

**DETEKSI DAN KLASIFIKASI TIPE BANGUNAN PADA CITRA  
SATELIT MENGGUNAKAN METODE  
*K NEAREST NEIGHBOR***

**SKRIPSI**

Oleh:  
**ADAM JAELANI**  
NIM. 14650050



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2020**

**DETEKSI DAN KLASIFIKASI TIPE BANGUNAN PADA CITRA  
SATELIT MENGGUNAKAN METODE  
K NEAREST NEIGHBOR**

**SKRIPSI**

**Diajukan Kepada:  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
ADAM JAELANI  
NIM. 14650050**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG**

**2020**

ii

**LEMBAR PERSETUJUAN**

**DETEKSI DAN KLASIFIKASI TIPE BANGUNAN PADA CITRA  
SATELIT MENGGUNAKAN METODE  
*K NEAREST NEIGHBOR***

**SKRIPSI**

Oleh:  
**ADAM JAELANI**  
**NIM. 14650050**

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: Mei 2020

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Fachrul Kurniawan, M.MT  
NIP. 19771020 200912 1 001

Dr. M Faisal, MT  
NIP. 19740510 200501 1 007

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdiان  
NIP. 19740424 200901 1 008

## LEMBAR PENGESAHAN

### DETEKSI DAN KLASIFIKASI TIPE BANGUNAN PADA CITRA SATELIT MENGGUNAKAN METODE *K NEAREST NEIGHBOR*

Oleh:  
**ADAM JAELANI**  
NIM. 14650050

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjan Komputer (S. Kom)  
Tanggal: Juni 2020

#### Susunan Dewan Penguji

#### Tanda Tangan

- |                       |  |     |
|-----------------------|--|-----|
| 1. Penguji Utama      | : <u>Yunifa Miftachul Arif, M.T</u><br>NIP. 198306162011011004   | ( ) |
| 2. Ketua Penguji      | : <u>Hani Nurhayati, M.T</u><br>NIP. 197806252008012006          | ( ) |
| 3. Sekertaris Penguji | : <u>Dr. Fachrul Kurniawan, M.MT</u><br>NIP. 19771020 2009121001 | ( ) |
| 4. Anggota Penguji    | : <u>Dr. M Faisal, MT</u><br>NIP. 19740510 2005011007            | ( ) |

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Adam Jaelani

NIM : 14650050

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : **DETEKSI DAN KLASIFIKASI TIPE BANGUNAN PADA  
CITRA SATELIT MENGGUNAKAN METODE  
*K NEAREST NEIGHBOR***

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Juni 2020  
Yang membuat pernyataan,



Adam Jaelani  
NIM. 14650050

## MOTTO

**“Not knowing oneself that's the worst”**

“Don't compare yourself with others, There is no comparison between the sun & the moon, they shine when it's their time”



## HALAMAN PERSEMBAHAN

Puji Syukur kehadirat Allah Subhanahu wa ta'ala atas segala limpahan RahmatNya, Serta Sholawat dan salam semoga selalu tercurah kepada Rasul Muhammad Shallallahu alihhi wassalam.

Ucapan Terimakasih dan Karya ini saya persembahkan kepada:

Kedua Orangtuaku tercinta Bapak Sucita dan Ibu Rohimah, atas doa dan kasih sayang yang selalu tercurah, Semoga Allah menganugrahkan umur panjang, kesehatan, ketaatan dan selalu dalam ridhoNya.

Never forget two people in mylife: The person who lost every thing just to make me win (My Father). The person who was with me in every pain (My Mother)

Ketiga Saudaraku tercinta, kakakku Rahmat Saleh dan Mustofa Hambali dan adikku Aprilia Rahmi semoga selalu dalam lindungan Allah dan dapat menjadi anak anak yang berbakti kepada orang tua.

Kakek, Nenek, Paman, Bibi dan semua keluarga besarku semoga dikumpulkan kembali didalam surgaNya.

Dosen pembimbing saya Bapak Fachrul Kurniawan, M.MT dan Bapak Dr. M Faisal, M.T. Seluruh dosen dan Staf Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang atas semua bimbingan dan ilmu yang diajarkan semoga selalu diberi keberkahan dan RidhoNya.

Sahabat sahabat seperjuangan Teknik Informatika B 14, Biner 14, dan semua angkatan Teknik Informatika Uin Malang.

Sahabat rantau dan sahabat Forum Silaturahmi Mahasiswa Lombok

Semua orang yang saya sayangi yang tidak bisa saya sebutkan satu persatu Saya ucapkan terimakasih yang luar biasa, semoga persahabatan dan kekeluargaan ini selalu dalam Ridho Allah Subhanahu wa ta'ala, Aamiin.

## KATA PENGANTAR

Segala puji bagi Allah Subhanahu wa ta'ala atas limpahan rahmat, taufiq dan hidayahNya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Deteksi dan Klasifikasi Tipe Bangunan Pada Citra Satelite Menggunakan Metode *k Nearest Neighbor*” sebagai syarat untuk memperoleh gelar sarjana Program Studi Teknik Informatika jenjang Strata- 1 Universitas Islam negeri Maulana Malik Ibrahim Malang.

Sholawat dan salam senantiasa tercurah kepada Rasul Mulia Muhammad Shalallahu alaihi wassalam, keluarga dan para sahabatnya yang telah membawa umat manusia dari zaman yang gelap gulita jahiliyah menuju ke zaman yang terang benderang, Islam yang diridhoi Allah Subhanahu wa ta'ala.

Penulis menyadari bahwa ada banyak keterbatasan yang penulis miliki, sehingga ada banyak pihak yang telah memberikan bantuan baik secara moril ataupun materil dalam menyelesaikan penelitian ini. Maka dari itu dengan segenap kerendahan hati penulis mengucapkan terimakasih kepada:

1. Prof Dr H Abd. Haris, M.Ag selaku Rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan selaku Ketua Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang.
4. Fachrul Kurniawan, M.MT selaku pembimbing I dan Dr. M Faisla M.T selaku pembimbing II yang senantiasa memberikan bimbingan, arahan dan masukan kepada penulis.

5. Seluruh Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
6. Segenap Civitas Akademik Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
7. Kedua Orang tua dan seluruh keluarga besar penulis yang senantiasa memberi dukungan.
8. Rekan rekan seperjuangan Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Penulis menyadari bahwa dalam karya ini masih banyak kekurangan, oleh karena itu penulis selalu menerima segala keritik dan saran dari pembaca. Semoga karya ini bermanfaat bagi seluruh pihak.

Malang, Juni 2020

Penulis

## DAFTAR ISI

<b>LEMBAR PENGAJUAN.....</b>	<b>ii</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>iv</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>v</b>
<b>HALAMAN MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>ABSTRAK .....</b>	<b>xvii</b>
<b>ABSTRACT .....</b>	<b>xviii</b>
<b>ملخص البحث.....</b>	<b>xix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Pernyataan Masalah.....	5
1.3. Tujuan Penelitian.....	5
1.4. Batasan Penelitian .....	6
1.5. Manfaat Penelitian.....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1. Penelitian Terkait .....	7
2.2. Landasan Teori.....	9
2.2.1. Model Warna (Color Models).....	10
2.2.1.1. Ruang Warna RGB .....	11
2.2.1.2. Ruang Warna $L^*a^*b$ .....	12
2.2.2. Segmentasi Gambar .....	14
2.2.3. Ekstarksi Ciri.....	18
2.2.3.1. Ekstraksi Ciri Bentuk .....	18
2.2.3.2. Ekstataksi CiriaUkuran .....	20
2.2.3.3. Ekstraksi CiriaGeometri.....	20

2.2.3.4. Ekstraksi Ciri Tekstur .....	21
2.2.3.5. Ekstraksi Ciri Warna .....	21
2.2.4. Operasi Morfologis .....	23
2.2.4.1. Translasia .....	23
2.2.4.2. Dilasi .....	24
2.2.4.3. Erosi .....	24
2.2.4.4. Opening .....	24
2.2.5. K Nearest Neighbora .....	25
2.2.5.1. Cara Kerja Algoritma <i>k-NN</i> .....	26
2.2.5.2. Perhitungan Jarak .....	28
2.2.5.3. Kinerja Algoritma <i>k-NN</i> .....	30
2.2.5.4. Kelebihan dan Kekurangan Algoritma <i>k-NN</i> .....	30
2.2.6. <i>K Fold Cross Validationa</i> .....	31
<b>BAB III DESAIN PENELITIAN .....</b>	<b>33</b>
3.1. Desain Sistem .....	33
3.1.1. Citra Satelit <i>RGB</i> .....	34
3.1.2. Segmentasi .....	34
3.1.2.1. Proses Segmentasi .....	35
3.1.3. Ekstraksi Ciri .....	41
3.1.4. Klasifikasi <i>k-Nearest Neighbora</i> .....	44
3.1.4.1. Flowchart Algoritma <i>k-Nearest Neighbor</i> .....	46
3.1.4.2. Menghitung <i>Euclidean Distance</i> .....	49
3.1.4.3. Klasifikasi Data .....	50
3.1.5. Perhitungan Akurasi .....	52
<b>BAB IV UJICOBAN DAN PEMBAHASAN .....</b>	<b>54</b>
4.1. Implementasi .....	54
4.1.1. Software .....	54
4.1.2. Hardware .....	54
4.2. Uji Coba Deteksi Bangunan .....	55
4.2.1. Citra Hasil Segmentasi .....	55
4.2.2. Citra Hasil Deteksi Bangunan .....	57
4.3. Proses Klasifikasi Bangunan .....	60
4.3.1. Ekstraksi Ciri Objek Deteksi .....	61

4.3.2.	Uji Coba Akurasi Klasifikasi KNN .....	62
4.3.2.1.	Dataset Uji Coba Klasifikasi KNN.....	62
4.3.2.2.	Skenario Pengujian 1 .....	64
4.3.2.3.	Skenario Pengujian 2: .....	68
4.3.2.4.	Skenario pengujian 3 .....	72
4.4.	Integrasi Dengan Al Quran.....	78
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>81</b>
5.1.	Kesimpulan.....	81
5.2.	Saran .....	81
<b>DAFTAR PUSTAKA .....</b>		<b>83</b>
<b>LAMPIRAN.....</b>		<b>85</b>



## DAFTAR GAMBAR

Gambar 2.1 Koordinat Warna RGB.....	11
Gambar 2. 2 <i>LAB Color</i> .....	13
Gambar 2. 3 Histogramadengan 1 lembahdiantara 2 puncak.....	15
Gambar 2. 4 Contoh Segmentasiadengan <i>Histogramadan Region</i> .....	15
Gambar 2. 5 GambaraSegmentasi Pesawat.....	17
Gambar 2. 6 Ilustrasi Perhitungan <i>Eccentricity</i> .....	19
Gambar 2. 7 Ilustrasi Perhitungan Matric .....	19
Gambar 2. 8 Contoh Ciri Ukuran.....	20
Gambar 2. 9 Contoh Bentuk Bentuk Geometri.....	20
Gambar 2. 10 Contoh Tekstu Citra .....	21
Gambar 2. 11 <i>RGB ColorSpace</i> dan <i>HSVaColor Space</i> .....	23
Gambar 2. 12 Contoh Klasifikasi <i>k- NN</i> .....	26
Gambar 3. 1 Desain Sistem.....	33
Gambar 3. 2 Citra Satelit Hasil Cropping.....	34
Gambar 3. 3 Proses Segmentasi .....	35
Gambar 3. 4 Inputan Citra <i>RGB</i> .....	35
Gambar 3. 5 Proses Konversi Warna .....	36
Gambar 3. 6 Hasil Konversi <i>RGB</i> ke <i>LAB</i> .....	37
Gambar 3. 7 Citra <i>RGB</i> .....	38
Gambar 3. 8 Konversi <i>RGB</i> ke <i>L*a*b</i> .....	38
Gambar 3. 9 Citra Hasil Segmentasi .....	39
Gambar 3. 10 Hasil Segmentasi <i>Cluster 1</i> .....	39
Gambar 3. 11 Hasil Segmentasi <i>Cluster 2</i> .....	39

Gambar 3. 12 Hasil Segmentasi <i>Cluster 3</i> .....	40
Gambar 3. 13 Cluster Terdapat <i>Foreground</i> Bangunan.....	40
Gambar 3. 14 Hasil Morfologi <i>Filling Holes</i> dan <i>Bwarea Open</i> .....	41
Gambar 3. 15 <i>Flowchart</i> Ekstraksi Ciri.....	42
Gambar 3. 16 Labelling Objek.....	42
Gambar 3. 17 Rumus <i>Eccentricity</i> .....	44
Gambar 3. 18 Rumus <i>Metric</i> .....	44
Gambar 3. 19 Flowchat Algoritma <i>k-NN</i> .....	46
Gambar 4. 1 Blok Deteksi Bangunan.....	55
Gambar 4. 2 Diagram Blok Klasifikasi sistem.....	60
Gambar 4. 3 Gambar Uji.....	61
Gambar 4. 4 Citra Bangunan.....	75
Gambar 4. 5 Interface Sistem.....	75

## DAFTAR TABEL

Tabel 2. 1 Ilustrasi <i>k-Fold Cross Validation</i> .....	31
Tabel 3. 1 Luas, Keliling, <i>Eccentricity</i> , dan <i>Metrik</i> Hasil Ekstraksi.....	43
Tabel 3. 2 Contoh Data Training Tipe Bangunan .....	45
Tabel 3. 3 Sampel Data Latih Bangunan .....	47
Tabel 3. 4 Sampel Data Training Tipe Bangunan.....	48
Tabel 3. 5 Tabel Pehitungan <i>Euclidean Distance</i> .....	49
Tabel 3. 6 Hasil Perhitngan <i>Euclidean Distance</i> .....	50
Tabel 3. 7 $k=3$ .....	50
Tabel 3. 8 $k=5$ .....	51
Tabel 3. 9 $k=7$ .....	51
Tabel 3. 10 $k=9$ .....	52
Tabel 3. 11 <i>Confusion Matric Model</i> .....	53
Tabel 4. 1 Hasil Segmentasi Citra.....	56
Tabel 4. 2 Tabel Hasil Deteksi Bangunan.....	57
Tabel 4. 3 Akurasi Deteksi Sistem.....	59
Tabel 4. 4 Tabel Hasil Ekstraksi Ciri .....	62
Tabel 4. 5 Tipe Bangunan A .....	63
Tabel 4. 6 Tipe Bangunan B .....	63
Tabel 4. 7 Tipe Bangunan C .....	63
Tabel 4. 8 Hasil Klasifikasi Knn .....	65
Tabel 4. 9 Rumus <i>Confussion Matrix</i> .....	66
Tabel 4. 10 <i>Confussion matrix</i> $k=3$ .....	66
Tabel 4. 11 <i>Confussion Matrix</i> $k=5$ .....	66

Tabel 4. 12 <i>Confussion Matrix k=7</i> .....	67
Tabel 4. 13 <i>Confussion Matrix k=9</i> .....	67
Tabel 4. 14 <i>Confussion Matrix k=11</i> .....	67
Tabel 4. 15 Akurasi Hasil Klasifikasi <i>k</i> .....	68
Tabel 4. 16 Pengujian <i>k-fold=5</i> .....	69
Tabel 4. 17 Test <i>fold 1</i> .....	69
Tabel 4. 18 Test <i>fold 2</i> .....	70
Tabel 4. 19 Test <i>fold 3</i> .....	70
Tabel 4. 20 Test <i>fold 4</i> .....	70
Tabel 4. 21 Test <i>fold 5</i> .....	71
Tabel 4. 22 Akurasi <i>k fold</i> .....	71
Tabel 4. 23 Perbandingan akurasi <i>k NN</i> dan <i>k Fold</i> .....	72
Tabel 4. 24 Hasil Deteksi dan Klasifikasi Sistem.....	77

## ABSTRAK

Jaelani, Adam. 2020. **Deteksi dan Klasifikasi Tipe Bangunan Pada Citra Satellite Menggunakan Metode  $k$  Nearest Neighbor**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fachrul Kurniawan, M. MT. (II) Dr. M Faisal, MT.

Kata kunci: Citra Satelit, Segmentasi Warna, Deteksi Bangunan,  $k$  Fold Cross Validation,  $k$  Nearest Neighbor

Kepadatan bangunan menimbulkan berbagai permasalahan di kota-kota besar di Indonesia, menjadikan kota cerdas muncul menjadi solusi untuk mengurangi masalah yang ditimbulkan, agar infrastruktur kota menjadi cerdas maka konsep identifikasi objek harus digunakan untuk mendeteksi dan menganalisa keberadaan mobil, jalan, dan bangunan. Dengan adanya citra satelit maka proses pengolahan citra digital terkait dengan objek-objek permukaan bumi akan sangat membantu dalam menyelesaikan permasalahan geografis, tata ruang kota dan lain-lain. Metode segmentasi warna digunakan untuk mendeteksi objek bangunan pada citra satelit digital dan metode  $k$  Nearest Neighbor digunakan untuk mengklasifikasi objek bangunan berdasarkan hasil ekstraksi ciri yang didapatkan. Rata-rata akurasi deteksi bangunan dengan segmentasi warna yaitu 90.8%, pengujian nilai akurasi  $k$  Nearest Neighbor dengan  $k=5$  mendapatkan akurasi terbaik 93%. Akurasi hasil pengujian ketepatan klasifikasi  $k$  Nearest Neighbor dengan  $k$  Fold Cross Validation menunjukkan rata-rata akurasi  $k$  Nearest Neighbor sebesar 93%.

## ABSTRACT

Jaelani, Adam. 2020. Detection and Classification of Building Types in Satellite Image Using *k Nearest Neighbor* Method. Thesis. Department of Informatics, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Supervisor: (I) Fachrul Kurniawan, M. MT. (II) Dr. M Faisal, MT.

Keywords: Satellite Imagery, Color Segmentation, Building Detection, *k Fold Cross Validation*, *k Nearest Neighbor*

Building density causes various problems in big cities in Indonesia, making smart cities emerge as a solution to reduce the problems that are caused, so that city infrastructure becomes intelligent, the concept of object identification must be used to detect and analyze the presence of cars, roads and buildings. With the presence of satellite imagery, the process of processing digital images associated with objects on the surface of the earth will be very helpful in solving geographical problems, the layout of cities and others. The color segmentation method is used to detect building objects in digital satellite imagery and the *k Nearest Neighbor* method is used to classify building objects based on the feature extraction obtained. The average accuracy of building detection with color segmentation is 90.8%, testing the accuracy value of *k Nearest Neighbor* with  $k = 5$  gets the best accuracy of 93%. The accuracy of the results of the classification of *k Nearest Neighbor* with *k Fold Cross Validation* shows the average accuracy of *k Nearest Neighbor* by 93%.

## ملخص البحث

جيلاني، آدم. ٢٠٢٠. كشف وتصنيف أنواع المباني في صورة الأقمار الصناعية باستخدام طريقة *k Nearest Neighbor*. البحث الجامعي. شعبة الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.

المشرف: فخر الكورنيوان، الماجستير، والدكتور محمد فيصل، الماجستير

الكلمات الرئيسية: صورة الأقمار الصناعية، تجزئة الألوان ، كشف المباني ، *k Fold Cross Validation* ، *k Nearest Neighbor* ، Validation

المبنى الممتلئ يسبب مشاكل مختلفة في المدن الكبرى في إندونيسيا، مما يجعل المدن الذكية أن تظهر كحل للحد المشاكل لان تصبح البنية التحتية للمدينة ذكية ، ويجب أن يستخدم مفهوم تحديد الكائن لاكتشاف وتحليل وجود السيارات والطرق والمباني. مع وجود صورة الأقمار الصناعية، ستكون عملية معالجة الصورة الرقمية التي تتعلق بالأجسام على سطح الأرض مفيدة جدًا في حل المشكلات الجغرافية، وتخطيط المدن وغيرها. استخدم طريقة تجزئة الألوان للكشف عن كائن المبني في صورة الأقمار الصناعية الرقمية ، واستخدم طريقة *k Nearest Neighbor* لتصنيف كائن المبني بناء على استخراج المعالم. دقة الكشف المباني المتوسط مع تجزئة الألوان هي 90.8% ، ويختبر قيمة دقة *k Nearest Neighbor* مع  $k = 5$  أفضل دقة أي 93%. دلت دقة نتائج الاختبار التصنيف *k Nearest Neighbor* مع *k Fold Cross Validation* دقة متوسط *k Nearest Neighbor* بنسبة 93% .

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Dari hasil sensus penduduk jumlah penduduk di Indonesia pada tahun 2010 adalah sebanyak 237.641.326 jiwa, yang mencakup mereka yang bertempat tinggal di daerah perkotaan sebanyak 118.320.256 jiwa (49.79%) dan di daerah pedesaan sebanyak 119.321.070 jiwa (50.21%) (bps.go.id:2010). Tahun ini, jumlah penduduk Indonesia yang tinggal di perkotaan diperkirakan telah mencapai 54%. Jika saat ini penduduk Indonesia sudah lebih dari 240 juta, artinya paling sedikit ada 129,6 juta orang yang menyesaki perkotaan. Angka ini melambung tinggi disbanding hasil sensus penduduk tahun 2010. Saat itu, sebanyak 49,8 % dari 237,6 juta penduduk Indonesia tinggal di kota (Wahyudi, 2012). Semakin hari kita akan melihat semakin tingginya migrasi dari zona desa menuju zona kota. Ketimpangan antara desa dengan kota menjadi penyebab terus meningkatnya Urbanisasi dari tahun ke tahun. Oleh karena itu permasalahan baru akan muncul di kota-kota besar di Indonesia seperti: kelangkaan sumberdaya, polusi, kemacetan lalulintas, kesehatan, infrastruktur yang tidak memadai, dan lain-lain.

Menjadikan kota 'cerdas' muncul menjadi solusi untuk mengurangi masalah yang ditimbulkan oleh pertumbuhan populasi perkotaan dan urbanisasi yang cepat. Lebih dari setengah penduduk dunia sekarang tinggal di daerah perkotaan. Pergeseran dari zona pedesaan menuju perkotaan diproyeksikan akan berlanjut dalam beberapa dekade mendatang. Akibatnya kota harus tumbuh, tetapi penting harmoni itu ada atau dihasilkan antara aspek spasial, sosial, ekonomi dan lingkungan. Kota disebut kota pintar harus membuat perencanaan kota itu lebih pintar.

Konsep *Smart City* untuk negara berkembang terdiri dari delapan parameter, yaitu *Smart Energy*, *Smart Infrastructure*, *Smart People*, *Smart Health*, *Smart Mobility*, *Smart Governance*, *Smart Educatuin*, dan *Smart Technology*. Perbedaan antara negara maju dan negara berkembang terletak pada ketersediaan infrastruktur perkotaan, sumber daya manusia yang kompeten, dan teknologi canggih, yang mungkin berbeda dalam beberapa tahun (Kurniawan, Nugroho, & Hariadi, 2018).

*Smart Infrastruktur* sangat membantu dalam meningkatkan perencanaan kota dalam membentuk *Smart City*, sehingga dapat menerapkan sistem pengalamatan bangunan cerdas di dalam kota. Agar infrastruktur kota menjadi cerdas maka konsep identifikasi objek harus digunakan untuk mendeteksi dan menganalisa keberadaan mobil, jalan, bangunan dan lain lain. Dari konteks ini penulis berfokus pada studi ini tentang topik penelitian deteksi bangunan yang merupakan batuan dasar dalam membangun infrastruktur *Smart City* yang lengkap.

Selain itu banyaknya jenis bangunan dengan berbagai bentuk dan ukuran membutuhkan algoritma yang bisa mengklasifikasikan tipe tipe bangunan sehingga dapat dibedakan antara bangunan satu dengan bangunan yang lain dengan mudah dan akurat. Algoritma *k Nearest Neighbor* merupakan teknik klasifikasi yang sangat populer yang telah terbukti menjadi algoritma sederhana yang baik. *K Nearest Neighbor* merupakan salah satu metode yang digunakan dalam pengklasifikasian dengan menggunakan algoritma *Supervised*. Tujuan dari algoritma ini adalah untuk

mengklasifikasikan objek baru berdasarkan jarak suatu objek yang akan diklasifikasikan terhadap data contoh. *Classifier* hanya menggunakan fungsi jarak dari data baru ke data *training* (Whidhiasih, Wahanani, & Supriyanto, 2013)

Pesatnya perkembangan teknologi khususnya dalam bidang pengolahan citra digital yang banyak dimanfaatkan di berbagai bidang untuk menyelesaikan banyak permasalahan. Pengolahan citra satelit merupakan salah satu topik penelitian yang banyak dikaji beberapa tahun belakangan ini, dengan adanya citra satelit maka proses pengolahan citra digital terkait dengan objek permukaan bumi akan sangat membantu manusia dalam menyelesaikan permasalahan seperti geografis, kependudukan, keamanan dan lain lain. Salah satu penelitian terkait pengolahan citra satelit digital adalah deteksi bangunan.

Deteksi bangunan dari data penginderaan jauh adalah sangat penting untuk industri perumahan, perencanaan kota, keamanan dalam negeri, manajemen bencana dan banyak aplikasi lainnya. Ekstraksi batas batas bangunan secara otomatis juga merupakan langkah penting menuju pembuatan model kota (Liang, Gong, Chen, & Han, 2008)

Dalam ayat Al Quran surat Ali Imran ayat 190- 191, dijelaskan bahwa manusia diperintahkan untuk memperhatikan tanda tanda kebesaran Allah sehingga manusia dapat mengambil pelajaran sekaligus manfaat dari apa yang diperhatikannya.

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَأَخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِأُولِي  
 الْأَلْبَابِ ۗ الَّذِينَ يَذْكُرُونَ اللَّهَ قِيَمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ  
 فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَذَا بَطْلًا سُبْحَانَكَ فَقِنَا  
 عَذَابَ النَّارِ ۗ

190. *Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal*

191. *(yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): "Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia, Maha Suci Engkau, maka peliharalah kami dari siksa neraka*

#### **Tafsir Al Quran Surat Ali Imran Ayat 190-191:**

Sesungguhnya dalam penciptaan langit dan bumi yang tanpa ada contoh sebelumnya dan pergantian malam dan siang dan perbedaan waktu keduanya dengan memanjang dan memendek benar benar merupakan petunjuk petunjuk dan bukti bukti yang agung atas keesaan Allah bagi orang orang yang mempunyai akal akal yang selamat (Tafsir al Muyassar).

Yaitu orang orang yang mengingat Allah dalam semua kondisi mereka, baik berdiri, duduk dan dalam keadaan mereka berbaring. Mereka mentadaburi dalam penciptaan langit dan bumi seraya berkata, “wahai Tuhan

kami, Engkau tidaklah menciptakan makhluk ciptaan ini dengan sia sia. Dan Engkau Maha Suci dari hal itu. Maka jauhkanlah dari kami siksaan neraka (Tafsir al Muyassar).

## 1.2. Pernyataan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dapat diketahui permasalahan yang dapat difokuskan pada penelitian ini adalah:

- a. Bagaimana membangun sistem deteksi bangunan berdasarkan segmentasi gambar dan klasifikasi tipe bangunan menggunakan metode *k Nearest Neighbor*?
- b. Bagaimana mendapatkan tingkat akurasi metode *k Nearest Neighbor* untuk klasifikasikan bangunan pada citra satelit?

## 1.3. Tujuan Penelitian

Berdasarkan identifikasi permasalahan yang telah diuraikan sebelumnya, maka tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Penelitian ini untuk membangun sistem deteksi dan klasifikasi tipe bangunan dari citra satelit menggunakan metode segmentasi warna dan klasifikasian tipe bangunan menggunakan metode *k Nearest Neighbor*.
- b. Penelitian ini untuk mendapatkan tingkat akurasi dari metode segmentasi gambar dan *k Nearest Neighbor* dalam mendeteksi dan mengklasifikasikan bangunan pada citra satelit.

#### 1.4. Batasan Penelitian

Batasan penelitian ini bertujuan untuk memfokuskan dan memudahkan dalam penelitian, maka diperlukan beberapa batasan-batasan dalam pengerjaannya, diantaranya sebagai berikut ini:

- a. Citra Satelit yang digunakan diambil dari citra satelit *Google Earth*.
- b. Citra satelit yang digunakan diambil dari ketinggian 700 – 800 meter.

#### 1.5. Manfaat Penelitian

Adapun hasil penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi penulis, penelitian ini dapat menambah pengetahuan dan wawasan mengenai permasalahan deteksi bangunan dari citra satelit dan pengklasifikasian tipe bangunan menggunakan metode *k Nearest Neighbor*.
- b. Bagi pengguna, hasil dari penelitian ini dapat membantu untuk melakukan deteksi bangunan pada suatu wilayah secara cepat dan mudah sehingga permasalahan akibat kepadatan bangunan dapat diatasi.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terkait

(Awrangjeb, Ravanbakhsh, & Fraser, 2010) *Automatic detection of residential buildings using LIDAR data and multispectral imagery*. Mengusulkan teknik deteksi bangunan otomatis menggunakan *LIDAR* data dan *colour orthoimagery* yang memiliki empat langkah utama. Pertama, *two mask*, ‘*primary building mask*’ dan ‘*secondary building mask*’ yang dihasilkan dari *LIDAR* data. *Mask* bangunan primer menunjukkan daerah kosong dimana tidak ada laser kembali dibawah ambang batas ketinggian tertentu. *Mask* bangunan sekunder menunjukkan area yang terisi, darimana objek ditinggikan diatas ambang ketinggian yang sama. Kedua, baris segmen dari seluruh area kosong di *mask* bangunan utama diekstrak. Segmen garis di sekitar pohon dihapus menggunakan *Normalized Different Vegetation Index (NDVI)* nilai diturunkan dari gambar *multispectral*. Ketiga, posisi awal bangunan dipulihkan berdasarkan segmen garis yang tersisa. Akhirnya, jejak bangunan yang lengkap diperoleh dari posisi awal menggunakan *two mask* dan *orthoimagery* didalam *YIQ colour system*.

(Akçay & Aksoy, 2010) *Building detection using directional spatial constraints*. Menggambarkan sebuah algoritma untuk mendeteksi bangunan pada citra spasial resolusi tinggi. Setelah *oversegmentation* awal, selanjutnya menggunakan *directional spatial constraints* untuk menemukan kandidat daerah bangunan yang dekat dengan bayangan di sepanjang sudut *azimuth* matahari. Wilayah bangunan dipilih dengan cara pengelompokan daerah kandidat menggunakan *spanning trees minimum*. Selanjutnya mengevaluasi pendekatan

yang diusulkan pada *scene* yang berbeda dengan karakteristik bangunan yang berbeda. Eksperimen menunjukkan bahwa algoritma yang diusulkan mampu mendeteksi bangunan dengan berbagai bentuk dan warna. Selain itu, setelah daerah bangunan terdeteksi, mereka dapat digunakan untuk meningkatkan analisis *scene* dan klasifikasi daerah perkotaan.

(Izadi & Saeedi, 2010) *Automatic building detection in aerial image using a hierarchical feature based image segmentation*. Memperkenalkan deteksi bangunan otomatis dari gambar udara. Metode yang diusulkan menggabungkan fitur *hirarki multilayer* berbasis teknik segmentasi gambar menggunakan warna. Sejumlah atribut geometri atau regional didefinisikan untuk mengidentifikasi daerah potensial dalam berbagai lapisan gambar tersegmentasi. Mekanisme berbasis *tree* digunakan untuk memeriksa daerah tersegmentasi menggunakan hubungan spasial satu sama lain dan karakteristik regional atau geometris. Proses ini memungkinkan terciptanya sekumpulan kandidat daerah yang tervalidasi sebagai atap rumah berdasarkan pada tumpang tindih dan prediksi bayangan setiap daerah sesuai dengan informasi perolehan gambar. Hasil eksperimen menunjukkan akurasi bentuk keseluruhan dan kelengkapan 96%.

(Setiawan & Damayanti, 2016) *Klasifikasi Citra Retina Menggunakan K Nearest Neighbor Untuk Mendeteksi Makulopati Diabetik*. Melakukan pendeteksian penyakit Makulopati Diabetik dari analisis citra fundus yang merupakan citra hasil foto retina menggunakan kamera fundus. Dataset yang digunakan yaitu *MESSIDOR* sebanyak 75 citra retina, terdiri dari 3 kelas yaitu Edema Makula (*EM*) tingkat 1, *EM* tingkat 2 dan *EM* tingkat 3. Ekstraksi fitur menggunakan *Two Dimensional Linear Discriminant Analysis*, sedangkan

klasifikasi menggunakan *k-Nearest Neighbor*. Dari hasil ujicoba didapatkan prosentase pengenalan maksimal hingga 93,33%. Hal ini dipengaruhi oleh semakin banyaknya data pelatihan maka semakin baik prosentase keakuratan.

(Subairi, Rahmadwati, & Yudaningtyas, 2018) ***Implementasi Metode K Nearest Neighbor Pada Pengenalan Pola Tekstur Citra Saliva Untuk Deteksi Ovulasi***. Mendeteksi masa ovulasi pada wanita berdasarkan citra pola *ferning saliva* menggunakan metode metode *k-Nearest Neighbor*. Data masukan berupa lima parameter dari fitur tekstur *GLCM (gray-level co-occurrence matrices)*, yang meliputi *ASM (Angular Second Moment)*, *contrast*, *IDM (Inverse Different Moment)*, *entropy*, dan *correlation*, yang diambil dari rata-rata dari masing-masing sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Sedangkan data keluaran adalah salah satu dari tiga klasifikasi dari masa ovulasi, yaitu *infertile*, *intermediate*, dan *fertile* dengan akurasi 93.3%.

## **2.2. Landasan Teori**

Pengolahan citra digital (*Digital Image Processing*) merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami manusia. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat  $(x, y)$  dan amplitude  $f(x, y)$ . Koordinat  $(x, y)$  menunjukkan letak atau posisi piksel dalam suatu citra, sedangkan amplitude  $f(x, y)$  menunjukkan nilai intensitas warna citra.

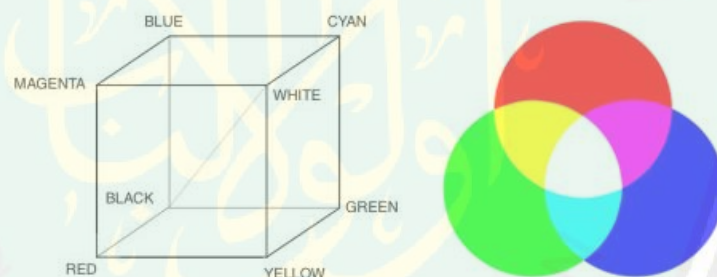
Berdasarkan bentuk sinyal penyusunnya, citra dapat digolongkan menjadi dua jenis yaitu citra analog dan citra digital. Citra analog adalah citra yang dibentuk dari sinyal analog yang bersifat kontinyu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit. Citra analog dihasilkan dari alat akuisisi citra analog, contohnya adalah mata manusia dan kamera analog. Gambar yang tertangkap oleh mata manusia dan foto atau film yang tertangkap oleh kamera analog merupakan contoh dari citra analog. Citra tersebut memiliki kualitas dengan tingkat kerincian (resolusi) yang sangat baik tetapi memiliki kelemahan diantaranya adalah tidak dapat disimpan, diolah, dan dipublikasikan di dalam komputer. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat  $(x, y)$  dan amplitudo  $f(x, y)$ .

### 2.2.1. Model Warna (Color Models)

Ruang warna adalah representasi matematis dari serangkaian warna (Gonzalez & Woods, 2002). Tiga model warna yang paling sering digunakan adalah *RGB* (digunakan di dalam grafik komputer), *YIQ*, *YUV*, dan *YCbCr* (digunakan dalam video system), *CMYK* (digunakan dalam *color printing*). Namun, tidak satupun dari ruang warna ini yang berhubungan dengan pengertian intuitif *hue*, *saturation*, dan *brightness*. Ini menghasilkan pengejaran sementara dari model lain seperti *HIS* dan *HSV*, untuk menyederhanakan pemrograman, pemrosesan dan manipulasi penggunaan akhir. Semua ruang warna dapat berasal dari informasi *RGB* yang disediakan oleh perangkat seperti kamera dan pemindai.

### 2.2.1.1. Ruang Warna RGB

Model warna red, green, and blue (*RGB*) banyak digunakan di seluruh komputer grafik, merupakan tiga warna aditif primer yaitu komponen individual ditambahkan secara bersama untuk membentuk warna yang diinginkan dan di wakikan oleh system kordinat kartesius tiga dimensi, gambar 2.1. Diagonal yang ditunjukkan dari kubus, dengan jumlah yang sama dari setiap komponen utama, mewakili berbagai tingkat abu-abu. Ruang warna RGB adalah pilihan paling umum untuk grafik komputer karena tampilan warna menggunakan merah, hijau, dan biru untuk membuat warna yang diinginkan. Oleh karena itu, pilihan ruang warna **RGB** menyederhanakan arsitektur dan desain sistem.



Gambar 2.1 Koordinat Warna RGB

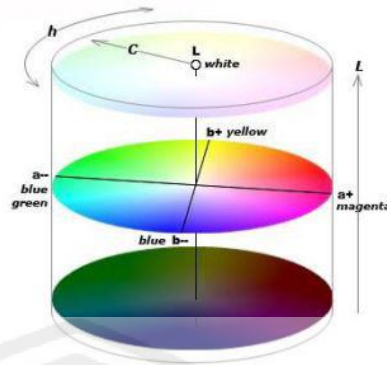
Gambar dalam bentuk *RGB* bekerja dengan 24 bit dimana warna tiap chanel (*R, G, B*) mengandung 8 bit. Sistem yang dirancang menggunakan ruang warna *RGB* dapat memanfaatkan sejumlah besar rutin perangkat lunak yang ada, karena ruang warna ini telah ada selama beberapa tahun.

### 2.2.1.2. Ruang Warna $L^*a^*b^*$

*CIELAB* merupakan model tiga dimensi, hanya dapat digambarkan apabila dalam ruang tiga dimensi, dan apabila diambil irisan kompones  $a^*$  dan  $b^*$ , maka akan mendapatkan diagram *chromaticity*  $a^*b^*$ . Dengan *CIELAB* diberikan makna dari setiap dimensi yang dibentuk:

- a) Besaran *CIE*  $L^*$  untuk mendeskripsikan kecerahan warna, 0 untuk hitam dan  $L^*=100$  untuk putih
- b) Dimensi *CIE*  $a^*$  mendeskripsikan jenis warna hijau-merah, angka negative  $a^*$ : warna hijau; *CIE*  $a^*$  positif mengidentifikasi warna merah.
- c) Dimensi *CIE*  $b^*$  untuk jenis warna biru-kuning, angka negative  $b^*$  mengidentifikasi warna biru dan sebaliknya *CIE*  $b^*$  positif mengidentifikasi warna kuning.

Pengukuran warna dengan metode ini jauh lebih cepat dengan ketepatan yang cukup baik. Pada system ini term penilaian terdiri atas tiga parameter yaitu  $L$ ,  $a$ , dan  $b$ . Lokasi warna pada system ini ditentukan dengan koordinat  $L^*$ ,  $a^*$ ,  $b^*$ . Notasi  $L^*$ : 0 (hitam); 100 (putih) menyatakan cahaya pantul yang menghasilkan warna akromatik putih, abu-abu dan hitam. Notasi  $a^*$ : warna kromatik campuran merah-hijau dengan nilai (+)  $a^*$  (positif) dari 0 sampai +80 untuk warna merah, dan nilai (-)  $a^*$  (negative) dari 0 sampai -80 untuk warna hijau. Notasi  $b^*$ : warna kromatik campuran biru kuning dengan nilai (+)  $b^*$  (positif) dari 0 sampai +70 untuk warna kuning dan nilai (-)  $b^*$  (negative) dari 0 sampai -70 untuk warna biru.



Gambar 2. 2 LAB Color

Ruang warna XYZ, beberapa warna direpresentasikan sebagai nilai yang selalu positif. Perhitungan untuk transformasi warna dari ruang warna RGB ke XYZ (dengan nilai referensi putih), adalah melalui perhitungan matriks transformasi (Sinaga, 2019). Konversi ruang warna dari *type RGB* ke *CIELab*.

Konversi *RGB-XYZ*:

$$[X] = [0.412453 \ 0.357580 \ 0.180423] [R]$$

$$[Y] = [0.212671 \ 0.715160 \ 0.072169] [G]$$

$$[Z] = [0.019334 \ 0.119193 \ 0.950227] [B]$$

Konversi *XYZ – L\*a\*b*:

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \text{ for } \frac{Y}{Y_n} > 0.008856$$

$$L^* = 903.3 \frac{Y}{Y_n}, \text{ for others}$$

$$a^* = 500 \left( f \left( \frac{X}{X_n} \right) - f \left( \frac{Y}{Y_n} \right) \right)$$

$$b^* = 200 \left( f \left( \frac{Y}{Y_n} \right) - f \left( \frac{Z}{Z_n} \right) \right)$$

### 2.2.2. Segmentasi Gambar

Segmentasi merupakan proses mempartisi citra menjadi beberapa daerah atau objek. Segmentasi citra pada umumnya berdasarkan pada sifat *Discontinuity* atau *Similarity* piksel.

- Pendekatan *Discontinuity*: mempartisi citra bila terdapat perubahan intensitas secara tiba-tiba (*Edge Based*).
- Pendekatan *Similarity*: mempartisi citra menjadi daerah-daerah yang memiliki kesamaan sifat tertentu (*Region Based*).

*Contoh: Thresholding, Region Growing, Region Splitting and Merging.*

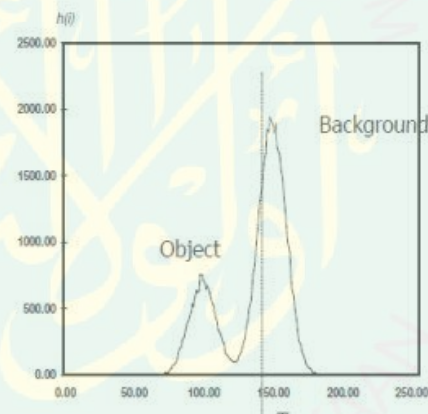
Citra digital berisi piksel-piksel, segmentasi merupakan partisi atau membagi citra ke dalam bagian-bagian yang lebih kecil (*segmen*). Segmentasi dilakukan dengan menggabungkan piksel-piksel yang mirip atau *similar* (piksel  $i$  dan  $j$ ),  $S_{ij} = S_{ij} \geq 0$ . Tujuan segmentasi adalah menggabungkan piksel-piksel yang mirip dengan nilai dari nilai  $S_{ij}$  yang besar, harus pada segmen yang sama, sedangkan piksel-piksel yang tidak mirip pada segmen yang berbeda-beda. Segmentasi dapat juga dianggap sebagai proses pembagian citra ke dalam kelompok-kelompok yang berarti yang saling berkaitan.

Tujuan dari segmentasi adalah membagi-bagi citra dalam bagian-bagian atau region yang penting untuk aplikasi tertentu. Segmentasi dikerjakan dengan berbasis pada: *Greylevel, Colour, Texture, Depth or Motion*. Segmentasi merupakan proses awal dari pengolahan citra selanjutnya, contoh aplikasi segmentasi antara lain: Identifikasi objek berdasarkan ukuran dan bentuk dan identifikasi objek dari *moving picture (objek based video compression)*

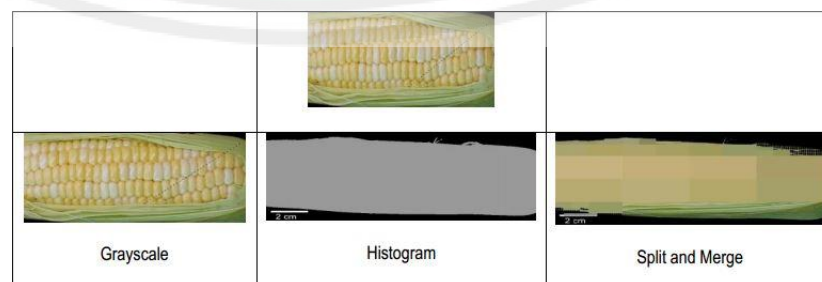
*MPEG4*). Terdapat beberapa macam segmentasi, antara lain *greylevel histogram based segmentation thresholding* dan *greylevel histogram based segmentation clustering*. Didalam tipe segmentasi, terlebih dahulu didefinisikan citra *input* sebagai citra *input signal-to-noise ratio* dalam hal rata rata (*mean*) dari nilai *greylevel* piksel objek dan *piksel background and standard deviasi noise*.

$$\frac{S}{N} = \frac{|\mu_b - \mu_o|}{\sigma}$$

Dapat mudah dipahami bahwa segmentasi berbasis *thresholding* dengan memeriksa *histogram* pada citra dengan *low noise* dan *background*, seperti gambar berikut:



Gambar 2. 3 Histogram dengan 1 lembah diantara 2 puncak



Gambar 2. 4 Contoh Segmentasi dengan *Histogram* dan *Region*

Dalam melakukan pengenalan sebuah objek di antara banyak objek dalam citra, komputer harus melakukan proses segmentasi terlebih dahulu. Berdasarkan cara kerjanya, terdapat dua jenis segmentasi citra yaitu:

1. Segmentasi berdasarkan intensitas warna (derajat keabuan).

Berasumsi bahwa objek-objek yang akan dipisahkan cenderung memiliki intensitas warna yang berbeda-beda dan masing-masing objek memiliki warna yang seragam. Salah satu teknik berdasarkan intensitas warna adalah *Mean Clustering*. Pada *mean clustering* dilakukan pembagian citra dengan membagi *histogram* citra. Kelemahan segmentasi berdasarkan intensitas warna (derajat keabuan) antara lain harus tau dengan tepat berapa jumlah objek yang ada pada citra serta citra hasil kurang bagus jika pada citra terdapat beberapa objek dengan warna pada masing-masing objeknya bervariasi atau pada setiap objek memiliki warna yang sama.

2. Segmentasi berdasarkan karakteristik. Yaitu mengelompokkan bagian-bagian citra yang memiliki karakteristik yang sama berupa perubahan warna antara titik yang berdekatan, nilai rata-rata dari bagian citra tersebut. Untuk menghitung atau menentukan karakteristik digunakan perhitungan statistik, misalnya *varian*, standar deviasi, teori probabilitas, transformasi *fourier*. Salah satu teknik segmentasi berdasarkan karakteristik adalah *Split and Merge*. Proses tersebut dilakukan secara rekursif karena pada setiap saat dilakukan proses yang sama tetapi dengan data yang selalu berubah.

Sedangkan berdasarkan metodenya, terdapat dua pendekatan utama dalam segmentasi citra yaitu didasarkan pada tepi (*Edge Based*) dan didasarkan pada wilayah (*Region Based*). Segmentasi berdasarkan tepi membagi citra berdasarkan diskontinuitas dia antara sub wilayah (*Sub Region*), sedangkan segmentasi berdasarkan wilayah bekerja berdasarkan keseragaman yang ada pada sub wilayah tersebut. Hasil dari segmentasi citra adalah sekumpulan wilayah yang melingkupi citra tersebut, atau sekumpulan kontur yang diekstrak dari citra (pada deteksi tepi). Tiap piksel dalam suatu wilayah mempunyai kesamaan karakteristik atau properti yang dapat dihitung (*Image Properties*), seperti: warna (*Color*), intensitas (*Intensity*), tekstur (*Texture*).



Gambar 2. 5 Gambar Segmentasi Pesawat

Segmentasi wilayah merupakan pendekatan lanjutan dari deteksi tepi. Dalam deteksi tepi segmentasi citra dilakukan melalui identifikasi batas batas objek (*boundaries of object*). Batas merupakan lokasi dimana terjadi perubahan intensitas. Dalam pendekatan didasarkan pada wilayah, maka identifikasi dilakukan melalui wilayah yang terdapat dalam objek tersebut. Salah satu cara untuk mendefinisikan segmentasi citra adalah sebagai berikut.

Sekumpulan wilayah  $\{R_1, R_2, R_3, \dots, R_n\}$  merupakan segmentasi citra  $R$  ke dalam  $n$  wilayah jika:

$$\text{I. } \bigcup_{i=1}^n R_i = R$$

$$\text{II. } R_i \cap R_j = \emptyset, i \neq j$$

III.  $R_i$  terhubung  $i = 1, 2, \dots, n$

IV. Terdapat suatu predikat yang merupakan ukuran homogenisis wilayah:

$$\text{a. } P(R_i) = \text{TRUE}, i = 1, 2, \dots, n$$

$$\text{b. } P(R_i \cup R_j = \text{FALSE}), i \neq j \text{ dan } R_i \text{ adjacent } R_j$$

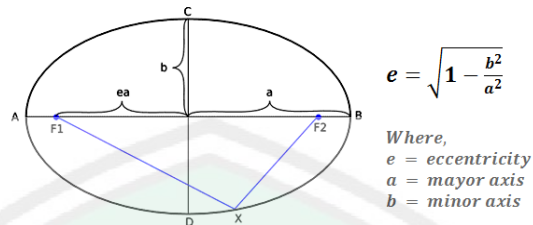
### 2.2.3. Ekstarksi Ciri

Merupakan tahapan mengekstarak ciri atau informasi dari objek di dalam citra yang ingin dikenali atau dibedakan dengan objek lainnya. Ciri yang telah diekstarkasi kemudian digunakan sebagai parameter atau nilai masukan untuk membedakan antara objek satu dengan lainnya pada tahapan identifikasi atau klasifikasi. Ciri umum yang diekstrak antara lain:

#### 2.2.3.1. Ekstraksi Ciri Bentuk

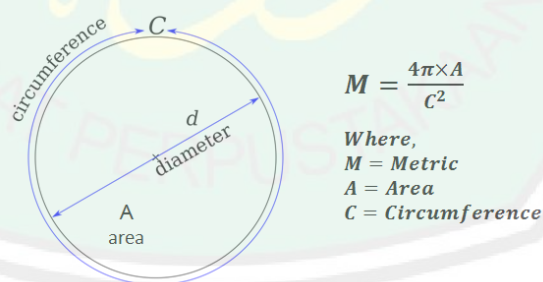
Untuk membedakan bentuk objek satu dengan lainnya, dapat menggunakan parameter yang disebut dengan 'eccentricity'. *Eccentricity* merupakan nilai perbandingan antara jarak *foci ellips minor* dengan *foci ellips mayor* suatu objek. *Eccentricity* memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau mendekati bentuk garis lurus, nilai *Eccentricity*nya mendekati angka 1, sedangkan objek yang berbentuk bulat atau lingkaran nilai

*Eccentricity*nya mendekati angka 0. Perhitungan *Eccentricity* diilustrasikan pada gambar di bawah ini:



Gambar 2. 6 Ilustrasi Perhitungan *Eccentricity*

Parameter lain yang dapat digunakan untuk membedakan bentuk suatu objek yaitu '*metric*'. *Metric* merupakan nilai perbandingan antara luas dan keliling objek. *Metric* memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau mendekati bentuk garis lurus, nilai *metric*nya mendekati angka 0, sedangkan objek yang berbentuk bulat atau lingkaran, nilai *metric*nya mendekati angka 1. Perhitungan *metric* diilustrasikan pada gambar dibawah ini:



Gambar 2. 7 Ilustrasi Perhitungan *Matric*

**2.2.3.2. Ekstataksi Ciri Ukuran**

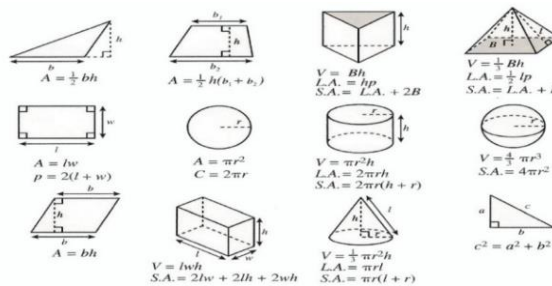
Untuk membedakan ukuran objek satu dengan yang lainnya menggunakan parameter luas dan keliling. Luas merupakan banyaknya piksel yang menyusun suatu objek. Sedangkan keliling merupakan banyaknya piksel yang mengelilingi suatu objek.



Gambar 2. 8 Contoh Ciri Ukuran

**2.2.3.3. Ekstraksi Ciri Geometri**

Ciri geometri merupakan ciri yang didasarkan pada hubungan antara dua buah titik, garis, atau bidang dalam citra digital. Ciri geometri di antaranya adalah jarak dan sudut. Jarak antara dua buah titik (dengan satuan piksel) dapat ditentukan menggunakan persamaan *Euclidean*, *Minkowski*, *Manhattan*, dan lain lain. Jarak dengan satuan piksel tersebut dapat dikonversikan menjadi satuan panjang seperti *millimeter*, *centimeter*, *meter*, dan lain lain dengan cara membaginya dengan resolusi spasial. Sedangkan sudut antara dua buah garis dapat ditentukan dengan perhitungan trigonometri maupun dengannalisis vektor.



Gambar 2. 9 Contoh Bentuk Bentuk Geometri

#### 2.2.3.4. Ekstraksi Ciri Tekstur

Untuk membedakan tekstur objek satu dengan yang lainnya dapat menggunakan ciri statistic orde pertama atau ciri statistik orde dua. Ciri orde pertama didasarkan pada karakteristik histogram citra. Ciri orde pertama umumnya digunakan untuk membedakan tekstur makrostruktur (perulangan pola local secara periodic). Ciri orde pertama antara lain: *Mean*, *Variance*, *Skewness*, *Kurtois*, dan *Entropy*. Sedangkan ciri orde dua didasarkan pada probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientas sudut tertentu. Ciri orde dua umumnya digunakan untuk membedakan tekstur mikrostruktur (pola local dan perulangan tidak begitu jelas). Ciri orde dua antara lain: *Angular Second Moment*, *Contras*, *Correlation*, *Variance*, *Invers Different Moment*, dan *Entropy*. Analisis tekstore juga dapat dilakukan dalam domain frekuensi antara lain menggunakan *Filter Gabor*.



Gambar 2. 10 Contoh Tekstu Citra

#### 2.2.3.5. Ekstraksi Ciri Warna

Untuk membedakan suatu objek dengan warna dapat menggunakan nilai *hue* yang merupakan representasi dari cahaya tampak (Merah, Jingga, Kuning, Hijau, Biru, Ungun). Nilai *hue* dapat dikombinasikan dengan nilai *Saturation* dan *Value* yang merupakan tingkat kecerahan suatu warna. Untuk mendapatkan ketiga

nilai tersebut, perlu dilakukan konversi ruang warna citra yang semula *RGB* (*Red*, *Green*, *Blue*) menjadi *HSV* (*Hue*, *Saturation*, *Value*) melalui persamaan berikut:

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Perhitungan nilai *Hue*:

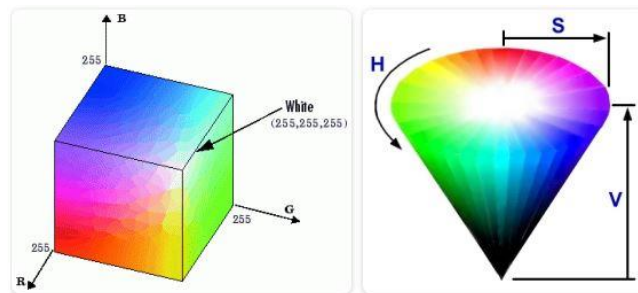
$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right), C_{max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right), C_{max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right), C_{max} = B' \end{cases}$$

Perhitungan nilai *Saturation*:

$$S = \begin{cases} 0, C_{max} = 0 \\ \frac{\Delta}{C_{max}}, C_{max} \neq 0 \end{cases}$$

Perhitungan nilai *Value*:

$$V = C_{max}$$



Gambar 2. 11 RGB Color Space dan HSV Color Space

#### 2.2.4. Operasi Morfologis

Pemrosesan gambar morfologis adalah kumpulan operasi yang terkait dengan bentuk atau morfologi fitur dalam suatu gambar. Operasi morfologis diterapkan untuk menghilangkan *noise* dan diskontinuitas dari latar depan yang diekstraksi. Operasi morfologis mempengaruhi bentuk, struktur, atau bentuk suatu objek. Diterapkan pada gambar biner. Mereka digunakan dalam pemrosesan pra atau pasca (penyaringan, penipisan, dan pemangkasan) atau untuk mendapatkan representasi atau deskripsi bentuk objek atau daerah. Dua operasi morfologi utama adalah *Dilation* dan *Erosion*. *Dilation* memungkinkan objek untuk mengembang, sehingga berpotensi mengisi lubang kecil dan menghubungkan objek terpisah. *Erosion* menyusutkan objek dengan mengikis batas-batasnya. Operasi ini dapat custom untuk suatu aplikasi dengan pemilihan yang tepat dari elemen penataan, yang menentukan dengan tepat bagaimana objek akan mengembang atau menyusut.

##### 2.2.4.1. Translasi

Translasi artinya sebuah citra didesens pada arah  $(x, y)$  dimana  $(x, y)$  adalah koordinat matriks. Operasi operasi translasi dinyatakan sebagai:

$$A_w = \{(a, b) + (x, y) : (a, b) \in A\}$$

#### 2.2.4.2. Dilasi

Operasi dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambah lapisan di sekeliling objek sehingga citra hasil dilasi cenderung menebal. Operasi dilasi akan melakukan proses pengisian pada citra asal yang memiliki ukuran lebih kecil dibandingkan *structuring element (strel)*. Dilasi  $A$  oleh  $B$  dinotasikan dengan  $A \oplus B$  dan didefinisikan sebagai:

$$A \oplus B = \bigcup_{x \in B} A_x$$

#### 2.2.4.3. Erosi

Operasi erosi adalah kebalikan dari operasi dilasi. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek sehingga citra hasil cenderung diperkecil menipis. Operasi erosi akan melakukan pengurangan pada citra asal yang lebih kecil dibanding elemen penstruktur (*strel*). Erosi  $A$  oleh  $B$  dinotasikan  $A \ominus B$  didefinisikan sebagai:

$$A \ominus B = \{w: B_w \subseteq A\}$$

$$A \ominus B = \bigcap_{b \in B} A_b$$

#### 2.2.4.4. Opening

Proses opening pada sebuah citra  $A$  oleh *strel*  $B$  dinotasikan dengan  $A \circ B$  dan didefinisikan sebagai proses erosi yang dilanjutkan dengan proses dilasi dimana kedua proses tersebut dilakukan secara berulang untuk semua titik  $((x,y))$ .

$$(A)_{opening} = A \circ B = (A \ominus B) \oplus B$$

### 2.2.5. *K Nearest Neighbor*

Dalam pengenalan pola, algoritma *k Nearest Neighbor* (*k- NN*) adalah metode non parametrik yang digunakan untuk kasifikasi dan regresi. Dalam kedua kasus, input terdiri dari *k* contoh pelatihan terdekat diruang fitur. Outpunya tergantung pada *k- NN* digunakan untuk klasifikasi atau regresi:

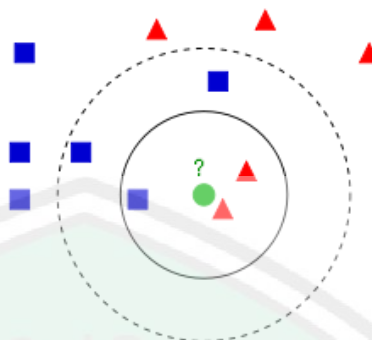
- Dalam klasifikasi *k-NN*, output adalah keanggotaan kelas. Suatu objek diklasifikasikan oleh suara pluralitas tetangganya, dengan objek yang ditugaskan ke kelas paling umum diantara tetangga terdekat. Jika  $k = 1$ , maka objek ditugaskan hanya untuk kelas tetangga terdekat itu.
- Dalam regresi *k-NN*, *output* adalah nilai properti untuk objek. Nilai ini adalah rata rata nilai *k* tetangga terdekat.

*k- NN* adalah jenis pembelajaran berbasis contoh (*Instace Based Learning*), atau pembelajaran malas (*Lazy Learning*), dimana fungsinya hanya didekati secara lokal dan semua perhitungan ditangguhkan hingga klasifikasi.

Baik untuk klasifikasi atau regresi, teknik yang berguna adalah menetapkan bobot pada kontribusi tetangga, sehingga tetangga yang lebih dekat berkontribusi lebih banyak ke rata rata daripada yang lebih jauh. Sebagai contoh, skema pembobotan umum terdiri dari memberikan masing masing tetangga  $1/d$ , dimana  $d$  adalah jarak ke tetangga.

Tetangga diambil dari satu set objek yang kelasnya (Untuk klasifikasi *k-NN*) atau nilai properti objek (untuk regresi *k-NN*) diketahui. Ini dapat dianggap sebagai pelatihan yang ditetapkan untuk algoritma, meskipun tidak diperlukan langkah pelatihan eksplisit.

### 2.2.5.1. Cara Kerja Algoritma $k$ -NN



Gambar 2. 12 Contoh Klasifikasi  $k$ - NN

Sampel uji (titik hijau) harus diklasifikasikan menjadi kotak biru atau segitiga merah. Jika  $k=3$  (lingkaran garis padat) itu ditugaskan untuk segitiga merah karena ada 2 segitiga dan hanya 1 persegi didalam lingkaran dalam. Jika  $k=5$  (lingkaran garis putus putus) itu ditugaskan ke kotak biru (3 kotak vs 2 segitiga didalam lingkaran luar).

Contoh pelatihan adalah vektor dalam ruang fitur multidimensi, masing masing dengan label kelas. Fase pelatihan algoritma hanya terdiri dari penyimpanan vektor fitur dan label kelas dari sampel pelatihan.

Dalam fase klasifikasi,  $k$  adalah konstanta yang ditentukan pengguna, dan vektor yang tidak berlabel (*query* atau *test point*) diklasifikasikan dengan menetapkan label yang paling sering diantara sampel pelatihan  $k$  yang terdekat dengan titik *query* itu..

Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan sampel latih. Pengklasifikasian tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik uji, akan ditemukan sejumlah  $k$  (titik training) yang paling dekat dengan titik uji. Klasifikasi

menggunakan voting mayoritas diantara klasifikasi dari  $k$  objek. Algoritma  $k$ -NN menggunakan klasifikasi ketetangga sebagai nilai prediksi dari sample uji yang baru. Matrik jarak yang umum digunakan untuk variable kontinu adalah *Euclidean Distance*. Untuk variable diskrit, seperti untuk klasifikasi teks, matrik lain dapat digunakan, seperti matrik tumpang tindih (atau *Hamming Distance*).

Algoritma  $k$ -NN menggunakan klasifikasi ketetangaan (*neighbor*) sebagai nilai prediksi dari *query instance* yang baru. Algoritma ini sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training* sampel untuk menentukan ketetanggaannya (Rizal, 2013).

Ketepatan algoritma  $k$ -NN sangat dipengaruhi ada atau tidaknya fitur-fitur yang tidak relevan atau bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik. Langkah-langkah untuk menghitung Algoritma  $k$ -NN:

1. Menentukan parameter  $k$  (jumlah parameter terdekat)
2. Menghitung kuadrat jarak *Euclidean* (*Query Instance*) masing-masing objek terhadap data sampel yang diberikan.
3. Mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak *Euclidean* terkecil.
4. Mengumpulkan kategori  $y$  (*Klasifikasi Nearest Neighbor*)
5. Dengan menggunakan kategori nearest neighbor yang paling mayoritas, maka dapat diprediksikan nilai *query instance* yang telah dihitung.

### 2.2.5.2. Perhitungan Jarak

Perhitungan jarak atau distance ( $d$ ), adalah bagian penting dalam algoritma *k-Nearest Neighbor*. Ada beberapa cara perhitungan jarak *k-NN* diantaranya:

#### 1. *Euclidean Distance*

*Euclidean Distance* adalah matrik jarak yang paling umum digunakan dalam set data dimensi rendah. Ini juga dikenal sebagai norma  $L_2$ .

*Euclidean Distance* adalah cara biasa dimana jarak diukur di dunia nyata. Rumus *Euclidean Distance*:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Dimana:

- $x_i$  = sampel data
- $y_i$  = data uji
- $i$  = variable data
- $d$  = jarak
- $n$  = dimensi data

Ide rumus ini dari rumus *Pythagoras*:

$$c = \sqrt{a^2 + b^2}$$

Sementara *Euclidean Distance* berguna dalam dimensi rendah, dan tidak bekerja dengan baik dalam dimensi tinggi dan untuk variable kategori. Kelemahan *Euclidean Distance* adalah mengabaikan kesamaan antara atribut. Setiap atribut diperlakukan berbeda dari semua atribut.

## 2. *Manhattan Distance*

Disebut berdasarkan pada kota *Manhattan* yang tersusun menjadi blok blok, sehingga sering disebut *City Block Distance*, juga sering disebut *Absolute Value*. Rumusnya adalah:

$$d(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|}$$

Rumus ini mencari jarak hanya dengan menjumlahkan semua selisih dari jarak  $x_i$  dan  $y_i$ .

## 3. *Minkowski Distance*

Adalah generalisasi jarak *Euclidean* dan *Manhattan*. Rumus ini diambil dari konsep aljabar dengan vektor berdimensi  $n$  dan  $r$ . Rumusnya adalah:

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^r \right)^{1/r}$$

Dimana:

- $x_i$  = sampel data
- $y_i$  = data uji
- $i$  = variable data
- $d$  = jarak
- $n$  = dimensi data
- $r$  = parameter

#### 4. *Chebychev Distance*

Algoritma ini adalah algoritma yang paling simpel dari dari semua algoritma mesin *learning*, rumus ini mencari jarak yang terbesar antara  $x_i$  dan  $y_i$ . Rumusnya adalah:

$$d(x, y) = \max_{i=1}^n |x_i - y_i|$$

#### 2.2.5.3. Kinerja Algoritma *k-NN*

Dipengaruhi oleh tiga faktor utama:

1. Fungsi jarak atau *metric* jarak digunakan untuk menentukan jarak tetangga terdekat.
2. Aturan keputusan digunakan untuk memperoleh klasifikasi dari *k-Nearest Neighbor*.
3. Jumlah tetangga yang digunakan untuk mengklasifikasikan contoh baru.

#### 2.2.5.4. Kelebihan dan Kekurangan Algoritma *k-NN*

Algoritma *k-NN* memiliki beberapa kelebihan antara lain:

- Sangat *non linier*
- Mudah dipahami dan diimplementasikan
- Tangguh terhadap data *training sample* yang *noisy*
- Efektif apabila data *training* sampelnya besar
- Memiliki konsistensi yang kuat
- *Asymptotically coorrect*

Algoritma  $k$ -NN juga memiliki beberapa kekurangan antara lain:

- Perlu menentukan  $k$  (jumlah tetangga terdekat)
- Tidak menangani nilai hilang (*missing value*) secara implisit
- Sensistif terhadap data pencilan (*outlier*)
- Rentan terhadap *variable* yang *non informative*
- Rentan terhadap dimensional yang tinggi
- Rentan terhadap perbedaan rentang variabel
- Pembelajaran berdasarkan jarak tidak jelas
- Nilai komputasi yang tinggi.

#### 2.2.6. *K Fold Cross Validation*

*K-fold cross validation* adalah salah satu metode untuk mengevaluasi kinerja *classifier*, *k-fold cross validation* mebagi data menjadi data *training* dan *testing* dalam  $K$  bagian. Metode *k-fold cross validation* bekerja dengan cara mempartisi himpunan data  $D$  secara acak menjadi  $k$  -fold (subhimpunan) yang saling bebas:  $f_1, f_2, f_3, \dots, f_k$ , sehingga masing masing *fold* berisi  $1/k$  bagian data (Suyatno, 2017). Sebagai gambaran, jika kita melakukan *5-Fold Cross-Validation* maka desain data eksperimennya sebagai berikut:

Tabel 2. 1 Ilustrasi *k-Fold Cross Validation*

K 1	test	train	train	train	train
K 2	train	test	train	train	train
K 3	train	train	test	train	train
K 4	train	train	train	test	train
K 5	train	train	train	train	test

Percobaan diatas adalah contoh ilustrasi dari *5 k-fold cross validation* yang artinya adalah melakukan percobaan sebanyak 5 kali tahapan. Dari hasil percobaan ini, kita akan mencatat nilai evaluasi performa dari model tersebut dengan menggunakan *confussion matrix*, kemudian menentukan nilai rata rata dari setiap percobaan. Maka akan ditemukan percobaan mana yang dapat dijadikan acuan dari penggunaan suatu model algoritma yang dipilih. K 1, yaitu menjadikan bagian partisi pertama menjadi data *testing* dan partisi lainnya menjadi data *training* dan seterusnya hingga K 5.

Tujuan *cross validation* adalah untuk menguji kemampuan model untuk memprediksi data baru yang tidak digunakannya, untuk menandai masalah seperti *overfitting* atau bias seleksi (Cawley & Talbot, 2010) dan untuk memberikan wawasan tentang bagaimana model akan digeneralisasikan ke dataset independen ( yaitu, dataset yang tidak diketahui misanya dari masalah nyata).

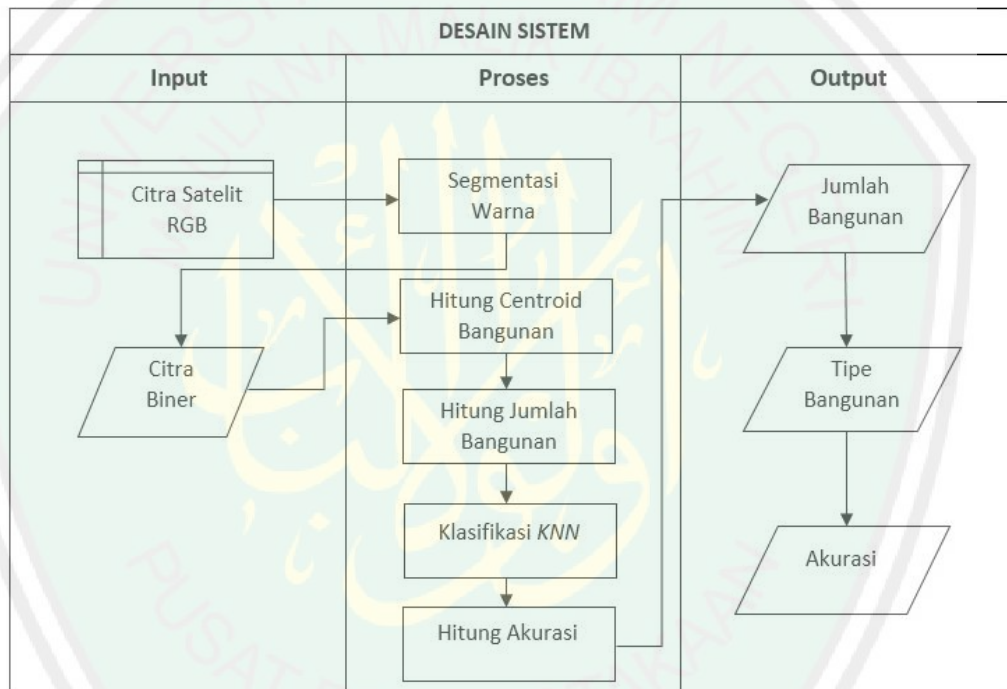
Satu putaran *cross validation* melibatkan partisi sebuah sampel dari data yang menjadi pelengkap subset, melakukan analisis pada suatu bagian (disebut *trainingset*), dan memvalidasi analisis pada subset lain (disebut validasi set atau pengujian set). Untuk mengurangi variabilitas, dalam sebagian besar metode, beberapa putaran validasi silang dilakukan dengan menggunakan partisi yang berbeda, dan hasil validasi digabungkan (dirata rata) selama putran untuk memberikan perkiraan kinerja prediksi model. Singkatnya *cross validation* menggabungkan (rata rata) ukuran kebugaran dalam prediksi untuk mendapatkan perkiraan kinerja prediksi model yang lebih akurat (Seni & Elder, 2010).

### BAB III

#### DESAIN PENELITIAN

##### 3.1. Desain Sistem

Untuk membangun aplikasi deteksi dan klasifikasi bangunan maka diperlukan desain sistem untuk penerapan aplikasi secara rinci dan teratur sesuai yang diinginkan. Berikut desain sistem yang diterapkan pada aplikasi ini.

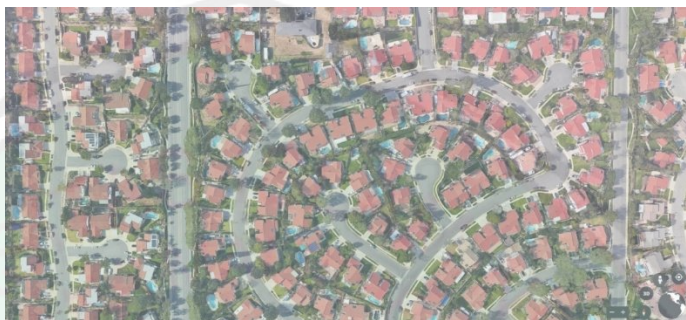


Gambar 3. 1 Desain Sistem

Pada desain sistem gambar 3.1 ditampilkan alur aplikasi mulai dari input, proses dan output sistem.

### 3.1.1. Citra Satelit *RGB*

Pada Tahap ini dilakukan *crooping* citra Satelit *Google Earth*, kemudian disimpan kedalam format (\*.jpg) yang selanjutnya akan diinputkan kedalam sistem untuk dilakukan proses segmentasi.



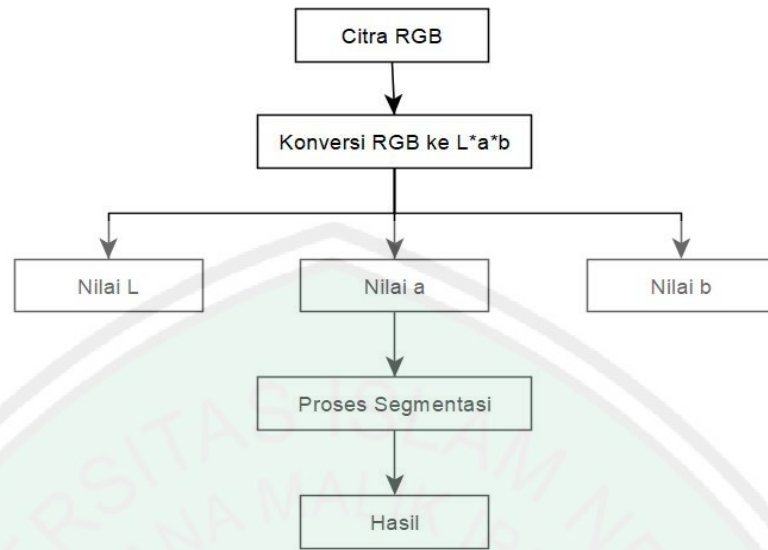
Gambar 3. 2 Citra Satelit Hasil Cropping

Citra satelit yang diambil oleh *Software Google Earth* merupakan hasil foto yang ditangkap dari satelit di luar angkasa. Pada *Software Google* ketinggian citra pengambilan citra diartikan sebagai nilai perbesaran gambar atau *zoom*. Satuan panjang yang digunakan untuk pengambilan ketinggian gambar pada *Google Earth* ini menggunakan satuan panjang meter. Dalam penelitian ini Citra satelit diambil dari ketinggian 700-800 meter. Ketinggian dan intensitas cahaya citra sangat berpengaruh terhadap kualitas gambar hasil *crooping*.

### 3.1.2. Segmentasi

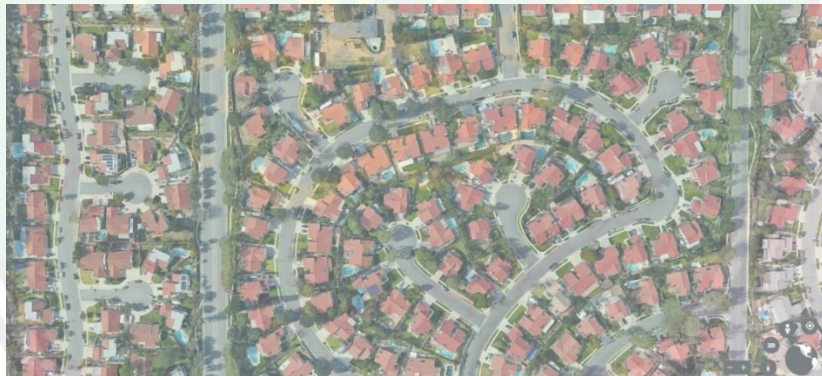
Pada tahap ini dilakukan segmentasi pada citra satelit setelah diinputkan kedalam sistem, proses segmentasi bertujuan untuk memisahkan antara objek *foreground* dengan *background*. Pada umumnya hasil segmentasi citra berupa citra biner dimana objek *foreground* yang dikehendaki berwarna putih (1), sedangkan *background* yang ingin dihilangkan berwarna hitam (0).

### 3.1.2.1. Proses Segmentasi



Gambar 3. 3 Proses Segmentasi

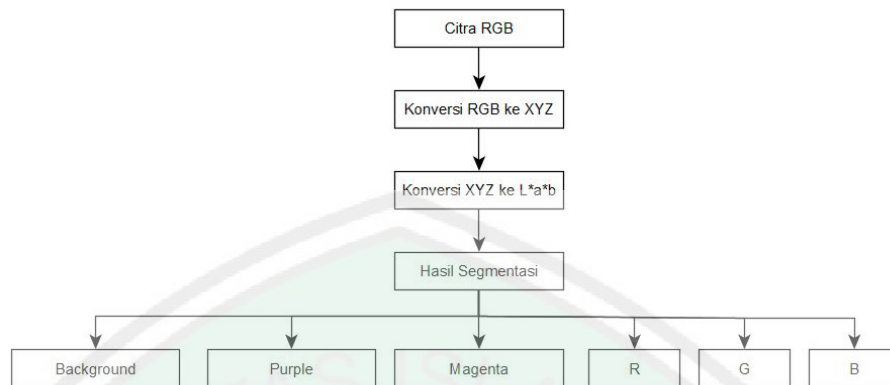
#### 1. Menampilkan citra *RGB*



Gambar 3. 4 Inputan Citra *RGB*

*Input citra rgb hasil Crooping dari Google Earth. Ruang warna RGB terdiri dari semua kemungkinan warna yang dibuat dengan kobinasi Red, Green dan Blue.*

## 2. Konversi warna



Gambar 3. 5 Proses Konversi Warna

Sebelum melakukan proses segmentasi citra *RGB* dikonversikan ke ruang warna *CIE L\*a\*b\**. *CIE L\*a\*b\** didefinisikan oleh cahaya (*L*), dan dimensi lawan warna *a* dan *b*, yang didasarkan pada kordinat ruang waran *XYZ* yang terkompresi.

Tranformasi perhitungan niali *RGB*:

$$R = \frac{r}{255}; \text{ indeks warna merah (red)} = \frac{R}{R+G+B}$$

$$G = \frac{g}{255}; \text{ indeks warna hijau (green)} = \frac{G}{R+G+B}$$

$$B = \frac{b}{255}; \text{ indeks warna biru (blue)} = \frac{B}{R+G+B}$$

Tranformasi *RGB* ke *CIELAB* dimulai dengan melakukan perhitungan sebagai berikut:

$$X = 0,412453R + 0,357580G + 0,180423B$$

$$Y = 0,212671R + 0,715160G + 0,072169B$$

$$Z = 0,019334R + 0,119193G + 0,950227B$$

Selanjutnya  $L^*a^*b^*$  didefinisikan sebagai berikut:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16$$

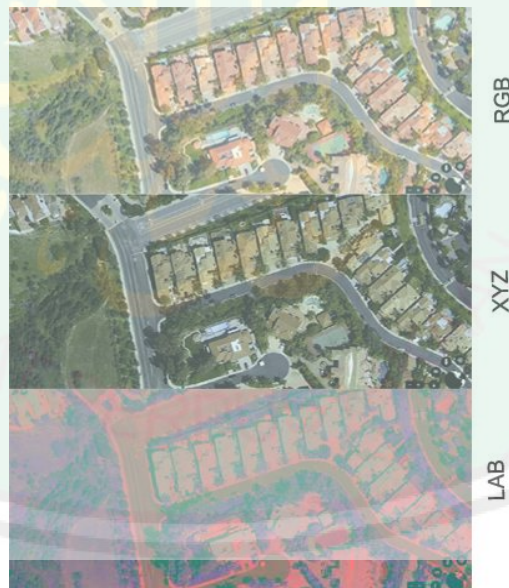
$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

Dalam hal ini,  $f(q)$  dihitung seperti berikut:

$$f(q) = \begin{cases} q^{1/3}, & \text{jika } q > 0,008856 \\ 7,787q + \frac{16}{116}, & \text{untuk yang lain} \end{cases}$$

$X_n Y_n Z_n$  diperoleh melalui  $R=G=B=1$  dengan jangkauan  $R,G,B$  berupa  $[0,1]$ .



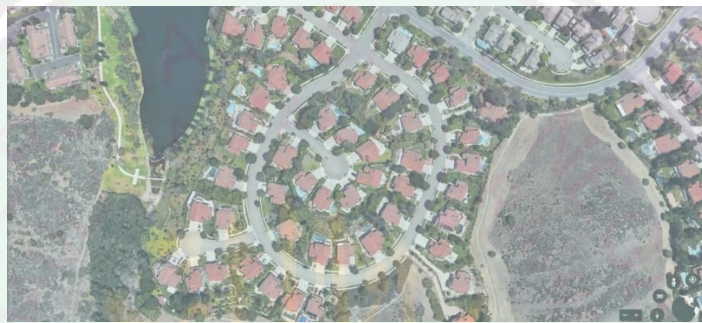
Gambar 3. 6 Hasil Konversi  $RGB$  ke  $LAB$

Segmentasi warna menggunakan warna  $L^*a^*b^*$ , menampilkan komposisi warna menjadi enam kelompok yaitu: *background*, kuning, magenta, ungu, merah dan hijau.

Segmentasi selanjutnya memisahkan objek *Foreground* dengan *Background* menggunakan metode *k Mean Clustering*. Proses *clustering* dilakukan dengan cara mengkonversikan ruang warna citra *RGB* menjadi *L\*a\*b*.

Komponen *a* dan *b* dari citra *L\*a\*b* digunakan sebagai nilai masukan dalam algoritma *k-Mean*.

a. Menampilkan citra *RGB*



Gambar 3. 7 Citra *RGB*

Inputan citra *RGB* kedalam sistem untuk di konversikan kedalam ruang warna *LAB*.

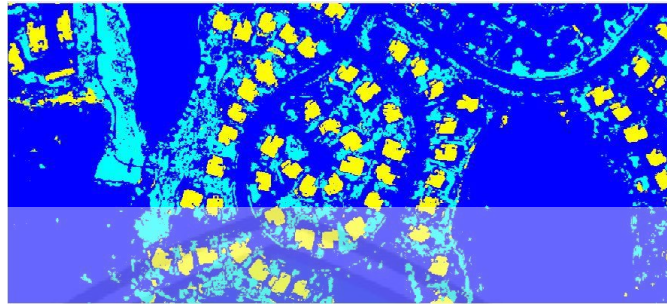
b. Konversi *RGB* ke *LAB*



Gambar 3. 8 Konversi *RGB* ke *L\*a\*b*

Setelah dikonversikan dari citra *RGB* ke *LAB* maka didapatkan citra *LAB* yang selanjutnya akan disegmentasi dengan membaginya menjadi beberapa kluster.

c. Hasil Segmentasi



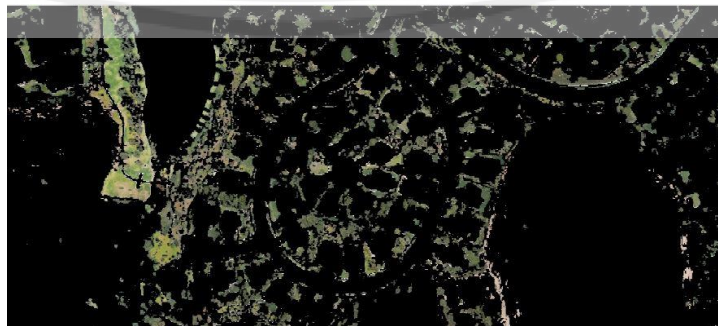
Gambar 3. 9 Citra Hasil Segmentasi

Hasil segmentasi menunjukkan citra *LAB* telah terbagi mejadi 3 cluster warna yang membedakan objek satu dengan yang lainnya.



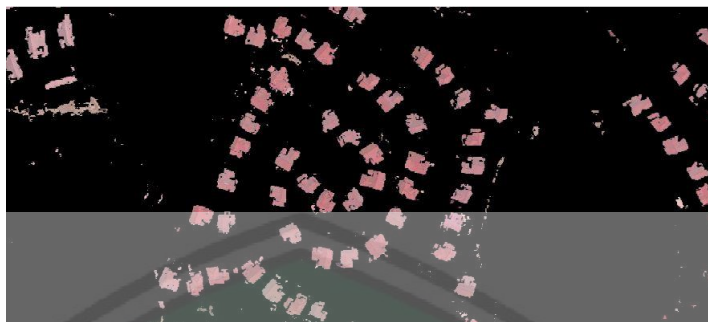
Gambar 3. 10 Hasil Segmentasi *Cluster 1*

Hasil segmentasi pada objek cluster 1 menunjukan objek jalan dan lahan dan objek objek yang memiliki luasan yang besar.



Gambar 3. 11 Hasil Segmentasi *Cluster 2*

Hasil segmentasi pada cluster 2 menunjukkan objek tumbuhan.



Gambar 3. 12 Hasil Segmentasi *Cluster 3*

Hasil segmentasi pada cluster 3 menunjukan objek bangunan

d. Memilih cluster



Gambar 3. 13 Cluster Terdapat *Foreground* Bangunan

Memilih kluster yang terdapat *foreground* (bangunan) Pemilihan didasarkan pada kluster yang memiliki luasan objek paling kecil. Setelah diperoleh citra *foreground* saja selanjutnya dilakukan proses operasi morfologi yaitu *Filling Holes* untuk menyempurnakan hasil segmentasi dan *bwareaopen* untuk menghilangkan *noise*.



Gambar 3. 14 Hasil Morfologi *Filling Holes* dan *Bwarea Open*

Hasil segmentasi berupa citra biner (bangunan yang terdeteksi) yang selanjutnya akan dilakukan ekstraksi ciri bentuk citra untuk menghitung jumlah dan klasifikasi tipe bangunan.

### 3.1.3. Ekstraksi Ciri

Proses pengenalan objek dalam citra umumnya membutuhkan suatu ciri yang dapat membedakan anatar objek satu dengan objek lainnya. Ektaksi ciri citra merupakan tahapan mengekstrak ciri atau informasi dari objek didalam citra yang ingin dikenali atau dibedakan dengan objek lainnya. Ciri yang telah diektrak selanjutnya digunakan sebagai parameter atau nilai masukan untuk membedakan antara objek satu dengan lainnya pada tahap identifikasi atau klasifikasi.

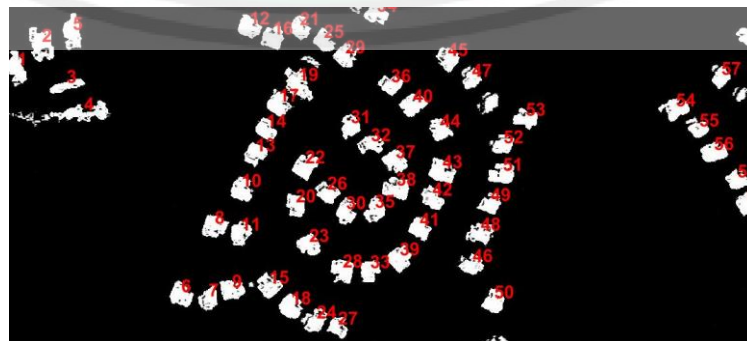
Ciri merupakan karakteristik unik dari suatu objek, ciri dibedakan menjadi ciri alami merupakan bagian dari gambar misalnya kecerahan dan tepi objek, sedangkan ciri buatan merupakan fitur yang diperoleh dengan operasi tertentu pada gambar misalnya histogram tingkat keabuan.

Citra disegmentasi dan didapatkan obejek bangunan hasil deteksi selanjutnya mengektraksi ciri masing masing objek dari citra hasil deteksi berdasarkan Luas, Keliling, *Eccentricity*, dan *Metric*.



Gambar 3. 15 Flowchart Ekstraksi Ciri

Citra biner hasil deteksi menunjukkan hasil bangunan terdeteksi yang akan diekstraksi ciri ukuran (Luas dan Keliling) dan bentuk (*Metric* dan *Eccentricity*) sebagai ciri yang akan membedakan objek satu dengan objek lainnya. Ciri ini akan digunakan sebagai acuan klasifikasi bangunan dengan metode *kNN* untuk mengklasifikasikan setiap bangunan yang terdeteksi.



Gambar 3. 16 Labelling Objek

Labeling objek dilakukan untuk menandai objek yang terdeteksi, kemudian menghitung jumlah objek (bangunan).

Tabel 3. 1 Luas, Keliling, *Eccentricity*, dan *Metrik* Hasil Ekstraksi

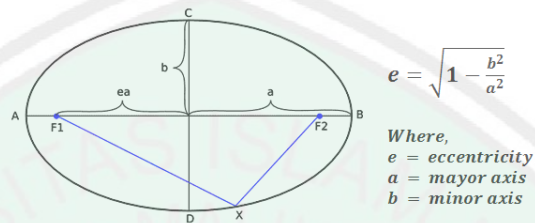
Object number = 1 Area = 2803 Perimeter = 1091,9483 Metric = 0,029541 Eccentricity = 0,98253	Object number = 2 Area = 948 Perimeter = 189,3381 Metric = 0,33231 Eccentricity = 0,82025	Object number = 3 Area = 1271 Perimeter = 197,7229 Metric = 0,40855 Eccentricity = 0,85386
Object number = 4 Area = 139 Perimeter = 54,3848 Metric = 0,59057 Eccentricity = 0,93575	Object number = 5 Area = 2842 Perimeter = 342,2325 Metric = 0,30492 Eccentricity = 0,95966	Object number = 6 Area = 774 Perimeter = 128,9533 Metric = 0,58491 Eccentricity = 0,74418
Object number = 7 Area = 1873 Perimeter = 289,463 Metric = 0,28091 Eccentricity = 0,90677	Object number = 8 Area = 2140 Perimeter = 256,0488 Metric = 0,41018 Eccentricity = 0,91055	Object number = 9 Area = 910 Perimeter = 124,8112 Metric = 0,73408 Eccentricity = 0,51193
Object number = 10 Area = 1516 Perimeter = 182,2082 Metric = 0,57382 Eccentricity = 0,61043	Object number = 11 Area = 1721 Perimeter = 218,4508 Metric = 0,45319 Eccentricity = 0,84461	Object number = 12 Area = 1017 Perimeter = 150,6102 Metric = 0,56341 Eccentricity = 0,55984

Luas (*Area*) merupakan banyak piksel yang menyusun suatu objek Pada citra biner dilakukan operasi *thresholding* untuk mempresentasikan luasan dari objek, Luas objek kemudian dihitung dengan cara menjumlahkan seluruh angka 1 (piksel berwarna putih) pada citra biner yang diperoleh. Luas objek dihitung dalam satuan piksel.

Keliling (*Perimeter*) merupakan banyak piksel yang berada pada boundary objek. Operasi morfologi dilakukan untuk mendapatkan boundaries dari citra biner. Kemudian dilakukan penghitungan keliling objek dengan cara

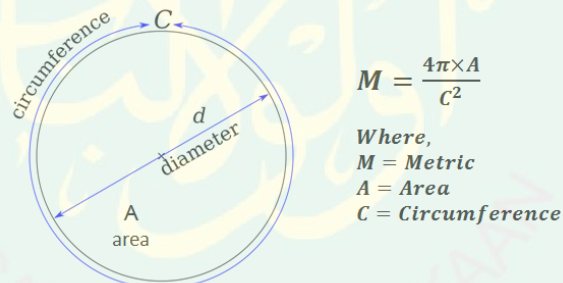
menjumlahkan seluruh angka 1 (piksel berwarna putih) pada citra biner yang diperoleh. Keliling objek dihitung dalam satuan piksel.

*Eccentricity* merupakan nilai perbandingan antara jarak *focil ellipsis minor* dengan *focil ellipsis mayor* suatu objek.



Gambar 3. 17 Rumus *Eccentricity*

*Metric* merupakan nilai perbandingan antara luas dan keliling.



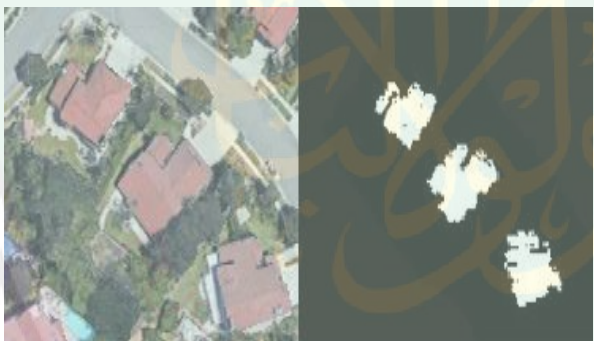


Gambar 3. 18 Rumus *Metric*

### 3.1.4. Klasifikasi *k*-Nearest Neighbor

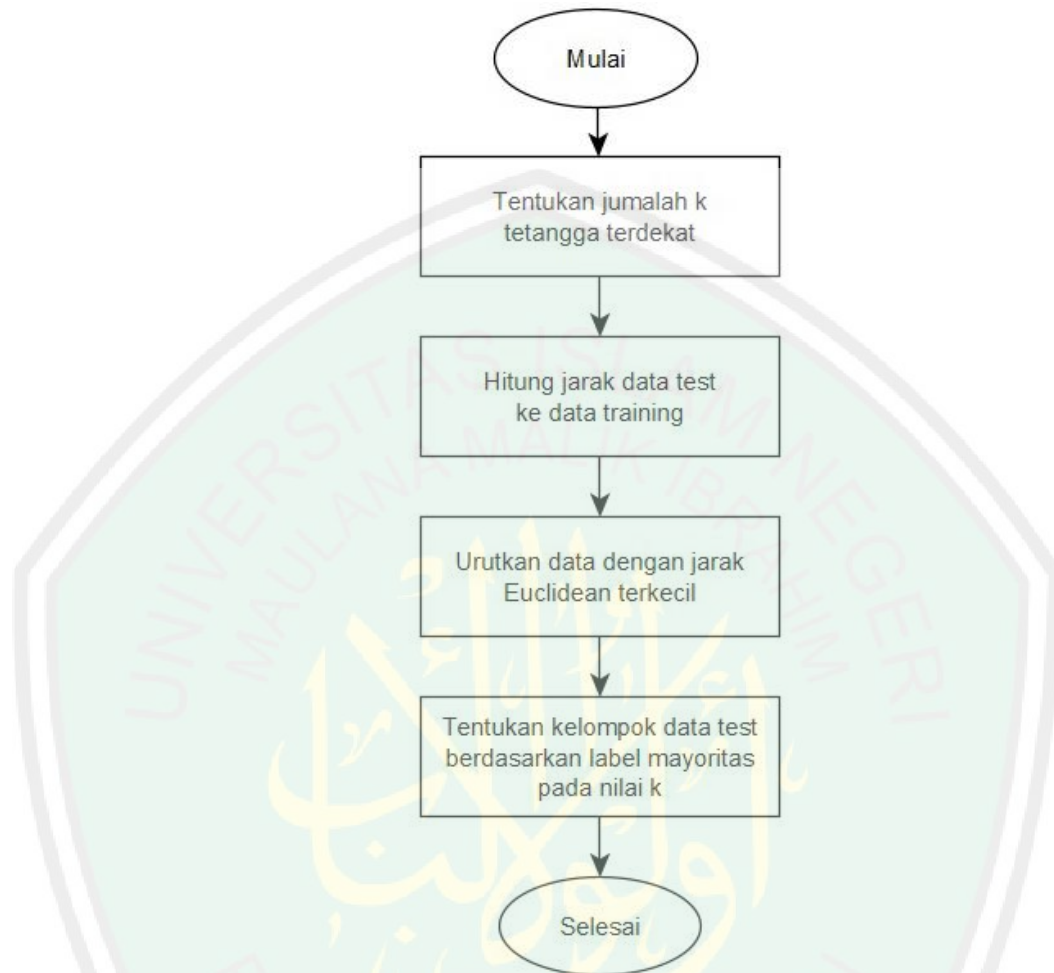
Pada tahap ini dilakukan klasifikasi tipe bangunan dengan menggunakan *k* Nearest Neighbor, data *training* dibagi menjadi 3 kelas bangunan yaitu tipe A (bangunan Besar), tipe B (bangunan sedang) dan tipe C (bangunan kecil).

Tabel 3. 2 Contoh Data Training Tipe Bangunan

	<p>Bangunan Tipe A=1 (Besar)</p>
	<p>Bangunan Tipe B=2 (Sedang)</p>
	<p>Bangunan Tipe C=3 (Kecil)</p>

Data training diambil dari citra satelit *Google Earth*, Setelah objek (bangunan) terdeteksi selanjut pengklasifikasian tipe bangunan menggunakan ciri objek yaitu: Luas dan Keliling. Klasifikasi tipe bangunan dengan *k-Nearest Neighbor* pada penelitian ini menggunakan perhitungan jarak *Euclidean Distance*.


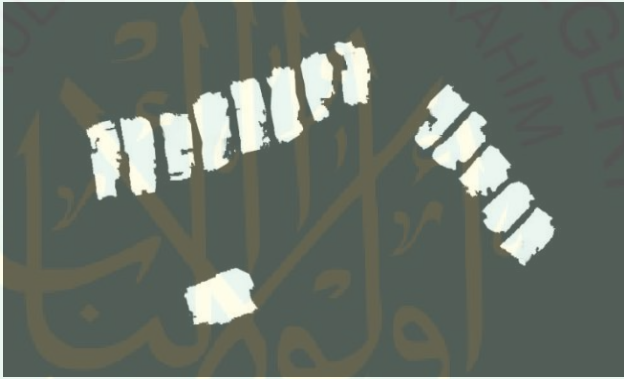
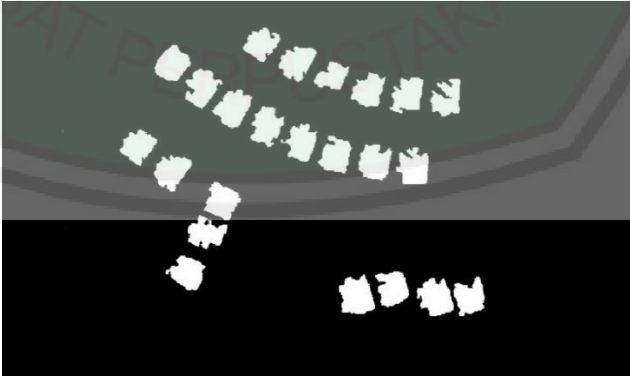
### 3.1.4.1. Flowchart Algoritma *k*-Nearest Neighbor



Gambar 3. 19 Flowchat Algoritma *k*-NN

Menentukan  $k$  tetangga terdekat adalah tahap pertama dalam klasifikasi *k*-Nearest Neighbor dalah menentukan parameter  $k$  tetatngga terdekat, misanyal  $k=3$ , berarti mencari jarak terdekat data uji terhadap 3 tetangga terdekat dari data trainingnya.

Tabel 3. 3 Sampel Data Latih Bangunan

Sampel Data latih Objek Bangunan Hasil deteksi dan Ekstraksi Ciri
 <p>Bangunan Tipe (A)</p>
 <p>Bangunan Tipe (B)</p>
 <p>Bangunan Tipe (C)</p>

*Sample* data latih adalah objek bangunan hasil deteksi citra satelit *Google Earth (rgb)* yang telah dikonversikan kedalam ruang warna biner untuk diekstraksi ciri tipe objeknya dan kemudian dikelompokkan berdasarkan kesamaan cirinya.

Tabel 3. 4 Sampel Data Training Tipe Bangunan

Data Training						
No.	Luas	Keliling	Metric	Eccentricity	Tipe	Kategori
1	8333	610,68	0,28	0,97	1	A
2	8786	603,69	0,30	0,97	1	A
3	7802	527,18	0,35	0,96	1	A
4	7855	519,91	0,37	0,96	1	A
5	8923	589,08	0,32	0,97	1	A
6	3104	264,79	0,56	0,69	2	B
7	3231	239,48	0,71	0,65	2	B
8	3196	238,51	0,71	0,77	2	B
9	3306	301,89	0,46	0,73	2	B
10	2940	253,61	0,57	0,80	2	B
11	1687	180,65	0,65	0,77	3	C
12	2157	209,28	0,62	0,61	3	C
13	2000	211,62	0,56	0,68	3	C
14	1834	193,54	0,62	0,68	3	C
15	1802	193,76	0,60	0,51	3	C
Data Uji						
1	2889	311,06	0,38	0,47	?	?

Data *training* diambil dari hasil ekstraksi ciri objek deteksi yaitu ciri Luas, Keliling, *Eccentricity* dan *Metric*nya kemudian dikelompokkan mejadi 3 kelompok tipe data bangunan berdasarkan cirinya. Data *training* ini akan menjadi acuan data uji untuk diklasifikasikan berdasarkan jarak terdekatnya. Data uji adalah data baru yang belum masuk kedalam kelompok, data uji diambil dari salah satu hasil deteksi kemudian diekstraksi cirinya untuk diuji dan diklasifikasi.

Dari hasil deteksi bangun dari citra *Satelit Google Earth*, setelah objek bangunan dideteksi maka selanjutnya adalah mengklasifikasikannya kedalam salah satu dari 3 tipe bangunan dengan mencari kecocokan cirinya dengan perhitungan *Euclidean distance* terhadap  $k$  mayoritas tetangga terdekat.

### 3.1.4.2. Menghitung *Euclidean Distance*

Tabel 3. 5 Tabel Pehitungan *Euclidean Distance*

No.	Euclidean Distance
1	$\sqrt{(8333 - 2889)^2 + (610,68 - 311,06)^2 + (0,28 - 0,38)^2 + (0,97 - 0,47)^2}$
2	$\sqrt{(8786 - 2889)^2 + (603,69 - 311,06)^2 + (0,30 - 0,38)^2 + (0,97 - 0,47)^2}$
3	$\sqrt{(7802 - 2889)^2 + (527,18 - 311,06)^2 + (0,35 - 0,38)^2 + (0,96 - 0,47)^2}$
4	$\sqrt{(7855 - 2889)^2 + (519,91 - 311,06)^2 + (0,37 - 0,38)^2 + (0,96 - 0,47)^2}$
5	$\sqrt{(8923 - 2889)^2 + (589,08 - 311,06)^2 + (0,32 - 0,38)^2 + (0,97 - 0,47)^2}$
6	$\sqrt{(3104 - 2889)^2 + (264,79 - 311,06)^2 + (0,56 - 0,38)^2 + (0,69 - 0,47)^2}$
7	$\sqrt{(3231 - 2889)^2 + (239,48 - 311,06)^2 + (0,71 - 0,38)^2 + (0,65 - 0,47)^2}$
8	$\sqrt{(3196 - 2889)^2 + (238,51 - 311,06)^2 + (0,71 - 0,38)^2 + (0,77 - 0,47)^2}$
9	$\sqrt{(3306 - 2889)^2 + (301,89 - 311,06)^2 + (0,46 - 0,38)^2 + (0,73 - 0,47)^2}$
10	$\sqrt{(2940 - 2889)^2 + (253,61 - 311,06)^2 + (0,57 - 0,38)^2 + (0,80 - 0,47)^2}$
11	$\sqrt{(1687 - 2889)^2 + (180,65 - 311,06)^2 + (0,65 - 0,38)^2 + (0,77 - 0,47)^2}$
12	$\sqrt{(2157 - 2889)^2 + (209,28 - 311,06)^2 + (0,62 - 0,38)^2 + (0,61 - 0,47)^2}$
13	$\sqrt{(2000 - 2889)^2 + (211,62 - 311,06)^2 + (0,56 - 0,38)^2 + (0,68 - 0,47)^2}$
14	$\sqrt{(1834 - 2889)^2 + (193,54 - 311,06)^2 + (0,62 - 0,38)^2 + (0,68 - 0,47)^2}$
15	$\sqrt{(1802 - 2889)^2 + (193,76 - 311,06)^2 + (0,60 - 0,38)^2 + (0,51 - 0,47)^2}$

Tabel perhitungan jarak pada table 3.5 menggunakan rumus perhitungan jarak *Euclidean Distance*:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Dimana:

- $dx_i$  = sampel data
- $y_i$  = data uji
- $i$  = variable data
- $d$  = jarak
- $n$  = dimensi data

Tabel 3. 6 Hasil Perhitngan *Euclidean Distance*

No.	Rangking	<i>Euclidean Dinstance</i>	Tipe
1	13	5452,24	A
2	14	5904,26	A
3	11	4917,75	A
4	12	4970,39	A
5	15	6040,40	A
6	2	219,92	B
7	4	349,41	B
8	3	315,46	B
9	5	417,10	B
10	1	76,83	B
11	10	1209,05	C
12	6	739,04	C
13	7	894,54	C
14	8	1061,53	C
15	9	1093,31	C

Tabel hasil perhitungan *Euclidean Distance* didapatkan dari hasil perhitungan jarak dari data uji terhadap data *training*.

### 3.1.4.3. Klasifikasi Data

Setelah hasil perhitungan jarak *Euclidean* didapatkan maka selanjutnya mengurutkan jarak terdekat data uji terhadap data *training* berdasarkan mayoritas nilai  $k$  kemudian mengelompokkannya kedalam tipe data bangunan.

Hasil pengujian data uji terhadap data *training* dengan nilai  $k=3, 5, 7$  dan  $9$ .

Tabel 3. 7  $k=3$ 

Ranking	<i>Euclidean distance</i>	Tipe
1	76,83	B
2	219,92	B
3	315,46	B

Berdasarkan hasil uji dengan  $k=3$ , mayoritas jarak terpendek terhadap label  $k$  adalah kategori  $B$  maka data uji masuk kedalam kelompok data Tipe  $B$  berdasarkan *Euclidean Distance*.

Tabel 3. 8  $k=5$ 

Ranking	Euclidean Distance	Tipe
1	76,83	B
2	219,92	B
3	315,46	B
4	349,41	B
5	417,10	B

Berdasarkan hasil uji dengan  $k=5$ , mayoritas jarak terpendek terhadap label  $k$  adalah kategori  $B$ , maka data uji termasuk dalam kelompok Tipe  $B$  berdasarkan *Euclidean Distance*.

Tabel 3. 9  $k=7$ 

Ranking	<i>Euclidean Distance</i>	Tipe
1	76,83	B
2	219,92	B
3	315,46	B
4	349,41	B
5	417,10	B
6	739,04	C
7	894,54	C

Berdasarkan hasil uji dengan  $k=7$ , mayoritas jarak terpendek terhadap label  $k$  adalah kategori  $B$ , maka data uji termasuk dalam kelompok Tipe  $B$  berdasarkan *Euclidean Distance*.

Tabel 3. 10  $k=9$ 

Ranking	<i>Euclidean Distance</i>	Tipe
1	76,83	B
2	219,92	B
3	315,46	B
4	349,41	B
5	417,10	B
6	739,04	C
7	894,54	C
8	1061,53	C
9	1093,31	C

Berdasarkan hasil uji dengan  $k=9$ , mayoritas jarak terpendek terhadap label  $k$  adalah kategori B, maka data uji termasuk dalam kelompok Tipe B berdasarkan *Euclidean Distance*.

Jadi berdasarkan nilai  $k=3$ ,  $k=5$ ,  $k=7$ , dan  $k=9$  menunjukkan mayoritas jarak terpendek terhadap tipe data B, maka dapat disimpulkan data uji termasuk kedalam kelompok bangunan Tipe B.

### 3.1.5. Perhitungan Akurasi

*Confusion matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Evaluasi dengan *confusion matrix* menghasilkan nilai akurasi, presisi dan *recall*. Akurasi dalam klasifikasi adalah persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi (Han & Kamber, 2006). Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar (Powers, 2011).

Tabel 3. 11 *Confusion Matric Model*

		Nilai Sebenarnya	
		True	False
Nilai Prediksi	True	<i>TP</i> (True Positive) Correct result	<i>FP</i> (False Positive) Unexpected Result
	False	<i>FN</i> (False Negative) Missing Result	<i>TN</i> (True Negative) Correct absence of result

Rumus untuk menghitung Akurasi, presisi dan *recall* pada *confusion matrix* adalah sebagai berikut (Gorunescu, 2011) :

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\%$$

Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\%$$

*Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual.

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

*Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

## BAB IV

### UJICOBA DAN PEMBAHASAN

Pada Bab ini akan dijelaskan rangkaian uji coba dan pembahasan dari hasil penelitian yang telah dilakukan. Uji coba dilakukan untuk melihat sejauh mana keberhasilan metode yang diterapkan, sedangkan pembahasan dilakukan dengan melakukan analisa hasil uji coba agar memperoleh kesimpulan dan saran untuk penelitian selanjutnya.

#### 4.1. Implementasi

Lingkungan implementasi deteksi dan klasifikasi bangunan pada citra satelit menggunakan *k- Nearest Neighbor* terbagi menjadi lingkungan *hardware* dan lingkungan *software*.

##### 4.1.1. Software

Perangkat lunak atau *software* yang digunakan untuk pembuatan dan pengujian sistem sebagai berikut:

- a) Sistem Operasi *Windows* 10 64 bit
- b) Bahasa Pemrograman *MATLAB* (R2015a)
- c) Program Citra Satelit *Google Earth*

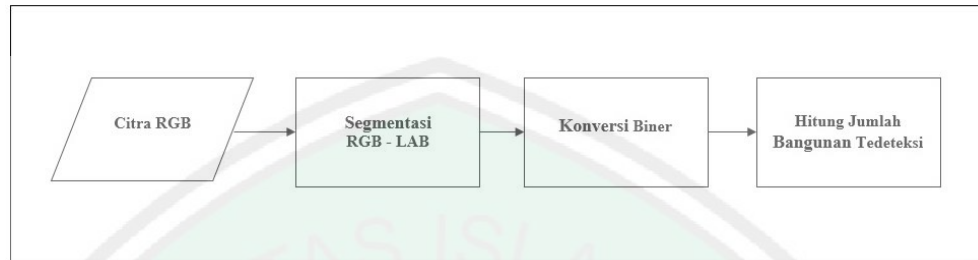
##### 4.1.2. Hardware

Perangkat keras atau *hardware* yang digunakan untuk pembuatan dan pengujian sistem adalah sebagai berikut:

- a) *Processor* : *Intel(R) Core(TM) i3-3110M*
- b) *Memory* : RAM DDR3 6,00 GB
- c) *Harddisk* : 500 GB

## 4.2. Uji Coba Deteksi Bangunan

Pada proses pengujian deteksi bangunan ini digunakan metode segmentasi warna untuk mendeteksi objek bangunan pada citra satelit.



Gambar 4. 1 Blok Deteksi Bangunan

Langkah langkah uji coba deteksi bangunan sebagai berikut:

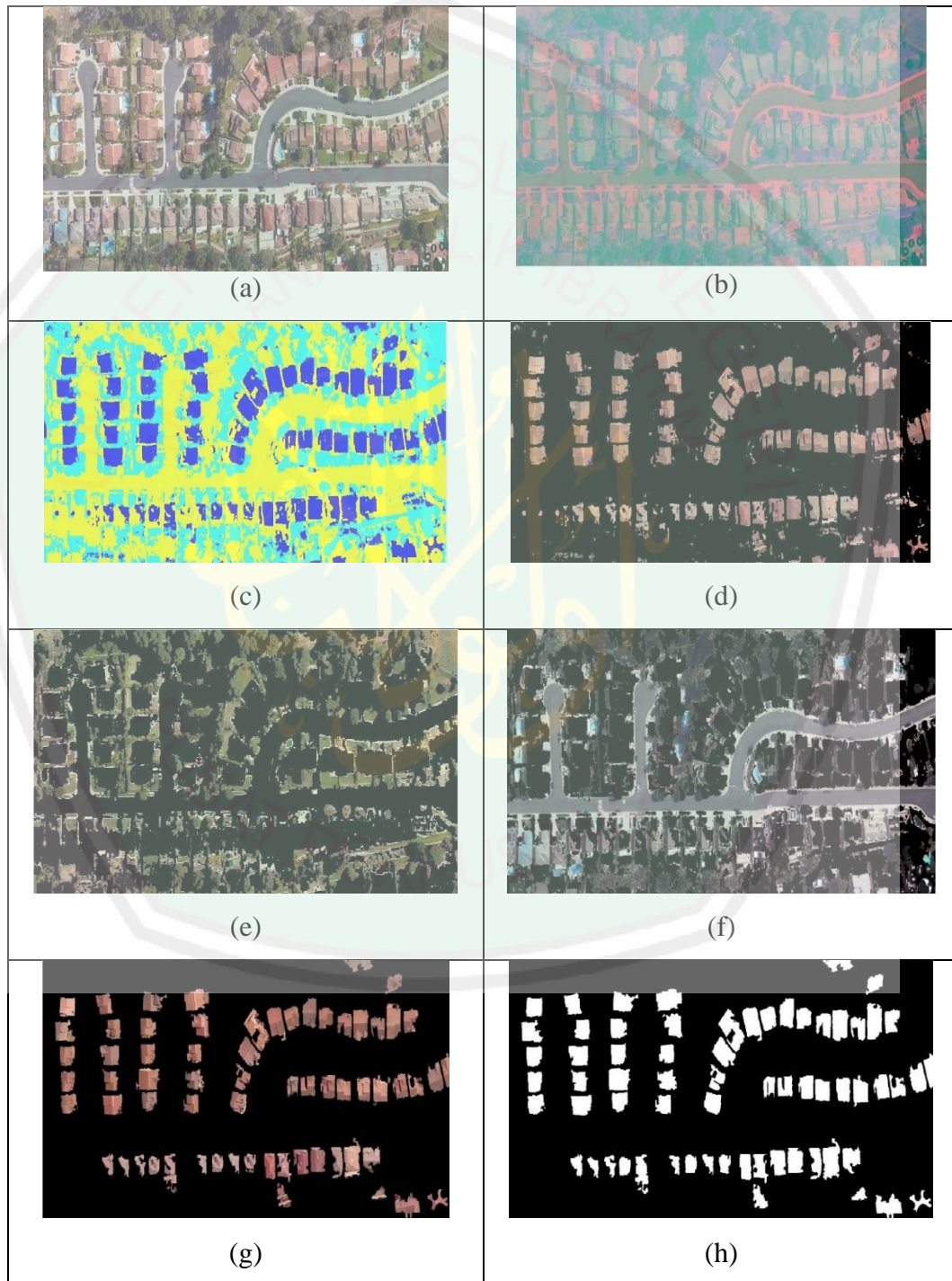
- a) Input citra satelit: citra satelit diambil dari *google earth* kemudian dilakukan *crooping* dan diinputkan kedalam sistem.
- b) Segmentasi warna: citra satelit *RGB* dikonversikan kedalam warna *CIE LAB*, kemudian *CIE LAB* dibagi menjadi beberapa kluster. Setipa kluster dianalisa untuk mencari kluster target (kluster bangunan).
- c) Konversi Biner: Setelah kluster bangunan ditemukan maka dilakukan konversi ke biner dan dilakukan perbaikan kualitas citra.
- d) Perhitungan Jumlah Bangunan: Objek yang terdeteksi dilabeling kemudian dihitung jumlah objek bangunan.

### 4.2.1. Citra Hasil Segmentasi

Segmentasi citra dilakukan setelah citra uji (citra bangunan *RGB*) diinputkan ke dalam sistem, kemudian citra *RGB* akan dikonversikan kedalam *CIE LAB*, warna *CIE LAB* akan dibagi menjadi beberapa cluster warna. Pada setiap cluster akan diidentifikasi objek bangunan. Setelah objek bangunan ditemukan, cluster

bangunan akan dikonversikan ke dalam ruang warna biner dan dilakukan perbaikan hasil deteksi selanjutnya melakukan labeling objek untuk menghitung jumlah objek bangunan terdeteksi.

Tabel 4. 1 Hasil Segmentasi Citra







Pada table hasil segmentasi citra di atas:

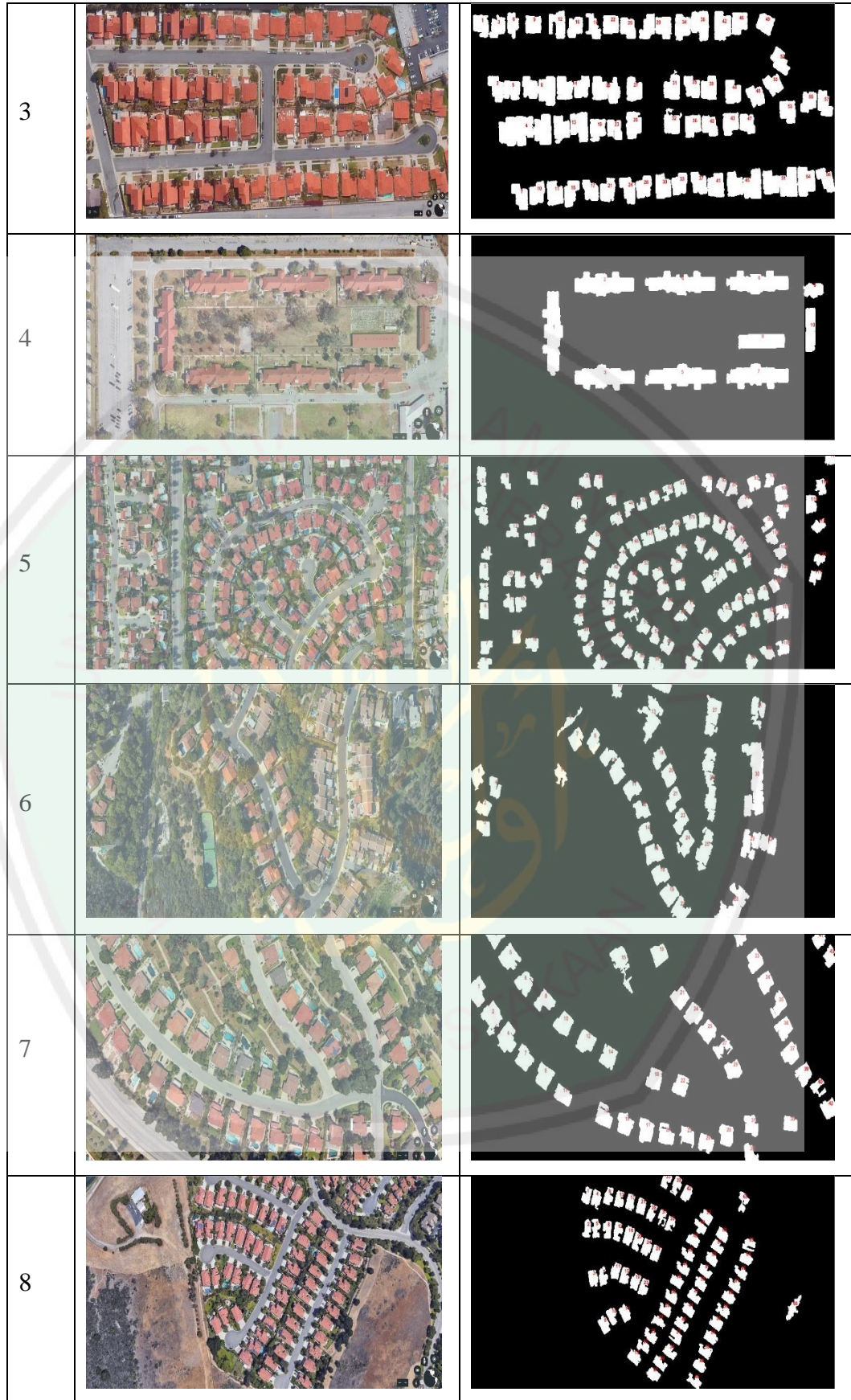
- a) Citra satelit *RGB* yang di inputkan kedalam sistem
- b) Konversi warna *RGB* ke *CIE LAB*
- c) *Labelling index* pada *CIE LAB*
- d) Objek pada *cluster 1*
- e) Objek pada *cluster 2*
- f) Objek pada *cluster 3*
- g) Objek Bangunan yang terdeteksi (*rgb*)
- h) Citra bangunan Biner setelah mengalami perbaikan hasil deteksi

#### 4.2.2. Citra Hasil Deteksi Bangunan

Objek bangunan hasil deteksi dari citra satelit dan akurasi deteksi akan ditampilkan pada pembahasan ini.

Tabel 4. 2 Tabel Hasil Deteksi Bangunan

No.	Citra Uji	Hasil Deteksi Bangunan
1		
2		





Pada table 4.1 ditampilkan hasil deteksi bangunan setelah melalui proses segmentasi sehingga objek bangunan dapat dihitung.

Nilai akurasi dibutuhkan untuk mengetahui seberapa besar ketepatan sistem dalam mendeteksi jumlah bangunan. Nilai akurasi dapat dinyatakan dalam satuan %. Adapun perhitungan untuk mencari nilai akurasi yaitu:

$$\text{Akurasi} = \frac{(\text{jumlah objek bngunan} + \text{jumlah objek bukan bangunan})}{\text{jumlah total objek}} \times 100\%$$

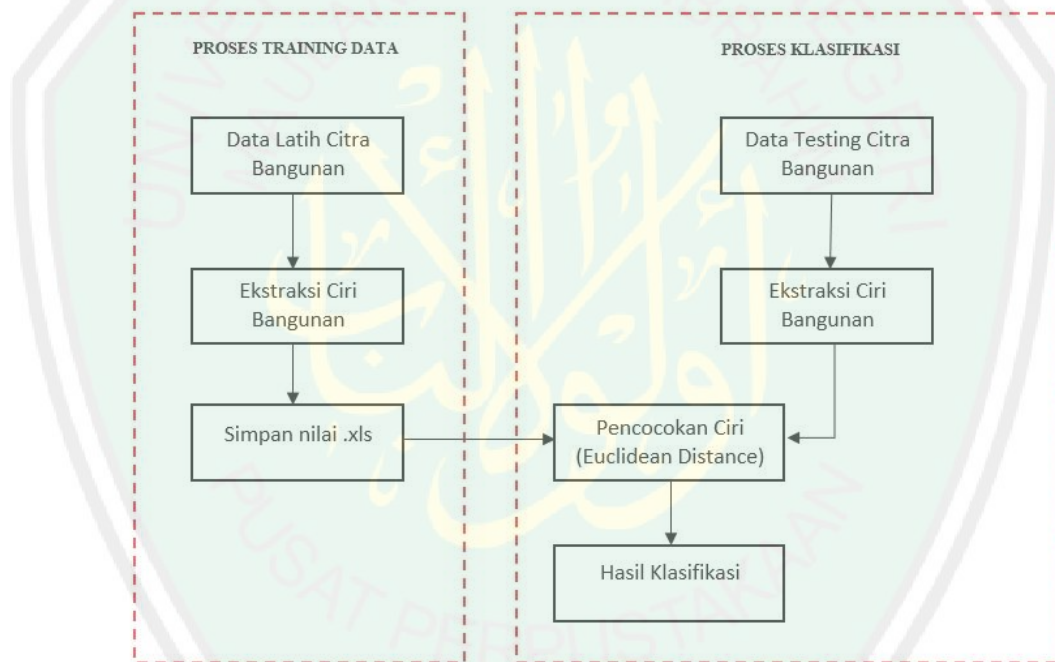
Tabel 4. 3 Akurasi Deteksi Sistem

No.	Objek	Jumlah Real	Jumlah Deteksi	Akurasi (%)
1	Gambar 1	70	65	92.8%
2	Gambar 2	57	55	96.4%
3	Gambar 3	66	57	86.3%
4	Gambar 4	10	10	100%
5	Gambar 5	137	115	83.9%
6	Gambar 6	40	32	80%
7	Gambar 7	46	43	93.4%
8	Gambar 8	80	66	82.5%
9	Gambar 9	50	48	96%
10	Gambar 10	63	61	96.8%
Akurasi rata rata (%)				90.8%

Akurasi rata rata sistem dalam mendeteksi bangun adalah 90.8% dengan menggunakan  $areaopen=700$  piksel yang berarti menghilangkan citra hasil deteksi yang bernilai piksel dibawah 700 piksel, sehingga dianggap sebagai *noise* kemudian dihilangkan.

### 4.3. Proses Klasifikasi Bangunan

Pada proses klasifikasi bangunan ini objek terdeteksi akan diklasifikasikan berdasarkan nilai ekstraksi cirinya ukurannya yaitu nilai Luas, Keliling, dengan cara mengukur jarak data uji dengan data latih.



Gambar 4. 2 Diagram Blok Klasifikasi sistem

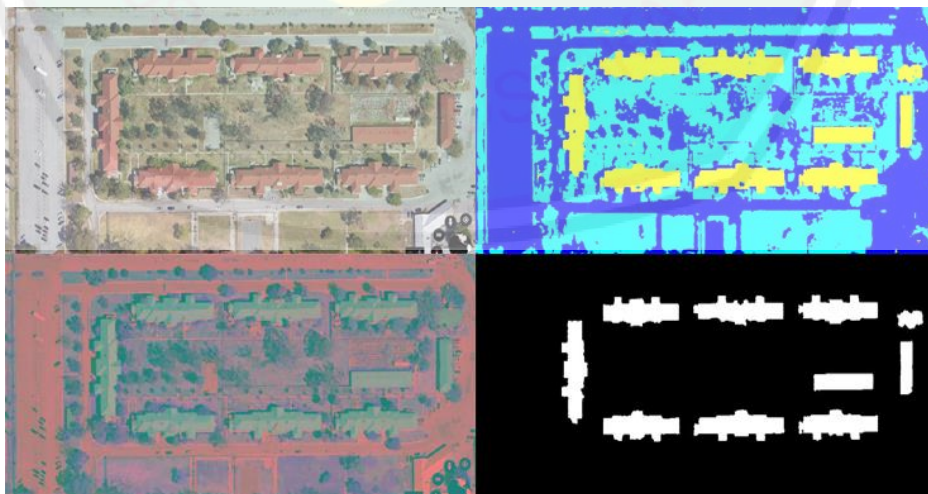
Pada diagram blok diatas akan dijelaskan proses klasifikasi bangun pada sistem sebagai berikut:

- Data latih citra bangunan merupakan data yang *dicrooping* dari *Google Earth* kemudian disimpan dalam format *.jpg*.

- Ekstraksi ciri bangunan merupakan proses mendapatkan ciri bangunan setelah dilakukan proses segmentasi gambar.
- Simpan nilai *.xls* adalah proses penyimpanan nilai ciri ekstraksi objek kedalam format *.xls* yang akan digunakan sebagai nilai referensi dan akan digunakan untuk pencocokan nilai dari data testing.
- Data *testing* adalah data baru hasil ekstraksi sistem yang akan diuji kecocokannya dengan data *training*.
- Pencocokan ciri adalah proses perhitungan jarak terdekat (*Euclidean distance*) data *testing* dengan data *training* untuk mencari nilai jarak terkecil dan pengklasifikasian menggunakan metode *k Nearest Neighbor* berdasarkan mayoritas jumlah tetangga terdekatnya.

#### 4.3.1. Ekstraksi Ciri Objek Deteksi

Pada tahap ini akan dilakukan ekstraksi ciri pada objek yang terdeteksi. Ciri yang dideteksi yaitu nilai Luas, Keliling, *Metric*, dan *Eccentricity*.



Gambar 4. 3 Gambar Uji

Pada gambar 4.3 objek akan diekstraksi cirinya setelah di konversikan ke warna biner dan hasil ekstraksi akan di tampilkan pada table hasil ekstraksi ciri.

Tabel 4. 4 Tabel Hasil Ekstraksi Ciri

Objek	Luas	Keliling	Meric	Eccentricity
1	10667	676,4163	0,293	0,9711
2	9576	591,8061	0,3436	0,9606
3	9606	588,1909	0,3489	0,9601
4	10174	677,4041	0,2786	0,9751
5	10801	657,0782	0,3144	0,9722
6	9260	607,1198	0,3157	0,9661
7	9539	622,333	0,3095	0,9637
8	5805	386,7696	0,4876	0,9703
9	2170	218,2082	0,5727	0,8221
10	3826	322,0244	0,4636	0,9652

Pada table hasil ekstraksi terdapat 10 objek yang terdeteksi nilai Luas objek, nilai Keliling objek, nilai *Metric* objek dan nilai *Eccentricity* objek. Nilai ekstraksi ciri yang terdeteksi akan digunakan sebagai nilai uji untuk pengklasifikasian objek terhadap data latih menggunakan metode *k Nearest Neighbor*.

#### 4.3.2. Uji Coba Akurasi Klasifikasi KNN

Pada tahan ini akan dilakukan beberapa tahap untuk menguji akurasi klasifikasi *k Nearest Neighbor* dalam mengklasifikasikan objek bangunan hasil deteksi dan ekstraksi ciri. Pengujian ini akan dilakukan dengan beberapa skenario pengujian.

##### 4.3.2.1. Dataset Uji Coba Klasifikasi KNN

Data yang digunakan dalam pengujian ini menggunakan dataset 3 kategori tipe bangunan hasil deteksi dan ekstraksi ciri yaitu tipe A, B dan C sebanyak 150 data dan telah dikategorikan berdasarkan ciri ukuran luas dan kelilingnya:

Tabel 4. 5 Tipe Bangunan A

Objek	Luas	Keliling	Metric	Eccentricity	Kategori	Tipe
1	5989	330,2914	0,6899	0,6391	1	A
2	5430	334,8356	0,6086	0,9163	1	A
3	6162	342,5757	0,6598	0,5436	1	A
4	5913	353,3625	0,5951	0,8293	1	A
5	5565	318,8772	0,6877	0,8577	1	A
...	...	...	...	...	...	...
50	12549	460,7767	0,7427	0,7013	1	A

Ciri bangunan tipe A dikategorikan dengan angka 1 sebanyak 50 objek bangunan.

Tabel 4. 6 Tipe Bangunan B

Objek	Luas	Keliling	Metric	Eccentricity	Kategori	Tipe
1	3922	280,0071	0,6286	0,8878	2	B
2	3021	230,0244	0,7175	0,782	2	B
3	3429	295,7473	0,4926	0,9315	2	B
4	3910	273,6224	0,6563	0,8558	2	B
5	3836	270,5513	0,6586	0,8584	2	B
...	...	...	...	...	...	...
50	3773	296,0904	0,5408	0,9176	2	B

Ciri bangunan tipe B dikategorikan dengan angka 2 sebanyak 50 objek bangunan.

Tabel 4. 7 Tipe Bangunan C

Objek	Luas	Keliling	Metric	Eccentricity	Kategori	Tipe
1	2315	208,7523	0,6676	0,509	3	C
2	1537	153,0955	0,8241	0,6472	3	C
3	2315	232,2914	0,5391	0,5007	3	C
4	2162	223,664	0,5431	0,9543	3	C
5	2501	251,0193	0,4988	0,6273	3	C
...	...	...	...	...	...	...
50	1854	214,3087	0,5073	0,6873	3	C

Ciri bangunan tipe C dikategorikan dengan angka 3 sebanyak 50 objek bangunan.

Berdasarkan data setiap dataset diatas menunjukkan informasi ciri dari 3 kategori bangunan yang terdeteksi yaitu luas, keliling, *eccentricity* dan *matric*.

#### 4.3.2.2. Skenario Pengujian 1

Tahap pertama dalam pengujian ini adalah proses klasifikasi dimana data uji bangunan hasil deteksi dan ekstraksi ciri yaitu luas dan keliling digunakan untuk mengklasifikasikan tipe bangunan dengan berdasarkan jarak terdekatnya terhadap data latihnya dengan menggunakan metode *k Nearest Neighbor*.

Pada pengujian ini akan mencari nilai *k* (nilai tertangga terdekat) terbaik untuk klasifikasi *k Nearest Neighbor*, proses pengujian dilakukan dengan menguji dataset bangunan sebanyak 150 tipe bangunan hasil ekstraksi yang telah dikategorikan berdasarkan ciri ukuran luas dan kelilingnya. Pengujian ini dilakukan dengan membagi dataset menjadi 2 bagian dengan perbandingan (80:20) yaitu 120 data tipe bangunan sebagai data latih dan 30 data tipe bangunan sebagai data uji yang diambil secara acak dari dataset. Klasifikasi tipe bangunan ini menggunakan fungsi *fitcknn* untuk mencari kesamaan karakteristik tipe bangunan berdasarkan jarak terdekat menggunakan perhitungan jarak *Euclidean Distance*.

Tabel 4. 8 Hasil Klasifikasi Knn

Data Uji			Target Klasifikasi		Pengujian dengan k-n				
Objek	Luas	Keliling	Kategori	Tipe	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$
1	5989	330,2914	1	A	1	1	1	1	1
2	5430	334,8356	1	A	1	1	1	1	1
3	6162	342,5757	1	A	1	1	1	1	1
4	5913	353,3625	1	A	1	1	1	1	1
5	5565	318,8772	1	A	1	1	1	1	1
6	5123	302,1493	1	A	1	1	1	1	1
7	8503	599,3036	1	A	1	1	1	1	1
8	5795	374,2325	1	A	1	1	1	1	1
9	7723	524,2498	1	A	1	1	1	1	1
10	7739	525,7056	1	A	1	1	1	1	1
11	3558	250,2082	2	B	2	2	2	2	2
12	3369	232,4924	2	B	2	2	2	2	2
13	3656	255,2792	2	B	2	2	2	2	2
14	2680	205,4386	2	B	3	3	3	3	3
15	2846	218,1665	2	B	2	2	2	2	2
16	3218	249,7645	2	B	2	2	2	2	2
17	2348	199,6812	2	B	3	3	3	3	3
18	2435	223,9239	2	B	2	2	3	3	3
19	4165	259,664	2	B	2	2	2	2	2
20	4223	289,1371	2	B	2	2	2	2	2
21	2315	208,7523	3	C	3	3	3	3	3
22	1537	153,0955	3	C	3	3	3	3	3
23	2315	232,2914	3	C	2	3	3	3	3
24	2162	223,664	3	C	3	3	3	3	3
25	2501	251,0193	3	C	3	3	3	2	2
26	2242	243,1198	3	C	3	3	3	3	3
27	1838	183,3381	3	C	3	3	3	3	3
28	1888	201,7645	3	C	3	3	3	3	3
29	1989	201,3209	3	C	3	3	3	3	3
30	1767	174,1249	3	C	3	3	3	3	3

Pada table hasil pengujian diatas didapatkan hasil klasifikasi menggunakan  $k=3$ ,  $k=5$ ,  $k=7$ ,  $k=9$ , dan  $k=11$ . Selanjutnya menghitung *accuracy* dari masing masing hasil klasifikasi dengan *confusion matrix*.

Tabel 4. 9 Rumus *Confussion Matrix*

		Kenyataan	
		<i>true</i>	<i>false</i>
Hasil Prediksi	<i>true</i>	TP	FP
	<i>false</i>	FN	TN

$$Accuracy = \frac{TP+TN}{TP+FP+FN} \times 100\%$$

Tabel 4. 10 *Confussion matrix k=3*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	8	2
Tipe C	0	1	9

Pada pengujian  $k=3$  terdapat 3 objek miss klasifikasi yaitu 2 objek tipe B yang di klasifikasikan ke tipe C dan 1 objek tipe C yang diklasifikasikan ke tipe B.

Tabel 4. 11 *Confussion Matrix k=5*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	8	2
Tipe C	0	0	10

Pada pengujian  $k=5$  terdapat 2 objek misklasifikasi yaitu 2 objek tipe B yang diklasifikasikan ke tipe C.

Tabel 4. 12 *Confussion Matrix k=7*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	7	3
Tipe C	0	0	10

Pada pengujian  $k=7$  terdapat 3 objek misklasifikasi yaitu 3 objek Tipe B yang diklasifikasikan ke Tipe C.

Tabel 4. 13 *Confussion Matrix k=9*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	7	3
Tipe C	0	1	9

Pada pengujian  $k=9$  terdapat 4 objek misklasifikasi yaitu 3 objek tipe B yang diklasifikasikan ke Tipe C dan 1 objek tipe C yang diklsifikasikan ke Tipe B.

Tabel 4. 14 *Confussion Matrix k=11*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	7	3
Tipe C	0	1	9

Pada pengujian  $k=11$  terdapat 4 objek misklasifikasi yaitu 3 objek tipe B yang diklasifikasikan ke Tipe C dan 1 objek tipe C yang diklsifikasikan ke Tipe B.

Tabel 4. 15 Akurasi Hasil Klasifikasi  $k$ 

Hasil Uji $k$	Akurasi
$k= 3$	90%
$k= 5$	93%
$k= 7$	90%
$k= 9$	86%
$k= 11$	86%
Rata rata akurasi	89%

Hasil akurasi terbaik dari pengujian klasifikasi  $k$  *Nearest Neighbor* adalah 93% menggunakan  $k=5$ . Maka klasifikasi pada sistem akan menggunakan  $k=5$  untuk  $k$  tetangga terdekatnya.

#### 4.3.2.3. Skenario Pengujian 2:

Pada tahap pengujian kedua ini akan menguji keakuratan model klasifikasi  $kNN$  dengan menggunakan *k-fold cross validation*. *Cross validation* merupakan metode untuk memvalidasi keakuratan sebuah model yang dibangun berdasarkan dataset tertentu. *K Fold Cross Validation* mengulang  $k$ -kali untuk membagi sebuah himpunan contoh secara acak menjadi  $k$  subset yang saling bebas, setiap ulangan disisikan satu subset untuk pengujian dan subset lainnya untuk pelatihan (Fu, 1994).

*Cross model* yang dibuat pada matlab:

[classreg.learning.partition.ClassificationPartitionedModel](#)

```
CrossValidatedModel: 'KNN'
PredictorNames: {'x1' 'x2'}
ResponseName: 'Y'
NumObservations: 150
KFold: 5
Partition: [1x1 cvpartition]
ClassNames: [1 2 3]
ScoreTransform: 'none'
```

Cross model akan memvalidasi model klasifikasi  $kNN$  dengan *PredictorNames* yaitu kolom  $x1=$  Luas dan  $x2=$  Keliling dan akan merespon  $Y=$

kategori bangunan yaitu kategori tipe A, B dan C dengan *Class names* [1 2 3]. Dataset yang diobservasi sebanyak 150 data tipe bangunan, dengan melakukan *kfold* = 5 atau 5 kali tahap pengujian.

*Cross model partition* pada matlab:

```
K-fold cross validation partition
NumObservations: 150
NumTestSets: 5
TrainSize: 120 120 120 120 120
TestSize: 30 30 30 30 30
```

Selanjutnya membuat partisi pengujian *k-fold* =5, yaitu dataset sebanyak 150 data tipe bangunan dibagi menjadi 5 bagian *trainingset* dengan setiap bagian *trainsize* 120 data , dan 5 bagian *testset* dengan setiap bagian *testsize* 30 data.

Tabel 4. 16 Pengujian *k-fold*=5

	Semua Data					
	Training data					Test data
	Fold1	Fold 2	Fold 3	Fold 4	Fold 5	} mencari parameter
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
evaluasi akhir						{test data

Pada tabel diatas dilakukan 5 kali tahap pengujian untuk mencari hasil kalsifikasi dari *kNN* dengan menggunakan *k Fold Cross Validation*.

Tabel 4. 17 Test *fold* 1

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	8	0
Tipe C	0	0	12

Pada pengujian *fold 1*, 30 kategori bangunan diklasifikasikan dengan benar, yaitu 10 tipe A, 8 tipe B dan 12 tipe C.

Tabel 4. 18 Test *fold 2*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	11	0	0
Tipe B	0	6	1
Tipe C	0	2	10

Pada pengujian *fold 2*, 27 bangunan dikategorikan dengan benar dan terdapat 3 misklasifikasi yaitu satu 1 bangunan tipe B diklasifikasi tipe C dan 2 tipe C diklasifikasi tipe B.

Tabel 4. 19 Test *fold 3*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	10	0	0
Tipe B	0	12	1
Tipe C	0	0	7

Pada pengujian *fold 3*, 29 bangunan dikategorikan dengan benar dan 1 bangunan misklasifikasi yaitu 1 bangunan tipe B yang di klasifikasikan ke tipe C.

Tabel 4. 20 Test *fold 4*

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	6	2	0
Tipe B	0	11	1
Tipe C	0	0	10

Pada pengujian *fold* 4, 27 bangunan dikategorikan dengan benar dan 3 bangunan misklasifikasi yaitu 2 bangunan tipe A diklasifikasikan tipe B dan bangunan 1 tipe B yang di klasifikasikan ke tipe C.

Tabel 4. 21 Test *fold* 5

Hasil prediksi	Kenyataan		
	Tipe A	Tipe B	Tipe C
Tipe A	11	0	0
Tipe B	0	9	1
Tipe C	0	1	8

Pada pengujian *fold* 5, 28 bangunan dikategorikan dengan benar dan 2 bangunan misklasifikasi yaitu 1 bangunan tipe B yang di klasifikasikan tipe C dan 1 bangunan tipe C diklasifikasikan tipe B.

Tabel 4. 22 Akurasi *k fold*

Skenario	Jumlah data		Akurasi
	Training	Testing	
<i>fold</i> 1	120	30	100%
<i>fold</i> 2	120	30	90%
<i>fold</i> 3	120	30	96%
<i>fold</i> 4	120	30	90%
<i>fold</i> 5	120	30	93%
Akurasi rata rata			93%

Pada table akurasi *k Fold Cross Validation* dengan *k fold* = 5 didapatkan hasil akurasi rata rata pengujian dari *k-fold* yaitu 93% ketepatan hasil validasi dari metode *k Nearest Neighbor*.

Tabel 4. 23 Perbandingan akurasi  $k$  NN dan  $k$  Fold

No.	Akurasi $k$ -NN		Akurasi $k$ -fold	
1	$k=3$	90%	<i>fold1</i>	100%
2	$k=5$	93%	<i>fold2</i>	90%
3	$k=7$	90%	<i>fold3</i>	96%
4	$k=9$	86%	<i>fold4</i>	90%
5	$k=11$	86%	<i>fold5</i>	93%
6	rata rata	89%	rata rata	93%

Pada table di atas menunjukkan bahwa akurasi klasifikasi  $k$  Nearest Neighbor dengan  $k=5$  menunjukkan tingkat akurasi paling optimal dalam mengklasifikasikan data, dan  $k$  Fold Cross Validation menunjukkan rata rata 93% ketepatan akurasi dari model klasifikasi  $k$  Nearest Neighbor.

#### 4.3.2.4. Skenario pengujian 3

Pada tahap pengujian ketiga ini setelah pengujian akurasi model klasifikasi  $k$  Nearest Neighbor dengan  $k$  Fold Cross Validation dan ditentukan  $k$  terbaik hasil pengujian yaitu  $k=5$ . Maka tahap selanjutnya akan menguji sistem deteksi dan klasifikasi bangunan yang telah dibangun pada matlab dengan data baru hasil deteksi bangunan pada citra satelit.

Sistem klasifikasi bangunan yang dibangun pada matlab menggunakan fungsi *fitcknn* untuk klasifikasi data.

Fungsi *knn* pada matlab menggunakan *syntax*:

```
Md1=fitcknn(X,Y,'NumNeighbors',5,'Standardize',1)
Class = predict(Md1,sample);
ClassificationKNN
  PredictorNames: {'x1' 'x2'}
  ResponseName: 'Y'
  ClassNames: [1 2 3]
  ScoreTransform: 'none'
  NumObservations: 150
  Distance: 'euclidean'
  NumNeighbors: 5
```

**Md1** adalah *classifier kNN* yang terlatih, dan beberapa propertinya muncul di *Command Window*. **Md1**, dikembalikan sebagai objek model *Clasification kNN*.

**Md1**= *fitcknn* (X, Y), mengembalikan model klasifikasi *k- Nearest Neighbor* berdasarkan data predictor **X** dan respon **Y**.

**X** adalah matrik numerik yang berisi dua pengukuran kelompok 137 data bangunan. **X**-*Predictor data (numeric matrix)* setiap baris sesuai dengan satu pengamatan juga dikenal sebagai *instance* atau contoh dan setiap kolom sesuai dengan satu variable predictor juga dikenal sebagai fitur. Panjang **Y** dan jumlah baris **X** harus sama.

**Y** adalah array sel vektor yang berisi kategori bangunan. **Y**- *Class Label*, label kelas ditentukan sebagai *categorical array, character array, string array, logical vector, numeric vector* atau *cell array of character vector*. Setiap baris **Y** mewakili klasifikasi baris *x* yang sesuai.

Selanjutnya menampilkan jarak terdekat hasil klasifikasi data uji dengan data latih menggunakan fungsi *knnsearch*, *Syntax*:

$$\text{Idx} = \text{knnsearch}(\text{X}, \text{Y})$$

**Idx**= *knnsearch*(X, Y) menemukan tetangga terdekat di X untuk setiap titik permintaan di Y dan mengembalikan indeks tetangga di **Idx** vektor kolom. **Idx** memiliki jumlah baris yang sama dengan Y.

**X** *input data (numeric matrix)*, input data ditentukan sebagai matrik numerik. Baris X sesuai dengan pengamatan dan kolom sesuai dengan variable.

**Y** *query point (numeric matrix)*, *query point* ditentukan sebagai matrik nnumerik. Baris Y sesuai dengan pengamatan, dan kolom sesuai dengan variable. Y harus memiliki jumlah kolom yang sama dengan X.

Langkah langkah pemrograman matlab untuk mengklasifikasikan bentuk suatu objek dalam citra digital adalah sebagai berikut:

1. Mempersiapkan data latih ekstraksi ciri objek bangunan, yaitu: 50 ciri objek bangunan tipe A (Bangunan Besar) yang dikategorikan ke nilai= 1, 50 ciri objek bangunan tipe B (Bangunan Sedang) yang dikategorikan ke nilai=2, 50 ciri objek bangunan tipe C (Bangunan Kecil) yang dikategorikan ke nilai=3, total 150 dataset tipe bangunan.
2. Mempersiapkan data uji berupa hasil ekstraksi ciri dari masing masing objek yang terdeteksi pada citra uji yang diproses didalam sistem.
3. Klasifikas Tipe bangunan dengan menggunakan metode *kNN* (*fitcknn*).
4. Mencari jarak terdekat masing masing objek terhadap data latih dengan menggunakan *kNN search*.
5. Menampilkan Hasil klasifikasi dan sebaran data klasifikasi.

Pengujian sistem deteksi dan klasifikasi bangunan dengan metode *knn* yang dibangun pada matlab sebagai berikut:

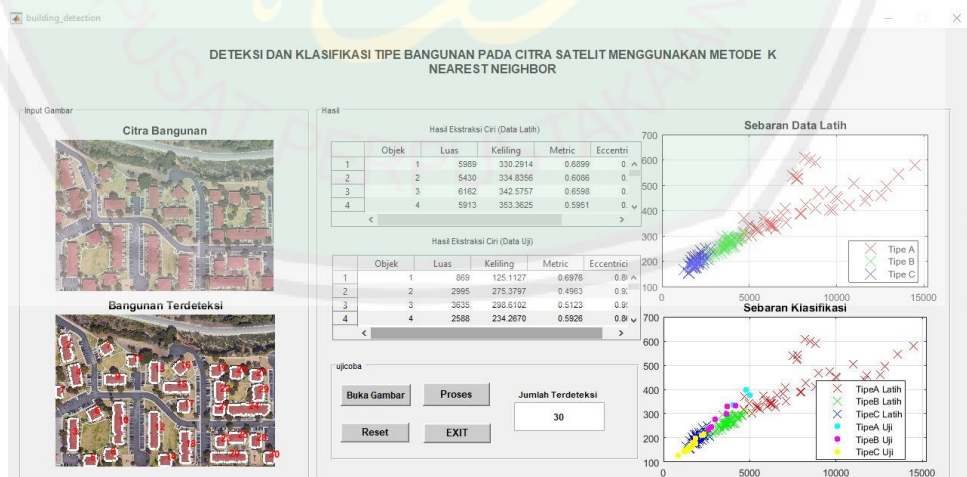
### A. Gambar Uji yang digunakan



Gambar 4. 4 Citra Bangunan

Citra satelit bangunan ini akan digunakan sebagai gambar uji untuk menguji kemampuan deteksi sistem dalam mendeteksi objek bangunan pada citra satelit dan hasil ekstraksi ciri yang didapatkan akan digunakan sebagai data uji untuk proses klasifikasi bangunan dengan menggunakan algoritma *knn* yang telah diterapkan didalam sistem sehingga didapatkan hasil akurasi klasifikasi sistem.

### B. Interface Uji Sistem



Gambar 4. 5 Interface Sistem

Pada interface ditampilkan jumlah bangunan terdeteksi, hasil ekstraksi ciri dan sebaran data bangunan. Dan proses segmentasi warna untuk mencari objek bangunan akan ditampilkan pada setiap figure yang muncul ketiga menekan *button* proses.

Proses yang ditampilkan pada *Figure*:

- a. *Figure 1* citra Rgb dikonversikan ke warna LAB
- b. *Figure 2* warna LAB dibagi menjadi 3 berdasarkan *cluster index*
- c. *Figure 3* menampilkan cluster 1
- d. *Figure 4* menampilkan cluster 2
- e. *Figure 5* menampilkan cluster 3
- f. *Figure 6* menampilkan hasil deteksi ( terdapat noise)
- g. *Figure 7* menampilkan hasil deteksi (tanpa noise)
- h. *Figure 8* menampilkan hasil konversi biner bangunan terdeteksi
- i. *Figure 9* menampilkan penyempurnaan hasil deteksi pada warna biner
- j. *Figure 10* menampilkan labeling objek terdeteksi.

Selanjutnya hasil klasifikasi dengan menggunakan *k Nearest Neighbor* akan ditampilkan pada *command window* matlab:

Tabel 4. 24 Hasil Deteksi dan Klasifikasi Sistem

Objek	Luas	Keliling	Klasifikasi	Tipe	Distance Luas	Distance Keliling
1	869	125,11	3	C	1,28	0,17
2	2995	275,38	2	B	3,02	0,23
3	3635	298,61	2	B	3,65	0,27
4	2588	234,27	2	B	2,58	0,21
5	1682	175,84	3	C	1,68	0,18
6	1734	181,24	3	C	1,74	0,20
7	1429	156,75	3	C	1,52	0,19
8	1674	179,05	3	C	1,68	0,18
9	2342	214,81	3	C	2,35	0,20
10	4791	398,69	1	A	4,70	0,37
11	1197	148,81	3	C	1,28	0,17
12	4165	333,68	2	B	4,18	0,30
13	2303	211,15	3	C	2,32	0,21
14	1687	168,43	3	C	1,68	0,18
15	2790	246,81	2	B	2,80	0,25
16	1465	155,24	3	C	1,52	0,19
17	1485	159,25	3	C	1,52	0,19
18	3688	329,14	2	B	3,74	0,28
19	1515	155,84	3	C	1,54	0,15
20	4023	337,48	1	A	4,06	0,26
21	1238	142,18	3	C	1,28	0,17
22	2184	211,30	3	C	2,16	0,22
23	1268	144,43	3	C	1,28	0,17
24	5001	375,52	1	A	4,94	0,34
25	1871	177,11	3	C	1,87	0,17
26	1529	159,50	3	C	1,54	0,15
27	1502	158,43	3	C	1,52	0,19
28	1575	159,94	3	C	1,55	0,16
29	1853	199,44	3	C	1,85	0,21
30	1402	147,36	3	C	1,28	0,17
Akurasi k Nearest Neighbor Sistem						
Correct Rate	Error Rate	Sensitivity		Specificity		
0,9667	0,0333	1		0,9900		

Pada table diatas hasil klasifikasi sistem ditampilkan pada *coman window matlab* yaitu 30 objek bangunan terdeteksi kemudian sistem mengklasifikasikan dengan *knn* berdasarkan jarak terdekat dengan 150 data latih menggunakan

*Euclidean Distance* dan parameter  $k=5$ . Akurasi *k Nearest Neighbor* sistem menunjukkna *Correct Rate: 96,67%*, *Error Rate: 3,33%*, *Sensitivity: 100%*, *Specificity: 99%*.

#### 4.4. Integrasi Dengan Al Quran

Penelitian ini menggunakan metode segmentasi warna untuk mendeteksi ciri objek yang terdeteksi kemudian metode *k Kearest Ninghbor* untuk mengklasifikasikan objek yang terdeteksi yaitu bangunan dengan ciri ukuran Luas dan Kelilingnya. Metode *kNN* bekerja dengan menghitung jarak terdekat data uji dengan data latih menggunakan perhitungan jarak *Euclidean* kemudian melakukan voting mayoritas nilai *k* terhadap jarak data uji ke jarak data latih.

Penelitian ini erat sekali kaitannya dengan proses pengamatan sehingga manusia bisa mengambil pelajaran dan manfaat dari apa yang diamatinya. Dalam Al Quran surat Ali Imran ayat 190-191 Allah berfirman:

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِّأُولِي الْأَلْبَابِ  
 ۝ الَّذِينَ يَذْكُرُونَ اللَّهَ قِيَمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ  
 السَّمَوَاتِ وَالْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَذَا بَطْلًا سُبْحَانَكَ فَقِنَا عَذَابَ النَّارِ ۝

190. *Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal*

191. *(yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadan berbaring dan mereka memikirkan tentang penciptaan langit dan*

*bumi (seraya berkata): "Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia, Maha Suci Engkau, maka peliharalah kami dari siksa neraka.*

Dalam Tafsir al Muyassar dijelaskan bahwa dengan mengamati penciptaan langit dan bumi yang terjadi tanpa ada contoh seblemunnya merupakan bukti dan petunjuk akan ke Esaan Allah bagi orang-orang yang berfikir. Sehingga manusia akan semakin bermanfaat dan semakin taat dengan menyadari bahwa tidak ada hal yang sia-sia yang diciptakan Allah.

Dalam penelitian ini objek deteksi yaitu bangunan diidentifikasi ciri-cirinya kemudian dikelompokkan dengan jarak terdekat menggunakan metode *kNN*. Metode *kNN* sangat erat kaitannya dengan hadis Nabi.

Dalam sebuah hadis Rasulullah shallallahu ‘alaihi wa sallam bersabda:

*“Agama seseorang tergantung dengan agama teman dekatnya. Hendaklah kalian memerhatikan, siapa yang dia jadikan teman dekatnya.” (HR. Abu Daud dan Tirmidzi, dishahihkan oleh Syaikh Al Albani dalam Silsilah Ash-Shahihah, no. 927)”*.

Pada hadis di atas kita bisa mengambil pelajaran bahwa cara kerja metode *KNN* hampir sama seperti penjelasan hadis di atas yaitu suatu objek akan digolongkan ke dalam objek terdekatnya. Begitu juga yang dijelaskan pada hadis yang lain.

Dari Abu Musa Al-Asy’ariy radhiyallahu ‘anhu berkata, Rasulullah shallallahu ‘alaihi wa sallam bersabda:

*“Permisalan teman duduk yang shalih dan buruk adalah seperti penjual minyak wangi dan tukang pandai besi. Adapun penjual minyak wangi, bisa jadi ia akan memberimu minyak wangi, atau kamu akan membeli darinya atau kamu akan*

*mendapat bau harum darinya. Adapun tukang pandai besi, bisa jadi ia akan membuat pakaianmu terbakar, atau kamu akan mendapat bau yang tidak sedap darinya.” (HR. Bukhari No. 2101, Muslim No. 2628)”.*

Pada ayat Al Quran dan Hadis diatas kita dapat mengambil pelajaran dari proses pengamatan dan pengelompokan, yaitu bagaiman kecendrungan suatu objek terhadap objek terdekatnya. Begitu juga dalam kehidupan sehari hari orang orang terdekat terutama lingkungan tempat tinggal akan mempengaruhi tumbuh kembang seseorang apakah akan menjadi lebih baik atau sebaliknya.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah disajikan pada bab sebelumnya maka kesimpulan dalam penelitian ini adalah sebagai berikut:

1. Sistem deteksi bangunan dibangun menggunakan segmentasi warna untuk mendeteksi objek bangunan pada citra satelit dengan rata rata akurasi deteksi bangunan 90,8% dan sistem klasifikasi dibangun dengan metode *k Nearest Neighbor* dengan perhitungan jarak *Euclidean Distance* menggunakan parameter  $k= 5$ .
2. Akurasi metode *k Nearest Neighbor* pada sistem klasifikasi tipe bangunan didapatkan dengan pengujian *k Fold Cross Validation* dengan  $k\ fold= 5$ , didapatkan tingkat akurasi rata rata *k Nearest Neighbor* 93% dalam ketepatan klasifikasi data *testing* dengan data *training*.

#### 5.2. Saran

Dari hasil penelitian deteksi dan klasifikasi tipe bangunan pada citra satelit menggunakan metode *k Nearest Neighbor* ada beberapa saran yang diberikan penulis untuk penelitian selanjutnya dengan tema yang sama:

1. Jarak ketinggian citra satelit dan kualitas warna citra yang akan *dicropping* dari *Google Earth* akan mempengaruhi ciri objek yang akan dideteksi.

2. Perbaikan kualitas citra hasil segmentasi mempengaruhi nilai hasil ekstraksi ciri setiap objek deteksi berupa penambahan dan pengurangan *pixel* objek yaitu luas, keliling, *metric* dan *eccentricity*
3. Menggunakan metode perhitungan jarak selain *Euclidean distance* antara lain *Manhatan distance*, *Minkowsky distance*, *Chebychev distance* dan lain lain dapat digunakan untuk meningkatkan akurasi deteksi,
4. Penambahan jumlah data *training* klasifikasi akan mempengaruhi tingkat akurasi klasifikasi *k Nearest Neighbor*.




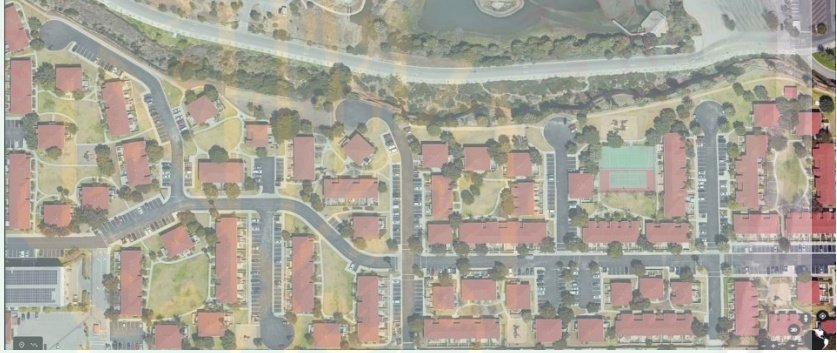

## DAFTAR PUSTAKA

- Akcay, H., & Aksoy, S. (2010). Building Detection Using Directional Spatial Constraints. *IEEE International Geoscience and Remote Sensing Symposium*, 1932-1935.
- Awrangjeb, M., Ravanbakhsh, M., & Fraser, C. (2010). Automatic detection of residential buildings using LIDAR data and Multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 457-467.
- Cawley, G. C., & Talbot, N. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, 2079-2107.
- Fu, L. (1994). *Neural Networks in Computer Intelligence*. Singapura: McGraw-Hill Companies.
- Gonzalez, R., & Woods, R. (2002). *Digital Image Processing. 2nd Edition*. Upper Saddle River: Prentice Hall.
- Gorunescu, F. (2011). *Data Mining: Concepts and Techniques*. Verlag berlin Heidelberg: Springer.
- Han, J., & Kamber, M. (2006). *Data Mining : Concepts and Techniques 2nd Edition*. San Fransisco: Elsevier.
- Izadi, M., & Saeedi, P. (2010). Automatic Building Detection in Aerial Images using a Hierarchical Feature Based Image Segmentation. *International Conference on Pattern Recognition*, 472-475.
- Kurniawan, F., Nugroho, S. M., & Hariadi, M. (2018). Smart City Priority Correlation Using Differential Equation. *IEEE Computing in Science & Engineering*.
- Liang, C., Gong, J., Chen, X., & Han, P. (2008). Building boundary extraction from high resolution imagery and LIDAR data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science 37 (Part B3)*, 693-698.
- Powers, D. M. (2011). Evaluation: From Precision, Recall, and F-Measure to ROC, Infomedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 37-63.
- Rizal, A. (2013). Perbandingan Performa antara Imputasi Metode Konvensional dan Imputasi dengan Algoritma Mutual Nearest Neighbor. *Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember. Surabaya*.





- Seni, G., & Elder, J. (2010). Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 1-126.
- Setiawan, W., & Damayanti, F. (2016). Klasifikasi Citra Retina Menggunakan K Nearest Neighbor Untuk Mendeteksi Mokolopati Diabetik. *Seminar Nasional Sains and Teknologi*, 1-6.
- Sinaga, A. S. (2019). Segmentasi Ruang Warna  $L^*a^*b$ . *Jurnal Mantik Penusa*, 43-46.
- Subairi, Rahmadwati, & Yudaningsy, E. (2018). Implementasi Metode k-Nearest Neighbor pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi. *Jurnal EECCIS Vol. 12,*, 9-14.
- Suyatno. (2017). *Data mining untuk klasifikasi dan klusterisasi data*. Bandung: Informatika Bandung.
- Wahyudi, M. Z. (2012, Agustus 23). *Nasional News*. Retrieved from [www.kompas.com](http://www.kompas.com):  
<https://nasional.kompas.com/read/2012/08/23/21232065/%20Hampir.54.Persen.Penduduk.Indonesia.Tinggal.di.Kota>
- Whidhiasih, R. N., Wahanani, N. A., & Supriyanto. (2013). Klasifikasi Buah Belimbing Berdasarkan Citra Red-Green-Blue Menggunakan k-NN dan LDA. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 29-35.
- Wu, X., & Kumar, V. (2009). *The Top Ten Algorithms In Data Mining*. London: CRC PRESS, Taylor & Francis Group.

## LAMPIRAN

Gambar Uji:

No.	Gambar
1	
2	
3	



8	
9	
10	
11	

12	 An aerial photograph of a residential development. The houses have prominent red-tiled roofs and are arranged in a somewhat circular or semi-circular pattern around winding asphalt roads. There are green spaces and trees interspersed among the buildings.
13	 Another aerial view of a residential development, very similar to image 12. It shows a dense cluster of houses with red-tiled roofs, winding roads, and green landscaping. The layout is more compact and circular than in image 12.
14	 An aerial view of a residential development with a more structured, grid-like street pattern. The houses have red-tiled roofs and are arranged in neat rows. There are green spaces and trees between the buildings.
15	 An aerial view of a residential development. It features a mix of red-tiled houses and a large, multi-story building complex with a red roof. There are winding roads, green spaces, and a tennis court visible in the lower right portion of the image.

