

Lampiran 1. Surat Observasi



POLITEKNIK HARAPAN BERSAMA

D-3 Teknik Komputer

Nomor : 228.03/KOM-PHB/III/2025
Lampiran : -
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.

Kepala Posyandu Slawi Wetan

Di Jl.serayu RT 16 RW 06 Kelurahan Slawi Wetan, Kecamatan Slawi, Kabupaten Tegal

Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi D III Teknik Komputer Politeknik Harapan Bersama Tegal, Maka dengan ini kami mengajukan izin observasi pengambilan data di Posyandu Slawi Wetan yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No	NIM	NAMA	NO HP
1	22040095	JAVANKA AMEDEO CAVENDISH	082314800448
2	22040071	YADI DHARMAWAN	082323942925

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 03 Maret 2025
Kas. Prodi DIII Teknik Komputer
Politeknik Harapan Bersama Tegal

Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Lampiran 2. Surat Kesiediaan Membimbing Dosen Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang berntanda tangan dibawah ini

Nama : Muhammad Bakhar, M.Kom
NIDN : 0622028602
NIPY : 04.014.179
Jabatan Struktural : Dosen tetap
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

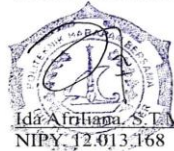
Nama : Javanka Amedeo Cavendish
NIM : 22040095
Program Studi : Diploma III Teknik Komputer

Judul TA : SISTEM INFORMASI TUMBUH KEMBANG BALITA
BERBASIS *WEBSITE*

Dengan ini menyatakan agar dilaksanakan sebagaimana mestinya.

Tegal, 24 Februari 2025

Mengetahui,
Ka Prodi DIII Teknik Komputer



Ida Afrhiana, S.T.M.Kom
NIPY. 12.013.168

Dosen Pembimbing I

A handwritten signature in black ink, likely belonging to Muhammad Bakhar, M.Kom.

Muhammad Bakhar, M.Kom
NIPY. 04.014.179

Lampiran 3. Surat Kesiediaan Membimbing Dosen Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini

Nama : Eko Budihartono, S.T, M.Kom.
NIDN : 0605037304
NIPY : 12.013.170
Jabatan Struktural : Sub Pelatihan dan Sertifikasi
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Javanka Amedeo Cavendish
NIM : 22040095
Program Studi : Diploma III Teknik Komputer

Judul TA : SISTEM INFORMASI TUMBUH KEMBANG BALITA
BERBASIS *WEBSITE*

Dengan ini menyatakan agar dilaksanakan sebagaimana mestinya.

Tegal, 3 Maret 2025

Mengetahui,
Ka Prodi DIII Teknik Komputer

Dosen Pembimbing II


Ida Afriliana, S.T.M.Kom
NIPY-12.013.168


Eko Budihartono, S.T, M.Kom
NIPY. 12.013.170

Lampiran 4. Source Code Program

```
import React, { useState, useEffect } from "react";
import { Form, Button, Container, Row, Col, Alert, Card, Spinner,
Badge, Modal } from "react-bootstrap";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import { format } from "date-fns";
import id from "date-fns/locale/id";
import SideBidan from "../SideBidan";

const STUNTING_THRESHOLD = -2;
const API_BASE_URL = "http://192.168.43.36:5000";

const stuntingRecommendations = [
  "Memberikan ASI eksklusif hingga bayi berusia 6 bulan",
  "Memberikan MPASI bergizi dan kaya protein hewani untuk bayi >6
bulan",
  "Memantau perkembangan anak dan membawa ke posyandu secara
berkala",
  "Mengonsumsi rutin Tablet Tambah Darah (TTD)",
  "Melakukan imunisasi rutin",
  "Memantau tumbuh kembang anak",
  "Menerapkan perilaku hidup bersih dan sehat",
  "Memakai jamban sehat",
  "Konsultasi dengan dokter atau ahli gizi",
  "Pertahankan gizi yang baik",
];

const StuntingPrediction = () => {
  const [formData, setFormData] = useState({
    id_anak: "",
    nik: "",
    nama_anak: "",
    nama_orang_tua: "",
    jenis_kelamin: "",
    umur: "",
    tinggi_badan: "",
    berat_badan: "",
  });

  const [anakList, setAnakList] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [predictionResult, setPredictionResult] = useState(null);
  const [error, setError] = useState(null);
```

```

const [showHistoryModal, setShowHistoryModal] = useState(false);
const [isFetchingAlatData, setIsFetchingAlatData] =
useState(false);
const [alatDataError, setAlatDataError] = useState(null);
const navigate = useNavigate();

useEffect(() => {
  const token = localStorage.getItem("token");
  if (!token) {
    navigate("/login");
    return;
  }

  try {
    const tokenData = JSON.parse(atob(token.split(".")[1]));
    if (Date.now() >= tokenData.exp * 1000) {
      localStorage.removeItem("token");
      navigate("/login");
    }
  } catch (err) {
    localStorage.removeItem("token");
    navigate("/login");
  }

  fetchAnakData();
  fetchPredictionHistory();
  fetchLatestAlatData();
}, [navigate]);

const fetchAnakData = async () => {
  try {
    const response = await axios.get(`${API_BASE_URL}/anak/list`,
{
    headers: { Authorization: `Bearer
${localStorage.getItem("token")}` },
    });
    setAnakList(response.data.data || []);
  } catch (err) {
    setError(err.response?.data?.message || "Gagal memuat data
anak");
  }
};

const fetchPredictionHistory = async () => {
  try {

```

```

const response = await
axios.get(`${API_BASE_URL}/stunting/history`, {
  headers: { Authorization: `Bearer
${localStorage.getItem("token")}` },
});
setHistoryData(response.data.data || []);
} catch (err) {
  console.error("Gagal memuat riwayat:", err);
}
};

const fetchLatestAlatData = async () => {
  setIsFetchingAlatData(true);
  setAlatDataError(null);
  try {
    const response = await
axios.get(`${API_BASE_URL}/stunting/alat/latest`);
    const { tinggi_badan, berat_badan } = response.data.data;

    if (tinggi_badan && berat_badan) {
      setFormData(prev => ({
        ...prev,
        tinggi_badan: tinggi_badan.toString(),
        berat_badan: berat_badan.toString()
      }));
    } else {
      setAlatDataError("Data dari alat tidak lengkap");
    }
  } catch (err) {
    console.error("Gagal mengambil data alat:", err);
    setAlatDataError("Gagal mengambil data dari alat");
  } finally {
    setIsFetchingAlatData(false);
  }
};

const handleNamaAnakChange = (e) => {
  const selectedNamaAnak = e.target.value;
  const anak = anakList.find((a) => a && a.nama_anak ===
selectedNamaAnak);
  if (anak) {
    setFormData({
      ...formData,
      id_anak: anak.id,
      nama_anak: anak.nama_anak,

```

```

        nik: anak.nik,
        nama_orang_tua: anak.nama_orang_tua,
        jenis_kelamin: anak.jenis_kelamin,
    });
} else {
    setFormData({
        ...formData,
        id_anak: "",
        nama_anak: selectedNamaAnak,
        nik: "",
        nama_orang_tua: "",
        jenis_kelamin: "",
    });
}
};

const handleSubmit = async (e) => {
    e.preventDefault();
    setError(null);
    setPredictionResult(null);
    setIsLoading(true);

    try {
        const payload = {
            ...formData,
            umur: parseFloat(formData.umur),
            tinggi_badan: parseFloat(formData.tinggi_badan),
            berat_badan: parseFloat(formData.berat_badan),
        };

        const response = await axios.post(
            `${API_BASE_URL}/stunting/predict`,
            payload,
            {
                headers: {
                    Authorization: `Bearer
${localStorage.getItem("token")}`,
                    "Content-Type": "application/json",
                },
            }
        );

        setPredictionResult(response.data.data);
        fetchPredictionHistory();
    } catch (err) {

```

```

        setError(err.response?.data?.message || "Terjadi kesalahan
saat memprediksi");
    } finally {
        setIsLoading(false);
    }
};

const getStuntingStatus = (zScore) => {
    const isStunting = zScore < STUNTING_THRESHOLD;
    return {
        status: isStunting ? "Stunting" : "Tidak Stunting",
        variant: isStunting ? "danger" : "success",
        textClass: isStunting ? "text-danger" : "text-success",
        badge: isStunting ? (
            <Badge bg="danger">Stunting</Badge>
        ) : (
            <Badge bg="success">Tidak Stunting</Badge>
        ),
    };
};

return (
    <div className="d-flex">
        <SideBidan />
        <Container className="py-4">
            <Card className="shadow-sm mb-4">
                <Card.Header className="bg-primary text-white">
                    <h4 className="mb-0">Prediksi Stunting</h4>
                </Card.Header>
                <Card.Body>
                    {error && <Alert variant="danger">{error}</Alert>}
                    {alatDataError && <Alert
variant="warning">{alatDataError}</Alert>}

                    <Form onSubmit={handleSubmit}>
                        <Row>
                            <Col md={6}>
                                <Form.Group className="mb-3">
                                    <Form.Label>Pilih Anak</Form.Label>
                                    <Form.Select
                                        name="nama_anak"
                                        value={formData.nama_anak}
                                        onChange={handleNamaAnakChange}
                                        required
                                    >

```



```

        <option value="">Pilih Nama Anak</option>
        {anakList.map((anak) => (
            <option key={anak.id}
value={anak.nama_anak}>
                {anak.nama_anak}
            </option>
        ))}
        </Form.Select>
    </Form.Group>
</Col>

    <Col md={6}>
        <Form.Group className="mb-3">
            <Form.Label>NIK</Form.Label>
            <Form.Control type="text" value={formData.nik}
readOnly />
        </Form.Group>
    </Col>

    <Col md={6}>
        <Form.Group className="mb-3">
            <Form.Label>Nama Orang Tua</Form.Label>
            <Form.Control type="text"
value={formData.nama_orang_tua} readOnly />
        </Form.Group>
    </Col>

    <Col md={6}>
        <Form.Group className="mb-3">
            <Form.Label>Jenis Kelamin</Form.Label>
            <Form.Select
name="jenis_kelamin"
value={formData.jenis_kelamin}
onChange={(e) => setFormData({ ...formData, jenis_kelamin:
e.target.value })}
required
>
                <option value="">Pilih Jenis Kelamin</option>
                <option value="1">Laki-laki</option>
                <option value="2">Perempuan</option>
            </Form.Select>
        </Form.Group>
    </Col>

    <Col md={6}>

```

```

<Form.Group className="mb-3">
  <Form.Label>Umur (bulan)</Form.Label>
  <Form.Control
    type="number"
    name="umur"
    value={formData.umur}
    onChange={(e) => setFormData({ ...formData,
umur: e.target.value })}
    min="0"
    max="60"
    required
  />
</Form.Group>
</Col>

<Col md={6}>
  <Form.Group className="mb-3">
    <Form.Label>Tinggi Badan (cm)</Form.Label>
    <div className="d-flex align-items-center">
      <Form.Control
        type="number"
        name="tinggi_badan"
        value={formData.tinggi_badan}
        readOnly
      />
      <Button
        variant="outline-secondary"
        onClick={fetchLatestAlatData}
        disabled={isFetchingAlatData}
        className="ms-2"
      >
        {isFetchingAlatData ? (
          <Spinner animation="border" size="sm" />
        ) : (
          "Refresh"
        )}
      </Button>
    </div>
  </Form.Group>
</Col>

<Col md={6}>
  <Form.Group className="mb-3">
    <Form.Label>Berat Badan (kg)</Form.Label>
    <Form.Control

```

```

        type="number"
        name="berat_badan"
        value={formData.berat_badan}
        readOnly
      />
    </Form.Group>
  </Col>
</Row>

  <div className="d-flex justify-content-between mt-3">
    <Button variant="secondary" onClick={() =>
setShowHistoryModal(true)}>
      Lihat Riwayat
    </Button>
    <Button variant="primary" type="submit"
disabled={isLoading || isFetchingAlatData}>
      {isLoading ? (
        <>
          <Spinner animation="border" size="sm" />
          <span className="ms-2">Memproses...</span>
        </>
      ) : (
        "Prediksi"
      )}
    </Button>
  </div>
</Form>

  {predictionResult && (
    <Alert
variant={getStuntingStatus(predictionResult.z_score).variant}
className="mt-4">
      <h5>Anak {formData.nama_anak} terdeteksi{" "}
        {getStuntingStatus(predictionResult.z_score).statu
s} dengan Z-score: {predictionResult.z_score.toFixed(2)}
      </h5>
      <div>{getStuntingStatus(predictionResult.z_score).ba
dge}</div>
      <ul>
        {stuntingRecommendations.map((rec, index) => (
          <li key={index}>{rec}</li>
        ))}
      </ul>
    </Alert>
  )}

```

```
        })  
      </Card.Body>  
    </Card>  
  </Container>  
</div>  
);  
};  
  
export default StuntingPrediction
```