



TEKNOLOGI dan rekayasa

PERBANDINGAN ALGORITMA KLASIFIKASI DALAM PENGKLASIFIKASIAN DATA PENYAKIT JANTUNG KORONER	161
Ardea Bagas Wibisono, Achmad Fahrurrozi	
SISTEM MONITORING GAS CO PADA PARKIRAN BASEMENT MALL DI JAKARTA MENGGUNAKAN METODE REAL-TIME BERBASIS INTERNET OF THINGS (IoT)	171
Fikri Faisal Adli, Ajib Setyo Arifin	
SISTEM PAKAR UNTUK DIAGNOSA AWAL PENYAKIT LAMBUNG MENGGUNAKAN METODE DEMPSTER-SHAFFER BERBASIS WEB	182
Rizky Ardiansyah, Fauziah, Andria Ningsih	
KLASIFIKASI KERUSAKAN BANGUNAN SEKOLAH MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DENGAN PRE-TRAINED MODEL VGG-16	197
Ade Muhammad Rizki, Nola Marlina	
KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)	207
Nur Fadlia, Rifki Kosasih	
RANCANG BANGUN PURWARUPA SISTEM PENGUNCI LEMARI DENGAN PENGENALAN SUARA	216
Laksamana Akbar Dzulfikar, Emy Haryatmi, Tri Agus Riyadi	
OPTIMASI RUTE TRUK PENGANGKUT SAMPAH DI KOTA DEPOK	226
Tegar Mujadid Mappa, Sudaryanto	

DEWAN REDAKSI JURNAL TEKNIK REKAYASA

Penanggung Jawab

Prof. Dr. E.S. Margianti, S.E., M.M.
Prof. Suryadi Harmanto, SSI., M.M.S.I.
Drs. Agus Sumin, M.M.S.I.

Dewan Editor

Dr. Desti Riminarsih, S.Si, M.Si, Universitas Gunadarma
Dr. Dina Indarti, S.Si, M.Si, Universitas Gunadarma
Dr. Ir. Asep Mohamad Noor, M.T., Universitas Gunadarma
Ajib Setyo Arifin, S.T., M.T., Ph.D., Universitas Indonesia
Dr. Sunny Arief Sudiro, STMIK Jakarta STI&K

Mitra Bebestari

Prof. Dr. Sarifuddin Madenda, Universitas Gunadarma
Prof. Dr.-Ing. Adang Suhendra, S.Si, Skom, Msc, Universitas Gunadarma
Prof. Ir. Busono Soerowirdjo, Msc, Phd, Universitas Gunadarma
Prof. Dr. Rer.Nat. A. Benny Mutiara, Universitas Gunadarma
Prof. Dr. Ir. Bambang Suryawan, MT, Universitas Gunadarma
Prof. Dr. B.E.F. da Silva, Universitas Indonesia
Prihandoko, S.Kom, MIT, Phd, Universitas Gunadarma
Dr. Tubagus Maulana Kusuma, Skom., Mengsc., Universitas Gunadarma
Dr. Ir. Rakhma Oktavina, MT., Universitas Gunadarma
Dr. RR Sri Poernomo Sari, M.T., Universitas Gunadarma
Dr. Lussiana ETP, Ssi., M.T., STMIK Jakarta STI&K

Sekretariat Redaksi

Universitas Gunadarma
Jalan Margonda Raya No. 100 Depok 16424
Phone : (021) 78881112 ext 516.

JURNAL ILMIAH TEKNOLOGI DAN REKAYASA

NOMOR 3, VOLUME 24, DESEMBER 2019

DAFTAR ISI

PERBANDINGAN ALGORITMA KLASIFIKASI DALAM PENGKLASIFIKASIAN DATA PENYAKIT JANTUNG KORONER Ardea Bagas Wibisono, Achmad Fahrurozi	161
SISTEM MONITORING GAS CO PADA PARKIRAN BASEMENT MALL DI JAKARTA MENGGUNAKAN METODE <i>REAL-TIME</i> BERBASIS <i>INTERNET OF THINGS</i> (IoT) Fikri Faisal Adli, Ajib Setyo Arifin	171
SISTEM PAKAR UNTUK DIAGNOSA AWAL PENYAKIT LAMBUNG MENGGUNAKAN METODE DEMPSTER-SHAFFER BERBASIS WEB Rizky Ardiansyah, Fuziah Fauziah, Andria Ningsih	182
KLASIFIKASI KERUSAKAN BANGUNAN SEKOLAH MENGGUNAKAN METODE <i>CONVOLUTIONAL NEURAL NETWORK</i> DENGAN <i>PRE-TRAINED MODEL VGG-16</i> Ade Muhammad Rizki , Nola Marina	197
KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE <i>CONVOLUTIONAL NEURAL NETWORK</i> (CNN) Nur Fadlia, Rifki Kosasih	207
RANCANG BANGUN PURWARUPA SISTEM PENGUNCI LEMARI DENGAN PENGENALAN SUARA Laksamana Akbar Dzulfikar, Emy Haryatmi, Tri Agus Riyadi	216
OPTIMASI RUTE TRUK PENGANGKUT SAMPAH DI KOTA DEPOK Tegar Mujadid Mappa, Sudaryanto	226

PERBANDINGAN ALGORITMA KLASIFIKASI DALAM PENGKLASIFIKASIAN DATA PENYAKIT JANTUNG KORONER

¹Ardea Bagas Wibisono, ²Achmad Fahrurozi

^{1,2}Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹ardeabagas22@gmail.com, ²achmad_fahrurozi@staffsite.gunadarma.ac.id

Abstrak

Penyakit Jantung Koroner (PJK) menjadi penyebab kematian tertinggi pada semua umur setelah stroke. Hal ini mendorong banyak penelitian terhadap penyakit jantung koroner, salah satunya menggunakan metode berbasis komputer. Pengolahan data dalam jumlah besar dapat dilakukan dengan klasifikasi menggunakan algoritma tertentu sehingga hasilnya cepat dan akurat. Metode klasifikasi yang umum digunakan antara lain Naïve Bayes, K-Nearest Neighbor, Decision Tree dan Random Forest. Metode Naïve Bayes menggunakan probabilitas disetiap data, metode K-Nearest Neighbor menggunakan perhitungan jarak, metode Decision Tree menggunakan pohon keputusan, sedangkan metode Random Forest menggunakan beberapa pohon keputusan yang disatukan. Penelitian ini bertujuan untuk membandingkan keempat algoritma tersebut dalam mengklasifikasikan data penyakit jantung koroner. Perbandingan algoritma akan dilihat berdasarkan performance measure yang terdiri dari tingkatan akurasi, recall disetiap kelas, dan presisi disetiap kelas. Pada setiap algoritma diuji menggunakan cross validation. Berdasarkan hasil perbandingan terhadap 300 dataset penyakit jantung koroner, algoritma Random Forest lebih baik dan optimal dibanding dengan Algoritma Naïve Bayes, K-Nearest Neighbor, dan Decision Tree untuk mengklasifikasikan penyakit jantung koroner. Hasil klasifikasi dengan algoritma Random Forest memiliki rerata tingkat akurasi sebesar 85,668 % dengan recall kelas '1' adalah 89 %, recall kelas '0' adalah 83,6%, presisi kelas '1' adalah 85%, dan presisi kelas '0' adalah 85,8%.

Kata Kunci: *Klasifikasi, machine learning, penyakit jantung koroner.*

Abstract

Coronary Artery Disease (CAD) is the highest cause of death in all age categories after stroke. This fact encourages numerous studies on coronary artery disease, one of which using computer-based methods. Processing big data can be done by using classification with certain algorithms to obtain fast and accurate results. The commonly used classification methods include Naïve Bayes, K-Nearest Neighbor, Decision Tree and Random Forest. The Naïve Bayes method uses probability in each data, the K-Nearest Neighbor uses distance calculation, the Decision Tree uses a decision tree, and the Random Forest uses several integrated decision trees. This study aims to compare the four algorithms in classifying coronary artery disease data. Algorithm comparison was determined based on performance measure of each algorithm, which consists of accuracy level, recall in each class, and precision in each class. Each algorithm was tested using cross validation. Based on the comparison results of 300 coronary artery disease data, the Random Forest algorithm was better and more optimum compared to the Naïve Bayes Algorithm, K-Nearest Neighbor, and Decision Tree in classifying coronary artery disease. Classification results with the Random Forest algorithm have an average accuracy rate of 85.668% based on the following data: the class '1' recall was 89%, the class '0' recall was 83.6%, the class '1' precision was 85%, and the class '0' precision was 85.8%.

Keywords: *classification, coronary artery disease, machine learning.*

PENDAHULUAN

Sistem peredaran darah manusia merupakan salah satu sistem yang penting pada tubuh manusia. Sistem ini memiliki dua fungsi utama, yakni untuk mengedarkan oksigen dan nutrisi keseluruh organ tubuh manusia serta mengangkut sisa hasil metabolisme. Salah satu organ yang penting dalam sistem peredaran darah manusia adalah jantung. Jantung bertugas sebagai alat pompa untuk mengedarkan darah keseluruh tubuh manusia. Jika jantung mengalami suatu gangguan atau kerusakan maka akan mengakibatkan terganggunya seluruh kinerja organ didalam tubuh manusia. Menurut survei *Sample Regristration System (SRS)* pada 2014 di Indonesia menunjukkan bahwa Penyakit Jantung Koroner (PJK) menjadi penyebab kematian tertinggi pada semua umur setelah stroke, yakni sebesar 12,9% [1].

Kebanyakan pasien penderita penyakit jantung tidak mengetahui gejala-gejala awal yang dirasakan dan tidak sedikit banyak penderita penyakit jantung konorer yang meninggal disebabkan oleh serangan jantung. Masih kurangnya kesadaran terhadap pola hidup yang sehat dan kurangnya informasi penyakit jantung konorer yang dapat membuat seseorang tidak dapat mengenali gejala awalnya. Proses untuk mendeteksi penyakit jantung dapat dilakukan dengan cara manual, yakni dengan konsultasi langsung ke dokter spesialis jantung dan melakukan beberapa pemeriksaan laboratorium yang

kemudian harus dikonsultasikan kembali oleh dokter spesialis jantung. Hal ini tentu saja memerlukan biaya yang relatif besar. Dengan resiko kematian yang sangat tinggi, maka diperlukan suatu sistem yang dapat mendeteksi penyakit jantung koroner pada penderita secara akurat serta dengan biaya yang tidak besar.

Hal ini mendorong banyak penelitian terhadap penyakit jantung koroner, salah satunya menggunakan metode berbasis komputer. Metode ini banyak dikembangkan dengan bantuan komputasi cerdas yang mampu mengolah data dalam jumlah yang besar. Pengolahan data dalam jumlah besar dapat dilakukan dengan klasifikasi menggunakan algoritma tertentu sehingga hasilnya cepat dan akurat.

Metode klasifikasi yang umum digunakan antara lain *Naïve Bayes* [2], *K-Nearest Neighbor* [2, 3], *Decision Tree* [2, 4], *Random Forest* [5] dan *Support Vector Machines* [6]. Pada penelitian ini, beberapa metode klasifikasi diimplementasikan pada kasus pengenalan penyakit jantung koroner untuk kemudian dibandingkan hasil *performance measure* (akurasi, *recall*, dan presisi).

Beberapa penelitian yang membahas klasifikasi penyakit jantung koroner diantaranya penelitian oleh Retnasari dan Rahmawati. Penelitian tersebut membandingkan dua algoritma, yakni algoritma *Naïve Bayes* dan algoritma *C4.5*. Pada hasil penelitian tersebut menunjukkan bahwa algoritma *Naïve Bayes* mendapatkan nilai akurasi 86,67% dan

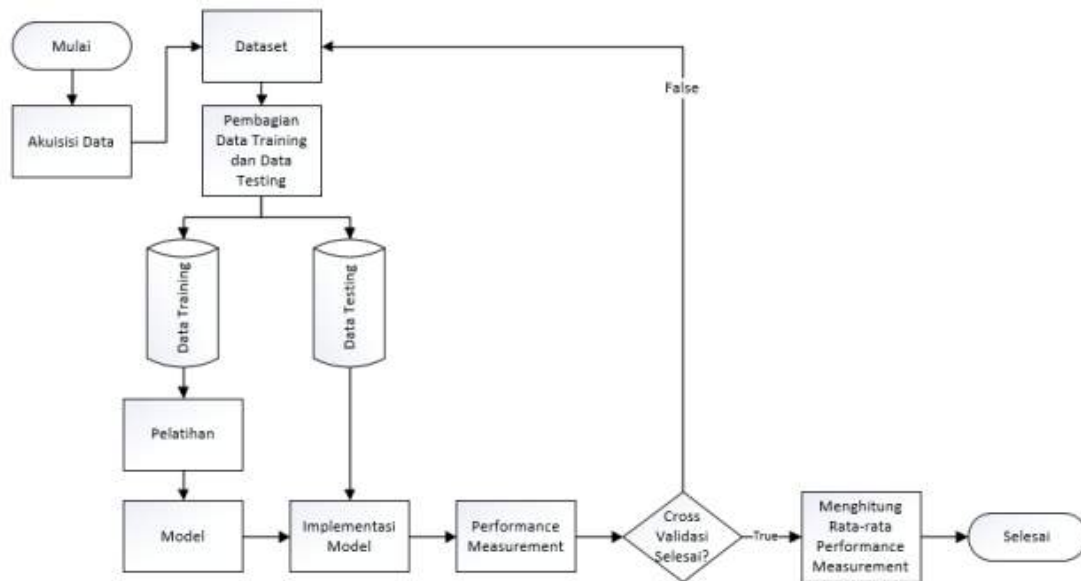
algoritma *C4.5* mendapatkan nilai 83,70% [7].

Algoritma *Naïve Bayes*, *Random Forest*, *Decision Tree* dan *K-Nearest Neighbor* adalah algoritma pengklasifikasian dokumen teks yang umumnya mendapat nilai akurasi relatif tinggi. Hal ini dibuktikan melalui penelitian yang dilakukan oleh Dewi. Penelitian tersebut membandingkan lima algoritma, yakni algoritma *Neural Network*, *Decision Tree*, *Naïve Bayes*, *K-Nearest Neighbor* dan *Logistic Regresion*. Hasil penelitian tersebut menunjukkan bahwa algoritma *Neural Network* mendapatkan nilai akurasi tertinggi yaitu 89,71%. Algoritma dengan nilai akurasi kedua tertinggi yaitu algoritma *Logistic Regresion* yang mendapatkan nilai akurasi 89,32%, lalu algoritma *Decision Tree* mendapatkan nilai akurasi 89,10%. Selanjutnya, algoritma *K-Nearest Neighbor* dengan nilai akurasi 87,79% dan yang terakhir algoritma *Naïve Bayes* dengan nilai akurasi 84,70% [8].

Algoritma *Naïve Bayes*, *Random Forest*, *Decision Tree* dan *K-Nearest Neighbor* masing-masing memiliki kelebihan dan kekurangan. Dengan demikian, pada penelitian ini dilakukan perbandingan antara keempat algoritma tersebut untuk memperoleh algoritma yang paling cocok dalam klasifikasi terhadap data penyakit jantung koroner. Parameter pembanding keempat algoritma tersebut yakni hasil *performance measure* (akurasi, *recall*, dan presisi).

METODE PENELITIAN

Pada metode penelitian terdiri atas beberapa tahapan proses, yakni mengakuisisi data yang akan digunakan untuk implementasi lalu menganalisa parameter yang digunakan untuk memproses data dengan melakukan pembagian data yang telah diolah. Pembagian data untuk data *training* dan data *testing* dengan perbandingan sebesar 80:20. Kemudian merancang sebuah model algoritma *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree* dan *Random Forest*. Setelah itu melakukan proses pelatihan data dengan algoritma yang telah dipilih. Pada tahap ini model akan mempelajari *feature* yang didapat dan melakukan pengubahan nilai parameter selama latihan berlangsung. Tahap berikutnya yakni mempelajari model dengan data uji yang kemudian data uji akan diklasifikasikan hasilnya dan dibandingkan dengan nilai parameter data uji yang sesungguhnya sehingga mendapatkan tingkat akurasi dalam klasifikasi proses uji data dan data uji yang sesungguhnya. Percobaan akan dilakukan dengan menggunakan *cross validation*, sehingga membutuhkan beberapa percobaan, dalam penelitian ini dilakukan 5 percobaan dalam satu algoritma. Hasil akhir dengan menggunakan *cross validation* dinyatakan dari rata-rata hasil akurasi dari 5 percobaan yang diterapkan pada setiap algoritma. Gambaran umum mengenai alur proses dari penelitian dapat dilihat pada Gambar 1.



Gambar 1. Skema Umum Penelitian

Akuisisi Data

Proses akuisisi data dilakukan dengan mengunduh *dataset* penyakit jantung koroner yang berasal dari lokasi *Cleveland Clinic Foundation* yang telah diadopsi oleh instansi *Hungarian Institute of Cardiology* di Budapest [9]. Data yang diambil terdiri dari 300 data pasien jantung koroner, di mana masing-masing data memiliki 14 variabel, terdiri atas 13 variabel independen dan 1 variabel dependen yang merupakan parameter hasil diagnosis penyakit jantung koroner (memiliki 2 buah nilai, yakni nilai 0 untuk positif dan nilai 1 untuk negatif).

Dataset

Pada *dataset* penyakit jantung koroner terdapat 14 variabel yang terbagi menjadi 13 independen dan 1 variabel dependen. Variabel independen sebagai berikut:

1. *Age*: Umur pasien.
2. *Sex*: Jenis kelamin pasien, atribut ini memiliki 2 nilai, yakni nilai 1 untuk laki-laki dan nilai 0 untuk perempuan.
3. *Cp*: Tipe nyeri dada yang diderita pasien. Atribut ini memiliki 4 nilai, yaitu :
 Nilai 0: *asymptomatic*
 Nilai 1: *atypical angina*
 Nilai 2: *non-anginal pain*
 Nilai 3: *typical angina*
4. *Trestbps*: *resting blood pressure* yaitu tekanan darah pasien ketika dalam keadaan istirahat. Satuan yang dipakai adalah mm Hg.
5. *Chol*: *Cholesterol* yaitu kadar kolesterol dalam darah pasien, dengan satuan mg/dl.
6. *Fbs*: *fasting blood sugar* yaitu kadar gula darah pasien, atribut *fbs* ini hanya memiliki 2 nilai yaitu 1 jika kadar gula

darah pasien lebih dari 120 mg/dl, dan 0 jika kadar gula darah pasien kurang dari sama dengan 120 mg/dl.

7. *Restecg*: *resting electrocardiographic* yaitu kondisi *ECG* pasien ketika dalam keadaan istirahat. Atribut ini memiliki 3 nilai yaitu nilai 1 untuk keadaan normal, nilai 2 untuk keadaan *ST-T wave abnormality* yaitu keadaan dimana gelombang inversions T dan atau ST meningkat maupun menurun lebih dari 0,5 mV dan nilai 3 untuk keadaan dimana ventricular kiri mengalami hipertropi.
8. *Thalach*: rata-rata detak jantung pasien dalam satu menit.
9. *Exang*: keadaan dimana pasien akan mengalami nyeri dada apabila berolah raga, 0 jika tidak nyeri, dan 1 jika menyebabkan nyeri.
10. *Oldpeak*: penurunan ST akibat olahraga.
11. *Slope*: *slope* dari puncak ST setelah berolah raga. Atribut ini memiliki 3 nilai yaitu 0 untuk *downsloping*, 1 untuk *flat*, dan 2 untuk *upsloping*.
12. *Ca*: banyaknya pembuluh darah yang terdeteksi melalui proses pewarnaan *flourosopy*.
13. *Thal*: detak jantung pasien. Atribut ini memiliki 3 nilai yaitu 1 untuk *fixed defect*, 2 untuk normal dan 3 untuk *reversal defect*.

Variabel dependen pada penelitian ini yaitu variabel target yang berarti hasil diagnosa penyakit jantung, memiliki 2 nilai, yakni 0 untuk terdiagnosa positif terkena

penyakit jantung koroner, dan 1 untuk negatif terkena penyakit jantung koroner.

Pembagian Data Training dan Data Testing

Pada tahap ini dilakukan pemisahan *dataset* menjadi dua bagian, yakni data *training* dan data *testing*. Jumlah data *training* dan data *testing* yang digunakan adalah perbandingan 80:20. Pada penelitian ini terdapat 300 data pasien jantung koroner, sehingga jumlah data *training* dan data *testing* berturut-turut adalah 240 data dan 60 data. Data *training* digunakan untuk melatih *classifier* dalam mengenali karakteristik pasien yang positif terkena jantung koroner maupun yang negatif. Data *testing* digunakan dalam uji coba terhadap model klasifikasi yang dihasilkan dan menentukan performa dari model klasifikasi dengan cara membandingkan hasil klasifikasi model terhadap tiap data dalam data *testing* dengan label sebenarnya.

Perancangan Model

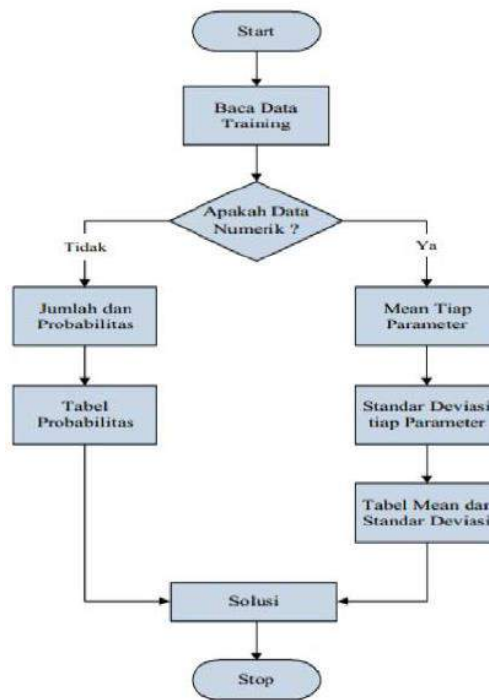
Perancangan model algoritma *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree* dan *Random Forest* yang dibangun akan digunakan sebagai *classifier* untuk mengelompokkan data yang positif terkena penyakit jantung koroner atau tidak berdasarkan parameter yang diberikan.

1. Perancangan Model Algoritma Naïve Bayes

Perancangan model algoritma *Naïve Bayes* tidak memiliki banyak pengaturan karena algoritma *Naïve Bayes* merupakan algoritma pengklasifikasian sederhana yang meng-

hitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari *dataset* yang diberikan. Dalam penelitian ini digunakan distribusi frekuensi Normal dalam model perhitungan

probabilitasnya, sehingga data penyakit jantung koroner yang dihimpun diasumsikan berdistribusi Normal standar. Gambar 2 merupakan alur pembentukan model *Naïve Bayes*.



Gambar 2. Alur Model *Naïve Bayes*

2. Algoritma *k-Nearest Neighbor*

Pada proses perancangan model algoritma *k-Nearest Neighbor*, model algoritma *k-Nearest Neighbor* dibangun dengan menggunakan beberapa nilai *k* yang umum digunakan, untuk kemudian dipilih nilai *k* yang menghasilkan performa terbaik. Ukuran jarak yang digunakan adalah *Euclidean Distance* dan *Manhattan Distance* untuk kemudian dibandingkan dan diambil yang memberikan performa terbaik dalam kasus data penyakit jantung koroner.

3. Algoritma *Decision Tree*

Pada proses perancangan model algoritma *Decision Tree*, model algoritma *Decision Tree* yang dibangun menggunakan algoritma *CART* sebagai *criterion*-nya, serta menggunakan *splitter best* untuk mendapatkan hasil akurasi terbaik dalam kasus data penyakit jantung koroner.

4. Algoritma *Random Forest*

Pada proses perancangan model algoritma *Random Forest*, model algoritma *Random Forest* yang dibangun dengan menggunakan 'n' pohon *Decision Tree* dengan mencari

nilai 'n' terbaik, maka didapatkan hasil optimum dari algoritma *Random Forest*, algoritma *CART* sebagai *criterionnya*, serta menggunakan *splitter best* untuk mendapatkan hasil akurasi terbaik dalam kasus data penyakit jantung koroner.

Pelatihan Model

Tahap selanjutnya adalah proses pelatihan model dengan menggunakan algoritma *Naïve Bayes*, *k-Nearest Neighbor*, *Decision Tree*, serta *Random Forest*. Proses pelatihan ini dilakukan dengan prinsip *k-fold cross validation*, yaitu dengan cara membagi data menjadi sejumlah *k* himpunan bagian, sedemikian sehingga setiap kali salah satu himpunan bagian digunakan sebagai data *testing*, maka $k - 1$ himpunan bagian sisanya disatukan untuk membentuk data *training*. Estimasi kesalahan dirata-rata pada tiap uji coba, di mana uji coba dilakukan berulang sebanyak *k* kali, sedemikian sehingga masing-masing himpunan bagian dari *k* himpunan bagian tersebut berperan menjadi himpunan *testing* sebanyak tepat satu kali. Hal ini dilakukan untuk mendapatkan efektivitas total dari *dataset* yang ada. Pengulangan data latih dan data uji ini secara signifikan mengurangi varians karena sebagian besar data juga digunakan dalam data uji.

Pengujian Model

Pengujian dilakukan terhadap model-model yang dihasilkan dari 4 buah algoritma, yaitu algoritma *Naïve Bayes*, algoritma *k-*

Nearest Neighbor, algoritma *Decision Tree* dan algoritma *Random Forest*, yang menghasilkan *performance measure* yang berbeda-beda terhadap kasus data penyakit jantung koroner. Berdasarkan prinsip *k-fold cross validation*, maka *performance measure* yang diberikan adalah nilai rata-rata performa dari tiap model yang dihasilkan dari *k* percobaan terhadap masing-masing model tersebut. Adapun *performance measure* yang digunakan adalah *recall*, presisi, dan akurasi yang disajikan dalam bentuk persentase dengan rumusan masing-masing berdasarkan nilai-nilai pada *confusion matrix*.

HASIL DAN PEMBAHASAN

Pengujian terhadap empat algoritma, yakni algoritma *Naïve Bayes*, algoritma *K-Nearest Neighbor*, algoritma *Decision Tree* dan algoritma *Random Forest* menghasilkan akurasi yang berbeda-beda terhadap kasus penyakit jantung koroner. Pengujian menggunakan 300 jumlah data dari *dataset* yang disediakan. Setiap data diuji terhadap seluruh data yang ada. Kemudian hasil pengujian tersebut diukur untuk mendapatkan tingkat ketepatan dalam mengklasifikasikan penyakit jantung koroner dengan menghitung rata-rata *recall*, presisi, dan akurasi dari setiap percobaan menggunakan *cross validation* yang disajikan dalam bentuk persentase. Tabel 1 menunjukkan rangkuman dari hasil *performance measure* terhadap seluruh hasil pengujian dalam masing-masing algoritma.

Tabel 1. Hasil *Performance Measure* Klasifikasi Penyakit Jantung Koroner

Algoritma	<i>Performance Measure</i>				
	<i>Recall '1'</i>	<i>Recall '0'</i>	<i>Presisi '1'</i>	<i>Presisi '0'</i>	<i>Akurasi</i>
<i>Naïve Bayes</i>	84,8 %	78,4 %	80,8 %	81,2 %	80,33 %
<i>K-Nearest Neighbor</i>	73,4 %	68 %	72,2 %	68,2 %	69,67 %
<i>Decision Tree</i>	82,8 %	78,2 %	81,2 %	79,4 %	80,33 %
<i>Random Forest</i>	89 %	83,6 %	85 %	85,8 %	85,668 %

Hasil *performance measure* pada Tabel 1 menunjukkan akurasi paling optimal untuk keempat algoritma. Berdasarkan hasil perbandingan pada Tabel 1 dapat disimpulkan bahwa hasil akurasi tertinggi terhadap klasifikasi penyakit jantung koroner diperoleh algoritma *Random Forest* sebesar 85,67 % dengan *recall* kelas '1' adalah 89 %, *recall* kelas '0' adalah 83,6%, presisi kelas '1' adalah 85%, dan presisi kelas '0' adalah 85,8%. Hasil akurasi yang diperoleh algoritma *Naïve Bayes* sebesar 80,33 % dengan *recall* kelas '1' adalah 84,8%, *recall* kelas '0' adalah 78,4%, presisi kelas '1' adalah 80,8%, dan presisi kelas '0' adalah 81,2%. Algoritma *Decision Tree* memiliki akurasi sebesar 80,33% dengan *recall* kelas '1' adalah 82,8%, *recall* kelas '0' adalah 78,2%, presisi kelas '1' adalah 81,2%, dan presisi kelas '0' adalah 79,4%. Hasil akurasi yang terendah yaitu algoritma *K-Nearest Neighbor* dengan akurasi sebesar 69,67% dengan *recall* kelas '1' adalah 73,4%, *recall* kelas '0' adalah 68%, presisi kelas '1' adalah 72,2%, dan presisi kelas '0' adalah 68,2%. Hasil tersebut menunjukkan bahwa model klasifikasi algoritma *Random Forest* menghasilkan *performance measure* yang terbaik untuk mengklasifikasikan penyakit

jantung koroner dibandingkan dengan model klasifikasi algoritma *Naïve Bayes*, Algoritma *Decision Tree* dan Algoritma *K-Nearest Neighbor*.

KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa implementasi dan perbandingan algoritma *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree* dan *Random Forest* terhadap data kasus penyakit jantung koroner berhasil direalisasikan. Implementasi dan uji coba telah dilakukan 300 *dataset* penyakit jantung koroner. Selain itu, telah dilakukan perbandingan terhadap hasil uji coba Algoritma *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree* dan *Random Forest*.

Berdasarkan perbandingan hasil uji coba, *performance measure* algoritma *Random Forest* memiliki hasil yang lebih baik dibanding dengan algoritma *Naïve Bayes*, *K-Nearest Neighbor* dan *Decision Tree* dengan metode *k-fold cross validation*. Algoritma *Random Forest* dapat memberikan rerata hasil akurasi sebesar 85,668 % dengan *recall* kelas '1' adalah 89 %, *recall* kelas '0' adalah 83,6%, presisi kelas '1' adalah 85%,

dan presisi kelas '0' adalah 85,8%. Dengan demikian dapat dikatakan bahwa algoritma *Random Forest* adalah algoritma yang terbaik dalam mengklasifikasi kasus penyakit jantung koroner dibanding dengan algoritma *Naïve Bayes*, *K-Nearest Neighbor* dan *Decision Tree*.

Pada penelitian ini dilakukan uji coba menggunakan data dalam jumlah kecil, sehingga pada penelitian selanjutnya dapat dikembangkan dengan melakukan uji coba pada data dengan jumlah yang lebih besar. Pada penelitian lebih lanjut dapat juga menambahkan algoritma klasifikasi lainnya sehingga mendapat perbandingan yang lebih beragam. Selain itu, penelitian lebih lanjut juga dapat mengembangkan klasifikasi kelas yang digunakan, agar menjadi rinci dalam mengklasifikasi penyakit jantung koroner berdasarkan level atau tingkatan tertentu.

DAFTAR PUSTAKA

- [1] Kementerian Kesehatan RI, "Penyakit jantung penyebab kematian tertinggi, kemenkes ingatkan cerdas," *Kementerian Kesehatan RI*, 2017. [Daring]. Tersedia: <http://www.depkes.go.id/article/view/17073100005/penyakit-jantung-penyebab-kematian-tertinggi-kemenkes-ingatkan-cerdik-.html>. [Diakses: 25 April 2019].
- [2] D. Sartika dan D. I. Sensuse, "Perbandingan algoritma klasifikasi Naive Bayes, Nearest Neighbour, dan Decision Tree pada studi kasus pengambilan keputusan pemilihan pola pakaian," *Jatiji*, vol. 1, no. 2, hal. 153 – 154, 2017.
- [3] M. Lestari, "Penerapan algoritma klasifikasi Nearest Neighbor (K-NN) untuk mendeteksi penyakit jantung," *Faktor Exacta*, vol. 7, no. 4, hal. 366 – 371, 2014.
- [4] F. Maspiyanti dan J. Gatc, "Diagnosa penyakit jantung pada ponsel menggunakan pohon keputusan," *Jurnal Teknologi Terpadu*, vol. 1, no. 1, hal. 13 – 20, 2015.
- [5] M. R. Amiarrahman dan T. Handhika, "Analisis dan implementasi algoritma klasifikasi Random Forest dalam pengenalan Bahasa Isyarat Indonesia (BISINDO)," *Prosiding Seminar Nasional Inovasi Teknologi*, 2017, hal. 83 – 88.
- [6] A. M. Puspitasari, D. E. Ratnawati, dan A. W. Widodo, "Klasifikasi penyakit gigi dan mulut menggunakan metode Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, hal. 802 – 810, 2018.
- [7] T. Retnasari dan E. Rahmawati, "Diagnosa prediksi penyakit jantung dengan model algoritma Naïve Bayes dan algoritma C4.5," *Prosiding Konferensi Nasional Ilmu Sosial dan Teknologi (KNiST)*, 2017, hal. 7 – 12.

- [8] S. Dewi, “Komparasi 5 metode algoritma klasifikasi data mining pada prediksi keberhasilan pemasaran produk layanan perbankan,” *Jurnal Techno Nusa Mandiri*, vol. 8, no. 1, hal. 60 – 66, 2016.
- [9] Kaggle Dataset, “Heart disease UCI,” *Kaggle Dataset*, 2019. [Daring]. Tersedia: <https://www.kaggle.com/ronitf/heart-disease-uci>. [Diakses: 22 April 2019].

SISTEM MONITORING GAS CO PADA PARKIRAN BASEMENT MALL DI JAKARTA MENGGUNAKAN METODE *REAL-TIME* BERBASIS *INTERNET OF THINGS* (IoT)

¹Fikri Faisal Adli, ²Ajib Setyo Arifin

^{1,2} Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia
Depok, Jawa Barat

¹fikri.faisal71@ui.ac.id, ²ajib@eng.ui.ac.id

Abstrak

Kadar kualitas udara yang ada pada suatu daerah menjadikan tolak ukur keamanan dan kebersihan daerah tersebut. Begitu juga dengan area tertutup seperti parkir basement. Persentase kadar gas yang terperangkap di parkir basement jauh lebih tinggi dibandingkan tempat tertutup lainnya. Hal ini dikarenakan pada area tersebut terdapat aktivitas keluar masuk kendaraan sehingga potensi terperangkapnya gas karbon monoksida (CO) yang berasal dari kendaraan sangat tinggi. Berdasarkan kondisi tersebut diperlukan untuk dibuat sistem monitoring kadar gas CO di area parkir. Pada penelitian ini, dibuat sistem monitoring kadar gas CO dengan metode real-time berbasis Internet of Things (IoT). Sensor gas yang dipakai adalah MQ-7 yang digabungkan dengan mikrokontroler Arduino Uno. Untuk jalur pengiriman data secara real-time digunakan WiFi modul ESP8266. Data yang telah diambil oleh sensor akan disimpan dalam ThingSpeak cloud melalui ESP8266. Berdasarkan hasil monitoring didapatkan nilai kadar gas CO parkir basement cukup tinggi, yaitu dengan nilai rata-rata 15,52 ppm dan nilai tertinggi ada pada 17,27 ppm.

Kata Kunci: basement, IoT, karbon monoksida, monitoring, real-time.

Abstract

Air quality index in an area is an index for reporting daily air quality for safety and cleanliness of the area, likewise with closed spaces such as basement parking. The percentage of gas trapped in the basement parking lot is much higher than in other enclosed places. It is because, in that area, there is an activity in and out of the vehicle so that the potential for trapping carbon monoxide (CO) gas coming from the vehicle is very high. Under these conditions, it is necessary to set up a CO gas content monitoring system in the parking area. In this study, the real-time method based on the Internet of Things (IoT) is used to create a CO gas content monitoring system. The gas sensor used is the MQ-7, that is combined with the Arduino Uno microcontroller. The WiFi module ESP8266 is used for the data transmission path in real-time. The researcher used ThingSpeak cloud via ESP8266 to store the data that has been taken by the sensor. Based on the monitoring results, the basement parking CO gas value is quite high, with an average value of 15.52 ppm, and the highest value is at 17.27 ppm.

Keywords: basement, carbon monoxide, IoT, monitoring, real-time.

PENDAHULUAN

Nilai index kualitas udara menjadi perhatian utama bagi kota-kota besar, terutama di kawasan ibu kota. Hal ini dikarenakan nilai tersebut menjadi tolak ukur bagi level kesehatan dan kenyamanan suatu daerah [1]. Kota-kota besar, contohnya Jakarta, pasti memiliki banyak gedung-gedung pencakar langit dan bangunan-bangunan besar yang dipakai untuk aktivitas perkantoran, perbelanjaan, dan penginapan. Lingkungan seperti itu mengakibatkan beberapa kota besar saat ini memiliki sedikit lahan kosong untuk dijadikan tempat parkir. Alhasil, gedung-gedung tersebut pun harus membuat tempat parkir dengan jenis *indoor* atau *basement* agar dapat menampung kendaraan yang terparkir. Kondisi seperti ini mengakibatkan kuantitas interaksi manusia dengan lingkungan sekitar menjadi sangat tinggi ditambah dengan jumlah penduduk yang sangat tinggi di kota-kota besar, pasti banyak juga aktivitas yang terjadi di gedung-gedung pencakar langit atau bangunan besar tersebut baik di saat hari kerja (*workday*) maupun akhir pekan (*weekend*) [2].

Aktivitas masyarakat perkotaan yang meningkat biasanya berdampak dengan penggunaan kendaraan pribadi yang berlebih dibandingkan dengan angkutan umum. Hal ini menyebabkan kadar kualitas udara menjadi menurun karena meningkatnya polusi udara. Berdasarkan kondisi ini diperlukan alat yang dapat mengukur kadar polusi udara [3,4]. Gas

karbon monoksida (CO) yang dikeluarkan oleh kendaraan memiliki kontribusi sekitar 70% dalam pencemaran kualitas udara. CO merupakan salah satu gas yang sangat berbahaya bagi makhluk hidup, khususnya manusia. Hal ini dikarenakan jenis gas CO yang memiliki sifat tidak berbau, tidak berasa dan tidak berwarna sehingga sangat sulit untuk dideteksi secara kasat mata. Gas ini juga memiliki sifat yang mudah masuk ke dalam aliran darah manusia sehingga sangat berbahaya.

Pembuatan alat untuk mendeteksi keberadaan suatu gas sudah dilakukan pada beberapa penelitian terdahulu. Salah satunya adalah penelitian mengenai teknologi Narrowband IoT (NB-IoT), yang jika dibandingkan dengan jaringan Generasi ke-4 (4G), ZigBee dan teknologi komunikasi jarak pendek lainnya, memiliki karakteristik kapasitas besar, jangkauan luas, biaya rendah, dan konsumsi daya rendah [5]. H. Amer, dkk membuat monitoring kualitas udara dengan versi *low budget* dengan cara menggunakan *NodeMCU* dan sistem komunikasi *Local Area Network* (LAN) [6]. Kiran Patil V, dkk membuat alat monitoring gas yang jadi penyebab efek Rumah Kaca (Greenhouse), yaitu karbon dioksida (CO₂), CO, Methane (CH₄), dan Nitrous Oxide (N₂O) [7]. Pendeteksian yang dilakukan adalah menggunakan Xbee dan Ethernet Shield sebagai alat komunikasi yang terhubung ke mikrokontroler Arduino Uno. Alat ini melakukan pendeteksian dengan menggunakan

metode *real-time*. Lalu data yang didapat akan dikirimkan ke *Thingspeak cloud* dengan berbasis IoT. Sistem pendeteksian itu akan memberikan notifikasi kepada pengguna tertentu melalui *Short Message Service (SMS)* notifikasi. Sistem pendeteksian ini sayangnya hanya dipakai untuk mendeteksi di area terbuka dan menggunakan sistem komunikasi *Point-to-point*.

Gagan Parmar, et al. membuat sistem monitoring polusi udara berbasis IoT dengan model *Low-cost* [8]. *Prototype* ini dibuat menggunakan sensor gas semikonduktor murah yang dikombinasikan dengan *Wireless Fidelity (WiFi)* modul. Sensor gas yang dipakai dapat mengukur kadar gas CO, CO₂, Sulfur Dioksida (SO₂), dan Nitrogen Dioksida (NO₂). Data yang telah diambil melalui sensor akan diolah menggunakan mikrokontroler Arduino Raspberry pi 3 yang terhubung ke *webserver*. Visualisasi data yang dihasilkan akan berbentuk *MEAN stack*.

Berdasarkan dari penelitian-penelitian yang sudah ada, sistem monitoring gas yang telah dibuat belum pernah ada yang mencoba dalam ruang tertutup seperti di area parkir *basement*. Area tertutup seperti parkir *basement* merupakan area yang sangat rentan terperangkapnya gas berbahaya. Meningkatnya aktifitas dalam gedung parkir *indoor* atau *basement* secara otomatis akan menghasilkan peningkatan gas CO yang berasal dari kendaraan bermotor yang terparkir di parkir tersebut. Kadar gas yang tinggi dapat menyebabkan keracunan gas CO bila tidak adanya monitoring

kondisi udara di parkir *indoor* atau *basement*.

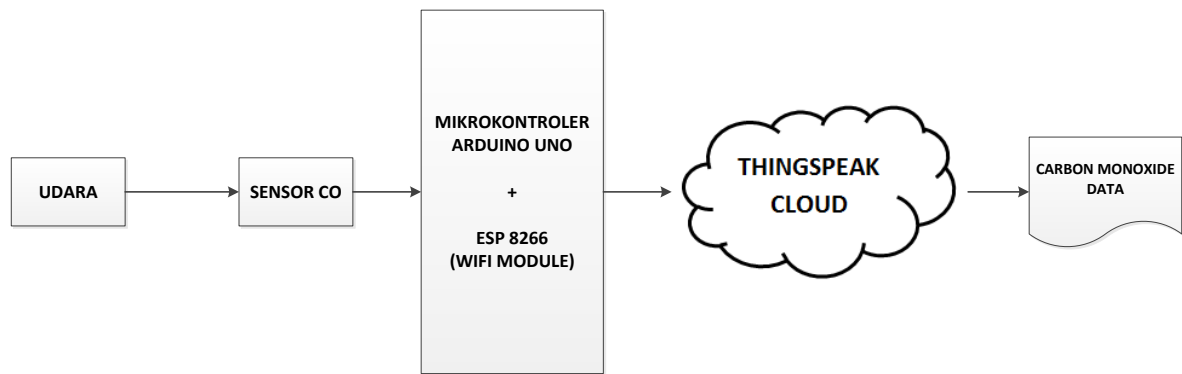
Pembuatan alat untuk memonitor kadar gas CO dalam parkir cukup penting sebagai salah satu solusi terkini untuk menanggulangi resiko terjadinya tingkat gas berbahaya serta terjadinya keracunan gas. Penelitian ini bertujuan untuk menciptakan alat yang praktis dan mudah digunakan untuk memonitoring gas CO dalam parkir *indoor* atau *basement* sehingga kejadian keracunan atau peningkatan gas CO berlebih tidak terjadi dan dapat segera di tanggulangi.

METODE PENELITIAN

Pembuatan sistem monitoring gas CO parkir *basement* berbasis IoT ini dilakukan dalam beberapa tahapan yang nantinya akan berpengaruh pada proses pengambilan data dan data yang dihasilkan. Berikut ini dijelaskan tahapan-tahapan penelitian yang dilakukan.

Diagram Alur Sistem

Sistem monitoring gas CO ini dibuat untuk melakukan pengukuran kadar gas CO yang terdapat pada area parkir *basement* mall-mall di Jakarta. Alat ini dapat digunakan secara kontiniu dengan sistem komunikasi berbasis IoT. Data yang didapatkan akan disimpan menggunakan *cloud storage* dari ThingSpeak Cloud dengan cara dikirimkan secara *real-time* menggunakan modul WIFI ESP 8266 yang terhubung langsung dengan sensor gas dan mikrokontroler Arduino Uno.



Gambar 1. Diagram alur sistem monitoring gas CO pada parkir *basement* berbasis IoT

Berdasarkan Gambar 1, sistem monitoring bekerja dengan cara sensor gas yang terhubung dengan rangkaian mikrokontroler Arduino Uno dan modul WIFI ESP8266 mengambil data kadar gas CO dari lingkungan sekitar area *basement*. Setelah sensor mendapatkan data selanjutnya data tersebut di proses oleh mikrokontroler Arduino Uno dan diteruskan ke ThingSpeak cloud melalui modul WIFI. Pada ThingSpeak cloud data diolah menjadi grafik yang nantinya akan dapat dilihat oleh pihak pengguna otoritas sehingga data dapat dilihat secara *real-time*.

ThingSpeak adalah *platform cloud IOT open source* dan API yang digunakan untuk menyimpan data sensor secara real time dan menganalisis data kapan pun diperlukan menggunakan perintah *HyperText Transfer Protocol* (HTTP) [9]. Untuk memasukkan data ke ThingSpeak harus dibuat akun terlebih dahulu. Setelah akun dibuat, pengguna akan mendapatkan saluran data (*channel*) yang bisa diisi dengan data dari sistem yang terhubung ke ThingSpeak.

Pengguna dapat membuat channel hingga 8 buah channel yang bisa diisi dengan data yang berbeda-beda. Penggunaan ThingSpeak cloud ini sangat memungkinkan untuk sistem monitoring yang memakai sistem komunikasi *multi-to-point* dengan maksimum modul sistem monitoring 8 buah. Data yang dimasukkan ke dalam cloud ini akan direpresentasikan dalam bentuk grafik dari masing-masing sensor.

Data Sampel

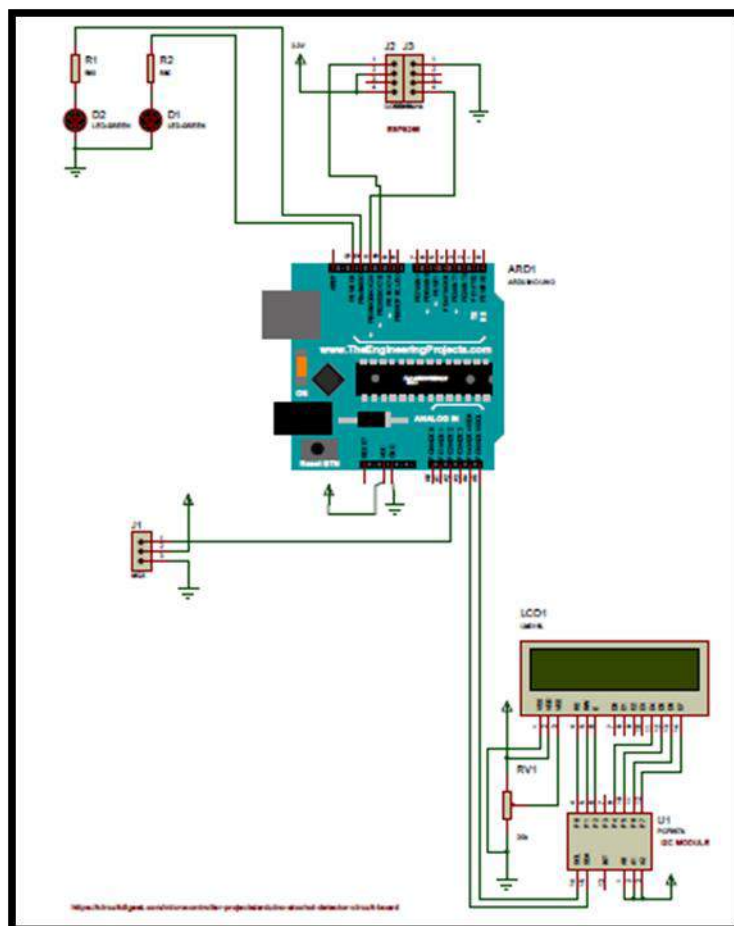
Pengambilan data untuk sistem monitoring ini akan dilakukan pada parkir *basement* mall-mall di Jakarta. Metode penelitian yang dipakai pada penelitian ini adalah dengan metode kuantitatif. Dengan begitu diperlukan perhitungan statistik untuk data-data yang diambil pada sistem monitoring ini. Pada penelitian ini, perhitungan statistik yang dipakai adalah menggunakan Teori Slovin. Untuk mendapatkan data monitoring gas CO pada parkir *basement* mall di Jakarta diperlukan perhitungan statistik agar dapat menentukan

berapa jumlah sample data minimal yang diperlukan mengingat jumlah parkir mall yang memiliki basement jumlahnya sangat banyak. Dengan menggunakan teori Slovin sampel data yang diperlukan dapat mewakili keseluruhan populasi yang ada [10]. Berikut adalah rumus Slovin yang dipakai untuk penentuan sample minimal yang diperlukan :

$$n = \frac{N}{1 + Ne^2} \quad (1)$$

Pada persamaan (1), nilai N adalah nilai dari jumlah total populasi mall di Jakarta yang memiliki parkir *basement*. Kemudian untuk nilai n adalah jumlah sampel minimal. Sedangkan nilai e adalah margin *error*. Berdasarkan rumus pada persamaan (1) nilai e ditentukan besarnya oleh peneliti berdasarkan tingkat kesalahan dalam bentuk persentase. Sehingga nilai e bisa disesuaikan dengan jenis penelitian yang dilakukan.

Sistem Monitoring berbasis Arduino Uno



Gambar 2. Skema diagram arsitektur sistem monitoring gas CO

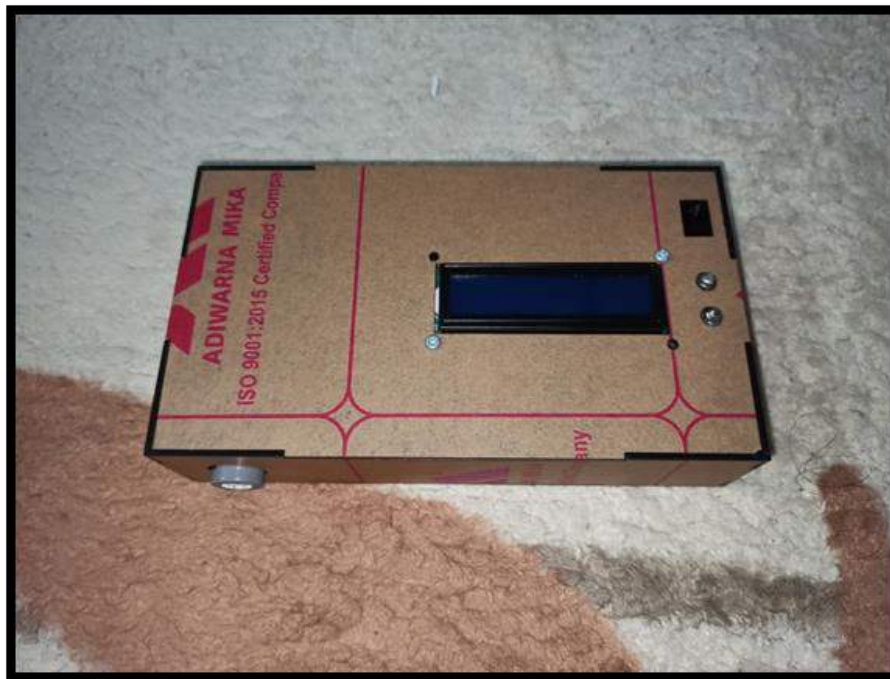
Pada modul monitoring ini juga dipasang *Liquid Crystal Display* (LCD) yang

berfungsi untuk pemantauan langsung modul monitoring saat dilakukan pengecekan

perawatan. Penggunaan lampu *Light Emmiting Diode* (LED) berwarna merah dan hijau juga difungsikan sebagai alat bantu notifikasi selain data ditampilkan melalui LCD panel. Semua alat-alat ini disimpan di dalam sebuah box kecil yang terbuat dari mika sehingga alat monitoring ini mudah dilepas pasang jika diaplikasikan.

Sesuai dari Gambar 2, sensor gas MQ 7 dihubungkan ke mikrokontroler Arduino Uno ke pin analog A2. Untuk WiFi modul

ESP8266 dihubungkan ke Arduino ke pin digital nomor 10 dan 11. Lalu untuk LCD dihubungkan dengan modul *Inter Integrated Circuit* (I2C) yang merupakan komunikasi jenis serial dua arah yang menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Rangkaian LCD dan modul I2C ini dihubungkan ke Arduino melalui pin analog A4 dan A5. Serta dua buah LED dihubungkan ke pin digital nomor 12 dan 13.



Gambar 3. Penampakan modul sistem monitoring gas CO bagian luar

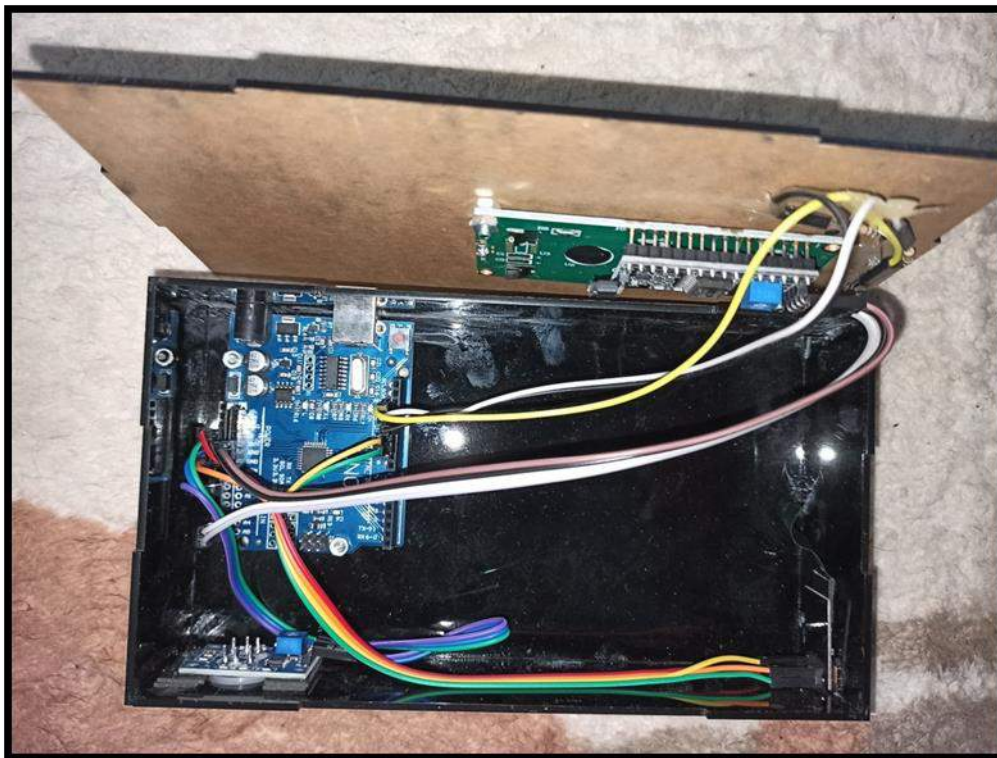
Pada Gambar 3 ditunjukkan penampakan dari bagian luar dari modul sistem monitoring gas CO. Pada bagian atas penampang modul terdapat LCD ukuran 16x2 sebagai alat bantu untuk menampilkan status koneksi modul ke server dan besar kadar gas CO yang dideteksi. Lalu ada 2 buah LED yang berwarna merah

sebagai tanda pengukuran gas melebihi ambang batas toleransi dan warna hijau untuk tanda udara masuk kategori aman karena kadar gas CO di bawah batas toleransi. Pada bagian samping terdapat sensor gas MQ7 yang berguna untuk mengambil data kadar gas CO dan modul WiFi ESP8266 yang

berfungsi untuk menghubungkan modul sistem monitoring ini ke server ThingSpeak *cloud*.

Pada bagian dalam modul terdapat mikrokontroler Arduino Uno yang terhubung sensor gas MQ7, LCD, LED dan modul WiFi. Pada rangkain modul ini, semua bagian-

bagian perangkat dihubungkan dengan menggunakan kabel *jumper* jenis tembaga (*cooper*). Pada modul ini, kabel jumper yang dipakai adalah yang warna warni agar dapat memudahkan *maintenance* alat. Setelah modul dibuka maka penampakan bagian dalam modul ditunjukkan pada Gambar 4 berikut.



Gambar 4. Penampakan modul sistem monitoring gas CO bagian dalam

Sistem monitoring gas CO ini dibuat dengan menggabungkan semua komponen-komponen yang diperlukan ke dalam sebuah kotak berbahan mika. Hal ini dilakukan agar sistem ini tidak mudah terekspose dengan area sekitar, aman, dan juga menjadikannya mudah untuk dibawa-bawa. Selain itu, dengan menggunakan kotak mika sebagai wadah, membuat susunan kabel menjadi lebih rapi.

HASIL DAN PEMBAHASAN

Penelitian ini dilakukan pada parkir *basement* mall-mall di Jakarta. Waktu pengambilan data diatur agar dilakukan pada waktu yang setara, yaitu pada hari Sabtu dan Minggu dari pukul 13.00 hingga pukul 20.00 WIB. Hal ini dilakukan karena pada rentang waktu tersebut tingkat aktivitas di parkir *basement* mall-mall

di Jakarta berada pada kondisi yang sangat tinggi. Sehingga data yang dikumpulkan dapat diolah dengan variabel yang sama walaupun dilakukan di beberapa tempat yang berbeda-beda.

Sebelum dilakukan pengukuran data, hal yang harus dilakukan adalah melakukan pengaturan pada sistem monitoring agar data yang diambil bisa diterjemahkan menjadi bentuk satuan *Part per Million* (PPM). Pembacaan data cukup menggunakan data matematis yang ada pada datasheet MQ-7 karena sensor gas yang dipakai adalah sensor MQ-7[12]. Akan tetapi, hasil yang didapatkan menjadi tidak valid karena perhitungan matematis dari sensor MQ-7 yang berasal dari *datasheet* secara langsung menjadikan data terlihat naik-turun sehingga sulit untuk dilakukan analisa. Modifikasi matematis dari *datasheet* sensor MQ-7 diperlukan untuk mengatasi hal tersebut.

Modifikasi perhitungan yang dilakukan adalah dengan cara mengubah nilai *Analog and Digital Converter* (ADCvalue) menjadi nilai voltase [13]. Pada perhitungan ini, nilai voltase yang dipakai adalah 5 volt.

$$VRL = ADCvalue \times \frac{5}{1024} \quad (2)$$

Nilai *ADCvalue* diubah menjadi nilai *VRL* menggunakan persamaan (2). Setelah nilai *ADCvalue*-nya diubah, selanjutnya adalah mencari nilai resistensi pada sensor (*Rs*). Untuk mencari nilai *Rs*, diperlukan nilai

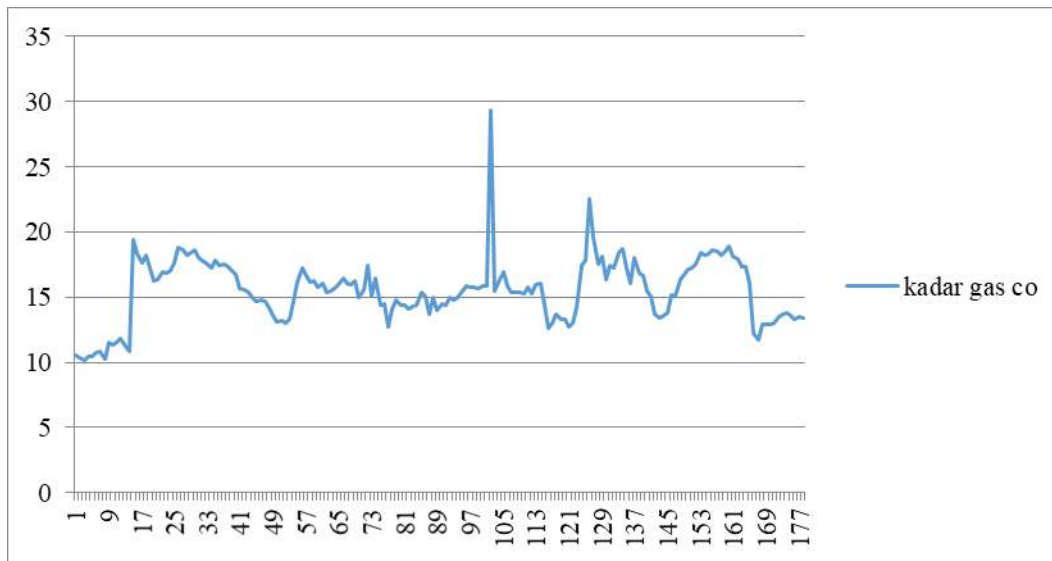
tegangan yang masuk pada sensor (*Vc*) yang biasanya sama dengan nilai voltase, yaitu 5 volt. Selanjutnya nilai tegangan output rangkaian (*VRL*) bisa didapatkan dengan mengukur fungsi ADC. Kemudian didapatkan nilai resistensi pada rangkaian (*RL*).

$$Rs = \left(Vc \times \frac{RL}{VRL} \right) - RL \quad (3)$$

$$ppm = 100 \times \left(\frac{Rs}{Ro} \right)^{-1,53} \quad (4)$$

Dengan menggunakan persamaan (3) didapatkan nilai resistensi saat udara bersih (*Rs*). Sebelum mencari nilai ppm, diperlukan nilai (*Ro*) tetapi karena adanya keterbatasan alat pembanding untuk mencari nilai *Ro*, maka akan dicari dengan menggunakan pendekatannya saja. Nilai *Ro* = nilai *Rs* pada CO kadar 100 ppm. Untuk membuat kondisi lingkungan dengan kadar 100 ppm agak sulit. Oleh karena itu, untuk perhitungan ini digunakan kadar ppm di udara normal (iklim udara ruangan normal). Kadar CO di udara ruangan normal adalah 5-15 ppm [14]. Karena nilai *Ro* adalah suatu tetapan maka dalam kondisi 20 ppm atau 100 ppm adalah sama. Setelah mendapatkan nilai *Ro*, nilai ppm bisa dicari dengan menggunakan persamaan (4).

Dari hasil pengukuran data sebanyak 20 parkir *basement* mall di Jakarta didapatkan hasil dalam grafik pada Gambar 5 berikut.



Gambar 5. Hasil pengukuran kadar gas CO di parkir *Basement* mall di Jakarta

Berdasarkan pada Gambar 5 dari grafik hasil pengukuran kadar gas CO di parkir *basement* mall di Jakarta dapat disimpulkan bahwa hasil pengukuran memiliki trend yang naik dan berada diatas kadar batas toleransi untuk kesehatan. Berdasarkan pengukuran tersebut didapat bahwa modus nilai ada pada kadar 17,27 ppm dan rata-rata kadar udara adalah 15,52 ppm. Hal ini membuktikan bahwa pada area parkir *basement* diperlukan suatu alat yang dapat melakukan pertukaran udara dengan baik sehingga kadar gas CO yang terperangkap di dalam area parkir *basement* bisa dibuang keluar dan kadar kualitas udara menjadi jauh lebih baik.

KESIMPULAN DAN SARAN

Pada sistem monitoring gas CO pada parkir *basement* ini dirasa sudah cukup baik untuk diaplikasikan secara nyata. Hal ini dikarenakan adanya keunggulan dari segi

ringkas alat yang dipakai dan proses penyimpanan data yang sudah menggunakan *cloud*. Selain itu, pengiriman data juga sudah berbasis IoT dengan memanfaatkan WiFi modul ESP8266 sebagai alat untuk mengirimkan data ke tempat penyimpanan *cloud* secara *real-time*. Tetapi alat ini belum memiliki integrasi dengan sistem ventilasi yang menjadi fasilitas pada parkir *basement*. Sistem monitoring ini bisa dikembangkan dengan lebih baik lagi dan dapat diintegrasikan dengan sistem ventilasi yang ada pada parkir *basement*.

DAFTAR PUSTAKA

- [1] P. Spachos, dan D. Hatzinakos, "Real-Time Indoor Carbon Dioxide Monitoring Through Cognitive Wireless Sensor Network", *IEEE Sensors Journal*, vol. 16, no. 2, hal. 506 - 514, 2016

- [2] Canadian Centre for Occupational Health and Safety, "Indoor Air Quality," July, 2016. [Daring]. Tersedia: https://www.ccohs.ca/oshanswers/chemicals/iaq_intro.html. [Diakses: 25 September, 2019]
- [3] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes," *IEEE Sensors Journal*, vol. 13, no. 10, hal. 3846-3853, 2013.
- [4] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang dan H. Ning, "A survey on gas sensing technology," *Sensors*, vol. 12, no. 7, hal. 9635–9665, 2012.
- [5] Song Hongru, Wang Yihuai, Yang Fan. Design and implementation of intelligent gas meter system based on narrowband IoT. *Sensors & micro Systems*, vol. 38, no. 3, hal. 113-116, 2019.
- [6] H. Aamer, R. Mumtaz, H. Anwar, and S. Posland, "A Very Low Cost, Open, Wireless, Internet of Things (IoT) Air Quality Monitoring Platform," In Proc. 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), 2018, hal. 102-106.
- [7] K. Patil V, Kiran B.N, dan R. Srinath, "Internet Of Things For Green House Gas Detection System," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 5, hal. 3053 – 3055, 2016.
- [8] G. Parmar, S. Lakhani, and M. K. Chattopadhyay, "An IoT Based Low Cost Air Pollution Monitoring System," In Proc. Recent Innovations in Signal Processing and Embedded Systems (RISE-2017) 27-29 October, 2017, hal.524 – 528.
- [9] <https://thingspeak.com/> [Diakses: 29 Oktober, 2019]
- [10] Anwar Hidayat, "Cara Perhitungan Rumus Slovin Besar Sampel Minimal", Desember, 2017. [Daring]. Tersedia: <https://www.statistikian.com/2017/12/hitung-rumus-slovin-sampel.html>. [Diakses: 29 Oktober, 2019]
- [11] Ivan Grokhotkov, "ESP8266 Arduino Core Documentation Release 2.4.0," Jan 02, 2018. [Online] Tersedia: <https://datasheet4u.com/datasheet-parts/ESP8266-datasheet.php?id=853652>. [Diakses: 8 November, 2019]
- [12] www.hwsensor.com, "Technical Data MQ-7 Gas Sensor", Huawei Electronics Co. [Diakses: 4 November, 2019].
- [13] Pambudi Giri Wahyu. "Cara menggunakan modul deteksi gas CO MQ-7 dengan Arduino," July, 2018. [Daring]. Tersedia: <https://www.cronyos.com/cara-menggunakan-modul-deteksi-gas-co-mq7-dengan-arduino/>. [Diakses: 4 Oktober, 2019].
- [14] Clean Air Initiative for Asian Cities Centre (CAI-Asia), "Air Quality in

Asia: Status and Trends”, 2010 Edition.

[Daring]. Tersedia:

<http://www.cleanairinitiative.org>.

[Diakses: 20 Oktober, 2019].

SISTEM PAKAR UNTUK DIAGNOSA AWAL PENYAKIT LAMBUNG MENGGUNAKAN METODE DEMPSTER-SHAFER BERBASIS WEB

¹Rizky Ardiansyah, ²Fuziah Fauziah, ³Andria Ningsih
Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional
Jl. Sawo Manila, Pejaten Ps Minggu Jakarta 12520
¹rizkiardiansyah3@gmail.com, ²fauziah@civitas.unas.ac.id,
³andrianingsih@civitas.unas.ac.id

Abstrak

Berdasarkan situs kesehatan Health Line menyebutkan 60% orang dewasa akan mengalami masalah asam lambung. Di Indonesia penderita GERD pada tahun 2018 adalah 27,4% dan menempati urutan 10 besar penyakit dengan penderita terbanyak. Hal ini dipicu juga kurangnya pengetahuan masyarakat terhadap informasi jenis-jenis penyakit serta gejala yang terdapat pada lambung. Selain itu jika gejala penyakit lambung muncul maka masyarakat kadang tidak segera berkonsultasi kepada dokter. Oleh karena itu pada penelitian ini dibuat sistem pakar yang dapat merekomendasikan diagnosa awal penyakit pada lambung berdasarkan gejala yang dipilih. Sistem pakar merupakan sistem computer yang digunakan untuk menyerap pengetahuan dan keahlian manusia. Aplikasi sistem pakar ini dibuat menggunakan bahasa pemrograman PHP dan database MySQL. Object pada penelitian ini hanya berfokus pada penyakit lambung yaitu Gastroparesis, GERD, Dispepsia, dan Tukak Lambung. Metode Dempster Shafer merupakan teori matematika berdasarkan fungsi kepercayaan dan pemikiran yang masuk akal. Pengujian akurasi sistem dengan menggunakan 20 sample mendapatkan presentase sebesar 95%.

Kata kunci: Dempster-Shafer, GERD, Penyakit Lambung, PHP, Sistem Pakar.

Abstract

According to the Health Line media website publishes health Information, 60% of adults will experience stomach acid problems. In Indonesia, patients with GERD in 2018 are 27.4% and ranks in the top 10 with the most sufferers. This is also triggered by the lack of public knowledge of information on the types of diseases and symptoms that occur in the stomach. In addition, if symptoms of gastric disease appear, the community sometimes does not immediately consult a doctor. Therefore, in this study an expert system was created that could recommend an initial diagnosis of diseases of the stomach based on the symptoms chosen. Expert systems are computer systems that are used to absorb human knowledge and expertise. This expert system application is created using the PHP programming language and MySQL database. The object of this study only focuses on gastric disease, namely Gastroparesis, GERD, Dyspepsia, and Gastric Ulcer. The Dempster Shafer method is a mathematical theory based on the function of belief and reasoning. Testing the accuracy of the system by using 20 samples get a percentage of 95%.

Keywords: Dempster-Shafer, Gerd, Gastric Disease, PHP, Expert System.

PENDAHULUAN

Menurut situs kesehatan Health Line menyebutkan 60% orang dewasa akan

mengalami masalah asam lambung. Penyakit asam lambung adalah suatu hal yang tidak dapat dianggap biasa, karena asam lambung dapat menyebabkan penyakit lambung lain

datang. Sebagai contoh penyakit lambung yaitu GERD. Menurut Yayasan *Gastroenterologi Indonesia (YGI)* persentase angka kejadian *GERD* di Indonesia pada tahun 2018 adalah 27,4%. Berdasarkan catatan dari Kementerian Kesehatan penyakit yang berhubungan dengan *gastrointestinal* berada pada 10 besar penyakit terbanyak penderitanya di Indonesia.

Lambung merupakan organ dalam tubuh manusia yang cukup rentan terinfeksi bakteri atau terluka. Salah satu penyebab gangguan kesehatan lambung diantaranya asam lambung yang meningkat [1]. Meningkatnya asam lambung dapat diakibatkan dari beberapa *factor* salah satunya adalah jenis makanan yang dikonsumsi. Terlalu sering mengonsumsi makanan yang berlemak dan buah yang rasanya masam dapat mengakibatkan asam lambung menjadi naik sehingga kinerja lambung tidak maksimal. Terdapat beberapa penyakit pada lambung antara lain *dyspepsia*, *Gastroesophageal Reflux Disease (GERD)*, tukak lambung, *gastroparesis*, *Gastroparesis* dan kanker lambung. Beberapa penyakit pada lambung tersebut dapat diakibatkan dari bakteri pada lambung, dan juga iritasi pada lambung yang disebabkan oleh meningkatnya asam lambung yang dapat menyebabkan terganggunya kinerja lambung. Kesehatan lambung merupakan suatu kebutuhan primer bagi manusia, namun terkadang beberapa orang kurang memperhatikan kesehatan pada lambungnya [1]. Semua manusia menyadari bila kesehatan lambungnya mengalami beberapa gangguan, tetapi sebagian besar tidak

mengetahui penyakit apa yang sedang diderita pada lambungnya serta bagaimana cara meredakan dan mencegahnya. Oleh karena itu pada penelitian ini dibuat suatu sistem pakar untuk memudahkan masyarakat umum dalam mendiagnosa dini penyakit lambung.

Pada penelitian terdahulu telah dibuat sistem pakar untuk mendiagnosis penyakit lambung menggunakan metode *dempster shafer* dengan 20 data uji. Hasil ujicoba perbandingan antara sistem dengan hitungan manual didapatkan tingkat keakurasian 88,40% [1]. Pada penelitian lain penerapan metode *dempster shafer* dengan menggunakan 100 data rekam medik pada 2 rumah sakit untuk dibandingkan dengan hasil *diagnose* sistem menghasilkan tingkat akurasi sebesar 84% [2]. Sistem pakar menggunakan metode *dempster shafer* dengan 104 data rekam medik yang digunakan untuk dibandingkan dengan *dignosa* sistem menghasilkan tingkat keakurasian 94,23% [3]. Pada penelitian lain menggunakan metode *dempster shafer* dengan 40 data uji diperoleh hasil perbandingan antara hasil *diagnosa* sistem dengan *diagnose* pakar menghasilkan tingkat akurasi 95% [4]. Pada penelitian lain mengenai sistem pakar metode *dempster shafer* menggunakan 35 data uji perbandingan antara *diagnose* sistem dengan *diagnose* pakar menghasilkan tingkat akurasi 91,42% [5].

Pada penelitian lain menghasilkan kesimpulan bahwa tingkat akurasi 90% berdasarkan 20 data sample pengujian antara sistem dengan pakar [6]. Pada penelitian

mengenai perbandingan metode *certainly factor* dengan *dempster shafer* menggunakan 20 data uji didapatkan hasil tingkat keakurasian *dempster shafer* lebih tinggi (90%) dibandingkan dengan metode *certainly factor* (85%)[7]. Pada penelitian lain tentang Teknik rekomendasi pengobatan menggunakan metode *dempster shafer* teori *dempster shafer* memiliki tingkat fleksibilitas untuk mewakili dan menggabungkan beberapa jenis bukti yang diperoleh dari beberapa sumber[8].

Pada penelitian terdahulu tersebut terdapat beberapa kekurangan diantaranya sistem tersebut hanya membahas beberapa penyakit dan tidak terdapat factor yang menjadikan penyebab penyakit tersebut dan solusi untuk pencegahan penyakit tersebut. Berdasarkan permasalahan tersebut maka pada penelitian ini dibuat sebuah sistem yang dapat melengkapi kekurangan pada penelitian sebelumnya. Identifikasi masalah pada penelitian ini meliputi kurangnya informasi pada masyarakat tentang penyakit yang menyerang lambung seperti *GERD*, *Gastritis*, *Dispepsia*, dan *Tukak Lambung*, dan juga kurangnya pengetahuan tentang gejala-gejala yang dapat menyebabkan penyakit tersebut. Pada penelitian ini pembahasan dibatasi hanya sistem pakar menggunakan metode *Dempster Shafer* dan hanya *focus* pada penyakit lambung *GERD*, *Gastroparesis*, *Dispepsia*, dan *Tukak Lambung*. Hasil penelitian diharapkan dapat memberikan output berupa hasil rekomendasi diagnosa awal, solusi dari penyakit yang dihasilkan, dan langkah-langkah pencegahan

penyakit tersebut. Aplikasi ini dirancang dengan menggunakan bahasa pemrograman PHP dengan MySQL. Tujuan dari penelitian ini adalah pembuatan sistem pakar yang dapat memberikan informasi berupa rekomendasi diagnosa awal, dan langkah-langkah pencegahan untuk masyarakat berdasarkan aturan yang ada dari gejala yang dipilih, menerapkan metode *dempster shafer* dalam membantu memberikan rekomendasi diagnosa awal penyakit lambung.

METODE PENELITIAN

Sistem pakar merupakan sistem *computer* yang digunakan untuk menyerap pengetahuan dan keahlian manusia [9]. Sistem pakar dirancang agar dapat membantu dalam memberikan informasi dan mengambil keputusan seperti yang dilakukan seorang pakar.

Terdapat dua bagian pokok dalam sistem pakar, yaitu lingkungan pengembangan, dan lingkungan konsultasi. Lingkungan pengembangan digunakan untuk membangun dari segi komponen dan basis pengetahuan. Lingkungan konsultasi digunakan oleh orang yang ingin mendapatkan informasi untuk berkonsultasi [10].

Dempster Shafer

Teori *dempster shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief functions* (fungsi kepercayaan) dan *plausible reasoning* (pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti)

untuk mengkalkulasi kemungkinan dari suatu peristiwa. Secara umum teori *dempster shafer* ditulis dalam suatu interval: [*Belief,Plausibility*]

Belief (Bel) merupakan ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Nilai *Belief* didapatkan dari pakar yang diperoleh dari ilmu pengetahuan pakar Nilai Bel ini berada dalam kisaran [0...1], Jika nilai Bel = 0 artinya tidak ada *evidence* dan Bel = 1 artinya kepastian. Fungsi *belief* dapat diformulasikan sebagai berikut:

$$Bel(X) = \sum_{Y \subset X} m(Y) \quad (1)$$

Plausibility (Plau) merupakan ukuran ketidakpercayaan terhadap evidence/gejala. *Plausibility* juga bernilai 0 sampai 1. Jika kita yakin akan x, maka dapat dikatakan bahwa $Bel(X)=1$, dan $Pl(X)=0$. *Plausibility* akan mengurangi tingkat kepercayaan dari evidence. *Plausibility* (Plau) dinotasikan sebagai berikut:

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \subset X} m(Y) \quad (2)$$

Dimana:

$$Bel(X) = Belief (X)$$

$$Pls(X) = Plausibility (X)$$

$$m(Y) = \text{mass function dari } (Y)$$

Pada teori *dempster shafer* adanya *frame of discernment* yang dinotasikan dengan θ dan *mass function* yang dinotasikan dengan m . *frame of discernment* adalah semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan environment. Sedangkan *mass function* (m) dalam teori Dempster-Shafer adalah tingkat kepercayaan

dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m). Untuk mengatasi sejumlah *evidence* tersebut gunakan aturan yang lebih dikenal dengan *Dempster's rule of combination* yaitu:

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - K} \quad (3)$$

Dimana :

$$m_3(Z) = \text{mass function dari evidence } Z$$

$$m_1(X) = \text{mass function dari evidence } X$$

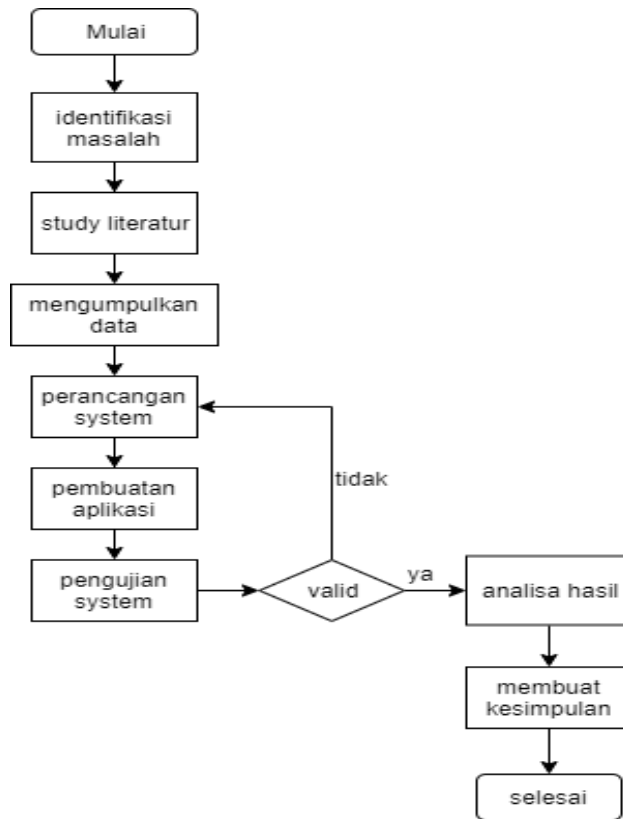
$$m_2(Y) = \text{mass function dari evidence } Y$$

$\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)$ adalah jumlah dari irisan pada perkalian $m_1(X)$ dan $m_2(Z)$.

K = perkalian dari *mass function* yang mengalami konflik *evidence* bila tidak terdapat irisan.

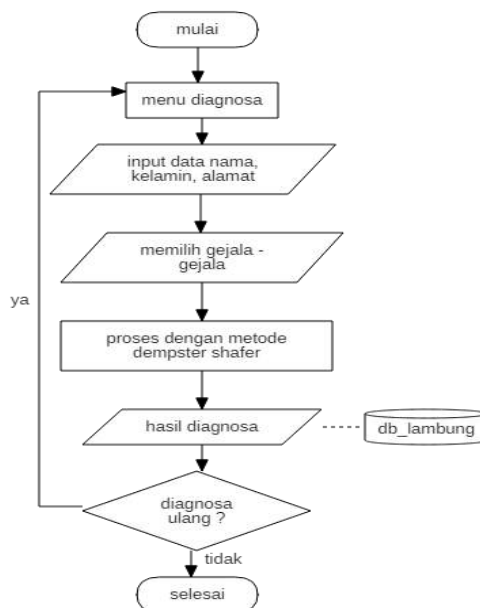
Pada Gambar 1 dijelaskan diagram alur penelitian. Proses pertama yang dilakukan adalah identifikasi masalah pada gejala-gejala penyakit yang termasuk dalam penyakit lambung. Tahap kedua dilakukan studi literatur yang menjadi referensi dalam mendapatkan informasi terkait. Tahap ketiga adalah proses pengumpulan data dengan metode wawancara dan mengutip beberapa bacaan yang terkait sistem pakar. Pada tahap keempat dilakukan perancangan sistem dan pembuatan aplikasi. Tahap terakhir adalah pengujian sistem untuk mengetahui apakah data informasi yang didapatkan valid atau tidak, jika tidak valid kembali ke tahap ke empat hingga tahap keenam. Analisis hasil merupakan proses penjabaran hasil yang sudah *valid*, dan tahap terakhir membuat kesimpulan hasil penelitian yang sudah di analisis.

Flowchart Penelitian



Gambar 1. Flowchart Penelitian

A. Flowchart Sistem

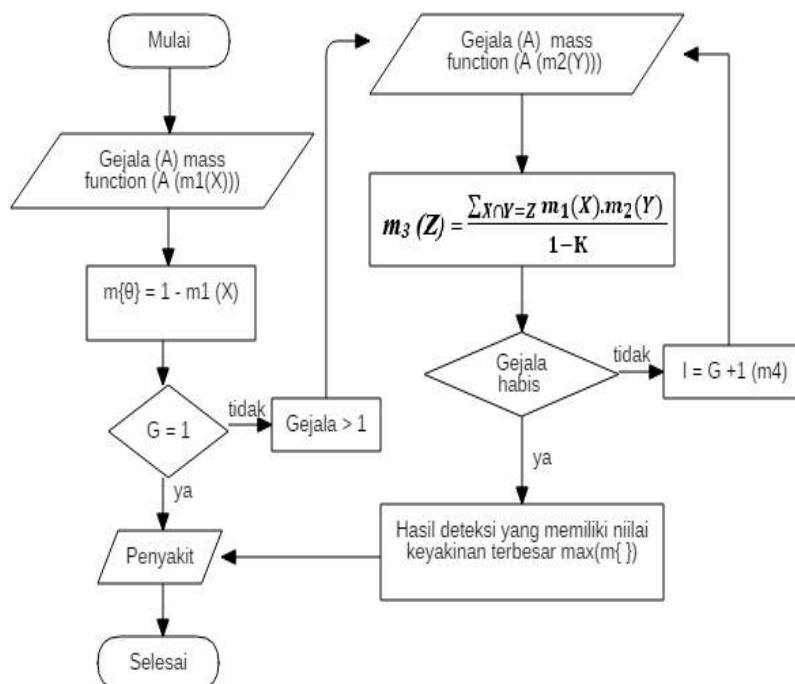


Gambar 2. Flowchart sistem

Pada Gambar 2 dijelaskan bagaimana alur jalannya aplikasi sistem. Aplikasi diawali dengan masuk ke dalam menu *diagnose*. User melakukan pengisian data diri pada menu *diagnose*, lalu user memilih gejala yang dialami. Gejala yang dipilih akan diproses menggunakan

metode *Dampster Shafer*, dan setelah proses perhitungan selesai maka akan ditampilkan hasil dari diagnosa dan hasil tersebut akan tersimpan ke dalam *database*. Jika ingin melakukan *diagnose* maka harus mengulang kembali ke menu *diagnose*, jika tidak maka selesai.

B. Flowchart Metode



Gambar 3. Flowchart Metode *Dampster Shafer*

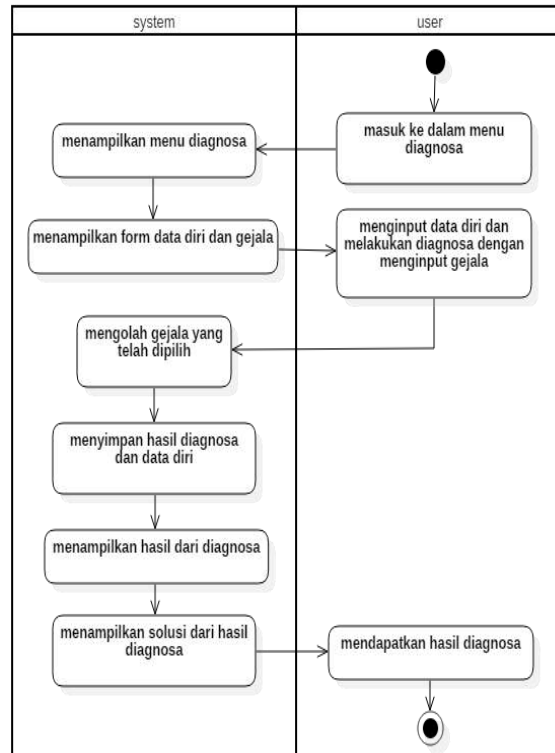
Pada Gambar 3 ditunjukkan diagram alur metode *Dampster Shafer* yang digunakan pada penelitian ini. Simbol X berarti penyakit, symbol I merupakan banyaknya iterasi gejala dan m merupakan *Mass Fuction*.

Pada Gambar 4 ditunjukkan *Activity Diagram* aplikasi sistem pakar. Diawali

dengan masuk ke diagnosa hingga memperoleh hasil diagnose.

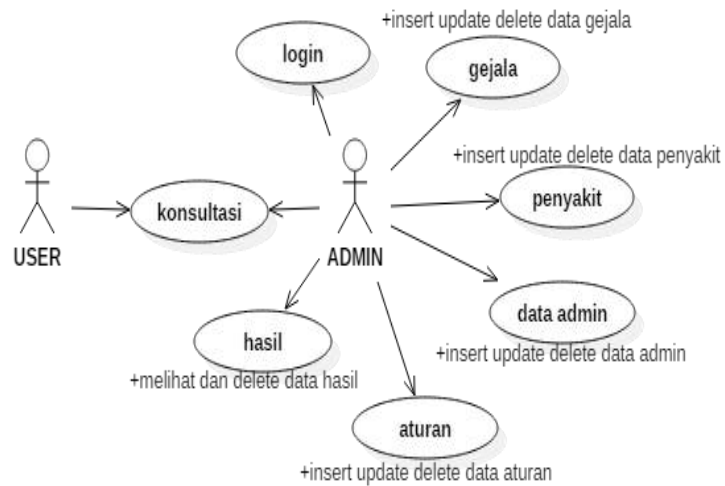
Pada Gambar 5 ditunjukkan *Usecase Diagram* dari aplikasi sistem pakar yang dibuat. Admin memiliki hak penuh pada sistem pada bagian admin termasuk insert, update, dan delete data. User hanya dapat melakukan konsultasi berdasarkan gejala yang dipilih.

C. Activity Diagram



Gambar 4. Activity Diagram

D. Usecase diagram



Gambar 5. Usecase Diagram

HASIL DAN PEMBAHASAN

Basis Pengetahuan

Hal yang pertama kali dilakukan dalam membuat sebuah sistem pakar adalah membuat struktur basis pengetahuan. Struktur basis pengetahuan dalam sistem pakar ini

adalah sebagai berikut :

1. Basis pengetahuan penyakit
2. Basis pengetahuan *factor* resiko
3. Basis pengetahuan gejala
4. Basis pengetahuan penyakit dan gejala
5. Basis pengetahuan nilai *believe*

Tabel 1. Basis pengetahuan penyakit

Kode Penyakit	Nama Penyakit	Pengertian
P1	<i>GERD</i>	GERD adalah penyakit yang menyebabkan munculnya rasa terbakar di dada akibat naiknya asam lambung
P2	Tukak Lambung	Tukak lambung adalah luka pada lambung yang menyebabkan keluhan sakit maag. Selain di lambung, luka tersebut dapat terbentuk di usus 12 jari atau bagian bawah kerongkongan
P3	<i>Gastroparesis</i>	Gastroparesis adalah gangguan pada otot lambung yang menyebabkan gerakan lambung untuk mendorong makanan ke usus menjadi lebih lambat. Gastroparesis ditandai dengan gejala berupa mual, muntah, dan mudah merasa kenyang.
P4	<i>Dispepsia</i>	Dispepsia adalah suatu kondisi yang bisa menyebabkan rasa tidak nyaman pada perut bagian atas karena asam lambung.

Pada Tabel.1 berisikan basis pengetahuan jenis-jenis penyakit pada lambung manusia. Setelah basis pengetahuan, selanjutnya adalah

menentukan pengetahuan factor resiko. Pada Tabel 2 diberikan basis pengetahuan factor resiko yang terdapat pada penyakit lambung.

Tabel 2. Basis pengetahuan faktor resiko

Kode	Nama
G1	Riwayat Penyakit DM
G2	Riwayat Pasca Operasi Lambung
G3	Riwayat meminum obat analgetik (aspirin)
G4	Penyakit lain yang disebabkan oleh infeksi : HIV/AIDS, Chron, dan Infeksi bakteri lainnya
G5	Perut tidak nyaman pasca makan berlemak
G6	Riwayat konsumsi obat antidepresi
G7	Sering Makan-makanan berlemak
G8	Riwayat pengobatan kanker seperti kemoterapi atau radioterapi
G9	Sering mengkonsumsi kopi, rokok dan minuman beralkohol

Tabel.3 berisikan basis pengetahuan gejala yang terdapat pada penyakit lambung.

Pada sistem pakar ini terdapat 36 gejala yang terdapat pada penyakit lambung.

Tabel 3. Basis pengetahuan gejala

Kode	Nama
G10	Berat badan menurun
G11	<i>Anoreksia</i>
G12	Muntah berwarna kuning dan pahit
G13	Perut Terasa Kembung
G14	Rongga Mulut Terasa Asam
G15	Nyeri Dada dan Tenggorokan
G16	Sendawa Berlebihan
G17	Cegukan Berlebihan
G18	Batuk Kering
G19	Sulit Menelan Makanan
G20	Muntah darah merah/hitam
G21	Nyeri Pada Ulu Ati yang sakit menetap
G22	Mual
G23	Muntah-muntah
G24	Tinja Berwarna Hitam
G25	Nyeri Pada Perut
G26	Nafsu Makan Menurun
G27	Mudah kenyang
G28	Cepat terasa kenyang setelah sedikit makan
G29	Nyeri Setelah Makan
G30	Muntah cairan asam /muntah air
G31	Muntah makanan yang dimakan beberapa jam sebelumnya
G32	Perut terasa panas dan tidak nyaman
G33	Rasa penuh setelah makan
G34	Buang gas yang berlebihan
G35	Rasa kembung pada saluran cerna atas

Pada Tabel 4 berisikan basis lambung dengan masing-masing penyakit pengetahuan penyakit dan gejala pada memiliki gejala tertentu berdasarkan basis penyakit lambung. Terdapat 4 jenis penyakit pengetahuan gejala penyakit lambung.

Tabel 4. Basis pengetahuan gejala penyakit

Kode Penyakit	Nama Penyakit	Aturan
P1	GERD	G1, G2, G5, G7, G10, G11, G12, G13, G14, G15, G16, G17, G18, G19, G22, G23, G27, G28, G30, dan G35
P2	Tukak Lambung	G1, G2, G3, G4, G6, G10, G11, G12, G13, G20, G21, G22, G24, G25, G26, G29, G30, G31, dan G32
P3	Gastroparesis	G1, G3, G5, G7, G8, G10, G12, G13, G16, G20, G21, G22, G23, G26, G27, G30, G33, dan G35

Tabel 5. Nilai belief masing-masing gejala

Kode Gejala	P1	P2	P3	P4	Believe
G1	✓	✓	✓		0,4
G2	✓	✓			0,6
G3		✓	✓	✓	0,4
G4		✓			0,8
G5	✓		✓		0,6
G6		✓			0,8
G7	✓		✓		0,6
G8			✓		0,8
G9				✓	0,8
G10	✓	✓	✓	✓	0,1
G11	✓	✓			0,6
G12	✓	✓	✓	✓	0,1
G13	✓	✓	✓	✓	0,1
G14	✓				0,8
G15	✓				0,7
G16	✓		✓	✓	0,4
G17	✓				0,7
G18	✓				0,8
G19	✓			✓	0,6
G20		✓	✓		0,7
G21		✓	✓	✓	0,4
G22	✓	✓	✓	✓	0,1
G23	✓		✓	✓	0,3
G24		✓			0,8
G25		✓			0,8
G26		✓	✓		0,6
G27	✓		✓	✓	0,4
G28	✓				0,9
G29		✓		✓	0,5
G30	✓	✓	✓	✓	0,2
G31		✓			0,8
G32		✓		✓	0,6
G33			✓	✓	0,7
G34				✓	0,7
G35	✓		✓	✓	0,4

Pada Tabel.5 merupakan basis pengetahuan nilai pada masing – masing gejala. Setelah dilakukan pengumpulan data

berdasarkan sumber pengetahuan yang didapat, dibuatlah Tabel daftar penyakit, Tabel *factor* resiko, Tabel gejala, dan juga

Tabel gejala penyakit dapat dilihat pada Tabel 1, Tabel 2, Tabel 3, dan Tabel 4.

Analisis Metode Dempster Shafer

Pada analisis metode ini dilakukan pengujian konsultasi dengan menggunakan 3 gejala yang diajukan oleh user yaitu:

1. Perut tidak nyaman pasca makan berlemak (G5)
2. Riwayat konsumsi obat antidepresi (G6)
3. Riwayat pengobatan kanker seperti kemo-terapi atau radioterapi (G8)

Berikut ini adalah contoh simulasi diagnose. Gejala pertama perut tidak nyaman pasca makan berlemak (G5) memiliki nilai believe 0,6. Gejala tersebut masuk kedalam penyakit P1,P3 sehingga $m_1 (P1,P3) = 0,6$ dan $m_1 (\theta) = 1 - 0,6 = 0,4$. Gejala kedua adalah riwayat konsumsi obat antidepresi (G6) memiliki nilai believe 0,8 yang masuk kedalam penyakit P2 sehingga $m_2 (P2) = 0,8$ dan $m_2 (\theta) = 1 - 0,8 = 0,2$. Pada Tabel 6 diberikan aturan kombinasi m_3 .

Tabel 6. Aturan kombinasi mass function 3 (m_3)

	$m_2 (P2) 0,8$	$m_2 (\theta) 0,2$
$m_1(P1,P3) 0,6$	(K) 0,48	(P1,P3)0,12
$m_1 (\theta) 0,4$	(P2) 0,32	(θ) 0,08

Berdasarkan Tabel 6 maka diperoleh nilai $m_3 (P1,P3) = 0,23077$, $m_3 (P2) = 0,61538$ dan $m_3 (\theta) = 0,15385$. Gejala ketiga adalah riwayat pengobatan kanker seperti

kemoterapi atau radioterapi (G8) memiliki nilai believe 0,8 , gejala tersebut masuk kedalam penyakit P3 sehingga diperoleh $m_4 (P3) = 0,8$ dan $m_4 (\theta) = 0,2$.

Tabel 7. Aturan kombinasi mass function 5 (m_5)

	$m_4 (P3) 0,8$	$m_4 (\theta) 0,2$
$m_3 (P1,P3) 0,23077$	(P3) 0,184616	(P1,P3) 0,046154
$m_3 (P2) 0,61538$	(K) 0,492304	(P2) 0,123076
$m_3 (\theta) 0,15385$	(P3) 0,12308	(θ) 0,03077

Pada Tabel 7 diberikan aturan kombinasi m_5 . Berdasarkan Tabel 7 diperoleh $m_5(P3)= 0,60606$, $m_5(P1,P3)= 0,090909$, $m_5 (P2) = 0,24242$ dan $m_5 (\theta) = 0,060611$. Nilai

keyakinan terbesar terdapat pada (P3) yaitu Gastroparesis dengan nilai $0,60606 \times 100\% = 60,6\%$.

Tampilan Interface

Tampilan pada sistem pakar *diagnose* penyakit lambung berbasis web dapat dilihat pada penjelasan berikut.

1. Halaman *Home*

Pada Gambar 6 ditunjukkan halaman Home.

Pada halaman home terdapat menu yang digunakan untuk mengakses, seperti menu konsultasi, info, dan panduan.

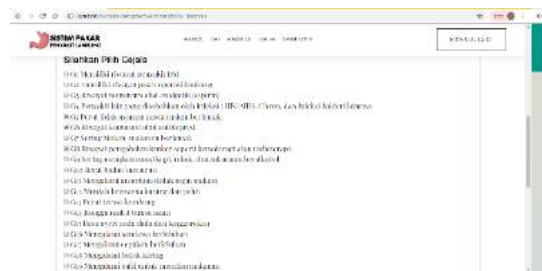


Gambar 6. Halaman *home*

2. Halaman Konsultasi

Gambar 7 merupakan tampilan halaman

konsultasi *user*. *User* memasukkan data diri dan memilih gejala yang dikeluhkan.

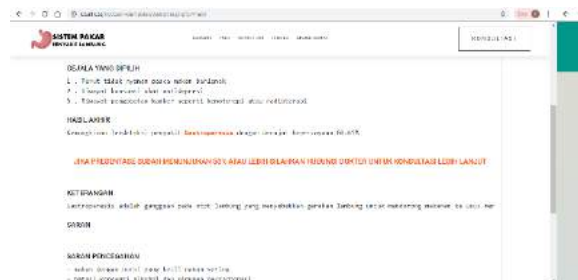


Gambar 7. Halaman konsultasi

3. Halaman Hasil

Pada Gambar 8 diberikan tampilan halaman

hasil. Pada halaman hasil ditampilkan hasil dari gejala yang telah *user* pilih.



Gambar 8. Halaman hasil

Pengujian Akurasi

Pengujian akurasi digunakan untuk membandingkan *diagnose* pakar dengan *diagnose* sistem yang telah dibuat pengujian

menggunakan 100 data uji dan diambil 20 sample sebagaimana yang ditampilkan pada Tabel.6. Pada Tabel 8 berikut diberikan sebagian hasil pengujian sistem pakar.

Tabel 8. Sebagian hasil pengujian akurasi

No	Gejala yang dipilih	Diagnosa		Hasil
		Sistem	Pakar	
1	G1, G5, G7, G8	Gastroparesis (80%)	Gastroparesis	S
2	G1, G2, G3, G4	Tukak Lambung (84,4%)	Tukak Lambung	S
3	G3, G7, G8, G12	Gastroparesis (84,8%)	Gastroparesis	S
4	G8,G10,G11,G12	Gastroparesis (61,5%)	Gastroparesis	S
5	G5, G12, G14, G21	GERD (70,59%)	GERD	S
6	G19,G20, G21, G22	Tukak lambung (48,28%)	Tukak Lambung	S
7	G21,G27, G28, G29	GERD (72,97%)	GERD	S
8	G24,G29, G30, G31	Tukak Lambung (96%)	Tukak Lambung	S
9	G24,G26, G27, G28	GERD (51,43%)	Tukak Lambung	TS
10	G32,G33, G34, G35	Dispepsia (84,76%)	Dispepsia	S
11	G28,G29, G30, G31	GERD (43,37%)	GERD	S
12	G6, G9, G10, G11	Tukak Lambung (60,61%)	Tukak Lambung	S
13	G5, G7, G10, G13	GERD (84%)	GERD	S
14	G10,G11, G13, G15	GERD (70%)	GERD	S
15	G2, G7, G18, G25	GERD (66,06%)	GERD	S
16	G5, G6, G8	Gastroparesis (60,61%)	Gastroparesis	S
17	G6, G17, G18, G26	GERD (55,62%)	GERD	S
18	G21,G27, G32, G33	Dispepsia (49,2%)	Dispepsia	S
19	G7, G15, G16, G17	GERD (91%)	GERD	S
20	G21,G27, G28, G29	GERD	GERD	S

(72,97%)

Ket : S adalah Sesuai, TS adalah Tidak sesuai

Berdasarkan 20 data sample pengujian akurasi perbandingan antara hasil diagnose sistem dengan hasil diagnose pakar, maka didapatkan 19 data sesuai dan 1 data tidak sesuai.

$$\frac{\text{Banyak data sample yang Sesuai}}{\text{Banyak data uji}} \times 100\% \quad (4)$$
$$\frac{19}{20} \times 100\% = 95\%$$

Sedangkan hasil dari 100 data pengujian akurasi perbandingan antara hasil diagnose sistem dengan hasil diagnose pakar dengan wawancara kepada dokter Renzana Rizkika maka didapatkan 94 data sesuai dengan *diagnose* pakar, dengan presentase 94%, dan 6 data yang tidak sesuai dengan presentase 6%.

$$\frac{\text{Banyak data Sesuai / TidakSesuai}}{\text{Banyak data uji}} \times 100\% \quad (5)$$

$$\frac{94}{100} \times 100\% = 94\% \text{ (data sesuai)}$$

$$\frac{6}{100} \times 100\% = 6\% \text{ (data tidak sesuai)}$$

Dari perhitungan 100 data uji perbandingan dan 20 data uji perbandingan, didapatkan penurunan presentase sebesar 1%. Adanya ketidaksesuaian antara diagnosa dokter terhadap diagnosa sistem. Hal ini diakibatkan karena kurang spesifiknya gejala yang di

masuk, karena sistem mendiagnosa penyakit berdasarkan besaran bobot dari tiap gejala, sedangkan dokter mendiagnosa penyakit berdasarkan banyaknya gejala yang diindikasikan pada penyakit tersebut. Sehingga dapat dikatakan sistem pakar diagnosa awal penyakit lambung memiliki tingkat akurasi 94%.

$$\frac{\text{Banyak Penyakit yang muncul}}{\text{Banyak data uji}} \times 100\% \quad (6)$$
$$\frac{41}{100} \times 100\% = 41\% \text{ (GERD)}$$

Dari 100 data uji perbandingan, sistem merekomendasikan adanya indikasi penyakit GERD dengan presentase 41%, selanjutnya penyakit tukak lambung 30% , gastroparesis 13% , dan dispepsia 16%.

KESIMPULAN DAN SARAN

Dari hasil penelitian tentang sistem pakar *diagnose* awal penyakit lambung dengan metode *Dempster Shafer* dapat disimpulkan. Besaran nilai belief pada gejala dapat mempengaruhi hasil dari diagnosa. Pengujian akurasi pada sistem dengan melakukan wawancara kepada dokter penyakit dalam menghasilkan tingkat akurasi 94%. Adanya sistem pakar ini mengedukasi masyarakat terhadap penyakit *GERD*, *Gastroparesis*, *Dispepsia*, dan *Tukak*

Lambung. Perbedaan *diagnose* sistem dengan pakar disebabkan adanya gejala penyakit yang dimasukan kurang spesifik dan sistem mendiagnosa berdasarkan nilai belief gejala, sedangkan. Dokter dengan melihat gejala penyakit terbanyak.

DAFTAR PUSTAKA

- [1] Kirman, A. Saputra, dan J. Sukmana. "Sistem pakar untuk mendiagnosis penyakit lambung dan penanganannya menggunakan metode Dempster Shafer," *Jurnal Pseudocode*, vol.6, no.1, hal. 58 – 66, 2019.
- [2] L. Khairiah, Tursina, dan T. Rismawan, "Sistem pakar diagnosis penyakit hati dengan metode Dempster Shafer berbasis android," *Jurnal Coding Sistem Komputer Untan*, vol. 5, no. 2, hal. 57-66, 2017.
- [3] R. Setiawan, dan C. S. S. Bahri, "Implementasi metode Dempster Shafer pada sistem pakar diagnosa penyakit tropis berbasis web," *Jurnal Coding Sistem Komputer Untan*, vol. 6, no. 3, hal. 97-106, 2018.
- [4] D. Hastari, dan F. Bimantoro. "Sistem pakar untuk mendiagnosis gangguan mental anak menggunakan metode Dempster Shafer," *J-COSINE*, vol. 2, no. 2, hal. 71 – 79, 2018.
- [5] D. W. Utomo, Suprpto, dan N. Hidayat, "Pemodelan sistem pakar diagnosis penyakit pada sistem endokrin manusia dengan metode Dempster-Shafer," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1. No. 9, hal. 893-903. 2017.
- [6] D. M. Khairina, H. R. Hatta, R. Rustam, dan S. Maharani, "Automation diagnosis of skin disease in humans using Dempster-Shafer method," In Proc. The 2nd International Conference on Energy, Environmental and Information System (ICENIS 2017), 2017, hal. 1 – 7.
- [7] D. T. Yuwono, A. Fadlil, dan Sunardi. "Comparative analysis of Dempster-Shafer method and certainty factor method on personality disorders expert systems," *Scientific Journal of Informatics*, vol. 6, no. 1, hal. 12 – 22, 2019.
- [8] A. Sagdoldanova, L. Atymtayeva dan Z. Yespolayeva, "Medicine recommendation technique by using Dempster-Shafer theory," *Adv. Eng. Tec. Appl.*, vol. 6, no. 3, hal. 27-32, 2017.
- [9] A. Susanto, "Pengendalian mutu statistik dengan sistem pakar (pengembangan sistem pakar diagnosis)," Skripsi, Universitas Indonesia, Depok, 1991
- [10] S. Kusumadewi, *Artificial Intelligence Teknik dan Aplikasinya*, Yogyakarta: Graha Ilmu, 2003.

KLASIFIKASI KERUSAKAN BANGUNAN SEKOLAH MENGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* DENGAN *PRE-TRAINED MODEL VGG-16*

¹Ade Muhammad Rizki , ²Nola Marina

^{1,2}Fakultas Teknologi Industri, Universitas Gunadarma
Jl Margonda Raya No. 100, Depok 16424, Jawa Barat
¹ademrizki@gmail.com, ²nolamarina@staff.gunadarma.ac.id

Abstrak

Bangunan sekolah merupakan komponen utama penunjang pelaksanaan proses belajar mengajar dan menjadi salah satu faktor penentu peningkatan mutu suatu lembaga pendidikan. Ketersediaan sarana dan prasarana sebagai penunjang kegiatan juga merupakan hal yang penting dalam peningkatan mutu itu sendiri, sehingga dibutuhkan pemeliharaan dan perawatan yang tepat dalam penggunaan bangunan tersebut. Pada proses penggunaannya, banyak bangunan sekolah yang tidak terawat dikarenakan kurangnya perhatian pada kualitas bangunan tersebut, maupun faktor-faktor yang tidak menentu, seperti kesalahan dalam merancang, cuaca, maupun bencana alam. Salah satu upaya yang dilakukan untuk penanganan bangunan rusak adalah dengan dilakukannya rehabilitasi sebagai penentuan penilaian kerusakan bangunan yang salah satunya dengan mengklasifikasi bangunan secara langsung maupun dengan kumpulan citra. Berdasarkan permasalahan tersebut, dibangun sebuah model Convolutional Neural Network (CNN) untuk mengklasifikasi kerusakan bangunan sekolah di Indonesia. Algoritma CNN yang dibangun menggunakan VGG-16 sebagai pre-trained modelnya. Algoritma CNN digunakan karena memiliki performa yang lebih baik untuk mempelajari data citra dibandingkan dengan metode konvensional lainnya. Model ini dilatih dan diuji menggunakan 3000 citra kerusakan bangunan, diantaranya memiliki 3 kelas kerusakan yang masing-masing terdiri dari 1000 citra per kelasnya. Pengujian model menggunakan 200 citra kerusakan bangunan dari setiap kelas kerusakan. Hasil penelitian menghasilkan nilai akurasi terbaik pada proses pelatihan 3000 citra dengan menghasilkan 67,8%.

Kata Kunci: Citra, Convolutional Neural Network, kerusakan bangunan, VGG-16.

Abstract

School building is a major component that supports the implementation of the teaching and learning process and it is one of the determining factors in improving the quality of an educational institution. The availability of facilities and infrastructure to support activities is also important in improving the quality itself, so that proper maintenance and attention are needed when using the building. In the process of using the school building, many school buildings are not maintained due to lack of attention to the quality of the building, as well as other erratic factors, such as mistakes in design, weather, or natural disasters. One of the efforts made for handling damaged buildings is by conducting rehabilitation as a determination of building damage assessment, one of which is by classifying buildings directly or by a collection of images. Based on these problems, a Convolutional Neural Network (CNN) model was built to classify damage to school buildings in Indonesia. CNN algorithm that was built using VGG-16 as a pre-trained model. CNN algorithm is used because it has better performance to learning image data compared to other conventional methods. This model was trained and tested using 3000 images of building damage, including 3 damage classes, each consisting of 1000 images per class. Testing model uses 200 building damage images from each class of damage. The results of the study produced the best accuracy value in the 3000 image training process with a value of 67.8%.

Keywords: Building Damage, Convolutional Neural Network, image, VGG-16.

PENDAHULUAN

Bangunan sekolah merupakan prasarana yang penting untuk menunjang mutu pendidikan di Indonesia. Namun, kondisi bangunan sekolah masih banyak yang mengalami kerusakan. Hal ini dikarenakan kurangnya perhatian pada kualitas bangunan tersebut maupun faktor-faktor yang tidak menentu lainnya seperti kesalahan dalam merancang, cuaca, maupun bencana alam.

Salah satu upaya yang dilakukan untuk penanganan bangunan rusak adalah dengan dilakukannya rehabilitasi. Rehabilitasi adalah memperbaiki bangunan yang telah rusak sebagian agar dapat digunakan sesuai dengan fungsi utama yang sudah ditetapkan, baik arsitektur maupun struktur bangunan gedung tetap dipertahankan seperti semula, sedangkan utilitasnya dapat berubah. Pada saat melakukan rehabilitasi, beberapa cara digunakan untuk menentukan penilaian kerusakan bangunan yang salah satunya adalah mengklasifikasi bangunan secara langsung maupun dengan kumpulan citra yang sudah ditentukan.

Berdasarkan permasalahan tersebut, pada penelitian ini dibuat model untuk mengklasifikasi kerusakan bangunan sekolah di Indonesia menggunakan metode *Convolutional Neural Network (CNN)* dengan modifikasi arsitektur VGG-16 sebagai salah satu jenis model dari CNN. Proses ini diharapkan dapat membantu dalam proses penilaian kerusakan bangunan untuk memulai perbaikan dan mempermudah proses pengajuan

dalam perawatan maupun perbaikan yang dapat dilakukan pada bangunan tersebut.

Terdapat beberapa penelitian terdahulu mengenai deteksi kerusakan bangunan dan beberapa metode yang digunakan. Penelitian pertama terkait dengan penilaian kondisi bangunan sekolah SDN menggunakan analisis pembobotan berdasarkan hasil survey langsung [1]. Selanjutnya, penelitian lain mengenai deteksi kecacatan pada bangunan juga dilakukan dengan pendekatan *Deep Learning* dengan metode CNN [2]. Berikutnya adalah penelitian mengenai klasifikasi citra menggunakan CNN pada Caltech 101 yang melakukan klasifikasi citra unggas [3].

Konstruksi bangunan sekolah merupakan sarana dan prasana yang harus menyesuaikan dengan standar pemerintah dalam hal kenyamanan dan kekuatan yang telah ditetapkan oleh Menteri Pendidikan Nasional melalui peraturan Nomor 24 Tahun 2007 [4]. Terdapat bangunan sekolah yang mengalami kerusakan diakibatkan umur bangunan, kesalahan manusia, maupun faktor alam. Berdasarkan banyak faktor kerusakan bangunan sekolah, faktor alam paling banyak mempengaruhi karena interaksi terhadap bangunan sangatlah sering seperti terkena angin, cuaca, hujan, dan beragam bencana alam.

Pada penelitian ini akan dilakukan klasifikasi kondisi kerusakan bangunan sekolah menggunakan CNN. Klasifikasi kerusakan gedung atau ruang kelas sekolah yang didefinisikan dalam penelitian ini dikelompokkan menjadi 3 bagian, yaitu rusak

ringan, rusak sedang, dan rusak berat. Rusak ringan didefinisikan berdasarkan visual ringan dari bangunan tersebut, seperti kerusakan warna cat bangunan. Rusak sedang merupakan kerusakan yang tidak mengganggu kerangka dasar bangunan tersebut, seperti kerusakan plafon dan keramik yang terlepas. Rusak berat merupakan kerusakan didefinisikan sebagai bangunan yang sebagian besar kerusakannya tidak layak digunakan dalam kegiatan belajar mengajar, seperti robohnya bangunan gendung atau ruang kelas akibat bencana alam.

METODE PENELITIAN

Pada penelitian ini dibuat model untuk mengklasifikasi kerusakan bangunan sekolah di Indonesia menggunakan metode *Convolutional Neural Network* (CNN) dengan modifikasi arsitektur VGG-16. CNN adalah salah satu metode *deep learning*. Arsitektur CNN yang digunakan adalah arsitektur VGG-16 yang dimodifikasi dan digunakan sebagai *transfer learning*. *Deep learning* adalah bagian disiplin ilmu dari *machine learning* untuk mengidentifikasi objek dalam sebuah citra, dilakukan transkrip dari ucapan menjadi teks, mencocokkan berita, menentukan post atau produk dengan minat pengguna, dan memilih hasil pencarian yang relevan [5]. *Transfer learning* merupakan bidang dalam *machine learning* yang memfokuskan pada pemanfaatan model yang sudah dilatih terhadap suatu *dataset* atau bisa juga disebut dengan istilah “*Pretrained Network*” [6].

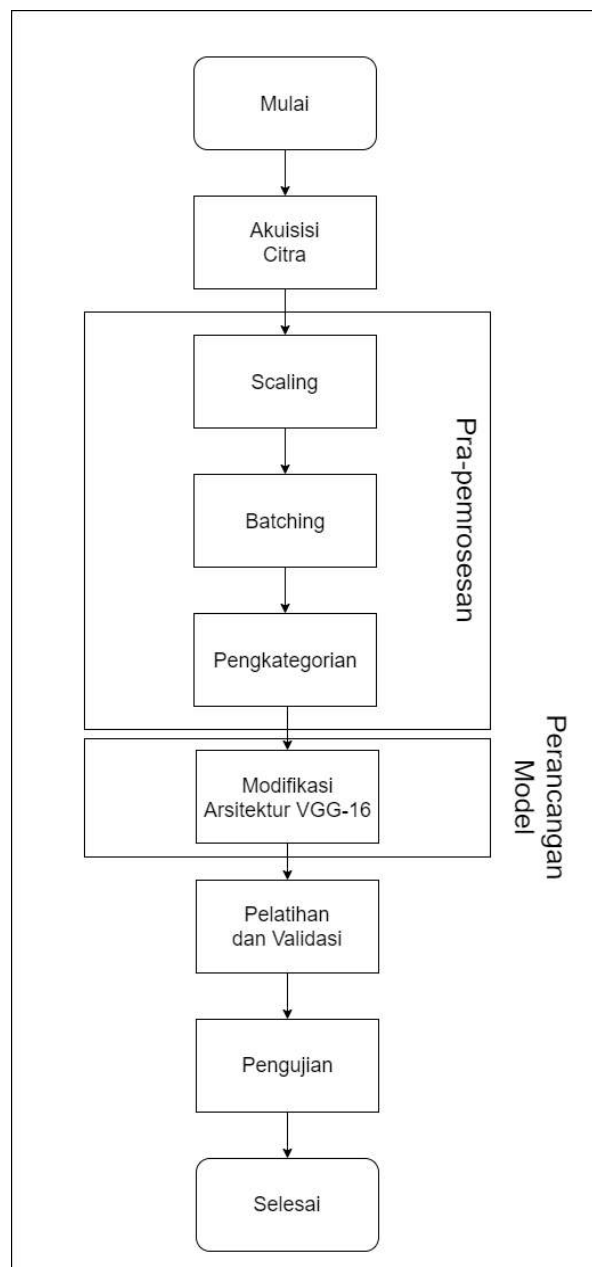
Convolutional Neural Networks atau CNN merupakan jaringan syaraf tiruan khusus untuk memproses data berbentuk matriks. Salah satu contoh berupa citra berwarna yang terdiri atas tiga kanal warna berbentuk matriks dua dimensi. Nama *Convolutional Neural Networks* diambil dari operasi matematika yang digunakan pada algoritma ini, yaitu konvolusi [5]. Arsitektur VGG-16 merupakan model CNN yang dikemukakan oleh K. Simonyan dan A. Zisserman dari Universitas Oxford dalam penulisan yang berjudul “*Very Deep Convolutional Networks for large-scale Image Recognition*”. Model tersebut berhasil mencapai 92,7% dan merupakan 5 besar akurasi tes pada dataset ImageNet, yang mana terdiri dari 14 juta gambar yang berasal dari 10000 kelas yang berbeda. [7]

Metode penelitian dijelaskan pada Gambar 1. Tahap metode penelitian ini dimulai dengan akuisisi citra, pra-pemrosesan, perancangan model, pelatihan dan validasi, serta pengujian. Pra-pemrosesan pada *dataset* dilakukan sebelum pembuatan model. Proses pra-pemrosesan meliputi beberapa tahapan yang dikerjakan secara berurutan. Tahapan tersebut meliputi akuisisi citra yang terdiri dari 3 jenis kerusakan bangunan dan kelas sekolah di seluruh Indonesia yang bersumber dari *Google Image*. Total gambar berjumlah 3000 dengan pembagian untuk data training sebesar 80% dan data testing sebesar 20%. Selanjutnya pada tahap pra-pemrosesan dilakukan proses

seperti *scaling*, *one-hot encoding*, dan *batching*.

Kemudian pada tahap perancangan model arsitektur yang digunakan untuk mengklasifikasi kerusakan bangunan dan kelas disekolah berdasarkan citra yang sudah di kumpulkan dan diolah sebelumnya. Tahap

terakhir pada pelatihan model arsitektur yang dibentuk menyesuaikan citra yang telah diolah sebelumnya. Model pada tahap ini akan mempelajari data untuk menentukan fitur yang optimal dalam mengklasifikasi kerusakan dan memvalidasi tiap operasi untuk menentukan akurasi dari model yang dibuat.



Gambar 1. Metode Penelitian

Akuisisi Citra

Citra bangunan dan kelas rusak dari sekolah di seluruh Indonesia yang digunakan untuk pelatihan dan pengujian pada model ini didapatkan dari *Google Image* dengan cara mengunduh citra dalam jumlah yang sudah ditentukan dalam kurun waktu tertentu untuk tiap jenis kerusakan bangunan. Terdapat 3 jenis kerusakan bangunan sekolah yang ditentukan untuk pengklasifikasian ini yaitu kerusakan bangunan dan kelas di sekolah berintensitas ringan, kerusakan bangunan dan kelas di sekolah berintensitas sedang, dan kerusakan bangunan dan kelas di sekolah berintensitas berat sesuai dengan ketentuan yang sudah didefinisikan sebelumnya. Setiap jenis kerusakan memiliki jumlah citra sebanyak 1000 citra, sehingga jika di total terdapat 3000 citra yang digunakan untuk model yang akan dibuat.

Pra-pemrosesan Citra

Pada tahap pra-pemrosesan citra dilakukan beberapa prosedur yang bertujuan untuk mempersiapkan data agar data siap di

dan *batching*. *Data generator* akan menghasilkan beberapa hasil pra-pemrosesan yang akan digunakan pada model pre-training VGG-16 dan lapisan *fully-connected*.

Perancangan Model

Model yang digunakan pada perancangan ini menggunakan Convolutional Neural Network (CNN) dengan arsitektur VGG-16 sebagai model pre-trained network-nya. Model CNN dan arsitektur VGG-16 digunakan untuk mengklasifikasi citra kerusakan

bangunan sekolah yang akan diketahui jenis kerusakannya.

Arsitektur VGG-16 digunakan sebagai metode transfer learning dimana model tersebut sudah dilatih terhadap suatu dataset untuk menyelesaikan masalah yang serupa dengan cara digunakan sebagai titik permulaan dengan memodifikasi atau mengubah parameternya sehingga dapat sesuai dengan dataset baru.

Modifikasi Arsitektur VGG-16

Penentuan ukuran citra sebagai masukan sebesar 256 x 256 dengan *channel* sebesar 3 yang terdiri dari *Red*, *Green*, dan *Blue* (karena bentuk citra yang berwarna). Citra yang melewati tiap lapisan ekstraksi fitur ukurannya akan direduksi 36 dan filter yang berjumlah 64 bertambah sebesar kelipatan 2 pada tiap blok mulai dari lapisan konvolusi pertama, hingga menjadi berukuran 8x8x512 pada blok terakhir VGG-16 dengan ukuran filter konvolusi tetap sebesar 3x3 dan *stride* sebesar 2. Pada blok terakhir VGG-16 dengan ukuran filter konvolusi tetap sebesar 3x3 dan *stride* sebesar 2, hingga pada lapisan konvolusi keempat pada blok tersebut. Lapisan *max-pooling* di akhir blok juga tetap berukuran 2x2 dengan *stride* sebesar yang menggunakan aktivasi ReLU.

Hasil dari lapisan ekstraksi fitur yang telah dilakukan pada model VGG-16 yang sudah dimodifikasi, kemudian melewati lapisan *Global Average Pooling* yang juga digunakan sebagai lapisan masukan pada bagian *fully-connected* dengan menghitung rata-rata nilai dari semua nilai pada dimensi masukan (panjang x lebar matriks) pada tiap

channel masukan tiap citra dan mempertahankan kedalaman atau *channel* dari citra tersebut. Keluaran dari proses *Global Average Pooling* (GAP) akan menjadi tensor sebesar $1 \times 1 \times$ (*channel* masukan).

Lapisan *fully-connected* pertama akan menerima hasil keluaran dari lapisan sebelumnya dengan penambahan bias dan melewati tensor berukuran 1024 diikuti penggunaan aktivasi ReLu. Setelah melewati lapisan *fully-connected* pertama, neuron yang tidak memiliki bobot yang cukup untuk diaktivasi akan di non-aktifkan secara acak sebanyak 25% dari total jumlah neuron yang digunakan melalui lapisan *dropout*. Hasil dari Lapisan *fully-connected* dan *dropout* sebelumnya akan melalui lapisan *fully-connected* kedua dengan jumlah tensor yang direduksi menjadi sebesar 512 diikuti penggunaan ReLu sebagai aktivasinya. Neuron yang tidak memiliki bobot yang cukup untuk diaktivasi, akan di non-aktifkan kembali secara acak sebanyak 15% dari total jumlah neuron yang digunakan dari lapisan *fully-connected* kedua. Lapisan terakhir atau lapisan keluaran akan menerima hasil akhir dari proses ekstraksi fitur hingga lapisan *fully-connected* dengan memiliki 3 neuron sesuai dengan kategori kelas kerusakan bangunan yang berjumlah 3 kelas, yaitu kerusakan bangunan ringan, sedang, dan berat.

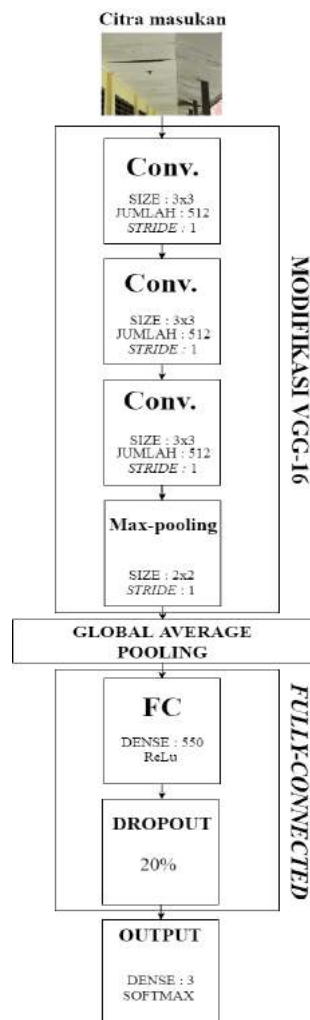
Lapisan keluaran juga menggunakan aktivasi *softmax* untuk menghasilkan probabilitas dari ketiga kelas yang diprediksi. Parameter lain yang digunakan pada model ini yaitu *categorical cross-entropy* yang digunakan untuk meng-

evaluasi penggunaan model pada data yang digunakan. Jika deviasi dari prediksinya terlalu tinggi, maka nilai *loss* akan membesar. Maka dari itu, *dropout* digunakan sebagai regularisasi untuk mengurangi nilai *loss* apabila terus membesar. *Categorical cross-entropy* berfungsi dalam permasalahan klasifikasi yang hanya menghasilkan 1 hasil yang benar. Parameter lain yang digunakan adalah pengoptimalan *Adam* sebagai pengoptimal bobot model saat proses pelatihan dengan mengurangi nilai jarak dari hasil *categorical cross-entropy*.

Bentuk keseluruhan model yang dibuat berdasarkan dari penggunaan blok terakhir dari arsitektur VGG-16 yang dimodifikasi hingga lapisan *fully-connected* untuk diklasifikasi pada lapisan keluaran terdapat pada Gambar 2.

Pelatihan dan Validasi Model

Model yang telah dibangun berdasarkan dataset kerusakan bangunan sekolah yang digunakan sebelumnya, akan dilatih untuk mempelajari fitur yang didapat dari masing-masing kategori kerusakan bangunan menggunakan citra yang telah diakuisisi. Seluruh citra yang telah diakuisisi berjumlah 3000, dibagi peruntukannya sebagai data latih dan data validasi. Data latih menggunakan 80% dari keseluruhan citra yang diakuisisi, sehingga jumlahnya sebesar 2400. Sedangkan, data validasi menggunakan 20% dari seluruh total citra yang telah diakuisisi sebelumnya, sehingga jumlahnya sebesar 600 gambar.



Gambar 2. Model CNN Yang Dirancang

HASIL DAN PEMBAHASAN

Pada tahap ini dibahas mengenai hasil pelatihan dan pengujian yang dilakukan dan keluaran model yang telah dibuat. Setelah melalui proses pelatihan, program akan menampilkan keluaran berupa nilai akurasi dari pelatihan dan validasi pada tiap iterasi. Program juga akan menampilkan hasil *loss* dari proses pelatihan dan validasi pada tiap iterasi. Keluaran dari proses pelatihan pada tiap iterasi dijelaskan pada gambar 3.

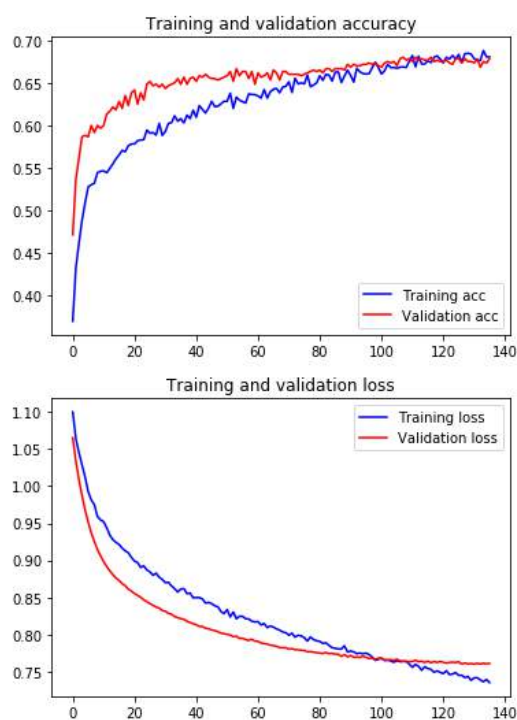
Pada Gambar 3 dapat dilihat bahwa tingkat pembelajaran pada model yang dibangun bernilai 73% untuk nilai *loss*, dan 76% untuk nilai validasi *loss* pada iterasi ke-136. Nilai akurasi didapat sebesar 68% dan 67% pada akurasi validasi pada iterasi ke-136. Proses penghitungan akurasi dan *loss* terjadi secara fluktuatif. Gambaran dari hasil penghitungan akurasi dan *loss* pada data pelatihan dan validasi dijelaskan pada gambar 4.

```

Epoch 13/150
2400/2400 [=====] - 0s 106us/step - loss: 0.9340 - acc: 0.5496 - val_loss: 0.8859 - val_acc: 0.6167
Epoch 14/150
2400/2400 [=====] - 0s 111us/step - loss: 0.9280 - acc: 0.5550 - val_loss: 0.8807 - val_acc: 0.6217
Epoch 15/150
2400/2400 [=====] - 0s 107us/step - loss: 0.9242 - acc: 0.5608 - val_loss: 0.8762 - val_acc: 0.6183
Epoch 16/150
2400/2400 [=====] - 0s 108us/step - loss: 0.9214 - acc: 0.5654 - val_loss: 0.8727 - val_acc: 0.6283
Epoch 17/150
2400/2400 [=====] - 0s 108us/step - loss: 0.9165 - acc: 0.5708 - val_loss: 0.8686 - val_acc: 0.6200
Epoch 18/150
2400/2400 [=====] - 0s 107us/step - loss: 0.9128 - acc: 0.5692 - val_loss: 0.8659 - val_acc: 0.6350
Epoch 19/150
2400/2400 [=====] - 0s 110us/step - loss: 0.9102 - acc: 0.5762 - val_loss: 0.8610 - val_acc: 0.6233
Epoch 20/150
2400/2400 [=====] - 0s 112us/step - loss: 0.9039 - acc: 0.5783 - val_loss: 0.8587 - val_acc: 0.6383
Epoch 21/150
2400/2400 [=====] - 0s 105us/step - loss: 0.8990 - acc: 0.5783 - val_loss: 0.8552 - val_acc: 0.6417
Epoch 22/150
2400/2400 [=====] - 0s 105us/step - loss: 0.8970 - acc: 0.5821 - val_loss: 0.8526 - val_acc: 0.6250
Epoch 23/150
2400/2400 [=====] - 0s 112us/step - loss: 0.8904 - acc: 0.5829 - val_loss: 0.8502 - val_acc: 0.6383
Epoch 24/150
2400/2400 [=====] - 0s 104us/step - loss: 0.8926 - acc: 0.5833 - val_loss: 0.8468 - val_acc: 0.6300
Epoch 25/150
2400/2400 [=====] - 0s 106us/step - loss: 0.8871 - acc: 0.5946 - val_loss: 0.8442 - val_acc: 0.6483

```

Gambar 3. Hasil Pelatihan Model



Gambar 4. Grafik Gradien Akurasi dan *Loss*

Pengujian Klasifikasi

Berdasarkan 6 citra uji yang diambil dari tiap kelas, diantaranya 4 citra berhasil diklasifikasi sesuai dengan kelas sebenarnya, sedangkan satu gambar tidak dapat diprediksi

dengan benar. Hasil tersebut sesuai dengan nilai validasi *loss* yang masih bernilai tinggi, yaitu 76%. Tabel 1 menjelaskan hasil dari klasifikasi pada data uji.

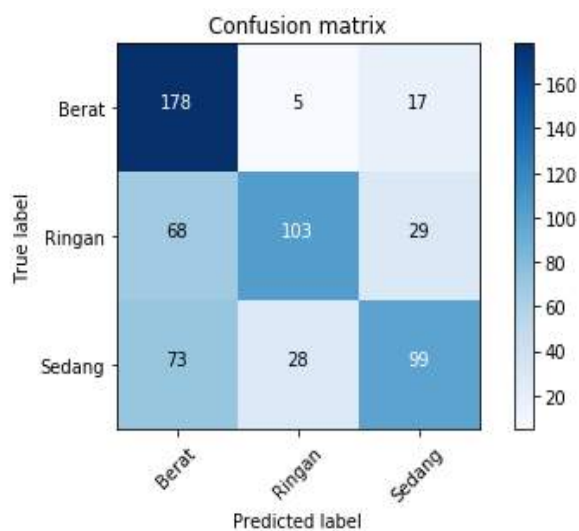
Tabel 1. Hasil Klasifikasi Beberapa Data Uji.

No	Citra Uji	Kelas Sebenarnya	Prediksi Kelas	Kebenaran Prediksi
1.		Rusak Berat	Rusak Berat	True
2		Rusak Berat	Rusak Berat	True
3		Rusak Sedang	Rusak Sedang	True

Analisis Hasil Pengujian

Pengujian ini menggunakan 600 citra yang berasal dari ketiga jenis kerusakan bangunan. Sehingga setiap jenis kerusakan bangunan akan memberikan 200 citra yang digunakan untuk proses pengujian. Data kerusakan bangunan yang digunakan merupakan data yang sudah ditetapkan

porsinya secara manual untuk penggunaan proses pengujian. Dalam pengujian, digunakan *confusion matrix* untuk mendapatkan nilai-nilai yang dibutuhkan seperti nilai akurasi, presisi, *recall*, dan F1. Gambar 5 menjelaskan bentuk plot dari hasil pengujian menggunakan *confusion matrix* pada setiap jenis kerusakan bangunan sekolah.



Gambar 5. *Confusion Matrix* Hasil Pengujian Model

KESIMPULAN DAN SARAN

Pada penelitian ini diperoleh hasil bahwa pada pelatihan terhadap model yang dibangun menghasilkan nilai loss sebesar 68% menunjukkan bahwa model *convolutional neural network* dengan *pre-trained* model VGG-16 dapat diterapkan untuk mengidentifikasi kerusakan bangunan sekolah walaupun cukup sulit dalam mengidentifikasi ciri dari tiap citra. Walaupun cukup sulit dalam mengidentifikasi ciri dari tiap citra karena memiliki bentuk data yang abstrak dan klasifikasinya berdasarkan kondisi citra yang bersifat spasial atau keruangan, sehingga menghasilkan nilai akurasi sebesar 67.8% pada saat pelatihan. Model ini dapat digunakan untuk mengklasifikasi kerusakan bangunan sekolah di Indonesia. Tetapi, nilai akurasi untuk mengukur performa model belum cukup besar 63%.

Pengembangan lebih lanjut dapat dilakukan dengan menggunakan citra yang ukurannya lebih konstan atau lebih besar dari sebelumnya, menggunakan citra dengan pendefinisian klasifikasi yang lebih detail pada pengelompokan tiap kategori, penentuan sudut pandang tertentu dalam proses akuisisi citra, atau menambah jumlah dataset yang digunakan.

DAFTAR PUSTAKA

[1] Parmo, M.H. Sucipto, dan Sumarkan. “Penilaian kondisi bangunan gedung sekolah dasar negeri studi kasus di sekolah dasar negeri se-kabupaten

madiun “, *EMARA Indonesian Journal of Architecture*, vol. 2, no. 1, hal.42 – 52, 2016.

- [2] H.Perez, J. H. M. Tah, dan A.Mosavi. “Deep learning for detecting building defects using convolutional neural networks”, *arXiv:1908.04392v1*, 2019, [Daring] Tersedia <https://arxiv.org/ftp/arxiv/papers/1908/1908.04392.pdf>. [Diakses: 2 Oktober, 2019]
- [3] I.W.Suartika, A.Y.Wijaya, R.Soelaiman. “Klasifikasi Citra Menggunakan Convolutional Neural Networks pada Caltech 101”. *Jurnal Teknik ITS*, vol. 5, no. 1 , hal. 65 – 69, 2016.
- [4] Pemerintah Republik Indonesia, “Peraturan Menteri Pendidikan Nasional Nomor 24 Tahun 2007 tentang Standar Sarana dan Prasarana Untuk SD/MI, SMP/MTs, SMA/ MA”, 2007.
- [5] Y. LeCun, B.Hengio, dan G.Hinton, “Deep learning”, *Nature Research Journal*, vol. 521, hal. 436–444, 2015.
- [6] C. Tan, F.Sun, T.Kong.,W.Zhang, C.Yhang, dan C.Liu, “A survey on deep transfer learning”, 2018 [Daring] Tersedia: <https://arxiv.org/pdf/1808.01974.pdf>. [Diakses: 2 Oktober, 2019].
- [7] K.Simonyan, dan A.Zisserman. “Very deep convolutional networks for large-scale image recognition”. *arXiv preprint arXiv:1409.1556*, Cornell University, 2014. [Daring] Tersedia:
- [8] <https://arxiv.org/pdf/1409.1556v3.pdf>. [Diakses: 2 Oktober, 2019].

KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

¹Nur Fadlia, ²Rifki Kosasih

^{1,2}Fakultas Teknologi Industri Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
¹nfadliaa19@gmail.com, ²rifki_kosasih@staff.gunadarma.ac.id

Abstrak

Kemacetan merupakan suatu keadaan padatnya lalu lintas yang dapat disebabkan oleh banyak faktor. Salah satu faktor yang menyebabkan kemacetan adalah banyaknya kendaraan yang melintas di jalan. Salah satu usaha yang dapat dilakukan untuk mengurangi masalah kemacetan yaitu dengan membuat jalur khusus yang hanya boleh dilalui kendaraan roda dua, empat atau lebih. Akan tetapi, terdapat pengendara yang masih menggunakan jalur yang tidak sesuai seperti di jalan Margonda Raya, Depok. Kendaraan roda dua (motor) yang telah disiapkan jalur khusus, sering kali mengambil jalur untuk kendaraan roda empat sehingga diperlukan suatu kegiatan pengawasan penggunaan jalur lalu lintas. Oleh karena itu, pada penelitian ini dibuat pengenalan jenis-jenis kendaraan menggunakan metode Convolutional Neural Network (CNN). Data yang digunakan dalam penelitian ini sebanyak 120 citra yang terdiri dari citra mobil, motor dan sepeda. Hasil uji coba dan evaluasi model terhadap tiga jenis kendaraan menggunakan package Keras menunjukkan akurasi sebesar 94,4% pada tahap pelatihan dan 73,3% pada tahap pengujian.

Kata Kunci: *convolutional neural network, deep learning, kendaraan, klasifikasi.*

Abstract

Congestion is a state of heavy traffic which can be caused by many factors. One factor that causes congestion is the number of vehicles passing on the road. One effort that can be done to reduce the problem of congestion is to make a special lane that can only be passed by two, four or more wheeled vehicles. However, there are motorists who still use inappropriate paths such as on Margonda Raya Street, Depok. Two-wheeled vehicles (motorcycles) that have been prepared for special lanes, often take the lane for four-wheeled vehicles so that an activity is needed to monitor the use of traffic lanes. Therefore, in this study an introduction to the types of vehicles was made using the Convolutional Neural Network (CNN) method. The data used in this study were 120 images consisting of images of cars, motorbikes and bicycles. The results of the trial and model evaluation of the three types of vehicles using the Keras package show an accuracy of 94.4% at the training stage and 73.3% at the testing stage.

Keywords: *clasification, convolutional neural network, deep learning, vehicles.*

PENDAHULUAN

Kemacetan lalu lintas merupakan suatu keadaan dimana jumlah kendaraan yang melintas lebih banyak sehingga melebihi kapasitas jalanan [1]. Sekarang ini, banyak kendaraan yang tidak berjalan sesuai jalurnya. Salah satu contoh di jalan Margonda Raya, Depok yang telah disiapkan jalur khusus kendaraan roda dua, sering kali ditemukan kendaraan roda dua mengambil jalur untuk kendaraan roda empat. Contoh lain adalah kendaraan roda dua memasuki jalur tol kendaraan roda empat yang tentu saja mengganggu lalu lintas. Salah satu usaha untuk mengurangi masalah kemacetan yaitu dengan menetapkan jalur-jalur khusus yang hanya boleh dilalui kendaraan roda dua, empat atau lebih. Teknologi seperti *artificial intelligence* dapat digunakan untuk dapat mengenali kendaraan-kendaraan yang melewati jalan sehingga kegiatan pengawasan lalu lintas dapat diserahkan pada aplikasi berbasis komputer.

Pengenalan jenis-jenis kendaraan dapat digunakan teknik *deep learning*. Salah satu teknik *deep learning* yang banyak digunakan adalah *Convolutional Neural Network* (CNN) yang mampu mengenali jenis-jenis kendaraan. *Deep learning* merupakan algoritma jaringan saraf tiruan yang menggunakan data sebagai masukan dan memprosesnya dengan menggunakan sejumlah lapisan tersembunyi. Setelah itu melakukan transformasi nonlinier dari data masukan untuk menghitung nilai keluaran [2].

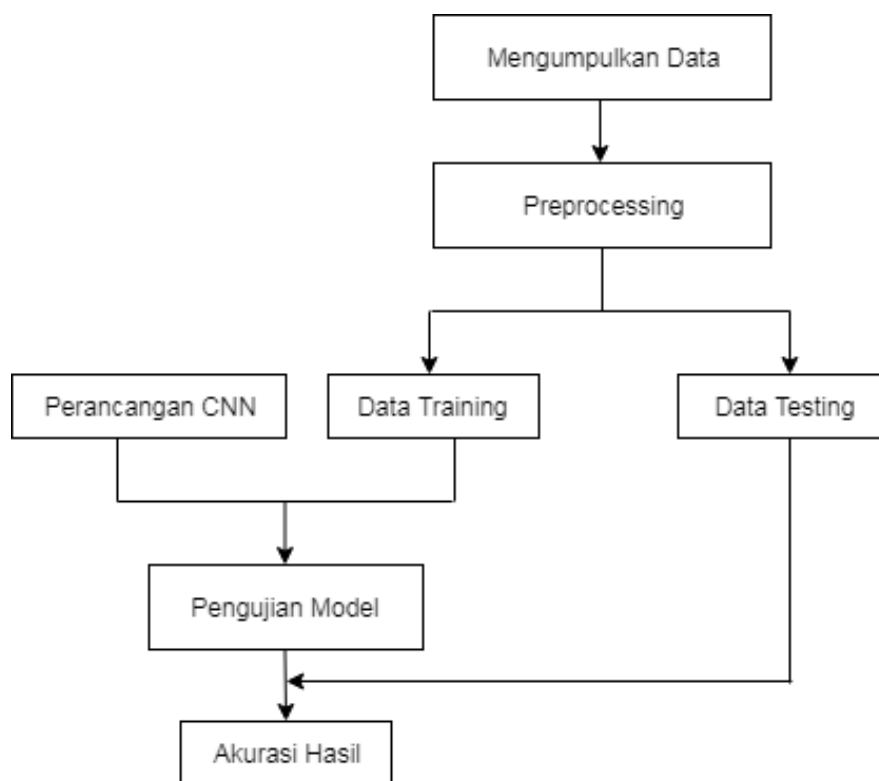
Beberapa penelitian mengenai pengolahan citra dengan menggunakan metode CNN mendapatkan hasil akurasi yang bagus, yaitu penelitian yang dilakukan oleh Rismiyati untuk sortasi salak ekspor berbasis citra digital. Hasil akurasi yang didapatkan yaitu dengan satu lapisan konvolusi adalah 81,5% dan didapatkan nilai akurasi sebesar 70,7% dengan dua lapisan konvolusi [3]. Penelitian menggunakan CNN juga dilakukan oleh Arrofiqoh dan Harintaka untuk klasifikasi tanaman pada citra resolusi tinggi. Hasil pengujian menghasilkan akurasi sebesar 100% terhadap data latih. Pengujian terhadap data validasi menghasilkan akurasi sebesar 93% dan akurasi terhadap data uji sebesar 82% [4]. Penggunaan CNN dengan 7 lapisan untuk pengenalan wajah menghasilkan tingkat akurasi mencapai 98,57% pada penelitian yang dilakukan oleh Santoso dan Ariyanto [5]. Pada penelitian yang dilakukan oleh Rokhana, Priambodo, Karlita, Sunarya, Yuniarno, Purnama, dan Purnomo menggunakan lima rancangan arsitektur CNN dalam pengenalan kontur tulang dan klasifikasi tulang yang patah. Hasil uji coba pada penelitian tersebut memiliki akurasi 95,3%, sensitivitas 95%, dan specificity 96% [6]. Pada penelitian mengenai penggunaan CNN untuk mengklasifikasi citra makanan tradisional telah dilakukan oleh Rohim, Sari, dan Tibyani. Arsitektur model CNN yang dibuat pada penelitian tersebut dapat mengklasifikasi citra makanan tradisional dengan nilai 73% presisi, 69% *recall* dan 69% *Fscore* [7].

Pada penelitian ini digunakan metode

Convolutional Neural Network untuk mengenali jenis kendaraan darat. Jenis kendaraan darat yang dapat dikenali pada penelitian ini yaitu mobil, motor dan sepeda.

METODE PENELITIAN

Pada penelitian ini pengklasifikasian kendaraan dilakukan dengan menggunakan metode *Convolutional Neural Network*. Proses perancangan dan implemetasi terdiri atas beberapa tahap. Tahapan penelitian disajikan pada Gambar 1.



Gambar 1. Tahapan Pengklasifikasian Kendaraan dengan Metode CNN

Pada Gambar 1 pengumpulan data dilakukan untuk mendapatkan data citra yang akan digunakan pada proses pelatihan dan pengujian model arsitektur *Convolutional Neural Network*. Pada tahap *preprocessing* dilakukan untuk mempersiapkan data agar dapat digunakan pada tahap selanjutnya. Setelah data diproses, tahapan berikutnya adalah membangun arsitektur model *Convolutional Neural Network* (CNN). Model ini akan digunakan untuk mengetahui

tingkat akurasi dari transportasi darat. Selanjutnya adalah tahap pelatihan model yang sudah dirancang menggunakan data yang telah diolah.

Pengumpulan Data

Pengumpulan data citra yang dilakukan menggunakan aplikasi tambahan pada Google yaitu Fatkun Batch Downloader. Aplikasi ini mampu mengunduh gambar dalam jumlah

yang banyak. Banyaknya citra yang diambil sebanyak 90 citra sebagai data latih dan 30 citra sebagai data uji. Citra tersebut terdiri dari gambar mobil sebanyak 40, motor sebanyak 40 dan sepeda sebanyak 40.

Preprocessing Data

Preprocessing citra dengan mengubah ukuran citra mobil, motor dan sepeda menjadi ukuran yang sama. Langkah selanjutnya dilakukan pembagian data yang telah dikumpulkan menjadi dua, yaitu data latih (*training*) dan data uji (*testing*). Langkah selanjutnya dilakukan pelabelan pada masing-masing kendaraan untuk pengenalan kendaraan berupa mobil, motor dan sepeda pada data latih dan data uji.

Perancangan CNN

Perancangan CNN merupakan tahapan dalam menyusun sebuah model yang digunakan untuk melatih data dalam mengenali objek yang diinginkan. Model yang disusun terdiri dari jumlah layer yang digunakan, penentuan *filter*, penentuan ukuran *kernel*, dan penentuan fungsi aktivasi dan ukuran *pooling*.

Pelatihan Model

Setelah dilakukan perancangan model CNN, tahapan selanjutnya adalah melakukan pelatihan model dengan menggunakan data latih yang telah dikumpulkan sebelumnya. Selanjutnya, pada proses pengujian model digunakan jumlah *epoch* (iterasi) untuk menentukan berapa kali jaringan akan melakukan pelatihan. Pada tahap ini terdapat

fungsi *loss* yang digunakan untuk melihat performa dari model CNN.

Pengujian Model

Tahapan selanjutnya adalah melakukan pengujian model pada data uji. Tahapan ini dilakukan untuk menguji tingkat akurasi dari model *Convolutional Neural Network* (CNN). Tingkat akurasi ini menunjukkan tingkat kebenaran dari pengklasifikasian jenis kendaraan. Tingkat akurasi yang tinggi menunjukkan bahwa model dapat melakukan pengklasifikasian ketiga jenis kendaraan dengan baik.

HASIL DAN PEMBAHASAN

Hasil Preprocessing Data

Setelah data terkumpul, tahapan selanjutnya yaitu memasukkan data dan membuat sistem untuk pengenalan kendaraan darat. Pengenalan kendaraan dilakukan dengan menggunakan metode *deep learning* yaitu metode *Convolutional Neural Network* (CNN). Data yang digunakan berupa data citra.

Dalam penelitian ini, langkah yang dilakukan sebelum memasukkan data yaitu melakukan instalasi *package* yang nantinya akan digunakan dalam pengolahan data. *Package* yang diinstalasi yaitu *package* Tensorflow, *package* Keras dan *package* EBImage

Pengenalan menggunakan metode CNN yaitu mengklasifikasikan gambar yang dimulai dengan cara melakukan pelatihan. Proses pelatihan dilakukan agar model yang dibuat dapat mengenali objek yang diinginkan.

Pembagian data latih dan data uji yang digunakan pada penelitian ini dapat dilihat pada Tabel 1.

Berdasarkan Tabel 1, data yang digunakan

sebanyak 120 data citra yang terdiri dari citra mobil, motor, dan sepeda. Data latih yang digunakan terdiri dari 30 citra mobil, 30 citra motor dan 30 citra sepeda.

Tabel 1 Pembagian Data Penelitian

Jenis Kendaraan	Jumlah Data Latih	Jumlah Data Uji
Mobil	30	10
Motor	30	10
Sepeda	30	10
Jumlah	90	30

Contoh data latih yang digunakan pada penelitian ini dapat dilihat pada Gambar 2.

Data uji terdiri dari 10 citra motor, 10 citra mobil dan 10 citra sepeda. Contoh data uji yang digunakan dapat dilihat pada Gambar 3.

Selanjutnya dilakukan pelabelan pada data latih dan data uji. Pelabelan dilakukan dengan menggunakan angka 0, 1, dan 2. Angka 0 (nol) untuk label mobil, angka 1 untuk label motor dan angka 2 untuk label sepeda.



Gambar 2. Contoh Data Latih



Gambar 3. Contoh Data Uji

Pembuatan Model

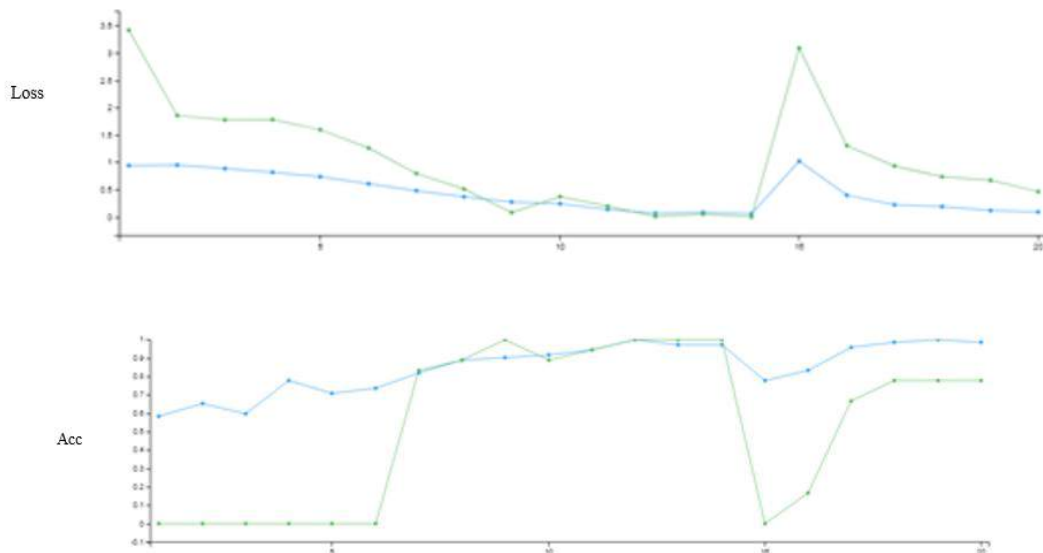
Pada pembuatan model, umumnya proses yang dilakukan terdiri dari proses konvolusi yang disertai fungsi aktivasi dan proses *pooling*. Banyaknya proses ini disesuaikan dengan kebutuhan peneliti. Model dibuat dengan memuat beberapa jenis lapisan yang berbeda, antara lain lapisan konvolusi (*layer_conv_2d*), lapisan *pooling*, lapisan *dropout*, lapisan *flatten* dan lapisan *dense*. Proses konvolusi ini dilakukan sebanyak 4 kali ditunjukkan oleh banyaknya *convolution layer* yang digunakan. Pada umumnya 2 sampai 3 lapisan cukup untuk mendapatkan model klasifikasi yang memiliki akurasi tinggi. Pada penelitian ini digunakan banyak lapisan untuk melatih model dan melihat bagaimana kinerja model tersebut.

Fungsi aktivasi yang digunakan pada model CNN adalah ReLu (*Rectified Linear Unit*) yang menjadikan tahap pelatihan menjadi lebih cepat. Ukuran *kernel* yang digunakan untuk setiap lapisan konvolusi sebesar 3×3. Pada proses *pooling* digunakan metode *maxpooling* dengan ukuran *filter* 2×2 sehingga nilai maksimum pada area 2×2 piksel tersebut yang akan dipilih pada setiap pergeseran.

Jumlah filter yang digunakan dalam *convolution layer* yang pertama dan kedua adalah 32. Filter yang digunakan pada *convolution layer* ketiga dan keempat adalah 64. Penggunaan jumlah filter yang lebih banyak pada dua lapisan konvolusi terakhir disebabkan oleh ukuran *input* pada kedua lapisan tersebut lebih kecil sehingga dibutuhkan lebih banyak filter untuk mengesktrak informasi citra.

Tahapan selanjutnya yaitu melakukan pelatihan data citra kendaraan ke dalam model dengan *fit model*. Pada saat melakukan *fit model* digunakan *epoch* = 20, *batch_size* = 32 dan *validation_split* = 0,2. *Epoch* digunakan untuk menentukan berapa kali jaringan akan melihat

seluruh kumpulan data. Nilai *batch_size* menunjukkan jumlah contoh pelatihan dalam satu *forward/backward pass*. Semakin tinggi nilai *batch_size* maka akan semakin banyak memori yang dibutuhkan. Hasil *loss* dan *accuracy* model yang terbentuk dapat dilihat pada Gambar 4.



Gambar 4. Grafik Hasil Iterasi Nilai *Loss* dan *Accuracy*

Pada Gambar 4 dapat dilihat bahwa pada iterasi ke 30 nilai *loss* yang dihasilkan pada data latih yaitu 0.171811 dan akurasi pada data latih yaitu 0.9444444. Selanjutnya, pengujian model pada data uji memperoleh hasil *loss* yaitu 0.9727243 dengan akurasi sebesar 0.7333333.

Berdasarkan hasil pengujian diperoleh informasi bahwa model yang dibuat mampu mengklasifikasikan jenis kendaraan dengan baik. Pergerakan *loss* yang mendekati nilai nol atau kurang dari satu dan akurasi yang terus meningkat menunjukkan hasil yang baik seiring dengan berjalannya *epoch*. Kondisi berhentinya proses pelatihan ditentukan dari

banyaknya *epoch* yang digunakan, dalam hal ini yaitu sebanyak 20 *epoch*. Proses pembelajaran tersebut akan berhenti jika sudah memenuhi kondisi tersebut.

Uji Coba dan Evaluasi

Hasil klasifikasi yang didapatkan dari data latih dan data uji ditunjukkan pada Gambar 5. Pada Gambar 5 ditunjukkan hasil prediksi untuk data latih. Citra mobil yang berhasil diklasifikasi adalah semua citra mobil yaitu sebanyak 30, citra motor yang berhasil diklasifikasi yaitu 29 dari 30 citra yang digunakan dan citra sepeda yang berhasil diklasifikasi yaitu sebanyak 26 citra.

		Actual		
		0	1	2
Predicted	0	30	1	0
	1	0	29	4
	2	0	0	26

Gambar 5. Hasil Klasifikasi Data Latih

Hasil klasifikasi data uji dapat dilihat pada Gambar 6. Pada Gambar 6 didapatkan hasil prediksi untuk data uji yang terdiri dari 10 citra untuk tiap jenis kendaraan. Citra mobil yang berhasil diklasifikasi yaitu sebanyak 8, citra motor yang berhasil diklasifikasi yaitu sebanyak 8 citra dari 10 citra dan citra sepeda yang berhasil diklasifikasi yaitu sebanyak 6 citra dari 10 citra yang digunakan.

		Actual		
		0	1	2
Predicted	0	8	2	0
	1	2	8	4
	2	0	0	6

Gambar 6. Hasil Klasifikasi Data Uji

KESIMPULAN DAN SARAN

Pada penelitian ini klasifikasi jenis kendaraan dengan metode *Convolutional Neural Network* (CNN) telah berhasil dilakukan dengan tingkat akurasi sebesar 73,33%. Model CNN yang digunakan terdiri dari empat lapisan konvolusi dengan ukuran filter 3×3, fungsi aktivasi yang digunakan yaitu reLu, dan 2 *pooling layer* dengan ukuran 2×2. Citra yang digunakan untuk penelitian ini yaitu sebanyak 120 citra. Berdasarkan hasil pelatihan

diperoleh tingkat akurasi dari model CNN yaitu sebesar 0.9444444 dengan *loss* 0.171811. Pada proses pengujian dihasilkan nilai *loss* yang masih cukup besar yaitu 0.9727243 dengan akurasi sebesar 0.7333333.

Pada penelitian selanjutnya dapat dilakukan untuk meningkatkan akurasi dan menurunkan nilai *loss* dengan melakukan beberapa hal seperti menambahkan data citra, menambahkan lapisan konvolusi dan menambahkan fungsi optimasi. Selain itu, pada penelitian lebih lanjut dapat digunakan metode

deep learning lainnya sehingga diperoleh akurasi klasifikasi yang lebih baik.

DAFTAR PUSTAKA

- [1] R. Azis dan Asrul, *Pengantar sistem dan perencanaan transportasi*. Yogyakarta: CV. Budi Utama, 2018.
- [2] R. Primartha, *Belajar machine learning teori dan praktek*. Bandung: Informatika, 2018.
- [3] Rismiyati, “Implementasi convolutional neural network untuk sortasi mutu salak ekspor berbasis citra digital,” *Tesis*, Universitas Gajah Mada, Yogyakarta, 2016.
- [4] E. N. Arrofiqoh dan Harintaka, “Implementasi metode convolutional neural network untuk klasifikasi tanaman pada citra resolusi tinggi,” *Jurnal Geomatika*, vol. 24, no. 2, hal. 61 – 68, 2018.
- [5] A. Santoso dan G. Ariyanto, “Implementasi deep learning berbasis keras untuk pengenalan wajah,” *Jurnal Emitor*, vol. 18, no. 1, hal. 15 – 21, 2018.
- [6] R. Rokhana, J. Priambodo, T. Karlita, I. M. G. Sunarya, E. M. Yuniarno, I. K. E. Purnama, dan M. H. Purnomo, “Convolutional neural network untuk pendeteksian patah tulang femur pada citra ultrasonik b-mode,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 8, no. 1, hal 59. – 67, 2019.
- [7] A. Rohim, Y. A. Sari, dan Tibyani, “Convolution Neural Network (CNN) untuk pengklasifikasian citra makanan tradisional,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 7, hal. 7037 – 7042, 2019.

RANCANG BANGUN PURWARUPA SISTEM PENGUNCI LEMARI DENGAN PENGENALAN SUARA

¹Laksamana Akbar Dzulfikar, ²Emy Haryatmi, ³Tri Agus Riyadi

¹²Fakultas Ilmu Komputer Universitas Gunadarma, ³Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹akbardzulfikar09@gmail.com, ²emy_h@staff.gunadarma.ac.id,

³ta_riyadi@staff.gunadarma.ac.id

Abstrak

Sistem keamanan berbasis teknologi modern telah banyak diterapkan di kehidupan sehari-hari salah satunya adalah penerapan sistem keamanan pada kunci pintu lemari. Perintah suara merupakan salah satu media pengoperasian sistem home automation yang banyak diminati. Pada penelitian ini dibuat sistem pengunci lemari dengan metode pengenalan suara. Sistem keamanan ini menggunakan suara untuk membuka kunci pintu lemari dan menguncinya. Alat ini dirancang dengan menggunakan modul elechouse voice recognition v3, Arduino Uno R3 sebagai mikrokontroler, motor servo 9g dan lampu LED. LED berfungsi sebagai indikator tambahan pada saat penguncian lemari. Perancangan sistem pengunci lemari ini menggunakan pengenalan suara dengan dua kondisi pergerakan motor servo. Hasil penelitian menunjukkan kondisi pertama jika suara pengguna terdeteksi sebagai suara dengan signature "on" maka motor servo akan bergerak membuka kunci lemari dan LED sebagai indikator tambahan akan menyala. Kondisi kedua yaitu jika suara pengguna terdeteksi sebagai suara dengan signature "off" maka motor servo akan bergerak menutup kunci lemari dan LED sebagai indikator tambahan akan mati. Baud rate yang memiliki tingkat keberhasilan paling tinggi adalah 2400 bps.

Kata Kunci: mikrokontroler, pengenalan suara, sistem keamanan, sistem pengunci.

Abstract

Modern technology-based security systems have been widely applied in daily life, one of which is the application of security systems on cabinet door locks. Voice commands are one of the most popular home automation media operating systems. In this study, a cabinet locking system was made with a speech recognition method. This security system uses sound to unlock cabinet doors and lock them. This tool is designed by using the v3 voice recognition elechouse module, Arduino Uno R3 as a microcontroller, 9g servo motor and LED lights. LED functions as an additional indicator when locking cabinets. The design of this cabinet locking system uses voice recognition with two servo motor movement conditions. The results show the first condition if the user's voice is detected as a sound with a signature "on" then the servo motor will move to unlock the cabinet and the LED as an additional indicator will light up. The second condition is if the user's voice is detected as a sound with an "off" signature, the servo motor will move to close the cabinet lock and the LED as an additional indicator will turn off. Baud rate has the highest success rate is 2400 bps.

Keywords: microcontroller, locking system, security system, voice recognition.

PENDAHULUAN

Teknologi modern pada zaman ini banyak dipergunakan pada berbagai aspek kehidupan salah satunya pada teknologi pengunci. Salah satu penerapan sistem pengunci adalah kunci berbagai pintu, seperti pada pintu lemari hingga kotak deposit. Sistem pembukaan dan penguncian pintu secara elektronik maupun mekanik telah banyak dipergunakan. Sistem penguncian yang telah diterapkan pada lemari antara lain adalah menggunakan akses berupa tombol angka *password* [1], pengenalan biometrik pada sidik jari manusia [2], dan dengan kartu yang memiliki kode unik [3]. Setiap akses yang telah disebutkan memiliki kelebihan dan kekurangan masing-masing. Penguncian dengan tombol angka *password* memiliki kekurangan dapat dimasukan oleh siapapun yang mengetahui angka *password* tersebut. Penguncian dengan *smartphone* dapat digunakan oleh siapapun yang menggunakan *smartphone* tersebut. Pada sistem penguncian dengan pengenalan sidik jari manusia hanya dapat dilakukan oleh seseorang yang sidik jarinya telah terekam oleh sistem, namun jika sidik jari tersebut terkena goresan atau luka dapat membuat sensor sidik jari menjadi susah dalam mengidentifikasi sidik jari.

Penelitian mengenai penggunaan pengenalan suara telah banyak dilakukan diantaranya adalah penelitian yang dilakukan oleh Imario, Sudiharto, dan Ariyanto [4]. Penelitian tersebut menguji validasi suara

berbasis pengenalan suara (*voice recognition*) menggunakan *easy VR 3.0*. Sensor utama yang digunakan adalah sensor pengenal suara, dengan menggunakan modul *easy VR 3.0*. Pengenalan suara digunakan sebagai akses pintu yang menggunakan kunci *solenoid* untuk membuka kunci dan menutup kunci. Suara yang dikenali memiliki keunikan sendiri yaitu jika suara yang direkam adalah suara laki-laki, modul beberapa kali mengenali suara yang terekam walaupun yang mengucapkannya bukan suara asli telah terekam ke dalam modul. Berbeda dengan perempuan, modul lebih dapat membedakan suara untuk pengguna perempuan yang telah terekam dalam modul.

Penelitian lain mengenai penggunaan pengenalan suara dilakukan oleh Seppiawan, Nurussa'adah, dan Siwindarto untuk sistem keamanan pintu pagar otomatis menggunakan *voice recognition* [5]. Pada penelitian tersebut menggunakan modul *Easy VR 3.0* sebagai modul pengenal suara. Suara dijadikan sebagai akses utama untuk pembukaan pintu pagar otomatis. Arduino Uno R3 digunakan sebagai mikrokontroler yang menghubungkan serial modul sebagai *input* (masukan) dan motor DC sebagai *output* (keluaran). Pengujian dilakukan dengan percobaan pengucapan suara dengan jarak-jarak yang berbeda, maka ditemukan jarak yang ideal dalam pengucapan suaranya. Suara yang masuk menggunakan *microphone* dihubungkan dengan modul *wireless* yang selanjutnya suara tersebut diterima oleh modul *Easy VR3* dengan jarak 10 meter. Untuk kecepatan respon pergerakan motor

DC yang digunakan untuk tiap suara didapatkan hasil respon yang sama.

Penelitian selanjutnya telah dilakukan oleh Dani, Adriansyah, dan Hermawan yaitu merancang aplikasi *voice command recognition* berbasis Android dan Arduino Uno [6]. Pada penelitian tersebut digunakan suara sebagai akses utama untuk menghidupkan dan mematikan kipas dan lampu. Namun suara yang terekam adalah suara yang dikenali berdasarkan kata yang diucapkan, bukan dikenali berdasarkan pengguna. Suara direkam melalui *smartphone* dan akan dideteksi oleh modul *Bluetooth* yang terhubung dengan Arduino Uno R3. Suara dapat dideteksi karena Arduino Uno R3 menggunakan Google *voice command* yang mana Google memiliki banyak kata yang telah disimpan sehingga Arduino Uno R3 dapat mengenali kata yang diucapkan dengan suara siapapun yang mengucapkannya.

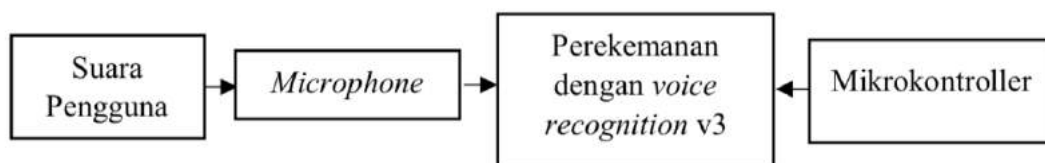
Penelitian mengenai *voice recognition* menggunakan modul *elechouse voice recognition* juga dibuat oleh Nugroho, Setyawan, dan Basuki [7]. Penelitian tersebut menggunakan modul *elechouse voice recognition* untuk alat bantu penderita disabilitas menggunakan mikrokontroler Arduino Uno R3. Pada penelitian tersebut, *microphone* yang digunakan adalah *microphone*

dengan kualitas standar, maka jika pengguna tidak mengucapkan suara dengan jarak dekat maka suara tidak dapat terekam dan tidak dikenali secara baik. Hal ini dimungkinkan karena nilai *baud rate* yang digunakan kurang tepat sehingga suara tidak dapat terekam dan tidak dikenali dengan baik. Berdasarkan hal tersebut, maka pada penelitian ini dirancang sebuah sistem yang dapat membuka dan menutup pintu kunci lemari dengan menggunakan metode pengenalan suara (*voice recognition*) berdasarkan beberapa nilai *baud rate* yang berbeda supaya suara dapat dikenali dengan baik.

METODE PENELITIAN

Perancangan sistem pengunci lemari dengan menggunakan pengenalan suara dibagi atas dua bagian, yaitu sistem perekaman suara dan sistem pengunci lemari. Sistem perekaman suara merupakan perekaman terhadap suara pengguna yang dapat digunakan untuk membuka dan mengunci pintu lemari. Sistem pengunci lemari ini berdasarkan dari sistem perekaman suara. Suara yang dapat digunakan pada sistem pengunci lemari untuk membuka dan mengunci lemari hanya suara yang telah tersimpan dalam sistem.

Perancangan Sistem Perekaman Suara



Gambar 1. Sistem Perekaman Suara

Gambar 1 merupakan perancangan untuk perekaman suara yang membutuhkan *input* (masukan) berupa suara yang dimasukkan pada modul *voice recognition* karena modul tersebut belum menyimpan suara. Suara pengguna yang direkam merupakan sinyal analog yang bersumber dari manusia dan direkam dengan menggunakan *microphone*. Pengucapan suara pengguna harus memenuhi beberapa kriteria seperti pengucapan yang jelas, tinggi rendah nada suara harus sesuai, serta intonasi suara. *Microphone* digunakan untuk mengubah *input* (masukan) berupa sinyal analog menjadi sinyal digital. *Microphone* yang digunakan adalah jenis *microphone* dengan *jack 3.5mm* jenis *stereo 3 pole TRS (Tip Ring Sleeve)*. Suara pengguna yang diucapkan pada *microphone* akan masuk ke modul *elechouse voice recognition V3*. Modul akan mendeteksi suara yang telah diubah dari sinyal analog menjadi sinyal digital pada perekaman suara dengan modul *voice recognition* yang membutuhkan mikrokontroler. Mikrokontroler yang digunakan adalah *Arduino Uno R3* dan dibutuhkan untuk mengaktifkan fungsi modul *voice recognition*.

Proses perekaman suara membutuhkan beberapa kali pengulangan pengucapan suara sehingga modul dapat merekam suara jika suara telah cocok pada percobaan pengulangan suara. Suara yang tersimpan dapat diproses menjadi *output* (keluaran) dengan mikrokontroler *Arduino Uno R3*. Pada proses perekaman suara, *Arduino Uno R3* akan mendefinisikan suara dengan variabel "*signature*" dan

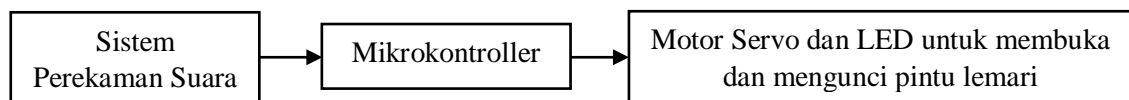
"*recordnum*". Teknik perekaman pada modul dengan melatih (*train*) modul terlebih dahulu. Ketika suara akan disimpan, maka masukan *command: sigtrain 0 on*. *Command* tersebut artinya suara direkam dan disimpan pada *record 0* dengan *signature "on"*, lalu serial monitor akan mencetak atau menunjukkan perintah "*Speak Now*". Lalu suara dari kata diucapkan sekali yaitu "*on*", dan saat serial monitor mencetak atau menunjukkan perintah "*Speak Again*" suara diucapkan sekali lagi "*on*". Jika dua suara yang telah diucapkan sama atau cocok, serial monitor akan mencetak perintah "*Success*" dan "*record 0*" telah terekam oleh modul. Jika dua suara tidak sama, ulangi prosedur sebelumnya hingga dua suara yang direkam sama atau cocok.

Pada saat merekam suara terdapat dua lampu LED pada modul yang digunakan saat proses perekaman. Setelah mengirimkan perintah perekaman, LED warna kuning akan berkedip secara cepat yang artinya bersiap untuk mengucapkan kata. Kata yang diucapkan harus jelas pada saat LED merah menyala dan proses perekaman berhenti ketika LED merah mati. Selanjutnya, LED kuning berkedip kembali untuk bersiap mengulang kata yang telah diucapkan sebelumnya. Pada saat kedua suara yang direkam dinyatakan sama maka LED merah dan LED kuning akan berkedip secara bersamaan. Namun jika perekaman gagal LED kuning dan LED merah akan berkedip secara cepat. Kata pertama yang direkam adalah suara dengan *signature "on"*. Pada serial monitor ketik *command "sigtrain*

0 on". Selanjutnya terkirim dan serial monitor akan memberikan perintah "Speak Now". Suara diucapkan dan serial monitor akan memberikan perintah "Speak Again", kemudian suara diucapkan kembali. Jika kedua suara telah sama atau cocok maka serial monitor akan mencetak "success", dan "Record 0" serta "SIG (signature) : on". Kata Kedua yang direkam adalah suara dengan signature "off" pada serial monitor ketika command "sigtrain 1 off". Selanjutnya terkirim dan serial

monitor akan memberikan perintah "Speak Now", kemudian suara diucapkan dan serial monitor akan memberikan perintah "Speak Again". Suara diucapkan kembali, jika kedua suara telah sama atau cocok maka serial monitor akan mencetak "success", dan "Record 1" serta "SIG (signature) : off". Suara yang telah terekam dapat disimpan dengan command "load 0 1" signature "on", dan suara dengan signature "off" ke modul voice recognition.

Perancangan Sistem Pengunci Lemari Dengan Pengenalan Suara

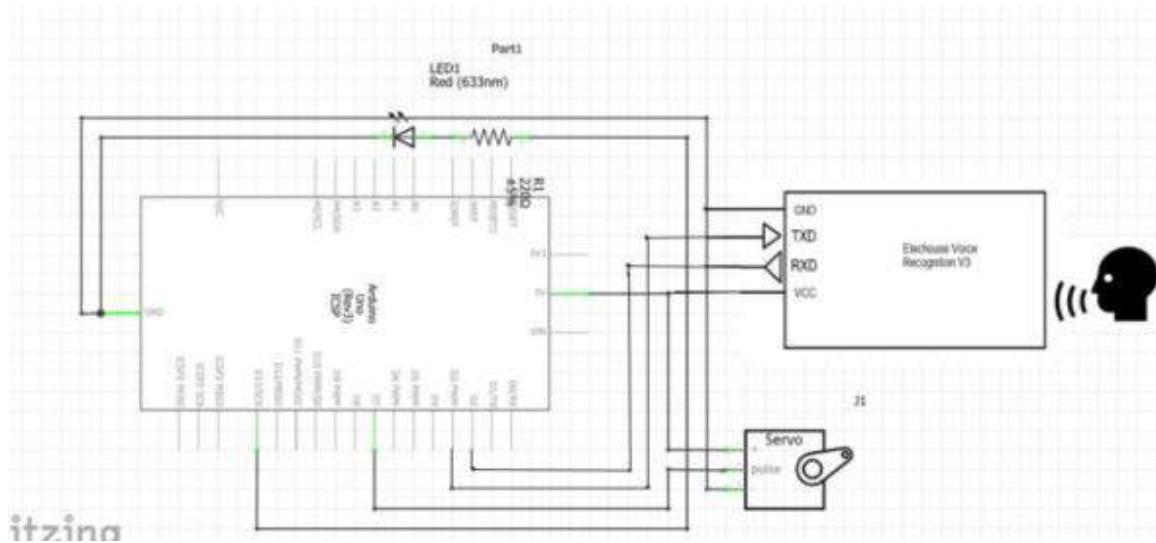


Gambar 2. Sistem Pengunci Lemari dengan Pengenalan Suara

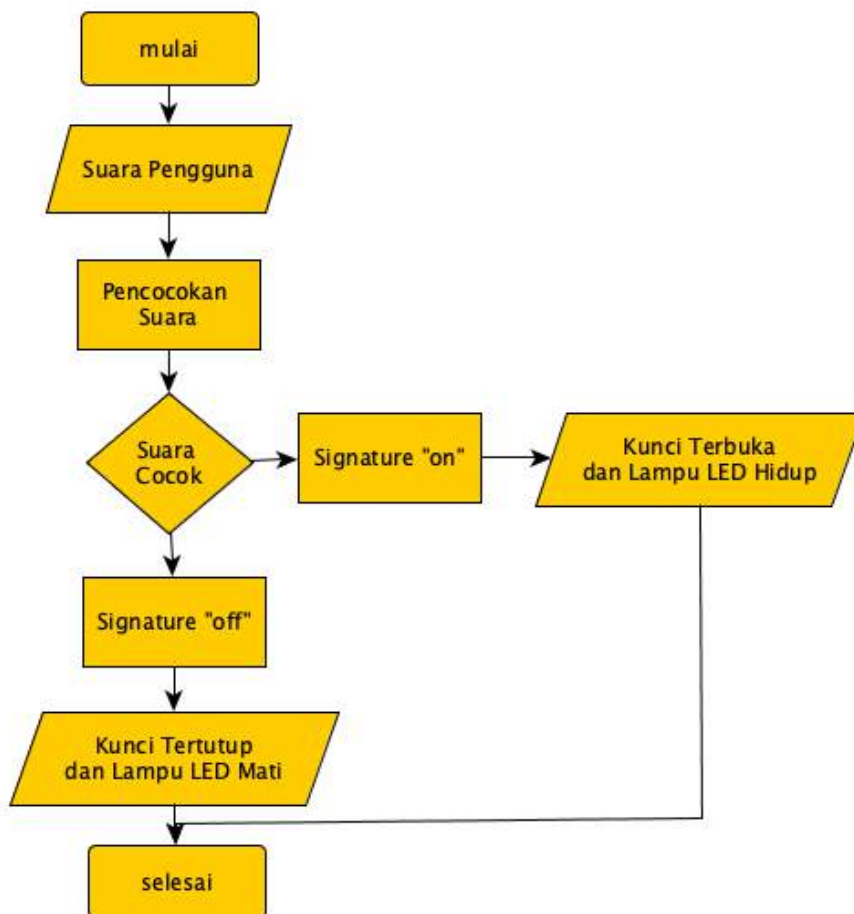
Perancangan sistem pengunci lemari dengan pengenalan suara terlihat pada Gambar 2. Sistem penguncian ini hanya merespon suara yang sama dengan suara yang telah terekam. Suara tersebut memiliki akses untuk membuka dan menutup kunci lemari. Apabila suara pengguna yang terdeteksi cocok dengan suara yang tersimpan, maka suara dikirimkan menuju Arduino Uno melewati jalur UART (*Universal Asynchronous Receiver-Transmitter*) dengan komunikasi serial RX (*received*) dan TX (*transmitted*). Suara yang diterima modul voice recognition selanjutnya dikirimkan

melalui pin TX modul voice recognition menuju pin digital 2 Arduino Uno yang dijadikan sebagai RX untuk menerima masukan (*input*) dari modul voice recognition seperti terlihat pada Gambar 3.

Suara pengguna yang masuk ke dalam mikrokontroler dikondisikan menjadi 2. Kondisi pertama untuk membuka kunci dan menghidupkan lampu LED. Kondisi kedua untuk menutup kunci dan mematikan lampu LED. Kunci lemari digerakkan oleh motor servo berdasarkan suara yang telah dikenali. Lampu LED membutuhkan resistor agar tidak mendapatkan muatan yang berlebih.



Gambar 3. Rangkaian Pengunci Lemari menggunakan Suara



Gambar 4. Flowchart Sistem Pengunci Lemari dengan Pengenalan Suara

Flowchart pada Gambar 4 merupakan alur kerja dari pengunci lemari dengan pengenalan suara untuk mengunci dan membuka kunci. Masukan suara yang masuk ke dalam mikrokontroller didefinisikan sebagai “rekaman”. Rekaman untuk membuka kunci dan menutup kunci seperti dengan kondisi yang telah dibuat. Jika suara yang masuk terdefinisi sebagai rekaman dengan *signature* “off” maka kunci lemari akan bergerak mengunci lemari dan lampu LED akan mati. Jika suara yang masuk terdefinisi sebagai rekaman dengan *signature* “on” maka kunci lemari akan bergerak membuka kunci lemari dan lampu LED akan menyala.

HASIL DAN PEMBAHASAN

Pengujian terhadap rancangan sistem pengunci lemari dilakukan berdasarkan 4 nilai *baud rate* yang berbeda-beda yaitu 2400 bps, 4800 bps, 9600 bps, dan 19200 bps. Setiap pengujian nilai *baud rate* dilakukan 10 kali percobaan sehingga total percobaan yang dilakukan sebanyak 40 kali. Pada pengujian ini digunakan masing-masing dua kata untuk membuka dan menutup kunci. Kata “buka” dan “alpha” digunakan untuk membuka kunci. Kata “tutup” dan “beta” digunakan untuk menutup kunci. Pengujian pertama dengan nilai *baud rate* sebesar 2400 bps diperlihatkan pada Tabel 1.

Tabel 1. Hasil Pengujian Perancangan dengan Nilai *Baud Rate* sebesar 2400 bps

Suara Pengguna	Pengujian	Hasil Percobaan Yang Berhasil	<i>Delay</i>	<i>Sigtrain</i>	Persentase Keberhasilan	Kondisi Kunci
Buka	1	7	< 2 detik	On	70%	Terbuka
Tutup	1	6	< 2 detik	Off	60%	Tertutup
Alpha	1	6	< 2 detik	On	60%	Terbuka
Beta	1	7	< 2 detik	Off	70%	Tertutup
Buka	2	7	< 2 detik	On	70%	Terbuka
Tutup	2	7	< 2 detik	Off	70%	Tertutup
Alpha	2	7	< 2 detik	On	70%	Terbuka
Beta	2	7	< 2 detik	Off	70%	Tertutup

Pada Tabel 1 ditunjukkan berdasarkan hasil pengujian dengan menggunakan *baud rate* sebesar 2400 bps dengan persentase keberhasilan pengenalan suara hingga kunci dapat terbuka dan tertutup sebesar 60% dan 70% pada pengujian pertama. Persentase keberhasilan pengenalan suara hingga kunci dapat terbuka dan tertutup sebesar 70% pada pengujian yang kedua. Pengenalan suara berhasil

membuka dan menutup kunci lemari. *Delay* yang dihasilkan antara pengucapan dengan kondisi kunci sebesar kurang dari 2 detik.

Pengujian kedua dengan dengan nilai *baud rate* sebesar 4800 bps diperlihatkan pada Tabel 2. Hasil pengujian dengan menggunakan nilai *baud rate* sebesar 4800 bps pada Tabel 2 memperlihatkan tingkat keberhasilan pengenalan suara hingga kunci terbuka dan tertutup

sebesar 40% dan 50% pada pengujian pertama. Pengujian kedua diperoleh tingkat keberhasilan pengucapan suara dengan kondisi kunci

sebesar 50%. *Delay* yang dihasilkan antara pengucapan dan kondisi kunci masih kurang dari 2 detik.

Tabel 2. Hasil Pengujian Perancangan dengan Nilai *Baud Rate* sebesar 4800 bps

Suara Pengguna	Pengujian	Hasil Percobaan Yang Berhasil	<i>Delay</i>	<i>Sigtrain</i>	Persentase Keberhasilan	Kondisi Kunci
Buka	1	5	< 2 detik	On	50%	Terbuka
Tutup	1	4	< 2 detik	Off	40%	Tertutup
Alpha	1	5	< 2 detik	On	50%	Terbuka
Beta	1	4	< 2 detik	Off	40%	Tertutup
Buka	2	5	< 2 detik	On	50%	Terbuka
Tutup	2	5	< 2 detik	Off	50%	Tertutup
Alpha	2	5	< 2 detik	On	50%	Terbuka
Beta	2	5	< 2 detik	Off	50%	Tertutup

Pengujian ketiga dengan menggunakan nilai *baud rate* sebesar 9600 bps. Hasil pengujian ditunjukkan pada Tabel 3. Hasil pengujian pertama dan kedua dengan *baud rate* 9600 bps pada Tabel 3 memperlihatkan persentase keberhasilan pengenalan suara

dengan kondisi kunci sebesar 30% dan 40%. *Delay* antara pengucapan dan kondisi kunci dibawah 2 detik. Walaupun tingkat keberhasilannya rendah, namun kondisi kunci masih memperlihatkan hasil yang sesuai dengan *sigtrain*.

Tabel 3. Hasil Pengujian Perancangan dengan Nilai *Baud Rate* sebesar 9600 bps

Suara Pengguna	Pengujian	Hasil Percobaan Yang Berhasil	<i>Delay</i>	<i>Sigtrain</i>	Persentase Keberhasilan	Kondisi Kunci
Buka	1	4	< 2 detik	On	40%	Terbuka
Tutup	1	3	< 2 detik	Off	30%	Tertutup
Alpha	1	4	< 2 detik	On	40%	Terbuka
Beta	1	4	< 2 detik	Off	40%	Tertutup
Buka	2	3	< 2 detik	On	30%	Terbuka
Tutup	2	4	< 2 detik	Off	40%	Tertutup
Alpha	2	4	< 2 detik	On	40%	Terbuka
Beta	2	3	< 2 detik	Off	30%	Tertutup

Pengujian terakhir dengan menggunakan *baud rate* sebesar 19200 bps diperlihatkan pada Tabel 4. Berdasarkan Tabel 4 persentase keberhasilan pengenalan suara dan kondisi kunci dengan *baud rate* sebesar 19200 bps

adalah 20% dan 30%. Dibandingkan dengan nilai *baud rate* yang lain maka pada *baud rate* 19200 bps memiliki keberhasilan paling kecil. *Delay* antara pengenalan suara dan kondisi kunci masih dibawah 2 detik.

Tabel 4. Hasil Pengujian Perancangan dengan Nilai *Baud Rate* sebesar 19200 bps

Suara Pengguna	Pengujian	Hasil Percobaan Yang Berhasil	<i>Delay</i>	<i>Sigtrain</i>	Persentase Keberhasilan	Kondisi Kunci
Buka	1	2	< 2 detik	On	20%	Terbuka
Tutup	1	3	< 2 detik	Off	30%	Tertutup
Alpha	1	2	< 2 detik	On	20%	Terbuka
Beta	1	2	< 2 detik	Off	20%	Tertutup
Buka	2	3	< 2 detik	On	30%	Terbuka
Tutup	2	3	< 2 detik	Off	30%	Tertutup
Alpha	2	2	< 2 detik	On	20%	Terbuka
Beta	2	2	< 2 detik	Off	20%	Tertutup

Berdasarkan percobaan yang dilakukan terhadap empat nilai baud rate, nilai *baud rate* sebesar 2400 bps memperlihatkan hasil yang paling baik dibandingkan dengan nilai lainnya. Hal ini dikarenakan tingkat keberhasilan pengenalan suara terhadap kondisi kunci lebih tinggi yaitu hingga 70% dari 10 kali pengujian untuk setiap kata yang diucapkan. Nilai *baud rate* sebesar 19200 bps menghasilkan tingkat keberhasilan pengenalan pengucapan dengan kondisi kunci hanya sebesar 20% - 30% dari 10 kali pengujian untuk setiap kata yang diucapkan.

KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan dan pengujian maka sebuah sistem pengunci lemari dengan pengenalan suara telah berhasil dibuat. Hasil pengujian memperlihatkan bahwa dengan *baud rate* sebesar 2400 bps, suara lebih banyak dikenali yaitu hingga 70% dari 10 kali pengujian dibandingkan dengan suara yang memiliki nilai *baud rate* lainnya yang telah diuji. Sistem berhasil mengenali

suara untuk membuka dan menutup kunci lemari. *Delay* yang dihasilkan dari semua pengujian antara pengucapan suara dengan kondisi kunci dibawah 2 detik. Hal tersebut menunjukkan perancangan sistem sudah baik.

Hasil pengujian pada penelitian ini hanya mencapai 70%, sehingga sistem ini perlu dirancang ulang supaya tingkat keberhasilan hingga 100%. Pada penelitian selanjutnya sistem pengunci lemari dengan pengenalan suara dapat dikembangkan menggunakan modul pengenalan suara lainnya yang telah dibuat sendiri atau yang tersedia dipasaran sehingga sistem pengunci lemari menjadi lebih akurat.

DAFTAR PUSTAKA

- [1] T. Pratama, *Merancang pengantar-mukaan kunci pembuka pintu mobil berbasis mikrocontroller*. Universitas Gunadarma: Jakarta, 2013.
- [2] R. Ulansari, "Perancangan Keamanan Akses Ruang Menggunakan Verifikasi Biometrik Pengenalan Wajah dan Sidik

- Jari,” *Tesis*, Universitas Gunadarma, Jakarta, 2015.
- [3] M. S. Alam, “Rancang Bangun Sistem Keamanan Benda Museum Menggunakan RFID Berbasis Mikrokontroler Atmega 8535,” *Skripsi*, Universitas Gunadarma, Jakarta, 2013.
- [4] A. Imario, D. W. Sudiharto, dan E. Ariyanto, “Uji Validasi Suara Berbasis Pengenalan Suara (Voice Recognition) Menggunakan Easy Vr 3.0,” *Prosiding SNATIF ke 4, Fakultas Teknik, Universitas Muria Kudus*, 2017, hal. 801 – 806.
- [5] A. N. Seppiawan, Nurussa'adah, dan P. Siwindarto, “Sistem Keamanan Pintu Pagar Otomatis Menggunakan Voice Recognition,” *Jurnal Mahasiswa Teknik Elektro Universitas Brawijaya*, vol. 2, no. 6, hal. 1 – 6, 2014.
- [6] A. W. Dani, A. Adriansyah, dan D. Hermawan, “Perancangan Aplikasi Voice Command Recognition Berbasis Android dan Arduino Uno,” *Jurnal Teknologi Elektro, Universitas Mercu Buana*, vol. 7, no. 1, hal. 11 – 19, 2016.
- [7] A. B. Nugroho, H. Setyawan, dan L. A. Basuki, “Pembuatan prototype robot beroda berbasis mikrokontroler dan sensor easy voice recognition sebagai alat bantu penderita disabilitas,” *Jurnal Teknologi Proses dan Inovasi Industri*, vol. 2, no. 1, hal. 37 – 41, 2016.

OPTIMASI RUTE TRUK PENGANGKUT SAMPAH DI KOTA DEPOK

¹Tegar Mujadid Mappa, ²Sudaryanto

^{1,2}Fakultas Teknologi Industri Universitas Gunadarma

Jalan Margonda Raya No 100, Pondok Cina, Depok, 16424

¹mujadidtegar@gmail.com, ²sudaryanto@staff.gunadarma.ac.id

Abstrak

Kota Depok merupakan salah satu wilayah dengan jumlah populasi penduduk terpadat di Indonesia. Jumlah penduduk yang besar menimbulkan permasalahan salah satunya adalah sampah. Pengelolaan sampah yang baik terutama pada pengangkutan sampah dari TPS hingga TPA harus diperhatikan supaya proses transportasi efektif dan efisien. Masalah ini terkait dengan penentuan rute truk sampah. Pada penelitian ini penentuan rute transportasi truk sampah di kota Depok dilakukan menggunakan metode Nearest Neighbour. Metode ini digunakan karena sesuai dengan permasalahan yang terjadi dimana terdapat kendala yaitu kapasitas muatan yang terbatas. Langkah penyelesaiannya yaitu truk mengawali perjalanan dari depot yang berlokasi di area TPA Cipayung menuju TPS terdekat lalu menuju TPA untuk membongkar muatan dan kembali ke depot. Berdasarkan penyelesaian menggunakan metode Nearest Neighbour didapatkan hasil terjadi penggabungan 6 rute baru. Jarak tempuh seluruh armada dalam satu hari yang sebelumnya 845,6 km menjadi 775,6 km sehingga penghematan jarak tempuh sekitar 8,3%. Jumlah truk untuk mengangkut sampah sebanyak 197 m³ yang sebelumnya 40 armada menjadi hanya 34 armada, sehingga ada penghematan sekitar 15%. Waktu waktu proses yang digunakan seluruh armada truk ke seluruh TPS hingga kembali ke depot sebelumnya adalah 9914 menit, dengan metode Nearest Neighbour didapatkan pengurangan waktu proses menjadi 9457 menit, sehingga menghemat waktu sekitar 4,6%.

Kata Kunci: kota Depok, nearest neighbour, optimasi, rute, truk sampah.

Abstract

Depok city is one of the regions with the densest population in Indonesia. A large population certainly raises problems and one of them is about waste, to overcome this, a good management is needed, especially waste management from TPS to TPA which involves the transportation process. The solution to the problem is the garbage truck transportation route in Depok, which uses the Nearest Neighbor method. This method is used because it is in accordance with the problems happened, namely limited load capacity. The completion step is the truck starts the journey from the storehouse located in the Cipayung landfill area to the nearest polling station and then goes to the landfill to unload and return to the storehouse. Based on the solution using the Nearest Neighbour method, the results obtained are the occurrence of combining 6 new routes. The mileage of the entire fleet in one day which was previously 845.6 km became 775.6 km so there was a savings of about 70 km or around 8.3%. The number of trucks used to transport garbage is 197 m³, of which 40 were previously only 34 fleets, resulting in savings of 6 truck fleets or around 15%. The processing time used by the entire fleet of garbage transportation trucks to all polling stations until returning to the previous depot was 9914 minutes, but after following the steps of the Nearest Neighbor method, it was found that the processing time was reduced to 9457 minutes, thus saving time by 4.6%.

Keywords: Depok city, garbage trucks, nearest neighbour, optimization, routes.

PENDAHULUAN

Prospek pekerjaan yang besar di kota membuat peluang mendapatkan pekerjaan juga semakin besar, hal ini dapat memberikan dampak yaitu semakin banyaknya perpindahan masyarakat dari desa ke kota. Perpindahan masyarakat ke kota secara besar-besaran tersebut dapat berpengaruh pada meningkatnya jumlah populasi penduduk di kota yang dapat menyebabkan masalah sosial. Permasalahan sosial adalah salah satu permasalahan yang memiliki dampak yang sangat besar terutama mengenai sampah.

Problematika mengenai sampah merupakan hal yang sangat penting terutama di daerah perkotaan yang berkaitan dengan budaya dan perilaku masyarakat terutama di wilayah perkotaan [1]. Kurangnya kesadaran dan kepedulian dari masyarakat mengenai permasalahan sampah akan berdampak pada kualitas lingkungan yang semakin mengalami penurunan. Peran dari instansi pemerintahan terkait sangat diperlukan untuk mengatasi permasalahan tersebut. Pengelolaan sampah dibagi menjadi 2 yaitu pengelolaan sampah setempat (individu) dan pengelolaan sampah terpusat untuk suatu lingkungan permukiman atau kota. Pengelolaan sampah secara terpusat memerlukan teknis operasional yang terkoordinir dan institusi yang menangani langsung pengelolaan persampahan [2].

Pengelolaan persampahan suatu daerah sangat ditentukan oleh peraturan yang mendukungnya terutama dalam hal pengelolaan

sampah yang dimulai dari Tempat Penampungan Sementara (TPS) sampai dengan Tempat Pemrosesan Akhir (TPA). Pengelolaan sampah dari TPS sampai TPA tersebut melibatkan proses transportasi yang dimana truk sampah digunakan sebagai kendaraan untuk melakukan pengangkutannya.

Proses pengangkutan sampah ke setiap wilayah mengalami beberapa permasalahan, salah satu contohnya masalah yang muncul di wilayah kota Depok. Masalah yang muncul adalah penentuan rute pengangkutan sampah yang kurang efisien dengan kendala kapasitas muatan dari truk sampah dan biaya operasional yang terbatas. Berdasarkan kondisi tersebut, maka diperlukan kajian terutama untuk mengoptimalkan rute pengangkutan sampah sehingga pengangkutan sampah dari TPS ke TPA menjadi efisien. Selain itu, dengan dilakukannya pengoptimalan rute transportasi truk sampah diharapkan dapat memaksimalkan pengangkutan sampah dan memperpendek jaraknya tempuhnya.

Masalah pengangkutan sampah dari TPS ke TPA merupakan salah satu contoh *Vehicle Routing Problem* (VRP). Terdapat 5 variasi dari bentuk VRP muncul tergantung pada suatu kondisi yang ada. Pertama *Vehicle Routing Problem With Time Windows* (VRPTW) adalah masalah optimasi kombinatorial yang berkaitan dengan terbatasnya waktu penyediaan layanan. Kedua, *Open Vehicle Routing Problem* (OVRP) yaitu model VRP dengan rute terbuka. Ketiga *Multi-Depot Vehicle Routing Problem* (MDVRP)

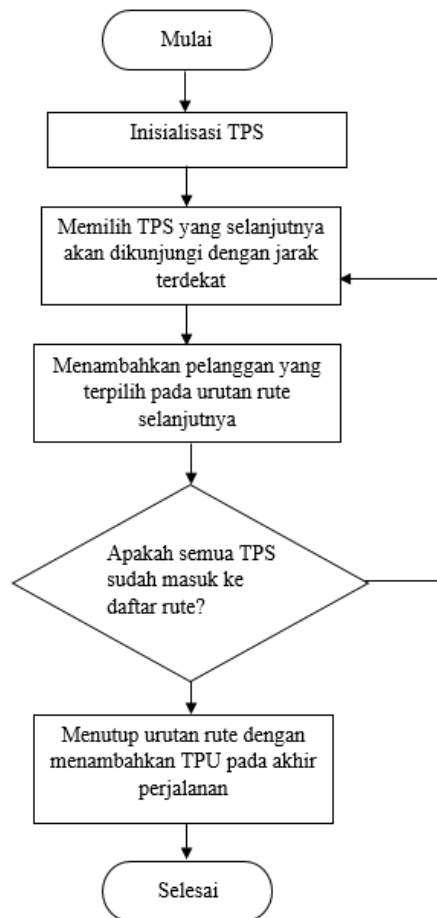
merupakan permasalahan VRP dengan kondisi dimana depot yang digunakan sebagai pusat distribusi barang lebih dari satu. Keempat *Site-Dependent Vehicle Routing Problem* (SDVRP) digunakan untuk melayani satu set pelanggan, tetapi ada ketergantungan kompatibilitas antara pelanggan dan jenis kendaraan. Kelima *Capacitated Vehicle Routing Problem* (CVRP) yaitu VRP yang memiliki batasan kapasitas kendaraan [3,4].

Penelitian terdahulu mengenai CVRP dilakukan untuk optimalisasi rute jasa pengiriman barang menggunakan pendekatan heuristik [3]. Permasalahan CVRP yang lain adalah kasus pendistribusian air isi ulang menggunakan metode *sequential insertion* dan berhasil menentukan rute distribusi dengan lebih efektif [5]. Penelitian terkait pengelolaan sampah sudah banyak dilakukan salah satunya untuk menghitung jumlah dump truck dan arm roll truck yang dibutuhkan untuk mengangkut sampah yang dihasilkan di Kecamatan Klungkung ke TPA [6]. Tujuan penelitian ini adalah optimasi rute truk pengangkutan sampah dari TPS ke TPA sehingga didapatkan rute terpendek dan sampah yang terangkut maksimal. Optimasi dapat dilakukan dengan algoritma optimasi deterministik dan probabilistik. Algoritma optimasi probabilistik digunakan untuk menyelesaikan ruang masalah dengan ruang solusi yang besar dan bahkan tak terbatas [7]. Pada penelitian ini, optimasi menggunakan algoritma *nearest neighbour*.

METODE PENELITIAN

Permasalahan pengangkutan sampah dari TPS ke TPA merupakan permasalahan untuk mengatur aktivitas distribusi sumber daya tertentu dengan keterbatasan jumlah armada dan kapasitas angkutnya. Permasalahan ini termasuk dalam permasalahan CVRP. Parameter-parameter model CVRP antara lain jumlah TPS, kapasitas setiap kendaraan, permintaan TPS dan jarak tempuh perjalanan dari TPS ke TPS. Parameter tersebut dianggap bernilai integer tidak negatif. Pendekatan yang digunakan adalah algoritma *nearest neighbour*.

Algoritma *nearest neighbour* merupakan prosedur memulai rute kendaraan dari jarak yang paling dekat dengan depot. Rute selanjutnya yaitu pelanggan yang paling dekat dengan pelanggan pertama yang sudah dikunjungi. Prosedur ini terus berulang sampai semua pelanggan masuk ke dalam rute perjalanan. Keistimewaan dari metode ini merupakan salah satu metode yang digunakan untuk menjadwalkan sejumlah terbatas kendaraan dengan memperhatikan kapasitas maksimum. Metode *nearest neighbour* digunakan karena metode ini memiliki karakteristik pembentukan rute distribusi sesuai dengan keadaan nyata yang terdapat pada kondisi lapangan. Alasan lain penggunaan metode ini karena lebih mudah dilakukan dibandingkan dengan metode VRP yang lain. Diagram alir metode *nearest neighbour* pada penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Algoritma *Nearest Neighbour*

Pada Gambar 1 ditunjukkan diagram alir algoritma *nearest neighbour*. Langkah pertama adalah menetapkan kapasitas muatan kendaraan, berawal dari depot kemudian menuju TPS yang belum dikunjungi dengan jarak terpendek. Langkah kedua yaitu ke pelanggan lain yang memiliki jarak terdekat dari pelanggan yang terpilih sebelumnya dan jumlahnya tidak melebihi kapasitas muatan kendaraan. Apabila ada pelanggan yang terpilih sebagai pelanggan berikutnya dan terdapat sisa kapasitas kendaraan, kembali ke langkah (2). Bila kendaraan tidak memiliki sisa kapasitas, kembali ke langkah (1). Bila tidak ada lokasi yang terpilih karena

jumlah pengiriman melebihi kapasitas kendaraan, maka kembali ke langkah (1). Dimulai lagi dari depot dan mengunjungi pelanggan yang belum dikunjungi yang memiliki jarak terdekat. Langkah ke tiga yaitu memastikan semua pelanggan telah dikunjungi. Bila semua pelanggan telah dikunjungi tepat satu kali maka algoritma berakhir [5].

Kota Depok terdiri dari 11 kecamatan dengan jumlah keseluruhan TPS adalah 32 TPS yang tersebar di setiap kecamatan. Tabel 1 menyajikan data TPS yang ada di kota Depok dengan depot di TPA Cipayung dengan kapasitas truk sebesar 6 m³.

Tabel 1. TPS di Kota Depok

Kecamatan	TPS	Jumlah Timbunan (perhari)	Kecamatan	TPS	Jumlah Timbunan (perhari)
Beji	TPS Sanin	6 m ³	Cimanggis	TPS Tugu	6 m ³
	TPS Curug	3 m ³		TPS Tugu	3 m ³
	TPS Rais	4 m ³		TPS Pasar Cisalak	6 m ³
Pancoran Mas	TPS Mawar	6 m ³	Cinere	TPS Pasar Cisalak	6 m ³
	TPS H. Naidih	5 m ³		TPS Pasar Cisalak	6 m ³
	TPS Keadilan	6 m ³		TPS Pasar Cisalak	6 m ³
	TPS Jaya	2 m ³		TPS Pasar Cisalak	6 m ³
Cipayung	TPS Jl. Raya Cipayung	5 m ³	Tapos	TPS Pasar Cisalak	6 m ³
	TPS Permata	3 m ³		TPS Rusunawa	4 m ³
	TPS Pitara	3 m ³		TPS Lewinanggung	6 m ³
	TPS Bulak	4 m ³		TPS Jl. Raya Tapos	4 m ³
Sukmajaya	TPS Dipo	6 m ³	Sawangan	TPS Pahlawan	6 m ³
	TPS Antena Raya	2 m ³		TPS Jabon	6 m ³
	TPS nail	5 m ³		Bojongsari	TPS Serua
TPS Merdeka	6 m ³	TPS H Suhaemi	6 m ³		
Cilodong	TPS Parung	5 m ³			
	TPS Gani	6 m ³			
Limo	TPS Iskandar	6 m ³			
	TPS Boulevard	6 m ³			
	TPS Aster	6 m ³			
	TPS Pendowo	3 m ³			

(Sumber: DLHK, 2019)

Pada Tabel 1 merupakan tempat penampungan sementara (TPS) yang ada di kota Depok. TPS tersebut tidak hanya berupa lahan kosong yang disediakan oleh dinas kebersihan kota Depok, namun juga berupa bak kontainer yang berukuran 6 m³ yang ditempatkan di lokasi-lokasi tertentu dan juga terdapat titik-titik kumpul dari gerobak sampah. Jumlah TPS yang ada di kota Depok sangat sedikit akibat keterbatasan lahan, sehingga terdapat titik-titik kumpul dari gerobak sampah yang dijadikan TPS. TPS yang berupa titik kumpul tersebut lokasinya tidak tetap, sehingga titik kumpul terkadang

harus pindah dari satu lokasi ke lokasi lain dikarenakan adanya penolakan dari warga.

Berdasarkan data lokasi TPS di kota Depok, maka dibuatlah penomoran dari setiap TPS, hal tersebut dilakukan untuk memudahkan dalam mengolah data pada tahap selanjutnya. Pemberian nomor tersebut hanya berdasarkan asumsi pribadi dari penulis, tidak diurutkan berdasarkan tiap wilayah kecamatan, sehingga dalam penomoran tersebut bisa saja antara TPS 1 dan TPS 2 berada di wilayah kecamatan yang berbeda. Tabel 2 menyajikan urutan penomoran TPS di Kota Depok

Tabel 2. Penomoran TPS di Kota Depok

No	Nama TPS	No	Nama TPS
1	TPS Pitara	17	TPS Merdeka
2	TPS Bulak	18	TPS Nail
3	TPS Dipo	19	TPS Pasar Cisalak
4	TPS H. Naidih	20	TPS Tugu
5	TPS Keadilan	21	TPS Mawar
6	TPS Jabon	22	TPS Rais
7	TPS Jl. Raya Cipayung	23	TPS Sanin
8	TPS Permata	24	TPS Jaya
9	TPS Boulevart	25	TPS Pendowo
10	TPS Gani	26	TPS Curug
11	TPS Parung	27	TPS Anggrek
12	TPS Rusunawa	28	TPS Nangka Timur
13	TPS Jl. Raya Tapos	29	TPS Aster
14	TPS Lewinanggun	30	TPS Pahlawan
15	TPS Iskandar	31	TPS Serua
16	TPS Antena Raya	32	TPS H. Suhaemi

(Sumber: DLHK, 2019)

Tabel 2 menyajikan data TPS yang ada di kota Depok. Data tersebut menunjukkan setiap TPS memiliki jumlah timbunan sampah yang berbeda-beda. TPS yang ada di kota Depok sebagian besar dikunjungi oleh satu armada truk sampah yang berukuran 6 m³, tetapi ada beberapa TPS yang dikunjungi oleh lebih dari satu armada truk sampah.

Formulasi Masalah

Pada penelitian ini ditetapkan kota Depok sebagai sampel dalam penelitian. Proses pengangkutan sampah tersebut menggunakan *dump truck* dengan kapasitas 6m³. Proses pengangkutan sampah dilakukan selama lima hari dimulai dari hari Senin sampai dengan hari Jumat dan beberapa sampai hari Sabtu. Pengangkutan dilakukan oleh empat orang petugas dimana satu orang sebagai supir dan tiga orang sebagai kernet yang bertugas

mengangkut sampah ke dalam truk. Jam operasi pengangkutan sampah dimulai pukul 5.30 WIB sampai dengan pukul 02.00 WIB, dengan titik awal pemberangkatan di TPA Cipayung dimana seluruh armada truk pengangkutan sampah di kota Depok ditempatkan di sana. Biaya bahan bakar yang disediakan untuk pengangkutan yaitu sebanyak 20 liter solar untuk setiap harinya sehingga jika diasumsikan harga bahan bakar solar saat ini adalah Rp 5.150/liter maka pengeluaran yang dibutuhkan setiap harinya adalah Rp 103.000,00. Pengangkutan sampah dilakukan di TPS yang tersebar di kota Depok, kemacetan diabaikan dan kecepatan truk konstan, yaitu 40 km/jam.

Berdasarkan data rute pengangkutan sampah saat ini diketahui proses pengangkutan sampah di kota Depok kurang memaksimalkan kapasitas volume dari truk yang digunakan.

Oleh sebab itu akan dibentuk rute baru untuk menyelesaikan proses pengangkutan sampah yang tidak dimaksimalkan pada rute sebelumnya. Fungsi tujuan dari model penentuan rute truk pengangkut sampah pada penelitian ini adalah meminimumkan total jarak tempuh truk pengangkut sampah dengan memperhatikan kendala. Kendala yang ada yaitu setiap rute melayani TPS diawali dari depot (TPA Cipayung), setiap TPS hanya dilayani tepat satu kali oleh satu truk, jumlah kapasitas muatan yang diangkut pada suatu *trip* dalam suatu rute tidak melebihi kapasitas maksimal truk, dan setiap rute diakhiri pada depot (dalam hal ini TPA Cipayung) dalam keadaan kosong (tidak membawa sampah).

HASIL DAN PEMBAHASAN

Pada penelitian ini digunakan asumsi-asumsi yaitu lokasi titik awal pemberangkatan truk sampah disimbolkan dengan 0. TPS disimbolkan dengan penomoran angka dan lokasi TPA disimbolkan dengan huruf X. Pemuatan sampah 1m³ memakan waktu 30 menit dan untuk proses pengosongan muatan memakan waktu 20 menit. Data jarak tempuh dari depot ke setiap TPS diperoleh dengan menggunakan *Google Maps*. Tabel 3 menunjukkan rute masing-masing truk sampah dari depot ke setiap TPS di kota Depok.

Tabel 3. Rute Truk Pengangkut Sampah di Kota Depok

No	Rute	Kapasitas Angkut (m ³)	Jarak Tempuh (Km)	Waktu Proses (Menit)
1	0-1-X-0	3	3,0	220
2	0-2-X-0	4	4,2	166
3	0-3-X-0	6	13,0	256
4	0-4-X-0	5	7,6	210
5	0-5-X-0	6	5,6	222
6	0-6-X-0	6	14,2	240
7	0-7-X-0	5	6,4	200
8	0-8-X-0	3	11,8	160
9	0-9-X-0	6	20,4	280
10	0-10-X-0	6	26,2	294
11	0-11-X-0	6	18,0	276
12	0-12-X-0	6	28,8	248
13	0-13-X-0	4	40,4	266
14	0-14-X-0	6	39,2	344
15	0-15-X-0	6	21,8	286
16	0-16-X-0	6	24,4	190
17	0-17-X-0	6	19,0	282
18	0-18-X-0	5	20,6	260
19	0-19 ¹ -X-0	6	30,4	294
20	0-19 ² -X-0	6	30,4	294
21	0-19 ³ -X-0	6	30,4	294
22	0-19 ⁴ -X-0	6	30,4	294
23	0-19 ⁵ -X-0	6	30,4	294

24	0-19 ⁶ -X-0	6	30,4	294
25	0-19 ⁷ -X-0	6	30,4	294
26	0-19 ⁸ -X-0	2	30,4	174
27	0-20 ¹ -X-0	6	25,6	300
28	0-20 ² -X-0	3	25,6	210
29	0-21-X-0	6	12,2	260
30	0-22-X-0	3	15,4	206
31	0-23-X-0	6	13,8	260
32	0-24-X-0	2	10,4	122
33	0-25-X-0	3	13,8	164
34	0-26-X-0	3	17,6	180
35	0-27-X-0	3	22,0	220
36	0-28-X-0	6	26,0	296
37	0-29-X-0	5	19,6	240
38	0-30-X-0	6	23,8	300
39	0-31-X-0	5	28,0	250
40	0-32-X-0	6	24,0	274
Total		197	845,6	9.914

Tabel 3 merupakan rute truk sampah yang beroperasi saat ini, yaitu dimulai dari depot menuju TPS, lalu menuju TPA untuk mengosongkan muatan dan kembali lagi ke depot. Pengoperasian truk yang dilakukan saat ini sebanyak satu ritasi dalam satu hari, sesuai dengan aturan yang ditetapkan. Jarak tempuh setiap TPS berbeda-beda. Jarak TPS terjauh berdasarkan pengukuran dari *Google Maps* yaitu TPS Jl. Raya Tapos dengan jarak 40,4 km. Waktu yang dihabiskan seluruh armada untuk mengangkut sampah di kota Depok yaitu 9.914 menit dan sampah yang terangkut 197 m³. Biaya operasional untuk bahan bakar per harinya yaitu sebanyak 20 liter solar sehingga cukup digunakan untuk mengangkut sampah yang memiliki lokasi terjauh.

Sistem pengelolaan per wilayah ini membuat pengelolaan pengangkutan sampah di kota Depok tidak efisien karena truk hanya melakukan pengangkutan sampah hanya di

TPS yang ditetapkan dan tidak mengangkut sampah di TPS lain yang berbeda wilayahnya walaupun masih memiliki muatan yang cukup. Berdasarkan hal tersebut dilakukan pembentukan rute baru menggunakan *Nearest Neighbour*. Pembentukan rute dilakukan berdasarkan jarak terdekat dengan depot tanpa memperhatikan wilayah-wilayah TPS.

Pembentukan rute pengangkut sampah dengan metode *Nearest Neighbour* terdiri dari beberapa langkah. Langkah pertama yaitu menetapkan kapasitas muatan kendaraan yang berawal dari depot yang berada di area TPA Cipayung, kemudian menuju lokasi yang belum dikunjungi dan memiliki jarak terpendek dari depot sebagai lokasi pertama. Langkah kedua yaitu ke TPS yang memiliki jarak terdekat dari pelanggan yang terpilih sebelumnya dan jumlahnya tidak melebihi kapasitas muatan kendaraan. Apabila ada pelanggan yang terpilih sebagai pelanggan berikutnya dan terdapat sisa kapasitas

kendaraan, kembali ke langkah (2). Bila kendaraan tidak memiliki sisa kapasitas, kembali ke langkah (1). Bila tidak ada lokasi yang terpilih karna jumlah pengiriman melebihi kapasitas kendaraan, maka kembali ke langkah (1). Dimulai lagi dari depot dan mengunjungi pelanggan yang belum dikunjungi yang memiliki jarak terdekat.

Langkah ketiga yaitu memastikan semua pelanggan telah dikunjungi. Bila semua pelanggan telah dikunjungi tepat satu kali maka algoritma berakhir. Berdasarkan langkah-langkah penyelesaian masalah dengan metode *Nearest Neighbour*, maka dibuat dalam bentuk model matematik yang disajikan pada Tabel 4.

Tabel 4. Pembentukan Rute pada TPS 1

Tahap	Algoritma
Langkah 1	Truk sampah mengawali perjalanan dari depot yang berlokasi di TPA Cipayung dengan muatan dalam kendaraan (Q) = 0 m ³ untuk rute pertama ($k = 1$) pada trip pertama ($t = 1$)
Langkah 2	Memilih TPS yang paling dekat dengan depot yaitu yang terdekat adalah TPS 1 dengan jarak 1,6 km, sehingga rute yang terbentuk menjadi 0 – 1 – X – 0.
Langkah 3	Hitung permintaan muatan ($Q = Q + d_i$). Volume sampah pada TPS 1 (d_1) = 3 m ³ , maka kapasitas sampah dalam truk (Q) = $Q + (d_1) = 0 + 3 = 3 \text{ m}^3 < Q_{maks}$.
Langkah 4	Jika $Q < Q_{maks}$, lanjutkan ke langkah (5). Jika $Q = Q_{maks}$, lanjutkan ke langkah (7). Jika $Q > Q_{maks}$, lanjut ke langkah (6). Berdasarkan pada langkah ke (3), diketahui muatan sampah dalam truk adalah $Q < Q_{maks}$
Langkah 5	Bentuk konsumen yang terpilih sebagai lokasi awal, kemudian ulangi langkah (2). Muatan sampah dalam truk adalah $Q < Q_{maks}$ sehingga dicari TPS yang terdekat dari TPS 1. TPS yang terpilih adalah TPS 2 dengan jarak 1 km
Langkah 6	Batalkan pemilihan konsumen terakhir, kemudian pilih konsumen lain yang belum terpilih yang paling dekat dengan lokasi awal dan lanjutkan ke langkah (3). Jika semua konsumen tidak ada yang layak, lanjutkan ke langkah (7). Sampah di TPS 2 adalah 4 m ³ , jika TPS 2 dipilih maka $Q > Q_{maks}$, sehingga dicari TPS lain yang terdekat dan terpilihlah TPS 8 lalu dilanjutkan dengan langkah (3)
Langkah 7	Kendaraan menuju X untuk membongkar muatan. Bentuk trip baru ($t = t + 1$), kemudian ulangi langkah (1). Rute yang terbentuk yaitu 0 – 1 – 8 – X – 0 dan muatan sampah dalam truk $Q = Q_{maks}$, maka truk menuju TPA Cipayung untuk membongkar muatan
Langkah 8	Batalkan pemilihan konsumen terakhir, kemudian kembali ke langkah (1) untuk

membentuk rute berikutnya ($k = k + 1$).

Langkah 9 Semua konsumen telah terpilih, maka pencarian rute selesai.

Berdasarkan pembentukan rute pada TPS 1 yang disajikan dalam bentuk Tabel 4 maka dijelaskan secara rinci langkah penyelesaiannya. Langkah tersebut dimulai dengan pembentukan trip pertama ($t = 1$) pada rute pertama ($k = 1$) yaitu truk mengawali perjalanan dalam keadaan kosong dari depot (0) yang berada di TPA Cipayung yang merupakan pangkalan semua armada truk, menuju TPS yang paling terdekat yaitu TPS 1 dengan jarak 1,6 km. Rute yang terbentuk menjadi $0 - 1 - X - 0$. Selanjutnya diuji kelayakannya yaitu volume sampah pada TPS 1 (d_1) = 3 m^3 , maka kapasitas sampah dalam truk (Q) = $Q + (d_1) = 0 + 3 = 3 \text{ m}^3 < Q \text{ maks}$. Oleh karena $Q \leq Q \text{ maks}$, maka pemilihan TPS 1 masuk dalam kategori layak sehingga rute $0 - 1 - X - 0$ dianggap layak.

Lokasi terakhir yang dikunjungi truk adalah TPS (1), maka dipilih TPS yang terdekat dengan TPS (1). Berdasarkan hasil pengukuran jarak, diketahui TPS yang memiliki jarak terdekat adalah TPS (2) dengan jarak 1 km dengan demikian maka rute yang terbentuk menjadi $0 - 1 - 2 - X - 0$. Selanjutnya diuji kelayakannya yaitu volume sampah pada TPS 2 (d_2) = 4 m^3 , maka kapasitas sampah dalam truk (Q) = $Q + (d_2) = 3 + 4 = 7 \text{ m}^3 > Q \text{ maks}$. oleh karena $Q \geq Q \text{ maks}$, maka pemilihan TPS 2 masuk dalam kategori tidak layak sehingga harus dicari rute terdekat dan volume sampah yang tidak

melebihi kapasitas. Berdasarkan pertimbangan jarak terdekat dan volume sampah maka TPS (8) terpilih sebagai rute selanjutnya. Selanjutnya diuji kelayakannya yaitu volume sampah pada TPS 8 (d_8) = 3 m^3 , maka kapasitas sampah dalam truk (Q) = $Q + (d_8) = 3 + 3 = 6 \text{ m}^3 = Q \text{ maks}$. Oleh karena $Q = Q \text{ maks}$, maka pemilihan TPS 8 masuk dalam kategori layak sehingga rute $0 - 1 - 8 - X - 0$ dianggap layak. Volume sampah dalam truk sama dengan kapasitas maksimal truk, sehingga truk menuju TPA Cipayung untuk membongkar muatan. Total jarak tempuh perjalanan yang dimulai dari Depot menuju TPS (1) lalu ke TPS (8) lalu ke TPA dan kembali ke depot adalah sejauh 12,7 km.

Trip selanjutnya juga dilakukan hal yang demikian sampai semua TPS telah dikunjungi. Berdasarkan aturan *Nearest Neighbour* adalah apabila semua pelanggan telah dikunjungi tepat satu kali maka algoritma berakhir. Seluruh TPS di kota Depok telah dikunjungi berdasarkan aturan *Nearest Neighbour*, maka diketahui ada enam penggabungan rute yang terjadi. Rute yang digabungkan yaitu TPS 1 dengan 8, TPS 2 dengan 24, TPS 12 dengan 16, TPS 19 dengan 20, TPS 25 dengan 22, dan terakhir TPS 26 dengan 27. Berdasarkan penyelesaian menggunakan metode *Nearest Neighbour* tersebut, maka didapatkan hasil seperti pada Tabel 5.

Tabel 5. Rute Baru Menggunakan *Nearest Neighbour*

No	Rute	Kapasitas Angkut (m ³)	Jarak Tempuh (km)	Waktu Proses (menit)
1	0-1-8-X-0	6	12,7	255
2	0-2-24-X-0	6	13,1	259
3	0-3-X-0	6	13,0	256
4	0-4-X-0	5	7,6	210
5	0-5-X-0	6	5,6	222
6	0-6-X-0	6	14,2	240
7	0-7-X-0	5	6,4	200
8	0-9-X-0	6	20,4	280
9	0-10-X-0	6	26,2	294
10	0-11-X-0	6	18,0	276
11	0-12-16-X-0	6	31,7	338
12	0-13-X-0	4	40,4	266
13	0-14-X-0	6	39,2	344
14	0-15-X-0	6	21,8	286
15	0-17-X-0	6	19,0	282
16	0-18-X-0	5	20,6	260
17	0-19 ¹ -X-0	6	30,4	294
18	0-19 ² -X-0	6	30,4	294
19	0-19 ³ -X-0	6	30,4	294
20	0-19 ⁴ -X-0	6	30,4	294
21	0-19 ⁵ -X-0	6	30,4	294
22	0-19 ⁶ -X-0	6	30,4	294
23	0-19 ⁷ -X-0	6	30,4	294
24	0-19 ⁸ -20 ² -X-0	5	33,3	313
25	0-20 ¹ -X-0	6	25,6	300
26	0-21-X-0	6	12,2	260
27	0-23-X-0	6	13,8	260
28	0-25-22-X-0	6	22,6	285
29	0-26-27-X-0	6	24,0	303
30	0-28-X-0	6	26,0	296
31	0-29-X-0	5	19,6	240
32	0-30-X-0	6	23,8	300
33	0-31-X-0	5	28,0	250
34	0-32-X-0	6	24,0	274
Total		197	775,6	9.457

Tabel 5 merupakan rute truk sampah setelah menggunakan metode *Nearest Neighbour*. Truk mengawali perjalanan dari depot yang berlokasi di area TPA Cipayung menuju TPS, lalu menuju TPA untuk mengosongkan muatan dan kembali lagi ke depot. Berdasarkan penyelesaian menggunakan metode *Nearest Neighbour*, maka terjadi

penggabungan enam rute truk sehingga total jarak tempuh yang dilalui seluruh armada truk dalam satu hari adalah 775,6 km dengan waktu beroperasi 9.457 menit dan total sampah yang terangkut yaitu 197 m³.

Berdasarkan rute truk pengangkut sampah yang disajikan pada Tabel 4 maka dilakukan optimalisasi hingga didapatkan

hasil yang disajikan pada Tabel 5. Proses pengangkutan sampah ke seluruh TPS menggunakan armada truk berukuran 6 m³ pada awalnya memakan total jarak tempuh sekitar 845,6 km perharinya. Penghematan terjadi setelah mengikuti langkah-langkah penyelesaian dengan menggunakan metode *Nearest Neighbour* dan didapat total jarak tempuh menjadi berkurang sekitar 775,6 km per harinya sehingga terjadi penghematan jarak tempuh 70 km atau sekitar 8,3%. Berkurangnya jarak tempuh tersebut terjadi karena ada penambahan rute yang biasanya pengambilan sampah hanya dilakukan di satu TPS menjadi lebih dari satu TPS dengan memperhatikan faktor kendala yaitu kapasitas muatan.

Sampah yang diangkut dari TPS menggunakan truk berkapasitas 6m³ berjumlah 197 m³ perharinya. Pengangkutan sampah tersebut biasanya dilakukan dengan menggunakan 40 armada truk. Setelah menggunakan metode *Nearest Neighbour* terjadi penggabungan sebanyak enam rute, sehingga armada yang digunakan untuk mengangkut sampah yang berjumlah 197 m³ hanya membutuhkan sekitar 34 armada truk. Terdapat penghematan 6 truk untuk mengangkut sampah tersebut atau penghematan sekitar 15%.

Perhitungan waktu proses digunakan asumsi-asumsi yaitu untuk pemuatan sampah 1m³ memakan waktu 30 menit dan untuk proses pengosongan muatan memakan waktu 20 menit. Total waktu proses setiap harinya yang dimulai dari depot menuju TPS lalu

menuju TPA dan kembali lagi ke depot sebelum menggunakan metode *Nearest Neighbour* adalah 9914 menit, namun setelah mengikuti langkah-langkah penyelesaian dalam metode *Nearest Neighbour* maka waktu proses berkurang menjadi 9457 menit, sehingga terjadi penghematan waktu 457 menit atau sekitar 4,6%. Waktu kerja yang disediakan apabila dimanfaatkan, maka dapat digunakan untuk pengangkutan hingga 2 ritasi di beberapa wilayah tertentu yang memiliki jam kerja sedikit, ataupun digunakan untuk melakukan dengan cara menyisir jalan.

Tantangan kedepannya dalam penerapan metode *Nearest Neighbour* yaitu para koordinator lapangan tidak hanya mengetahui permasalahan sampah yang ada di wilayah sendiri, tapi juga harus mengetahui permasalahan sampah yang ada di wilayah lainnya sehingga dengan armada yang tersedia dapat dimanfaatkan sebaik-baiknya. Perubahan rute yang terjadi tentunya juga berdampak pada jarak tempuh yang bertambah dari biasanya.

Perbaikan kedepannya mengenai pengelolaan sampah khususnya dalam hal pengangkutan sampah yaitu dinas kebersihan harus melakukan pertemuan dengan seluruh koordinator lapangan di seluruh kecamatan untuk melakukan evaluasi, sehingga pengelolaan sampah yang biasanya dilakukan per wilayah kini dapat dilakukan secara bersama-sama. Pengelolaan sampah juga harus dilakukan di setiap wilayah kecamatan agar TPA tidak menanggung beban seluruh

sampah yang ada di kota Depok. Pengelolaan dapat dilakukan dengan membangun TPS 3R di seluruh kecamatan bahkan hingga tiap kelurahan, sehingga nantinya dengan pemanfaatan TPS 3R yang lebih baik dapat menghasilkan nilai ekonomi dari sampah. Penambahan jumlah armada perlu dilakukan sehingga dapat mengangkut sampah lebih banyak karena populasi penduduk tiap tahunnya meningkat. Penambahan jumlah armada tersebut tentunya harus disesuaikan dengan kondisi geografis setiap wilayah, karena setiap wilayah memiliki geografis yang berbeda-beda. Penambahan jumlah armada akan berdampak pada penambahan jumlah pekerja baik itu supir dan kernet sehingga harus dilakukan penelitian lebih lanjut.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan dengan mengikuti tahapan dalam metode *Nearest Neighbour* didapatkan hasil berupa jarak tempuh yang sebelumnya adalah 845,6 km berkurang menjadi sekitar 775,6 km sehingga terjadi penghematan jarak tempuh 70 km atau sekitar 8,3%. Berkurangnya jarak tempuh tersebut karena ada enam penggabungan rute truk pengangkut sampah, yaitu TPS 1 dengan 8, TPS 2 dengan 24, TPS 12 dengan 16, TPS 19 dengan 20, TPS 25 dengan 22, dan terakhir TPS 26 dengan 27. Truk yang digunakan untuk mengangkut sampah sebanyak 197 m³ sebelumnya berjumlah 40 armada. Setelah mengikuti

langkah-langkah metode *Nearest Neighbour* maka jumlah armada truk yang dibutuhkan untuk mengangkut sampah sebanyak 197 m³ menjadi sekitar 34 armada sehingga terjadi penghematan sebanyak 6 armada atau sekitar 15%. Waktu proses yang digunakan seluruh armada truk pengangkutan sampah ke seluruh TPS hingga kembali ke depot sebelumnya adalah 9914 menit. Setelah mengikuti langkah-langkah metode *Nearest Neighbour* maka didapatkan pengurangan waktu proses menjadi 9457 menit, sehingga terjadi penghematan waktu sebanyak 457 menit atau sekitar 4,6%.

Pada penelitian selanjutnya sebaiknya DLHK melakukan koordinasi kepada koordinator lapangan agar dalam hal pengelolaan pengangkutan sampah dilakukan secara bersama-sama sehingga pemanfaatan armada yang telah diberikan oleh DLHK dapat maksimal dilakukan. DLHK juga perlu melakukan penelitian lanjutan mengenai pengelolaan pengangkutan sampah yang dilakukan truk dengan cara menyisir jalan atau dari rumah ke rumah akibat tidak tersedianya lahan, sehingga jumlah armada truk dengan timbunan sampah yang ada sesuai agar tidak menimbulkan penumpukan sampah dan terjadi penurunan kualitas lingkungan.

DAFTAR PUSTAKA

- [1] R. Candrakirana, "Penegakan hukum lingkungan dalam bidang pengelolaan

- sampah sebagai perwujudan prinsip good environmental governance di kota Surakarta,” *Jurnal Yustisia*, vol. 4, no. 3, hal. 581 – 601, 2015.
- [2] M. Rizal, “Analisis pengelolaan persampahan perkotaan (studi kasus pada kelurahan Boya kecamatan Banawa kabupaten Donggala),” *Jurnal Sipil Mesin Arsitek Elektro (SMARTek)*, vol. 9, no. 2, hal. 155 – 172, 2011.
- [3] S. Wirasambada dan D. I. Handayani, “Vehicle routing untuk pick up problem dengan pendekatan most valueable neighborhood dan nearest neighbor pada jasa pengiriman barang,” *Jurnal Teknik Waktu*, vol. 14, no. 2, hal. 43 – 46, 2016.
- [4] M. K. Karim, B. D. Setiawan, dan P. P. Adikara, “Optimasi Vehicle Routing Problem With Time Windows (VRPTW) pada rute Mobile Grapari (MOGI) Telkomsel cabang Malang menggunakan algoritme genetika,” *Jurnal Pengembangan Teknologi Informasi dan Komputer*, vol. 2, no. 8, hal. 2702 – 2709, 2018.
- [5] M. Sari, A. Dhoruri, dan E. R. Sari, “Penyelesaian capacitated vehicle routing problem menggunakan saving matriks, sequential insertion, dan nearest neighbour di Victoria Road,” *Jurnal Matematika*, vol. 5, no. 3, hal. 1 – 11, 2016.
- [6] I. K. T. S. Pramatha, I. A. R. Widhiawati, dan Y. Ciawi, “Analisis pengelolaan pengangkutan sampah di kecamatan Klungkung kabupaten Klungkung,” *Jurnal Ilmiah Elektronik Infrastruktur Teknik Sipil*, vol. 2, no. 2, hal. IX-1 – IX-6, 2013.
- [7] N. H. N. Wirum, “Optimasi pembagian tugas karyawan menggunakan metode Hungarian,” *Skripsi*, Universitas Islam Negeri Alauddin, Makassar, 2017.
- [8] DLHK, “Perda kota Depok tahun 2014 tentang pengelolaan sampah,” *Pemerintah Kota Depok*. [Daring]. Tersedia: <http://dlhk.depok.go.id/wp-content/uploads/2018/08/PERDA-KOTA-DEPOK-THN-2014-NO-05-TTG-PENGELOLAAN-SAMPAH.pdf>. [Diakses: 10 Oktober 2019].