

LAMPIRAN



Lampiran 01 Form Pengantar Kelaikan Etik

	KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI UNIVERSITAS PENDIDIKAN GANESHA FAKULTAS TEKNIK DAN KEJURUAN Alamat : Jalan Udayana No. 11 Singaraja-Bali Website fk.undiksha.ac.id Telepon : 082141321955, Email : fk@undiksha.ac.id Kode Pos 81116
	FL/06-06
SURAT PENGANTAR KELAIKAN ETIK	
Singaraja 06 Februari 2026	
Yth. Ketua Komite Etik Penelitian Fakultas Kedokteran Universitas Pendidikan Ganesha Singaraja	
Dengan hormat, Saya yang bertandatangan di bawah ini bermaksud mengajukan permohonan kaji etik penelitian:	
Nama (dengan gelar) NIM/NIP/NIR/NIK Pekerjaan Institusi Judul Penelitian	: I Gde Made Hanura : 2215101062 : Mahasiswa : Universitas Pendidikan Ganesha : KLASIFIKASI JENIS KELAMIN BERBASIS CITRA MATA MANUSIA DENGAN PENDEKATAN VISION TRANSFORMER (VIT)
Pembimbing 1/Promotor : Dr. Putu Hendra Suputra, S.Kom., M.Cs. Pembimbing 2/Ko Promotor : Ni Putu Novita Puspa Dewi, S.Kom., M.Cs., MIM.S	
Demikian surat permohonan ini saya buat dengan sebenar-benarnya, atas perhatiannya saya ucapkan terima kasih.	
Menyetujui, Pembimbing 1/ Promotor  (Dr. Putu Hendra Suputra, S.Kom., M.Cs.)	Pengusul  (I Gde Made Hanura)
 Mengetahui, Atasan institusi  (Prof. Dr. Kadek Rihendra Dantes, S.T., M.T.)	

Lampiran 02 Surat Pernyataan Peneliti

	KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI UNIVERSITAS PENDIDIKAN GANESHA FAKULTAS KEDOKTERAN KOMITE ETIK PENELITIAN Jl. Udayana No.11, Singaraja, Bali, Kode Pos 81116
	FL/07-06
SURAT PERNYATAAN PENELITIAN	
Saya yang bertanda tangan di bawah ini:	
Nama	: I Gde Made Hanura
NIM/NIP/NIR/NIK	: 2215101062
Judul Penelitian	: KLASIFIKASI JENIS KELAMIN BERBASIS CITRA MATA MANUSIA DENGAN PENDEKATAN VISION TRANSFORMER (ViT)
Asal Instansi	: Universitas Pendidikan Ganesha
Nomor HP	: 081339277560
Dengan sesungguhnya menyatakan bahwa saya telah mengisi seluruh data dengan sebenar-benarnya dan bersedia mematuhi semua prinsip yang tertuang dalam pedoman etik WHO 2011 dan CIOMS 2016. Apabila saya melanggar salah satu prinsip tersebut dan terdapat bukti adanya pemalsuan data, maka saya bersedia diberikan sanksi sesuai dengan kebijakan dan aturan yang berlaku.	
Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya, atas perhatiannya saya mengucapkan banyak terima kasih.	
Singaraja, 6 Februari 2026 Peneliti utama,   (I Gde Made Hanura)	

Lampiran 03 Kode Arsitektur Vit-B/16

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

model = ViTForImageClassification.from_pretrained(
    "google/vit-base-Patch16-224-in21k",
    num_labels=2,
    ignore_mismatched_sizes=True,
    output_attentions=True
)

model.to(device)
```



Lampiran 04 Kode Augmentasi Data Vit-B/16

```
train_transform = transforms.Compose([
    transforms.Grayscale(num_output_channels=3),
    transforms.Resize((224,224)),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
    transforms.Normalize([0.5]*3,[0.5]*3)
])

val_test_transform = transforms.Compose([
    transforms.Grayscale(num_output_channels=3),
    transforms.Resize((224,224)),
    transforms.ToTensor(),
    transforms.Normalize([0.5]*3,[0.5]*3)
])
```



Lampiran 05 Kode Training dan Validasi Model

```

import time
from tqdm import tqdm

EXPERIMENT_NAME = "D0"
SAVE_PATH = f'best_model_{EXPERIMENT_NAME}.pth'

best_val_loss = float('inf')
best_val_acc = 0.0
best_Epoch = 0
patience = 4
patience_counter = 0

history = {
    'train_loss': [], 'train_acc': [],
    'val_loss': [], 'val_acc': []
}

for Epoch in range(num_Epochs):
    start_time = time.time()
    model.train()
    train_loss, train_correct, train_total = 0.0, 0, 0

    train_bar = tqdm(train_loader,
                      desc=f'Epoch {Epoch+1}/{num_Epochs} [Train]',
                      leave=True)

    for images, labels in train_bar:
        images, labels = images.to(device), labels.to(device)

        optimizer.zero_grad()
        outputs = model(images).logits
        loss = criterion(outputs, labels)
        loss.backward()

        torch.nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)

        optimizer.step()
        scheduler.step()

        batch_size = labels.size(0)
        train_loss += loss.item() * batch_size
        _, preds = torch.max(outputs, 1)
        train_correct += (preds == labels).sum().item()
        train_total += batch_size

    train_bar.set_postfix({

```

```

        "loss": f"{loss.item():.4f}",
        "acc": f"{train_correct/train_total:.4f}",
        "lr": f"{scheduler.get_last_lr()[0]:.2e}"
    })
    avg_train_loss = train_loss / train_total
    avg_train_acc = train_correct / train_total

    model.eval()
    val_loss, val_correct, val_total = 0.0, 0, 0

    with torch.no_grad():
        val_bar = tqdm(val_loader,
                        desc=f'Epoch {Epoch+1}/{num_Epochs} [Val]',
                        leave=False)

        for images, labels in val_bar:
            images, labels = images.to(device), labels.to(device)

            outputs = model(images).logits
            loss = criterion(outputs, labels)
            batch_size = labels.size(0)
            val_loss += loss.item() * batch_size
            _, preds = torch.max(outputs, 1)
            val_correct += (preds == labels).sum().item()
            val_total += batch_size

            val_bar.set_postfix({
                "loss": f"{loss.item():.4f}",
                "acc": f"{val_correct/val_total:.4f}"
            })

    avg_val_loss = val_loss / val_total
    avg_val_acc = val_correct / val_total
    Epoch_time = time.time() - start_time

    history['train_loss'].append(avg_train_loss)
    history['train_acc'].append(avg_train_acc)
    history['val_loss'].append(avg_val_loss)
    history['val_acc'].append(avg_val_acc)

```

Lampiran 06 Kode Evaluasi Model

```

import torch
import numpy as np
from tqdm import tqdm
from sklearn.metrics import confusion_matrix

# =====
# EVALUASI MODEL
# =====

model.eval()

all_preds = []
all_labels = []

with torch.no_grad():
    test_bar = tqdm(
        test_loader,
        desc="Evaluating [Test]",
        total=len(test_loader),
        leave=True
    )

    for images, labels in test_bar:
        images = images.to(device)
        labels = labels.to(device)

        outputs = model(images).logits
        preds = torch.argmax(outputs, dim=1)

        all_preds.extend(preds.cpu().numpy())
        all_labels.extend(labels.cpu().numpy())

# =====
# CONFUSION MATRIX
# =====

cm = confusion_matrix(all_labels, all_preds)

# Label:
# 0 = Pria
# 1 = Wanita
#
#       Prediksi
#       Pria  Wanita
# Aktual Pria  M11   M12
# Aktual Wanita M21   M22

M11 = cm[0, 0]

```



```
M12 = cm[0, 1]
M21 = cm[1, 0]
M22 = cm[1, 1]

print("\n=====")
print("CONFUSION MATRIX")
print("=====")
print(cm)

print("\nNilai Elemen Confusion Matrix")
print(f"M11 (Pria → Pria) : {M11}")
print(f"M12 (Pria → Wanita) : {M12}")
print(f"M21 (Wanita → Pria) : {M21}")
print(f"M22 (Wanita → Wanita) : {M22}")
```



Lampiran 07 Demo Model

Gender Classification from Eye Images

Upload an image or use the webcam. The system will automatically detect and crop the eye region before classification.

Features

- Vision Transformer (ViT-Base)
- Automatic Eye Detection
- Webcam Support
- Confidence Visualization
- Attention Map Visualization
- CLAHE / Grayscale Preprocessing

Input Image



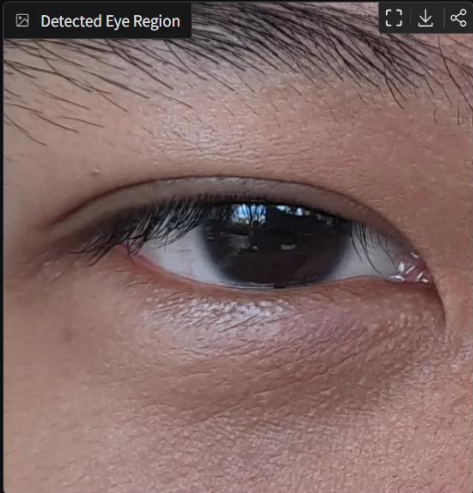
Upload, Webcam, Copy

Prediction Confidence

Male

Male	95%
Female	5%

Detected Eye Region



Fullscreen, Download, Share

Predict Gender

Clear

Lampiran 08 Riwayat Hidup**RIWAYAT HIDUP**

I Gde Made Hanura lahir di Tabanan pada tanggal 27 Oktober 2024. Kini penulis tinggal di Banjar Sakenan Baleran. Jalan Danau Toba Gg 1 No 1, Delod Peken Tabanan. Penulis menyelesaikan pendidikan dasar di SD Negeri 1 Tabanan. Kemudian penulis melanjutkan di SMP Negeri 3 Tabanan dan lanjut melakukan pendidikan di SMA Negeri 1 Tabanan Bali mengambil kelas MIPA 2. Kemudian penulis lanjut menjadi mahasiswa aktif dari Universitas Pendidikan Ganesha mengambil jurusan Teknik Informatika dalam Prodi Ilmu Komputer.

