

BAB III

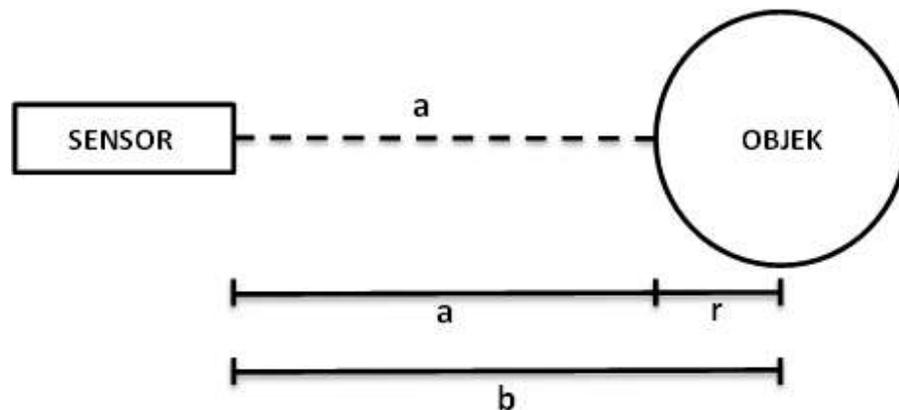
METODOLOGI PENELITIAN

A. Waktu dan Tempat Penelitian

Penelitian ini dilakukan di workshop dan laboratorium instrumentasi Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam (FPMIPA) selama dua bulan terhitung dari 1 Maret sampai dengan 30 April 2015.

B. Metode Penelitian

Pada penelitian ini sistem scanner tiga dimensi dibuat untuk mendapatkan koordinat-koordinat silinder dari benda lalu menampilkannya dalam bentuk model 3D secara *realtime*. Ketiga koordinat silinder benda diperoleh dengan cara yang berbeda-beda. Sensor ultrasonik pada penelitian ini digunakan untuk mendapatkan jari-jari (r) dari suatu benda. Caranya adalah dengan mengukur jarak antara sensor dan titik pusat benda menggunakan mistar lalu dikurangi dengan jarak antara sensor dan permukaan benda yang diperoleh dari hasil pemindaian sensor tersebut. Cara mendapatkan jari-jari benda dapat dilihat pada Gambar 3.1.



Gambar 3.1 Cara mendapatkan jari-jari benda.

Dari Gambar 3.1, rumus untuk mendapatkan jari-jari benda yaitu sebagai berikut:

$$r = b - a \quad (3.1)$$

Koordinat sudut (θ) dari bidang yang dipindai diperoleh dari rotasi benda menggunakan *motor stepper* 1. Koordinat z dari benda diperoleh dari pergerakan naik dari sensor menggunakan *motor stepper* 2. Pada penelitian ini *motor stepper* yang digunakan memiliki spesifikasi dengan maksimal 200 *step* dalam satu kali rotasi atau $1,8^\circ$ untuk setiap *step*. Akan tetapi banyaknya *step* dalam sekali rotasi dapat diatur sesuai keinginan menggunakan program. Untuk *motor stepper* yang menggerakkan sensor ultrasonik, pergerakan *motor stepper* diatur menjadi 20 *step* setiap kali bergerak. Dan setelah dilakukan pengukuran menggunakan mistar, 20 *step* dari *motor stepper* tersebut adalah 0,45 cm. Itu artinya sensor akan naik per 0,45 cm.

Setelah koordinat r , θ , dan z diperoleh, langkah selanjutnya adalah mengubah koordinat tersebut ke koordinat kartesian yaitu x , y , z . Caranya adalah dengan menggunakan rumus trigonometri seperti yang terlihat pada Persamaan 2.1a dan Persamaan 2.1b.

Pembuatan sistem *scanner* 3D pada penelitian ini terdiri dari tiga tahapan yaitu: pembuatan sistem mekanik, pembuatan sistem digital, dan pengambilan data.

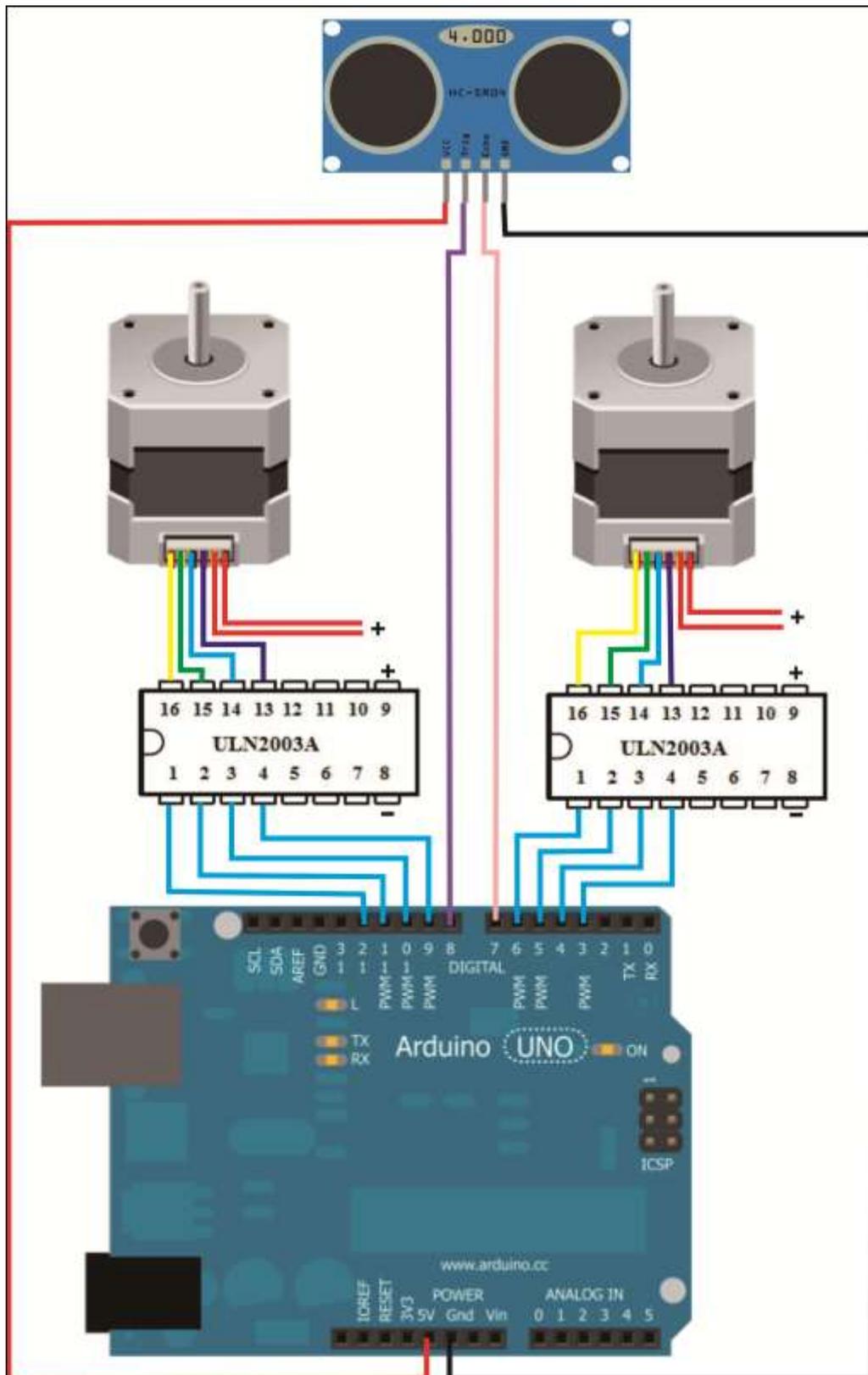
1. Sistem Mekanik

Pada penelitian ini, sistem mekanik dibuat untuk menggerakkan sensor dan benda yang akan dipindai secara otomatis. Sistem ini terdiri dari beberapa komponen seperti: papan arduino uno, dua buah *motor stepper unipolar*, dua buah IC ULN2003A, sensor ultrasonik, kabel penghubung, *v-belt*, *gear*, *breadboard*, plat besi, dan triplek. Satu *motor stepper* digunakan untuk memutar benda yang akan dipindai, sedangkan *motor stepper* satunya untuk menggerakkan sensor ultrasonik naik dan turun. Untuk memutar benda digunakan sebuah alas dari triplek yang telah dipotong bundar dan dilekatkan pada salah satu *motor stepper*. Sedangkan untuk menggerakkan sensor ultrasonik digunakan *v-belt* yang dipasangkan pada *motor stepper* dan sebuah *gear*. Sensor diletakan pada sebuahudukan yang di tempelkan pada *v-belt*, sehingga saat *v-belt* bergerak, maka sensorpun akan ikut bergerak. Bentuk dari sistem mekanik yang dibuat pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Sistem mekanik *scanner* 3D.

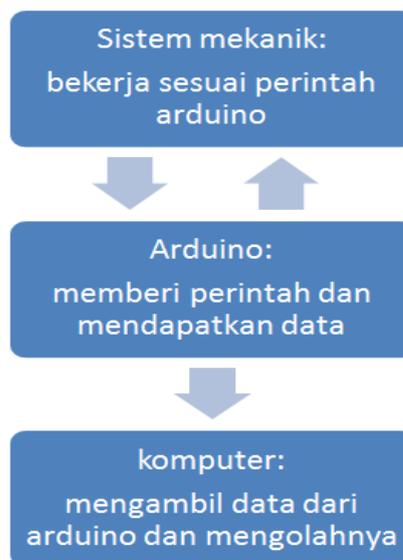
Seluruh pergerakan dari sistem mekanik dikontrol oleh arduino. Sebelum dapat dikontrol, bagian dari sistem mekanik yang terdiri dari arduino uno, dua *motor stepper*, IC ULN2003A dan sensor ultrasonik harus dirangkai seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Rangkaian sistem mekanik scanner 3D.

2. Sistem Digital

Pada penelitian ini sistem digital dibuat untuk beberapa fungsi seperti: menggerakkan dua buah *motor stepper* secara otomatis, menjalankan sensor ultrasonik, mengambil data dari sensor, serta mengolah data dan menampilkannya dalam bentuk gambar tiga dimensi secara *realtime*. Sistem ini terdiri dari dua buah *software* yaitu Arduino IDE dan Matlab. Arduino IDE digunakan untuk membuat bahasa pemrograman Arduino untuk mengatur gerakan dua buah *motor stepper* dan menjalankan sensor ultrasonik. Sedangkan Matlab digunakan untuk membuat bahasa pemrograman untuk mengambil data dari arduino menggunakan serial komunikasi, mengolahnya, dan menampilkannya dalam bentuk gambar tiga dimensi secara *realtime*. Skema dari sistem *scanner* 3D pada penelitian ini dapat dilihat pada Gambar 3.4.

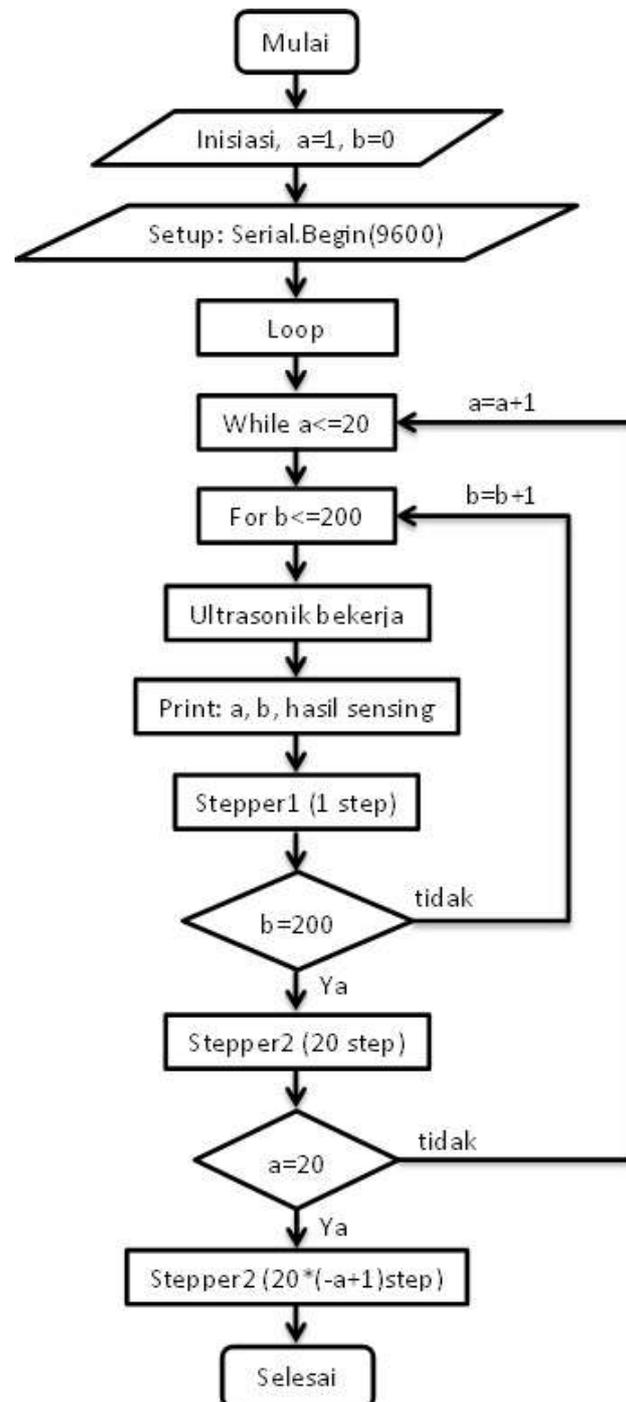


Gambar 3.4 Skema sistem *scanner* 3D dengan tampilan *realtime*.

Sistem mekanik yang terdiri dari sensor ultrasonik dan *motor stepper* bekerja sesuai dengan apa yang diprogram pada arduino dan arduino juga mendapatkan data dari hasil pemindaian sensor ultrasonik. Selanjutnya matlab mengambil data secara langsung dan berulang, matlab mengolah data dan menampilkannya dalam bentuk gambar tiga dimensi secara *realtime*. Pembuatan sistem digital ini terbagi menjadi dua yaitu pembuatan bahasa pemrograman pada Arduino IDE dan Matlab.

a. Bahasa Pemrograman Arduino IDE

Program Arduino dibuat untuk mengontrol kerja sensor ultrasonik dan dua buah *motor stepper*. Bahasa pemrograman pada Arduino IDE ini terdiri dari tiga tahap yaitu inisiasi, *setup*, dan *loop*. Diagram alir pemrograman arduino pada penelitian ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Diagram alir pemrograman Arduino.

Proses inisiasi dibuat untuk menentukan informasi awal pada suatu pemrograman seperti nilai awal, pin yang digunakan dan lain-lain. Coding inisiasi pada pemrograman ini dapat dilihat pada Gambar 3.6. Coding inisiasi ini terdiri dari informasi awal tentang *motor stepper* yang digunakan, nilai maksimum dan minimum yang bisa dipindai sensor ultrasonik serta nilai a dan b yang akan digunakan pada fungsi *loop*.

```
#include <Stepper.h>
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin

// initialize the stepper
Stepper myStepper(200, 9,10,11,12);
Stepper myStepper2(200, 3,4,5,6);

// initialize the ultrasonic
float maximumRange = 7.5; // Maximum range needed
float minimumRange = 0; // Minimum range needed
float duration, distance; // Duration used to calculate distance

int a=1;
int b=0;
```

Gambar 3.6 Program inisiasi Arduino.

Fungsi *setup* akan dipanggil saat program dimulai. Fungsi ini digunakan untuk menginisiasi pin mode, *serial begin*, dan lain-lain. Pin mode digunakan untuk menentukan suatu pin bertindak sebagai input atau output, sedangkan *serial begin* digunakan untuk membuka *serial port* dan mengatur nilai *baudrate* untuk transmisi data serial. *Baudrate* khusus untuk komunikasi dengan komputer adalah 9600. Fungsi *setup* harus dimasukkan dalam program sekalipun tidak ada pernyataan untuk dilakukan (Evans, 2007). Fungsi *setup* pada pemrograman ini dapat dilihat pada Gambar 3.7. Fungsi *setup*nya terdiri dari *serial begin*, pengaturan kecepatan *motor stepper* dan pin mode untuk sensor ultrasonik.

```

void setup() {
  Serial.begin (9600);
  // set the speed at 60 rpm:
  myStepper.setSpeed(60);
  myStepper2.setSpeed(60);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

```

Gambar 3.7 Program setup Arduino.

Fungsi *loop* digunakan untuk melakukan pengulangan, merespon, dan mengontrol papan Arduino. Pada program ini digunakan fungsi *while* dan *for* untuk mengatur kerja dua *motor stepper* dan sensor ultrasonik. Fungsi *loop* pada penelitian ini dapat dilihat pada Gambar 3.7. Pada program ini dibuat fungsi tambahan yaitu fungsi ultrasonik yang digunakan untuk mengontrol sensor ultrasonik dan mencetak nilai dari a, b, dan jarak hasil pemindaian seperti yang terlihat pada Gambar 3.9. Sebenarnya coding dalam Fungsi ini bisa langsung diletakkan pada fungsi *for*, tetapi agar tidak terlalu berantakan maka dibuatlah fungsi baru tersebut.

```

void loop() {
  while ( a<=40)
  {
    for( b=0; b<=200;)
    {
      ultrasonik();
      myStepper2.step(1);
      b++;
      delay(100);
    }
    myStepper2.step(-1);
    myStepper.step(20);
    a++;
  }
  myStepper.step(20*(-a+1));
}

```

Gambar 3.8. Program loop Arduino.

```

void ultrasonik()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  //Calculate the distance (in cm) based on the speed of sound.
  distance = duration/58.2;

  Serial.print(a);
  Serial.print("\t");
  Serial.print(b);
  Serial.print("\t");

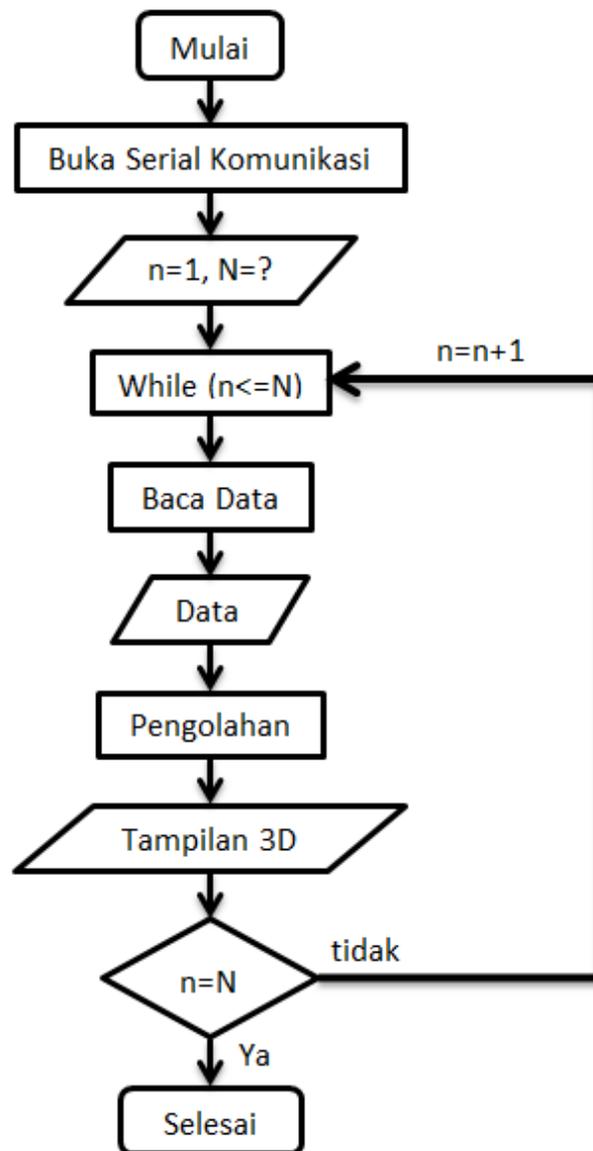
  if (distance >= maximumRange || distance <= minimumRange){
    Serial.println("7.5");
  }
  else {
    Serial.println(distance);
  }
  //Delay 50ms before next reading.
  delay(50);
}

```

Gambar 3.9 Program dari fungsi ultrasonik.

b. Bahasa Pemrograman Matlab

Pemrograman pada Matlab dilakukan untuk mengambil dan mengolah data dari arduino dan menampilkannya dalam bentuk gambar tiga dimensi secara realtime. Program pada penelitian ini terdiri dari dua bagian yaitu inisiasi dan pengulangan. Diagram alir pemrograman matlab pada penelitian ini dapat dilihat pada Gambar 3.10. Pada proses inisiasi dibuat program untuk membuka serial komunikasi antara Matlab dan Arduino serta nilai-nilai dan matrik-matrik yang akan digunakan pada saat proses pengulangan. Program inisiasi pada penelitian ini dapat dilihat pada Gambar 3.11.



Gambar 3.10 Diagram alir pemrograman Matlab.

```

clear
clc
s=serial('COM7','BaudRate',9600);
fopen(s)

A=40;
B=201;
C=A*B;
i=1;
x1=NaN(A,B);
y1=NaN(A,B);
z1=NaN(A,B);
l=NaN;
m=NaN;
n=NaN;
w=0;

```

Gambar 3.11 Program inisiasi Matlab.

Program selanjutnya adalah pengulangan. Untuk melakukan proses pengulangan, pada program ini digunakan fungsi *while*. Proses pengulangan terjadi sebanyak *i*-kali, tergantung dari benda yang dipindai. Pada saat proses pengulangan, hal pertama yang dilakukan adalah membaca data dari arduino. Setelah data diperoleh, data dibagi menjadi tiga bagian yang akan merepresentasikan nilai dari ketinggian, sudut, dan jari-jari. Setelah data diperoleh data diolah agar bisa diplot secara *realtime*. Agar bisa diplot, proses pengolahan yang dilakukan pada program ini adalah mengubah data ketinggian, sudut, dan jari-jari yang merupakan bagian dari koordinat polar menjadi koordinat kartesian yaitu *x*, *y*, dan *z*.

Pada penelitian ini, dibuat dua plot berbeda secara bersamaan yaitu *plot3* dan *surf*. Pada *plot3*, data ditampilkan dalam bentuk garis tiga dimensi. Sedangkan *surf*, data ditampilkan dalam bentuk bidang tiga dimensi. Untuk melakukan *plot3*, data *x*, *y*, dan *z* harus dalam bentuk matrik baris dengan panjang yang sama. Sedangkan untuk *surf*, data *x*, *y*, dan *z* harus dalam bentuk matrik (baris,kolom) dengan ukuran yang sama. Pada saat proses pengulangan berakhir, coding terakhir yang dibuat adalah menutup serial komunikasi. Program pengulangan pada penelitian ini dapat dilihat pada Gambar 3.12.

```

while(i<=C)
    a=fscanf(s,'%f');
    b=a(1,:);
    c=a(2,:);
    d=a(3,:);
    d=roundn(d,-1);

    e=b*0.45;
    t=c*2*pi/200;
    r=9.5-d;
    w=[w;r];
    x=r*cos(t); x1(b,c+1)=x;
    y=r*sin(t); y1(b,c+1)=y;
    z1(b,c+1)=e;
    l=[1;x];
    m=[m;y];
    n=[n;e];

    subplot(1,2,1)
    plot3(l,m,n)
    axis equal
    grid on
    hold off
    drawnow
    subplot(1,2,2)
    surf(x1,y1,z1)
    axis equal
    grid on
    hold off
    drawnow

    i=i+1;
end
fclose(s)

```

Gambar 3.12 Program pengulangan Matlab.

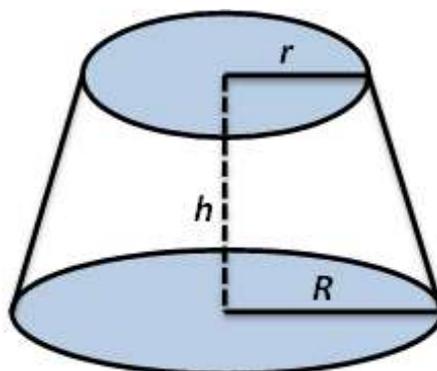
3. Teknik Pemindaian dan Pengambilan Data

Untuk mendapatkan model hasil pemindaian yang bagus, pemindaian harus dilakukan dengan beberapa cara. Hal ini dilakukan karena hasil dari pemindaian masih banyak terdapat *noise* yang disebabkan ketidakstabilan hasil dari pembacaan sensor ultrasonik. Untuk mengatasi masalah ini, teknik pemindaian dilakukan dengan menggunakan beberapa cara. Teknik pemindaian dapat dilihat pada Tabel 3.1.

Tabel 3.1 Teknik-teknik pemindaian

Cara	<i>Stepper 1 (Step)</i>	<i>Stepper 2 (Step)</i>	Plot	Keterangan
1	8	20	Pertitik	Data polar dirubah ke kartesian lalu diplot pertitik dan berkelanjutan
2	4	20	Pertitik	Data polar dirubah ke kartesian lalu diplot pertitik dan berkelanjutan
3	1	20	Pertitik	Data polar dirubah ke kartesian lalu diplot pertitik dan berkelanjutan
4	4	20	Pertitik	Data pertama dirubah ke kartesian lalu diplot. Data jari-jari kedua digabungkan dengan data pertama lalu diambil modulusnya, kemudian dirubah ke kartesian lalu diplot, dan begitu seterusnya
5	1	20	Pertitik	Data pertama dirubah ke kartesian lalu diplot. Data jari-jari kedua digabungkan dengan data pertama lalu diambil modulusnya, kemudian dirubah ke kartesian lalu diplot, dan begitu seterusnya
6	1	20	Perlingkaran	Scan dilakukan satu lingkaran lalu diambil modulus dari jari-jarinya, kemudian dirubah ke kartesian lalu diplot menjadi satu lingkaran sempurna

Setelah model 3D dari benda terbentuk, langkah selanjutnya adalah mencocokkan model 3D yang dibentuk dengan bentuk benda yang sebenarnya dalam penelitian ini yaitu botol. Cara untuk mencocokkannya adalah dengan mencocokkan volume model dengan botol yang sebenarnya. Untuk mengukur volume botol yang sebenarnya cukup dengan mengisi penuh botol tersebut dengan air dan mengukur volume air tersebut dengan gelas ukur. Sedangkan untuk menghitung volume model adalah dengan menggunakan rumus. Model yang diperoleh berbentuk kerucut tegak terpancung, yaitu bentuk yang menyerupai silinder namun berbeda jari-jari dari ketinggian satu dengan ketinggian berikutnya seperti yang terlihat pada Gambar 3.13.



Gambar 3.13 Kerucut tegak terpancung.
Sumber: Purcell & Varberg, 1987.

Maka dari itu rumus yang digunakan untuk menghitung volume model adalah rumus volume kerucut tegak terpancung. Rumus dari volume kerucut tegak terpancung adalah sebagai berikut (Purcell & Varberg, 1987):

$$Volume = \frac{1}{3}\pi(r^2 + rR + R^2)h \quad (3.2)$$

dimana r adalah jari-jari lingkaran atas, R adalah jari-jari lingkaran bawah, dan h adalah ketinggian kerucut tegak terpancung tersebut. Volume model dihitung dengan menjumlahkan volume model persuku. Hal ini dilakukan dengan program yang dibuat pada matlab seperti yang terlihat pada Gambar 3.14.

```
clc
i=2;
h=0.45;
V=0;
V1=74;
V2=438;
V3=324;
V4=512;
V5=1466;
while (i<41)
    R=w(i);
    r=w(i+1);
    V=[V+(1/3*pi*(r^2+r*R+R^2 )*h)];
    i=i+1;
end
A=V/V1*100;
disp(['Volume Benda = ',num2str(V1),' cm^3'])
disp(['Volume Model = ',num2str(V),' cm^3'])
disp(['Akurasi = ',num2str(A),' %'])
```

Gambar 3.14 Program Matlab untuk menghitung volume model.