

KLASIFIKASI RAMBU LALU LINTAS MENGGUNAKAN DECISION TREE J48 DAN LOCAL BINARY PATTERN

Kevin Tanuwidjaya^{a1}, Ericson^{a2}, Lukman Hakim^{a3}

^aJurusan Informatika, Universitas Bunda Mulia, Indonesia

e-mail: ¹kevin.tanuwidjaya@gmail.com, ²ericsonwen123@gmail.com, ³lhakim2710@gmail.com

Abstrak

Teknologi kemudi otomatis atau autopilot pada mobil sedang berkembang dengan cepat. Dengan fitur ini pengemudi dimudahkan karena mobil dapat berjalan sesuai arahan program. Pengemudi tentu masih dapat mengambil kendali mobil secara manual sesuai keinginan, sehingga tidak menutup kemungkinan pengemudi melanggar rambu lalu lintas baik disengaja maupun tidak. Penelitian ini berusaha untuk membuat sebuah sistem pengenalan rambu lalu-lintas yang dapat membantu mengurangi pelanggaran yang dilakukan secara sadar oleh pengemudi. Pengujian dilakukan menggunakan gabungan algoritma Local Binary Pattern sebagai ekstraksi ciri dan algoritma Decision Tree J48 sebagai sistem klasifikasi untuk mengenali rambu lalu lintas. Diharapkan penelitian ini dapat memberikan gambaran tentang kemampuan gabungan dua algoritma ini dalam mengklasifikasi jenis rambu lalu lintas.

Kata kunci: Klasifikasi, J48, Local Binary Pattern, Rambu Lalu Lintas

Abstract

Automatic steering technology or autopilot in cars is developing rapidly. This feature makes it easier for the driver because the car can run according to the program directions. The driver of course can still take control of the car manually as desired, so it is possible for the driver to violate traffic signs, whether intentionally or not. This study seeks to create a traffic sign recognition system that can help reduce violations committed by drivers knowingly. The test is carried out using a combination of the Local Binary Pattern algorithm as feature extraction and Decision Tree J48 algorithm as a classification system to recognize traffic signs. It is hoped that this research can provide an overview of the combined ability of these two algorithms in classifying types of traffic signs.

Keywords: Classification, J48, Local Binary Pattern, Traffic Sign

1. Introduction

Computer vision adalah bidang kecerdasan buatan yang melatih komputer menafsirkan dan memahami dunia visual. Menggunakan gambar digital dari kamera dan video serta model pembelajaran mendalam, mesin-mesin dapat dengan akurat mengidentifikasi dan mengklasifikasikan objek, kemudian bereaksi terhadap apa yang dilihatnya.

Berdasarkan UU no 14 tahun 1992 pasal 8a, rambu lalu lintas adalah salah satu perlengkapan jalan untuk keselamatan, keamanan, ketertiban dan kelancaran lalu lintas serta kemudahan bagi pemakai jalan. Rambu lalu lintas terbagi menjadi beberapa kelompok yaitu rambu peringatan, rambu petunjuk, rambu larangan, dan rambu perintah. Rambu lalu lintas jalan merupakan alat pengendali lalu lintas untuk menyampaikan informasi berupa peringatan, larangan, perintah, atau petunjuk dengan tujuan untuk menjaga keamanan, ketertiban, kelancaran dan kenyamanan bagi pengguna jalan.

Penelitian ini diharapkan dapat menjadi dasar pengembangan aplikasi pendukung dari teknologi autopilot atau kendaraan yang dapat berjalan secara otomatis. Sistem akan memberikan informasi yang didapatkan sesuai dengan rambu-rambu yang terdapat di jalan. Informasi ini akan membantu AI autopilot untuk mengetahui kondisi yang ada di jalan dan memberikan keputusan yang lebih akurat. Dengan adanya teknologi ini, diharapkan juga dapat membantu dalam meminimalisir angka pelanggaran yang disebabkan oleh pengemudi.

Dalam melakukan pengenalan terhadap rambu lalu lintas, terdapat proses ekstraksi ciri dan klasifikasi. Pada kedua proses tersebut, terdapat beberapa algoritma yang dapat digunakan. Pada penelitian ini, penulis menggunakan algoritma Local Binary Pattern untuk melakukan ekstraksi ciri dan algoritma J48 untuk melakukan klasifikasi. Melalui penelitian ini, penulis akan melakukan implementasi algoritma Local Binary Pattern dan algoritma J48 untuk pengenalan rambu lalu lintas menggunakan computer vision.

2. Research Method / Proposed Method

Dalam membangun sistem dan menjalankan penelitian, pengerjaannya terbagi menjadi beberapa tahapan metodologi. Berikut kami paparkan tahapan pengerjaan penelitian ini:

- a. Analisa dan Perancangan Sistem
Pada tahap ini, kami melakukan analisis terhadap pokok masalah dari penelitian yang akan dilakukan. Kami juga membuat gambaran rancangan sistem dalam bentuk *activity diagram* dan *sequence diagram* untuk memudahkan pengembangan sistem yang diperlukan.
- a. Studi Literatur
Pada tahap ini, kami berusaha mencari informasi-informasi yang berhubungan dengan subjek penelitian. Informasi-informasi tersebut kami peroleh dari berbagai jenis artikel yang kesahihannya dapat dipertanggung jawabkan.
- b. Pengembangan Sistem
Setelah merancang sistem yang akan dibangun serta memiliki informasi yang cukup, pada tahap ini kami mulai mengembangkan aplikasi sesuai dengan rancangan yang ada. Aplikasi dikembangkan menggunakan bahasa pemrograman Python 3 pada sistem operasi Windows 10. Pengembangan aplikasi juga memanfaatkan library OpenCv dan python-weka-wrapper untuk membantu mempercepat pengerjaan.
- c. Pengujian Terhadap Metode
Pada tahap ini, aplikasi yang sudah selesai dikembangkan akan digunakan oleh peneliti untuk melakukan pengujian metode. Hasil dari pengujian yang dilakukan akan disediakan oleh aplikasi melalui output terminal.
- d. Analisa Hasil Pengujian
Pada tahap terakhir, peneliti akan melakukan analisis terhadap hasil uji yang dihasilkan aplikasi. Melalui data uji tersebut, peneliti dapat mengetahui keakuratan sistem dalam mengenali gambar rambu lalu lintas.

3. Literature Study

3.1 Rambu Lalu Lintas

Rambu lalu lintas dapat diartikan sebagai suatu lambang, huruf, kalimat, angka, atau gabungan dari ketiganya sebagai pemberi arahan, larangan, maupun petunjuk kepada pengguna jalan. Di Indonesia, peraturan yang mengatur rambu lalu lintas adalah Peraturan Menhub No. 13 Tahun 2014.[3]

Berdasarkan pesan yang disampaikan, rambu lalu lintas dibedakan menjadi:[3]

- Rambu peringatan
- Rambu ajar
- Rambu larangan
- Rambu perintah

3.2 Citra

Citra adalah nilai dari intensitas cahaya yang biasanya diterapkan kedalam bidang dua dimensi. Citra dapat digolongkan berdasarkan bentuk dari sinyal yang menyusun citra tersebut, diantaranya:[4]

- Citra Analog
Citra ini terbentuk dari alat akuisisi citra analog. Contoh citra analog antara lain citra yang tertangkap mata manusia dan foto atau video yang ditangkap menggunakan kamera analog. Meskipun memiliki kualitas gambar yang sangat baik, citra analog akan sulit hingga tidak dapat disimpan maupun diolah menggunakan komputer.
- Citra Digital
Citra yang terbentuk dari kumpulan *pixel* (*picture element*). *Pixel* merepresentasikan nilai-nilai cahaya serta memiliki titik koordinat (x, y) dan amplitudo f(x, y), dimana titik koordinat menyimpan informasi letak *pixel*, dan amplitudo menyimpan informasi intensitas cahaya pada *pixel*

3.3 Pengolahan Citra Digital

Cabang ilmu yang fokus mempelajari cara citra terbentuk, diolah, serta dianalisis untuk menghasilkan informasi yang mudah dipahami oleh manusia, adalah pengertian dari pengolahan citra digital.[4]

3.4 Computer Vision

Computer Vision merupakan pemanfaatan data optis berupa citra digital oleh komputer untuk memperoleh suatu informasi dari data optis tersebut. Dalam *Computer Vision*, komputer memiliki tugas untuk dapat memperoleh, menganalisis, memproses, hingga memahami suatu gambar digital, dan ekstraksi data dimensi tinggi dari dunia nyata. Hasil dari proses-proses yang dilakukan tersebut adalah informasi numerik atau simbolis, contohnya seperti sebuah keputusan [2].

3.5 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) merupakan suatu metode yang digunakan untuk menganalisis suatu tekstur dengan menggunakan model statistika dan struktur. Proses pengolahan *Local Binary Pattern* dapat dijelaskan menggunakan persamaan berikut [1]:

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(i_n - i_c)2^n$$

Gambar 1. Rumus persamaan *Local Binary Pattern*

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

Gambar 2. Rumus persamaan *thresholding Local Binary Pattern*

Pertama-tama, terjadi proses pengurutan *pixel* ketetangaan (i_n) dengan *pixel* tengah (i_c). Operator dasar yang digunakan LBP berukuran 3 x 3 dan menggunakan 8 *pixel* ketetangaan dari *pixel* tengah. Selanjutnya, *pixel* ketetangaan ke-n akan diolah menggunakan proses *thresholding* dengan menggunakan nilai keabuan dari *pixel* tengah menggunakan persamaan pada gambar 2. Terakhir, nilai dari piksel ketetangaan tersebut akan disusun berlawanan dengan arah jarum jam dan 8 bit biner tersebut dikonversi menjadi nilai desimal untuk menggantikan nilai piksel tengah.

Kelebihan dari menggunakan *Local Binary Pattern* adalah kemudahannya untuk diimplementasikan juga tingkat komputasinya yang rendah, sehingga proses ekstraksi fitur dapat dilakukan dalam waktu yang relatif cepat [1].

3.6 Decision Tree J48

Algoritma *decision tree* J48 merupakan implementasi dari algoritma C4.5 yang terdapat pada Weka. Algoritma C4.5 sendiri adalah algoritma yang digunakan untuk mengklasifikasikan objek-objek [5]. Proses algoritma C4.5 dalam membangun pohon keputusan antara lain [5]:

1. Memilih atribut yang akan dijadikan sebagai akar
2. Membuat cabang untuk setiap nilai
3. Membagi case dalam cabang

4. Ulangi tahapan proses untuk setiap cabang hingga semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih satu atribut yang dijadikan sebagai akar, perlu mencari nilai gain tertinggi yang terdapat dari atribut-atribut yang ada [5]. Berikut adalah persamaannya:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n * Entropy(S_i)$$

Gambar 3. Rumus persamaan pemilihan atribut sebagai akar *decision tree* J48

Keterangan:

S = himpunan kasus

A = atribut

n = jumlah partisi atribut A

|Si| = jumlah kasus pada partisi ke-i

|S| = jumlah kasus dalam S

Berikut adalah rumus dasar dari entropy:

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i$$

Gambar 4. Rumus dasar entropy *decision tree* J48

Keterangan:

S = himpunan kasus

A = fitur

n = jumlah partisi S

pi = proporsi dari Si terhadap S

4. Result and Discussion

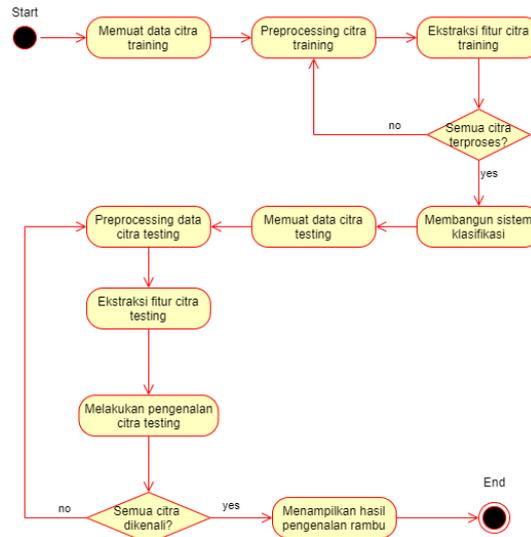
4.1 Perancangan Sistem

Untuk melakukan *preprocessing* ekstraksi fitur daripada citra, kami menggunakan bantuan library openCv. Untuk melakukan proses klasifikasi dan pengenalan citra, kami menggunakan bantuan aplikasi Weka yang berjalan pada environment Java. Dalam menghubungkan antara Python 3 dengan Java sendiri, kami menggunakan bantuan dari JVM yang dapat dijalankan melalui command python sehingga kami hanya perlu menggunakan bahasa Python 3 untuk menjalankan keseluruhan aplikasi.

Setelah aplikasi selesai melakukan processing terhadap citra yang diberikan, aplikasi akan menampilkan hasil-hasil pengerjaan yang dilakukan seperti jenis rambu yang terdeteksi dan tingkat akurasi di *terminal line* menggunakan bahasa yang dapat dimengerti oleh manusia.

Agar lebih jelas, berikut adalah diagram yang menampilkan rancangan cara kerja aplikasi kami:

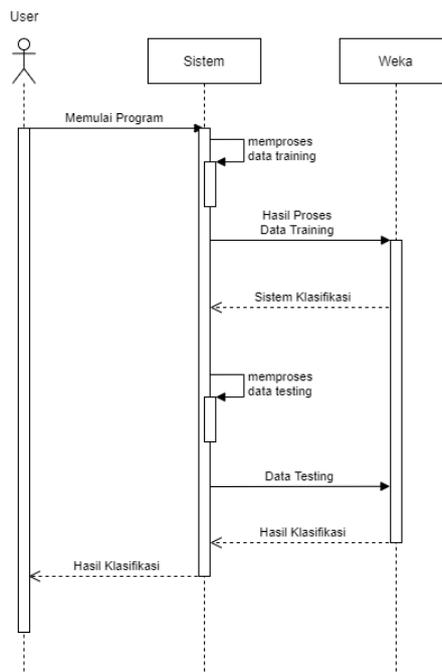
1. Activity Diagram



Gambar 5. Activity Diagram Proses Klasifikasi Rambu Lalu Lintas

Gambar 5 merupakan activity diagram yang menampilkan cara kerja dari aplikasi kami. Berdasarkan gambar activity diagram, proses sistem dimulai dari memuat data citra training untuk preprocessing citra training. Setelah itu, data citra training yang sudah disiapkan akan dilakukan ekstraksi fitur menggunakan algoritma Local Binary Pattern. Ketika semua data citra training yang disiapkan telah diproses, sistem akan membangun sistem klasifikasi yang menggunakan algoritma J48. Sistem klasifikasi ini akan menggunakan hasil ekstraksi ciri algoritma Local Binary Pattern yang didapatkan dari data citra training. Setelah sistem klasifikasi telah dibangun, selanjutnya sistem akan memuat data citra testing untuk preprocessing data citra training. Data citra testing yang sudah disiapkan akan dilakukan ekstraksi ciri terlebih dahulu menggunakan algoritma Local Binary Pattern. Hasil dari ekstraksi ciri akan digunakan untuk pengenalan citra testing melalui sistem klasifikasi yang telah dibuat. Setelah semua citra telah diproses, sistem akan menampilkan hasil pengenalan rambu sebagai aktivitas terakhir.

2. Sequence Diagram



Gambar 6. Sequence Diagram Proses Klasifikasi Rambu Lalu Lintas

Gambar 6 merupakan sequence diagram yang menampilkan interaksi antara user, sistem, dan Weka. Berdasarkan gambar sequence diagram, proses dimulai dari user memulai program. Setelah user memulai program, sistem akan memproses data training dan mengirimkan hasil proses data training ke Weka. Kemudian hasil data training yang diterima dari sistem, Weka akan membangun sistem klasifikasi yang kemudian akan dikirimkan ke sistem. Setelah itu, sistem akan memproses data testing dan mengirimkan hasil proses data testing ke Weka. Weka akan mengklasifikasi data testing sesuai dengan sistem klasifikasi sehingga menghasilkan hasil klasifikasi. Hasil klasifikasi tersebut akan dikirimkan ke sistem yang kemudian diteruskan sistem ke user.

4.2 Hasil Pengujian

Berikut ini merupakan hasil dari pengujian yang dilakukan.

Tabel 1. Hasil Pengujian Klasifikasi Rambu Lalu Lintas

	<i>Data Testing A</i>	<i>Data Testing B</i>	<i>Data Testing C</i>
<i>Correctly Classified Instances</i>	14	4	0
<i>Incorrectly Classified Instances</i>	2	0	4
<i>Total Number of Instances</i>	16	4	4

Pada tabel di atas, terdapat tiga jenis data yang digunakan sebagai data testing dengan hasil uji yang berbeda-beda, sebagai berikut:

1. Data Testing A

Data yang digunakan pada pengujian ini adalah data yang sama persis dengan data yang digunakan untuk melakukan training. Pada pengujian ini, terdapat 16 data yang digunakan sebagai data testing. Berdasarkan pengujian ini, dapat diketahui bahwa terdapat 14 data yang berhasil diklasifikasikan dan 2 data yang gagal diklasifikasikan dari total 16 data. Yang berarti tingkat akurasi dari 87,5% untuk tipe data testing A.

2. Data Testing B

Data yang digunakan pada pengujian ini adalah data yang mirip dengan data yang digunakan untuk melakukan training. Pada pengujian ini, terdapat 4 data yang digunakan sebagai data testing. Berdasarkan pengujian ini, dapat diketahui bahwa terdapat 4 data yang berhasil diklasifikasikan dan 0 data yang gagal diklasifikasikan dari total 4 data. Yang berarti tingkat akurasi dari 100% untuk tipe data testing B.

3. Data Testing C

Data yang digunakan pada pengujian ini adalah data yang berbeda dengan data yang digunakan untuk melakukan training. Pada pengujian ini, terdapat 4 data yang digunakan sebagai data testing. Berdasarkan pengujian ini, dapat diketahui bahwa terdapat 0 data yang berhasil diklasifikasikan dan 4 data yang gagal diklasifikasikan dari total 4 data. Yang berarti tingkat akurasi dari 0% untuk tipe data testing C.

5. Conclusion

Penerapan algoritma Local Binary Pattern dan algoritma J48 dalam pengenalan rambu lalu lintas menghasilkan 87,5% tingkat akurasi untuk penggunaan data testing yang sama dengan data training, 100% tingkat akurasi untuk data testing yang mirip dengan data training, dan 0% tingkat akurasi untuk data testing yang berbeda dengan data training

References

- [1] Amat, R., Sari, J. Y., & Ningrum, I. P. (2017). Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada Smartphone. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 15(2), 152.
 - [2] Davies, E. R. Computer Vision: Principles, Algorithms, Applications, Learning. In: *Computer Vision: Principles, Algorithms, Applications, Learning*. 5th ed. USA. Academic Press, 2017: hal. 1–13.
 - [3] Hakim, M. F. T. (2018). *RANCANG BANGUN SISTEM DETEKSI RAMBU – RAMBU LALU LINTAS MENGGUNAKAN JARINGAN SYARAF*. Institut Teknologi Sepuluh Nopember.
 - [4] <https://pemrogramanmatlab.com/2017/07/26/pengolahan-citra-digital/>, diakses tanggal 6 Oktober 2021.
 - [5] Imron, M., Perdanawanti, L., Informasi, S., & AMIKOM Purwokerto Jl Letjend Pol Sumarto, S. (2017). Algoritma Decision Tree-J48, K-Nearest, Dan Zero-R Pada Kinerja Akademik. *Seminar Nasional Teknologi Informasi*, 12–18.
-