

# Peningkatan Deep Neural Network pada Kasus Prediksi Diabetes Menggunakan PSO

*Improvement of Deep Neural Network in Diabetes Prediction Cases Using PSO*

Rusmal Firmansyah<sup>1</sup>, Guruh Fajar Shidik<sup>2</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

E-mail: <sup>1</sup>p31202202525@mhs.dinus.ac.id, <sup>2</sup>guruh.shidik@dsn.dinus.ac.id

## Abstrak

Diabetes adalah ancaman utama bagi kesehatan penduduk dunia yang saat ini merupakan penyebab utama kematian pada penduduk dunia yang berusia kurang dari 60 tahun. Dengan menggunakan *Machine Learning* diharapkan mampu memprediksi diabetes. Dengan menggunakan dataset *Pima Indians Diabetes* (PIMA Dataset). Pada penelitian ini dilakukan pengujian dengan menggunakan 2 Algoritma dan 1 Algoritma yang dioptimasi. Pengujian Pertama Menggunakan *Support Vector Machine* (SVM), Pengujian Kedua menggunakan *Deep Neural Network* (DNN) dan Pengujian Ketiga menggunakan DNN yang dikombinasikan dengan *Particle Swarm Optimize* (PSO). Pemilihan data yang digunakan sebagai *training* dilakukan dengan menggunakan *Non-Random Sampling*. Dalam Penelitian ini pengujian pertama dengan menggunakan SVM dengan melakukan pengujian tanpa menggunakan *kernel* dan menggunakan *kernel Linear, Sigmoid, Polynomial dan Radial Basis Function (RBF)*. Untuk Pengujian Kedua dilakukan dengan menggunakan DNN tanpa menggunakan Optimalisasi atau DNN original dengan dilakukan pengujian dengan jumlah *hidden layer* 2 dan jumlah *neuron* 8 sampai 10 pada setiap *hidden layer*. Pengujian Ketiga dilakukan dengan menggunakan DNN yang dioptimalisasi dengan menggunakan PSO. Pada Pengujian Ketiga dilakukan pengujian dengan jumlah *hidden layer* 2 dan jumlah *neuron* 8 sampai 10 pada setiap *hidden layer*. Hasil yang diperoleh menunjukkan bahwa DNN yang dioptimasi dengan PSO mampu memberikan akurasi tertinggi dengan jumlah *hidden layer* 1 sebanyak 9 *node* dan jumlah *hidden layer* 2 sebanyak 8 *node* dengan jumlah *iterasi* pada PSO sebanyak 166 iterasi.

Kata kunci: Diabetes, DNN, PSO, SVM

## Abstract

*Diabetes is a major threat to the health of the world population and currently the leading cause of death among individuals under 60 years old. By using Machine Learning, it is hoped to be able to predict diabetes using the Pima Indians Diabetes dataset (PIMA Dataset). This study conducted testing using 2 algorithms and 1 optimized algorithm. The first testing used Support Vector Machine (SVM), the second testing used Deep Neural Network (DNN), and the third testing used DNN combined with Particle Swarm Optimization (PSO). The selection of data used for training was done using Non-Random Sampling. In this study, the first testing was performed using SVM by conducting testing without using a kernel and using Linear Kernel, Sigmoid Kernels, Polynomial Kernels, and Radial Basis Function (RBF) Kernels. The second testing was performed using DNN without optimization or original DNN, with testing conducted with 2 hidden layers and 8 to 10 neurons in each hidden layer. The third testing was performed using DNN optimized with PSO. In the third testing, testing was conducted with 2 hidden layers and 8 to 10 neurons in each hidden layer. The results obtained showed that the DNN optimized with PSO was able to provide the highest accuracy with 9 nodes in the first hidden layer and 8 nodes in the second hidden layer, with a total of 166 iterations in the PSO.*

Keywords: Diabetes, DNN, PSO, SVM

## 1. PENDAHULUAN

Saat ini diabetes adalah ancaman utama bagi kesehatan penduduk dunia. Setiap tujuh detik, diabetes menjadi penyebab kematian seseorang yang berusia kurang dari 60 tahun[1]. Dilansir dalam situs resmi Organisasi Kesehatan Dunia (WHO), Diabetes merupakan kondisi yang terjadi akibat pankreas tidak mampu menghasilkan insulin sesuai dengan kebutuhan yang dibutuhkan oleh tubuh manusia. Dengan kondisi seperti ini maka Diabetes digolongkan dalam penyakit yang kronis. Pada tahun 2014 terdapat 8,5% penduduk dalam kategori dewasa (usia 18 tahun ke atas) menderita diabetes. Pada tahun 2019, diabetes menjadi penyebab kematian utama pada 1,5 juta penduduk didunia dan 48% dari seluruh kematian yang disebabkan oleh diabetes terjadi pada usia kurang dari 70 tahun. Sebanyak 460.000 kematian dari penyakit ginjal disebabkan oleh diabetes, dan peningkatan kadar glukosa darah menyebabkan sekitar 20% kematian pada penyakit kardiovaskular. Antara tahun 2000 sampai 2019, ada peningkatan 3% standar angka kematian usia akibat diabetes. Di negara berpenghasilan menengah ke bawah, angka kematian akibat diabetes meningkat 13% [2].

Disisi lain teknologi saat ini berkembang sangat pesat, khususnya *Machine Learning*. *Machine Learning* banyak digunakan dalam berbagai macam permasalahan, seperti kesehatan, pendidikan, perbankan dan lain-lain[3]. Dalam *Machine Learning* dikenal beberapa jenis pemodelan, seperti Klasifikasi, Klustering, Asosiasi. Dengan banyaknya kasus diabetes tersebut dan didukung dengan kemajuan teknologi, diharapkan dapat membantu memprediksi penyakit diabetes tersebut. Dalam kasus ini, digunakan pemodelan Klasifikasi data. Klasifikasi Yang digunakan dengan berdasarkan Algoritma *Support Vector Machine* dan *Deep Neural Network*.

Pada penelitian yang dilakukan oleh Febrian dan kawan-kawan pada tahun 2023 dengan dataset *Pima Indians Diabetes Database* menggunakan Algoritma *K-Nearest Neighbour* (KNN) dan *Naïve Baiyes* memperoleh Akurasi Rata-rata 73,33% untuk KNN dan 76,07% dengan *Naïve Baiyes*[4].

Pada Tahun 2020, Tigga beserta kawan-kawannya, dengan membanding Dataset Diabetes Mereka Sendiri dan *Dataset Pima Indians Diabetes*, diperoleh Hasil Akurasi untuk PIMA dataset *Logistic Regression* (74,4%), KNN (70,8%), *Support Vector Machine* (SVM) (74,4%), *Naive Baiyes* (68,9%), *Decision Tree* (69,7%) dan *Random Forest* (75%). Sedangkan pada Dataset mereka sendiri memiliki 952 data dengan pengidap diabetes sebanyak 267 dan normal 685 dengan 17 *feature* yaitu *Age, Gender, Family history with diabetes, Diagnosed with high blood pressure, Walk/run/physically active, BMI, Smoking, Alcohol consumption, Hours of sleep, Hours of sound sleep, Regular intake of medicine?, Junk food Consumption, Stress, Blood pressure level, Number of Pregnancies, Gestation diabetes, dan Frequency of Urination*. Hasil untuk Dataset mereka sendiri *Logistic Regression* (85,7%), *K-Nearest Neighbour* (77,3%), *Support Vector Machine* (SVM) (86,5%), *Naive Bayes* (80,6%), *Decision Tree* (84%) dan *Random Forest* (94,1%)[5].

Azrar pada tahun 2018, dengan Dataset PIMA menggunakan 3 Algoritma, KNN (65.19%), *Naïve Baiyes* (71,74%) dan *Decision Tree* (75,65)[6].

Sisodia dengan menggunakan Dataset PIMA, melakukan perbandingan 3 Algoritma yaitu *Naïve Bayes*, SVM dan *Decision Tree*. Perbandingan 3 Algoritma tersebut menghasilkan Accuracy Berturut-turut, *Naïve Bayes* (76,3%), SVM (51,3%) dan *Decision Tree* (73,82%)[7].

Chang pada tahun 2022 dengan menggunakan *dataset subset of Behavioral Risk Factor Surveillance System*, menguji 5 Algoritma dengan hasil akurasi *Decision Tree* 81,02%, *Random Forest* 82,26%, KNN 80,55%, *Logistic Regression* 72,64% dan *Naïve Bayes* 70,56% [8].

Dengan menggunakan hasil Elektrokardiogram (EKG) dari masing-masing 20 orang dari kelompok diabetes dan normal, *swapna* memperoleh hasil akurasi 95,7% dengan menggunakan *Convolution Neural Network*, LSTM with SVM (CNN5-LSTM with SVM)[9].

Tabel 1 Perbandingan Algoritma

Ref	Dataset	Algoritma	Algoritma Unggul	Akurasi
[4]	<i>Pima Indians Diabetes Database</i>	<i>KNN, Naïve Bayes</i>	<i>Naïve Bayes</i>	76,07%
[5]	<i>Pima Indians Diabetes Database</i>	<i>Logistic Regression, KNN, SVM, Naïve Bayes, Decision Tree, Random Forest</i>	<i>Random Forest</i>	75%
[5]	<i>Private dataset</i>	<i>Logistic Regression, K-NN, SVM, Naïve Bayes, Decision Tree, Random Forest</i>	<i>Random Forest</i>	94,1%
[6]	<i>Pima Indians Diabetes Database</i>	<i>KNN, Naïve Baiyes, Decision Tree</i>	<i>Decision Tree</i>	75,65%
[7]	<i>Pima Indians Diabetes Database</i>	<i>Naïve Bayes, SVM, Decision Tree</i>	<i>Naïve Bayes</i>	76,3%
[8]	<i>Subset of Behavioral Risk Factor Surveillance System</i>	<i>Decision Tree, Random Forest, KNN, Logistic Regression, Naïve Bayes</i>	<i>Random Forest</i>	82,26%
[9]	<i>Elektrokardiogram (EKG)</i>	<i>CNN5-LSTM with SVM</i>	<i>CNN5-LSTM with SVM</i>	95,7%

Pada penelitian sebelumnya, masih didominasi pada penggunaan *traditional machine learning* seperti KNN, Naïve Baiyes, SVM dan semacamnya. Oleh karena itu, pada penelitian ini dilakukan menggunakan deep learning atau DNN yang akan dioptimalkan dengan algoritma PSO. Pemilihan peningkatan DNN dengan PSO didasari dengan penggabungan kedua algoritma PSO dan DNN pada kasus Digital Modulation Recognition[10].

DNN adalah salah satu algoritma yang sangat dipengaruhi penentuan bobot. Dalam penelitian ini, PSO akan melakukan pencarian bobot terbaik yang akan digunakan oleh DNN. Penelitian ini diharapkan dapat memberikan yang kontribusi signifikan dengan mengintegrasikan Particle Swarm Optimization (PSO) dalam Deep Neural Network (DNN) untuk meningkatkan prediksi diabetes, dan memungkinkan pengembangan model prediksi yang lebih akurat. Tujuan penelitian adalah melakukan optimasi pada algoritma DNN dengan menggunakan algoritma optimasi PSO pada studi kasus prediksi diabetes.

## 2. METODE PENELITIAN

### 2.1 Dataset

Dalam Penelitian ini digunakan Dataset dari *Pima Indians Diabetes Database* (PIMA). Dataset ini merupakan hasil tes diabetes pada beberapa wanita yang memiliki usia minimal 21 tahun dan tinggal di Phoenix, Arizona, AS[6]. Jumlah Dataset 768 dengan 8 fitur. Fitur dalam dataset ini antara lain *Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function* dan *Age*. Dataset ini diklasifikasikan menjadi 2 kelas (Outcome), nilai 1 sebagai penderita diabetes dan nilai 0 sebagai bukan penderita.

### 2.2 Normalisasi Data

Data dinormalisasi dengan menggunakan *z-score* normalisasi. Formula *z-score* sebagai berikut:

$$y = \frac{(x - \mu)}{\sigma} \tag{1}$$

$$\sigma = \sqrt{\frac{(x_i - \mu)^2}{n}} \tag{2}$$

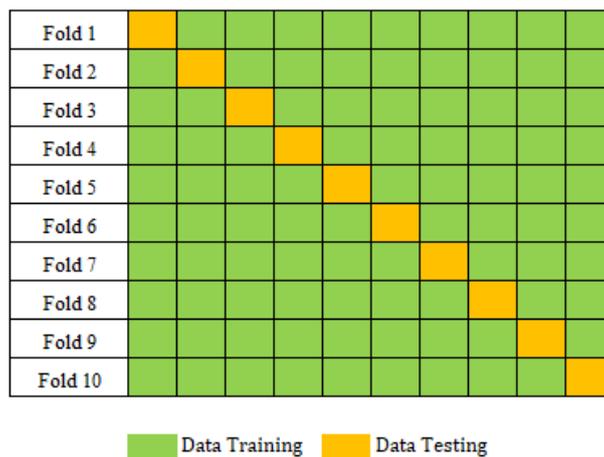
Dengan  $y$  mewakili nilai normalisasi,  $x$  mewakili nilai fitur,  $\mu$  mewakili nilai rata-rata,  $\sigma$  mewakili simpangan baku, dan  $n$  mewakili jumlah data.

Karena kita berhadapan dengan 2 kelas, nilai hasil dibagi menjadi 2 kolom, 1 - 0 untuk menunjukkan adanya kondisi dan 0 - 1 untuk menunjukkan kondisi normal

### 2.3 Seleksi Data

Karna kondisi Data yang mengalami *Imbalanced Data*. Dataset yang tidak seimbang jika digunakan pada kecerdasan buatan, *machine learning*, dan data mining maka sering kali memberikan hasil yang keliru. Hal tersebut dikarenakan algoritma *machine learning* dirancang untuk berkerja secara optimal dengan data yang seimbang[11]. Pemilihan Data didasarkan pada metode *Non-Random Sampling*. *Non-Random sampling* adalah metode pengambilan data yang berdasar pada pemilihan suatu karakteristik atau ciri-ciri untuk mendapatkan *sampel* relevan agar tujuan dari sebuah penelitian dapat tercapai.

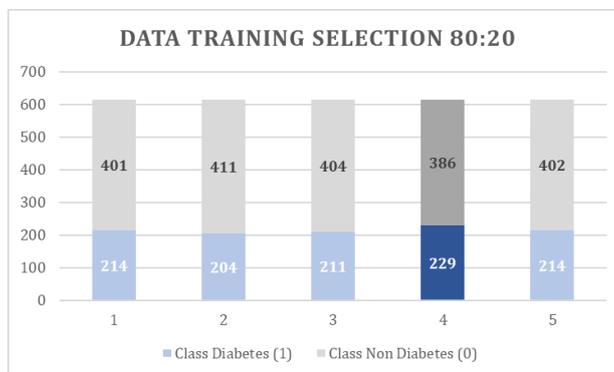
Untuk memilih data, digunakan konsep *K-Fold Cross Validation* untuk memilih area data mana saja yang memiliki *imbalanced data* terkecil. *K-Fold Cross Validation*, melakukan sampel awal dengan membagi dataset menjadi  $K$  subsampel. Satu subsampel dipertahankan sebagai data test dan sampel  $K-1$  lainnya digunakan untuk data latih. *Cross Validation* diulang  $K$  kali, dengan setiap subsampel diverifikasi satu kali. Hasilnya kemudian dirata-ratakan  $K$  kali atau kombinasi lain digunakan untuk mendapatkan estimasi tunggal. Keuntungan dari metode ini adalah subsampel yang dihasilkan secara acak berulang kali digunakan untuk pelatihan dan verifikasi, dan hasilnya diverifikasi sekali setiap kali. Validasi silang 10 kali lipat adalah metode yang paling umum digunakan[12]. Pada Penelitian ini akan dilakukan pengujian dengan pembagian 90:10 untuk data training dan testing serta melakukan pengujian untuk pembagian data 80:20 untuk data training dan testing.



Gambar 1 K-Fold Cross Validation Konsep



Gambar 2 Pemilihan Data 90:10



Gambar 3 Pemilihan Data 80:20

Untuk data 90:10, Data terpilih pada 7 Fold, menunjukkan perbedaan data antara 2 kelas lebih baik. Sehingga dipilih untuk data Training Berada Pada Index 0 – 461 dan 539 – 767, sedangkan untuk data Uji berada pada index 462 – 538. Sedangkan untuk data 80:20, data terpilih pada 4 Fold. Pada *index* 0 – 461 dan 616 – 767 akan digunakan sebagai data training sedangkan *index* 462 – 615 akan digunakan sebagai data uji.

#### 2.4 Support Vector Machine

Pada Pengujian Pertama digunakan *Support Vector Machine* (SVM). SVM merupakan Algoritma *Machine Learning* berbasis *kernel* yang paling banyak digunakan khususnya dalam klasifikasi. SVM, pada dasarnya merupakan pengklasifikasi *biner linier*, yang mengidentifikasi batas tunggal antara dua kelas[13]. *Support Vector Machine* dalam pengujian ini sebagai bahan referensi hasil yang diperoleh dari *traditional machine learning*. SVM dipilih karena memiliki basis yang sama dengan DNN yaitu menggunakan bobot.

Hasil dari seleksi data pada proses sebelumnya, dilanjutkan ke proses pengujian SVM. SVM yang digunakan adalah SVM yang disediakan oleh python. Perlakuan yang diberikan pada percobaan SVM ini dengan melihat pengaruh hyperparameter pada kernel. Dalam Pengujian ini dilakukan 5 Pemodelan Kernel yang ada dalam SVM. Pemodelan yang digunakan adalah standart (Tanpa Kernel) dan juga menggunakan 4 pemodelan berbasis kernel.

Untuk pemodelan SVM Standart diformulasikan dengan formula sebagai berikut[14] :

$$y_i (w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_k \cdot x_k) + b \geq 1 \quad (3)$$

Untuk pemodelan SVM berbasis kernel dirumuskan sebagai berikut :

*Kernel Linear*

$$K(x_i, x_j) = x_i \cdot x_j \quad (4)$$

*Kernel Radial Basis Function (RBF)*

$$K_{RBF}(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|}{\sigma^2}\right) \quad (5)$$

*Kernel Polynomial*

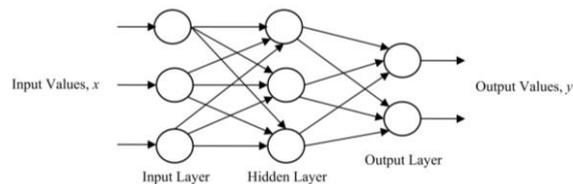
$$K_{ploy}(x_i, x_j) = (x_i^T x_j + 1)^d \quad (6)$$

*Kernel Sigmoid*

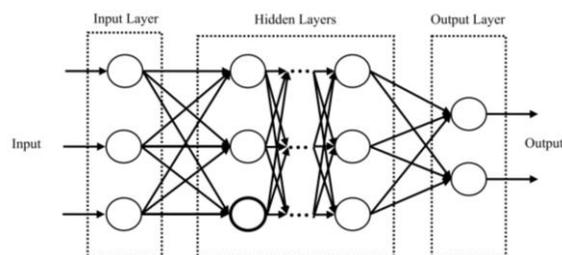
$$K_{sigmoid}(x_i, x_j) = \tanh(x_i^T x_j + 1) \quad (7)$$

### 2.5 Deep Neural Network

Pada Pengujian Kedua digunakan *Deep Neural Network* (DNN). Penemu neurokomputer pertama Dr. Robert Hecht Nielsen, memperkenalkan Neural Network atau yang di kenal dengan *artificial neural network* (ANN) untuk Pertama kalinya. Dr. Robert Hecht Nielsen, mendefinisikan sebagai “*a computing system made up of a number of simple, highly interconnected processing elements which process information by heir dynamic state response to external inputs*”[15]. *Deep neural network* (DNN) adalah perpanjangan dari ANN. DNN akan selalu terdiri dari lapisan input, beberapa lapisan tersembunyi, dan lapisan *output*. Setiap lapisan terhubung melalui *node* dimana setiap lapisan tersembunyi memberikan hasil prediksi berdasarkan prediksi dari lapisan sebelumnya. Perbedaan utama antara NN dan DNN adalah NN memiliki satu lapisan tersembunyi sedangkan DNN memiliki dua atau lebih dari dua lapisan tersembunyi[15].



Gambar 4 Arsitektur Neural Network



Gambar 5 Arsitektur Deep Neural Network

Pada pengujian DNN ini akan dilakukan pengujian dengan menggunakan 1 *layer input*, 2 *hidden layer* dan 1 *output layer*. Jumlah *neuron* pada *input layer* berjumlah 8 menyesuaikan dengan jumlah fitur yang ada. Untuk jumlah *Output Layer* berjumlah 2 menyesuaikan jumlah

kelas untuk melakukan klasifikasi. Sedangkan untuk *hidden layer* akan dilakukan pengujian dengan jumlah *neuron* pada *hidden layer* 1 adalah 8 sampai 10 dan juga jumlah *neuron* pada *hidden layer* 2 adalah 8 sampai 10.

*Deep Neural Network* diformulasikan dengan bobot ( $w$ ) dikali dengan nilai input ( $x$ ). Kemudian ditambah dengan nilai bias ( $b$ ). Nilai  $w$  dan  $b$  diperoleh dari nilai random.

$$f(x) = (w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_k \cdot x_k) + b \quad (8)$$

Dalam pengujian ini, digunakan 2 fungsi aktivasi. Pertama adalah fungsi aktivasi *sigmoid* pada setiap *hidden layer* dan fungsi aktivasi *softmax* pada *output layer*.

Fungsi Aktivasi *Sigmoid* :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

Fungsi Aktivasi *Softmax* :

$$f(x) = \frac{\exp(x_i)}{\sum \exp(x_j)} \quad (10)$$

## 2.6 Particle Swarm Optimize

Untuk menentukan bobot yang sesuai digunakan algoritma Particle Swarm Optimization (PSO). PSO adalah salah satu algoritma heuristik yang dianggap sebagai algoritma komputasi evolusioner, yang memiliki karakteristik kesederhanaan, efisiensi tinggi dan konvergensi yang cepat[16].

Algoritma *Particle Swarm Optimization* (PSO) adalah algoritma *swarm intelligence* yang khas, yang berasal dari pengembangan kolaborasi kelompok berdasarkan algoritma pencarian stokastik untuk perilaku mencari makan kawanan burung. Ide intinya adalah, dalam ruang solusi, untuk menggunakan pertukaran informasi antara individu dalam populasi untuk membuat gerak seluruh populasi berevolusi dari ketidakteraturan menjadi keteraturan, yaitu untuk menemukan solusi optimal. Setiap partikel dalam populasi mewakili solusi potensial untuk masalah optimisasi (fungsi kebugaran)[17].

Ini mensimulasikan perilaku predator burung di alam. Perilaku setiap individu dalam kawanan dipengaruhi oleh pengalamannya sendiri dan pengalaman tetangganya. Algoritma melalui komunikasi informasi dan umpan balik positif antara individu yang tidak cerdas, algoritma mendapatkan solusi optimal dari masalah setelah beberapa kali iterasi.[16]

Bobot inersia ditentukan dengan nilai  $w=0.789$  dan bobot kognitif ( $c_1$ ), dan bobot sosial ( $c_2$ ) masing-masing 1.49445 dan 1.49445[18]. Pada tahap ini akan dilakukan pengujian dengan jumlah partikel adalah 10 dan jumlah iterasi berkisar dari 0 sampai 500, kemudian pada setiap iterasi akan dilakukan pengecekan akurasi sesuai dengan bobot terbaik. Hasil akurasi dari setiap bobot terbaik akan menjadi referensi untuk melihat tingkat keberhasilan optimalisasi DNN. Entropy akan digunakan untuk menentukan referensi bobot terbaik dalam PSO yang digunakan.

$$v_i^{t+1} = wv_i^t + c_1r_1(p_i^t - x_i^t) + c_2r_2(g^t - x_i^t) \quad (11)$$

$$x_i^{t+1} = v_i^{t+1} + x_i^t \quad (12)$$

### 2.7 Evaluasi

Untuk melakukan evaluasi kinerja dari hasil pengujian, dilakukan pemeriksaan akurasi, *presis* dan *recall*. *True positive* (TP) didefinisikan sebagai jumlah gambar yang diklasifikasikan dengan benar; *false positive* (FP) sebagai nilai negatif tetapi model memperkirakannya sebagai positif; *true negative* (TN) sebagai jumlah gambar yang diidentifikasi dengan benar sebagai negatif; dan *false negative* (FN) sebagai positif tetapi model memperkirakannya sebagai negatif.

Akurasi menunjukkan proporsi klasifikasi yang diidentifikasi dengan benar dan dihitung sebagai berikut[19]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (13)$$

Presisi didefinisikan sebagai ukuran seberapa sering label positif diberikan ke kategori yang benar dan dihitung sebagai berikut[19]:

$$Precision = \frac{(TP)}{(TP + FP)} \quad (14)$$

*Recall* didefinisikan sebagai pengukuran setiap klasifikasi yang diklasifikasikan dengan benar dan dihitung sebagai berikut[19]:

$$Recall = \frac{(TP)}{(TP + FN)} \quad (15)$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Model SVM

Pengujian pertama dilakukan dengan menggunakan SVM yang merupakan salah satu algoritma *traditional machine learning*. Pemilihan SVM didasari karena SVM dan DNN memiliki konsep yang sama yaitu dipengaruhi oleh bobot. Pengujian pertama ini dilakukan dengan menggunakan memperlakukan SVM dengan Hyperparameter pada parameter kernel. Parameter kernel yang diuji adalah tanpa kernel (No Kernel), Kernel Linear, Kernel Polynomial, Kernel RBF dan Kernel RBF. Kernel dipilih sebagai penentu hyperparameter karena SVM merupakan salah satu algoritma berbasis kernel. Model SVM tanpa kernel, Kernel Polynomial dan Kernel RBF memperoleh akurasi sebesar 80,5% pada pembagian data 90:10, yang menunjukkan kemampuan yang cukup baik dalam membedakan kasus diabetes dan bukan diabetes. Penggunaan kernel linear sedikit meningkatkan akurasi menjadi 84.4% pada data 90:10 dan akurasi 83% pada data 80:20 sehingga menunjukkan bahwa kernel linear memberikan hasil cukup efektif dalam skenario ini Hasil Evaluasi keseluruhan pada Model SVM dengan melihat Akurasi (Acc), *Presision* (Pre) dan *Recall* (Rec) pada perbandingan data Training dan Testing, 90:10 dan 80:20 disajikan pada Tabel 2.

Tabel 2 Pengujian Model SVM

Kernel Type	Data 90:10			Data 80:20		
	Acc	Pre	Rec	Acc	Pre	Rec
No Kernel	0.805	0.875	0.889	0.823	0.884	0.877
Kernel Linear	0.844	0.918	0.889	0.830	0.879	0.895
Kernel Polynomial	0.805	0.875	0.889	0.810	0.824	0.947
Kernel RBF	0.805	0.875	0.889	0.824	0.885	0.877
Kernel Sigmoid	0.481	0.756	0.540	0.719	0.859	0.746

### 3.2 Model DNN

Pengujian kedua dilakukan dengan menggunakan algoritma DNN. Pada pengujian ini DNN diuji tanpa menggunakan optimasi. Pengujian DNN hanya didasari dengan penggunaan *Hidden Layer*. Pengujian dibatasi hanya menggunakan 2 Hidden Layer, masing-masing Hidden Layer diberi neuron 8 sampai 10. Hasil pengujian dengan menggunakan 8 neuron pada Hidden Layer 1 dan Hidden Layer menunjukkan akurasi tertinggi pada data 90:10 maupun data 80:20 yang masing-masing memiliki nilai akurasi 82% dan 75%. Ketika jumlah neuron pada Hidden Layer 1 diganti menjadi 9 neuron dan Hidden Layer 2 diganti menjadi 10 neuron, akurasi menurun drastis menjadi 18% untuk data 90:10 dan menjadi 25% untuk data 80:20. Hasil Evaluasi keseluruhan pada Model DNN tanpa dilakukan optimasi dengan melihat Akurasi (A), *Presision* (P) dan *Recall* (R) pada perbandingan data *Training* dan *Testing*, 90:10 dan 80:20. Pada Pengujian ini diujikan jumlah neuron 8 sampai 10 pada *hidden layer* 1 (HL1) dan jumlah *neuron* 8 sampai 10 pada *hidden layer* 2 (HL2).

Tabel 3 Pengujian DNN Original

HL1	HL2	Data 90:10			Data 80:20		
		A	P	R	A	P	R
8	8	0.82	0	0	0.75	0	0
9	9	0.18	0.18	1	0.25	0.25	1
9	10	0.18	0.18	1	0.25	0.25	1
10	9	0.82	0	0	0.75	0	0
10	10	0.18	0.18	1	0.25	0.25	1

### 3.3 Model DNN-PSO

Pada Pengujian Ketiga dilakukan percobaan dengan mengoptimalkan algoritma DNN dengan menggunakan PSO. Jumlah Hidden Layer dan Neuron disetiap hidden layer diberi perlakuan yang sama dengan DNN tanpa optimalisasi. PSO akan dilakukan sebanyak 500 iterasi pada setiap percobaan (pada setiap pengujian Hidden Layer). Mulanya DNN akan menentukan bobot awal secara random untuk dikalikan dengan setiap feature. Hasil perhitungan dengan menggunakan bobot awal akan dicek akurasinya. Bobot terbaik akan ditentukan dengan menggunakan formula Entropy pada setiap iterasi PSO. Kemudian PSO akan mencari bobot random lainnya. Proses perkalian bobot baru dan feature akan terus berlangsung secara terus menerus sesuai dengan jumlah iterasi pada PSO. Proses ini memberikan komputasi yang lama tergantung jumlah hidden layer, jumlah iterasi. Hasil pengujian model Deep Neural Network (DNN) yang dioptimalkan menggunakan algoritma Particle Swarm Optimization (PSO) untuk prediksi diabetes menunjukkan sejumlah pola menarik. Pada konfigurasi Hidden Layer 1 dengan 8 neuron dan Hidden Layer 2 dengan 8 neuron pada iterasi ke 2 memberikan akurasi sebesar 65% pada data 80:20. Pada konfigurasi dengan 2 hidden layer yang memiliki 9 dan 8 neuron pada iterasi ke 166, model mencapai tingkat akurasi sebesar 87%, menunjukkan kinerja yang sangat baik dalam mengidentifikasi kasus diabetes. Hasil Evaluasi secara keseluruhan pada Model DNN digabungkan dengan PSO dengan melihat Akurasi (A), *Presision* (P) dan *Recall* (R) pada perbandingan data *Training* dan *Testing*, 90:10 dan 80:20. Pada Pengujian ini diujikan jumlah neuron 8 sampai 10 pada *hidden layer* 1 (HL1) dan jumlah *neuron* 8 sampai 10 pada *hidden layer* 2 (HL2) dengan jumlah iterasi/epoch 1 sampai 500 pada algoritma PSO (ITR) yang disajikan pada Tabel 4

Tabel 4 Pengujian DNN Dan PSO

			Data 90:10			Data 80:20		
HL1	HL2	ITR	A	P	R	A	P	R
8	8	1	0.62	0.17	0.29	0.64	0.3	0.31
8	8	2	0.64	0.16	0.21	0.65	0.29	0.26
9	8	166	0.87	0.75	0.43	0.75	0.51	0.51
10	9	496	0.80	0.47	0.64	0.83	0.65	0.72
10	10	499	0.67	0.30	0.57	0.76	0.52	0.72
10	10	500	0.67	0.30	0.57	0.76	0.52	0.72

#### 4. KESIMPULAN DAN SARAN

Dalam penelitian ini, kami menyajikan pengujian dengan menggunakan SVM dan DNN dengan PSO pada 768 records pada PIMA dataset. Pemilihan Data Latih dilakukan dengan menggunakan *Non-Random Sampling* dengan membagi 90% data *training* dan 10% data *testing* serta membagi juga dengan 80% data *training* dan 20% data uji. Pengujian ini dievaluasi dengan menggunakan Akurasi, *Presisi* dan *Recall*. Penelitian ini berhasil menunjukkan peningkatan dalam pemodelan prediksi diabetes. Integrasi Particle Swarm Optimization (PSO) dengan Deep Neural Network (DNN) telah menghasilkan hasil yang lebih baik dan lebih akurat dalam memprediksi diabetes dibandingkan dengan algoritma DNN tanpa optimasi dan algoritma *traditional machine learning* (SVM) serta menunjukkan peningkatan dari penelitian sebelumnya yang menggunakan dataset yang sama[4], [5]. Hal ini mengindikasikan bahwa kombinasi DNN dan PSO dapat menjadi metode yang efektif untuk meningkatkan kinerja prediksi diabetes. Peningkatan ini dapat memiliki implikasi positif dalam pemahaman dan manajemen penyakit diabetes, serta dapat digunakan dalam aplikasi medis dan klinis. Hasil akurasi DNN menghasilkan akurasi tertinggi sebesar 82% sedangkan PSO-DNN menghasilkan akurasi tertinggi sebesar 87% pada jumlah neuron *hidden layer* 1 sebanyak 9 dan *hidden layer* 2 sebanyak 8 pada perbandingan data 90% data *training* dan 10% data *testing*.

#### DAFTAR PUSTAKA

- [1] S. C. Gupta and N. Goel, "Predictive Modeling and Analytics for Diabetes using Hyperparameter tuned Machine Learning Techniques," *Procedia Comput Sci*, vol. 218, pp. 1257–1269, 2023, doi: 10.1016/j.procs.2023.01.104.
- [2] "Diabetes." Accessed: May 30, 2023. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/diabetes>
- [3] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
- [4] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," *Procedia Comput Sci*, vol. 216, pp. 21–30, 2023, doi: 10.1016/j.procs.2022.12.107.
- [5] N. P. Tigga and S. Garg, "Prediction of Type 2 Diabetes using Machine Learning Classification Methods," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 706–716. doi: 10.1016/j.procs.2020.03.336.
- [6] A. Azrar, M. Awais, Y. Ali, and K. Zaheer, "Data Mining Models Comparison for Diabetes Prediction," 2018. [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)

- [7] D. Sisodia and D. S. Sisodia, "Prediction of Diabetes using Classification Algorithms," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 1578–1585. doi: 10.1016/j.procs.2018.05.122.
- [8] V. Chang, M. A. Ganatra, K. Hall, L. Golightly, and Q. A. Xu, "An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators," *Healthcare Analytics*, vol. 2, Nov. 2022, doi: 10.1016/j.health.2022.100118.
- [9] G. Swapna, R. Vinayakumar, and K. P. Soman, "Diabetes detection using deep learning algorithms," *ICT Express*, vol. 4, no. 4, pp. 243–246, Dec. 2018, doi: 10.1016/j.icte.2018.10.005.
- [10] W. Shi, D. Liu, X. Cheng, Y. Li, and Y. Zhao, "Particle Swarm Optimization-Based Deep Neural Network for Digital Modulation Recognition," *IEEE Access*, vol. 7, pp. 104591–104600, 2019, doi: 10.1109/ACCESS.2019.2932266.
- [11] F. Ismail and I. I. Lawanda, "IMPLEMENTASI EDMS DALAM PENATAAN DOKUMEN DI RAIL DOCUMENT SYSTEM PT. KERETA API INDONESIA (PERSERO) DAERAH OPERASI 1 JAKARTA," *BACA: JURNAL DOKUMENTASI DAN INFORMASI*, vol. 41, no. 2, p. 143, Dec. 2020, doi: 10.14203/j.baca.v41i2.563.
- [12] N. Li, F. He, W. Ma, R. Wang, and X. Zhang, "Wind Power Prediction of Kernel Extreme Learning Machine Based on Differential Evolution Algorithm and Cross Validation Algorithm," *IEEE Access*, vol. 8, pp. 68874–68882, 2020, doi: 10.1109/ACCESS.2020.2985381.
- [13] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13. Institute of Electrical and Electronics Engineers Inc., pp. 6308–6325, 2020. doi: 10.1109/JSTARS.2020.3026724.
- [14] H. Zhao, Y. Gao, H. Liu, and L. Li, "Fault diagnosis of wind turbine bearing based on stochastic subspace identification and multi-kernel support vector machine," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 350–356, Mar. 2019, doi: 10.1007/s40565-018-0402-8.
- [15] R. Thanki, "A deep neural network and machine learning approach for retinal fundus image classification," *Healthcare Analytics*, vol. 3, Nov. 2023, doi: 10.1016/j.health.2023.100140.
- [16] H. Cao, F. Wang, M. Deng, X. Wang, and Y. Chu, "Software multiple-fault localization using particle swarm optimization via genetic operation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 4, pp. 21–35, Apr. 2023, doi: 10.1016/j.jksuci.2023.02.023.
- [17] L.-H. Ye, S.-J. Chen, Y.-F. Shi, D.-H. Peng, and A.-P. Shi, "Remaining useful life prediction of lithium-ion battery based on chaotic particle swarm optimization and particle filter," *Int J Electrochem Sci*, vol. 18, no. 5, p. 100122, May 2023, doi: 10.1016/j.ijoes.2023.100122.
- [18] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation, CEC 2000*, IEEE Computer Society, 2000, pp. 84–88. doi: 10.1109/CEC.2000.870279.
- [19] I. Kanjanasurat, K. Tenghongsakul, B. Purahong, and A. Lasakul, "CNN–RNN Network Integration for the Diagnosis of COVID-19 Using Chest X-ray and CT Images," *Sensors*, vol. 23, no. 3, Feb. 2023, doi: 10.3390/s23031356.