

Klasifikasi Kanker Kulit pada Data Tidak Seimbang Menggunakan *Ensemble CNN* dan *Support Vector Machine* Berbasis *Feature Fusion*

1st Mirza Helga Prabatanadi
SI Teknik Informatika

Telkom University Purwokerto
Purwokerto, Indonesia

mirzaa@student.telkomuniversity.ac.id

2nd Nicolaus Euclides Wahyu Nugroho
SI Teknik Informatika

Telkom University Purwokerto
Purwokerto, Indonesia

nicolausn@telkomuniversity.ac.id

3rd Wahyu Andi Saputra
SI Teknik Informatika

Telkom University Purwokerto
Purwokerto, Indonesia

andiwahyu@telkomuniversity.ac.id

Abstrak — Kanker kulit merupakan masalah kesehatan serius yang memerlukan deteksi dini akurat untuk meningkatkan peluang kesembuhan. Tantangan utama dalam diagnosis otomatis berbantuan komputer adalah ketidakseimbangan data dermoskopi, seperti pada dataset HAM10000, yang menyebabkan model cenderung bias memprediksi kelas mayoritas. Penelitian ini bertujuan membangun sistem klasifikasi yang adaptif menggunakan pendekatan Hybrid Learning. Metode yang diusulkan menerapkan strategi penyeimbangan data Hybrid Sampling dan teknik Feature Fusion yang menggabungkan representasi fitur visual dari Ensemble CNN yang terdiri dari EfficientNet, DenseNet, dan VGG16. Vektor fitur gabungan tersebut kemudian diklasifikasikan menggunakan algoritma Support Vector Machine (SVM). Hasil penelitian mencatat perolehan akurasi pelatihan sebesar 99% dan akurasi validasi mencapai 82%. Pencapaian ini menunjukkan bahwa integrasi fitur multi-model dengan kemampuan diskriminatif SVM efektif mengatasi dominasi kelas mayoritas serta menghasilkan performa diagnostik yang lebih stabil dibandingkan penggunaan arsitektur model tunggal dalam membedakan tujuh jenis penyakit kulit.

Kata Kunci — kanker kulit, ensemble cnn, support vector machine, feature fusion, data tidak seimbang, deep learning

I. PENDAHULUAN

Kulit merupakan organ terbesar tubuh manusia yang memiliki peran vital sebagai pelindung dari faktor lingkungan. Namun, dalam beberapa tahun terakhir, paparan radiasi *ultraviolet* (UV) yang meningkat telah memicu kenaikan kasus keganasan pada kulit [1]. Mengacu pada data statistik *Global Cancer Statistics* (GLOBOCAN), insiden kanker kulit terus menunjukkan tren peningkatan di seluruh dunia, menjadikannya salah satu prioritas masalah kesehatan global [2].

Tantangan terbesar dalam penanganan penyakit ini terletak pada deteksi dini. Diagnosis kanker kulit pada

stadium awal merupakan langkah kritis untuk menurunkan angka morbiditas dan mortalitas [2]. Namun, dalam praktik klinis deteksi awal sering kali menghadapi kendala karena manifestasi visual yang mirip antara lesi jinak (seperti *melanocytic nevi* atau *benign keratosis*) dan lesi ganas (seperti *melanoma* atau *basal cell carcinoma*) [3].

Dalam dunia medis, dikenal beberapa jenis kanker kulit yang sering dijumpai, seperti *melanocytic nevi*, *melanoma*, *benign keratosis-like lesions*, *basal cell carcinoma*, *actinic keratoses*, *vascular lesions*, dan *dermatofibroma*. Dari semua jenis tersebut, *melanoma* merupakan tipe yang paling berbahaya karena penyebarannya yang agresif dan potensi kematian yang tinggi jika tidak ditangani sejak dini. Oleh karena itu, proses klasifikasi yang akurat terhadap jenis kanker kulit sangat penting untuk menunjang diagnosis dan penanganan lebih lanjut [5].

Dalam ranah penelitian dermatologi digital, validasi model memerlukan dataset yang representatif secara klinis. Oleh karena itu, penelitian ini menggunakan dataset HAM10000 (*Human Against Machine with 10000 training images*) sebagai data latih dan uji. Dataset yang dipopulerkan oleh Tschandl *et al.* ini terdiri dari 10.015 citra dermoskopi yang dikumpulkan dari berbagai populasi dan modalitas alat, mencakup tujuh kelas diagnostik utama: *melanocytic nevi*, *melanoma*, *benign keratosis*, *basal cell carcinoma*, *actinic keratosis*, *vascular lesions*, dan *dermatofibroma* [6]. Kompleksitas visual dan volume data dalam HAM10000 menjadikannya standar baku emas (*gold standard*) untuk menguji performa arsitektur *Deep Learning*.

Meskipun HAM10000 menjadi standar riset (*benchmark*), penerapan teknologi ini di dunia nyata masih terhambat oleh dua masalah fundamental. Masalah pertama adalah karakteristik data medis yang tersedia secara publik seringkali sangat tidak seimbang (*imbalanced*). Penelitian terbaru oleh Khan *et al.* [7] menyoroti bahwa dataset standar yang populer digunakan dalam riset mempunyai ketidakseimbangan yang sangat besar dikarenakan tingginya variasi jumlah sampel gambar di kelas jinak, yang mencapai rasio tidak proporsional dibandingkan kelas kanker.

Masalah kedua terletak pada keterbatasan arsitektur model tunggal (*Single-Model*). Penelitian terdahulu

umumnya hanya mengandalkan satu jenis arsitektur CNN, padahal setiap arsitektur memiliki keunggulan ekstraksi fitur yang berbeda. Misalnya, VGG16 unggul dalam representasi fitur spasial yang solid [10], DenseNet121 memiliki propagasi fitur yang kuat [11], dan EfficientNetB0 menawarkan efisiensi parameter yang tinggi [12]. Mengandalkan satu model saja berpotensi kehilangan informasi fitur penting lainnya. Selain itu, penggunaan *classifier* standar (*Softmax*) pada layer terakhir CNN seringkali kurang optimal dalam memisahkan kelas pada ruang fitur berdimensi tinggi dibandingkan dengan *Support Vector Machine* (SVM) [13].

II. KAJIAN TEORI

A. Kanker Kulit

Kanker kulit adalah kondisi di mana terjadi proliferasi sel kulit yang abnormal secara tidak terkendali, yang bisa bersifat ganas (melanoma, karsinoma) atau jinak (nevus, lesi jinak lain). Paparan sinar ultraviolet (UV) dari matahari atau sumber buatan, faktor genetik, dan penuaan kulit merupakan faktor risiko utama [14].

B. Dataset Penelitian

HAM10000 memuat total 10.015 citra dengan resolusi 600x450 piksel yang mencakup tujuh jenis lesi kulit. Meskipun memiliki kualitas visual standar medis yang tinggi, dataset ini memiliki karakteristik distribusi kelas yang sangat tidak seimbang (*highly imbalanced*) [6].



GAMBAR 1
(Sample Citra HAM10000)

C. Penanganan Data Tidak Seimbang

Dalam dataset medis, sering ditemukan kondisi di mana jumlah sampel pada satu kelas (kelas mayoritas/negatif) jauh melebihi jumlah sampel pada kelas lain (kelas minoritas/positif). Kondisi ini dikenal sebagai *Class Imbalance*. Pada dataset HAM10000, terdapat ketimpangan signifikan antara kelas *Melanocytic nevi* (mayoritas) dibandingkan dengan *Melanoma* atau karsinoma lainnya (minoritas) [16].

Untuk mengatasi masalah ini, diterapkan teknik penyeimbangan data (*data balancing*). Berikut adalah ilustrasi mekanisme dari metode yang diterapkan:

1. **Undersampling** (Pengurangan Sampel) Teknik ini dilakukan dengan mengurangi jumlah sampel pada kelas mayoritas secara acak hingga mencapai jumlah tertentu.
2. **Random Oversampling** (Duplikasi Sampel) Teknik ini memperbanyak sampel kelas minoritas dengan cara menduplikasi data yang ada (*sampling with replacement*).

D. Data Augmentation

Data Augmentation merupakan strategi krusial dalam pelatihan *Deep Learning*, khususnya pada dataset medis yang sering kali memiliki jumlah sampel terbatas namun variasi visual yang tinggi. Liang *et al.* [17], menunjukkan bahwa metode augmentasi seperti *Mixup* dapat meningkatkan akurasi model dengan mencampurkan dua gambar berbeda untuk menghasilkan citra baru yang lebih realistis, serta memberikan variasi yang lebih beragam dalam proses pelatihan.

E. Model Deep Learning

EfficientNetB0 Dikembangkan oleh Tan dan Le (2019), Kualitas hasil penskalaan model sangat bergantung pada arsitektur dasar (*baseline network*) yang digunakan. Oleh karena itu, Tan dan Le (2019) tidak hanya mengusulkan metode penskalaan, tetapi juga mengembangkan arsitektur dasar baru yang disebut EfficientNet-B0. Pengembangan ini dilakukan menggunakan pendekatan *Multi-objective Neural Architecture Search* (NAS) yang bertujuan untuk mengoptimalkan keseimbangan antara akurasi dan kompleksitas komputasi (*FLOPS*).

Berbeda dengan pendekatan sebelumnya yang berfokus pada latensi perangkat keras spesifik (seperti pada MnasNet), EfficientNet mengoptimalkan fungsi tujuan

$$ACC(m) \times \left[\frac{FLOPS(m)}{T} \right]^W \quad (1)$$

Fungsi ini mencari model yang memiliki akurasi tertinggi namun dengan batasan operasi *floating-point* yang efisien. Hasil pencarian arsitektur tersebut menghasilkan EfficientNet-B0, yang strukturnya didominasi oleh blok *Mobile Inverted Bottleneck Convolution* (MBConv).

Arsitektur kedua yang digunakan dalam penelitian ini adalah *Densely Connected Convolutional Networks* (DenseNet121).

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (2)$$

Dimana $[x_0, x_1, \dots, x_{l-1}]$ merepresentasikan operasi penggabungan (*concatenation*) fitur. Hal ini berbeda dengan ResNet yang menggunakan penjumlahan (*summation*). Struktur Arsitektur DenseNet121 terdiri dari empat Dense Block yang dipisahkan oleh Transition Layer.

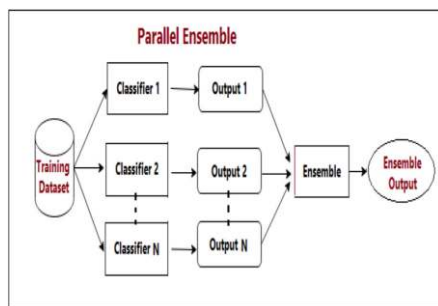
Arsitektur ketiga yang digunakan adalah VGG16, sebuah model *Deep Convolutional Neural Network* klasik yang dikembangkan oleh *Visual Geometry Group*. Dalam penelitian ini, penerapan VGG16 merujuk pada studi yang dilakukan oleh Gulzar *et al.* [18] dalam jurnal *Symmetry*.

Struktur Arsitektur Berdasarkan deskripsi Gulzar *et al.* arsitektur VGG16 terdiri dari 13 lapisan konvolusi (*convolutional layers*) dan 3 lapisan terhubung penuh (*fully connected layers*), yang terbagi menjadi lima blok utama.

F. Parallel Ensemble

Penelitian ini mengadopsi pendekatan *Parallel Ensemble Learning* tipe heterogen, sebuah strategi di mana beberapa model dengan arsitektur berbeda dilatih secara independen untuk menyelesaikan tugas yang sama. Berbeda dengan pendekatan *sequential* (seperti Boosting) yang berfokus memperbaiki error model sebelumnya, strategi paralel bertujuan mengeksplorasi keberagaman (*diversity*) antar-model untuk mengurangi varians prediksi [19].

Berdasarkan tinjauan komprehensif oleh Mohammed dan Kora [19], keberhasilan metode *ensemble* sangat bergantung pada prinsip "Ketidaksepakatan Strategis" (*Strategic Disagreement*). Setiap arsitektur *Deep Learning* (seperti VGG, DenseNet, dan EfficientNet) memiliki bias induktif dan cara ekstraksi fitur yang unik..



GAMBAR 3
(Parallel Ensemble)

Dengan menggabungkan model-model ini secara paralel, sistem dapat menutupi titik buta (*blind spots*) dari satu model tunggal. Mohammed dan Kora, [19] menjelaskan bahwa penggabungan model yang tidak berkorelasi sempurna (*uncorrelated errors*) secara matematis akan menghasilkan batas keputusan (*decision boundary*) yang lebih stabil dan generalis, sehingga mampu mengungguli model tunggal mana pun dalam hal akurasi dan ketahanan (*robustness*) terhadap *noise*.

G. Strategi Penggabungan Fitur (*Feature Fusion*)

Dalam arsitektur *Deep Learning Multi-model*, strategi penggabungan (*fusion*) memegang peranan krusial untuk meningkatkan kinerja klasifikasi. Menurut survei komprehensif oleh Ganaie *et al.* [20] teknik fusi dalam *ensemble learning* secara umum dibagi menjadi tiga kategori utama: fusi tingkat awal (*early fusion*), fusi tingkat keputusan (*decision fusion*), dan fusi tingkat fitur (*feature fusion*)

Penelitian ini menerapkan Fusi Tingkat Fitur (*Feature Fusion*). Berbeda dengan *decision fusion* yang hanya merata-rata probabilitas hasil akhir, *feature fusion* bekerja dengan cara menggabungkan vektor fitur yang diekstraksi dari *layer* terakhir (biasanya *global average pooling* atau *fully connected layer*) sebelum masuk ke tahap klasifikasi akhir.

$$(D_{total}) = D_{Model\ 1} + D_{Model\ 2} + D_{Model\ 3} \quad (3)$$

H. Teknik Konkatensi (*Concatenation*)

Strategi penggabungan fitur (*feature fusion*) dalam penelitian ini dilakukan melalui teknik konkatensi. Konkatensi adalah proses menggabungkan vektor fitur dari berbagai sumber (model) menjadi satu vektor representasi tunggal yang lebih komprehensif tanpa mereduksi informasi yang terkandung di dalamnya..

Secara matematis, jika f_i merepresentasikan vektor fitur dari masing-masing model CNN, maka vektor fitur gabungan F didefinisikan sebagai hasil penggabungan dimensi (*concatenation*) vektor-vektor tersebut dalam ruang fitur. Meskipun Latif *et al.* [21] menggunakan notasi umum $\sum$ (sigma) untuk merepresentasikan agregasi fitur ini, operasi yang sebenarnya dilakukan adalah penyusunan vektor secara sekuensial:

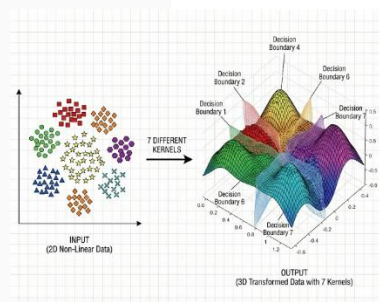
$$F = \text{Concat}(f_{EffNet}, f_{DenseNet}, f_{VGG16}) \quad (4)$$

Atau dalam notasi vektor:

$$F = [f_1, f_2, f_3] \quad (5)$$

I. Hybrid CNN-SVM

Dalam arsitektur *Deep Learning* standar, lapisan klasifikasi biasanya menggunakan fungsi aktivasi *Softmax* yang berbasis probabilitas. Namun, penelitian ini mengusulkan pendekatan *Hybrid* dengan menggantikan lapisan tersebut menggunakan Support Vector Machine (SVM). Strategi ini menggabungkan kekuatan ekstraksi fitur otomatis dari CNN dengan kemampuan klasifikasi margin maksimal dari SVM.



GAMBAR 5
(Ilustrasi konsep Kernel Trick pada SVM Multiclass)

Dalam SVM, pemilihan fungsi kernel sangat krusial. Salah satu kernel yang paling umum digunakan untuk klasifikasi citra medis adalah Radial Basis Function (RBF). Kernel RBF (juga dikenal sebagai kernel Gaussian) memiliki kemampuan untuk menangani hubungan non-linear antara label kelas dan atribut fitur [22].

J. Strategi dan Pendekatan dalam Fine-Tuning

Dalam pengembangan model *deep learning*, terdapat berbagai strategi *fine-tuning* yang dapat diterapkan untuk menyesuaikan model *pre-trained* dengan tugas spesifik.

Berikut adalah tiga pendekatan utama yang dievaluasi berdasarkan literatur terkini:

1. Weighted Fine-Tuning

Secara spesifik, Liu et al. mengimplementasikan *multiscale patch-based weighted fine-tuning*. Teknik ini memungkinkan model untuk lebih fokus pada *patch* atau area lokal yang memiliki informasi krusial, sehingga meningkatkan efektivitas model dalam menangani tugas kompleks seperti deteksi keaslian wajah (*anti-spoofing*) dengan memanfaatkan fusi data multimodal [23].

2. Partial Fine-Tuning

Partial fine-tuning adalah strategi di mana hanya sebagian lapisan (*layers*) dari jaringan saraf yang dilatih ulang, sementara lapisan lainnya dibekukan (*frozen*). Sebuah studi mengenai klasifikasi lesi kulit (*melanoma* dan *nevus*) menggunakan arsitektur AlexNet membandingkan efektivitas antara *full fine-tuning* (melatih seluruh lapisan) dan *partial fine-tuning* (hanya melatih lapisan ke-13 hingga ke-25) [24].

3. Two-Stage Fine-Tuning

Untuk menangani masalah ketidakseimbangan kelas (*class-imbalanced data*) pada model *pre-trained*, ValizadehAslani et al. [25], mengusulkan pendekatan dua tahap (*two-stage fine-tuning*). Metode ini dirancang untuk mempertahankan performa baik pada kelas mayoritas maupun minoritas.

K. Callbacks

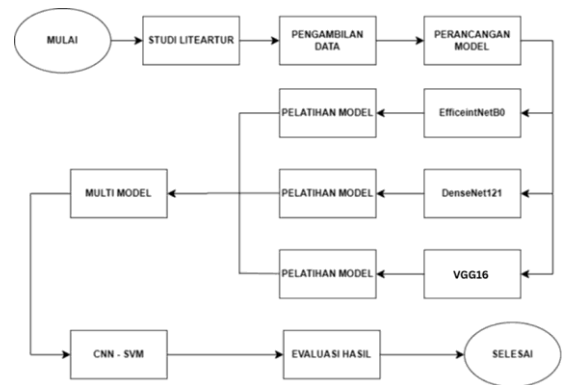
Dalam proses pelatihan ketiga model CNN (EfficientNet, DenseNet, VGG16), penelitian ini menerapkan mekanisme *Callbacks* untuk mengontrol perilaku pelatihan secara otomatis. *Callbacks* bertugas memantau metrik validasi secara *real-time* dan mengambil keputusan strategis untuk mengoptimalkan konvergensi serta mencegah *overfitting*.

L. Evaluasi Model: Akurasi, Precision, Recall, F1-Score, dan Confusion Matrix

Metrik Evaluasi Kinerja Mengingat adanya ketidakseimbangan data pada kasus deteksi kanker kulit, penggunaan metrik Akurasi saja dapat menyesatkan (*misleading*). Oleh karena itu, evaluasi model akan difokuskan pada *Confusion Matrix* dan *Classification Report*, khususnya pada akurasi F1-Score.

III. METODE

Untuk memberikan gambaran sistematis mengenai pelaksanaan penelitian, langkah-langkah yang dilakukan divisualisasikan dalam diagram alir (*flowchart*). Alur ini mencakup tahapan mulai dari studi literatur, pengolahan data, perancangan arsitektur *hybrid*, hingga evaluasi performa model. Secara rinci, diagram alir penelitian disajikan pada Gambar 6.



GAMBAR 6
(Diagram Alir)

Berdasarkan Gambar 6 di atas, berikut adalah penjelasan mendetail mengenai mekanisme teknis pada setiap tahapan penelitian:

A. Pra-pemrosesan dan Pembagian Data

Tahap ini krusial untuk memastikan data siap digunakan oleh model *Deep Learning*. Proses dimulai dengan pengumpulan dataset HAM10000 yang berisi 10.015 citra..

Tahapan ini bertujuan menyiapkan data mentah agar kompatibel dengan arsitektur model. Proses teknis yang dilakukan meliputi:

1. **Resizing:** Citra asli dataset HAM10000 diubah dimensinya (*resize*) menjadi 224 x 224 piksel. Langkah ini dilakukan untuk menyeragamkan input sebelum masuk ke dalam arsitektur CNN (VGG16, DenseNet, EfficientNet).
2. **Penanganan *Imbalance* dan Augmentasi Terintegrasi:** Mengingat adanya ketimpangan kelas yang ekstrem, penelitian ini menerapkan strategi dua lapis (*two-tier strategy*) untuk menyeimbangkan data:

B. Perancangan Arsitektur dan Strategi Pelatihan Model

Penelitian ini mengusulkan arsitektur *Multi-Model Ensemble* yang mengintegrasikan tiga jaringan saraf konvolusional (CNN) berbeda sebagai pengekstraksi fitur. Pemilihan ketiga model ini didasarkan pada karakteristik struktural yang saling melengkapi: VGG16 untuk kedalaman spasial, DenseNet121 untuk efisiensi aliran fitur, dan EfficientNetB0 untuk optimasi parameter.

Untuk memaksimalkan performa pada dataset dermatoskopi yang terbatas, diterapkan strategi pelatihan transfer (*Transfer Learning*) dengan mekanisme *Fine-Tuning* yang spesifik untuk setiap model.

1. Model 1: EfficientNetB0 dengan *Weighted Fine-Tuning*

Untuk mengatasi tantangan ketidakseimbangan kelas dan kemiripan visual yang tinggi antar lesi kulit, penelitian ini menerapkan strategi pelatihan dua fase yang dinamakan '*The Efficient Specialist*'.

Proses pelatihan dibagi menjadi dua tahap krusial:

a. Fase 1 (Feature Extraction)

Model dilatih pada lapisan *head* klasifikasi saja dengan *backbone* yang dibekukan (*frozen*).

```
base_model = tf.keras.applications.EfficientNetB0(
    include_top=False,
    weights="imagenet",
    input_tensor=x_eff
)

base_model.trainable = True
for layer in base_model.layers[:-50]:
    layer.trainable = False

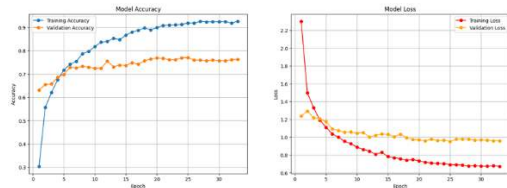
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.BatchNormalization()(x)

x = tf.keras.layers.Dense(256, activation='relu')(x)
x = tf.keras.layers.Dropout(0.5)(x)

outputs = tf.keras.layers.Dense(7, activation='softmax')(x)

model = tf.keras.Model([inputs, outputs])
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss=tf.keras.losses.CategoricalCrossentropy(label_smoothing=0.1),
    metrics=['accuracy', 'auc']
)
```

GAMBAR 6
(Proses Fase 1 Generalisasi EfficientNetB0)



GAMBAR 7
(Plot Garis Fase 1 EfficientNetB0)

Pembekuan layer (*layer freezing*) dilakukan untuk menerapkan strategi Transfer Learning yang efisien.

b. Fase 2 (Weighted Fine-Tuning):

Seluruh lapisan model dibuka (*unfrozen*) untuk memungkinkan model mempelajari detail tekstur kulit yang sangat halus.

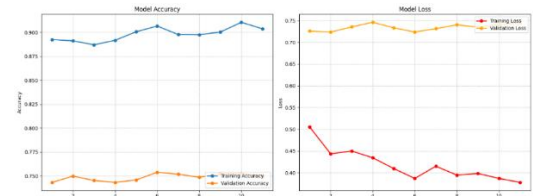
```
print("\nBobot Awal:", class_weights_dict)

class_weights_dict[4] *= 2.0
class_weights_dict[2] *= 1.5
class_weights_dict[1] *= 1.2
class_weights_dict[0] *= 1.2

print("Bobot Baru (Fokus Kanker):", class_weights_dict)

base_model = model.layers[2]
base_model.trainable = True
```

GAMBAR 8
(Proses Fase 2 (*Weighted Fine-Tuning*) EfficientNetB0)



GAMBAR 9
(Plot Garis Training Fase 2 EfficientNetB0)

2. Model 2: DenseNet121 dengan *Partial Fine-Tuning*

Model kedua, DenseNet121, difokuskan untuk memaksimalkan aliran informasi fitur melalui koneksi padat (*dense connections*).

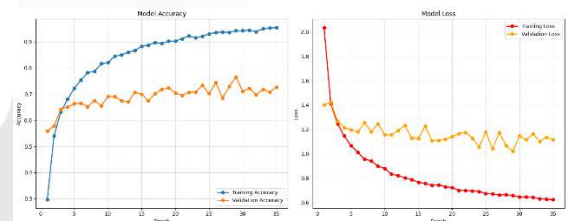
a. *Partial Fine-Tuning*

Strategi pelatihan menerapkan pembekuan selektif, di mana hanya 30 lapisan terakhir yang diizinkan untuk belajar (*trainable*).

```
base_model_2 = tf.keras.applications.DenseNet121(
    include_top=False,
    weights="imagenet",
    input_tensor=x_dense
)

base_model_2.trainable = True
for layer in base_model_2.layers[:-30]:
    layer.trainable = False
```

GAMBAR 9
(Proses Partial Fine Tuning DenseNet 121)



GAMBAR 10
(Plot Garis Training Partial Fine Tuning DenseNet 121)

b. *Classification Head*

```
x = tf.keras.layers.GlobalAveragePooling2D()(base_model_2.output)
x = tf.keras.layers.BatchNormalization()(x)
x = tf.keras.layers.Dense(256, activation='relu')(x)
x = tf.keras.layers.Dropout(0.5)(x)
outputs = tf.keras.layers.Dense(7, activation='softmax')(x)
```

GAMBAR 11
(Proses Classification Head)

3. Model 3: VGG16 dengan *Two-Stage Fine-Tuning*

Model ketiga memanfaatkan arsitektur VGG16 yang dikenal memiliki kedalaman spasial yang kuat untuk mengekstraksi fitur visual.

Berbeda dengan pendekatan sebelumnya, model ini menerapkan strategi pelatihan dua tahap (*Two-Stage Training*) dengan konfigurasi sebagai berikut:

a. Preprocessing & Arsitektur Head

Karena VGG16 dilatih menggunakan standar dataset ImageNet yang berbeda, model ini menerapkan *preprocessing* khusus melalui lapisan Lambda.

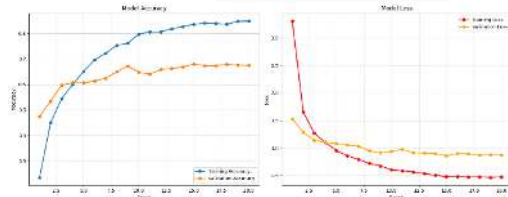
```
base_model = VGG16(include_top=False, weights='imagenet', input_tensor=preprocessed_input)
base_model.trainable = False
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
outputs = Dense(7, activation='softmax')(x)
model = Model(inputs=input_layer, outputs=outputs)
return model, base_model
model, base_model = build_vgg_model()
```

GAMBAR 12
(Preprocessing & Arsitektur Head VGG16)

b. Fase 1 (*Frozen Head Training*): Pada tahap awal, seluruh lapisan ekstraktor fitur (*backbone*) VGG16 dibekukan (*frozen*).

```
checkpoint = ModelCheckpoint('VGG16_skin_best.keras', monitor='val_accuracy', save_best_only=True, mode='max', verbose=1)
early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True, verbose=1)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-6, verbose=1)
history = model.fit(
    x_train, y_train,
    validation_data=(x_val, y_val),
    epochs=50,
    batch_size=32,
    callbacks=[checkpoint, early_stop, reduce_lr])
```

GAMBAR 13
(Fase 1 (Frozen Head Training) VGG16)



GAMBAR 14
(Plot Garis Fase 1 (Frozen Head Training) VGG16)

Pada fase 1 ini, VGG16 dengan Membekukan *Head Training* mendapatkan val_ akurasi sebesar 0.67931.

c. Fase 2 (*Specific Block Tuning*)

Setelah model stabil, dilakukan strategi *Fine-Tuning* selektif. Mekanisme *unfreezing* hanya diterapkan pada Block 5, yang merupakan blok konvolusi terakhir (terdiri dari 4 lapisan teratas).

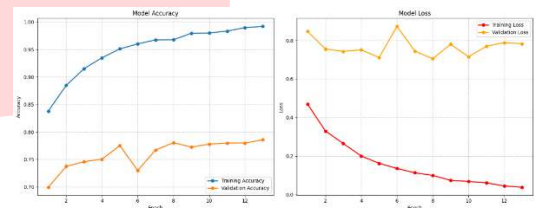
```
base_model.trainable = True

for layer in base_model.layers[:4]:
    layer.trainable = False

model.compile(optimizer=Adam(learning_rate=1e-5),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

print("Melanjutkan training (Fine Tuning)...")
history_fine = model.fit(
    x_train, y_train,
    validation_data=(x_val, y_val),
    epochs=40,
    batch_size=32,
    callbacks=[checkpoint, early_stop, reduce_lr])
```

GAMBAR 15
(Fase 2 (Specific Block Tuning) VGG16)



GAMBAR 16
(Plot Garis Fase 2 (Specific Block Tuning) VGG16)

Pada fase 2 ini, VGG dengan *Specific Block Tuning* mendapatkan angka val_ accuracy optimal di bandingkan fase 1 yaitu sebesar 0,78576. Hasil ini yang akan kita pakai untuk Model VGG16

C. Implementasi Arsitektur *Parallel Ensemble* (Penyatuan Fitur)

Setelah ketiga model CNN (*EfficientNetB0*, *DenseNet121*, *VGG16*) dilatih secara independen hingga mencapai konvergensi optimal, tahap selanjutnya adalah membentuk arsitektur *Parallel Ensemble*.

Mekanisme penyatuan fitur dilakukan melalui langkah-langkah berikut:

1. Ekstraksi Fitur Simultan

Setiap citra dermatoskopi diproses secara paralel oleh ketiga model. Lapisan prediksi probabilitas (*Softmax*) pada setiap model dibuang, sehingga output yang diambil adalah representasi fitur numerik dari lapisan *Global Average Pooling* atau *Fully Connected* terakhir.

TABEL 1
(Fitur Setiap Model)

Model	Lapisan Ekstraksi	Fitur
EfficientNetB0	Global Average Pooling	1280
DenseNet121	Global Average Pooling	1024

VGG16	Block5_Pool / Global Average Pooling	512
Total Gabungan	Concatenated Vektor	2.816

2. Konkatenasi Fitur (*Feature Fusion*)

Ketiga vektor fitur independen tersebut kemudian disatukan menggunakan teknik *Concatenation*. Teknik ini menggabungkan informasi visual yang beragam. Mulai dari efisiensi parameter, detail tekstur, hingga kedalaman spasial menjadi satu kesatuan representasi yang utuh.

Secara matematis, dimensi vektor gabungan (D_{total}) dihitung sebagai:

$$(D_{total}) = D_{EffNet} + D_{DenseNet} + D_{VGG16} \quad (10)$$

$$D_{total} = 1280 + 1024 + 512 = 2.816 \text{ fitur}$$

Vektor gabungan inilah yang disebut sebagai "*Super Vector*", yang memuat informasi komprehensif mengenai karakteristik lesi kulit.

```
input_tensor = Input(shape=(224, 224, 3))
feat_group1 = eff_extractor(input_tensor)
feat_group2 = dense_extractor(input_tensor)
feat_group3 = vgg_extractor(input_tensor)

final_features = Concatenate(name="concatenated_trio")([feat_group1, feat_group2])
ultimate_extractor = Model(inputs=input_tensor, outputs=final_features)
```

GAMBAR 17
(Konkatenasi Fitur Model)

D. Klasifikasi Hybrid Menggunakan Support Vector Machine (SVM)

Tahap akhir dari metodologi ini menerapkan pendekatan Hybrid CNN-SVM. Pendekatan ini menggantikan lapisan klasifikasi standar (*Softmax*) dengan algoritma *Machine Learning* konvensional, yaitu *Support Vector Machine* (SVM).

Alasan pemilihan SVM sebagai klasifikator final didasarkan pada prinsip *Structural Risk Minimization* (SRM). Berbeda dengan *Softmax* yang meminimalkan *empirical risk* (error pada data latih), SVM bertujuan meminimalkan batas atas kesalahan generalisasi, sehingga lebih stabil dalam menangani data berdimensi tinggi [26].

Pada tahap ini, sistem tidak lagi bekerja dengan data piksel mentah, melainkan beroperasi pada representasi fitur tingkat tinggi yang dihasilkan oleh arsitektur *Parallel Ensemble*. Fokus utama beralih dari pembelajaran fitur visual (yang dilakukan CNN) ke pemetaan keputusan diagnosis yang optimal..

1. Mekanisme Penangkapan dan Pemetaan Fitur

Setelah proses ekstraksi fitur selesai, SVM menerima input berupa "*Super Vector*" berdimensi 2.816. Mekanisme penangkapan fitur oleh SVM dilakukan melalui prinsip *Kernel Mapping* dan *Margin Maximization*:

- Transformasi Ruang Fitur (*Feature Space Transformation*)
- Pembentukan *Decision Boundary*

```
from sklearn.svm import SVC
svm_clf = SVC(kernel='rbf', probability=True, decision_function_shape='ovo', random_state=2)
svm_clf.fit(X_features, y_labels)

print("SVM Selesai Dilatih")

# Cek performa Training
print("===== HASIL TRAINING (SELF CHECK) =====")
predictions = svm_clf.predict(X_features)
print(classification_report(y_labels, predictions, target_names=generator.class_indices.keys()))
```

GAMBAR 18
(Proses Train SVM)

2. Mekanisme Inferensi (Pengujian)

Pada saat pengujian data baru atau implementasi sistem, alur kerja berjalan secara *feed-forward*:

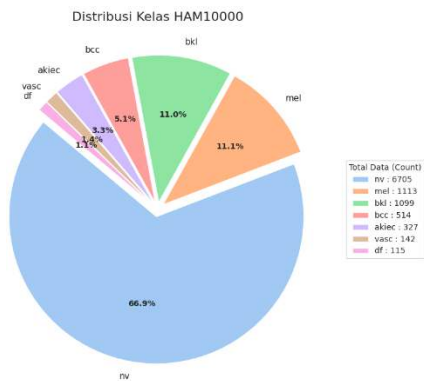
- Citra baru diproses oleh CNN (yang sudah *frozen*) untuk menghasilkan vektor fitur gabungan.
- Vektor tersebut dimasukkan ke dalam model SVM yang telah dilatih (disimpan dalam format *.joblib*).
- SVM menghitung posisi vektor terhadap *hyperplane*. Berdasarkan posisi tersebut, SVM memberikan output final berupa label kelas diagnosis (salah satu dari 7 kategori lesi kulit).

Dengan mekanisme ini, sistem menggabungkan keunggulan CNN dalam mengekstraksi detail visual (tekstur, bentuk, warna) dengan keunggulan matematis SVM dalam menentukan batas keputusan yang tegas.

IV. HASIL DAN PEMBAHASAN

A. Pra-pemrosesan Data (*Preprocessing*) dan Strategi Penyeimbangan Kelas

Data awal yang diperoleh dari dataset HAM10000 menunjukkan ketidakseimbangan kelas yang signifikan (*imbalanced dataset*). Sebagaimana terlihat pada Gambar 4.1, kelas *Melanocytic nevi* (nv) mendominasi distribusi populasi data hingga mencapai 66,9%. sementara kelas lain seperti *Dermatofibroma* (df) dan *Vascular lesions* (vasc) memiliki representasi yang sangat minim (kurang dari 1,5%).



GAMBAR 17
(Distribusi Kelas HAM10000)

Ketimpangan distribusi ini berpotensi menyebabkan bias mayoritas, di mana model akan cenderung memprediksi kelas dominan dan gagal mengenali fitur pada kelas kanker kulit yang kritis.

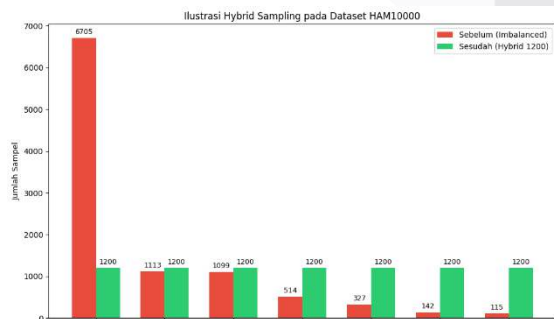
257/257	0.706	3s/step	- accuracy: 0.6678 - auc: 0.8638 - loss: 1.3834 - recall: 0.0031 - val_accuracy: 0.6694
epoch 7/25			
257/257	0.736	3s/step	- accuracy: 0.6880 - auc: 0.8678 - loss: 1.3577 - recall: 0.0059 - val_accuracy: 0.6694
epoch 8/25			
257/257	0.306	3s/step	- accuracy: 0.6688 - auc: 0.8635 - loss: 1.3951 - recall: 0.0088 - val_accuracy: 0.6694
epoch 9/25			
257/257	0.736	3s/step	- accuracy: 0.6738 - auc: 0.8648 - loss: 1.3851 - recall: 0.0125 - val_accuracy: 0.6694
epoch 10/25			
257/257	0.742	3s/step	- accuracy: 0.6744 - auc: 0.8669 - loss: 1.3709 - recall: 0.0147 - val_accuracy: 0.6694
epoch 11/25			
257/257	0.826	3s/step	- accuracy: 0.6745 - auc: 0.8673 - loss: 1.3669 - recall: 0.0168 - val_accuracy: 0.6694
epoch 12/25			
257/257	0.582	3s/step	- accuracy: 0.6680 - auc: 0.8637 - loss: 1.3995 - recall: 0.0180 - val_accuracy: 0.6694
epoch 13/25			
257/257	0.706	3s/step	- accuracy: 0.6761 - auc: 0.8645 - loss: 1.3824 - recall: 0.0211 - val_accuracy: 0.6694
epoch 14/25			
257/257	0.812	3s/step	- accuracy: 0.6788 - auc: 0.8671 - loss: 1.3852 - recall: 0.0195 - val_accuracy: 0.6694

GAMBAR 18
(Proses train tanpa Hybrid Sampling)

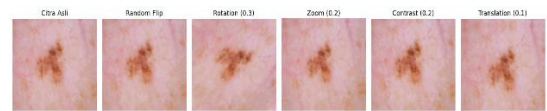
Terlihat di Gambar 18, bahwa training model tanpa *Hybrid Sampling* akan menyebabkan overfitting dan model gagal membedakan fitur minoritas (bias mayoritas). Hasilnya *val Accuracy* stagnan di 0.6694 atau 66.94% yang berarti model asal menebak kelas mayoritas (nv).

Strategi ini dilakukan melalui tiga tahapan terintegrasi untuk menjamin keseimbangan statistik sekaligus mencegah overfitting:

1. *Random Undersampling*: Diterapkan pada kelas mayoritas (nv) dengan mengambil sampel secara acak tanpa pengembalian hingga jumlahnya turun menjadi 1.200.
2. *Random Oversampling*: Diterapkan pada enam kelas minoritas lainnya. Sampel diduplikasi secara acak (*random replication*) hingga mencapai target 1.200 citra. Bukti kuantitatif hasil penyeimbangan ini dapat dilihat pada kolom "Jumlah Data Latih" dalam Tabel 2



GAMBAR 19
(Ilustrasi Hybrid Sampling)



GAMBAR 20
(Augmentasi pada dataset)

Penting untuk dicatat bahwa Data Validasi (15%) dipisahkan sebelum proses balancing. Hal ini dilakukan untuk menjaga integritas pengujian, sehingga model dievaluasi pada distribusi data yang alami (murni) dan tidak bias. Distribusi data dapat dilihat pada Tabel 2.

TABEL 2
(Distribusi Data Latih dan Validasi)

Kelas Diagnostik	Kode Label	Jumlah Awal (Raw)	Jumlah Data Latih (Setelah Balancing)	Jumlah Data Validasi (Murni)
Actinic keratoses	Akiec	327	1200	49
Basal cell carcinoma	Bcc	514	1200	77
Benign keratosis-like	Bkl	1098	1200	165
Dermatofibroma	Df	115	1200	17
Melanoma	Mel	1112	1200	167
Melanocytic nevi	Nv	6705	1200	1006
Vascular lesions	Vasc	142	1200	22

Selain penyeimbangan, seluruh citra diubah ukurannya (*resize*) menjadi dimensi 224 x 224 piksel untuk menyesuaikan dengan *input layer* dari arsitektur CNN yang digunakan.

B. Performa Training Model Dasar (Baseline)

Sebelum digabungkan menjadi ensemble, performa masing-masing model dievaluasi secara individual. Hasil pelatihan disajikan pada Tabel 3.

TABEL 3
(Train Dasar)

Model	Accuracy	Loss	Val Accuracy	Val Loss
EfficientNetB0	89,76%	39,45%	74,85%	95,82%
DenseNet121	93,51%	66,92%	74,38%	104,08%
VGG16	99,20%	3,80%	78,58%	78,28%

C. Implementasi Hybrid Ensemble CNN-SVM

Setelah tahap ekstraksi fitur selesai, dilakukan penggabungan (*concatenation*) vektor fitur dari ketiga arsitektur CNN. Berdasarkan skema rancangan sistem yang telah dipaparkan pada Diagram Alir Penelitian (Sub-bab 3.2), output dari

setiap model tidak langsung diklasifikasikan, melainkan disatukan untuk membentuk representasi fitur yang komprehensif. Ringkasan arsitektur model hibrida yang terbentuk disajikan pada Tabel 4.

TABEL 4
Ringkasan Model Hibrida (Model Summary)

Layer (type)	Nama Komponen (Deskripsi)	Output Shape	Parameter	Connected to
input_layer (InputLayer)	Input Citra	(None, 224, 224, 3)	0	-
functional (Functional)	EffNetB0	(None, 128, 0)	4,382,627	input_layer [0][0]
functional_1 (Functional)	DenseNet121	(None, 1024)	7,304,000	input_layer [0][0]
functional_2 (Functional)	VGG16	(None, 512)	14,714,688	input_layer [0][0]
Concatenate_d_trio (Concatenate)	Feature Fusion Layer	(None, 2816)	25,801,763	functional_2 [0][0] functional_1 [0][0] functional [0][0]

Berdasarkan Tabel 4, proses pembentukan vektor fitur gabungan (*Super Vector*) terbentuk dari penjumlahan dimensi output *Global Average Pooling* murni dari ketiga model dasar:

1. EfficientNetB0: Menyumbang 1.280 fitur.
2. DenseNet121: Menyumbang 1.024 fitur.
3. VGG16: Menyumbang 512 fitur.

Total dimensi fitur yang masuk ke SVM adalah:

$Dimensi_{Total} = 1.280 + 1.024 + 512 = 2.816$ Fitur
Alasan Pemilihan Kernel RBF (*Radial Basis Function*) Vektor fitur berdimensi 2.816 ini kemudian diklasifikasikan menggunakan algoritma SVM. Sesuai dengan spesifikasi parameter yang ditetapkan pada Metodologi Penelitian, Dengan Kernel RBF, data dipetakan ke ruang dimensi tak hingga (*infinite-dimensional space*) sehingga SVM dapat membentuk batas keputusan (*decision boundary*) yang melengkung dan fleksibel mengikuti distribusi data yang rumit. Hal ini menjamin akurasi yang lebih tinggi dibandingkan pendekatan linear.

D. Evaluasi Kinerja dan Analisis Komparatifasil Evaluasi Model

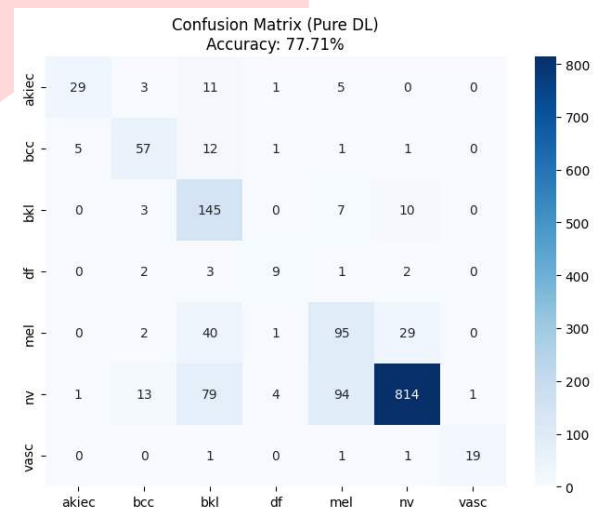
Evaluasi dilakukan menggunakan Data Validasi (1.503 citra) yang tidak terlibat dalam proses pelatihan (*training*) maupun penyeimbangan (*resampling*). Data ini dijaga kemurniannya

untuk mensimulasikan kondisi pengujian dunia nyata (*real-world*), di mana model dihadapkan pada distribusi data alami. Evaluasi ini terbagi menjadi dua tahap: analisis performa model Hybrid SVM dan komparasi langsung terhadap model dasar (*baseline*) tanpa SVM.

1. Hasil Pengujian Model Baseline Softmax

Sebelum menerapkan SVM sebagai klasifikator akhir, pengujian dilakukan menggunakan klasifikator standar *Softmax* (Fully Connected Layer). Model ini menggunakan arsitektur gabungan (*Ensemble*) dari VGG16, DenseNet121, dan EfficientNetB0, namun lapisan akhirnya menggunakan fungsi aktivasi probabilitas standar.

Pengujian ini bertujuan untuk mengetahui batas kemampuan (*baseline*) metode Deep Learning murni tanpa bantuan *Machine Learning* konvensional. Hasil prediksi pada 1.503 data validasi divisualisasikan pada Gambar 21



GAMBAR 21
(Confusion Matrix Model Baseline Softmax)

Berdasarkan Gambar 21 terlihat bahwa meskipun model mampu mengenali kelas mayoritas (*Nevus*) dengan baik, terdapat kelemahan signifikan dalam membedakan kelas kanker. Secara kuantitatif, performa model dirangkum dalam Tabel 5.

TABEL 5
(Laporan Klasifikasi Model Baseline (Ensemble CNN - Softmax))

	precision	recall	f1-score	support
Akiec	0.83	0.59	0.69	49
Bcc	0.71	0.74	0.73	77
Bkl	0.50	0.88	0.64	165
Df	0.56	0.53	0.55	17
Mel	0.47	0.57	0.51	167
Nv	0.95	0.81	0.87	1006
Vasc	0.95	0.86	0.90	22
Accuracy			0.78	1503
Macro avg	0.71	0.71	0.70	1503
Weighted avg	0.83	0.78	0.79	1503

Analisis Kinerja Model Baseline: Berdasarkan hasil di atas, ditemukan beberapa kelemahan fundamental pada penggunaan klasifikator *Softmax*:

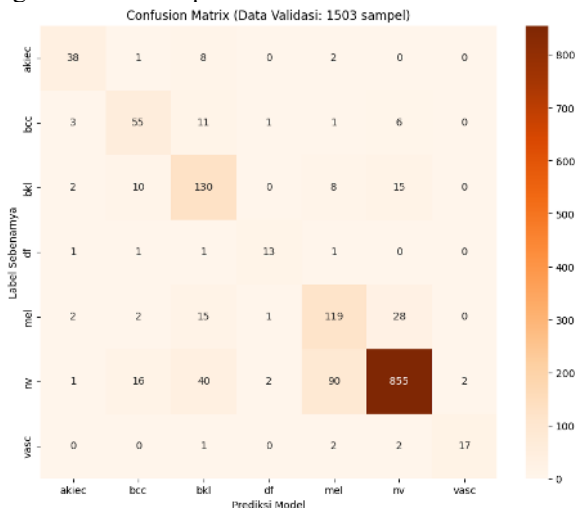
1. Akurasi Stagnan: Model *Softmax* hanya mampu mencapai akurasi 78%. Angka ini menunjukkan bahwa lapisan *Dense* standar kesulitan menarik garis pemisah yang tegas pada fitur gabungan yang berdimensi tinggi.
2. Ketimpangan Akurasi per Kelas: Meskipun akurasi rata-rata tertolong oleh kelas mayoritas (*Nevus*), akurasi pada kelas vital seperti Melanoma sangat rendah (terlihat dari *F1-Score* 0.51). Model gagal mengenali hampir separuh dari kasus kanker kulit ganas. Ini membuktikan bahwa akurasi 78% tersebut bersifat semu (bias ke data mayoritas) dan tidak mencerminkan keandalan model yang sesungguhnya.

2. Hasil Pengujian Model Hybrid CNN - SVM

Untuk mengatasi kelemahan pada model *baseline*, penelitian ini mengganti lapisan klasifikasi akhir dengan *Support Vector Machine* (SVM) menggunakan kernel RBF. Input untuk SVM adalah vektor fitur yang sama persis dengan yang digunakan pada model *baseline*.

1. Confusion Matrix Model Hybrid CNN - SVM

Hasil pengujian pada data validasi yang sama (1.503 citra) menunjukkan perubahan distribusi prediksi yang signifikan, sebagaimana terlihat pada Gambar 22.



GAMBAR 22
(Confusion Matrix Model Hybrid CNN – SVM)

Gambar 22 mengilustrasikan peta distribusi prediksi model *Hybrid CNN-SVM* terhadap label diagnosis sebenarnya. Visualisasi ini menjadi landasan utama dalam membedah komponen pembentuk akurasi model.

Laporan Klasifikasi (*Classification Report*)

Rincian performa statistik untuk setiap kelas disajikan pada Tabel 6. Format tabel telah disesuaikan untuk memisahkan metrik per-kelas dan metrik rata-rata global agar interpretasi lebih jelas.

TABEL 6
(Laporan Klasifikasi)

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>Akiec</i>	0.81	0.78	0.79	49
<i>Bcc</i>	0.65	0.71	0.68	77
<i>Bkl</i>	0.63	0.79	0.70	165
<i>Df</i>	0.76	0.76	0.76	17
<i>Mel</i>	0.53	0.71	0.61	167
<i>Nv</i>	0.94	0.85	0.89	1006

<i>Vasc</i>	0.89	0.77	0.83	22
<i>Accuracy</i>			0.82	1503
<i>Macro avg</i>	0.75	0.77	0.75	1503
<i>Weighted avg</i>	0.84	0.82	0.82	1503

Analisis Implementasi SVM terbukti memberikan dampak positif terhadap kinerja sistem diagnosis:

Meskipun CNN konvensional memberikan hasil yang cukup baik, analisis lebih lanjut menunjukkan bahwa integrasi SVM berhasil meningkatkan stabilitas klasifikasi secara signifikan dibandingkan penggunaan CNN konvensional. Peningkatan ini terjadi karena penggunaan kernel RBF yang efektif membantu memisahkan fitur-fitur yang berhimpitan antar kelas lesi kulit yang memiliki kemiripan visual.

V. KESIMPULAN

Berdasarkan hasil analisis dan pengujian sistem klasifikasi kanker kulit menggunakan metode *Hybrid Learning*, dapat ditarik kesimpulan sebagai berikut:

1. Efektivitas Penyeimbangan Data: Penerapan teknik penyeimbangan data (*Hybrid Sampling*) pada dataset HAM10000 terbukti efektif dalam mengatasi bias mayoritas.
2. Metode Hybrid Ensemble-SVM: Penggabungan ekstraksi fitur dari tiga arsitektur CNN (*EfficientNet*, *DenseNet*, *VGG16*) dengan klasifikasi SVM terbukti menghasilkan performa yang handal.

REFERENSI

[1] S. Handayani Pusadan, A. Makahrnun, M. Yazdi, and Z. Sakka, "Pattern Recognition untuk Klasifikasi Penyakit Kanker Kulit menggunakan Artificial Intelligence (AI)," 2025. [Online]. Available: <https://jurnal.unw.ac.id/index.php/IKN>

[2] H. Sung *et al.*, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA Cancer J. Clin.*, vol. 71, no. 3, pp. 209–249, May 2021, doi: 10.3322/caac.21660.

[3] International Agency for Research on Cance, "Skin cancer," <https://www.iarc.who.int/cancer-type/skin-cancer/>.

[4] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, 2017, doi: 10.1038/nature21056.

[5] A. De, N. Mishra, and H. T. Chang, "An approach to the dermatological classification of histopathological skin images using a hybridized CNN-DenseNet model," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/peerj-cs.1884.

[6] P. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018.

[7] A. Bibi *et al.*, "Skin lesion segmentation and classification using conventional and deep learning based framework," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 2477–2495, 2022, doi: 10.32604/cmc.2022.018917.

[8] A. Al-Rasheed, A. Ksibi, M. Ayadi, A. I. A. Alzahrani, M. Zakariah, and N. Ali Hakami, "An Ensemble of

- Transfer Learning Models for the Prediction of Skin Cancers with Conditional Generative Adversarial Networks,” *Diagnostics*, vol. 12, no. 12, Dec. 2022, doi: 10.3390/diagnostics12123145.
- [9] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, and G. P. Figueredo, “Deep learning with dynamically weighted loss function for sensor-based prognostics and health management,” *Sensors (Switzerland)*, vol. 20, no. 3, Feb. 2020, doi: 10.3390/s20030723.
- [10] R. Singh *et al.*, “FasNet: a hybrid deep learning model with attention mechanisms and uncertainty estimation for liver tumor segmentation on LiTS17,” *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-98427-9.
- [11] Z. S. Kahhoul *et al.*, “Mathematical Analysis and Performance Evaluation of CBAM-DenseNet121 for Speech Emotion Recognition Using the CREMA-D Dataset,” *Applied Sciences (Switzerland)*, vol. 15, no. 17, 2025, doi: 10.3390/app15179692.
- [12] R. Rani, S. Bharany, D. H. Elkamchouchi, A. Ur Rehman, R. Singh, and S. Hussien, “VGG-EffAttnNet: Hybrid Deep Learning Model for Automated Chili Plant Disease Classification Using VGG16 and EfficientNetB0 With Attention Mechanism,” *Food Sci. Nutr.*, vol. 13, no. 7, 2025, doi: 10.1002/fsn3.70653.
- [13] N. E. W. Nugroho and A. Harjoko, “Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 221, Jul. 2021, doi: 10.22146/ijccs.66062.
- [14] S. Tansil Tan, M. Ghaznawie, and G. Reginata, “Deteksi Dini Karsinoma Sel Basal.”
- [15] E. H. Houssein, D. A. Abdelkareem, G. Hu, M. A. Hameed, I. A. Ibrahim, and M. Younan, “An effective multiclass skin cancer classification approach based on deep convolutional neural network,” *Cluster Comput.*, Dec. 2024, doi: 10.1007/s10586-024-04540-1.
- [16] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, “The class imbalance problem in deep learning,” *Mach. Learn.*, vol. 113, no. 7, 2024, doi: 10.1007/s10994-022-06268-8.
- [17] W. Liang, Y. Liang, and J. Jia, “MiAMix: Enhancing Image Classification through a Multi-Stage Augmented Mixed Sample Data Augmentation Method,” *Processes*, vol. 11, no. 12, 2023, doi: 10.3390/pr11123284.
- [18] Y. Gulzar, Y. Hamid, A. B. Soomro, A. A. Alwan, and L. Journaux, “A convolution neural network-based seed classification system,” *Symmetry (Basel)*, vol. 12, no. 12, pp. 1–18, Dec. 2020, doi: 10.3390/sym12122018.
- [19] A. Mohammed and R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” 2023. doi: 10.1016/j.jksuci.2023.01.014.
- [20] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, “Engineering Applications of Artificial Intelligence Ensemble deep learning : A review,” *Eng. Appl. Artif. Intell.*, vol. 115, no. March, 2022.
- [21] M. A. Latif *et al.*, “Enhanced Classification of Coffee Leaf Biotic Stress by Synergizing Feature Concatenation and Dimensionality Reduction,” *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3314590.
- [22] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, “A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images,” *IRBM*, vol. 43, no. 4, 2022, doi: 10.1016/j.irbm.2021.06.003.
- [23] W. Liu, X. Wei, T. Lei, X. Wang, H. Meng, and A. K. Nandi, “Data-Fusion-Based Two-Stage Cascade Framework for Multimodality Face Anti-Spoofing,” *IEEE Trans. Cogn. Dev. Syst.*, vol. 14, no. 2, 2022, doi: 10.1109/TCDS.2021.3064679.
- [24] S. O. Ngesthi, I. Setyawan, and I. K. Timotius, “The Effect of Partial Fine Tuning on AlexNet for Skin Lesions Classification,” in *2021 13th International Conference on Information Technology and Electrical Engineering, ICITEE 2021*, 2021. doi: 10.1109/ICITEE53064.2021.9611885.
- [25] T. ValizadehAslani *et al.*, “Two-stage fine-tuning with ChatGPT data augmentation for learning class-imbalanced data,” *Neurocomputing*, vol. 592, 2024, doi: 10.1016/j.neucom.2024.127801.
- [26] S. Lin and W. Lin, “Research on the Mental Health Status of College Students Based on Machine Learning,” in *2025 5th International Symposium on Computer Technology and Information Science, ISCTIS 2025*, 2025. doi: 10.1109/ISCTIS65944.2025.11064987.