

**TUGAS PEMROGRAMAN VISUAL:  
TUGAS AKHIR**



**DISUSUN OLEH :**

Baiq Luthfida khairunnisa

(F1D022037)

**KELAS C**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
2025**

## PENDAHULUAN

### 1. Tentang Aplikasi

Aplikasi "Yuk Hitung Gizi Balita" dibuat untuk membantu memeriksa status gizi balita berdasarkan umur, tinggi badan, dan berat badan. Dengan memasukkan data tersebut, pengguna bisa langsung mengetahui apakah anak berada pada kategori gizi buruk, normal, atau gizi lebih.

Selain itu, aplikasi ini juga bisa menyimpan data anak ke database, menampilkan daftar data dalam bentuk tabel, mengekspor ke file CSV, dan menyajikan grafik distribusi status gizi dengan tampilan yang menarik dan user-friendly. Cocok digunakan oleh guru, tenaga kesehatan, atau orang tua yang ingin memantau kondisi gizi anak dengan mudah.

### 2. Tahapan Pembuatan Aplikasi

Dalam membuat aplikasi hitung gizi balita ini, berikut dijabarkan proses pembuatannya secara beruntun:

#### a. Menentukan Topik dan Tujuan Aplikasi

Pertama, topik ditentukan berdasarkan minat dalam bidang kesehatan anak. Tujuan aplikasinya adalah mempermudah orang tua, guru, atau petugas kesehatan dalam menghitung status gizi balita menggunakan data antropometri (umur, tinggi badan, berat badan).

#### b. Membuat Struktur Proyek

Struktur proyek dibuat menggunakan PyQt5 sebagai GUI (Graphical User Interface). File Python utama diberi nama, lalu dibuat tampilan jendela utama (QMainWindow) lengkap dengan tab-tab untuk input data dan grafik.

#### c. Membuat UI

Tampilan dirancang dengan gaya lucu dan warna pastel agar lebih menarik. Menggunakan komponen seperti:

- 1) QLineEdit untuk input teks,
- 2) QComboBox untuk pilihan jenis kelamin,
- 3) QTableWidget untuk menampilkan data,
- 4) QPushButton untuk tombol aksi.

Desain diperhalus dengan ditambahkan icon dan stylesheet agar tampilannya lebih estetik (misalnya warna pink pastel dan font lucu seperti Comic Sans MS atau Poppins).

d. Membuat fitur-fitur

Membuat fitur-fitur dalam aplikasi antara lain ada simpan, cari, hapus data, kemudian fungsi utamanya adalah untuk mengecek gizi balita dan ekspor CSV, fitur lain di tambahkan untuk pencarian nama anak serta tambahan untuk autofill dibagian menubar.

e. Menghubungkan ke Database SQLite

Dibuat koneksi ke file SQLite untuk menyimpan data anak, termasuk nama, jenis kelamin, umur, tinggi badan, berat badan, dan status gizi.

f. Logika Perhitungan Gizi

Logika status gizi ditentukan secara manual menggunakan kondisi if berdasarkan rentang umur, tinggi, dan berat badan. Hasilnya dikategorikan ke dalam:

- 1) Gizi Buruk
- 2) Gizi Normal
- 3) Gizi Lebih

g. Menambahkan Visualisasi Grafik

Grafik Lingkaran dibuat menggunakan Matplotlib, yang menampilkan jumlah anak berdasarkan status gizinya. Grafik ini diletakkan di tab kedua sebagai ringkasan visual yang mudah dibaca.

## PEMBAHASAN

### 3. Penjelasan Fungsi

#### a. save\_data()

```
def save_data(self):  
    nama = self.name_input.text()  
    jenis_kelamin = self.gender_input.currentText()  
    umur = self.umur_input.text()  
    tb = self.tb_input.text()  
    bb = self.bb_input.text()  
    status = self.cek_gizi_manual(umur, tb, bb)  
    if not nama or not jenis_kelamin or not umur or  
not tb or not bb:  
        QMessageBox.warning(self, "Input Kosong",  
"Semua kolom harus diisi sebelum menyimpan.")  
        return  
    cur = self.conn.cursor()  
    cur.execute("INSERT INTO gizi (nama, kelamin,  
umur, tb, bb, status_gizi) VALUES (?, ?, ?, ?, ?, ?)", (nama,  
jenis_kelamin, umur, tb, bb, status))  
    self.conn.commit()  
    self.name_input.clear()  
    self.gender_input.setCurrentIndex(0)  
    self.umur_input.clear()  
    self.tb_input.clear()  
    self.bb_input.clear()  
    self.load_data()
```

Fungsi ini bertugas untuk menyimpan data anak ke dalam database SQLite setelah pengguna mengisi form. Di dalamnya juga terdapat pemanggilan fungsi `cek_gizi_manual()` untuk menghitung status gizi berdasarkan umur, tinggi badan, dan berat badan. Setelah data tersimpan, semua input akan dikosongkan dan tabel akan diperbarui.

b. cek\_gizi\_manual(umur, tb, bb)

```
def cek_gizi_manual(self, umur, tb, bb):
    try:
        umur = int(umur)
        tb = float(tb)
        bb = float(bb)

        # Contoh aturan gizi berdasarkan rentang nilai
manual
    if umur <= 12:
        if tb < 65 or bb < 6:
            return "Gizi Buruk"
        elif 65 <= tb <= 75 and 6 <= bb <= 9:
            return "Gizi Normal"
        else:
            return "Gizi Lebih"

    elif 13 <= umur <= 24:
        if tb < 75 or bb < 8:
            return "Gizi Buruk"
        elif 75 <= tb <= 85 and 8 <= bb <= 11:
            return "Gizi Normal"
        else:
            return "Gizi Lebih"

    elif 25 <= umur <= 36:
        if tb < 85 or bb < 10:
            return "Gizi Buruk"
        elif 85 <= tb <= 95 and 10 <= bb <= 13:
            return "Gizi Normal"
        else:
            return "Gizi Lebih"

    elif 37 <= umur <= 60:
        if tb < 95 or bb < 12:
            return "Gizi Buruk"
        elif 95 <= tb <= 110 and 12 <= bb <= 17:
```

```

        return "Gizi Normal"
    else:
        return "Gizi Lebih"

    else:
        return "Data luar rentang balita (0-60
bulan)"

    except:
        return "Input tidak valid"

```

Ini adalah logika inti dari aplikasi. Fungsi ini mengecek status gizi anak menggunakan aturan sederhana berbasis range (rentang) dari umur, tinggi badan, dan berat badan.

Outputnya bisa berupa: Gizi Buruk, Gizi Normal, atau Gizi Lebih.

c. `load_data()`

```

def load_data(self, text=None):
    # Pastikan text adalah string, kalau None jadi
    string kosong
    if text is None:
        text = self.search_input.text()
        search = text.strip() # bersihkan spasi

    query = "SELECT * FROM gizi WHERE nama LIKE ?" if
    search else "SELECT * FROM gizi"
    cur = self.conn.cursor()
    if search:
        cur.execute(query, ('%' + search + '%',))
    else:
        cur.execute(query)

    records = cur.fetchall()
    self.table.blockSignals(True)
    self.table.setRowCount(0)
    for row_data in records:
        row_number = self.table.rowCount()
        self.table.insertRow(row_number)
        for column_number, data in enumerate(row_data):

```

```

        item = QTableWidgetItem(str(data))
        item.setTextAlignment(Qt.AlignCenter)
        self.table.setItem(row_number,
column_number, item)
        self.table.blockSignals(False)

```

Fungsi ini menampilkan semua data anak yang sudah disimpan dalam bentuk tabel. Kalau pengguna mengetikkan sesuatu di kolom pencarian, fungsi ini akan menampilkan hasil pencarian yang cocok.

d. `delete_data()`

```

def delete_data(self):
    selected = self.table.currentRow()
    if selected < 0:
        QMessageBox.warning(self, "Tidak Ada Data",
"Pilih satu buku terlebih dahulu untuk dihapus.")
        return

    judul = self.table.item(selected, 1).text()
    reply = QMessageBox.question(
        self, "Konfirmasi Hapus", f"Apakah yakin ingin
menghapus buku berjudul '{judul}'?",
        QMessageBox.Yes | QMessageBox.No,
        QMessageBox.No
    )
    if reply == QMessageBox.Yes:
        id_item = self.table.item(selected, 0)
        cur = self.conn.cursor()
        cur.execute("DELETE FROM gizi WHERE id = ?",
(id_item.text(),))
        self.conn.commit()
        self.load_data()

```

Fungsi ini digunakan untuk menghapus data anak yang dipilih dari tabel. Sebelum menghapus, aplikasi akan menampilkan pop-up konfirmasi dulu supaya tidak salah hapus.

e. `export_to_csv()`

```
def export_to_csv(self):
    path, _ = QFileDialog.getSaveFileName(self,
    "Simpan File", "", "CSV Files (*.csv)")
    if path:
        cur = self.conn.cursor()
        cur.execute("SELECT * FROM gizi")
        records = cur.fetchall()
        with open(path, 'w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(["ID", "Nama", "Kelamin",
            "Umur", "Tinggi Badan", "Berat Badan", "Status Gizi"])
            writer.writerows(records)
```

Fungsi ini digunakan untuk menyimpan semua data ke dalam file CSV.

Fitur ini berguna jika pengguna ingin mencetak atau menyimpan data di luar aplikasi.

f. `plot_gizi_data()`

```
def plot_gizi_data(self):
    # Ambil data dari database
    cur = self.conn.cursor()
    cur.execute("SELECT status_gizi, COUNT(*)
    FROM gizi GROUP BY status_gizi")
    data = cur.fetchall()

    # Pisahkan label dan jumlah
    labels = [row[0] for row in data]
    counts = [row[1] for row in data]

    # Clear axis & plot
    self.ax.clear()

    self.ax.bar(labels, counts,
    color="#FF69B4") # pink fuchsia
    self.ax.set_title("Distribusi Status Gizi
    Anak")

    self.ax.set_ylabel("Jumlah Anak")
    self.ax.set_xlabel("Status Gizi")
    self.canvas.draw()
```



Fungsi ini membuat grafik batang (bar chart) untuk menunjukkan jumlah anak berdasarkan status gizinya. Visualisasi ini membantu pengguna melihat distribusi gizi anak dengan lebih jelas dan menarik.

g. auto\_fill()

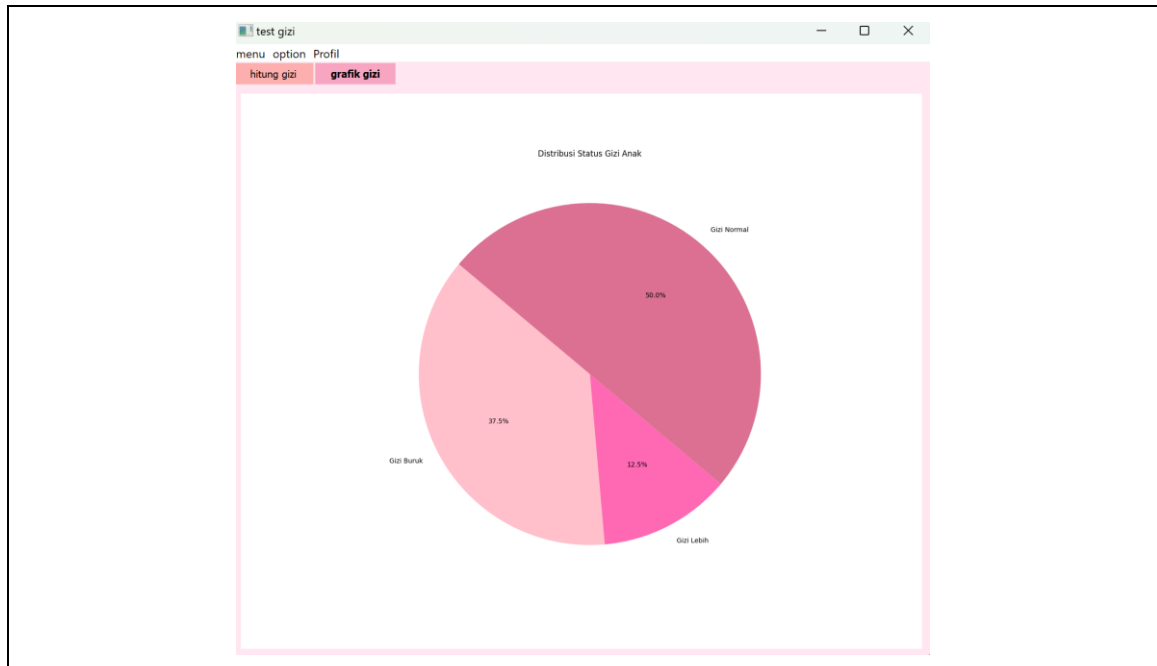
```
def auto_fill(self):  
    self.name_input.setText("fulan")  
    self.gender_input.setCurrentIndex(0)  
    self.umur_input.setText("41")  
    self.tb_input.setText("100")  
    self.bb_input.setText("15")
```

Fungsi ini membantu pengguna mengisi data form secara otomatis dengan data contoh yang sudah di uji cobakan sebelumnya.

#### 4. Screen Shots UI

ID	Nama	Kelamin	umur	TB	BB	Status	
1	2	fulan	Laki-laki	41	100	15	Gizi Normal
2	3	fulan	Laki-laki	41	100	15	Gizi Normal
3	4	putu	Laki-laki	1	65	6,4	Gizi Normal
4	5	cinta	Perempuan	24	76	10	Gizi Normal
5	6	putri	Perempuan	48	89	10	Gizi Buruk
6	7	harip	Perempuan	3	55	5	Gizi Buruk
7	8	ipin	Laki-laki	18	77	7,2	Gizi Buruk
8	9	emir	Laki-laki	5	100	8	Gizi Lebih

Pada gambar diatas dapat dilihat hasil tampilan aplikasi yang sudah berjalan dan dibuat interaktif bagi pengguna, dibuat tombol yang jelas dengan warna yang mencolok agar memudahkan pengguna dalam mencari tombol yang akan ditekan, serta penggunaan combo box untuk mempermudah inputan pengguna.



Pada Gambar diatas merupakan tampilan grafik dari perolehan gizi yang didapatkan, fitur ini memudahkan, guru dan bahkan tenaga Kesehatan dalam menghitung persentase di sebuah tempat, manakan status gizi yang mendominasi. Sehingga dapat di wujudkakan Solusi terbaik dalam penanganan gizi.