

APPENDICES

Appendix 1. Interview Guidelines
 Resource Person : Mr Made
 Date : 19 Dec 2024
 Location : Clandys jalan Dewi Sartika

Table 1.
 Interview Results with Mr. Made

Pertanyaan	Jawaban
Sejak kapan Anda mulai membeli atau menjual melon?	Sudah lama, melon dibeli lagi saat sudah hampir habis stoknya.
Apa yang Anda perhatikan pertama kali ketika melihat melon hijau dan oranye?	Saya belah terlebih dahulu.
Bagaimana Anda mendeskripsikan perbedaan warna antara melon hijau dan oranye?	Dilihat dari daging buahnya sudah terlihat.
Apakah ada perbedaan dalam tekstur kulit antara kedua jenis melon tersebut? Jika ya, bisa Anda jelaskan?	Saya kurang bisa, tapi biasanya ada ddi bentuk buahnya.
Apakah ada metode yang Anda gunakan untuk membedakan melon hijau dan oranye tanpa membelahnya?	Tidak ada, saya harus membelahnya dulu
Apakah Anda menggunakan alat atau teknologi tertentu untuk membantu dalam identifikasi melon? Jika ya, alat apa yang Anda gunakan?	Tidak ada.
Apakah Anda pernah mengalami kesulitan dalam membedakan melon hijau dan oranye? Jika ya, apakah bisa Anda ceritakan pengalaman tersebut?	Saya pribadi sulit membedakan.
Dalam situasi apa Anda merasa sulit untuk membedakan kedua jenis melon ini?	Karena kemiripan.

Pertanyaan	Jawaban
Apakah Anda menggunakan indera penciuman atau rasa untuk membantu membedakan melon? Jika ya, bagaimana cara Anda melakukannya?	Tidak.
Apa tanggapan yang sering Anda terima dari pelanggan mengenai kesulitan mereka dalam membedakan melon hijau dan oranye	Tidak ada.
Apakah Anda pernah mendengar keluhan dari pelanggan tentang kesalahan dalam memilih jenis melon?	Ini pertanyaan untuk supplier buah
Apa saran Anda untuk konsumen yang ingin lebih baik dalam membedakan melon hijau dan oranye?	Ini pertanyaan untuk supplier buah
Apakah Anda berharap ada teknologi atau metode baru yang dapat membantu dalam klasifikasi melon?	Iya.
Apakah Anda memiliki pengetahuan atau informasi lain yang ingin Anda bagikan terkait dengan melon hijau dan oranye?	Tidak ada.
Adakah sumber informasi (buku, artikel, atau pelatihan) yang Anda rekomendasikan untuk memahami lebih lanjut tentang melon?	Tidak ada.

Resource Person : Mrs. Novi
 Date : 19 December 2024
 Location : Alfamart near Singaraja 6th Junior High School

Table 2.
 Interview Results with Mrs. Novi

Pertanyaan	Jawaban
Sejak kapan Anda mulai membeli atau menjual melon?	Sudah lama, tiap hari selalu ada yang baru.
Apa yang Anda perhatikan pertama kali ketika melihat melon hijau dan oranye?	Saya belah terlebih dahulu.
Bagaimana Anda mendeskripsikan perbedaan warna antara melon hijau dan oranye?	Dilihat dari daging buahnya sudah terlihat.
Apakah ada perbedaan dalam tekstur kulit antara kedua jenis melon tersebut? Jika ya, bisa Anda jelaskan?	Saya bisa, ada perbedaan sedikit jika diteliti.
Apakah ada metode yang Anda gunakan untuk membedakan melon hijau dan oranye tanpa membelahnya?	Tidak ada metode khusus, mungkin karena saya sudah terbiasa
Apakah Anda menggunakan alat atau teknologi tertentu untuk membantu dalam identifikasi melon? Jika ya, alat apa yang Anda gunakan?	Tidak ada.
Apakah Anda pernah mengalami kesulitan dalam membedakan melon hijau dan oranye? Jika ya, apakah bisa Anda ceritakan pengalaman tersebut?	Awal-awal sulit, namun karena sudah terbiasa, jadi bisa membedakan
Dalam situasi apa Anda merasa sulit untuk membedakan kedua jenis melon ini?	Keduanya banyak kemiripan.

Pertanyaan	Jawaban
Apakah Anda menggunakan indera penciuman atau rasa untuk membantu membedakan melon? Jika ya, bagaimana cara Anda melakukannya?	Tidak.
Apa tanggapan yang sering Anda terima dari pelanggan mengenai kesulitan mereka dalam membedakan melon hijau dan oranye	Tidak ada.
Apakah Anda pernah mendengar keluhan dari pelanggan tentang kesalahan dalam memilih jenis melon?	Ini pertanyaan untuk supplier buah
Apa saran Anda untuk konsumen yang ingin lebih baik dalam membedakan melon hijau dan oranye?	Ini pertanyaan untuk supplier buah
Apakah Anda berharap ada teknologi atau metode baru yang dapat membantu dalam klasifikasi melon?	Iya.
Apakah Anda memiliki pengetahuan atau informasi lain yang ingin Anda bagikan terkait dengan melon hijau dan oranye?	Tidak ada.
Adakah sumber informasi (buku, artikel, atau pelatihan) yang Anda rekomendasikan untuk memahami lebih lanjut tentang melon?	Tidak ada.

Resource Person : Mr. Ketut Suastrawan
 Date : 19 December 2024
 Location : Arta Mart Banyuning

Table 3.
 Results of Interview with Mr. Ketut

Pertanyaan	Jawaban
Sejak kapan Anda mulai membeli atau menjual melon?	Sudah lama, tiap hari saya ke pasar.
Apa yang Anda perhatikan pertama kali ketika melihat melon hijau dan oranye?	Dari dagangnya sudah memilah, diberi tanda.
Bagaimana Anda mendeskripsikan perbedaan warna antara melon hijau dan oranye?	Dilihat dari daging buahnya sudah terlihat.
Apakah ada perbedaan dalam tekstur kulit antara kedua jenis melon tersebut? Jika ya, bisa Anda jelaskan?	Saya tidak bisa.
Apakah ada metode yang Anda gunakan untuk membedakan melon hijau dan oranye tanpa membelahnya?	Tidak ada, saya harus membelahnya dulu
Apakah Anda menggunakan alat atau teknologi tertentu untuk membantu dalam identifikasi melon? Jika ya, alat apa yang Anda gunakan?	Tidak ada.
Apakah Anda pernah mengalami kesulitan dalam membedakan melon hijau dan oranye? Jika ya, apakah bisa Anda ceritakan pengalaman tersebut?	Sulit, saya tidak bisa membedakan. Karena sudah ada langganan jadi saya percaya padanya.
Dalam situasi apa Anda merasa sulit untuk membedakan kedua jenis melon ini?	Karena kemiripan.

Pertanyaan	Jawaban
Apakah Anda menggunakan indera penciuman atau rasa untuk membantu membedakan melon? Jika ya, bagaimana cara Anda melakukannya?	Tidak.
Apa tanggapan yang sering Anda terima dari pelanggan mengenai kesulitan mereka dalam membedakan melon hijau dan oranye	Tidak ada.
Apakah Anda pernah mendengar keluhan dari pelanggan tentang kesalahan dalam memilih jenis melon?	Ini pertanyaan untuk supplier buah
Apakah saran Anda untuk konsumen yang ingin lebih baik dalam membedakan melon hijau dan oranye?	Ini pertanyaan untuk supplier buah
Apakah Anda berharap ada teknologi atau metode baru yang dapat membantu dalam klasifikasi melon?	Iya.
Apakah Anda memiliki pengetahuan atau informasi lain yang ingin Anda bagikan terkait dengan melon hijau dan oranye?	Tidak ada.
Adakah sumber informasi (buku, artikel, atau pelatihan) yang Anda rekomendasikan untuk memahami lebih lanjut tentang melon?	Tidak ada.

Appendix 2.Documentation During Interview



Figure 1.
Interview Documentation with
Mrs. Novi



Figure 2.
Documentation During Interview
with Mrs. Novi



Figure 3.
Interview Documentation with Mr. Ketut



Figure 4.
Interview Documentation with Mr. Ketut

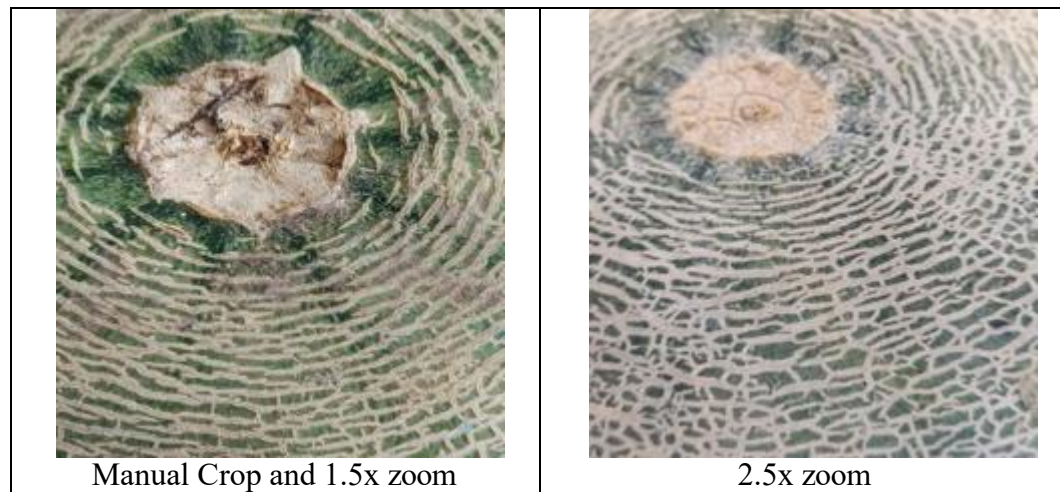


Figure 5.
Interview Documentation with Mr.
Made



Figure 6.
Interview Documentation with Mr. Made

Appendix 3. Sample Dataset



Appendix 4. Source Code Zoom and Crop

```
# 2. Setup Path
source_path = '/content/drive/MyDrive/Datatest_New' # Path dataset mentah
output_path = '/content/Zoomed_Resized_Datatest_Melon' # Folder untuk hasil
target_size = (224, 224) # Ukuran akhir yang diinginkan

# 3. Buat folder output
os.makedirs(output_path, exist_ok=True)

# ganti zoom factornya jadi 1.5 untuk zoom 1.5x
# 4. Fungsi untuk zoom 2x + resize ke 224x224
def process_image(image_path, output_path, zoom_factor=2.5):
    try:
        img = Image.open(image_path).convert('RGB')

        # Langkah 1: Zoom 2.5x (perbesar gambar)
        width, height = img.size
        img = img.resize((int(width * zoom_factor), int(height * zoom_factor)))

        # Langkah 2: Crop bagian tengah
        left = (img.width - width) // 2
        top = (img.height - height) // 2
        right = left + width
        bottom = top + height
        img = img.crop((left, top, right, bottom))

        # Langkah 3: Resize ke 224x224
        img = img.resize(target_size)

        # Simpan gambar
        img.save(output_path)
        return True
```

```

except Exception as e:
    print(f'Error processing {image_path}: {str(e)}')
    return False

```

Appendix 5. Source Code Model Architecture

```

# --- MODEL UTAMA: Pure VGG16 (Transfer Learning) ---
def build_model_vgg16():
    base = VGG16(weights="imagenet", include_top=False,
input_shape=IMAGE_SIZE+(3,))
    base.trainable = False # freeze convolutional base

    inputs = layers.Input(shape=IMAGE_SIZE+(3,))
    x = base(inputs, training=False)
    x = layers.GlobalAveragePooling2D()(x)
    outputs = layers.Dense(1, activation="sigmoid")(x)

    model = models.Model(inputs, outputs)
    model.compile(
        optimizer=Adam(1e-4),
        loss="binary_crossentropy",
        metrics=["accuracy"]
    )
    return model

# --- MODEL PEMBANDING: Custom CNN VGG-like ---
def build_model_custom():
    inputs = layers.Input(shape=IMAGE_SIZE+(3,))

    # Blok Conv 1
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(inputs)
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 2
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 3
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 4
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
    x = layers.GlobalAveragePooling2D()(x)

```

```

outputs = layers.Dense(1, activation="sigmoid")(x)

model = models.Model(inputs, outputs)
model.compile(
    optimizer=Adam(1e-4),
    loss="binary_crossentropy",
    metrics=["accuracy"]
)
return model

```

Appendix 6. Training Configuration

```

# === BLOK 5: CALLBACKS ===
early_stop = tf.keras.callbacks.EarlyStopping(
    monitor="val_loss",
    patience=10,
    restore_best_weights=True
)
import numpy as np
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

def run_kfold_training(df, model_builder, splits, out_root, model_name):
    """
    Executes K-Fold Cross Validation Training.
    Focus: Training logic, Class Weighting, and Model Checkpointing.
    """
    fold_metrics = []

    for fold_id, (tr_idx, va_idx) in enumerate(splits, start=1):
        print(f"\n===== {model_name}: Fold {fold_id}/{len(splits)} =====")

        # 1. Data Splitting
        df_tr = df.iloc[tr_idx].copy()
        df_va = df.iloc[va_idx].copy()
        gen_tr, gen_va = make_generators(df_tr, df_va) # Data Augmentation
        applied here

        # 2. Model Initialization
        model = model_builder()

        # 3. Handle Class Imbalance (Compute Class Weights)
        y_train_fold = df.iloc[tr_idx]["label_idx"].to_numpy()
        classes = np.unique(y_train_fold)
        weights = compute_class_weight(class_weight="balanced",
        classes=classes, y=y_train_fold)
        class_weight = {int(c): float(w) for c, w in zip(classes, weights)}

```

```

# 4. Callbacks Configuration
# Save only the best weights based on validation loss
ckpt_path = out_root / model_name / f'fold{fold_id}' /
"best_weights.keras"
ckpt_cb = ModelCheckpoint(
    filepath=str(ckpt_path),
    save_weights_only=True,
    save_best_only=True,
    monitor="val_loss",
    mode="min"
)
# Stop training if no improvement for 10 epochs
early_stop = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)

# 5. Core Training Execution
history = model.fit(
    gen_tr,
    validation_data=gen_va,
    epochs=1000, # Max epochs (controlled by Early Stopping)
    callbacks=[early_stop, ckpt_cb],
    class_weight=class_weight,
    verbose=1
)

# 6. Evaluation (Inference)
y_prob = model.predict(gen_va, verbose=0).ravel()
y_pred = (y_prob >= 0.5).astype(int)
y_true = df_va["label_idx"].values

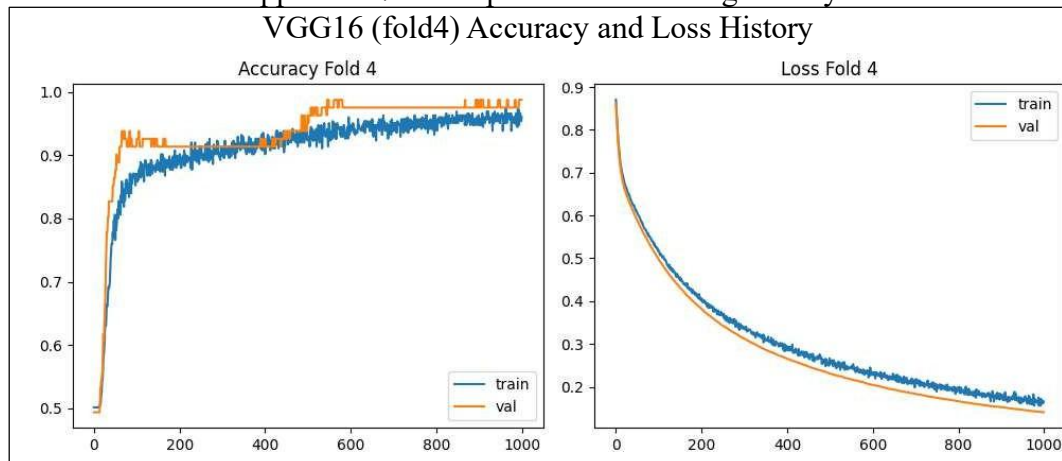
# Metrics calculation (Code simplified for display)
# ... (Accuracy, Precision, Recall, F1, AUC calculation) ...

fold_metrics.append({"fold": fold_id, "acc": acc, "f1": f1, "auc": auc})

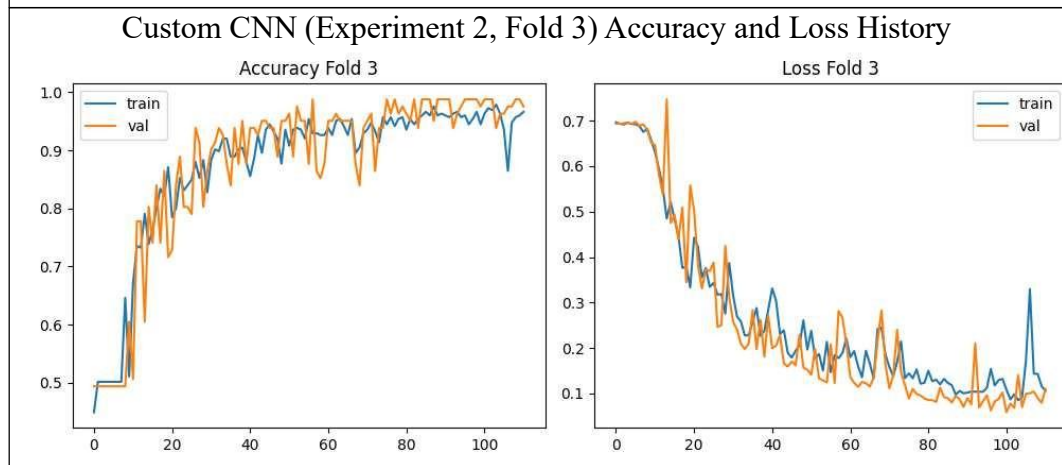
return fold_metrics

```

Appendix 7. Champion Model Testing History VGG16 (fold4) Accuracy and Loss History



Custom CNN (Experiment 2, Fold 3) Accuracy and Loss History



Appendix 8. Sample Detail Prediction

Table 4.
Sample VGG16 Fold 4

Image Filename	True Label	Predicted Label	Confidence Score	Status
orange_melon38.jpg	Orange	Green	0.8905 (89.05% Green)	WRONG (FN)
orange_melon26.jpg	Orange	Orange	0.6731 (67.31% Orange)	CORRECT
orange_melon28.jpg	Orange	Orange	0.9928 (99.28% Orange)	CORRECT
green_melon1.jpg	Green	Green	0.9929 (99.29% Green)	CORRECT
green_melon36.jpg	Green	Orange	0.6515 (65.15% Orange)	WRONG (FP)

Table 5.
Sample Custom CNN Exp 2-fold 3

Image Filename	True Label	Predicted Label	Confidence Score	Status
orange_melon38.jpg	Orange	Green	0.9922 (99.2% Green)	WRONG (FN)
orange_melon26.jpg	Orange	Orange	0.8234 (82.3% Orange)	CORRECT
orange_melon20.jpg	Orange	Orange	1.0000 (100% Orange)	CORRECT
green_melon17.jpg	Green	Green	0.9994 (99.9% Green)	CORRECT
green_melon26.jpg	Green	Orange	0.7834 (78.3% Orange)	WRONG (FP)

Appendix 9. Custom CNN History

```
# --- MODEL PEMBANDING: Custom CNN VGG-like ---
def build_model_custom():
    inputs = layers.Input(shape=IMAGE_SIZE+(3,))

    # Blok Conv 1
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(inputs)
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 2
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 3
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok Conv 4
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
```

```
x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
x = layers.GlobalAveragePooling2D()(x)
```

```
outputs = layers.Dense(1, activation="sigmoid")(x)
```

```
model = models.Model(inputs, outputs)
model.compile(
    optimizer=Adam(1e-4),
    loss="binary_crossentropy",
    metrics=["accuracy"]
)
return model
```

```
def build_model_B_hidden_flatten():
    inputs = layers.Input(shape=IMAGE_SIZE+(3,))

    # --- Feature Extractor ---
    # Blok 1
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(inputs)
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 2
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 3
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 4
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
```

```

x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
# PENTING: Saya tambah MaxPool di sini biar hasil Flatten gak meledak
x = layers.MaxPooling2D()(x)

# --- Classifier Klasik (Permintaan Dosen) ---
# GANTI GAP JADI FLATTEN
x = layers.Flatten()(x)

# Hidden Layer (Dense)
# Karena pakai Flatten, input ke sini BANYAK BANGET.
# Wajib pakai Dropout biar gak overfitting parah.
x = layers.Dense(256, activation="relu")(x)
x = layers.Dropout(0.5)(x)

outputs = layers.Dense(1, activation="sigmoid")(x)

model = models.Model(inputs, outputs, name="Model_B_Flatten_Hidden")
model.compile(optimizer=Adam(1e-4), loss="binary_crossentropy",
metrics=["accuracy"])
return model

```

```

def build_model_history_C_basic():
    inputs = layers.Input(shape=IMAGE_SIZE+(3,))

    # Struktur Dangkal (Shallow)
    # Blok 1
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(inputs)
    x = layers.MaxPooling2D()(x)

    # Blok 2
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 3 (Stop di sini, belum dalam)
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)

```

```

x = layers.MaxPooling2D()(x)

# Classifier Sederhana
x = layers.Flatten()(x)
x = layers.Dense(64, activation="relu")(x) # Hidden layer kecil
outputs = layers.Dense(1, activation="sigmoid")(x)

model = models.Model(inputs, outputs, name="Model_C_Basic")
model.compile(optimizer=Adam(1e-4), loss="binary_crossentropy",
metrics=["accuracy"])

return model

```

```

def build_model_history_D_complex():
    inputs = layers.Input(shape=IMAGE_SIZE+(3,))

    # Struktur Dalam (Deep) - Mirip Final
    # Blok 1
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(inputs)
    x = layers.Conv2D(32, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 2
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(64, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 3
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(128, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

    # Blok 4
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
    x = layers.Conv2D(256, (3,3), padding="same", activation="relu")(x)
    x = layers.MaxPooling2D()(x)

```

```
# Classifier "Gemuk" (Penyebab Masalah)
x = layers.Flatten()(x) # Ini biang keroknya
x = layers.Dense(512, activation="relu")(x) # Hidden layer besar
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)

model = models.Model(inputs, outputs, name="Model_D_Complex")
model.compile(optimizer=Adam(1e-4), loss="binary_crossentropy",
metrics=["accuracy"])
return model
```

