

Klasifikasi Kemacetan Lalu Lintas di Buah Batu Menggunakan K - Nearest Neighbor (KNN)

Tugas Akhir

diajukan untuk memenuhi salah satu syarat
memperoleh gelar sarjana
dari Program Studi Teknologi Informatika
Fakultas Informatika Universitas Telkom

1301160781

Rima Rizita Indasari



Program Studi Sarjana Teknologi Informatika
Fakultas Informatika
Universitas Telkom
Bandung

2021

LEMBAR PENGESAHAN

Klasifikasi Kemacetan Lalu Lintas di Buah Batu Menggunakan K - Nearest Neighbor
(KNN)

Classification of Traffic Congestion in Buah Batu Using K - Nearest Neighbor (KNN)

NIM: 1301160781

Rima Rizita Indasari

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknologi Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 18 September 2021

Menyetujui

Pembimbing I

Dr. Putu Harry Gunawan, S.Si.,M.Si.,M.Sc.

NIP: 16860043

Ketua Program Studi

Sarjana Teknologi Informatika,

Dr. Erwin Budi Setiawan, S.Si., M.T.

NIP: 00760045

LEMBAR PERNYATAAN

Dengan ini saya, Rima Rizita Indasari, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul "Klasifikasi Kemacetan Lalu Lintas di Buah Batu Menggunakan K - Nearest Neighbor (KNN)" beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 18 September 2021

Yang Menyatakan,



Rima Rizita Indasari

Klasifikasi Kemacetan Lalu Lintas di Buah Batu Menggunakan K - Nearest Neighbor (KNN)

Rima Rizita Indasari¹, Putu Harry Gunawan²

^{1,2,3} Fakultas Informatika, Universitas Telkom, Bandung

⁴ Divisi Digital Service PT Telekomunikasi Indonesia

¹rimirizita@students.telkomuniversity.ac.id, ²phgunawan1@telkomuniversity.ac.id

Abstrak

Kemacetan merupakan salah satu permasalahan yang ada di jalan selain kecelakaan kendaraan, padatnya jumlah kendaraan, dan set time lampu lalu lintas yang kurang sesuai dengan keadaan yang dapat menghambat lalu lintas. Kemacetan biasanya sering terjadi di sebuah persimpangan. Dalam paper ini mengambil contoh kemacetan yang terjadi di persimpangan Buah Batu, Bandung, Jawa Barat. Kemacetan yang ada di persimpangan Buah Batu tidak hanya terjadi pada jam - jam padat atau jam - jam sibuk seperti pagi hari dan sore hari, pada siang hari dan malam hari kemacetan di persimpangan Buah Batu juga sering terjadi. Penelitian ini menggunakan metode K - Nearest Neighbor (KNN) dengan data dari Dinas Perhubungan (DISHUB) kota Bandung, dimana data yang sudah di dapat akan di olah dan di split sebanyak 40% data test dan 60% data train. Hasil akurasi yang didapatkan sebesar 98% dengan mengambil nilai k terbaik menggunakan metode KNN dengan nilai k = 6.

Kata kunci : kemacetan, persimpangan, KNN

Abstract

Congestion is one of the problems on the road in addition to vehicle accidents, the dense number of vehicles, and the set time of traffic lights that are less by the circumstances that can hamper traffic. Traffic jams usually occur at an intersection. This paper takes the example of congestion that occurs at the intersection of Buah Batu, Bandung, West Java. Congestion at the intersection of Buah Batu does not only occur during the hours - hours or hours - rush hours such as morning and evening, during the day and at night congestion at the intersection of Buah Batu is also common. This research uses the K - Nearest Neighbor (KNN) method with data from the Transportation Department (DISHUB) of Bandung, where the data that has been processed will be processed and split as much as 40% of test data and 60% of data trains. The accuracy result was obtained by 98% by taking the best k value using the KNN method with a value of k = 6.

Keywords: congestion, intersection, KNN

1. Pendahuluan

Keadaan lalu lintas akan berkembang dengan cepat sesuai dengan kondisi kepadatan suatu kota atau daerah yang menandakan bahwa daerah atau kota tersebut maju. Kemacetan lalu lintas merupakan salah satu permasalahan yang selalu di hadapi oleh masyarakat Indonesia, terutama mereka yang tinggal di kota - kota besar di Indonesia, salah satunya kota Jalan Buah Batu, Bandung. Kemacetan yang sering terjadi adalah di setiap persimpangan, dimana biasanya masalah utamanya adalah set time untuk lampu lalu lintas tidak sesuai dengan keadaan jumlah kendaraan di setiap persimpangannya. Hal - hal yang mempengaruhi kemacetan selain durasi atau set time untuk lampu lalu lintas adalah banyaknya kendaraan yang melintas, lebar jalan, dan juga terkadang sikap pengendara juga menjadi salah satu penyebab kemacetan. Maka dari itu sangat dibutuhkan solusi yang tepat untuk mengatasi kemacetan yang terjadi. Pada laporan ini penulis menggunakan metode K - Nearest Neighbor (KNN) untuk mengatasi permasalahan tersebut.

Sebelumnya ada banyak yang sudah melakukan penelitian mengenai kondisi atau keadaan kemacetan lalu lintas. Penelitian [13] membahas tentang "Deteksi Kemacetan Lalu Lintas Jalan Raya Menggunakan Metode Moving Object Detection". Penelitian dilakukan di kota Bali, dan dalam penelitian tersebut mereka menghasilkan klasifikasi untuk kemacetan yang mereka bagi dalam 4 bagian (C, D, E, dan F) menggunakan teknik Moving Object Detection dengan metode background subtraction. Sedangkan peneliti lain juga ada yang melakukan analisa menggunakan metode Recurrent Neural Network (RNN)[11]. RNN adalah salah satu dari beberapa macam arsitektur jaringan pada jaringan syaraf tiruan. Dalam paper [11] RNN cocok untuk digunakan dalam penelitian

mereka, karena arsitektur jaringan syaraf tiruan yaitu koneksi antar neuron belum ditentukan. Hasil untuk pengaturan durasi lampu lalu lintas adaptif dengan skenario pengujian mendapatkan akurasi fix time sebesar 87,191%. Pada penelitian yang lain, ada juga yang menggunakan metode Backpropagation untuk menentukan durasi lampu lalu lintas yang sesuai, dimana dalam paper [1], Backpropagation merupakan salah satu metode pembelajaran dalam jaringan syaraf tiruan dimana proses pertama sebelum ke algoritma Backpropagation adalah proses normalisasi. Metode tersebut dalam pembelajarannya dilakukan dengan penyesuaian bobot – bobot (w) dengan arah mundur berdasarkan nilai error. Dalam hasil penelitian tersebut diperoleh hasil RMSE sebesar 0,0888978841028. Pada penelitian [12], Algoritma Dijkstra digunakan dalam penelitian mereka dengan menentukan bobot terkecil dari masing – masing ruas jalan. Dimana hasil dari penelitian mereka menghasilkan pemilihan rute terbaik sebagai solusi untuk mengurangi kemacetan di Purwokerto.

Dalam penelitian ini penulis mendapatkan data dari Dinas Perhubungan (DISHUB) Kota Bandung dimana masih belum sesuai. Penulis melakukan olah data sehingga didapatkan data – data untuk mendapatkan hasil yang diinginkan. Data yang didapatkan dari mengolah data asli DISHUB Kota Bandung sebanyak 1081 data. Gambar 1 dibawah ini menunjukkan kemacetan yang terjadi di salah satu persimpangan Buah Batu yang mengarah ke Telkom University.



Gambar 1. kemacetan di salah satu persimpangan Buah Batu

Gambar 1 menunjukkan kepadatan kendaraan yang terjadi di salah satu persimpangan yang menyebabkan kendaraan yang seharusnya dapat langsung belok kiri harus terjebak macet dan tidak bisa langsung belok dikarenakan kendaraan di depan yang menutupi sebagian jalan. Gambar tersebut di ambil menggunakan drone pada sore hari.

Batasan masalah yang dialami selama proses penulisan laporan ini adalah olah data yang didapatkan dari DISHUB akan mengalami perubahan dikarenakan class yang ada masih belum sesuai. Data tersebut juga akan ada tahapan preprocessing data, dimana data akan disesuaikan dengan model yang digunakan. Pembersihan data juga diperlukan agar tidak terjadi noise pada data, dan pada penelitian ini dilakukan secara manual, sehingga membutuhkan waktu untuk mendapatkan data.

Tujuan dari penelitian ini adalah menemukan nilai akurasi dari data kemacetan lalu lintas di persimpangan Buah Batu sehingga hasilnya dapat digunakan sebagai pedoman atau dapat menjadi acuan untuk membantu meminimalisir kemacetan yang sering terjadi dengan tepat. Untuk pembahasan selanjutnya dengan terperinci dilanjutkan pada bab berikutnya. Studi Terkait berisi tentang studi atau penelitian yang telah dilakukan sebelumnya dengan tema yang sama. Sistem yang Dibangun berisi tentang rancangan dan sistem yang dihasilkan. Evaluasi berisi tentang hasil dan juga analisa dari hasil yang sudah didapatkan. Kesimpulan adalah bab terakhir yang berisikan kesimpulan dan saran dari penulis untuk penelitian – penelitian selanjutnya dengan tema atau topik yang sama.

2. Studi Terkait

2.1 K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan metode klasifikasi yang populer dan sering digunakan, dimana KNN merupakan metode supervised learning yang pada pengklasifikasiannya menggunakan data dan nilai k yang sudah ditentukan [5][10]. Tahapan dalam KNN sebagai berikut:

1. Menemukan nilai k yang optimal.
2. Menghitung jarak terdekat antar titik dengan metode distance.
3. Didapatkan hasil klasifikasi.

Berikut adalah macam-macam metode distance:

2.1.1 Euclidean Distance

Euclidean Distance adalah yang paling umum digunakan dalam metode k-NN. Euclidean Distance adalah ukuran untuk mencari jarak dari dua titik, dimana Euclidean Distance adalah kasus khusus dari Minkowski dengan $p = 2$ [4][10]. Rumus dari Euclidean Distance[10][7] :

$$d(x_i, y_j) = \sqrt{\sum_{r=1}^n (x_{ir} - y_{jr})^2} \quad (1)$$

dimana:

x_i = sampel data atau data training

y_j = data uji atau data testing

r = atribut individu

$d(x, y)$ = dissimilarity atau jarak

n = jumlah atribut setiap kasus

2.1.2 Minkowski Distance

Minkowski Distance adalah metode untuk menemukan jarak berdasarkan ruang Euclidean [4][10]. Dibawah adalah rumus Minkowski Distance[10][7] :

$$d(x_i, y_j) = \sqrt[p]{\sum_{r=1}^n |x_{ir} - y_{jr}|^p} \quad (2)$$

dimana:

x_i = sampel data atau data training

y_j = data uji atau data testing

r = atribut individu

$d(x, y)$ = dissimilarity atau jarak

n = jumlah atribut setiap kasus

Untuk nilai $p = 1, 2, \dots, \infty$

2.1.3 Cityblock Distance (Manhattan Distance)

Cityblock distance atau Manhattan distance adalah kasus khusus dari Minkowski distance dengan $p = 1$. [4]. Rumus untuk Manhattan Distance adalah [10][7] :

$$d(x_i, y_j) = \sum_{r=1}^n |x_{ir} - y_{jr}| \tag{3}$$

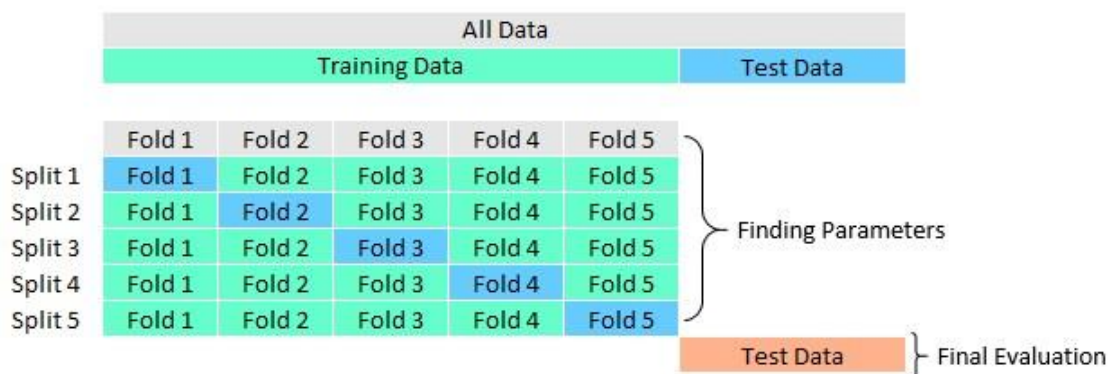
dimana:

- x_i = sampel data atau data training
- y_j = data uji atau data testing
- r = atribut individu
- $d(x, y)$ = dissimilarity atau jarak
- n = jumlah atribut setiap kasus

2.2 Metode Cross Validation

Penggunaan Cross Validation bertujuan untuk mengambil data sample ulang yang digunakan dalam menilai kemampuan evaluasi validitas prediktif dari model dan mencegah overfitting[3]. Dataset yang tersedia untuk memfbangun dan mengevaluasi model prediktif disebut dengan Learning Set. Dataset ini diasumsikan sebagai sampel dari populasi yang diinginkan. Metode random subsampling digunakan untuk menghasilkan Training Set dan Test Set dari Learning Set. Model tersebut kemudian dibangun (atau dilatih) menggunakan Training Set dan diuji pada Test Set. Berbagai metode random subsampling berbeda sehubungan dengan bagaimana training dan test set dihasilkan [2].

Pada metode random subsampling salah satunya adalah cross-validation dimana mirip dengan random subsampling berulang yang pengambilan sampelnya dilakukan sedemikian rupa agar tidak ada dua test set yang saling tumpang tindih. Dalam cross-validation, learning set yang tersedia dipartisi menjadi k subset yang terpisah dengan ukuran yang rata-ratanya sama. Cross-validation sering kali melibatkan stratified random sampling (pengambilan sampel acak berlapis), yang berarti bahwa pengambilan sampel dilakukan sedemikian rupa sehingga proporsi kelas dalam subset individu mencerminkan proporsi dalam learning set.



Gambar 2. Proses Cross Validation

Gambar 2 adalah contoh dari 5-fold cross-validation dimana melakukan training dengan 5 tahapan, yaitu:
 Percobaan pertama, menjadikan bagian dari partisi pertama dari keseluruhan dataset menjadi data test dan 4 bagian lainnya sebagai data train.
 Percobaan kedua, menjadikan bagian dari partisi kedua dari keseluruhan dataset menjadi data test dan 4 bagian lainnya sebagai data train.
 Percobaan ketiga, menjadikan bagian dari partisi ketiga dari keseluruhan dataset menjadi data test dan 4 bagian lainnya sebagai data train, dan begitu seterusnya.

2.3 Data

Data berasal dari Dinas Perhubungan Kota Bandung. Dimana data yang didapatkan sebelumnya harus diolah terlebih dahulu sehingga mendapatkan data yang benar – benar dibutuhkan dan sesuai dengan model yang digunakan. Data yang sudah didapatkan dikumpulkan dalam file excel atau .xlsx seperti pada gambar 3 dan gambar 4.

1	waktu	posisi	volume kendaraan lolos	kapaistas jalan	derajat kejenuhan	red detik	level of service
2	07.00-08.00	timur	964	2671	0,360913516	286	B
3		barat	689	2671	0,257955822	286	B
4		utara	1592	2514	0,633253779	286	C
5		selatan	1581	943	1,676564157	286	F
6	07.15-08.15	timur	1030	2671	0,385623362	286	B
7		barat	664	2671	0,248596031	286	B
8		utara	1894	2514	0,753381066	286	D
9	07.30-08.30	selatan	1626	943	1,724284199	286	F
10		timur	992	2671	0,371396481	286	B
11		barat	612	2671	0,229127668	286	B
12	07.45-08.45	utara	2070	2514	0,823389021	286	D
13		selatan	1589	943	1,68504772	286	F
14		timur	941	2671	0,352302508	286	B
15	08.00-09.00	barat	548	2671	0,205166604	286	B
16		utara	2067	2514	0,822195704	286	D
17		selatan	1530	943	1,622481442	286	F
18	08.00-09.00	timur	893	2671	0,334331711	286	B
19		barat	472	2671	0,176712842	286	B
20		utara	1714	2514	0,681782021	286	C
21		selatan	1527	943	1,619300106	286	F
22	12.00-13.00	timur	598	2671	0,223886185	286	B
23		barat	293	2671	0,109696743	286	A
24		utara	1054	2514	0,419252188	286	B
25		selatan	1124	943	1,191940615	286	F

Gambar 3. gambar data bersih

1	waktu	posisi	volume_kendaraan_lolos	kapaistas_jalan	derajat_kejenuhan	red_detik	level_of_service
2	07:00-08:00	Timur	964	2617	0,368360718	286	0
3	07:00-08:00	Barat	689	2671	0,257955822	286	0
4	07:00-08:00	Utara	1592	2514	0,633253779	286	0
5	07:00-08:00	Selatan	1581	943	1,676564157	286	1
6	07:15-08:15	Timur	1030	2671	0,385623362	286	0
7	07:15-08:15	Barat	664	2671	0,248596031	286	0
8	07:15-08:15	Utara	1894	2514	0,753381066	286	1
9	07:15-08:15	Selatan	1626	943	1,724284199	286	1
10	07:30-08:30	Timur	992	2671	0,371396481	286	0
11	07:30-08:30	Barat	612	2671	0,229127668	286	0
12	07:30-08:30	Utara	2070	2514	0,823389021	286	1
13	07:30-08:30	Selatan	1589	943	1,68504772	286	1
14	07:45-08:45	Timur	941	2671	0,352302508	286	0
15	07:45-08:45	Barat	548	2671	0,205166604	286	0
16	07:45-08:45	Utara	2067	2514	0,822195704	286	1
17	07:45-08:45	Selatan	1530	943	1,622481442	286	1
18	08:00-09:00	Timur	893	2671	0,334331711	286	0
19	08:00-09:00	Barat	472	2671	0,176712842	286	0
20	08:00-09:00	Utara	1714	2514	0,681782021	286	0
21	08:00-09:00	Selatan	1527	943	1,619300106	286	1
22	12:00-13:00	Timur	598	2671	0,223886185	286	0
23	12:00-13:00	Barat	293	2671	0,109696743	286	0

Gambar 4. gambar data yang diolah

Dalam gambar 3 adalah data bersih yang didapatkan dari DISHUB sedangkan gambar 4 merupakan data yang telah disesuaikan dengan model yang digunakan. Penjelasan class dari kedua gambar adalah, waktu menunjukkan dimana setiap 15 menit sampai satu jam dilakukan perhitungan. Posisi adalah lokasi atau arah dari persimpangan Buah Batu, seperti dari arah Utara, Selatan, Timur, dan Barat. Volume kendaraan lolos menjelaskan tentang jumlah kendaraan yang lolos pada saat lampu hijau disetiap Posisi. Kapasitas jalan adalah jumlah total kendaraan yang bisa ditampung oleh jalan dimana dalam menentukan keadaan suatu jalan tersebut macet atau tidaknya. Untuk rumus dari kapasitas jalan itu sendiri terdapat dalam data asli dari DISHUB Kota Bandung. Red detik adalah lamanya lampu merah di persimpangan dalam satuan detik. Derajat kejenuhan merupakan penentu apakah keadaan lalu lintas dalam keadaan macet atau tidak sesuai dengan aturan kemacetan dari DISHUB Kota Bandung, dimana untuk menentukannya menggunakan rumus

$$dk = \frac{v}{kj} \quad (4)$$

dimana:

dk = Derajat Kejenuhan

v = Volume Kendaraan Lolos

kj = Kapasitas Jalan

2.4 Tingkatan Kemacetan

Dalam mengatur lalu lintas, menteri perhubungan memiliki beberapa aturan yang tertulis dalam "Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 96 Tahun 2015". Salah satu aturan, dimana tingkat kemacetan lalu lintas terdapat pada "Penetapan Tingkat Pelayanan Yang Diinginkan" dengan dibedakan dalam poin - poin sebagai berikut:

1. Tingkat pelayanan A
Kepadatan lalu lintas sangat rendah dengan kondisi tundaan kurang dari 5 detik perkendaraan dengan kecepatan kurang dari 80km/jam. Untuk derajat kejenuhan antara 0.01 – 0.20.
2. Tingkat pelayanan B
Kepadatan lalu lintas rendah hambatan internal lalu lintas belum mempengaruhi kecepatan dengan kondisi tundaan lebih dari 5 sampai 15 detik perkendaraan dengan kecepatan kurang dari 70km/jam. Untuk derajat kejenuhan antara 0.20 – 0.45.
3. Tingkat pelayanan C
Kepadatan lalu lintas sedang karena hambatan internal lalu lintas meningkat dengan kondisi tundaan lebih dari 15 sampai 25 detik perkendaraan dengan kecepatan kurang dari 60km/jam. Untuk derajat kejenuhan antara 0.45 – 0.75.
4. Tingkat pelayanan D
Kepadatan lalu lintas sedang namun volume lalu lintas menyebabkan penurunan kecepatan, tapi masih ditolerir namun sangat terpengaruh oleh perubahan arus dengan kondisi tundaan lebih dari 25 sampai 40 detik perkendaraan dengan kecepatan kurang dari 50km/jam. Untuk derajat kejenuhan antara 0.75 – 0.90.
5. Tingkat pelayanan E
Kepadatan lalu lintas tinggi karena hambatan internal lalu lintas tinggi dengan kondisi tundaan lebih dari 40 sampai 60 detik perkendaraan dengan kecepatan kurang dari 30km/jam. Untuk derajat kejenuhan antara 0.90 – 1.
6. Tingkat pelayanan F
Kepadatan lalu lintas sangat tinggi dan volume rendah serta terjadi kemacetan untuk durasi yang cukup lama dengan kondisi tundaan lebih dari 60 detik perkendaraan dengan kecepatan dibawah 30km/jam. Untuk derajat kejenuhan lebih dari 1.

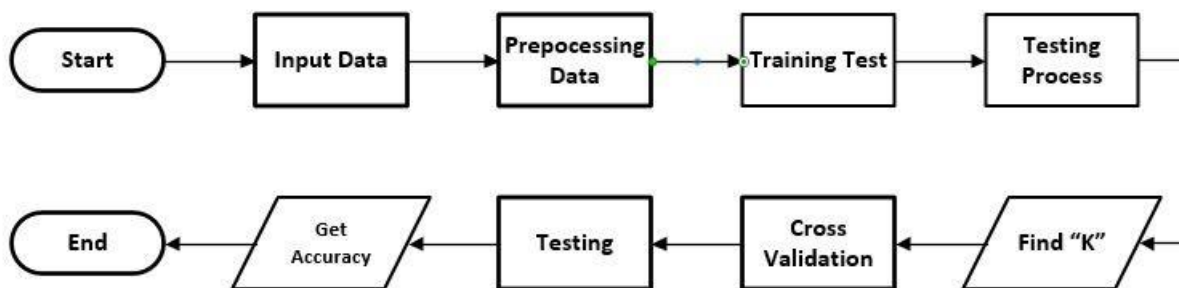
Tingkatan kemacetan diatas digunakan untuk menggolongkan hasil dari derajat kejenuhan ke dalam kategori macet atau tidak pada class level of service, dimana semakin mendekati 1 maka keadaan jalan pada saat itu dianggap sedang dalam keadaan macet. Apabila keadaan suatu lalu lintas telah mencapai kondisi D - F maka harus dilakukan perubahan atau menemukan solusi yang terbaik agar keadaan berubah menjadi lebih baik dari kondisi sebelumnya. Pada penelitian ini, penulis menggunakan binary untuk mengklasifikasikan kemacetan sesuai aturan yang sudah tersedia dari DISHUB. Karena menggunakan binary, maka penjabarannya seperti pada gambar 5

NO	BINARY	
	0	1
1	A	D
2	B	E
3	C	F

Gambar 5. gambar klasifikasi

Pada gambar 5 adalah penjabaran dari level of service pada gambar 3 untuk tabel level of service dimana nantinya tabel level of service akan berubah menjadi binary sesuai dengan aturan pada gambar 5 seperti pada gambar 4 yang akan digunakan pada data dalam pengujian ini. Pada binary 0 dimana menandakan bahwa keadaan jalan pada saat itu dalam keadaan tidak macet atau lancar. Sedangkan binary 1 yang menandakan bahwa keadaan jalan pada saat itu adalah dalam kondisi macet.

3. Sistem yang Dibangun



Gambar 6. flowchart sistem

Berikut penjelasannya:

Input Data

Setelah melakukan olah data, maka akan dilakukan tahap – tahapan selanjutnya berdasarkan gambar 6. Langkah pertama yang dilakukan adalah input data terlebih dahulu, dimana data dipanggil dari github agar lebih mudah dalam pencarian data apabila kita lupa meletakkan data dimana. Setelah data di input, data akan diolah kembali agar data sesuai dengan model yang digunakan dan menghilangkan data yang tidak diperlukan.

Preprocessing Data

	waktu	posisi	derajat_kejenuhan	level_of_service
0	0	2	0.368361	0
1	0	0	0.257956	0
2	0	3	0.633254	0
3	0	1	1.676564	1
4	1	2	0.385623	0

Gambar 7. hasil processing data

Data yang digunakan pada tahap preprocessing data adalah data pada gambar 4. Data tersebut selanjutnya mengalami perubahan setelah melalui proses penghilangan class yang tidak diperlukan untuk proses berikutnya, seperti pada gambar 7, dimana class volume kendaraan lolos dan class kapasitas jalan yang ada pada data sebelumnya, pada gambar 4, di hapus karena telah diwakilkan oleh class derajat kejenuhan dimana merupakan perbandingan antara dua class yang dihapus sesuai dengan rumus 4. Selain menghapus data yang tidak diperlukan, juga dibutuhkan merubah class atau n-coding waktu dan juga class "posisi" agar terlihat rapi dan berbentuk integer agar mudah di baca, seperti pada class "waktu" pada pukul 07:00 – 08:00 dimisalkan 0 sedangkan pukul 07:15 – 08:15 dimisalkan 1 begitu seterusnya, kemudian untuk class "posisi" dimisalkan 0 – 3 berurutan sesuai dengan abjad. Derajat kejenuhan diisi sesuai dengan rumus 3. Waktu dan posisi masih tetap dipertahankan dan tidak dihapus karena waktu dan posisi juga masih menentukan class level of service, dimana waktu dan posisi yang berbeda bisa menentukan macet atau tidaknya pada ruas jalan tersebut. Untuk level of service class menggunakan binary dimana penjelasannya sudah dijelaskan pada Bab 2.3 Tingkatan Kemacetan.

Data Test dan Data Training

Langkah selanjutnya setelah mendapatkan data yang sesuai adalah tahap split data yang menghasilkan data training sebesar 60% dan data test sebesar 40% dimana selanjutnya akan ada trining data dan testing process untuk menemukan nilai k terbaik. Setelah menemukan nilai k selanjutnya tahap testing untuk menghasilkan confusion matrix dimana terdapat nilai akurasi sebagai hasil akhir yang dibutuhkan dalam laporan ini. Confusion matrix merupakan salah satu metode yang paling umum untuk menyajikan hasil yang diperoleh oleh pengklasifikasi[8].

a. Sensitivity Recall

Recall adalah rasio prediksi benar positif (TP) yang dibandingkan dengan keseluruhan data yang benar positif (Actual Class, Positive) dan hasilnya 0.9137931034 atau 0.91 dengan rumus 5 :

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

b. Precision

Precision adalah rasio prediksi benar positif (TP) yang dibandingkan dengan keseluruhan hasil yang diprediksi positif (Predict Class, Positive) dan hasilnya 1 atau 1.00 dengan rumus 6 :

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

c. Accuracy

Accuracy adalah perbandingan antara total data yang diklasifikasikan benar dengan keseluruhan data dan hasilnya 0.9768518518518519 atau 0.98 atau 98% dengan rumus 7 :

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+TN+FP} \quad (7)$$

d. F_1 - Score

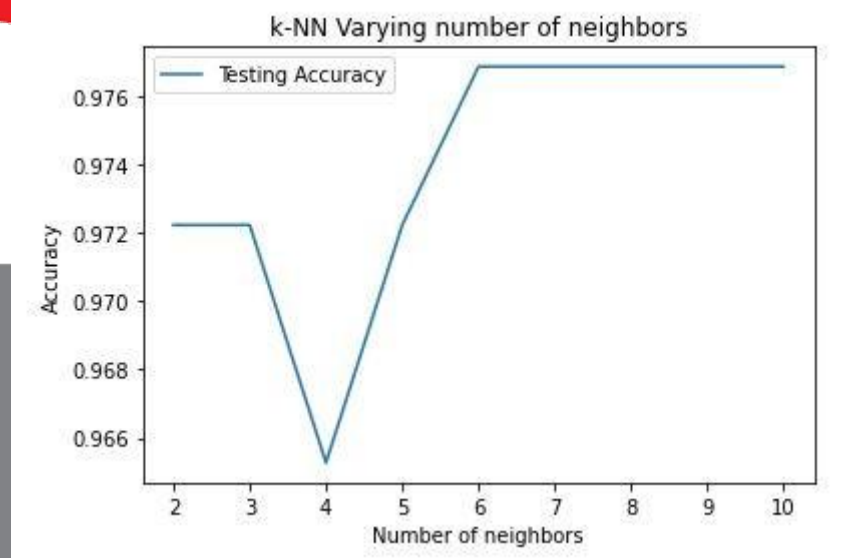
F_1 Score adalah perbandingan rata – rata precision dan recall yang dibobotkan dan hasilnya 0.954954955 atau 0.95 dengan rumus 8 :

$$F1score = 2 \frac{(PRC)(SNS)}{PRC + SNS} \quad (8)$$

4. Evaluasi

4.1 Hasil Nilai k

Berdasarkan model yang telah dibentuk didapatkan hasil nilai akurasi dengan berbagai kemungkinan nilai k seperti pada gambar 8. Akurasi yang didapat rata – rata diatas 90%. Data yang digunakan agar mendapatkan grafik untuk nilai k terbaik adalah data pada gambar 7 dimana selanjutnya data akan melalui tahap split data sebanyak 40% untuk data test dan sisanya 60% untuk data train.

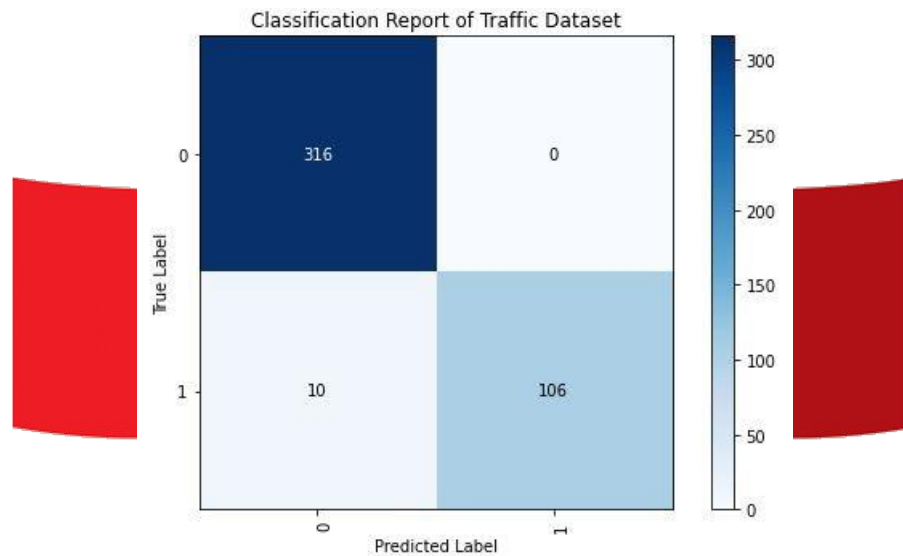


Gambar 8. kurva atau graf menentukan nilai k

Dalam paper ini untuk nilai k yang digunakan adalah k = 6 yang didapatkan berdasarkan dari grafik 8 sehingga menghasilkan nilai akurasi yang tinggi. Pada nilai k = 7, k = 8, k = 9, dan k = 10 memiliki hasil akurasi yang sama dengan nilai k = 6, namun pada penelitian ini menggunakan k = 6 karena hasil akurasi yang di dapat lebih tinggi daripada angka sebelumnya dan juga merupakan angka terkecil yang menghasilkan akurasi tertinggi di awal sebelum angka selanjutnya, yaitu angka 7, 8, 9, dan seterusnya. Hasil akurasi yang didapatkan setelah memilih k adalah sebesar 0.9768518518518519 atau 98%.

4.2 Hasil Confusion Matrix

Hasil dari confusion matrix digambarkan seperti pada gambar 9



Gambar 9. Confusion Matrix

Hasil dan penjelasan dari gambar 9 [8][9][6] Confusion Matrix dengan k = 6 dalam paper ini seperti dibawah ini:

1. Berdasarkan gambar 9, True Negative yang ditandai hasil Prediksi (Predicted Label) dengan nilai sebenarnya (True Label) sebanyak 316 variabel. Dan pada False Positive yang didapatkan sebanyak 0 variabel.
2. Berdasarkan gambar 9, True Positive yang ditandai hasil Prediksi (Predicted Label) dengan nilai sebenarnya (True Label) sebanyak 106 variabel. Dan pada False Negative yang didapatkan sebanyak 10 variabel.

Menggunakan hasil diatas, dapat dihitung nilai Sensitivity (Recall), Precision dan F1 score dengan cara sebagai berikut:

- Sensitivity Recall

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{106}{106+10} = 0.9137931$$

$$\% \text{Recall} = 91,37\%$$

- Precision

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{106}{106+0} = 1.00$$

$$\% \text{Precision} = 100\%$$

- Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+TN+FP} = \frac{106+316}{106+10+316+0} = 0.976851$$

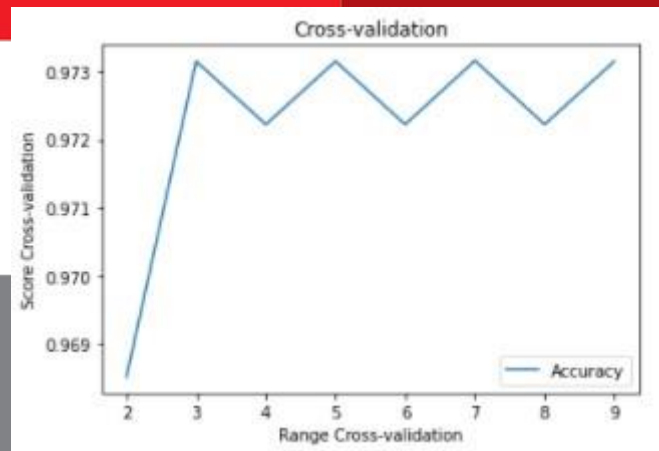
$$\% \text{Accuracy} = 97,68\%$$

- F1 Score

$$F1Score = 2 \frac{(PRC)(SNS)}{PRC + SNS} = 2 \frac{1.00 \cdot 0.9137931}{1.00 + 0.9137931} = 0.954954955$$

%F1Score = 95,49%

4.3 Hasil dari Training menggunakan Cross Validation (CV)



Gambar 10. Hasil Cross-validation

Dalam pengujian k-Nearest Neighbors, didapatkanlah sebuah hasil bahwa nilai cross-validation yang terbaik pada nilai 5. Pengujian dilakukan dari cross-validation nilai 2-10 seperti pada gambar 10. Dengan nilai accuracy dari setiap cross-validation sebagai berikut:

Proses CV pertama, dengan partisi pertama sebagai test data, dan 4 partisi lainnya sebagai training data, menghasilkan akurasi 0.97685185.

Proses CV kedua, dengan partisi kedua sebagai test data, dan 4 partisi lainnya sebagai training data, menghasilkan akurasi 0.9537037.

Proses CV ketiga, dengan partisi ketiga sebagai test data, dan 4 partisi lainnya sebagai training data, menghasilkan akurasi 0.98611111.

Proses CV keempat, dengan partisi ketiga sebagai test data, dan 4 partisi lainnya sebagai training data, menghasilkan akurasi 0.96759259.

Proses CV kelima, dengan partisi ketiga sebagai test data, dan 4 partisi lainnya sebagai training data, menghasilkan akurasi 0.98148148.

Pada penjelasan, proses CV hanya di ambil sebanyak 5 CV karena pada gambar 10 grafik paling bagus pada saat CV 3, 5, 7, dan 9, untuk data paling bagus adalah CV dari 3 sampai 6, maka diambil tengah - tengah yaitu CV = 5. Berdasarkan kelima proses CV yang sudah dilakukan, didapatkan nilai rata-ratanya adalah 0.9731481481481483. Maka performa yang didapatkan tidak jauh berbeda dari menggunakan cross-validation atau tidak.

5. Kesimpulan

Pada penelitian ini bertujuan untuk menemukan nilai akurasi di persimpangan Buah Batu agar dapat digunakan sebagai pedoman atau dapat menjadi acuan untuk membantu meminimalisir kemacetan yang sering terjadi. Padatnya jumlah kendaraan, pengaturan lampu lalu lintas yang belum maksimal, dan permasalahan yang lainnya mengakibatkan kemacetan panjang. Untuk mengatasi permasalahan tersebut, dilakukanlah penelitian ini dimana metode yang digunakan adalah K-Nearest Neighbor (KNN). Berdasarkan hasil dari penelitian ini yang menggunakan metode KNN, didapatkan hasil akurasi yang cukup tinggi dimana dalam laporan ini menggunakan nilai k yang terbaik, $k = 6$, yaitu sebesar 0.98 atau 98%. Hasil dari Confusion Matrix yang lainnya selain akurasi adalah Sensivity (Recall) sebesar 0.91 atau 91%, Precision sebesar 1.00 atau 100%, dan F1 Score sebesar 0.95 atau 95%. Dengan hasil akurasi yang didapatkan bernilai besar, maka diharapkan model klasifikasi pada penelitian ini dapat menjadi acuan untuk membantu mengatasi kemacetan lalu lintas di persimpangan Buah Batu dan juga di tempat yang lainnya. Saran untuk penelitian selanjutnya dengan topik yang sama dan menggunakan metode KNN maupun menggunakan metode yang lain adalah diharapkan lebih memperhatikan data, karena penulis juga melakukan wawancara agar data yang didapatkan masih sesuai dengan data yang diberikan dari DISHUB yang juga membutuhkan waktu, dimana terkadang pihak DISHUB tidak bisa dihubungi atau sibuk sehingga menghambat proses pembetulan data. Pada confusion matrix perbaikan yang lainnya juga bisa dilakukan dengan augmentasi pada data dan digabungkan dengan algoritma Machine Learning lain contohnya Convolutional Neural Network (CNN) agar keakuratan (Accuracy) bisa mencapai nilai yang lebih tinggi.



Daftar Pustaka

- [1] S. A. Azizy. Penentuan Durasi Nyala Lampu Lalu Lintas Berdasarkan Panjang Antrian Kendaraan Menggunakan Metode Backpropagation. PhD thesis, Universitas Brawijaya, 2018.
- [2] D. Berrar. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology*, 1:542–545, 2018.
- [3] M. W. Browne. Cross-validation methods. *Journal of Mathematical Psychology*, 4(1):108–132, 2000.
- [4] K. Chomboon, P. Chujai, P. Teerarassamee, K. Kerdprasop, and N. Kerdprasop. An empirical study of distance metrics for k-nearest neighbor algorithm. In *Proceedings of the 3rd international conference on industrial application engineering*, pages 280–285, 2015.
- [5] Y. I. Claudy, R. S. Perdana, and M. A. Fauzi. Klasifikasi dokumen twitter untuk mengetahui karakter calon karyawan menggunakan algoritme k-nearest neighbor (knn). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN, 2548:964X, 2018.
- [6] A. F. Hidayatullah, A. A. Fadila, K. P. Juwairi, and R. A. Nayoan. Identifikasi konten kasar pada tweet bahasa indonesia. *Jurnal Linguistik Komputasional*, 2(1):1–5, 2019.
- [7] Y. I. Kurniawan and T. I. Barokah. Klasifikasi penentuan pengajuan kartu kredit menggunakan k-nearest neighbor. *Jurnal Ilmiah Matrik*, 22(1):73–82, 2020.
- [8] A. Luque, A. Carrasco, A. Martín, and A. de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.
- [9] M. Maulidah, W. Gata, R. Aulianita, and C. I. Agustyaningrum. Algoritma klasifikasi decision tree untuk rekomendasi buku berdasarkan kategori buku. *E-Bisnis: Jurnal Ilmiah Ekonomi dan Bisnis*, 13(2):89–96, 2020.
- [10] S. A. Medjahed, T. A. Saadi, and A. Benyettou. Breast cancer diagnosis by using k-nearest neighbor with different distances and classification rules. *International Journal of Computer Applications*, 62(1), 2013.
- [11] R. A. Pramudya, M. Imrona, and F. N. Indonesia. Perancangan pengaturan durasi lampu lalu lintas adaptif. In *Indonesia Symposium On Computing 2015*, pages 175–180, 2015.
- [12] U. M. Rifanti. Pemilihan rute terbaik menggunakan algoritma dijkstra untuk mengurangi kemacetan lalu lintas di purwokerto. *JMPM: Jurnal Matematika dan Pendidikan Matematika*, 2(2):90–99, 2017.
- [13] I. G. A. Wibawa, L. Arida, and A. Rahning. Deteksi kemacetan lalu lintas jalan raya menggunakan metode moving object detection. no. November, pages 2–4, 2016.

Lampiran

Lampiran dapat berupa detil data dan contoh lebih lengkapnya, data-data pendukung, detail hasil pengujian, analisis hasil pengujian, detail hasil survey, surat pernyataan dari tempat studi kasus, screenshot tampilan sistem, hasil kuesioner dan lain-lain.

