

Prediksi Kemacetan Lalu Lintas Jaringan Menggunakan Algoritma Random Forest Berbasis Data Mikrotik

Desmita Wahyu Putri*¹, Budi Sutomo*²

^{1,2,3}Jurusan Teknik Informatika, FTBS Universitas Dharma Wacana,
Jl. Kenanga No.3, Mulyojati, Kec. Metro Barat, Kota Metro, Lampung 34121
e-mail: *¹desmitaw23@gmail.com, ²budi.atmel@gmail.com

Abstrak

Kemacetan lalu lintas jaringan merupakan tantangan utama dalam pengelolaan jaringan komputer, terutama di lingkungan kampus yang memiliki jumlah pengguna dan kebutuhan akses internet yang terus meningkat. Penggunaan jaringan untuk berbagai aktivitas akademik dan operasional sering kali menyebabkan kemacetan, yang berdampak pada penurunan kualitas layanan dan kinerja pengguna dalam mengakses informasi serta berkolaborasi secara online. Permasalahan seperti volume data yang tinggi, ketidakstabilan kinerja jaringan, alokasi bandwidth yang tidak efisien, serta kurangnya sistem prediksi kemacetan menjadi faktor penyebab utama. Penelitian ini bertujuan untuk memprediksi kemacetan jaringan menggunakan algoritma Random Forest. Data yang digunakan diperoleh dari perangkat MikroTik dan analisis paket menggunakan Wireshark, mencakup informasi tentang pengguna aktif, jitter, dan bandwidth. Metode penelitian meliputi pengumpulan data, pra-pemrosesan, pelatihan model, dan evaluasi kinerja model. Hasil penelitian menunjukkan bahwa algoritma Random Forest mampu memprediksi kemacetan jaringan. Hal ini dapat memberikan wawasan penting bagi pengelola jaringan dalam mengambil langkah-langkah untuk mengoptimalkan kinerja jaringan dan mengurangi kemacetan. Dengan menyediakan pendekatan berbasis data kuantitatif, penelitian ini diharapkan dapat menjadi solusi efektif untuk mengatasi permasalahan kemacetan jaringan, khususnya di lingkungan yang kompleks seperti kampus.

Kata Kunci: *Kemacetan jaringan, Random Forest, MikroTik, Wireshark, prediksi.*

Abstract

Network traffic congestion is a major challenge in computer network management, especially in campus environments with a growing number of users and increasing internet access needs. The use of the network for various academic and operational activities often causes congestion, which impacts the quality of service and user performance in accessing information and collaborating online. Issues such as high data volume, network performance instability, inefficient bandwidth allocation, and lack of congestion prediction systems are the main contributing factors. This research aims to predict network congestion using the Random Forest algorithm. The data used was obtained from MikroTik devices and packet analysis using Wireshark, including information about active users, jitter, and bandwidth. The research methods include data collection, preprocessing, model training, and model performance evaluation. The research results show that the Random Forest algorithm is capable of predicting network congestion. This can provide important insights for network managers in taking steps to optimize network performance and reduce congestion. By providing a quantitative data-based approach, this research is expected to be an effective solution for addressing network congestion issues, particularly in complex environments such as campuses.

Keywords: *Network congestion, Random Forest, MikroTik, Wireshark, prediction.*

1. PENDAHULUAN

Perkembangan internet dan jaringan yang terjadi di masa sekarang ini memberikan keuntungan dan kemudahan kepada para pengguna komputer untuk dapat berbagi sumber daya informasi dalam suatu jaringan lokal maupun internasional. Kemacetan pada jaringan komputer adalah salah satu bentuk kegagalan pada jaringan. Hal ini dapat membuat jaringan berada dalam masalah dan mengurangi kualitas pelayanan jaringan. Kemampuan untuk secara akurat memprediksi kegagalan pada jaringan komputer dapat membantu meminimalkan biaya administrasi [1].

Kemacetan lalu lintas jaringan merupakan tantangan utama yang sering dihadapi dalam pengelolaan jaringan komputer, terutama di lingkungan kampus. Penggunaan internet yang semakin meningkat, baik untuk kegiatan akademik maupun operasional, seringkali menyebabkan kemacetan yang berdampak pada penurunan kualitas layanan jaringan. Hal ini mempengaruhi kinerja pengguna dalam mengakses informasi dan berkolaborasi secara online, yang pada akhirnya menurunkan efisiensi dan produktivitas. Kemacetan pada jaringan, salah satunya, ditandai dengan tingginya waktu akses internet dalam membuka suatu situs. Hal ini ditandai dengan waktu akses internet yang lebih lama saat membuka website. Hal ini terjadi karena antrian data yang panjang dan tingginya kehilangan data (*packet loss*).

Berbagai penelitian telah dilakukan untuk mengatasi masalah kemacetan jaringan, dalam penelitian sebelumnya yang berjudul *Prediksi Kemacetan pada Jaringan Komputer Menggunakan Metode Naive Bayesian Classifier* [2] Penelitian ini menggunakan simulasi training dari parameter jaringan yang dikonversi ke dalam nilai probabilitas, yang kemudian diproses dengan metode *Bayesian*. Hasilnya menunjukkan metode ini cukup efektif dalam memprediksi kemacetan jaringan. Penelitian lain berjudul *Pengendalian Kemacetan Jaringan Melalui Per-Flow Multipath Routing* Penelitian [3] mengusulkan penggabungan teknik multipath routing dan pengendalian kemacetan jaringan pada protokol *OSPF*. Teknik ini terbukti mampu mengurangi rasio kemacetan melalui pengalihan jalur dan pemerataan beban pada jaringan, sehingga meningkatkan performansi jaringan secara keseluruhan.

Selain itu, penelitian lainnya yang berjudul "Studi mengenai kemacetan lalu lintas dan domain penyiaran di dalam jaringan komputer di lingkungan STKIP PGRI Pacitan" [4] membahas tentang evaluasi kemacetan lalu lintas dan penyiaran di jaringan komputer untuk meningkatkan kinerja jaringan yang ada serta memberikan pengawasan terhadap munculnya kemacetan. Menguraikan metode dalam menerapkan daftar akses. dan cara penyiaran untuk meningkatkan domain Anda. Domain menjadi hal yang sangat krusial. bisa dengan mudah diminimalkan. Penelitian lainnya menggunakan algoritma *Long Short-Term Memory (LSTM)* dalam *Deep Learning* untuk memprediksi kemacetan jaringan di PT XYZ [4] Hasil penelitian tersebut menunjukkan bahwa LSTM mampu memprediksi kemacetan dengan akurasi yang tinggi, memberikan alternatif solusi berbasis machine learning yang menjanjikan dalam menangani kemacetan lalu lintas jaringan.

Melihat berbagai pendekatan yang telah dilakukan, upaya prediksi kemacetan jaringan tetap menjadi fokus yang relevan untuk menjaga ketersediaan layanan jaringan yang stabil dan berkualitas. Dalam konteks ini, algoritma *Random Forest* berbasis data MikroTik menawarkan potensi besar untuk memberikan solusi prediktif yang akurat dan efisien, sehingga mampu mengatasi kemacetan jaringan dengan lebih baik.

2. METODE PENELITIAN

2.1 Pendekatan Penelitian

Metode penelitian ini menggunakan pendekatan kuantitatif dengan eksperimen pada data jaringan yang dikumpulkan menggunakan Wireshark, Wireshark ini merupakan software untuk melakukan analisa aktivitas jaringan komputer yang memiliki fungsi-fungsi yang berguna bagi profesional jaringan, administrator, peneliti, hingga pengembang piranti lunak jaringan. Tools dapat bekerja secara real time dalam menangkap paket - paket data/informasi yang berjalan dalam jaringan. [5] Algoritma *Random Forest* digunakan sebagai model *machine learning* untuk memprediksi kemacetan jaringan berdasarkan pola yang teridentifikasi dalam data.

2.2 Tahapan Penelitian

Identifikasi Masalah

Kemacetan jaringan menjadi tantangan utama di lingkungan kampus seperti Universitas Dharma Wacana, terutama pada waktu sibuk, mengakibatkan penurunan kualitas layanan dan gangguan produktivitas. Data Mikrotik dan data Wireshark, yang mencatat informasi seperti jumlah pengguna aktif, volume *traffic bandwidth*, dan status jaringan, belum dimanfaatkan secara optimal untuk analisis dan prediksi kemacetan.

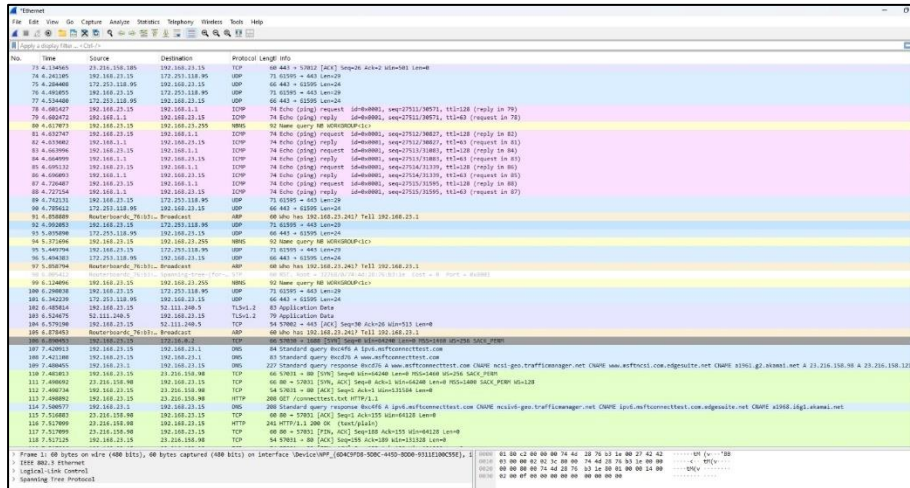
Kemacetan jaringan bersifat dinamis, dipengaruhi oleh lonjakan pengguna dan penggunaan bandwidth yang tidak terkendali. Saat ini, pengelola jaringan hanya dapat bereaksi setelah kemacetan terjadi, tanpa sistem prediksi yang memungkinkan tindakan preventif. Dengan memanfaatkan algoritma machine learning seperti Random Forest, data Mikrotik dapat digunakan untuk memprediksi kemacetan jaringan secara akurat. Hal ini memungkinkan pengelola untuk mengambil tindakan proaktif, seperti pembatasan bandwidth atau peningkatan kapasitas jaringan, guna meningkatkan kualitas layanan dan kepuasan pengguna. Manajemen bandwidth adalah proses mengatur lalu lintas Internet jaringan sedemikian rupa sehingga tidak ada perbedaan kualitas layanan Internet dengan pengguna jaringan lainnya. Manajemen bandwidth memberikan kemampuan untuk mengelola bandwidth jaringan dan memberikan tingkat layanan sesuai dengan kebutuhan dan prioritas pelanggan. [6]

a. Pengumpulan Data

Dalam penelitian prediksi kemacetan jaringan, pengumpulan data dilakukan menggunakan dua alat utama, yaitu Wireshark dan Winbox. Masing-masing alat memiliki fungsi dan peran yang berbeda dalam menangkap informasi terkait aktivitas jaringan. Proses pengumpulan data dengan Wireshark dilakukan sebagai berikut:

1. Menyiapkan Wireshark
 - a. Jalankan Wireshark di perangkat yang terhubung ke jaringan.
 - b. Pilih interface jaringan (seperti Wi-Fi atau Ethernet) untuk memantau lalu lintas yang sedang berlangsung.
2. Menangkap Data
 - a. Wireshark mulai menangkap paket data yang melewati jaringan, seperti HTTP, TCP, UDP, atau protokol lainnya.
 - b. Data yang dikumpulkan mencakup informasi seperti:
 - 1) Alamat IP sumber dan tujuan: Untuk mengidentifikasi pengguna aktif.
 - 2) Ukuran paket: Untuk menghitung total bandwidth yang digunakan.
 - 3) Jitter: Mengukur perbedaan waktu antar paket yang dikirim.

3. Filter Data
Gunakan filter (misalnya, "ip.addr == [alamat IP]" atau "tcp") untuk menyaring data yang relevan sesuai dengan kebutuhan analisis.
4. Ekspor Data
Data yang telah difilter diekspor dalam format CSV atau teks untuk diproses lebih lanjut.



Gambar 1. Data Jaringan melalui tools Wireshark

Kemudian disini untuk pengambilan data melalui Winbox prosesnya adalah sebagai berikut:

1. Akses Mikrotik dengan Winbox:
Login ke perangkat Mikrotik melalui Winbox menggunakan IP, username, dan password.
2. Melihat Statistik Jaringan:
 - a. Setelah berhasil login, buka menu Interfaces di panel sebelah kiri.
 - b. Pilih interface jaringan yang ingin dimonitor, misalnya ether1, ether2, atau wlan1.
 - c. Klik kanan pada interface yang dipilih, lalu pilih Torch.
 - d. Kemudian mengatur parameter untuk memfilter lalu lintas jaringan yaitu interface yang ingin dipantau, alamat IP Address, port dan protokol.
 - e. Selanjutnya mencatat atau membuat tangkapan layar hasil monitoring untuk dianalisis lebih lanjut.

Setelah Data yang dikumpulkan dari Wireshark dan Winbox kemudian disusun untuk dianalisis untuk melihat pola dan hubungan antar parameter, yang kemudian digunakan sebagai input untuk model prediksi.

Address	MAC Address	Client ID	Server	Active Address	Active MAC Address	Active Host Name	Expires After	Status
192.168.23.226	00:ED:4C:CC:66:09	1:0:a0:4c:cc:66:9	dhcp5	192.168.23.226	00:ED:4C:CC:66:09	DESKTOP-43812HL	00:47:36	bound
192.168.23.38	00:ED:4C:F8:86:08	1:0:e0:4c:f8:86:8	dhcp5	192.168.23.38	00:ED:4C:F8:86:08	CL01-16	00:36:36	bound
115.50.154	02:7F:21:68:8E:F5	1:2:0:21:68:8e:f5	dhcp4	115.50.154	02:7F:21:68:8E:F5	Intra-SMART-6	00:07:49	bound
115.50.247	04:ED:33:68:53:6c	1:4:e0:33:68:53:6c	dhcp4	115.50.247	04:ED:33:68:53:6c	novan	00:56:18	bound
115.50.168	12:25:8D:08:31:4A	1:12:25:8d:08:31:4a	dhcp4	115.50.168	12:25:8D:08:31:4A	Realme-C15	00:38:55	bound
115.50.223	12:84:3F:4E:30:19	1:12:84:3f:4e:30:19	dhcp4	115.50.223	12:84:3F:4E:30:19	Galaxy-A545ys	00:45:47	bound
115.50.172	16:72:45:DA:2A:D2	1:16:72:45:da:2a:d2	dhcp4	115.50.172	16:72:45:DA:2A:D2	OPPO-A78-5G	00:33:01	bound
115.50.7	1E:4C:5F:8A:42:C4	1:1e:4c:5f:8a:42:c4	dhcp4	115.50.7	1E:4C:5F:8A:42:C4	vivo-1918	00:00:43	bound
115.50.32	26:81:F5:6E:50:09	1:26:81:f5:6e:50:09	dhcp4	115.50.32	26:81:F5:6E:50:09	cherry	00:37:40	bound
115.50.24	3E:AD:A0:02:36:6F	1:3e:ad:a0:02:36:6f	dhcp4	115.50.24	3E:AD:A0:02:36:6F		00:56:44	bound
115.50.228	42:30:8B:36:E9:88	1:42:30:8b:36:e9:88	dhcp4	115.50.228	42:30:8B:36:E9:88	V2111	00:48:09	bound
115.50.29	46:F4:2F:6F:93:20	1:46:f4:2f:6f:93:20	dhcp4	115.50.29	46:F4:2F:6F:93:20	V2131	00:58:21	bound
115.50.221	5E:0B:D8:91:41:07	1:5e:0b:d8:91:41:07	dhcp4	115.50.221	5E:0B:D8:91:41:07	V2131	00:39:22	bound
115.50.2	64:6E:E9:9E:FC:28	1:64:6e:e9:9e:fc:28	dhcp4	115.50.2	64:6E:E9:9E:FC:28	LAPTOP-PSQEPAZL	00:41:01	bound
115.50.14	6C:D7:1F:2C:73:C5	1:6c:d7:1f:2c:73:c5	dhcp4	115.50.14	6C:D7:1F:2C:73:C5	OPPO-A1k	00:51:46	bound
115.50.19	70:D6:23:3A:61:C3	1:70:d6:23:3a:61:c3	dhcp4	115.50.19	70:D6:23:3A:61:C3	Nea	00:40:15	bound
115.50.41	7A:AB:33:17:C1:EA	1:7a:ab:33:17:c1:ea	dhcp4	115.50.41	7A:AB:33:17:C1:EA	V2310	00:13:52	bound
192.168.23.169	8C:EC:4B:87:D7:18	1:8c:ec:4b:87:d7:18	dhcp5	192.168.23.169	8C:EC:4B:87:D7:18	CL03-23	00:46:36	bound
192.168.23.177	8C:EC:4B:87:D7:18	1:8c:ec:4b:87:d7:18	dhcp5	192.168.23.177	8C:EC:4B:87:D7:18	PC93-07	00:55:04	bound
192.168.23.37	8C:EC:4B:87:D7:18	1:8c:ec:4b:87:d7:18	dhcp5	192.168.23.37	8C:EC:4B:87:D7:18	CL03-24	00:37:25	bound
192.168.23.11	98:D7:02:DE:17:96	1:98:d7:02:de:17:96	dhcp5	192.168.23.11	98:D7:02:DE:17:96		00:40:45	bound
115.50.202	A0:1A:59:48:36:8B	1:a0:1a:59:48:36:8b	dhcp4	115.50.202	A0:1A:59:48:36:8B	DESKTOP-4P1LR2P	00:04:22	bound
115.50.176	9C:FC:E8:F9:52:B3	1:9c:fc:e8:f9:52:b3	dhcp4	115.50.176	9C:FC:E8:F9:52:B3	LAPTOP-PPVFKUJ	00:16:52	bound
115.50.140	A2:2F:88:F7:79:70	1:a2:2f:88:f7:79:70	dhcp4	115.50.140	A2:2F:88:F7:79:70	Galaxy-A54	00:31:31	bound
192.168.23.15	A8:A1:59:48:36:8B	1:a8:a1:59:48:36:8b	dhcp5	192.168.23.15	A8:A1:59:48:36:8B	DESKTOP-4P1LR2P	00:58:35	bound
115.50.142	AC:D5:64:B1:E7:DB	1:ac:d5:64:b1:e7:db	dhcp4	115.50.142	AC:D5:64:B1:E7:DB	CL02-08	00:37:29	bound
115.50.179	CC:89:D7:1F:70:AD	1:cc:89:d7:1f:70:ad	dhcp4	115.50.179	CC:89:D7:1F:70:AD	made	00:39:22	bound
192.168.23.241	C4:65:16:30:65:28	1:c4:65:16:30:65:28	dhcp5	192.168.23.241	C4:65:16:30:65:28	CL02-08	00:37:27	bound
115.50.226	CE:AA:BC:44:37:B6	1:ce:aa:bc:44:37:b6	dhcp4	115.50.226	CE:AA:BC:44:37:B6	realme-10	00:49:16	bound
115.50.241	DE:E7:8A:38:71	1:de:e7:8a:38:71	dhcp4	115.50.241	DE:E7:8A:38:71	A15-mk-Desmita	00:31:19	bound
115.50.198	F8:89:71:C3:11:3E	1:f8:89:71:c3:11:3e	dhcp4	115.50.198	F8:89:71:C3:11:3E	Carlson_A1	00:37:20	bound
115.50.29	FA:EC:F2:53:90:6B	1:fa:ec:f2:53:90:6b	dhcp4	115.50.29	FA:EC:F2:53:90:6B	M210K7BNY	00:37:15	bound

Gambar 2. Data User Aktif di Mikrotik

Src. Address	Dest. Address	Src. Address	Dest. Address	MAC Protocol	Protocol	Port	VLAN ID	DSCP
0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	all	any	any	any	any
800 (ip) 1c	115.50.138	115.50.1						
800 (ip) 17c	115.50.15813730	103.157.33.965000 (wp)						
800 (ip) 6 (ip)	115.50.15525324	197.240.15.615222						
800 (ip) 17c	0.0.0.0 (bootpc)	255.255.255.67 (boot)						
800 (ip) 17c	115.50.1548 (bootpc)	115.50.147 (bootpc)						
800 (ip) 2 (ip)	115.50.15	224.0.0.22						
800 (ip) 17c	115.50.15842062	103.157.33.965000 (wp)						
800 (ip) 17c	115.50.155331	224.0.0.251 (mcast)						
800 (ip) 17c	115.50.1581113	115.50.153 (ip)						
800 (ip) 17c	115.50.157092	115.50.153 (ip)						
800 (ip) 17c	115.50.151850	115.50.153 (ip)						
800 (ip) 17c	115.50.1522626	115.50.153 (ip)						
800 (ip) 17c	115.50.151634 (mcast)	115.50.153 (ip)						
800 (ip) 17c	115.50.1534182	115.50.153 (ip)						
800 (ip) 6 (ip)	115.50.1541088	197.240.235.34443 (mcast)						
800 (ip) 17c	115.50.1541003	115.50.153 (ip)						
800 (ip) 6 (ip)	115.50.1547790	142.250.190.3.80 (mcast)						
800 (ip) 6 (ip)	115.50.1546612	246.239.39.30443 (mcast)						
800 (ip) 6 (ip)	115.50.1546076	71.18.255.145443 (mcast)						
800 (ip) 6 (ip)	115.50.1547795	142.250.190.3.80 (mcast)						
800 (ip) 6 (ip)	115.50.1546754	197.240.235.34.443 (mcast)						
800 (ip) 6 (ip)	115.50.15402320	74.125.130.188.5223						
800 (ip) 6 (ip)	115.50.15838958	178.204.72.61.10000						
800 (ip) 6 (ip)	115.50.1542550	13.214.52.63443 (mcast)						
800 (ip) 17c	115.50.1512200	115.50.153 (ip)						
800 (ip) 17c	115.50.15846630	103.157.33.965000 (wp)						
800 (ip) 6 (ip)	115.50.1536370	197.240.235.61.80 (mcast)						
800 (ip) 6 (ip)	115.50.1546000	71.18.255.145443 (mcast)						
800 (ip) 6 (ip)	115.50.1547976	52.77.55.64.443 (mcast)						
800 (ip) 6 (ip)	115.50.1547050	179.194.77.188.5228						
800 (ip) 1c	115.50.15	115.50.1						
800 (ip) 6 (ip)	115.50.1546498	115.50.150 (ip)						
800 (ip) 6 (ip)	115.50.1582890	103.157.33.965000 (wp)						
800 (ip) 6 (ip)	115.50.1546734	142.250.190.3.80 (mcast)						
800 (ip) 50		802-11:89:da:46						
800 (ip) 50		802-11:89:da:46						
800 (ip) 50		802-11:89:da:46						
800 (ip) 50		802-11:89:da:46						
800 (ip) 17c	115.50.15846630	103.157.33.965000 (wp)						

Gambar 3. Data Realtime Traffic Jaringan

c. Pembersihan dan Validasi Data

Data mentah yang diambil dari Mikrotik perlu melalui proses pembersihan untuk memastikan kualitas dataset. Proses ini mencakup:

- 1) *Cleansing*: Menghapus entri yang tidak relevan, duplikat, atau data yang hilang.
- 2) *Normalisasi*: Menyelaraskan skala data agar dapat dibandingkan, misalnya dengan menggunakan teknik *Min-Max Scaling*.
- 3) *Validasi*: Memeriksa konsistensi data dengan membandingkan hasil logging dari beberapa hari atau perangkat lain untuk memastikan akurasi.

d. Penyimpanan Data

Data yang telah bersih dan divalidasi disimpan di basis data atau file sistem dengan struktur yang terorganisasi, seperti:

- 1) Kolom untuk jumlah pengguna aktif, *traffic bandwidth*, dan *jitter*.
- 2) Dataset disiapkan untuk digunakan sebagai input dalam proses analisis dan pelatihan model *machine learning*.

e. Implementasi *Random Forest*

Random Forest adalah algoritma yang andal untuk tugas prediksi, termasuk prediksi kemacetan jaringan. Algoritma ini memanfaatkan data historis untuk menemukan pola yang kompleks dan memberikan prediksi yang akurat. Dengan evaluasi yang tepat, *Random Forest* dapat diintegrasikan ke dalam sistem jaringan untuk memberikan wawasan penting dan mendukung pengambilan keputusan secara real-time. *Random Forest* merupakan salah satu metode yang digunakan untuk klasifikasi dengan membangun banyak pohon klasifikasi. Metode ini dapat meningkatkan hasil akurasi, dengan cara membangkitkan simpul anak untuk setiap node (simpul di atasnya) dan dilakukan pemilihan secara acak. Kemudian hasil klasifikasi dari setiap pohon diakumulasikan dan dipilih hasil klasifikasi yang paling banyak muncul [7].

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Data

Data yang akan digunakan dalam penerapan algoritma *Random Forest* terdiri dari beberapa variabel penting yang berhubungan dengan kondisi jaringan. Berikut adalah penjelasan mengenai setiap variabel:

1. Waktu

Menunjukkan timestamp atau waktu pengambilan data untuk memahami pola kemacetan jaringan pada waktu tertentu.

2. User_aktif

Jumlah pengguna aktif yang terhubung ke jaringan pada waktu tertentu. Variabel ini dapat memberikan indikasi langsung tentang beban jaringan, di mana semakin banyak pengguna aktif, semakin besar kemungkinan terjadinya kemacetan.

3. Jitter

Jitter atau variasi kedatangan yang diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir. Mengukur variasi dalam waktu pengiriman paket data. *Jitter* berhubungan erat dengan *delay* yang terjadi pada transmisi jaringan [8]. Pada tabel berikut menunjukkan jitter menurut standar TIPHONE

Tabel 1. Indeks Jitter TIPHONE

Kategori Degradasi	Peak Jitter	Indeks
Sangat Bagus	0 ms	4
Bagus	>0 s/d 75 ms	3
Sedang	>75 s/d 125 ms	2
Buruk	>125 s/d 225 ms	1

4. Traffic

Mengukur jumlah data yang ditransfer dalam jaringan pada waktu tertentu, biasanya dalam satuan kilobit per detik (Kbps). *Traffic* yang tinggi sering kali berhubungan dengan kemacetan jaringan.

5. Kepadatan

Kategori yang menunjukkan tingkat kemacetan jaringan, yang dapat diklasifikasikan sebagai Rendah, Sedang, atau Tinggi. Ini adalah variabel target yang ingin diprediksi oleh model *Random Forest*.

3.2. Penerapan *Random Forest* dalam Prediksi Kemacetan Lalu Lintas Jaringan

1. Pra Pemrosesan Data

Sebelum data dimasukkan kedalam model maka yang harus dilakukan adalah melakukan pembersihan data. Tujuan dari pembersihan data adalah agar data yang akan dimasukkan kedalam model sesuai [9], sebelum membangun model, data perlu dipersiapkan melalui beberapa langkah pra-pemrosesan yaitu sebagai berikut :

- a. Siapkan pustaka Python untuk analisis data, visualisasi, dan pembelajaran mesin. *Pandas* digunakan untuk mengolah data dalam bentuk tabel, sedangkan *Matplotlib* membantu memvisualisasikan data. Dengan *Scikit-learn*, dataset dibagi menjadi data pelatihan dan pengujian, kemudian model *Random Forest* dilatih untuk melakukan klasifikasi. Evaluasi model dilakukan dengan metrik seperti precision, recall, dan confusion matrix. Terakhir, modul *Google Colab* memungkinkan pengelolaan file untuk mengunggah atau mengunduh data saat menggunakan *Google Colab*.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from google.colab import files
```

Gambar 4. Import Library yang dibutuhkan

- b. Membaca Data dengan menggunakan *Pandas* untuk membaca file CSV yang berisi data.

```
# 1. Unggah dan Baca Data
uploaded = files.upload()
file_name = list(uploaded.keys())[0]
data = pd.read_csv(file_name)
```

Gambar 5. Program Membaca file CSV

- c. Menghapus Kolom yang Tidak Diperlukan atau tidak relevan untuk analisis.
- d. Menangani Nilai yang Hilang (*Missing Value*) yaitu Mengisi atau menghapus nilai yang hilang untuk memastikan integritas data
- e. Mengonversi Variabel Kategorika target (Kepadatan) menjadi format numerik jika diperlukan. Konversi ini sangat penting karena memungkinkan model pembelajaran mesin untuk memahami dan memproses data dengan lebih efektif. Dengan menggunakan *LabelEncoder*, dapat dengan mudah mengubah label kategori menjadi angka, sehingga mempersiapkan data untuk analisis lebih lanjut dan pelatihan model.

```
# Konversi label kategori menjadi numerik
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

Gambar 6. Program Konversi Label Kategori Menjadi Numerik

2. Pembagian Data

Setelah itu melakukan pra-pemrosesan data, langkah selanjutnya adalah membagi data menjadi dua bagian utama: set pelatihan (training set) dan set pengujian (test set),

- a. `test_size=0.3`: Menentukan bahwa 30% dari data akan digunakan sebagai set pengujian.
- b. `random_state=42`: Menetapkan seed untuk pengacakan, sehingga pembagian data dapat direproduksi di lain waktu.

```
# 3. Membagi Data
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.3, random_state=42)
```

Gambar 7. Program Konversi Label Kategori Menjadi Numerik

3. Membuat Model Random Forest

Dengan mengatur parameter yang tepat dan melatih model pada data pelatihan, kita dapat membangun model yang mampu memprediksi kemacetan jaringan dengan akurasi yang baik. Proses ini memanfaatkan kekuatan *ensemble* dari *Random Forest* untuk meningkatkan kinerja klasifikasi dibandingkan dengan menggunakan satu pohon keputusan saja.

Selama pelatihan, setiap pohon keputusan dalam Random Forest akan:

- a. Mengambil sampel acak dari data pelatihan (dikenal sebagai bootstrap sampling).
- b. Membangun pohon keputusan berdasarkan sampel tersebut, di mana pada setiap node, hanya subset acak dari fitur yang dipertimbangkan untuk pemisahan.
- c. Mengulangi Proses, mengulangi langkah ini untuk jumlah pohon yang ditentukan (dalam hal ini, 100 pohon).

Setelah model dilatih, ia akan dapat digunakan untuk memprediksi label pada data baru berdasarkan pola yang telah dipelajari dari data pelatihan.

```
# 4. Membuat Model Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

Gambar 8. Program Membuat model Random Forest

4. Evaluasi Model

Dengan menggunakan metode `predict`, kita dapat menghasilkan prediksi untuk data pengujian dan menyimpan hasilnya dalam variabel `y_pred`. Langkah ini memungkinkan untuk menilai seberapa baik model yang telah dilatih dapat menggeneralisasi dan memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya


```
# 5. Evaluasi Model
y_pred = rf_model.predict(X_test)
```

Gambar 9. Program Evaluasi Model

5. Laporan Klasifikasi (Classification Report)
Matrics Confusion Multiclass adalah tabel yang sering digunakan untuk menggambarkan kinerja model klasifikasi pada sutau set data testing yang nilai-nilai yang sebenarnya sudah diketahui [10], kita dapat melihat dengan jelas di mana model melakukan prediksi yang benar dan salah untuk setiap kelas. Laporan klasifikasi memberikan metrik yang lebih mendalam untuk menilai kinerja model, termasuk *precision*, *recall*, dan *f1-score* untuk setiap kelas, dimana *Accuracy* merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data, *Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif, *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif dan *F1 Score* merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan.[11] Informasi ini sangat berguna untuk memahami kekuatan dan kelemahan model, serta untuk melakukan perbaikan jika diperlukan.

```
# Confusion Matrix dan Classification Report
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
# Dapatkan label string, bukan label numerik
target_names = [str(cls) for cls in label_encoder.classes_]
print(classification_report(y_test, y_pred, target_names=target_names))
```

Gambar 10. Program Classification Report

6. Mengubah label numerik kembali ke kategori
 Dengan menggunakan metode *inverse_transform* dari *LabelEncoder*, kita dapat mengonversi label numerik kembali ke kategori aslinya, sehingga memudahkan interpretasi hasil dan analisis kinerja model. Proses ini membantu memastikan bahwa kita dapat memahami dan menjelaskan hasil prediksi model dengan cara yang lebih intuitif dan informatif.
7. Membuat DataFrame untuk visualisasi
 Dengan membuat DataFrame yang berisi label asli dan label yang diprediksi, kita dapat mempersiapkan data untuk analisis dan visualisasi lebih lanjut. Struktur data ini memungkinkan kita untuk dengan mudah membandingkan hasil prediksi model dengan data asli, sehingga memberikan wawasan yang lebih baik tentang kinerja model dalam memprediksi kemacetan jaringan.

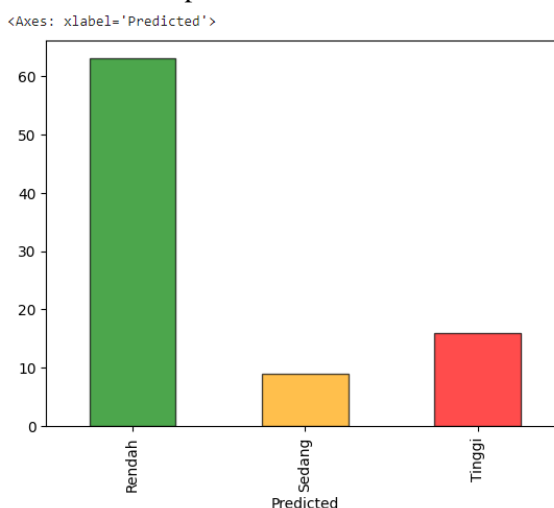
```
# Membuat DataFrame untuk visualisasi
visual_df = pd.DataFrame({
    'Actual': y_test_labels,
    'Predicted': y_pred_labels
})
```

Gambar 11. Potongan Program DataFrame visualisasi

8. Visualisasi Kesimpulan Hasil Prediksi

Proses visualisasi kesimpulan kepadatan jaringan menggunakan Matplotlib. Dengan membuat figure dan menentukan ukurannya, kita mempersiapkan area di mana semua elemen visualisasi akan digambar. Ini membantu memastikan bahwa visualisasi yang dihasilkan akan terorganisir, informatif, dan mudah dibaca, yang sangat penting untuk analisis dan presentasi data.

Dengan menggunakan grafik batang, kita dapat dengan jelas melihat jumlah prediksi untuk setiap kategori kemacetan jaringan. Pengaturan warna, transparansi, dan warna tepi membantu meningkatkan keterbacaan dan estetika grafik, sehingga memudahkan analisis dan interpretasi hasil.



Gambar 12. Hasil Visualisasi Prediksi Kemacetan Jaringan

4. KESIMPULAN

Penelitian ini bertujuan untuk memprediksi kemacetan lalu lintas jaringan menggunakan algoritma *Random Forest* berbasis data MikroTik. Berikut adalah kesimpulan yang dapat diambil:

1. Penelitian ini menunjukkan bahwa algoritma *Random Forest* efektif untuk memprediksi kemacetan jaringan dengan memanfaatkan data dari perangkat MikroTik dan Wireshark.
2. Dengan menggunakan fitur seperti jumlah pengguna aktif, jitter, dan traffic, model ini mampu memberikan prediksi terkait kondisi kemacetan jaringan.
3. Hasil evaluasi menunjukkan bahwa algoritma *Random Forest* layak diterapkan dalam manajemen jaringan untuk memantau dan mengelola kemacetan.
4. Faktor utama yang memengaruhi kemacetan jaringan adalah jumlah pengguna aktif yang tinggi, yang menyebabkan peningkatan beban pada jaringan dan Jitter yang signifikan, yang mencerminkan ketidakstabilan waktu pengiriman paket data.
5. Algoritma *Random Forest* memiliki potensi besar untuk digunakan sebagai alat prediksi dalam sistem monitoring jaringan berbasis *real-time*, membantu pengelola jaringan membuat keputusan yang lebih baik berdasarkan data.

5. SARAN

Berdasarkan hasil penelitian, disarankan agar institusi mengintegrasikan model prediksi ini ke dalam sistem monitoring jaringan berbasis web atau cloud untuk memungkinkan pemantauan kondisi jaringan secara *real-time*. Langkah ini akan membantu administrator jaringan dalam mengantisipasi dan mengatasi kemacetan dengan lebih cepat dan efisien. Selain itu, penelitian lanjutan sebaiknya memperluas cakupan data dari berbagai jenis jaringan untuk menguji generalisasi model serta menambahkan parameter tambahan seperti *latency* dan *packet loss* guna meningkatkan akurasi prediksi. Peneliti selanjutnya juga dapat mengeksplorasi algoritma lain, seperti *Deep Learning*, untuk membandingkan performa dan memberikan solusi yang lebih optimal dalam manajemen jaringan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang ikut berkontribusi dalam melakukan penyusunan penelitian ini dan kepada pihak penerbit yang telah menerbitkan penelitian ini.

DAFTAR PUSTAKA

- [1] H. Maulana and S. L. Mufreni, "Prediksi Kemacetan Pada Jaringan Komputer Menggunakan Metode Naive Bayesian Classifier," *Pros. Semin. Nas. Inform. Bela Negara*, vol. 2, no. L, pp. 41–43, 2021, doi: 10.33005/santika.v2i0.92.
- [2] H. Maulana and S. L. Mufreni, "Prediksi Kemacetan Pada Jaringan Komputer Menggunakan Metode Naive Bayesian Classifier," *Pros. Semin. Nas. Inform. Bela Negara*, vol. 2, no. 1, pp. 41–43, 2021, doi: 10.33005/santika.v2i0.92.
- [3] M. P. M. Routing, F. S. Mukti, A. Basuki, and O. Setyawati, "Pengendalian Kemacetan Jaringan," *J. EECCIS*, vol. 12, no. 1, pp. 15–19, 2018, [Online]. Available: <https://jurnaleeccis.ub.ac.id/index.php/eccis/article/view/465>
- [4] S. S. Nurashila, F. Hamami, and T. F. Kusumasari, "Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 8, no. 3, pp. 864–877, 2023, doi: 10.29100/jipi.v8i3.3961.
- [5] T. M. Diansyah, "Analisa Pencegahan Aktivitas Ilegal Didalam Jaringan Menggunakan Wireshark," *J. TIMES*, vol. 4, no. 2, pp. 20–23, 2016, doi: 10.51351/jtm.4.2.2015229.
- [6] N. Asyifah and D. Ramayanti, "Optimasi Kinerja Jaringan Di Smk Al Fudhola Bekasi: Pengaturan Bandwidth Dengan Mikrotik Rb 951ui-2hnd Dan Penerapan Algoritma Simple Queue," *J. Ilm. Ilk. - Ilmu Komput. Inform.*, vol. 7, no. 1, pp. 33–46, 2024, doi: 10.47324/ilkominfo.v7i1.210.
- [7] Kurniawan Fransiska Amalia, "Analisis Dan Implementasi Random Forest dan Classification dan Regression Tree (CART) untuk Klasifikasi pada Misuse Intrusion Detection System," *Univ. Telkom*, 2011.

- [8] I. K. S. Satwika, "Analisis Quality of Service Jaringan Virtual Private Network (Vpn) Di Stmik Stikom Indonesia," *J. Ilm. Inform.*, vol. 7, no. 01, p. 60, 2019, doi: 10.33884/jif.v7i01.1016.
- [9] R. Nisa, S. Amriza, and D. Supriyadi, "Komparasi Metode," *Penelit. Ilmu dan Teknol. Komput.*, vol. 13, no. 2, pp. 130–139, 2021.
- [10] V. W. Siburian and I. E. Mulyana, "Prediksi Harga Ponsel Menggunakan Metode Random Forest," *Annu. Res. Semin. 2018*, vol. 4, no. 1, pp. 144–147, 2018.
- [11] N. Sari, F. Panjaitan, D. Palembang, J. Jendral Ahmad Yani No, and P. Kota Palembang, "Penerapan Quality of Service Dalam Menganalisis Kualitas Kinerja Metode Simple Queue Dan Queue Tree," *J. Jupiter*, vol. 15, no. 1, pp. 570–576, 2023.