

**LAPORAN PERHITUNGAN PROGRAM ESTIMASI BERAT JANIN
MATEMATIKA KOMPUTER 3**



DOSEN PENGAMPU

M. ZAINAL ARIFIN, S.SI., M.KOM.

DISUSUN OLEH

ALDY RAHMAT YULIANTO	(220535610051)
ALVALEN SHAFELBILYUNAZRA	(220535608548)
ARDHA ARDHANA PUTRA A.	(220535608503)
AVAN FABIAN DANISWARA	(220535603620)
AZARYA A. K. MOELJONO	(220535608951)
BURHANUDIN YUSUF A	(220535605001)
BIMA AHMADANI D H	(220535603999)
DAVID SATRIA ALAMSYAH	(220535610311)
DIMAS ARDIMINDA E P	(220535608555)
DIMAS ADI PRATAMA	(200535626858)
DIMAS MAULANA RAMADHAN	(220535605869)
ENANTO HARUN SATRIO	(220535600376)

OFFERING

TI 22 A

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI MALANG
2023**

A. PENDAHULUAN

Pengukuran berat janin adalah salah satu aspek kunci dalam pemantauan kehamilan yang bertujuan untuk memastikan pertumbuhan janin yang sehat dan perkembangan yang tepat. Selama beberapa dekade terakhir, teknologi USG (Ultrasonografi) telah menjadi alat penting dalam pengukuran berat janin secara non-invasif dan akurat. Penelitian ini bertujuan untuk menginvestigasi metode pengukuran berat janin menggunakan USG, serta implikasinya dalam perawatan kehamilan dan pemantauan pertumbuhan janin.

Keberhasilan perawatan kehamilan dan kelahiran bayi yang sehat sangat bergantung pada pemahaman yang mendalam tentang pertumbuhan janin. Salah satu indikator vital dalam memantau pertumbuhan janin adalah berat janin. Berat janin yang tidak sesuai dengan usia kehamilan dapat mengindikasikan masalah potensial seperti pertumbuhan terhambat atau makrosomia. Oleh karena itu, penting untuk memiliki metode pengukuran yang akurat dan non-invasif, yang dapat memberikan informasi yang tepat tentang berat janin.

Laporan ini bertujuan untuk membuat sebuah program perhitungan estimasi berat janin. Program ini didasarkan pada Penerapan perhitungan matematis Integral. Program ditujukan untuk mempermudah perhitungan estimasi berat bayi berdasarkan sebuah gambar USG. Dari gambar tersebut akan diketahui ukuran-ukuran dari bagian-bagian janin tersebut yang kemudian akan diaplikasikan pada formula integral yang telah dibuat.

Dalam Program ini user akan diminta untuk menginputkan gambar USG dari janin dan memilih titik-titik per bagian badan janin dari kaki bawah, kaki atas, lengan, badan, dan juga kepala. Berdasarkan data inputan titik-titik tersebut, akan dihitung menggunakan rumus integral volume sehingga akan ditemukan volume dari masing-masing bagian yang kemudian dijumlahkan dan dikonversikan ke berat dalam kilogram (kg).

Program ini dikembangkan menggunakan Bahasa pemrograman Python. Python dipilih sebagai bahasa pemrograman untuk pengembangan program estimasi berat janin karena kemudahan pengembangan, keterbacaan kode, dan dukungan pustaka matematis seperti NumPy dan SciPy yang penting untuk perhitungan integral.

B. FORMULA

Pada penghitungan berat janin ini kami menggunakan integral volume benda putar menggunakan persamaan garis lurus dan persamaan lingkaran. Untuk bagian tubuh mulai dari badan hingga kaki kami menggunakan rumus integral volume benda putar dengan persamaan garis lurus, sementara untuk bagian kepala menggunakan integral volume bola dengan persamaan lingkaran.

1. Formula Persamaan Garis Lurus Yang Melalui Dua Titik

Formula persamaan garis lurus adalah rumus matematika yang digunakan untuk menggambarkan dan mengidentifikasi garis lurus dalam ruang dua dimensi. Alasan kami menggunakan persamaan garis lurus untuk menghitung sumbu y dalam perhitungan integral volume benda putar karena persamaan garis lurus hanya memerlukan empat titik koordinat ketika digambar. Hal ini menjadikan sebagai pilihan yang sangat nyaman bagi pengguna yang ingin menentukan titik pada gambar yang akan dihitung.

$$\text{Rumus: } \frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1}$$

2. Formula Persamaan Lingkaran

Persamaan lingkaran adalah bentuk matematis yang digunakan untuk menggambarkan lokasi semua titik pada lingkaran dalam sistem koordinat dua dimensi. Formula Persamaan Lingkaran digunakan dalam perhitungan volume benda putar untuk menganalisis estimasi berat kepala janin pada gambar 2D. Pusat lingkaran yang digunakan dalam persamaan ini berada di tengah gambar penampang kepala janin, sementara jari-jarinya diukur dari pusat ke tepi penampang.

$$\text{Rumus: } x^2 + y^2 = r^2$$

$$y = \sqrt{r^2 - x^2}$$

3. Formula Integral Volume Benda Putar

Formula integral volume benda putar adalah alat matematis yang sangat berguna ketika kita memiliki data dalam bentuk gambar 2D yang ingin diubah menjadi volume tiga dimensi. Dalam kasus perhitungan berat janin, gambar penampang melintang janin yang didapatkan melalui teknik seperti ultrasonografi (USG) atau pencitraan medis seringkali hanya menampilkan informasi dalam dua dimensi, yaitu panjang dan lebar. Untuk mendapatkan informasi volume janin, kita perlu memperhitungkan dimensi ketiga, yaitu kedalaman.

Integral volume benda putar dapat mengonversi informasi dari gambar 2D ke dalam representasi tiga dimensi yang dapat digunakan untuk menghitung volume janin. Dengan diberikannya gambar 2D, kita dapat menggunakan sistem koordinat y sebagai sumbu untuk menghitung volume janin dengan tingkat akurasi yang baik, karena gambar penampang melintang janin sering kali ditampilkan dalam dimensi x dan y .

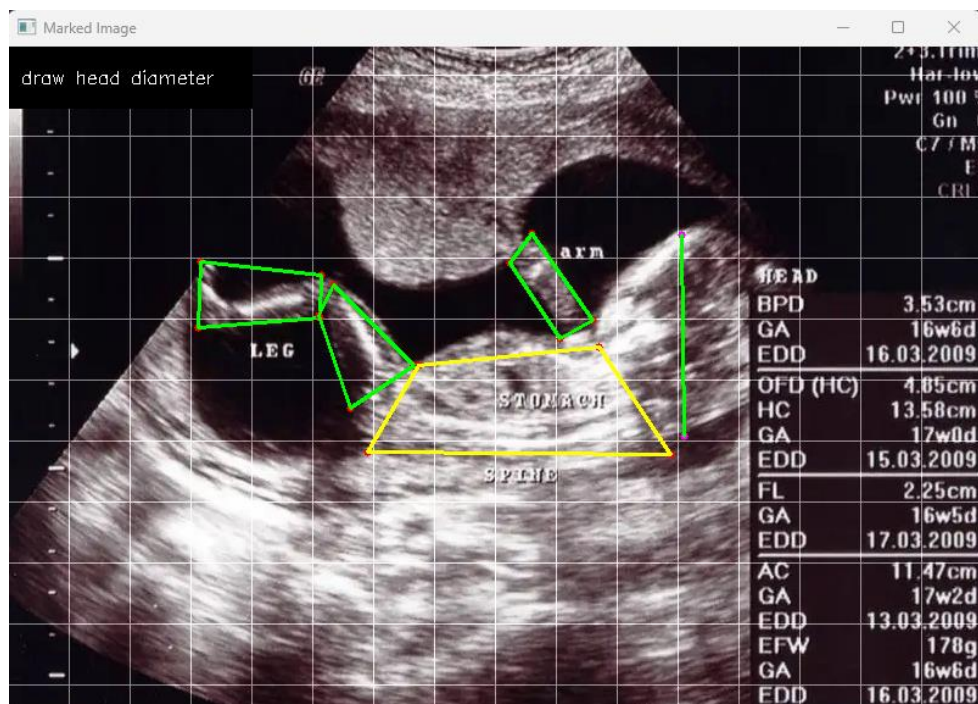
$$\text{Rumus : } V = \pi \int_a^b [f(x)]^2 dx$$

4. Formula Integral Volume Bola Dengan Persamaan lingkaran

Formula integral volume bola adalah rumus matematika yang digunakan untuk menghitung volume dari sebuah bola, yaitu benda tiga dimensi yang berbentuk bulat dengan semua titik pada permukaannya berjarak sama dari pusat bola. Alasan kami memilih menggunakan metode persamaan lingkaran dalam perhitungan volume bola, khususnya untuk mengestimasi volume kepala janin, karena bentuk kepala janin seringkali lebih mendekati bentuk bola atau bulat. Persamaan lingkaran dapat menggambarkan dengan baik struktur bulat dalam gambar kepala janin. Selain itu gambar kepala janin yang digunakan untuk perhitungan berat janin disajikan dalam dua dimensi (2D), sehingga menghitung volume dengan persamaan lingkaran menjadi lebih tepat dan praktis.

Dengan memandang persamaan lingkaran pada sumbu- x dan sumbu- y positif maka lingkaran yang terbentuk adalah seperempat lingkaran atau jika diputar terhadap sumbu- x maka akan terbentuk setengah bola. Sehingga untuk mencari volumenya yaitu dengan cara mengintegralkan persamaan lingkaran dengan batas atas dan batas bawah masing-masing 0 dan r dan dikalikan 2.

$$\text{Rumus : } V_{\text{bola}} = 2 \pi \int_0^r [\sqrt{r^2 - x^2}]^2 dx$$



Gambar 1. Proses perhitungan berat janin melalui inputan titik koordinat

Contoh Perhitungan

- Bottom_Foot

$$X1 = 0$$

$$Y1 = 1,6561461024015967$$

$$X2 = 2,686128688279447$$

$$Y2 = 1,0278$$

$$\frac{y - y1}{y2 - y1} = \frac{x - x1}{x2 - x1}$$

$$\frac{y - 1,6561461024015967}{1,0278 - 1,6561461024015967} = \frac{x - 0}{2,686128688279447 - 0}$$

$$2,686128688279447y - 4,44862156 = -0,628346102x$$

$$y = -0,233922561x + 1,6561461$$

$$V = \pi \frac{1}{2} \int_a^b [f(x)]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{2,686128688279447} [-0,233922561x + 1,6561461]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{2,686128688279447} (0,0547197645447987x^2 - 0,7748198742043242x + 2,74281990454521) dx$$

$$V = \pi \frac{1}{2} \int_0^{2,686128688279447} \left(\frac{1}{3} 0,0547197645447987x^3 - \frac{1}{2} 0,7748198742043242x^2 + 2,74281990454521x \right) dx$$

$$V = \pi \frac{1}{2} \int_0^{2,686128688279447} (0,0182399215149329x^3 - 0,3874099371021621x^2 + 2,74281990454521x) dx$$

$$V = \pi \frac{1}{2} [0,0182399215149329x^3 - 0,3874099371021621x^2 + 2,74281990454521x]_0^{2,686128688279447}$$

$$\begin{aligned} V &= \pi \frac{1}{2} [0,0182399215149329(2,686128688279447)^3 \\ &- 0,3874099371021621(2,686128688279447)^2 \\ &+ 2,74281990454521(2,686128688279447)] - 0 \end{aligned}$$

$$V = \pi \frac{1}{2} (0,353511389779044 - 2,795274010688499 + 7,367567232382783) - 0$$

$$V = \pi \frac{1}{2} (0,353511389779044 - 2,795274010688499 + 7,367567232382783) - 0$$

$$V = \pi \frac{1}{2} (4,925804611473328)$$

$$V = \frac{1}{2} 15,46702648002625$$

$$V = 7,733513240013125 \text{ cm}^3$$

• **Upper Foot**

$$X1 = 0$$

$$Y1 = 1.0528719770228476$$

$$X2 = 1.8184803662673952$$

$$Y2 = 1.4848744938546152$$

$$\frac{y - y1}{y2 - y1} = \frac{x - x1}{x2 - x1}$$

$$\frac{y - 1.0528719770228476}{1.4848744938546152 - 1.0528719770228476} = \frac{x - 0}{1.8184803662673952 - 0}$$

$$1.8184803662673952y - 1.91462702 = 0.4320025168317x$$

$$y = \frac{0.4320025168317x + 1.91462702}{1.8184803662673952}$$

$$y = 0.2375623761714989x + 1.0528719778976525$$

$$V = \pi \frac{1}{2} \int_a^b [f(x)]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{1.8184803662673952} [0.2375623761714989x + 1.0528719778976525]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{1.8184803662673952} (0.05643588257224874x^2 + 0.5002455377475044x + 1.1085394018421149) dx$$

$$V = \pi \frac{1}{2} \int_0^{1.8184803662673952} \left(\frac{1}{3} 0.05643588257224874x^3 + \frac{1}{2} 0.5002455377475044x^2 + 1.1085394018421149x \right) dx$$

$$V = \pi \frac{1}{2} \int_0^{1.8184803662673952} (0.018811960857416248x^3 + 0.2501227688737522x^2 + 1.1085394018421149x) dx$$

$$V = \pi \frac{1}{2} [0.018811960857416248x^3 + 0.2501227688737522x^2 + 1.1085394018421149x]_0^{1.8184803662673952}$$

$$\begin{aligned} V &= \pi \frac{1}{2} [0.018811960857416248(1.8184803662673952)^3 \\ &+ 0.2501227688737522(3.3068708424999995)^2 \\ &+ 1.1085394018421149(1.8184803662673952)]_0^{1.8184803662673952} \end{aligned}$$

$$V = \pi \frac{1}{2} (0.11312534474960292 + 0.8271236914339776 + 2.015857137483688) - 0$$

$$V = \pi \frac{1}{2} (2.9561061736672682)$$

$$V = \frac{1}{2} (9.282173385315224)$$

$$V = 4.641086692657612 \text{ cm}^3$$

- **Hand / Arm**

$$X1 = 0$$

$$Y1 = 0.9391170880140558$$

$$X2 = 2.0332731936461466$$

$$Y2 = 0.8180459354461703$$

$$\frac{y - y1}{y2 - y1} = \frac{x - x1}{x2 - x1}$$

$$\frac{y - 0,9391170880140558}{0,8180459354461703 - 0,9391170880140558} = \frac{x - 0}{2,0332731936461466 - 0}$$

$$2.0332731936461466y - 1.90948 = -0,1210711525678855x$$

$$\frac{y = -0,1210711525678855x + 1.90948}{2,0332731936461466}$$

$$y = -0.059545x + 0.939116$$

$$V = \pi \frac{1}{2} \int_a^b [f(x)]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{2.0332731936461466} [-0.059545x + 0.939116]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{2.0332731936461466} (0.0035456x^2 - 0.11183932x + 0.88193886) dx$$

$$V = \pi \frac{1}{2} \int_0^{2.0332731936461466} \left(\frac{1}{3} 0.0035456x^3 - \frac{1}{2} 0.11183932x^2 + 0.88193886x \right) dx$$

$$V = \pi \frac{1}{2} \int_0^{2.0332731936461466} (0.00118187x^3 - 0.05591966x^2 + 0.88193886x) dx$$

$$V = \pi \frac{1}{2} [0.00118187x^3 - 0.05591966x^2 + 0.88193886x]_0^{2.0332731936461466}$$

$$V$$

$$= \pi \frac{1}{2} [0.00118187(2.0332731936461466)^3 - 0.05591966(2.0332731936461466)^2 + 0.88193886(2.0332731936461466)] \frac{2.0332731936461466}{0}$$

$$V = \pi \frac{1}{2} (0.00993472 - 0.231183 + 1.793222642472841775796876) - 0$$

$$V = \pi \frac{1}{2} (1.571974362472481775796876)$$

$$V = \frac{1}{2} 4,9385$$

$$V = 2,46925 \text{ cm}^3$$

- Body**

$$X1 = 0$$

$$Y1 = 2.4009175079539906$$

$$X2 = 7.1389273572785426$$

$$Y2 = 3.1762436371443545$$

$$\frac{y - y1}{y2 - y1} = \frac{x - x1}{x2 - x1}$$

$$\frac{y - 2.4009175079539906}{3.1762436371443545 - 2.4009175079539906} = \frac{x - 0}{7.1389273572785426 - 0}$$

$$\frac{y - 2.4009175079539906}{0.7753261291903639} = \frac{x - 0}{7.1389273572785426 - 0}$$

$$7.1389273572785426 y - 17.139975680101767 = 0.7753261291903639 x$$

$$y = \frac{0.7753261291903639x + 17.139975680101767}{7.1389273572785426}$$

$$y = 0.10860540952274501x + 2.4009175079539906$$

$$V = \pi \frac{1}{2} \int_a^b [f(x)]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{7.1389273572785426} [0.10860540952274501x + 2.4009175079539906]^2 dx$$

$$V = \pi \frac{1}{2} \int_0^{7.1389273572785426} (0.011795134977603153x^2 + 0.5215052583633432x + 5.7644048800000001) dx$$

$$V = \pi \frac{1}{2} \int_0^{7.1389273572785426} \left[\frac{1}{3} 0.011795134977603153(x)^3 + \frac{1}{2} 0.5215052583633432(x)^2 + 5.7644048800000001(x) \right] dx$$

$$V = \pi \frac{1}{2} \int_0^{7.1389273572785426} \left[0.0039317116592010514(x)^3 + 0.2607526291816716(x)^2 + 5.7644048800000001(x) \right] dx$$

$$V = \pi \frac{1}{2} \left[(0.0039317116592010514(x)^3 + 0.2607526291816716(x)^2 + 5.7644048800000001(x)) \right]_0^{7.1389273572785426}$$

$$V = \pi \frac{1}{2} \left[0.0039317116592010514(7.1389273572785426)^3 + 0.2607526291816716(7.1389273572785426)^2 + 5.7644048800000001(7.1389273572785426) - 0 \right]$$

$$V = \pi \frac{1}{2} (1.4304759109307046 + 13.289070998470278 + 41.15166769626194)$$

$$V = \pi \frac{1}{2} (55.87121460566292)$$

$$V = \frac{1}{2} (175.43561386178158)$$

$$V = 87.71780693089079 \text{ cm}^3$$

- **Head**

$$\begin{aligned}
 V &= 2\pi \int_0^r y^2 dx \\
 &= 2\pi \int_0^r (\sqrt{r^2 - x^2})^2 dx \\
 &= 2\pi \int_0^r (r^2 - x^2) dx \\
 &= 2\pi \int_0^r (r^2 - (r \sin \theta)^2) dx \\
 &= 2\pi \int_0^r (r^2 - r^2 \sin^2 \theta) dx \\
 &= 2\pi \int_0^r r^2 (1 - \sin^2 \theta) dx \\
 &= 2\pi \int_0^r r^2 \cos^2 \theta dx
 \end{aligned}$$

Karena $\sin \theta = \frac{x}{r}$, maka $x = r \sin \theta$, kemudian diturunkan kedua ruas menghasilkan $dx = r \cos \theta d\theta$, substitusi dx sehingga menghasilkan:

$$\begin{aligned}
 &= 2\pi \int_0^r r^2 \cos^2 \theta (\cos \theta d\theta) \\
 &= 2r^3 \pi \int_0^r \cos^2 \theta \cos \theta d\theta \\
 &= 2r^3 \pi \int_0^r (1 - \sin^2 \theta) (\cos \theta d\theta)
 \end{aligned}$$

Dilanjutkan dengan integral substitusi

Misalnya : $u = \sin \theta$ maka $du = \cos \theta d\theta$ sehingga diperoleh

$$\begin{aligned}
 &= 2r^3 \pi \int_0^r (1 - u^2) du \\
 &= 2r^3 \pi \left(\sin \theta - \frac{1}{3} u^3 \right) \Big|_0^r
 \end{aligned}$$

Subtitusikan $u = \sin \theta$, diperoleh

$$= 2r^3 \pi \left(\sin \theta - \frac{1}{3} \sin^3 \theta \right) \Big|_0^r$$

Karena $\sin \theta = \frac{x}{r}$, maka

$$\begin{aligned}
&= 2r^3\pi \left(\frac{x}{r} - \frac{1}{3} \left(\frac{x}{r} \right)^3 \right) \Big|_0^r \\
&= 2r^3\pi \left[\left(\frac{r}{r} - \frac{1}{3} \left(\frac{r}{r} \right)^3 \right) - \left(\frac{0}{r} - \frac{1}{3} \left(\frac{0}{r} \right)^3 \right) \right] \\
&= 2r^3\pi \left[\left(1 - \frac{1}{3} \right) - (0 - 0) \right] \\
&= 2r^3\pi \left(\frac{2}{3} \right) \\
&= \frac{4}{3} r^3 \pi
\end{aligned}$$

Diameter Kepala : 4.682548157787595

Jari-jari Kepala : 2.341274078893798

$$\begin{aligned}
\text{Volume} &= \frac{4}{3} 2.341274078893798^3 \pi \\
&= \frac{4}{3} 12.83384443664556 \pi \\
&= 17.11179258219407\pi \\
&= 53.7582818659732 \text{ cm}^3
\end{aligned}$$

Volume Total :

$$\begin{aligned}
&= 2 \times (\text{volume bottom foot} + \text{volume upper foot}) + (2 \times \text{volume hand}) + \text{volume} \\
&\quad \text{body} + \text{volume head} \\
&= 2 \times (7,733513240013125 \text{ cm}^3 + 4,641086692657612 \text{ cm}^3) + (2 \times 2,46325 \text{ cm}^3) + \\
&\quad 87,71780693089079 \text{ cm}^3 + 53,7582818659732 \text{ cm}^3 \\
&= 24,749199865341474 + 4,9265 + 87,71780693089079 + 53,7582818659732 \\
&= 171,151788662205464 \text{ cm}^3
\end{aligned}$$

Konversi Volume ke Berat :

Berat = Volume Total x massa jenis

$$\text{Berat} = 171,151788662205464 \text{ cm}^3 \times 1 \text{ g/cm}^3$$

$$\text{Berat} = 171,151788662205464 \text{ g} = 0,171151788662205464 \text{ kg}$$

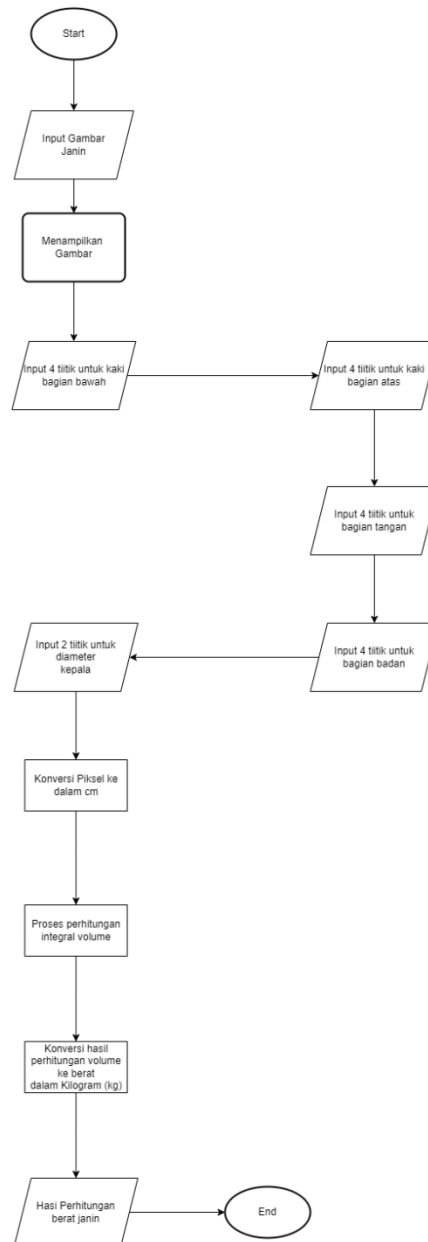
C. ALGORITMA

Langkah pertama pada program ini adalah user meng-input gambar USG dari file manager. Program kemudian akan menampilkan gambar USG kepada user dengan *grid* untuk mempermudah proses penandaan titik koordinat.

Setelah gambar tampil. User akan diminta untuk memasukkan titik-titik koordinat pada gambar untuk masing-masing bagian badan dari janin. Terdapat total 5 bagian badan janin, yaitu kaki bawah, kaki atas, lengan, perut, dan kepala. Untuk titik kaki bagian bawah akan diminta 4 titik inputan, 4 titik untuk kaki bagian atas, 4 titik untuk bagian tangan, 4 titik untuk bagian badan, dan 2 titik untuk diameter kepala. Titik-titik tersebut dimulai dari x_1 , x_2 , y_2 , dan terakhir y_1 untuk kepala sendiri hanya 2 titik karena yang dicari adalah diameter kepala.

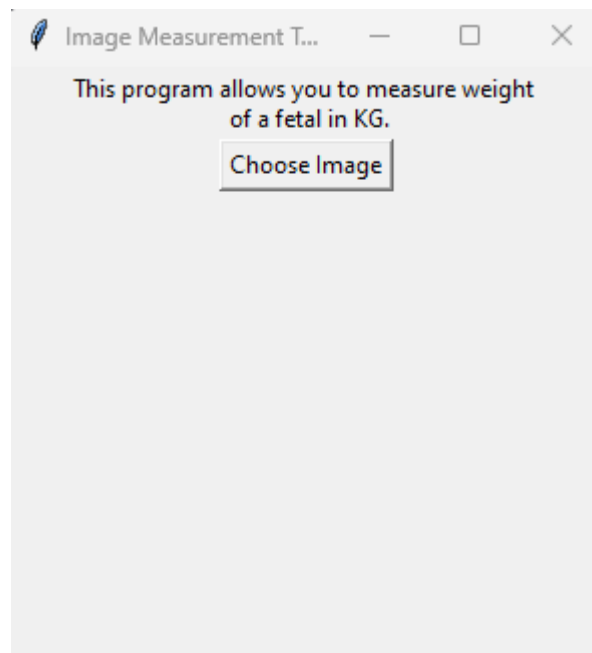
Jarak antar masing-masing titik tersebut akan dihitung dalam pixel kemudian dikonversi menjadi centimeter (cm) dengan cara mengkalikan dengan skala gambar yang harus dikalibrasi manual. Kemudian, setelah user menginputkan semua titik dari masing-masing bagian tubuh, program akan menghitung volume setiap bagian yang diukur menggunakan formula sebelumnya, yaitu untuk bagian kaki, tangan dan perut akan digunakan formula **Persamaan Garis Lurus Yang Melalui Dua Titik** dan **Integral Volume Benda Putar**. Sedangkan untuk kepala digunakan formula **Integral Volume Bola**.

Selanjutnya hasil dari volume kaki bagian atas dan bawah akan dijumlahkan dan dikalikan dua. Untuk bagian tangan juga akan dikalikan dua. Lalu hasil perhitungan dari masing-masing volume bagian tubuh dijumlahkan dan disimpan sebagai variabel total yang selanjutnya dikonversi menjadi berat dalam kilogram (kg). Hasil dari konversi tersebut ditampilkan sebagai output.

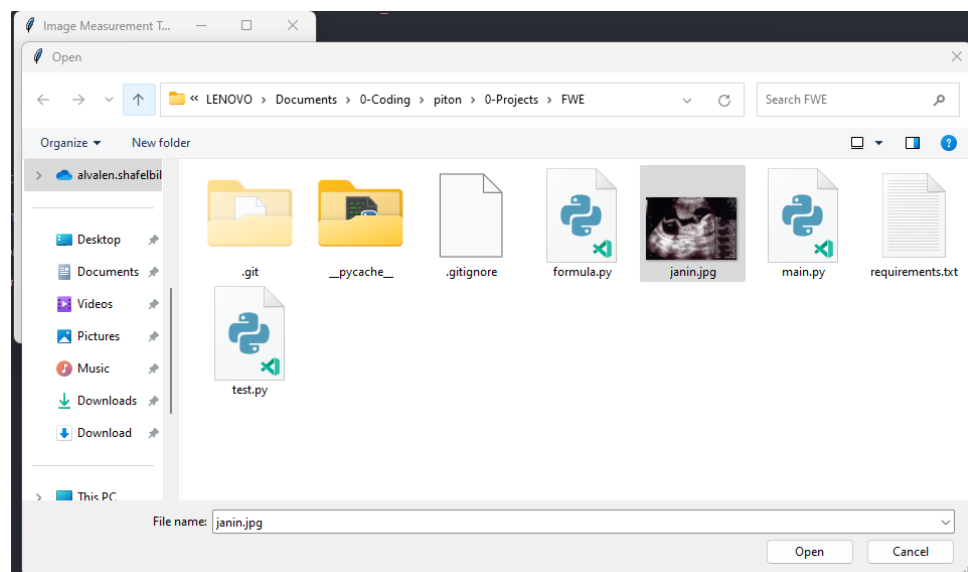


Gambar 2. *Flowchart*

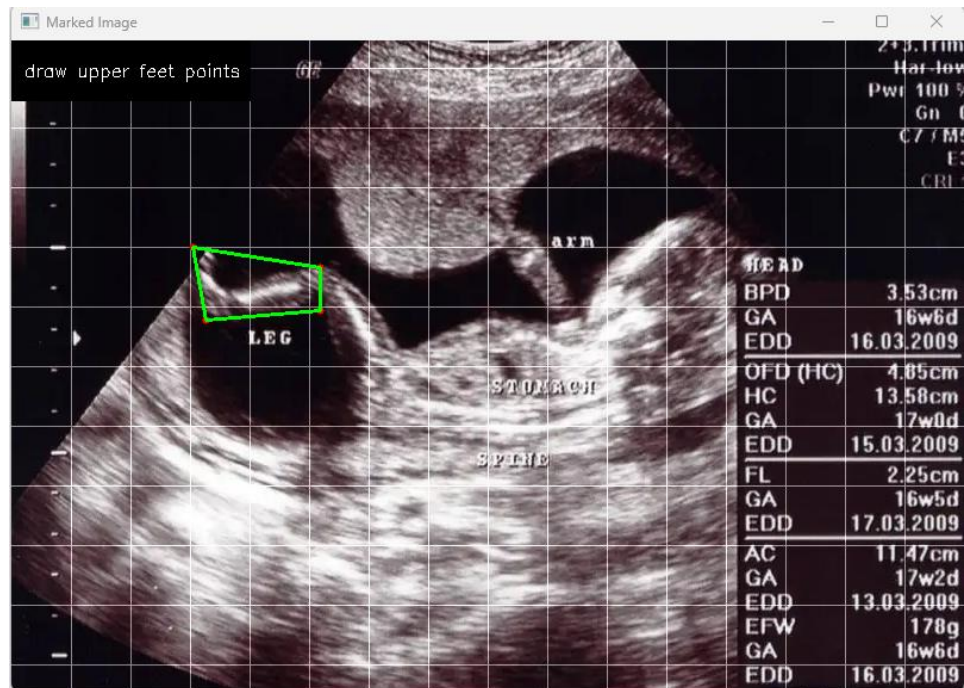
D. SCREENSHOT PROGRAM



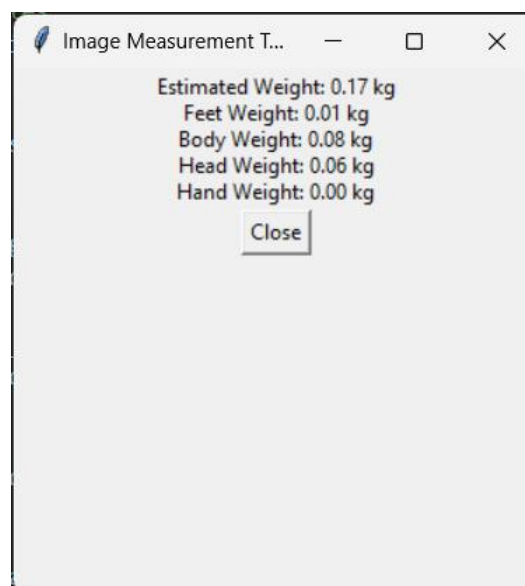
Gambar 3. Tombol untuk memilih gambar inputan



Gambar 4. Proses pemilihan gambar untuk digunakan sebagai input program



Gambar 5. Penginputan titik koordinat pada bagian tubuh janin



Gambar 6. Output hasil penghitungan berat janin

E. KODE PROGRAM

Dalam pengembangan program ini, Bahasa yang digunakan adalah python versi 3.11.x ke atas. Dengan menggunakan Text Editor visual studio code dan beberapa library python untuk membantu proses perhitungan maupun visualisasi. Beberapa library yang digunakan antara lain adalah scipy, numpy,

math untuk perhitungan, tkinter untuk UI dan opencv untuk visualisasi. Dalam kode ini file terpisah menjadi dua yaitu formula.py dan main.py.

- **Kode Formula (formula.py)**

```
from scipy import integrate
import numpy as np
import math

scale = 0.02855 # Adjust this value according to your scale

def calculate_integrate_volume(points):
    x1, x2, y1, y2 = points
    X = np.array([x1, x2])
    Y = np.array([y1, y2])
    coefficients = np.polyfit(X, Y, 1)
    slope = coefficients[0]
    intercept = coefficients[1]

    def volume_function(x):
        return (slope * x + intercept)**2

    volume, _ = integrate.quad(volume_function, x1, x2)
    volume = (volume * np.pi) * 0.5
    return volume

# Function to calculate distance between two points in
# physical units
def calculate_distance(point1, point2):
    # Calculate the Euclidean distance in physical units
    distance = float(math.sqrt((point2[0] - point1[0])**2 +
    (point2[1] - point1[1])**2) * scale)
    return distance

def calculate_head_volume(diameter):
    R = diameter / 2 # Inisialisasi nilai R dengan diameter
    / 2
    r_lower = 0
    r_upper = R

    def integrand(x, R):
        return np.pi * (R**2 - x**2)

    # Hitung integral menggunakan scipy.integrate.quad
    result, _ = integrate.quad(integrand, r_lower, r_upper,
    args=(R))
    volume_kepala = result * 2
    return volume_kepala
```

- **Kode Program Utama (main.py)**

```
import tkinter as tk
from tkinter import filedialog
import cv2
import numpy as np
```

```

from formula import calculate_integrate_volume,
calculate_distance, calculate_head_volume

# init tkinter
root = tk.Tk()
# Variables to store clicked points
bottom_feet_coordinates = []
upper_feet_coordinates = []
hand_coordinates = []
body_coordinates = []
head_diameter = []
status_text = 'Click on the four points of the bottom
feet'
image = None

# Function to draw lines and distance text between
points
def draw_lines_and_distances(image, points, color=(0,
255, 0)):
    for i in range(1, len(points)):
        cv2.line(image, points[i - 1], points[i],
color, 2)
        distance = calculate_distance(points[i - 1],
points[i])
        text_position = (
            int((points[i - 1][0] + points[i][0]) /
2),
            int((points[i - 1][1] + points[i][1]) / 2)
        )
        # cv2.putText(image, f'{distance:.2f} cm',
text_position, cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0,
255), 2)

# Function to handle mouse clicks
def mark_point(event, x, y, flags, param):
    global bottom_feet_coordinates,
upper_feet_coordinates, body_coordinates,
head_diameter, marked_image, hand_coordinates
    if event == cv2.EVENT_LBUTTONDOWN:
        if len(bottom_feet_coordinates) < 4:
            bottom_feet_coordinates.append((x, y))
            cv2.circle(marked_image, (x, y), 3, (0, 0,
255), -1)

            if len(bottom_feet_coordinates) == 4:
                draw_lines_and_distances(marked_image,
bottom_feet_coordinates)

```

```

        draw_lines_and_distances(marked_image,
[bottom_feet_coordinates[0],
bottom_feet_coordinates[3]])
        draw_instruction(marked_image, "draw
upper feet points")
        elif len(upper_feet_coordinates) < 4:
            upper_feet_coordinates.append((x, y))
            # Draw a dot at the clicked point
            cv2.circle(marked_image, (x, y), 3, (0, 0,
255), -1)

            if len(upper_feet_coordinates) == 4:
                draw_lines_and_distances(marked_image,
upper_feet_coordinates)
                draw_lines_and_distances(marked_image,
[upper_feet_coordinates[0],
upper_feet_coordinates[3]])
                draw_instruction(marked_image, "draw
hands points")
            elif len(hand_coordinates) < 4:
                hand_coordinates.append((x, y))
                # Draw a dot at the clicked point
                cv2.circle(marked_image, (x, y), 3, (0, 0,
255), -1)

                if len(hand_coordinates) == 4:
                    draw_lines_and_distances(marked_image,
hand_coordinates)
                    draw_lines_and_distances(marked_image,
[hand_coordinates[0], hand_coordinates[3]])
                    draw_instruction(marked_image, "draw
body points")

            elif len(body_coordinates) < 4:
                body_coordinates.append((x, y))
                # Draw a dot at the clicked point
                cv2.circle(marked_image, (x, y), 3, (0, 0,
255), -1)

                # If both points are obtained for
diameter, draw lines and distances
                if len(body_coordinates) == 4:
                    draw_lines_and_distances(marked_image,
body_coordinates, (0, 255, 255))
                    draw_lines_and_distances(marked_image,
[body_coordinates[0], body_coordinates[3]], (0, 255,
255))

                    draw_instruction(marked_image, "draw
head diameter")

```

```

        elif len(head_diameter) < 2:
            head_diameter.append((x, y))
            cv2.circle(marked_image, (x, y), 3, (255,
0, 255), -1)
            # If both points are obtained for
diameter, draw lines and distances
            if len(head_diameter) == 2:
                draw_lines_and_distances(marked_image,
head_diameter)

                if len(bottom_feet_coordinates) == 4 and
len(upper_feet_coordinates) == 4 and
len(body_coordinates) == 4 and len(hand_coordinates)
== 4 and len(head_diameter) == 2:
                    cv2.destroyAllWindows()
                    next_screen()

                # Display the marked image
                cv2.imshow('Marked Image', marked_image)

def draw_instruction(image, text):
    # create a black rectangle
    cv2.rectangle(image, (0, 0), (200, 50), (0, 0, 0),
-1)
    # create text
    cv2.putText(image, text, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
    return image

# Function to proceed to the next screen
def next_screen():
    # close opencv window
    cv2.destroyAllWindows()
    # change tkinter window
    root.geometry("300x300")
    # calculate length and diameter

    #vottom feet
    feet_x2 =
calculate_distance(bottom_feet_coordinates[0],
bottom_feet_coordinates[1])
    feet_y1 =
calculate_distance(bottom_feet_coordinates[0],
bottom_feet_coordinates[3])
    feet_y2 =
calculate_distance(bottom_feet_coordinates[1],
bottom_feet_coordinates[2])

    # upper feet

```

```

        upper_feet_x2 =
calculate_distance(upper_feet_coordinates[0],
upper_feet_coordinates[1])
        upper_feet_y1 =
calculate_distance(upper_feet_coordinates[0],
upper_feet_coordinates[3])
        upper_feet_y2 =
calculate_distance(upper_feet_coordinates[1],
upper_feet_coordinates[2])

        #hand
        hand_x2 = calculate_distance(hand_coordinates[0],
hand_coordinates[1])
        hand_y1 = calculate_distance(hand_coordinates[0],
hand_coordinates[3])
        hand_y2 = calculate_distance(hand_coordinates[1],
hand_coordinates[2])

        # body
        body_x2 = calculate_distance(body_coordinates[0],
body_coordinates[1])
        body_y1 = calculate_distance(body_coordinates[0],
body_coordinates[3])
        body_y2 = calculate_distance(body_coordinates[1],
body_coordinates[2])
        head_diameter_cm =
calculate_distance(head_diameter[0], head_diameter[1])

        feet_volume = calculate_integrate_volume([0,
feet_x2, feet_y1, feet_y2])

        upper_feet_volume = calculate_integrate_volume([0,
upper_feet_x2, upper_feet_y1, upper_feet_y2])

        hand_volume = calculate_integrate_volume([0,
hand_x2, hand_y1, hand_y2])

        body_volume = calculate_integrate_volume([0,
body_x2, (body_y1), body_y2])

        head_volume =
calculate_head_volume(head_diameter_cm)
        total = ((feet_volume + upper_feet_volume) * 2) +
(hand_volume * 2) + body_volume + head_volume

        open_button.pack_forget()
        description.pack_forget()
        text = f"Estimated Weight: {total * 0.001:.2f}
kg\nFeet Volume: {feet_volume * 0.001:.2f} kg\nBody

```

```

Volume: {body_volume * 0.001:.2f} kg\nHead Volume:
{head_volume * 0.001:.2f} kg\nHand Volume:
{hand_volume * 0.001:.2f} kg"

    label = tk.Label(root, text=text)
    label.pack()
    close_button = tk.Button(root, text="Close",
command=root.destroy)
    close_button.pack()

# Function to process the selected image
def process_image(image_path):
    global image, marked_image
    # Load the ultrasound image
    image = cv2.imread(image_path)
    marked_image = image.copy()
    draw_instruction(marked_image, "draw feet points")
    # Create a window for the image
    cv2.namedWindow('Marked Image')
    cv2.setMouseCallback('Marked Image', mark_point)

    # Get the image dimensions
    height, width, _ = marked_image.shape

    # Create a grid matrix image
    grid_image = np.zeros((height, width, 3),
np.uint8)

    # Define the grid spacing (adjust as needed)
    grid_spacing = 50 # Adjust the spacing as needed

    # Draw grid lines and label coordinates
    for x in range(width // 2, width, grid_spacing):
        cv2.line(grid_image, (x, 0), (x, height),
(255, 255, 255), 1)

    for x in range(width // 2, 0, -grid_spacing):
        cv2.line(grid_image, (x, 0), (x, height),
(255, 255, 255), 1)

    for y in range(height // 2, height, grid_spacing):
        cv2.line(grid_image, (0, y), (width, y), (255,
255, 255), 1)

    for y in range(height // 2, 0, -grid_spacing):
        cv2.line(grid_image, (0, y), (width, y), (255,
255, 255), 1)

```

```

        # Combine the grid matrix image with the marked
        image
        marked_image = cv2.addWeighted(marked_image, 1,
        grid_image, 0.5, 0)

        # Display the initial image
        cv2.imshow('Marked Image', marked_image)

        cv2.waitKey(0)
        reset_globals()
        cv2.destroyAllWindows()

def open_image():
    file_path = filedialog.askopenfilename()
    if file_path:
        process_image(file_path)

def reset_globals():
    global image, marked_image,
    bottom_feet_coordinates, body_coordinates,
    head_diameter, upper_feet_coordinates,
    hand_coordinates
    image = None
    bottom_feet_coordinates = []
    upper_feet_coordinates = []
    hand_coordinates = []
    body_coordinates = []
    head_diameter = []

if __name__ == '__main__':
    root.title("Image Measurement Tool")
    # windoww settings
    root.geometry("300x300")
    root.eval('tk::PlaceWindow . center')

    # add program description
    description = tk.Label(root, text="This program
allows you to measure weight \n of a fetal in KG.")
    description.pack()

    # Add a button to open the file dialog
    open_button = tk.Button(root, text="Choose Image",
command=open_image)
    open_button.pack()

    root.mainloop()

```

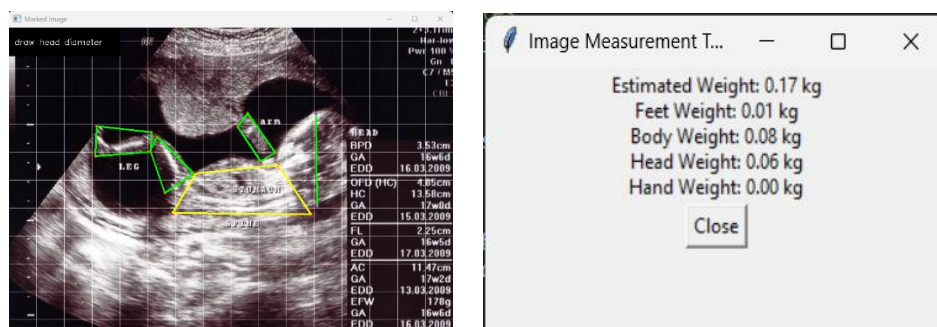

F. KESIMPULAN

Pengukuran berat janin adalah salah satu aspek kunci dalam pemantauan kehamilan yang bertujuan untuk memastikan pertumbuhan janin yang sehat dan perkembangan yang tepat. Pada studi kasus ini kami mencoba menghitung berat janin berdasarkan gambar berikut.



Gambar 7. Gambar janin hasil pencitraan USG

Dalam gambar USG janin tersebut diketahui estimasi berat janin (EFW) atau estimated fetal weight adalah 178g atau sekitar 0.17kg. Dengan gambar USG yang sama pada program ini estimasi yang dihasilkan adalah 0.17kg (dibulatkan).



Gambar 8. Proses perhitungan berat janin dan output estimasi berat janin

Untuk perbandingan, terdapat juga perhitungan manual dengan rincian hasil sebagai berikut:

1. Feet Weight

$$1.1 \text{ bottom foot} : 7,733513240013125 \text{ cm}^3$$

$$1.2 \text{ upper foot} : 4,641086692657612 \text{ cm}^3$$

2. Body Weight : 87.71780693089079 cm³

3. Head Weight : 53.7582818659732 cm³
4. Hand weight : 2,46925 cm³

Setelah mendapatkan data di atas dapat di kalkulasikan perhitungan sebagai berikut untuk mendapatkan perkiraan berat total janin :

$$\begin{aligned} \text{Perkiraan berat janin} &= ((7,733513240013125 + \\ &4,641086692657612) \times 2) + 87.71780693089079 + \\ &53.7582818659732 + (2,46925 \times 2) = 171.16378966220547 \text{ gram} \\ &\text{Di rubah ke satuan KG menjadi } 0,17116378966220547 \text{ kg} \end{aligned}$$

Dapat disimpulkan bahwa hasil perhitungan dari program maupun manual yang dilakukan adalah cukup akurat dengan hasil 0.17kg menggunakan formula integral volume dengan membagi-bagi bagian badan menjadi 5 bagian jika dibandingkan dengan gambar acuan sebelumnya yaitu 178g atau sekitar 0.17kg.

LAMPIRAN

Link Source Code Github: <https://github.com/AlephZe/FWE-prediksi-berat-janin>