

Prediksi Kemacetan Lalu Lintas Jalur Armada Bus Trans Metro Bandung Berbasis *Machine Learning* dan *Internet of Things*

1st Muhammad Buushiri
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

muhammadbuushiri@student.telkomuniversity.ac.id

2nd Rendy Munadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rendymunadi@telkomuniversity.ac.id

3rd Sussi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

sussiss@telkomuniversity.ac.id

Abstrak— Kemacetan adalah suatu hal yang tidak dapat dihindari dalam kehidupan sehari-hari bagi setiap orang khususnya untuk kota Bandung. Sistem monitoring untuk memantau kondisi lalu lintas di kota Bandung khususnya di jalur armada bus Trans Metro Bandung dapat menjadi solusi dalam mengambil tindakan cepat oleh pihak yang berwajib ketika ada kemungkinan terjadi kemacetan. Selain itu, calon penumpang juga dapat mengetahui kondisi lalu lintas sepanjang jalur yang dilewati oleh bus Trans Metro Bandung pada jam dan hari-hari tertentu. Dalam penelitian tugas akhir ini bertujuan untuk membuat sistem monitoring kemacetan dengan memanfaatkan speed bus Trans Metro Bandung ketika sedang beroperasi untuk dijadikan sebagai indikasi kemacetan. Speed bus diperoleh dari GPS dengan Raspberry Pi 3B sebagai Microcomputer yang telah terpasang pada bus dan diproses ke model machine learning untuk mengklasifikasikan kemacetan berdasarkan historical data dari GPS. Adapun model Machine Learning yang digunakan adalah Decision Tree dan Random Forest. Pengujian hasil prediksi pada masing-masing model dilakukan dengan menggunakan confusion matrix dengan mengukur parameter Precision, Recall, F1-Score, dan Accuracy. Penentuan nilai Max Depth berdasarkan data train dan data test juga dilakukan untuk mendapatkan performansi terbaik dan melihat model apakah mengalami overfitting, underfitting atau tidak. Didapatkan Decision Tree sebagai model dengan performansi terbaik dengan nilai Accuracy sebesar 96.8% pada split data 80% data train dan 20% data test.

Kata kunci— Bus Rapid Transit, Internet of Things, Machine Learning, Decision Tree, Random Forest, Trans Metro Bandung.

I. PENDAHULUAN

Padatnya jumlah penduduk tersebut tentu bukan merupakan satu-satunya penyebab dari tingginya tingkat kemacetan. Masih banyaknya masyarakat yang masih memilih menggunakan kendaraan pribadi ketimbang menggunakan moda transportasi umum juga menjadi penyebab utama dari tingginya tingkat kemacetan di Bandung. Salah satu upaya yang dilakukan oleh Pemerintah Kota Bandung untuk mengatasi kemacetan adalah dengan memberdayakan transportasi umum yaitu bus Trans Metro Bandung (TMB). Seiring dengan berjalannya waktu, permasalahan dan hambatan pun mulai dirasakan. Jaringan jalan yang tersedia kurang memungkinkan TMB memiliki

jalur khusus yang terpisah dari lalu lintas sehingga ketika terjadi kemacetan bus TMB juga merasakan dampaknya.

Beberapa penelitian telah dilakukan sebelumnya untuk mengatasi permasalahan kemacetan tersebut seperti pada jurnal [4] memprediksi kemacetan berdasarkan kecepatan rata-rata kendaraan yang diperoleh menggunakan sensor yang dipasang hanya di persimpangan jalan sehingga ketika terjadi kemacetan di sepanjang jalan lainnya maka tidak dapat terdeteksi.

Pada jurnal [6] deteksi kemacetan dari gambar kamera menggunakan Deep Convolution Neural Networks. Namun, model ini hanya melakukan deteksi kemacetan saja, tingkat kemacetan dan deteksi *speed* tidak ditentukan. Selain itu, komputasi berat jika model diimplementasikan juga salah satu kekurangan dari model sistem ini.

Pada jurnal [7] deteksi kondisi jalan di Italia menggunakan Twitter Stream Analysis. Namun permasalahan yang dihadapi dengan sistem model ini adalah mengingat jaranganya dan keragaman data dari Twitter terutama yang menyangkut hal kemacetan jalan. Oleh karena itu.

Pada penelitian ini bertujuan untuk memprediksi kemacetan pada sepanjang jalur yang dilalui oleh bus TMB dengan mengacu kepada kecepatan dari bus TMB ketika melewati jalur tersebut sehingga kemacetan bisa dideteksi tidak hanya pada persimpangan jalan saja tapi sepanjang jalan yang dilalui oleh Bus Trans Metro Bandung. Untuk mendapatkan data kecepatan Bus TMB tersebut, maka dipasanglah GPS Breakout V3 Module pada Bus TMB yang dihubungkan dengan Raspberry Pi 3B sebagai *Microcomputer*. Dari data kecepatan bus tersebut akan dilakukan prediksi dengan menggunakan model Decision Tree dan Random Forest. Kemudian kedua model tersebut akan dilakukan pengujian untuk mengetahui model manakah yang memiliki performansi terbaik jika menggunakan data kemacetan yang telah dikumpulkan.

II. KAJIAN TEORI

A. Internet of Things

Internet of Things (IoT) adalah suatu konsep yang memungkinkan perangkat elektronik dengan perangkat fisik seperti sensor dapat saling berkomunikasi satu sama lain melalui koneksi internet untuk menemukan solusi terhadap permasalahan disekitar [8]. IoT menjadi paradigma terciptanya gaya hidup yang lebih modern seperti adanya *smart transportation*, *smart city*, *smart farming* dan lain-lain.

B. Raspberry Pi 3 Model B

Raspberry Pi adalah komputer papan tunggal yang dapat digunakan pengguna untuk menjalankan dan mengembangkan script/program menggunakan bahasa pemrograman python, desain animasi, game atau video [9]. Kelebihan Raspberry Pi adalah dapat berfungsi sebagai layaknya komputer, namun memiliki daya yang kecil dan relatif murah. Selain itu, Raspberry Pi dapat dihubungkan dengan internet karena mempunyai sambungan LAN yang dapat digabungkan dengan router menjadi jaringan nirkabel yang dapat diakses oleh beberapa perangkat dengan akses internet yang didapatkan melalui modem dan wirelessrouter. Dengan begitu Raspberry Pi dapat dihubungkan juga dengan web server sehingga dapat dikendalikan dari jarak jauh secara remote [10].



Gambar 1(B) Raspberry Pi 3 Model B

C. Machine Learning

Machine Learning adalah suatu cabang keilmuan kecerdasan buatan (Artificial Intelligence) yang mengembangkan mesin untuk dapat belajar sendirinya untuk memperoleh, mempelajari, dan menganalisa data dalam melakukan tugas tertentu seperti memprediksi masa depan dan mengklasifikasikan sesuatu ke dalam berbagai kelas/kategori. Machine Learning sering digunakan untuk memecahkan berbagai masalah dalam berbagai bidang seperti lalu lintas, industri, kesehatan, teknologi, dan lain-lain [11]. Metode pada Machine Learning dikategorikan menjadi tiga, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*.

D. Global Positioning System (GPS)

Global Positioning System (GPS) adalah suatu sistem navigasi berbasis satelit yang dapat memberikan informasi

tentang suatu posisi, rute yang dilewati dan kecepatan kendaraan tersebut secara *Real-Time* [12]. Selain itu pengguna juga dapat mengetahui kondisi lalu lintas terkini dengan menggunakan sistem navigasi tersebut. Posisi kendaraan dapat diketahui dengan mendapatkan data dari Satelit GPS yang berupa titik koordinat (*longitude* dan *latitude*) serta ketinggian. Kapanpun dan dimanapun pengguna dapat menggunakan GPS ini karena sistem navigasi ini tersedia diseluruh penjuru permukaan bumi dan bekerja 24 jam dalam sehari.

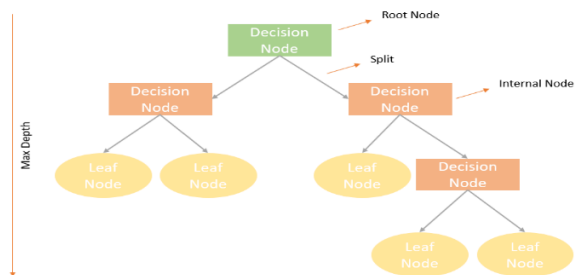
E. Python

Python merupakan bahasa pemrograman yang open-source dan berorientasi pada objek. Saat ini Python menjadi salah satu bahasa pemrograman yang populer karena mudah dimengerti sehingga cocok untuk pemula. Python menawarkan sejumlah besar library yang berguna untuk scientific computing dan Machine Learning. Oleh karena itu dengan Python program kode-kode dan fungsi yang kompleks bisa menjadi mudah [13].

F. Decision Tree

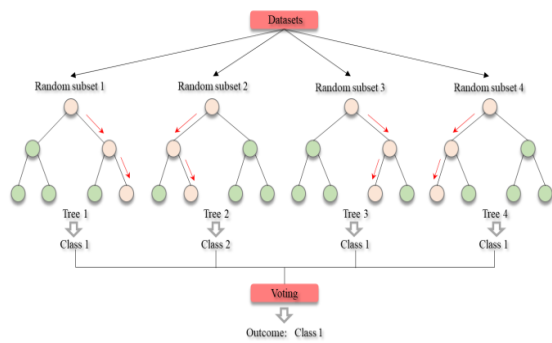
Decision Tree merupakan salah satu algoritma klasifikasi yang sering digunakan karena kemudahan dalam eksekusi dan mudah untuk dimengerti jika dibandingkan dengan algoritma klasifikasi yang lainnya [14]. Decision tree selain dapat mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simple, juga dapat memadukan eksplorasi data dan pemodelan sehingga sangat baik sebagai langkah awal dalam proses pemodelan [15]. Belakangan ini Decision tree telah diterapkan dalam mengatasi masalah-masalah seperti memprediksi penyakit pasien, memprediksi lalu lintas, prediksi kepuasan pelanggan, dll.

Gambar 2(F) Sistem Kerja Decision Tree



G. Random Forest

Random Forest adalah algoritma yang dapat digunakan untuk pengklasifikasian dan regresi pada dataset dengan jumlah yang besar. Metode ini merupakan kumpulan metode Decision Tree sebagai *base classifier* yang dibangun dan dikombinasikan. *Bootstrap sampling* untuk dilakukan untuk membangun pohon prediksi dengan predictor acak pada tiap-tiap pohon keputusan. Dengan cara *Majority Vote*, Random Forest melakukan prediksi dengan mengkombinasikan hasil dari setiap pohon keputusan untuk klasifikasi dan rata-rata regresi [16].



Gambar 3(G) Sistem Kerja Random Forest [17]

H. Confusion Matrix

Confusion Matrix atau *Error Matrix* adalah tabel yang digunakan untuk mengukur kinerja model Machine Learning untuk masalah klasifikasi dengan menguji prediksi jumlah yang benar dan salah dengan perhitungan pada satu set tes yang nilainya sebenarnya diketahui. Dengan Confusion Matrix tidak hanya dapat mengetahui kesalahan yang dibuat oleh *Classifier* tetapi yang lebih penting adalah kesalahan yang sedang dibuat [18]. Pada pengukuran performa model Machine Learning dengan menggunakan Confusion Matrix dikenal beberapa istilah, diantaranya sebagai berikut:

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Gambar 4(H) Confusion Matrix [15]

Keterangan:

- **True Positive (TP):** total data positif yang diprediksi dengan benar oleh sistem
- **True Negative (TN):** total data negatif yang diprediksi dengan benar oleh sistem
- **False Negative (FN):** total data positif yang terdeteksi sebagai data negatif oleh sistem
- **False Positive (FP):** total data negatif yang terdeteksi sebagai data positif oleh sistem.

Untuk dapat mengetahui performansi model pada klasifikasi yaitu dengan memasukkan nilai parameter diatas dengan rumus sebagai berikut:

1. **Accuracy**, menunjukkan seberapa akurat model Machine Learning dalam mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2. **Precision**, menunjukkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Precision* yang nilainya tinggi artinya permintaan informasi dengan jawaban memiliki probabilitas kecocokan yang tinggi [19].

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

3. **Recall**, menunjukkan keberhasilan model dalam menemukan kembali sebuah informasi. Recall dengan nilai yang tinggi artinya kelas dikenali dengan benar [18].

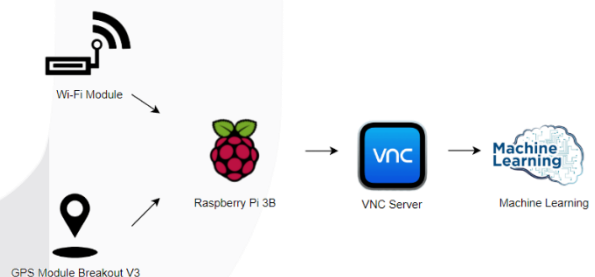
$$Recall = \frac{TP}{TP + FN}$$

4. **F1-Score**, menunjukkan representasi dari penggabungan antara *Precision* dan *Recall* yang diukur secara bersamaan dengan menggunakan *Harmonic Mean* pada *Arithmetic Mean*.

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

III. METODE

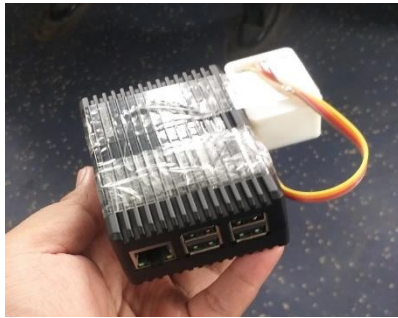
A. Desain Sistem



Gambar 1(A) Desain Sistem

Pada penelitian ini membahas tentang desain sistem prediksi kemacetan dengan menggunakan modul IoT untuk memperoleh data yang dibutuhkan nantinya yaitu *speed* bus saat melintasi sepanjang jalur Trans Metro Bandung. GPS Module Breakout V3 membantu dalam memperoleh data *timestamp*, *latitude*, *longitude*, dan *speed* dari bus itu sendiri. GPS Module Breakout V3 terhubung dengan Raspberry Pi 3B sebagai *Microcomputer*. Wi-Fi Module juga terhubung ke Raspberry Pi 3B agar dapat di *remote* dengan VNC Server sebagai antarmuka. Data hasil *tracking* bus yang sudah diperoleh akan diunduh melalui Google Drive dan diproses ke Machine Learning untuk dapat melakukan prediksi apakah di sepanjang jalur yang dilalui oleh bus mengalami kemacetan atau tidak.

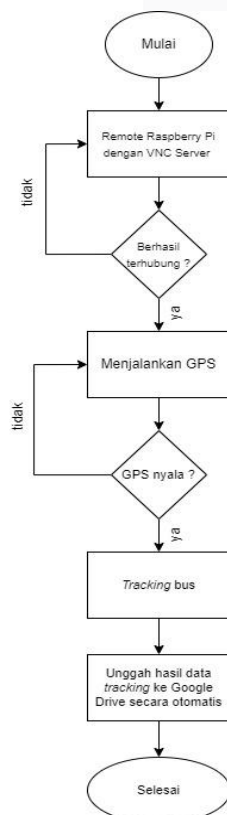
B. Desain Alat



Gambar 2(B) Desain Alat

Pada penelitian ini, perancangan perangkat keras terdiri dari GPS Module Breakout V3 untuk melacak lokasi dan kecepatan bus ketika beroperasi, Raspberry Pi 3B sebagai *Microcomputer* agar dapat menjalankan program, dan Modem agar perangkat IoT terhubung dengan internet. Komponen IoT dilengkapi dengan *Case* untuk menjaga dan merawat alat dari kejadian yang tidak diinginkan seperti terkena air, debu, dan tidak mudah tersenggol oleh orang lain. Raspberry Pi 3B dialirkan listrik melalui sumber listrik dari Bus yang masih bertegangan DC sehingga diperlukan Inverter untuk dapat mengubah tegangan DC menjadi AC.

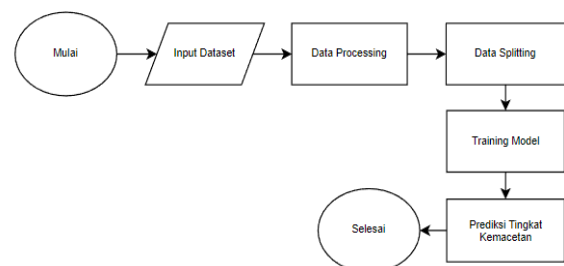
C. Pengambilan Data



Gambar 3(C) Flowchart Proses Pengambilan Data GPS

Modul GPS merekam posisi bus berupa data *latitude* dan *longitude*, *speed* bus dan *datetime* untuk keterangan waktu kapan proses *tracking* dilakukan. *Tracking* bus dilakukan ketika bus mulai beroperasi dari titik awal keberangkatan bus hingga titik akhir pemberhentian bus. Untuk melakukan *tracking* bus, diperlukan akses terlebih dahulu ke Raspberry Pi karena *tracking* dilakukan secara *remote* melalui VNC Server. Setelah berhasil mengakses ke Raspberry Pi, GPS perlu dinyalakan terlebih dahulu dengan menjalankan program yang sudah dibuat. Jika GPS berhasil nyala, proses *tracking* telah berjalan dan akan merekam posisi bus selama beroperasi. Setelah itu, jika bus telah berhenti di titik pemberhentian terakhir atau selesai beroperasi maka *tracking* akan dihentikan. Hasil data *tracking* akan tersimpan secara local di Raspberry Pi dan diunggah secara otomatis ke Google Drive menggunakan Google Drive API. Dengan data yang diperoleh berupa titik koordinat yaitu *latitude*, *longitude*, *timestamp*, dan kecepatan. *Latitude* dan *longitude* akan menjadi informasi lokasi dimana bus berada. *Timestamp* akan menjadi data menyatakan keterangan waktu. Kecepatan bus akan menjadi parameter untuk menentukan tingkat kemacetan yang nantinya akan dirata-rata terlebih dahulu. Data hasil *tracking* ini akan dijadikan dataset untuk diproses ke machine learning.

D. Perancangan Model Machine Learning

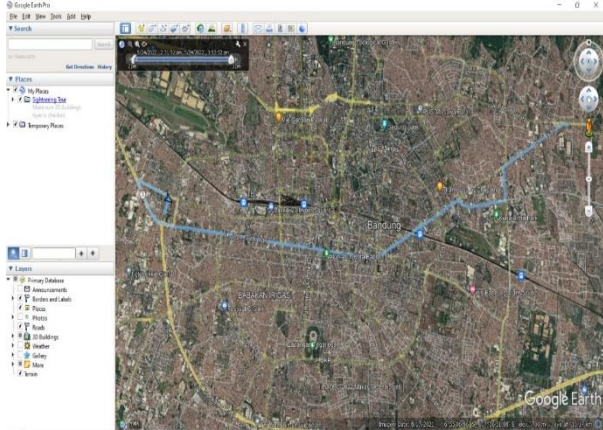


Gambar 4(D) Flowchart Perancangan Model Machine Learning

Sebelum dataset digunakan untuk melakukan *training* pada model, perlu dilakukan beberapa tahapan pengolahan data seperti konversi data, *labelling* data dan pengecekan data NAN. Konversi data perlu dilakukan dikarenakan format file hasil *tracking* GPS masih berupa NMEA. Format tersebut tidak dapat diproses ke model machine learning sehingga perlu dikonversi menjadi format CSV menggunakan bahasa pemrograman Python. Selain itu, *labelling* juga digunakan untuk menambah atribut yang dibutuhkan pada dataset. Pengecekan data NAN juga dilakukan agar jika ada data kosong dapat di *drop* sehingga tidak mengganggu komputasi setelahnya. Pada proses ini disebut juga dengan tahapan *Preprocessing* data.

Setelah itu, dataset akan dipisahkan menjadi 2 bagian, yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih model melalui algoritma agar model yang dilatih bisa belajar pola data yang diberikan. Sedangkan data *testing* digunakan untuk menguji dan membuktikan apakah model sudah sesuai dan akurat serta dapat mengetahui performansi model yang didapatkan pada *testing*. Pada proses ini disebut juga dengan tahapan *Data Splitting*. Pada *Data*

splitting dengan rasio 60:40 artinya 60% sebagai data



	A	B	C	D	E	F	G	H	I	J	K
1	date_time	lat	lon	speed							
2	22-05-24 06:37:01.000	-6.90239	107.65643	0							
3	22-05-24 06:37:02.000	-6.90239	107.65643	1							
4	22-05-24 06:37:03.000	-6.90239	107.656428	1							
5	22-05-24 06:37:04.000	-6.902388	107.656418	1							
6	22-05-24 06:37:05.000	-6.902388	107.656408	2							
7	22-05-24 06:37:06.000	-6.902385	107.656393	3							
8	22-05-24 06:37:07.000	-6.902378	107.656385	3							
9	22-05-24 06:37:08.000	-6.902358	107.656375	4							
10	22-05-24 06:37:09.000	-6.902348	107.656372	4							
11	22-05-24 06:37:10.000	-6.90234	107.656373	4							
12	22-05-24 06:37:11.000	-6.902333	107.656372	3							
13	22-05-24 06:37:12.000	-6.902327	107.656372	3							
14	22-05-24 06:37:13.000	-6.902322	107.656365	3							
15	22-05-24 06:37:14.000	-6.902318	107.65636	3							
16	22-05-24 06:37:15.000	-6.902313	107.656357	3							
17	22-05-24 06:37:16.000	-6.902313	107.656353	2							
18	22-05-24 06:37:17.000	-6.90231	107.656352	1							

Gambar 1(A) Data Hasil Tracking GPS dengan Format TXT

training dan 40% sebagai data *testing*.

Selanjutnya, pada *training* model, akan dilakukan dengan menggunakan algoritma Decision Tree dan Random Forest. Dataset yang sudah melalui tahap *preprocessing* akan digunakan pada tahapan ini dengan menentukan parameter *Feature* (variabel X) dan Target (variabel Y). Dengan kata lain, variabel X merupakan variabel input dan variabel Y merupakan variabel output. Setelah itu proses *training* model dilakukan sesuai dengan algoritma yang digunakan dan jika proses *training* selesai dapat dilihat performansi model dengan melihat nilai *Accuracy*. Jika hasil *Accuracy* yang dihasilkan bagus, maka model siap digunakan untuk prediksi.

Gambar 2(A) Visualisasi Data Hasil Tracking GPS di Google Earth dengan Format NMEA

B. Hasil Preprocessing Data

Konversi data dilakukan untuk mengkonversi data file dengan format NMEA tersebut menjadi format CSV agar dijadikan dataset dan bisa diproses di machine learning. Setelah itu juga dilakukan *cleaning* data dan pelabelan data yang nanti akan diperlukan dalam pemrosesan machine learning. Berikut data hasil *tracking* dengan format CSV dan keterangan pelabelan data yang ditambahkan pada dataset:

E. Pengujian Model

Pengujian model dilakukan untuk mengetahui performansi model machine learning yaitu Decision Tree dan Random Forest dengan menggunakan Confusion Matrix dengan mengukur *Precision*, *Recall*, *F1-Score*, dan *Accuracy* pada model.dengan melihat nilai *Accuracy*.

Gambar 3(B) Data Mentah Hasil Konversi dari NMEA ke CSV

Tabel 1(B) Pelabelan Data

Feature	Keterangan	Label
Jam_sibuk	Ya	1
	Tidak	2
Kondisi	Macet (<i>avg speed</i> <= 8 km/jam)	1
	Sedang (9 <= <i>avg speed</i> <= 17 km/jam)	2
	Lancar (<i>avg speed</i> >= 18km/jam)	3
Jalur	Terminal Cicaheum-Halte Padasuka	1
	Halte Padasuka-Halte A.Yani	2
	Halte A.Yani-Halte Bank Mahyapada	3
	Halte Bank Mahyapada-Halte BTM	4
	Halte BTM-Halte Jl.Jakarta	5
	Halte Jl Jakarta-Halte KONI	6
	Halte KONI-Halte Plaza IBCC	7
	Halte Plaza IBCC-Halte Jaya Plaza	8

IV.HASIL DAN PEMBAHASAN

A. Data dan Visualisasi Hasil Tracking GPS

Data hasil *tracking* GPS diambil dari Raspberry Pi berupa format TXT kemudian dikonversi menjadi format NMEA terlebih dahulu agar hasil *tracking* GPS bisa ditampilkan di Google Earth.

```

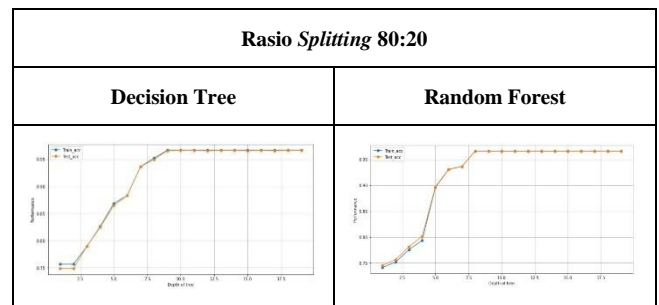
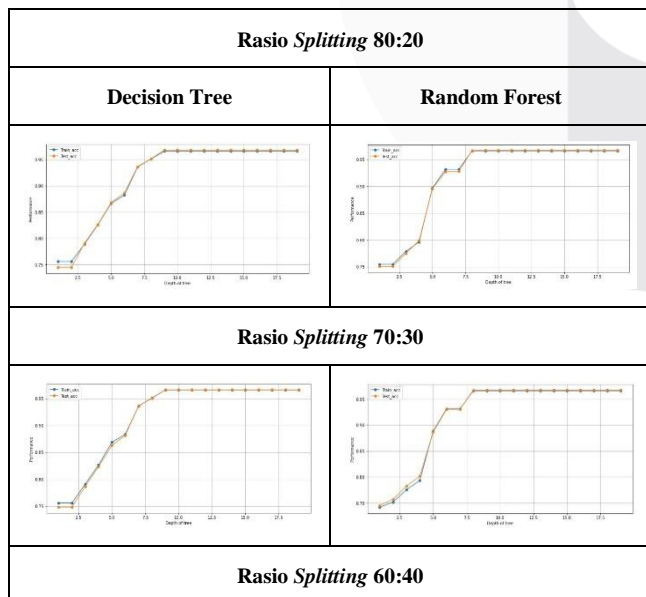
File Edit Format View Help
$GPGGA,082116.000,0654.7948,S,10734.5484,E,2,08,0.93,754.1,M,3.1,M,0.00,0.00
$GPGSA,A,3,20,17,14,13,05,19,12,06,,,,,1.29,0.93,0.89*08
$GPRMC,082116.000,A,0654.7948,S,10734.5484,E,1.31,30.72,250522,,D*44
$GPVTG,30.72,T,M,1.31,N,2.42,K,D*09
$GPGGA,082117.000,0654.7945,S,10734.5487,E,2,08,0.93,754.1,M,3.1,M,0.00,0.00
$GPGSA,A,3,20,17,14,13,05,19,12,06,,,,,1.29,0.93,0.89*08
$GPGSV,4,1,14,41,60,284,25,05,51,281,16,20,45,333,28,19,43,064,29*72
$GPGSV,4,2,14,13,43,181,18,17,39,091,27,12,29,309,20,14,23,152,18*74
$GPGSV,4,3,14,15,18,211,15,02,14,346,,30,11,124,13,06,09,028,22*75
$GPGSV,4,4,14,24,08,244,14,193,,*7E
$GPRMC,082117.000,A,0654.7945,S,10734.5487,E,1.32,34.08,250522,,D*41
$GPVTG,34.08,T,M,1.32,N,2.45,K,D*04
$GPGGA,082118.000,0654.7942,S,10734.5489,E,2,08,0.93,754.0,M,3.1,M,0.00,0.00
$GPGSA,A,3,20,17,14,13,05,19,12,06,,,,,1.29,0.93,0.89*08
$GPRMC,082118.000,A,0654.7942,S,10734.5489,E,1.30,35.88,250522,,D*4C
$GPVTG,35.88,T,M,1.30,N,2.41,K,D*08
$GPGGA,082119.000,0654.7939,S,10734.5491,E,2,08,0.93,754.0,M,3.1,M,0.00,0.00
$GPGSA,A,3,20,17,14,13,05,19,12,06,,,,,1.29,0.93,0.89*08
$GPRMC,082119.000,A,0654.7939,S,10734.5491,E,1.36,39.71,250522,,D*44
$GPVTG,39.71,T,M,1.36,N,2.52,K,D*05
$GPGGA,082120.000,0654.7936,S,10734.5494,E,2,08,0.93,753.9,M,3.1,M,0.00,0.00
$GPGSA,A,3,20,17,14,13,05,19,12,06,,,,,1.29,0.93,0.89*08
  
```

Feature	Keterangan	Label
	Halte Jaya Plaza-Halte Jl.Ketapang	9
	Halte Jl Ketapang-Halte HSBC	10
	Halte HSBC-Halte Alun-Alun	11
	Halte Alun-Alun-Halte KEB Hana	12
	Halte KEB Hana-Halte Mahaypada Tower	13
	Halte Mahaypada Tower-Halte Jendral Sudirman	14
	Halte Jendral Sudirman-Halte Bunderan Sudirman	15
	Halte Bunderan Sudirman-Halte Jendral Sudirman 3	16
	Halte Jendral Sudirman 3-Terminal Damri Elang	17

C. Menentukan Nilai Max Depth

Dalam menentukan Max Depth, dilakukan dengan melihat hasil grafik akurasi dari data train dan data test berdasarkan nilai Max Depth dari range 1 sampai dengan 20. Nilai Max Depth yang mempunyai akurasi tinggi pada data test akan dimasukkan ke parameter Max Depth pada model untuk mendapatkan performansi model terbaik. Berikut grafik yang dihasilkan pada model Decision Tree dan Random Forest:

Tabel 2(C) Grafik Akurasi Data *Training* dan Data *Testing* Berdasarkan Nilai Max Depth



Berdasarkan Tabel diatas pada model Decision Tree akurasi pada data test mencapai angka tertinggi pada nilai Max Depth sebesar 9. Sedangkan pada model Random Forest mencapai akurasi tertinggi pada nilai Max Depth sebesar 8. Jika nilai Max Depth pada masing-masing model lebih besar dari nilai tersebut, akurasi cenderung akan sama, tidak akan bertambah ataupun berkurang. Nilai tersebut akan ditetapkan pada Max Depth untuk dimasukkan ke model sebagai parameter.

D. Pengujian Performansi Model

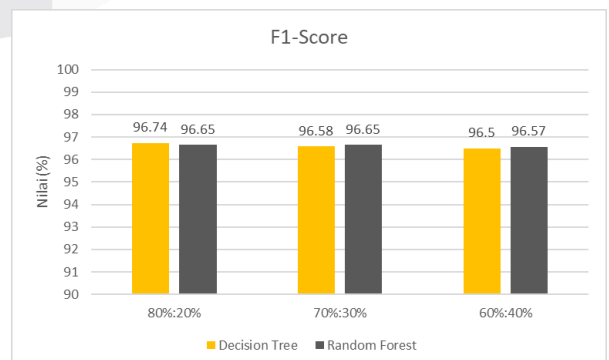
Pengujian ini dilakukan untuk melihat akurasi performansi model dengan menggunakan confusion matrix pada masing-masing rasio *splitting* data. Parameter yang



akan diukur adalah *Precision*, *Recall*, *F1-Score*, dan *Accuracy*. Berikut hasil pengukurannya:

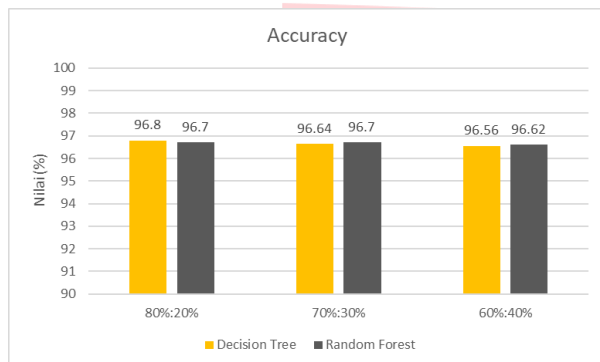
Gambar 4(D) Perbandingan Nilai *Precision*

Gambar 5(D) Perbandingan Nilai *Recall*



Gambar 6(D) Perbandingan Nilai *F1-Score*Gambar 7(D) Perbandingan Nilai *Accuracy*

Berdasarkan hasil pengukuran yang telah didapatkan, diperoleh model Decision Tree sebagai model dengan performansi lebih baik daripada Random Forest pada *splitting* data 80:20 dengan nilai *Precision* 96.95%, *Recall* 96.8%, *F1-Score* 96.74%, dan *Accuracy* 96.8%. Namun demikian, Random Forest akan tetap memiliki performansi lebih unggul dari pada Decision Tree dikarenakan selisih nilai yang didapatkan tidak terlalu jauh dan lebih cocok untuk pengklasifikasian data yang berjumlah banyak.



E. Pengujian Prediksi Tingkat Kemacetan dan Visualisasi di Website

Tabel 3(E) Data Aktual

Hari	Jam Sibuk	Jalur	Kondisi
2	1	12	1
4	2	5	2
3	1	17	3
2	1	15	3
3	2	8	2

Pada Tabel diatas merupakan data aktual yang akan menjadi acuan untuk mengetahui keakuratan hasil prediksi tingkat kemacetan dengan melakukan perbandingan pada hasil prediksi menggunakan Decision Tree dan Random Forest.

Tabel 4(E) Data Prediksi Kemacetan Decision Tree

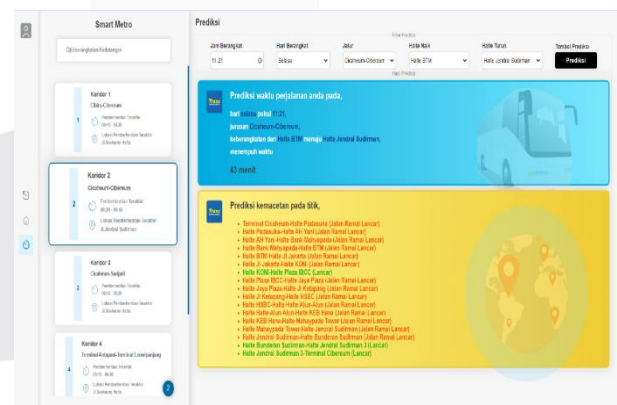
Input			Output
Hari	Jam Sibuk	Jalur	Kondisi
2	1	12	1
4	2	5	2

Input			Output
Hari	Jam Sibuk	Jalur	Kondisi
3	1	17	3
2	1	15	3
3	2	8	2

Tabel 5(E) Data Prediksi Kemacetan Random Forest

Input			Output
Hari	Jam Sibuk	Jalur	Kondisi
2	1	12	1
4	2	5	2
3	1	17	3
2	1	15	3
3	2	8	2

Berdasarkan Tabel diatas menunjukkan hasil prediksi pada Decision Tree dan Random Forest. Pada hasil prediksi yang dihasilkan antara kedua model menghasilkan *output* yang sama. Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa kedua model tersebut menghasilkan hasil prediksi yang akurat dengan membandingkan dengan data aktual. Kemudian hasil prediksi ditampilkan di website seperti berikut:



Gambar 8(E) Tampilan Hasil Prediksi di Website

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut bahwa:

1. Pada penelitian tugas akhir ini telah diusulkan suatu sistem untuk prediksi tingkat kemacetan di jalur armada bus Trans Metro Bandung dengan memanfaatkan Internet of Things dan Machine Learning.

2. GPS Breakout V3 module menghasilkan data hasil tracking bus yang tidak menyimpang jauh dari jalur yang dilewati oleh bus TMB.
3. Dalam menentukan nilai Max Depth terbaik diperoleh nilai Max Depth 9 pada model Decision Tree dan pada model Random Forest diperoleh nilai Max Depth 8.
4. Data Splitting dengan rasio 80:20 terpilih sebagai split data terbaik yang memiliki hasil accuracy tertinggi.
5. Dengan menggunakan nilai Max Depth terbaik dan data splitting terbaik diperoleh model Decision Tree dengan nilai Precision 96.95%, Recall 96.8%, F1-Score 96.74%, dan Accuracy 96.8%.
6. Dengan menggunakan nilai Max Depth terbaik dan data splitting terbaik diperoleh model Random Forest dengan nilai Precision 96.87%, Recall 96.7%, F1-Score 96.65%, dan Accuracy 96.7%.
7. Diperoleh model Decision Tree sebagai model dengan performansi lebih baik daripada Random Forest pada dataset yang digunakan. Namun demikian, Random Forest akan tetap memiliki performansi lebih unggul dari pada Decision Tree dikarenakan selisih nilai yang didapatkan tidak terlalu jauh dan lebih cocok untuk pengklasifikasian data yang berjumlah banyak.

VI. REFERENSI

- [1] CNN, "Bandung Kota Tarmac se-Indonesia, Ragam Solusi Disiapkan," 2019. <https://www.cnnindonesia.com/nasional/20191007205754-20-437595/bandung-kota-termacet-se-indonesia-ragam-solusi-disiapkan>.
- [2] V. B. Kusnandar, "Jumlah Penduduk Kota Bandung Sebanyak 2,44 juta Jiwa pada 2020," 2021. <https://databoks.katadata.co.id/datapublish/2021/10/01/jumlah-penduduk-kota-bandung-sebanyak-244-juta-jiwa-pada-2020>.
- [3] E. Kinerja Trans Metro Bandung, L. dan Reni Puspitasari, L. dan Reni Puspitasari Puslitbang Perhubungan Darat dan Perkeretaapian, and J. Medan Merdeka Timur Nomor, "Evaluasi Kinerja Trans Metro Bandung Performance Evaluation of Trans Metro Bandung," pp. 65–78, 2015.
- [4] S. Devi and T. Neetha, "Machine Learning based traffic congestion prediction in a IoT based Smart City," *Int. Res. J. Eng. Technol.*, pp. 3442–3445, 2017, [Online]. Available: www.irjet.net.
- [5] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," *Proc. - 2017 10th Int. Symp. Comput. Intell. Des. Isc. 2017*, vol. 2, pp. 361–364, 2018, doi: 10.1109/ISCID.2017.216.
- [6] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks," *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 222–231, 2018, doi: 10.1177/0361198118777631.
- [7] D. E., D. P., L. B., and M. F., "Real-Time Detection of Traffic from Twitter Stream Analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2269–2283, 2015, [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84938802216&partnerID=40&md5=5747f5988b7a2dbb4d7f8597f78d5572>.
- [8] M. A. Khan and S. F. Khan, "IoT based framework for Vehicle Over-speed detection," *1st Int. Conf. Comput. Appl. Inf. Secur. ICCAIS 2018*, pp. 1–4, 2018, doi: 10.1109/CAIS.2018.8441951.
- [9] C. Wai Zhao, J. Jegatheesan, and S. Chee Loon, "Exploring IOT Application Using Raspberry Pi," *Int. J. Comput. Networks Appl.*, vol. 2, no. 1, pp. 27–34, 2015, [Online]. Available: <http://www.digi.com>.
- [10] E. Susanti and J. Triyono, "Pengembangan Sistem Pemantau dan Pengendalian Menggunakan Raspberry Pi dan Firebase," *J. Konf. Nas. Teknol. Inf. dan Komun. (KNASTIK 2016)*, vol. 1, no. 1, pp. 144–153, 2016.
- [11] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.
- [12] A. Ajay Shingare and N. C. and P. Deshpande, "GPS Supported City Bus Tracking & Smart Ticketing System," pp. 93–98, 2015.
- [13] Y. Ren, "Python Machine Learning: Machine Learning and Deep Learning With Python," *Int. J. Knowledge-Based Organ.*, vol. 11, no. 1, pp. 67–70, 2021.
- [14] A. Priyam, R. Gupta, A. Rathee, and S. Srivastava, "Comparative Analysis of Decision Tree Classification Algorithms," pp. 334–337, 2013.
- [15] Z. Azmi and M. Dahria, "Decision Tree Berbasis Algoritma Untuk Pengambilan Keputusan," *Saintikom*, vol. 12, pp. 157–164, 2013, [Online]. Available: <http://demo.pohonkeputusan.com/files/DECISION TREE BERBASIS ALGORITMA UNTUK PENGAMBILAN KEPUTUSAN.pdf?i=1>.
- [16] E. Fitri, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *J. Transform.*, vol. 18, no. 1, p. 71, 2020, doi: 10.26623/transformatika.v18i1.2317.
- [17] J. Yang, J. Gong, W. Tang, Y. Shen, C. Liu, and J. Gao, "Delineation of Urban Growth Boundaries Using a Patch-Based Cellular Automata Model under Multiple Spatial and Socio-Economic Scenarios."

- [18] A. Sharma, "Confusion Matrix in Machine Learning," *Www.Geeksforgeeks.Org*, 2018, [Online]. Available: <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>.
- [19] A. Yuliana and D. B. Pratomo, "Memprediksi Kepuasan Mahasiswa Terhadap Kinerja Dosen Politeknik TEDC Bandung," *Semnasinotek 2017*, pp. 377–384, 2017.

