



## Segmentasi Citra Sel Sabit dengan Menggunakan Algoritma K-Nearest Neighbor Untuk Deteksi Penyakit Anemia

Katarina Febrianti<sup>1</sup>, Rizki Dian Rahayani ,S.T., M.T<sup>2</sup> dan Wahyuni Khabzli,S.T., M.T<sup>3</sup>

<sup>1</sup>Politeknik Caltex Riau, email: katarinafebrianti23@gmail.com

<sup>2</sup>Politeknik Caltex Riau, email: uki@pcr.ac.id

<sup>3</sup>Politeknik Caltex Riau, email: ayu@pcr.ac.id

### Abstrak

Anemia sel sabit (*Sickle cell Anemia*) merupakan penyakit kekurangan sel darah merah normal yang disebabkan oleh kelainan genetik pada tubuh manusia dimana sel-sel darah merah berbentuk sabit. Pada penelitian ini mencoba melakukan analisis otomatis berdasarkan citra sel darah masukan, yang kemudian akan di bedakan antara sel sehat dan sel sabit. Penelitian ini terdiri dari dua tahapan. Tahapan pertama adalah pengolahan citra (grayscale, thresholding, resize image) yang akan digunakan sebagai masukan untuk tahapan kedua. Tahapan kedua adalah deteksi anemia sel sabit untuk mengidentifikasi anemia sel sabit, menggunakan algoritma *k-nearest neighbor*. Proses identifikasi terdiri dari tahapan pengolahan citra (grayscale, thresholding), pengukuran sel darah untuk mendapatkan data numerik sel yang akan digunakan sebagai masukan *k-nearest neighbor*, untuk kemudian dibedakan apakah sel darah masukan termasuk sel sehat atau sel sabit. Setelah melakukan 50 pengambilan data dengan nilai K yang bervariasi (  $k=3$ ,  $k=5$ ,  $k=7$ ,  $k=9$ ,  $k=12$ ) dapat disimpulkan bahwa K terbaik untuk penelitian ini adalah 3 dengan nilai akurasi 80%.

**Kata kunci** : Anemia sel sabit, pengolahan citra digital, algoritma *k-nearest neighbor* (k-NN atau KNN).

### Abstract

*Sickle cell anemia (Sickle cell anemia) is a deficiency disease normal red blood cells caused by a genetic abnormality in the human body where blood cells are sickle-shaped red. This research is trying to perform automated analysis based on the input image of blood cells, which then differentiate between healthy cells and sickle cells. The study consisted of two phases. The first stage is the image processing (grayscale, thresholding, resize image) which will be used as input to the second stage. The second stage is the detection of sickle cell anemia to identify sickle cell anemia, using k - nearest neighbor algorithm. The identification process consists of the stages of image processing (grayscale, thresholding), measurement of blood cells to get the data numerik cells that will be used as input k - nearest neighbor , to then distinguish whether the blood cells of inputs including healthy cells or sickle cell. After doing 50 to collect data by varying the value of K (  $k = 3$ ,  $k = 5$ ,  $k = 7$ ,  $k = 9$ ,  $k = 12$ ) can be concluded that the best K for the study was 3 with a value of 80% accuracy*

**keywords** : *Sickle cell anemia, digital image processing, algorithms k-nearest neighbor (k-NN or KNN).*

## 1. Pendahuluan

### 1.1 Latar Belakang

Anemia sel sabit (*Sickle cell Anemia*) merupakan penyakit kekurangan sel darah merah normal yang disebabkan oleh kelainan genetik pada tubuh manusia dimana sel-sel darah merah berbentuk sabit. Sulit bagi sel darah merah berbentuk sabit untuk melewati pembuluh darah, terutama di bagian pembuluh darah yang menyempit atau pada persimpangan pembuluh darah. Perkembangan teknologi yang pesat telah memberikan banyak manfaat dalam kemajuan diberbagai macam aspek, salah satunya adalah aspek kesehatan. Sistem identifikasi manual dengan waktu yang lama menjadi alasan untuk membuat penelitian yang dapat mempercepat proses identifikasi. Penelitian ini dirancang ini mengidentifikasi citra darah masukan yang terlebih dahulu telah mengalami proses pengolahan citra (*grayscale, thresholding*) untuk mendapatkan data numberik citra, setelah itu citra masukan akan diidentifikasi menggunakan algoritma K-NN untuk mengetahui apakah citra darah masukan tersebut termasuk dalam sel sehat atau sel sabit. Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean.

## 2. Tinjauan Pustaka

### 2.1 Studi Literatur

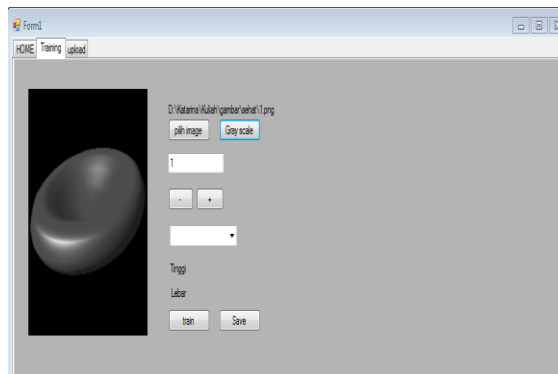
Penelitian terdahulu yang dilakukan oleh Elvia Budianita<sup>1</sup>, Jasril<sup>2</sup>, Lestari Handayani<sup>3</sup> pada penelitian ini membahas tentang Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi. Pada penelitian ini akan dibuat suatu sistem pengolahan citra untuk membedakan daging sapi dan babi menggunakan metode HSV, GLCM, dan klasifikasi K-Nearest Neighbour (K-NN). Tujuannya adalah untuk mengetahui hasil akurasi metode tersebut dalam klasifikasi citra daging sapi dan babi.

Penelitian selanjutnya dilakukan oleh Sufiatul Maryana, Lita Karlitasari, Arie Qur'ania, pada penelitian ini Tujuan dari penelitian ini adalah *k-nearest neighbor* KNN pada pengenalan wajah dengan praproses transformasi wavelet. Data yang digunakan berupa gambar atau citra yang diperoleh dari beberapa objek. Pada percobaan ini terdapat 80 citra dari 10 objek. Sebanyak 60 citra digunakan sebagai database pelatihan dan 16 digunakan untuk uji pengenalan pola. Metode yang digunakan adalah transformasi wavelet untuk ekstraksi fitur dan KNN untuk identifikasi.

Pada penelitian ini penulis menggunakan sistem identifikasi anemia sel sabit dengan menggunakan citra darah sebagai masukan, kemudian akan diolah dengan menggunakan beberapa proses pengolahan citra digital seperti grayscale, thresholding untuk mendapatkan data numberik dari citra tersebut. Setelah mengalami proses pengolahan citra, maka citra masukan akan diproses dengan Algoritma *k-nearest neighbor* (k-NN atau KNN) untuk mengetahui apakah citra tersebut termasuk sel sehat atau sel sabit.

### 2.2 Grayscale

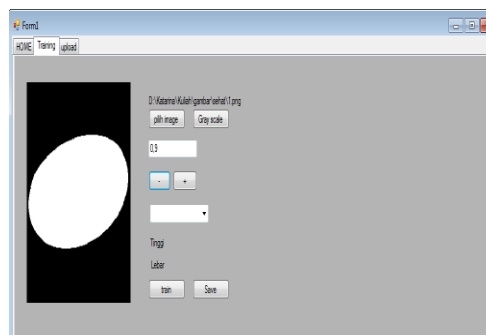
Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu. Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra beraras keabuan hanya perlu menyatakan nilai intensitas untuk tiap piksel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pikselnya. Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra gray-scale, hal ini digunakan untuk menyederhanakan model citra.



Gambar 2.2 proses grayscale

### 2.3 Thresholding

*Thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra *biner* atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra. Setelah dilakukan proses *Thresholding* maka citra sel darah berwarna hitam dan citra latar berwarna putih, kemudian citra *biner* tersebut diinvers untuk menghasilkan citra sel darah berwarna putih dan citra latar berwarna hitam. Proses *thresholding* sering disebut dengan proses binerisasi.



Gambar 2.3 proses thresholding.

### 2.2 Resize Image

Mengubah ukuran horizontal dan vertikal suatu citra. Pada tahap ini dilakukan perubahan dan menetapkan satu ukuran resolusi citra. Resolusi citra biasanya dinyatakan dalam satuan piksel. Cara kerja ini merubah ukuran sebenarnya menjadi satu ukuran yang sudah di tetapkan pada sistem. Ada kalanya ukuran citra berubah menjadi lebih kecil dari file aslinya, namun bisa juga terjadi yang sebaliknya.

### 2.3 Algoritma *k-nearest neighbor* (k-NN atau KNN)

Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Tiap data *learning* merepresentasikan sebuah titik, yang ditandai dengan  $c$ , dalam ruang  $n$ -dimensi. Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *K-Nearest Neighbor* akan mencari  $k$  buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang  $n$ -dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Euclidean Distance*. Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data, secara umumnya, nilai  $k$  yang tinggi akan mengurangi efek noise pada klasifikasi.

2.5 Flowchat algoritma  $K$ -NN

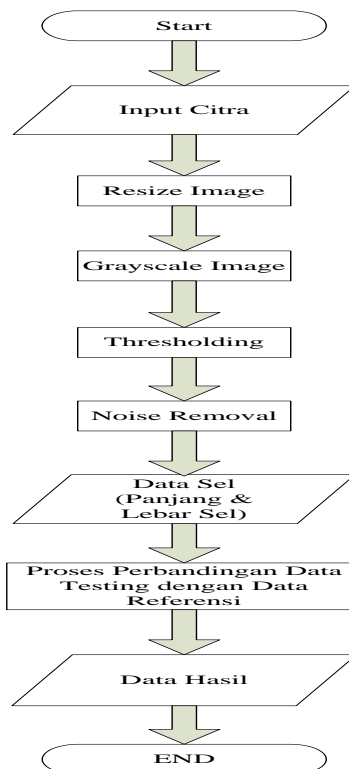
### 3. Perancangan

#### 3.1 Langkah-langkah Perancangan

Agar Tugas Akhir ini mencapai tujuannya dengan lancar, tentunya perlu dibuat *timeline* dan langkah pengerjaan yang sistematis dan terintegrasi satu dengan yang lain sehingga Tugas Akhir ini selesai dengan hasil yang memuaskan.

##### 3.1.1 *Flowchart sistem*

Berikut merupakan alur pengiriman data pada Tugas Akhir ini, agar pengerjaan Tugas Akhir ini dapat berjalan secara teratur.

Gambar 3.1 *Flowchart* sistem secara keseluruhan.

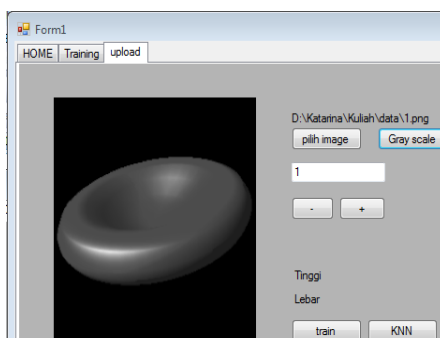
Pada gambar 3.1 dapat dijelaskan bahwa Citra yang akan diidentifikasi di masukkan (citra dalam format .jpeg dan .jpg) , kemudian citra yang telah di inputkan akan dilanjutkan pada tahap *resize image*, Cara kerja dari *resize image* ini merubah ukuran sebenarnya menjadi satu ukuran yang sudah di tetapkan pada sistem. Pada *resize image* ini dilakukan pengubahan ukuran citra menjadi 100X120 piksel, selanjutnya konversi citra asli menjadi citra *grayscale*, hal ini dilakukan untuk mengubah bobot atau kekuatan. Kemudian akan dilanjutkan pada tahap *Thresholding*, tahap ini berfungsi untuk memisahkan objek yang ingin di identifikasi dengan background nya. Setelah citra masukan mengalami beberapa proses *image processing*, maka citra akan di simpan dalam database. Data testing yang di inputkan akan di bandingkan dengan database yang telah disimpan sebelumnya dengan menggunakan Algoritma *k-nearest neighbor*. Hasil klasifikasi oleh algoritma *k-nearest neighbor* akan menunjukkan apakah data testing termasuk kedalam sel sehat atau sel sabit.

#### 4. Pengujian

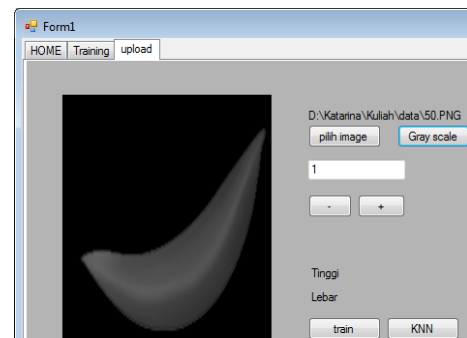
Setelah perancangan dan pembuatan sistem maka langkah selanjutnya adalah melakukan pengujian dan analisa terhadap sistem yang telah dibuat. Pengujian ini bertujuan untuk mengetahui apakah sistem yang telah direalisasikan dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pengujian ini akan dibagi kedalam 2 tahap. Tahap pertama adalah pengujian pada pengolahan citra, dan tahap kedua adalah pengujian pada Algoritma *k-nearest neighbor* (k-NN atau KNN).

##### 4.1 Pengujian Pengolahan Citra

###### A. Proses Grayscale

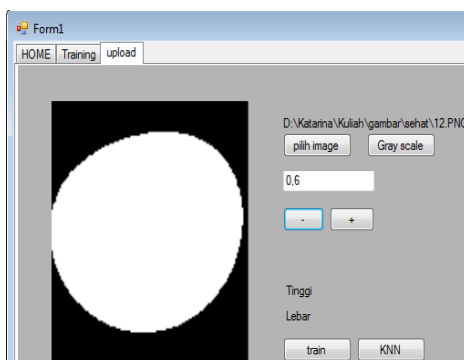


Gambar 4.1 (a) grayscale pada sel sehat

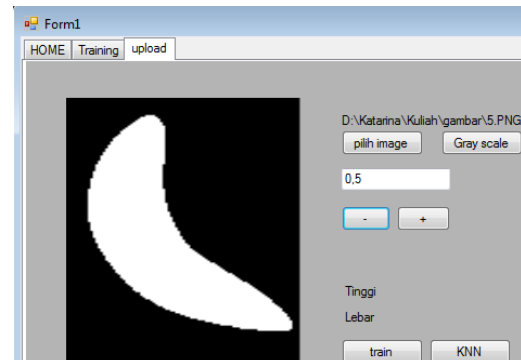


Gambar 4.2 (b) grayscale pada sel sabit

###### B. Thresholding

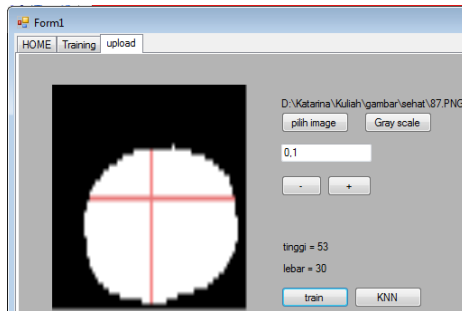


Gambar 4.1 (c) thresholding pada sel sehat

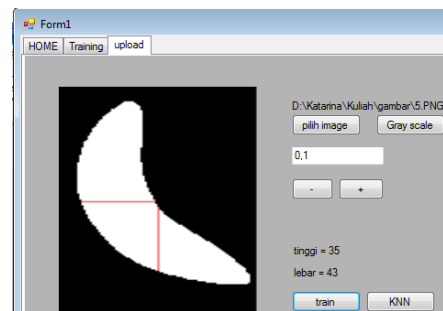


Gambar 4.1 (c) thresholding sel sabit

C. Pengukuran Sel

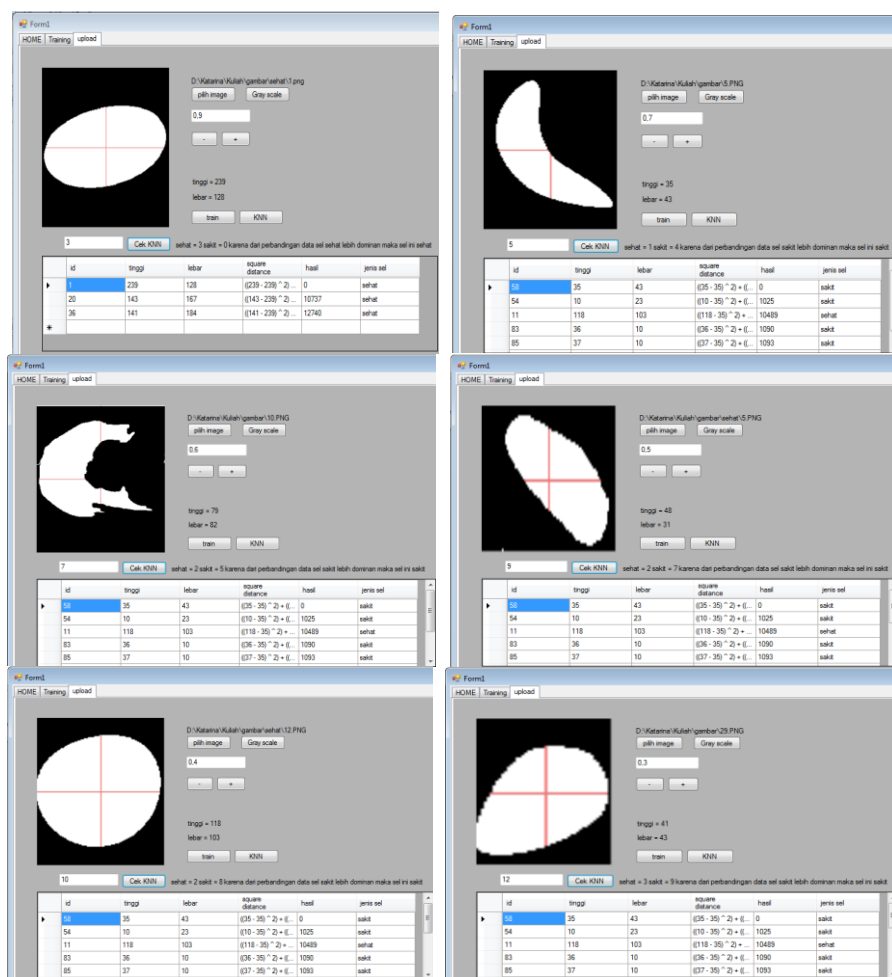


Gambar 4.1 (d) sel pengukuran sehat



Gambar 4.1(d) pengukuran sel sabit

4.2 Pengujian Algoritma *k-nearest neighbor* (k-NN atau KNN)



Gambar 4.1 Data Pengujian Algoritma k-nearest neighbor (k-nn atau KNN)

Dari 10 Data testing yang di uji dengan nilai K yang bervariasi (K=3, K=5, K=7, K=9, K=12). Dapat dilihat bahwa ketika menggunakan nilai K=3 persentase error yang diperoleh lebih kecil, yaitu 20% dengan tingkat akurasi tertinggi yaitu 80%. Nilai K yang digunakan akan mempengaruhi jumlah tetangga terdekat dengan data testing. Dari hasil yang diperoleh dapat disimpulkan bahwa nilai K yang cocok untuk proses identifikasi ini adalah K=3.

### 4.3 Tabel Data testing

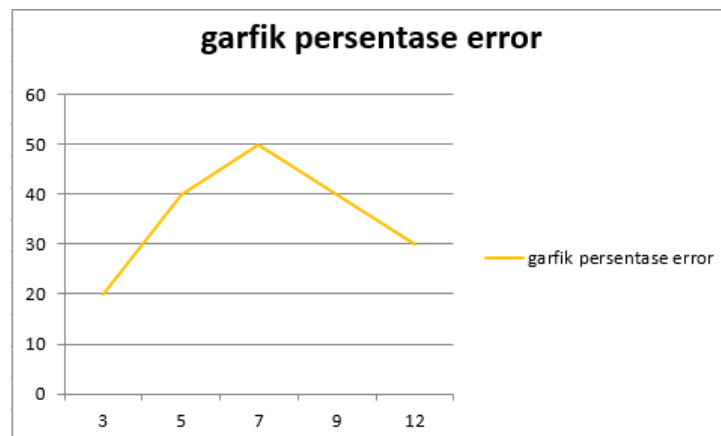
No	Id Sel	Nilai K	Keterangan	Persentase	Tingkat
1	6 (P=98 L=84)	3	Sistem Berhasil	Error 20%	Akurasi 80%
2	8 (P=33 L=40)		Sistem Berhasil		
3	12 (P=34 L=44)		Sistem Berhasil		
4	13 (P=47 L= 60)		Sistem Berhasil		
5	26 (P=89 L=66)		Sistem Berhasil		
6	35 (P=88 L=57)		Tidak Berhasil		
7	43 (P=79 L=49)		Tidak Berhasil		
8	50 (P=34 L=36)		Sistem Berhasil		
9	95 (P= 86 L=85)		Sistem Berhasil		
10	98 (P=96 L=65)		Sistem Berhasil		
11	6 (P=98 L=84)	5	Tidak Berhasil	40%	60%
12	8 (P=33 L=40)		Sistem Berhasil		
13	12 (P=34 L=44)		Sistem Berhasil		
14	13 (P=47 L= 60)		Sistem Berhasil		
15	26 (P=89 L=66)		Sistem Berhasil		
16	35 (P=88 L=57)		Tidak Berhasil		
17	43 (P=79 L=49)		Tidak Berhasil		
18	50 (P=34 L=36)		Sistem Berhasil		
19	95 (P= 86 L=85)		Sistem Berhasil		
20	98 (P=96 L=65)		Tidak Berhasil		
21	6 (P=98 L=84)	7	Tidak Berhasil	50%	50%
22	8 (P=33 L=40)		Sistem Berhasil		
23	12 (P=34 L=44)		Sistem Berhasil		
24	13 (P=47 L= 60)		Sistem Berhasil		
25	26 (P=89 L=66)		Sistem Berhasil		
26	35 (P=88 L=57)		Tidak Berhasil		
27	43 (P=79 L=49)		Tidak Berhasil		
28	50 (P=34 L=36)		Sistem Berhasil		
29	95 (P= 86 L=85)		Tidak Berhasil		
30	98 (P=96 L=65)		Tidak Berhasil		
31	6 (P=98 L=84)	9	Sistem Berhasil	40%	60%
32	8 (P=33 L=40)		Sistem Berhasil		
33	12 (P=34 L=44)		Sistem Berhasil		
34	13 (P=47 L= 60)		Sistem Berhasil		
35	26 (P=89 L=66)		Sistem Berhasil		
36	35 (P=88 L=57)		Tidak Berhasil		
37	43 (P=79 L=49)		Tidak Berhasil		
38	50 (P=34 L=36)		Sistem Berhasil		
39	95 (P= 86 L=85)		Tidak Berhasil		
40	98 (P=96 L=65)		Tidak Berhasil		

No	Id Sel	Nilai K	Keterangan	Persentase	Tingkat
41	6 (P=98 L=84)	12	Sistem Berhasil	30%	70%
42	8 (P=33 L=40)		Sistem Berhasil		
43	12 (P=34 L=44)		Sistem Berhasil		
44	13 (P=47 L= 60)		Sistem Berhasil		
45	26 (P=89 L=66)		Sistem Berhasil		
46	35 (P=88 L=57)		Tidak Berhasil		
47	43 (P=79 L=49)		Tidak Berhasil		
48	50 (P=34 L=36)		Sistem Berhasil		
49	95 (P= 86 L=85)		Sistem Berhasil		
50	98 (P=96 L=65)		Tidak Berhasil		

#### 4.4 Grafik Persentase Error

Tabel 4.4 perbandingan nilai K dan Persentase Error

Nilai K	Persentase Error
3	20
5	40
7	50
9	40
12	30



Berdasarkan grafik diatas dapat kita lihat bahwa, nilai K terbaik untuk proses identifikasi anemia sel sabit dengan pengolahan citra dan algoritma *K-Nearest Neighbor (KNN)* adalah 3, hal ini dapat di lihat bahwa ketika menggunakan nilai K=3, persentase error yang diperoleh sangat kecil yaitu 20 % dengan tingkat akurasi yang tertinggi yaitu 80%.



### 5.1 Kesimpulan

1. Algoritma *k-nearest neighbor* (k-NN atau KNN) dapat mengidentifikasi sel darah yang berbentuk sabit dan sel darah sehat.
2. Setelah dilakukan pengujian terhadap 10 sampel darah (5 sel sehat dan 5 sel sabit) dengan menggunakan nilai k yang bervariasi, dapat disimpulkan nilai k terbaik untuk proses klasifikasi ini adalah 3.
3. Setelah melakukan pengujian terhadap 10 sampel darah, tingkat akurasi yang paling tinggi diperoleh ketika menggunakan nilai  $K=3$ , yaitu 80%.

### 5.2 Saran

1. Untuk penelitian selanjutnya dapat menggunakan metode klasifikasi lainnya seperti JST (jaringan syaraf tiruan).
2. Dapat menggunakan metode *edge detection* untuk mendapatkan nilai tepi dari objek.
3. Sistem ini belum bisa mengidentifikasi objek yang menumpuk (overlapping). Untuk itu kedepannya diperlukan pengembangan penelitian yang bisa melakukan identifikasi pada citra darah yang mengalami overlapping.

### Daftar Pustaka

1. Sibero, Alexander F.K. (2010). Dasar - dasar Visual Basic.NET. Yogyakarta :Mediakom. Hal 9.
2. Sofwan, Akhmad. Belajar Mysql dengan Php my admin. Fakultas Teknologi Informasi, Universitas Budi Luhur: 2007.
3. Budianti Elvia, Jasril, Handayani Lestari. Implementasi Pengolahan Citra Dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi. Fakultas Sains Dan Teknologi, UIN Sultan Syarif Qasim Riau: 2015.
4. Maryana Sufiatul, Karlitasari Lita, Qur'ania Arie. Pemanfaatan K-Nearest Neighbour Pada Penngenaln Wajah Dengan Praproses Transformasi Wavelet. Fakultas MIPA, UNPAK.
5. Arandika Azhar, Drs.Mardji, Cholissodin Imam S.Si. Implementasi Algoritma Klasifikasi K-Nearest Neighbour(K-NN) Untuk Klasifikasi Data Wine. Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Subekti Imam. Identifikasi Sel Darah Berbentuk Sabit Pada Citra Sel Darah Penderita Anemia. Jurusan Teknik Elektro ITS.
7. Callista Media. Perancangan dan Implementasi Aplikasi Rekomendasi Ukuran Baju menggunakan IP Camera. Jurusan Teknik informatika Politeknik Caltex Riau.