

# Ensemble Classifier untuk Klasifikasi Kanker Payudara

Khadijah<sup>1</sup>, Retno Kusumaningrum<sup>2</sup>

Departemen Ilmu Komputer/ Informatika, Fakultas Sains dan Matematika, Universitas Diponegoro<sup>1,2</sup>  
khadijah@live.undip.ac.id<sup>1</sup>, retno@live.undip.ac.id<sup>2</sup>

---

## Article Info

### History :

Dikirim 01 Agustus 2019

Direvisi 15 Agustus 2019

Diterima 22 Agustus 2019

---

### Kata Kunci :

Klasifikasi  
Ensemble Classifier  
Kanker Payudara  
ELM  
SVM  
KNN

---

## Abstrak

Kanker payudara merupakan jenis kanker yang paling banyak diderita oleh kaum wanita di Indonesia. Penyakit tersebut dapat berakibat pada kematian jika terlambat ditangani. Oleh karena itu, deteksi dini kanker payudara merupakan langkah awal untuk menyelamatkan nyawa pasien. Pada penelitian ini telah dilakukan klasifikasi kanker payudara berdasarkan data anthropometric serta data dari hasil tes darah rutin menggunakan *single classifier* (ELM, SVM dan kNN) dan *ensemble classifier* yang menggabungkan ketiga algoritma tersebut dengan penentuan kelas *majority voting*. Pembagian data dilakukan dengan *three way data split*. Hasil eksperimen menunjukkan bahwa saat menggunakan keseluruhan fitur penggunaan *ensemble classifier* lebih baik daripada *single classifier* dalam hal akurasi maupun G-mean. Namun, saat menggunakan 4 fitur terbaik (resistin, glucose, age, dan BMI) penggunaan *ensemble classifier* sedikit lebih baik dalam hal G-mean, akan tetapi lebih rendah dalam hal akurasi. Hal ini disebabkan minimnya *diversity* di antara *classifier* sehingga saat digabungkan tidak mampu memperbaiki hasil.

© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

---

## Koresponden:

Khadijah

Departemen Ilmu Komputer/ Informatika, Fakultas Sains dan Matematika

Universitas Diponegoro,

Jl. Prof. Soedarto, S.H. Tembalang Semarang, Indonesia, 50275

Email : khadijah@live.undip.ac.id

---

## 1. PENDAHULUAN

Kanker adalah penyakit yang diakibatkan oleh sel-sel abnormal yang tumbuh dan menyebar secara tidak terkendali [1]. Berdasarkan data Globocan Tahun 2018 Indonesia menempati peringkat 8 di Asia Tenggara dan 23 di Asia dalam hal jumlah kejadian kanker. Khusus untuk kaum wanita di Indonesia, kanker payudara merupakan jenis kanker yang banyak diderita dengan angka kejadian 42,1 per 100.000 penduduk dan angka kematian rata-rata 17 per 100.000 penduduk [2]. Penderita kanker memiliki kesempatan lebih besar untuk sembuh atau bertahan hidup jika segera mendapatkan *treatment* yang tepat [1]. Oleh karena itu, deteksi dini penyakit kanker sangat perlu dilakukan agar pasien dapat segera menjalani *treatment* seawal mungkin.

Deteksi kanker dapat dilakukan melalui beberapa cara, misalnya berdasarkan struktur morfologis atau kelainan pada ekspresi gen. Akan tetapi, struktur morfologis antara jenis kanker yang berbeda memiliki kemiripan yang cukup tinggi, sehingga deteksi berdasarkan hal tersebut

sulit untuk dilakukan [3]. Sedangkan eksperimen microarray untuk mendapatkan nilai ekspresi gen membutuhkan biaya yang cukup besar [4]. Oleh karena itu, [5] mencoba mencari kandidat biomarker yang dapat digunakan untuk deteksi kanker payudara dengan biaya yang lebih murah berdasarkan data *anthropometric* dan hasil tes darah rutin. Penelitian tersebut lebih terfokus pada pemilihan fitur atau biomarker yang representatif untuk diagnosis kanker payudara. Penelitian tersebut juga mencoba beberapa algoritma untuk membangun *classifier* berdasarkan fitur yang dipilih, yaitu *Support Vector Machine* (SVM), Logistic Regression dan Random Forest. Hasil eksperimen menunjukkan bahwa SVM memberikan performa terbaik pada berbagai kombinasi fitur yang digunakan.

Di samping algoritma SVM, algoritma klasifikasi lain yang juga memberikan hasil yang baik dalam permasalahan klasifikasi kanker adalah *Extreme Learning Machine* (ELM) [6][7] dan *k-Nearest Neighbor* (kNN)[8][9]. ELM adalah algoritma pembelajaran untuk *single hidden layer feedforward network* yang lebih baik dari algoritma *gradient descent learning* karena memerlukan waktu pelatihan yang lebih singkat, jumlah parameter pelatihan yang lebih sedikit dan kemampuan generalisasi yang lebih baik [10][11]. kNN adalah *lazy learner* yang mengklasifikasikan data berdasarkan label kelas mayoritas dari sejumlah tetangga terdekatnya [12]. Meskipun konsep algoritma kNN sederhana, akan tetapi algoritma ini mampu mencapai akurasi 98,70% pada klasifikasi Wisconsin Breast Cancer Database [8] dan 96,33% pada klasifikasi *gastric cancer* [9].

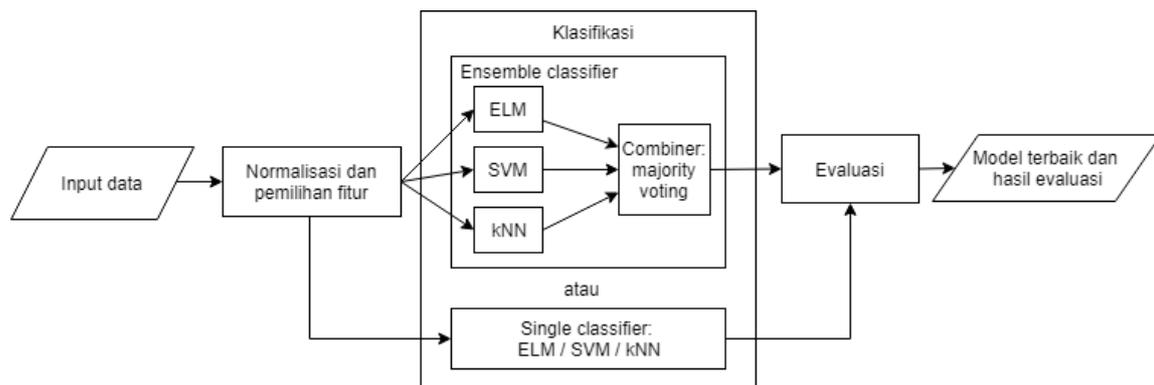
Beberapa penelitian mencoba meningkatkan hasil klasifikasi dengan menerapkan *ensemble method*. *Ensemble method* melatih sejumlah *classifier* atau *base learner* dan kemudian mengkombinasikan hasilnya. *Base learner* yang digunakan dapat berupa algoritma klasifikasi apapun, seperti *decision tree*, Naïve Bayes classifier, jaringan syaraf tiruan, SVM dan sebagainya [13]. Hasil eksperimen menunjukkan bahwa penggunaan *ensemble classifier* memberikan performa yang lebih baik daripada *single classifier* [14][15]. Hal ini disebabkan, sebuah *classifier* atau *base learner* umumnya memiliki letak kesalahan (*error*) pada *instance data* tertentu. *Classifier* yang berbeda juga dapat memiliki letak kesalahan pada *instance data* yang berbeda. Hal itu disebut dengan istilah *diversity* di antara *classifier*. Oleh karena itu, *ensemble method* menggabungkan sejumlah *classifier* yang berbeda dengan tujuan agar letak kesalahan pada sebuah *classifier* dapat diperbaiki oleh *classifier* lainnya, sehingga pada akhirnya didapat *error* yang lebih kecil daripada saat menggunakan sebuah *classifier* saja. Akan tetapi supaya tujuan tersebut dapat dicapai, harus terdapat *diversity* pada sejumlah *classifier* yang digunakan [16]. Salah satu cara untuk mencapai *diversity* adalah dengan menggunakan algoritma klasifikasi yang berbeda [14]. Oleh karena itu, pada penelitian ini digunakan *ensemble classifier* yang menggabungkan ketiga algoritma yang berasal dari konsep yang berbeda untuk mencapai *diversity*, yaitu SVM (*statistical classifier*), ELM (*artificial neural network*) dan kNN (*lazy learner*). Selanjutnya, dapat dibandingkan hasil yang dicapai oleh *ensemble classifier* dan *single classifier*.

## 2. METODE PENELITIAN

Gambaran umum proses pada penelitian ini dapat dilihat pada Gambar 1. Terdapat tiga proses utama, yaitu normalisasi dan pemilihan fitur, klasifikasi dan evaluasi. Detail untuk masing-masing proses tersebut dijelaskan pada sub-bab berikutnya. Sebelumnya, dijelaskan terlebih dahulu mengenai *dataset* dan pembagian data yang digunakan.

### 2.1 Dataset

*Dataset* yang digunakan pada penelitian ini adalah *dataset* Breast Cancer Coimbra yang diunduh dari UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra>). *Dataset* tersebut dikumpulkan dari Gynaecology Department pada University Hospital Centre of Coimbra (CHUC) selama tahun 2009-2013 yang terdiri atas 52 pasien normal (negatif) dan 64 pasien kanker payudara (positif). Setiap *record* data terdiri atas 10 atribut, terbagi atas 9 atribut fitur bertipe numerik dan satu atribut keputusan bertipe kategori yang menyatakan positif/ negatif kanker. Atribut fitur terdiri atas data *anthropometric* (usia dan BMI) serta data yang didapat dari hasil tes darah rutin (glucose, insulin, HOMA, leptin, adiponectin, resistin, MCP-1) seperti pada Tabel 1 [5].



Gambar 1. Gambaran Umum Proses Penelitian

Tabel 1. Deskripsi Atribut pada Dataset [5]

Fitur	Keterangan	Nilai Minimal	Nilai Maksimal
Usia	Usia pasien	24	89
BMI	<i>Body Mass Index</i>	18,37	38,579
Glucose	Kadar glukosa dalam serum	60	201
Insulin	Kadar insulin dalam serum	2,432	58,46
HOMA	Nilai indeks Homeostasis Model Assessment	0,467	25,050
Leptin	Kadar leptin dalam serum	4,311	90,280
Adiponectin	Kadar adiponectin dalam serum	1,656	38,040
Resistin	Kadar resistin dalam serum	3,21	82,1
MCP-1	Kadar Monocyte Chemoattractant Protein 1 dalam serum	45,843	1698,440
Kategori	Kategori pasien	1= negatif	2 = positif

## 2.2. Pembagian Data

Klasifikasi terdiri atas dua proses utama, yaitu pelatihan dan pengujian. Pelatihan bertujuan untuk membangun model atau *classifier*, selanjutnya pengujian bertujuan untuk mengevaluasi *classifier* tersebut. Agar hasil pengujian minimal dari bias, maka *subset* data yang dipakai untuk pelatihan dan pengujian harus berbeda. Oleh karena itu, perlu dilakukan pembagian dataset untuk memisahkan *subset* data untuk proses pelatihan dan pengujian.

Pembagian data dilakukan dengan *three-way data split*, yaitu *dataset* dibagi menjadi data latih, data validasi dan data uji. Pertama, dataset dibagi menjadi data latih (93 sampel) dan data uji (23 sampel) menggunakan *random subsampling* dengan perbandingan yang seimbang antar kelas positif dan negatif dan diulang sebanyak sepuluh kali. Selanjutnya, data latih dibagi lagi menjadi data latih yang sesungguhnya dan data validasi menggunakan *stratified k-fold cross validation* ( $k = 5$ ). Data validasi digunakan untuk mengevaluasi hasil pelatihan dalam rangka memilih kombinasi parameter terbaik untuk pelatihan, sedangkan data uji digunakan untuk mengevaluasi *classifier* akhir yang telah dilatih menggunakan kombinasi parameter terbaik. Pembagian data dengan cara seperti ini dilakukan agar data latih dan data uji benar-benar independen, yang bertujuan untuk meminimalkan bias pada hasil akhir yang diperoleh.

## 2.3. Normalisasi dan Pemilihan Fitur

Normalisasi bertujuan untuk menyeragamkan *range* data pada setiap jenis fitur, sehingga mencegah *overweighting* pada fitur yang memiliki *range* data besar terhadap atribut yang memiliki *range* data kecil. Normalisasi yang digunakan pada penelitian ini adalah *min-max normalization* dengan *range*  $[-1,1]$ , artinya *range* nilai setiap fitur diskalakan sehingga nilai minimumnya menjadi -1 dan nilai maksimumnya menjadi 1 [12]. Sedangkan untuk pemilihan fitur akan dicoba dengan

dua cara, yaitu menggunakan keseluruhan fitur dan menggunakan empat fitur terbaik pada penelitian sebelumnya (resistin, glucose, age dan BMI) [5].

### 2.3. Klasifikasi

Pada penelitian ini klasifikasi dilakukan dengan *ensemble classifier* dan *single classifier*. *Single classifier* yang digunakan meliputi ELM, SVM dan kNN, sedangkan *ensemble classifier* menggabungkan ketiga algoritma tersebut.

#### 2.3.1. Ensemble Method

*Ensemble method* melatih sejumlah *base learner* untuk menyelesaikan suatu permasalahan dan kemudian mengkombinasikan hasilnya. *Ensemble method* dapat dibangun dengan menggunakan satu jenis algoritma klasifikasi namun dilatih dengan *subset* data yang berbeda (*homogeneous ensembles*), atau menggunakan algoritma klasifikasi yang berbeda (*heterogenous ensembles*) [13]. Karena jumlah *dataset* yang terbatas, penelitian ini menggunakan *dataset* yang sama untuk melatih beberapa *classifier* yang berbeda, yaitu ELM, SVM dan kNN. Selanjutnya, *output* dari masing-masing *classifier* atau *learner* dikombinasikan. Aturan kombinasi yang digunakan pada penelitian ini adalah *majority voting* karena aturan tersebut sederhana dan banyak digunakan. Pada *majority voting*, setiap *learner* akan memberikan output berupa sebuah label kelas untuk sebuah masukan. Selanjutnya, label kelas final adalah label kelas yang memiliki jumlah *vote* lebih dari setengah jumlah *learner* yang digunakan dalam *ensemble method*. Jika tidak ada label kelas yang memenuhi kondisi tersebut, maka *ensemble method* akan mengeluarkan opsi *rejection* atau dengan kata lain tidak ada prediksi label yang dihasilkan [13].

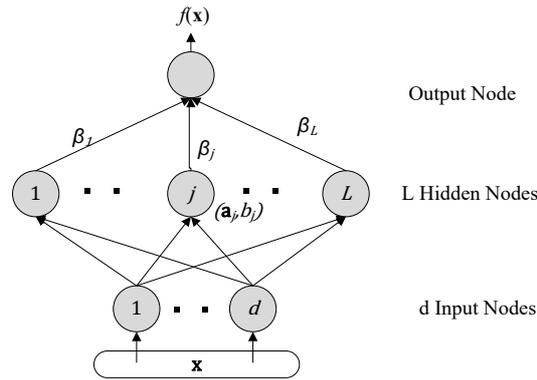
#### 2.3.2 Extreme Learning Machine

*Extreme learning machine* (ELM) pertama kali dikembangkan pada tahun 2004 [10] dan disempurnakan menjadi ELM untuk *generalized single hidden layer feedforward neural network* (SLFN) [11]. Contoh arsitektur jaringan syaraf tiruan dengan satu *node output* yang dapat dilatih dengan ELM ditunjukkan pada Gambar 2. Nilai  $\mathbf{a}_j$  dan  $b_j$  adalah parameter pada *hidden node* ke- $j$ , sedangkan  $\boldsymbol{\beta} = [\beta_1, \dots, \beta_L]^T$  adalah vektor bobot dari *hidden node* ke sebuah *output node*. Fungsi *output* untuk sebuah *output node* pada ELM dapat dilihat pada persamaan (1) dengan  $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbf{R}^d$  adalah vektor *input* dan  $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_L(\mathbf{x})]^T$  adalah vektor *output* dari *hidden layer* untuk sebuah masukan  $\mathbf{x}$ . Pada penelitian ini  $\mathbf{x}$  adalah *feature set* yang telah dinormalisasi. Nilai keluaran pada *hidden node* ke- $j$ , yaitu  $h_j(\mathbf{x})$  dihitung menggunakan fungsi aktivasi  $G(\mathbf{a}_j, b_j, \mathbf{x})$ . Fungsi aktivasi yang digunakan dapat berupa fungsi sigmoid atau hard-limit pada *feedforward neural network*, serta fungsi Gaussian atau multiquadric pada *RBF network*. Selanjutnya, label kelas menggunakan fungsi  $\text{sign}(f(\mathbf{x}))$ . Jika digunakan *multioutput node*, maka jumlah *node output* diset sesuai jumlah kelas, dan label dari suatu masukan ditentukan dari indeks *node output* yang memberikan nilai tertinggi [11].

$$f(\mathbf{x}) = \sum_{j=1}^L \beta_j h_j(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \quad (1)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \dots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_N) & \dots & h_L(\mathbf{x}_N) \end{bmatrix}_{N \times L} \quad (2)$$

$$\boldsymbol{\beta} = \mathbf{H}^T \left( \frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \quad (3)$$

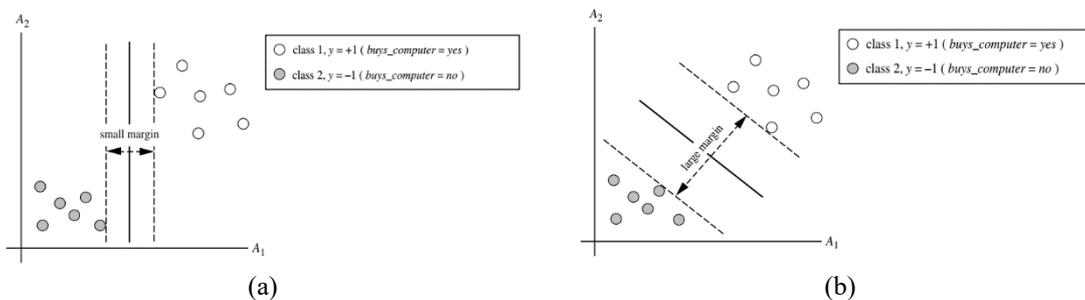


Gambar 2. Contoh Arsitektur Jaringan Syaraf Tiruan untuk ELM [17]

Algoritma pelatihan ELM menggunakan  $N$  pasangan *input-output*  $(x_i, t_i)_{i=1}^N$  dengan  $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T \in R^d$  adalah vektor *input data* ke- $i$  dan  $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$  adalah nilai target data ke- $i$ . Parameter pada *hidden node*  $(a_j, b_j)_{j=1}^L$  di-generate secara random, sedangkan bobot output  $\beta$  dihitung dengan persamaan (2) dengan  $H$  adalah *hidden layer output matrix* seperti persamaan (3) dan  $T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$  adalah matriks target untuk data latih ke  $1..N$ ,  $I$  adalah matriks identitas dan  $C$  adalah parameter regularisasi [11].

**2.3.3. Support Vector Machine**

*Support Vector Machine* (SVM) adalah algoritma untuk klasifikasi biner yang memisahkan data pada dua kelas berbeda dengan mencari *maximum marginal hyperplane* (MMH), yaitu optimal *hyperplane* dengan margin terbesar. Sebagai contoh Gambar 3 menunjukkan pemetaan data  $(x_i, y_i)$  dari dua kelas yang berbeda yang terdiri atas dua atribut, yaitu  $A_1$  dan  $A_2$ . Gambar 3(b) lebih baik dari Gambar 3(a) sebab garis pemisahannya memiliki margin yang lebih besar.



Gambar 3. Contoh Pemetaan Data 2 Kelas (a) *small margin* (b) *large margin* [12]

*Hyperplane* yang mendefinisikan kedua sisi margin dapat dituliskan dengan  $H_1$  dan  $H_2$  seperti persamaan (4) dan (5) dengan  $W = \{w_0, w_1, w_2\}$  adalah vektor bobot. Data latih yang berada di  $H_1$  atau  $H_2$  disebut *hyperplane* sebagai *support vector*. Algoritma pelatihan SVM bertujuan untuk menemukan MMH dan *support vector*.

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \text{ for } y_i = +1 \tag{4}$$

$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \text{ for } y_i = -1 \tag{5}$$

Selanjutnya, sebuah data baru  $x_{test}$  dapat diklasifikasikan menggunakan persamaan (6) dengan  $x_i$  adalah *support vector* ke- $i$ ,  $y_i$  adalah kelas data untuk *support vector*  $x_i$  serta  $\alpha_i$  dan  $b_0$

adalah parameter numerik yang didapat dari algoritma pelatihan SVM. Data  $\mathbf{x}_{test}$  diklasifikasikan ke kelas +1 jika  $d(\mathbf{x}_{test})$  bernilai positif atau ke kelas -1 jika  $d(\mathbf{x}_{test})$  bernilai negatif.

$$d(\mathbf{x}_{test}) = \sum_{i=1}^l y_i \alpha_i \mathbf{x}_i \mathbf{x}_{test} + b_0 \quad (6)$$

Untuk kasus *non-linearly separable data*, maka data akan dipetakan terlebih dahulu ke dimensi yang lebih tinggi menggunakan *non-linear mapping*. *Nonlinear mapping*  $\phi(\mathbf{X}_i)$  dapat diterapkan menggunakan fungsi kernel seperti persamaan (7). Beberapa pilihan fungsi kernel yang dapat digunakan antara lain linear, Gaussian atau *radial basis function*.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (7)$$

#### 2.3.4. *k*-Nearest Neighbor

Metode *k-nearest neighbor* (*k*-NN) adalah metode klasifikasi yang termasuk *lazy learner* sebab metode ini tidak memerlukan proses pelatihan untuk membangun sebuah model *classifier*. Pada metode ini, setiap data atau *tuple*  $\mathbf{x}$  merepresentasikan sebuah titik pada ruang berdimensi  $d$ . Selanjutnya, *k*-NN melakukan prediksi kelas dari sebuah *test tuple* (titik) dengan mencari  $k$  *tuple* (titik) terdekat dengan *test tuple* tersebut pada ruang berdimensi  $d$ . Untuk menentukan titik-titik terdekat dibutuhkan ukuran kedekatan dan ukuran kedekatan yang banyak digunakan adalah jarak Euclidean. [12]

Langkah-langkah dalam algoritma *k*-NN untuk memprediksi label kelas dari sebuah *test tuple*  $\mathbf{x}_{test}$  adalah sebagai berikut [12]:

1. Menghitung jarak antara *test tuple*  $\mathbf{x}_{test}$  dengan semua *tuple* yang ada di data latih  $(\mathbf{x}_i, t_i)_{i=1}^N$  dengan  $\mathbf{x}_i$  adalah vektor input dan  $t_i$  adalah nilai target data ke- $i$
2. Menemukan  $k$  tetangga terdekat, yaitu  $k$  *tuple* di data latih yang memiliki jarak terdekat dengan *test tuple*  $\mathbf{x}_{test}$ .
3. Menentukan label kelas dari *test tuple*  $\mathbf{x}_{test}$  dengan cara:
  - a. Jika  $k = 1$ , maka *test tuple*  $\mathbf{x}_{test}$  diklasifikasikan ke kelas data dari *tuple* tetangga.
  - b. Jika  $k > 1$ , maka *test tuple*  $\mathbf{x}_{test}$  diklasifikasikan berdasarkan kelas terbanyak dari *tuple* tetangga (*majority voting*).

#### 2.4. Metrik Evaluasi

Permasalahan yang diselesaikan dalam penelitian ini adalah klasifikasi biner. Dengan demikian, model atau *classifier* yang dihasilkan dievaluasi menggunakan metrik berupa akurasi, *sensitivity* dan *specificity*. Akurasi menyatakan perbandingan antara jumlah data yang diklasifikasikan dengan benar oleh *classifier* dengan jumlah data keseluruhan. *Sensitivity* menyatakan perbandingan antara jumlah data di kelas positif yang diklasifikasikan dengan benar oleh *classifier* ke kelas positif (*true positive*) dengan jumlah data yang seharusnya di kelas positif. *Specificity* menyatakan perbandingan antara jumlah data di kelas negatif yang diklasifikasikan oleh *classifier* ke kelas negatif (*true negative*) dengan jumlah data yang seharusnya di kelas negatif [12]. Di samping itu, untuk mengetahui kemampuan rata-rata pengenalan *classifier* di kelas positif maupun negatif, dihitung pula nilai G-mean. G-mean merupakan hasil kali nilai *sensitivity* dan *specificity* yang nilainya tidak terpengaruh saat perbandingan jumlah data di kelas positif dan negatif tidak sama [18].

### 3. HASIL DAN PEMBAHASAN

Pada eksperimen ini dilakukan dua skenario. Skenario pertama menggunakan keseluruhan fitur pada dataset sebagai masukan untuk klasifikasi, sedangkan skenario kedua hanya menggunakan 4 fitur terbaik yang telah dipilih pada penelitian sebelumnya [5]. Pada masing-masing skenario dilakukan klasifikasi menggunakan *classifier* tunggal untuk mendapatkan kombinasi parameter terbaik pada masing-masing *classifier*. Selanjutnya, masing-masing *classifier* tersebut digabungkan membentuk *ensemble classifier*.

Pada *classifier* ELM terdapat 2 parameter yang diuji coba untuk mendapatkan hasil yang optimal, yaitu fungsi aktivasi pada *hidden node* dan nilai parameter regularisasi. Fungsi aktivasi *hidden node* yang dicoba adalah sigmoid, hard-limit, Gaussian dan multiquadric. Nilai parameter regularisasi yang dicoba adalah dalam rentang  $\{2^{-25}, 2^{-24}, \dots, 2^{24}, 2^{25}\}$  dan jumlah *hidden node* ( $L$ ) yang digunakan adalah 1000 sesuai nilai yang digunakan pada penelitian [11]. Pada *classifier* SVM fungsi kernel yang digunakan diuji coba dengan beberapa kemungkinan, yaitu fungsi Linear, Gaussian, dan RBF. Pada *classifier* kNN ukuran kedekatan yang digunakan adalah jarak Euclidean dan jumlah tetangga ( $k$ ) yang digunakan diuji coba dengan beberapa nilai, yaitu 5, 7, 11 dan 11.

### 3.1. Skenario 1

Hasil eksperimen skenario pertama yang menggunakan keseluruhan fitur pada dataset, dapat dilihat pada Tabel 2, 3 dan 4 secara berurutan untuk *classifier* ELM, SVM, dan kNN. Saat menggunakan *classifier* ELM, hanya ditampilkan nilai  $C$  yang memberikan hasil terbaik. Eksperimen tersebut dilakukan dengan *5-fold cross validation* hanya pada 93 sampel data dan bertujuan untuk mendapatkan kombinasi parameter terbaik pada masing-masing *classifier*. Pada Tabel 2 terlihat bahwa hasil terbaik didapat saat menggunakan fungsi aktivasi sigmoid. Akurasi terbaik pada *classifier* SVM didapat saat menggunakan fungsi kernel Gaussian atau RBF. Hal ini dapat disebabkan karena data yang bersifat *non-linearly separable*, sehingga saat menggunakan fungsi kernel linear akurasi menjadi lebih rendah. Saat menggunakan *classifier* kNN, terlihat bahwa semakin banyak jumlah tetangga yang digunakan akurasi maupun G-mean semakin tinggi. Hal ini dapat disebabkan karena data dari kelas yang berbeda menempati ruang yang cukup berdekatan, akibatnya semakin banyak tetangga yang diperlukan sebagai dasar penentuan label kelas untuk suatu data agar didapat hasil prediksi lebih tepat.

Tabel 2. Rata-Rata Hasil 5-Cross Validation pada ELM Menggunakan 9 Fitur

Fungsi Aktivasi	C	Akurasi	Sensitivity	Specificity	G-mean
<b>Sigmoid</b>	$2^0$	81.50%	81.48%	81.07%	81.28%
Hard limit	$2^{17}$	77.82%	77.78%	77.38%	77.58%
Gaussian	$2^1$	79.39%	80.74%	77.14%	78.92%
Multiquadric	$2^0$	80.12%	82.22%	77.14%	79.64%

Tabel 3. Rata-Rata Hasil 5-Cross Validation pada SVM Menggunakan 9 Fitur

Fungsi Kernel	Akurasi	Sensitivity	Specificity	G-mean
<b>Gaussian</b>	74.12%	88.89%	55.36%	70.15%
<b>RBF</b>	74.12%	88.89%	55.36%	70.15%
Linear	72.75%	74.07%	70.83%	72.44%

Tabel 4. Rata-Rata Hasil 5-Cross Validation pada kNN Menggunakan 9 Fitur

Jumlah Tetangga ( $k$ )	Akurasi	Sensitivity	Specificity	G-mean
5	70.44%	68.89%	71.79%	70.32%
7	72.94%	75.56%	69.29%	72.35%
9	76.62%	75.56%	77.50%	76.52%
<b>11</b>	79.12%	77.78%	80.36%	79.06%

Setelah mendapatkan kombinasi parameter terbaik, *classifier* dibangun menggunakan 93 data latih dan kombinasi parameter tersebut, kemudian diuji menggunakan 23 data uji. Rata-rata hasil pengujian dari 10 eksperimen pada masing-masing *classifier* dan *ensemble classifier* ditunjukkan pada Tabel 5. Terlihat bahwa akurasi yang didapat oleh *single classifier* ELM, SVM dan kNN hampir sama atau tidak jauh berbeda, akan tetapi jika dilihat dari G-mean, SVM memberikan hasil yang terendah (terlihat dari nilai *sensitivity* dan *specificity* yang tidak seimbang). Namun, ketika masing-masing *classifier* tersebut digabung menjadi *ensemble classifier* akurasi dan

G-mean yang didapat lebih tinggi daripada hasil dari masing-masing *classifier*. Hal ini menunjukkan bahwa penggunaan *ensemble classifier* mampu meningkatkan performa klasifikasi dalam hal akurasi maupun G-mean.

Tabel 5. Rata-Rata Hasil Pengujian Menggunakan 9 Fitur

<i>Classifier</i>	Akurasi	<i>Sensitivity</i>	<i>Specificity</i>	G-mean
ELM	70.43%	73.08%	67.00%	69.29%
SVM	69.13%	90.00%	42.00%	60.32%
kNN	70.87%	72.31%	69.00%	69.82%
<b>Ensemble</b>	<b>72.17%</b>	<b>78.46%</b>	<b>64.00%</b>	<b>70.35%</b>

### 3.2. Skenario 2

Pada skenario kedua masukan ke *classifier* hanya menggunakan 4 fitur terbaik yang dipilih pada penelitian sebelumnya [5], yaitu Glucose, Resistin, BMI dan Age. Hasil eksperimen ditunjukkan pada Tabel 6, 7, dan 8 secara berurutan untuk *classifier* ELM, SVM, dan kNN. Hasil yang didapat mengalami peningkatan dibanding saat menggunakan keseluruhan fitur, baik pada algoritma ELM, SVM maupun kNN. Pada hasil tersebut juga terlihat bahwa kombinasi parameter terbaik untuk *classifier* ELM dan kNN berbeda dengan saat menggunakan keseluruhan fitur, kecuali pada SVM. Hal ini menunjukkan bahwa saat menggunakan 4 fitur pemetaan data menjadi berbeda dengan saat menggunakan keseluruhan fitur, akan tetapi karakteristik *non-linearly separable* tetap ada, sehingga saat menggunakan SVM, fungsi kernel Gaussian atau RBF memberikan hasil yang lebih baik daripada fungsi kernel linear. Pada *classifier* ELM hasil terbaik didapat saat menggunakan fungsi aktivasi multiquadric. Pada *classifier* kNN terlihat bahwa tidak terdapat pola yang teratur antara penambahan jumlah tetangga yang digunakan dengan peningkatan atau penurunan hasil yang didapat. Hasil terbaik didapat saat menggunakan  $k = 7$ .

Tabel 6. Rata-Rata Hasil 5-Cross Validation pada ELM Menggunakan 4 Fitur

Fungsi Aktivasi	C	Akurasi	<i>Sensitivity</i>	<i>Specificity</i>	G-mean
Sigmoid	$2^7$	85.17%	90.37%	78.57%	84.26%
Hard limit	$2^{-1}$	83.55%	83.70%	83.10%	83.40%
Gaussian	$2^{-3}$	87.67%	91.11%	82.98%	86.95%
<b>Multiquadric</b>	$2^{-3}$	<b>88.06%</b>	<b>88.89%</b>	<b>86.67%</b>	<b>87.77%</b>

Tabel 7. Rata-Rata Hasil 5-Cross Validation pada SVM Menggunakan 4 Fitur

Fungsi Kernel	Akurasi	<i>Sensitivity</i>	<i>Specificity</i>	G-mean
<b>Gaussian</b>	82.72%	91.11%	72.14%	81.07%
<b>RBF</b>	82.72%	91.11%	72.14%	81.07%
Linear	75.37%	68.89%	82.86%	75.55%

Tabel 8. Rata-Rata Hasil 5-Cross Validation pada kNN Menggunakan 4 Fitur

Jumlah Tetangga ( $k$ )	Akurasi	<i>Sensitivity</i>	<i>Specificity</i>	G-mean
5	80.22%	84.44%	74.29%	79.20%
<b>7</b>	<b>81.40%</b>	<b>88.89%</b>	<b>71.43%</b>	<b>79.68%</b>
9	77.65%	80.00%	74.29%	77.09%
11	78.97%	84.44%	71.79%	77.86%

Selanjutnya, *classifier* dibangun kembali menggunakan 93 data latih dan kombinasi parameter terbaik yang telah didapatkan, lalu diuji menggunakan 23 data uji. Rata-rata hasil pengujian dari 10 eksperimen pada masing-masing *classifier* dan *ensemble classifier* ditunjukkan pada Tabel 9. Terlihat bahwa secara keseluruhan hasil klasifikasi menggunakan 4 fitur saja lebih

baik dibandingkan saat menggunakan keseluruhan fitur. Hal ini menunjukkan bahwa 4 fitur tersebut mampu mempermudah *classifier* untuk menemukan bidang pemisah data dari kelas yang berbeda, sehingga *classifier* dapat memberikan hasil yang lebih baik. Jika dilihat dari masing-masing *classifier* nampak bahwa kNN memberikan hasil yang paling rendah dari *classifier* lainnya, sedangkan ELM dan SVM memberikan hasil yang hampir sama dalam hal akurasi dan G-mean. Akan tetapi, performa *ensemble classifier* tidak lebih baik dari *classifier* tunggal ELM maupun SVM dalam hal akurasi, namun nilai G-mean yang didapat sedikit lebih baik dibanding ELM dan SVM.

Perbandingan hasil prediksi saat menggunakan *single classifier* maupun *ensemble classifier* pada *subsample* data uji ke-9 dapat dilihat pada Tabel 10. Terlihat bahwa kNN paling banyak memberikan prediksi yang salah, sedangkan pada SVM dan ELM sebagian besar letak kesalahan prediksi berada pada *instance data* yang sama (data nomor 3, 4, 7 dan 17), sehingga *diversity* antara kedua *classifier* tersebut rendah, akibatnya saat digabungkan tidak mampu menurunkan jumlah kesalahan prediksi.

Tabel 9. Rata-Rata Hasil Pengujian Menggunakan 4 Fitur

<i>Classifier</i>	Akurasi	<i>Sensitivity</i>	<i>Specificity</i>	G-mean
ELM	83.04%	82.31%	84.00%	80.18%
SVM	82.61%	90.00%	73.00%	80.74%
kNN	76.52%	82.31%	69.00%	74.54%
<b>Ensemble</b>	82.17%	87.69%	75.00%	80.85%

Tabel 10. Perbandingan Hasil Prediksi pada *Subsample* Data Ke-9

No	ELM	SVM	kNN	Ensemble	Target	No	ELM	SVM	kNN	Ensemble	Target
1	0	0	0	0	0	13	1	1	1	1	1
2	0	0	1	0	0	14	1	1	1	1	1
3	1	1	1	1	0	15	1	1	1	1	1
4	1	1	1	1	0	16	1	1	1	1	1
5	0	0	0	0	0	17	0	0	0	0	1
6	0	0	0	0	0	18	1	1	1	1	1
7	1	1	1	1	0	19	1	1	1	1	1
8	0	0	0	0	0	20	1	1	1	1	1
9	0	0	0	0	0	21	1	1	1	1	1
10	1	0	0	0	0	22	1	1	1	1	1
11	1	1	1	1	1	23	1	1	1	1	1
12	1	1	1	1	1						

#### 4. KESIMPULAN

Pada penelitian ini telah dilakukan klasifikasi kanker payudara berdasarkan data *anthropometric* (usia dan BMI/ Body Mass Index) serta data dari hasil tes darah rutin (glucose, insulin, HOMA, leptin, adiponectin, resistin, MCP-1). Penelitian ini membandingkan klasifikasi dengan *single classifier* (ELM, SVM dan kNN) dan klasifikasi dengan *ensemble classifier* yang menggabungkan ketiga algoritma tersebut dengan penentuan kelas *majority voting*. Pembagian data dilakukan dengan *three way data split*. Hasil yang didapat menunjukkan bahwa:

1. Pada saat menggunakan keseluruhan fitur penggunaan *ensemble classifier* lebih baik daripada *single classifier* dalam hal akurasi maupun G-mean. Pada skenario ini, *single classifier* yang digunakan, yaitu ELM, SVM dan kNN memberikan hasil yang hampir sama (berdekatan nilainya).
2. Pada saat menggunakan fitur resistin, glucose, age, dan BMI, kNN memberikan performa yang jauh lebih rendah dibanding *single classifier* lainnya. Sedangkan penggunaan *ensemble classifier* sedikit lebih baik dalam hal G-mean, namun sedikit lebih rendah dalam hal akurasi. Hal ini disebabkan *classifier* yang digunakan sebagian besar memiliki letak kesalahan pada *instance data* yang sama (*diversity* antar *classifier* rendah), sehingga saat digabungkan tidak mampu menurunkan tingkat kesalahan.

Penelitian selanjutnya dapat mencoba model *ensemble method* yang lain, seperti *bagging* dan *boosting*, serta algoritma seleksi fitur lainnya untuk meningkatkan hasil akurasi.

## DAFTAR PUSTAKA

- [1] American Cancer Society, *Cancer Facts & Figures 2019*. Atlanta: American Cancer Society, 2019.
- [2] Kementerian Kesehatan, "Hari Kanker Sedunia 2019," 2019. [Online]. Available: <http://sehatnegeriku.kemkes.go.id/baca/fokus-utama/20190131/2329273/hari-kanker-sedunia-2019>. [Accessed: 21-Jul-2019].
- [3] T. R. Golub *et al.*, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," vol. 286, no. October, pp. 531–537, 1999.
- [4] M. K. Kerr and G. A. Churchill, "Experimental Design for Gene Expression Microarrays," *Biostatistics*, vol. 2, no. 2, pp. 183–201, 2001.
- [5] M. Patricio *et al.*, "Using Resistin, Glucose, Age and BMI to Predict the Presence of Breast Cancer," *BMC Cancer*, vol. 18(1), no. 29, pp. 1–8, 2018.
- [6] R. Zhang, G. Huang, N. Sundararajan, and P. Saratchandran, "Multicategory Classification Using an Extreme Learning Machine for Microarray Gene Expression Cancer Diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 4, no. 3, pp. 485–495, 2007.
- [7] Khadijah, Rismiyati, and A. J. Mantau, "Multiclass Classification of Cancer Based on Microarray Data Using Extreme Learning Machine," in *Proceeding of The 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 159–164.
- [8] S. A. Medjahed, "Breast Cancer Diagnosis by Using k-Nearest Neighbor with Different Distances and Classification Rules," *Int. J. Comput. Appl.*, vol. 62, no. 1, pp. 1–5, 2013.
- [9] C. Li *et al.*, "Using the K-Nearest Neighbor Algorithm for the Classification of Lymph Node Metastasis in Gastric Cancer," *Comput. Math. Method Med.*, vol. 2012, pp. 1–11, 2012.
- [10] G. Huang, Q. Zhu, and C. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks," in *Proceeding of International Joint Conference on Neural Networks (IJCNN)*, 2004, no. July, pp. 985–990.
- [11] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Trans. Syst. Man, Cybern. - Part B Cybern.*, vol. 42, no. 2, pp. 513–528, 2012.
- [12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques Second Edition*. San Farnsisco: Elsevier Inc., 2006.
- [13] Z.-H. Zhou, *Ensemble Methods Foundations and Algorithms*, Machine Le. Boca raton: CRC Press, 2012.
- [14] M. Graña and E. Corchado, "A Survey of Multiple Classifier Systems as Hybrid Systems," *Inf. Fusion*, vol. 16, pp. 3–17, 2014.
- [15] N. Joshi and S. Srivastava, "Improving Classification Accuracy Using Ensemble Learning Technique (Using Different Decision Trees)," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 5, pp. 727–732, 2014.
- [16] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity Creation Methods: A Survey and Categorisation," *Hournal Inf. Fusion*, vol. 6, no. 1, pp. 1–28, 2005.
- [17] G. Huang, "Extreme Learning Machine - Learning Without Iterative Tuning." Tutorial in IJCNN2012/WCCI2012, Brisbane, 2012.
- [18] I. K. Timotius and S. G. Miaou, "Arithmetic Means of Accuracies: A Classifier Performance Measurement for Imbalanced Data Set," in *International Conference on Audio, Language and Image Processing (ICALIP)*, 2010, pp. 1244–1251.

**BIOGRAFI PENULIS**

 A portrait of a woman wearing a white hijab and a dark blue blazer, set against a light blue background.	<p><b>Khadijah</b> obtained Bachelor Degree in Informatics Engineering from Universitas Diponegoro in 2011 and obtained Master Degree in Computer Science from Universitas Gadjah Mada in 2014. She has been a Lecturer with the Department of Computer Science/ Informatics, Universitas Diponegoro since 2014. Her current research interests include artificial intelligence and machine learning.</p>
 A portrait of a woman wearing a pink hijab and a purple patterned top, smiling, set against a background of red curtains.	<p><b>Retno Kusumaningrum</b> obtained Bachelor Degree in Mathematics from Universitas Diponegoro in 2003, obtained Master Degree in Computer Science from Universitas Indonesia in 2010, and obtained Doctoral of Computer Science from Universitas Indonesia in 2014. She has been a Lecturer with the Department of Informatics, Universitas Diponegoro, since 2005. Her current research interests include computational linguistics, natural language processing and machine learning.</p>