

Kompleksitas Algoritma GLCM untuk Ekstraksi Ciri Tekstur pada Penyakit Glaucoma

Anindita Septiarini^{#1}, Retantyo Wardoyo^{*2}

[#]*Program Studi Ilmu Komputer*
Fakultas MIPA Universitas Mulawarman, Samarinda

¹*anindita.septiarini@gmail.com*

^{*}*Program Studi Ilmu Komputer*
Fakultas MIPA Universitas Gadjah Mada, Yogyakarta

Program Pascasarjana Universitas Gadjah Mada

²*rrw@ugm.ac.id*

Abstract — Complexity is part of a computer science related to the use of the algorithm in a particular application. The complexity of the algorithm can be time complexity, where it is measured by the amount of computation performed stages. This study discusses about the statistically characteristic complexity algorithm of Gray Level Co-Occurrence Matrix (GLCM) method which can be used to extract the characteristics of the disease glaucoma. In this algorithm there are basic operations such as measuring direction and statistical characteristics, where the amount of direction and statistical characteristics applied to this algorithm could be different. It depend on the problems. Such as directions and the statistical characteristics are not used all. The results of the calculation complexity statistical characteristics is $O(n^2)$. GLCM texture feature extraction algorithm can be applied to problems with the data input of fundus images, such as for extracting PPA characteristic of the disease glaucoma.

Keywords— algorithm complexity, feature extraction, GLCM, retinal fundus images.

I. PENDAHULUAN

Kompleksitas merupakan bagian dari ilmu komputer yang berhubungan dengan penggunaan algoritma pada suatu aplikasi tertentu. Kompleksitas dari algoritma dibedakan menjadi dua, yaitu kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari sejumlah data n (ukuran masukan). Sedangkan untuk kompleksitas ruang diukur dari memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari sejumlah data n .

Pada aplikasi yang berhubungan dengan pengolahan citra terdapat beberapa tahapan yang membutuhkan penerapan suatu algoritma. Salah satu tahapan yang dilakukan pada pengolahan citra adalah tahap ekstraksi ciri, dimana jenis ekstraksi ciri dapat berupa ekstraksi ciri warna, bentuk dan tekstur. Penelitian ini membahas mengenai ekstraksi ciri

tekstur yang telah digunakan pada beberapa penelitian sebelumnya untuk pendeteksian penyakit glaucoma dengan data masukan berupa citra fundus. Penelitian yang berhubungan dengan pendeteksian glaucoma dikembangkan karena glaucoma merupakan salah satu penyakit mata penyebab terjadinya kebutaan terbesar kedua di dunia [1].

Terdapat beberapa algoritma ekstraksi ciri tekstur untuk pendeteksian glaucoma pada citra fundus diantaranya *gabor filter*, *fractal dimension*, *Discrete Wavelet Transform (DWT)* dan *Gray Level Co-occurrence Matrix (GLCM)*. Pada penelitian [2] melakukan penggabungan ekstraksi ciri warna, bentuk dan tekstur. Untuk ekstraksi ciri tekstur dilakukan dengan menerapkan algoritma *Gabor Filter* dengan hasil akurasi mencapai 86%. Algoritma *fractal dimension* diterapkan pada penelitian([3], [4]) dengan hasil akurasi secara berurutan mencapai 93% dan 95%. Penerapan algoritma DWT dapat mencapai hingga akurasi 95% ([5], [6]). Pada [7] menerapkan algoritma GLCM dengan akurasi mencapai 95%. Algoritma GLCM juga bisa digunakan untuk mengekstraksi salag satu ciri dari penyakit glaucoma yaitu PPA dengan akurasi mencapai 92.5% [8].

Penelitian ini membahas mengenai kompleksitas algoritma GLCM yang digunakan untuk ekstraksi ciri tekstur GLCM. Algoritma tersebut dipilih karena dapat menghasilkan akurasi hingga 95%. Tujuan dari penelitian ini adalah melakukan perhitungan kompleksitas dari algoritma GLCM. Pada bagian selanjutnya (2) menjelaskan mengenai kompleksitas, glaucoma dan citra fundus dijelaskan pada bagian (3). Selanjutnya ekstraksi ciri, kompleksitas algoritma GLCM, kesimpulan secara berurutan dijelaskan pada bagian (4), (5) dan (6).

II. TINJAUAN TEORI

A. Kompleksitas

Algoritma yang baik adalah algoritma yang mangkus (*efficient*). Kemangkusan algoritma diukur dari waktu (*time*) eksekusi algoritma dan kebutuhan ruang (*space*) memori.

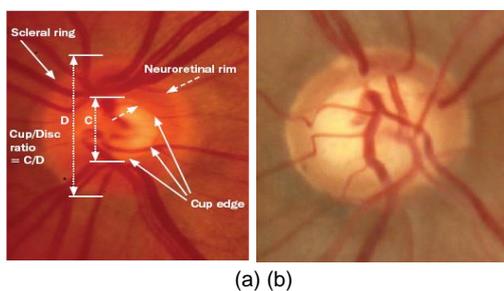
Algoritma yang mangkus ialah algoritma yang meminimumkan kebutuhan waktu dan ruang. Kebutuhan waktu dan ruang suatu algoritma bergantung pada ukuran masukan (n), yang menyatakan jumlah data yang diproses. Kemangkusan algoritma dapat digunakan untuk menilai algoritma yang baik dari sejumlah algoritma penyelesaian masalah. Besaran yang dipakai untuk menerangkan model abstrak pengukuran waktu/ruang ini adalah kompleksitas algoritma [9].

Ada dua macam kompleksitas algoritma, yaitu kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu, $T(n)$, diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n .

- Kompleksitas ruang, $S(n)$, diukur dari memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran masukan n .
- Kompleksitas waktu dihitung berdasarkan jumlah tahapan komputasi operasi yang dilakukan dalam sebuah algoritma. Dalam prakteknya, operasi yang dihitung hanya operasi khas yang mendasari suatu algoritma. Misalnya, operasi khas di dalam algoritma pencarian di dalam larik adalah perbandingan elemen larik. Operasi khas algoritma pengurutan adalah perbandingan dan pertukaran elemen.

B. Glaucoma

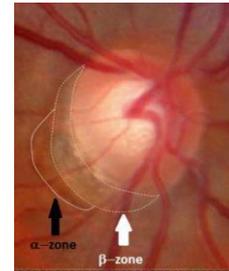
Glaucoma merupakan salah satu penyakit mata penyebab terjadinya kebutaan terbesar kedua di dunia [10]. Pendeteksian glaucoma merupakan suatu proses untuk mengetahui kondisi mata tergolong normal atau glaucoma. Pendeteksian dibutuhkan karena gejala tahap awal dari penyakit ini tidak dirasakan oleh penderitanya dan proses perkembangan penyakit lambat. Hal tersebut mengakibatkan jika penyakit tersebut terdeteksi kerusakan yang terjadi pada *Optic Nerve Head* (ONH) sudah terlanjur parah. ONH merupakan bagian dari retina mata yang dapat membedakan antara mata normal dan glaucoma berdasarkan pengamatan pada citra fundus. Contoh citra fundus ONH yang tergolong normal atau glaucoma dapat dilihat pada Gambar 1 [11].



Gambar 1. ONH pada Mata (a) Normal dan (b) Glaucoma

Pada Gambar 1 terdapat dua bagian berbentuk bulatan, dimana bulatan dengan ukuran yang lebih besar disebut *disc* dan bulatan dengan ukuran yang lebih kecil disebut *cup*.

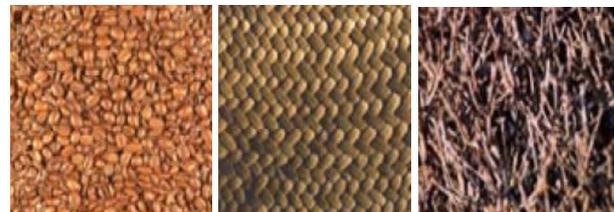
Untuk area yang terletak antara *disc* dan *cup* disebut *neuroretinal rim*. Rasio *cup/disc* (C/D) merupakan perbandingan diameter *cup* dan *disc*. Salah satu ciri yang dapat membedakan antara mata normal dan glaucoma adalah terjadinya pembesaran ukuran *cup* sehingga dapat mempengaruhi area *neuroretinal rim* dan nilai rasio C/D . Ciri lain yang dapat membedakan antara mata normal dan glaucoma adalah munculnya area PPA pada mata penderita glaucoma [8]. Adapun ciri PPA yang dimaksud dapat dilihat pada Gambar 2 [8].



Gambar 2. Area PPA pada Mata Glaucoma

C. Ekstraksi Ciri Tekstur

Ekstraksi ciri tekstur merupakan salah satu tahapan proses pada pengenalan kelas (objek) untuk menghasilkan ciri pada setiap citra berdasarkan teksturnya. Proses ini bertujuan untuk memberikan informasi yang detail mengenai perbedaan antar kelas dari beberapa citra, sehingga antara citra yang satu dan yang lain bisa dimungkinkan berada dalam kelas yang berbeda [12].



Gambar 3. Contoh citra tekstur dari *VisTex Database*

Pada proses ini objek di dalam citra dihitung properti-properti objek yang berkaitan sebagai ciri. Ciri tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah (di dalam citra) yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Pengertian dari tekstur dalam hal ini kurang lebih adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Adapun contoh citra tekstur dapat dilihat pada Gambar 3 [12].

Untuk membentuk suatu tekstur setidaknya ada dua persyaratan yang harus dipenuhi antara lain [12]:

1. Terdiri dari satu atau lebih piksel yang membentuk pola-pola primitif (bagian-bagian terkecil). Bentuk-bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.

- Munculnya pola-pola primitif yang berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.

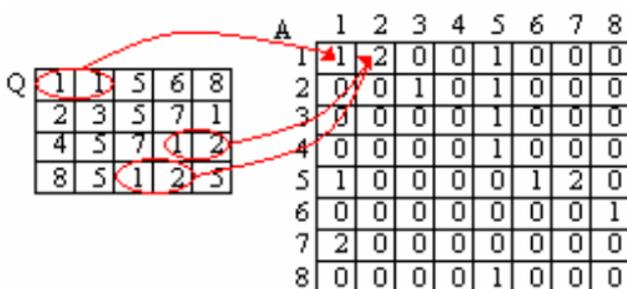
D. Gray Level Co-Occurrence Matrix (GLCM)

Co-occurrence matrix menggunakan matriks derajat keabuan adalah untuk mengambil contoh bagaimana suatu derajat keabuan tertentu terjadi dalam hubungannya dengan derajat keabuan yang lain. Matriks derajat keabuan adalah suatu matriks yang elemen-elemennya merupakan frekuensi relatif kejadian bersama dari kombinasi level keabuan antar pasangan piksel dengan hubungan spasial tertentu [13].

Misal diketahui sebuah citra $Q(i,j)$, $p(i,j)$ merupakan posisi dari operator dan A adalah sebuah matriks $N \times N$. Elemen $A(i,j)$ menyatakan jumlah titik tersebut terjadi dengan grey level (intensitas) $g(i)$ terjadi, pada posisi tertentu menggunakan operator p , relatif terhadap titik dengan intensitas $g(j)$. Matriks A merupakan co-occurrence (GLCM) matrix yang didefinisikan oleh p . Operator p didefinisikan dengan terlebih dahulu menentukan sebuah sudut θ ($0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$ dan 315°) dan jarak d . Berdasarkan matriks A dapat dihitung nilai-nilai ciri tekstur. Langkah-langkah dalam membentuk GLCM adalah sebagai berikut [14]:

- Konversi nilai keabuan citra asal dari range 0-255 ke dalam skala keabuan baru.
- Pembentukan matriks GLCM
- Hitung probabilitas setiap elemen GLCM
- Hitung semua ciri statistik untuk setiap arah yang terbentuk

Pada Gambar 4 menunjukkan ilustrasi pembentukan matriks GLCM dengan tingkat keabuan = 8, nilai $d = 1$ dan sudut $\theta = 0^\circ$ [14].



Gambar 4. Representasi Co-occurrence Matrix

Ciri statistik GLCM adalah sebagai berikut [14]:

- Maximum probability (MP)

MP adalah nilai maksimum atau respon terkuat dari co-occurrence matrix. Nilai MP menyatakan tingkat keteraturan dari suatu citra. Semakin tinggi nilai MP, maka semakin teratur teksturnya. Nilai MP dapat dicari menggunakan persamaan 1.

$$MP = \max(P(i, j)) \tag{1}$$
- Entropy

Mengukur tingkat keacakan piksel pada citra. Entropy mencapai nilai tertinggi jika semua elemen dalam matrix P sama. Semakin tinggi nilai entropy, maka semakin acak teksturnya. Nilai entropy dapat dicari menggunakan persamaan 2.

$$E_2 = -\sum_{i,j} P(i, j) \log P(i, j) \tag{2}$$

- Energy

Energy mengukur tingkat keseragaman piksel-piksel pada suatu citra. Energi mencapai nilai tertinggi saat persebaran level keabuan konstan atau bersifat periodik. Semakin tinggi nilai energy, maka semakin seragam teksturnya. Nilai energy dapat dicari menggunakan persamaan 3.

$$E_1 = \sum_{i,j} P(i, j)^2 \tag{3}$$

- Correlation

Correlation menyatakan ukuran hubungan linear dari nilai graylevel piksel ketetanggaan. Nilai correlation dapat dicari menggunakan persamaan 4.

$$C_2 = \sum_{i,j} \frac{(1 - \mu_i)(j - \mu_j)P(i, j)}{\sigma_i \sigma_j} \tag{4}$$

- Contrast

Contrast menyatakan kandungan variasi lokal pada citra. Semakin tinggi nilai contrast maka semakin tinggi tingkat kekontrasannya. Nilai Contrast dapat dicari menggunakan persamaan 5.

$$C_1 = \sum_{i,j} |i - j|^2 P(i, j) \tag{5}$$

- Homogeneity

Homogeneity menyatakan ukuran kedekatan setiap elemen dari co-occurrence matrix (kehomogenan piksel). Nilainya dapat dicari menggunakan persamaan 6.

$$H = \sum_{i,j} \frac{P(i, j)}{1 + |i - j|} \tag{6}$$

III. KOMPLEKSITAS ALGORITMA GLCM

Pada bagian ini akan melakukan perhitungan kompleksitas operasi dasar dan ciri statistik dari algoritma GLCM. Perhitungan kompleksitas algoritma ini dilakukan dengan menghitung langkah-langkah yang ada pada operasi dasar dari algoritma tersebut. Adapun potongan program dari operasi dasar yang dilakukan pada algoritma GLCM adalah sebagai berikut:

- Konversi nilai keabuan citra asal dari range 0-255 ke dalam skala keabuan baru.

```
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    Q[i,j] = (R_asli[i,j] + G_asli[i,j] +
             B_asli[i,j])/3
```

Analisa kompleksitas pada tahap ini adalah sebagai berikut:



- Baris ke-3 $\rightarrow O(1)$
- Baris ke-2 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-3 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(n^2)$

2. Perhitungan probabilitas setiap elemen GLCM.

```
total=0;
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    total = total+A[i,j];
```

Analisa kompleksitas pada tahap ini adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(1)+O(n^2) = O(n^2)$

```
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    A_prob[i,j] = A[i,j]/total;
```

Analisa kompleksitas pada tahap ini adalah sebagai berikut:

- Baris ke-3 $\rightarrow O(1)$
- Baris ke-2 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-1 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(n^2)$

3. Hitung ciri statistik untuk setiap arah yang terbentuk

a. Maximum probability (MP)

```
max=0;
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    if (max<A[i,j])
      max= A[i,j];
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (1) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-5 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1)$. $O(1) = O(n)$. $O(1) \cdot O(1) = O(n \cdot 1 \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(n^2)$

b. Entropy

```
jml=0;
nilai=0;
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    if (A[i,j]==0)
      nilai= A[i,j];
    else
      nilai=A[i,j]*Math.Log(A[i,j],2);
    jml=jml+nilai;
jml=jml*(-1);
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (2) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-2 $\rightarrow O(1)$

- Baris ke-6 $\rightarrow O(1)$
- Baris ke-8 $\rightarrow O(1)$
- Baris ke-5 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow n \cdot O(1)$. $O(1) = O(n)$. $O(1) \cdot O(1) = O(n \cdot 1 \cdot 1) = O(n)$
- Baris ke-3 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$
- Baris ke-10 $\rightarrow O(1)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(n^2)+O(1) = O(n^2)$

c. Energy

```
jml=0;
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
    jml=jml+Math.Pow(A[i,j],2);
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (3) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(1)+O(n^2) = O(n^2)$

d. Correlation

```
jml=0;
for (i=0; i<n; i++)
  for (j=0; j<m; j++)
    jml=jml+((i-Mi)+(j-Mj))*A[i,j]/(SDi*SDj);
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (4) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(1)+O(n^2) = O(n^2)$

e. Contrast

```
jml=0;
for (i=0; i<n; i++)
  for (j=0; j<m; j++)
    jml=jml+Math.Pow(i-j, 2) *A[i,j];
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (5) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(1)+O(n^2) = O(n^2)$

f. Homogeneity

```
jml=0;
for (i=0; i<n; i++)
  for (j=0; j<m; j++)
```



```
jml=jml+(A[i,j]/1+Math.Abs(i-j));
```

Analisa kompleksitas pada tahap ini dilakukan berdasarkan persamaan (6) adalah sebagai berikut:

- Baris ke-1 $\rightarrow O(1)$
- Baris ke-4 $\rightarrow O(1)$
- Baris ke-3 $\rightarrow n \cdot O(1) = O(n)$. $O(1) = O(n \cdot 1) = O(n)$
- Baris ke-2 $\rightarrow n \cdot O(n) = O(n)$. $O(n) = O(n \cdot n) = O(n^2)$

Jadi kompleksitas algoritma pada tahap ini adalah $O(1)+O(n^2) = O(n^2)$

Berdasarkan hasil perhitungan kompleksitas pada beberapa tahapan algoritma GLCM, kompleksitas dari algoritma GLCM adalah $O(n^2)$.

IV. SIMPULAN

Pada penelitian ini membahas mengenai kompleksitas dari algoritma ekstraksi ciri tekstur yaitu ciri statistik GLCM. Hasil dari perhitungan kompleksitas dari beberapa tahapan dari algoritma perhitungan ciri statistik GLCM adalah $O(n^2)$. Algoritma GLCM dapat dikembangkan untuk menghasilkan ciri-ciri khusus pada suatu citra. Jumlah sudut dan ciri statistik yang digunakan tergantung dari permasalahan yang akan diselesaikan. Berdasarkan penelitian sebelumnya algoritma ini dapat diterapkan pada permasalahan dengan data masukan berupa citra fundus, salah satunya pada permasalahan untuk mengekstraksi ciri PPA pada penyakit glaucoma.

DAFTAR PUSTAKA

- [1] H. A. Quigley, "Number of people with glaucoma worldwide," pp. 389–393, 1996.
- [2] R. Bock, J. Meier, G. Michelson, L. Nyul, and J. Hornegger, "Classifying Glaucoma with Image-Based Features from Fundus Photographs," *Springer-Verlag*, pp. 355–364, 2007.
- [3] R. Kolar and J. Jan, "Detection of Glaucomatous Eye via Color Fundus Images Using Fractal Dimensions," *RADIOENGINEERING*, vol. 17, no. 3, pp. 109–114, 2008.
- [4] D. Lamani, Ramegowda, and T. Manjunath, "Fractal Dimension as Diagnostic Parameter to Detect Glaucoma," *Int. J. Innov. Eng. Technol.*, vol. 2, no. 1, pp. 63–69, 2013.
- [5] M. R. K. Mookiah, U. R. Acharya, C. M. Lim, A. Petznick, and J. S. Suri, "Data Mining Technique for Automated Diagnosis of Glaucoma using Higher Order Spectra and Wavelet Energy Features," *Knowledge-Based Syst.*, vol. 33, pp. 73–82, 2012.
- [6] N. Annu and J. Justin, "Classification of Glaucoma Images using Wavelet based Energy Features and PCA," *Int. J. Sci. Eng. Res.*, vol. 4, no. 5, pp. 1369–1374, 2013.
- [7] S. Karthikeyan and N. Rengarajan, "Performance Analysis of Gray Level Co- Occurrence Matrix Texture Features for Glaucoma Diagnosis," *Am. J. Appl. Sci.*, vol. 11, no. 2, pp. 248–257, 2014.
- [8] M. S. Haleem, L. Han, J. Van Hemert, and B. Li, "Automatic Extraction of Retinal Features from Color Retinal Images for Glaucoma Diagnosis: A Review," *Comput. Med. Imaging Graph.*, 2013.
- [9] R. Munir, *Kompleksitas Algoritma*. Bandung, 2009.
- [10] H. A. Quigley, "Number of people with glaucoma worldwide," pp. 389–393, 1996.
- [11] R. R. A. Bourne, "GLOSSARY The optic nerve head in glaucoma," *COMMUNITY EYE Heal.*, vol. 19, no. 59, pp. 44–45, 2006.
- [12] A. J. Arriawati, I. Santoso, and Y. Christyono, "KLASIFIKASI CITRA TEKSTUR MENGGUNAKAN k-NEAREST NEIGHBOUR BERDASARKAN EKSTRAKSI CIRI METODE MATRIKS KOOKURENSI Asri," Universitas Diponegoro.
- [13] M. E. Osadebey, "Integrated Content-Based Image Retrieval Using Texture, Shape and Spasial Information," Umea University, 2006.
- [14] R. Y. Dillak, M. G. Bintiri, and A. Harjoko, *Retinopati Diabetes Sistem Deteksi Penyakit Menggunakan Jaringan Syaraf Tiruan*. Graha Ilmu, 2012.