

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002024190641, 21 September 2024

Pencipta

Nama : **Rudi Setiawan, Lestari Agusalim dkk**
Alamat : Pura Bojonggede Blok A1 No.11, Tajurhalang, Bogor, Jawa Barat, 16320
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Rudi Setiawan, Lestari Agusalim dkk**
Alamat : Pura Bojonggede Blok A1 No.11, Tajurhalang, Bogor, Jawa Barat, 16320
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Model Deteksi Penyakit Diabetes Pada Ibu Hamil Menggunakan Metode Hybrid Machine Learning**

Tanggal dan tempat diumumkan untuk pertama kali : 21 September 2024, di Jakarta
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000763117

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

IGNATIUS M.T. SILALAH
NIP. 196812301996031001

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Rudi Setiawan	Pura Bojonggede Blok A1 No.11, Tajurhalang, Bogor
2	Lestari Agusalim	Jl. TMP. Kalibata No. 1, Pancoran, Jakarta Selatan, Pancoran, Jakarta Selatan
3	Gatot Tri Pranoto	Pondok Sukmajaya Permai, Blok BB/23, Sukmajaya, Depok
4	Rahayu Khairiah	Jl. Swadaya No.7, RT.001/RW.014, Jatibening, Kec. Pd. Gede, Pondok Gede, Bekasi
5	Lisa Handayani	Tulung Buyut, Kabupaten Way Kanan, Blambangan Umpu, Way Kanan
6	Royandi Novantara	Jl. TMP. Kalibata No.1, Pancoran, Pancoran, Jakarta Selatan
7	Izzaturamadhan	Jl. TMP. Kalibata No. 1, Pancoran, Jakarta Selatan, Pancoran, Jakarta Selatan

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Rudi Setiawan	Pura Bojonggede Blok A1 No.11, Tajurhalang, Bogor
2	Lestari Agusalim	Jl. TMP. Kalibata No. 1, Pancoran, Jakarta Selatan, Pancoran, Jakarta Selatan
3	Gatot Tri Pranoto	Pondok Sukmajaya Permai, Blok BB/23, Sukmajaya, Depok
4	Rahayu Khairiah	Jl. Swadaya No.7, RT.001/RW.014, Jatibening, Kec. Pd. Gede, Pondok Gede, Bekasi
5	Lisa Handayani	Tulung Buyut, Kabupaten Way Kanan, Blambangan Umpu, Way Kanan
6	Royandi Novantara	Jl. TMP. Kalibata No.1, Pancoran, Pancoran, Jakarta Selatan
7	Izzaturamadhan	Jl. TMP. Kalibata No. 1, Pancoran, Jakarta Selatan, Pancoran, Jakarta Selatan





MODEL DETEKSI PENYAKIT DIABETES PADA IBU HAMIL MENGUNAKAN METODE HYBRID MACHINE LEARNING

Abstract

Pengembangan model meliputi kegiatan penerapan metode machine learning untuk mendeteksi dini resiko diabetes pada ibu hamil menggunakan algoritma Support Vector Machine dengan optimisasi parameter menggunakan algoritma Walrus Optimization. Pada tahap preprocessing dilakukan penanganan imbalanced dataset menggunakan metode SMOTE. Dengan kombinasi ini, model hybrid mampu menangani kompleksitas data.

Diusulkan oleh

Rudi Setiawan
Lestari Agusalim
Gatot Tri Pranoto
Rahayu Khairiah
Lisa Handayani
Royandi Novantara
Izzaturamadhan

DAFTAR ISI

A. Deskripsi Model.....	2
B. Dataset Diabetes.....	2
C. Pengembangan Model.....	2

A. Deskripsi Model

Pengembangan model meliputi kegiatan penerapan metode machine learning untuk mendeteksi dini resiko diabetes pada ibu hamil menggunakan algoritma Support Vector Machine dengan optimisasi parameter menggunakan algoritma Walrus Optimization. Pada tahap preprocessing dilakukan penanganan imbalanced dataset menggunakan metode SMOTE. Dengan kombinasi ini, model hybrid mampu menangani kompleksitas data.

B. Dataset Diabetes

Dataset diabetes yang digunakan bersumber dari data pemeriksaan ibu hamil yang ada di Unit Pelayanan Teknis Pusat Kesehatan Masyarakat (UPT Puskesmas) Gunung Labuhan, Kabupaten Way Kanan, Provinsi Lampung. Jumlah data sebanyak 158 yang terdiri dari fitur serta label data sebagai berikut:

Nama	Keterangan	Nilai
Jumlah kehamilan	Jumlah kehamilan yang pernah dialami	0 - 4
Glukosa	Konsentrasi glukosa plasma pada 2 jam dalam tes toleransi glukosa	0 - 197
Tekanan darah	Ukuran tekanan darah	0 - 122
Body mass index (BMI)	Indeks Massa Tubuh	0 - 55
Diabetes pedigree function	Indikator riwayat diabetes dalam keluarga	0.085 – 2.288
Usia	Usia saat hamil	21-49
Kategori	Class / Kategori (0 = tidak terindikasi diabet dan 1 = terindikasi diabetes)	0, 1

C. Pengembangan Model

Pengembangan model dilakukan menggunakan bahasa pemrograman Python dan melibatkan beberapa library yang dibutuhkan dalam pemodelan.

1. Install library Mealpy yang dibutuhkan untuk melakukan optimasi parameter pada pemodelan

```
pip install git+https://github.com/thieu1995/mealpy
```

2. Import library yang diperlukan untuk melakukan data preprocessing dan pemodelan

```
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import datasets, metrics
from mealpy import FloatVar, StringVar, Problem, WaOA, MixedSetVar
import pandas as pd
```

- Masukkan dataset dan tentukan variable independent (features) dan dependent (target), masing-masing simpan pada variabel X dan y

```
data = pd.read_csv("diabetes_gunung_labuhan.csv")

X = data.drop(Kategori, axis=1)
y = data[Kategori]
```

- Lakukan one-hot encoding pada variable dependent (target) yang semula dalam bentuk categorical menjadi binary vector

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

- Terapkan teknik SMOTE untuk menangani permasalahan imbalanced dataset

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='auto', k_neighbors=2)
X_augmented, y_augmented = smote.fit_resample(X, y)
```

- Split dataset menjadi data training dan data testing

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_augmented,
y_augmented, test_size=0.2, random_state=42, stratify=y_augmented)
```

- Lakukan standarisasi fitur dengan menghilangkan mean dan skala

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

- Buat Class untuk melakukan prediksi dengan algoritma SVC dengan nilai kembalian pada fungsi objective function berupa nilai accuracy dari hasil prediksi.

```
class SvmOptimizedProblem(Problem):
    def __init__(self, bounds=None, minmax="min", data=None, **kwargs):
        self.data = data
        super().__init__(bounds, minmax, **kwargs)

    def obj_func(self, x):
        x_decoded = self.decode_solution(x)
        C_paras, kernel_paras, gamma = x_decoded["C_paras"],
x_decoded["kernel_paras"], x_decoded["gamma_paras"]

        X_train_std, X_test_std, y_train, y_test = self.data

        svc = SVC(C=C_paras, kernel=kernel_paras, gamma=gamma,
random_state=1)
```

```

svc.fit(X_train_std, y_train)

y_predict = svc.predict(X_test_std)

print(classification_report(y_test,y_predict))
print(confusion_matrix(y_test,y_predict))

return metrics.accuracy_score(y_test, y_predict)

```

9. Masukkan data hasil standardize kedalam variable data

```
data = [X_train_scaled, X_test_scaled, y_train, y_test]
```

10. Tentukan batasan / seting nilai parameter yang dibutuhkan oleh SVM

```

my_bounds = [
    FloatVar(lb=0.01, ub=1000., name="C_paras"),
    StringVar(valid_sets=('linear', 'poly', 'rbf', 'sigmoid'),
              name="kernel_paras"),
    MixedSetVar(valid_sets=('scale', 'auto', 0.01, 0.05, 0.1, 0.5, 1.0),
                name="gamma_paras")
]

```

11. Menyiapkan variable "problem" untuk menampung nilai kembalian dari hasil pemanggilan class SvmOptimizedProblem

```
problem = SvmOptimizedProblem(bounds=my_bounds, minmax="max", data=data)
```

12. Menyiapkan variable model untuk menampung class dari algoritma walrus optimization beserta nilai parameter yang telah ditentukan

```

model = WaOA.OriginalWaOA(epoch=50, pop_size=50)
model.solve(problem)

```

13. Cetak nilai terbaik dari agent, solusi, acuraccy dan parameter terbaik yang didapat dari hasil pemodelan

```

print(f"Best agent: {model.g_best}")
print(f"Best solution: {model.g_best.solution}")
print(f"Best accuracy: {model.g_best.target.fitness}")
print(f"Best parameters:
{model.problem.decode_solution(model.g_best.solution)}")

```

14. Simpan histroy dari perolehan nilai global objective, local objective, global best fitness, local best fitness, run time dan persentase explorasi dan exploitasi hasil pemodelan kedalam file gambar berupa grafik.

```

model.history.save_global_objectives_chart(filename="hasil/goc")
model.history.save_local_objectives_chart(filename=" hasil /loc")
model.history.save_global_best_fitness_chart(filename=" hasil /gbfc")
model.history.save_local_best_fitness_chart(filename=" hasil /lbfc")
model.history.save_runtime_chart(filename=" hasil /rtc")

```

```
model.history.save_exploration_exploitation_chart(filename=" hasil  
/eec")  
model.history.save_diversity_chart(filename="hasil /dc")
```