

PERANCANGAN APLIKASI DETEKSI KEMACETAN BERDASARKAN AUDIO PROCESSING MENGGUNAKAN METODE ZERO CROSSING RATE DAN AVERAGE ENERGY BERBASIS ANDROID

(Design of The Congestion Detection Application Based on Audio Processing Using Zero Crossing Rate and Average Energy Based on Android)

Putu Cinthia Wikessa¹, Dr. Ir. Bambang Hidayat, DEA², Ratri Dwi Atmaja, S.,T M.T.³

¹²³Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹cwikessa@gmail.com

²bhidavadat@telkomuniversity.co.id

³ratriidwiatmaja@telkomuniversity.co.id

Abstrak

Pemrosesan digital seringkali dimanfaatkan untuk menganalisis suatu hal yang bersifat audio, visual, ataupun video digital. Para *engineer* banyak mengembangkan sistem yang menggunakan pemrosesan digital seperti *intelligent transport system* (ITS) atau aplikasi *monitoring system* kemacetan di kota metropolitan. Kemacetan lalu lintas menjadi masalah utama bagi pengguna jalan, untuk itu sangat dibutuhkan sebuah aplikasi yang dapat memberikan informasi kondisi kepadatan lalu lintas di suatu ruas jalan agar pengguna jalan dapat terhindar dari kemacetan.

Pada tugas akhir ini dibuat suatu aplikasi berbasis android yang dapat mengolah audio yang direkam menggunakan *handphone*. Secara umum, prinsip kerja sistem ini adalah dengan membandingkan sampel pada *database* dengan sampel baru menggunakan metode *zero crossing rate* dan *average energy*. Sampel yang nantinya akan diklasifikasi sehingga dapat menentukan kondisi lalu lintas. Proses merekam suara dengan menggunakan *audio recorder* pada *handphone* seluler berbasis android. Android adalah sistem operasi yang menyediakan *platform* terbuka sehingga dapat digunakan oleh bermacam peranti penggerak.

Pada Tugas Akhir ini telah berhasil menstimulasikan sistem yang mampu mengklasifikasikan kondisi jalan macet dan lancar. Nilai akurasi yang didapat sebesar 73,33% menggunakan metode klasifikasi *nearest-neighbor*.

Kata Kunci : kepadatan lalu lintas, *zero crossing rate*, *average energy*, android

Abstract

Digital processing is often used to analyze some things that are audio, visual, or digital video. The engineers developed a system that uses a lot of digital processing such as *intelligent transport system* (ITS) or application *monitoring system* congestion in the metropolis. Traffic congestion a major problem for road users, it is very necessary for an application that can provide information on the condition of traffic in a road section so that road users can avoid the congestion.

In this final project created an android based application that can process audio recorded using a mobile phone. In general, the working principle of this system is to compare samples in the database with new samples using *zero crossing rate* and *average energy*. Samples will be classified so it can determine traffic conditions. Proses recording sound with the *audio recorder* on a mobile phone based on Android. Android is an operating system that provides an open platform that can be used by various propulsion devices.

In this final project has been successfully stimulating system that is able to classify the traffic conditions and smoothly. Values obtained accuracy is 73,33% using *nearest-neighbor* classification method.

Keywords : traffic density, *zero crossing rate*, *average energy*, android

1. Pendahuluan

Intelligent Transportation System adalah upaya dalam pengembangan informasi dan teknologi informasi dalam infrastruktur transportasi dengan mengelola kejadian yang terjadi pada lalu lintas jalan seperti kepadatan kendaraan, kecelakaan yang terjadi, dan rute alternatif untuk mengurangi kemacetan, sehingga dapat meningkatkan waktu perjalanan, mengurangi polusi udara serta bahan bakar minyak (BBM). *Intelligent Transportation System* dibangun dari beberapa subsistem, seperti pengolahan video, audio, *database* sistem dan sebagainya [1]. Dengan adanya ITS, diharapkan dapat memberikan solusi bagi *traffic* lalu lintas disuatu perkotaan. Dengan memanfaatkan salah satu subsistem pada ITS seperti audio, dengan sinyal audio digital kita dapat memproses subsistem tersebut yang dapat menggambarkan kondisi kepadatan jalan pada saat itu. Sehingga, kondisi kepadatan lalu lintas pada suatu tempat dapat diketahui oleh pengendara dan pihak berwajib, yang dapat membantu dalam pengaturan lalu lintas terutama pada saat *peak hour*.

Pada penelitian sebelumnya, telah dilakukan pendeteksian kepadatan lalu lintas yang dapat diakses oleh *user* melalui *sms gateway*. Namun, kondisi saat ini telah berubah, dan fitur sms sudah mulai ditinggalkan. Masyarakat, terutama *user* lebih tertarik menggunakan suatu aplikasi pada *smartphone*, atas dasar ini dibuatlah sebuah sistem deteksi kemacetan yang dapat diakses via aplikasi android. Pada tugas akhir ini, aplikasi ITS akan dibangun dari dua subsistem. Sistem yang pertama akan menggunakan subsistem pengolahan audio. Pada subsistem ini, audio yang telah direkam akan dibagi kedalam beberapa *frame*, yang selanjutnya akan dilakukan perhitungan pada tiap *frame* untuk penentuan status pada waktu tersebut di ruas jalan. Keputusan yang diambil nantinya ada dua kategori, yaitu macet dan lancar. Pada subsistem yang kedua, akan digunakan subsistem *database*. Subsistem ini akan menyimpan hasil dari pengolahan sinyal subsistem audio yang dapat diakses *user* melalui aplikasi pada android. Diharapkan dengan adanya aplikasi ini dapat membantu memberikan informasi kepadatan lalu lintas yang mudah untuk diakses oleh para pengguna jalan.

2. Dasar Teori

2.1 Intelligent Transportation System

ITS merupakan salah satu upaya pengembangan teknologi informasi untuk meningkatkan fungsi infrastruktur transportasi secara menyeluruh. Sistem ini mampu memberikan informasi *real-time* yang terjadi pada lalu lintas jalan seperti kepadatan kendaraan, kecelakaan yang terjadi, dan rute alternatif untuk mengurangi kemacetan. Secara umum, teknologi ITS yang telah berkembang di dunia terdiri dari [1]:

1. *Advance Navigation System/Advanced Traveller Information System*
2. *Advance Traffic Management System*
3. *Incident Management System*
4. *Advance for Save driving*
5. *Advanced Bus Information System*

2.1 Pengolahan Suara

2.2.1 Sinyal Audio

Audio diartikan sebagai suara atau reproduksi suara. Suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya dan membentuk sebuah pola. Pola osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola-sama yang berulang pada interval tertentu, yang disebut sebagai periode. Karena suara adalah gelombang, maka seperti halnya gelombang lain, ia akan memiliki amplitudo dan juga frekuensi. Amplitudo akan menentukan intensitas suara, frekuensi akan menentukan tinggi rendahnya nada. Manusia mendengar bunyi pada gelombang bunyi, yaitu getaran di udara atau medium lain, sampai ke gendang telinga manusia. Batas frekuensi bunyi yang dapat didengar oleh telinga manusia berkisar antara 20 Hz sampai 20 kHz.[2]

2.2.2 Sampling

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu atau untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan *sample*. [3] Proses mengubah gelombang suara kontinyu ke dalam bentuk waktu disebut *sampling*, yang menghasilkan representasi digital sinyal suara.

2.2.3 Bit Depth

Kedalaman bit biasanya dinyatakan dalam bit per sampel. *Bit depth* atau kedalaman bit menyatakan jumlah bit yang diperlukan untuk merepresentasikan tiap sampel atau jumlah tingkatan level suara. Audio 8 bit menyediakan 2 pangkat delapan atau 256 level. Audio 16 bit menyediakan 65.536 level dan audio 32 bit memiliki jumlah jangkauan 2 pangkat 32. Semakin banyak jumlah bit yang digunakan untuk merepresentasikan sebuah citra, maka semakin baik kualitas citra tersebut. Begitu pula dengan ukuran *file* yang juga semakin besar.[4]

2.2.4 Sample Rate

Sample rate menyatakan banyaknya sampel yang direkam tiap 1 detik. Semakin banyak *frame* yang bergerak tiap detiknya maka kualitas suara yang diterima semakin akurat. *Sample rate* dinyatakan dalam *sample per second*. Agar suara dapat didengar oleh telinga manusia, sebaiknya tidak merekam suara pada *sample rate* dibawah 44.100 Hz karena formula dari "*Frekuensi Nyquist*" yang menunjukkan bahwa bandwidth audio dari sinyal sampel dibatasi setengah dari *sampling rate*. Jadi untuk mencakup rentang sekitar 20.000 Hz untuk pendengaran manusia, peralatan harus *sampling* pada sampel lebih dari 40.000 (40k) per detik.[5]

2.3 Framing

Framing adalah pembagian sinyal audio menjadi beberapa *frame* yang nantinya dapat memudahkan dalam perhitungan dan analisa sinyal, satu *frame* terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi *sampling*-nya.[6]

Pada proses *framing*, sinyal audio dibagi menjadi *frame-frame* N sampel, dengan *frame-frame* berdekatan dengan spasi M ($M < N$). *Frame* pertama terdiri dari N sampel pertama. Proses ini berlanjut hingga semua sinyal suara dihitung dalam satu atau banyak *frame*. [7]

2.4 Overlapping

Untuk menjaga kualitas sinyal audio, maka diperlukannya proses *overlapping* pada sinyal yang sudah dibagi kedalam bentuk *frame*. Proses ini diawali dengan membandingkan panjang data yang akan di-*overlapping*, yang kemudian dilanjutkan dengan mencari mana yang paling pendek dari keduanya yang nantinya akan dijadikan sebagai patokan dalam penentuan banyak data yang akan di proses.[6]

Frame pertama yang berisi N-sampel kemudian dilanjutkan dengan *frame* kedua dengan M sampel setelah *frame* pertama, dan *overlap* dengan N-M sampel. Dengan cara yang sama, *frame* ketiga dimulai 2M sampel setelah *frame* pertama (atau M sampel setelah *frame* kedua) dan *overlap* dengan N- 2M sampel. [8]

2.5 Zero Crossing Rate (ZCR)

Zero crossing adalah suatu keadaan di mana suatu fungsi menyentuh titik nol atau saat dimana suatu fungsi berpindah dari nilai positif ke negatif. Metode ini telah banyak digunakan dalam pengenalan suara maupun informasi jenis musik. Dalam konteks sinyal diskrit, *zero crossing* dapat terjadi jika sampel mempunyai tanda aljabar yang berbeda. Ini berarti bahwa ada hubungan antara *zero crossing* dan konten frekuensi sinyal. Formula *zero crossing rate* adalah sebagai berikut :

$$z_c = \frac{1}{T} \sum_{n=1}^{T-1} II\{s(n) \cdot s(n-1) < 0\} \tag{2.1}$$

dimana s adalah sinyal dari panjang T dan fungsi indikator II{A} akan bernilai 1 jika A adalah benar, dan bernilai 0 jika A adalah salah.[9]

2.6 Average Energy

Average Energy adalah suatu parameter atau ratio untuk mendeteksi kenyaringan suatu sinyal audio. *Average Energy* dapat dihitung dengan *mean-square value*, yaitu[10]:

$$E = 1/N \sum_{n=0}^{N-1} x(n)^2 \tag{2.2}$$

Dimana N adalah jumlah sampel per unit waktu, E adalah *average energy* dari sinyal x(n), dan n adalah sinyal ke - n.

2.7 Nearest - Neighbor

Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. [14]

2.6 Android

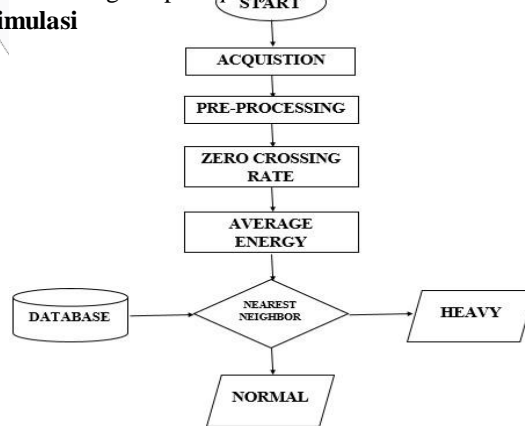
Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi Android dapat dikembangkan pada sistem operasi Windows (XP Vista/Seven, dll), Mac OS X (Mac OS X 10.4.8 atau lebih baru) dan Linux. Android dipuji sebagai “*platform mobile* pertama yang lengkap, terbuka, dan bebas” [11].

Lengkap (*Complete Platform*) : Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.

Terbuka (*Open Source Platform*) : Pengembang dapat dengan bebas untuk mengembangkan aplikasi.

Bebas (*Free Platform*): Android adalah platform atau aplikasi yang bebas untuk *developed*. Tidak ada lisensi atau biaya royalti, biaya keanggotaan untuk dikembangkan pada *platform* Android.

3. Perancangan Sistem dan Simulasi



Gambar 3.1 Gambar Diagram Alir Sistem

3.1 Akuisisi Data

Akuisisi data merupakan tahapan yang menggambarkan bagaimana cara perekaman suara. Proses perekaman suara dilakukan dengan menggunakan media telepon genggam jenis Samsung Galaxy Mega2 8GB. Perekaman suara

dilakukan sebanyak 180 kali pada 1 titik lokasi dengan kondisi jalan acak, yaitu pada saat jalanan macet, ramai dan lancar. Masing-masing rekaman tersebut memiliki panjang kira-kira 10 detik, 20 detik, dan 30 detik dengan jarak posisi pengambilan suara yang sama di tiap lokasi. Frekuensi *sampling* yang digunakan adalah 44100 Hz. Pada tugas akhir ini data suara dibagi menjadi dua bagian yaitu 90 rekaman *pre-recorded* sebagai data suara latih dan 90 rekaman *real-time* sebagai data suara uji. Pada data suara latih, data suara tersebut akan diproses untuk mendapatkan ciri latih.

3.2 Pre Processing

3.2.1 Framing

Pada proses ini sinyal audio yang dibagi menjadi sejumlah frekuensi dalam range tertentu atau kedalam beberapa frame yang diinginkan. Panjang frame yang digunakan untuk pemrosesan sinyal adalah 20 ms. Karena jika panjang frame terlalu kecil, informasi yang di dapat tidak cukup akurat. Sedangkan jika ukuran frame terlalu besar maka informasi yang dibawa akan berubah-ubah. Panjang *frame* yang digunakan akan mempengaruhi keberhasilan dalam analisa spektral.

3.2.2 Overlapping

Proses *overlapping* diawali dengan membandingkan panjang data yang akan di-*overlapping*, yang kemudian dilanjutkan dengan mencari mana yang paling pendek dari keduanya yang nantinya akan dijadikan sebagai patokan dalam penentuan banyak data yang akan di proses. Pada proses ini, nilai *overlapping* yang digunakan sebesar 25% karena dari percobaan sebelumnya untuk *audio processing* nilai *overlapping* sebesar 25% memiliki nilai akurasi lebih baik dibandingkan dengan nilai *overlapp* 50% atau 75%. Disebutkan bahwa semakin kecil nilai *overlapping* yang digunakan maka akan semakin baik hasil dari sebuah identifikasi [15].

3.3 Ekstraksi Ciri Dengan Zero Crossing Rate

Zero crossing rate digunakan karena ZCR merupakan salah satu jenis metode yang bekerja pada domain waktu. Pada konteks pewaktu sinyal diskrit, *zero crossing* terjadi ketika sampel sebelumnya memiliki tanda aljabar yang berbeda dengan sampel saat ini. Implementasi dari metode ZCR ini akan dilakukan menggunakan aplikasi Android. Pada proses ini, hasil *pre-processing* akan diolah untuk mendapatkan ciri dari sebuah suara. Dari perhitungan ZCR ini nantinya akan didapat nilai *sample rate* dari tiap sinyal suara yang akan diuji. Selanjutnya, nilai *sample rate* ini akan dibandingkan dengan nilai pada *database* untuk menentukan tingkat kepadatan lalu lintas.

3.4 Ekstraksi Ciri Dengan Average Energy

Perhitungan *average energy* dilakukan untuk menemukan sifat-sifat perbedaan dari kondisi jalan yang macet dan kondisi jalan yang lancar. Data audio yang sudah diambil akan melakukan *query* dengan data latih seracara berulang sehingga mendapatkan keterangan informasi dari data audio.

3.5 Klasifikasi Dengan Nearest-Neighbor

Hasil ekstrasi ciri digunakan untuk mengklasifikasikan kondisi ruas jalan berdasarkan *audio processing*, yaitu dengan membandingkan nilai *sample rate* data uji dengan nilai *sample rate* pada database. Klasifikasi audio ini pada dasarnya dikelompokkan menjadi 2 kelas, yaitu lancar dan macet. Pada tahap latih hasil dari ekstraksi ciri disimpan dalam sebuah *database*. Hasil dari ciri uji dibandingkan dengan *database* ciri latih kemudian diklasifikasikan menggunakan metode *nearest-neighbor*. Parameter yang diubah adalah panjang data audio yang di-*set* manual pada saat proses *recording*. Pada tahap klasifikasi, hasil ciri uji ekstraksi ciri akan dihitung jaraknya berdasarkan jarak terdekatnya hasil ciri latih. Jika hasil ciri uji mendekati nilai ciri latih pada *database* maka data uji tersebut bernilai benar. Ketika ciri uji bernilai benar, maka akan masuk sesuai dengan kelas yang disediakan yaitu macet dan lancar.

3.6 Parameter Pengujian Sistem

Perhitungan dalam akurasi sistem yang telah dibuat dibutuhkan untuk mengetahui seberapa tingkat akurasi dari sistem pendeteksi kemacetan ini yang telah dibuat. Tingkat akurasi sistem (TAS) merupakan nilai dari ketepatan suatu sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang sesuai dengan tujuan pembuatan sistem tersebut. Perhitungan dari TAS sebagai berikut:

$$ACC = \frac{TP + TN}{P + N} \quad (3.1)$$

Error menyatakan nilai selisih absolut antara jumlah sampel benar yang dihasilkan oleh sistem dengan jumlah *frame event* aktual. Sedangkan jumlah *frame system* aktual menyatakan jumlah frame jika dihitung secara manual oleh kasat mata (telinga). Secara sistematis perhitungan *Error* dapat dituliskan sebagai berikut:

$$Error = 100\% - \text{Tingkat akurasi sistem} \quad (3.2)$$

3.7 Proses Penanaman Software ke Android

Pada proses ini akan dibuat aplikasi deteksi kemacetan dengan data audio sebagai masukan pada sistem android. Setelah aplikasi pendeteksi kemacetan dinyatakan layak dan memenuhi syarat, maka akan dilakukan pemindahan aplikasi tersebut menjadi aplikasi berbasis android yang nantinya dapat digunakan oleh *user* sebagai sarana bantu dalam perjalanan mereka.

3.8 Interface Aplikasi

Langkah terakhir dari aplikasi ini adalah menampilkan *output* hasil pendeteksian berupa *user interface* yang akan menjadi daya tarik tersendiri bagi *user* dalam pemakaian aplikasi ini. Tahapan ini merupakan bagian krusial dalam suatu aplikasi *smartphone* terutama android, karena dengan memiliki *user interface* yang bagus, *user* akan semakin senang menggunakan aplikasi pendeteksi kemacetan ini.

4. Analisis Dan Hasil Keluaran Sistem

4.1 Skenario Pengujian Sistem

Pengujian sistem dilakukan dengan mengubah parameter waktu dan pengujian nilai *overlapping* untuk mendapatkan tingkat akurasi yang terbaik. Pada pengujian ini terdapat 4 skenario pengujian. Skenario pengujian pertama dengan merubah parameter nilai *overlapping*, yaitu pengujian sistem dengan menggunakan nilai overlapp sebesar 25%, 50%, dan 75% dengan inputan sebanyak 20 data pada kondisi jalan macet dan lancar. Tiga skenario pengujian selanjutnya adalah dengan mengubah parameter waktu, yaitu pengujian sistem dengan menggunakan panjang data 10 detik, 20 detik, dan 30 detik. Proses *recording* dilakukan secara *real time* sebanyak 90 kali, dimana tiap kelas panjang data memiliki 15 rekaman pada kondisi jalan macet dan 15 rekaman pada kondisi jalan lancar. Tujuan dari pengujian ini adalah untuk mengetahui berapa panjang data audio dan besarnya nilai *overlapping* yang akan menghasilkan tingkat akurasi terbaik.

4.4.1 Hasil Pengujian Skenario ke-1

Pada pengujian skenario pertama dilakukan dengan mengubah nilai *overlapping* sebesar 25%, 50%, dan 75%. Pengujian dilakukan menggunakan aplikasi matlab sebagai *software* tambahan untuk menguji besarnya nilai *overlapping* yang akan diimplementasikan pada sistem android. Pengujian pertama dilakukan dengan menggunakan input audio pada *database* dengan panjang data 10 detik pada kondisi jalan lancar sebanyak 10 kali. Dilanjutkan dengan pengujian dengan data input dengan panjang data yang sama saat kondisi macet sebanyak 10 kali. Setelah pengujian tersebut kemudian dilakukan perhitungan rata-rata dari masing – masing parameter yang dihasilkan, dan dapatkan nilai nilai sebagai berikut:

Pengujian Dengan Input Lancar			
	25%	50%	75%
Overlapp	221	441	662
Waktu Komputasi	0,0648	0,0715	0,0863

(a)

Pengujian Dengan Input Macet			
	25%	50%	75%
Overlapp	221	441	662
Waktu Komputasi	0,0256	0,0276	0,0530

(b)

Tabel 4. 1 Hasil Pengujian Skenario 1 (a) data lancar (b) data macet

Dari Tabel 4. 1 dapat dilihat jika besarnya persentase nilai *overlapping* yang digunakan akan mempengaruhi nilai *overlapp* dan waktu komputasi yang dihasilkan. Dan pengujian dengan dua kondisi input yang berbeda akan menghasilkan waktu komputasi yang berbeda pula, namun tidak berpengaruh pada nilai *overlapp* yang dihasilkan.

4.4.2 Hasil Pengujian Skenario ke-2

Pada skenario pengujian kedua pengujian dilakukan dengan menggunakan aplikasi berbasis android yang sudah terinstal pada *mobile phone*. Pada sistem pendeteksian ini, besarnya nilai *overlapping* yang digunakan sebesar 25% dengan panjang data sebesar 10 detik. Data uji diambil sebanyak 30 data, dimana 15 data diambil saat jalanan macet dan 15 data diambil saat jalanan lancar. Berikut merupakan tabel data benar pada pengujian skenario kedua:

Panjang Data 10s		
	Heavy	Normal
Heavy	13	2
Normal	6	9

Tabel 4. 2 Hasil pengujian skenario 2

4.4.3 Hasil Pengujian Skenario ke-3

Pada pengujian ketiga pengambilan data uji dialukan sebanyak 30 kali menggunakan aplikasi android dengan nilai *overlapp* sebesar 25%, dimana 15 data merupakan data jalanan macet dan 15 data pada jalanan lancar. Panjang data yang digunakan adalah 20 detik. Berikut tabel data benar pada skenario pengujian kedua:

Panjang Data 20s		
	Heavy	Normal
Heavy	11	4
Normal	6	9

Tabel 4. 3 Hasil Pengujian Skenario 3

4.4.4 Hasil Pengujian Skenario ke-4

Pada pengujian terakhir dilakukan dengan menggunakan panjang data sebesar 30 detik. Banyak data dan cara pengambilan data sama dengan pada skenario pertama dan kedua. Berikut tabel data benar pada skenario keempat:

Panjang Data 30s		
	Heavy	Normal
Heavy	10	5
Normal	9	6

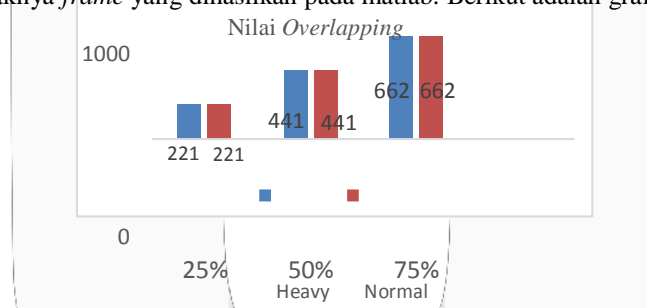
Tabel 4. 4 Hasil Pengujian Skenario 4

4.5 Analisa Hasil Pengujian Sistem

Sistem akan dianalisa berdasarkan *software* yang digunakan. Berikut adalah hasil analisa dari pengujian sistem:

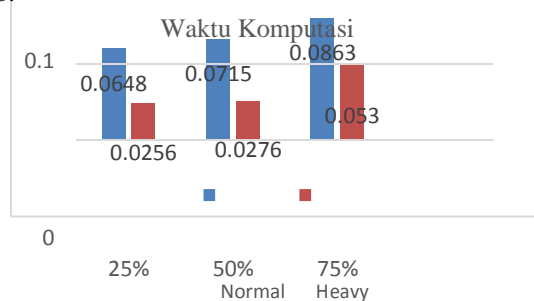
4.5.1 Analisa Pengujian Sistem Pada Matlab

Dari pengujian pertama dapat diketahui bahwa besarnya nilai *overlapping* berpengaruh terhadap nilai *overlapp*, waktu komputasi dan banyaknya *frame* yang dihasilkan pada matlab. Berikut adalah grafik dari keluaran sistem:



Gambar 4. 1 Grafik Nilai *Overlapp*

Dari Gambar 4. 1 dapat diketahui bahwa nilai *overlapping* yang paling baik digunakan adalah sebesar 25%. Karena semakin kecil nilai *overlapp*-nya maka semakin baik hasil dari sebuah identifikasi. Berbeda dengan besarnya nilai *overlapping*, kondisi jalan saat macet atau lancar tidak berpengaruh pada nilai *overlapping*, karena durasi yang digunakan sama yaitu 10 detik jadi akan menghasilkan nilai *overlapping* yang sama pula. Berikut adalah hasil waktu komputasi pada pengujian matlab:



Gambar 4. 2 Grafik Waktu Komputasi

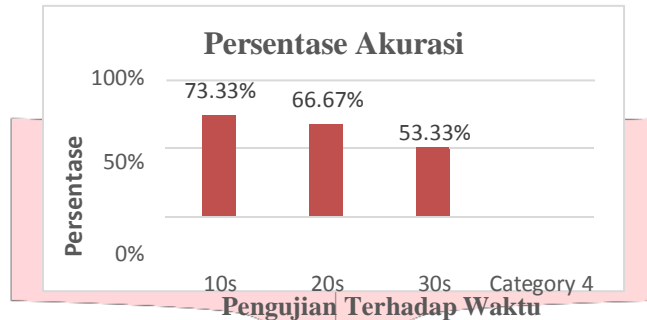
Dari Gambar 4. 2 dapat disimpulkan pada kondisi macet atau lancar, waktu komputasi paling cepat adalah pada nilai *overlapping* sebesar 25% yaitu sebesar 0,0648 sekon saat kondisi jalan normal dan 0,0256 sekon saat kondisi jalan macet. Hal ini akan sangat berpengaruh pada lamanya proses setelah diimplementasikan pada android. Dari kedua hasil pengujian pada matlab, dapat disimpulkan bahwa nilai *overlapping* terbaik untuk *audio processing* menggunakan metode *zero crossing rate* dan *average energy* adalah sebesar 25%.

4.5.2 Hasil Pengujian Sistem Pada Android

Dari pengujian dengan mengubah parameter waktu dapat diketahui bahwa durasi atau panjang data dapat mempengaruhi jumlah data yang benar. Perbedaan jumlah data benar pada pengujian, akan berpengaruh terhadap tingkat akurasi pada sistem yang dibuat. Berikut adalah tabel data benar dari pengujian sistem:

	10s	20s	30s
Heavy	13	11	10
Normal	9	9	6

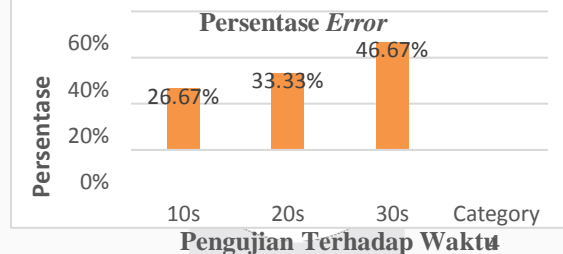
Tabel 4. 5 Data Benar Sistem



Gambar 4. 3 Persentase Akurasi Tiap Detik

Gambar 4. 2 menunjukkan bahwa dengan panjang data atau durasi 10 detik, tingkat akurasi yang didapatkan lebih tinggi yaitu sebesar 73,33%, dibandingkan dengan menggunakan panjang data 20 detik sebesar 66,67% dan 30 detik sebesar 53,33%, dimana tingkat akurasi ini di dapat dari perhitungan jumlah data benar pada panjang data 10 detik yaitu sebanyak 21 data benar dari 30 kali percobaan.

Dengan panjang data 10 detik, ciri dari data yang menunjukkan kondisi jalan sudah didapat serta variasi ciri dan noise yang ada pada ruas jalan tidak terlalu banyak. Sehingga lebih mudah saat proses klasifikasinya. Sedangkan pada panjang data 20 detik dan 30 detik, terdapat banyak noise yang membuat proses pengklasifikasian semakin sulit, sehingga mempengaruhi tingkat akurasi sistem.



Gambar 4. 4 Persentase Error tiap detik

Gambar 4. 4 menunjukkan nilai *error* pada sistem yang telah dibuat. Dimana persentase nilai *error* terkecil pada durasi 10 detik dengan persentase nilai *error* sebesar 26,67%.

5. Kesimpulan Dan Saran

1.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan pada proses deteksi kemacetan berdasarkan *audio processing* dengan menggunakan metode *zero crossing rate* dan *average energy* sebagai ekstraksi ciri, maka dapat diambil kesimpulan sebagai berikut:

1. Metode *zero crossing rate* dan *average energy* sebagai ekstraksi ciri dan *nearest-neighbor* sebagai klasifikasi dapat digunakan untuk membuat sistem pendeteksi ruas jalan dengan keluaran macet dan lancar
2. Panjang data atau durasi rekaman berpengaruh terhadap akurasi sistem. Panjang data audio pada ruas jalan yang menghasilkan tingkat akurasi terbaik dan dapat diimplementasikan pada perangkat *mobile* adalah data audio dengan panjang durasi 10 detik
3. Perancangan sistem pengklasifikasian kondisi jalanan macet dan lancar menghasilkan nilai akurasi terbaik sebesar 73,33% menggunakan metode pengklasifikasian *nearest-neighbor*.

1.2 Saran

Berikut adalah saran yang diberikan untuk dapat memperbaiki kekurangan – kekurangan yang ada pada penelitian ini guna pengembangan penelitian lebih lanjut:

1. Sistem dapat dikembangkan dengan menggunakan metode ekstraksi ciri dan klasifikasi lain yang dapat diimplementasikan pada sistem android sehingga mendapatkan tingkat performansi yang lebih baik
2. Mengubah parameter pengujian pada ekstraksi ciri dan klasifikasi
3. Menambah jumlah data latih dan data uji sistem yang dapat diklasifikasikan sehingga tingkat akurasi akan semakin baik
4. Menambah kelas pada keluaran dari klasifikasi sistem
5. Menambah proses filtering untuk membedakan suara pengujian dan noise yang ada
6. Perbaiki *interface* aplikasi sehingga lebih menarik untuk digunakan oleh *user*.

6. Referensi

- [1] Mandaku, Hanok dan Marcus Tukan. 2010. *Studi Penerapan Transportation System (ITS) di Kabupaten Serang Bagian Barat*. Arika.
- [2] Purnama, Agus. "Elektronika Dasar". 26 April 2016.
<http://elektronika-dasar.web.id/sinyal-audio-gelombang-suara/>
- [3] Anton. "Suara Dan Audio". 26 April 2016.
<http://lecturer.ukdw.ac.id/anton/download/multimedia3.pdf>
- [4] Sandha, Ryan. "Pengolahan Sinyal Digital". 26 April 2016.
<http://ryanzps.blogspot.co.id/2015/10/pengolah-sinyal-digital.html>
- [5] Yusuf, Dody Maulana. 2014. *Analysis and Implementation of Wavelet Daubechies Method in Mobile Based Guitar Tuning Application*. Bandung: UNIKOM
- [6] <http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2007-2-00247-IF-Bab%202.pdf>
- [7] Shete, Patil. 2014. *Zero Crossing Rate and Energy of the Speech Signal of Devanagari Script*. India: IOSR Journals.
- [8] Setiawan, Angga, dkk. 2011. *Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Kursor Komputer*. Semarang: UNDIP
- [9] Pradypta, Adhitya Ahmad. "Pengolahan Sinyal Digital Menggunakan Komponen dspFFT di Delphi 7". 27 April 2016.
<http://myblog.adhityapradyptha.id/2015/01/pengolahan-sinyal-digital-menggunakan.html>
- [10] https://en.wikipedia.org/wiki/Zero-crossing_rate#cite_note-2
- [11] Fuadi, Vanessa Catalina. 2013. *Implementasi dan Analisis Kondisi Mesin Mobil Berdasarkan Ciri Akustik Dengan Metode Transformasi Wavelet Pada Platfrom Android*. Bandung: Universitas Telkom.
- [12] Developer, Android. "Android Studio". 25 November 2016.
<https://developer.android.com/studio/index.html?hl=id>
- [13] Yani Lubis, D.S. 2010. *Perancangan dan Simulasi Sistem Informasi Lalu Lintas Berdasarkan Pengolahan Video Digital Dengan Metode Frame Difference Dan SMS Gateway*. Bandung: Institut Teknologi Telkom.
- [14] http://elearning.amikom.ac.id/index.php/download/materi/190302125-ST078-5/2011/10/20111020_Contoh.
- [15] Ibrahim, Nanda Rizki. 2016. *Perancangan Aplikasi Analisis Barcoding Gap Gen Matk dan RBCL Dari Tumbuhan Monokotol Dan Dikotil*. Banda Aceh: UNSYIAH.