



Terbit online pada laman web jurnal :
<http://ejournal.amikompuwoko.ac.id/index.php/telematika/>

Telematika

Accredited SINTA “2” Kemenristek/BRIN, No. 85/M/KPT/2020



Sistem Komunikasi Augmentatif dan Alternatif Berbasis Tracking Realtime Mata

Yogi Ramadhani¹, Dita Dayu A.W.², dan Retno Supriyanti^a

^{1,2,3} Jurusan Teknik Elektro, Fakultas Teknik
 Universitas Jenderal Soedirman

Email : yogi.ramadhani@unsoed.ac.id¹, ditadayu11@gmail.com², retno.supriyanti@unsoed.ac.id³

ARTICLE INFO

Article History:

Received May 31, 2020

Revised June 20, 2020

Received August 24, 2020

Online August 31, 2020

Keywords:

Augmentative

Alternative

Communication

Eyes Tracking

Kata kunci:

Komunikasi

Augmentatif

Alternatif

Tracking

Mata

Correspondence:

Telepon: +62 816 69 7221

E-mail:

yogi.ramadhani@unsoed.ac.id

ABSTRACT

Communication focuses on understanding of a person to another person, a simple communication can occur if there are similarities between the delivery of messages and people who receive messages. However, there are some people who lose communication skills or language and speech disorder which is a side effect of some diseases. One think to facilitate the problem is building a Augmentative and Alternative Communication (AAC) Systems. AAC is a communication method to support speaking or writing abilities for someone who have oral and writing disorders. In this research, eye-gaze boards used to implant AAC method. The computer programs created to implement that methods, with centroid iris tracking as input variable. The tests were carried out on various respondents with various eye shapes. The result of this research show that iris centroid could represent eyes direction, with six classifications (left – middle – right and upper – lower).

ABSTRAK

Proses komunikasi berfokus pada pemahaman antar pihak yang berkomunikasi. Dengan kata lain, komunikasi terjadi bila ada kesamaan persepsi antara pengirim pesan dan penerima pesan. Namun demikian, beberapa orang kehilangan kemampuan berkomunikasi atau memiliki gangguan bahasa dan bicara karena efek dari penyakit yang dideritanya. Salah satu upaya untuk memfasilitasi hal tersebut adalah dengan membuat suatu sistem komunikasi augmentatif dan alternatif (AAC). AAC merupakan metode komunikasi untuk membantu maupun menggantikan kemampuan bicara maupun menulis bagi penderita gangguan komunikasi lisan serta tulis. Pada penelitian ini, penerapan metode AAC menggunakan eye-gaze boards. Metode tersebut diimplementasikan dalam suatu program komputer, dengan input berupa tracking centroid iris. Pengujian dilakukan pada berbagai responden dengan berbagai bentuk mata. Hasil dari penelitian menunjukkan bahwa centroid iris dapat merepresentasikan arah penglihatan, dengan klasifikasi enam arah (kiri-tengah-kanan dan atas-bawah).

PENDAHULUAN

Komunikasi adalah pengiriman dan penerimaan berita atau pesan dari dua orang atau lebih supaya pesan yang dimaksud bisa dipahami (KBBI, 2016). Komunikasi merupakan proses dimana seseorang yang sedang berusaha memberikan pengertian dengan cara pemindahan pesan. Secara sederhana komunikasi dapat terjadi apabila ada kesamaan antara penyampaian pesan dan orang yang menerima pesan (Rohim, 2016). Hal ini menunjukkan komunikasi menitikberatkan pada pemahaman ataupun pengertian suatu pihak terhadap pihak yang lain. Namun ada beberapa orang yang kehilangan kemampuan komunikasi atau gangguan bahasa dan bicara yang merupakan efek samping dari beberapa penyakit seperti *cerebral palsy*,

amyotrop hiclateral sclerosis (ALS) dan *Parkinson's disease*. Berkurangnya kemampuan komunikasi ini dapat menyebabkan penderita menganggap seolah-olah orang di sekelilingnya tidak lagi dapat memahami dirinya. Hal ini dapat menyebabkan stres/depresi ketika orang di sekelilingnya salah memahami (Beukelman & Light, 2020).

Salah satu upaya untuk membantu komunikasi para penderita gangguan komunikasi adalah dengan membangun sistem komunikasi augmentatif dan alternatif atau *augmentative and alternative communication (AAC)* (ASHA, 2017). Pada aplikasinya sistem AAC umumnya membutuhkan gerakan, gestur maupun bahasa isyarat yang menyulitkan penderita karena *penderita cerebral palsy, amyotrophic lateral sclerosis* dan *Parkinson's disease* kehilangan kemampuan gerak atau motorik, sehingga dibutuhkan gerak yang sangat minim yaitu melalui gerakan bola mata (Kamble, 2017).

Penggunaan alat bantu komputer untuk menerjemahkan pola gerak bola mata menjadi suatu rangkaian kata tentu akan sangat membantu para penderita gangguan komunikasi (Vanderheiden, 2003). Salah satu aplikasinya adalah dengan mengembangkan suatu perangkat komunikasi AAC menggunakan sistem *tracking realtime* mata.

METODE PENELITIAN

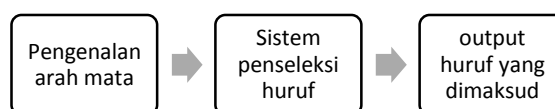
1. Alat dan Bahan

Pada pengembangan sistem komunikasi ini, alat dan bahan yang digunakan adalah sebagai berikut:

- MATLAB 2017
- Komputer dengan spesifikasi cukup untuk menjalankan MATLAB 2017, dalam hal ini digunakan komputer dengan processor setara Intel i3, RAM 8GB, dengan sistem operasi Windows 8 64-bit.
- Kamera mini / *webcam* dengan resolusi minimal 1MP.
- Kawat untuk membentuk pegangan kamera .

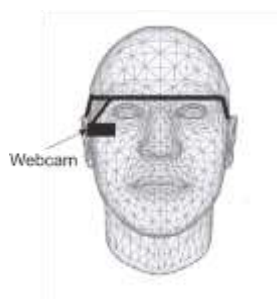
2. Desain Sistem

Desain secara singkat digambarkan dalam diagram blok pada Gambar 1. Sebagai input awal, sistem akan mendeteksi posisi bola mata, yang selanjutnya akan mengenali arah pergerakan bola mata. Setelah pengenalan arah mata didapat, sistem akan menerjemahkannya sehingga didapat hasil berupa huruf yang dimaksud oleh pengguna.



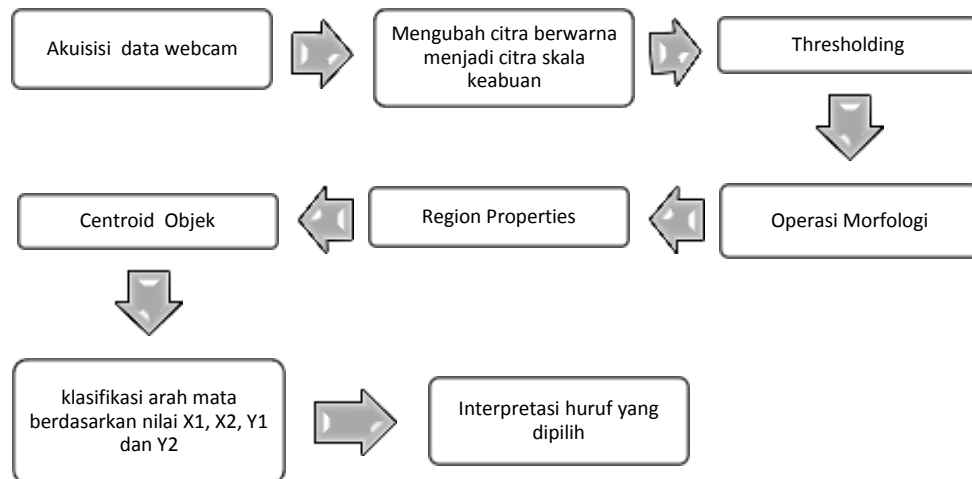
Gambar 1. Diagram blok desain sistem

Pada proses pengenalan arah mata, data diambil dari akuisisi citra mata oleh *webcam*, yang diletakkan di depan mata seperti pada Gambar 2.



Gambar 2. Posisi Webcam

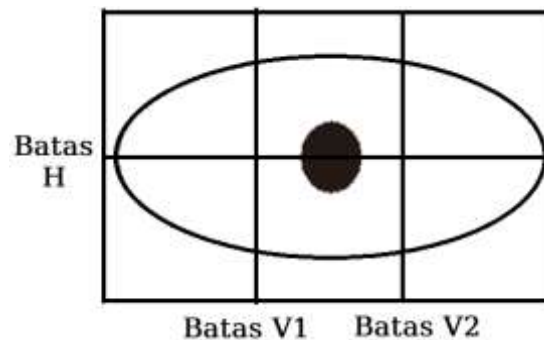
Lebih lanjut, dilakukan proses pengenalan bola mata dengan metode pengolahan citra digital. Diagram alir proses ini ditunjukkan pada Gambar 3.



Gambar 3. Proses Deteksi Arah Mata

Data citra bola mata yang diakuisisi dari *web cam* akan diubah menjadi citra skala abu-abu, kemudian dilakukan operasi *thresholding*. Operasi morfologi citra hasil *thresholding* dilakukan untuk mencari atribut centroid mata, yang selanjutnya dapat digunakan untuk mengklasifikasi arah mata. Hasil klasifikasi arah mata diterjemahkan oleh sistem untuk menginterpretasi huruf yang dipilih (Bilal, 2018).

Pada penelitian ini, menggunakan 6 arah mata yaitu kiri atas, tengah atas, kanan atas, kiri bawah, tengah bawah, dan kanan bawah. Dengan koordinat titik tengah atau *centroid* lingkaran, dapat diketahui koordinat tersebut lebih cenderung ke kiri atas, tengah atas, kanan atas, kiri, tengah, kanan, kiri bawah, tengah bawah, atau kanan bawah, seperti ditunjukkan pada Gambar 4.



Gambar 4. Skema Area Penglihatan

Pada Gambar 4, terlihat penentuan batas koordinat pada bidang X dan Y. Hal ini dilakukan untuk membedakan arah mata apakah melihat kanan, tengah ataupun kiri. Aturan/*rule* yang digunakan adalah sebagai berikut.

- Jika $X_{mata} < \text{Batas V1}$ dan $Y_{mata} < \text{Batas H}$ maka mata melihat **kiri atas**.
- Jika $X_{mata} < \text{Batas V1}$ dan $Y_{mata} > \text{Batas H}$ maka mata melihat **kiri bawah**.
- Jika $\text{Batas V1} < X_{mata} < \text{Batas V2}$ dan $Y_{mata} < \text{Batas H}$ maka mata melihat **tengah atas**.
- Jika $\text{Batas V1} < X_{mata} < \text{Batas V2}$ dan $Y_{mata} > \text{Batas H}$ maka mata melihat **tengah bawah**.
- Jika $X_{mata} > \text{Batas V2}$ dan $Y_{mata} < \text{Batas H}$ maka mata melihat **kanan atas**.
- Jika $X_{mata} > \text{Batas V2}$ dan $Y_{mata} > \text{Batas H}$ maka mata melihat **kanan bawah**.

Metode seleksi yang digunakan mengadaptasi sistem *eye-gaze board* (S.V. Kamble, 2017). Pada *eye-gaze board* terdapat kelompok huruf yang masing-masing terletak pada sisi yang berbeda. Pengguna pertama akan diminta untuk melihat kelompok huruf yang diinginkan kemudian pengguna akan diminta untuk melihat letak huruf yang diinginkan (Wanjari & Khode, 2014). Contohnya ketika pengguna ingin memilih huruf B maka pengguna tersebut akan melihat kiri atas untuk memilih kelompok ABCDEF, kemudian karena huruf B berada di tengah atas maka kemudian pengguna tersebut harus melihat tengah atas.



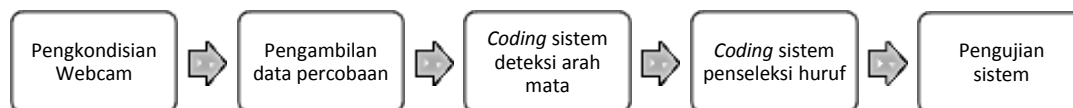
Gambar 5. Papan *Eye-Gaze Board* yang Digunakan

3. Pengujian

Pada tahap pengujian akan dilakukan pengujian terhadap sistem AAC dengan melihat keakuratan sistem pengenalan arah mata. Untuk pengujian pengenalan arah mata dilakukan pengujian terhadap enam arah penglihatan. Pengujian pada 24 orang dengan masing-masing melakukan gerakan mata enam (total 144 proses). Hasil pengujian menunjukkan nilai kesalahan relatif kecil, mendekati 7%.

4. Alur Penelitian

Ringkasan alur penelitian ini dapat dilihat pada Gambar 6.



Gambar 6. Diagram Alur Penelitian

Pada Gambar 6, langkah awal adalah mengondisikan *web cam* agar dapat terhubung ke sistem, kemudian mengambil sampel data uji untuk dikompilasi pada proses *coding* sistem deteksi arah mata hingga sistem penyeleksi huruf. Langkah selanjutnya adalah uji coba implementasi sistem.

HASIL DAN PEMBAHASAN

1. Pengambilan Data Mata

Proses pengambilan data dilakukan dengan menggunakan webcam. Untuk mengakuisi data dari *webcam*, menggunakan *add-on webcam* pada MATLAB. Sintaks yang digunakan pengambilan data dilakukan dengan menggunakan webcam sebagai berikut.

```

cam=webcam(2);
NumberOfFrames=10000;
...
for count=1:NumberOfFrames
    ...
    % proses pengolahan
    ...
end
  
```

Pada sintaks pengambilan data menggunakan webcam, Angka 2 menunjukkan indeks atau urutan koneksi kamera dengan komputer. Pada komputer yang digunakan pada penelitian ini, webcam (1) adalah kamera yang sudah terintegrasi pada komputer dan webcam (2) adalah USB webcam eksternal.

Pengambilan gambar dilakukan secara *realtime*, data yang diambil bergantung pada gerakan mata. Sebagai pembatas, digunakan 10.000 frame untuk pengambilan gambar, di mana setiap frame digunakan untuk 1 data huruf. Hasil pengambilan gambar ditunjukkan pada gambar 7(a).

2. Pengolahan Citra Mata

Data mata yang telah didapatkan kemudian diproses dengan prinsip pengolahan citra digital. Proses pengolahan citra, antara lain meliputi : *grayscale*, *thresholding*, *image filling*, *erotion*, dan *image filtering* (Sutoyo, Mulyanto, Suhartono, & Nurhayati, 2009). Proses ini pada akhirnya mendapatkan suatu obyek *centroid* mata

a. Grayscale

Pada proses ini, data gambar yang telah diambil dikonversi, dari citra berwarna menjadi citra *grayscale*. Sintaks yang digunakan adalah sebagai berikut.

```
gambar=snapshot (cam) ;
gambar=rgb2gray (gambar)
```

Fungsi `rgb2gray` akan mengkonversi citra gambar dan menyimpan hasil konversi kembali ke variabel gambar.



7(a) Hasil Gambar Awal



7(b) Hasil Gambar Setelah *Grayscale*

Gambar 7. Gambar Mata Sebelum dan Setelah Proses *Grayscale*

b. Thresholding

Thresholding adalah proses mengubah citra *grayscale* menjadi citra biner dengan menggunakan nilai ambang sebagai patokan untuk mengubah piksel menjadi hitam atau putih (Gonzales, 2008). Secara matematis proses ini dirumuskan sebagai berikut.

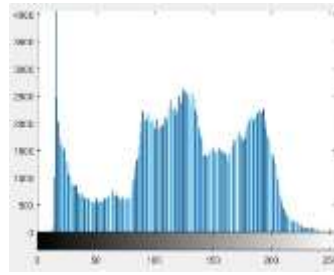
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \quad \dots\dots\dots (1)$$

Piksel menjadi 1 ketika piksel citra lebih besar dari nilai *threshold* dan mengubah piksel menjadi 0 ketika piksel citra lebih kecil atau sama dengan nilai *threshold*.

```
gmbrad=imadjust (gambar) ;
gmbr=gmbrad;
gmbr=im2bw (gmbr, 0.1) ;
```

Sintaks diatas akan mengubah citra grayscale gambar menjadi citra biner yang disimpan pada variabel gmbr dengan threshold luminan T, dengan nilai T dari 0 hingga 1. Penentuan nilai




T dilakukan dengan analisis terhadap luminans citra, dengan menggunakan histogram citra seperti pada Gambar 8.



Gambar 8. Histogram dari Citra pada Gambar 7






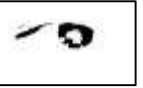
Dari Gambar 8, histogram terlihat bahwa tidak ada bagian yang terpisah secara tegas sehingga diperlukan sampel piksel dari beberapa bagian mata. Tabel 1 menunjukkan sampel nilai luminans (intensitas cahaya) pada beberapa bagian mata.

Tabel 1. Nilai Intensitas Cahaya

<i>Bagian</i>	<i>Citra</i>	<i>Nilai Luminan</i>
Kelopak Bawah		0,917
Iris (terhitam)		0,06275
Iris bagian pinggir		0,09804

Dari analisis luminans citra, nilai yang dapat memisahkan bagian iris dengan yang lain berkisar 0,0675 hingga 0,1. Selanjutnya, dilakukan variasi nilai *threshold* mulai dari 0,5; 0,4; 0,3; 0,2; 0,1; dan 0,005.

Tabel 2. Perbandingan Citra dengan Nilai Threshold Bervariasi

<i>T</i>	0,5	0,4	0,3	0,2	0,1	0,05
<i>Citra</i>						

Pada Tabel 2, nampak bahwa nilai *T* yang paling optimal adalah 0,1, dimana objek bukan iris terlihat paling sedikit, dan citra iris paling sempurna. Dengan demikian nilai *threshold* yang digunakan adalah 0,1.

c. *Image Filling*

Hasil *threshold* menunjukkan terdapat *noise* di tengah iris yang diakibatkan oleh cahaya pantulan lampu. *Noise* tersebut dapat dihilangkan dengan proses *image filling*, setelah sebelumnya dilakukan inversi citra terlebih dahulu. Sintaks pada MATLAB adalah sebagai berikut.

```
gambarinvers=~gibr;      % invers citra
gibr=imfill(in,'holes'); % image filling
```



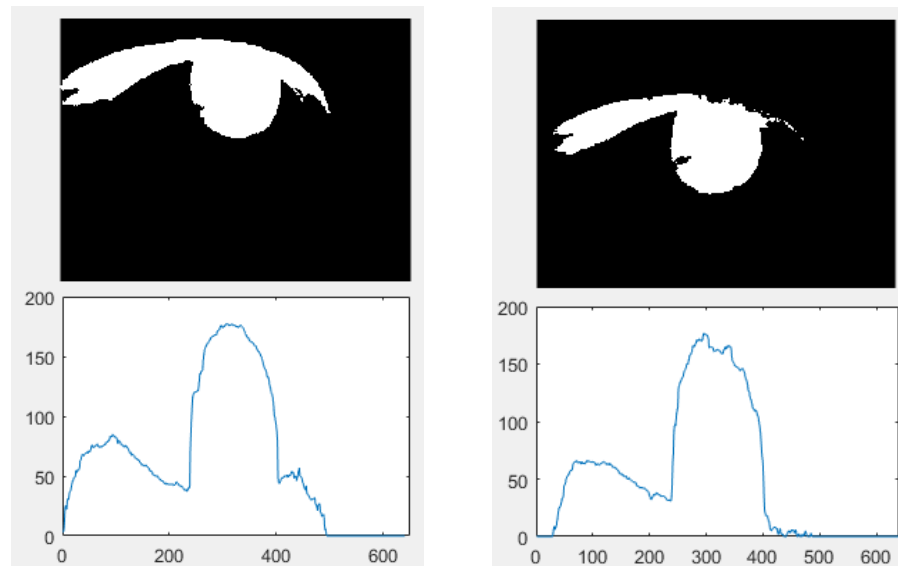
Gambar 7(a) Citra hasil *inversi* Gambar 7(b) Citra Hasil *Image Filling*

Hasil citra inversi terlihat pada Gambar 7(a), masih terdapat *noise* pada bagian tengah.

Image filling digunakan untuk menghilangkan *noise*, seperti tampak pada Gambar 7(b).

d. Erosi Citra

Setelah dilakukan *thresholding* terdapat beberapa bagian bukan iris yang masih terdeteksi sebagai objek seperti lekukan kelopak mata, untuk menghilangkan atau memutus objek menjadi iris dan kelopak mata maka dilakukan pengikisan piksel atau yang disebut erosi. Erosi memberikan efek penipisan sebuah objek pada citra biner. Pada citra biner operasi erosi akan menghapus piksel-piksel pada lapisan terluar objek sesuai dengan strel yang digunakan. Operasi erosi akan menghapus objek yang lebih kecil daripada strel yang digunakan (Kadir & Susanto, 2013).

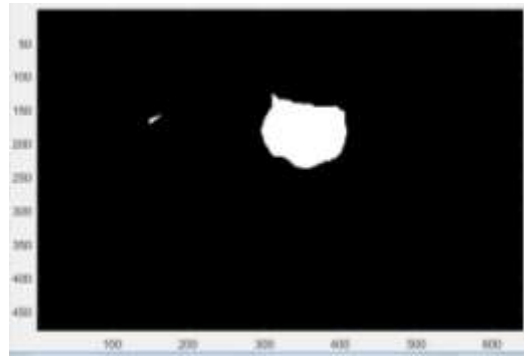


Gambar 8. Sampel Histogram 2 Citra Mata yang Berbeda

Pada Gambar 8, terlihat bahwa bagian lipatan kelopak memiliki ketebalan paling tebal sekitar 70 hingga 90 piksel dan ketebalan kelopak pada titik pertemuan dengan iris sekitar 30 sampai 40 piksel. Dengan demikian, dapat dilakukan erosi dengan memanfaatkan fungsi “*strel disk*”. Nilai strel ditetapkan pada angka 45, agar obyek citra iris tidak rusak. Citra hasil erosi ditampilkan dengan sintaks *imerode*. Berikut adalah potongan kode sumber untuk menampilkan citra tersebut.

```
...
sel=strel('disk',45);
...
e=imerode(gibr,sel);
```

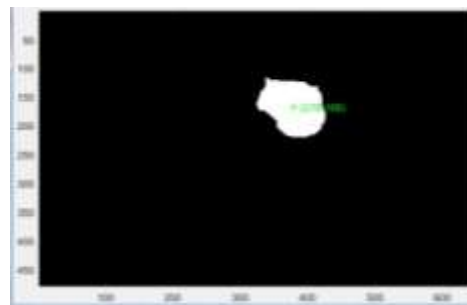
Hasil eksekusi program dengan kode tersebut, akan menampilkan citra hasil erosi seperti pada Gambar 9. Terlihat bentuk citra sudah lebih fokus, namun masih perlu dilakukan *filtering*.



Gambar 9. Citra Hasil Erosi

e. *Image Filtering*

Setelah dilakukan erosi terkadang objek alis masih muncul meski berukuran kecil, namun objek ini dapat berpengaruh pada deteksi letak *centroid* objek sehingga dapat menurunkan tingkat akurasi. Untuk itu, digunakan operasi image filtering untuk membuang obyek yang tidak diinginkan (Gonzalez & Woods, 2018). Dalam kasus ini, dilakukan dengan sintaks : `e=bwareafilt(e,1)`. Hasil eksekusi sintaks kode tersebut terlihat pada Gambar 10, nampak satu noise berupa lingkaran kecil yang ada seperti pada Gambar 9, sudah hilang.

Gambar 10. Citra Hasil *Image Filtering*

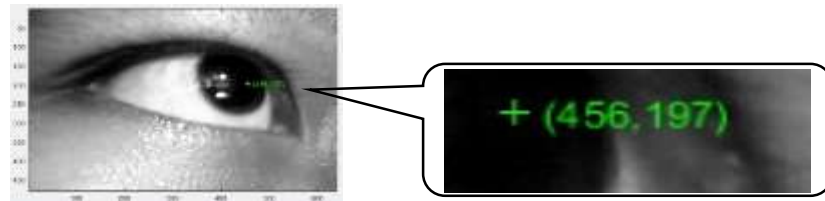
f. *Centroids Object*

Titik tengah dari obyek iris perlu dicari untuk proses klasifikasi arah mata. Koordinat centroid dapat diperoleh dengan instruksi berikut.

```
s = regionprops(e, 'centroid');
objk=size(s,1);
if objk < 1 || isempty(s)
    continue
end
centroids = cat(1, s.Centroid);
X=centroids(1);
Y=centroids(2);
```

Potongan kode sumber di atas, berfungsi untuk menentukan titik tengah *centroid*, yang berasal dari citra sebelumnya (Gambar 10). Koordinat X dan Y adalah titik tengah dari *centroid*. Dari nilai koordinat ini, didapat data koordinat mata saat melihat ke enam arah, yang kemudian digunakan untuk membuat sistem klasifikasi arah mata yang tepat.

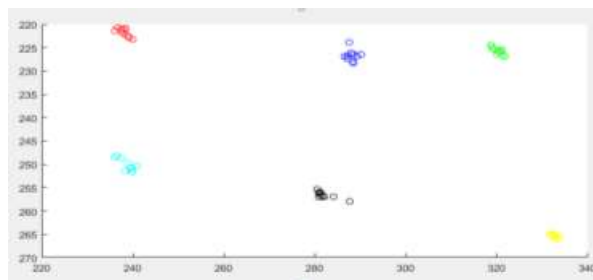
Pada Gambar 11, terlihat sampel citra saat melihat ke kanan atas. Hasil eksekusi fungsi `regionprobs` pada citra *centroid*, didapatkan nilai koordinat titik tengahnya yaitu X=456, dan Y = 197.



Gambar 11. Koordinat Centroid Mata

3. Klasifikasi Arah Mata

Klasifikasi adalah proses pengelompokan berdasarkan kesamaan sifat antar objek. Pada penelitian ini akan dilakukan klasifikasi arah mata menjadi 6 kelompok (dapat dilihat pada gambar 12) yaitu kiri atas (hijau), tengah atas (biru), kanan atas (merah), kanan bawah (sian), tengah bawah (hitam), dan kiri bawah (kuning).



Gambar 12. Plotting Koordinat Centroid Mata

Pada Gambar 12, terlihat bahwa setiap data koordinat arah mata memiliki korelasi yang jelas dengan arah yang sedang dilihat sehingga koordinat ini dapat digunakan untuk mengklasifikasi arah mata. Selain itu, terlihat pula bahwa setiap kelompok data terpisah secara linear sehingga dapat dibuat suatu garis linear pembatas untuk setiap kelompok ke kelompok yang lain.

4. Penseleksi Huruf

Dalam pemilihan huruf digunakan metode papan *eye-gaze* dimana untuk memilih huruf maka pengguna melihat kelompok huruf, kemudian menunjuk huruf yang dimaksud (Cazzato, 2018). Untuk mempermudah maka tiap arah diberi nomor index kelompok.

ABC DEF	GHI JKL	MNO PQR
STU VWX	YZ? !..	<<< > 0

Gambar 13. Eye-Gaze Board

Eye-Gaze Board tersebut dapat diinisiasikan ke variabel matriks “huruf”, dengan baris akan melambangkan nomor indeks kelompok, dan kolom melambangkan nomor indeks huruf. Untuk membuat *board* tersebut menggunakan potongan kode sumber sebagai berikut.

```
...
huruf=[ 'ABCDEF' ; 'GHIJKL' ; 'MNOPQR' ; 'STUVWX' ; 'YZ?!.,' ; ' 12345' ];
...
kata (indexhuruf)=huruf (X1, Y1)
set(handles.text1, 'string', num2str(kata));
indexhuruf=indexhuruf+1;
```

```

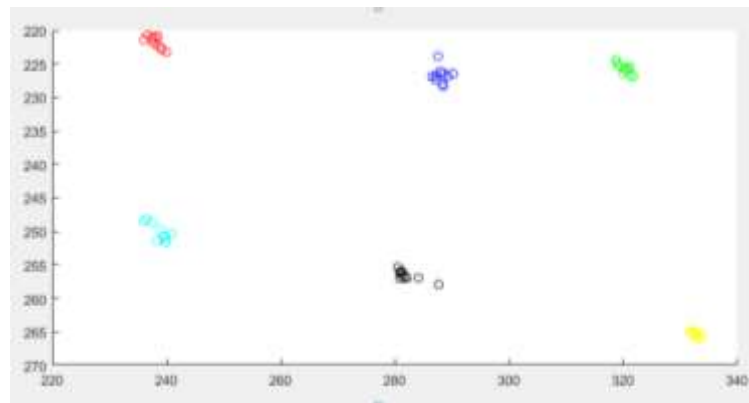
...
...
if pt==1
  set(handles.text10,'string','A');
  set(handles.text11,'string','B');
...

```

Variabel huruf merupakan matriks, digunakan untuk mengeset indeks kelompok huruf A sampai Z termasuk tanda baca dan instruksi untuk memilih huruf.

5. Klasifikasi Arah Mata

Klasifikasi arah mata menjadi 6 kelompok yaitu kiri atas, tengah atas, kanan atas, kanan bawah, tengah bawah, dan kiri bawah. Hasil eksekusi program, menghasilkan plot koordinat (x,y) dari titik tengah mata seperti ditunjukkan pada Gambar 14, Warna hijau menunjukkan koordinat mata saat melihat kiri atas layar, warna biru menunjukkan koordinat mata saat melihat tengah atas layar, warna merah menunjukkan koordinat mata saat melihat kanan atas, warna *cyan* menunjukkan koordinat mata saat melihat kanan bawah layar, warna hitam menunjukkan koordinat mata saat melihat tengah bawah layar dan warna kuning menunjukkan koordinat mata saat melihat kiri bawah



Gambar 14. Plot Koordinat Centroid

Untuk memisahkan setiap arah koordinat yang dituju, perlu dilakukan pembatasan koordinat. Pada Gambar 15, terlihat garis horizontal yang memisahkan kelompok atas dan bawah, yang diperoleh dengan menghitung jarak antara titik tengah atas dengan titik tengah bawah dalam koordinat Y, digunakan algoritma sebagai berikut.

```

TPBY=mean(b4);
TCY=mean(b1); # TCY adalah koordinat Y titik kalibrasi
batash=(TPBY-TCY)/2;

```

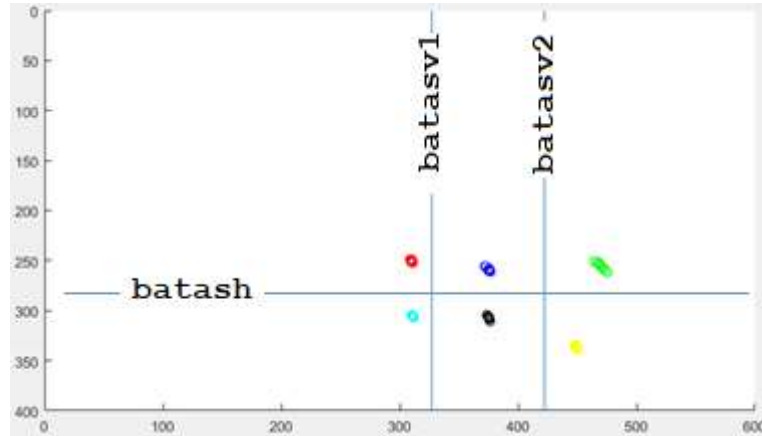
Sedangkan untuk garis pembatas vertikal yang memisahkan kelompok kiri dan tengah dan satu untuk memisahkan kelompok tengah dan kanan, digunakan algoritma sebagai berikut.

```

TPX=mean(a);
TCX=mean(a1); # TCX adalah titik tengah kalibrasi
batasv=(TPX-TCX)/1;

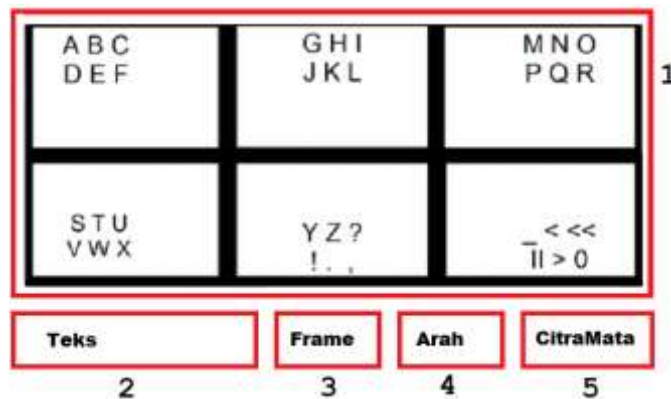
```

Lebih lanjut, untuk mendapatkan batas kiri dan tengah maka digunakan titik tengah antara titik kiri atas dan tengah atas, menggunakan sintaks `batasv1=TCX+batasv`. Sedangkan untuk mendapat batas tengah dan kanan dapat dihitung dengan selisih titik tengah atas dan kiri atas, dengan sintaks `batasv2=TCX-batasv`.



Gambar 15. Batas plot horizontal dan vertikal

6. Implementasi pada GUI MATLAB



Gambar 16. Desain GUI pada MATLAB

Gambar 16, menunjukkan desain GUI yang digunakan dalam sistem ini. Terdapat 5 bagian utama dari GUI yang dijelaskan lebih lanjut pada Tabel 3.

Tabel 3. Keterangan Desain GUI pada MATLAB

No	Keterangan
1	Layar Pemilihan huruf, di layar ini akan ditampilkan pilihan huruf yang tersedia dan menampilkan arah yang sedang dilihat melalui <i>highlight</i> kotak yang ada pada masing-masing kotak.
2	Penampil teks, di kolom ini ditampilkan susunan huruf yang membentuk kata dan kalimat yang telah disusun oleh pengguna
3	Bagian ini menampilkan jumlah <i>frame</i> yang telah diambil oleh sistem
4	Bagian ini menunjukkan arah penglihatan pengguna
5	Bagian ini menampilkan citra mata yang diambil oleh sistem

7. Pengujian Sistem

Pengujian sistem dilakukan pada 24 responden secara acak, dengan berbagai bentuk mata. Dengan demikian, dapat dilihat performa sistem yang dibuat terhadap berbagai bentuk mata pengguna. Saat pengujian, tiap pengguna melihat ke enam arah secara bergantian sehingga didapatkan hasil dari 144 arah terdapat 10 arah yang tidak sesuai dengan arah yang sedang dilihat. Hal ini dapat terjadi karena ada responden memiliki mata yang terlalu sipit atau memiliki kantung mata yang tebal.

Dari hasil pengujian, didapat tingkat kesalahan sistem mendekati 7% dengan perhitungan sebagai berikut.

$$\text{Nilai kesalahan} = \frac{\text{Jumlah Tidak Sesuai}}{\text{Jumlah Perlakuan}} \times 100\% = \frac{10}{144} \times 100\% = 6,94 \%$$

Dengan demikian perkiraan akurasi sistem adalah $100\% - 7\% = 93\%$. Sistem relatif cukup akurat sebagai sistem AAC dengan *tracking* mata untuk membantu penderita gangguan komunikasi bicara maupun tulis.

KESIMPULAN DAN SARAN

Dari hasil pembahasan pada penelitian ini, dapat diambil beberapa kesimpulan: *Centroid* dari objek iris dapat merepresentasikan arah penglihatan, Koordinat dari keenam arah penglihatan dapat terpisahkan secara linear, dan dengan kalibrasi tiga arah yaitu kiri atas, tengah atas dan tengah bawah maka dapat dibentuk garis klasifikasi enam arah. Hal tersebut dapat dilihat pada Gambar 15, yang menunjukkan posisi *centroid* mata. Namun demikian jarak pengguna dengan layar dan pencahayaan ruangan sangat mempengaruhi pembacaan sistem.

Penelitian ini masih dapat dikembangkan lebih lanjut. Beberapa saran untuk pengembangan berikutnya antara lain : perlu pengujian lebih lanjut untuk penentuan akurasi sistem, desain antarmuka dan algoritme perlu dikembangkan agar lebih portabel dan interaktif. Pada penelitian selanjutnya dapat menggunakan pupil sebagai patokan sehingga mata yang sipit dapat terbaca sistem. Metode *thresholding* adaptif dapat diterapkan pada penelitian berikutnya, karena perbedaan intensitas cahaya pada ruangan mempengaruhi hasil pembacaan. Dengan demikian diharapkan sistem ini dapat diterapkan di berbagai kondisi ruangan.

DAFTAR PUSTAKA

- ASHA. (2017). *Augmentative and Alternative Communication (AAC)*. Diambil kembali dari <http://www.asha.org/public/speech/disorders/AAC>
- Beukelman, D., & Light, J. (2020). *Augmentative & Alternative Communication: Supporting Children and Adults with Complex Communication Needs*. Baltimore: Paul H. Brookes Publishing Co.
- Dario Cazzato, F. D. (2018, February). Real-time gaze estimation via pupil center tracking. *Paladyn, Journal of Behavioral Robotics*, 9(1), 6-18. doi: <https://doi.org/10.1515/pjbr-2018-0002>
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing, 4th Edition*. New York: Pearson.
- Ji, Q., & Yang, X. (2002). Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. *Real-Time Imaging*, 8(5), 357-377. doi: <https://doi.org/10.1006/rtim.2002.0279>
- Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi Offset.
- KBBI. (2016). *Kamus Besar Bahasa Indonesia Edisi V*. Diambil kembali dari <https://kbbi.kemdikbud.go.id>
- M. Mehrubeoglu, L. M. (2011). Real-time eye tracking using a smart camera. *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 1-7. doi: <https://doi.org/10.1109/AIPR.2011.6176373>.
- Rohim, S. (2016). *Teori Komunikasi: Perspektif, Ragam, dan Aplikasi*. Jakarta: Rineka Cipta.
- S.Deivanayagi, V. S. (2018). Pupil Detection Algorithm Based on Feature Extraction for Eye Gaze. *International Journal of Recent Technology and Engineering (IJRTE)*, 73-76. doi: <https://doi.org/10.35940/ijrte.B1016.0782S519>
- S.V. Kamble, S. N. (2017, January). Eye Gaze Communication. *International Advanced Research Journal in Science, Engineering and Technology*, 4(3), 140-144. doi: <https://doi.org/10.17148/IARJSET>

- Sara Bilal, M. H. (2018, December). Design a Real-Time Eye Tracker. *ICVIP 2018: Proceedings of the 2018 the 2nd International Conference on Video and Image Processing*, 187-191. doi: <https://doi.org/10.1145/3301506.3301509>
- Sutoyo, T., Mulyanto, E., Suhartono, V., & Nurhayati, O. D. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Andi Offset.
- Vanderheiden, G. C. (2003). A Journey Through Early Augmentative Communication and Computer Access. *Journal of Rehabilitation Research and Development Vol. 39 No. 6*, 39-53.
- Vinson, B. P. (2001). *Essential for Speech Language Pathologist*. Canada: Thompson Learning.
- Wanjari, A. G., & Khode, S. S. (2014). The Eye gaze Communication System. *International Journal of Research Studies in Science, Engineering and Technology [IJRSSET]*, 1(1), 4-9.
- .