

IMPLEMENTASI PLOTTER MENGGUNAKAN RASBERRY PI DENGAN INPUT KOORDINAT PIXEL

PLOTTER IMPLEMENTATION USING RASBERRY PI WITH INPUT PIXEL COORDINATE

Widi Santoso¹, Erwin Susanto,ST.,MT.,Ph.D², Agung Surya Wibowo.,ST.,MT³

Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹wsantoso1996@gmail.com,²erwinelektro@telkomuniversity.ac.id,³agungsw@telkomuniversity.ac.id

Abstrak

Alat percetakan atau *Printer* adalah suatu alat yang sering digunakan untuk mencetak sebuah data berupa gambar dan tulisan. Terkadang timbul berbagai masalah saat penggunaan *printer*, Contohnya seperti terbatasnya area atau media untuk mencetak gambar maupun tulisan. Teknologi *printer* juga butuh pembaruan agar bisa mengikuti perkembangan jaman dan bisa digunakan lebih efektif dari segi waktu maupun biaya.

Berdasarkan permasalahan tersebut, Maka dari itu alat yang diperlukan untuk mengurangi biaya adalah teknologi mesin plotter yang mampu membuat sebuah data gambar dan tulisan dengan sebuah pensil yang dapat digunakan di berbagai jenis area atau media cetak seperti dinding. Masukan data yang diperoleh berasal dari mini komputer yang berupa sebuah aplikasi untuk mengubah gambar menjadi koordinat *pixel*. Setelah data gambar atau tulisan terdeteksi, maka plotter akan menggambarkan setiap detail posisi *pixel* dalam bentuk koordinat *X* dan *Y*.

Pada tugas akhir ini, penulis akan mengimplementasikan sebuah masukan data berupa koordinat *pixel* (*G-Code*) yang didapatkan dari aplikasi *open source* (*Processing*). Harapan penulis adalah menjawab semua permasalahan dalam penggunaan media cetak pada *printer* dan mengembangkan teknologi mesin cetak yang lebih kreatif dan bermanfaat bagi semua orang.

Kata Kunci : printer,koordinat pixel,plotter

Abstract

Printing tools or Printers is a tool that is often used to print a data in the form of images and text. Sometimes problems arise during the use of the printer, such as the limited area or media to print images or text. Printer technology also needs updates to keep up with the times and can be used more effectively in terms of time and cost.

*Based on the problem, therefore the tools needed to reduce the cost is a plotter machine technology that is able to create an image data and writing with a pencil that can be used in various types of areas or print media such as walls. The data input obtained comes from a computer hardware in the form of an application to convert the image into pixel coordinates. After the image or text data is detected, the plotter will illustrate every detail of pixel position in the form of *X* and *Y* coordinates.*

*In this final project, the writer will implement a data input in the form of pixel coordinate (*G-Code*) obtained from open source application (*Processing*). The author's hope is to answer all the problems in the use of print media on printers and to develop a more creative and useful printing machine technology for everyone.*

Keywords: Printer,Coordinate pixel,plotter

1. Pendahuluan

Salah satu masalah pada industri desain dan produk adalah membuat desain sketsa gambar yang masih menggunakan kinerja tangan manusia, Menggambar desain produk membutuhkan ketelitian dan kecepatan agar mempercepat waktu sebuah pekerjaan. Oleh karena itu dibutuhkan alat yang dapat dikendalikan secara otomatis hanya menentukan koordinat tertentu.

Salah satu metode yang dapat digunakan untuk menggambar yaitu metode koordinat *pixel*. Koordinat *pixel* merupakan sebuah metode dasar yang menentukan sebuah pola dari sebuah gambar. inti dari sistem koordinat *pixel* adalah menggambarkan sebuah pola yang akan tersusun dari beberapa titik. Pola tersebut memiliki tingkat kesulitan yang berbeda, tergantung dari gambar yang akan di implementasikan pada gambar yang akan dibentuk.

Pada masalah ini juga dibutuhkan juga sebuah alat agar mempermudah dalam membuat gambar dari metode koordinat *pixel* yaitu plotter. Plotter adalah sebuah alat mekanik yang akan membentuk sebuah gambar dengan sebuah pensil. Dalam perancanganya juga membutuhkan beberapa komponen pendukung, komponen tersebut

antara lain adalah, *Raspberry pi 3* yang berfungsi sebagai *Monitoring data*, *Arduino Uno* berfungsi sebagai *controller*, *Dua Motor steppers* sebagai penggerak arah motor, *Servo motor* sebagai penggerak pensil, *Driver motor* sebagai penggerak *Motor steppers*.

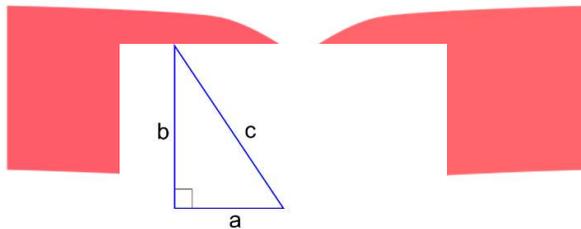
2. Dasar Teori

2.1 Plotter

Plotter merupakan jenis *printer* yang dirancang secara khusus guna menghasilkan keluaran komputer yang berupa gambar ataupun grafik [1]. Dengan menghubungkan ploter pada sistem komputer, maka berbagai bentuk gambar akan dapat digambar secara praktis. Perbedaan plotter dengan *printer*, yaitu dengan sistem *digital* untuk *printer* sedangkan plotter yaitu *analog*. Contoh ploter grafik adalah *Electro Cardiograph (ECG)* yaitu alat yang digunakan untuk mengetahui potensial dari denyutan jantung, atau jarum *seismograph* untuk mencatat getaran bumi. Ploter dapat menggambar grafik pada kertas, plastik, maupun pada plastik transparan untuk digunakan dalam proyektor.

2.2 Rumus Pythagoras

Teorema *Pythagoras* adalah suatu keterkaitan dalam geometri *Euklides* antara tiga sisi sebuah segitiga siku-siku. Teorema ini dinamakan menurut nama filsuf dan matematikawan Yunani abad ke-6 SM yakni *Pythagoras* [1].



Gambar II-1. Segitiga siku-siku[2]

$$a^2 + b^2 = c^2 \dots\dots\dots(1)$$

2.3 Metode Koordinat Pixel

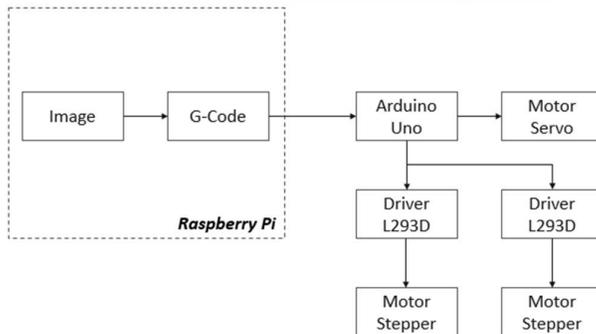
Metode koordinat ini digunakan untuk menghitung berapa jarak antara koordinat pada posisi awal misalkan $x_0 = (0,0)$ menuju koordinat selanjutnya $x_1 = (1,1)$, selain itu juga menghitung berapa jumlah step yang dibutuhkan motor stepper agar bisa menempuh posisi koordinat yang baru.

Metode ini menghitung setiap jumlah koordinat dengan cara mengurangnya agar mendapatkan nilai delta (x,y). Dengan metode ini juga akan menghasilkan koordinat baru dengan cara menjumlahkan atau mengurangi nilai koordinat yaitu dengan fungsi *increment*.

Fungsi *increment* disini bisa sebagai penambahan tergantung dari koordinat pada posisi awal sedangkan fungsi *decrement* berfungsi sebagai pengurangnya.

3. Perancangan Sistem

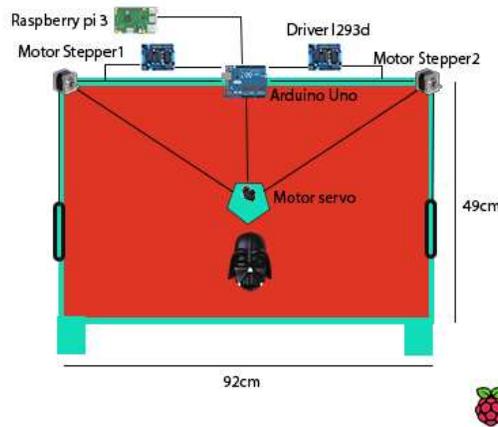
3.1 Diagram Blok Umum



Gambar III-1. Diagram Alir Seluruh system[1]

Pada Gambar III-3. Dijelaskan pertama Raspberry pi akan mengolah data gambar, lalu mengkonversikannya ke file *G-code* yang berisi koordinat (x,y).

3.2 Desain Perangkat Keras



Gambar III-3. Desain Perangkat keras[3]

Pada perancangan perangkat keras ini penulis menggunakan mikrokontroler arduino uno dan raspberry pi sebagai monitoring, lalu 2 motor stepper sebagai penggerak pensil (x,y) dan motor servo penggerak naik dan turun stepper untuk menulis/menggambar koordinat.

3.3 Cara Kerja Sistem

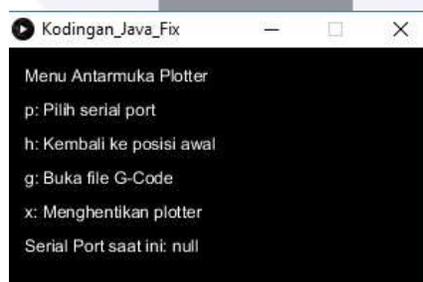
Pada tugas akhir ini menggunakan perangkat keras yang digunakan untuk diimplementasikan. Pertama alat dihidupkan kemudian Raspberry pi mengirimkan inputan data berupa koordinat pixel berbentuk *G-code*. Kemudian arduino akan membaca inputan tersebut dan mengkonversi data pixel ke jumlah step/millimeter untuk setiap koordinat. Setelah itu kedua motor stepper akan bergerak dari posisi koordinat awal (x0,y0) menuju koordinat akhir atau baru (x1,y0), Kemudian untuk menggerakkan motor servo membutuhkan kalibrasi terlebih dahulu dengan memposisikan pensil secara manual. Posisi servo akan bergerak ketika mendapatkan inputan data sepanjang koordinat yang ditempuh dari koordinat awal menuju koordinat akhir.

4. Hasil Percobaan dan Analisa

4.1 Implementasi dan Respon alat terhadap interfaces

Dalam proses mendapatkan data pengujian respon koordinat dari *G-code*, dilakukan dengan cara memasukan nilai koordinat yang terbaca oleh aplikasi *open source* yaitu *processing*.

Tahap pertama file *G-code* akan ditampilkan oleh program yang diambil dari aplikasi *processing*. Tampilan *interface* dari *processing* sebagai berikut :



Gambar IV-1. Tampilan dari Interface Dari *Processing*

Hal yang pertama pilih atau ketik P untuk serial port fungsi ini untuk menentukan Port berapa yang akan digunakan. Selanjutnya ketik G untuk membuka file *G-code* yang akan digunakan. Untuk menghentikan plotter bisa digunakan fungsi X dan fungsi H untuk mengembalikan posisi pensil ke posisi awal.

4.2 Pengujian Sistem Terhadap Perubahan Step Ke Milimeter

Pada pengujian tahap ini akan dilakukan dengan cara mengatur atau kalibrasi jumlah step untuk 1 milimeter.

$$\text{StepPerRevolution} = 360^\circ / 1.8 = 200 \text{step/rev} \dots \dots \dots (1)$$

Rumus diatas digunakan untuk mencari jumlah 1 putaran penuh dari motor stepper, yang pertama 360° akan dibagi dengan 1.8 yaitu nilai *default* untuk *mode full step* pada motor stepper. Maka didapatkan 200 step/rev.

Selanjutnya konversi step ke milimeter dilakukan secara manual menggunakan penggaris dengan mencoba kode program berikut :

```
#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

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

void loop() {
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);
```



Kalibrasi alat diperlukan agar dalam perancangan ini, untuk menghasilkan ukuran yang sama dengan seperti ukuran pada komputer atau raspberry pi. Pada perancangan sumbu X dan sumbu Y memiliki sistem pergerakan linier yang sama, yakni menggunakan gear dan tali. *Pitch* atau jarak antar gigi gear adalah 2,4mm dengan jumlah *pulley* 16 keliling lingkaranya adalah 16 x 2,4 = 38,4mm. tiap satu step motor stepper mengalami perpindahan linier sejauh :

$$\text{Perpindahan linier} = \frac{\text{perpindahan angular}}{\text{besar sudut satu putaran penuh}} \times (\text{pitch} \times \text{jumlah gigi}) \dots (1)$$

$$\frac{1.8^\circ}{360^\circ} \times 38.4 = 0.192 \text{mm}$$

Jadi untuk pergerakan linier 1 milimeter, jumlah step pada motor stepper adalah:

$$\text{Jumlah step/mm} = \frac{1}{\text{perpindahan linier}}$$

$$\frac{1}{0.192} = 5,2 \text{ step/mm}$$

Untuk memperoleh hasil pengujian yang maksimal, percobaan dilakukan berkali-kali dengan intruksi jarak perpindahan antar sumbu berbeda-beda agar dapat mengetahui jarak real yang diproses oleh plotter, Hasil pengujian dapat dilihat pada tabel dibawah ini:

Tabel IV-1. Hasil Pengujian Kalibrasi

| No | Sumbu X pada Software | Sumbu Y pada Software | Sumbu X pada Hardware | Sumbu Y pada Hardware |
|----|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 10 | 10 | 10 | 10 |
| 2 | 20 | 20 | 20.5 | 20 |
| 3 | 50 | 50 | 50 | 50.5 |

| | | | | |
|---|-----|-----|-------|-----|
| 4 | 80 | 80 | 80 | 81 |
| 5 | 100 | 100 | 100.8 | 100 |

Persentasi terjadinya *error* pada plotter ini dapat dihitung pada setiap percobaan menggunakan rumus :

$$\text{Error} = \frac{\text{nilai yang dikehendaki} - \text{nil real}}{\text{nilai yang dikehendaki}} \times 100\% \dots \dots \dots (1)$$

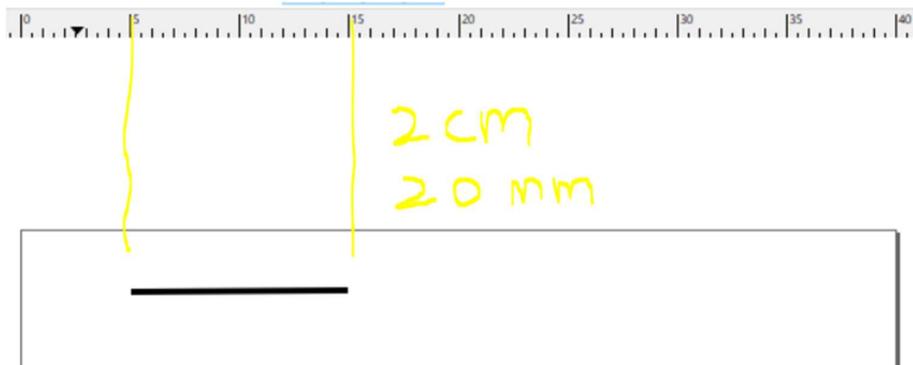
Tabel IV-2. Hasil Perhitungan Nilai Error

| No | Sumbu X (%) | Sumbu Y (%) |
|-----------------|-------------|-------------|
| 1 | 0 | 0 |
| 2 | -2.5 | 0 |
| 3 | 0 | -1 |
| 4 | 0 | -1.25 |
| 5 | -0.8 | 0 |
| Rata-rata error | -0.66 | -0.45 |

3 Uji coba Gambar garis, kotak, dan segitiga

Pada pengujian ini akan dilakukan dengan mencoba kalibrasi dalam beberapa bentuk gambar sederhana :

1. Gambar Garis



Gambar IV-2. Tampilan Gambar lurus Pada *Software*

Tampilan pada Gambar IV-2 adalah contoh gambar garis lurus yang akan di uji coba terhadap plotter yang berukuran 2cm atau 20mm.

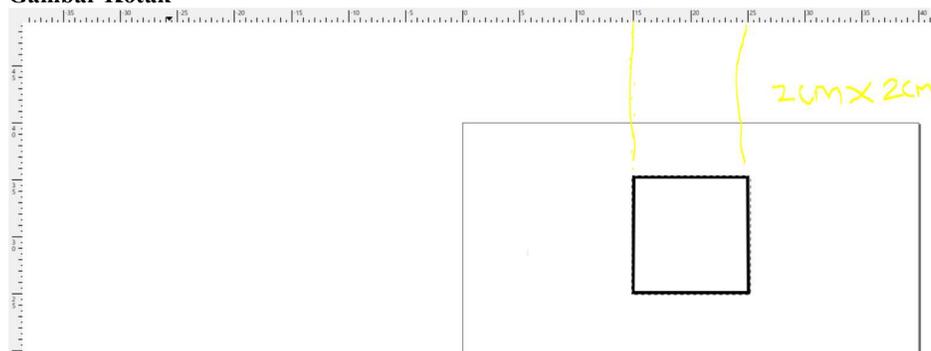


Gambar IV-3. Tampilan Gambar lurus Pada *Hardware*

Hasil setelah plotter menggambar garis lurus dapat dilihat pada gambar IV-3, nilai yang didapatkan 2,1cm.

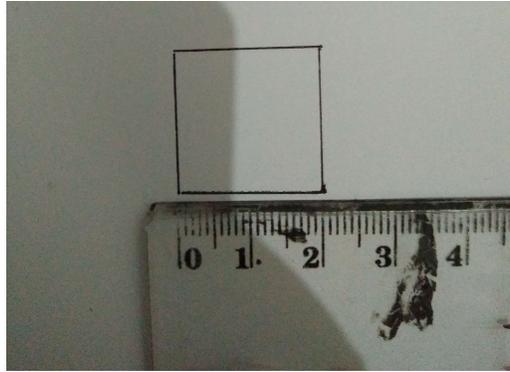
$$\text{Nilai error} = \frac{20-21}{20} \times 100\% = -0.05$$

2. Gambar Kotak



Gambar IV-4. Tampilan Gambar Kotak Pada *Software*

Tampilan pada Gambar IV-4 adalah contoh gambar kotak yang akan di uji coba terhadap plotter yang berukuran 2cm x 2cm.

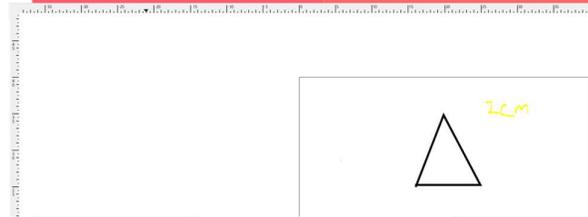


Gambar IV-5. Tampilan Gambar lurus Pada *Hardware*

Hasil setelah plotter menggambar garis lurus dapat dilihat pada gambar IV-5, nilai yang didapatkan 2cm untuk setiap garis.

$$\text{Nilai error} = \frac{20-20}{20} \times 100\% = 0$$

3. Gambar Segitiga



Gambar IV-6. Tampilan Gambar Segitiga Pada *Software*

Tampilan pada Gambar IV-4 adalah contoh gambar garis lurus yang akan di uji coba terhadap plotter yang berukuran 2cm untuk setiap garisnya.



Gambar IV-7. Tampilan Gambar Segitiga Pada *Hardware*

Tampilan pada Gambar IV-7 adalah contoh gambar garis lurus yang akan di uji coba terhadap plotter yang berukuran 2.3cm untuk setiap garisnya.

$$\text{Nilai error} = \frac{20}{20} \times 100\% = -0.2$$

4.4 Pengujian hasil akurasi gambar

Pada percobaan poin 4.3 dapat disimpulkan setiap gambar memiliki akurasi dan presisi yang berbeda-beda, maka dari itu untuk hasil persentasi keberhasilan pada alat ini dapat dilihat dari nilai error yang didapatkan untuk setiap bentuk gambar.

Nilai error yang dihasilkan dari 3 percobaan gambar garis lurus,kotak dan segitiga adalah :

Tabel IV-3. Hasil Perhitungan Nilai Error Semua Gambar

| No | Garis lurus | Kotak | Segitiga |
|-----------|-------------|-------|----------|
| 1 | -0.05 | 0 | -0.2 |
| 2 | 0 | 0 | -0.025 |
| 3 | -0.025 | 0 | -0.0375 |
| Rata-rata | -0.075 | 0 | -0.2625 |

Dari hasil perhitungan nilai rata-rata *error* dari 3 kali percobaan untuk setiap gambar yang diperoleh untuk garis dengan panjang 20mm, 40mm, dan 80mm adalah -0.075. Sedangkan untuk garis kotak diperoleh 0 dan untuk gambar segitiga adalah -0.2625.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian pada tugas akhir ini dapat ditarik beberapa kesimpulan, diantaranya:

1. Pada pengujian implementasi dan respon dari *G-code*, alat dapat merealisasikan gambar dari berbagai titik koordinat yang dihubungkan satu sama lain sesuai dengan gambar aslinya.
2. Dari data hasil uji coba untuk membentuk sebuah gambar memerlukan nilai koordinat yang di dapatkan dari 2 titik koordinat awal dan akhir ,kedua koordinat tersebut dapat menghubungkan sebuah garis agar membentuk sebuah gambar.
3. Gambar yang dibentuk dari koordinat $x_0(0),y_0(0)$ ke $x_1(20),y_1(20)$ dapat membentuk sebuah garis yang dapat dihubungkan dan membentuk sebuah gambar kotak,segitiga dan garis lurus horizontal kearah kanan.

5.2 Saran

Dari hasil penelitian yang telah dilakukan, tugas akhir ini memiliki beberapa hal yang harus diperbaiki atau bisa disempurnakan lagi berhubung dengan alat ini yang masih belum berstandarisasi atau layak uji di pasaran.

1. Untuk pengerjaan tugas akhir selanjutnya untuk skala ukuran dicoba untuk diperbesar.
2. Penentuan bidang yang digunakan bisa diterapkan pada media yang baru.
3. Menganti tempat elektronik yang layak dan berstandar pabrik.
4. Pada perancangan *user interface* lebih baik dicoba dengan aplikasi yang berbeda seperti *visual studio* agar *user interface* menjadi lebih baik dari sebelumnya.

Daftar Pustaka

- [1] Kadir, Abdul, 2013 Panduan Praktis Mempelajari Mikrokontroler dan Pemrogramannya menggunakan Arduino, Andi Offset, Yogyakarta.
- [2] Raspberry. *Raspberry pi 3 model B*.
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [3] Siswono, Tatang Yuli Eko ; Netty Lastiningsih(2007).Matematika 2 SMP Dan Mts untuk Kelas VIII.Jakarta:Esis/Erlangga.ISBN 979-734-666-8.