Laporan Tugas Besar
Algoritma & Pemograman

Program Kontrol Pencegahan Stunting
Kecamatan Mandalajati

Oleh :

Maitsa Luthfiyyah Durrotunnisa 1202223002

SI 46 01

Telkom Univeristy

Bandung

BAB I
Pendahuluan

1.1. Latar Belakang

Stunting adalah kondisi dimana anak memiliki tinggi badan yang lebih pendek dari standar usianya karena kekurangan gizi dalam jangka panjang. Kekurangan gizi ini bisa terjadi pada ibu saat hamil atau anak selama masa pertumbuhannya.

Menurut WHO, negara yang memiliki jumlah kasus stunting di atas 20% dianggap mengalami masalah stunting. Pada tahun 2018, Indonesia memiliki jumlah kasus stunting sebesar 30,8%, kementrian Kesehatan mengumumkan hasil survei status gizi Indonesia memang mengalami peurunan yaitu pada tahun 2021 24,4% menjadi 21,6% di tahun 2022, tapi nayatanya angka yang kita miliki masih diatas yang WHO berikan, makan kita tidak boleh lengah, perlu melakukan pengontrolan terhadap pola pertumbuhan anak dan pememuhan gizi terhadap mereka.

Bagaimana pun mereka adalah agen pembangun bangsanya suatu saat nanti, bangsa ini akan mengalami kehancuran jika para penerusnya tidak dalam kondisi yang baik, maka sangat perlu ada pengontrolan agar kita mengerti bagaimana kondisi anak saat ini, apa yang mereka butuhkan.

1.2.Pengertian dan Penjelasan Ide

Screening menjadi tahapan penting dalam pencegahan stunting. Dengan mengetahui kondisi anak, langkah-langkah selanjutnya dapat diambil dengan lebih mudah. Oleh karena itu, program ini hadir dengan tujuan melakukan pendataan secara digital agar hasilnya dapat diketahui dengan cepat.

Stunting sangat terkait erat dengan kondisi ibu dan anak, sehingga program ini menyediakan fitur pendataan anak dan ibu. Pada fitur anak, terdapat beberapa data yang perlu dilengkapi seperti nama, tanggal lahir, berat badan, tinggi badan, dan proses ASI.

Data-data ini menjadi sangat penting dalam proses screening karena indikasi stunting dapat dilihat dari berat badan dan tinggi badan, meskipun tidak dapat langsung disimpulkan. Oleh karena itu, penting untuk mendata informasi tentang ibunya juga. Beberapa data penting yang perlu didata adalah pendidikan, pekerjaan, dan penghasilan ibu. Pendidikan akan mencerminkan pemahaman ibu dalam mendidik dan merawat anak, sedangkan pekerjaan dan penghasilan akan menggambarkan kemampuan keluarga untuk memenuhi kebutuhan anak.

BAB II
Pembahasan

2.1. Source Code

2.1.1. Menu Utama

```
import tkinter as tk
import tkinter.ttk as ttk
import sqlite3
import matplotlib.pyplot as plt
import numpy as np
from PIL import ImageTk, Image

# Membuat GUI
root = tk.Tk()
root.title('Anak Sehat')
gambar = ImageTk.PhotoImage(file='Frame_3__5_-removebg-preview.png')

# Membuat frame utama
main_frame = tk.Frame(root)
main_frame.pack(fill=tk.BOTH, expand=True, padx=20, pady=20)

# Membuat judul
judul_label = tk.Label(main_frame, image=gambar)
judul_label.pack(side=tk.TOP, padx=10, pady=10)

# Membuat button untuk menampilkan tabel anak
button_tabel_anak = tk.Button(main_frame, bg='#0BB4AC' ,text="Data Anak",font=("Inika", 16),
command=tabel_anak, border=8,)
button_tabel_anak.pack(pady=10)

# Membuat button untuk menampilkan tabel ibu
button_tabel_ibu = tk.Button(main_frame, bg='#0BB4AC' ,text="Data Ibu", font=("Inika", 16),command=tabel_ibu,
border=8)
button_tabel_ibu.pack(pady=10)

create_table_anak()
create_table_ibu()

root.mainloop()
```

2.1.2. Membuat Tabel Anak

```
# Fungsi untuk membuat tabel anak
def create_table_anak():
    conn = sqlite3.connect('data_stunting.db')
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS anak(
                    id INTEGER PRIMARY KEY,
                    id_ibu INTEGER,
                    nama_anak TEXT,
                    tanggal_lahir TEXT,
                    tempat_lahir TEXT,
                    usia INTEGER,
                    alamat TEXT,
                    kelurahan TEXT,
                    berat_badan REAL,
                    tinggi_badan REAL,
                    ASI TEXT,
                    FOREIGN KEY (id_ibu) REFERENCES ibu (id_ibu)
                )''')
    conn.commit()
    conn.close()
```

2.1.3. Membuat Tabel Ibu

```python
# Fungsi untuk membuat tabel ibu
def create_table_ibu():
    conn = sqlite3.connect('data_stunting.db')
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS ibu(
                    id_ibu INTEGER PRIMARY KEY,
                    nama_ibu TEXT,
                    tanggal_lahir TEXT,
                    tempat_lahir TEXT,
                    usia INTEGER,
                    alamat TEXT,
                    kelurahan TEXT,
                    pendidikan TEXT,
                    pekerjaan TEXT,
                    penghasilan TEXT
                )''')
    conn.commit()
    conn.close()
```

2.1.4. Tabel Ibu

```python
def tabel_ibu():
    # Fungsi untuk menambahkan data ke database
    def insert_data():
        id_ibu = entry_id.get()
        nama = entry_nama.get()
        tanggal_lahir = entry_tanggal_lahir.get()
        tempat_lahir = entry_tempat_lahir.get()
        usia = int(entry_usia.get())
        alamat = entry_alamat.get()
        kelurahan = kelurahan_combobox.get()
        pendidikan = pendidikan_combobox.get()
        pekerjaan = pekerjaan_combobox.get()
        penghasilan = penghasilan_combobox.get()

        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('''INSERT INTO ibu(id_ibu, nama_ibu, tanggal_lahir, tempat_lahir, usia, alamat, kelurahan,
pendidikan, pekerjaan, penghasilan)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)''',
                (id_ibu, nama, tanggal_lahir, tempat_lahir, usia, alamat, kelurahan, pendidikan, pekerjaan,
penghasilan))
        conn.commit()
        conn.close()

        show_data()

        # Mengosongkan input fields setelah data ditambahkan
        entry_id.delete(0, 'end')
        entry_nama.delete(0, 'end')
        entry_tanggal_lahir.delete(0, 'end')
        entry_tempat_lahir.delete(0, 'end')
        entry_usia.delete(0, 'end')
        entry_alamat.delete(0, 'end')
```

```python
# Fungsi untuk menghapus data dari database
    def delete_data():
        selected_item = table.selection()
        if not selected_item:
            return

        item_id = table.item(selected_item)['text']

        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('DELETE FROM ibu WHERE id_ibu=?', (item_id,))
        conn.commit()
        conn.close()

        show_data()
```

```python
def show_data():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('SELECT * FROM ibu')
        rows = c.fetchall()
        conn.close()

        # Menghapus data yang ada di tabel sebelum menampilkan data baru
        table.delete(*table.get_children())

        for row in rows:
            table.insert('', 'end', text=row[0], values=row[1:])
```

```python
def refresh_table():
        # Menghapus semua baris dalam tabel
        for row in table.get_children():
            table.delete(row)

        # Mengambil data dari tabel
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute("SELECT * FROM ibu")
        data = c.fetchall()
        conn.close()

        # Menambahkan data ke dalam tabel
        for row in data:
            table.insert('', 'end', values=row)
```

```python
def chart_pendidikan():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()

        # Mengambil data ASI dari tabel anak
        c.execute("SELECT COUNT(*) FROM ibu WHERE pendidikan = 'SD'")
        data_SD = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pendidikan = 'SLTP'")
        data_SLTP = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pendidikan = 'SLTA'")
        data_SLTA = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pendidikan = 'Diploma I/II/III/IV'")
        data_Diploma = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pendidikan = 'Sarjana I/II/III'")
        data_Sarjana = c.fetchone()[0]

        conn.close()

        # Membuat diagram lingkaran
        labels = ['SD', 'SLTP', 'SLTA', 'Diploma I/II/III/IV', 'Sarjana I/II/III']
        sizes = [data_SD, data_SLTP, data_SLTA, data_Diploma, data_Sarjana]
        colors = ['red', 'blue', 'green', 'yellow', 'pink']
        explode = (0.1, 0, 0, 0, 0)

        plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
        plt.axis('equal')
        plt.show()
```

```python
def chart_pekerjaan():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()

        # Mengambil data ASI dari tabel anak
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Ibu Rumah Tangga'")
        data_IRT = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'PNS'")
        data_PNS = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Guru'")
        data_Guru = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Pedagang'")
        data_Pedagang = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Buruh Harian Lepas'")
        data_BHL = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Swasta'")
        data_Swasta = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM ibu WHERE pekerjaan = 'Lainnya'")
        data_lainnya = c.fetchone()[0]

        conn.close()

        # Membuat diagram lingkaran
        labels = ['Ibu Rumah Tangga', 'PNS', 'Guru', 'Pedagang', 'Buruh Harian Lepas', 'Swasta', 'Lainnya']
        sizes = [data_IRT, data_PNS, data_Guru, data_Pedagang, data_BHL, data_Swasta, data_lainnya]
        colors = ['red', 'blue', 'green', 'yellow', 'pink', 'purple', 'orange']
        explode = (0.1, 0, 0, 0, 0, 0, 0)

        plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
        plt.axis('equal')
        plt.show()
```

```python
def visualisasi_penghasilan():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()

        # Mengambil data penghasilan terbanyak di setiap kelurahan
        c.execute("SELECT kelurahan, penghasilan, COUNT(*) as jumlah FROM ibu GROUP BY kelurahan, penghasilan")
        data = c.fetchall()

        conn.close()

        # Mengelompokkan data penghasilan berdasarkan kelurahan
        penghasilan_kelurahan = {}
        for row in data:
            kelurahan = row[0]
            penghasilan = row[1]
            jumlah = row[2]
            if kelurahan not in penghasilan_kelurahan:
                penghasilan_kelurahan[kelurahan] = {}
            penghasilan_kelurahan[kelurahan][penghasilan] = jumlah

        # Menyiapkan data untuk grafik
        penghasilan_labels = ['Kurang dari Rp 1.000.000', 'Rp 1.000.000 - Rp 3.000.000', 'Rp 3.000.000 - Rp
5.000.000', 'Rp 5.000.000 - Rp 10.000.000', 'Lebih dari Rp 10.000.000']
        kelurahan_data = []
        for kelurahan, penghasilan_data in penghasilan_kelurahan.items():
            penghasilan_jumlah = [penghasilan_data.get(label, 0) for label in penghasilan_labels]
            kelurahan_data.append(penghasilan_jumlah)

        # Menggambar grafik
        kelurahan_labels = list(penghasilan_kelurahan.keys())
        x = np.arange(len(kelurahan_labels))
        width = 0.15
        colors = ['red', 'blue', 'green', 'yellow', 'pink']

        fig, ax = plt.subplots()
        for i in range(len(penghasilan_labels)):
            ax.bar(x + (i * width), [data[i] for data in kelurahan_data], width, label=penghasilan_labels[i],
color=colors[i])

        ax.set_xlabel('Kelurahan')
        ax.set_ylabel('Jumlah Ibu')
        ax.set_title('Penghasilan Terbanyak di Setiap Kelurahan')
        ax.set_xticks(x)
        ax.set_xticklabels(kelurahan_labels)
        ax.legend()

        plt.show()
```

```
    tabel_ibu_frame = tk.Toplevel(root)
    gambar = ImageTk.PhotoImage(file='Frame_3__9_-removebg-preview.png')

    # Membuat frame untuk input data anak
    input_ibu_frame = tk.Frame(tabel_ibu_frame)
    input_ibu_frame.pack(side=tk.LEFT, fill=tk.BOTH, padx=10)

    # Membuat judul tabel anak
    judul_label = tk.Label(input_ibu_frame, image=gambar)
    judul_label.pack(side=tk.TOP, padx=10, pady=10)
    judul_label.image = gambar


    # Membuat frame tabel
    table_frame = tk.Frame(tabel_ibu_frame)
    table_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

    # Membuat tabel
    table = ttk.Treeview(table_frame, columns=('id_ibu', 'nama_ibu', 'tanggal_lahir', 'tempat_lahir', 'usia',
'alamat', 'kelurahan', 'pendidikan', 'pekerjaan', 'penghasilan'), show='headings')
    table.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

    # Mengatur lebar kolom tabel
    table.column('id_ibu',width=90)
    table.column('nama_ibu', width=90)
    table.column('tanggal_lahir', width=90)
    table.column('tempat_lahir', width=90)
    table.column('usia', width=90)
    table.column('alamat', width=90)
    table.column('kelurahan', width=90)
    table.column('pendidikan', width=90)
    table.column('pekerjaan', width=90)
    table.column('penghasilan', width=90)

    # Mengatur nama header kolom tabel
    table.heading('id_ibu', text='ID')
    table.heading('nama_ibu', text='Nama')
    table.heading('tanggal_lahir', text='Tanggal Lahir')
    table.heading('tempat_lahir', text='Tempat Lahir')
    table.heading('usia', text='Usia')
    table.heading('alamat', text='Alamat')
    table.heading('kelurahan', text='Kelurahan')
    table.heading('pendidikan', text='Pendidikan')
    table.heading('pekerjaan', text='Pekerjaan')
    table.heading('penghasilan', text='Penghasilan')

    # Membuat scrollbar untuk tabel
    scrollbar = ttk.Scrollbar(table_frame, orient='vertical', command=table.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
    table.configure(yscrollcommand=scrollbar.set)
```

```
# Membuat label dan entry untuk id Ibu
    label_id = tk.Label(input_ibu_frame, text='ID Ibu:')
    label_id.pack(anchor='w')
    entry_id = tk.Entry(input_ibu_frame, width=50)
    entry_id.pack(anchor='w')

    # Membuat label dan entry untuk Nama Ibu
    label_nama = tk.Label(input_ibu_frame, text='Nama:')
    label_nama.pack(anchor='w')
    entry_nama = tk.Entry(input_ibu_frame, width=50)
    entry_nama.pack(anchor='w')

    # Membuat label dan entry untuk Tanggal Lahir Ibu
    label_tanggal_lahir = tk.Label(input_ibu_frame, text='Tanggal Lahir:')
    label_tanggal_lahir.pack(anchor='w')
    entry_tanggal_lahir = tk.Entry(input_ibu_frame, width=50)
    entry_tanggal_lahir.pack(anchor='w')

    # Membuat label dan entry untuk Tempat Lahir ibu
    label_tempat_lahir = tk.Label(input_ibu_frame, text='Tempat Lahir:')
    label_tempat_lahir.pack(anchor='w')
    entry_tempat_lahir = tk.Entry(input_ibu_frame, width=50)
    entry_tempat_lahir.pack(anchor='w')

    # Membuat label dan entry untuk Usia
    label_usia = tk.Label(input_ibu_frame, text='Usia:')
    label_usia.pack(anchor='w')
    entry_usia = tk.Entry(input_ibu_frame, width=50)
    entry_usia.pack(anchor='w')

    # Membuat label dan entry untuk Alamat
    label_alamat = tk.Label(input_ibu_frame, text='Alamat:')
    label_alamat.pack(anchor='w')
    entry_alamat = tk.Entry(input_ibu_frame, width=50)
    entry_alamat.pack(anchor='w')

    # Membuat label dan combobox untuk Kelurahan ibu
    label_kelurahan = tk.Label(input_ibu_frame, text='Kelurahan:')
    label_kelurahan.pack(anchor='w')
    kelurahan_combobox = ttk.Combobox(input_ibu_frame, state='readonly',  width=47)
    kelurahan_combobox.pack(anchor='w')
    kelurahan_combobox['values'] = ('Kelurahan Karang Pamulang', 'Kelurahan Pasir Impun', 'Kelurahan
Sindangjaya')

    # Membuat label dan combobox untuk Pendidikan ibu
    label_pendidikan = tk.Label(input_ibu_frame, text='Pendidikan:')
    label_pendidikan.pack(anchor='w')
    pendidikan_combobox = ttk.Combobox(input_ibu_frame, state='readonly',  width=47)
    pendidikan_combobox.pack(anchor='w')
    pendidikan_combobox['values'] = ('SD', 'SLTP', 'SLTA', 'Diploma I/II/III/IV', 'Sarjana I/II/III')

    # Membuat label dan combobox untuk Pekerjaan ibu
    label_pekerjaan = tk.Label(input_ibu_frame, text='Pekerjaan:')
    label_pekerjaan.pack(anchor='w')
    pekerjaan_combobox = ttk.Combobox(input_ibu_frame, state='readonly',  width=47)
    pekerjaan_combobox.pack(anchor='w')
    pekerjaan_combobox['values'] = ('Ibu Rumah Tangga', 'PNS', 'Guru', 'Pedagang', 'Buruh Harian Lepas',
'Swasta','Lainnya')

    # Membuat label dan combobox untuk Penghasilan ibu
    label_penghasilan = tk.Label(input_ibu_frame, text='Penghasilan:')
    label_penghasilan.pack(anchor='w')
    penghasilan_combobox = ttk.Combobox(input_ibu_frame, state='readonly',  width=47)
    penghasilan_combobox.pack(anchor='w')
    penghasilan_combobox['values'] = ('Kurang dari Rp 1.000.000', 'Rp 1.000.000 - Rp 3.000.000', 'Rp 3.000.000
- Rp 5.000.000', 'Rp 5.000.000 - Rp 10.000.000', 'Lebih dari Rp 10.000.000')
```

```
#membuat button
    button_frame1 = tk.Frame(input_ibu_frame)
    button_frame1.pack(side=tk.TOP, padx=10, pady=10)

    button_insert_ibu = tk.Button(button_frame1, text="Insert", command=insert_data, bg='#CFDC2F')
    button_insert_ibu.pack(side=tk.LEFT, padx=5)

    button_delete_ibu = tk.Button(button_frame1, text="Delete", command=delete_data,bg='#CFDC2F')
    button_delete_ibu.pack(side=tk.LEFT, padx=5)

    button_frame2 = tk.Frame(input_ibu_frame)
    button_frame2.pack(side=tk.TOP, padx=10, pady=10)

    button_pendidikan = tk.Button(button_frame2, text="Pendidikan", command=chart_pendidikan, bg='#0BB4AC')
    button_pendidikan.pack(side=tk.LEFT, padx=5)

    button_pekerjaan = tk.Button(button_frame2, text="Pekerjaan", command=chart_pekerjaan, bg='#0BB4AC')
    button_pekerjaan.pack(side=tk.LEFT, padx=5)

    button_penghasilan = tk.Button(button_frame2, text="Penghasilan", command=visualisasi_penghasilan,
bg='#0BB4AC')
    button_penghasilan.pack(side=tk.LEFT, padx=5)

    refresh_table()
```

2.1.5. Tabel Anak

```python
def tabel_anak():
    # Fungsi untuk menambahkan data ke database
    def insert_data():
        id = entry_id.get()
        id_ibu = entry_id_ibu.get()
        nama = entry_nama.get()
        tanggal_lahir = entry_tanggal_lahir.get()
        tempat_lahir = entry_tempat_lahir.get()
        usia = int(entry_usia.get())
        alamat = entry_alamat.get()
        kelurahan = kelurahan_combobox.get()
        berat_badan = float(entry_berat_badan.get())
        tinggi_badan = float(entry_tinggi_badan.get())
        asi = ASI_combobox.get()

        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('''INSERT INTO anak (id, id_ibu, nama_anak, tanggal_lahir, tempat_lahir, usia, alamat,
kelurahan, berat_badan, tinggi_badan, ASI)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)''',
                (id, id_ibu, nama, tanggal_lahir, tempat_lahir, usia, alamat, kelurahan, berat_badan,
tinggi_badan, asi))
        conn.commit()
        conn.close()

        # Mengambil ID data yang baru ditambahkan
        item_id = c.lastrowid

        conn.close()

        show_data()

        # Mengosongkan input fields setelah data ditambahkan
        entry_id.delete(0, 'end')
        entry_id_ibu.delete(0, 'end')
        entry_nama.delete(0, 'end')
        entry_tanggal_lahir.delete(0, 'end')
        entry_tempat_lahir.delete(0, 'end')
        entry_usia.delete(0, 'end')
        entry_alamat.delete(0, 'end')
        entry_berat_badan.delete(0, 'end')
        entry_tinggi_badan.delete(0, 'end')
```

```python
# Fungsi untuk menghapus data dari database
    def delete_data():
        selected_item = table.selection()
        if not selected_item:
            return

        item_id = table.item(selected_item)['text']

        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('DELETE FROM anak WHERE id=?', (item_id,))
        conn.commit()
        conn.close()

        show_data()
```

```python
def show_data():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('SELECT * FROM anak')
        rows = c.fetchall()
        conn.close()

        # Menghapus data yang ada di tabel sebelum menampilkan data baru
        table.delete(*table.get_children())

        for row in rows:
            table.insert('', 'end', text=row[0], values=row[1:])
```

```python
def refresh_table():
        # Menghapus semua baris dalam tabel
        for row in table.get_children():
            table.delete(row)

        # Mengambil data dari tabel
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute("SELECT * FROM anak")
        data = c.fetchall()
        conn.close()

        # Menambahkan data ke dalam tabel
        for row in data:
            table.insert('', 'end', values=row)
```

```python
def chart_berat():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('SELECT kelurahan, AVG(berat_badan) FROM anak GROUP BY kelurahan')
        rows = c.fetchall()
        conn.close()

        kelurahan = []
        berat_badan = []

        for row in rows:
            kelurahan.append(row[0])
            berat_badan.append(row[1])


        # Menampilkan visualisasi rata-rata berat
        plt.figure(figsize=(8, 6))
        plt.bar(kelurahan, berat_badan, label='Berat Badan')
        plt.ylim(6, 15)
        plt.xlabel('Kelurahan')
        plt.ylabel('Rata-rata')
        plt.title('Rata-rata Berat Badan Anak')
        plt.legend()
        plt.show()
```

```python
def chart_tinggi():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()
        c.execute('SELECT kelurahan, AVG(tinggi_badan) FROM anak GROUP BY kelurahan')
        rows = c.fetchall()
        conn.close()

        kelurahan = []
        tinggi_badan = []

        for row in rows:
            kelurahan.append(row[0])
            tinggi_badan.append(row[1])


        # Menampilkan visualisasi rata-rata berat
        plt.figure(figsize=(8, 6))
        plt.bar(kelurahan, tinggi_badan, label='Tinggi Badan')
        plt.xlabel('Kelurahan')
        plt.ylabel('Rata-rata')
        plt.ylim(50, 85)
        plt.title('Rata-rata Tinggi Badan Anak')
        plt.legend()
        plt.show()
```

```python
def chart_ASI():
        conn = sqlite3.connect('data_stunting.db')
        c = conn.cursor()

        # Mengambil data ASI dari tabel anak
        c.execute("SELECT COUNT(*) FROM anak WHERE ASI = 'ASI Eksklusif'")
        data_asi_eksklusif = c.fetchone()[0]
        c.execute("SELECT COUNT(*) FROM anak WHERE ASI = 'ASI Non Eksklusif'")
        data_asi_non_eksklusif = c.fetchone()[0]

        conn.close()

        # Membuat diagram lingkaran
        labels = ['ASI Eksklusif', 'ASI Non Eksklusif']
        sizes = [data_asi_eksklusif, data_asi_non_eksklusif]
        colors = ['#ff9999', '#66b3ff']
        explode = (0.1, 0)

        plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
        plt.axis('equal')
        plt.show()
```

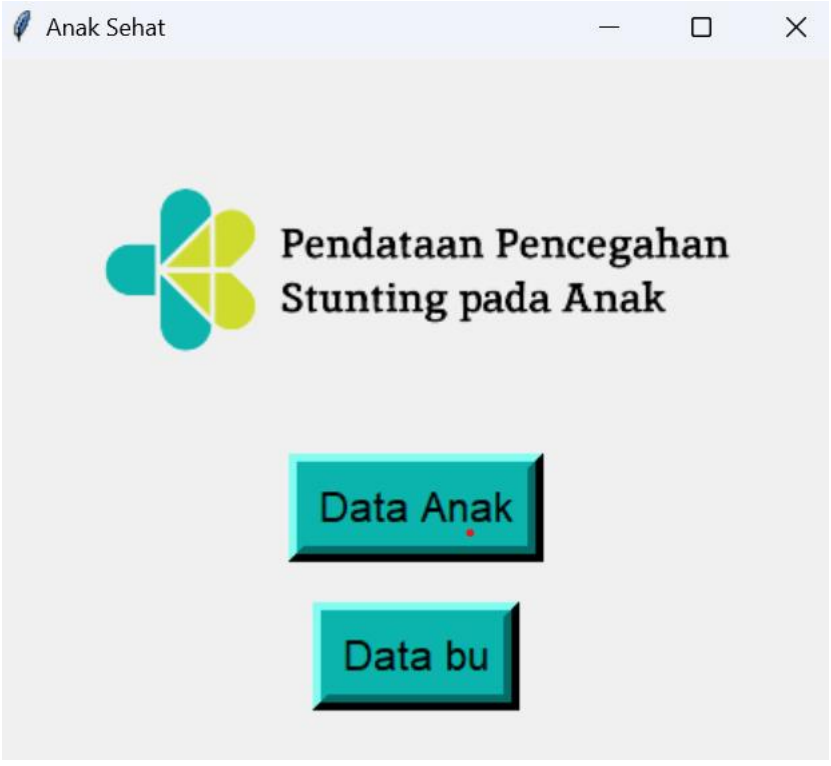
```python
tabel_anak_frame = tk.Toplevel(root)
    gambar = ImageTk.PhotoImage(file='Frame_3__6_-removebg-preview.png')

    # Membuat frame untuk input data anak
    input_anak_frame = tk.Frame(tabel_anak_frame)
    input_anak_frame.pack(side=tk.LEFT, fill=tk.BOTH, padx=10)

    # Membuat judul tabel anak
    judul_label = tk.Label(input_anak_frame, image=gambar)
    judul_label.pack(side=tk.TOP, padx=10, pady=10)
    judul_label.image = gambar

    # Membuat frame tabel
    table_frame = tk.Frame(tabel_anak_frame)
    table_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

    # Membuat tabel
    table = ttk.Treeview(table_frame, columns=('id','id_ibu', 'nama_anak', 'tanggal_lahir', 'tempat_lahir',
'usia', 'alamat', 'kelurahan', 'berat_badan', 'tinggi_badan', 'ASI'), show='headings')
    table.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

    # Mengatur lebar kolom tabel
    table.column('id',width=90)
    table.column('id_ibu',width=90)
    table.column('nama_anak', width=90)
    table.column('tanggal_lahir', width=90)
    table.column('tempat_lahir', width=90)
    table.column('usia', width=90)
    table.column('alamat', width=90)
    table.column('kelurahan', width=90)
    table.column('berat_badan', width=90)
    table.column('tinggi_badan', width=90)
    table.column('ASI', width=90)

    # Mengatur nama header kolom tabel
    table.heading('id', text='ID Anak')
    table.heading('id_ibu', text='ID Ibu')
    table.heading('nama_anak', text='Nama')
    table.heading('tanggal_lahir', text='Tanggal Lahir')
    table.heading('tempat_lahir', text='Tempat Lahir')
    table.heading('usia', text='Usia')
    table.heading('alamat', text='Alamat')
    table.heading('kelurahan', text='Kelurahan')
    table.heading('berat_badan', text='Berat Badan')
    table.heading('tinggi_badan', text='Tinggi Badan')
    table.heading('ASI', text='ASI')

    # Membuat scrollbar untuk tabel
    scrollbar = ttk.Scrollbar(table_frame, orient='vertical', command=table.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
    table.configure(yscrollcommand=scrollbar.set)
```

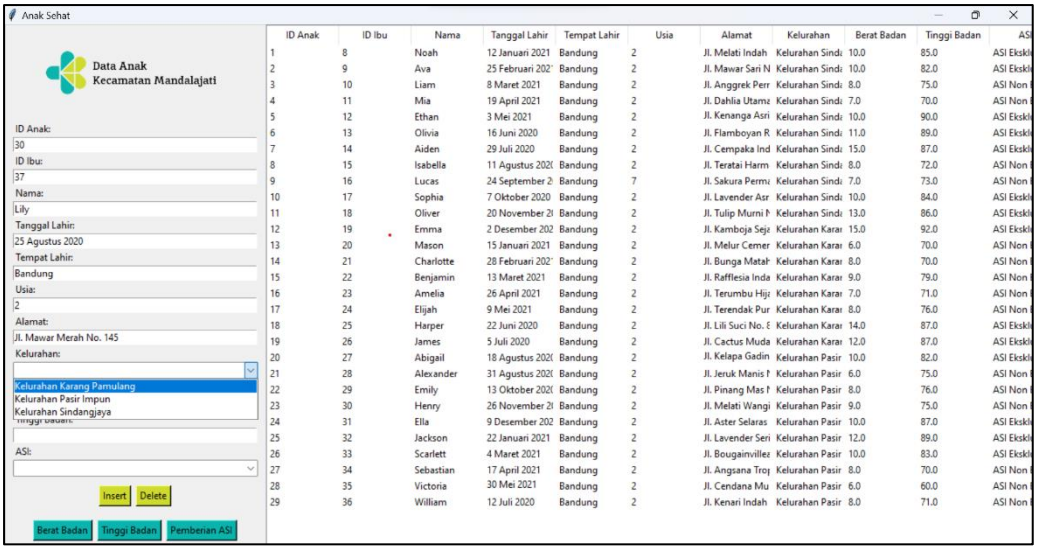
```
# Membuat label dan entry untuk id anak
    label_id = tk.Label(input_anak_frame, text='ID Anak:')
    label_id.pack(anchor='w')
    entry_id = tk.Entry(input_anak_frame, width=50)
    entry_id.pack(anchor='w')

    # Membuat label dan entry untuk id Ibu
    label_id_ibu = tk.Label(input_anak_frame, text='ID Ibu:')
    label_id_ibu.pack(anchor='w')
    entry_id_ibu = tk.Entry(input_anak_frame, width=50)
    entry_id_ibu.pack(anchor='w')

    # Membuat label dan entry untuk Nama
    label_nama = tk.Label(input_anak_frame, text='Nama:')
    label_nama.pack(anchor='w')
    entry_nama = tk.Entry(input_anak_frame, width=50)
    entry_nama.pack(anchor='w')

    # Membuat label dan entry untuk Tanggal Lahir
    label_tanggal_lahir = tk.Label(input_anak_frame, text='Tanggal Lahir:')
    label_tanggal_lahir.pack(anchor='w')
    entry_tanggal_lahir = tk.Entry(input_anak_frame, width=50)
    entry_tanggal_lahir.pack(anchor='w')

    # Membuat label dan entry untuk Tempat Lahir
    label_tempat_lahir = tk.Label(input_anak_frame, text='Tempat Lahir:')
    label_tempat_lahir.pack(anchor='w')
    entry_tempat_lahir = tk.Entry(input_anak_frame, width=50)
    entry_tempat_lahir.pack(anchor='w')

    # Membuat label dan entry untuk Usia
    label_usia = tk.Label(input_anak_frame, text='Usia:')
    label_usia.pack(anchor='w')
    entry_usia = tk.Entry(input_anak_frame, width=50)
    entry_usia.pack(anchor='w')

    # Membuat label dan entry untuk Alamat
    label_alamat = tk.Label(input_anak_frame, text='Alamat:')
    label_alamat.pack(anchor='w')
    entry_alamat = tk.Entry(input_anak_frame, width=50)
    entry_alamat.pack(anchor='w')

    # Membuat label dan combobox untuk Kelurahan ibu
    label_kelurahan = tk.Label(input_anak_frame, text='Kelurahan:')
    label_kelurahan.pack(anchor='w')
    kelurahan_combobox = ttk.Combobox(input_anak_frame, state='readonly',  width=47)
    kelurahan_combobox.pack(anchor='w')
    kelurahan_combobox['values'] = ('Kelurahan Karang Pamulang', 'Kelurahan Pasir Impun', 'Kelurahan
Sindangjaya')

    # Membuat label dan combobox untuk berat badan
    label_berat_badan = tk.Label(input_anak_frame, text='Berat Badan:')
    label_berat_badan.pack(anchor='w')
    entry_berat_badan = tk.Entry(input_anak_frame, width=50)
    entry_berat_badan.pack(anchor='w')

    # Membuat label dan combobox untuk tinggi badan
    label_tinggi_badan = tk.Label(input_anak_frame, text='Tinggi Badan:')
    label_tinggi_badan.pack(anchor='w')
    entry_tinggi_badan = tk.Entry(input_anak_frame, width=50)
    entry_tinggi_badan.pack(anchor='w')

    # Membuat label dan combobox untuk ASI
    label_ASI = tk.Label(input_anak_frame, text='ASI:')
    label_ASI.pack(anchor='w')
    ASI_combobox = ttk.Combobox(input_anak_frame, state='readonly',  width=47)
    ASI_combobox.pack(anchor='w')
    ASI_combobox['values'] = ('ASI Eksklusif', 'ASI Non Eksklusif')
```

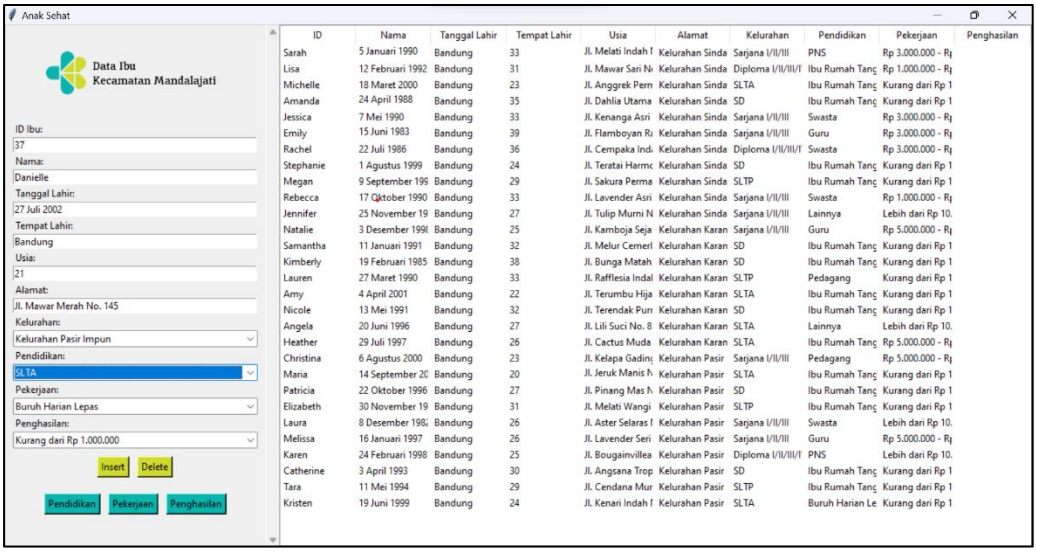
```
# Membuat button
    button_frame1 = tk.Frame(input_anak_frame)
    button_frame1.pack(side=tk.TOP, padx=10, pady=10)

    button_insert_anak = tk.Button(button_frame1, text="Insert", command=insert_data, bg='#CFDC2F')
    button_insert_anak.pack(side=tk.LEFT, padx=3, pady=1)

    button_delete_anak = tk.Button(button_frame1, text="Delete", command=delete_data, bg='#CFDC2F')
    button_delete_anak.pack(side=tk.LEFT, padx=3, pady=1)

    button_frame2 = tk.Frame(input_anak_frame)
    button_frame2.pack(side=tk.TOP, padx=10, pady=5)

    button_berat_anak = tk.Button(button_frame2, text="Berat Badan", command=chart_berat, bg='#0BB4AC')
    button_berat_anak.pack(side=tk.LEFT, padx=3, pady=1)

    button_tinggi_anak = tk.Button(button_frame2, text="Tinggi Badan", command=chart_tinggi,bg='#0BB4AC' )
    button_tinggi_anak.pack(side=tk.LEFT, padx=3, pady=1)

    button_ASI = tk.Button(button_frame2, text="Pemberian ASI", command=chart_ASI, bg='#0BB4AC')
    button_ASI.pack(side=tk.LEFT, padx=3, pady=1)

    refresh_table()
```

2.2. Output

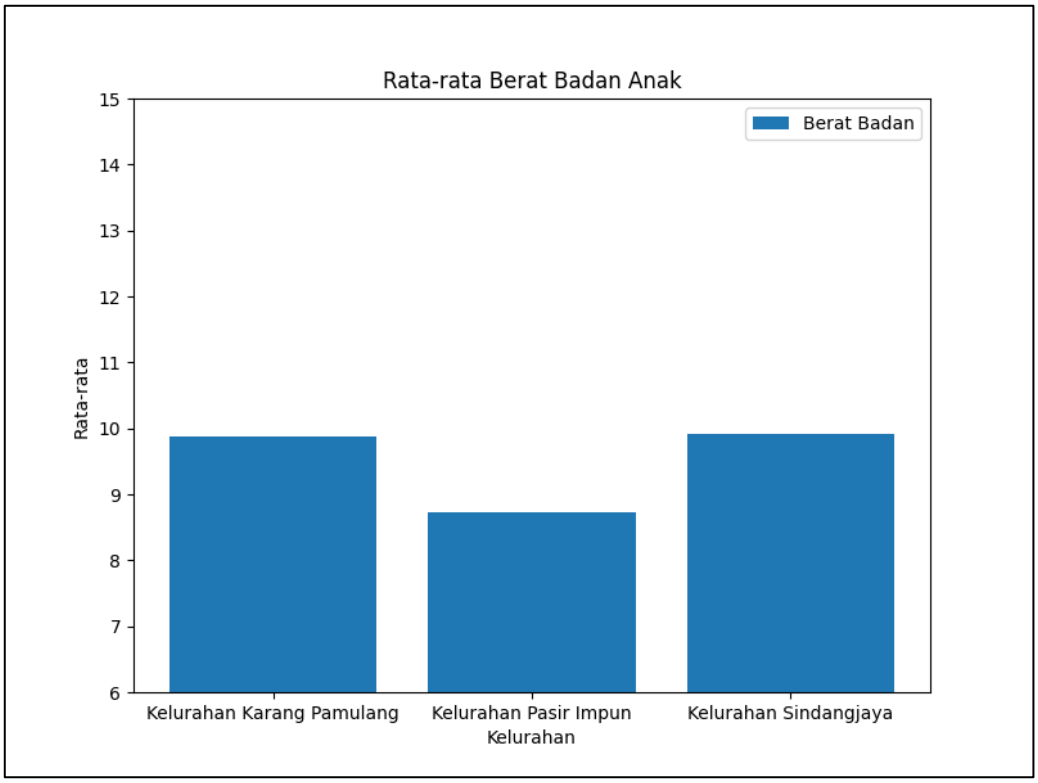
2.2.1. Tampilan GUI




Anak Sehat
Data Anak
Kecamatan Mandalajati
ID Anak:
30
ID Ibu:
37
Nama:
Lily
Tanggal Lahir:
25 Agustus 2020
Tempat Lahir:
Bandung
Usia:
2
Alamat:
Jl. Mawar Merah No. 145
Kelurahan:
Kelurahan Karang Pamulang
Kelurahan Pasir Impun
Kelurahan Sindangjaya
ASI:
Insert
Delete
Berat Badan
Tinggi Badan
Pemberian ASI
ID Anak | ID Ibu | Nama | Tanggal Lahir | Tempat Lahir | Usia | Alamat | Kelurahan | Berat Badan | Tinggi Badan | ASI
1 | 8 | Noah | 12 Januari 2021 | Bandung | 2 | Jl. Melati Indah | Kelurahan Sinda | 10.0 | 85.0 | ASI Ekskl
2 | 9 | Ava | 25 Februari 202 | Bandung | 2 | Jl. Mawar Sari N | Kelurahan Sinda | 10.0 | 82.0 | ASI Ekskl
3 | 10 | Liam | 8 Maret 2021 | Bandung | 2 | Jl. Anggrek Perr | Kelurahan Sinda | 8.0 | 75.0 | ASI Non
4 | 11 | Mia | 19 April 2021 | Bandung | 2 | Jl. Dahlia Utama | Kelurahan Sinda | 7.0 | 70.0 | ASI Non
5 | 12 | Ethan | 3 Mei 2021 | Bandung | 2 | Jl. Kenanga Asri | Kelurahan Sinda | 10.0 | 90.0 | ASI Ekskl
6 | 13 | Olivia | 16 Juni 2020 | Bandung | 2 | Jl. Flamboyan R | Kelurahan Sinda | 11.0 | 89.0 | ASI Ekskl
7 | 14 | Aiden | 29 Juli 2020 | Bandung | 2 | Jl. Cempaka Ind | Kelurahan Sinda | 15.0 | 87.0 | ASI Ekskl
8 | 15 | Isabella | 11 Agustus 202(| Bandung | 2 | Jl. Teratai Harm | Kelurahan Sinda | 8.0 | 72.0 | ASI Non
9 | 16 | Lucas | 24 September 2 | Bandung | 7 | Jl. Sakura Perm | Kelurahan Sinda | 7.0 | 73.0 | ASI Non
10 | 17 | Sophia | 7 Oktober 2020 | Bandung | 2 | Jl. Lavender Asr | Kelurahan Sinda | 10.0 | 84.0 | ASI Ekskl
11 | 18 | Oliver | 20 November 2(| Bandung | 2 | Jl. Tulip Murni N | Kelurahan Sinda | 13.0 | 86.0 | ASI Ekskl
12 | 19 | Emma | 2 Desember 202 | Bandung | 2 | Jl. Kamboja Seja | Kelurahan Karar | 15.0 | 92.0 | ASI Ekskl
13 | 20 | Mason | 15 Januari 2021 | Bandung | 2 | Jl. Melur Cemer | Kelurahan Karar | 6.0 | 70.0 | ASI Non
14 | 21 | Charlotte | 28 Februari 202 | Bandung | 2 | Jl. Bunga Matah | Kelurahan Karar | 8.0 | 70.0 | ASI Non
15 | 22 | Benjamin | 13 Maret 2021 | Bandung | 2 | Jl. Rafflesia Inda | Kelurahan Karar | 9.0 | 79.0 | ASI Non
16 | 23 | Amelia | 26 April 2021 | Bandung | 2 | Jl. Terumbu Hija | Kelurahan Karar | 7.0 | 71.0 | ASI Non
17 | 24 | Elijah | 9 Mei 2021 | Bandung | 2 | Jl. Terendak Pur | Kelurahan Karar | 8.0 | 76.0 | ASI Non
18 | 25 | Harper | 22 Juni 2020 | Bandung | 2 | Jl. Lili Suci No. 8 | Kelurahan Karar | 14.0 | 87.0 | ASI Ekskl
19 | 26 | James | 5 Juli 2020 | Bandung | 2 | Jl. Cactus Muda | Kelurahan Karar | 12.0 | 87.0 | ASI Ekskl
20 | 27 | Abigail | 18 Agustus 202(| Bandung | 2 | Jl. Kelapa Gadin | Kelurahan Pasir | 10.0 | 82.0 | ASI Ekskl
21 | 28 | Alexander | 31 Agustus 202(| Bandung | 2 | Jl. Jeruk Manis N | Kelurahan Pasir | 6.0 | 75.0 | ASI Non
22 | 29 | Emily | 13 Oktober 202(| Bandung | 2 | Jl. Pinang Mas N | Kelurahan Pasir | 8.0 | 76.0 | ASI Non
23 | 30 | Henry | 26 November 2(| Bandung | 2 | Jl. Melati Wangi | Kelurahan Pasir | 9.0 | 75.0 | ASI Non
24 | 31 | Ella | 9 Desember 202 | Bandung | 2 | Jl. Aster Selaras | Kelurahan Pasir | 10.0 | 87.0 | ASI Ekskl
25 | 32 | Jackson | 22 Januari 2021 | Bandung | 2 | Jl. Lavender Seri | Kelurahan Pasir | 12.0 | 89.0 | ASI Ekskl
26 | 33 | Scarlett | 4 Maret 2021 | Bandung | 2 | Jl. Bougainvillea | Kelurahan Pasir | 10.0 | 83.0 | ASI Ekskl
27 | 34 | Sebastian | 17 April 2021 | Bandung | 2 | Jl. Angsana Trop | Kelurahan Pasir | 8.0 | 70.0 | ASI Non
28 | 35 | Victoria | 30 Mei 2021 | Bandung | 2 | Jl. Cendana Mu | Kelurahan Pasir | 6.0 | 60.0 | ASI Non
29 | 36 | William | 12 Juli 2020 | Bandung | 2 | Jl. Kenari Indah | Kelurahan Pasir | 8.0 | 71.0 | ASI Non


Anak Sehat
Data Ibu
Kecamatan Mandalajati
ID Ibu:
37
Nama:
Danielle
Tanggal Lahir:
27 Juli 2002
Tempat Lahir:
Bandung
Usia:
21
Alamat:
Jl. Mawar Merah No. 145
Kelurahan:
Kelurahan Pasir Impun
Pendidikan:
SLTA
Pekerjaan:
Buruh Harian Lepas
Penghasilan:
Kurang dari Rp 1.000.000
Insert
Delete
Pendidikan
Pekerjaan
Penghasilan
ID | Nama | Tanggal Lahir | Tempat Lahir | Usia | Alamat | Kelurahan | Pendidikan | Pekerjaan | Penghasilan
Sarah | 5 Januari 1990 | Bandung | 33 | Jl. Melati Indah I | Kelurahan Sinda | Sarjana I/II/III | PNS | Rp 3.000.000 - Rp
Lisa | 12 Februari 1992 | Bandung | 31 | Jl. Mawar Sari No | Kelurahan Sinda | Diploma I/II/III/I | Ibu Rumah Tang | Rp 1.000.000 - Rp
Michelle | 18 Maret 2000 | Bandung | 23 | Jl. Anggrek Perm | Kelurahan Sinda | SLTA | Ibu Rumah Tang | Kurang dari Rp 1
Amanda | 24 April 1988 | Bandung | 35 | Jl. Dahlia Utama | Kelurahan Sinda | SD | Ibu Rumah Tang | Kurang dari Rp 1
Jessica | 7 Mei 1990 | Bandung | 33 | Jl. Kenanga Asri | Kelurahan Sinda | Sarjana I/II/III | Swasta | Rp 3.000.000 - Rp
Emily | 15 Juni 1983 | Bandung | 39 | Jl. Flamboyan Ra | Kelurahan Sinda | Sarjana I/II/III | Guru | Rp 3.000.000 - Rp
Rachel | 22 Juli 1986 | Bandung | 36 | Jl. Cempaka Ind | Kelurahan Sinda | Diploma I/II/III/I | Swasta | Rp 3.000.000 - Rp
Stephanie | 1 Agustus 1999 | Bandung | 24 | Jl. Teratai Harmo | Kelurahan Sinda | SD | Ibu Rumah Tang | Kurang dari Rp 1
Megan | 9 September 199 | Bandung | 29 | Jl. Sakura Perma | Kelurahan Sinda | SLTP | Ibu Rumah Tang | Kurang dari Rp 1
Rebecca | 17 Oktober 1990 | Bandung | 33 | Jl. Lavender Asri | Kelurahan Sinda | Sarjana I/II/III | Swasta | Rp 1.000.000 - Rp
Jennifer | 25 November 19 | Bandung | 27 | Jl. Tulip Murni N | Kelurahan Sinda | Sarjana I/II/III | Lainnya | Lebih dari Rp 10.
Natalie | 3 Desember 199(| Bandung | 25 | Jl. Kamboja Seja | Kelurahan Karan | Sarjana I/II/III | Guru | Rp 5.000.000 - Rp
Samantha | 11 Januari 1991 | Bandung | 32 | Jl. Melur Cemerl | Kelurahan Karan | SD | Ibu Rumah Tang | Kurang dari Rp 1
Kimberly | 19 Februari 1985 | Bandung | 38 | Jl. Bunga Matah | Kelurahan Karan | SD | Ibu Rumah Tang | Kurang dari Rp 1
Lauren | 27 Maret 1990 | Bandung | 33 | Jl. Rafflesia Indal | Kelurahan Karan | SLTP | Pedagang | Kurang dari Rp 1
Amy | 4 April 2001 | Bandung | 22 | Jl. Terumbu Hija | Kelurahan Karan | SLTA | Ibu Rumah Tang | Kurang dari Rp 1
Nicole | 13 Mei 1991 | Bandung | 32 | Jl. Terendak Pur | Kelurahan Karan | SD | Ibu Rumah Tang | Kurang dari Rp 1
Angela | 20 Juni 1996 | Bandung | 27 | Jl. Lili Suci No. 8 | Kelurahan Karan | SLTA | Lainnya | Lebih dari Rp 10.
Heather | 29 Juli 1997 | Bandung | 26 | Jl. Cactus Muda | Kelurahan Karan | SLTA | Ibu Rumah Tang | Rp 5.000.000 - Rp
Christina | 6 Agustus 2000 | Bandung | 23 | Jl. Kelapa Gading | Kelurahan Pasir | Sarjana I/II/III | Pedagang | Rp 5.000.000 - Rp
Maria | 14 September 20 | Bandung | 20 | Jl. Jeruk Manis N | Kelurahan Pasir | SLTA | Ibu Rumah Tang | Kurang dari Rp 1
Patricia | 22 Oktober 1996 | Bandung | 27 | Jl. Pinang Mas N | Kelurahan Pasir | SD | Ibu Rumah Tang | Kurang dari Rp 1
Elizabeth | 30 November 19 | Bandung | 31 | Jl. Melati Wangi | Kelurahan Pasir | SLTP | Ibu Rumah Tang | Kurang dari Rp 1
Laura | 8 Desember 198 | Bandung | 26 | Jl. Aster Selaras I | Kelurahan Pasir | Sarjana I/II/III | Swasta | Lebih dari Rp 10.
Melissa | 16 Januari 1997 | Bandung | 26 | Jl. Lavender Seri | Kelurahan Pasir | Sarjana I/II/III | Guru | Rp 5.000.000 - Rp
Karen | 24 Februari 1998 | Bandung | 25 | Jl. Bougainvillea | Kelurahan Pasir | Diploma I/II/III/I | PNS | Lebih dari Rp 10.
Catherine | 3 April 1993 | Bandung | 30 | Jl. Angsana Trop | Kelurahan Pasir | SD | Ibu Rumah Tang | Kurang dari Rp 1
Tara | 11 Mei 1994 | Bandung | 29 | Jl. Cendana Mur | Kelurahan Pasir | SLTP | Ibu Rumah Tang | Kurang dari Rp 1
Kristen | 19 Juni 1999 | Bandung | 24 | Jl. Kenari Indah I | Kelurahan Pasir | SLTA | Buruh Harian Le | Kurang dari Rp 1

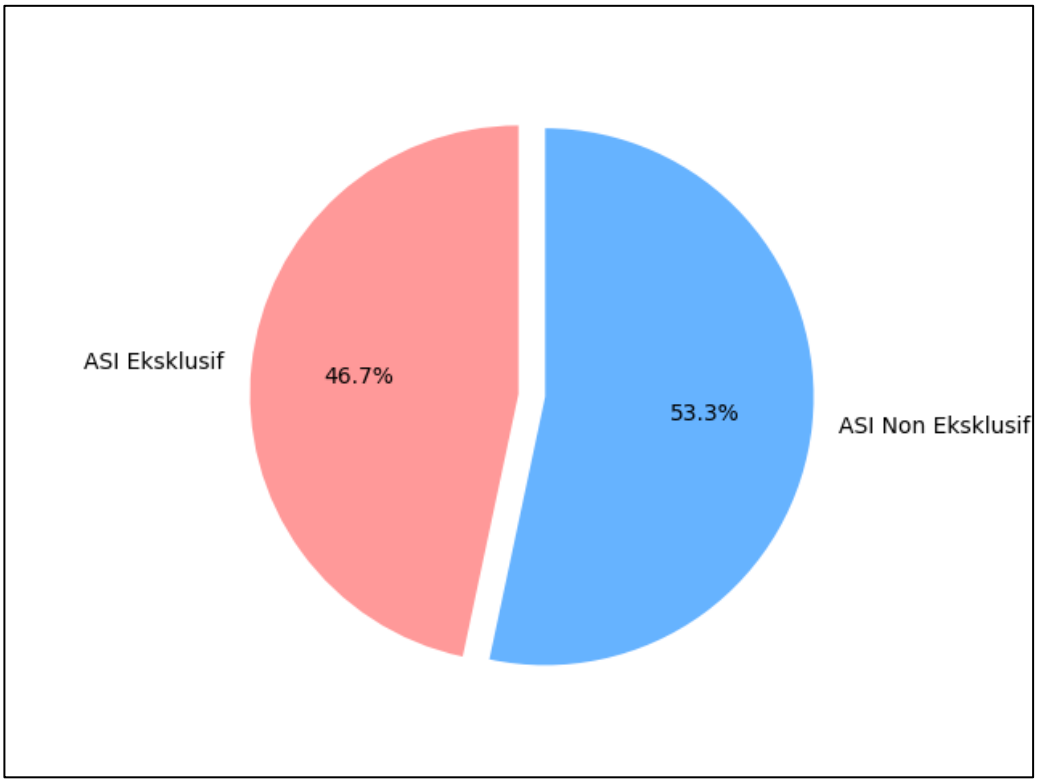
2.2.2. Tampilan Data Visualisasi



Kesimpulan : Keluarahan Pasir Impun memiliki rata-rata dibawah batas minimal berat badan menurut WHO yaitu 10-15Kg

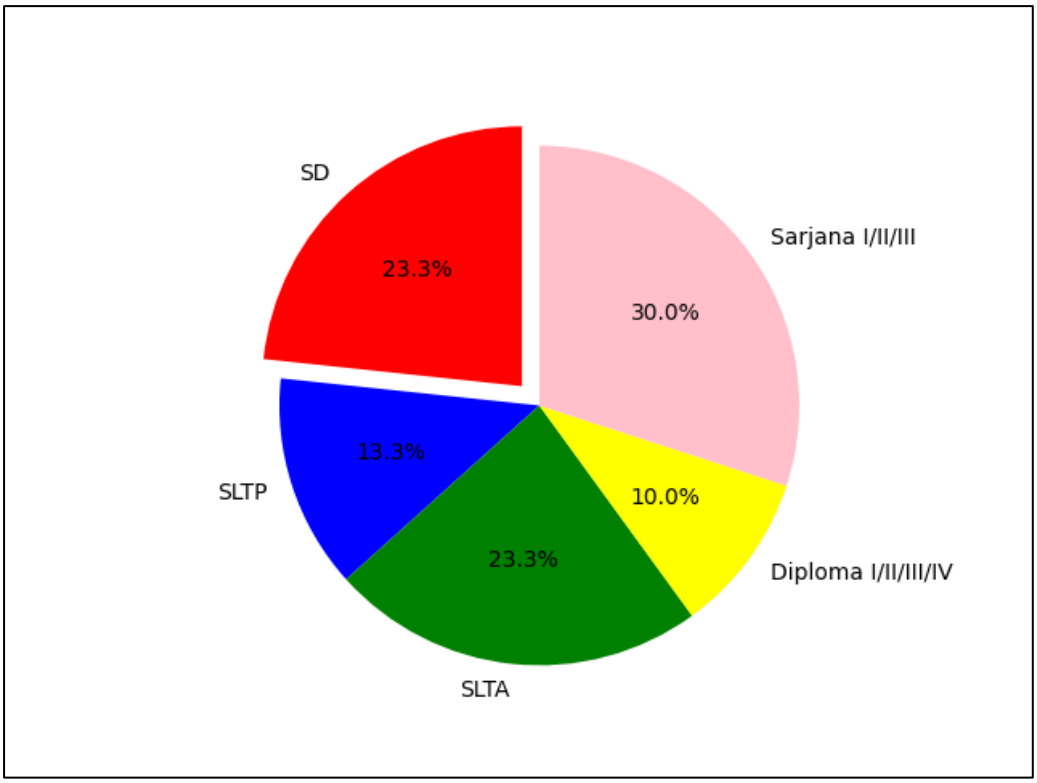
Kesimpulan : Keluarahan Sindangjaya dan Pasir Impun memiliki rata-rata dibawah tinggi dibawah batas minimal WHO yaitu 82–92 cm

Persentase Pemberian ASI Eksklusif



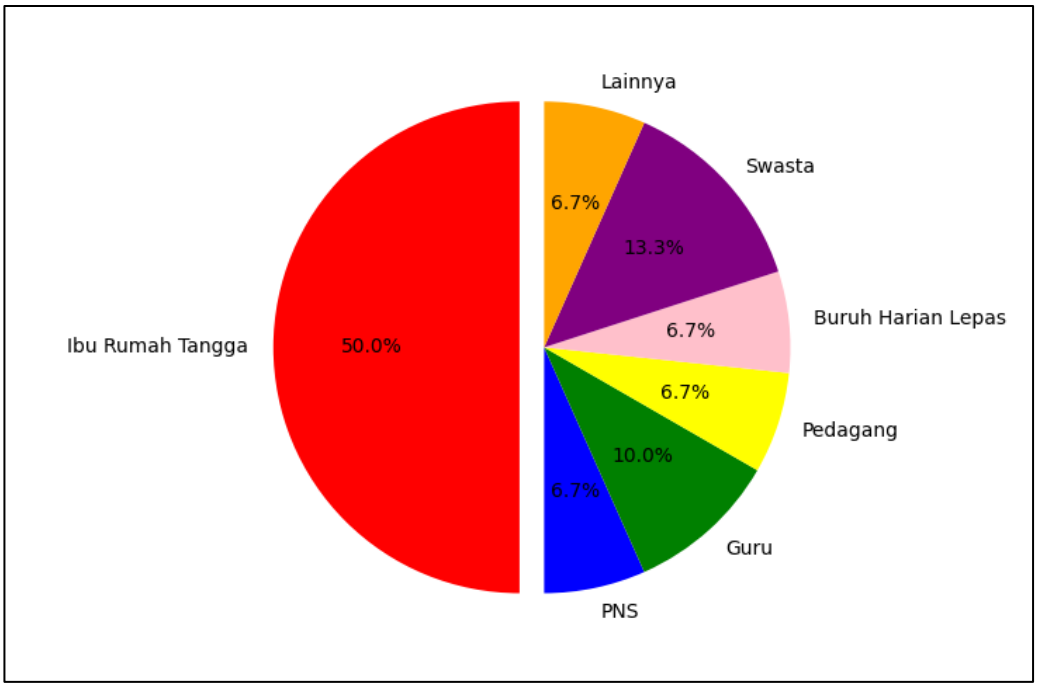
Kesimpulan : Hasil dari rata-rata berat badan dan tinggi badan yang menunjukan banyajnya anak yang berpotensi stunting, cukup bisa diterima karena ternyata banyak ibu yang tidak memberikan ASI eksklusifnya

Persentasi Pendidikan Ibu

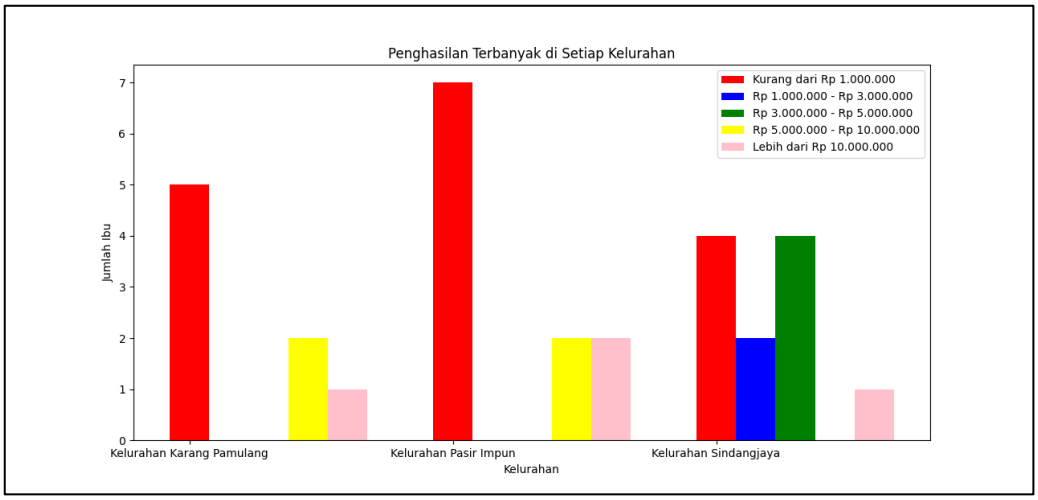


Kesimpulan : dari data diatas sekitar 30% ibu penempuh Pendidikan sampai sarjana I/II/II namun data ini masih sangat kurang tidak menjumlahkan data yang masih belum punya kesempatan melakukan proses Pendidikan

Persentase Pekerjaan Ibu



Kesimpulan : sebanyak 50% ibu memilih tidak bekerja dan menjadi ibu rumah tangga saja.



Kesimpulan : Tingkat kemiskinan masih sangat tinggi di berbagai kelurahan, ini juga bisa menjadi hipotesa mengapa tingkat anak yang terindikasi stunting cukup tinggi, karena ketidak mampuan membeli makanan-makanan yang bergizi