

## ANALISIS KINERJA ALGORITMA MESIN PEMBELAJARAN UNTUK KLASIFIKASI PENYAKIT STROKE MENGGUNAKAN CITRA CT SCAN

Nur Sakinah\*<sup>1</sup>, Tessa Badriyah<sup>2</sup>, Iwan Syarif<sup>3</sup>

<sup>1,2,3</sup>Politeknik Elektronika Negeri Surabaya, INDONESIA

Email: <sup>1</sup>nursakinah@pasca.student.pens.ac.id, <sup>2</sup>tessy@pens.ac.id, <sup>3</sup>iwanarif@pens.ac.id

\*Penulis Korespondensi

(Naskah masuk: 20 April 2020, diterima untuk diterbitkan: 22 Juli 2020)

### Abstrak

Stroke adalah suatu kondisi dimana pasokan darah ke otak terganggu sehingga bagian tubuh yang dikendalikan oleh area otak yang rusak tidak dapat berfungsi dengan baik. Penyebab stroke antara lain adalah terjadinya penyumbatan pada pembuluh darah (stroke iskemik) atau pecahnya pembuluh darah (stroke hemoragik). Pasien yang terkena stroke harus segera ditangani secepatnya karena sel otak dapat mati dalam hitungan menit. Tindakan penanganan stroke secara cepat dan tepat dapat mengurangi resiko kerusakan otak dan mencegah terjadinya komplikasi. Penelitian ini bertujuan untuk mengembangkan perangkat lunak yang dapat membaca dan menganalisis citra CT scan dari otak, dan kemudian secara otomatis memprediksi apakah citra CT scan tersebut stroke iskemik atau stroke hemoragik. Data citra CT scan berasal dari Rumah Sakit Umum Haji Surabaya yang diambil selama periode Januari-Mei 2019 dan berasal dari 102 pasien yang terindikasi stroke. Sebelum data gambar tersebut diolah dengan menggunakan beberapa algoritma mesin pembelajaran, data tersebut melalui tahap pre-processing yang bertujuan untuk meningkatkan kualitas citra meliputi konversi citra, pemotongan citra, penskalaan, greyscaling, penghilangan noise dan augmentasi. Tahap selanjutnya adalah ekstraksi fitur menggunakan metode Gray-Level Co-Occurrence Matrix (GLCM). Penelitian ini juga bertujuan untuk membandingkan kinerja lima algoritma mesin pembelajaran yaitu Naïve Bayes, Logistic Regression, Neural Network, Support Vector Machine dan Deep Learning yang diterapkan untuk memprediksi penyakit stroke. Hasil percobaan menunjukkan bahwa algoritma Deep Learning menghasilkan tingkat performansi paling tinggi yaitu nilai akurasi 96.78%, presisi 97.59% dan recall 95.92%.

**Kata kunci:** deteksi penyakit stroke, deep learning, neural network, support vector machine

## PERFORMANCE ANALYSIS MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION OF STROKE USING CT SCAN IMAGES

### Abstract

Stroke is a condition in which the blood supply to the brain is interrupted so that parts of the body that are controlled by damaged brain areas cannot function properly. Causes of strokes include blockages in blood vessels (ischemic stroke) or rupture of blood vessels (hemorrhagic stroke). Stroke patients must be treated as soon as possible because brain cells can die within minutes. The handling of stroke patients quickly can reduce the risk of brain damage and prevent complications. This study aims to develop software that can read and analyze CT scan images from the brain, and then automatically predict whether the CT scan images are ischemic stroke or hemorrhagic stroke. The CT scan image data came from the Surabaya Hajj General Hospital which was taken during the January-May 2019 period and came from 102 patients who had indicated a stroke. Before the image data is processed using several machine learning algorithms, the data goes through a pre-processing phase which aims to improve image quality including image conversion, image cutting, scaling, greyscaling, noise removal and augmentation. The next step is feature extraction using the Gray-Level Co-Occurrence Matrix (GLCM) method. This study also aims to compare the performance of five machine learning algorithms, namely Naïve Bayes, Logistic Regression, Neural Networks, Support Vector Machines and Deep Learning which are applied to predict stroke. The experimental results show that the deep learning algorithm produces the highest level of performance where the accuracy value is 96.78%, 97.59% precision and 95.92% recall.

**Keywords:** stroke prediction, deep learning, neural network, support vector machine

## 1. PENDAHULUAN

Stroke merupakan penyebab utama mortalitas dan morbiditas yang tinggi dalam menyebabkan kecacatan dan kematian di banyak negara, secara global 5,5 juta orang meninggal karena stroke setiap tahunnya, selain itu lebih dari 13,7 juta stroke baru setiap tahun dan satu dari empat di atas usia 25 diantaranya akan mengalami stroke (WSO, 2016). Stroke merupakan masalah yang cukup serius karena serangan stroke sebagai keadaan darurat medis yang dapat mengancam kecacatan dan kematian pada pasien jika tidak cepat dan tepat dalam penanganannya. Pada penyakit stroke, *neuroimaging* selalu memainkan peranan penting pada diagnosis stroke, mayoritas pasien stroke melakukan pemeriksaan dengan menggunakan modalitas radiologi *Computerized Tomograph Scanning* (CT scan). Namun demikian hasil citra CT scan setiap pasien berbeda-beda sesuai dengan jumlah waktu yang berlalu sejak *onset* stroke. Sehingga seorang radiolog berperan besar dalam manajemen pasien stroke sehingga pengetahuan tentang gambaran radiologis pasien akan menentukan perawatan yang akan dijalani oleh pasien (Yuyun, 2016). Stroke menyebabkan berkurangnya atau terhentinya aliran darah yang mengakibatkan matinya sel otak. Berdasarkan penyebabnya stroke terbagi menjadi dua yaitu stroke iskemik dimana suplai darah berhenti mengalir ke otak karena penyumbatan dan stroke hemoragik dimana terjadi pendarahan pada jaringan otak (Marbun dkk, 2018). Penting untuk menerima diagnosa yang benar sebelum perawatan stroke dimulai, karena perawatan untuk stroke berbeda sesuai jenis stroke. Jika pasien gagal menerima perawatan yang cepat dan tepat maka stroke akan menimbulkan konsekuensi serius dan menyebabkan kerusakan permanen pada otak hingga kematian pada pasien. Perawatan dan diagnosis penyakit stroke dilakukan dengan pemeriksaan klinis, kemudian dilanjutkan dengan melakukan pemeriksaan modalitas radiologi seperti CT Scan (Chin dkk, 2017).

Penelitian terkait yang berhubungan dengan diagnosa dan prediksi penyakit stroke dengan melakukan pengembangan sistem deteksi hanya pada satu jenis stroke yaitu mendeteksi iskemik dini secara otomatis menggunakan algoritma *Convolutional Neural Network* (CNN), hasil klasifikasi memperoleh nilai akurasi 90%, (Chin dkk, 2017).

Penelitian lain menggunakan algoritma CNN menggunakan *open dataset* penyakit stroke dari [www.radiopaedia.org](http://www.radiopaedia.org) untuk mengklasifikasi data pasien dalam tiga kelas yaitu normal, stroke iskemik dan stroke hemoragik melalui gambar CT Scan (Marbun dkk, 2018). Sedangkan peneliti lain dengan menggunakan *dataset* yang sama yaitu melakukan *optimasi hyperparameter* pada algoritma *Deep Learning* untuk meningkatkan akurasi diagnosa

stroke. Optimasi *random search* dan *bayesian search* untuk *tuning hyperparameter* pada *Deep Learning* mampu meningkatkan akurasi hingga mencapai 100% (Badriyah dkk, 2019).

Algoritma *Support Vector Machine* (SVM) memberikan hasil yang memuaskan pada penelitian prediksi penyakit stroke dengan menggunakan kinerja berbagai fungsi *kernel* pada metode SVM. Hasil percobaan diperoleh presisi terbaik pada kernel *linier function* dengan nilai akurasi 91%. Penelitian ini hanya menggunakan data klinis pasien dari rekam medis pasien (Jenna and Kumar, 2017).

Penelitian lainnya menggunakan pendekatan algoritma mesin pembelajaran berdasarkan citra CT scan adalah klasifikasi terhadap penyakit kanker paru-paru menggunakan perbandingan beberapa algoritma yaitu *K-Nearest Neighbour*, *Super Vector Machine*, *Decision Tree*, *Naïve Bayes*, *Stochastic Gradient Descent*, *Random Forest* dan *Multilayer Perceptron* yang merupakan salah satu jenis arsitektur *Deep Learning*. Terdapat lima tahap yang berbeda yaitu akuisisi gambar, pemrosesan gambar, segmentasi, ekstraksi fitur, klasifikasi, kemudian melakukan evaluasi terhadap kinerja algoritma. Hasil klasifikasi *dataset* citra medis sebanyak 15.750 dengan distribusi 6.910 untuk kelas *benign* dan 8.84% untuk kelas *malignant*, hasil evaluasi kinerja algoritma didapat akurasi tertinggi 88.55% dengan pendekatan *Multilayer Perceptron* (Singh dan Gupta, 2018).

Penelitian lainnya membandingkan metode *Neural Network* dengan algoritma *Naïve Bayes*, *K-Nearest Neighbors*, *Linear Regression*, *Logistic Regression*, dan *Support Vector Machine* untuk melakukan klasifikasi data preeklampsia dengan 17 atribut yang dianggap mempengaruhi karakteristik risiko terjadinya preeklampsia. Hasilnya algoritma *Neural Network* mencapai nilai akurasi tertinggi sebesar 96,66% (Tahir dkk, 2018a)

Penelitian lainnya membahas perbandingan teknik *data mining* dengan menggunakan dua metode yang berbeda yaitu *Logistic Regression* dan metode *Bayesian* untuk memprediksi tingkat resiko diabetes, menggunakan data pasien yang memiliki 9 atribut. Metode *Bayesian* memiliki akurasi tertinggi dengan nilai 91.0% metode *Logistic Regression* memiliki akurasi dengan nilai 90.8% (Wicaksono dkk, 2016)

Berdasarkan hasil penelitian sebelumnya telah menunjukkan akurasi yang baik pada Algoritma *Deep Learning* dengan arsitektur CNN maupun *Deep Learning* dengan arsitektur *Multilayer Perceptron*, *Neural Network*, *Logistic Regression*, *Naïve Bayes*, dan *Support Vector Machine* untuk memprediksi suatu penyakit, namun hasil tersebut diperoleh berdasarkan penggunaan data dan proses yang berbeda-beda, mulai dari data masukan yang digunakan, atribut atau fitur yang dimiliki hingga proses implementasi algoritma terhadap data

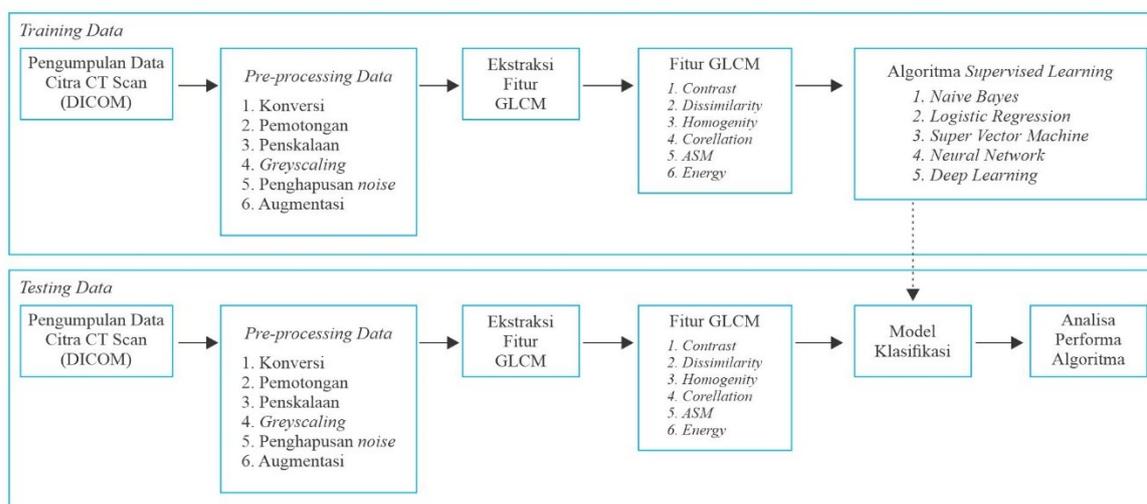
masuk. Karena setiap proses yang dilakukan akan mempengaruhi hasil prediksi. Penelitian ini bertujuan untuk mengembangkan perangkat lunak yang dapat membaca dan menganalisis citra CT scan dari otak, dan kemudian secara otomatis memprediksi apakah citra CT scan tersebut stroke iskemik atau stroke hemoragik dengan membandingkan kinerja lima algoritma mesin pembelajaran yaitu *Naïve Bayes*, *Logistic Regression*, *Neural Network*, *Support Vector Machine* dan *Deep Learning* yang diterapkan untuk memprediksi penyakit stroke.

Prediksi penyakit stroke menggunakan data yang belum teruji sebelumnya dari Rumah Sakit Umum Haji Surabaya. Data CT scan diekstraksi menggunakan metode tekstur fitur *Gray-Level Co-Occurrence Matrix* (GLCM) sebagai data masukan. Untuk mendapatkan performa algoritma terbaik, prediksi penyakit stroke dengan proses klasifikasi diimplementasikan menggunakan *library scikit-learn* untuk algoritma *Naive Bayes*, *Logistic Regression*, *Support Vector Machine*, *Neural Network* dan *library keras* untuk *Deep Learning*. Kemudian menganalisis beberapa metode optimal untuk dibandingkan kinerja berbagai pendekatan algoritma mesin pembelajaran menggunakan empat parameter yaitu *accuracy*, *precision*, *recall*, *f1-*

*measure* dengan *k-fold cross-validation* dan *leave-one-out cross-validation* sebagai validasi untuk mengoptimalkan kinerja algoritma sehingga menghasilkan akurasi prediksi yang tinggi.

## 2. METODE PENELITIAN

Perancangan sistem yang diusulkan terdiri dari beberapa tahap yaitu (1) pengumpulan data, (2) *pre-processing* data, penelitian ini menggunakan data citra CT scan yang perlu dilakukan *pre-processing* data citra, sehingga dapat meningkatkan kualitas data citra dengan tahapan konversi citra DICOM ke jpg, pemotongan, penskalaan, *greyscaling*, penghapusan *noise* dan augmentasi (3) ekstraksi fitur dan (4) analisis performa algoritma mesin pembelajaran. Implementasi prediksi penyakit stroke menggunakan *library scikit-learn* untuk empat algoritma yaitu *Naive Bayes*, *Logistic Regression*, *Neural Network*, *Support Vector Machine* dan *library keras* untuk algoritma *Deep Learning* menggunakan bahasa pemrograman *python*. Gambar 1 merupakan perancangan sistem dari penelitian ini, setiap langkah akan dijelaskan lebih terinci dibagian 2.1 -2.5



Gambar 1. Perancangan Sistem

### 2.1. Pengumpulan Data

Tahapan pengumpulan data penelitian ini adalah mengumpulkan data citra hasil dari CT scan otak pasien yang terdiri dari data citra stroke iskemik dan data citra stroke hemoragik. Data citra CT scan berasal dari Rumah Sakit Umum Haji Surabaya yang diambil selama periode Januari-Mei 2019 dan berasal dari 102 pasien yang terindikasi stroke. Citra yang diambil memiliki ketebalan *slice thickness* 5.0 mm, pengambilan data dengan ukuran tersebut didasarkan pada ukuran hasil citra CT scan memiliki batas ketebalan antara 1.0 – 10.0 mm sebagai parameter *Digital Imaging and Communication in*

*Medicine* (DICOM), dengan ketebalan 5.0 mm didapatkan 25 citra setiap pasien. Dari 25 citra yang diambil satu pasien didapat citra yang baik dimana terlihat objek stroke antara 1-10 citra. Data yang dihasilkan dari 102 pasien adalah sebanyak 274 yang terbagi dua kelas yaitu stroke sebanyak 248 dan stroke hemoragik sebanyak 26 citra.

### 2.2. Pre-processing Data

*Pre-processing* data merupakan tahapan untuk meningkatkan kualitas dari data citra CT scan pasien stroke. Data yang digunakan berupa data citra dengan format DICOM. *Pre-processing* data pada

CT scan dilakukan melalui lima langkah berikut yaitu: (1) konversi (2) pemotongan (3) penskalaan yang berfungsi untuk mengatur ukuran piksel yang digunakan (4) *greyscaling* yang berfungsi untuk penyeragaman derajat keabuan (5) penghapusan *noise* berfungsi untuk menghilangkan gangguan yang ditemukan pada data CT scan. (6) augmentasi dengan tujuan untuk memperluas *dataset* pelatihan. Tahapan *pre-processing data* dapat masing-masing proses akan dijelaskan lebih terinci dibagian 2.2.1-2.2.5.

### 2.2.1. Konversi

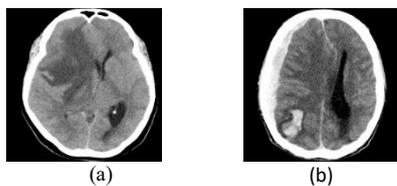
Pada proses konversi citra dilakukan menggunakan aplikasi yang disediakan dari rumah sakit yaitu *syngo fastview*. Data yang diperoleh dari rumah sakit masih dalam bentuk format DICOM, data citra DICOM hanya bisa terlihat menggunakan komputer yang terintegrasi dengan alat CT scan, sehingga perlu dilakukan konversi dengan menggunakan *syngo fastview* untuk mengkonversi citra DICOM ke format jpg. Berikut merupakan hasil konversi *syngo fastview* pada gambar 2.



Gambar 2. Konversi (a) iskemik, (b) hemoragik

### 2.2.2. Pemotongan

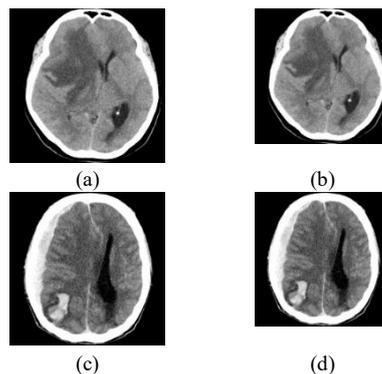
Proses kedua yaitu pemotongan data citra, hal ini dilakukan karena hasil konversi *syngo fastview* menghasilkan dimensi 1105 x 650, proses pemotongan dilakukan untuk lebih fokus pada objek stroke.



Gambar 3. (a) citra pemotongan iskemik (b) citra pemotongan hemoragik

### 2.2.3. Penskalaan

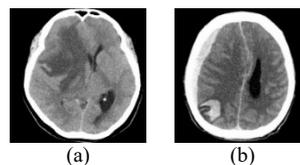
Proses pemotongan citra memberikan hasil dimensi citra yang berbeda-beda, sehingga digunakan proses penskalaan agar dimensi citra menjadi seragam dan mengembalikan nilai dimensi citra DICOM yaitu 512x512. Pada proses pemotongan citra menghasilkan nilai yang tidak sama setiap data karena data konversi setiap pasien stroke memiliki nilai tengah yang berbeda-beda.



Gambar 4 (a) sebelum penskalaan iskemik, (b) penskalaan iskemik, (c) sebelum penskalaan hemoragik, (d) penskalaan hemoragik

### 2.2.4. Greyscaling

*Greyscaling* berfungsi untuk penyeragaman derajat keabuan citra. Hasil citra radiografi memiliki nilai piksel citra digital antara 0 hingga 255, sehingga dari proses penskalaan citra akan dilakukan proses *greyscaling* yang menggunakan warna hitam sebagai warna minimal yang direpresentasikan dengan 0 dan warna putih sebagai warna maksimal yang direpresentasikan dengan 1.



Gambar 5. (a) *greyscaling* iskemik, (b) *greyscaling* hemoragik

### 2.2.5. Penghapusan Noise

Penghapusan *noise* dalam citra medis adalah tugas penting dalam *pre-processing* citra untuk menghilangkan *noise* dan meningkatkan gambar lebih baik sehingga terjadi perbaikan pada gambar medis (Vijaya dan Suhasuni, 2014). penelitian ini, metode yang digunakan untuk penghapusan *noise* menggunakan *bilateral filtering* yang didefinisikan sebagai rata-rata piksel tertimbang yang memperhitungkan variasi intensitas untuk mempertahankan tepian. Penelitian yang menghasilkan perbaikan citra dengan melakukan penghapusan *noise* untuk citra CT scan stroke hasil terbaik terdapat pada *bilateral filtering* mendapatkan dari perbandingan 2 filter lainnya yaitu *median filtering* dan *gaussian filtering* yang dievaluasi menggunakan PSNR dan MSE (Badriyah dkk, 2019b)

### 2.2.6. Augmentasi

Augmentasi adalah proses memperbanyak citra tanpa menghilangkan esensinya karena kelangkaan data yang tersedia. Proses ini dilakukan karena data stroke sulit untuk didapatkan, terutama data stroke hemoragik. Praktik terkini untuk melakukan augmentasi data adalah kombinasi transformasi

gambar *affine* dan modifikasi warna (Mikołajczyk A. dan M. Grochowski, 2018). Metode umum transformasi *affine* yang digunakan dalam augmentasi yaitu *random brightness*, *horizontal flip*, *vertical flip* dan *random rotation*. Setelah dilakukan augmentasi, data yang digunakan pada setiap kelas stroke memiliki jumlah yang berbeda yaitu 2500 data citra untuk stroke hemoragik dan 2500 data untuk stroke iskemik.

### 2.3. Ekstraksi Fitur

Ekstraksi fitur menggunakan metode *Gray-Level Co-Occurrence Matrix* (GLCM). Metode GLCM baik dalam mewakili karakteristik tekstur citra dan perhitungan fitur tekstur yang terkait. Ekstraksi fitur data citra terdiri dari piksel dengan intensitas tingkat abu-abu tertentu. Terdapat 6 fitur yang digunakan dalam ekstraksi ciri GLCM (Badriyah dkk, 2019a). Masing-masing fitur akan dijelaskan lebih terinci pada point 2.3.1-2.3.6

#### 2.3.1. Contrast

*Contrast* merupakan perhitungan jumlah keberagaman intensitas keabuan dalam citra, jika piksel tetangganya mempunyai nilai yang sama, *contrast* akan bernilai 0. *Contrast* dirumuskan sebagai berikut:

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \quad (1)$$

Keterangan

- P = matriks *co-occurrence*
- i, j = indeks pada matriks
- N = level keabuan matriks *co-occurrence*

#### 2.3.2. Dissimilarity

*Dissimilarity* adalah hasil mengukur perbedaan setiap piksel, hasilnya akan tinggi jika teksturnya acak dan akan rendah jika nilainya seragam. *Dissimilarity* dirumuskan sebagai berikut:

$$Dissimilarity = \sum_{i,j=0}^{N-1} P_{i,j}|i-j| \quad (2)$$

Keterangan

- P = matriks *co-occurrence*
- i, j = indeks pada matriks
- N = level keabuan matriks *co-occurrence*

#### 2.3.3. Homogeneity

*Homogeneity* nilai tinggi ada ketika semua piksel memiliki nilai yang sama. Fitur ini kebalikan dari kontras, yang sangat bagus jika memiliki nilai piksel yang sama ketika *energy* diperbaiki. *Homogeneity* dirumuskan sebagai berikut:

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i+j)^2} \quad (3)$$

Keterangan

- P = matriks *co-occurrence*
- i, j = indeks pada matriks
- N = level keabuan matriks *co-occurrence*

#### 2.3.4. Correlation

*Correlation* merupakan hasil pengukuran linearitas dari sejumlah pasangan piksel. *Correlation* dirumuskan sebagai berikut:

$$Correlation = \sum_{i,j=0}^{N-1} P_{i,j} \left[ \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (4)$$

Keterangan

- P = matriks *co-occurrence*
- i, j = indeks pada matriks
- N = level keabuan matriks *co-occurrence*
- M = nilai mean piksel
- $\sigma$  = nilai varian piksel

#### 2.3.5. Angular Second Moment (ASM)

ASM merupakan hasil pengukuran dari keseragaman. Jika nilai piksel mirip satu sama lain *energy* akan bernilai tinggi, sebaliknya akan bernilai kecil. menandakan nilai dari GLCM normalisasi adalah heterogen. ASM dirumuskan sebagai berikut:

$$ASM = \sum_{i,j=0}^{N-1} P_{i,j}^2 \quad (5)$$

Keterangan

- P = matriks *co-occurrence*
- i, j = indeks pada matriks
- N = level keabuan matriks *co-occurrence*

#### 2.3.6. Energy

*Energy* merupakan hasil pengukuran keseragaman tekstur (pengulangan pasangan piksel). *Energy* dirumuskan sebagai berikut:

$$Energy = \sqrt{ASM} \quad (6)$$

Perhitungan fitur tekstur menyediakan ukuran untuk variasi intensitas pada piksel dan dapat dihitung menggunakan GLCM. Dua parameter digunakan untuk menghitung GLCM yang dikenal sebagai (1) jarak relatif antara pasangan piksel 'd' diukur dalam jumlah piksel dan (2) relatif orientasi pasangan piksel 'd', nilai dari parameter  $\theta$  dikuantisasi dalam empat arah dengan nilai  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$  (Singh dan Gupta, 2018). Fitur GLCM memberikan deskripsi gambar tekstur yang baik, tetapi membutuhkan sumber daya pemrosesan yang kuat. Banyak pendekatan telah diusulkan untuk menghadapi masalah ini, dengan mengurangi kuantisasi yang dilakukan sehingga tidak

menyebabkan dimensi matriks GLCM menurun. Selain itu tahapan *pre-processing* ini akan mengurangi jarangnya matriks GLCM. Pendekatan lain dengan pertimbangan pasangan jarak (d) dan arah (θ) yang lebih sedikit, ini menunjukkan bahwa fitur diekstraksi dari matriks GLCM sesuai dengan “(d, θ) = (1,0)” sudah cukup untuk dijelaskan tekstur sebagian besar citra (Mirzapour F. dan Ghassemian H, 2015).

**2.4. Analisis Performa Algoritma Mesin Pembelajaran**

Analisis performa akan diterapkan secara langsung dari data *training* dan data *testing* dalam proses klasifikasi. Prediksi stroke pada otak dengan proses klasifikasi pada penelitian ini menghasilkan kelas stroke iskemik dan kelas stroke hemoragik. Untuk mengevaluasi kinerja algoritma, dibutuhkan nilai-nilai matriks antara lain sebagai berikut (Badriyah dkk, 2019):

Table 1. *Performance confusion matrix*

		Label Prediksi	
		Positive	Negative
Label	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

*Metrics score* yang digunakan untuk mengevaluasi algoritma adalah *performance measurement method* menggunakan 4 parameter antara lain *accuracy*, *precision*, *recall*, dan *f1 Measure*. Masing-masing parameter akan dijelaskan lebih terinci dibagian 2.4.1. – 2.4.4.:

**2.4.1. Accuracy**

*Accuracy* merupakan *metric score* untuk menghitung akurasi dari hasil klasifikasi. *Accuracy* dirumuskan sebagai berikut:

$$Accuracy: \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

**2.4.2. Precision**

*Precision* merupakan *metric score* yang digunakan untuk menghitung presisi atau data prediksi. *Precision* dirumuskan sebagai berikut:

$$Precision: \frac{TP}{TP + FP} \tag{8}$$

**2.4.3. Recall**

*Recall* merupakan *metric score* yang digunakan untuk menghitung jumlah predeksi kelas yang sama. *Recall* dapat dirumuskan sebagai berikut:

$$Recall: \frac{TP}{TP + FN} \tag{9}$$

**2.4.4. F1-Measure**

*F1-measure* merupakan *metric score* yang menghitung nilai keseimbangan dari *precision* dan *recall*, *F1-measure* dirumuskan sebagai berikut:

$$F1 - measure: 2 \frac{(Precision * Recall)}{(Precision + Recall)} \tag{10}$$

**2.5. Keluaran Performa Klasifikasi**

Keluaran performa klasifikasi menggunakan validasi sampling, proses ini dilakukan untuk membagi antara data *training* dan *testing* sebelum algoritma klasifikasi membangun model. Teknik sampling yang digunakan dalam penelitian ini adalah *k-fold cross validation* dengan k=10 dan *leave-one-out cross validation* (LOO)

Teknik validasi *k-fold cross validation* dalam percobaan ini memiliki jumlah k=10. Persamaan *k-fold cross validation* dirumuskan sebagai berikut:

$$E = \frac{1}{10} \sum_{i=1}^{10} E_i \tag{11}$$

Keterangan:

- E = Eksperimen
- E<sub>i</sub> = jumlah percobaan-i
- N = jumlah contoh

Sedangkan validasi sampling *leave one out cross validation* (LOO) menggunakan seluruh data sebagai nilai k pada *k-fold cross validation*. Persamaan LOO dirumuskan sebagai berikut:

$$E = \frac{1}{N} \sum_{i=1}^N E_i \tag{12}$$

Keterangan:

- E = Eksperimen
- E<sub>i</sub> = jumlah percobaan-i
- N = jumlah contoh

**2.6. Algoritma Mesin Pembelajaran untuk Klasifikasi**

Pengujian performa klasifikasi menggunakan 5 algoritma mesin pembelajaran untuk dapat membandingkan hasil performa masing-masing algoritma, menggunakan *library scikit-learn* pada 4 algoritma yaitu *Naive Bayes*, *Logistic Regression*, *Support Vector Machine*, *Neural Network*, dan *library keras* untuk *Deep Learning*. Algoritma mesin pembelajaran dijelaskan secara terinci pada bagian 2.6.1-2.6.5

**2.6.1. Naive Bayes (NB)**

*Naive Bayes* merupakan metode *supervised learning* untuk klasifikasi serta memecahkan

masalah diagnostik dan prediksi menggunakan probabilitas dan statistik berdasarkan teorema *Bayes* dengan asumsi independensi yang kuat antara fitur yang dimiliki sebuah objek. Teorema *Naïve Bayes*, dirumuskan sebagai berikut (Fahrudin dkk, 2017):

$$P(C|F_1, \dots, F_n) = \frac{P(C)P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)} \quad (14)$$

Keterangan

- P = probabilitas setiap kelas  
 C = variable mewakili kelas  
 F1...Fn = variabel mewakili karakteristik arah untuk klasifikasi

### 2.6.2. Logistic Regression (LR)

*Logistic Regression* adalah metode umum untuk menciptakan probabilitas model prediksi dari suatu kejadian. *Logistic Regression* hanya digunakan jika variabel keluaran dari model yang digunakan didefinisikan sebagai kategori *biner*. Dalam rumus persamaan, Pbj adalah probabilitas yang diprediksi oleh kode 1 dan (1-Pbj) diperkirakan probabilitas dengan keputusan lain dan dikodekan dengan 0 (Wicaksono dkk, 2016). *Logistic Regression* dapat dirumuskan sebagai berikut:

$$\log\left(\frac{P_{bj}}{1 - P_{bj}}\right) = a + \beta_1 \cdot X_{1j} + \beta_2 \cdot X_{2j} + \dots + \beta_n \cdot X_{nj} \quad (15)$$

Keterangan

- a = *intercept*  
 X1j...Xnj = atribut independen dalam data -j  
 β1...βn = kemiringan untuk atribut independen  
 n = nomor atribut independen.  
 j = jumlah data

### 2.6.3. Support Vector Machine (SVM)

Metode SVM menggunakan metode *kernel* untuk mencari *hyperplane* pemisah optimal ke ruang vektor berdimensi lebih tinggi (Muis dan Affandes, 2015). SVM menggunakan dua konsep utama untuk memecahkan masalah yaitu pemisahan margin besar dan fungsi *kernel* (Vanitha dkk, 2015) Pemilihan fungsi *kernel* yang tepat sangat penting, karena fungsi *kernel* menentukan ruang fitur dimana contoh pelatihan akan diklasifikasikan. Fungsi *kernel* dalam SVM dirumuskan dengan  $K(x_i, x_j) \equiv \Phi(x_i)^T \Phi(x_j)$ , dalam penelitian ini fungsi *kernel* yang digunakan adalah fungsi *kernel Radial Basis Function* (RBF) dengan parameter C dan *Gamma*. Fungsi *kernel* RBF di rumuskan sebagai berikut (Srivastava, 2010):

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (22)$$

$K(x_i, x)$  = pemetaan variabel amatan x ke-i dan x oleh fungsi *kernel* K,  $\gamma$  adalah parameter *kernel*.

### 2.6.4. Neural Network (NN)

*Neural Network* atau *Artificial Neural Networks* (ANN) adalah kinerja yang meniru jaringan sel-sel saraf (*neuron*) di otak manusia. komputasi metode ini adalah dalam hubungan antar *neuron*. setidaknya ada dua lapisan yang terdiri dari layer masukan dan layer keluaran sedangkan pada *Multilayer Neural Network*, layer masukan dan layer keluaran ada *hidden layer* yang berisi nilai-nilai *node* keluaran yang digunakan sebagai nilai *node* masukan lain (Tahir dkk, 2018). *Neural Network* dapat dirumuskan sebagai berikut:

$$Z = a_1w_1 + a_2w_2 + \dots + a_k w_k + b \quad (20)$$

Keterangan

- a1 = nilai masukan  
 w1 = nilai bobot  
 B = variabel mewakili karakteristik arah untuk klasifikasi

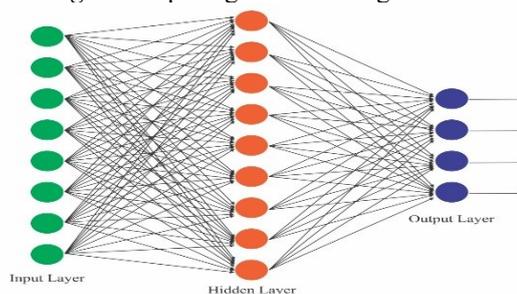
Rumus yang digunakan sebagai *activation function* (AF) dalam penelitian ini adalah *sigmoid*, dapat dirumuskan sebagai berikut:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (21)$$

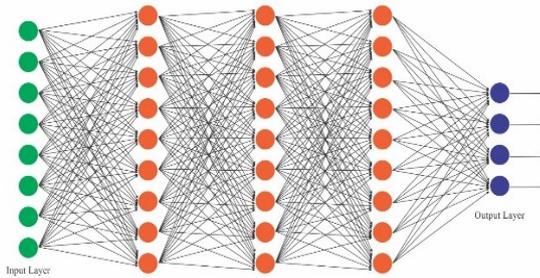
Karakteristik suatu fungsi aktivasi yaitu kontinu dan monoton menurun. Fungsi *sigmoid* yang memiliki range (0,1).

### 2.6.5. Deep Learning (DL)

*Deep Learning* adalah metode *representation-learning* terdiri dari seperangkat algoritma yang dimodelkan dengan banyak level representasi, dari masukan mentah ke tingkat representasi yang lebih tinggi (Lecun, 2015). *Deep learning* memungkinkan model komputasi terstruktur dalam beberapa lapisan untuk mempelajari data yang diwakili oleh tingkat abstraksi yang berbeda. Representasi berlapis ini dipelajari dari model yang disebut *Neural Network*, yang terstruktur secara literal dan lapisan yang ditumpuk satu per satu (Tahir dkk, 2018). Ilustrasi perbedaan *Neural Network* dan *Deep Learning* dilihat pada gambar 6 dan gambar 7.



Gambar 6. Ilustrasi *Neural Network*



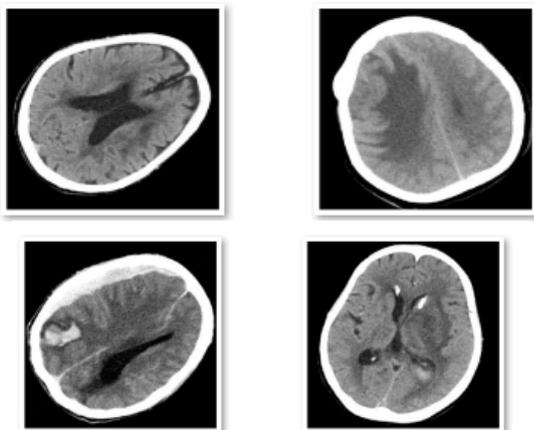
Gambar 7. Ilustrasi Deep Learning

### 3. HASIL DAN PEMBAHASAN

Pada penelitian ini, melakukan proses klasifikasi menggunakan data berupa citra CT scan untuk memprediksi 2 jenis stroke yaitu stroke iskemik dan stroke hemoragik dengan melakukan ekstraksi fitur yang menghasilkan 6 fitur antara lain *contrast*, *dissimilarity*, *homogeneity*, *correlation*, *ASM*, *energy*, keenam fitur tersebut menjadi fitur pembelajaran data *testing* dan *training* yang akan diproses menggunakan lima algoritma mesin pembelajaran menggunakan validasi sampling data *k-fold cross validation* dan *LOO* yang akan dievaluasi menggunakan 4 parameter *metric score* yaitu *accuracy*, *precision*, *recall*, dan *f1 measure*, yang akan menampilkan hasil performa kasifikasi dari algoritma mesin pembelajaran. Berikut merupakan pembahasan hasil proses ekstraksi fitur dan hasil klasifikasi yang dilakukan dapat dilihat secara terperinci pada bagian 3.1.- 3.3

#### 3.1. Hasil Ekstraksi fitur menggunakan GLCM

Gambar 8 menunjukkan sampel citra CT scan untuk diproses ekstraksi fitur. Sampel citra stroke iskemik dan sampel citra stroke hemoragik



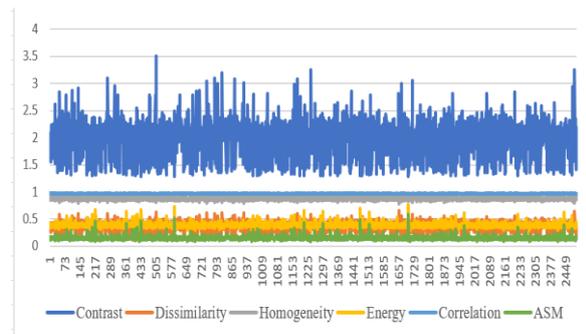
Gambar 8. Sampel data citra Ct scan untuk ekstraksi fitur

Pada table 2 menjelaskan hasil nilai ekstraksi 6 fitur menggunakan parameter “(d, θ) = (1,0)” dengan jumlah sampel 15 sampel dari 5000 data ekstraksi. data sampel berasal dari 7 data stroke hemoragik dengan kelas 0 dan 8 data stroke iskemik dengan kelas 1.

Table 2. Hasil ekstraksi fitur glcm (1) *contrast*, (2) *dissimilarity* (3) *homogeneity* (4) *correlation*, (5) *ASM*, (6) *energy*

1	2	3	4	5	6	Kelas
1.844	0.361	0.878	0.348	0.971	0.121	0
2.030	0.343	0.892	0.416	0.973	0.173	1
1.852	0.365	0.874	0.362	0.972	0.131	1
2.141	0.445	0.849	0.354	0.967	0.126	0
2.083	0.430	0.851	0.338	0.967	0.114	1
2.073	0.380	0.875	0.380	0.969	0.144	1
1.701	0.328	0.887	0.357	0.970	0.127	0
1.812	0.322	0.895	0.356	0.964	0.127	1
2.015	0.385	0.869	0.338	0.969	0.114	1
1.709	0.328	0.889	0.362	0.970	0.131	0
1.939	0.372	0.874	0.345	0.971	0.119	0
2.067	0.394	0.868	0.347	0.969	0.120	0
1.555	0.263	0.912	0.365	0.963	0.133	1
2.228	0.404	0.868	0.381	0.967	0.145	1

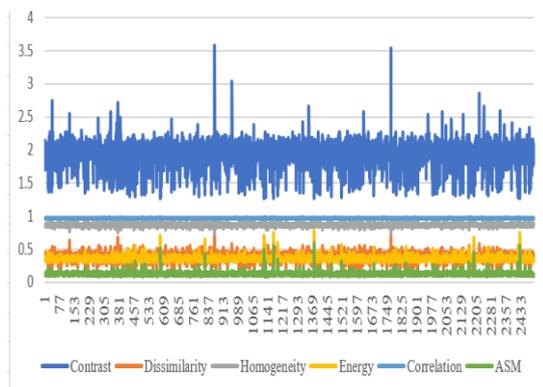
Hasil ekstraksi fitur secara keseluruhan digambarkan dalam bentuk grafik yang menjelaskan nilai 6 fitur dari GLCM yaitu *contrast*, *dissimilarity*, *homogeneity*, *corellation*, *ASM*, *energy*. Pada gambar 9 merupakan grafik seluruh distribusi nilai ekstraksi fitur stroke iskemik yang berjumlah 2500, pada nilai fitur *contrast* menunjukkan angka terkecil 1,283 dan angka 3,551 sebagai nilai terbesar fitur, *dissimilarity* menunjukkan nilai terkecil pada angka 0,107 dan angka 0.651 sebagai nilai terbesar, fitur *homogeneity* menunjukkan nilai terkecil pada angka 0,777 dan angka 0,980 sebagai nilai terbesar, fitur *correlation* menunjukkan nilai terkecil pada angka 0,956 dan angka 0,976 sebagai nilai terbesar, fitur *ASM* menunjukkan nilai terkecil pada angka 0.085 dan angka 0.584 sebagai nilai terbesar. fitur *energy* menunjukkan nilai terkecil pada angka 0,292 dan angka 0,764 sebagai nilai terbesar.



Gambar 9. Hasil ekstraksi fitur stroke iskemik

Sedangkan pada Gambar 10. merupakan grafik seluruh distribusi nilai ekstraksi fitur stroke hemoragik yang berjumlah 2500, pada nilai fitur *contrast* menunjukkan angka terkecil 1,268 dan angka 3,577 sebagai nilai terbesar, fitur *dissimilarity* menunjukkan nilai terkecil pada angka 0,093 dan angka 0.820 sebagai nilai terbesar, fitur *homogeneity* menunjukkan nilai terkecil pada angka 0,773 dan angka 0,987 sebagai nilai terbesar, fitur *correlation* menunjukkan nilai terkecil pada angka 0,301 dan angka 0,766, fitur *ASM* menunjukkan nilai terkecil pada angka 0,951 dan angka 0,976 sebagai nilai terbesar, fitur *energy* menunjukkan nilai terkecil

pada angka 0.091 dan angka 0.587 sebagai nilai terbesar.



Gambar 10. Hasil ekstraksi fitur stroke hemoragik

### 3.2. Keluaran Performa Klasifikasi Algoritma

Implementasi algoritma mesin pembelajaran untuk prediksi stroke dalam proses klasifikasi menggunakan library *scikit-learn* pada algoritma *Naive Bayes*, *Logistic Regression*, *Support Vector Machine*, *Neural Network*, dan library *keras* untuk *Deep Learning* berikut merupakan keluaran performa klasifikasi masing-masing algoritma:

#### 1. *Naive Bayes*

Metode ini akan menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari data yang dimasukkan. Algoritma *Naive Bayes* dilakukan dengan menghitung jumlah kelas, menentukan nilai probabilitas dari setiap atribut berdasarkan masing-masing kelas kemudian proses untuk mendapatkan prediksi data baru dengan melakukan perkalian pada nilai probabilitas setiap atribut dengan masing-masing kelas. Performa klasifikasi dapat dilihat pada table 3.

Tabel 3. Hasil performa *Naive Bayes*

Validasi	Accuracy	Precision	Recall	F1-measure
10-Fold	69.34%	72.29%	69.34%	70.78%
LOO	83.72%	86.24%	80.24%	83.13%

Dari tabel 3 dapat dilihat bahwa performa *accuracy* dengan validasi LOO mendapatkan hasil yang baik dengan nilai 83.72% dibandingkan dengan 10-fold yang hanya mendapatkan hasil dengan nilai 69.34%.

#### 2. *Logistic Regression*

*Logistic Regression* digunakan untuk menganalisis *multivariat* yang berguna untuk memprediksi variabel *independent*, jenis *Logistic Regression* yang digunakan dalam klasifikasi adalah *binary Logistic Regression* yang hanya memiliki keluaran yaitu 0 dan 1. Hasil performa *Logistic Regression* dapat dilihat pada tabel 4.

Tabel.4 Hasil performa *Logistic Regression*

Validasi	Accuracy	Precision	Recall	F1-measure
10-Fold	79.80%	79.85%	79.80%	79.82%
LOO	79.86%	7.86%	79.87%	79.86%

Dari tabel 4 dapat dilihat bahwa performa *accuracy Logistic Regression* dengan validasi LOO mendapatkan hasil dengan nilai 79.86% hanya berbeda 0,6 lebih tinggi dari 10-fold yang hanya mendapatkan nilai 79.80%.

#### 3. *Super Vector Machine (SVM)*

Metode SVM menggunakan *hyperplane* pemisah optimal untuk memisahkan dua kelas yang berbeda, pada klasifikasi ini menggunakan fungsi kernel RBF. Hasil SVM dapat dilihat pada tabel 5.

Tabel.5 Hasil performa *Super Vector Machine*

Validasi	Accuracy	Precision	Recall	F1-measure
10-Fold	82.22%	82.33%	82.22%	82.27%
LOO	82.22%	82.22%	82.28%	82.25%

Dari tabel 5 dapat dilihat bahwa performa *accuracy Super Vector Machine* dengan validasi 10-fold dan LOO memiliki hasil yang sama dengan nilai 82.22% nilai *precision* dan *f-measure* yang paling tinggi terdapat pada validasi 10-fold.

#### 4. *Neural Network*

Metode *Neural Network* pada klasifikasi ini menggunakan arsitektur *Multilayer Perceptron*, berikut parameter yang digunakan (1) *hidden layer: 1*, (2) jumlah *neuron: 100*, (3) *epoch: 200*, (4) *optimizer: adam*, (5) *bath size: 32*, (6) *learning\_rate: 0.0001*, (7) fungsi aktivasi *hidden: Relu*, (8) fungsi aktifasi *output: Sigmoid*. Hasil performa klasifikasi *Neural Network* dapat dilihat pada tabel 6.

Tabel 6. Hasil Performa *Neural Network*

Validasi	Accuracy	Precision	Recall	F1-measure
10-Fold	85.34%	86.71%	93.15%	89.76%
LOO	86.22%	88.03%	83.84%	85.88%

Dari tabel 6 menunjukkan bahwa performa *accuracy* dengan validasi LOO dari *Neural Network* memiliki tingkat akurasi tinggi jika dibandingkan dengan *Naive Bayes*, *Logistic Regression* dan SVM. Hasil *accuracy* dengan nilai 86.22% lebih besar dibandingkan dengan 10-fold mendapatkan nilai akurasi dibawahnya yaitu 85.34%.

#### 5. *Deep Learning*

Metode *Deep Learning* pada klasifikasi ini menggunakan parameter yang diinisialisasi berdasarkan pendekatan empiris dari banyak

percobaan yang telah dilakukan sebelumnya. Parameter *deep learning* diinisialisasi sebagai berikut (1) *hidden layer*:5, (2) jumlah *neuron*: 25 setiap *hidden layer*, (3) *epoch*: 1000, (4) *optimizer*: *adam*, (5) *bath size*: 32, (6) *dropout*:0.9 (7) *learning\_rate*: 0.0001, (8) fungsi aktivasi *hidden* “Relu”, (9) fungsi aktifasi *output* “Sigmoid”. Hasil performa klasifikasi *deep learning* dapat dilihat pada tabel 7.

Tabel.7 Hasil performa *Deep Learning*

Validasi	Accuracy	Precision	Recall	F1-measure
10-Fold	95.16%	93.69%	95.92%	96.00%
LOO	96.78%	97.59%	95.92%	96.75%

Pada tabel 7 dapat dilihat bahwa *accuracy* dengan validasi LOO dari *Deep Learning* memiliki tingkat akurasi yang paling tinggi dengan nilai 96.78% lebih besar dibandingkan dengan *10-fold* dengan nilai akurasi 95.16%.

Performa terbaik dari hasil perbandingan dari *10-fold cross validation* dan LOO dari masing masing algoritma unggul pada teknik sampling LOO, namun algoritma SVM unggul pada *10-fold* dan algoritma *Deep Learning* menunjukkan bahwa hasil performa paling tinggi diantara algoritma lainnya. Hasil perbandingan performa klasifikasi terbaik algoritma dapat dilihat pada tabel 8.

Tabel 8 Hasil perbandingan performa klasifikasi terbaik

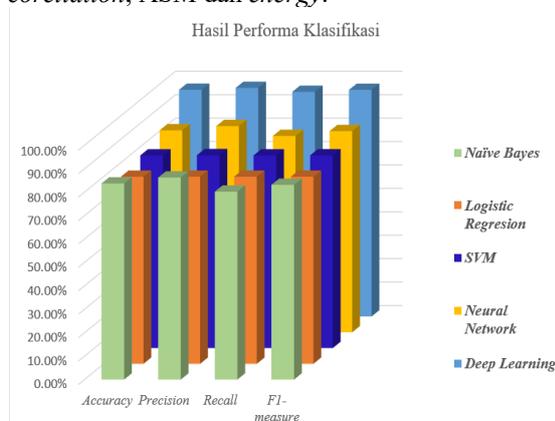
Algoritma	Accuracy	Precision	Recall	F1-measure
<i>Naïve Bayes</i>	83.72%	86.24%	80.24%	83.13%
<i>Logistic Regresion</i>	79.86%	7.86%	79.87%	79.86%
<i>SVM</i>	82.22%	82.33%	82.22%	82.27%
<i>Neural Network</i>	86.22%	88.03%	83.84%	85.88%
<i>Deep Learning</i>	96.78%	97.59%	95.92%	96.75%

Gambar 11 merupakan grafik perbandingan hasil performa klasifikasi pada lima algoritma mesin pembelajaran terbaik dari validasi sampling *10-fold cross validation* dan LOO, *Deep Learning* menunjukkan hasil signifikan pada seluruh performa *metric score* dibandingkan dengan algoritma lain dengan perfoma sebagai berikut dengan nilai *accuracy* 96.79%, *precision* 97.59% *recall* 95.92% dan *F-Measure* 96.75%. Berdasarkan analisa kinerja performa algoritma *Deep Learning* memiliki nilai performa klasifikasi paling tinggi diantara metode algoritma lainnya.

#### 4. KESIMPULAN

Perangkat lunak yang dikembangkan dalam penelitian ini menunjukkan hasil yang memuaskan. Ujicoba dilakukan pada citra CT scan yang berasal

dari 102 pasien. Metode GLCM digunakan untuk ekstraksi fitur dari citra CT scan sehingga didapatkan fitur *contrast*, *dissimilarity*, *homogeneity*, *corellation*, ASM dan *energy*.



Gambar 11. Grafik hasil performa klasifikasi

Fitur-fitur ini kemudian diproses dengan menggunakan lima algoritma mesin pembelajaran yaitu *Naive Bayes*, *Logistic Regression*, *Support Vector Machine*, *Neural Network* dan *Deep Learning*. Hasil eksperimen menunjukkan bahwa algoritma *Deep Learning* menghasilkan nilai akurasi yang terbaik yaitu 96.79%. Hasil ini diperoleh dengan menggunakan sejumlah parameter yaitu jumlah *hidden layer*, jumlah *neuron*, *epoch*, *bacth size*, *dropout*, *learning rate* dan fungsi aktifasi.

Untuk pengembangan ke depan, perangkat lunak ini akan dilengkapi dengan fitur optimasi parameter untuk mendapatkan kombinasi nilai parameter terbaik untuk algoritma *Deep Learning* sehingga dapat menghasilkan kinerja yang optimal. Selain itu, penulis akan meminta saran dari dokter dan pihak rumah sakit untuk perbaikan dan peningkatan fitur agar aplikasi ini bermanfaat.

#### DAFTAR PUSTAKA

- The World Stroke Organization, 2016. Global stroke fact sheet, [online] Tersedia di: <[https://www.worldstroke.org/assets/downloads/WSO\\_Global\\_Stroke\\_Fact\\_Sheet.pdf](https://www.worldstroke.org/assets/downloads/WSO_Global_Stroke_Fact_Sheet.pdf)> [diakses 9 Juni 2020].
- YUYUN, Y., 2016. Pencitraan Pada Stroke. Malang:Universitas Brawijaya Press.
- MARBUN J. T. DKK. 2018. Classification of stroke disease using convolutional neural network. *2nd International Conference on Computing and Applied Informatics 2017*, 978 (1), pp.1-6. doi:10.1088/1742-6596/978/1/012092
- CHIN C. L. DKK. 2017. An automated early ischemic stroke detection system using CNN deep learning algorithm. *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, pp. 368-372. doi:10.1109/ICAWS.2017.8256481
- BADRIYAH T. DKK. 2019a. Improving stroke

- diagnosis accuracy using hyperparameter optimized deep learning. *International Journal of Advances in Intelligent Informatics*. 5(3), pp. 256–272. doi:<https://doi.org/10.26555/ijain.v5i3.427>
- JEENA R. S. AND KUMAR S. 2016. Stroke prediction using SVM. *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 600-602. doi:10.1109/ICCICCT.2016.7988020
- SINGH, G. A. P., DAN GUPTA P. K., 2018. Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans. *Neural Computing and Applications Springer London*. doi:<https://doi.org/10.1007/s00521-018-3518-x>
- TAHIR, M., BADRIYAH, T. & SYARIF, I. 2018a. Neural Networks Algorithm to Inquire Previous Preeclampsia Factors in Women with Chronic Hypertension During Pregnancy in Childbirth Process. 2018 *International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*. pp. 51-55. doi: 10.1109/KCIC.2018.8628588
- WICAKSONO A. P., BADRIYAH T., & BASUKI A. 2016. Comparison of The Data-Mining Methods in Predicting The Risk Level of Diabetes. *EMITTER International Journal of Engineering Technology*. 4 (1), pp. 164–178. doi: <https://doi.org/10.24003/emitter.v4i1.119>
- VIJAYA G. AND SUHASINI A. 2019. An Adaptive Preprocessing of Lung CT Images with Various Filters for Better Enhancement. *International Research Journal of Engineering and Technology (IRJET)* 7(3) pp. 179–184.
- BADRIYAH T. DKK. 2019b. Segmentation Stroke Objects based on CT Scan Image using Thresholding Method. *2019 First International Conference on Smart Technology & Urban Development (STUD)*, pp.1-6. doi:10.1109/STUD49732.2019.9018825
- MIKOŁAJCZYK A. DAN GROCHOWSKI M. 2018. Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117-122. doi: 10.1109/iiphdw.2018.8388338
- MIRZAPOUR F. AND GHASSEMIAN H. 2015. Fast GLCM and Gabor Filters for Texture Classification of Very High Resolution Remote Sensing Images. *International Journal Of Information And Communication Technology Research Summer*, 7 (3). pp 21-30.
- FAHRUDIN T. M., SYARIF I., & A. R. BARAKBAH. 2017. Data Mining Approach for Breast Cancer Patient Recovery. *EMITTER International Journal of Engineering Technology*. 5(1), pp. 36–71. <https://doi.org/10.24003/emitter.v5i1.190>
- MUIS I A. DAN AFFANDES, M. 2015. Penerapan Metode Support Vector Machine ( SVM ) Menggunakan Kernel Radial Basis Function ( RBF ) Pada Klasifikasi Tweet. *Jurnal Sains, Teknologi dan Industri*, 12(2), pp. 189–197
- VANITHA, C. D. A, DEVARAJ, D. AND VENKATESULU M. 2014. Gene expression data classification using Support Vector Machine and mutual information-based gene selection', *ELSIVER Procedia Computer Science*, 47(C), pp. 13–21.
- SRIVASTAVA, D. K. DAN BHAMBHU, L. 2010. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 12 (1), pp. 1–7.
- LECUN, Y., BENGIO, Y. & HINTON, G. 2015 .Deep learning. *Nature*. 521, 436–444 <https://doi.org/10.1038/nature14539>
- TAHIR, M., BADRIYAH, T. & SYARIF, I. 2018b. Classification Algorithms of Maternal Risk Detection For Preeclampsia With Hypertension During Pregnancy Using Particle Swarm Optimization. *EMITTER International Journal of Engineering Technology*, 6. (2). pp. 236–253. doi: <https://doi.org/10.24003/emitter.v6i2.287>

*Halaman ini sengaja dikosongkan*