

Perbandingan Model CNN, LSTM, dan FNN dalam Klasifikasi Kulit Penderita Diabetes

Samsul Arifin^{*1}, Imam Tahyudin², Fandy Setyo Utomo³

^{1,2,3}Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Indonesia

Email: ¹samsulchin21@gmail.com, ²imam.tahyudin@amikompurwokerto.ac.id,

³fandy_setyo_utomo@amikompurwokerto.ac.id

Abstrak

Penelitian ini mengembangkan dan membandingkan kinerja tiga algoritma Deep Learning, yaitu Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), dan Feedforward Neural Networks (FNN), untuk mengklasifikasi gambar kulit penderita diabetes dan kulit sehat. Data yang digunakan terdiri dari gambar kulit yang diproses melalui tahapan pra-pemrosesan, pembangunan model, pelatihan, dan evaluasi. Parameter yang diuji meliputi akurasi klasifikasi masing-masing model. Hasil menunjukkan bahwa LSTM mencapai akurasi tertinggi sebesar 94%, diikuti oleh CNN dengan 87%, dan FNN dengan 82%. Model terbaik diimplementasikan dalam aplikasi berbasis web menggunakan Flask, yang dapat memberikan prediksi otomatis untuk mendukung diagnosis dini. Penelitian ini berkontribusi pada pengembangan teknologi diagnostik yang dapat membantu mencegah komplikasi serius pada pasien diabetes melalui deteksi dini kondisi kulit.

Kata kunci: CNN, deep learning, FNN, klasifikasi kulit, LSTM

Comparison of CNN, LSTM, and FNN Models in Diabetes Skin Classification

Abstract

This research develops and compares the performance of three Deep Learning algorithms, namely Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Feedforward Neural Networks (FNN), to classify diabetic and healthy skin images. The data used consists of skin images processed through pre-processing, model building, training, and evaluation stages. The parameters tested include the classification accuracy of each model. The results show that LSTM achieved the highest accuracy of 94%, followed by CNN with 87%, and FNN with 82%. The best model was implemented in a web-based application using Flask, which can provide automatic predictions to support early diagnosis. This research contributes to the development of diagnostic technologies that can help prevent serious complications in diabetic patients through early detection of skin conditions.

Keywords: CNN, deep learning, FNN, LSTM, skin classification

1. PENDAHULUAN

Diabetes mellitus adalah salah satu penyakit metabolik yang terus meningkat secara global, dengan prevalensi mencapai lebih dari 460 juta orang di seluruh dunia pada tahun 2019, dan diperkirakan akan meningkat menjadi lebih dari 700 juta pada tahun 2045 (*World Health Organization, 2020*). Salah satu komplikasi yang sering dialami oleh penderita diabetes adalah gangguan kulit, seperti ulkus kulit diabetes, yang dapat menyebabkan infeksi dan amputasi apabila tidak terdeteksi dan ditangani dengan baik. Penanganan dini terhadap perubahan kulit akibat diabetes sangat penting untuk mencegah komplikasi lebih lanjut. Menurut [1] ulkus diabetik merupakan penyebab utama amputasi pada penderita diabetes, dengan angka kejadian yang terus meningkat setiap tahunnya. Oleh karena itu, pengembangan sistem deteksi otomatis untuk mengklasifikasi kulit penderita diabetes menjadi suatu kebutuhan yang mendesak.

Dalam beberapa tahun terakhir, kemajuan teknologi dalam bidang *Deep Learning*, khususnya dalam pengolahan citra medis, telah membuka peluang besar untuk mendeteksi dan mengklasifikasikan kondisi medis secara lebih cepat dan akurat. Pendekatan *Deep Learning*, seperti *Convolutional Neural Networks (CNN)*, *Long Short-Term Memory (LSTM)*, dan *Feedforward Neural Networks (FNN)*, telah terbukti efektif dalam tugas klasifikasi citra medis. CNN, sebagai salah satu metode yang paling populer, mampu mengekstraksi fitur penting dalam citra medis dengan efisiensi tinggi [2]. Sementara itu, LSTM, yang lebih dikenal analisis dalam urutan data, menunjukkan potensi besar dalam pemrosesan citra berurutan untuk meningkatkan akurasi deteksi [3]. Di sisi lain,

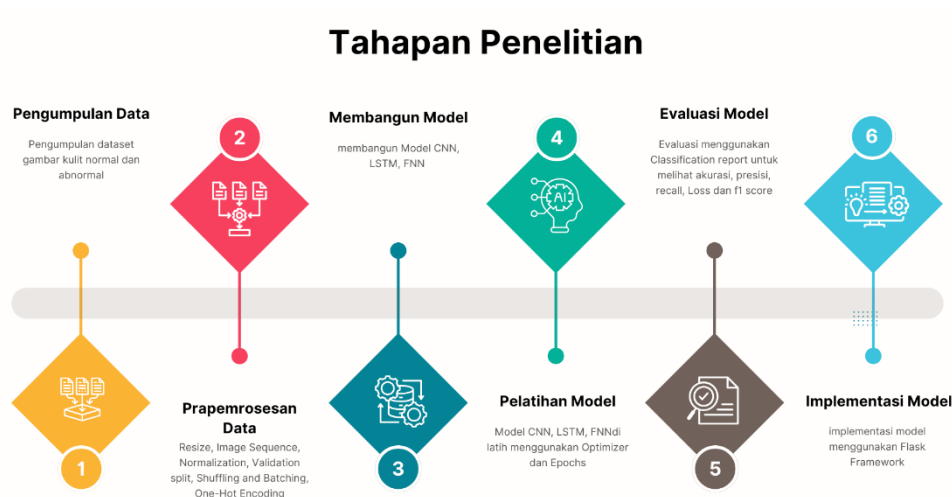
FNN, meskipun lebih sederhana, dapat digunakan untuk mengidentifikasi hubungan non-linier antar fitur dalam citra medis [4].

Namun, meskipun berbagai pendekatan telah digunakan dalam pengklasifikasian citra medis, penelitian yang secara khusus fokus pada klasifikasi kondisi kulit diabetes masih terbatas. Sebagian besar penelitian sebelumnya hanya mengeksplorasi penggunaan *Deep Learning* untuk mendeteksi kanker kulit [5] maupun retinopati diabetik [6]. Namun, meskipun ada kemajuan yang signifikan, masih terdapat tantangan dalam mengembangkan model yang dapat secara efektif membedakan antara kondisi kulit diabetes dan kulit normal pada tingkat akurasi yang tinggi. Selain itu Penelitian-penelitian sebelumnya menggunakan dataset terbatas atau teknik klasifikasi yang hanya fokus pada satu model *Deep Learning*, seperti CNN, tanpa mempertimbangkan kombinasi model lain yang dapat meningkatkan akurasi prediksi [7].

Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan membandingkan tiga algoritma Deep Learning—CNN, LSTM, dan FNN—dalam mengklasifikasi citra kulit penderita diabetes dan kulit normal. Pendekatan ini tidak hanya memberikan wawasan baru tentang keunggulan relatif masing-masing model, tetapi juga mengintegrasikan hasil terbaik ke dalam sistem deteksi berbasis web yang dapat digunakan untuk diagnosis dini dalam praktik medis [8]. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan teknologi diagnostik yang lebih akurat dan aplikatif dalam pengelolaan komplikasi diabetes.

2. METODE PENELITIAN

Metode penelitian ini di rancang dalam beberapa tahapan utama, yaitu: Pengumpulan Data, Pra-Pemrosesan Data, Pelatihan Model, Evaluasi Model, dan Implementasi Model [9]. Tahapan penelitian ini dapat digambarkan pada gambar 1 sebagai berikut :



Gambar 1. Tahapan Penelitian

2.1. Pengumpulan Data

Pengumpulan data adalah langkah pertama dalam penelitian ini. Pada penelitian ini data yang digunakan diambil dari sumber dataset terbuka dari website kaggle <https://www.kaggle.com/datasets/laithjj/diabetic-foot-ulcer-dfu>. Dataset tersebut memiliki 2 jenis gambar citra yaitu luka kulit diabetes dan kulit sehat. Data tersebut disusun dalam folder 2 folder utama *Validation* dan *Training*, dengan setiap folder memiliki 2 kelas gambar yang disimpan dalam direktori terpisah, seperti *Abnormal(Ulcer)* dan *Normal(Healthy skin)*. Dataset ini berisi gambar dengan format seperti .jpg atau .png yang perlu diproses lebih lanjut agar dapat digunakan oleh model.

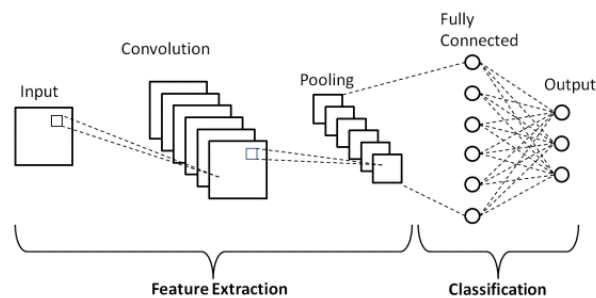
2.2. Pra Pemrosesan Data

Setelah data terkumpul, tahapan selanjutnya adalah pra pemrosesan data, yang bertujuan untuk mempersiapkan data agar dapat digunakan oleh model [10]. Pertama, gambar diubah ukurannya ke dimensi yang sesuai dengan model yang digunakan (128x128). Selanjutnya, piksel gambar dinormalisasi dengan membagi setiap nilai piksel dengan 255.0, sehingga gambar berada dalam rentang [0, 1], yang mempercepat pelatihan model. Data kemudian dibagi menjadi dua set, yaitu set pelatihan (80%) dan set validasi (20%) menggunakan teknik *validation*.

split. Selain itu, data diacak dan dibagi menjadi *batch* (32 gambar per *batch*) untuk efisiensi pelatihan dan untuk menghindari urutan yang terulang. Label kelas diubah menjadi *one-hot encoding* untuk klasifikasi *Multi-Class*, dimana setiap kelas diberi representasi biner [1, 0] untuk *Abnormal* dan [0, 1] untuk *Normal*).

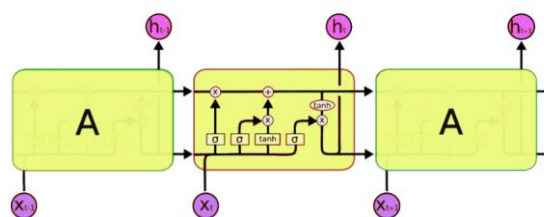
2.3. Pembangunan Model

Setelah tahap *Preprocessing* selesai tahap selanjutnya adalah pembangunan model. Tiga algoritma *Deep Learning* (*CNN*, *LSTM*, *FNN*) di pilih karena mempunyai berbagai kelebihan.



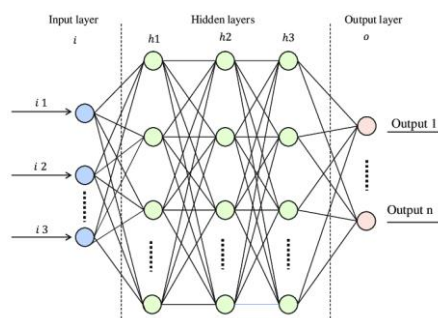
Gambar 2. Arsitektur CNN

Gambar 2 Merupakan Arsitektur CNN, CNN adalah jenis jaringan syaraf tiruan yang sangat efektif dalam menangani data berbentuk gambar atau citra. Model ini digunakan untuk ekstraksi fitur secara otomatis melalui lapisan konvolusi dan pooling [11]. Model ini sangat cocok untuk klasifikasi citra medis, termasuk klasifikasi penyakit kulit.



Gambar 1. Arsitektur LSTM

Gambar 3 Merupakan Arsitektur LSTM, LSTM adalah jenis *Recurrent Neural Network (RNN)* yang dirancang untuk mengatasi masalah *vanishing gradien* pada *RNN* tradisional, yang membatasi kemampuannya dalam mempelajari dependensi jangka panjang dalam urutan data [12]. LSTM sangat efektif dalam menangani data yang memiliki urutan panjang, termasuk dalam masalah klasifikasi citra yang memerlukan pemahaman urutan atau pola temporal.



Gambar 2. Arsitektur FNN

Gambar 4 Merupakan Arsitektur FNN. FNN adalah jenis jaringan syaraf yang paling sederhana di mana informasi mengalir dari input ke output melalui lapisan tersembunyi tanpa adanya umpan balik (umpan balik) [13].

FNN sering digunakan untuk masalah klasifikasi sederhana dan dapat digunakan dalam klasifikasi penyakit kulit, khususnya kulit luka diabetes dan kulit sehat.

2.4. Pelatihan Model

Pada tahap ini, model CNN, LSTM, atau FNN yang telah dirancang dilatih menggunakan data pelatihan. Sebelum pelatihan, model dikompilasi menggunakan optimizer Adam, fungsi loss yang sesuai dengan tugas klasifikasi (misalnya, categorical cross-entropy), dan metrik akurasi untuk evaluasi performa. Proses pelatihan dilakukan selama 10-20 epoch, di mana data pelatihan dimasukkan ke dalam model secara batch. Model memprediksi output untuk setiap batch data, dan selisih antara prediksi dan nilai sebenarnya dihitung menggunakan fungsi loss. Model kemudian melakukan backpropagation untuk menghitung gradien loss dan memperbarui bobotnya menggunakan algoritma optimizer. Untuk memastikan generalisasi model, data pelatihan dibagi menjadi data training (80%) dan data validasi (20%) [14]. Kinerja model selama pelatihan dipantau melalui metrik akurasi dan loss pada data validasi, yang membantu dalam mengidentifikasi overfitting atau underfitting. Selain itu, pelatihan diulang sebanyak tiga kali dengan pengaturan hyperparameter yang sama untuk memastikan konsistensi hasil. Model terbaik berdasarkan akurasi validasi tertinggi disimpan dalam format file (.h5) untuk evaluasi lebih lanjut. Langkah ini bertujuan untuk menjamin replikasi hasil dan meningkatkan keandalan model dalam aplikasi nyata.

2.5. Evaluasi Model

Setelah pelatihan selesai, tahap berikutnya adalah evaluasi model. Evaluasi dilakukan dengan menggunakan metrik evaluasi seperti akurasi dan loss, yang mengukur seberapa baik model dalam memprediksi data yang belum pernah dilihat sebelumnya. Selain itu, *confusion matrix* dan metrik lain seperti *precision*, *recall*, dan *F1-score* dapat digunakan untuk menganalisis kinerja model lebih mendalam [15]. Dengan cara ini, kita dapat mengetahui apakah model mengalami *overfitting* (terlalu menyesuaikan dengan data pelatihan) atau *underfitting* (kurang belajar dari data pelatihan).

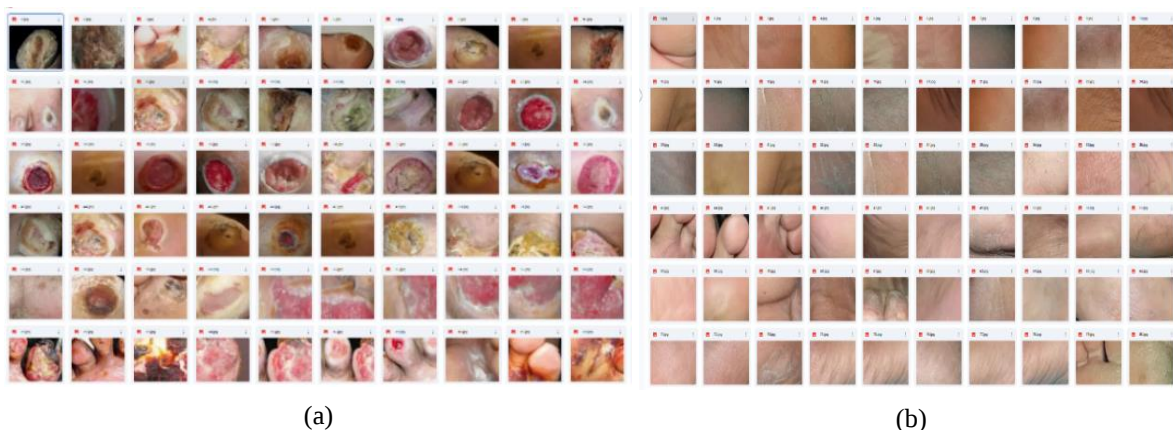
2.6. Implementasi Model

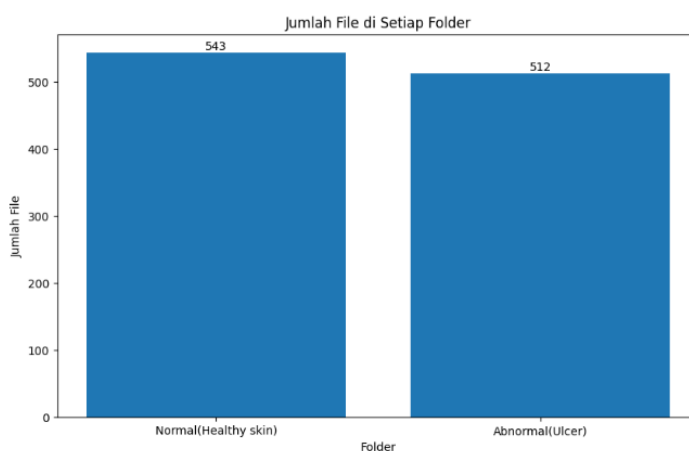
Setelah model *Deep Learning* selesai dilatih dan dievaluasi, langkah selanjutnya adalah mengimplementasikannya dalam sebuah aplikasi web menggunakan *Framework Flask*, sebuah *framework web Python* yang ringan dan mudah digunakan. *Flask* dipilih karena *framework* ini sangat cocok untuk membuat aplikasi web dengan cepat dan mengintegrasikan model deep learning yang telah dilatih untuk digunakan dalam prediksi.

3. HASIL DAN PEMBAHASAN

3.1. Dataset

Pada penelitian ini, Dataset terdiri dari dengan dua kategori utama, yaitu *Normal (Healthy Skin)* yang merepresentasikan kulit sehat, dan *Abnormal (Ulcer)* yang merepresentasikan luka kulit akibat diabetes. Data ini menjadi dasar dalam membangun dan mengevaluasi performa model deep learning untuk mendeteksi luka diabetes.



Gambar 5. (a) *Abnormal(ulcer)* (b) *Normal (Helathy skin)*

Gambar 6. Distribusi Jumlah Data

Gambar 6 adalah Distribusi Dataset yang digunakan dalam penelitian ini berjumlah 1055 data gambar. Dengan distribusi data 543 data *Normal (Healthy Skin)* dan 512 data *Abnormal (Ulcer)*. Sumber Data Dataset yang digunakan diambil dari sumber terbuka di website Kaggle dengan nama "Diabetic Foot Ulcer (DFU)". Dataset ini dirancang untuk mendukung penelitian dan pengembangan algoritma dalam deteksi luka diabetes. Dataset tersebut memiliki struktur folder yang telah terorganisasi untuk keperluan pelatihan (*training*) dan validasi (*validation*). Distribusi ini divisualisasikan pada Gambar 6 yang menunjukkan keseimbangan relatif antara kedua kelas, dengan sedikit kelebihan data pada kategori Normal (*Healthy Skin*).

Dataset *Diabetic Foot Ulcer* yang digunakan dalam penelitian ini menyediakan fondasi yang kuat untuk mengembangkan model deep learning dalam deteksi luka diabetes. Dengan distribusi yang cukup seimbang antara kelas Normal dan Abnormal, serta pengorganisasian data yang baik, dataset ini memfasilitasi proses pelatihan dan evaluasi model. Meskipun dataset memiliki keterbatasan ukuran, teknik augmentasi data yang diterapkan dapat mengatasi masalah tersebut dan memungkinkan model untuk belajar pola dengan lebih baik. Analisis mendalam terhadap dataset ini menunjukkan potensi besar untuk digunakan dalam aplikasi medis, terutama dalam mendukung diagnosis luka diabetes secara otomatis.

3.2. Pra Pemrosesan Data

Tabel 1. Pra Pemrosesan Data

No	Tahapan Preprocessing	Penjelasan
1	Resize	Mengubah ukuran gambar menjadi dimensi yang seragam (128x128 piksel). Tujuannya adalah untuk memastikan semua gambar memiliki dimensi yang konsisten, sehingga memudahkan proses komputasi.
2	Image Sequence	Membentuk data gambar menjadi urutan (sequence) dengan panjang tertentu (misalnya 10 gambar per urutan). Hal ini penting untuk model RNN yang membutuhkan data dalam bentuk sekuensial.
3	Normalization	Menormalisasi nilai piksel gambar dengan membagi setiap nilai piksel dengan 255. Langkah ini memastikan nilai piksel berada dalam rentang [0, 1], yang mempercepat proses pelatihan model dan stabilitas konvergensi.
4	Validation Split	Membagi dataset menjadi dua bagian: 80% untuk data training dan 20% untuk data validation. Ini memastikan bahwa model dapat dievaluasi pada data yang belum pernah dilihat selama pelatihan.
5	Shuffling and Batching	Mengacak urutan data untuk menghindari bias selama pelatihan, lalu membagi data ke dalam batch dengan ukuran tertentu (contoh: 32 gambar per batch). Tujuannya adalah meningkatkan efisiensi pelatihan model.
6	One-Hot Encoding	Mengubah label data menjadi format one-hot encoding untuk klasifikasi multi-kelas. Misalnya, label untuk kelas "Abnormal"

menjadi [1, 0], dan "Normal" menjadi [0, 1]. Format ini membantu model memahami target output dengan lebih baik.

CNN fokus pada preprocessing seperti *resize*, *normalization*, *shuffling*, *batching*, dan *one-hot encoding*. Tidak memerlukan tahapan *image sequence*. LSTM membutuhkan tambahan *preprocessing image sequence* untuk menghasilkan urutan gambar (*sequence*) yang menjadi input. FNN Menggunakan *preprocessing* standar seperti *resize*, *normalization*, *validation split*, *shuffling*, *batching*, dan *one-hot encoding* tanpa memanfaatkan data sekuensial.

3.3. Membangun Model

Gambar 11 adalah Model CNN yang dibangun untuk klasifikasi citra kulit penderita diabetes dan kulit normal, yang terdiri dari beberapa lapisan utama lapisan konvolusi (*Conv2D*) untuk mengekstraksi fitur dari citra, diikuti oleh lapisan pooling (*MaxPooling2D*) untuk mereduksi dimensi dan mengurangi kompleksitas. Setelah itu, output dari lapisan konvolusi dan pooling diratakan (*Flatten*) dan diproses melalui lapisan *fully connected* (*Dense*) yang dilengkapi dengan fungsi aktivasi *Relu* untuk menambah non-linearitas. Untuk mencegah *overfitting*, digunakan lapisan *dropout*. Akhirnya, lapisan *output* dengan fungsi aktivasi *sigmoid* menghasilkan prediksi *biner*, mengklasifikasikan citra sebagai kulit sehat atau kulit diabetik. Model ini dilatih menggunakan dataset citra kulit dan diuji untuk evaluasi kinerjanya, menghasilkan akurasi yang menjanjikan dalam mendeteksi kondisi kulit diabetik secara otomatis [16].

```
# Membangun model CNN
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid') # Output binary (0 atau 1)
])
```

Gambar 11. Model CNN

```
# Model Architecture
model = Sequential([
    # TimeDistributed CNN for Feature Extraction
    TimeDistributed(Conv2D(32, (3, 3), activation='relu', input_shape=
        (SEQUENCE_LENGTH, IMG_HEIGHT, IMG_WIDTH, 3))),
    TimeDistributed(MaxPooling2D((2, 2))),
    TimeDistributed(Conv2D(64, (3, 3), activation='relu')),
    TimeDistributed(MaxPooling2D((2, 2))),
    TimeDistributed(Conv2D(128, (3, 3), activation='relu')),
    TimeDistributed(MaxPooling2D((2, 2))),
    TimeDistributed(Flatten()),

    # LSTM Layers
    LSTM(128, return_sequences=True),
    Dropout(0.5),
    LSTM(64),

    # Dense for Binary Classification
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

Gambar 12. Model LSTM

```
# Inisialisasi model
model = Sequential()
# Flatten layer untuk merubah data gambar menjadi satu dimensi
model.add(Flatten(input_shape=(image_size[0], image_size[1], 3)))
# Layer tersembunyi pertama
model.add(Dense(128, activation='relu'))
# Layer tersembunyi kedua
model.add(Dense(64, activation='relu'))
# Output layer dengan 2 kelas
model.add(Dense(2, activation='softmax'))
# Menampilkan ringkasan model
model.summary()
```

Gambar 13. Model FNN

Gambar 12 adalah model LSTM yang mempunyai beberapa tahapan. LSTM menggunakan *TimeDistributed* untuk menerapkan lapisan *Conv2D* (dengan filter 32, 64, dan 128) dan *MaxPooling2D* pada setiap citra dalam urutan, memungkinkan ekstraksi fitur dari setiap gambar secara terpisah. Setelah itu, hasil ekstraksi fitur tersebut diratakan (*flatten*) untuk diproses oleh lapisan LSTM yang menangani urutan temporal. LSTM pertama dengan 128 unit memori mengembalikan output sekuensial untuk mempertahankan informasi dari seluruh urutan gambar, yang kemudian diteruskan ke LSTM kedua dengan 64 unit untuk merangkum informasi penting dari urutan tersebut. Untuk mengurangi *overfitting*, model menggunakan lapisan *Dropout* setelah LSTM dan *fully connected layers*. Pada akhirnya, lapisan *Dense* dengan fungsi aktivasi *sigmoid* menghasilkan klasifikasi biner, menentukan apakah citra tersebut termasuk dalam kategori kulit sehat atau kulit diabetik [17]. Model ini menggabungkan kemampuan CNN dalam mengekstraksi fitur citra dengan kekuatan LSTM dalam memproses informasi urutan, memberikan hasil klasifikasi yang akurat berdasarkan pola yang teridentifikasi dalam data sekuensial.

Gambar 13 adalah model *Feedforward Neural Network (FNN)* yang digunakan untuk klasifikasi gambar dalam dua kelas. Dimulai dengan lapisan *Flatten* yang mengubah data gambar 2D menjadi vektor 1D agar bisa diproses oleh lapisan *dense*. Setelah itu, model memiliki dua lapisan tersembunyi *Dense*, masing-masing dengan 128 dan 64 unit, serta menggunakan fungsi aktivasi *Relu* untuk menangkap pola non-linear dalam data. Lapisan-lapisan tersembunyi ini bertugas mempelajari representasi fitur dari gambar, dengan lapisan pertama mengidentifikasi fitur yang lebih umum dan lapisan kedua merangkum informasi tersebut. Akhirnya, lapisan *Dense* dengan 2 unit dan fungsi aktivasi *softmax* digunakan untuk menghasilkan probabilitas untuk dua kelas yang berbeda, misalnya, kulit sehat dan kulit diabetik. *Softmax* mengonversi *output* menjadi nilai probabilitas yang berjumlah 1, yang membantu dalam menentukan kelas mana yang lebih mungkin [18]. Model ini cukup sederhana namun efektif dalam menangani tugas klasifikasi citra *biner* dengan dua kelas.

3.4. Pelatihan Model

Gambar 14 Merupakan Proses pelatihan model *Convolutional Neural Network (CNN)* menggunakan 10 *epoch*. dalam log tersebut menunjukkan perkembangan yang signifikan dalam kinerja model dari *epoch* ke *epoch*. Pada *epoch* pertama, model menunjukkan akurasi pelatihan yang rendah (54.41%) dan *loss* yang tinggi (1.0224), yang menunjukkan bahwa model masih dalam tahap belajar dan belum optimal dalam memprediksi data. Namun, seiring berjalannya pelatihan, akurasi pelatihan meningkat secara signifikan, mencapai 94.40% pada *epoch* ke-10, sementara *loss* menurun drastis menjadi 0.1652, menunjukkan bahwa model semakin baik dalam memprediksi data yang dilatih.

Pada sisi lain, akurasi validasi yang dimulai dari 79.52% pada *epoch* pertama, juga meningkat stabil hingga mencapai 87.14% pada *epoch* terakhir, meskipun ada sedikit fluktuasi antara *epoch*. *Loss* validasi yang dimulai dari 0.4445 juga menunjukkan penurunan yang konsisten, meskipun sedikit lebih tinggi dibandingkan dengan *loss* pelatihan. Hal ini menunjukkan bahwa model mampu menggeneralisasi dengan cukup baik pada data validasi yang tidak terlihat sebelumnya, tanpa mengalami *overfitting* yang signifikan. Secara keseluruhan, model menunjukkan perkembangan yang sangat baik selama pelatihan, dengan akurasi yang semakin tinggi dan *loss* yang semakin rendah, yang menandakan bahwa model CNN berhasil belajar dengan baik dan mampu memberikan prediksi yang lebih akurat pada data yang belum pernah dilihat sebelumnya.

Gambar 15 adalah proses pelatihan model LSTM yang dilakukan selama 20 *epoch*, terlihat adanya peningkatan signifikan dalam akurasi dan penurunan *loss* pada data pelatihan dan validasi. Pada *epoch* pertama, model masih memiliki akurasi 50.82% dan *loss* 0.6978, tetapi seiring berjalannya waktu, performa model meningkat pesat. Pada *epoch* ke-6, akurasi pelatihan mencapai 95.35% dengan *loss* 0.1329, sementara akurasi

validasi juga meningkat menjadi 94.37%. Pada *epoch* ke-7, model mencapai akurasi pelatihan 96.06% dan akurasi validasi 97.18%, yang menunjukkan kemampuan model dalam menggeneralisasi dengan baik.

```

/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your
self._warn_if_super_not_called()
Epoch 2/10 ----- 193s 6s/step - accuracy: 0.5441 - loss: 1.0224 - val_accuracy: 0.7952 - val_loss: 0.4445
Epoch 3/10 ----- 128s 4s/step - accuracy: 0.8242 - loss: 0.4119 - val_accuracy: 0.7619 - val_loss: 0.4961
Epoch 4/10 ----- 118s 4s/step - accuracy: 0.8086 - loss: 0.4874 - val_accuracy: 0.8667 - val_loss: 0.3345
Epoch 5/10 ----- 150s 5s/step - accuracy: 0.9185 - loss: 0.2592 - val_accuracy: 0.8810 - val_loss: 0.3019
Epoch 6/10 ----- 117s 4s/step - accuracy: 0.9238 - loss: 0.2388 - val_accuracy: 0.8810 - val_loss: 0.3044
Epoch 7/10 ----- 143s 4s/step - accuracy: 0.9278 - loss: 0.2189 - val_accuracy: 0.8667 - val_loss: 0.3793
Epoch 8/10 ----- 140s 4s/step - accuracy: 0.9213 - loss: 0.2403 - val_accuracy: 0.8857 - val_loss: 0.3300
Epoch 9/10 ----- 159s 5s/step - accuracy: 0.9390 - loss: 0.1895 - val_accuracy: 0.8571 - val_loss: 0.3520
Epoch 10/10 ----- 123s 4s/step - accuracy: 0.8756 - loss: 0.2910 - val_accuracy: 0.8714 - val_loss: 0.3101
Epoch 10/10 ----- 115s 4s/step - accuracy: 0.9440 - loss: 0.1652 - val_accuracy: 0.8714 - val_loss: 0.3471

```

Gambar 14. Pelatihan Model CNN

```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/wrapper.py:27: UserWarning: Do not pass an `input_shape`/:
super().__init__(**kwargs)
Epoch 1/20 ----- 15/15 458s 30s/step - accuracy: 0.5082 - loss: 0.6978 - val_accuracy: 0.5208 - val_loss: 0.6783
Epoch 2/20 ----- 15/15 378s 26s/step - accuracy: 0.5822 - loss: 0.6746 - val_accuracy: 0.6761 - val_loss: 0.6044
Epoch 3/20 ----- 15/15 383s 25s/step - accuracy: 0.6685 - loss: 0.5865 - val_accuracy: 0.7887 - val_loss: 0.4376
Epoch 4/20 ----- 15/15 372s 24s/step - accuracy: 0.8972 - loss: 0.2578 - val_accuracy: 0.9014 - val_loss: 0.2682
Epoch 5/20 ----- 15/15 403s 27s/step - accuracy: 0.8809 - loss: 0.3089 - val_accuracy: 0.9271 - val_loss: 0.2855
Epoch 6/20 ----- 15/15 379s 26s/step - accuracy: 0.9535 - loss: 0.2082 - val_accuracy: 0.9437 - val_loss: 0.1329
Epoch 7/20 ----- 15/15 374s 25s/step - accuracy: 0.9606 - loss: 0.1125 - val_accuracy: 0.9718 - val_loss: 0.0821
Epoch 8/20 ----- 15/15 373s 25s/step - accuracy: 0.9822 - loss: 0.0664 - val_accuracy: 0.8873 - val_loss: 0.5086
Epoch 9/20 ----- 15/15 396s 26s/step - accuracy: 0.9949 - loss: 0.0402 - val_accuracy: 0.9583 - val_loss: 0.1965
Epoch 10/20 ----- 15/15 371s 26s/step - accuracy: 0.9872 - loss: 0.0510 - val_accuracy: 0.9296 - val_loss: 0.3875
Epoch 11/20 -----

```

Gambar 13. Pelatihan Model LSTM

```

Epoch 1/30 ----- 27/27 [=====] - 1s 47ms/step - loss: 0.5826 - accuracy: 0.7050 - val_loss: 0.8284 - val_accuracy: 0.5782
Epoch 2/30 ----- 27/27 [=====] - 1s 44ms/step - loss: 0.7538 - accuracy: 0.6460 - val_loss: 0.5840 - val_accuracy: 0.6019
Epoch 3/30 ----- 27/27 [=====] - 1s 42ms/step - loss: 0.7607 - accuracy: 0.6363 - val_loss: 0.6054 - val_accuracy: 0.6919
Epoch 4/30 ----- 27/27 [=====] - 1s 44ms/step - loss: 0.5798 - accuracy: 0.7109 - val_loss: 0.6528 - val_accuracy: 0.5118
Epoch 5/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.6085 - accuracy: 0.6991 - val_loss: 0.5676 - val_accuracy: 0.7441
Epoch 6/30 ----- 27/27 [=====] - 1s 44ms/step - loss: 0.5162 - accuracy: 0.8092 - val_loss: 0.5470 - val_accuracy: 0.7678
Epoch 7/30 ----- 27/27 [=====] - 1s 49ms/step - loss: 0.5162 - accuracy: 0.7642 - val_loss: 0.8739 - val_accuracy: 0.5640
Epoch 8/30 ----- 27/27 [=====] - 1s 50ms/step - loss: 0.5783 - accuracy: 0.7014 - val_loss: 0.5926 - val_accuracy: 0.7156
Epoch 9/30 ----- 27/27 [=====] - 1s 48ms/step - loss: 0.6198 - accuracy: 0.6564 - val_loss: 0.5418 - val_accuracy: 0.7773
Epoch 10/30 ----- 27/27 [=====] - 1s 48ms/step - loss: 0.5890 - accuracy: 0.7026 - val_loss: 0.5224 - val_accuracy: 0.7725
Epoch 11/30 ----- 27/27 [=====] - 1s 48ms/step - loss: 0.4919 - accuracy: 0.8081 - val_loss: 0.6996 - val_accuracy: 0.6445
Epoch 12/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.4828 - accuracy: 0.8009 - val_loss: 0.4969 - val_accuracy: 0.8057
Epoch 13/30 ----- 27/27 [=====] - 1s 42ms/step - loss: 0.4913 - accuracy: 0.7773 - val_loss: 0.4872 - val_accuracy: 0.7393
Epoch 14/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.4432 - accuracy: 0.8223 - val_loss: 0.4707 - val_accuracy: 0.8152
Epoch 15/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.4762 - accuracy: 0.7855 - val_loss: 0.4679 - val_accuracy: 0.7536
Epoch 16/30 ----- 27/27 [=====] - 1s 44ms/step - loss: 0.5071 - accuracy: 0.7678 - val_loss: 0.8124 - val_accuracy: 0.5261
Epoch 17/30 ----- 27/27 [=====] - 1s 41ms/step - loss: 0.5100 - accuracy: 0.7630 - val_loss: 0.4612 - val_accuracy: 0.8152
Epoch 18/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.4338 - accuracy: 0.8223 - val_loss: 0.4885 - val_accuracy: 0.7915
Epoch 19/30 ----- 27/27 [=====] - 1s 43ms/step - loss: 0.4211 - accuracy: 0.8211 - val_loss: 0.6142 - val_accuracy: 0.7014
Epoch 20/30 ----- 27/27 [=====] - 1s 42ms/step - loss: 0.4225 - accuracy: 0.8282 - val_loss: 0.6980 - val_accuracy: 0.5388

```

Gambar 14. Pelatihan model FNN

Namun, meskipun akurasi pelatihan mencapai 100% pada *epoch-epoch* terakhir, akurasi validasi mengalami sedikit fluktuasi, dengan puncaknya tercatat pada epoch ke-7 (97.18%). Meskipun begitu, model tetap stabil dengan akurasi validasi yang konsisten antara 92%-95% pada *epoch* berikutnya. *Loss* pelatihan menurun secara signifikan dari 1.0224 pada *epoch* pertama hingga 0.00063468 pada *epoch* terakhir, menunjukkan bahwa model semakin baik dalam memprediksi data. *Loss* validasi juga menurun dari 0.6783 menjadi 0.3677, yang menandakan bahwa model mampu mengurangi kesalahan pada data yang belum pernah dilihat sebelumnya.

Gambar 16 adalah Model *Feedforward Neural Network (FNN)* yang sedang mengalami pelatihan selama 30 *epoch* dengan hasil yang bervariasi. Pada *epoch* pertama, model memiliki akurasi 70.50% dan loss 0.5826, yang

menunjukkan bahwa model masih kesulitan dalam mengklasifikasikan data dengan benar. Seiring berjalannya pelatihan, pada epoch ke 10, akurasi meningkat menjadi 70.26%, dan *loss* relatif menurun pada 0.5890. hal ini menunjukkan bahwa model mulai belajar lebih baik.

Setelah 30 *epoch*, model diuji pada data pengujian, dan *test_accuracy* tercatat sebesar 82.82% dengan *test_loss* sebesar 0.4225, yang menunjukkan bahwa model belajar cukup baik, akurasi yang dicapai masih bisa ditingkatkan. Hasil ini mengindikasikan bahwa meskipun FNN berhasil mempelajari pola dasar dalam data, performa klasifikasinya belum optimal, terutama dalam hal kemampuan generalisasi pada data yang tidak terlihat sebelumnya

3.5. Hasil Akurasi Model

Berdasarkan Gambar 17, Model CNN menunjukkan kinerja yang sangat baik dengan akurasi validasi sekitar 87%. *Precision Overall* 89 %, *recall Overall* 87%, dan *F1-Score Overall* 87%. Hal ini menunjukkan hasil yang seimbang antara kedua kelas menunjukkan bahwa model memiliki keseimbangan yang baik antara *Precision* dan *Recall*.

	precision	recall	f1-score	support
0.0	0.99	0.75	0.85	102
1.0	0.80	0.99	0.89	108
accuracy			0.87	210
macro avg	0.90	0.87	0.87	210
weighted avg	0.89	0.87	0.87	210

Gambar 15. Akurasi Model CNN

	precision	recall	f1-score	support
0	1.00	0.89	0.94	36
1	0.90	1.00	0.95	35
accuracy			0.94	71
macro avg	0.95	0.94	0.94	71
weighted avg	0.95	0.94	0.94	71

Gambar 16. Akurasi Model LSTM

Classification Report for FNN Model:				
	precision	recall	f1-score	support
Abnormal(Ulcer)	0.97	0.66	0.79	110
Normal(Healthy skin)	0.73	0.98	0.84	101
accuracy			0.82	211
macro avg	0.85	0.82	0.81	211
weighted avg	0.86	0.82	0.81	211

Gambar 19. Akurasi Model FNN

Berdasarkan Gambar 18, Model LSTM menunjukkan kinerja yang sangat baik dengan akurasi 94% *Precision Overall* 95 %, *recall Overall* 94%, dan *F1-Score Overall* 94%. *Precision* dan *recall* yang tinggi menunjukkan bahwa model mampu membedakan antara kelas 0 dan 1 dengan sangat baik. *F1-Score* yang tinggi pada kedua kelas menunjukkan bahwa model tidak hanya akurat tetapi juga seimbang dalam mendeteksi kedua kelas. Nilai *Loss* yang relatif rendah (0.56) menunjukkan bahwa model sudah sangat baik dalam meminimalkan kesalahan saat mengklasifikasikan data.

Berdasarkan Gambar 19, Model FNN menunjukkan akurasi 82% pada data uji, dengan *Precision Overall* 85 %, *recall Overall* 81%, dan *F1-Score Overall* 81%. *precision* yang baik pada kelas "Abnormal" (97%) namun

recall yang rendah (66%). Hal ini mengindikasikan bahwa model lebih banyak salah mengidentifikasi kelas "Abnormal" meskipun memiliki ketepatan yang tinggi saat memprediksi kelas ini.

3.6. Perbandingan Model

Berdasarkan tabel 2, model LSTM menunjukkan performa terbaik dengan *accuracy* 94%, diikuti oleh CNN dengan 87%, dan FNN dengan 82%. Model LSTM memiliki *precision*, *recall*, dan *F1 score* yang lebih tinggi dibandingkan CNN dan FNN, meskipun *loss*-nya sedikit lebih tinggi. Model CNN memiliki *precision*, *recall*, dan *F1 score* yang baik, serta *loss* terendah. Sementara itu, FNN menunjukkan performa terendah dalam hal *accuracy*, *precision*, *recall*, dan *F1 score*, meskipun *loss*-nya masih cukup baik. Secara keseluruhan, LSTM memberikan hasil terbaik dalam klasifikasi, diikuti oleh CNN, dan FNN di posisi terakhir.

Tabel 2. Perbandingan Model

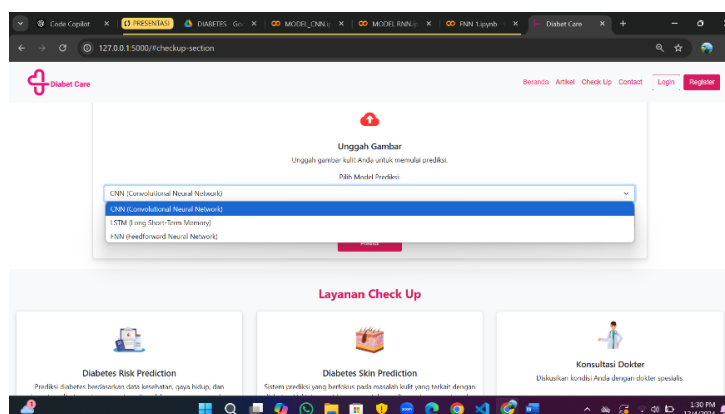
Nama Model	Accuracy	Precision (overall)	Recall (overall)	F1 Score (overall)	Loss
CNN	87 %	0.94	0.94	0.92	0.3125
LSTM	94 %	0.90	0.94	0.94	0.5639
FNN	82 %	0.85	0.81	0.81	0.5639

3.7. Implementasi Sistem

Setelah semua model di evaluasi selanjutnya 3 model (CNN, LSTM dan FNN) akan di implementasikan ke sistem berbasis web untuk proses klasifikasi kulit. Berikut ini adalah Tampilan Utama dari sistem.

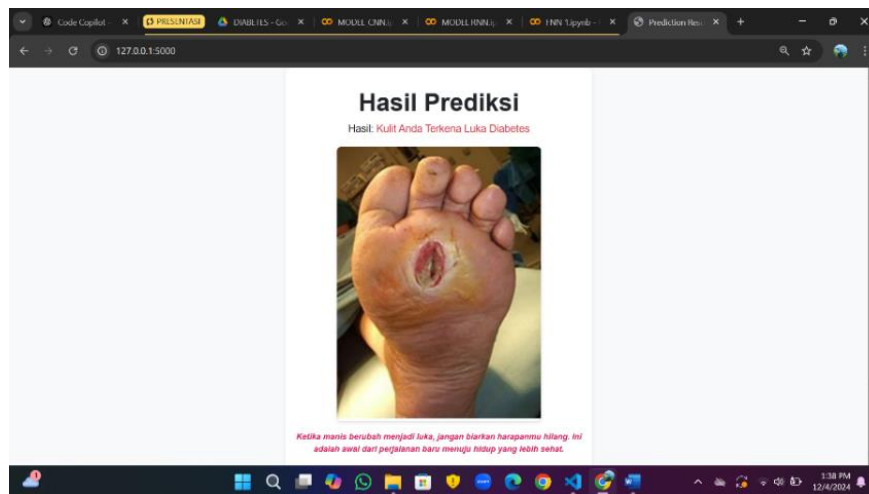


Gambar 20. Dashboard Utama Sistem

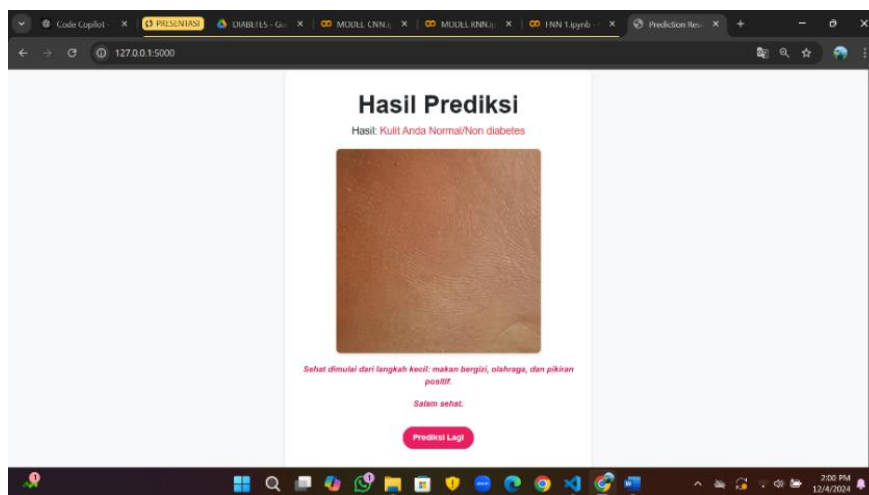


Gambar 21. Halaman untuk Klasifikasi

Gambar 20 adalah halaman *Dashboard* utama sistem, Gambar 21 adalah halaman Klasifikasi terdapat 2 form input, pertama untuk *option choice* model yang akan digunakan, kemudian ada *form* unggah file, yang berfungsi sebagai form unggah file gambar yang akan di prediksi. selanjutnya ada *button* prediksi yang berfungsi untuk memulai memprediksi gambar.



Gambar 22. Hasil prediksi Kulit Abnormal (Ulcer)



Gambar 23. Hasil Prediksi Kulit Normal (Healthy Skin)

Gambar 22 dan 23 Merupakan hasil prediksi model setelah di implementasikan ke web. Sistem berhasil memprediksi dengan akurat antara kulit luka diabetes (*Abnormal*) dan kulit normal (*Healthy skin*). Dengan menggunakan teknologi deep learning, *CNN*, *LSTM* dan *FNN* yang dilatih untuk mengidentifikasi perbedaan antara kedua kondisi kulit, aplikasi web ini mampu memberikan hasil yang dapat diandalkan. Sistem ini tidak hanya menunjukkan kemampuan untuk mendeteksi ulkus pada kulit penderita diabetes, tetapi juga dapat membedakan dengan baik antara kulit yang sehat.

3.8. Diskusi

Hasil penelitian menunjukkan bahwa model *LSTM* memberikan performa terbaik dalam klasifikasi gambar kulit dengan akurasi mencapai 94%, diikuti oleh *CNN* dengan 87%, dan *FNN* dengan 82%. Selain akurasi, model *LSTM* juga memiliki nilai precision, recall, dan F1 score yang lebih tinggi dibandingkan *CNN* dan *FNN*, meskipun loss-nya sedikit lebih tinggi. Hal ini menunjukkan bahwa *LSTM*, yang dirancang untuk menangani data berurutan, mampu menangkap pola kompleks dalam data citra kulit diabetes. Temuan ini konsisten dengan penelitian sebelumnya oleh [3], yang menunjukkan bahwa *LSTM* efektif dalam pengolahan data yang memiliki hubungan temporal atau spasial.

Model *CNN*, meskipun berada di posisi kedua, menunjukkan keunggulan dalam loss yang lebih rendah serta precision, recall, dan F1 score yang tetap kompetitif. Kemampuan *CNN* untuk mengekstraksi fitur penting dari data citra menjadi alasan utama performanya yang baik, seperti juga dilaporkan oleh [19] dalam konteks klasifikasi citra medis. Di sisi lain, *FNN* menunjukkan performa terendah dalam semua metrik evaluasi kecuali loss, yang mengindikasikan keterbatasannya dalam menangkap pola kompleks dari data citra dibandingkan dua model lainnya.

Implikasi dari hasil ini sangat penting dalam pengembangan sistem deteksi dini untuk pasien diabetes. Model LSTM, dengan akurasi tertinggi dan kemampuan untuk menangkap pola kompleks, dapat diimplementasikan dalam sistem berbasis web untuk membantu dokter dalam diagnosis dini. Namun, hasil ini juga menunjukkan bahwa CNN tetap menjadi alternatif yang kuat jika efisiensi komputasi menjadi prioritas, karena memiliki loss yang lebih rendah. FNN, meskipun menunjukkan performa terendah, tetap relevan untuk aplikasi dengan kebutuhan pemrosesan ringan.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa model LSTM merupakan model terbaik untuk klasifikasi gambar kulit diabetes dengan akurasi 94%, precision, recall, dan F1 score yang tinggi, terutama dalam mendeteksi kelas Normal. Model CNN berada di posisi kedua dengan akurasi 87% dan precision yang baik untuk kelas Abnormal, meskipun recall untuk kelas Normal sedikit lebih rendah. Sementara itu, FNN menunjukkan performa terendah dengan akurasi 82% dan kesulitan dalam mendeteksi kelas Abnormal. Implementasi sistem berbasis model CNN dan LSTM berjalan dengan baik, mampu memproses dan mengklasifikasi gambar secara cepat dan akurat. Sebaliknya, FNN kurang optimal karena tidak memiliki kemampuan untuk mengekstraksi fitur spasial atau menangani hubungan antar piksel, yang esensial dalam klasifikasi gambar.

Berdasarkan hasil ini, model LSTM direkomendasikan untuk diimplementasikan dalam sistem deteksi dini kulit diabetes karena akurasi dan performanya yang unggul. Model CNN juga menjadi alternatif yang baik jika efisiensi komputasi menjadi prioritas. Penelitian di masa mendatang dapat memperluas dataset, mengeksplorasi teknik ensemble model untuk meningkatkan akurasi, dan mengembangkan pendekatan baru untuk meningkatkan performa FNN pada data gambar. Selain itu, integrasi sistem dengan perangkat berbasis IoT dapat menjadi langkah inovatif untuk mendukung diagnosis dini di berbagai lingkungan medis.

DAFTAR PUSTAKA

- [1] A. F. Zubir, S. Brisma, A. Zulkarnaini, and M. Anissa, "Gambaran Penderita Ulkus Diabetikum Yang Menjalani Tindakan Operasi," 2024, <https://doi.org/10.56260/sciena.v3i4.151>
- [2] R. Yohannes and M. E. Al Rivan, "Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM," *algoritme*, vol. 2, no. 2, Art. no. 2, Apr. 2022, doi: 10.35957/algoritme.v2i2.2363.
- [3] G. E. H. Cahyadi, S. Sukemi, and D. P. Rini, "Peningkatan Akurasi Prediksi Cnn-Lstm Dan Cnn-Gru Untuk Mendiagnosa Skizofrenia Melalui Sinyal Eeg," *JSI*, vol. 14, no. 2, Art. no. 2, Oct. 2022, doi: 10.18495/jsi.v14i2.19071.
- [4] P. V. De Campos Souza and M. Dragoni, "IFNN: Enhanced interpretability and optimization in FNN via Adam algorithm," *Information Sciences*, vol. 678, p. 121002, Sep. 2024, doi: 10.1016/j.ins.2024.121002.
- [5] Qorry Aina Fitroh and Shofwatul 'Uyun, "Deep Transfer Learning untuk Meningkatkan Akurasi Klasifikasi pada Citra Dermoskopi Kanker Kulit," *JNTETI*, vol. 12, no. 2, Art. no. 2, May 2023, doi: 10.22146/jnteti.v12i2.6502.
- [6] S. Rizal, N. Ibrahim, N. K. C. Pratiwi, S. Saidah, and R. Y. N. Fu'Adah, "Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNet," *ELKOMIKA*, vol. 8, no. 3, Art. no. 3, Aug. 2020, doi: 10.26760/elkomika.v8i3.693.
- [7] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," *SJI*, vol. 8, no. 1, pp. 144–153, May 2021, doi: 10.15294/sji.v8i1.26888.
- [8] C. Kavitha, S. Priyanka, M. P. Kumar, and V. Kusuma, "Skin Cancer Detection and Classification using Deep Learning Techniques," *Procedia Computer Science*, vol. 235, pp. 2793–2802, 2024, doi: 10.1016/j.procs.2024.04.264.
- [9] K. A. H. Wibowo and A. Alamsyah, "Klasifikasi Diabetic Retinopathy Menggunakan CNN dengan Arsitektur yang Dimodifikasi," *ijmns*, vol. 46, no. 1, Art. no. 1, Apr. 2023, doi: 10.15294/ijmns.v46i1.46172.
- [10] C. R. Mulyasari, A. I. Hadiana, and A. Komarudin, "DETEKSI PENYAKIT DIABETES, KATARAK DAN GLAUKOMA PADA CITRA FUNDUS RETINA MATA MANUSIA MENGGUNAKAN CNN DENGAN ARSITEKTUR ALEXNET". 2024, <https://doi.org/10.26874/jumanji.v8i1.341>
- [11] D. Irfansyah, M. Mustikasari, and A. Suroso, "Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi," *JPIT*, vol. 6, no. 2, pp. 87–92, May 2021, doi: 10.30591/jpit.v6i2.2802.

-
- [12] M. Rizki, S. Basuki, and Y. Azhar, "Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang," *JR*, vol. 2, no. 3, Jan. 2024, doi: 10.22219/repositor.v2i3.30499.
- [13] A. Yudhistira and A. Wirasto, "Penggunaan Logika Fuzzy dalam Deteksi Penyakit Kanker," vol. 1, no. 1, 2024, <http://dx.doi.org/10.30646/sinus.v15i1.259>.
- [14] R. Saifan and F. Jubair, "Six skin diseases classification using deep convolutional neural network," *IJECE*, vol. 12, no. 3, p. 3072, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3072-3082.
- [15] M. Emhandyksa, I. Soesanti, and R. Susilowati, "Pengembangan Deep Learning untuk Sistem Deteksi Dini Komplikasi Kaki Diabetik Menggunakan Citra Termogram," *JTIK*, vol. 10, no. 6, Art. no. 6, Dec. 2023, doi: 10.25126/jtiik.1067382.
- [16] Y. Liu, H. Pu, and D.-W. Sun, "Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices," *Trends in Food Science & Technology*, vol. 113, pp. 193–204, Jul. 2021, doi: 10.1016/j.tifs.2021.04.042.
- [17] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, Apr. 2021, doi: 10.3390/s21082852.
- [18] S. Li, M. Ardabilian, and A. Zine, "Quantitative Analysis of Skin using Diffuse Reflectance for Non-invasive Pigments Detection:," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Online Streaming, --- Select a Country ---: SCITEPRESS - Science and Technology Publications, 2021, pp. 604–614. doi: 10.5220/0010326806040614.
- [19] M. Muslih and E. H. Rachmawanto, "CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI CITRA PENYAKIT DIABETES RETINOPATHY," *SKANIKA*, vol. 5, no. 2, Art. no. 2, Jul. 2022, doi: 10.36080/skanika.v5i2.2945.