa isyarat ini, sehingga menyulitkan orang biasa untuk berkomunikasi dengan penyandang tunarungu. Dari permasalahan tersebut, maka dirancanglah sebuah sistem untuk mendeteksi simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan *Convolutional Neural Network*. *Convolutional Neural Network* itu sendiri adalah pengembangan dari *Machine Learning*. ada 2 tahap utama pada *Convolutional Neural Network*, yaitu *Feature Extraction* dan *Classification*. Pada penelitian ini, *Dataset* simbol isyarat yang digunakan yaitu 6 simbol, 3 kata ganti orang (Saya, Kamu, Dia) dan 3 kata ganti perasaan (Maaf, Cinta, Sedih). Total dari *Dataset* yang digunakan yaitu 540 citra sebagai *data Training*, 60 citra sebagai *data validasi* dan 60 citra sebagai *Data Testing*. 6 simbol inilah yang *ditraining* menggunakan *Convolutional Neural Network* agar dapat diklasifikasikan. Hasil *Training* yang dilakukan menghasilkan akurasi sebesar 91%, sedangkan validasi yang dilakukan memiliki akurasi sebesar 90%. Pada hasil prediksi mendapatkan akurasi pada simbol “Cinta” di angka 80%, simbol “Saya” dan “Dia” pada angka 70%, simbol “Kamu” sebesar 90%, simbol “Sedih” sebesar 100% dan simbol “Maaf” di angka 0%.

Kata kunci: SIBI (Sistem Isyarat Bahasa Indonesia), tuna rungu, *Convolutional Neural Network*, *Dataset*.

Masyarakat Penyandang disabilitas khususnya tunarungu sudah terbilang sangat banyak di Indonesia, menurut (Jelita, 2018) dalam artikel di Media Indonesia, disebutkan bahwa penyanda tunarungu yang menjadi pemilih tetap pada pemilu 2019 yaitu berjumlah 475.852 orang. Penyandang tunarungu dalam berkomunikasi menggunakan bahasa isyarat. Menurut (Murti, et al., 2016) dalam penelitiannya menjelaskan bahwa bahasa isyarat adalah salah satu bahasa komunikasi yang dilakukan dengan menggunakan pergerakan tangan, pergerakan tubuh atau ekspresi wajah. Penyandang tunarungu menggunakan BISINDO (Bahasa Isyarat Indonesia) dan BISINDO memiliki keberagaman, dalam artian beberapa daerah memiliki perbedaan pada bahasa

isyarat, hingga pada tahun 1994 dengan keputusan dari Mendikbud No.0161/U/2994 yang berisi mengenai pembakuan dari SIBI (Sistem Isyarat Bahasa Indonesia). Dengan adanya SIBI (Sistem Isyarat Bahasa Indonesia), lahirlah bahasa yang lebih *universal* atau bahasa pemersatu.

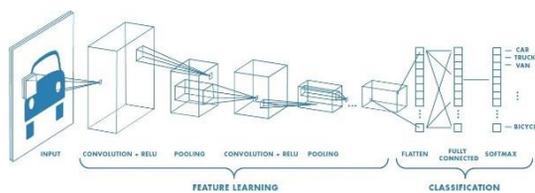
Namun masyarakat Indonesia banyak yang masih asing atau awam dengan bahasa isyarat ini, sehingga ketika penyandang tunarungu berusaha berkomunikasi dengan orang biasa, mereka merasa kesulitan. Di era sekarang ini, sudah banyak pengembangan pada teknologi deteksi objek, khususnya penggunaan pada *Deep Learning* atau *Machine Learning*. Salah satu metode pengembangan yang digunakan adalah

Convolutional Neural Network. Peneliti umumnya menggunakan *Convolutional Neural Network* sebagai deteksi objek dalam *Multi Class, Binary Class* atau dapat juga dengan *Multi Label*. Contoh pada penelitian sebelumnya yang dilakukan oleh (Darmatiasia, 2020), yang membahas pengenalan pada alfabet SIBI (Sistem Isyarat Bahasa Indonesia) dengan menggunakan *Gradient-Convolutional Neural Network*. Akurasi yang didapatkan mencapai angka 98%. Namun *Dataset* yang digunakan, dibatasi 5 alfabet yaitu A, B, C, D, E.

Atas permasalahan tersebut, maka dirancanglah sebuah sistem deteksi pada simbol SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan *Convolutional Neural Network*. Pada penelitian ini menggunakan *Dataset* baru dimana peneliti membuat sendiri. Namun simbol isyarat dibatasi dengan 6 simbol yaitu Cinta, Dia, Kamu, Maaf, Saya, Sedih. Model yang digunakan pada *Convolutional Neural Network* adalah Model *Sequential*.

Convolutional Neural Network

Convolutional Neural Network adalah pengembangan dari Multilayer Perceptron (MLP) yang termasuk dalam *neural network* bertipe *feed forward* (bukan berulang). *Convolutional Neural Network* dapat mengolah dan mengklasifikasikan data 2 dimensi (Nugroho, et al., 2020), sedangkan menurut (Arrofiqoh & , 2018) menjelaskan bahwa *Convolutional Neural Network* termasuk dalam jenis *Deep Learning* karena kedalaman jaringannya.

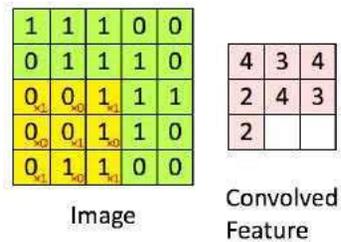


Gambar 1. Contoh arsitektur CNN
Sumber : (Sena, 2017)

Pada dasarnya arsitektur pada *convolutional Neural Network* memiliki 2 tahap seperti pada gambar 1, yaitu *Feature Learning* dan *Classification*. *Feature Learning* adalah tahap dimana citra input dilakukan fitur ekstraksi dengan beberapa *layer*. Dalam proses inilah yang menjadikan *Convolutional Neural Network*, yaitu citra ekstraksi dilakukan konvolusi. *Output* dari *Feature Learning* yang nantinya menentukan fitur-fitur setiap data input dan *class*. *Layer* pada *Feature Learning* antara lain:

a. Konvolusi

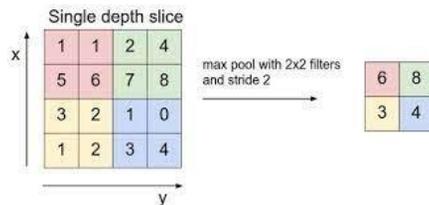
Konvolusi berfungsi untuk mengekstraksifitur pada citra input. Dalam implementasi nya konvolusi menggunakan *kernel filter* dengan ukuran tertentu dengan pergeseran *filter* (*strides*). Contoh proses konvolusi dapat dilihat pada gambar 2. *Output* pada konvolusi biasanya dilakukan fungsi aktivasi dengan ReLU (*RectifiedLinier Unit*), berfungsi untuk menormalisasikan nilai negatif menjadi 0 pada *output* konvolusi.



Gambar 2. Proses konvolusi CNN
(Sumber: Suartika, et al., 2016)

b. Max Pooling

Output dari hasil konvolusi dilakukan *Max Pooling* untuk mengurangi dimensi pada citra, agar memudahkan komputasi pada saat *Training*. Hampir sama dengan konvolusi, *Max Pooling* juga menggunakan *kernel filter* dan *strides* pada proses nya, namun filter yang dilakukan *Max Pooling* adalah mengambil nilai terbesar lalu memasukkannya pada dimensi baru seperti pada gambar 3.



Gambar 3. Proses Max Pooling CNN
(Sumber: Suartika, et al., 2016)

Pada tahap *Classification* atau *Fully Connected*, dimana *layer* ini merupakan kelanjutan dari proses sebelumnya yaitu *feature learning*, pada *layer* ini, *output* dari proses sebelumnya ditentukan sesuai perhitungan pada fitur untuk dilakukan prediksi *class*. *Output* dari *Feature Learning* perlu dilakukan proses *Flatten* yaitu proses mengubah citra 2 dimensi menjadi 1 dimensi, tujuannya dapat dilakukan pengklasifikasian secara linear. Fungsi aktivasi yang digunakan untuk klasifikasi yaitu *sigmoid* untuk *binary class* dan *softmax* untuk *multiclass*.

METODE

Pengumpulan Dataset

Dataset yang digunakan adalah olahan pribadi seperti pada tabel 1. *Dataset* diambil dengan 5 subjek yang berbeda dan dilakukan standarisasi yaitu dengan memberikan *green screen* pada *background* dan pakaian yang digunakan berwarna putih. Hal ini dilakukan untuk memudahkan saat proses *Image Pre-processing*. Total *Dataset* terbagi menjadi 540 *Data Training*, 60 *Data Validation* dan 60 *Data Testing*.

Tabel 1. Pengambilan dataset

Subjek/ gender	Usia	Tanggal ambil data	
Subjek 1/ Laki-laki	23 Tahun	“Saya”	(12/04/2021)
		“Kamu”	(19/04/2021)
		“Dia”	(19/04/2021)
		“Maaf”	(19/04/2021)
		“Cinta”	(19/04/2021)
Subjek 2/ Laki-laki	21 Tahun	“Sedih”	(20/04/2021)
		“Saya”	(20/04/2021)
		“Kamu”	(20/04/2021)
		“Dia”	(20/04/2021)
		“Maaf”	(20/04/2021)
Subjek 3/ Laki-laki	23 Tahun	“Cinta”	(20/04/2021)
		“Sedih”	(20/04/2021)
		“Saya”	(20/04/2021)
		“Kamu”	(20/04/2021)
		“Dia”	(20/04/2021)
Subjek 4/ Wanita	51 Tahun	“Maaf”	(20/04/2021)
		“Cinta”	(20/04/2021)
		“Sedih”	(20/04/2021)
		“Saya”	(23/04/2021)
		“Kamu”	(23/04/2021)
Subjek 5/ Wanita	22 Tahun	“Dia”	(23/04/2021)
		“Maaf”	(23/04/2021)
		“Cinta”	(23/04/2021)
		“Sedih”	(23/04/2021)
		“Saya”	(27/04/2021)
		“Kamu”	(27/04/2021)
		“Dia”	(27/04/2021)
		“Maaf”	(27/04/2021)
		“Cinta”	(27/04/2021)
		“Sedih”	(27/04/2021)

Contoh dari *Data Training* adalah pada gambar 4, *Data Testing* ditunjukkan pada gambar 5 dan gambar 6. *Data Testing* mempunyai 2 variasi, yang pertama adalah dengan standar seperti halnya *Data Training* yaitu berwarna putih, sedangkan variasi yang kedua adalah dengan menggunakan pakaian selain putih, ini dilakukan untuk mengetahui perbedaan pada prediksi jika menggunakan pakaian berwarna lain.



Gambar 4. *Data Training* simbol “Saya”



Gambar 5. *Data Testing* simbol "Saya"



Gambar 6. *Data Testing* variasi simbol "Saya"

Image Pre-processing

Beberapa tahap *Image Pre-processing* dilakukan terlebih dahulu terhadap *Dataset*. Hal ini dilakukan agar proses klasifikasi dapat terfokus dengan bahasa isyarat, tanpa *Image Pre-processing*, maka citra diklasifikasikan sebagai orang saja.

Proses pertama yang dilakukan adalah *remove background* pada citra asli dengan masking dengan *threshold* batas atas dan batas bawah warna hijau. Rentang nilai diatur dari 4-85, sehingga nilai antara 4-85 menjadi nilai 255, dan selain itu menjadi nilai 0. Hasilnya seperti pada gambar 7.



Gambar 7. Hasil proses *Remove Background*

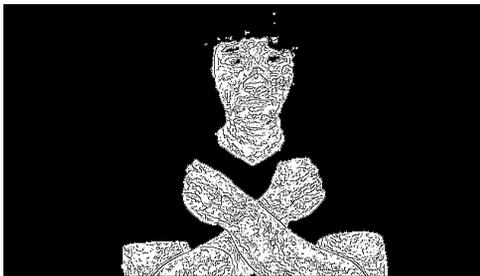
Hasil dari *remove background* dilakukan 2 tahap *Image Pre-processing* lagi. Yang pertama mengubah menjadi warna hitam putih seperti gambar 8. Yang kedua adalah proses filter dengan Canny untuk mendapatkan garis tepi dari objek seperti pada gambar 9. Pemilihan filter Canny untuk proses deteksi garis tepi adalah filter yang terbaik menurut (Katiya & Arun, 2014). Proses akhir adalah dengan *masking* pada gambar 8 dan gambar 9, hasilnya ditunjukkan pada gambar 10. lalu hasil proses di-*resize* menjadi dimensi 100x100.



Gambar 8. Hasil *Image* menjadi hitam putih



Gambar 9. Hasil proses Canny



Gambar 10. Hasil akhir *Image Pre-processing*

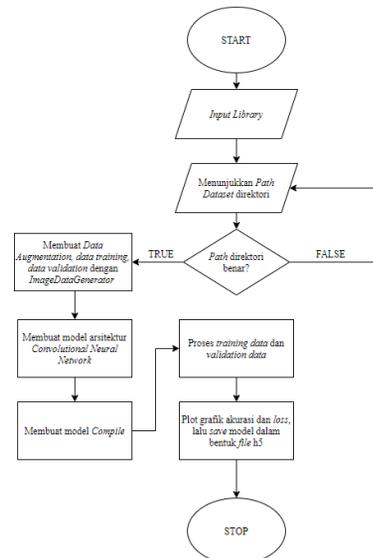
Proses Data Training

Proses *Data Training* menggunakan *model sequential* dengan arsitektur *Convolutional Neural Network* seperti pada gambar 11. Arsitektur menggunakan 5 layer pada *Feature Learning*. *Kernel filter* yang digunakan berukuran 3x3 dengan nilai pergeseran *strides* adalah 1. Setiap *Layer* digunakan *padding*, bertujuan untuk memanipulasi citra dengan cara menambahkan dimensi tambahan pada sisi-sisi citra dengan nilai 0, sehingga fitur pada citra dapat terekstraksi lebih

banyak. Pada tahap *classification* menggunakan 3 *Layer dense* yaitu 512, 256 dan 6 (jumlah *class*) dan dikarenakan *class* prediksi sebanyak 6 (*multiclass*), maka digunakan fungsi aktivasi *softmax* untuk klasifikasi nya.

```
Model: "sequential"
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)             (None, 100, 100, 32)     320
max_pooling2d (MaxPooling2D) (None, 49, 49, 32)       0
conv2d_1 (Conv2D)           (None, 49, 49, 32)       9248
max_pooling2d_1 (MaxPooling2D) (None, 24, 24, 32)       0
conv2d_2 (Conv2D)           (None, 24, 24, 64)       18496
max_pooling2d_2 (MaxPooling2D) (None, 12, 12, 64)       0
conv2d_3 (Conv2D)           (None, 12, 12, 64)       36928
max_pooling2d_3 (MaxPooling2D) (None, 11, 11, 64)       0
conv2d_4 (Conv2D)           (None, 11, 11, 128)      73856
max_pooling2d_4 (MaxPooling2D) (None, 10, 10, 128)      0
flatten (Flatten)           (None, 12800)            0
dropout (Dropout)          (None, 12800)            0
dense (Dense)               (None, 512)              6554112
dense_1 (Dense)             (None, 256)              131328
dense_2 (Dense)             (None, 6)                1542
-----
Total params: 6,825,830
Trainable params: 6,825,830
Non-trainable params: 0
```

Gambar 11. Arsitektur CNN



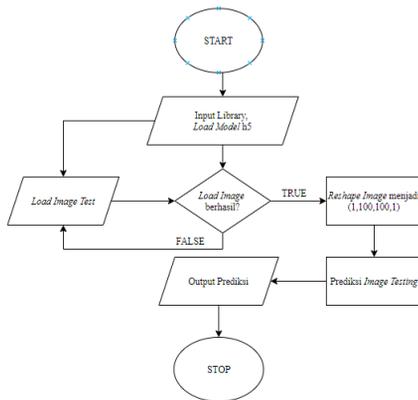
Gambar 12. Flowchart proses Deep Learning

Alur proses terdapat pada *Flowchart* di gambar 12. *Data* direktori yang digunakan, diakses dari Google Drive. Karena *Dataset* berjumlah relatif sedikit, maka *Dataset* dilakukan *Data* augmentasi agar tidak terjadi *underfitting*, yang menjadikan hasil akurasi rendah. Selain itu, juga sebagai penambahan variasi *Dataset*. Proses *Training* menggunakan *optimizer* "Adam" dengan *learning rate* bernilai 0,001. *Training* dilakukan sebanyak 250 *Epochs*. Model yang sudah di-*training* kemudian disimpan dengan model h5, sehingga tidak perlu melakukan proses

Training secara berulang.

Proses Data Testing

Proses *Data Testing* adalah proses dimanahasil *Training* dilakukan prediksi dengan *Data Testing* untuk menentukan apakah prediksi pada *class* akurat. *Flowchart* pada *Data Testing* terdapat pada gambar 13.



Gambar 13. *Flowchart Data Testing*

HASIL DAN PEMBAHASAN

Penunjang Pengujian

Adapun penunjang pengujian yang digunakan adalah:

- Dataset* menjadi peran penting sebagai *input* dari *Data Training*, *Data Validation* dan *Data Testing*.
- Arsitektur *Convolutional Neural Network*, arsitektur pada setiap kasus berbeda-beda, namun arsitektur CNN ini penting dikarenakan pada *Feature Learning* terdapat proses konvolusi yang menjadi inti dari metode ini.
- Image Pre-processing*, proses ini berfungsi agar *Dataset* dapat fokus pada bahasa isyarat ketika proses *Training*, tanpa proses ini, maka *Dataset* terprediksi sebagai orang bukan bahasa isyarat.
- Proses *Training* dan *Validation*, proses ini adalah proses dimana komputer mempelajari citra (*Deep Learning*) dan mendeteksi citra sesuai klasifikasi tertentu. Hasil *Training* disimpan dengan model h5.

Hasil Training dan Validation

Hasil proses ditunjukkan dengan akurasi pada *Training* dan *Validation*. Data dalam tabel 2 dan tabel 3 merupakan hasil dari 5 *Layer* pada arsitektur CNN yang merupakan hasil terbaik. Data

diambil dari 10 *epochs* terakhir, namun grafik pada gambar 14 dan 15 adalah hasil dari *epochs* 1-250. Hasil akhir pada *Training* menunjukkan akurasi sebesar 91%, sedangkan pada *validation* memiliki akurasi sebesar 90%.

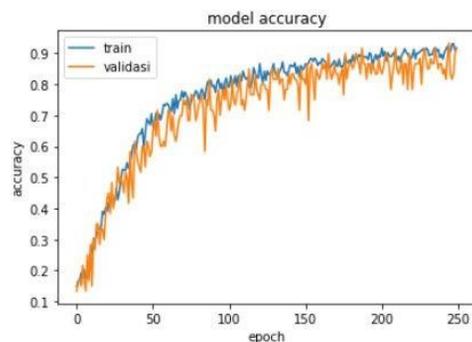
Hasil kedua grafik terlihat fluktuatif, namun pada hasil grafik tidak terjadi *Overfitting* dimana ditunjukkan dengan hasil *loss* dan akurasi yang rendah. Hasil grafik masih berhimpit dan pada grafik akurasi cenderung meningkat, sedangkan pada grafik *loss* cenderung menurun.

Tabel 2. Hasil akurasi *Training* dan *Validation*

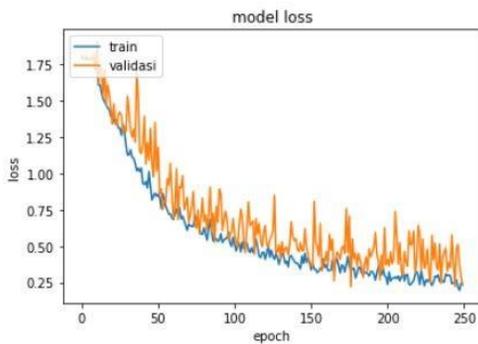
Epochs	Akurasi Training (%)	Akurasi Validation (%)
241	89	83
242	89	85
243	92	82
244	93	88
245	91	93
246	92	83
247	93	82
248	94	83
249	92	90
250	91	90

Tabel 3. Hasil *Loss Training* dan *Validation*

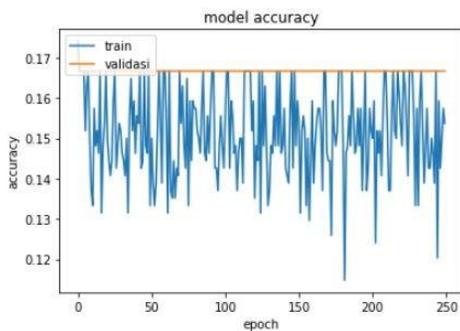
Epochs	Akurasi Training (%)	Akurasi Validation (%)
241	28	39
242	31	36
243	22	58
244	26	38
245	23	22
246	25	48
247	25	51
248	17	37
249	22	29
250	25	23



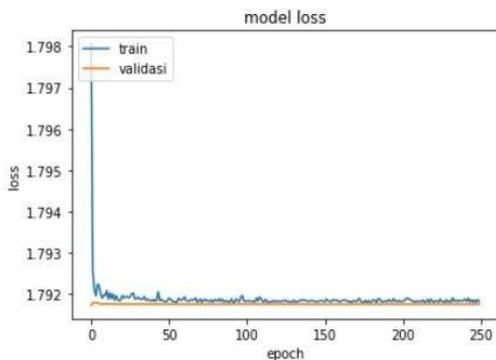
Gambar 14. Grafik akurasi 5 *Layer*



Gambar 15. Grafik Loss 5 Layer



Gambar 16. Grafik akurasi 6 Layer



Gambar 17. Grafik Loss 6 Layer

Hasil perbandingan grafik arsitektur 5 Layer dengan 6 Layer sangat terlihat secara signifikan. Arsitektur yang menggunakan 6 Layer hasil Training dan Validation didapatkan sangat kurang. Terlihat pada gambar 16 akurasi sangat fluktuasi, bahkan cenderung tidak ada peningkatan pada grafik akurasi, sama halnya dengan grafik Loss pada gambar 17. Seharusnya grafik Loss yang baik adalah yang cenderung menurun serta hasil Loss yang rendah. Dari analisis diatas, hasil arsitektur yang terbaik adalah arsitektur CNN dengan 5 Layer.

Tabel 4. Parameter uji Precision dan Recall

	Precision	Recall
“Cinta”	1.00	1.00
“Dia”	0.80	1.00
“Kamu”	1.00	0.86
“Maaf”	0.50	1.00
“Saya”	1.00	0.57
“Sedih”	1.00	1.00
Macro Avg	0.88	0.90
Weighted Avg	0.93	0.88

Tabel 5. Parameter uji F1-Score dan Support

	F1-Score	Support
“Cinta”	1.00	6
“Dia”	0.89	4
“Kamu”	0.92	7
“Maaf”	0.67	3
“Saya”	0.73	7
“Sedih”	1.00	5
Accuracy	0.88	32
Macro Avg	0.87	32
Weighted Avg	0.88	32

Parameter uji pada tabel 4 Precision (spesifitas) mendapatkan hasil 88%, sedangkan Recall (sensivitas) mendapatkan hasil 90%. Kedua hasil diatas merupakan hasil rata-rata jumlah nilai setiap class dari masing-masing parameter. Tabel 5 menampilkan nilai F1-score, yang merupakan acuan pada hasil model untuk proses prediksi.

Perbandingan Hasil Pada 4 Arsitektur

Pengujian perbandingan variasi arsitektur Layer bertujuan untuk mengetahui arsitektur CNN terbaik. Variasi arsitektur yaitu 4 Layer, 5 Layer, 6 Layer dan 7 Layer. Hasil dapat dilihat pada tabel 6 untuk parameter akurasi dan tabel 7 untuk parameter Loss.

Tabel 6. Tabel perbandingan hasil akurasi

Jumlah Layer	Epochs	Akurasi Training (%)	Akurasi Validation (%)
4 Layer	241	87	83
	242	89	83
	243	90	83
	244	86	83
	245	89	80
	246	88	80
	247	88	88

Jumlah Layer	Epochs	Akurasi Training (%)	Akurasi Validation (%)
5 Layer	248	87	83
	249	86	86
	250	86	83
	241	89	83
	242	89	85
	243	92	82
	244	93	88
	245	91	93
	246	92	83
	247	93	82
6 Layer	248	94	83
	249	92	90
	250	91	90
	241	16	17
	242	15	17
	243	16	17
	244	19	17
	245	12	17
	246	17	17
	247	16	17
7 Layer	248	15	17
	249	19	17
	250	17	17
	241	15	17
	242	13	17
	243	18	17
	244	16	17
	245	14	17
	246	16	17
	247	18	17
248	14	17	
249	18	17	
250	17	17	

Tabel 7. Tabel perbandingan hasil Loss

Jumlah Layer	Epochs	Training Loss (%)	Validation Loss (%)
4 Layer	241	35	59
	242	33	47
	243	28	39
	244	42	40
	245	27	51
	246	36	44
	247	30	33
	248	38	42
	249	31	29
	250	42	41
5 Layer	241	28	39
	242	31	36
	243	22	58
	244	26	38
	245	23	22
	246	25	48

6 Layer	247	25	51
	248	17	37
	249	22	29
	250	25	23
	241	179	179
	242	179	179
	243	179	179
	244	179	179
	245	179	179
	246	179	179
7 Layer	247	179	179
	248	179	179
	249	179	179
	250	179	179
	241	179	179
	242	179	179
	243	179	179
	244	179	179
	245	179	179
	246	179	179
247	179	179	
248	179	179	
249	179	179	
250	179	179	

Tabel 6 menunjukkan bahwa 5 Layer mendapatkan akurasi 91%, sedangkan pada 4 Layer akurasinya sebesar 86%, pada 6 Layer dan 7 Layer nilai akurasi didapatkan 17%. Pada Tabel 7 menunjukkan hasil Loss untuk 5 Layer sebesar 25%, 4 Layer sebesar 42%, 6 Layer dan 7 Layer sebesar 179%. Hasil dari perbandingan arsitektur Layer tersebut, hasil terbaik terdapat pada 5 Layer.

Hasil Prediksi

Persentase pada hasil prediksi adalah jumlah prediksi yang benar dikalikan dengan 10%, karena setiap class memiliki 10 Data Testing, sehingga 1 Data mewakili 10%.

Tabel 8. Prediksi Simbol "Cinta"

No	Subjek	Cinta
1	Subjek 1	Citra 1 Prediksi: Cinta
		Citra 2 Prediksi: Cinta
2	Subjek 2	Citra 1 Prediksi: Sedih
		Citra 2 Prediksi: Cinta
3	Subjek 3	Citra 1 Prediksi: Cinta
		Citra 2 Prediksi: Sedih
4	Subjek 4	Citra 1 Prediksi: Cinta
		Citra 2 Prediksi: Cinta

No	Subjek	Cinta
5	Subjek 5	Citra 1 Prediksi: Cinta Citra 2 Prediksi: Cinta
	Persentase	Benar: 8 Salah: 2 Persentase: 80%

Tabel 9. Prediksi Simbol "Dia"

No	Subjek	Cinta
1	Subjek 1	Citra 1 Prediksi: Dia Citra 2 Prediksi: Dia
2	Subjek 2	Citra 1 Prediksi: Saya Citra 2 Prediksi: Dia
3	Subjek 3	Citra 1 Prediksi: Kamu Citra 2 Prediksi: Dia
4	Subjek 4	Citra 1 Prediksi: Dia Citra 2 Prediksi: Dia
5	Subjek 5	Citra 1 Prediksi: Saya Citra 2 Prediksi: Dia
	Persentase	Benar: 7 Salah: 3 Persentase: 70%

Tabel 10. Prediksi Simbol "Kamu"

No	Subjek	Kamu
1	Subjek 1	Citra 1 Prediksi: Kamu Citra 2 Prediksi: Kamu
2	Subjek 2	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Kamu
3	Subjek 3	Citra 1 Prediksi: Kamu Citra 2 Prediksi: Kamu
4	Subjek 4	Citra 1 Prediksi: Kamu Citra 2 Prediksi: Kamu
5	Subjek 5	Citra 1 Prediksi: Kamu Citra 2 Prediksi: Kamu
	Persentase	Benar: 9 Salah: 1 Persentase: 90%

Tabel 11. Prediksi Simbol "Maaf"

No	Subjek	Cinta
1	Subjek 1	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Sedih
2	Subjek 2	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Sedih
3	Subjek 3	Citra 1 Prediksi: Cinta Citra 2 Prediksi: Sedih
4	Subjek 4	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Sedih
5	Subjek 5	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Sedih
	Persentase	Benar: 0 Salah: 10 Persentase: 0%

Tabel 12. Prediksi Simbol "Saya"

No	Subjek	Cinta
1	Subjek 1	Citra 1 Prediksi: Saya Citra 2 Prediksi: Saya
2	Subjek 2	Citra 1 Prediksi: Saya Citra 2 Prediksi: Sedih
3	Subjek 3	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Saya
4	Subjek 4	Citra 1 Prediksi: Saya Citra 2 Prediksi: Sedih
5	Subjek 5	Citra 1 Prediksi: Saya Citra 2 Prediksi: Saya
	Persentase	Benar: 7 Salah: 3 Persentase: 70%

Tabel 13. Prediksi Simbol "Sedih"

No	Subjek	Cinta
1	Subjek 1	Citra 1 Prediksi: Sedih Citra 2 Prediksi: Sedih
2	Subjek 2	Citra 1 Prediksi:

No	Subjek	Cinta
		Sedih
		Citra 2 Prediksi:
		Sedih
3	Subjek 3	Citra 1 Prediksi:
		Sedih
		Citra 2 Prediksi:
		Sedih
4	Subjek 4	Citra 1 Prediksi:
		Sedih
		Citra 2 Prediksi:
		Sedih
5	Subjek 5	Citra 1 Prediksi:
		Sedih
		Citra 2 Prediksi:
		Sedih
	Persentase	Benar: 10
		Salah: 0
		Persentase: 100%

Prediksi yang salah ditunjukkan dengan *text* berwarna merah. Hasil akurasi prediksi simbol “Cinta” sebesar 80%, simbol “Saya” dan “Dia” sebesar 70%, simbol “Kamu” sebesar 90%, simbol “Sedih” sebesar 100% dan simbol “Maaf” sebesar 0%.

Hasil prediksi dapat dipengaruhi dari 2 hal, yang pertama adalah bentuk tangan, contoh pada simbol “Maaf” terprediksi sebagai simbol “Sedih” dikarenakan kedua simbol itu hampir mirip. Pengaruh lain dari hasil prediksi adalah pakaian yang digunakan. Subjek 2 pada citra 1 memiliki *noise* berupa logo pada citra, sehingga pada prediksi simbol “Cinta”, “Dia”, “Kamu” dan “Maaf” terjadi kesalahan prediksi. Logo pada pakaian tersebut saat proses *Image Pre-processing* tidak di-*remove* melainkan ikut terfilter, sehingga mempengaruhi pada hasil prediksi.

SIMPULAN

Berdasarkan hasil pengujian yang dilakukan dengan *Image Pre-processing*, *Training* dan *Validation Dataset* dan prediksi (*Data Testing*), dapat ditarik kesimpulan:

1. Pengujian akurasi hasil *Training* dan *Validation* menunjukkan nilai sebesar 91% untuk *Training* dan 90% untuk *Validation*. Pada hasil grafik tidak mengalami *Overfitting*.
2. *Dataset* perlu dilakukan *Image Pre-processing* terlebih dahulu sebelum dilakukan *Training* dan *Validation*. Tujuannya adalah untuk memudahkan mendeteksi bahasa isyarat.
3. Pada hasil prediksi mendapatkan hasil akurasi prediksi sebagai berikut:
 - a. Simbol “Cinta”: 80%
 - b. Simbol “Dia”: 70%
 - c. Simbol “Kamu”: 90%

- d. Simbol “Maaf”: 0%
- e. Simbol “Saya”: 70%
- f. Simbol “Sedih”: 100%

SARAN

Pengembangan pada *Deep Learning* khususnya pada metode *Convolutional Neural Network* sebagai berikut, yaitu:

1. Pada *Dataset* dapat diberi variasi simbol bahasa isyarat selain yang ada pada penelitian ini. Aspek lain dapat dari kuantitas pada *Dataset* karena pada penelitian ini jumlah pada *Dataset* relatif sedikit. Lalu variasi yang lain dapat menggunakan citra tanpa *background green screen*.
2. *Dataset* ditingkatkan kualitasnya dengan menggunakan *Labelling*, *Annotation*, *Ground Truth* atau *Caption*.
3. Metode dapat menggunakan yang lain seperti YOLO (*You Only Look Once*) atau SSD (*Single Shot Detector*).
4. Menambahkan *Transfer Learning (Pre-trained Model)* pada *Convolutional Neural Network*, seperti VGG16 atau ResNet50 agar grafik halus tidak fluktuatif.

DAFTAR PUSTAKA

- Arrofiqoh, E. N. & H., 2018. Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi. *Geomatika*, 24(2), pp. 61-68.
- Darmatasia, 2020. Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Gradient Convolutional Neural Network. *Jurnal Informatika Sains dan Teknologi*.
- Jelita, I. N., 2018. *MEDIA INDONESIA*. [Online] Available at: <https://m.mediaindonesia.com/politik-dan-hukum/204499/tercatat-12-juta-penyandang-disabilitas-masuk-dpt-pemilu-2019>
- Katiya, S. K. & Arun, P. V., 2014. Comparative Analysis of Common Edge Detection Techniques of Context of Object Extraction. *IEEE TGRS*, 50(11b), pp. 68-79.
- Nugroho, P. A., Fenriana, I. & Ariyanto, R., 2020. Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Jurnal Algor*.
- Sena, S., 2017. *Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN)*.
- Ilham Rizaldy, Heri Pratikno, Musayyanah
JCONES Vol. 10, No. 2 (2021) Hal: 68

[Online]
Available at:
<https://medium.com/@samuelsena/pengenal-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>
[Accessed 1 April 2021].

S., Murti, D. H. & Khotimah, W. N., 2016. Pengenalan Sistem Isyarat Bahasa Indonesia Menggunakan Kombinasi Fitur Statis dan Fitur Dinamis LMC Berbasis L-GCNN. *Jurnal Ilmiah Teknologi Informasi*, 14(2), pp. 217-230.

Suartika, I. W., Wijaya, A. Y. & Soelaiman, R., 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network. *Jurnal Teknik ITS*, V(1).