



Lampiran 1. Surat Izin Penelitian



PEMERINTAH KABUPATEN BULELENG
DINAS PENANAMAN MODAL DAN PELAYANAN
TERPADU SATU PINTU
Lantai 3 Pasar Banyuasri, Kelurahan Banyuasri, Kecamatan Buleleng
Telp. (0362) 22063 Singaraja
Alamat e-mail : dpmpstp@bulelengkab.go.id
Website : dpmpstp.bulelengkab.go.id

Nomor : 503/628/REK/DPMPSTP/2025
Lamp :
Perihal : **Surat Keterangan Penelitian**

Kepada :
Yth. Kepala Dinas Kesehatan
Kabupaten Buleleng

di-
Tempat

- I. Dasar :
- Peraturan Menteri Dalam Negeri RI Nomor : 3 Tahun 2018 tentang Penerbitan Surat Keterangan Penelitian;
 - Peraturan Menteri Dalam Negeri RI Nomor : 138 Tahun 2017 tentang Penyelenggaraan Pelayanan Terpadu Satu Pintu Daerah;
 - Surat dari Ketua Jurusan Matematika Universitas Pendidikan Ganesha Nomor 184/UN48.9.5/TU/2025 Tanggal 8 Juli 2025 Perihal Surat Ijin Penelitian
- II. Setelah mempelajari dan meneliti rencana kegiatan yang diajukan, maka dapat diberikan Surat Keterangan Penelitian Kepada :
- Nama : Welzeen Sinaga
NIK : 1208291310040001
Pekerjaan : Mahasiswa
Alamat : Buttu Ganjang, Kel. Siporkas, Kec. Raya, Kab. Simalungun, Prov. Sumatra Utara
- Bidang / Judul : Pengembangan Aplikasi Stunting Di Kecamatan Buleleng Dengan Menerapkan Algoritma Naive Bayes
- Jumlah Peserta : 1 Orang
Lokasi : Dinas Kesehatan Kabupaten Buleleng
Lamanya : 3 bulan (14 Juli - 20 September 2025)
- III. Dalam melakukan kegiatan agar yang bersangkutan mematuhi ketentuan sebagai berikut :
- Sebelum mengadakan kegiatan agar melapor kepada Kepala Dinas Penanaman Modal dan PTSP Kabupaten Buleleng atau Pejabat yang Berwenang;
 - Tidak dibenarkan melakukan kegiatan yang tidak ada kaitannya dengan bidang/ judul dimaksud, apabila melanggar ketentuan akan dicabut ijinnya dan menghentikan segala kegiatannya;
 - Menaati segala ketentuan perundang-undangan yang berlaku serta mengindahkan adat istiadat dan budaya setempat;
 - Apabila masa berlaku Surat Keterangan Penelitian / Ijin ini telah berakhir, sedangkan pelaksanaan kegiatan belum selesai maka perpanjangan Surat Keterangan Penelitian / Ijin agar ditujukan kepada Instansi pemohon;
- Demikian Surat Keterangan Penelitian ini dibuat untuk dipergunakan sebagaimana mestinya.

DITETAPKAN : SINGARAJA
PADA TANGGAL : 11 Juli 2025



Tembusan ini disampaikan kepada yth:

- Kepala Dinas Penanaman Modal dan PTSP Prov. Bali
- Kepala Badan Kesatuan Bangsa dan Politik Kabupaten Buleleng
- Yang bersangkutan



yang diterbitkan oleh Balai Sertifikasi Elektronik, Badan Siber Dan Sandi Negara

Lampiran 2. Surat Izin Permohonan Penelitian



KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
JURUSAN MATEMATIKA

Alamat : Jalan Udayana Singaraja-Bali
 Telepon (0362) 25072 Fax. (0362) 25335 Pos 81116

Nomor : 184/UN48.9.5/TU/2025
 Lampiran : -
 Perihal : Surat Ijin Penelitian

Singaraja, 8 Juli 2025

Yth : Kepala Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu

Dengan hormat, dalam rangka penyusunan skripsi yang berjudul "**Pengembangan Aplikasi Stunting di Kecamatan Buleleng Dengan Menerapkan Algoritma Naive Bayes**", bersama ini dimohon bantuannya untuk memberikan ijin melakukan pengambilan data status gizi dan stunting pada balita di Kecamatan Buleleng terkait penelitian kepada mahasiswa berikut.

Nama : Welzeen Sinaga
 NIM : 2213101012
 Program Studi : S1 Matematika

Demikian surat ini disampaikan, atas perkenan dan kerjasamanya diucapkan terima kasih.

Mengetahui
 Ketua Jurusan Matematika,



Prof. Dr. I Putu Wisna Ariawan, M.Si.
 NIP. 196805191993031001

Lampiran 3 Surat Pernyataan Penelitian

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Welzeen Sinaga
 Tempat/Tgl Lahir : Buttu Ganjang, 13 Oktober 2003
 Alamat : Buttu Ganjang, Desa Siporkas, Keamatan Raya
 No HP : 081362465033

Dengan ini menyatakan bahwa dalam melaksanakan kegiatan penelitian, peneliti bersedia untuk :

1. Melapor kepada Bupati/Walikota/Instansi yang bersangkutan sebelum memulai penelitian.
2. Menaati dan menghormati ketentuan perundang- undangan yang berlaku di daerah penelitian.
3. Dikenakan sanksi apabila melakukan kegiatan diluar tujuan penelitian yang telah ditetapkan.
4. Melapor Kembali kepada Pemerintah Kabupaten Buleleng melalui Dinas Penanaman Modal dan PTSP Kabupaten Buleleng setelah selesai melakukan kegiatan penelitian serta menyerahkan 1 (satu) eksemplar hasil penelitian.

Demikian surat pernyataan ini dibuat dengan sebenarnya tanpa adanya tekanan dari pihak manapun, agar dapat dipergunakan sebagai kelengkapan administrasi permohonan izin/rekomendasi penelitian.

Singaraja, 10 Juli 2025

 Welzeen Sinaga

Lampiran 4 Surat Pernyataan Keabsahan Dokumen

SURAT PERNYATAAN KEABSAHAN DOKUMEN

Saya yang bertanda tangan dibawah ini :

- Nama : Welzeen Sinaga
- Tempat/tanggal lahir : Buttu Ganjang, 13 Oktober 2003
- Agama : Kristen
- Jenis Kelamin : Laki-laki
- Pekerjaan : Mahasiswa
- Identitas / KTP : 1208291310040001
- Alamat : Buttu Ganjang, Desa Siporkas, Kecamatan Raya

Dengan ini menyatakan bahwa seluruh dokumen yang menjadi kelengkapan dalam permohonan yang saya ajukan kepada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kabupaten Buleleng sudah sah dan sesuai dengan ketentuan perundang undangan yang berlaku.

Demikian pernyataan ini saya buat tanpa adanya tekanan dari pihak manapun dan surat pernyataan ini dapat dipergunakan sebagaimana mestinya.

Singaraja, 11 Juli 2025


(Welzeen Sinaga)



Lampiran 5 Data Standar Antropometri Balita

<https://drive.google.com/file/d/1xxS-xkM6IMvGqcemgcldQvkeSXg1sCyJ/view?usp=sharing>

Lampiran 6 Data Set Balita

<https://drive.google.com/file/d/13OrT3npB8vkcYiiVBsCmpwJND9mOYl0M/view?usp=sharing>

Lampiran 7 *Data Cleaning* Balita

https://drive.google.com/file/d/1aeCM7QhBD5SLtPHOkPfSzOuTwu-PSNX0/view?usp=drive_link

Lampiran 8 Data *Training* Balita

https://drive.google.com/file/d/1SALUSNyoh7akQ0iEWTgMOsBoEIZJEjG2/view?usp=drive_link

Lampiran 9 Data *Tesing* Balita

https://drive.google.com/file/d/1Ftt5qFcw62Y8i3-7Lk6a-xieAyirSF/view?usp=drive_link

Lampiran 10 Hasil Perhitungan Manual Klasifikasi *Naive Bayes*

<https://docs.google.com/spreadsheets/d/12LIZghSdTJ8qqYzif5Spzjb3ffpYdfNS/edit?usp=sharing&oid=118274215636493304486&rtpof=true&sd=true>

Lampiran 11. *Source Code* Algoritma *Naïve Bayes*

Berikut ini adalah *source code* implementasi algoritma *Naïve Bayes* pada aplikasi deteksi *stunting* balita yang dibangun menggunakan *framework Django* (*Python*). *Source code* terbagi menjadi dua bagian utama, yaitu file *naive_bayes_service.py* yang berisi seluruh logika algoritma, dan file *views.py* yang memanggil fungsi-fungsi tersebut melalui antarmuka web.

A. *Import Library* dan Definisi Konstanta

Bagian ini menginisialisasi seluruh *library* yang dibutuhkan, yaitu *scikit-learn* untuk implementasi *GaussianNB* dan metrik evaluasi, serta *NumPy* untuk operasi array. Selain itu, didefinisikan juga daftar fitur (*features*) dan kategori nilai tiap fitur (*categories*) yang digunakan sebagai acuan encoding data kategorik.

```
"""
naive_bayes_service.py
Implementasi algoritma Naïve Bayes (GaussianNB) untuk klasifikasi
status stunting balita menggunakan 6 fitur kategorik.
"""

import numpy as np
import pickle
import os

from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, confusion_matrix, classification_report
)
```

```

# Path penyimpanan model yang sudah dilatih
MODEL_PATH = os.path.join(os.path.dirname(__file__), "saved_model.pkl")

# Fitur yang digunakan sebagai input algoritma Naïve Bayes
FEATURES = [
    "jenis_kelamin",
    "umur",
    "status_bbu",
    "status_tbu",
    "status_gizi",
    "status_asih",
]

TARGET = "status_stunting"

# Kategori tiap fitur (sesuai data aktual, diurutkan alphabetical)
# digunakan untuk membangun LabelEncoder yang konsisten
CATEGORIES = {
    "jenis_kelamin": ["L", "P"],
    "umur": ["1-34 Bulan", "35-69 Bulan"],
    "status_bbu": ["Kurang", "Normal", "Risiko Lebih", "Sangat Kurang"],
    "status_tbu": ["Normal", "Pendek", "Sangat Pendek"],
    "status_gizi": ["Gizi Baik", "Gizi Buruk", "Gizi Kurang", "Gizi Lebih", "Risiko Gizi Lebih"],
    "status_asih": ["Tidak", "Ya"],
    "status_stunting": ["Potensi Stunting", "Stunting", "Tidak"],
}

```

B. Label Encoding Fitur Kategorik

Algoritma *Naïve Bayes* membutuhkan input berupa bilangan numerik. Oleh karena itu, setiap nilai fitur kategorik (seperti "L"/"P", "Stunting"/"Tidak") diubah menjadi angka menggunakan *LabelEncoder*. Fungsi *get_label_encoders()*

membangun *encoder* secara deterministik berdasarkan daftar kategori yang telah ditetapkan, sehingga hasil encoding konsisten antara proses training dan prediksi.

```
def _norm_gizi(val: str) -> str:

    """Normalisasi nilai Status Gizi: strip spasi + title case."""

    return str(val).strip().title()


def get_label_encoders():

    """

    Membuat LabelEncoder untuk setiap fitur kategorik.

    Setiap encoder di-fit dengan daftar kategori yang telah ditentukan
    agar encoding bersifat konsisten dan deterministik.

    """

    encoders = {}

    for col, cats in CATEGORIES.items():

        le = LabelEncoder()

        le.fit(cats)

        encoders[col] = le

    return encoders
```

C. Persiapan Data (*prepare_dataframe*)

Fungsi ini mengonversi data balita dari *queryset Django* menjadi *array NumPy* yang siap diproses oleh algoritma. Setiap *record* diambil nilai keenam

fiturnya, kemudian masing-masing di-*encode* menjadi bilangan bulat. Data yang tidak lengkap atau memiliki nilai tidak dikenal dilewati secara otomatis.

```
def prepare_dataframe(qs):
    """
    Mengonversi queryset Django menjadi array NumPy (X, y)
    yang siap digunakan oleh algoritma Naïve Bayes.
    Proses:
    1. Setiap record diambil nilai 6 fiturnya.
    2. Setiap nilai fitur di-encode menjadi bilangan bulat
       menggunakan LabelEncoder.
    3. Hasilnya dikumpulkan menjadi array X (fitur) dan y (label).
    """
    encoders = get_label_encoders()
    rows, labels = [], []
    for b in qs:
        try:
            row = []
            for feat in FEATURES:
                val = str(getattr(b, feat, "") or "").strip()
                if feat == "status_gizi":
                    val = norm_gizi(val)
                row.append(float(encoders[feat].transform([val])[0]))
            rows.append(row)
            if b.status_stunting:
                labels.append(
                    int(encoders[TARGET].transform(
                        [b.status_stunting.strip()])[0])
                )
        except Exception:
            continue # lewati data yang tidak lengkap
    return (
        np.array(rows, dtype=float),
        np.array(labels, dtype=int),
    )
```

```

        encoders
    )

```

D. Pelatihan Model (*train_model*)

Fungsi *train_model* melatih model *GaussianNB* menggunakan data *training*. Model *GaussianNB* mengasumsikan bahwa distribusi setiap fitur dalam setiap kelas bersifat *Gaussian* (normal), dan menghitung *probabilitas posterior* menggunakan Teorema Bayes:

$$P(C | X) = P(X | C) \times P(C) / P(X)$$

di mana $P(C)$ adalah *prior probability* kelas, $P(X|C)$ adalah *likelihood* fitur pada kelas tersebut, dan $P(X)$ adalah bukti (*evidence*). Model yang sudah dilatih disimpan dalam file *saved_model.pkl* menggunakan *library pickle*.

```

def train_model(training_qs):
    """
    Melatih model Naïve Bayes (GaussianNB) menggunakan data training.
    Langkah:
    1. Konversi queryset ke array NumPy (X_train, y_train).
    2. Inisialisasi model GaussianNB dari scikit-learn.
    3. Latih model dengan model.fit(X_train, y_train).
    4. Simpan model dan encoder ke file .pkl (pickle).

    GaussianNB mengasumsikan bahwa setiap fitur berdistribusi
    Gaussian (normal) dalam setiap kelas, dan menghitung probabilitas
    posterior menggunakan Teorema Bayes:

        P(C|X) = P(X|C) * P(C) / P(X)
    """
    X, y, encoders = prepare_dataframe(training_qs)
    if len(X) == 0:
        raise ValueError("Tidak ada data training yang valid.")

```

```

model = GaussianNB()

model.fit(X, y)

# Simpan model dan encoder ke disk
with open(MODEL_PATH, "wb") as f:
    pickle.dump({"model": model, "encoders": encoders}, f)

return model, encoders, X, y

```

E. Evaluasi Model (*evaluate_model*)

Setelah model dilatih, fungsi *evaluate_model* menghitung empat metrik evaluasi terhadap data testing, yaitu akurasi, presisi (*weighted*), recall (*weighted*), dan *F1-Score*, serta menghasilkan *confusion matrix* dan *classification report* per kelas.

```

def evaluate_model(model, encoders, testing_qs):
    """
    Mengevaluasi performa model terhadap data testing.
    Metrik yang dihitung:
    - Akurasi : proporsi prediksi yang benar dari total data.
    - Presisi : TP / (TP + FP), weighted average per kelas.
    - Recall : TP / (TP + FN), weighted average per kelas.
    - F1-Score : rata-rata harmonik presisi dan recall.
    - Confusion Matrix: tabel aktual vs prediksi.
    """
    X_test, y_test, _ = prepare_dataframe(testing_qs)
    if len(X_test) == 0:
        raise ValueError("Tidak ada data testing yang valid.")

    y_pred = model.predict(X_test)
    classes = list(encoders[TARGET].classes_)

```

```

acc = accuracy_score(y_test, y_pred)
prec = precision_score(
    y_test, y_pred, average="weighted", zero_division=0)
rec = recall_score(
    y_test, y_pred, average="weighted", zero_division=0)
f1 = f1_score(
    y_test, y_pred, average="weighted", zero_division=0)
cm = confusion_matrix(y_test, y_pred)
cr = classification_report(
    y_test, y_pred, target_names=classes,
    zero_division=0, output_dict=True
)
return {
    "akurasi": acc,
    "presisi": prec,
    "recall": rec,
    "f1_score": f1,
    "confusion_matrix": {
        "labels": classes,
        "matrix": cm.tolist(),
        "y_true": y_test.tolist(),
        "y_pred": y_pred.tolist(),
    },
    "classification_report": cr,
    "jumlah_testing": len(X_test),
}

```

F. Prediksi Data Baru (*predict_single*)

Fungsi *predict_single* digunakan untuk memprediksi status *stunting* satu data balita yang diinputkan melalui form aplikasi. Fungsi ini memuat model yang tersimpan, meng-*encode* nilai input, lalu memanggil *model.predict* dan

model.predict_proba untuk mendapatkan kelas prediksi dan probabilitas *posterior* setiap kelas.

```
def load_model():
    """Memuat model dan encoder dari file pickle."""
    if not os.path.exists(MODEL_PATH):
        return None, None
    with open(MODEL_PATH, "rb") as f:
        data = pickle.load(f)
    return data["model"], data["encoders"]

def predict_single(input_data: dict):
    """
    Melakukan prediksi untuk satu data balita.
    Parameter:
        input_data (dict): dictionary dengan key:
            jenis_kelamin, umur, status_bbu, status_tbu,
            status_gizi, status_asih.
    Return:
        predicted_class (str): label kelas hasil prediksi.
        proba_dict (dict) : probabilitas tiap kelas.
    Proses prediksi menggunakan Teorema Bayes:
        
$$P(C|X) = P(X|C) * P(C)$$

    Model mengembalikan kelas dengan probabilitas posterior tertinggi.
    """
    model, encoders = load_model()
    if model is None:
        raise ValueError(
            "Model belum dilatih. Silakan latih model terlebih dahulu.")

    # Encode input fitur ke bentuk numerik
    row = []
    for feat in FEATURES:
```

```

val = str(input_data.get(feats, "")).strip()

if feat == "status_gizi":
    val = _norm_gizi(val)

row.append(float(encoders[feat].transform([val])[0]))

X = np.array([row], dtype=float)

# Prediksi kelas dan probabilitas posterior
pred_encoded = model.predict(X)[0]
proba = model.predict_proba(X)[0]
label_enc = encoders[TARGET]
predicted_class = label_enc.inverse_transform([pred_encoded])[0]

# Susun dict probabilitas: {nama_kelas: nilai_probabilitas}
proba_dict = {
    cls: float(proba[i])
    for i, cls in enumerate(label_enc.classes_)
}

return predicted_class, proba_dict

```

G. Pemanggilan Algoritma di *views.py*

Berikut adalah bagian dari file *views.py* yang memanggil fungsi-fungsi algoritma *Naïve Bayes* di atas, baik untuk proses pelatihan model maupun prediksi data balita melalui antarmuka web aplikasi.

```

# =====
# views.py - bagian yang memanggil algoritma Naïve Bayes
# =====

from . import naive_bayes_service as nbs

# --- 1. Melatih Model ---
def train_model_view(request):
    training_qs = Balita.objects.filter(dataset_type="training") \

```

```

        .exclude(status_stunting__isnull=True) \
        .exclude(status_stunting="")

# Latih model Naïve Bayes dengan data training
model, encoders, X_train, y_train = nbs.train_model(training_qs)

# Evaluasi model terhadap data testing
testing_qs = Balita.objects.filter(dataset_type="testing") \
        .exclude(status_stunting__isnull=True)
metrics = nbs.evaluate_model(model, encoders, testing_qs)

# Simpan hasil evaluasi ke database
model_obj.akurasi = metrics["akurasi"]
model_obj.presisi = metrics["presisi"]
model_obj.recall = metrics["recall"]
model_obj.f1_score = metrics["f1_score"]
model_obj.save()

# --- 2. Prediksi Satu Data ---
def prediksi_view(request):
    input_data = form.cleaned_data # data dari form input user

# Panggil fungsi prediksi Naïve Bayes
predicted_class, proba_dict = nbs.predict_single(input_data)

result = {
    "kelas": predicted_class,
    # "Stunting" / "Potensi Stunting" / "Tidak"
    "probabilitas": proba_dict,
    # probabilitas tiap kelas
}

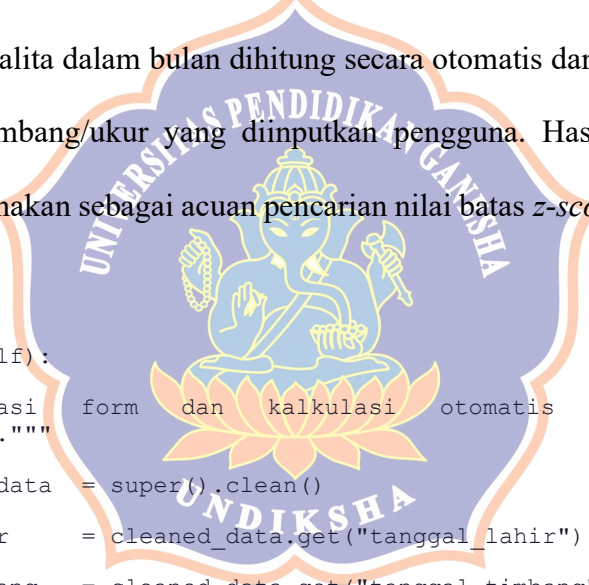
```


Lampiran 12 *Source Code* Perhitungan *Z-Score* Antropometri

Berikut adalah implementasi fungsi *Python* yang digunakan untuk menghitung secara otomatis status BB/U (Berat Badan menurut Umur), TB/U (Tinggi Badan menurut Umur), dan Status Gizi berdasarkan BB/TB (Berat Badan menurut Tinggi Badan). Perhitungan mengacu pada standar *z-score* WHO yang diadaptasi dalam Standar Antropometri Anak. Seluruh kode berada pada file *apps/ml_engine/forms.py*.

I. Penghitungan Otomatis Umur Balita

Umur balita dalam bulan dihitung secara otomatis dari selisih tanggal lahir dan tanggal timbang/ukur yang diinputkan pengguna. Hasil perhitungan umur kemudian digunakan sebagai acuan pencarian nilai batas *z-score* pada tabel standar antropometri.



```
def clean(self):
    """Validasi form dan kalkulasi otomatis semua indikator antropometri."""
    cleaned_data = super().clean()
    tgl_lahir = cleaned_data.get("tanggal_lahir")
    tgl_timbang = cleaned_data.get("tanggal_timbang")

    if tgl_lahir and tgl_timbang:
        # Validasi urutan tanggal
        if tgl_timbang < tgl_lahir:
            raise forms.ValidationError(
                "Tanggal timbang tidak boleh sebelum tanggal lahir.")
        if tgl_timbang > date.today():
            raise forms.ValidationError(
                "Tanggal timbang tidak boleh melebihi hari ini.")
```

```

# Hitung umur dalam bulan
umur_bulan = (
    (tgl_timbang.year - tgl_lahir.year) * 12
    + (tgl_timbang.month - tgl_lahir.month)
)

if tgl_timbang.day < tgl_lahir.day:
    umur_bulan -= 1

umur_bulan = max(0, umur_bulan)

if umur_bulan < 1 or umur_bulan > 69:
    raise forms.ValidationError(
        f"Umur balita ({umur_bulan} bulan) harus antara 1-69
bulan.")

# Tentukan kelompok umur untuk fitur model
cleaned_data["umur_bulan"] = umur_bulan
cleaned_data["umur"] = (
    "1-34 Bulan" if umur_bulan <= 34 else "35-69 Bulan")

jk = cleaned_data.get("jenis_kelamin")
bb = cleaned_data.get("berat_badan")
tb = cleaned_data.get("tinggi_badan")

# Hitung status BB/U (z-score WHO)
if jk and bb is not None:
    cleaned_data["status_bbu"] = _hitung_status_bbu(jk,
umur_bulan, bb)

# Hitung status TB/U (z-score WHO)
if jk and tb is not None:
    cleaned_data["status_tbu"] = _hitung_status_tbu(jk,
umur_bulan, tb)

# Hitung status gizi berdasarkan BB/TB (z-score WHO)
if jk and bb is not None and tb is not None:

```

```

        cleaned_data["status_gizi"] = _hitung_status_gizi(
            jk, umur_bulan, bb, tb)

    return cleaned_data

```

J. Fungsi Klasifikasi BB/U

Fungsi ini mengklasifikasikan status berat badan balita menurut umur (BB/U) berdasarkan tabel *z-score* WHO.

```

def _hitung_status_bbu(jk, umur_bulan, bb):
    """
    Klasifikasi BB/U (Berat Badan menurut Umur) – z-score WHO.
    Parameter:
        jk          : Jenis kelamin ("L" atau "P")
        umur_bulan  : Umur dalam bulan (1-60)
        bb          : Berat badan dalam kilogram (kg)
    Return:
        str: "Sangat Kurang" | "Kurang" | "Normal" | "Risiko Lebih"
    """
    tabel = _BBU_L if jk == "L" else _BBU_P
    bulan = min(umur_bulan, 60)
    if bulan not in tabel:
        bulan = min(tabel.keys(), key=lambda x: abs(x - bulan))

    # Format tabel: (median, -2SD, -3SD, +2SD)
    median, sd_neg2, sd_neg3, sd_pos2 = tabel[bulan]

    if bb < sd_neg3:
        return "Sangat Kurang"
    elif bb < sd_neg2:
        return "Kurang"
    elif bb <= sd_pos2:
        return "Normal"
    else:

```

```
return "Risiko Lebih"
```

```
# Tabel BB/U Laki-laki – format: {bulan: (median, -2SD, -3SD, +2SD)}
```

```
# Sumber: WHO Child Growth Standards 2006
```

```
_BBU_L = {
```

```
    1: (4.3, 3.4, 2.9, 5.8),    2: (5.6, 4.3, 3.8, 7.1),
```

```
    3: (6.4, 5.0, 4.4, 8.0),    4: (7.0, 5.6, 4.9, 8.7),
```

```
    5: (7.5, 6.0, 5.3, 9.3),    6: (7.9, 6.4, 5.7, 9.8),
```

```
    # ... (data lengkap sampai bulan 60)
```

```
    60: (18.3, 15.0, 13.0, 24.7),
```

```
}
```

```
# Tabel BB/U Perempuan – format: {bulan: (median, -2SD, -3SD, +2SD)}
```

```
_BBU_P = {
```

```
    1: (4.2, 3.2, 2.8, 5.5),    2: (5.1, 3.9, 3.4, 6.6),
```

```
    3: (5.8, 4.5, 3.9, 7.5),    4: (6.4, 5.0, 4.4, 8.1),
```

```
    5: (6.9, 5.4, 4.8, 8.8),    6: (7.3, 5.7, 5.1, 9.3),
```

```
    # ... (data lengkap sampai bulan 60)
```

```
    60: (18.8, 14.8, 13.0, 25.4),
```

```
}
```

K. Fungsi Klasifikasi TB/U

Fungsi ini mengklasifikasikan status tinggi badan balita menurut umur (TB/U). TB/U < -2SD dikategorikan sebagai Pendek (*stunted*), sedangkan < -3SD dikategorikan Sangat Pendek (*severely stunted*).

```
def _hitung_status_tbu(jk, umur_bulan, tb):
```

```
    """
```

```
    Klasifikasi TB/U (Tinggi Badan menurut Umur) – z-score WHO.
```

```
    Parameter:
```

```
        jk          : Jenis kelamin ("L" atau "P")
```

```
        umur_bulan : Umur dalam bulan (1-60)
```

```

        tb          : Tinggi badan dalam sentimeter (cm)
Return:
        str: "Sangat Pendek" | "Pendek" | "Normal"
"""
tabel = _TBU_L if jk == "L" else _TBU_P
bulan = min(umur_bulan, 60)
if bulan not in tabel:
    bulan = min(tabel.keys(), key=lambda x: abs(x - bulan))

# Format tabel: (median, -2SD, -3SD)
median, sd_neg2, sd_neg3 = tabel[bulan]

if tb < sd_neg3:
    return "Sangat Pendek"
elif tb < sd_neg2:
    return "Pendek"
else:
    return "Normal"

```



```

# Tabel TB/U Laki-laki – format: {bulan: (median, -2SD, -3SD)}
# Sumber: WHO Child Growth Standards 2006
_TBU_L = {
    1: (54.7, 50.8, 48.9),    2: (58.4, 54.4, 52.4),
    3: (61.4, 57.3, 55.3),    4: (63.9, 59.7, 57.6),
    5: (65.9, 61.7, 59.6),    6: (67.6, 63.3, 61.2),
    # ... (data lengkap sampai bulan 60)
    60: (113.0, 106.5, 103.2),
}

# Tabel TB/U Perempuan – format: {bulan: (median, -2SD, -3SD)}
_TBU_P = {
    1: (53.7, 49.8, 47.8),    2: (57.1, 53.0, 51.0),
    3: (59.8, 55.6, 53.5),    4: (62.1, 57.8, 55.6),

```

```

5: (64.0, 59.6, 57.4),    6: (65.7, 61.2, 58.9),
# ... (data lengkap sampai bulan 60)
60: (113.3, 107.5, 104.3),
}

```

L. Fungsi Klasifikasi Status Gizi BB/TB

Fungsi ini mengklasifikasikan status gizi balita berdasarkan rasio berat badan terhadap tinggi badan (BB/TB). Berbeda dengan BB/U dan TB/U yang menggunakan umur sebagai referensi, BB/TB menggunakan nilai tinggi badan aktual untuk mencari baris yang sesuai pada tabel standar.

```

def _hitung_status_gizi(jk, umur_bulan, bb, tb):
    """
    Klasifikasi Status Gizi berdasarkan BB/TB — z-score WHO.
    Sesuai Permenkes RI No. 2 Tahun 2020.
    Parameter:
        jk      : Jenis kelamin ("L" atau "P")
        umur_bulan : Umur dalam bulan (digunakan jika diperlukan)
        bb      : Berat badan dalam kilogram (kg)
        tb      : Tinggi badan dalam sentimeter (cm), rentang 45-110
    Return:
        str: "Gizi Buruk" | "Gizi Kurang" | "Gizi Baik" |
            "Risiko Gizi Lebih" | "Gizi Lebih"
    """
    tabel = _BBTB_L if jk == "L" else _BBTB_P
    keys = sorted(tabel.keys())

    # Cari nilai TB terdekat di tabel (interval 0.5 cm)
    if tb <= keys[0]:
        key = keys[0]
    elif tb >= keys[-1]:
        key = keys[-1]
    else:

```

```

key = min(keys, key=lambda x: abs(x - tb))

# Format tabel: (-3SD, -2SD, median, +1SD, +2SD)
neg3, neg2, median, pos1, pos2 = tabel[key]

if bb < neg3:
    return "Gizi Buruk"
elif bb < neg2:
    return "Gizi Kurang"
elif bb <= pos1:
    return "Gizi Baik"
elif bb <= pos2:
    return "Risiko Gizi Lebih"
else:
    return "Gizi Lebih"

# Tabel BB/TB Laki-laki - format: {tb_cm: (-3SD, -2SD, median, +1SD, +2SD)}
# Sumber: Standar Antropometri Anak, Kemenkes RI 2020
_BBTB_L = {
    45.0: (1.9, 2.1, 2.4, 2.7, 3.1),
    45.5: (2.0, 2.2, 2.5, 2.8, 3.2),
    # ... (interval 0.5 cm sampai 110.0 cm)
    75.0: (10.2, 11.2, 12.6, 14.7, 17.2),
    75.5: (10.4, 11.3, 12.8, 14.9, 17.5),
    # ...
    110.0: (19.2, 22.0, 25.8, 32.9, 43.0),
}

# Tabel BB/TB Perempuan - format: {tb_cm: (-3SD, -2SD, median, +1SD, +2SD)}
_BBTB_P = {
    45.0: (1.9, 2.1, 2.4, 2.6, 3.0),

```



```
45.5: (2.0, 2.2, 2.5, 2.8, 3.1),  
# ... (interval 0.5 cm sampai 110.0 cm)  
75.0: (9.6, 10.5, 11.8, 13.7, 16.5),  
75.5: (9.7, 10.6, 12.0, 13.9, 16.7),  
# ...  
110.0: (19.0, 21.7, 25.9, 33.7, 46.0),  
}
```



Lampiran 13. Dokumentasi Penelitian



RIWAYAT HIDUP



Welzeen Sinaga lahir pada 13 Oktober 2003. Penulis lahir dari pasangan suami istri Bapak Elson Sarisen Sinaga dan Ibu Ermalina Purba. Penulis berkebangsaan Indonesia dan beragama Kristen Protestan. Penulis tinggal di Kabupaten Simalungun. Penulis menempuh pendidikan sekolah dasar di SD Negeri 091329 Bahpasunsang dan lulus pada tahun 2016. Jenjang menengah pertama penulis ditempuh di SMP N 1 Raya dan lulus pada tahun 2019. Jenjang menengah atas penulis ditempuh di SMA N 1 Raya dan diselesaikan pada tahun 2022. Penulis melanjutkan ke Strata 1 Jurusan Matematika di Universitas Pendidikan Ganesha pada tahun 2022. Pada semester akhir tahun 2026 penulis menyelesaikan Tugas Akhir yang berjudul “Pengembangan Aplikasi Prediksi *Stunting* dengan Menerapkan Algoritma *Naïve Bayes* di Kecamatan Buleleng”. Tahun 2022 sampai dengan sekarang, penulis masih menjadi mahasiswa aktif di Program S1 Matematika Universitas Pendidikan Ganesha.

