

ROUGH SETS K-MEANS CLUSTERING DAN BACKPROPAGATION PADA OPTIMASI KASUS PENGENALAN SUARA

Ivan Eka Pratama¹, Ednawati Rainarli²

Program Studi Teknik Informatika

^{1,2} Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

Jl. Dipati Ukur 114 Bandung

E-mail: ivan.e.pratama@gmail.com¹, irene_edna@yahoo.com²

ABSTRAK

Pengenalan suara atau *speaker recognition* adalah suatu teknik mengenali suara berdasarkan sumbernya. Di dalam teknik tersebut ada suatu proses di mana suara diverifikasi berdasarkan identitas seseorang yang berbicara atau dalam bahasa asing disebut *speaker verification*. Pada penelitian ini akan diterapkan implementasi Rough Sets K-Means dan Backpropagation pada optimasi kasus pengenalan suara. Pada tahap ekstraksi fitur menggunakan MFCC sebagai ekstraksi ciri suara yang berguna bagi proses pengenalan suara. Rough Sets K-Means digunakan untuk mengoptimalkan hasil pada filter MFCC dan Backpropagation digunakan sebagai klasifikasi. Pada penelitian ini menggunakan *data set voice* yang berformat .wav didapatkan dari perekaman suara dari 5 pembicara. Berdasarkan pengujian *K-Fold Cross Validation* terhadap parameter yang digunakan didapatkan rata-rata akurasi yaitu sebesar 25% dan kata yang dipilih dapat digunakan dalam proses pengenalan.

Kata Kunci: *Speaker Recognition*, MFCC, *Backpropagation*, *Rough Sets K-Means*, Ekstraksi Ciri.

1. PENDAHULUAN

Pengenalan suara atau *speaker recognition* adalah suatu teknik mengenali suara berdasarkan sumbernya. Di dalam teknik tersebut ada suatu proses di mana suara diverifikasi berdasarkan identitas seseorang yang berbicara atau dalam bahasa asing disebut *speaker verification*. Pada saat melakukan pengenalan suara, data suara akan dilakukan proses ekstraksi fitur agar didapatkan

informasi yang ada dalam data suara. Tetapi hasil dari ekstraksi ciri berukuran besar, sehingga membutuhkan proses yang lebih banyak bila langsung digunakan untuk proses pengenalan pembicara. Oleh karena itu, dibutuhkan peranan dari metode *clustering* untuk membuat beberapa vektor pusat sebagai wakil dari keseluruhan vektor data yang ada. Metode *clustering* juga dapat mengoptimalkan hasil dari ekstraksi ciri tetapi banyak metode *clustering* yang dapat digunakan untuk mengimplementasikan pada kasus optimasi suara dengan hasil yang berbeda [1].

Metode *clustering* yang digunakan dalam penelitian ini adalah *Rough Sets K-Means*, keunggulan dalam menggunakan metode *rough sets k-means clustering* yaitu dapat meminimalkan jumlah memori yang digunakan untuk analisis informasi dan mengurangi perhitungan yang digunakan untuk mencari kemiripan dari vektor spektral [2]. Di samping itu terdapat pula penelitian perbandingan *K-Means*, *Rough Sets K-Means* pada kasus yang sama yaitu pengenalan suara. Disimpulkan bahwa *rough sets k-means* memiliki akurasi lebih baik pada kasus pengenalan suara dengan persentase 80% dibandingkan dengan *K-Means* yang mencapai 75% [3]. Karenanya *Rough K-Means* dapat digunakan untuk optimasi pengenalan suara.

Dalam penelitian ini proses pengenalan suara akan dilakukan menggunakan *Backpropagation*. *Backpropagation* merupakan sebuah jaringan syaraf yang bersifat *supervised* [4]. Dari penelitian sebelumnya yaitu penggunaan metode *backpropagation* pada pengenalan suara instrumen didapat bahwa metode *backpropagation* adalah klasifikasi yang optimal dalam mengenali suara instrumen berdasarkan jenis alat musik [5]. Dan

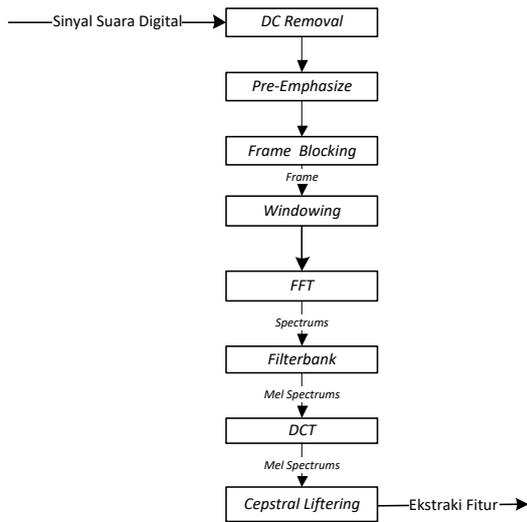
juga penelitian lainnya yaitu klasifikasi *Backpropagation* untuk kasus pengenalan wajah dapat disimpulkan bahwa metode *backpropagation* mendapat tingkat akurasi 37,33% [6]. Karenanya dalam penelitian ini akan dilakukan klasifikasi yang sama yaitu klasifikasi *backpropagation* pada kasus pengenalan suara manusia.

Berdasarkan uraian di atas, dalam penelitian ini akan dilakukan implementasi *Rough Sets K-Means* dan *Backpropagation* pada optimasi kasus pengenalan suara. Pada tahap ekstraksi fitur menggunakan MFCC sebagai ekstraksi ciri suara yang berguna bagi proses pengenalan suara. *Rough Sets K-Means* digunakan untuk mengoptimalkan hasil pada filter MFCC dan *Backpropagation* digunakan sebagai klasifikasi.

2. LANDASAN TEORI

2.1. MFCC

Metode ekstraksi ciri MFCC terdapat 8 proses yang akan dilakukan, tiap-tiap proses terdapat pada alur proses MFCC dapat dilihat pada Gambar 1.



Gambar 1. Alur MFCC

Berdasarkan Gambar 1 data masukan suara dengan kata telah dipilih akan masuk proses pertama yaitu *DC Removal* untuk mendapatkan nilai normal dari sinyal suara, lalu dilakukan proses *Pre-emphasize* untuk mengurangi *noise* pada sampel suara, lalu proses *Frame Blocking* untuk membagi sampel sinyal menjadi beberapa *frame*, lalu proses *Windowing* untuk mengurangi efek diskontinuitas pada ujung-ujung *frame* yang sebelumnya dihasilkan oleh proses *frame blocking*, lalu pada proses FFT diperoleh sampel sinyal dalam frekuensi domain, lalu proses *Filter bank* untuk

mendapatkan nilai energi dari setiap frekuensi *band*, lalu proses DFT untuk menghasilkan nilai *mel cepstrums*, langkah terakhir proses *Cepstral Lifting*, bertujuan menghaluskan *spectrum signal* [7].

2.2. Rough Sets-K-Means Clustering

Rough Sets-K-Means algoritma adalah penggunaan teori *rough set* dan algoritma *k-means* untuk menangani ketidakpastian yang terlibat pada analisis cluster. Dalam *rough clustering* setiap kluster memiliki dua penaksiran (*approximations*), yaitu *lower approximation* (nilai penaksiran rendah) dan *upper approximation* (nilai penaksiran tinggi). *Lower approximation* adalah bagian dari *upper approximation*. Anggota dalam *lower approximation* sudah dipastikan milik sebuah *cluster*, oleh karena itu mereka tidak dapat masuk ke kluster lain. Sedangkan data objek pada *upper approximation* belum dipastikan masuk dalam kluster atau mungkin masuk pada *cluster* lain. Karena keanggotaan *upper approximation* belum dipastikan maka setidaknya mereka masuk ke dalam kluster lain. [8]

Langkah-langkah dalam metode *clustering Rough Sets-K-Means* adalah sebagai berikut:

1. Pilih *cluster* awal n objek ke dalam *cluster* k .
2. Menghitung *Centroid* C_j pada kluster, dengan kondisi-kondisi sebagai berikut.

$$\text{If } \underline{U}(K) \neq \emptyset \text{ and } \overline{U}(K) - \underline{U}(K) = \emptyset$$

$$C_j = \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} \quad (1)$$

$$\text{Else } \underline{U}(K) = \emptyset \text{ and } \overline{U}(K) - \underline{U}(K) \neq \emptyset$$

$$C_j = \sum_{x \in \underline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|} \quad (2)$$

$$\text{Else}$$

$$C_j = W_{lower} \times \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} + W_{upper} \times \sum_{x \in \underline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|} \quad (3)$$

3. Tetapkan setiap objek pada batas *lower* $\underline{U}(K)$ atau batas *upper* $\overline{U}(K)$ pada *cluster* masing-masing i . Untuk setiap objek vektor x , tetapkan $d(X, C_j)$ menjadi jarak antara dirinya sendiri dan centroid pada kluster C_j .

$$d(X, C_j) = \min_{1 \leq j \leq K} d(X, C_j) \quad (4)$$

Rasio $d(X, C_i) / d(X, C_j), 1 \leq j \leq K$ digunakan untuk menentukan keanggotaan dari x adalah sebagai berikut $d(X, C_i) / d(X, C_j) \leq \epsilon$, pada pasangan (i, j) , pada $x \in \overline{U}(C_i)$ dan $x \in \overline{U}(C_j)$ dan x tidak akan jadi bagian pada setiap *lower approximation*. Jika tidak,

$x \in \bar{U}(C_i)$, seperti $d(X, C_i)$ adalah nilai minimal dari $1 \leq j \leq K$. Sebagai tambahan $x \in \bar{U}(C_i)$.

4. Jika kriteria konvergen terpenuhi, yaitu pusat *cluster* yang sama dengan yang di iterasi sebelumnya maka iterasi akan berhenti, jika tidak maka ulangi langkah 2 dan 3.

2.3. Backpropagation

Backpropagation melatih suatu jaringan untuk mengenali pola yang digunakan dalam proses *training* lalu memberikan nilai yang benar terhadap pola *input* yang memiliki kemiripan dengan pola yang dipakai selama pelatihan [9].

Proses pelatihan *backpropagation* memiliki kesamaan pada jaringan syaraf lainya. Pada jaringan *feed forward*, proses pelatihan dilakukan sebagai perhitungan bobot sehingga pada akhir pelatihan akan menghasilkan bobot-bobot yang optimal. Selama proses pelatihan, bobot-bobot disusun secara iterasi untuk meminimalkan *error*. *Error* dihitung dari rata-rata kuadrat kesalahan. Selanjutnya rata-rata kuadrat kesalahan akan dijadikan dasar perhitungan untuk fungsi aktivasi. [10]

Algoritma *backpropagation* digunakan untuk memperbaiki bobot-bobot jaringan yang menjadikan fungsi aktivasi menurun. Terdapat 3 fase Pelatihan *backpropagation* menurut Siang [9] antara lain:

1. Propagasi maju (*Forward Propagation*).

Dalam propagasi maju, setiap sinyal masukan dihitung maju menuju *hidden layer* sampai *output layer* dengan menggunakan fungsi aktivasi yang ditentukan.

2. Propagasi mundur (*Backward Propagation*).

Nilai *error* (selisih antara *output layer* dengan target) dihitung mundur mulai dari garis yang berhubungan langsung dengan unit-unit di *output layer*.

3. Perubahan bobot.

Disini dilakukan perubahan bobot untuk mengurangi kesalahan yang terjadi. Ketiga fase tersebut dilakukan berulang kali hingga mencapai kondisi penghentian.

Algoritma pelatihan untuk jaringan dengan satu *hidden layer* (menggunakan fungsi aktivasi *sigmoid* biner) adalah sebagai berikut.

1. Inisialisasi seluruh bobot dengan bilangan *random* bernilai kecil.
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah dari 3 sampai 11
3. Pada tiap data pelatihan lakukan langkah 4 sampai 10

I : Forward Propagation

4. Setiap unit masukan menerima sinyal dan diteruskan ke *hidden layer*

5. Hitung semua keluaran di *hidden layer*

$$z_j \quad (j = 1, 2, \dots, p)$$

$$z_in_j = v_{j0} + \sum_{t=1}^n x_t v_{jt} \quad (5)$$

$$z_j = f(z_in_j) = \frac{1}{1 + e^{-z_in_j}} \quad (6)$$

6. Hitung semua keluaran di *output layer*

$$y_k \quad (k = 1, 2, \dots, m)$$

$$y_in_k = w_{k0} + \sum_{j=1}^n z_j w_{kj} \quad (7)$$

$$y_j = f(y_in_k) = \frac{1}{1 + e^{-y_in_k}} \quad (8)$$

II : Backward Propagation

7. Hitung faktor unit keluaran berdasarkan *error* di setiap unit keluaran

$$y_k \quad (k = 1, 2, \dots, m)$$

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

$$= (t_k - y_k) y_k (1 - y_k) \quad (9)$$

δ_k merupakan unit *error* yang akan digunakan dalam perubahan bobot *layer* di bawahnya (langkah 8)

8. Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk mengubah bobot w_{kj})

dengan laju percepatan α ;

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (10)$$

9. Hitung faktor *hidden layer* berdasarkan *error* di setiap unit tersembunyi

$$z_j \quad (j = 1, 2, \dots, p)$$

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{kj} \quad (11)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_in_j f'(z_in_j) = \delta_in_j z_j (1 - z_j) \quad (12)$$

10. Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk merubah bobot v_{ji})

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (13)$$

III : Perubahan bobot

11. Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke *output layer*:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (14)$$

Perubahan bobot garis yang menuju ke *hidden layer*:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (15)$$

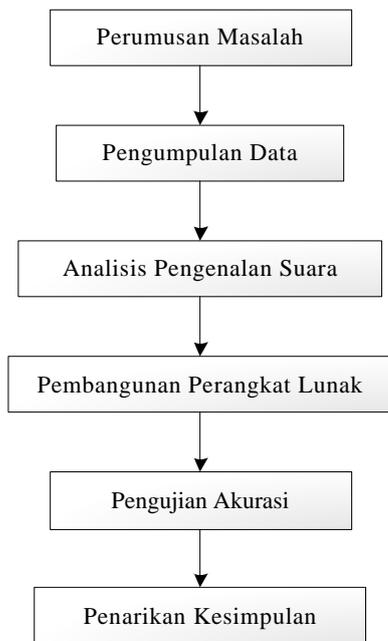
Setelah proses pelatihan dilakukan, jaringan dapat digunakan untuk mengenali pola. Dalam proses mengenali, hanya dilakukan propagasi maju saja yang dilakukan untuk menentukan *output*.

Proses pelatihan akan menghasilkan nilai yang baik apabila dilakukan pemilihan bobot awal yang optimal, karena bobot awal akan sangat berpengaruh seberapa cepat konvergensinya. Oleh karena itu dalam standar *backpropagation*, bobot

dan bias diisi dengan bilangan *random* kecil dan biasanya bobot awal diinisialisasi secara acak dengan nilai antara -1 sampai 1 atau interval yang lainnya.

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian eksperimen. Alur penelitian dapat dilihat pada Gambar 2



Gambar 2. Alur Metode Penelitian

Berdasarkan alur penelitian pada Gambar 2 dapat dijelaskan sebagai berikut. Pada tahap perumusan masalah, peneliti merumuskan masalah bagaimana mengimplementasikan metode *rough sets-k-means clustering* dan klasifikasi *backpropagation* pada kasus pengenalan suara. Pada tahap pengumpulan data yang dilakukan pada penelitian ini yaitu, kajian pustaka dan pengumpulan *dataset*. Pada tahap analisis pengenalan suara yang dilakukan dalam penelitian di antaranya yaitu:

1. Ekstraksi Fitur Dengan MFCC

Tahapan ekstraksi MFCC merupakan tahapan awal di mana sinyal suara masukan akan di olah dengan berbagai proses di dalamnya agar sinyal suara siap untuk dilakukan proses *clustering*. Proses pada tahapan ini meliputi *DC Removal, Pre-Emphasize, Frame Blocking, Windowing, Fast Fourier Transform, Filter bank, Discrete Cosine Transform, Cepstral Liftering*.

2. *Rough Sets-K-Means Clustering*

Pada fungsi *Clustering*, nilai *input* sebenarnya merupakan *cepstrums* hasil dari *feature extraction*. Terdapat proses dalam fungsi *Clustering* untuk mendapatkan vektor pusat. Pencarian vektor pusat dilakukan secara berulang ulang sehingga didapatkan vektor pusat yang mewakili seluruh vektor hasil *feature extraction*.

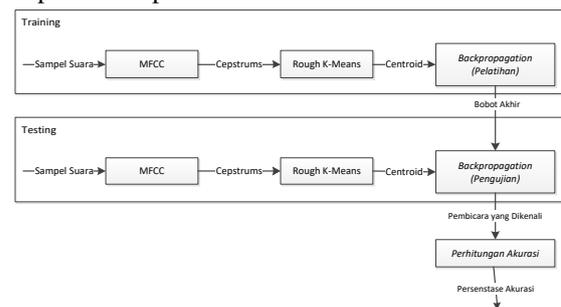
3. Klasifikasi *Backpropagation*

Untuk dapat melakukan pencocokan suara, maka data baru yang masuk akan disamakan dengan data yang telah ada dalam *database* sebelumnya. Setiap data dibandingkan dengan semua yang ada pada salah satu model *database* secara bergantian.

Model pembangunan perangkat lunak yang digunakan adalah model *prototype*. Pada tahap pengujian akurasi dari hasil pengenalan pada *rough sets-k-means* akan menggunakan metode *K-Fold Cross Validation*. Tahapan akhir ini adalah menarik kesimpulan terhadap aplikasi yang telah dibangun.

4. HASIL DAN PEMBAHASAN

Dalam pembangunan sistem pengenalan suara pada penelitian ini meliputi proses data latih (data training) dan proses data pengujian (data testing). Dalam proses yang dilakukan ada beberapa metode yang yang diterapkan yaitu, fitur ekstraksi menggunakan metode Mel Frequency Cepstrums Coefficients (MFCC), kemudian *Rough Sets K-Means Clustering* selanjutnya dilakukan klasifikasi *Backpropagation*. Alur proses pengenalan suara dapat dilihat pada Gambar 3.



Gambar 3. Alur Proses Pengenalan Suara

Dalam penelitian ini pengujian yang dilakukan akan menggunakan *K-Folds Cross Validation*, Jumlah dataset yang digunakan sebanyak 100 data. Tiap dataset berjumlah 20 data yang terdiri dari 5 orang pembicara mengucap 5 kata berbeda dalam bahasa Indonesia yaitu “ada”, “adalah”, ”bisa”, ”jadi” dan “sebut”. Pengujian menggunakan dataset yang akan dibagi menjadi 5 bagian dengan iterasi pengujian dilakukan sebanyak 5 kali dimana tiap

dataset akan secara bergantian menjadi data latih dan data uji.

Berdasarkan skenario pengujian dengan menggunakan *5-Fold Cross Validation* pada algoritma *backpropagation* dengan menggunakan parameter *hidden layer*=30 lalu menggunakan parameter lain dengan melakukan percobaan pada nilai yang berbeda didapatkan hasil akurasi sebagai berikut.

1) Pengujian pergantian jumlah kluster

Pengujian pertama dilakukan untuk melihat seberapa besar pengaruh jumlah kluster terhadap akurasi pengenalan suara. Berikut adalah hasil pengujian akurasi berdasarkan pergantian jumlah kluster dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Akurasi Berdasarkan Jumlah Kluster

Jumlah Kluster	K=3	K=4	K=5	K=6	K=7
Akurasi	18%	20%	25%	22%	22%

Berdasarkan hasil pengujian pada Tabel 1 nilai akurasi dengan jumlah kluster 3 adalah 18% lalu dengan mencoba mengganti jumlah kluster dengan jumlah yang bertingkat yaitu 4 dan didapatkan nilai akurasi 20% dan seterusnya bertambah dengan jumlah kluster 5 didapatkan nilai 25%. Lalu setelah jumlah kluster bertambah menjadi 6 nilai akurasi menurun menjadi 22% dan didapatkan pula nilai akurasi yang sama pada jumlah kluster 7 yaitu 22%. Didapatkanlah akurasi terbesar dengan nilai 25% dengan menggunakan jumlah kluster K=5.

2) Pengujian pergantian *epoch*

Pengujian ini menggunakan parameter *learning rate*=0.01, dan *epoch* 100, 500 dan 1000. Nilai akurasi dapat dilihat pada tabel Tabel 2.

Tabel 2. Hasil Akurasi Percobaan Pergatian Epoch

No	<i>epoch</i>	<i>Learning rate</i>	Akurasi	
			Non	<i>Clustering</i>
1	100	0.01	50%	20%
2	500	0.01	62%	22%
3	1000	0.01	65%	25%

Berdasarkan nilai akurasi pada Tabel 2 ada tiga nilai akurasi tanpa menggunakan *clustering* dan menggunakan *clustering*, dari kedua-duanya menggunakan nilai *learning rate* yang sama yaitu 0.01. Dari hasil ketiga pengujian berdasarkan

pergantian nilai *epoch* dapat dilihat bahwa akurasi yang didapatkan meningkat seiring dengan penambahan nilai *epoch*. Dengan nilai *epoch* 100 didapatkan nilai akurasi 50% tanpa penggunaan *clustering* dan 20% dengan menggunakan *clustering*. Lalu pada nilai *epoch* 500 terdapat peningkatan sehingga didapatkan nilai akurasi 62% dan 22%. Pada pengujian terakhir dengan *epoch* 1000 didapatkan nilai akurasi 65% dan 25%. Nilai akurasi tertinggi yaitu pengujian ketiga dengan nilai *epoch* 1000 yang akan dilakukan pada pengujian selanjutnya yaitu pengujian pergantian *learning rate*.

3) Pengujian pergantian *learning rate*.

Pengujian ini menggunakan parameter *epoch* yang optimal pada pengujian tahap sebelumnya yaitu 1000 dan *learning rate* 0.01, 0.001 dan 0.0001. Nilai akurasi dapat dilihat pada tabel Tabel 3.

Tabel 3. Hasil Pengujian Percobaan Pergatian *Learning Rate*

No	<i>epoch</i>	<i>Learning rate</i>	Akurasi	
			Non	<i>Clustering</i>
1	1000	0.01	65%	25%
2	1000	0.001	67%	30%
3	1000	0.0001	62%	22%

Berdasarkan nilai akurasi pada Tabel 3 ada tiga nilai akurasi tanpa menggunakan *clustering* dan menggunakan *clustering*, dari kedua-duanya menggunakan nilai *epoch* yang sama yaitu 1000. Dari hasil ketiga pengujian berdasarkan pergantian nilai *learning rate* dapat dilihat bahwa akurasi yang didapatkan meningkat seiring dengan penambahan nilai *learning rate*. Dengan nilai *learning rate* 0.01 didapatkan nilai akurasi 65% tanpa penggunaan *clustering* dan 25% dengan menggunakan *clustering*. Lalu pada nilai *learning rate* 0.001 terdapat peningkatan sehingga didapatkan nilai akurasi 60% dan 22%. Pada pengujian terakhir dengan *learning rate* 0.0001 terdapat penurunan nilai akurasi menjadi 67% dan 30%. Nilai akurasi tertinggi yaitu pengujian ketiga dengan nilai *learning rate* 0.001.

Dari keseluruhan pengujian akurasi yang dilakukan didapatkan parameter optimal yaitu *epoch* 1000, *learning rate* 0.001 bila tidak menggunakan *clustering* dan *learning rate* 0.0001 pada penggunaan *clustering*.

5. Kesimpulan

Berdasarkan hasil pengujian pada penelitian ini, dapat diambil kesimpulan bahwa metode clustering Rough Set K-Means belum dapat dijadikan sebagai metode optimal yang dapat meningkatkan proses pengenalan suara dengan metode Backpropagation sebagai algoritma klasifikasi dengan memanfaatkan hasil ekstraksi ciri MFCC. Dalam tahap pengujian didapatkan parameter hidden layer=30, epoch 1000, learning rate 0.001 bila tidak menggunakan clustering dan learning rate 0.0001 pada penggunaan clustering, dengan nilai rata-rata akurasi sebesar 25% dengan menggunakan jumlah kluster K=5.

Adapun saran untuk penelitian kedepannya yaitu dengan menggunakan metode clustering yang lebih baik agar dapat meminimalkan jumlah fitur tanpa mengurangi akurasi pada klasifikasi. Dalam proses pra-proses diharapkan untuk menambahkan metode yang lebih baik yang dapat mengatasi masalah *noise* agar dapat mempengaruhi pada tingkat keberhasilan pengenalan suara. Selain itu bisa juga menggunakan metode ekstraksi ciri dan klasifikasi yang lebih optimal untuk mendapatkan hasil pengenalan yang akurat. Agar mendapatkan hasil akurasi yang lebih baik bisa dilakukan untuk mencoba menggunakan variasi lain terhadap jumlah parameter yang akan digunakan seperti jumlah *hidden layer*, *learning rate*, *error* yang akan berpengaruh kepada hasil akurasi.

DAFTAR PUSTAKA

- [1] H. S. Manunggal, "Perancangan dan pembuatan perangkat lunak pengenalan suara pembicara dengan menggunakan analisa MFCC feature extraction," *Bachelor thesis, Petra Christian University.*, vol. I, 2005.
- [2] M. Ghassemian dan K. Strange, *Speaker Identification Features, Models and Robustness*, Denmark: Kongens Lyngby, 2009.
- [3] G. B. Djamal, "Perbandingan Metode Ekstraksi Ciri Rough Set-K-Means Dan K-Means Pada Optimasi Kasus Pengenaan Suara," 2017.
- [4] K. Setiawan, *Paradigma Sistem Cerdas*, Surabaya: Bayumedia, 2003.
- [5] Gunawan, A. D. Gunawan dan S. N. Soenardjo, "Penerapan algoritma backpropagation untuk klasifikasi musik dengan solo instrumen," *Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009)*, 2009.
- [6] M. D. Wuryandari dan I. Afrianto, "Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah," *Jurnal Komputer dan Informatika (KOMPUTA)*, vol. I, Maret 2012.
- [7] A. Resmawan dan D. Putra, "Verifikasi Biometrika Suara Menggunakan MFCC dan DTW," *LONTAR KOMPUTER*, vol. II, 2011.
- [8] G. Peters, "Some refinements of rough k-means clustering," *Pattern Recognition* 39.
- [9] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan MATLAB*, Yogyakarta: Andi Offset, 2005.
- [10] S. Kusumadewi, *Membangun Jaringan Syaraf Tiruan dengan Matlab Dan Excel Link*, Yogyakarta: Graha Ilmu, 2004.