Aplikasi Computer Vision dengan Metode *Fuzzy Image* Processing pada Text Reader

Aditya Kurniawan
Politeknik Kota Malang
Kompleks Pendidikan Internasional Tlogowaru No 3, telp/fax (0341) 754088
e-mail: aditya@poltekom.ac.id

Abstrak – *Algoritma ini didesain untuk menginterpretasikan objek gambar berupa huruf yang nantinya akan dianalisa dan diubah dalam bentuk teks/string. Penelitian ini menggunakan scanner untuk mengambil gambar, setelah itu gambar akan diproses kedalam tahap penginterpretasian image. Sistem akan menyimpan huruf-huruf teks tersebut menjadi array dan mengurutkannya sehingga menjadi sebuah kalimat yang sama persis dengan image yang diambil. Penelitian ini mengimplementasikan sebuah metode fuzzy image processing dengan menggunakan fuzzy Multi Attribute Decision Making (MADM), yaitu sebuah logika fuzzy yang digunakan bukan sebagai control plant, melainkan digunakan untuk pengambilan keputusan berdasarkan atribut yang berjumlah lebih dari dua. Keputusan yang diambil oleh metode yang diterapkan akan menghasilkan sebuah keputusan berupa huruf yang terbaca dari sebuah image. Berdasarkan hasil uji coba dapat disimpulkan bahwa hubungan antara ukuran font dan akurasi recognizing adalah saling mempengaruhi. Huruf yang dapat di baca paling baik adalah pada index font 9, 10, dan 11 yaitu untuk ukuran font 36, 48 dan 72. Hubungan antara akurasi dan kerapatan teks untuk single dan double adalah tidak saling mempengaruhi. Hubungan antara akurasi dan kerapatan teks untuk banyaknya huruf dalam 1 baris adalah saling mempengaruhi. Sedangkan urutan huruf yang paling sulit di baca adalah kelompok huruf (E, I, X, Y, B, C) setelah itu urutan yang lebih mudah dibaca/dikenali adalah sisanya yaitu huruf A, D,E,F,G,H,J,K,L,M,N,O,P,Q,R,S,T,U,V,W, dan Z.*

Kata kunci*: Fuzzy image processing, point extraction, Histrogram image, pemrograman visual basic, text to speech*

1. Pendahuluan

Keterbatasan seseorang yang memiliki kekurangan dalam hal penglihatan (tuna netra), telah hampir pasti menjadi tembok yang sangat besar untuk orang tersebut untuk menuntut ilmu. Hal ini terjadi cobaan yang berat dalam hidup mereka, khususnya bagi mereka yang tidak mengerti huruf braile. Menjadi harga mati bagi mereka jika tidak dapat menguasainya, maka tidak ada kesempatan lagi untuk dapat membaca. Akhirnya, banyak dari mereka yang tidak dapat menjalani kehidupannya secara normal. Tuna serta, secara otomatis menjadikan mereka tuna aksara. Ketidakmampuan untuk membaca dan meraih ilmu serta cita-cita yang mereka inginkan jauh di dalam lubuk hati mereka.

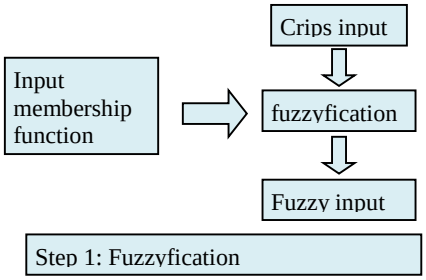
Alat ini didesain untuk menginterpretasikan objek gambar berupa huruf yang nantinya akan dianalisa menggunakan algoritma yang diperlukan untuk mengubahnya dalam bentuk teks. Alat ini menggunakan scanner untuk mengambil gambar setelah diproses dalam tahap penginterpretasian *image*, setelah selesai, sistem akan menyimpan huruf-huruf dalam teks tersebut menjadi array dan mengurutkannya sehingga menjadi sebuah kalimat yang sama persis dengan *image* yang diambil. Jika terdapat beberapa objek yang tidak dibaca sebagai teks, maka akan melakukan filtering sehingga yang tersisa hanya objek teks saja.

2. Metode Riset

2.1 Operasi Logika *Fuzzy*

a. Fuzzification

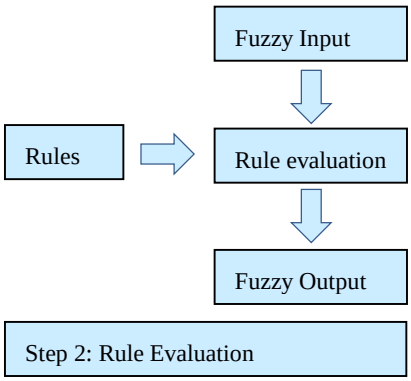
Langkah pertama yang diterapkan pada proses logika *fuzzy* adalah *fuzzy* fication yang akan mengubah input yang berupa variabel crisp (crisp Input) menjadi variabel *Fuzzy* (*fuzzy* Input). *Fuzzy* fiksasi akan mengambil nilai input crisp secara real time, seperti pembacaan temperatur dan mengombinasikannya dengan fungsi keanggotaan (membership funcation) yang tersimpan untuk menghasilkan masukan *fuzzy* sebagai contoh masukan crisp RGB 47 akan ditransformasikan menjadi agak putih dalam bentuk *fuzzy* dan 90 mph akan transformasikan menjadi cepat dan lain sebagainya. Membership function ini biasanya dinamakan membership fuction input. Dari membership function input ini nantinya akan dapat diketahui berapa derajat keanggotaan membership functionya ini. Berikut ini adalah gambar diagram mengenai *fuzzy*fication.



Gambar 1. Diagram fuzzyfication

b. Rule Evaluation

Langkah yang kedua yaitu evaluasi rule, dalam evaluasi rule ini akan dicari nilai akhir dengan memberikan bobot pada setiap aturan yang di berikan. Prosesnya adalah sebagai berikut suatu nilai *fuzzy* input yang berasal dari proses *fuzzy*fication kemudian dimasukkan ke dalam sebuah rule yang telah dibuat untuk kemudian diubah menjadi sebuah output. Gambar blok diagram dari proses ini dapat dilihat pada gambar 2 berikut ini.



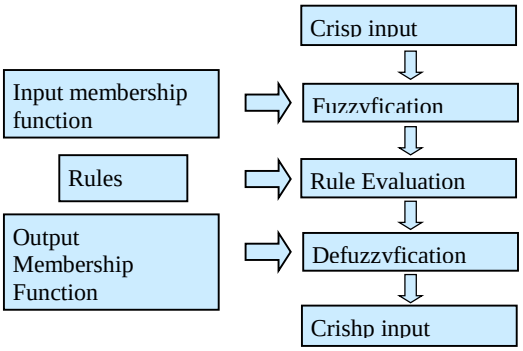
Gambar 2. Diagram proses fuzzy

c. De*fuzzy*fication

Langkah yang terakhir yaitu de*fuzzy*fikasi. Tugas dari ini adalah merubah variabel *fuzzy*yang terbentuk dari proses rule evaluation menjadi variabel crips. Proses ini merupakan tahap akhir dari proses logika *fuzzy* yang mengubah output *fuzzy* menjadi output crisp. Satu nilai *fuzzy* output yang berasal dari *rule evaluation* di ambil kemudian dimasukkan ke dalam suatu membership fuction output adalah bentuk singleton yaitu garis lurus *vertikal* ke atas. Besar nilai bentuk singleton yaitu garis lurus *vertikal* ke atas. Besar nilai *fuzzy* output dinyatakan sebagai *degree of membership function output*. Nilai-nilai tersebut nantinya akan dimasukkan ke dalam suatu rumus

ISSN (e): 2540-9123
ISSN (p): 2502-1982

yang dinamakan COG (*center of gravity*) untuk mendapatkan hasil akhir yang disebut crisp output. *Crisp output* adalah suatu nilai analog yang dibutuhkan untuk mengolah data pada system yang dirancang. Struktur dasar dari controller ini adalah sebagai berikut.



Gambar 3. Struktur dasar fuzzy

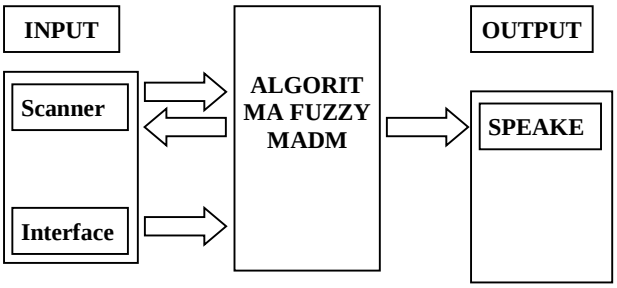
Adapun algoritma yang digunakan adalah

$$\underset{[y]}{\text{Crisp Output}} = \frac{\sum_i (\text{fuzzy output}_i) \text{ x } (\text{Singleton position on x axis}_i)}{\sum_i (\text{fuzzy output}_i)} \quad (1)$$

3. Hasil dan Pembahasan

3.1 Blok Diagram Sistem

Berikut adalah blok diagram sistem



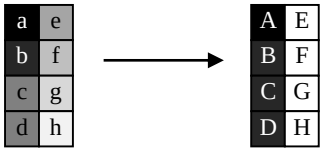
Gambar 4. Blok diagram sistem

3.2 Proses Algoritma Text Reader

a. Binerisasi

Binerisasi *Image* adalah sebuah langkah/ tahap untuk memperkecil dan mengurangi kemungkinan gradasi warna yang muncul dalam sebuah *image* memilik 3 komposisi warna untuk setiap pixel yang menyusunnya. Yaitu komposisi warna R, G, dan B. Dengan R untuk varian nilai warna merah, G untuk hijau. Dan B untuk biru. Hasil capture huruf yang telah didapatkan telah dibatasi untuk warna hitam saja, tetapi karena iluminasi / pencahayaan yang kompleks dan beragan untuk setiap cetakan huruf, maka proses binerasasi ini menjadi sangat penting, di samping untuk memudahkan proses recognizing yang menggunakan logika digita sebagai input *fuzzy*.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

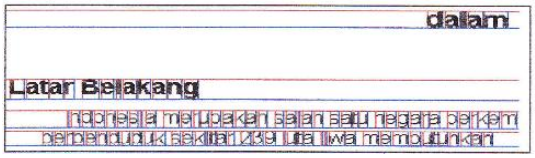


Gambar 5. Proses binarisasi

b. Text Wrapping

Text wrapping atau *clustering* adalah sebuah proses untuk memisahkan / melokalisir huruf dari kata / kalimat / dan paragraf dalam *image* text. Keakurasian dan keberhasilan proses ini sangatlah penting untuk proses *recognizing*, dikarenakan dalam proses inilah sebuah *image* text yang belum memiliki aglomerasi / pengelompokan yang jelas akan dijadikan beberapa segmen / cluster *image* yang mengindikasikan posisi dari tiap-tiap huruf yang terdapat dalam *image* tersebut.

$$S = \frac{\sum_{i=0}^{X-1} (P(I,x_0)) \,.\, (P(I,x_i))}{X}$$



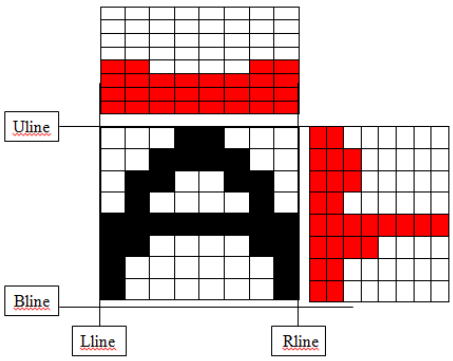
Gambar 6. Contoh *wrapped text*

c. Histogram *Image*

Histogram *image* adalah proses untuk mendata./mempresentasikan jumlah pixel yang menyusun sebuah *image*. Histogram dalam proses juga dipakai sebagai penanda karakteristik / pembeda sebuah *image*. Maka dari itu hasil dari histogram ini akan dimasukkan sebagai input *fuzzy* sebagai bahan pertimbangan pengambilan keputusan. Proses ini penting untuk menentukan tingkat keberhasilan pengenalan teks. Sebuah *image* yang telah melewati proses binerisasi dan *clustering* telah siap untuk dikenali.

Proyeksi histogram V adalah representasi dari semua komponen pixel pembangun *image* A dan diproyeksikan pada axis I dan H adalah representasi dari semua komponen pixel pembangun *image* A dan diproyeksikan pada axis J.histogram V dan H.

$$Vmid = Bline - \sum_{x=Uline}^{Bline} (P(I,x)) \quad (2)$$



Gambar 7. Histogram pada huruf A

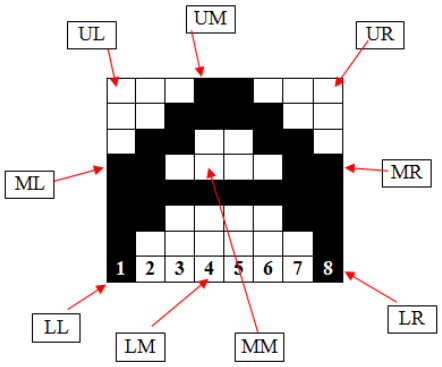
ISSN (e): 2540-9123
ISSN (p): 2502-1982

d. Point Extraction

Point extraction adalah sebuah proses dimana pixel pada posisi tertentu (dalam hal ini terdapat 9 point) memiliki nilai tertentu atau tidak. Sebuah *image* huruf adalah sebuah *image* yang memiliki karakteristik yang berbeda. Salah satu karakteristik dari *image* huruf tersebut adalah posisi pixel pembentuknya yang sudah pasti berbeda. Karena itulah proses point *extraction* ini diperlukan untuk diferensiasi *image* tersebut dan nantinya sebagai input kedua dari *fuzzy* sebagai pengambil keputusan. Proses ini penting untuk system ini Karena digunakan sebagai input dalam *fuzzy*. Terdapat 2 proses dalam tahap ini, yaitu

1. *Invert value* dan
2. *Point extraction*

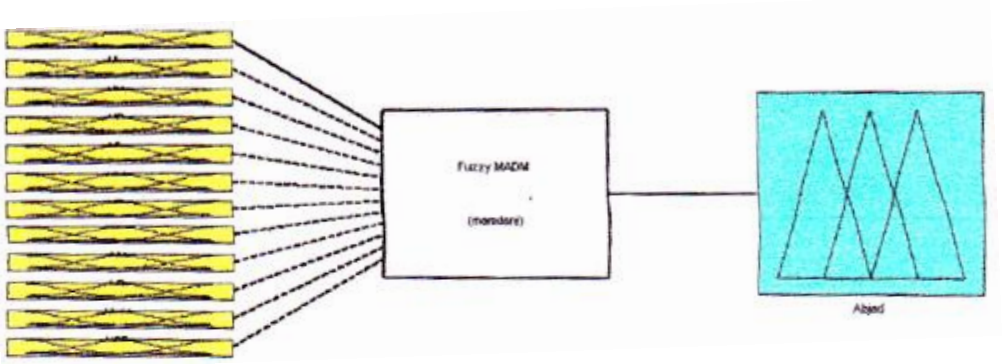
Karena dalam proses ini nilai 1 dan 0 dari *image* tidak mempresentasikan komposisi warna, melainkan ada atau tidaknya pixel pembangun *image*, maka dilakukan invert value dengan kondisi active high, yaitu dimana terdapat pixel pembangun *image* pada posisi tertentu, maka nilainya adalah 1 (*active*)/ON. Jika tidak maka 0 (*inactive*)/OFF.



Gambar 8. Posisi point pada huruf A

3.3 Data capture Kinect sensor dengan averaging dua layer

Fuzzy MADM / *fuzzy Multi Athibute Decision Making* adalah sebuah *fuzzy* logic yang digunakan untuk pengambilan keputusan dengan menggunakan input berjumlah banyak (multi atribute). Dalam hal ini, decision / keputusan yang dibuat adalah untuk memutuskan abjad sebagai ouput dari *fuzzy*, dengan 12 input yang didapatkan dari proses/tahap yang telah dilakukan sebelumnya. 3 input didapatkan dari proses histogram *image*, dan 9 input didapatkan dari proses *point extraction*. Dengan 9 input *point extraction* yaitu UL, UM, UR, ML, MM, MR, LL, LM, LR dengan menggunakan konvensi crisp set. Dan 3 input histogram Vmid, H25 dan H75 dengan menggunakan *fuzzy* set.



Gambar 9. Diagram blok fuzzy MADM

ISSN (e): 2540-9123
ISSN (p): 2502-1982

A. Konvensional crisp set input

Jika X1 adalah nilai *fuzzy* untuk input UL dengan range nilai 0 - l, X2 untuk input UM, X3 untuk input UR, X4 untuk input ML, X5 untuk input MM, X6 untuk input MR, X7 untuk input LL, X8 untuk input LM, dan X9 untuk input LR maka membership function untuk input ini adalah

$$X1 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X2 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X3 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases}$$

$$X4 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X5 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X6 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases}$$

$$X7 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X8 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases} \qquad X9 = \begin{cases} 0, & x < 1 \\ 1,, & x \geq 1 \end{cases}$$

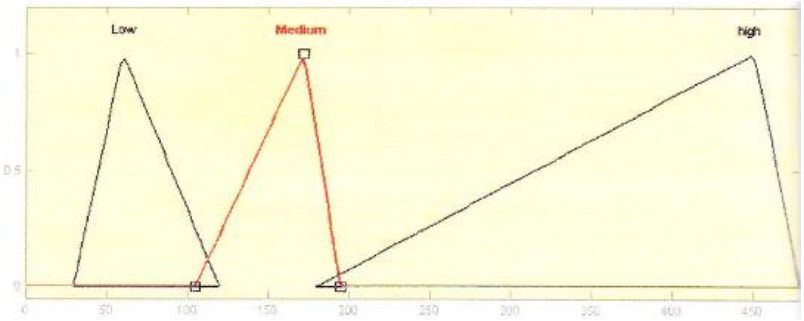
Jika X10 adalah nilai *fuzzy* LOW *membership function* Vmid, X11 adalah nilai *fuzzy* MEDIUM serta X12 adalah nilai *fuzzy* HIGH, dengan Xa untuk titik pangkal *fuzzy* set, Xb untuk titik ujung, dan x untuk nilai input dengan range nilai 0 – 470 untuk Vmid, maka

$$X10 = \begin{cases} 0 & ,120 < x < 30 \\ (Xa - x)/(Xa - Xb) & ,30 < x < 60 \\ 1 & ,x = 60 \\ (x - Xa)/(Xb - Xa) & ,60 < x < 120 \end{cases}$$

$$X11 = \begin{cases} 0 & ,195 < x < 105 \\ (Xa - x)/(Xa - Xb) & ,105 < x < 175 \\ 1 & ,x = 172 \\ (x - Xa)/(Xb - Xa) & ,172 < x < 195 \end{cases}$$

$$X12 = \begin{cases} 0 & ,480 < x < 180 \\ (Xa - x)/(Xa - Xb) & ,180 < x < 450 \\ 1 & ,x = 450 \\ (x - Xa)/(Xb - Xa) & ,450 < x < 480 \end{cases}$$

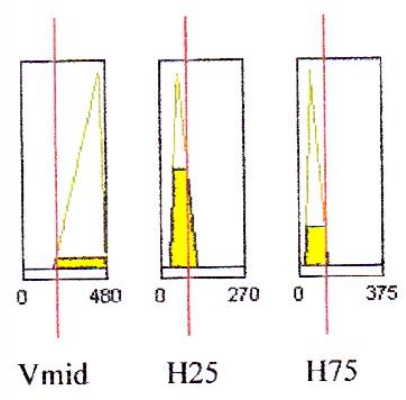
Dengan perhitungan di atas, maka didapatkan *fuzzy* set untuk input Vmid seperti yang terlihat pada gambar



Gambar 10. Fuzzy set untuk Vmid

Gambar input *fuzzy* untuk Vmid, H25 dan H75 dapat dilihat dari gambar 11.

ISSN (e): 2540-9123
ISSN (p): 2502-1982



Gambar 11. Fuzzy set untuk H25, H75

Hasil dari proses ini menunjukkan bahwa output *fuzzy* adalah 0.52. meliha crisp *fuzzy* output Abjad seperti yang ada pada gambar 3,30, maka keputusan atau *decision* dari proses ini adalah abjad A.

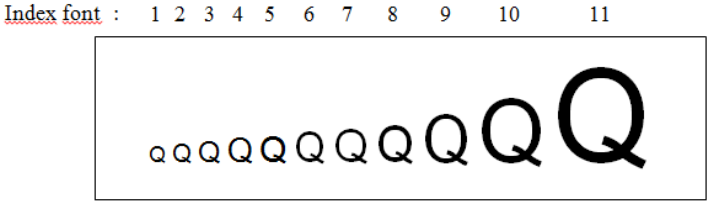
3.4 Data capture Kinect sensor dengan averaging 3 layer

Target pengujian pada penelitian ini yaitu:

a. Mendapatkan persentase keakuratan pada tiap tiap setting antara ukuran font dan kerapatan teks dalam 10 kali ulangan, serta
b. Mengetahui dan menganalisa sebab sebab sebuah huruf memiliki error pada saat pembacaan.

1. Bahan Uji

Bahan uji yang digunakan adalah huruf Q dalam index ukuran 1 sampai dengan 11



Gambar 12. Spesimen uji huruf Q

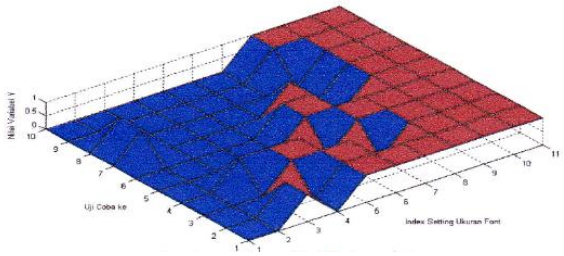
Jika keakuratan dianggap sebagai variabel Y dengan presentase nilaI/banyaknya ulangan dengan ukuran font sebagai variabel terikat X maka hubungan Untuk X dengan index 1 (font 12) sampai 11 (font 72), sehingga semakin besar nilai index menandakan ukuran font yang semakin besar dan Y (0%) sampai (100%),

2. Hasil Uji Coba

Hasil rekam data uji coba dapat dilihat di tabel pada lampiran A tentang tabel hasiluji coba untuk hubungan keakuratan dan variasi ukuran font.

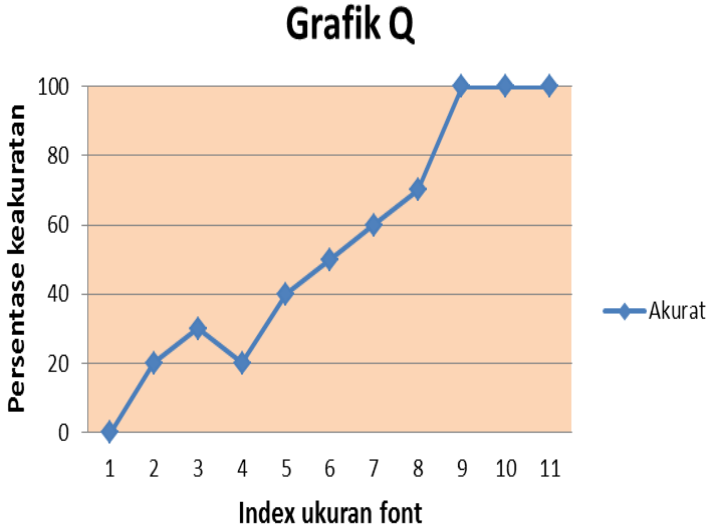
Grafik penyajian data uji coba untuk huruf Q.

ISSN (e): 2540-9123
ISSN (p): 2502-1982



Gambar 13. Hasil uji coba keakuratan

Dengan melakukan proses mean (rataan) dari 10 uji coba, maka didapatkan plot grafik Q untuk presentase keakuratan tiap setting sebagai berikut.

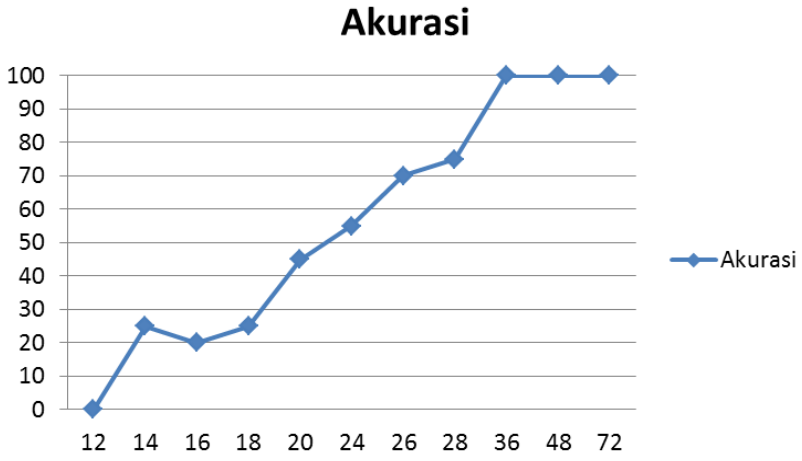


Gambar 14. Grafik Q

3. Analisa

Dari data yang telah didapatkan di atas, maka dapat disimpulkan bahwa hubungan antara ukuran font dan akurasi recognizing adalah **saling mempengaruhi** atau dapat disimpulkan semakin besar ukuran font yang di baca, maka akurasi pembacaan akan semakin besar. Huruf yang dapat dibaca paling baik adalah pada index font 9, 10, dan 11 untuk ukuran font 36, 48, dan 72. Berdasarkan analisa yang telah dilakukan pada bab 4 maka jawaban untuk rumusan masalah yang pertama yaitu akurasi pembacaan untuk tiap setting font adalah, untuk variable X sebagai ukuran font dan Y sebagai akurasi dengan menggunakan netide mean (rata-rata) adalah

ISSN (e): 2540-9123
ISSN (p): 2502-1982



Gambar 15. Grafik akurasi

1. Ukuran font 12 : 10%
2. Ukuran font 14 : 25%
3. Ukuran font 16 : 20%
4. Ukuran font 18 : 25%
5. Ukuran font 20 : 45%
6. Ukuran font 24 : 55%
7. Ukuran font 28 : 75%
8. Ukuran font 36 : 100%
9. Ukuran font 48 : 100%
10. Ukuran font 72 : 100%

4. Kesimpulan

Berdasarkan analisa di atas, maka akurasi teks yang dapat dibaca/dikenali dengan keberhasilan 100% masih berukuran cukup besar yaitu font 36 sampai dengan 72, Penelitian mendatang diharapkan dapat mengurangi kelemahan ini yaitu diantaranya dengan :

1. Peningkatan DPI *image*
2. Pengaplikasian algoritma deblurring sehingga akurasi pembacaan pada image berukuran *index* 1 sampai dengan 7 pada DPI yang rendah dapat menjadi lebih baik.

Daftar Pustaka

[1] *Al-Shabi, Mohammad :* Text Detection and Character Recognition Using Fuzzy Image Processing Journal of ELECTRICAL ENGINEERTNG, *VOL. 57, NO. 5, 2006*, 258-267

[2] *Peng. H – Long, F. – Siu, W. – Chi, Z. – Feng. D.* : Document Image Matching Based on Component Blocks, *Proceedings of 2000 International Conference*, Canada, 2000; 2, 601 – 604.

[3] Castleman, K. (1979). Digital image processing. 1st ed. Englewood Cliffs, N.J.: Prentice-Hall.

[4] Kurniawan, A., & Wardani, K. (2017). Pengaruh Isolated Neighborhood-Averaging Filters Pada Kinect Structural Noise Sebagai Sistem Navigasi Robot Wild Thumper. *TELKA-Telekomunikasi, Elektronika, Komputasi dan Kontrol*, *3*(1), 49-56.