



ANALISIS PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK SISTEM DETEKSI KATARAK

Akmal Rusdy Prasetyo ^a, Sussi, S.Si, M.T.^b, Bagus Aditya S.T., M.T. ^c

^a S1 Teknik Telekomunikasi, akmalrusdy@student.telkomuniversity.ac.id, Universitas Telkom,

^b S1 Teknik Telekomunikasi, sussiss@telkomuniversity.ac.id, Universitas Telkom,

^c S1 Teknik Telekomunikasi, goesaditya@telkomuniversity.ac.id, Universitas Telkom

Abstract

One of the most vital senses for humans is sight. Humans utilize their eyes to take in visual information that is used for a variety of tasks, but vision problems are prevalent, ranging from minor disorders to serious disorders that can result in blindness. Cataracts are one of the factors contributing to this vision loss. In Indonesia, cataracts alone account for 81 percent of cases of blindness, and 40 percent of those affected don't even realize they have them. A technique for early cataract identification using digital photographs is one way to lower the incidence of cataract-related blindness. We used Support Vector Machine (SVM) and Convolutional Neural Network (CNN) techniques to create a cataract detection system. These two techniques are used to categorize normal eye classes, immature cataracts, and mature cataracts. A digital image that has been downsized to 64x64 pixels in Joint Photographic Group (JPG) format that was taken from earlier study serves as the input data. Support Vector Machine (SVM) and Convolutional Neural Network (CNN) methods used in the cataract detection process provide optimal results. The Support Vector Machine (SVM) method itself produces an accuracy value of 96.67%, while the Convolutional Neural Network method produces a better accuracy value of 98.89%.

Keywords: Cataract, Digital Image, Machine Learning, Support Vector Machine, Convolutional Neural Network

Abstrak

Mata adalah salah satu indera terpenting bagi manusia. Melalui mata, manusia menyerap informasi visual yang digunakan untuk melakukan berbagai aktivitas, tetapi gangguan terhadap penglihatan banyak terjadi, dari gangguan ringan hingga gangguan serius yang dapat menyebabkan kebutaan, kebutaan. Salah satu penyebab gangguan penglihatan ini adalah katarak. Katarak sendiri menjadi penyebab tertinggi kebutaan di Indonesia (81%) dengan 40% penderitanya tidak mengetahui bahwa dirinya menderita katarak. Salah satu solusi untuk mengurangi prevalensi kebutaan yang disebabkan oleh penyakit katarak yaitu dengan sistem deteksi dini penyakit katarak dengan memanfaatkan citra digital. Kami merancang sistem deteksi katarak dengan metode Support Vector Machine (SVM) dan Convolutional Neural Network (CNN). Kedua metode ini digunakan sebagai metode klasifikasi atas kelas mata normal, katarak imatur dan katarak matur. Data masukan berupa citra digital yang sudah di *resize* menjadi 64x64 *pixel* berformat *Joint Photographic Group* (JPG) yang diperoleh dari penelitian sebelumnya. Metode *Support Vector Machine* (SVM) dan *Convolutional Neural Network* (CNN) yang digunakan dalam proses deteksi penyakit katarak memberikan hasil yang optimal. Metode Support Vector Machine (SVM) sendiri menghasilkan nilai *accuracy* sebesar 96.67%, Sedangkan untuk Metode Convolutional Neural Network menghasilkan nilai *accuracy* yang lebih baik sebesar 98.89%.

Kata Kunci: Katarak, Citra Digital, *Machine Learning*, *Support Vector Machine*, *Convolutional Neural Network*

1. PENDAHULUAN

Mata adalah salah satu indera terpenting bagi manusia. Melalui mata, manusia menyerap informasi visual yang digunakan untuk melakukan berbagai aktivitas. Namun gangguan terhadap penglihatan banyak terjadi, mulai dari gangguan yang bersifat ringan hingga gangguan yang berat yang dapat mengakibatkan kebutaan. Salah satu penyebab dari gangguan penglihatan ini adalah katarak. Katarak merupakan kelainan lensa mata yang keruh di dalam bola mata. Katarak terjadi akibat kekeruhan pada lensa mata yang mengakibatkan terganggunya cahaya masuk ke dalam bola mata, sehingga penglihatan menjadi kabur dan lama kelamaan dapat menyebabkan kebutaan[1]. Katarak merupakan penyebab utama kebutaan yang terjadi di Indonesia. Berdasarkan Hasil Survey Kebutuhan *Rapid Assessment of Avoidable Blindness* (RAAB) tahun 2014 - 2016 oleh Persatuan Dokter Spesialis Mata Indonesia (Perdami) dan Badan Litbangkes Kementerian Kesehatan di 15 Provinsi, diketahui angka kebutaan mencapai 3% dan katarak menjadi penyebab kebutaan tertinggi (81%)[2].

Kebutaan yang diakibatkan oleh katarak merupakan kebutaan yang dapat disembuhkan melalui operasi dengan biaya yang tidak terlalu mahal. Besarnya proporsi kebutaan yang diakibatkan oleh katarak menunjukkan bahwa masih banyaknya penderita katarak yang belum dioperasi. Alasan utama kecilnya angka operasi katarak di Indonesia disebabkan oleh banyak faktor, salah satunya adalah masyarakat tidak mengetahui kalau dirinya menderita katarak. Hal ini sendiri dipengaruhi oleh kurangnya penyebaran dokter mata di Indonesia sehingga tidak mencakup keseluruhan provinsi di Indonesia. Semakin lama seorang pasien menderita katarak atau menerima perawatan yang terlambat, maka akan semakin parah kerusakan yang akan terjadi pada penglihatan penderita katarak.

Dikutip dari IBM, *machine learning* merupakan cabang dari *Artificial Intelligence* yang berfokus untuk membuat sistem atau algoritma yang terus belajar dari data dan meningkatkan akurasi nya dari waktu ke waktu[3]. Algoritma *machine learning* sudah digunakan untuk menyelesaikan berbagai macam permasalahan dan banyak sekali digunakan dalam bidang sains. Dalam mengimplementasikan *machine learning*, ada banyak algoritma yang bisa digunakan untuk melakukan klasifikasi diantaranya adalah *Support Vector Machine* (SVM) dan *Convolutional Neural Network* (CNN). Algoritma *Support Vector Machine* (SVM) merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi dan regresi. SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik klasifikasi lainnya[4]. Sedangkan *Convolutional Neural Network* (CNN) adalah sebuah teknik yang terinspirasi dari cara mamalia menghasilkan persepsi visual. CNN biasanya digunakan pada data citra dan digunakan untuk mendeteksi dan mengklasifikasi objek pada sebuah gambar[5].

Pada penelitian sebelumnya oleh Vaibhav Agarwal, Vaibhav Gupta, Vivasvan Manasvi Vashisht, Kiran Sharma, dan Neetu Sharma pada sebuah paper yang berjudul *Mobile Application Based Cataract Detection System*[6], telah dilakukan sebuah penelitian penerapan *machine learning* untuk mendeteksi katarak. Penelitian ini menggunakan 3 jenis algoritma *machine learning* yang berbeda, yaitu KNN, SVM dan Naïve Bayes. Kekurangan dari penelitian ini adalah perbandingan yang dilakukan hanya berfokus pada algoritma *machine learning* dan tidak membandingkan performansi dengan algoritma *deep learning*.

Melalui penelitian sebelumnya, maka dikembangkan lah analisis sistem deteksi katarak dengan menggunakan dua algoritma yang berbeda yakni algoritma *Support Vector Machine* (SVM) dan algoritma *Convolutional Neural Network*. Pada klasifikasi ini, hasilnya akan dibagi menjadi tiga kelas yaitu mata normal, katarak imatur dan katarak matur. Penggunaan kedua algoritma ini ditujukan untuk mencari algoritma terbaik untuk metode deteksi mata katarak. Kedua algoritma ini juga nantinya akan diimplementasikan berbasis *website* dan aplikasi android untuk menjangkau pengguna yang lebih luas agar katarak bisa dideteksi secara dini.

2. TINJAUAN PUSTAKA

2.1. Katarak

Katarak adalah kekeruhan alami lensa *intraocular* yang memfokuskan cahaya yang masuk ke mata pada retina. Kekeruhan ini dapat mempengaruhi penglihatan dan dapat menyebabkan kebutaan jika tidak ditangani. Katarak sering berkembang secara perlahan dan tanpa rasa sakit, sehingga penglihatan dan gaya hidup dapat terpengaruh tanpa seseorang sadari[7]. Katarak adalah salah satu penyebab utama gangguan penglihatan, terhitung 47,9% dari kebutaan di seluruh dunia. Diperkirakan lebih dari setengah orang yang berusia di atas 65 tahun memiliki beberapa derajat perkembangan katarak pada satu atau kedua mata[8].

Katarak umumnya berkembang seiring waktu sejak seseorang menginjak usia 40-50 tahun. Pada awalnya pasien mungkin tidak menyadari adanya gangguan penglihatan karena lensa masih bisa berfungsi dengan baik meskipun katarak telah terbentuk. Namun seiring bertambah usia, katarak akan semakin parah dan menimbulkan beberapa gejala seperti pandangan kabur dan kabur, mata akan menjadi lebih sensitif saat melihat cahaya terang hingga munculnya halo saat melihat sumber cahaya.

2.2. Support Vector Machine

Support Vector Machine (SVM) adalah sebuah teknik yang digunakan untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop Computational Learning Theory (COLT). SVM memiliki prinsip dasar *linear classifier* yaitu kasus klasifikasi yang secara *linear* dapat dipisahkan, namun SVM telah dikembangkan agar dapat bekerja pada problem *non-linear* dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi. Dalam ruang berdimensi tinggi ini, SVM mencoba mencari *hyperplane* terbaik yang dapat memaksimalkan jarak (margin) antar lapisan data[9].

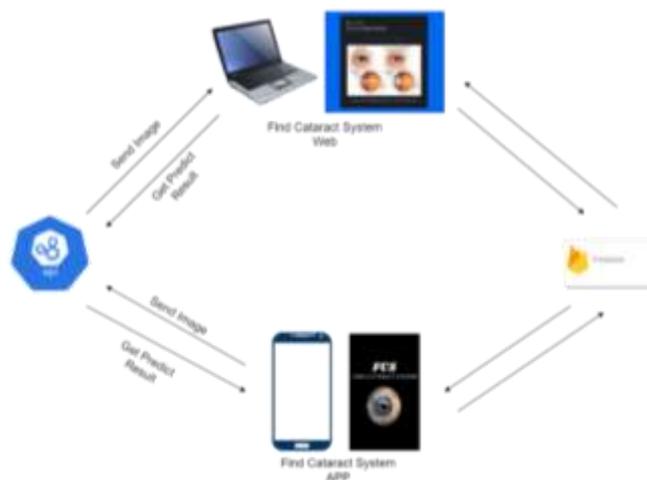
2.3. Convolutional Neural Network

Convolution Neural Network (CNN) adalah pengembangan dari *Multi-Layer Perceptron (MLP)* yang dirancang untuk memproses data dua dimensi. CNN termasuk dalam kategori *deep learning* karena kedalaman jaringannya dan banyak diterapkan pada data citra. *Multi-Layer Perceptron* sendiri tidak cocok untuk digunakan dalam klasifikasi citra karena ketidakmampuannya untuk menyimpan informasi spasial dari data citra dan memperlakukan setiap piksel sebagai fitur independen, yang mengarah pada hasil yang kurang baik[10]. Arsitektur CNN terdiri dari tiga jenis: *Convolutional Layer*, *Pooling Layer*, dan *Fully-Connected Layer*

3. METODOLOGI PENELITIAN

3.1. Desain Sistem

Sistem deteksi katarak di rancang untuk mempermudah deteksi katarak secara dini oleh pengguna. Sistem deteksi katarak ini tersedia dalam bentuk *website* dan aplikasi *android*. Konsep dari sistem dibuat agar pengguna bisa melakukan deteksi dini pada matanya untuk mendeteksi apakah pengguna menderita penyakit katarak dan seberapa parah kondisi dari matanya.

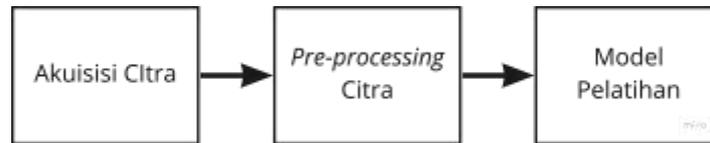


Gambar 1. Desain Sistem

3.2. Diagram Alir Sistem Deteksi Katarak

Pada tugas akhir ini, akan dirancang sebuah sistem pengolahan citra digital untuk mendeteksi penyakit mata katarak. Pembuatan sistem yang dapat mendeteksi penyakit katarak dirancang menggunakan metode *Support Vector Machine (SVM)* dan *Convolutional Neural Network (CNN)*. Klasifikasi pada penelitian ini terbagi

menjadi tiga kelas yaitu, citra mata normal, katarak imatur dan katarak matur. Proses secara umum tersebut dapat digambarkan dalam diagram alir berikut:



Gambar 2. Diagram Alir Sistem

3.3. Akuisisi Citra

Pada penelitian ini, dataset yang akan digunakan adalah dataset sekunder yang didapatkan dari penelitian sebelumnya berupa citra mata [7]. Data yang digunakan berjumlah 150 citra mata dan terbagi menjadi 50 citra mata normal, 50 citra mata katarak imatur dan 50 citra mata katarak matur dalam format .JPEG. Untuk contoh dataset dari ketiga kelas yang ada dapat dilihat pada Gambar 3.



Gambar 3. Klasifikasi Citra Penyakit Katarak

3.4. Pre-processing

Pada proses *pre-processing*, dilakukan operasi augmentasi data (*horizontal flip* dan *random brightness*) yang bertujuan memperbanyak jumlah data yang akan digunakan untuk pelatihan model sehingga performansi model yang didapat semakin baik. Dataset yang telah di augmentasi secara keseluruhan berjumlah 450 citra dengan komposisi tiap kelas sebanyak 150 citra yang berukuran 512×512 piksel dan dilakukan tahapan *resize* menjadi 64×64 piksel agar mempercepat proses komputasi pada sistem yang dibangun.

3.5. Model Pelatihan

Pada penelitian ini, terdapat dua buah model klasifikasi yang akan dibandingkan satu dengan yang lain. Model yang akan diuji adalah *Support Vector Machine* (SVM) dengan *Convolutional Neural Network* (CNN).

3.5.1. Support Vector Machine (SVM)

Pada metode *Support Vector Machine*, Citra mata yang sudah melalui tahap *pre-processing* akan melewati tahapan *feature extraction* terlebih dahulu. Fungsi dari tahap *feature extraction* ini adalah untuk melewatkan suatu objek tertentu dengan meminimalisir ciri yang tidak penting. Pada penelitian ini *feature extraction* yang digunakan adalah *Gabor Wavelet*. Tujuan utama dari *Gabor Wavelet* adalah untuk mengekstraksi fitur dari citra yang telah di konvolusi terhadap kernel. Dengan menggunakan orientasi yang berbeda-beda dapat menghasilkan filter yang akan digunakan dalam tahap ini. Setelah mengekstraksi fitur dari citra mata, dilakukan tahap klasifikasi dengan algoritma *Support Vector Machine* (SVM). Tahap klasifikasi dimulai dengan memasukkan hasil dari *feature extraction* dan menentukan jenis kernel yang akan digunakan beserta nilai parameter C (*Cost*) dan γ (*Gamma*). Pada penelitian ini, Fungsi kernel yang akan digunakan adalah *Linear*, *Polynomial* dan *Radial Basis Function* (RBF).

3.5.2. Convolutional Neural Network (CNN)

Pada metode *Convolutional Neural Network* (CNN), Citra mata yang sudah melalui tahapan *pre-processing* akan menjadi *input* bagi model *Convolutional Neural Network* (CNN). Selanjutnya, input citra akan masuk ke dalam tahap *feature extraction learning* dimana akan dilakukan proses *convolution*, *pooling* dan aktivasi ReLU. Dalam penelitian ini, digunakan 4 *Convolutional Layer* dengan fungsi *max-pooling* dan aktivasi ReLU di setiap lapisannya. Citra yang sudah melewati tahap *feature extraction learning* akan masuk pada tahap klasifikasi. Tahap klasifikasi dimulai dengan proses *flatten* untuk mengubah feature map yang sudah didapat menjadi sebuah vektor 1 dimensi. Selanjutnya hasil dari *flatten* akan diteruskan menuju *Dense Layer*. Proses

ini bisa disebut juga dengan *Fully Connected Layer*. Terakhir, fungsi aktivasi *softmax* akan digunakan untuk mengklasifikasikan citra menjadi tiga keluaran yaitu normal, matur, dan imatur.

3.6. Parameter Performansi

Tahap berikutnya adalah tahap performansi sistem. Tahap ini memiliki empat hal yang digunakan sebagai acuan dalam menentukan baik buruknya suatu sistem, yaitu *accuracy*, *recall*, *precision*, dan *f1-score*

- Accuracy

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* merupakan rasio prediksi benar (positif dan negatif) dari keseluruhan data[11].

$$Accuracy(\%) = \frac{TP + FN}{TP + TN + FP + FN} * 100\% \quad (1)$$

- Precision

Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model[11].

$$Precision(\%) = \frac{TP}{TP + FP} * 100\% \quad (2)$$

- Recall

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif[11].

$$Recall(\%) = \frac{TP}{TP + FN} * 100\% \quad (3)$$

- F1-Score

F1-Score digunakan untuk mempermudah analisis dengan penggabungan nilai *recall* dan *precision* ke dalam variabel nilai baru, karena ada kecenderungan jika nilai *recall* tinggi maka nilai *precision* rendah begitupun sebaliknya. Nilai terbaik *F1-Score* adalah 1.0 dan nilai terburuknya adalah 0.

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right) \quad (4)$$

Keterangan:

TP (*True Positive*) = Jumlah data positif yang terdeteksi benar.

FP (*False Positive*) = Jumlah data negatif namun terdeteksi sebagai data positif.

FN (*False Negative*) = Jumlah data positif yang terdeteksi sebagai data negatif.

TN (*True Negative*) = Jumlah data negatif yang terdeteksi sebagai data negatif

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Jumlah Data Latih dan Data Uji

Pengujian pertama adalah pengujian terhadap perbandingan jumlah data latih dan data uji yang digunakan. Perbandingan jumlah data latih dan data uji yang digunakan adalah 60:40, 70:30 dan 80:20.

Tabel 1. Hasil Pengujian Pengaruh Rasio Data terhadap Model SVM & CNN

Model	Rasio	Accuracy	Precision	Recall	F1-Score
SVM	60:40	87.22	87.93	87.22	87.35
	70:30	85.19	85.50	85.19	85.20
	80:20	93.33	93.62	93.33	93.36
CNN	60:40	91.11	91.58	91.11	91.22
	70:30	88.15	88.26	88.15	87.83
	80:20	96.67	96.84	96.67	96.62

Berdasarkan Tabel 1, Penggunaan rasio 80:20 menghasilkan performansi terbaik dengan model SVM mendapatkan nilai *accuracy* 93.33%, *precision* 93.62%, *recall* 93.33%, *f1-score* 93.36% dan model CNN mendapatkan nilai *accuracy* 96.67%, *precision* 96.84%, *recall* 96.67%, *f1-score* 96.62%. Hal ini menunjukkan bahwa semakin banyak data selama proses pelatihan, maka tingkat akurasi yang didapatkan oleh model semakin optimal. Sebaliknya, jika data selama proses pelatihan semakin sedikit, maka model tidak menghasilkan hasil akurasi yang optimal

4.2. Pengujian Parameter Model SVM

Pengujian kedua adalah pengujian parameter pada model Support Vector Machine (SVM). Pada pengujian ini, digunakan 3 fungsi kernel yaitu Linear, Polynomial, dan Radial Basis Function (RBF). Parameter yang diuji dalam penelitian ini adalah Parameter $C(Cost)$ dan Parameter $\gamma (Gamma)$.

4.2.1. Pengaruh Parameter $C(Cost)$ terhadap Kernel SVM

Pengujian ini ditujukan untuk mengetahui nilai $C(Cost)$ yang optimal dan mengetahui pengaruh dari nilai $C(Cost)$ terhadap performansi dari model yang digunakan. Dalam pengujian ini, digunakan 3 nilai $C(Cost)$ yang berbeda yaitu 1.0, 0.1 dan 0.001

Tabel 2. Hasil Pengujian Pengaruh Parameter $C(Cost)$ terhadap Kernel SVM

Kernel	C	Accuracy	Precision	Recall	F1-Score
Linear	1.0	96.67	96.84	96.67	96.62
	0.1	96.67	96.84	96.67	96.62
	0.01	96.67	96.84	96.67	96.62
Polynomial	1.0	95.56	96.08	95.56	95.62
	0.1	84.44	86.37	84.44	84.67
	0.01	55.56	73.09	55.56	51.53
RBF	1.0	93.33	93.62	93.33	93.36
	0.1	62.22	70.40	62.22	61.54
	0.01	62.22	70.40	62.22	61.54

Berdasarkan Tabel 2, Nilai $C(Cost) = 1.0$ menghasilkan performansi terbaik dengan model SVM dengan fungsi kernel *Linear* mendapatkan nilai *accuracy* 96.67%, *precision* 96.84%, *recall* 96.67%, *f1-score* 96.62%, Model SVM dengan fungsi kernel *Polynomial* mendapatkan nilai *accuracy* 95.56%, *precision* 96.08%, *recall* 95.56%, *f1-score* 95.62%, dan Model SVM dengan fungsi kernel *Radial Basis Function* (RBF) mendapatkan nilai *accuracy* 93.33%, *precision* 93.62%, *recall* 93.33%, *f1-score* 93.36%. Dapat disimpulkan bahwa penggunaan parameter $C(Cost)$ bernilai 1.0 menghasilkan performansi yang optimal untuk model yang digunakan.

4.2.2. Pengaruh Parameter $\gamma (Gamma)$ terhadap Kernel SVM

Pengujian ini ditujukan untuk mengetahui nilai $\gamma (Gamma)$ yang optimal dan mengetahui pengaruh dari nilai $\gamma (Gamma)$ terhadap performansi dari model yang digunakan. Dalam pengujian ini, digunakan 3 nilai $\gamma (Gamma)$ yang berbeda yaitu 1.0e-1, 1.0e-3 dan 1.0e-6

Tabel 3. Hasil Pengujian Pengaruh Parameter Gamma terhadap Kernel SVM

Kernel	$\gamma (Gamma)$	Accuracy	Precision	Recall	F1-Score
Linear	1.0e-1	96.67	96.84	96.67	96.62
	1.0e-3	96.67	96.84	96.67	96.62
	1.0e-6	96.67	96.84	96.67	96.62
Polynomial	1.0e-1	95.56	96.08	95.56	95.62
	1.0e-3	95.56	96.08	95.56	95.62
	1.0e-6	55.56	73.39	55.56	50.36
RBF	1.0e-1	64.44	82.80	64.44	64.12
	1.0e-3	93.33	93.62	93.33	93.36
	1.0e-6	56.67	68.39	56.67	54.38

Berdasarkan Tabel 3, Nilai γ (*Gamma*) sebesar $1.0e-3(0.001)$ menghasilkan performansi terbaik dengan model SVM dengan fungsi kernel *Linear* mendapatkan nilai *accuracy* 96.67%, *precision* 96.84%, *recall* 96.67%, *f1-score* 96.62%, *Polynomial* mendapatkan nilai *accuracy* 95.56%, *precision* 96.08%, *recall* 95.56%, *f1-score* 95.62%, dan Model SVM dengan fungsi kernel *Radial Basis Function* (RBF) mendapatkan nilai *accuracy* 93.33%, *precision* 93.62%, *recall* 93.33%, *f1-score* 93.36%. Dapat disimpulkan bahwa penggunaan parameter γ (*Gamma*) bernilai $1.0e-3(0.001)$ menghasilkan performansi yang optimal untuk model yang digunakan.

4.3. Pengaruh Parameter Model CNN

Pengujian ketiga adalah pengujian parameter pada model Convolutional Neural Network (CNN). Pada pengujian ini, dilakukan 3 pengujian terhadap parameter model CNN yaitu *Optimizer*, *Learning Rate*, dan *Epoch*.

4.3.1. Pengaruh *Optimizer* yang digunakan terhadap Model CNN

Pengujian ini ditujukan untuk mengetahui jenis optimizer yang cocok dengan model yang digunakan dan mengetahui pengaruh jenis *optimizer* terhadap performansi sistem. Dalam pengujian ini, digunakan 3 jenis *optimizer* yang berbeda yaitu Adam, Adamax dan RMSprop.

Tabel 4. Hasil Pengujian Pengaruh *Optimizer* terhadap Model CNN

<i>Optimizer</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Adam	98.89	98.92	98.89	98.89
Adamax	88.89	91.03	88.89	89.05
RMSprop	96.67	96.97	96.67	96.66

Berdasarkan **Tabel 4**, Penggunaan *Optimizer* Adam menghasilkan performansi terbaik dengan nilai *accuracy* 98.89%, *precision* 98.92%, *recall* 98.89%, *f1-score* 98.89%. Dapat disimpulkan bahwa penggunaan *Optimizer* Adam menghasilkan performansi yang optimal untuk model yang digunakan.

4.3.2. Pengaruh *Learning Rate* terhadap Model CNN

Pengujian ini ditujukan untuk mengetahui nilai *learning rate* yang optimal dan mengetahui pengaruh dari nilai *learning rate* terhadap performansi dari model yang digunakan. Dalam pengujian ini, digunakan 4 nilai *learning rate* yang berbeda yaitu 0.1, 0.01, 0.001, dan 0.0001

Tabel 5. Hasil Pengujian Pengaruh *Learning Rate* terhadap Model CNN

<i>Learning Rate</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.1	33.33	11.11	33.33	16.67
0.01	93.33	93.84	93.33	93.23
0.001	98.89	98.92	98.89	98.89
0.0001	86.67	86.96	86.67	86.71

Berdasarkan Tabel 6, Penggunaan jumlah *Epoch* sebesar 25 menghasilkan performansi terbaik dengan nilai *accuracy* 98.89%, *precision* 98.92%, *recall* 98.89%, *f1-score* 98.89%. Dapat disimpulkan bahwa penggunaan jumlah *Epoch* sebesar 25 menghasilkan performansi yang optimal untuk model yang digunakan.

4.3.3. Pengaruh *Epoch* terhadap Model CNN

Pengujian ini ditujukan untuk mengetahui jumlah *epoch* yang optimal dan mengetahui pengaruh dari jumlah *epoch* terhadap performansi dari model yang digunakan. Dalam pengujian ini, digunakan 3 jumlah *epoch* yang berbeda yaitu 10, 25 dan 50.

Tabel 6. Hasil Pengujian Pengaruh Jumlah *Epoch* terhadap Model CNN.

<i>Epoch</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
10	90.00	92.31	90.00	89.96
25	98.89	98.92	98.89	98.89
50	97.78	97.78	97.78	97.78

Berdasarkan Tabel 6, Penggunaan jumlah *Epoch* sebesar 25 menghasilkan performansi terbaik dengan nilai *accuracy* 98.89%, *precision* 98.92%, *recall* 98.89%, *f1-score* 98.89%. Dapat disimpulkan bahwa penggunaan jumlah *Epoch* sebesar 25 menghasilkan performansi yang optimal untuk model yang digunakan.

4.4. Perbandingan Performansi SVM dengan CNN

Untuk mengetahui dan memverifikasi performansi model dalam mendeteksi katarak, dilakukan perbandingan antara model Support Vector Machine (SVM) dan Convolutional Neural Network (CNN).

4.4.1. Perbandingan Performansi Klasifikasi SVM dengan CNN

Perbandingan ini ditujukan untuk mengetahui performansi kedua model dalam melakukan klasifikasi. Perbandingan ini menggunakan *test data* yang sama berjumlah 90 citra dengan 30 citra dalam masing-masing kelasnya. Hasil perbandingan kedua model tersebut ditunjukkan pada Gambar 4.



Gambar 4. Grafik Perbandingan Performansi Klasifikasi SVM dengan CNN

Pada Gambar 4, dapat diketahui bahwa model *Convolutional Neural Network* (CNN) memiliki nilai performansi yang lebih baik jika dibandingkan dengan model *Support Vector Machine*.

4.5. Perbandingan Waktu Komputasi SVM dengan CNN

Perbandingan ini ditujukan untuk mengetahui performansi waktu komputasi dari kedua model dalam melakukan klasifikasi. Perbandingan ini dilakukan sejumlah *test data* yaitu 90 citra dan menghitung rata-rata waktu komputasi yang dibutuhkan untuk kedua model dalam melakukan klasifikasi. Hasil perbandingan kedua model tersebut ditunjukkan pada Gambar 5.



Gambar 5. Grafik Perbandingan Waktu Komputasi SVM dengan CNN

Pada Gambar 5, dapat diketahui bahwa model *Convolutional Neural Network* (CNN) memiliki waktu komputasi yang lebih baik jika dibandingkan dengan model *Support Vector Machine* (SVM).

4.5.1. Perbandingan Performansi Klasifikasi SVM dengan CNN

Perbandingan ini ditujukan untuk mengetahui penggunaan memori dari kedua model dalam melakukan klasifikasi. Perbandingan ini dilakukan sejumlah *test data* yaitu 90 citra dan menghitung rata-rata penggunaan memori yang dibutuhkan untuk kedua model dalam melakukan klasifikasi. Hasil perbandingan kedua model tersebut ditunjukkan pada Gambar 6.



Gambar 6. Grafik Perbandingan Performansi SVM dengan CNN

Pada Gambar 6, dapat diketahui bahwa model *Support Vector Machine* (SVM) menggunakan memori lebih sedikit jika dibandingkan dengan model *Convolutional Neural Network* (CNN). Penggunaan memori yang lebih sedikit akan meminimalisir biaya dan konsumsi daya *server*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang dilakukan pada sistem deteksi katarak menggunakan metode *Support Vector Machine* (SVM) dan *Convolutional Neural Network* (CNN), didapatkan kesimpulan bahwa sistem sudah berfungsi secara optimal dalam mendeteksi 3 jenis kondisi mata, yaitu mata normal, katarak imatur dan katarak matur dimana dengan metode SVM hanya terdapat 3 kesalahan deteksi pada kelas mata normal dan dengan metode CNN hanya terdapat 1 kesalahan deteksi pada kelas mata normal. Metode SVM mendapatkan performansi terbaik saat menggunakan rasio pembagian data 80:20, *linear* kernel, dengan parameter *C* (*Cost*) berjumlah 1 dan nilai γ (*Gamma*) sebesar $1.0e-3$ dengan sistem mencapai *accuracy* sebesar 96.67%, *precision* 96.84%, *recall* 96.67% dan *f1-score* 96.62% sedangkan metode CNN mendapatkan performansi terbaik saat menggunakan rasio pembagian data 80:20, *optimizer* Adam, dengan nilai *learning rate* sebesar 0.001 dan *epoch* berjumlah 25 dengan sistem mencapai nilai *accuracy* sebesar 98.89%, *precision* 98.92%, *recall* 98.89% dan *f1-score* 98.89%. Hasil performansi sistem terbaik berdasarkan keseluruhan skenario pengujian didapatkan dengan menggunakan metode CNN saat menggunakan rasio pembagian data 80:20, *optimizer* Adam, dengan nilai *learning rate* sebesar 0.001 dan *epoch* berjumlah 25 dengan sistem mencapai nilai *accuracy* sebesar 98.89%, *precision* 98.92%, *recall* 98.89% dan *f1-score* 98.89%

DAFTAR PUSTAKA

- [1] Sidarta Ilyas, *Ilmu penyakit mata*, Fifth. Jakarta: Badan Penerbit FKUI., 2005.
- [2] F. Ismandari, "Situasi Gangguan Penglihatan," *Pusat Data dan Informasi Kementerian Kesehatan RI*, Jakarta, 2018.
- [3] IBM Cloud Education, "What is Machine Learning?," *IBM*, Jul. 15, 2020. <https://www.ibm.com/cloud/learn/machine-learning> (accessed Oct. 22, 2021).
- [4] Samsudiney, "Penjelasan Sederhana tentang Apa Itu SVM?," *Medium*, Jul. 25, 2019. <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02> (accessed Oct. 25, 2021).
- [5] Q. Lina, "Apa itu Convolutional Neural Network?," *Medium*, Jan. 02, 2019. <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4> (accessed Oct. 25, 2021).

- [6] V. Agarwal, V. Gupta, V. M. Vashisht, K. Sharma, and N. Sharma, "Mobile Application Based Cataract Detection System," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 780–787. doi: 10.1109/ICOEI.2019.8862774.
- [7] B. H. Feldman and S. Heersink, "Cataract," *American Academy of Ophthalmology*, May 01, 2011.
- [8] E. Nash, "Cataracts," *InnovAiT*, vol. 6, no. 9, pp. 555–562, Sep. 2013.
- [9] B. Santosa, *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu, 2007.
- [10] I. W. S. E. Putra, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, Mar. 2016, doi: 10.12962/j23373539.v5i1.15696.
- [11] K. S. Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning," *Medium*, Nov. 13, 2019. <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-supervised-machine-learning-bc4b1ae9ae3f> (accessed Jul. 21, 2022).