

## KOMPARASI METODE KLASIFIKASI UNTUK DETEKSI EKSPRESI WAJAH DENGAN FITUR FACIAL LANDMARK

Fitra A. Bachtiar<sup>\*1</sup>, Muhammad Wafi<sup>2</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Universitas Brawijaya Malang  
Email: <sup>1</sup>fitra.bachtiar@ub.ac.id, <sup>2</sup>muhammadwafi@student.ub.ac.id

\*Penulis Korespondensi

(Naskah masuk: 01 Desember 2020, diterima untuk diterbitkan: 18 Oktober 2021)

### Abstrak

*Human machine interaction*, khususnya pada *facial behavior* mulai banyak diperhatikan untuk dapat digunakan sebagai salah satu cara untuk personalisasi pengguna. Kombinasi ekstraksi fitur dengan metode klasifikasi dapat digunakan agar sebuah mesin dapat mengenali ekspresi wajah. Akan tetapi belum diketahui basis metode klasifikasi apa yang tepat untuk digunakan. Penelitian ini membandingkan tiga metode klasifikasi untuk melakukan klasifikasi ekspresi wajah. Dataset ekspresi wajah yang digunakan pada penelitian ini adalah JAFFE dataset dengan total 213 citra wajah yang menunjukkan 7 (tujuh) ekspresi wajah. Ekspresi wajah pada dataset tersebut yaitu *anger*, *disgust*, *fear*, *happy*, *neutral*, *sadness*, dan *surprised*. Facial Landmark digunakan sebagai ekstraksi fitur wajah. Model klasifikasi yang digunakan pada penelitian ini adalah ELM, SVM, dan *k*-NN. Masing masing model klasifikasi akan dicari nilai parameter terbaik dengan menggunakan 80% dari total data. 5- fold cross-validation digunakan untuk mencari parameter terbaik. Pengujian model dilakukan dengan 20% data dengan metode evaluasi akurasi, F1 Score, dan waktu komputasi. Nilai parameter terbaik pada ELM adalah menggunakan 40 hidden neuron, SVM dengan nilai  $C = 10^5$  dan 200 iterasi, sedangkan untuk *k*-NN menggunakan 3 *k* tetangga. Hasil uji menggunakan parameter tersebut menunjukkan ELM merupakan algoritme terbaik diantara ketiga model klasifikasi tersebut. Akurasi dan F1 Score untuk klasifikasi ekspresi wajah untuk ELM mendapatkan nilai akurasi sebesar 0.76 dan F1 Score 0.76, sedangkan untuk waktu komputasi membutuhkan waktu  $6.97 \times 10^{-3}$  detik.

**Kata kunci:** deteksi, ekspresi wajah, perbandingan, klasifikasi

## COMPARISON OF CLASSIFICATION METHODS FOR FACIAL EXPRESSION RECOGNITION USING FACIAL LANDMARK FEATURE

### Abstract

Human-machine interaction, especially facial behavior is considered to be use in user personalization. Feature extraction and classification model combinations can be used for a machine to understand the human facial expression. However, which classification base method should be used is not yet known. This study compares three classification methods for facial expression recognition. JAFFE dataset is used in this study with a total of 213 facial images which shows seven facial expressions. The seven facial expressions are anger, disgust, fear, happy, neutral, sadness, dan surprised. Facial Landmark is used as a facial component features. The classification model used in this study is ELM, SVM, and *k*-NN. The hyperparameter of each model is searched using 80% of the total data. 5-fold cross-validation is used to find the hyperparameter. The testing is done using 20% of the data and evaluated using accuracy, F1 Score, and computation time. The hyperparameter for ELM is 40 hidden neurons, SVM with  $C = 10^5$  and 200 iteration, while *k*-NN used 3 *k* neighbors. The experiment results show that ELM outperforms other classification methods. The accuracy and F1 Score achieved by ELM is 0.76 and 0.76, respectively. Meanwhile, time computation takes  $6.97 \times 10^{-3}$  seconds.

**Keywords:** detection, facial expression, comparison, classification

### 1. PENDAHULUAN

Seiring dengan berkembangnya kecerdasan buatan, *Intelligent Human Machine Interaction* khususnya pada bidang *facial behavior* saat ini

mulai diperhatikan (Martinez & Valstar, 2015). Ekspresi wajah adalah salah satu bagian penting pada komunikasi non-verbal (Torre & Cohn, 2011). Penggunaan ekspresi wajah juga telah digunakan

dalam *Human Computer Interaction* (HCI) untuk melakukan personalisasi yang disesuaikan dengan pengguna (McDuff dkk, 2013). Beberapa aplikasi telah dibuat dengan menggunakan ekspresi wajah di berbagai bidang yaitu pendidikan (McDaniel dkk. 2007), kesehatan (Girard dkk, 2013), industri (Busso & Jain, 2012), dan hiburan (Bloom dkk, 2014).

Penggunaan sumber wajah banyak dilakukan karena ekspresi emosi dapat dengan mudah diketahui dari ekspresi pada wajah. Emosi dari raut wajah memberikan proporsi sebesar 55% pada komunikasi manusia. Sedangkan percakapan memberikan proporsi emosi sebesar 7% (Mehrabian, 2008). Selain itu ekspresi wajah lebih mudah dipahami mesin dibandingkan dengan informasi sinyal fisiologis lain dari manusia. Proporsi dan kemudahan tersebut menunjukkan dengan menggunakan wajah saja dapat memberikan informasi yang cukup besar untuk mengetahui keadaan emosi manusia. Banyak dari penelitian sebelumnya menggunakan emosi dasar yang dikemukakan oleh Ekman (Ekman dkk., 2013) untuk mengenali emosi pada manusia. Emosi yang dideteksi yaitu *happy*, *sad*, *anger*, *joy*, *disgust*, dan *neutral*. Untuk dapat mengklasifikasikan emosi, ekstraksi fitur pada wajah dan model klasifikasi digunakan.

Kombinasi dari ekstraksi fitur wajah dengan metode klasifikasi dapat dilatih pada mesin agar mengenali ekspresi wajah seseorang. Salah satu cara untuk mendeteksi bagian dari pada wajah adalah Facial Landmark. Facial Landmark dikembangkan oleh Kazemi dan Sullivan (Kazemi & Sullivan, 2014). Facial Landmark memberikan penanda pada wajah dengan memberikan titik-titik tertentu pada wajah. Jumlah titik-titik tersebut adalah 68 titik yang menyatakan pasangan nilai koordinat titik ( $x,y$ ). Nilai titik-titik tersebut adalah pasangan koordinat yang akan membentuk fitur vektor.

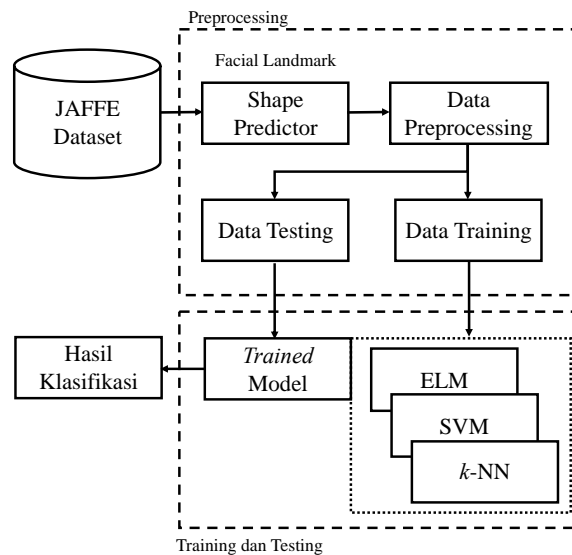
Penelitian sebelumnya oleh (Dino & Abdulrazaq, 2019) membuat model klasifikasi pengenalan ekspresi wajah dengan menggunakan metode klasifikasi. Tiga metode klasifikasi tersebut adalah SVM,  $k$ -NN, dan MLP. Ekspresi emosi yang akan di klasifikasikan adalah emosi dasar yaitu *normal*, *happy*, *angry*, *contempt*, *surprise*, *sad*, *fear*, dan *disgust*. Ekstraksi fitur yang digunakan pada penelitian tersebut menggunakan Histogram of Oriented Gradients (HOG). PCA digunakan untuk menurunkan nilai dimensi dari fitur. Pada penelitian ini SVM menghasilkan nilai klasifikasi tertinggi dengan nilai 93.53%. Pada penelitian lain, studi komparasi dilakukan untuk membandingkan  $k$ -NN, SVM dan Random Forest untuk klasifikasi ekspresi wajah (Nugrahaendi & Mutijarsa, 2016). Pada penelitian ini ekstraksi fitur dilakukan dengan menggunakan jarak dari setiap fitur wajah. Ekspresi emosi yang diklasifikasikan pada penelitian ini yaitu *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad* dan *surprise*. Akurasi tertinggi dihasilkan dari model  $k$ -

NN dengan akurasi 98.85%, sedangkan untuk Random Forest sebesar 98.5% dan SVM sebesar 90%.  $k$ -NN digunakan secara khusus untuk mengenali ekspresi wajah (Sohail & Bhattacharya, 2007). Ekstraksi pada wajah yang dilakukan pada penelitian ini menggunakan Action Unit (AU) yang difokuskan pada lokasi pada mulut, mata, alis. Ekspresi emosi yang akan diklasifikasikan adalah 6 (enam) emosi dasar. Hasil yang didapatkan dari penelitian ini  $k$ -NN mampu memberikan klasifikasi yang cukup tinggi yaitu sebesar 90.76%. Dari beberapa penelitian tersebut satu metode bisa lebih unggul dibandingkan dengan metode lain. Sehingga, belum diketahui secara pasti metode mana yang lebih baik untuk pengenalan ekspresi wajah.

Pada penelitian ini akan dilakukan perbandingan metode klasifikasi untuk mendeteksi ekspresi wajah manusia. Beberapa metode yang digunakan pada penelitian ini diwakili oleh metode berbasis Neural Network, berbasis kernel, dan berbasis tetangga. Metode-metode tersebut yaitu, Extreme Learning Machine (ELM), Support Vector Machine (SVM), dan  $k$ -NN. Penelitian ini diharapkan dapat digunakan sebagai dasar sebagai pertimbangan awal untuk memilih metode untuk pengembangan selanjutnya.

## 2. METODE PENELITIAN

Secara umum alur penelitian ini dapat dilihat pada Gambar 1 berikut:



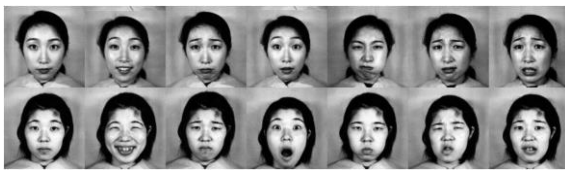
Gambar 1. Alur penelitian

Tahapan penelitian dimulai dari ekstraksi dataset dengan menggunakan *Shape Predictor* pada FL. Setelah itu data akan di *preprocess* terlebih dahulu. Kemudian dataset dibagi menjadi 2 (dua) bagian yaitu, data *training* dan data *testing*. Data *training* kemudian digunakan untuk membuat pemodelan dengan menggunakan 3 (tiga) metode klasifikasi. Model klasifikasi yang digunakan pada penelitian ini yaitu ELM, SVM, dan  $k$ -NN. Model yang telah dilatih kemudian digunakan untuk

menguji data yang belum pernah digunakan pada proses *training*.

**2.1. Pengumpulan Data**

Dataset pada penelitian ini menggunakan dataset sekunder. Dataset yang digunakan pada penelitian yaitu JAFFE (Japanese Female Face Expression). Dataset JAFFE memiliki 213 objek gambar yang terdiri dari 7 (tujuh) ekspresi wajah. Ekspresi wajah tersebut adalah *anger*, *disgust*, *fear*, *happy*, *neutral*, *sadness*, dan *surprised*. Ekspresi wajah tersebut diambil dari 60 subjek. Data ini dapat diakses pada website <http://www.kasrl.org/jaffe.html>. Contoh dari dataset JAFFE dapat dilihat pada Gambar 2.



Gambar 2. Cuplikan dataset JAFFE

Total terdapat 213 data dengan proporsi ekspresi *anger* sebanyak 30, ekspresi *disgust* sebanyak 29, ekspresi *fear* sebanyak 32, ekspresi *happy* sebanyak 31 dan ekspresi *neutral* sebanyak 30, ekspresi *sadness* sebanyak 31, dan ekspresi *surprised* sebanyak 30.

**2.2. Data Preprocessing**

*Preprocessing* yang dilakukan pada tahap ini bertujuan untuk mendeteksi letak titik pada wajah, menormalisasi data dan membagi data.

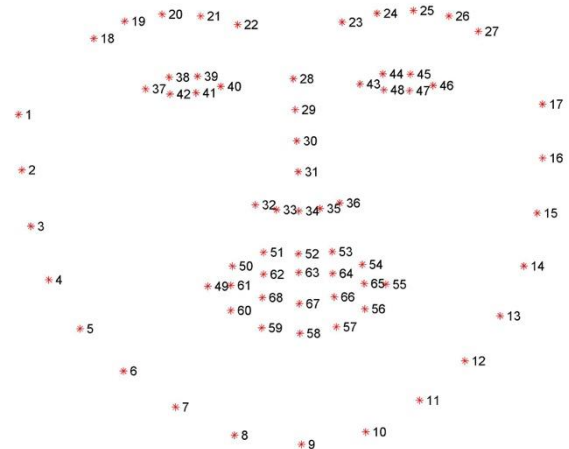
Langkah pertama pada tahapan ini adalah melakukan deteksi *Landmark*. Deteksi *Landmark* dilakukan pada wajah dengan teknik *Shape Predictor*. Teknik ini bertujuan untuk melakukan estimasi untuk menemukan 68 titik dengan koordinat (x, y) titik dari wajah pada gambar yang dimasukkan agar dapat dilakukan deteksi titik pada wajah. Ilustrasi *landmark* tersebut dapat dilihat pada Gambar 4. Dari titik-titik tersebut akan di ekstraksi menjadi nilai-nilai vektor sejumlah 136 titik. FL yang digunakan pada penelitian ini menggunakan library *DLib* yang mengimplementasikan penelitian yang dilakukan oleh Kazemi & Sullivan (2014).

Langkah selanjutnya adalah normalisasi. Normalisasi dilakukan ketika titik data pada wajah telah ditemukan. Metode normalisasi data yang digunakan pada tahap ini adalah L2-Norm untuk mengubah skala pada data menjadi lebih kecil. Persamaan L2-Norm dapat dilihat pada Persamaan 1.

$$\|x\|_2 = (\sum_{i=1}^N |x_i|^2)^{1/2} \tag{1}$$

Data yang telah ternormalisasi kemudian dibagi menjadi sejumlah proporsi 80% data *training*

dan 20% data *testing*. Pembagian proporsi data tersebut menggunakan metode *Stratified k-Fold*.



Gambar 3. Facial Landmark

**2.3. Metode Klasifikasi**

Tahapan setelah dilakukan pengumpulan data dan *pre-prosesing* data adalah pemodelan. Pemodelan yang dilakukan pada penelitian ini adalah mengklasifikasikan emosi yang didapatkan dari fitur ekstraksi wajah dengan menggunakan FL. Model klasifikasi yang digunakan pada penelitian ini adalah ELM, SVM, dan *k*-NN.

**ELM**

Algoritme ELM dikemukakan oleh Huang dkk. (2006). Algoritme ELM memiliki 2 (dua) tahapan utama yaitu proses *training* dan *testing*. Pada tahap *training*, algoritme ini akan menginisialisasi nilai matriks *input* bobot dan nilai bias. Kemudian akan menghitung nilai matriks *H'*. Setelah itu akan dihitung nilai matriks *H* sebagai *output* dari *hidden layer* menggunakan fungsi aktivasi. Fungsi aktivasi yang digunakan pada penelitian ini menggunakan fungsi aktivasi *sigmoid*. Langkah selanjutnya adalah menghitung matriks *H<sup>+</sup>* Matriks *H<sup>+</sup>* adalah matriks *Moore-Penrose Generalized Inverse*. Langkah terakhir menghitung bobot *output*. Sedangkan untuk proses *testing* dilakukan dengan menghitung matriks *H* dengan menggunakan bobot masukan nilai bias dan  $\hat{\beta}$  (matriks *output weight*) dari proses *training*. Kemudian menghitung nilai *output* dengan Persamaan 2.

$$\hat{Y} = H \cdot \hat{\beta} \tag{2}$$

Dimana  $\hat{\beta}$  adalah matriks *output weight*; *H* adalah matriks fungsi aktivasi;  $\hat{Y}$  adalah matriks *output* hasil klasifikasi. Secara umum algoritma ELM dapat di deskripsikan pada Tabel 1.

Tabel 1. ELM Pseudocode	
<b>Algoritme ELM</b>	
<b>Input:</b>	
$w_i$ (input random bobot)	
$b_i$ (input random bias $[1 \times N]$ )	

Algoritme ELM
$N$ (jumlah <i>hidden layer</i> )
$g(x)$ (fungsi aktivasi)
$y$ ( <i>query image</i> )
<b>Ouput:</b> Hasil prediksi kelas $x$
1. Tetapkan parameter hidden node secara acak ( $w_i, b_i$ )
2. Hitung output matriks $\mathbf{H}$ , Dimana $\mathbf{H} = \{h_{ij}\}$ ( $i = 1, \dots, x$ , dan $j = 1, \dots, N$ ) dan $h_{ij} = g(w_j \cdot x_i + b_j)$
3. Hitung matriks bobot keluaran dengan $\hat{\beta} = \mathbf{H}^{\dagger} \mathbf{T}$ , dimana $\mathbf{H}^{\dagger}$ adalah <i>Moore-Penrose inverse</i> matriks dari $\mathbf{H}$
4. Hitung nilai output aktual $\mathbf{o}$ yang berhubungan dengan $\mathbf{y}$
5. Label( $\mathbf{y}$ ) = $\text{argmax}(\mathbf{o})$

**SVM**

Secara sederhana algoritme SVM adalah proses untuk mencari *hyperplane* terbaik yang dimana dapat memisahkan dua buah kelas pada *input space* (Boser dkk., 1992). Proses klasifikasi pada SVM adalah proses menemukan garis batasan (*hyperplane*) yang membedakan atau memisahkan kedua kelas atau lebih. *Support Vector Machine* adalah suatu metode klasifikasi yang bertujuan untuk menemukan *Maximum Marginal Hyperplane* yang merupakan batas pemisah maksimal untuk semua kelas.

Untuk mendapatkan *hyperplane* pemisah terbaik diantara kedua kelas adalah dengan menghitung nilai margin *hyperplane* dan mencari titik dengan nilai tertinggi. Margin merupakan jarak antara *hyperplane* dengan pola terdekat dari masing-masing kelas. Pola yang paling dekat ini disebut sebagai *Support Vector*. Semakin besar *margin* maka semakin tinggi akurasi. Proses untuk menemukan lokasi *hyperplane* ini adalah inti dari proses pembelajaran pada SVM. SVM yang digunakan pada penelitian ini adalah SVM dengan *Kernel Linear*. *Pseudocode* algoritme SVM dapat dilihat pada Tabel 2.

Tabel 2. SVM Pseudocode

Algoritme SVM
<b>Input:</b>
$N_{in}$ (jumlah <i>input</i> vektor)
$N_{sv}$ (jumlah <i>support</i> vektor)
$N_{ft}$ (jumlah fitur dalam <i>support</i> vektor)
$SV[N_{sv}]$ ( <i>suport</i> vektor array)
$IN[N_{in}]$ ( <i>input</i> vektor array)
$b^*$ (bias)
<b>Ouput:</b> $F$ ( <i>fungsi keputusan</i> )
for $i \leftarrow 1$ to $N_{in}$ do
$F = 0$
for $j \leftarrow 1$ to $N_{sv}$ do
$dist = 0$
for $k \leftarrow 1$ to $N_{ft}$ do
$dist += (SV[j].feature[k] - IN[i].feature[k])^2$
end for
$k = \exp(-\gamma \times dist)$
$F += SV[j].a^* \times k$
end for
$F = F + b^*$
end for

**k-NN**

Algoritme *k*-Nearest Neighbor (*k*-NN) merupakan algoritme klasifikasi pencarian data

dalam *k* terdekat (Aggarwal, 2015). Algoritme *k*-NN dikemukakan oleh Cover & Hart (1967). Apabila data  $x$  memiliki sejumlah *k* tetangga terdekat, dan  $x$  didominasi oleh kelas  $c_i$  maka  $x$  merupakan kelas  $c$ . Metode ini membutuhkan nilai *k* dan perhitungan jarak didalamnya untuk menentukan kelas yang akan diprediksi.

Perhitungan jarak pada algoritme *k*-NN menggunakan metode perhitungan jarak *Euclidean Distance* yang dapat dilihat pada Persamaan 3.

$$d(x_1, x_2) = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2} \quad (3)$$

Tabel 3. K-NN Pseudocode

Algoritme k-NN
<b>Input:</b> $x, S, d$
<b>Ouput:</b> hasil prediksi kelas $x$
for $(x', c') \in S$ do
Hitung jarak $d(x', x)$
end for
Urutkan jarak $ S $ dari yang terdekat hingga terjauh
Pada <i>k</i> nearest neighbors, hitung jumlah kemunculan setiap kelas $c_j$
Tentukan $x$ berdasarkan kelas mayoritas

Dimana  $x_1$  dan  $x_2$  adalah data ke-1 dan ke-2;  $d(x_1, x_2)$  adalah *Euclidian distance*;  $n$  adalah jumlah fitur. Setelah menghitung jarak sesuai dengan *k* yang ditentukan maka hitung jumlah tetangga terdekat dengan kelas terbanyak. Kelas yang merupakan tetangga terbanyak akan mengklasifikasikan data  $x$  menjadi kelas ekspresi wajah tersebut. *Pseudocode* algoritme *k*-NN terdapat pada Tabel 3.

**2.4. Hyperparameter, Training dan Testing**

Pembagian data dilakukan sebelum tahap pencarian parameter terbaik (*hyperparameter*). Pembagian dataset menjadi dataset *training* dan *testing* dilakukan dengan proporsi 80% data *training* dan 20% data *testing*. Pembagian data dilakukan dengan memperhatikan proporsi kelas ekspresi emosi. Pembagian data tersebut dilakukan dengan menggunakan teknik *stratified sampling*. *Stratified sampling* memastikan untuk setiap kelas ekspresi emosi akan mempunyai perwakilan yang nanti akan digunakan pada saat proses *training* maupun *testing*.

*Hyperparameter* terbaik dari setiap model akan dicari terlebih dahulu untuk mendapatkan hasil akurasi yang maksimal. *Hyperparameter* dicari dengan menggunakan 80% dari data *training*. Dari data tersebut akan dibagi lagi menjadi proporsi 80% data *training* dan 20% data *testing*. Langkah selanjutnya akan dilakukan sebanyak *5-fold cross validation*. Pada model klasifikasi ELM akan dicari parameter jumlah *hidden neuron* terbaik dengan range 5 - 50. Untuk model klasifikasi SVM akan dicari nilai kombinasi parameter *C* dengan nilai  $C = 10^{-5}$  sampai  $10^5$  dan jumlah iterasi terbaik dari nilai 50 sampai 1000 dengan *Kernel Linear*. Sedangkan

untuk  $k$ -NN akan dicari nilai  $k$ -tetangga terbaik dari range 3 sampai 35.

Proses *training* dilakukan setelah mendapatkan hasil parameter terbaik dari tahapan sebelumnya. Parameter terbaik tersebut digunakan untuk proses *training* dengan menggunakan 80% data. Sedangkan untuk *testing* menggunakan 20% data yang belum pernah digunakan pada saat proses pencarian *hyperparameter* maupun proses *training*.

## 2.5. Metode Evaluasi

Akurasi, *Micro-Precision*, *Micro-Recall*, dan F1 Score (*micro*) digunakan untuk mengetahui kemampuan metode klasifikasi untuk deteksi ekspresi wajah. Persamaan Akurasi, *Micro-Precision*, *Micro-Recall*, dan F1 Score (*micro*) dapat dilihat pada Persamaan (4) – Persamaan (7). Selain itu, waktu komputasi juga digunakan untuk mengetahui kecepatan proses komputasi.

$$\text{Akurasi} = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (4)$$

$$\text{Micro - Precision} = \frac{TP1+TP2}{TP1+FP1+TP2+FP2} \quad (5)$$

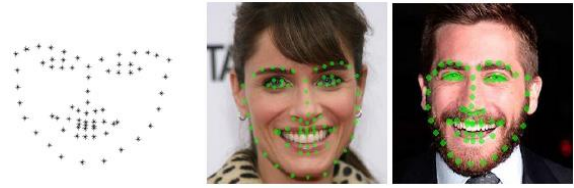
$$\text{Micro - Recall} = \frac{TP1+TP2}{TP1+FN1+TP2+FN2} \quad (6)$$

$$F1 = 2 \times \frac{\text{Micro-Precision} * \text{Micro-Recall}}{\text{Micro-Precision} + \text{Micro-Recall}} \quad (7)$$

## 3. FACIAL LANDMARK (FL)

FL merupakan salah satu bagian dari permasalahan deteksi bentuk yang dikhususkan pada wajah. FL mendeteksi bagian geometris dari wajah. Pada bagian wajah akan diberikan titik-titik tertentu yang akan membentuk wajah beserta titik-titik lainnya yang ada pada bagian wajah, seperti alis, mata, hidung, dan mulut. Titik-titik pada wajah dan komponen pada wajah akan diberikan titik untuk dapat digunakan mendeteksi perubahan yang terjadi pada wajah seperti terlihat pada Gambar 4.

Pada dasarnya, tahapan FL dapat dilakukan dalam 2 (dua) tahap. Tahap 1 yaitu lokalisasi bagian wajah. Lokalisasi ini ditujukan untuk mendapatkan bagian wajah terlebih dahulu. Tujuannya adalah untuk membatasi ruang deteksi agar tidak terlalu besar untuk menentukan titik-titik penting pada wajah. Tahap 2 mendeteksi titik-titik pada wajah dan bagian-bagian pada wajah. Total terdapat 68 titik di wajah yang merepresentasikan bentuk wajah dan komponen-komponen pada wajah. Beberapa metode deteksi yang ada dapat melokalisasi titik-titik diantaranya: dagu, mulut, hidung, mata kanan, mata kiri, alis kanan, dan alis kiri.



Gambar 4. Facial Landmark  
(Wu & Ji, 2019)

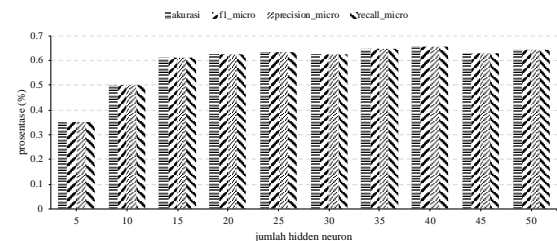
## 4. HASIL EKSPERIMEN

### 4.1. Parameter terbaik

Hasil dari parameter terbaik di setiap model memberikan hasil ketepatan akurasi yang berbeda-beda. Pada pengujian parameter terbaik didapatkan nilai akurasi antara 61% sampai 81%.

### ELM

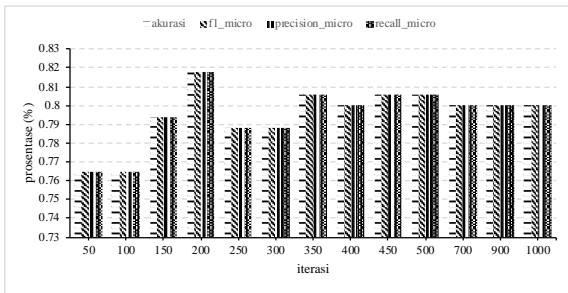
Parameter jumlah *hidden neuron* yang mendapatkan akurasi terbaik dari model ELM adalah 40 *hidden neuron*. Penggunaan jumlah *hidden neuron* yang kecil kurang memberikan nilai akurasi yang baik. Penambahan jumlah di *hidden neuron* berdampak pada naiknya nilai akurasi, akan tetapi pada jumlah *hidden neuron* dengan lebih dari 20 nilai akurasinya fluktuatif. Nilai akurasi tertinggi didapatkan pada jumlah *hidden neuron* sejumlah 40. Akurasi yang didapat dari jumlah *hidden neuron* tersebut adalah 61%. Hasil dari eksperimen tersebut dapat dilihat pada Gambar 5.



Gambar 5. Hasil akurasi terbaik pada ELM

### SVM

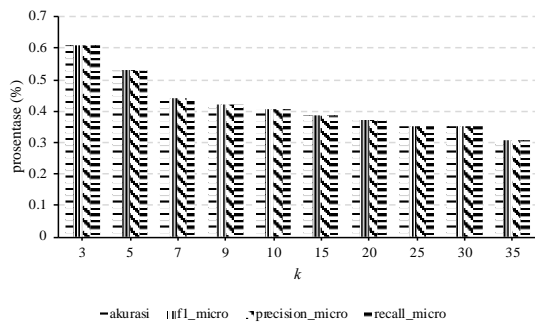
Pada pemodelan SVM kombinasi nilai *hyperparameter*  $C = 10^{-5}$  dan 200 iterasi menunjukkan hasil akurasi yang paling tinggi dibandingkan dengan kombinasi parameter lainnya. Nilai  $C = 10^{-5}$  memberikan nilai akurasi yang fluktuatif di awal iterasi dengan akurasi yang meningkat secara signifikan pada iterasi ke 200. Nilai akurasi tersebut kemudian menurun dan fluktuatif pada nilai antara 78% - 80%. Hasil eksperimen pencarian *hyperparameter* terbaik tersebut dapat dilihat pada Gambar 6.



Gambar 6. Hasil akurasi terbaik pada SVM

**k-NN**

Pada model klasifikasi *k*-NN, nilai *k* tetangga yang menghasilkan akurasi terbaik didapatkan nilai *k* = 3. Pada nilai *k* tetangga kecil menghasilkan nilai akurasi yang lebih tinggi dibandingkan dengan *k* tetangga dengan nilai tetangga yang semakin besar. Semakin besar *k* tetangga yang digunakan untuk proses klasifikasi menghasilkan nilai akurasi yang semakin kecil. Hasil eksperimen pencarian *hyperparameter* terbaik *k*-NN dapat dilihat pada Gambar 7.



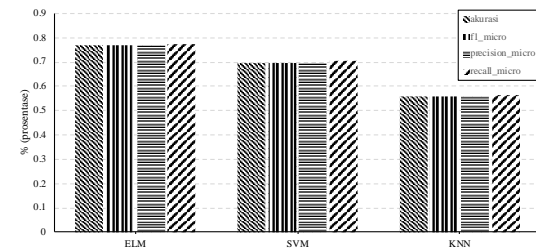
Gambar 7. Hasil akurasi terbaik pada k-NN

**4.2. Perbandingan Klasifikasi**

Pada tahap ini hasil *hyperparameter* terbaik dari setiap model digunakan untuk mengklasifikasikan 20% data *testing* ekspresi wajah. Hasil dari perbandingan antara ketiga model tersebut dapat dilihat pada Tabel 4. Hasil dari Tabel 4 didapatkan bahwa untuk akurasi tertinggi didapatkan oleh model ELM dengan akurasi dan F1 Score dengan nilai 0.76. Untuk algoritme SVM menghasilkan akurasi dan F1 Score sebesar 0.69. Sedangkan untuk algoritme *k*-NN nilai akurasi dan F1 Score yaitu sebesar 0.55.

Tabel 4. Rancangan Analisis Komputasi

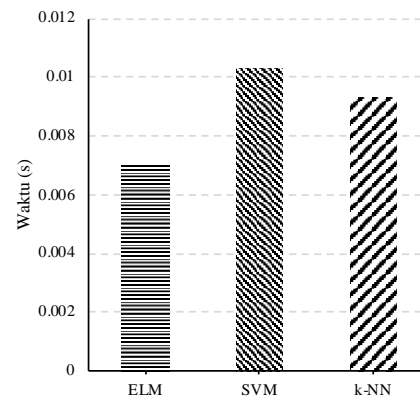
Evaluasi	ELM (%)	SVM (%)	k-NN (%)
akurasi	0.76	0.69	0.55
f1_micro	0.76	0.69	0.55
precision_micro	0.76	0.69	0.55
recall_micro	0.77	0.70	0.56



Gambar 8. Perbandingan 3 metode klasifikasi

Visualisasi dari Tabel 4 dapat dilihat pada Gambar 8. Dari Gambar 8 dapat dilihat bahwa model ELM memberikan hasil evaluasi yang lebih baik dibandingkan dengan SVM dan *k*-NN. Akurasi, F1 Score, *Micro-Precision*, dan *Micro-Recall* pada pemodelan ELM mengungguli nilai-nilai evaluasi pada pemodelan SVM dan *k*-NN.

Pebandingan lain dari segi waktu komputasi menunjukkan model ELM lebih cepat dibandingkan dengan model klasifikasi yang lainnya. Model klasifikasi ELM membutuhkan waktu  $6.97 \times 10^{-3}$  detik untuk mengklasifikasikan ekspresi emosi. Sedangkan untuk model SVM dan *k*-NN membutuhkan waktu  $1.02 \times 10^{-2}$  dan  $9.32 \times 10^{-3}$ . Hasil perbandingan waktu komputasi tersebut dapat dilihat pada Gambar 9.



Gambar 9. Perbandingan waktu komputasi

**5. KESIMPULAN DAN SARAN**

Penelitian ini melakukan perbandingan model klasifikasi untuk deteksi ekspresi emosi pada wajah manusia. Perbandingan dilakukan pada tiga model klasifikasi berbasis Neural Network yang diwakili oleh ELM, kernel yang diwakili oleh SVM, dan tetangga yang diwakili oleh *k*-NN. Pada penelitian ini hasil model klasifikasi terbaik ditunjukkan oleh ELM baik dari segi akurasi maupun waktu komputasi. Dari hasil eksperimen ini model klasifikasi ELM memberikan akurasi tertinggi dibandingkan dengan model klasifikasi lain dengan nilai akurasi dan F1 Score dengan nilai 0.76. Sedangkan untuk waktu komputasi, model ELM membutuhkan waktu  $6.97 \times 10^{-3}$  detik.

**DAFTAR PUSTAKA**

- AGGARWAL, C.C., 2015. *Data mining: the textbook*. Switzerland: Springer International Publishing.
- BOSER, B.E., GUYON, I.M. and VAPNIK, V.N., 1992, July. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152).
- BUSSO, C. and JAIN, J., 2012. Advances in multimodal tracking of driver distraction. In *Digital Signal Processing for In-Vehicle Systems and Safety* (pp. 253-270). Springer, New York, NY.
- BLOM, P.M., BAKKES, S., Tan, C.T., WHITESON, S., ROIJERS, D., VALENTI, R. and GEVERS, T., 2014, September. Towards personalised gaming via facial expression recognition. In Tenth Artificial Intelligence and Interactive Digital Entertainment Conference.
- COVER, T. and HART, P., 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), pp.21-27.
- DE LA TORRE, F. and COHN, J.F., 2011. Guide to Visual Analysis of Humans: Looking at People, chapter Facial Expression Analysis. *Springer*, 9, pp.31-33.
- DINO, H.I. and ABDULRAZZAQ, M.B., 2019, April. Facial expression classification based on SVM, KNN and MLP classifiers. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 70-75). IEEE.
- EKMAN, P., FRIESEN, W.V. and ELLSWORTH, P., 2013. *Emotion in the human face: Guidelines for research and an integration of findings* (Vol. 11). Elsevier.
- GIRARD, J.M., COHN, J.F., MAHOOR, M.H., MAVADATI, S. and ROSENWALD, D.P., 2013, April. Social risk and depression: Evidence from manual and automatic facial expression analysis. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (pp. 1-8). IEEE.
- HUANG, G.B., ZHU, Q.Y. and SIEW, C.K., 2006. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), pp.489-501.
- KAZEMI, V. and SULLIVAN, J., 2014. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1867-1874).
- MARTINEZ, B. and VALSTAR, M.F., 2016. Advances, challenges, and opportunities in automatic facial expression recognition. In *Advances in face detection and facial image analysis* (pp. 63-100). Springer, Cham.
- MCDUFF, D., EL KALIOUBY, R., DEMIRDJIAN, D. and PICARD, R., 2013, April. Predicting online media effectiveness based on smile responses gathered over the internet. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)* (pp. 1-7). IEEE.
- MCDANIEL, B., D'MELLO, S., KING, B., CHIPMAN, P., TAPP, K. and GRAESSER, A., 2007. Facial features for affective state detection in learning environments. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 29, No. 29).
- MEHRABIAN, A., 2008. Communication without words. *Communication theory*, 6, pp.193-200.
- NUGRAHAENI, R.A. and MUTIJARSA, K., 2016, August. Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. In *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)* (pp. 163-168). IEEE.
- SOHAIL, A.S.M. and BHATTACHARYA, P., 2007, March. Classification of facial expressions using k-nearest neighbor classifier. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications* (pp. 555-566). Springer, Berlin, Heidelberg.
- WU, Y. and JI, Q., 2019. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2), pp.115-142.

*Halaman ini sengaja dikosongkan*