

IMPLEMENTASI METODE *AUDIO FINGERPRINTING* PADA APLIKASI PENGENALAN LAGU BERBASIS SISTEM OPERASI ANDROID

Nanda Ariawan Putra^{*)}, Achmad Hidayatno dan Maman Somantri

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)E-mail:} ariawanputraaa@gmail.com

Abstrak

Musik selalu berkembang, hampir setiap hari muncul lagu baru dengan genre yang semakin beragam. Namun, jumlah lagu yang sangat banyak dapat menyulitkan penikmat musik untuk mengingat judul beberapa lagu yang disukainya. *Audio Fingerprinting* merupakan teknologi yang dapat memudahkan penikmat musik untuk mengetahui judul bermacam lagu tersebut yang mana metode ini mengekstrak ciri yang relevan dari sebuah konten audio. Oleh karena itu, dirancang sebuah aplikasi yang bernama 'Music and Match' berbasis sistem operasi Android yang memanfaatkan teknologi *Audio Fingerprinting* untuk membantu memberikan informasi dari lagu yang didengarkan melalui mikrofon ponsel berupa judul lagu dan informasi tentang penyanyinya. Penelitian ini akan membahas bagaimana aplikasi 'Music and Match' memproses lagu yang didengarkan, serta bagaimana aplikasi menampilkan informasi dari lagu tersebut. Aplikasi ini menggunakan Android Studio sebagai antarmuka utama, dan menggunakan GNSDK (*Gracenote Software Development Kit*) untuk memproses semua keperluan *Audio Fingerprinting*. Hasil implementasi menunjukkan aplikasi telah dibuat sesuai yang direncanakan. Berdasarkan hasil pengujian dengan variasi kondisi lagu saat didengarkan, bagian awal, tengah, dan akhir lagu memiliki rata-rata waktu pencocokan berturut-turut adalah 3,045 detik, 2,043 detik, dan 3,257 detik. Perbedaan bit rate tidak mempengaruhi keberhasilan pengenalan lagu dan presentase keberhasilan proses pencocokan lagu dengan 10 variasi amplitudo pada derau adalah 76%.

Kata Kunci: Musik, *Audio Fingerprinting*, Android, GNSDK

Abstract

Music is always developing, every day there are new songs released with more variant genres. However, the number of songs that already too much makes it difficult to remember all of the songs title. *Audio Fingerprinting* is the technology that can help to know those songs title easier. *Audio Fingerprinting* is a features extraction method of a piece of audio content. An android based application named Music and Match is designed using *Audio Fingerprinting* technology to provide information about the song, like song title and the artist, that listened through mobile phone's microphone. This research focus on how the app processing the song that listened, and how this app displays the information. The application uses Android Studio as interface, and GNSDK for processing all the *Audio Fingerprinting* components. The implementation shows that the application made in accordance with the design. Based on the testing results with various conditions of the songs that listened, the beginning, the middle, and the end of the song have average matching time respectively 3,045 s, 2,043 s, and 3,257 s. Bit rate difference doesn't affect the matching's success rate and the percentage of the matching's success rate with various amplitude on noise is 76%.

Keywords: Music, *Audio Fingerprinting*, Android, GNSDK

1. Pendahuluan

Musik selalu digemari masyarakat, apakah itu hanya untuk sekedar didengarkan maupun sebagai mata pencaharian. Musik selalu berkembang dari dulu hingga sekarang, dari masih menggunakan piringan hitam hingga saat ini yang dengan mudah didapatkan dan disimpan di telepon genggam. Terdapat banyak genre musik, seperti pop, rock, jazz, blues, klasik, hingga beberapa genre dari musik elektronik yang sedang digemari saat ini. Sudah banyak sekali lagu yang tercipta, hal ini membuat

semakin mudahnya lagu untuk didengarkan dimanapun dan kapanpun pada media yang beragam [1].

Seiring dengan perkembangan teknologi telepon genggam, terdapat aplikasi di perangkat *smartphone* dengan sistem operasi Android yang dapat memudahkan masyarakat untuk mengetahui judul lagu yang didengarkan. Salah satu teknologi yang dapat mewujudkan hal tersebut adalah *Audio Fingerprinting*.

Inti dari metode *Audio Fingerprinting* adalah mengekstraksi ciri atau karakteristik yang relevan dari sebuah konten audio [2][3].

Pada penelitian sebelumnya, telah dibuat aplikasi Android sebagai sarana edukatif panduan shalat [4], penjualan tanah [5], sistem kuliah *online* [6], dan persebaran objek wisata di Semarang [7]. Sedangkan untuk implementasi *Audio Fingerprinting*, telah diperkenalkan aplikasi untuk mendeteksi duplikasi lagu [8].

Pada penelitian ini dibuat aplikasi pengenalan lagu dengan nama ‘Music and Match’ berbasis Android, dengan menggunakan metode *Audio Fingerprinting* untuk keperluan aplikasi pengenalan lagu. Aplikasi ini diharapkan mampu membantu masyarakat untuk mengetahui judul sebuah lagu yang didengarnya secara lebih cepat dan efektif.

2. Metode

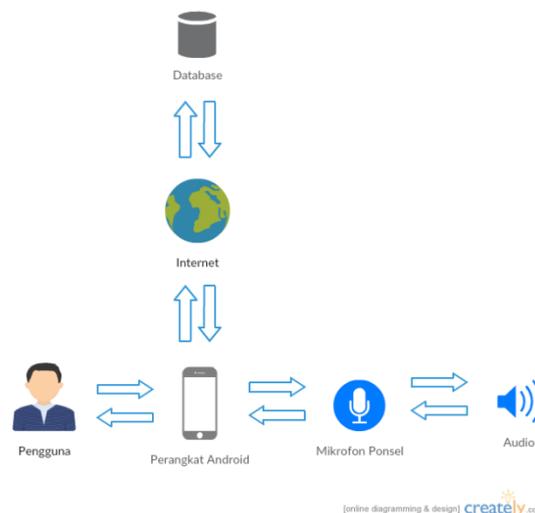
2.1. Deskripsi Sistem

Pada penelitian ini dirancang dan dibangun sebuah aplikasi yang diberi nama ‘Music and Match’ berbasis sistem operasi android. Konsep yang dibahas adalah bagaimana aplikasi ini mengubah masukan audio, yaitu sebagian dari lagu, menjadi informasi tentang lagu tersebut seperti judul dan artis yang membawakan lagu, semua informasi tersebut berasal dari *database*. Aplikasi ini menyajikan antarmuka yang dirancang melalui bantuan IDE Android Studio dengan minimal API level 15 [9] dan metode *audio fingerprinting* yang digunakan untuk mengidentifikasi lagu [10]. Pengolah metode *audio fingerprinting* yang digunakan pada aplikasi ini adalah GNSDK (*Gracenote Software Development Kit*) dan untuk *database* yang digunakan adalah *database* yang disediakan oleh Gracenote [11].

Aplikasi ‘Music and Match’ membutuhkan akses internet agar dapat mengakses *database* yang disediakan oleh Gracenote. *Database* ini berisi *audio fingerprint* dari berbagai lagu yang tersebar dari seluruh dunia. Aplikasi ini akan mengintegrasikan Android dengan metode *audio fingerprinting* melalui pemrograman Android (Java dan XML) untuk perancangan interface dan fungsionalitas aplikasi.

Pada gambar 1 menunjukkan bahwa terjadi pertukaran data antara perangkat Android dengan sisi *database*. Perangkat Android meminta informasi lagu kepada *database* dengan mengirimkan *audio fingerprint* sebagai *query* dari lagu yang didengarkan melalui mikrofon ponsel, saat lagu didengarkan melalui mikrofon, perangkat dapat menampilkan visualisasi audio sesuai dengan tempo dan volumenya. Selanjutnya *query* tersebut akan dicocokkan dengan *file audio fingerprint* lagu yang ada di *database*. Jika lagu yang didengarkan sebelumnya tidak sesuai, perangkat dapat membatalkan proses

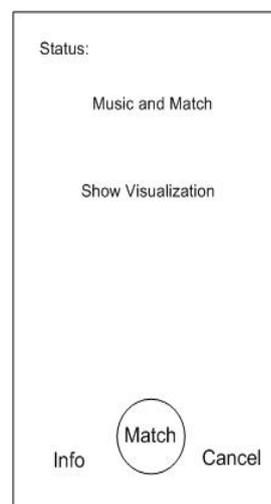
pencocokan sebelum selesai. Setelah proses pencocokan berhasil, perangkat Android akan menerima informasi lagu sesuai dengan audio yang didengarkan melalui mikrofon ponsel.



Gambar 1. Desain aplikasi

2.2. Desain Antarmuka Aplikasi

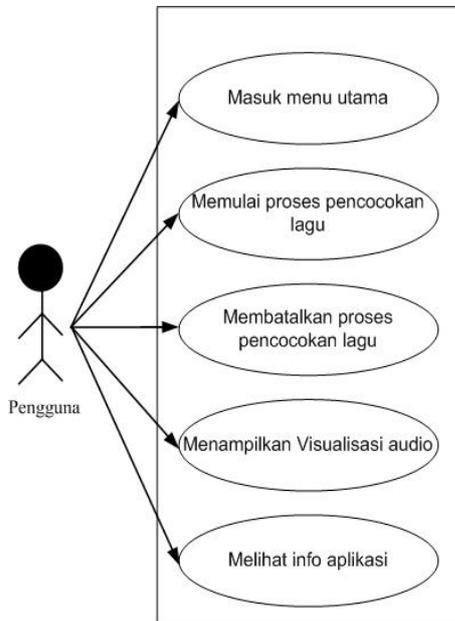
Desain aplikasi ‘Music and Match’, yang berfungsi sebagai antarmuka pengguna, menyajikan beberapa menu utama yang akan mengantarkan pengguna kepada berbagai fungsionalitas sistem. Pengguna dapat memilih 4 menu yang tersedia pada antarmuka menu utama. Pengguna dapat memilih menu *match* untuk memulai proses pencocokan lagu, menu *cancel* untuk membatalkan proses pencocokan lagu, menu *show visualization* untuk melihat visualisasi audio yang didengarkan, dan menu *info* untuk menampilkan informasi dari aplikasi ‘Music and Match’. Berikut merupakan rancangan antarmuka menu utama pada aplikasi.



Gambar 2. Rancangan antarmuka aplikasi

2.3. Diagram Use Case

Pada aplikasi ‘Music and Match’, diagram *Use Case* menggambarkan fungsi-fungsi yang ada pada sistem dan berfokus pada fitur-fitur sistem dari sudut pandang pihak luar, dalam hal ini pengguna.



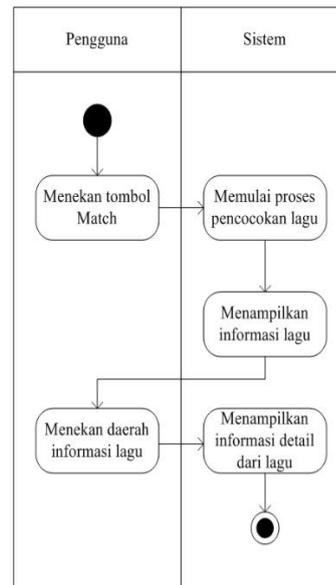
Gambar 3. Diagram use case pengguna

Gambar 3 menampilkan fitur-fitur yang dapat diakses oleh pengguna aplikasi ‘Music and Match’. Terdapat 4 menu, yaitu *Match*, *Cancel*, *Show Visualization*, dan *Info*. Pada menu *Match*, pengguna memulai proses pencocokan lagu yang nantinya setelah dikenali akan muncul info tentang lagu tersebut, seperti judul lagu, album, dan musisi yang membawakan lagu tersebut. Pada menu *Cancel*, pengguna dapat membatalkan proses pencocokan lagu sebelum lagu dikenali. Pada menu *Show Visualization*, pengguna dapat melihat visualisasi audio dari lagu yang didengarkan. Pada menu *Info*, pengguna dapat melihat informasi tentang aplikasi Music and Match.

2.4. Diagram Aktivitas

Diagram aktivitas menggambarkan logika prosedural dan aliran kerja dalam sistem yang dirancang. Diagram aktivitas juga dapat digunakan untuk menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas dibagi menjadi beberapa *swimlane object* untuk menggambarkan objek mana yang bertanggungjawab terhadap aktivitas tertentu.

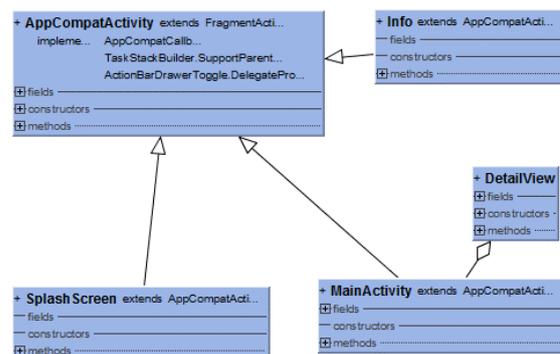
Pada gambar 4 ditampilkan salah satu contoh dari diagram aktivitas pada aplikasi Music and Match yaitu aktivitas *match*.



Gambar 4. Diagram aktivitas menu *match*

2.5. Diagram Kelas

Diagram kelas menggambarkan komponen-komponen pembentuk sistem yang berupa kelas atau interface dan hubungan antar komponen tersebut. Pada aplikasi ‘Music and Match’ berbasis Android, diagram kelas mencakup semua fungsi pada aplikasi yaitu menu *match*, *cancel*, *show visualization* dan *info*.

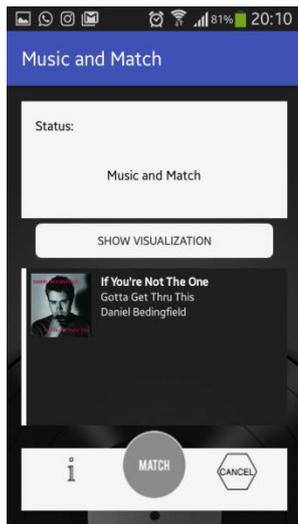


Gambar 5. Diagram kelas pada aplikasi Music and Match

3. Hasil dan Analisis

3.1. Implementasi

Pada aplikasi ‘Music and Match’ dilakukan implementasi hasil perancangan pada antarmuka sistem. Hal ini dilakukan untuk menerapkan sistem yang telah dibuat menjadi antarmuka aplikasi yang sesuai. Implementasi antarmuka sistem ini menggambarkan bagaimana program berjalan dari inialisasi sampai akhir. Gambar 6 berikut merupakan salah satu contoh dari hasil implementasi yang mana adalah tampilan setelah pencocokan lagu berhasil.



Gambar 6. Tampilan setelah pencocokan lagu berhasil

3.2. Pengujian Sistem

Tabel 1. Data pengujian pencocokan lagu

Uji Coba	Bentuk Pengujian	Hasil Pengujian	Status	Waktu (s)
1	Mengenali lagu Faded	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,1
2	Mengenali lagu Kisah Kasih di Sekolah	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	3,5
3	Mengenali lagu More Than Words	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,6
4	Mengenali lagu Cake By The Ocean	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,0
5	Mengenali lagu Jangan Ada Dusta di Antara Kita	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	4,0
6	Mengenali lagu Pergilah Kasih	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	3,1
7	Mengenali lagu Mungkinkah	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	3,2
8	Mengenali lagu Nothing's Gonna Stop Us	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,1
9	Mengenali lagu September	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,7
10	Mengenali lagu Goodbye	Muncul informasi lagu dan teks <i>Match Found</i>	Berhasil	2,2

Pada penelitian ini, pengujian dilakukan dengan melakukan pengujian alfa yaitu pengujian fungsionalitas dari aplikasi. Pengujian alfa bertujuan untuk identifikasi dan menghilangkan masalah sebelum akhirnya sampai ke pengguna yang sebenarnya. Pengujian alfa yang diterapkan pada aplikasi 'Music and Match' berbasis Android menggunakan model pengujian *blackbox*, yang mana merupakan metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi berdasarkan hasil eksekusi melalui data uji. Pengujian aplikasi Android ini

terdiri dari beberapa variabel hasil implementasi dari aplikasi, dimana pengujian ini dilakukan dalam sepuluh kali. Berikut ini merupakan hasil pengujian pencocokan lagu.

3.3. Pengujian Audio Fingerprinting

Pada pengujian ini dilakukan untuk mengetahui performa keandalan teknologi *Audio fingerprinting* yang digunakan. Pengujian ini dilakukan dengan menguji beberapa keadaan yang dapat mempengaruhi keberhasilan dari proses pencocokan lagu menggunakan aplikasi ini. Hal ini untuk mengetahui apakah aplikasi dapat digunakan dengan baik kapanpun dalam kondisi atau keadaan yang beragam.

Kondisi yang diuji pada penelitian ini adalah bagian lagu saat didengarkan, adanya derau saat lagu didengarkan, dan nilai *bit rate* dari berkas lagu yang didengarkan.

3.3.1. Pengaruh Bagian Lagu Terhadap Proses Pencocokan Lagu

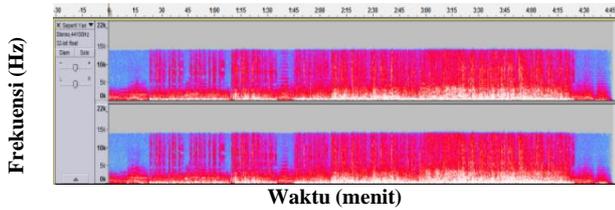
Pengujian ini dilakukan untuk mengetahui hubungan bagian lagu yang didengarkan dengan keberhasilan pencocokan lagu. Tujuan dari pengujian ini adalah untuk mengetahui apakah bagian lagu yang didengarkan akan mempengaruhi keberhasilan semua proses pencocokan lagu, atau hanya mempengaruhi sebagian, atau sama sekali tidak mempengaruhi proses pencocokan lagu.

Tabel 2. Hasil pengujian hubungan bagian lagu dengan keberhasilan pencocokan lagu

No.	Judul Lagu	Waktu Pencocokan (s)		
		Bagian Awal Lagu	Bagian Tengah Lagu	Bagian Akhir Lagu
1	Seperti Yang Kau Minta	3,78	2,58	3,68
2	Nothing's Gonna Stop Us Now	3,08	2,06	3,19
3	Mungkinkah	3,60	2,36	3,88
4	More Than Words	2,94	2,15	3,30
5	Kisah Kasih Di Sekolah	3,67	2,10	4,18
6	Faded	2,01	1,62	2,22
7	Pergilah Kasih	3,55	1,97	3,41
8	If You're Not The One	2,36	1,84	3,03
9	Jangan Ada Dusta Di Antara Kita	3,36	2,18	3,49
10	Accidentally In Love	2,10	1,57	2,18
	Rata-rata	3,045	2,043	3,257

Pada tabel 2 terlihat bahwa rata-rata waktu pencocokan untuk bagian awal lagu adalah 3,045 detik, untuk bagian tengah lagu adalah 2,043 detik, dan untuk bagian akhir lagu adalah 3,257 detik. Bagian tengah lagu memiliki rata-rata waktu pencocokan paling cepat, ini membuktikan bahwa bagian tengah lagu merupakan bagian lagu yang paling mudah untuk dikenali. Hal tersebut dikarenakan bagian tengah lagu memiliki frekuensi lebih tinggi dan stabil dibandingkan bagian lagu

lainnya. Untuk contoh, berikut merupakan gambar spektrogram dari lagu berjudul ‘Seperti yang Kau Minta’ yang dibawakan oleh Chrisye.



Gambar 7. Spektrogram dari lagu seperti yang kau minta

Seperti yang terlihat pada gambar 7, frekuensi pada bagian awal lagu cukup rendah dan belum stabil, tidak berbeda jauh dengan frekuensi bagian akhir lagu yang memiliki frekuensi yang rendah pula. Sedangkan untuk frekuensi pada bagian tengah lagu memiliki frekuensi yang tinggi dan stabil. Namun, hal tersebut tidak berarti membuat bagian awal dan akhir sama sekali tidak dapat dikenali oleh aplikasi ini, hanya membutuhkan waktu sedikit lebih lama untuk dikenali dibandingkan dengan bagian tengah lagu. Dengan demikian, bagian dari lagu yang didengarkan hanya mempengaruhi sebagian dari proses pencocokan lagu.

3.3.2. Pengaruh Bit Rate Terhadap Keberhasilan Pencocokan Lagu

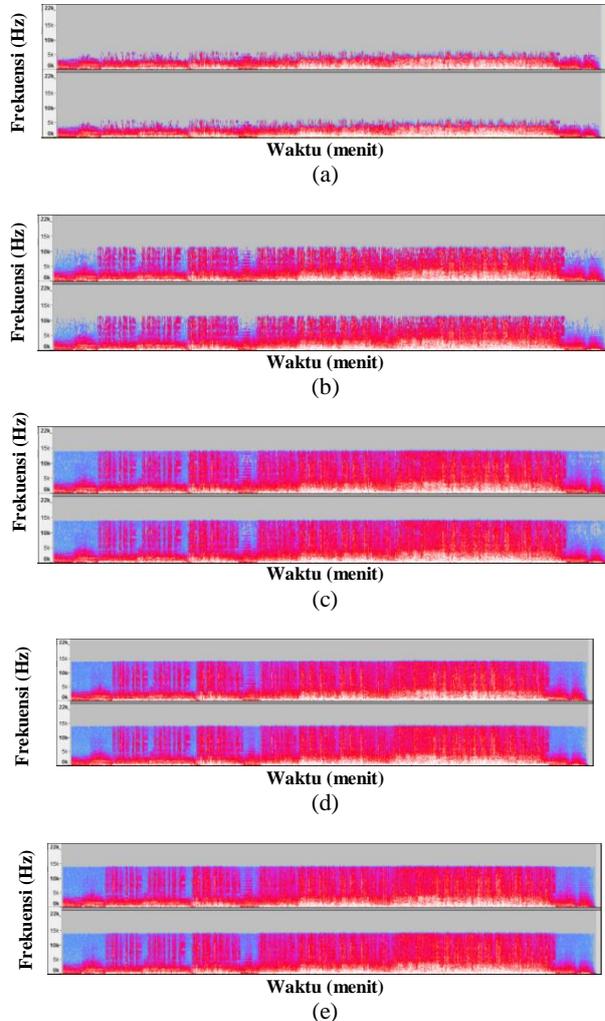
Pengujian ini dilakukan untuk mengetahui pengaruh perbedaan *bit rate* pada berkas MP3 yang digunakan saat proses pencocokan lagu menggunakan aplikasi ‘Music and Match’. Pada berkas MP3, *bit rate* merupakan ukuran dari data audio yang diproses dalam jumlah waktu tertentu. Semakin kecil *bit rate* pada berkas MP3 maka kualitas suaranya akan semakin menurun, sebaliknya, semakin besar *bit rate* pada berkas MP3 maka kualitas suaranya akan semakin baik. Pengujian ini akan menentukan apakah perbedaan kualitas suara dari berkas MP3 akan mempengaruhi keberhasilan pencocokan lagu.

Tabel 3. Hasil pengujian pengaruh *bit rate* terhadap keberhasilan pencocokan lagu

Judul lagu	Bit Rate	Keterangan
‘Seperti Yang Kau Minta’	32 kbps	Dikenali
	64 kbps	Dikenali
	96 kbps	Dikenali
	128 kbps	Dikenali
	192 kbps	Dikenali

Dari tabel 3 dapat dilihat bahwa lagu dengan 5 variasi *bit rate* yang berbeda, dari *bit rate* 32 kbps yang umumnya hanya digunakan untuk memperdengarkan ucapan, hingga *bit rate* 192 kbps yang memiliki kualitas suara yang baik, dapat dikenali menggunakan aplikasi Music and Match. Hal ini menunjukkan bahwa aplikasi dapat mengenali lagu meski dengan *bit rate* yang berbeda-beda. Aplikasi tetap dapat mengenali lagu meskipun kualitas suara dari

lagu yang didengarkan buruk karena amplitudo dari frekuensi lagu tersebut sebagian besar tetap sama sehingga lagu masih dapat dikenali dengan tepat. Berikut merupakan spektrogram dari lagu Seperti yang Kau Minta dengan 5 variasi *bit rate*.



Gambar 8. Spektrogram dari lagu ‘Seperti yang Kau Minta’ dengan 5 variasi *bit rate*
(a) 32 kbps (b) 64 kbps (c) 96 kbps
(d) 128 kbps (e) 192 kbps

Pada gambar 8 terlihat perbedaan yang jelas pada spektrogram berkas MP3 dengan *bit rate* 32 dan 64 kbps dibandingkan dengan spektrogram yang lainnya. Kedua spektrogram tersebut memiliki frekuensi yang lebih rendah yang disebabkan oleh proses pemampatan pada berkas MP3. Namun, terlihat pula bahwa amplitudo frekuensi dari spektrogram sebagian besar masih sama diantara kelima spektrogram, sehingga jumlah *peak* atau puncak pada spektrogram pun sebagian besar sama. Hal tersebut membuat aplikasi ‘Music and Match’ tetap dapat mengenali lagu sekalipun dengan kualitas keluaran suara yang buruk. Ini membuktikan bahwa perbedaan *bit rate* pada audio tidak mempengaruhi keberhasilan pencocokan lagu menggunakan aplikasi.

3.3.3 Ketahanan Terhadap Derau

Pengujian ini dilakukan untuk mengetahui ketahanan aplikasi ‘Music and Match’ terhadap derau yang dapat berupa suara televisi, atau suara lainnya yang bukan merupakan bagian dari lagu yang sedang didengarkan. Pengujian ini dilakukan menggunakan lagu yang digunakan sebelumnya, dengan diberikan 10 variasi derau untuk setiap lagunya. Derau yang digunakan pada pengujian ini merupakan *white noise*. Variasi yang digunakan pada pengujian ini adalah besarnya amplitudo pada *white noise*, yaitu dari 0,1 hingga 1. Berikut merupakan beberapa contoh hasil uji ketahanan aplikasi terhadap derau dari total 10 lagu yang diuji.

Tabel 4. Hasil uji ketahanan aplikasi terhadap derau

No.	Judul Lagu	Amplitudo Derau	Keterangan
1	Seperti Yang Kau Minta	0,1	Dikenali
		0,2	Dikenali
		0,3	Dikenali
		0,4	Dikenali
		0,5	Dikenali
		0,6	Dikenali
		0,7	Dikenali
		0,8	Dikenali
		0,9	Tidak dikenali
		1,0	Tidak dikenali
2	Mungkinkah	0,1	Dikenali
		0,2	Dikenali
		0,3	Dikenali
		0,4	Dikenali
		0,5	Dikenali
		0,6	Dikenali
		0,7	Dikenali
		0,8	Tidak dikenali
		0,9	Tidak dikenali
		1,0	Tidak dikenali
3	More Than Words	0,1	Dikenali
		0,2	Dikenali
		0,3	Dikenali
		0,4	Dikenali
		0,5	Dikenali
		0,6	Dikenali
		0,7	Tidak dikenali
		0,8	Tidak dikenali
		0,9	Tidak dikenali
		1,0	Tidak dikenali
4	Accidentally In Love	0,1	Dikenali
		0,2	Dikenali
		0,3	Dikenali
		0,4	Dikenali
		0,5	Dikenali
		0,6	Dikenali
		0,7	Dikenali
		0,8	Dikenali
		0,9	Dikenali
		1,0	Dikenali

Terlihat pada tabel 4 bahwa setiap lagu memiliki hasil yang berbeda terhadap 10 variasi amplitudo terhadap derau, dengan presentasi keberhasilan proses pencocokan lagu sebesar 76% dari 10 lagu yang diuji. Terlihat lagu berjudul ‘More Than Words’ tidak dapat dikenali saat amplitudo derau sebesar 0,7 sampai 1. Adapun lagu yang

tidak dapat dikenali ketika amplitudo derau sebesar 0,8 sampai 1, yaitu ‘Mungkinkah’. Kemudian, terdapat lagu yang tidak dapat dikenali hanya ketika amplitudo derau sebesar 0,9 dan 1, yaitu lagu ‘Seperti Yang Kau Minta’. Terakhir, lagu berjudul ‘Accidentally In Love’ dapat dikenali dengan semua variasi amplitudo derau. Hal ini disebabkan oleh kenyaringan, frekuensi, intensitas dari frekuensi, dan jumlah *peak* yang berbeda-beda tiap lagunya sehingga menyebabkan hasil pengujian yang berbeda-beda pula. Dengan demikian, aplikasi ‘Music and Match’ sudah berjalan dengan cukup baik dan memiliki ketahanan terhadap derau yang cukup baik pula.

4. Kesimpulan

Berdasarkan hasil pengujian didapatkan bahwa bagian tengah lagu memiliki rata-rata waktu pencocokan paling cepat dibandingkan kedua bagian lagu lainnya, yaitu 2,043 detik. Sementara bagian lagu yang lain memiliki rata-rata waktu pencocokan sebesar 3,045 detik untuk bagian awal lagu dan 3,257 detik untuk bagian akhir lagu. Perbedaan *bit rate* pada MP3 dari lagu yang didengarkan tidak mempengaruhi keberhasilan proses pencocokan lagu menggunakan aplikasi ‘Music and Match’, sehingga pengujian pengenalan lagu dengan variasi *bit rate* 32 kbps, 64 kbps, 96 kbps, 128 kbps, dan 192 kbps pada berkas MP3 memiliki presentase keberhasilan pengenalan lagu sebesar 100%. Pengujian ketahanan derau dengan 10 variasi amplitudo derau sebesar 0,1 sampai 1 memiliki presentase keberhasilan pengenalan lagu sebesar 76% dari total 100 percobaan pemutaran lagu dengan hasil yang berbeda-beda tiap lagu, dengan 2 lagu yang tidak dikenali saat amplitudo derau sebesar $\geq 0,7$, 4 lagu yang tidak dikenali saat amplitudo derau sebesar $\geq 0,8$, 2 lagu yang tidak dikenali hanya saat amplitudo derau sebesar $\geq 0,9$, dan 2 lagu yang dapat dikenali pada semua amplitudo derau yang diujikan. Penelitian selanjutnya dapat mempertimbangkan untuk menambahkan fitur lirik pada aplikasi dan menambahkan basis data untuk lagu Indonesia agar lebih lengkap

Referensi

- [1] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of audio fingerprinting,” *J. VLSI Signal Process. Syst. Signal Image. Video Technol.*, vol. 41, no. 3 SPEC. ISS., pp. 271–284, 2005.
- [2] M. Müller, *Fundamentals of Music Processing*. Erlangen: Springer, 2015.
- [3] G. Tzanetakis, “Music Mining,” in *Academic Press Library in Signal Processing: Volume 1 Signal Processing Theory and Machine Learning*, vol. 1, Victoria: Elsevier Masson SAS, 2014, pp. 1453–1492.
- [4] A. D. Utomo, “Implementasi Teknologi Augmented Reality Sebagai Panduan Salat Berbasis Sistem Operasi Android,” *TRANSMISI*, Vol 19, No 2 April (2017): 89–94.

- [5] M. Syarif, M. Somantri, Y. Christiyono, "Perancangan Aplikasi Bernama My Landmark Berbasis SIG untuk Informasi Pnjualan Tanah pada Perangkat Bergerak Android," *Transient*, vol. 5, no. 2, pp. 1–8, 2016.
- [6] C. W. Fitriyani, "Pengembangan Sistem Kuliah Online Universitas Bergerak Berbasis Android," *Transmisi*, vol. 4, no. 3, pp. 878–884, 2015.
- [7] M.R. Andikasani, M. Awaluddin, A. Suprayogi, "Aplikasi Persebaran Objek Wisata Di Kota Semarang Berbasis Mobile GIS Memanfaatkan Smartphone Android," *Jurnal Geodesi Undip*, Volume 3, Nomor 2, Tahun 2014, h. 28-39
- [8] R. Yusuf, H. Kusniyati, and E. Estrada, "Implementasi Algoritma Sidik Jari Audio Untuk Mendeteksi Duplikat Lagu," *Studia Informatika: Jurnal Sistem Informasi*, 8(1), 2015, 1-11
- [9] F. A. Maulana, *Semua Tentang Android*. Serpong: Universitas Surya, 2014.
- [10] R. N. Hidayat, "Rancang Bangun Pembuatan Aplikasi 'Voice Recognition Secure' Sebagai Media Keamanan Data Berbasis Android," *Tek. Inform.*, vol. 1, pp. 1–7, 2015.
- [11] P. Street, *GNSDK for Mobile (Android) Developers*. 2016.