Indonesian Journal of Data and Science (IJODAS) ISSN: 2715-9930
Vol 2, No 1, March 2021, pp. 01-12

Analisis Klasifikasi Dataset Citra Penyakit *Pneumonia* Menggunakan Metode K-*Nearest Neighbor* (KNN)

Andi Ainun Dzariah Halim[a,1], Siska Anraeni[a,2]

[a] *Program Studi Teknik Informatika, Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM.05, Makassar dan 90231, Indonesia*
[1]*13020160201@umi.ac.id;* [2]*siska.anraeni@umi.ac.id;*

INFORMASI ARTIKEL

Diterima : 18 – 12 – 2020
Direvisi : 28 – 02 – 2021
Diterbitkan : 31 – 03 – 2021

Kata Kunci:
Deteksi Tepi Sobel
Ekstraksi Fitur *Moment Invariant*
K-Nearest Neighbor
Akurasi
Penyakit Pneumonia

ABSTRAK

Pneumonia adalah peradangan paru yang menyebabkan nyeri saat bernafas dan keterbatasan *intake* oksigen. Pneumonia dapat disebabkan oleh bakteri, virus, dan jamur. Penelitian ini menggunakan 1000 dataset citra. Dataset citra tersebut dikelola oleh Paul Mooney yang dikumpulkan dari pasien anak berusia 1-5 tahun di Guangzhou Women and Children's Medical Center, Guangzhou pada 22 Maret 2018 hingga 25 Maret 2018. Penelitian ini bertujuan untuk menganalisis nilai performa (akurasi, presisi, *recall*, dan *f-measure*) pada proses klasifikasi dataset citra penyakit pneumonia dan tidak *pneumonia*. Tahapan yang dilakukan yaitu membagi dataset dengan berbagai simulasi rasio, deteteksi tepi sobel, ektraksi fitur *moment invariant*, klasifikasi metode KNN, nilai K=2 sampai K=900. Hasil Penelitian menunjukkan performa terbaik terdapat pada simulasi rasio 20:80 dengan memperoleh nilai akurasi 96%, presisi 97%, recall 97%, f-measure 97% dengan menggunakan nilai K=3.

I. Pendahuluan

Pneumonia adalah peradangan paru yang menyebabkan nyeri saat bernafas dan keterbatasan intake oksigen. *Pneumonia* dapat disebabkan oleh bakteri, virus, dan jamur. Virus penyebab pneumonia adalah *Respiratory Syncial Virus* (RVS). Virus ini kebanyakan menyerang saluran pernapasan bagian atas, pada balita gangguan ini bisa memicu *pneumonia*.[1] Penyakit *pneumonia* merupakan salah satu penyakit yang dianggap serius di Indonesia. Sebab, dari tahun ke tahun penyakit *pneumonia* selalu berada di peringkat atas dalam daftar penyakit penyebab kematian bayi dan balita. Berdasarkan hasil Riskesdas 2007, pneumonia menduduki peringkat kedua pada proporsi penyebab kematian anak umur 1-4 tahun. Oleh karena itu terlihat bahwa penyakit *pneumonia* menjadi masalah kesehatan yang utama di Indonesia. Jumlah kasus *pneumonia* pada balita (< 5 tahun) lebih tinggi dibandingkan dengan usia ≥ 5 tahun. Pada tahun 2007 dan 2008, perbandingan kasus *pneumonia* pada dua kelompok umur tersebut yaitu 7:3. Artinya bila terdapat 7 kasus *pneumonia* pada anak umur < 5 tahun, maka akan terdapat 3 kasus *pneumonia* pada anak ≥ 5 tahun. Pada tahun 2009 perbandingan tersebut berubah menjadi 6:4. Walaupun demikian tetap dapat disimpulkan bahwa proporsi kasus *pneumonia* pada kelompok umur balita menjadi yang terbesar.[2]

Pengolahan citra digital sekarang berkembang cepat, dan dapat digunakan di dalam dunia medis seperti menganalisis gambar *rontgen*, sehingga dapat memecahkan permasalahan analisis citra. Hasil citra *rontgen* sering nampak kabur, kurang kontras, dan sebagainya, sehingga satu citra diamati oleh beberapa pengamat dapat menghasilkan pembacaan yang berbeda-beda. Buruknya hasil visualisasi citra *rontgen* disebabkan karena sedikitnya perbedaan redaman sinar-X. Untuk mengatasi masalah tersebut digunakan pengolahan citra untuk meningkatkan dan memperbaiki mutu citra. Kemudian *region* paru ini dilakukan deteksi tepi berbasis operator sobel. Dalam penelitian sebelumnya ditemukan perbandingan *pixel* antara hasil deteksi tepi terhadap *region* paru, yang mengklasifikasikan 6 (enam) jenis penyakit paru dengan *interval persentase* untuk penyakit bronkitis sebesar 1,43% - 1,59%, penyakit pleuritis 1,43% - 1,59%, penyakit *pneumonia* 2,00% - 2,50%, penyakit TBC 2,86% - 3,79%, penyakit emfisema 4,16% - 4,76% dan penyakit kanker paru 76,72% - 94,85%.[3]

Segmentasi merupakan proses yang penting dalam pengolahan citra yang bertujuan untuk memecahkan suatu citra ke dalam beberapa segmentasi dengan suatu kriteria tertentu. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai input. Jenis operasi ini berkaitan erat dengan pengenalan pola.[4]

Salah satu bagian dari segmentasi citra adalah deteksi tepi. Deteksi tepi merupakan langkah awal dari segmentasi citra untuk mendapatkan informasi dalam citra. Tepi berisi kumpulan dari titik yang mempunyai

perbedaan tinggi dengan yang lainnya. Hasil dari deteksi tepi adalah garis batas dari tingkat kecerahan yang berbeda dari suatu objek yang berada dalam citra. Banyak metode yang digunakan untuk melakukan deteksi tepi citra, dengan menggunakan metode deteksi tepi *canny* dan sobel, didapatkan hasil analisa keluaran program meningkatkan dan mendeteksi tepi gambar akan lebih baik jika masukan gambar mempunyai banyak tekstur, hasil dari gambar keluaran ditentukan dengan faktor perkalian kernel (*Marks*). Dari hasil analisa keluaran program *operator canny* dan operator sobel sama-sama mendeteksi tepi citra dengan baik.[4]

Proses ekstraksi fitur merupakan proses yang bertujuan untuk membentuk fitur atau ciri dari suatu objek. Selanjutnya untuk mengenali pola atau mengelompokkan suatu objek tertentu maka dilakukan proses pencocokan antara objek tersebut dengan objek-objek yang cirinya telah dibentuk oleh sistem (Septiarini, 2012). Penelitian sebelumnya telah melakukan klasifikasi daun dengan perbaikan fitur citra menggunakan ekstraksi fitur moment invariant, dengan metode klasifikasi yaitu K-*Nearest Neighbor* karena metode ini dikenal cepat dalam data pelatihan, efektif untuk data pelatihan besar, sederhana dan mudah dipelajari. Hasil pengujian klasifikasi daun dari citra yang ada pada dataset didapatkan nilai akurasi sekitar 86,67%.[5]

Algoritma K-*Nearest Neighbor* (K-NN) merupakan algoritma klasifikasi berdasarkan kedekatan jarak suatu data dengan data yang lain. Pada algoritma K-NN, data berdimensi q, jarak dari data tersebut ke data yang lain dapat dihitung. Nilai jarak inilah yang digunakan sebagai nilai kedekatan/kemiripan antara data uji dengan data latih. Nilai K pada K-NN berarti K-data terdekat dari data uji. Untuk menangani masalah efektifitas dan akurasi dalam mendeteksi penyakit jantung maka dibuatlah sistem pendeteksi penyakit jantung menggunakan algoritma klasifikasi K-*Nearest Neighbor* (KNN).[6]

Penelitian ini mencoba untuk mencari nilai performa terbaik dari dataset citra penyakit pneumonia dengan berbagai simulasi rasio yang terdiri dari rasio perbandingan 20:80 (200 citra penyakit pneumonia, 800 citra tidak pneumonia), 50:50 (500 citra penyakit pneumonia, 500 citra tidak pneumonia), 80:20 (800 citra pneumonia, 200 citra tidak pneumonia) serta beragam nilai K. Analisis dilakukan dengan menghitung performa (akurasi, presisi, *recall*, dan *f-measure*) dengan metode K-*Nearest Neighbor* (KNN) pada objek tersebut.

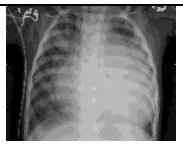
II. Metode

A. Dataset

Dataset citra penyakit pneumonia dan tidak pneumonia diperoleh dari *Kaggle dataset library*. Dataset citra *Chest X-ray* penyakit pneumonia di publikasikan oleh Paul Mooney pada tanggal 22 Maret 2018 dan terakhir diperbaharui di kaggle pada tanggal 25 Maret 2018, data tersebut dikumpulkan dari pasien anak berusia 1-5 tahun di Guangzhou Women and Children's Medical Center, Guangzhou. Tabel 1 menunjukkan contoh dataset citra penyakit pneumonia:

Tabel 1. Dataset Citra Penyakit Pneumonia

Id	Citra	Ket	Id	Citra	Ket
0		*Pneumonia*	5		Tidak Pneumonia
1		*Pneumonia*	6		Pneumonia
2		*Pneumonia*	7		Tidak Pneumonia
3		Tidak Pneumonia	8		Pneumonia
4		Pneumonia	9		Tidak Pneumonia

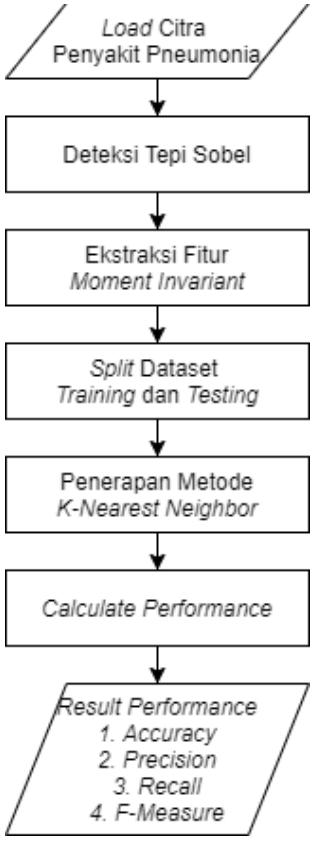
…	…	…	…	…	…
…	…	…	…	…	…
996		Tidak Pneumonia	998		Tidak Pneumonia
997		Pneumonia	999		Pneumonia

B. Perancangan Sistem

Dalam penelitian ini secara garis besar langkah-langkahnya terdiri dari load dataset, memilah data set menjadi data training dan data testing kemudian tahap persentasi data training kedalam implementasi metode K-*Nearest Neighbor* (KNN)[5] pada data *testing* kemudian menghitung nilai akurasi, presisi, *recall*, dan *F-measure*[7].

1) Perancangan Proses

Adapun alur perancangan proses mulai dari pembacaan dataset untuk penyakit pneumonia sampai pengujian performance dari implementasi metode KNN, disajikan pada Gambar 1.



Gambar 1. Perancangan Proses

2) Tahapan Penelitian

- Load Dataset

Load citra penyakit pneumonia dan tidak pneumonia, dimana proses ini akan me-load citra penyakit pneumonia yang terbagi dalam 3 macam kombinasi dataset yang berbeda rasio perbandingan. Rasio 20:80 (kombinasi ini terdapat 200 dataset citra penyakit *pneumonia* dan 800 dataset citra tidak *pneumonia*), rasio 5050 (kombinasi ini terdapat 500 dataset citra penyakit *pneumonia* dan 500 dataset citra tidak *pneumonia*), dan rasio 80:20 (kombinasi ini terdapat 800 dataset citra penyakit pneumonia dan 200 dataset citra tidak pneumonia).[8]

- Deteksi Tepi Sobel

Pada tahap ini deteksi tepi akan mengoptimalkan pendeteksian tepi pada citra asli yang bernoise, dengan menggunakan gaussian blur dimana berfungsi untuk memperhalus noise pada citra. Dalam proses deteksi tepi sobel menggunakan *library* cv2.

- Ekstraksi Fitur *Moment Invariant*

Pada tahap ekstraksi fitur *moment invariant* merupakan proses perubahan data citra yang dikonversi menjadi data numerik, dimana menghasilkan 7 array nilai yang dibeli label H1-H7, serta label aktual. Hasil konversi data tersebut kemudian di *export* dalam *format file* .csv (*Comma Separated Values*) yang disimpan kedalam *google drive*. Dalam proses ektraksi fitur menggunakan *library* cv2.

- *Split Dataset*

Pada tahap *split* (membagi) dataset dibagi menjadi 90% data *training* dan 10% data testing yang memiliki nilai *random state* = 0. Random state digunakan agar dataset tidak berubah-ubah, sehingga diberikan nilai ketetapannya.

- Metode K-*Nearest Neighbor* (KNN)

Pada tahap ini merupakan proses pengklasifikasian untuk mengetahui apakah termasuk dataset citra penyakit pneumonia atau tidak pneumonia menggunakan rumus euclidean dan manhattan euclidean dari metode K-*Nearest Neighbor* (KNN) yang menggunakan *library scikit learn*.

- *Calculate Performance*

Pada tahap ini, dilakukan perhitungan performa yang terdiri dari akurasi, presisi, *recall*, dan *f-measure* pada dataset citra penyakit *pneumonia* dan tidak *pneumonia*.

Dalam Proses ini, kita melakukan perhitungan hasil nilai performa yang terdiri dari nilai akurasi, presisi, *recall* dan *f-measure* pada dataset penyakit *Cardiovascular*.

Performa merupakan bentuk tindakan, perbuatan, pekerjaan yang telah dicapai atau dilaksanakan, Kinerja dari klasifikasi dapat dievaluasi berdasarkan perhitungan performa nilai akurasi, presisi, *recall*, *f-measure*.[9]

1. Akurasi

   Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. rumus akurasi. Pada persamaan 2.
2. Presisi

   Presisi didefinisikan sebagai rasio item relevan yang dipilih terhadap semua item yang terpilih. Presisi dapat diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. rumus presisi ditunjukkan pada persamaan 3.
2. *Recall*

   Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap total jumlah item relevan yang tersedia. Rumus *Recall* diuraikan pada persamaan4.
3. *F- measure*

   F- measure adalah *harmonic* mean antara nilai presisi dan recall, *F-measure* juga kadang disebut dengan nama *F-measure*. Rumus *F-measure* dijabarkan pada persamaan 5.

$$AKURASI = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad \text{..... (1)}$$

$$PRESISI = \frac{TP}{TP+FP} \quad \text{..... (2)}$$

$$RECALL = \frac{TP}{TP+FN} \quad \text{..... (3)}$$

$$F-measure = 2 \ \frac{Presisi\ x\ Recall}{Presisi+Recall} \quad \text{..... (4)}$$

Keterangan:

TP : *True Positive*
TN : *True Negative*
FP : *False Positive*
FN : *False Negative*

- *Result Performance*

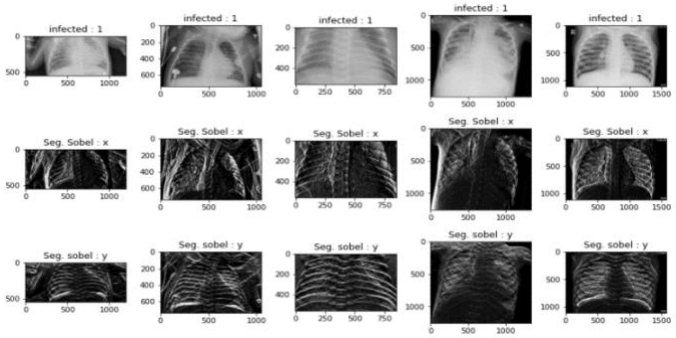
Result performance merupakan hasil dari tahap *calculate performance*, yaitu hasil dari perhitungan nilai akurasi, presisi, *recall*, dan *f-measure* pada daatset citra penyakit *pneumonia* dan tidak *pneumonia*.

III. Hasil dan Pembahasan

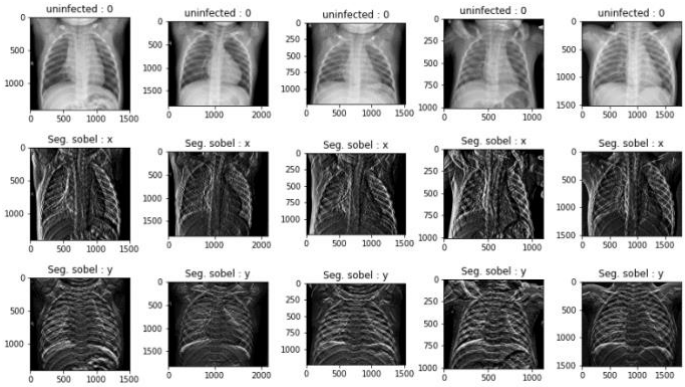
A. *Implementasi*

1) *Implementasi Deteksi Tepi Sobel*

Pada tahap ini dilakukan deteksi tepi pada dataset citra, pada Gambar 2 merupakan contoh implementasi dari deteksi tepi *sobel* pada citra penyakit *pneumonia*. Gambar 3 merupakan implementasi dari deteksi tepi *sobel* pada citra tidak *pneumonia*.



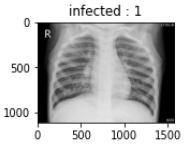
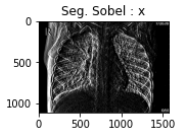
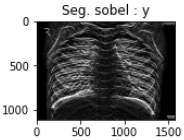
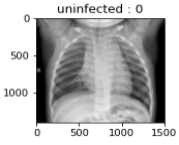
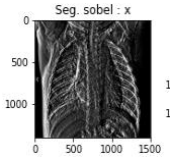
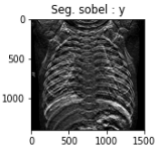
Gambar 2. Hasil Deteksi Tepi Citra Penyakit *Pneumonia*



Gambar 3. Hasil Deteksi Tepi Citra Tidak *Pneumonia*

Pada deteksi tepi *sobel*, kita akan menggunakan *gradien* G(x,y), yang merupakan sebuah vektor yang terdiri dari dua unsur yaitu Gx dan Gy. Deteksi tepi dilakukan dengan cara membaca setiap *pixel* pada citra dengan cara membaca dari *pixel* paling kiri atas (timur utara) dan bergerak ke *pixel* paling kanan bawah (barat selatan). Pada Tabel 2. menunjukkan perbedaan deteksi tepi *sobel* Gx, dan Gy pada citra penyakit pneumonia:

Tabel 2. Perbandingan *Gradien* GX dan Gy Pada Citra *Pneumonia*

No	Citra Asli	Gx	Gy
1			
2			

2) *Implementasi Ekstraksi Fitur Moment Invariant*

Implementasi Ekstraksi Fitur, pada tahap ini merupakan proses perubahan dataset citra yang dikonversi menjadi data numerik, yang menghasilkan 7 label *array* yaitu H1-H7, dan aktual. Dimana aktual merupakan label yang menyimpan hasil dari dataset citra apakah termasuk citra penyakit *pneumonia* atau citra tidak *pneumonia*. Setelah melakukan proses konversi data numerik, maka akan disimpan dalam format file.cvs (*Comma Separated Values*) di *google drive*. Hasil dari implementasi

Ektraksi Fitur *Moment* Invariant penyakit pneumonia ditunjukkan pada Tabel 3 dan Pada Tabel 4. menunjukkan hasil ekstraksi fitur *moment invariant* citra tidak pneumonia.

Tabel 3. Hasil Ekstraksi *Moment Invariant* Citra Penyakit *Pneumonia*

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.003205	2.83E-06	2.96E-10	6.41E-11	-1.88E-22	-2.68E-14	8.83E-21	1
2	0.003585	3.78E-06	1.53E-09	1.69E-09	2.68E-18	3.17E-12	-3.70E-19	1
3	0.002964	1.89E-06	3.39E-10	3.76E-10	1.34E-19	5.15E-13	1.21E-20	1
4	0.002294	1.96E-07	1.18E-10	1.34E-10	1.65E-20	4.79E-14	3.32E-21	1
5	0.002748	8.52E-07	3.12E-10	1.26E-10	2.47E-20	1.16E-13	-3.42E-21	1
6	0.002994	1.12E-06	8.51E-10	7.61E-10	4.97E-19	4.88E-13	-3.58E-19	1
7	0.002570	3.81E-09	2.40E-10	2.44E-10	7.93E-20	1.41E-13	-3.42E-20	1
8	0.003026	3.81E-09	2.40E-10	1.72E-11	8.12E-22	-1.04E-15	-7.53E-22	1
9	0.003338	1.68E-06	1.83E-09	1.92E-09	3.54E-18	2.30E-12	-6.20E-19	1
10	0.002680	1.79E-06	3.50E-10	3.39E-10	1.15E-19	4.51E-13	-1.87E-20	1

Tabel 4. Hasil Ekstraksi *Moment Invariant* Citra Tidak *Pneumonia*

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.002501	5.47E-07	4.63E-11	7.50E-11	4.20E-21	5.44E-14	-1.36E-21	0
2	0.002322	2.17E-07	3.28E-11	1.50E-10	9.28E-21	6.41E-14	-4.86E-21	0
3	0.002372	1.32E-07	6.11E-11	6.23E-11	3.56E-21	2.10E-14	1.43E-21	0
4	0.002254	1.34E-09	1.32E-10	2.89E-11	1.54E-21	-1.73E-16	-9.07E-22	0
5	0.002262	5.83E-09	9.68E-11	5.07E-11	2.06E-21	-7.75E-16	2.89E-21	0
6	0.002214	4.46E-08	8.38E-11	3.93E-11	1.64E-21	8.28E-15	-1.54E-21	0
7	0.002608	2.47E-08	4.29E-10	4.44E-11	5.28E-21	4.33E-15	3.10E-21	0
8	0.002422	3.55E-08	3.11E-11	6.23E-11	2.73E-21	1.17E-14	2.93E-22	0
9	0.00245	6.27E-08	2.35E-11	1.89E-11	3.03E-22	4.72E-15	2.59E-22	0
10	0.002454	3.17E-07	2.60E-10	9.18E-11	1.40E-20	4.83E-14	-2.09E-21	0

3) Implementasi Metode K-Nearest Neighbor (KNN)

Implementasi metode *K-Nearest Neighbor* (KNN) merupakan contoh proses perhitungan manual mulai dari penetapan data *training* dan data *testing*, implementasi metode *K-Nearest Neighbor* (KNN) hingga perhitungan performa. Pada Tabel 5 menunjukkan 10 *sample* data *training*, dan pada Tabel 6 menunjukkan 2 *sample* data *testing*. 12 *sample* tersebut yang akan diterapkan untuk perhitungan manual.

Tabel 5. *Sample* Data *Training*

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.003205	2.83E-06	2.96E-10	6.41E-11	-1.88E-22	-2.68E-14	8.83E-21	1
2	0.003585	3.78E-06	1.53E-09	1.69E-09	2.68E-18	3.17E-12	-3.70E-19	1
3	0.002964	1.89E-06	3.39E-10	3.76E-10	1.34E-19	5.15E-13	1.21E-20	1
4	0.002294	1.96E-07	1.18E-10	1.34E-10	1.65E-20	4.79E-14	3.32E-21	1
5	0.002748	8.52E-07	3.12E-10	1.26E-10	2.47E-20	1.16E-13	-3.42E-21	1
6	0.0025	5.47E-07	4.63E-11	7.50E-11	4.20E-21	5.44E-14	-1.36E-21	0
7	0.00232	2.17E-07	3.28E-11	1.50E-10	9.28E-21	6.41E-14	-4.86E-21	0
8	0.00237	1.32E-07	6.11E-11	6.23E-11	3.56E-21	2.10E-14	1.43E-21	0
9	0.00225	1.35E-09	1.32E-10	2.89E-11	1.54E-21	-1.73E-16	-9.07E-22	0
10	0.00226	5.83E-09	9.68E-11	5.07E-11	2.06E-21	-7.75E-16	2.89E-21	0

Tabel 6. *Sample* Data *Testing*

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.002994	1.12E-06	8.51E-10	7.61E-10	4.97E19	4.88E-13	-3.58E-19	1
2	0.002214	4.46E-08	8.38E-11	3.93E-11	1.64E-21	8.28E-15	-1.54E21	0

Pada Tabel 5 menunjukan 10 data yang digunakan sebagai data *training*. Sedangkan pada Tabel 6 menunjukkan 2 data sebagai data *testing*. Setelah menentukan data *testing* tahap selanjutnya yaitu proses implementasi metode *K-Nearest Neighbor* (KNN) dengan menggunakan persamaan 6 dan persamaan 7.

Pada Tabel 7. menunjukkan hasil perhitungan data *testing* dengan id=1, dan Pada Tabel 8. menunjukkan hasil perhitungan data *testing* dengan id=2 menggunakan rumus persamaan 6 (*Euclidean Distance*). Dimana setiap satu data *testing* dihitung ke semua data *training*. Pada perhitungan dibawah ini setiap atribut yang akan disandingkan pada atribut data *training* sesuai dengan rumus *euclidean distance K-Nearest Neighbor* (KNN) kemudian di urutkan sesuai jumlah *k* yang telah ditentukan dimana hasil dari urutan tersebut dilihat *class* yang paling dominan, dari hasil inilah yang menjadi acuan pengukuran performa dengan mencocokkan TP, TN, FP dan FN pada *Confusion Matrix*[8], [10]–[13].

Tabel 7. Data *Testing* I

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.002994	1.12E-06	8.51E-10	7.61E-10	4.97E19	4.88E-13	-3.58E-19	1

Tabel 8. Perhitungan Data *Testing* I (*Euclidean Distance*)

	Hasil Data Testing	K=5	Urutan Hasil Jarak Terdekat	Actual
1	2.110069E-04	2	3.000991E-05	1
2	5.910060E-04	5	2.110069E-04	1
3	3.000991E-05	1	2.460002E-04	1
4	7.000006E-04	8	4.930003E-04	0
5	2.460002E-04	3	5.910060E-04	1
6	4.930003E-04	4	6.220008E-04	0
7	6.720006E-04	7	6.720006E-04	0
8	6.220008E-04	6	7.000006E-04	1
9	7.400009E-04	10	7.320009E-04	0
10	7.320009E-04	9	7.400009E-04	0

Kesimpulan pada Tabel 9 dan Tabel 10 yaitu data *testing* id=1 yang dihitung menggunakan *euclidean distance* dengan nilai K=5 mempunyai hasil prediksi (klasifikasi) yang tepat, dimana nilai yang paling dominan yaitu nilai 1 (*pneumonia*).

Tabel 9. Data *Testing* II

Id	H1	H2	H3	H4	H5	H6	H7	Actual
2	0.002214	4.46E-08	8.38E-11	3.93E-11	1.64E-21	8.28E-15	-1.54E21	0

Tabel 10. Perhitungan Data *Testing* II (*Euclidean Distance*)

	Hasil Data Testing	K=5	Urutan Hasil Jarak Terdekat	Actual
1	9.910039E-04	9	4.000002E-05	0
2	1.371005E-03	10	4.800002E-05	0
3	7.500023E-04	8	8.000014E-05	1
4	8.000014E-05	3	1.080001E-04	0
5	5.340006E-04	7	1.580000E-04	0
6	2.870004E-04	6	2.870004E-04	0
7	1.080001E-04	4	5.340006E-04	1
8	1.580000E-04	5	7.500023E-04	1
9	4.000002E-04	1	9.910039E-04	1
10	4.800002E-05	2	1.371005E-03	1

Jadi, Kesimpulan dari Tabel 9. yaitu perhitungan data *testing* id=2 menggunakan rumus *euclidean distance* dengan nilai K=5 memiliki hasil prediksi (klasifikasi) yang tepat dimana nilai dominannya yaitu 0 (tidak *pneumonia*).

Pada Tabel 11 dan Tabel 12 menunjukkan hasil perhitungan data *testing* id=1 dan pada Tabel 13 menunjukkan hasil perhitungan data *testing* dengan id=2. Dimana setiap satu atribut data testing akan disandingkan dengan atribut data *training* menggunakan rumus *manhattan distance*, kemudian di urutkan sesuai jumlah *k* yang telah ditentukan dimana hasil dari urutan tersebut dilihat *class* yang paling dominan, dari hasil inilah yang menjadi acuan pengukuran performa dengan mencocokkan TP, TN, FP dan FN pada *Confusion Matrix*.

Tabel 11. Data *Testing* I (*Euclidean Distance*)

Id	H1	H2	H3	H4	H5	H6	H7	Actual
1	0.002994	1.12E-06	8.51E-10	7.61E-10	4.97E19	4.88E-13	-3.58E-19	1

Tabel 12. Perhitungan Data *Testing* I (*Euclidean Distance*)

	Hasil Data Testing	K=5	Urutan Hasil Jarak Terdekat	Actual
1	2.127098E-04	2	2.922965E-05	1
2	5.936546E-04	5	2.127098E-04	1
3	2.922965E-05	1	2.462732E-04	1
4	7.009287E-04	8	4.935783E-04	0
5	2.462732E-04	3	5.936546E-04	1
6	4.935783E-04	4	6.229929E-04	0
7	6.729080E-04	7	6.729080E-04	0
8	6.229929E-04	6	7.009287E-04	1
9	7.411237E-04	10	7.331193E-04	0
10	7.331193E-04	9	7.411237E-04	0

Hasil dari perhitungan data *testing* id=1 dengan nilai K=5 mendapatkan hasil prediksi (klasifikasi) yang tepat yaitu memiliki nilai dominan 1 (*pneumonia*).

Tabel 13. Data *Testing* II (*Manhattan Distance*)

Id	H1	H2	H3	H4	H5	H6	H7	Actual
2	0.002214	4.46E-08	8.38E-11	3.93E-11	1.64E-21	8.28E-15	-1.54E21	0

Tabel 14. Perhitungan Data *Testing* II (*Manhattan Distance*)

	Hasil Data Testing	K=5	Urutan Hasil Jarak Terdekat	Actual
1	9.937904E-04	9	3.995682E-05	0

2	1.374735E-03	10	4.796129E-05	0
3	7.518509E-04	8	8.015191E-05	1
4	8.015191E-05	3	1.081726E-04	0
5	5.348074E-04	7	1.580877E-04	0
6	2.875023E-04	6	2.875023E-04	0
7	1.081726E-04	4	5.348074E-04	1
8	1.580877E-04	5	7.518509E-04	1
9	3.995682E-05	1	9.937904E-04	1
10	4.796129E-05	2	1.374735E-03	1

Hasil perhitungan data *testing* dengan id=2 dengan nilai K=5 mendapatkan hasil prediksi (klasifikasi) yang tepat dengan nilai dominan 0 (tidak pneumonia).

B. *Pembahasan*

Pada penelitian ini menggunakan bahasa pemrograman *python* dan *library scikit learn* yang ada di dalam pemrograman *python.* Mulai dari tahap load Dataset citra penyakit *pneumonia* , deteksi tepi sobel, ektraksi fitur *moment invariant*, *split* data *testing* dan *training*, implementasi metode K-*Nearest Neighbor* (KNN)[14], hingga proses pehitungan performa (Akurasi, Presisi, *Recall*, dan *F-Measure*).

1) *Pembahasan Dataset*

Dataset yang digunakan pada penelitian ini yaitu dataset citra penyakit pneumonia dan tidak pneumonia yang diperoleh dari Kaggle dataset *library*. Dataset citra *Chest X-ray* penyakit pneumonia di publikasikan oleh Paul Mooney pada tanggal 22 Maret 2018 hingga 25 Maret 2018, data tersebut dikumpulkan dari pasien anak berusia 1-5 tahun di Guangzhou Women and Children's Medical Center, Guangzhou. Pada Tabel 15. menunjukkan simulasi pembagian dataset yang digunakan pada penelitian ini:

Tabel 15. Simulasi Pembagian Dataset

Dataset	Jumlah	Rasio Dataset	Split Dataset
Dataset 1	1000	Pneumonia 500 Tidak Pneumonia 500	90% Training 10% Testing
Dataset 2	1000	Pneumonia 800 Tidak Pneumonia 200	90% Training 10% Testing
Dataset 3	1000	Pneumonia 200	90% Training 10% Testing

Pada proses split (membagi) dataset dibagi menjadi 90% data training dan 10% data testing yang memiliki nilai random state = 0. *Random state* digunakan agar dataset tidak berubah-ubah, sehingga diberikan nilai ketetapannya.

2) *Pengujian Performa*

Pengujian yang dilakukan pada penelitian ini yaitu pengujian performa menggunakan metode K-*Neraest Neighbor* (KNN) pada dataset citra penyakit pneumonia sebanyak 1000 dataset, dimana 90% sebagai data training, dan 10% sebagai data testing. Pengujian performa terdiri dari perhitungan nilai akurasi, presisi, *recall* ,dan *f-measure*. Hasil implementasi metode K-*Nearest Neighbor* (KNN) ditunjukkan dalam bentuk confusion matrix, kemudian dilakukan perhitungan nilai perfoma (akurasi, presisi, *recall* dan *f-measure*). Setiap pengujian nilai K memiliki *confusion matrix*-nya masing-masing.

Pada Tabel 16. menunjukkan *confussion matrix* untuk nilai K = 3 pada dataset citra penyakit *pneumonia* dan tidak *pneumonia* yang berjumlah 1000 data dengan rasio 50:50.

Tabel 16. *Confusion Matrix* Nilai K = 5 Pada Rasio Dataset 50:50

	Actual Pneumonia	Actual Tidak Pneumonia
Predicted Pneumonia	TP = 47	FP = 13
Predicted Tidak Pneumonia	FN = 1	TN = 39

Setelah nilai TP, FP, FN, dan TN diketahui, maka nilai performa (akurasi, presisi, *recall*, dan *f-measure*) dapat dihitung seperti yang diilustrasikan dibawah ini:

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$= \frac{(47 + 39)}{(47 + 39 + 13 + 1)} = 0{,}86 = 86\%$$

$$Presisi = \frac{TP}{TP + FP}$$

$$= \frac{47}{47+13} = 0{,}975 = 97\%$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$= \frac{47}{47+1} = 0{,}97 = 97\%$$

$$F - Measure = 2\frac{Presisi\ x\ recall}{Presisi +\ recall}$$

$$= 2\frac{0{,}97\ \text{x}\ 0{,}97}{0{,}97 + 0{,}97} = 0{,}87 = 87\%$$

Pada Tabel 17. menunjukkan *confussion matrix* untuk nilai K = 3 pada dataset citra penyakit *pneumonia* dan tidak *pneumonia* yang berjumlah 1000 data dengan rasio 20:80.

Tabel 17. *Confusion Matrix* Nilai K = 7 Pada Rasio Dataset 20:80

	Actual Pneumonia	**Actual Tidak Pneumonia**
Predicted Pneumonia	TP = 86	FP = 4
Predicted Tidak Pneumonia	FN = 1	TN = 9

Setelah nilai TP, FP, FN, dan TN diketahui, maka nilai performa (akurasi, presisi, *recall*, dan *f-measure*) dapat dihitung seperti yang di ilustrasikan dibawah ini:

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$= \frac{(86+9)}{(86+9+4+1)} = 0{,}95 = 95\%$$

$$Presisi = \frac{TP}{TP + FP}$$

$$= \frac{86}{86+4} = 0{,}95 = 95\%$$

$$Recall = \frac{TP}{TP + FN}$$

$$= \frac{86}{86+1} = 0{,}98 = 98\%$$

$$F - Measure = 2\frac{Presisi\ x\ recall}{Presisi +\ recall}$$

$$= 2\frac{0{,}95\ \text{x}\ 0{,}98}{0{,}95 + 0{,}98} = 0{,}97 = 97\%$$

Pada Tabel 18. menunjukkan *confussion matrix* untuk nilai K = 81 pada dataset citra penyakit *pneumonia* dan tidak *pneumonia* yang berjumlah 1000 data dengan rasio 80:20.

Tabel 18. *Confusion Matrix* Nilai K = 5 Pada Rasio Dataset 80:20

	Actual Pneumonia	**Actual Tidak Pneumonia**
Predicted Pneumonia	TP = 13	FP = 7
Predicted Tidak Pneumonia	FN = 4	TN = 76

Setelah nilai TP, FP, FN, dan TN diketahui, maka nilai performa (akurasi, presisi, *recall*, dan *f-measure*) dapat dihitung seperti yang diilustrasikan dibawah ini:

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$= \frac{(13+76)}{(13+76+7+4)} = 0{,}89 = 89\%$$

$$Presisi = \frac{TP}{TP + FP}$$

$$= \frac{13}{13+7} = 0{,}65 = 65\%$$

$$Recall = \frac{TP}{TP + FN}$$

$$= \frac{13}{13+4} = 0{,}76 = 76\%$$

$$F - Measure = 2\frac{Presisi\ x\ recall}{Presisi +\ recall}$$

$$= 2\frac{0{,}65 \text{ x } 0{,}76}{0{,}65 + 0{,}76} = 0{,}70 = 70\%$$

3) Hasil Pengujian Performa

Pengujian performa pada penelitian ini dilakukan pada nilai K=2 hingga K= banyaknya data *training* untuk mengetahui nilai performa pada dataset citra penyakit pneumonia dan tidak pneumonia dengan rasio 50:50. Pada Tabel 19 menunjukkan hasil performa pada dataset rasio 50:50.

Tabel 19. Hasil Uji Performa Metode K-*Nearest Neighbor* (KNN) Dataset Rasio 50:50

		Manhattan Distance					Euclidean Distance				
		Nilai Performa					Nilai Performa				
		Nilai K	Akurasi	Presisi	Recall	F-Measure	Nilai K	Akurasi	Presisi	Recall	F-Measure
Gaussian Blur	x	3	91	87	97	92	5	92	87	98	92
	y	13	87	80	99	89	11	87	81	97	88
	xy	5	86	80	96	87	7	86	80	96	87
Tanpa Gaussian Blur	x	11	89	83	99	90	14	88	82	98	89
	y	23	85	78	98	87	29	85	78	98	87
	xy	13	86	81	97	88	7	87	83	95	88

Berdasarkan Tabel 19 maka dapat disimpulkan bahwa dengan menguji nilai K = 2 hingga K = banyaknya data *training*, diperoleh hasil performa terbaik pada nilai K = 5 mempunyai akurasi 92%, presisi 87%, *recall* 98% dan *f-measure* 98%.

Pada Tabel 20. pengujian performa pada penelitian ini dilakukan pada nilai K = 2 hingga K= banyaknya data training untuk mengetahui nilai performa pada dataset citra penyakit pneumonia dan tidak pneumonia dengan rasio 20:80. Pada Tabel 20 menunjukkan hasil performa pada dataset rasio 20:80.

Tabel 20. Hasil Uji Performa Metode K-*Nearest Neighbor* (KNN) Dataset Rasio 20:80

		Manhattan Distance					Euclidean Distance				
		Nilai Performa					Nilai Performa				
		Nilai K	Akurasi	Presisi	Recall	F-Measure	Nilai K	Akurasi	Presisi	Recall	F-Measure
Gaussian Blur	x	2	95	94	100	97	7	95	95	98	97
	y	107	95	94	100	97	96	95	94	100	97
	xy	6	95	94	100	97	4	95	95	98	97
Tanpa Gaussian Blur	x	8	95	94	100	97	3	96	97	97	97
	y	169	94	93	100	96	159	94	93	100	96
	xy	31	95	94	100	97	33	95	94	100	97

Berdasarkan Tabel 20 maka dapat disimpulkan bahwa dengan menguji nilai K = 2 hingga K = banyaknya data training, diperoleh hasil performa terbaik pada nilai K = 3 mempunyai akurasi 96%, presisi 97%, *recall* 97% dan *f-measure* 97%.

Pada Tabel 21. pengujian performa pada penelitian ini dilakukan pada nilai K = 2 hingga K= banyaknya data training untuk mengetahui nilai performa pada dataset citra penyakit *pneumonia* dan tidak pneumonia dengan rasio 80:20. Pada Tabel 21 menunjukkan hasil performa pada dataset rasio 80:20.

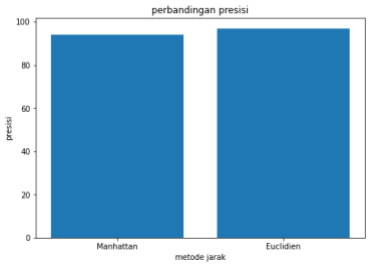
Tabel 21. Hasil Uji Performa Metode K-*Nearest Neighbor* (KNN) Dataset Rasio 80:20

		Manhattan Distance					Euclidean Distance				
		Nilai Performa					Nilai Performa				
		Nilai K	Akurasi	Presisi	Recall	F-Measure	Nilai K	Akurasi	Presisi	Recall	F-Measure
Gaussian Blur	x	67	89	66	70	68	81	89	65	76	70
	y	7	90	73	64	68	9	90	70	70	70
	xy	148	89	62	88	73	148	91	68	88	76
Tanpa Gaussian Blur	x	351	89	75	52	62	350	89	80	47	59
	y	7	90	73	64	68	17	89	66	70	68
	xy	244	89	61	94	74	263	89	68	64	66

Berdasarkan Tabel 21 maka dapat disimpulkan bahwa dengan menguji nilai K = 2 hingga K = banyaknya data *training*, diperoleh hasil performa terbaik pada nilai K = 148 mempunyai akurasi 91%, presisi 68%, *recall* 88% dan *f-measure* 76%. Gambar 3 menunjukkan grafik perbandingan perhitungan performa akurasi antara *euclidean distance* dan *mahattan distance*.

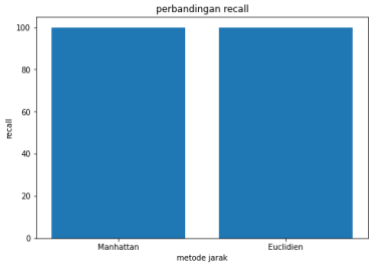
Gambar 4. Grafik Perbandingan Akurasi *Mahanttan Distance* dan *Euclidean Distance*

Kesimpulan dari Gambar 4 perhitungan jarak menggunakan *euclidean distance* lebih tinggi dibanding perhitungan jarak *manhattan distance*. Nilai akurasi perhitungan *euclidean distance* sebesar 96%, sedangkan perhitungan *manhattan distance* sebesar 95%. Gambar 5 menunjukkan grafik perbandingan perhitungan performa presisi antara *euclidean distance* dan *mahattan distance*.



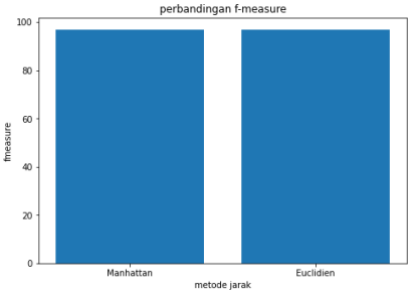
Gambar 5. Grafik Perbandingan Presisi *Mahanttan Distance* dan *Euclidean Distance*

Kesimpulan dari Gambar 5 perhitungan jarak menggunakan *euclidean distance* lebih tinggi dibanding perhitungan jarak menggunakan *manhattan distance*. Nilai presisi *euclidean distance* sebesar 97%, sedangkan *manhattan distance* sebesar 94%. Gambar 6 menunjukkan grafik perbandingan perhitungan performa *recall* antara *euclidean distance* dan *mahattan distance*[14], [15].



Gambar 6. Grafik Perbandingan *Recall Mahanttan Distance* dan *Euclidean Distance*

Kesimpulan dari Gambar 6 menunjukkan perhitungan nilai performa *recall* antara *euclidean distance* dan *manhattan distance* mendapatkan nilai yang sama yaitu sebesar 100%. Gambar 7 menunjukkan grafik perbandingan perhitungan performa *f-measure* antara *euclidean distance* dan *mahattan distance*.



Gambar 7. Grafik Perbandingan *f-measure Mahanttan Distance* dan *Euclidean Distance*

Kesimpulan dari Gambar 7 menunjukkan nilai performa yang sama antara euclidean distance dan manhattan distance yaitu sebesar 97%.

Berdasarkan hasil pengujian, perhitungan performa dataset citra penyakit pneumonia dan bukan pneumonia nilai akurasi tertinggi didapatkan dengan menggunakan rumus perhitungan jarak *euclidean* yaitu sebesar 96%. Nilai performa presisi tertinggi sebesar 97% dengan menggunakan perhitungan jarak *euclidean.* Untuk nilai performa *recall* perhitungan jarak antara *euclidean distance* dan *manhattan distance* yaitu masing-masing memiliki nilai performa *recall* sebesar 100%. Untuk nilai performa *f-measure*

perhitungan jarak antara *euclidean distance* dan *manhattan distance* memiliki nilai performa *f-measure* sebesar 97%.

IV. Kesimpulan

Berdasarkan hasil penelitian ini maka penulis dapat menarik kesimpulan dengan menguju nilai K=2 hingga K=900. Pada dataset simulasi rasio 50:50 diperoleh nilai performa terbaiknya pada K=5 dengan nilai akurasi mencapai 92%, presisi 87%, recall 98%, dan f-measure 92%, dengan kombinasi jenis sobel x menggunakan rumus euclidean distance, Pada dataset dengan simulasi rasio 80:20 diperoleh nilai performa terbaiknya pada K=148 dengan nilai akurasi mencapai 91%, presisi 68%, recall 88%, dan f-measure 76%, dengan kombinasi sobel xy menggunakan rumus euclidean distance, Pada dataset dengan simulasi rasio 20:80 diperoleh nilai performa terbaiknya pada K=3 dengan nilai akurasi mencapai 96%, presisi 97%, recall 97%, dan f-measure 97%, dengan menggunakan kombinasi jenis sobel x menggunakan rumus euclidean distance, dan Pada penelitian ini perhitungan performa yang memiliki nilai akurasi tertinggi yaitu menggunakan rumus perhitungan jarak euclidean yaitu sebesar 96%.

Daftar Pustaka

[1] Y. Farida, A. Trisna, and D. Nur, "Study of Antibiotic Use on Pneumonia Patient in Surakarta Referral Hospital Studi Penggunaan Antibiotik Pada Pasien Pneumonia di Rumah Sakit Rujukan Daerah Surakarta," *J. Pharm. Sci. Clin. Res.*, vol. 02, no. 01, pp. 44–52, 2017, doi: 10.20961/jpscr.v2i01.5240.

[2] Kemenkes RI, "Pneumonia Balita." Buletin Jendela Epidemiologi.

[3] R. Rahmadewi and R. Kurnia, "Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen dengan Metoda Segmentasi Sobel," *J. Nas. Tek. Elektro*, vol. 5, no. 1, p. 7, 2016, doi: 10.25077/jnte.v5n1.174.2016.

[4] A. Zalukhu, "Implementasi Metode Canny Dan Sobel Untuk Mendeteksi Tepi Citra," *J. Ris. Komput.*, vol. 3, no. 6, pp. 25–29, 2016.

[5] F. Liantoni, "Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor," *J. Ultim.*, vol. 7, no. 2, pp. 98–104, 2016, doi: 10.31937/ti.v7i2.356.

[6] S. H. A. Aini, Y. A. Sari, and A. Arwan, "Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, 2018.

[7] H. Zhang, "The optimality of Naive Bayes," *Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004*, vol. 2, pp. 562–567, 2004.

[8] Hasran, "Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor," *Indones. J. Data Sci.*, vol. 1, no. 1, pp. 1–4, 2020.

[9] C. A. U. Hassan, M. S. Khan, and M. A. Shah, "Comparison of Machine Learning Algorithms in Data classification," *Int. Conf. Autom. Comput.*, 2018.

[10] A. Maulida, "Penerapan Metode Klasifikasi K-Nearest Neigbor pada Dataset Penderita Penyakit Diabetes," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 29–33, 2020.

[11] N. Fadhillah, H. Azis, and D. Lantara, "Validasi Pencarian Kata Kunci Menggunakan Algoritma Levenshtein Distance Berdasarkan Metode Approximate String Matching," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf*, vol. 1, pp. 3–7.

[12] A. A. Karim, H. Azis, and Y. Salim, "Kinerja Metode C4.5 dalam Penyaluran Bantuan Dana Bencana 1," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 84–87, 2018.

[13] M. M. Baharuddin, T. Hasanuddin, and H. Azis, "Analisis Performa Metode K-Nearest Neighbor untuk Identifikasi Jenis Kaca," *Ilk. J. Ilm.*, vol. 11, no. 28, pp. 269–274, 2019.

[14] D. Cahyanti, A. Rahmayani, and S. Ainy, "Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 39–43, 2020.

[15] Y. Lukito and A. R. Chrismanto, "Perbandingan Metode-Metode Klasifikasi untuk Indoor Positioning System," *J. Tek. Inform. dan Sist. Inf.*, vol. 1, no. 2, pp. 123–131, 2015, doi: 10.28932/jutisi.v1i2.373.