

# Deteksi Penyakit Epilepsi dengan Menggunakan Entropi Permutasi, K-means *Clustering*, dan *Multilayer Perceptron*

Yunita Ardilla, Handayani Tjandrasa, dan Isye Arieshanti

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*e-mail*: handatj@its.ac.id

**Abstrak**—Epilepsi didefinisikan sebagai kumpulan gejala dan tanda-tanda klinis yang muncul disebabkan gangguan fungsi otak secara intermiten, yang terjadi akibat lepas muatan listrik abnormal atau berlebihan dari neuron-neuron secara paroksimal dengan berbagai macam etiologi. Banyak pasien yang tidak menyadari adanya gejala epilepsi dalam dirinya. Oleh karena itu diperlukan sistem yang bisa memprediksi apakah seseorang menderita epilepsi bebas kejang, atau epilepsi kejang. Dalam artikel ini diimplementasikan perangkat lunak pendeteksi penyakit epilepsi dengan menggunakan entropi permutasi, K-means *clustering*, dan *multilayer perceptron*. Hasil model dari algoritma *multilayer perceptron* akan digunakan dalam proses prediksi. Dataset yang digunakan dalam proses uji coba berisi lima himpunan (A-E) EEG dari manusia sehat dan yang menderita epilepsi yang tersedia online ("Klinik für Epileptologie, Universität Bonn"). Performa terbaik yang dihasilkan oleh model adalah akurasi sebesar 96,5%, *specificity* sebesar 95,45%, dan *sensitivity* sebesar 97,97%.

**Kata Kunci**—entropi permutasi, entropi, klasifikasi, K-means *clustering*, *multilayer perceptron*.

## I. PENDAHULUAN

EPILEPSI didefinisikan sebagai kumpulan gejala dan tanda-tanda klinis yang muncul disebabkan gangguan fungsi otak secara intermiten, yang terjadi akibat lepasnya muatan listrik abnormal atau berlebihan dari neuron-neuron secara paroksimal dengan berbagai macam etiologi. Sedangkan serangan atau bangkitan epilepsi yang dikenal dengan *epileptic seizure* adalah manifestasi klinis yang serupa dan berulang secara paroksimal, yang disebabkan oleh hiperaktivitas listrik sekelompok sel saraf di otak yang spontan dan bukan disebabkan oleh suatu penyakit otak akut [1].

Manifestasi serangan atau bangkitan epilepsi secara klinis dapat dicirikan sebagai gejala yang timbulnya mendadak, hilang spontan, dan cenderung untuk berulang. Sedangkan gejala dan tanda-tanda klinis tersebut sangat bervariasi dapat berupa gangguan tingkat penurunan kesadaran, gangguan sensorik (subjektif), gangguan motorik atau kejang (objektif), gangguan otonom (vegetatif), dan perubahan tingkah laku (psikologis). Semuanya itu tergantung dari letak fokus epileptogenesis atau sarang epileptogen dan penjarannya sehingga dikenal bermacam jenis epilepsi.

Diagnosa pasti epilepsi adalah dengan menyaksikan secara langsung terjadinya serangan, namun serangan epilepsi jarang bisa disaksikan langsung oleh dokter, sehingga diagnosa epilepsi hampir selalu dibuat berdasarkan *alloanamnesis*. Namun *alloanamnesis* yang terbaik dan akurat sulit didapatkan, karena gejala yang diceritakan oleh orang sekitar penderita yang menyaksikan sering kali tidak khas, sedangkan penderitanya sendiri tidak tahu sama sekali bahwa ia baru saja mendapat serangan epilepsi. Satu-satunya pemeriksaan yang membantu diagnosa penderita epilepsi adalah rekaman *electroencephalogram* (EEG). Perekaman EEG dilakukan secara terus menerus selama jangka waktu beberapa hari dan akibatnya sebagian besar data kemudian harus dianalisis secara visual oleh para ahli agar dapat mengidentifikasi penyakit epilepsi. Namun proses ini membutuhkan banyak waktu dan menghabiskan biaya yang besar. Oleh karena itu diperlukan pembuatan sistem klasifikasi pembuat keputusan yang cepat dan akurat.

Sebelumnya telah dilakukan beberapa penelitian untuk melakukan klasifikasi penyakit epilepsi dengan menggunakan dataset EEG. Metode yang digunakan untuk melakukan klasifikasi sebelumnya di antaranya adalah *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Bayesian Logistic Regression*. Penggunaan metode metode SVM memberikan akurasi sebesar 86,1% [2]. Pada artikel ini, akan dibangun perangkat lunak implementasi entropi permutasi, K-means *clustering*, dan jaringan saraf tiruan *multilayer perceptron* untuk deteksi *electroencephalogram* epilepsi yang bertujuan untuk menentukan pola dari kelas terkait (positif menderita epilepsi atau tidak) berdasarkan dataset EEG.

## II. DASAR TEORI

### A. Entropi Permutasi

Entropi Permutasi (PE) merupakan metode yang berguna untuk analisis semua jenis data deret waktu, karena memungkinkan penggambaran distribusi pada probabilitas dari semua kemungkinan keadaan dari deret waktu. Entropi permutasi digunakan untuk proses pengukuran kompleksitas pada deret waktu berdasarkan perbandingan nilai-nilai tetangga. Deret waktu dipetakan ke dalam urutan simbolis

yang mendeskripsikan hubungan antara nilai yang sekarang dengan sejumlah nilai dalam jarak tertentu.

Mula-mula pemetaan dilakukan dengan cara mengubah deret waktu  $x(t)$ ,  $t = 1, 2, \dots$  ke dalam ruang berdimensi  $m$ , sehingga sesuai dengan (1).

$$X(t) = [x(t), x(t+\tau), \dots, x(t+m\tau)] \quad (1)$$

Di mana  $m$  merupakan dimensi (jumlah sampel yang dimasukkan ke dalam setiap motif) dan  $\tau$  adalah jarak waktu (jumlah sampel terentang oleh setiap bagian dalam motif). Untuk setiap dimensi, terdapat  $m!$  permutasi (motif) yang memungkinkan. Jika setiap permutasi dianggap sebagai suatu simbol, maka  $X(t)$  dapat dilambangkan dengan sebuah urutan symbol,  $j$ , di mana masing-masing memiliki distribusi peluang  $p_j$ . Oleh karena itu, berdasarkan definisi entropi Shannon, PE yang dinormalisasi ( $H_p$ ) untuk suatu deret waktu  $x(t)$  didefinisikan pada (2).

$$H_p(m) = -\frac{1}{\ln(m!)} \sum_{j=1}^J p_j \ln(p_j) \quad (2)$$

Entropi permutasi digunakan untuk ekstraksi fitur data EEG. Metode ini dipilih dikarenakan karakteristik dari sinyal EEG yang tidak periodik, tidak mempunyai pola baku, serta mempunyai amplitudo tegangan yang kecil, maka menyebabkan sinyal EEG sangat mudah tertimbun *noise* sehingga sulit untuk dilakukan proses prediksi. Permasalahan ini mampu diatasi dengan entropi permutasi karena metode ini mampu melacak dinamika aktivitas otak dengan baik [3].

### B. K-means Clustering

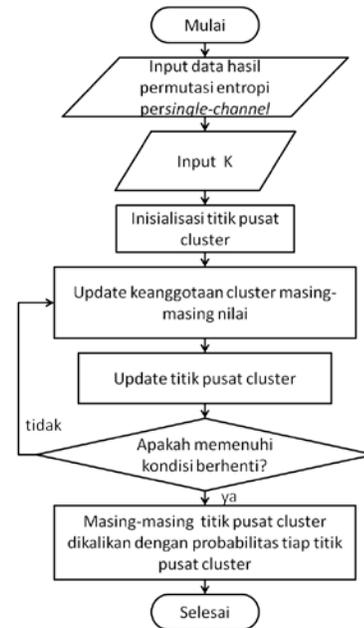
K-means adalah algoritma untuk membuat *cluster*  $n$  objek berdasarkan atribut menjadi  $k$  partisi, dimana  $k < n$ . Algoritma ini disusun atas dasar ide yang sederhana. Mula-mula ditentukan berapa *cluster* yang akan dibentuk. Elemen pertama dalam setiap *cluster* dapat dipilih untuk dijadikan sebagai titik tengah (*centroid point*) *cluster*. Algoritma K-means selanjutnya akan melakukan pengulangan langkah-langkah berikut hingga tidak ada objek atau elemen yang dapat dipindahkan.

- 1) Menentukan jumlah *cluster*
- 2) Menentukan nilai *centroid*, dalam menentukan nilai titik tengah untuk awal iterasi, nilai awal *centroid* dilakukan secara acak. Sedangkan jika menentukan nilai *centroid* yang merupakan tahap dari iterasi, maka digunakan (3).

$$\bar{v}_{ij} = \frac{1}{N_i} \sum_{k=0}^{N_i} x_{kj} \quad (3)$$

Keterangan:

- $v_{ij}$  adalah titik tengah dibagi rata-rata *cluster* ke-1 untuk variabel ke- $j$ ,
  - $N_i$  adalah jumlah data yang menjadi anggota *cluster* ke- $i$ ,
  - $I$  dan  $k$  adalah indeks dari *cluster*,
  - $j$  adalah indeks dari variabel,
  - $x_{kj}$  adalah nilai data ke- $k$  yang ada di dalam *cluster* tersebut untuk variabel ke- $j$ .
- 3) Menghitung jarak antara titik centroid dengan titik tiap objek. Untuk menghitung jarak tersebut dapat menggunakan *euclidean distance* seperti pada (4).



Gambar 1. Diagram alir K-means clustering

$$D(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4)$$

Keterangan:

- $D$  adalah *euclidean distance*,
- $i$  adalah banyaknya dimensi,
- $(q, p)$  adalah koordinat objek.

Untuk menentukan anggota cluster adalah dengan memperhitungkan jarak minimum objek. Nilai yang diperoleh dalam keanggotaan data pada *distance* matriks adalah 0 atau 1, dimana nilai 1 untuk data yang dialokasikan ke *cluster* dan nilai 0 untuk data yang dialokasikan ke *cluster* yang lain.

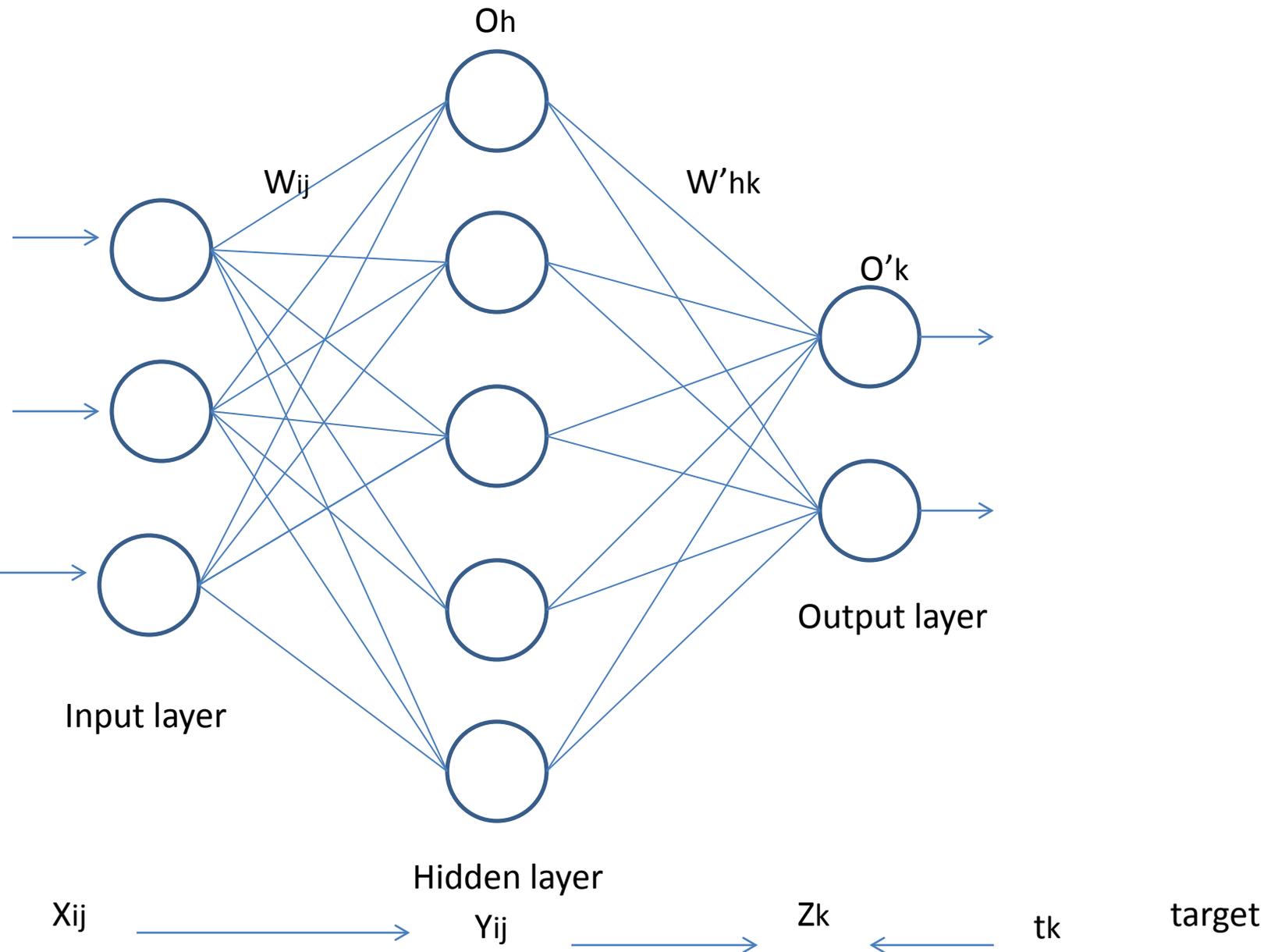
- 4) Kembali ke tahap 2, lakukan perulangan hingga nilai titik tengah yang dihasilkan tetap dan anggota *cluster* tidak berpindah-pindah ke *cluster* lain.

Penggunaan K-means pada umumnya adalah digunakan untuk membentuk *cluster* dari sekumpulan data. Namun, pada artikel ini K-means digunakan untuk melakukan reduksi fitur. Penggunaan K-means untuk reduksi fitur diawali dengan proses yang sama seperti yang telah dijelaskan sebelumnya. Data yang digunakan pada proses K-means adalah nilai-nilai PE pada satu *single-channel* EEG. Jadi, proses K-means akan dilakukan sebanyak jumlah *single-channel* EEG [4].

Proses K-means clustering dieksekusi untuk masing-masing nilai entropi permutasi dari *single-channel* EEG. Setelah proses K-means clustering dilakukan akan didapatkan sejumlah  $k$  titik pusat. Setiap titik pusat dikalikan dengan probabilitas setiap *cluster*. Kegunaan dari probabilitas tersebut sebagai faktor bobot dari masing-masing nilai *centroid*. Setelah proses K-means clustering selesai dilakukan, maka akan terbentuk dataset dengan atribut sejumlah  $k$ . Gambar 1 menunjukkan diagram alir dari proses K-means clustering.

### C. Multilayer Perceptron

Pada dasarnya algoritma *multilayer perceptron* ada dua tahap, yaitu *feedforward* dan *backpropagation*. Pada tahap



Gambar 2. Lapisan dan sinyal dari algoritma *backpropagation*

*feedforward*, sinyal input diberikan ke dalam jaringan. Komputasi dilakukan oleh neuron pada setiap lapisan sehingga menghasilkan *output*. Pada tahap ini, bobot pada jaringan tidak mengalami perubahan. Sebaliknya, pada tahap *backpropagation*, semua bobot diperbaiki berdasarkan *error* jaringan. Bobot diperbaiki untuk membuat *output* yang dihasilkan semakin mendekati *output* yang diinginkan.

Di *feedforward*, vektor *input* dari *training* set diterapkan untuk unit *input* jaringan dan dilewatkan dari lapisan ke lapisan sehingga menghasilkan *output* akhir. Selama *backpropagation*, *output* dari jaringan dibandingkan dengan *output* yang diinginkan untuk mendapatkan nilai *error*. *Error* yang dihasilkan digunakan untuk mengubah bobot [5]. Gambar 2 merupakan gambar lapisan dan sinyal dari algoritma *backpropagation*.

Berikut adalah penjelasan *pseudocode* algoritma *backpropagation*.

- Buat sebuah *network* dengan  $n_{in}$  unit *input*,  $n_{hidden}$  unit tersembunyi, dan  $n_{out}$  unit *output*.
- Inisialisasi semua bobot dan bias ke angka random yang kecil (antara -0,05 dan 0,05).
- Sampai kondisi terminasi tercapai, kemudian untuk tiap data *training*, lakukan:
  1. Masukkan data *training* sebagai *input* dan hitung *output* jaringan  $O_u$
  2. Untuk tiap unit *output*  $k$ , hitung term *error*  $\delta_k$   

$$\delta_k \leftarrow O_k (1 - O_k) (t_k - O_k)$$
  3. Untuk tiap unit tersembunyi hitung term *error*  $\delta_h$   

$$\delta_h \leftarrow O_h (1 - O_h) \sum_{k \in \text{output } s} w_{h,k} \delta_k$$
  4. Ubah tiap bobot dan bias  $w_{ij}$   

$$W_{ij} \leftarrow W_{ij} + \Delta W_{ij}$$

Di mana

$$\Delta W_{ij} \leftarrow \eta \delta_j x_{ij}$$

Keterangan:

- $\delta_k$  adalah *error output*
  - $O_k$  adalah keluaran dari lapisan *output*
  - $t_k$  adalah *output* yang diharapkan
  - $\delta_h$  adalah *error* pada lapisan tersembunyi
  - $O_h$  adalah keluaran dari lapisan tersembunyi
  - $W_{h,k}$  adalah bobot antara lapisan *output* dan lapisan tersembunyi
  - $w_{ij}$  adalah semua bobot dan bias pada jaringan saraf
  - $\eta$  adalah *learning rate*
  - $x_{ij}$  adalah keluaran dari lapisan
- $\Delta W_{ij}$  adalah selisih bobot saat ini dengan bobot sebelumnya.

### III. PERANCANGAN PERANGKAT LUNAK

Ketika akan memulai proses, mula-mula dataset yang terdiri dari 5 himpunan dilakukan *preprocessing* dengan metode entropi permutasi dengan tujuan untuk memetakan data dan mendapatkan fitur-fitur yang ada dalam dataset. Sebelum melakukan proses ini perlu mengisi parameter jumlah dimensi entropi permutasi.

Dataset yang didapatkan dari metode entropi permutasi tersebut akan menjadi data masukan untuk tahap *preprocessing* selanjutnya, yaitu dengan menggunakan metode K-means *clustering*. Sebelum melakukan proses ini, perlu diperhatikan untuk mengisi parameter K-means *clustering*, yaitu jumlah *cluster*. Setelah proses K-means *clustering* selesai, dilakukan perhitungan probabilitas masing-masing titik pusat *cluster*. Nilai-nilai probabilitas didapatkan dengan cara jumlah anggota tiap *cluster* dibagi dengan jumlah anggota seluruh *cluster*. Nilai akhir fitur yang akan digunakan pada tahap berikutnya adalah titik pusat masing-masing *cluster* yang sudah dikalikan dengan nilai probabilitas masing-masing *cluster*. Hasil dari proses K-means *clustering* ini akan digunakan sebagai data masukan untuk proses selanjutnya.

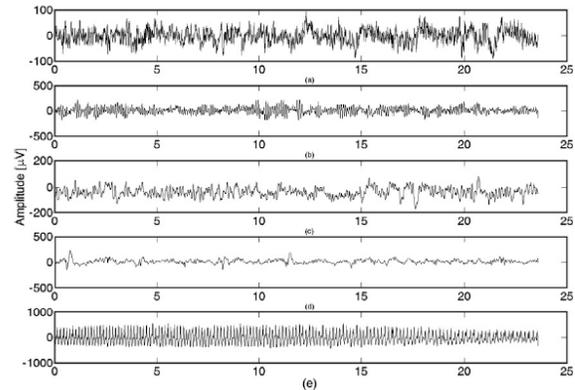
Setelah tahap *preprocessing* dilakukan, selanjutnya akan dilakukan proses pembelajaran. Dalam proses pembelajaran digunakan metode jaringan saraf tiruan *multilayer* perceptron. Proses pembelajaran terpisah dilakukan oleh setiap kelas. Proses ini dilakukan dengan empat macam kombinasi dataset, yaitu A dan E, B dan E, C dan E, serta D dan E. Setelah proses pembelajaran selesai dilakukan, maka akan didapatkan empat model akhir dari sistem klasifikasi.

Setelah model selesai dibuat, tahap berikutnya adalah mengukur performa model dalam tahap uji. Jika performa model yang dihasilkan mempunyai nilai yang baik, maka model dapat digunakan untuk klasifikasi data yang belum diketahui kelasnya.

### IV. IMPLEMENTASI

Sistem perangkat lunak ini dibangun dengan menggunakan perangkat lunak Microsoft Visual Studio 2010 dengan bahasa pemrograman C#. Data yang digunakan pada sistem ini berisi lima himpunan (A-E) rekaman EEG dari manusia sehat dan yang menderita epilepsi dan tersedia online ("Klinik für

Epileptologie, Universität Bonn") [6]. Gambar 3 merupakan contoh sinyal dari lima himpunan dataset. Pada tahap pra proses, masing-masing data *single-channel* EEG diolah dengan menggunakan metode entropi permutasi sehingga setiap *single-channel* EEG akan dihasilkan 23 nilai entropi permutasi.



Gambar 3. Contoh sinyal dari lima himpunan dataset.

Data hasil dari proses entropi permutasi akan diolah dengan metode K-means *clustering* hal ini bertujuan untuk mereduksi fitur pada dataset. Dari proses K-means *clustering* akan dihasilkan sejumlah data dengan *k* fitur.

Setelah tahap pra proses dilakukan, langkah selanjutnya yaitu tahap latih. Dalam tahap ini digunakan metode *multilayer* perceptron. Data masukan pada proses ini berupa data hasil dari proses K-means *clustering*. Tahap latih akan menghasilkan empat model klasifikasi yaitu model dari dataset A dan E, B dan E, C dan E, serta D dan E.

### V. UJI COBA

Uji coba dilakukan untuk mengetahui apakah algoritma entropi permutasi, K-means *clustering*, dan *multilayer* perceptron dapat diimplementasikan untuk proses prediksi penyakit epilepsi. Skenario uji coba yang dilakukan ada tiga skenario. Skenario pertama adalah mengganti jumlah dimensi entropi permutasi. Skenario kedua adalah mengganti jumlah kluster pada K-means *clustering*, dan skenario ketiga adalah mengganti jumlah neuron *hidden layer* pada *multilayer* perceptron.

Nilai performa masing-masing uji coba diukur dalam tiga parameter, yaitu akurasi, *sensitivity*, dan *specificity*. Masing-masing nilai performa secara berurutan dihitung dengan (5) hingga (7). Hasil skenario uji coba dapat dilihat pada Tabel 1.

$$Akurasi = \frac{(TP+TN)}{TP+TN+FN+FP} \tag{5}$$

$$Specificity = \frac{TN}{TN+FP} \tag{6}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{7}$$

### VI. KESIMPULAN

Dari hasil pengamatan selama perancangan, implementasi, dan uji coba perangkat lunak, dapat diambil kesimpulan

bahwa model klasifikasi *multilayer* perceptron akan menghasilkan performa yang maksimal ketika dilakukan eksekusi pada Algoritma entropi permutasi dengan jumlah dimensi antara 3 hingga 6, dan eksekusi pada Algoritma K-means *clustering* yang telah dimodifikasi dengan jumlah

Tabel 1.  
Hasil Skenario Uji Coba

Dataset	Akurasi (%)	<i>Specificity</i> (%)	<i>Sensitivity</i> (%)
A dan E	96,5	95,45	97,97
B dan E	79	74,26	89,86
C dan E	90,5	87,08	96,09
D dan E	83,5	78,92	92,16

*cluster* antara 3 serta pada jumlah *hidden layer neuron* antara nilai 3.

Model klasifikasi yang paling optimal performanya ketika model klasifikasi antara dataset A dan E dengan akurasi sebesar 96,5%, *specificity* sebesar 95,45%, dan *sensitivity* sebesar 97,97%. Sedangkan model klasifikasi yang paling rendah performanya ketika model klasifikasi antara dataset B dan E dengan akurasi 79%, *specificity* sebesar 74,26%, dan *sensitivity* yang bernilai 89,86%.

#### UCAPAN TERIMA KASIH

Penulis Y.A mengucapkan terima kasih kepada Yayasan Karya Salemba Empat yang telah memberikan dukungan finansial melalui beasiswa Karya Salemba Empat tahun 2013-2014. Penulis juga mengucapkan terima kasih kepada Prof. Ir Handayani Tjandrasa M.Sc., Ph.D dan Isye Arieshanti S.Kom., M.Phil sebagai pembimbing penulis dalam mengerjakan penelitian.

#### DAFTAR PUSTAKA

- [1] G Shorvon, S, Handbook of Epilepsy Treatment, London: Springer, 2000.
- [2] Nicolaou. N, and Gergiou. J, "Detection of Epileptic Electroencephalogram based on Permutation Entropy and Support Vector Machines," *Expert Systems with Applications* 39, 2012.
- [3] Bruzzo. A.G, "Permutation Entropy to Detect Vigilance Changes and Preictal States from Scalp EEG in Epileptic Patients," pp. 3-9, 2008.
- [4] Orhan. U, Hekim. M, and Ozer. M, "EEG Signlas Classification Using the K-means Clustering and Multilayer Perceptron Neural Network Model", *Expert Systems with Applications* 38, pp. 13475-13481, 2011.
- [5] T. Mitchell, and McGraw Hill, "Machine Learning: Artificial Neural Network," pp. 75-100, 1997.
- [6] Andrzejak. R. G, Lehenertz. K, Rieke. C, Mormann. F, David. P, and Elger. C. E, "Indications of Nonlinear Deterministic and finite Dimensional Sturctures in Time ", *Physical Review E*, 2001.