

**PENGARUH PENGGUNAAN *COLOR MODEL LAB*
DALAM KALIBRASI WARNA LUKA
MENGUNAKAN METODE SEGMENTASI
K-MEANS DAN *MEAN SHIFT***

Skripsi

**Disusun untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer**



**Oleh:
Aprilia Khairunnisa
3145160091**

**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA**




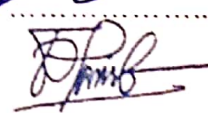
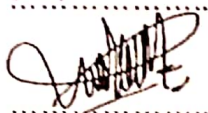
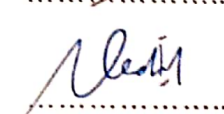

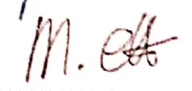
2021

LEMBAR PERSETUJUAN HASIL SIDANG SKRIPSI

PENGARUH PENGGUNAAN *COLOR MODEL LAB* DALAM KALIBRASI WARNA LUKA MENGUNAKAN METODE SEGMENTASI *K-MEANS* DAN *MEAN SHIFT*

Nama : Aprilia Khairunnisa

No. Registrasi : 3145160091

	Nama	Tanda Tangan	Tanggal
Penanggung Jawab			
Dekan	: Prof. Dr. Muktiningsih N, M.Si NIP. 19640511 198903 2 001	 	27 Agustus 2021
Wakil Penanggung Jawab			
Wakil Dekan I	: Dr. Esmar Budi, S.Si., M.T NIP. 19720728 199903 1 002		25 Agustus 2021
Ketua	: Ir. Fariani Hermin Indiyah, M.T NIP. 19600211 198703 2 001		21 Agustus 2021
Sekretaris	: Ari Hendarno, S.Pd, M.Kom NIDK. 8857650017		20 Agustus 2021
Penguji	: Med Irzal, M.Kom NIP. 19770615 200312 1 001		22 Agustus 2021
Pembimbing I	: Drs. Mulyono, M.Kom NIP. 19660517 199403 1 003		23 Agustus 2021
Pembimbing II	: Muhammad Eka Suryana, M.Kom NIP. 19851223 201212 1 002		23 Agustus 2021

Dinyatakan lulus ujian skripsi tanggal : 16 Agustus 2021

LEMBAR PENGESAHAN

Dengan ini saya mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Negeri Jakarta

Nama : Aprilia Khairunnisa

No. Registrasi : 3145160091

Program Studi : Ilmu Komputer

Judul : Pengaruh Penggunaan *Color Model* LAB dalam Kalibrasi
Warna Luka Menggunakan Metode Segmentasi *K-Means*
dan *Mean Shift*

Menyatakan bahwa skripsi ini telah siap diajukan untuk sidang skripsi.

Menyetujui,

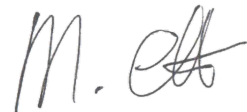
Dosen Pembimbing I

Dosen Pembimbing II



Drs. Mulyono, M. Kom.

NIP. 19660517 199403 1 003



Muhammad Eka Suryana, M. Kom.

NIP. 19851223 201212 1 002

Mengetahui,

Koordinator Program Studi Ilmu Komputer



Ir. Fariani Hermin Indiyah, M.T.

NIP. 19600211 198703 2 001

LEMBAR PERNYATAAN

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul **Pengaruh Penggunaan *Color Model LAB* dalam Kalibrasi Warna Luka Menggunakan Metode Segmentasi *K-Means* dan *Mean Shift*** yang disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Ilmu Komputer Universitas Negeri Jakarta adalah karya ilmiah saya dengan arahan dari dosen pembimbing.

Sumber informasi yang diperoleh dari penulis lain yang telah dipublikasikan yang disebutkan dalam skripsi ini, telah dicantumkan dalam Daftar Pustaka sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Jika dikemudian hari ditemukan sebagian besar skripsi ini bukan hasil karya saya sendiri dalam bagian-bagian tertentu, saya bersedia menerima sanksi pencabutan gelar akademik yang saya sanding dan sanksi-sanksi lainnya sesuai dengan peraturan perundang-undangan yang berlaku.

Jakarta, Juli 2021

Aprilia Khairunnisa

HALAMAN PERSEMBAHAN

Atas izin dan pertolongan Allah Subhanahu Wa Ta'ala
untuk Alm. Papah, Mamah, dan semua Adikku

KATA PENGANTAR

Ungkapan Puji dan Syukur penulis panjatkan kehadirat Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Adapun jenis penelitian yang dipilih yaitu kajian teori dengan judul Pengaruh Penggunaan *Color Model* LAB dalam Kalibrasi Warna Luka Menggunakan Metode Segmentasi *K-Means* dan *Mean Shift*.

Dalam menyelesaikan skripsi ini, penulis sedikit banyak menghadapi hambatan. Namun, dengan bantuan dan dorongan dari berbagai pihak, hambatan tersebut dapat diatasi sehingga penulis dapat menyusun dan menyelesaikan skripsi ini dengan baik. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini khususnya kepada:

1. Bapak Drs. Mulyono M.Kom selaku Dosen Pembimbing I yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap skripsi ini.
2. Bapak M. Eka Suryana M.Kom selaku Dosen Pembimbing II yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap skripsi ini.
3. Ibu Ir. Fariani Hermin Indiyah, M.T selaku Koordinator Program Studi Ilmu Komputer yang telah memberikan bimbingan dan bantuan sehingga skripsi ini dapat diselesaikan tepat pada waktunya.
4. Teman-teman Program Studi Ilmu Komputer 2016 yang telah membantu dan mendukung sehingga skripsi ini dapat diselesaikan tepat pada waktunya.

5. Keluarga dan sahabat tercinta yang telah memberikan dukungan, dorongan serta bantuan yang sangat besar kepada penulis.
6. Dan semua pihak yang juga telah membantu dengan tidak mengurangi rasa hormat penulis yang tidak dapat disebutkan satu persatu.

Dalam penulisan skripsi ini, penulis menyadari bahwa dengan keterbatasan ilmu dan pengetahuan penulis, skripsi ini masih jauh dari sempurna, baik dari segi penulisan, penyajian materi, maupun bahasa. Oleh karena itu, penulis sangat membutuhkan kritik dan saran yang dapat dijadikan sebagai pembelajaran serta dapat membangun penulis agar lebih baik lagi kedepannya.

Akhir kata, penulis berharap ini bermanfaat bagi semua pihak khususnya penulis sendiri, serta menjadi semangat dan motivasi bagi rekan-rekan yang akan melaksanakan skripsi berikutnya. Semoga Allah SWT senantiasa membalas kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini.

Terima kasih,
Jakarta, Juli 2021

Penulis

ABSTRAK

Luka kronis merupakan ancaman besar bagi kesehatan dan ekonomi masyarakat. Selain dapat menyebabkan depresi dan pemisahan sosial di kalangan masyarakat, biaya perawatan untuk luka kronis pun relatif tinggi, yang berkisar mulai dari Rp. 61.500,00 hingga Rp. 600.000,00. Penilaian luka yang akurat pun sangat dibutuhkan untuk memberikan keputusan diagnosis yang akurat, menghindari terjadinya komplikasi, serta dapat menghemat biaya perawatan luka. Namun, teknik pemeriksaan dan perawatan luka yang tersedia saat ini belum dapat mengukur dimensi luka secara akurat dan alat yang digunakan dapat menyebabkan infeksi pada luka. Penelitian ini dilakukan untuk melihat pengaruh model warna LAB terhadap segmentasi warna *Red*, *Yellow*, dan *Black* dengan menggunakan metode *K-Means* dan *Mean Shift* yang diharapkan dapat membantu kalangan dokter dan perawat terkait penilaian luka kronis agar dapat memberikan hasil aproksimasi yang lebih akurat. Data yang digunakan adalah 15 buah citra luka yang terdiri dari 5 buah citra luka merah, 5 buah citra luka kuning, dan 5 buah citra luka hitam. Citra luka terlebih dahulu di-*overlay* menggunakan *overlay mask* sebagai filter. Kemudian, citra hasil *overlay* ditransformasi menjadi citra dengan model warna LAB. Citra hasil transformasi ini kemudian disegmentasi menggunakan metode *k-means* dan *mean shift*. Penentuan nilai *k* dan nilai *centroid* awal untuk metode *k-means* dan penentuan nilai *w* dan *sigma* untuk metode *mean shift* sangat memengaruhi hasil dari segmentasi pada kedua metode ini. Berdasarkan hasil segmentasi dari metode *k-means* dan *mean shift*, menunjukkan bahwa pada citra warna RGB, metode *k-means* dapat mensegmentasi warna merah, kuning, dan hitam lebih baik dibandingkan dengan metode *mean shift*. Namun, pada citra warna LAB, hasil segmentasi belum dapat menunjukkan pengaruh dari penggunaan model warna LAB pada proses segmentasi, baik pada metode *k-means* maupun *mean shift*. Salah satu kekurangan dari penelitian ini, yaitu penelitian yang dilakukan masih kurang komprehensif, di mana hasil dari segmentasi masih belum dapat memperlihatkan pengaruh dari penggunaan model warna LAB pada proses segmentasi. Komprehensif yang dimaksud, yaitu belum dapat dilakukannya pengembangan lebih lanjut terhadap algoritma yang digunakan, namun untuk pengimplementasian dari algoritmanya sudah selesai dan kebenaran dari algoritmanya sudah di cek oleh Pembimbing.

Kata Kunci: Luka kronis, segmentasi, model warna LAB, *k-means*, *mean shift*.

ABSTRACT

Chronic wounds are a major threat to public health and economy. Besides being able to cause depression and social separation in the community, the cost of treatment for chronic wounds is relatively high, which ranges from Rp. 61.500,00 to Rp. 600.000,00. Accurate wound assessment is also needed to provide accurate diagnosis decisions, avoid complications, and can save wound care costs. However, currently available wound examination and treatment techniques have not been able to accurately measure wound dimensions and the tools used can cause excessive wound infection. This study was conducted to see the effect of the LAB color model on Red, Yellow, and Black color segmentation using K-Means and Mean Shift methods which are expected to help doctors and nurses regarding chronic wound assessment in order to provide more accurate approximation results. The data used are 15 wound images consisting of 5 red wound images, 5 yellow wound images, and 5 black wound images. The wound image is first overlaid using an overlay mask as a filter. Then, the overlaid image is transformed into an image using the LAB color model. The transformed image is then segmented using k-means and mean shift methods. The determination of k value and initial centroid value for k-means method and the determination of w and sigma values for mean shift method greatly affect the result of segmentation in these two methods. Based on the segmentation result from k-means and mean shift methods, it shows that in RGB color images, k-means method can segment red, yellow, and black colors better than mean shift method. However, in LAB color images, the segmentation result have not been able to show the effect of using LAB color model on the segmentation process, both in k-means and mean shift methods. One of the shortcomings of this study is that the research conducted is still not comprehensive, where the results of segmentation still cannot show the effect of using the LAB color model on the segmentation process. Comprehensive is meant, namely that further development of the algorithm used has not been carried out, but the implementation of the algorithm has been completed and the correctness of the algorithm has been checked by the Supervisor.

Keywords: *Chronic wounds, segmentation, LAB color model, k-means, mean shift.*

DAFTAR ISI

LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR	vii
ABSTRAK	viii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	10
1.3 Pembatasan Masalah	10
1.4 Tujuan Penelitian	11
1.5 Manfaat Penelitian	11
II KAJIAN PUSTAKA	13
2.1 Citra	13
2.2 Warna	14
2.2.1 Model Warna RGB dan XYZ	15
2.2.2 Model Warna L^*a^*b dan L^*u^*v	29

2.2.3	Model warna HSV dan HLS	35
2.3	Segmentasi Citra (<i>Image Segmentation</i>)	39
2.4	<i>Clustering</i>	40
2.5	<i>K-means</i>	41
2.6	<i>Mean shift</i>	44
III	METODOLOGI PENELITIAN	53
3.1	Data	54
3.2	<i>Color Model Transformation</i>	57
3.3	<i>Color Segmentation</i>	58
3.4	Evaluasi	60
IV	HASIL DAN PEMBAHASAN	61
4.1	Pengujian	61
4.2	Implementasi <i>Color Model Transformation</i>	61
4.3	Implementasi <i>Color Segmentation</i>	64
4.3.1	Implementasi Metode <i>K-Means</i>	64
4.3.2	Implementasi Metode <i>Mean Shift</i>	66
4.4	Evaluasi	68
V	KESIMPULAN DAN SARAN	70
5.1	Kesimpulan	70
5.2	Saran	71
	DAFTAR PUSTAKA	77
	DAFTAR LAMPIRAN	78
A	Sampel Kode Proses <i>Overlay</i> Menggunakan <i>Overlay Mask</i>	78

B	Sampel Kode Proses <i>Color Model Transformation</i>	79
C	Sampel Kode Segmentasi Menggunakan Metode <i>K-Means</i>	81
D	Sampel Kode Segmentasi Menggunakan Metode <i>Mean Shift</i>	84

DAFTAR GAMBAR

Gambar 2.1	Model warna CIE RGB dan XYZ	17
Gambar 2.2	<i>Colour matching function</i> atau fungsi pencocokan warna . . .	19
Gambar 2.3	Model warna CIE LAB	34
Gambar 2.4	Model warna HSV dan HLS	36
Gambar 2.5	1) k inisial " <i>means</i> " dipilih secara acak dari data set, 2) k <i>cluster</i> dibuat dengan mengaitkan setiap observasi dengan <i>mean</i> terdekat, 3) <i>centroid</i> (pusat) dari masing-masing k <i>clus-</i> <i>ter</i> menjadi <i>mean</i> baru, 4) ulangi tahap 2 dan 3 sampai kon- vergensi tercapai	42
Gambar 2.6	Jenis piksel tetangga: (a) 4-tetangga +, (b) 4-tetangga x, (c) 8-tetangga	50
Gambar 3.1	Diagram alur penelitian	53
Gambar 3.2	Proses <i>color model transformation</i>	58
Gambar 3.3	Proses <i>color segmentation</i>	59
Gambar 4.1	Hasil <i>overlay</i> pada citra luka merah	61
Gambar 4.2	Hasil <i>overlay</i> pada citra luka kuning	62
Gambar 4.3	Hasil <i>overlay</i> pada citra luka hitam	62
Gambar 4.4	Hasil transformasi warna LAB pada citra luka merah	63
Gambar 4.5	Hasil transformasi warna LAB pada citra luka kuning	63
Gambar 4.6	Hasil transformasi warna LAB pada citra luka hitam	64
Gambar 4.7	Hasil segmentasi metode <i>k-means</i> pada citra luka merah . . .	65
Gambar 4.8	Hasil segmentasi metode <i>k-means</i> pada citra luka kuning . . .	65
Gambar 4.9	Hasil segmentasi metode <i>k-means</i> pada citra luka hitam . . .	66
Gambar 4.10	Hasil segmentasi metode <i>mean shift</i> pada citra luka merah . .	67

Gambar 4.11 Hasil segmentasi metode *mean shift* pada citra luka kuning . 67

Gambar 4.12 Hasil segmentasi metode *mean shift* pada citra luka hitam . . 68

DAFTAR TABEL

Tabel 3.1	Rincian data luka merah	55
Tabel 3.2	Rincian data luka kuning	56
Tabel 3.3	Rincian data luka hitam	57

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Luka merupakan suatu kondisi terputusnya jaringan lunak pada kulit yang dapat disebabkan karena terkena sumber panas, seperti api dan zat kimia, hasil tindakan pembedahan, trauma benda tajam atau tumpul, dan perubahan kondisi tubuh, yang dapat menyebabkan fungsi dan struktur anatomi tubuh terganggu (Purnama et al., 2017). Ketika timbul luka, beberapa efek yang muncul di antaranya hilangnya seluruh atau sebagian fungsi organ, pendarahan dan pembekuan darah, kontaminasi bakteri serta kematian sel. Luka yang memiliki waktu penyembuhan yang lama dikhawatirkan akan mengalami komplikasi dan menimbulkan infeksi (Amalia, 2015).

Penyembuhan luka merupakan proses regenerasi jaringan kulit yang rusak menjadi jaringan yang baru. Proses penyembuhan luka terjadi dalam empat fase, yaitu: hemostasis (0 - beberapa jam setelah cedera), inflamasi (1 - 3 hari), proliferasi (4 - 21 hari), dan *remodelling* (21 hari - 1 tahun) (Landén et al., 2016). Fase-fase penyembuhan luka ini dapat terjadi secara bertahap maupun tumpang tindih (*overlap*). Agar luka sembuh secara sempurna, keempat fase tersebut harus terjadi secara tepat dan terartur. Namun, sering kali ditemukan luka dengan durasi yang memanjang atau memiliki proses penyembuhan yang lambat. Kondisi ini disebut sebagai luka kronis. Luka kronis dapat disebabkan oleh beberapa hal, di antaranya kegagalan pemulihan karena menderita penyakit tertentu, seperti diabetes dan kanker, infeksi secara terus menerus, dan rendahnya tindakan pengobatan yang diberikan. Tiga jenis luka yang dianggap sebagai luka kronis utama, yaitu luka kaki diabetik (*diabetic foot ulcer*), luka vena (*venous ulcer*), dan luka dekubitus (*pressure ulcer*) (Rahmansyah et al.,

2016).

Setiap tahunnya angka kejadian luka semakin meningkat. Hal ini ditunjukkan pada sebuah penelitian di Amerika yang menyatakan bahwa prevalensi pasien dengan luka sebanyak 3,50 per 1000 populasi penduduk dengan tingkat prevalensi untuk luka kronis adalah sekitar 2% dari populasi umum dan 8,5% dari orang tua. Luka yang paling banyak diderita oleh penduduk dunia adalah luka karena pembedahan atau trauma (48.00%), ulkus kaki (28.00%), dan luka dekubitus (21.00%) (Huda et al., 2018). Menurut data dari Riset Kesehatan Dasar (Riskesdas) tahun 2013, prevalensi luka di Indonesia sebanyak 8.2% dengan prevalensi tertinggi terdapat di Sulawesi Selatan sebanyak 12.8% dan prevalensi terendah terdapat di Jambi sebanyak 4.5%. Sebanyak 70.9% adalah luka lecet atau memar dan luka iris atau robek sebanyak 23.3%. Penyebab luka terbanyak yaitu karena jatuh dan kecelakaan motor dengan prevalensi sebanyak 40.9% dan 40.6% (Kesehatan, 2013).

Secara global, luka kronis adalah ancaman besar bagi kesehatan dan ekonomi masyarakat karena memiliki efek buruk pada kualitas hidup pasien yang dapat menyebabkan depresi, pemisahan sosial, dan biaya tinggi untuk perawatannya (Mukherjee et al., 2014). Di Indonesia, luka kronis masih merupakan masalah yang terbilang berat dalam penanganannya, baik dalam perawatan maupun rehabilitasinya (Rahmansyah et al., 2016). Tidak hanya berdampak buruk bagi pasien, luka kronis pun berdampak buruk bagi instansi kesehatan, di mana menyebabkan tingginya pembayaran asuransi perawatan luka per minggu karena perawatan luka menghabiskan 66% waktu perawat (HSE, 2007). Di Amerika, luka kronis setiap tahunnya menelan biaya \$ 20 miliar dan mempengaruhi 5,7 juta orang penderita (Brown et al., 2018). Berdasarkan data dari alodokter.com dan caradise.com, biaya perawatan luka di Indonesia berkisar mulai dari Rp 61.500,00-Rp 600.000,00. Biaya tersebut belum termasuk biaya balutan.

Mengidentifikasi area luka adalah langkah pertama dalam penilaian dan pe-

rawatan luka kronis. Penilaian luka yang akurat dapat membantu untuk memberikan keputusan diagnosis yang akurat, menentukan pemberian balutan yang tepat, memantau perbaikan luka, menghindari terjadinya komplikasi, serta dapat menghemat biaya perawatan luka. Penilaian luka kronis umumnya didasarkan pada dua jenis tinjauan klinis: tinjauan visual untuk identifikasi jaringan luka dari warna dominan dan tinjauan manual untuk pemeriksaan luka untuk menggambarkan bentuk luka (area, perimeter, kedalaman, dan lain-lain). Saat ini, tersedia dua teknik untuk pemeriksaan ini: (1) metode manual langsung yang digunakan oleh dokter dan perawat untuk mengukur secara berkala dimensi luka menggunakan penggaris, (2) melacak batas luka pada kertas kalkir transparan yang ditempatkan pada kotak metrik. Kertas kalkir merupakan sebuah kertas yang memiliki permukaan yang tembus pandang dan sering digunakan oleh desainer untuk merancang desain atau gambar. Kertas ini memiliki struktur seperti kaca yang dapat dilihat secara tembus pandang ke bagian belakang kertas kalkir tersebut. Permukaan luka kemudian ditentukan dengan menghitung jumlah kotak secara manual setelah memindai gambar. Namun, kedua metode tersebut tidak dapat mengukur dimensi luka secara akurat dan kertas kalkir yang digunakan dapat menyebabkan infeksi pada luka (Gupta, 2017).

Metode berbasis *digital image processing* telah diusulkan sebagai teknik alternatif untuk penilaian luka karena dapat memberikan langkah-langkah yang objektif, lebih akurat, dan dapat direproduksi. Selain itu, tidak akan menimbulkan infeksi pada luka karena tidak ada kontak antara luka dan sistem pengukuran. Salah satu metode berbasis *digital image processing* yang sering digunakan adalah segmentasi citra (*image segmentation*). Segmentasi citra merupakan proses mempartisi gambar digital menjadi beberapa segmen (set piksel, disebut juga superpiksel) yang memiliki fitur atau atribut yang sama. Segmentasi bertujuan untuk menyederhanakan atau mengubah representasi suatu citra menjadi sesuatu yang lebih bermakna dan lebih

mudah untuk dianalisis. Segmentasi citra biasa digunakan untuk menemukan objek dan batas (seperti garis, kurva, dan lain-lain) dalam gambar (Shmmla and Ashour, 2013). Biasanya segmentasi menggunakan informasi lokal dalam gambar digital untuk menghitung segmentasi terbaik, seperti informasi warna yang digunakan untuk membuat histogram atau informasi yang mengindikasikan tepi, batas atau informasi tekstur (Khattab et al., 2014).

Segmentasi warna citra (*color image segmentation*) didasarkan pada fitur warna piksel gambar yang mengasumsikan bahwa warna-warna homogen pada gambar bersesuaian dengan kelompok yang terpisah. Dengan kata lain, setiap kelompok mendefinisikan kelas piksel yang memiliki properti warna yang sama. Karena hasil segmentasi bergantung pada ruang warna (*color space*) yang digunakan, tidak ada ruang warna tunggal yang dapat memberikan hasil yang dapat diterima untuk semua jenis gambar. Karena alasan ini, banyak penulis yang mencoba menentukan ruang warna yang sesuai dengan masalah segmentasi warna citra spesifik mereka (Khattab et al., 2014). Beberapa macam dari ruang warna (*color space*), yaitu *RGB*, *CMY(K)*, *HSV*, *CIE*, L^*a^*b , L^*u^*v , dan *YCrCb*. Setiap ruang warna (*color space*) mempunyai sekurang-kurangnya 3 elemen warna dasar.

Beberapa algoritma telah dikembangkan untuk segmentasi citra. Teknik-teknik ini seringkali harus dikombinasikan dengan pengetahuan domain agar dapat secara efektif menyelesaikan segmentasi citra. Beberapa metode yang digunakan dalam segmentasi citra, yaitu: metode *clustering*, metode berbasis histogram, deteksi tepi, metode *region growing*, metode level set, metode partisi graph, transformasi *watershed*, segmentasi berbasis model, segmentasi multi-skala, segmentasi semi-otomatis, dan segmentasi jaringan saraf. Metode-metode ini digunakan untuk mencapai segmentasi citra berbasis warna untuk berbagai jenis gambar (Shmmla and Ashour, 2013).

Unsupervised learning merupakan salah satu algoritma dalam *machine lear-*

ning di mana datanya tidak memiliki label maupun target variabel. Algoritma ini dianggap dapat menemukan kesamaan karakteristik dari atribut yang dimiliki, seperti kepadatan, struktur, segmen serupa, dan fitur serupa lainnya. Salah satu tujuan dari algoritma ini, yaitu mengelompokkan objek yang hampir sama dalam suatu area tertentu. *Clustering* dianggap sebagai salah satu masalah *unsupervised learning* yang paling terkenal dalam penelitian *machine learning*. *Clustering* adalah partisi dari kumpulan data atau objek menjadi kelompok atau *cluster*. *Cluster* dapat didefinisikan sebagai kumpulan benda yang menyerupai satu sama lain dan berbeda dengan objek lainnya (Shmmla and Ashour, 2013). Salah satu algoritma *clustering* yang paling sering digunakan adalah algoritma *K-Means*. *K-Means* merupakan metode pengelompokan dari sejumlah *cluster* yang terpisah. "*K*" mengacu pada jumlah *cluster* yang ditentukan. Prosedur dari algoritma *K-Means* adalah untuk mempartisi set data input menjadi set "*k*" berdasarkan jarak yang melekat satu sama lain (Yadav et al., 2013). Algoritma ini akan mempartisi data yang berkarakteristik sama ke dalam suatu kelompok yang sama dan mengelompokkan data yang berkarakteristik berbeda ke kelompok lainnya. Pengelompokan dilakukan untuk meminimalkan variasi di dalam suatu kelompok dan memaksimalkan variasi antar kelompok (Gustientiedina et al., 2019).

Algoritma lain yang digunakan pada segmentasi citra adalah algoritma *mean shift*. *Mean shift* adalah teknik analisis *non-parametric feature space* atau analisis ruang fitur berbasis ruang non-parametrik untuk menentukan nilai maksimum lokal dari suatu probabilitas kepadatan yang diberikan. Analisis non-parametrik digunakan ketika distribusi dari data yang diamati tidak diketahui. Analisis ini juga digunakan ketika data tidak berdistribusi secara normal. Dalam pengolahan citra, *feature space* adalah ruang abstrak di dalam citra yang setiap contoh polanya digambarkan sebagai sebuah titik di dalam ruang berdimensi n atau *n-dimensional space*. Besar dari dimensi

tersebut ditentukan oleh jumlah dari *feature* yang digunakan untuk mendeskripsikan pola. Jadi, analisis *non-parametric feature space* merupakan suatu teknik analisis pada ruang abstrak di dalam suatu citra berdimensi n yang datanya memiliki distribusi yang tidak diketahui dan tidak berdistribusi secara normal. Dengan menggunakan segmentasi *mean shift*, suatu citra akan memiliki warna-warna yang homogen dalam suatu wilayah. Metode *mean shift* digunakan untuk menemukan kepadatan mode dari suatu gambar (Panjaitan, 2016).

Ada beberapa penelitian yang terkait dengan segmentasi warna luka, salah satunya adalah penelitian yang dilakukan oleh Ramya and Jenitta (2017). Dalam penelitiannya, mereka menerapkan sistem penilaian baru untuk pasien diabetes yang digunakan untuk menganalisis luka dengan diabetes tipe 1 dan tipe 2. Gambar pasien dikumpulkan dari klinik dan disimpan di PC atau ponsel. Ia melakukan *pre-processing*, penghalusan dan mensegmentasi citra biner. Algoritma yang digunakan adalah *Mean shift segmentation* dan *K-means clustering*. Kedua algoritma ini digunakan karena dapat memberikan akurasi yang lebih baik. *Back Propagation Neural Network classifier* digunakan untuk mengklasifikasikan gambar yang kemudian akan menghasilkan *output* yang hasilnya dapat ditransfer kepada orang lain menggunakan teknologi GSM. Sistem ini memberikan resolusi tinggi, hasil yang jauh lebih baik, dan sistem yang berbiaya rendah.

Ada beberapa kelompok penelitian lain yang terlibat dalam *image processing* warna luka. Wales, Jones, Plassman telah mengembangkan alat, yang dikenal sebagai MAVIS, untuk mengukur dimensi luka kulit. Kelompok ini melakukan eksperimen untuk membagi gambar menjadi salah satu dari tiga jenis jaringan: kulit yang sehat, jaringan luka, dan jaringan epitelisasi. Mereka mencoba enam parameter pengukuran: bidang warna R , G , dan B ; *Hue*, *Saturation*, dan intensitas skala abu-abu. Bidang warna R , G , dan B menunjukkan bahwa ambang batas bidang warna yang lurus tidak

dapat menghasilkan segmentasi yang baik yang membedakan antara luka dan jaringan kulit. Mereka menyimpulkan bahwa dalam melihat histogram 1D, segmentasi hanya dapat dicapai sebagian, tetapi menggunakan ruang histogram *RGB* 3D, volume *cluster* mungkin lebih banyak dipisahkan (Kolesnik and Fexa, 2004). Mekkes and Westerhof (1995) telah membuat beberapa kemajuan dengan teknik pengelompokan histogram warna *RGB* 3D untuk menilai penyembuhan luka. Mereka menemukan bahwa klaster dalam ruang *RGB* untuk jenis jaringan tertentu membentuk awan 3D yang berbentuk tidak teratur, dan dengan demikian pengukuran yang sederhana di sepanjang sumbu *R*, *G*, dan *B* tidak akan membantu untuk membagi gambar ke dalam tiga jenis jaringan ini.

Teknik lain yang dilaporkan pada segmentasi luka, yaitu penggunaan skema klasifikasi *red-yellow-black* untuk mengevaluasi aktivitas debridemen luka; metode untuk mengoreksi konvektivitas tungkai dalam gambar video berwarna untuk mengukur ukuran luka dan ulserasi kulit; penggunaan teknik *color image processing* dasar untuk mengevaluasi perbaikan luka pada manusia dan hewan (Kolesnik and Fexa, 2004). Pengkajian luka dengan menggunakan klasifikasi warna RYB sudah banyak diketahui untuk membedakan luka kronis atau akut dengan membedakan warna dan kelembaban luka. Klasifikasi ini dapat digunakan pada luka akut ataupun kronis dan manfaatnya pun sudah dapat dirasakan karena mudah untuk diaplikasikan, sederhana, dan mampu memandu perawat luka untuk menentukan intervensi selanjutnya. Warna merah mengindikasikan adanya jaringan granulasi yang menandakan bahwa luka akan sembuh, warna kuning mengindikasikan bahwa perlunya *wound cleansing* yang optimal, dan warna hitam mengindikasikan adanya nekrosis sehingga diperlukan debridemen luka (Aryani et al., 2019).

Penelitian lain yang dilakukan oleh Wannous et al. (2011) melaporkan tingkat akurasi segmentasi rata-rata 73,1% dengan algoritma JSEG. Meskipun berbagai algo-

ritma telah digunakan untuk segmentasi, hanya *color space RGB* dan *HSI* saja yang digunakan untuk segmentasi. Namun, beberapa literatur mengatakan bahwa penggunaan *color space* yang berbeda dan analisis komparatif yang mereka gunakan untuk segmentasi luka masih kurang.

Ketika sebuah gambar luka diambil dengan pose yang sama oleh kamera yang berbeda, warna yang tersimpan mungkin saja berbeda. Hal ini dikarenakan setiap kamera memiliki format RGB mereka sendiri yang menghasilkan ruang warna RGB yang berbeda. Salah satu solusi untuk masalah ini adalah menggunakan format *device independent sRGB*. Van Poucke et al. (2010) mengusulkan sebuah proses kalibrasi untuk mengubah gambar dengan format *device dependent RGB* menjadi format *sRGB* yang memiliki hubungan dengan ruang warna CIE *colorimetric* dengan menggunakan *chart* referensi *MacBeth Colour Checker Chart Mini* [MBCCC]. Percobaan dilakukan sebanyak dua kali. Percobaan pertama dilakukan untuk memeriksa apakah pengaturan kamera atau kondisi pencahayaan berdampak negatif pada kualitas kalibrasi kolorimetri. Proses kalibrasi dilakukan menggunakan koleksi besar gambar normal dengan pengaturan kamera yang tidak tepat dan pencahayaan yang kurang. Presisi dan akurasi warna setelah dikalibrasi akan dinilai secara kuantitatif menggunakan distribusi probabilitas untuk perbedaan perseptual (dE_{ab}). Delta-E (dE) merupakan suatu angka yang mewakili jarak antara dua warna. Ide dari dE , yaitu bahwa dE 1,0 adalah perbedaan warna terkecil yang dapat dilihat mata manusia. Jadi, setiap dE yang nilainya kurang dari 1,0 perbedaannya tidak terlihat dan setiap dE yang nilainya lebih besar dari 1,0 maka perbedaannya akan terlihat. dE biasanya mengacu pada dE_{76} (delta-E yang muncul pada tahun 1976, bersamaan dengan munculnya model warna LAB), atau yang juga dikenal sebagai dE_{ab} . Percobaan kedua dirancang untuk menilai dampak dari prosedur kalibrasi otomatis, yaitu efek dari deteksi *chart* MBCCC pada pengukuran luka sebenarnya. Menggunakan 40 gambar yang berbeda

dari luka nyata yang diperoleh dan dari setiap gambar dipilih wilayah atau area yang menjadi perhatian (*region of interest* (ROI)). Hasilnya menunjukkan sebagian besar *error* berada di bawah 1 dE_{ab}, yang menunjukkan bahwa deteksi *chart* kuat.

Selain menggunakan kamera digital, beberapa penelitian juga menggunakan *smartphone* untuk melakukan penilaian luka, salah satunya dalam menentukan batas luka. Wang et al. (2015) merancang dan mengembangkan sebuah kotak pengambilan gambar menggunakan kamera *smartphone* dan dua set cermin, untuk mengambil gambar ulkus kaki dasar. Setelah gambar luka ditangkap oleh kamera *smartphone*, selanjutnya gambar akan diproses untuk mendapatkan garis kaki berdasarkan warna kulit dan batas luka ditentukan menggunakan algoritma *mean shift* yang diikuti dengan segmentasi warna dalam area luka dan disempurnakan lagi menggunakan *region adjacency graph* (RAG). Setelah estimasi dari batas luka diperoleh, analisis luka dilakukan dengan mengklasifikasikan semua piksel ke dalam kategori warna *red-yellow-black* (RYB) dan mengelompokkannya menggunakan algoritma *K-Means clustering*. Metode ini dievaluasi ke 34 data pasien yang berada di klinik UMass di Worcester. Kemudian Wang et al. (2017) kembali melakukan penelitian yang masih berfokus pada penentuan batas luka namun kali ini dengan menggunakan SVM. Metode yang dilakukan adalah sebagai berikut, pertama gambar disegmentasi menjadi superpiksel menggunakan metode linear *iterative clustering* sederhana (SLIC). Kemudian dari superpiksel tersebut fitur warna dan tekstur yang signifikan diekstraksi. Deskriptor fitur dan warna yang berbeda diekstraksi untuk persiapan dari dua tahap pengklasifikasi SVM. Pada tahap pertama, warna dan representasi BoW (*bag-of-words*) diekstraksi menggunakan *dense scale invariant feature transformation* (DSIFT). Untuk tahap kedua, warna dan *wavelet texture* diekstraksi. Tahap SVM pertama mengimplementasi set *k-binary SVM classifier* menggunakan subset yang berbeda dari seluruh data gambar. Pada tahap SVM kedua, set tes klasifikasi yang salah diimplementasi

ke SVM *binary* tunggal. Setelah selesai, hasilnya disempurnakan lagi menggunakan teknik *conditional random field* (CRF). Penelitian yang dilakukan oleh Wang et al. ini memberikan hasil yang memuaskan, di mana salah satu dari penelitian memberikan hasil akurasi sensitivitas rata-rata sebesar 73,3% dan spesifisitas sebesar 94,6%. Namun, semua percobaan yang dilakukan oleh Wang et al. hanya berfokus pada ulkus kaki (*close wound*) saja.

Penulis bersama dengan Muhammad Rizki dari Ilmu Komputer angkatan 2016 Universitas Negeri Jakarta, sebagai tim riset *medical wound imaging* akan mengembangkan beberapa metode terkait dengan *assessment* luka. Pada penelitian ini, Rizki secara khusus berfokus untuk mengembangkan metode PACMs dan mendeteksi keliling luka dari data gambar luka kronis, sedangkan penulis akan berfokus untuk melihat pengaruh ruang warna (*color space*) LAB terhadap segmentasi warna *red*, *yellow*, dan *black* pada citra luka kronis dengan menggunakan metode *K-Means* dan *Mean shift*. Penelitian ini bertujuan untuk membantu kalangan dokter dan perawat terkait penilaian luka kronis agar dapat memberikan hasil aproksimasi yang lebih akurat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, fokus permasalahan pada penelitian ini adalah "Bagaimana melihat pengaruh ruang warna (*color space*) LAB terhadap segmentasi warna *red*, *yellow*, dan *black* pada citra luka kronis dengan menggunakan metode *K-Means* dan *Mean shift*?".

1.3 Pembatasan Masalah

Pembatasan masalah pada penelitian ini, yaitu

1. *Color model* atau model warna yang digunakan hanya model warna LAB.

2. Metode segmentasi yang digunakan hanya metode *K-Means* dan *Mean Shift*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini, yaitu untuk melihat pengaruh ruang warna (*color space*) LAB terhadap segmentasi warna *red*, *yellow*, dan *black* pada citra luka kronis dengan mengujikan metode *K-Means* dan *Mean Shift* agar dapat membantu kalangan dokter dan perawat terkait penilaian luka kronis sehingga dapat memberikan hasil aproksimasi yang lebih akurat.

1.5 Manfaat Penelitian

1. Bagi peneliti

Penelitian ini merupakan media penerapan ilmu pengetahuan, khususnya dalam penggunaan metode *K-Means* dan *Mean Shift* pada segmentasi warna luka.

2. Instansi terkait

Diharapkan dapat membantu dokter, perawat dan instansi terkait dalam proses pengkajian luka kronis dengan lebih terukur, lebih cepat, serta mempermudah dalam pengambilan keputusan.

3. Bagi ilmu pengetahuan

- Mahasiswa

Diharapkan penelitian ini dapat digunakan sebagai penunjang referensi, khususnya pustaka tentang segmentasi warna luka menggunakan metode *K-Means* dan *Meanshift*.

- Bagi peneliti selanjutnya

Diharapkan penelitian ini dapat digunakan sebagai dasar atau kajian awal bagi peneliti lain yang ingin meneliti permasalahan yang sama.

BAB II

KAJIAN PUSTAKA

2.1 Citra

Citra atau *image* adalah artifak yang menampilkan persepsi visual dari suatu objek dalam bentuk dua dimensi. Dalam konteks gelombang, citra adalah sinyal dua dimensi yang bersifat kontinu dan dapat diamati oleh sistem visual manusia. Pada citra, setiap titik pada posisi tertentu dapat menghasilkan nilai intensitas cahaya tertentu, sehingga secara sistematis, citra adalah fungsi dua dimensi yang menyatakan nilai intensitas cahaya pada bidang dua dimensi (Teresa, 2019). Berdasarkan bentuk sinyal penyusunnya, citra dibedakan menjadi citra analog dan citra digital.

Citra analog merupakan citra hasil dari transformasi gelombang cahaya yang terbentuk dari sinyal yang kontinu. Citra analog dihasilkan melalui alat akuisisi citra analog, yaitu mata manusia dan kamera analog. Citra ini memiliki kualitas resolusi (tingkat kerincian) yang sangat baik, namun tidak dapat disimpan, diolah, maupun diduplikasi di dalam komputer. Sedangkan citra digital didefinisikan sebagai fungsi dua variabel $f(x, y)$ dalam bentuk diskrit (Pamungkas, 2017). Citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, di mana perpotongan antara kolom dan baris disebut dengan piksel. Piksel merupakan elemen terkecil dari suatu citra yang mempunyai dua parameter, yaitu koordinat dan intensitas atau warna.

Representasi citra digital adalah suatu kegiatan dalam pengolahan citra digital untuk menampilkan kembali sebuah citra yang telah di-*digitizing* dengan cara memotong atau mempartisi gambar ke dalam bentuk titik-titik warna yang ditandai dengan angka yang menunjukkan tingkat kecerahan warna tersebut dan dipetakan menggunakan sistem koordinat. *Digitizing* merupakan proses mengubah bentuk citra analog

ke citra digital agar dapat dimanipulasi dan diolah di dalam komputer.

Secara sistematis, citra digital direpresentasikan sebagai suatu fungsi intensitas $f(x, y)$. x dan y merupakan koordinat posisi, di mana x menunjukkan baris dan y menunjukkan kolom, dan $f(x, y)$ merupakan fungsi intensitas pada setiap titik (x, y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Dalam proses matematisnya, tidak terdapat perbedaan antara koordinat matriks dan koordinat piksel pada suatu citra. Perbedaan di antara keduanya, yaitu koordinat matriks menunjukkan letak suatu titik pada citra asli, sedangkan koordinat piksel menunjukkan letak suatu titik citra pada layar monitor.

Mekanisme representasi citra digital, yaitu suatu elemen matriks diisi oleh angka-angka yang mewakili warna-warna yang tampak pada mata. Kumpulan angka tersebut kemudian disimpan dalam komputer dengan berbagai format citra yang ada. Berdasarkan nilai yang disimpan pada citra digital, citra dibagi menjadi empat jenis, yaitu citra biner, citra *grayscale*, citra berwarna (*true color*) dan citra warna berindeks.

2.2 Warna

Representasi sebuah warna didasarkan pada hubungan antara cahaya berwarna dan persepsi. Cahaya merupakan gelombang elektromagnetik yang mengenai suatu benda. Disaat gelombang elektromagnetik ini mengenai suatu objek, sebagian dari frekuensi cahaya diserap dan sebagian lainnya dipantulkan ke mata manusia. Frekuensi yang dipantulkan ke mata manusia ini yang kita anggap sebagai warna.

Panjang gelombang cahaya dapat menentukan identitas dari suatu warna. Panjang gelombang yang bisa ditangkap oleh mata manusia berkisar antara 380-700 nm. Suatu model yang memberikan representasi numerik untuk warna tertentu dan menentukan hubungan suatu warna dengan warna lain disebut dengan model warna atau

color model. Tujuan dari suatu model warna adalah untuk mewakili warna sedemikian rupa sehingga suatu gambar merupakan representasi akurat dari cara kita memandang sesuatu, dan untuk dapat menciptakan atau merepresentasikan kembali warna tersebut (Nixon and Aguado, 2020).

2.2.1 Model Warna RGB dan XYZ

Model warna RGB dan XYZ (disebut juga CIE RGB dan CIE XYZ) merupakan model warna yang memberikan deskripsi warna sesuai dengan persepsi manusia. Kedua model warna ini mencirikan sifat warna yang sama, namun menggunakan warna dasar yang berbeda. Di dalam retina mata manusia, terdapat sel penerima sinar yang bertanggung jawab terhadap penglihatan warna. Sel tersebut disebut sebagai sel kerucut (*cone cell*). Pada mata manusia, terdapat tiga sel kerucut, yang di mana ketiga sel ini merupakan sel yang paling banyak merespon cahaya yang memiliki panjang gelombang yang sama dengan panjang gelombang dari warna RGB yang ditangkap oleh mata manusia. CIE RGB digunakan untuk merepresentasikan warna atau cahaya yang ditangkap oleh sel kerucut pada mata manusia.

CIE RGB terdiri dari tiga warna, yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Model warna ini disebut sebagai warna aditif, di mana penggabungan dari jumlah warna merah, hijau, dan biru yang berbeda akan menghasilkan warna lain. Penggabungan dari warna merah dan hijau akan menghasilkan warna kuning, hijau dan biru akan menghasilkan warna *cyan*, dan merah dan biru akan menghasilkan warna magenta. Gabungan dari ketiga warna tersebut dalam intensitas penuh (maksimal) akan menghasilkan warna putih dan apabila dalam intensitas minimum (intensitas 0) akan menghasilkan warna hitam.

CIE RGB merupakan model warna yang paling sering digunakan dan biasanya digunakan untuk menampilkan citra atau gambar dalam perangkat elektronik. Alasan

mengapa RGB menjadi model warna standar di sebagian aplikasi adalah karena RGB menawarkan pilihan warna yang terluas. Jadi, dengan menggabungkan tiga warna dalam jumlah yang bervariasi, akan didapatkan warna apapun yang diinginkan dengan akurasi yang kuat (Andy, 2020).

Pada layar monitor, warna dalam model warna RGB direpresentasikan oleh berbagai titik cahaya, sehingga warna yang ditampilkan merupakan penggabungan dari masing-masing cahaya merah, hijau, dan biru. Warna dituliskan dalam bentuk triplet $RGB (r, g, b)$, di mana setiap bagiannya dapat bervariasi dari nol sampai nilai maksimum yang ditetapkan. Jangkauan untuk nilai maksimum dapat digambarkan dalam beberapa cara, yaitu

1. Dari 0 sampai 1, dengan sembarang nilai pecahan di antaranya. Representasi ini digunakan untuk analisis teoritis.
2. Dapat juga ditulis dalam bentuk presentase (0% sampai 100%).
3. Dalam komputer, biasanya disimpan sebagai angka integer dari 0 sampai 255, masing-masing warna memiliki kisaran maksimum 8 bit, dan dapat dituliskan ke dalam bentuk desimal maupun heksadesimal.

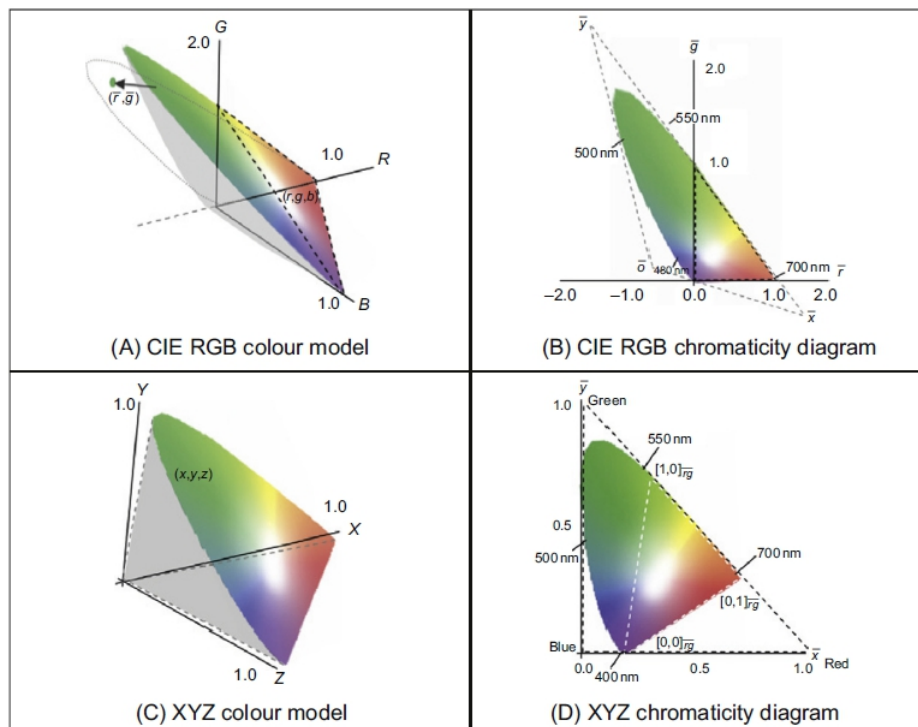
Secara umum, cara untuk mengilustrasikan CIE RGB adalah dengan mempertimbangkan kromatisitas warna. Kromatisitas warna merupakan spesifikasi objektif dari kualitas suatu warna terlepas dari luminansi (kecerahan). Energi cahaya dihilangkan dengan melakukan normalisasi terhadap energi total. Koordinat kromatisitas didefinisikan sebagai

$$\bar{r} = \frac{r}{r + g + b}, \bar{g} = \frac{g}{r + g + b}, \bar{b} = \frac{b}{r + g + b} \quad (2.1)$$

di mana \bar{r} , \bar{g} , dan \bar{b} adalah koordinat kromatisitas dari CIE RGB, r , g , dan b adalah komponen warna dari CIE RGB, dan $r + g + b$ adalah energi total. Jumlah seluruh

nilai kromatisitas warna adalah satu, yang berarti $r + g + b = 1$. Untuk menentukan kromatisitas warna, hanya diperlukan dua nilai, yaitu nilai r dan g . Apabila nilai r dan g telah diketahui, maka nilai b dapat dihitung. Misalnya untuk mendapatkan nilai b , dapat digunakan persamaan berikut.

$$\bar{b} = 1 - \bar{r} - \bar{g} \quad (2.2)$$



Gambar 2.1: Model warna CIE RGB dan XYZ

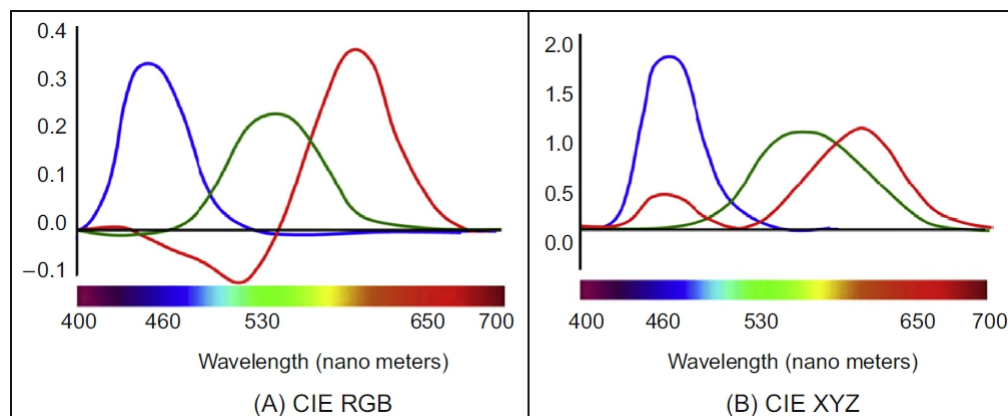
Saat ini, ada beragam standar warna yang digunakan di antaranya (Zakaria, 2020):

1. sRGB merupakan standar ruang warna RGB yang ditentukan oleh HP dan Microsoft dan digunakan oleh perangkat digital saat ini, seperti monitor, printer, dan internet. sRGB merupakan standar warna yang paling umum digunakan karena memiliki cakupan warna yang paling kecil dibandingkan dengan stan-

dar warna lainnya, sehingga hampir semua *device* dari yang terkini hingga yang terlama bisa menampilkan semua warna dengan *value* yang sama.

2. Adobe RGB, dibuat oleh perusahaan Adobe untuk mengatasi kekurangan pada sRGB dan memiliki cakupan warna yang lebih luas. Daerah yang diperluas pada standar warna ini adalah daerah warna hijau. Standar warna ini sering digunakan untuk penyuntingan gambar.
3. NTSC (*National Television Standards Committee*) merupakan standar warna untuk televisi. NTSC dikembangkan untuk memperbaiki ketidakstabilan dan ketidakakuratan CRT monitor atau TV tabung dalam menampilkan warna.
4. DCI-P3 (*Digital Cinema Initiatives*) merupakan standar warna yang dikembangkan untuk kebutuhan industri cinema dan film bioskop dan memiliki cakupan warna 25% lebih banyak dari sRGB. Standar warna ini cocok digunakan untuk *smartphone*, kamera, dan monitor.

Pengembangan model warna RGB dilakukan melalui eksperimen pencocokan warna menggunakan warna primer, yaitu merah, hijau, dan biru dengan masing-masing panjang gelombang 700, 546.1, dan 435.8 nm. Hasil dari eksperimen tersebut adalah banyak warna yang tidak bisa dihasilkan dengan penambahan warna primer, tetapi hanya bisa dilakukan dengan pengurangan nilai cahaya. Dalam prakteknya, eksperimen ini memiliki keterbatasan pada rangkaian warnanya. Untuk mengatasi keterbatasan ini, dibutuhkan sebuah mekanisme yang dapat digunakan untuk menyimpulkan komponen warna yang mungkin sesuai dengan hasil yang diperoleh. Mekanisme ini didasarkan pada fungsi pencocokan warna (Nixon and Aguado, 2020).



Gambar 2.2: *Colour matching function* atau fungsi pencocokan warna

Fungsi pencocokan warna mendefinisikan nilai intensitas warna dasar yang menghasilkan warna monokromatik apapun dengan intensitas yang dinormalisasi. Gambar 2.2A menunjukkan hasil normalisasi untuk CIE RGB dan gambar 2.2B menunjukkan hasil normalisasi untuk CIE XYZ. Pada gambar 2.2A, dapat dilihat bahwa pada CIE RGB terdapat nilai negatif pada warna merah. Nilai negatif ini disebut dengan *lost value*. *Lost value* ini membuat warna merah terlihat seperti memiliki nilai negatif, namun sebenarnya maksud dari *lost value* ini adalah warna merah bernilai 0 atau warna merah menghilang setelah dilakukan normalisasi. Ini merupakan salah satu masalah yang ada pada CIE RGB.

CIE XYZ dikembangkan untuk mengatasi dan memperbaiki masalah yang ada pada CIE RGB. CIE XYZ merupakan sebuah sistem warna yang dapat mencakup semua warna yang dapat dilihat oleh mata manusia. Warna primer dari CIE XYZ merupakan warna imajiner, yaitu warna yang tidak dapat diwujudkan secara fisik. Y merupakan luminansi (kecerahan warna), yang memiliki nilai minimum 0 untuk warna hitam dan nilai maksimum 100 untuk warna putih. Z merespons sebagian besar cahaya yang memiliki panjang gelombang yang pendek, sedangkan X merespons cahaya baik cahaya yang memiliki panjang gelombang yang panjang maupun yang pendek (Paschotta, 2020).

X , Y , dan Z merupakan ekstrapolasi dari nilai RGB yang dibuat secara sistematis untuk menghindari adanya nilai negatif. Dengan ketiga nilai ini, kita dapat menggambarkan *color gamut* atau gamut warna yang dapat mencakup seluruh warna yang ada. *Color gamut* atau gamut warna sendiri adalah suatu rentang warna yang dihasilkan oleh suatu perangkat tertentu sehingga warna tersebut dapat dilihat oleh mata manusia (Wolfcrow, 2021). Seperti yang terlihat pada gambar 2.2B, tidak terdapat nilai negatif pada CIE XYZ setelah dilakukan normalisasi. Selain itu, rentang sensitivitas pada CIE XYZ lebih luas dibandingkan dengan CIE RGB, warna biru memiliki rentang 6 kali lebih besar, warna hijau 5 kali lebih besar, dan warna merah 2,5 kali lebih besar. Namun, terdapat *overlapping* dari warna merah pada warna biru. Hal ini menandakan, bahwa warna biru tidaklah murni biru, tetapi bisa jadi merupakan hasil dari percampuran warna merah dan biru.

Cara lain untuk mewakili CIE XYZ, yaitu dengan menggunakan CIE xyY . Pada CIE xyY , Y merupakan luminansi dan x dan y merupakan nilai krominan yang diturunkan dari nilai tristimulus X , Y , dan Z dalam model warna CIE XYZ. Nilai tristimulus merupakan kombinasi dari tiga koordinat warna primer, di mana setiap koordinat warna menentukan kekuatan kontribusi dari satu warna primer. Nilai x dan y digunakan untuk menggambarkan diagram kromatisitas dari model warna CIE XYZ (Wolfcrow, 2021). Sesuai dengan persamaan 2.1 dan 2.2, maka koordinat kromatisitas untuk CIE XYZ didefinisikan sebagai

$$\bar{x} = \frac{x}{x + y + z}, \bar{y} = \frac{y}{x + y + z}, \bar{z} = \frac{z}{x + y + z} = 1 - \bar{x} - \bar{y} \quad (2.3)$$

Perbedaan antara CIE RGB dan CIE XYZ, yaitu CIE XYZ merupakan model warna yang *device-independent* atau tidak bergantung pada perangkat, di mana setiap warna primer (X , Y , Z) selalu konstan. Sedangkan CIE RGB merupakan model warna yang *device-dependent*, yang di mana di setiap perangkat (monitor, *scanner*,

kamera, dan lain-lain) memiliki versi RGB-nya sendiri. Bahkan untuk ruang warna standar RGB, seperti sRGB didasarkan pada perangkat aktual, seperti layar CRT model lama (Phil, 2021).

Untuk mentransformasi nilai RGB ke dalam model warna CIE XYZ, dapat dilakukan melalui proses transformasi matriks 3×3 . Transformasi ini melibatkan nilai-nilai tristimulus, yang merupakan susunan dari tiga komponen warna. Berbeda dengan CIE RGB, komponen warna dalam model warna XYZ tidak ditentukan melalui eksperimen pencocokan warna, melainkan diperoleh dari komponen model CIE RGB dengan transformasi linier (Nixon and Aguado, 2020). Persamaan untuk mentransformasi CIE RGB ke CIE XYZ adalah sebagai berikut.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (2.4)$$

M merupakan matriks *non-singular* 3×3 . Dengan demikian, maka persamaan untuk mengubah model XYZ ke RGB adalah sebagai berikut.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M^{-1} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (2.5)$$

Persamaan 2.4 mendefinisikan sistem tiga persamaan, yang di mana jika kita mengetahui nilai-nilai komponen CIE RGB dan CIE XYZ, maka kita dapat mensubstitusi nilai-nilai tersebut ke dalam persamaan 2.4 dan mencari nilai M . Maka, untuk mendefinisikan model XYZ, kita hanya perlu mencari nilai dari tiga titik. Penentuan titik-titik ini dilakukan dengan memperhitungkan kriteria yang diperlukan pada diagram kromatisitas. Karena nilai M didefinisikan oleh titik, maka pengembangan

dari model warna XYZ dianggap sebagai suatu pemetaan segitiga. Hal ini ditunjukkan pada gambar 2.1. Pada gambar ini, segitiga putus-putus pada diagram CIE RGB (gambar 2.1B) diubah menjadi segitiga putus-putus pada diagram XYZ (gambar 2.1D). Pada gambar 2.1D, sisi-sisi dari segitiga bertepatan dengan sumbu dari model XYZ dan warna yang terlihat dibatasi oleh segitiga yang berada pada kuadran positif. Dengan menyesuaikan segitiga dengan sumbu XYZ, maka dapat dipastikan bahwa transformasi akan bernilai positif karena letak segitiga berada di kanan dan atas sumbu XYZ ($x > 0$ dan $y > 0$). Maka, menurut definisi kromatisitas pada persamaan 2.2 didapatkan

$$\bar{x} + \bar{y} = 1 - \bar{z} \quad (2.6)$$

Jadi, supaya nilai z berkisar dari nol sampai satu, maka diperlukan nilai

$$\bar{x} + \bar{y} \leq 1 \quad (2.7)$$

Ini berarti bahwa warna harus berada di bawah garis diagonal. Setelah segitiga pada diagram kromatisitas XYZ ditentukan, maka transformasi M pada persamaan 2.4 dapat ditentukan dengan mencari segitiga yang sesuai dengan diagram CIE RGB, yang dapat dilakukan dengan menetapkan kriteria untuk mencari garis yang sesuai dengan kedua diagram. Kriteria pertama, yaitu memberikan kecerahan pada satu komponen.

Pada model XYZ, kecerahan ditentukan oleh nilai Y dan Y disebut sebagai intensitas warna. Hal ini dikarenakan mata manusia lebih sensitif terhadap warna yang panjang gelombangnya mendekati warna hijau dan perubahan kecil di sepanjang sumbu Y akan menghasilkan variasi intensitas yang kuat. Pada CIE RGB, semua komponen berkontribusi pada intensitas warna yang tergantung pada koefisien luminansi dari warna dasar, yaitu $[1 \ 4,59 \ 0,06]$. Maka, fungsi luminositas pada

CIE RGB adalah

$$V = r + 4,59g + 0,06b \quad (2.8)$$

Karena pada model warna XYZ intensitas warna hanya ditentukan oleh komponen Y , maka untuk warna $y = 0$ harus memiliki nilai $V = 0$, sehingga jika nilai $y = 0$, maka

$$r + 4,59g + 0,06b = 0 \quad (2.9)$$

Persamaan 2.9 mendefinisikan suatu bidang yang melewati titik asal pada ruang warna CIE RGB tiga dimensi. Proyeksi ke diagram kromatisitas diperoleh menurut persamaan 2.1, yaitu

$$\begin{aligned} r + g + b &= 1 + 4,59 + 0,06 = 5,65 \\ \bar{r} &= \frac{r}{r + g + b} = \frac{1}{5,65} = 0,17 \\ \bar{g} &= \frac{g}{r + g + b} = \frac{4,59}{5,65} = 0,81 \\ \bar{b} &= \frac{b}{r + g + b} = \frac{0,06}{5,65} = 0,01 \end{aligned} \quad (2.10)$$

Maka didapatkan

$$0,17\bar{r} + 0,81\bar{g} + 0,01 = 0 \quad (2.11)$$

Garis ini melewati titik \bar{o} dan \bar{x} yang ditunjukkan pada gambar 2.1B, dan sesuai dengan garis \overline{ox} pada gambar 2.1D. Warna-warna yang ada pada garis ini disebut dengan *alychne* atau warna dengan *luminance* nol dan terbentuk dari nilai negatif hijau atau merah. Menurut fungsi luminositas dan garis pada diagram kromatisitas, warna-warna ini tidak menghasilkan intensitas yang dirasakan oleh mata manusia dan merupakan warna yang tidak terlihat.

Garis \overline{xy} didefinisikan dengan asumsi bahwa garis yang melewati titik $[1 \ 0]$

dan $[0 \ 1]$ pada diagram kromatisitas CIE RGB dapat memetakan garis diagonal pada diagram kromatisitas XYZ dengan baik. Hal ini dikarenakan garis \overline{xy} dapat memaksimalkan area dari warna yang terlihat dan membatasi area warna yang bersinggungan dengan area warna yang terlihat pada rentang gelombang yang besar. Namun, garis ini tidak mencakup semua warna yang terlihat dan didefinisikan oleh

$$\bar{r} + \bar{g} = 1 \quad (2.12)$$

Di mana titik-titik pada atau dibawah garis harus memenuhi batasan berikut.

$$\bar{r} + \bar{g} \leq 1 \quad (2.13)$$

Warna biru dapat diperoleh menurut nilai z pada persamaan 2.3 dan digunakan dalam persamaan berikut.

$$\bar{r} + \bar{g} = 1 - \bar{b} \quad (2.14)$$

Kendala dari persamaan 2.13 adalah persamaan tersebut akan bernilai benar jika \bar{b} bernilai positif. Namun, terdapat nilai negatif antara 546 dan 600 nm pada fungsi pencocokan warna yang membuat beberapa warna berada di atas garis. Untuk mengatasi kendala ini, kemiringan garis pada persamaan 2.12 digeser sedikit untuk mendapatkan garis yang mendefinisikan model XYZ. Perubahan kecil pada kemiringan menghasilkan garis yang mengandung warna dari komponen biru minimum. Maka, garis kedua yang mendefinisikan model XYZ adalah

$$\bar{r} + 0,99\bar{g} = 1 \quad (2.15)$$

Garis ini ditunjukkan pada gambar 2.1B sebagai garis putus-putus yang melewati titik x dan y . Garis yang sesuai pada diagram kromatisitas XYZ dapat dilihat pada gambar

2.1D.

Garis \overline{oy} pada diagram kromatisitas CIE XYZ didefinisikan untuk memaksimalkan area yang menutupi warna yang terlihat dan didapat dengan menentukan garis tangensial ke titik yang mendefinisikan warna 500 nm. Pada gambar 2.1B, posisi garis ditunjukkan oleh titik \bar{o} dan \bar{y} dan garis ini sesuai dengan sumbu vertikal diagram XYZ. Persamaan garis pada diagram kromatisitas CIE RGB didefinisikan sebagai

$$2,62\bar{r} + 0,99\bar{g} = -0,81 \quad (2.16)$$

Maka, garis-garis yang menentukan segitiga pada diagram CIE RGB adalah persamaan 2.11, 2.15, dan 2.16. Dengan menghitung perpotongan dari ketiga garis ini, maka didapatkan titik puncak segitiga, yaitu $[1,27 \ -0,27]$, $[-1,74 \ 2,76]$, dan $[-0,74 \ 0,14]$. Namun, pada ruang warna CIE RGB diperlukan komponen \bar{b} yang dapat diperoleh dengan menerapkan persamaan 2.2. Maka, koordinat kromatisitas pada CIE RGB adalah $[1,27 \ -0,27 \ 0,002]$, $[-1,74 \ 2,76 \ -0,02]$, dan $[-0,74 \ 0,14 \ 1,6]$ dan komponen warna yang didefinisikan oleh koordinat ini adalah

$$\begin{aligned} \alpha[1,27 \ -0,27 \ 0,002] \\ \beta[-1,74 \ 2,76 \ -0,02] \\ \gamma[-0,74 \ 0,14 \ 1,6] \end{aligned} \quad (2.17)$$

α , β , dan γ menunjukkan konstanta normalisasi. Ketiga nilai ini diperoleh dari tiga titik dalam bentuk $[\bar{r} \ \bar{g}]$.

Segitiga pada model XYZ didefinisikan bertepatan dengan sumbu dan menyebabkan titik-titik pada persamaan 2.17 berubah menjadi $[1 \ 0 \ 0]$, $[0 \ 1 \ 0]$, dan $[0 \ 0 \ 1]$. Transformasi M dapat ditentukan dengan mensubstitusi titik tersebut ke

persamaan 2.4. Namun, terdapat cara yang lebih sederhana, yaitu dengan menggunakan persamaan 2.5 karena titik pada sistem XYZ mengandung nilai 0 pada kedua elemennya. Saat ketiga titik tersebut disubstitusi ke persamaan 2.5, maka didapatkan tiga persamaan

$$\begin{aligned} \alpha \begin{bmatrix} 1,27 \\ -0,27 \\ 0,002 \end{bmatrix} &= M^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \beta \begin{bmatrix} -1,74 \\ 2,76 \\ -0,02 \end{bmatrix} &= M^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \gamma \begin{bmatrix} -0,74 \\ 0,14 \\ 1,6 \end{bmatrix} &= M^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2.18)$$

Ruas kanan pada persamaan pertama merupakan kolom pertama dari M^{-1} , begitu juga ruas kanan pada persamaan kedua dan persamaan ketiga. Maka, persamaannya berubah menjadi

$$M^{-1} = \begin{bmatrix} 1,27\alpha & -1,74\beta & -0,74\gamma \\ -0,27\alpha & 2,76\beta & 0,14\gamma \\ 0,002\alpha & -0,02\beta & 1,6\gamma \end{bmatrix} \quad (2.19)$$

Matriks pada persamaan 2.19 ditulis ulang ke dalam bentuk matriks *product* (bentuk perkalian matriks), yaitu

$$M^{-1} = \begin{bmatrix} 1,27 & -1,74 & -0,74 \\ -0,27 & 2,76 & 0,14 \\ 0,002 & -0,02 & 1,6 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} \quad (2.20)$$

Penentuan konstanta normalisasi dilakukan dengan asumsi bahwa titik referensi warna putih sama pada kedua model. Cara yang lebih sederhana untuk mengubah titik referensi tersebut adalah dengan menggunakan kebalikan dari bentuk matriks *product* pada persamaan 2.20, yaitu

$$M = \begin{bmatrix} 1/\alpha & 0 & 0 \\ 0 & 1/\beta & 0 \\ 0 & 0 & 1/\gamma \end{bmatrix} \begin{bmatrix} 0,90 & 0,50 & 0,37 \\ 0,09 & 0,41 & 0,005 \\ -0,00002 & 0,006 & 0,62 \end{bmatrix} \quad (2.21)$$

Pada CIE RGB, nilai referensi warna putih adalah $[0,33 \ 0,33 \ 0,33]$. Apabila pada model warna XYZ titik ini bernilai sama, maka menurut persamaan 2.4 diperoleh persamaan

$$\eta \begin{bmatrix} 0,33 \\ 0,33 \\ 0,33 \end{bmatrix} = \begin{bmatrix} 1/\alpha & 0 & 0 \\ 0 & 1/\beta & 0 \\ 0 & 0 & 1/\gamma \end{bmatrix} \begin{bmatrix} 0,90 & 0,50 & 0,37 \\ 0,09 & 0,41 & 0,005 \\ -0,00002 & 0,006 & 0,62 \end{bmatrix} \begin{bmatrix} 0,33 \\ 0,33 \\ 0,33 \end{bmatrix} \quad (2.22)$$

Konstanta normalisasi η pada persamaan ini muncul karena terdapat batasan di mana koordinat kromatisitas titik putih harus sama tetapi tidak dengan komponen warnanya. Dengan mensubstitusi ke persamaan 2.1, terlihat bahwa $[0,33 \ 0,33 \ 0,33]$ dan $\eta[0,33 \ 0,33 \ 0,33]$ memiliki nilai kromatisitas yang sama. Apabila persamaan 2.22 dikembangkan, maka didapatkan

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \frac{1}{\eta} \begin{bmatrix} \frac{1}{0,33} & 0 & 0 \\ 0 & \frac{1}{0,33} & 0 \\ 0 & 0 & \frac{1}{0,33} \end{bmatrix} \begin{bmatrix} 0,90 & 0,50 & 0,37 \\ 0,09 & 0,41 & 0,005 \\ -0,00002 & 0,006 & 0,62 \end{bmatrix} \begin{bmatrix} 0,33 \\ 0,33 \\ 0,33 \end{bmatrix} \quad (2.23)$$

Jadi,

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \frac{1}{\eta} \begin{bmatrix} 1,84 \\ 0,52 \\ 0,62 \end{bmatrix} \quad (2.24)$$

Dengan mensubstitusi nilai-nilai tersebut ke persamaan 2.19 dan 2.21, maka didapatkan

$$M = \frac{1}{\eta} \begin{bmatrix} 0,489 & 0,31 & 0,20 \\ 0,17 & 0,81 & 0,01 \\ 0,00 & 0,01 & 0,99 \end{bmatrix}, \quad M^{-1} = \eta \begin{bmatrix} 2,36 & -0,89 & -0,45 \\ -0,51 & 1,42 & -0,088 \\ -0,005 & -0,01 & 1,00 \end{bmatrix} \quad (2.25)$$

Baris kedua dari transformasi digunakan untuk mendapatkan nilai η , yaitu

$$y = \frac{0,17r + 0,81g + 0,01b}{\eta} \quad (2.26)$$

Menurut persamaan 2.8, maka persamaannya berubah menjadi

$$r + 4,59g + 0,06b = \frac{0,17r + 0,81g + 0,01b}{\eta} \quad (2.27)$$

Maka, didapatkan nilai η , yaitu

$$\eta = \frac{0,17r + 0,81g + 0,01b}{r + 4,59g + 0,06b} = 0,17697 \quad (2.28)$$

Dengan demikian, maka transformasi untuk mengubah model RGB ke XYZ menjadi (Szeliski, 2010):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (2.29)$$

Baris kedua dari matriks M mendefinisikan koefisien luminansi dari model CIE RGB dan memberikan kecerahan warna yang dirasakan.

2.2.2 Model Warna L^*a^*b dan L^*u^*v

Model warna XYZ sangat berguna untuk menggambarkan hubungan antara warna-warna yang terlihat. Namun, model XYZ memiliki kekurangan dalam hal keseragaman atau linieritas persepsi, yang berarti perbedaan persepsi antara dua warna pada jarak yang sama dalam kromatisitas dapat sangat berbeda. Dalam praktiknya, keseragaman dan linieritas merupakan hal yang penting apabila kita menggunakan ukuran dari perbedaan warna sebagai indikasi dari seberapa miripnya suatu warna pada sistem visual. Salah satu contoh dari pentingnya menggunakan sistem warna yang seragam, yaitu penggunaan ukuran warna untuk menentukan keakuratan sistem reproduksi warna. Dalam hal ini, kualitas suatu sistem ditentukan oleh seberapa bedanya warna yang sebenarnya dirasakan dengan warna pada diagram kromatisitas.

Ketidakseragaman pada model XYZ biasanya diilustrasikan menggunakan sebuah elips yang diperoleh dari percobaan yang dilakukan oleh MacAdam, yang disebut dengan elips MacAdam. Dalam percobaannya, pengamat diminta untuk menyesuaikan komponen dari suatu warna sampai cocok dengan warna yang ada pada diagram kromatisitas. Hasil dari percobaan tersebut menunjukkan bahwa kemampuan manusia untuk membedakan warna tidak sama untuk semua warna, sehingga jarak dalam diagram kromatisitas bukan merupakan ukuran perbedaan warna yang baik. Idealnya, perbedaan yang diamati harus digambarkan dengan lingkaran yang memi-

liki jari-jari yang sama sehingga jarak antara warna memiliki arti yang sama terlepas dari posisinya pada diagram. Oleh karena itu, dilakukan pencarian model warna lain yang memiliki linieritas yang lebih baik. Model tersebut adalah model warna L^*a^*b dan L^*u^*v (disebut juga CIE LAB dan CIE LUV). Kedua model warna ini mengubah komponen warna dari XYZ agar menjadi lebih seragam.

CIE LUV mengubah komponen warna model XYZ dan didefinisikan pada persamaan berikut.

$$\begin{aligned} u &= \frac{4x}{x + 15y + 3z}, \\ v &= \frac{9y}{x + 15y + 3z} \end{aligned} \quad (2.30)$$

Awalnya, transformasi ini digunakan untuk meningkatkan linieritas persepsi pada model warna sebelumnya. Namun, saat ini transformasi ini dinyatakan sebagai definisi dari model warna CIE LUV. Menurut definisi dari koordinat kromatisitas CIE XYZ, persamaannya berubah menjadi

$$\begin{aligned} x &= \bar{x}(x + y + z), \\ y &= \bar{y}(x + y + z), \\ \bar{z} &= \frac{z}{x + y + z} \\ &= 1 - \bar{x} - \bar{y} \end{aligned} \quad (2.31)$$

Dengan mensubstitusi persamaan 2.31 ke persamaan 2.30, maka didapatkan nilai u , yaitu

$$\begin{aligned}
u &= \frac{4(\bar{x}(x+y+z))}{\bar{x}(x+y+z) + 15(\bar{y}(x+y+z)) + 3(\bar{z}(x+y+z))} \\
&= \frac{4\bar{x}(x+y+z)}{(x+y+z)(\bar{x} + 15\bar{y} + 3\bar{z})} \\
&= \frac{4\bar{x}}{\bar{x} + 15\bar{y} + 3\bar{z}} \\
&= \frac{4\bar{x}}{\bar{x} + 15\bar{y} + 3(1 - \bar{x} - \bar{y})} \\
&= \frac{4\bar{x}}{\bar{x} + 15\bar{y} + 3 - 3\bar{x} - 3\bar{y}} \\
&= \frac{4\bar{x}}{-2\bar{x} + 12\bar{y} + 3}
\end{aligned} \tag{2.32}$$

Dan nilai v , yaitu

$$\begin{aligned}
v &= \frac{9(\bar{y}(x+y+z))}{\bar{x}(x+y+z) + 15(\bar{y}(x+y+z)) + 3(\bar{z}(x+y+z))} \\
&= \frac{9\bar{y}(x+y+z)}{(x+y+z)(\bar{x} + 15\bar{y} + 3\bar{z})} \\
&= \frac{9\bar{y}}{\bar{x} + 15\bar{y} + 3\bar{z}} \\
&= \frac{9\bar{y}}{\bar{x} + 15\bar{y} + 3(1 - \bar{x} - \bar{y})} \\
&= \frac{9\bar{y}}{\bar{x} + 15\bar{y} + 3 - 3\bar{x} - 3\bar{y}} \\
&= \frac{9\bar{y}}{-2\bar{x} + 12\bar{y} + 3}
\end{aligned} \tag{2.33}$$

Model warna LUV menggunakan titik referensi warna putih yang dilambangkan dengan $[u_n, v_n]$, yang dapat diperoleh dengan menggunakan persamaan 2.32 dan 2.33. Misal, titik $[u_n, v_n]$ dicari dengan menggunakan *Standard Observer* 2° dan *Standard Illuminant C* dengan nilai $x = 0,31006$ dan $y = 0,31616$ (Pascale, 2003), maka didapatkan

$$\begin{aligned}
u_n &= \frac{4 \times 0,31006}{-2 \times 0,31006 + 12 \times 0,31616 - 3} = 0,2009 \\
v_n &= \frac{9 \times 0,31006}{-2 \times 0,31006 + 12 \times 0,31616 - 3} = 0,4610
\end{aligned} \tag{2.34}$$

Kedua titik referensi tersebut digunakan untuk menentukan komponen warna pada model LUV, yaitu

$$u^* = 13L^*(u - u_n), \quad v^* = 13L^*(v - v_n) \tag{2.35}$$

di mana u^* merupakan warna pada sumbu merah sampai hijau, v^* merupakan warna pada sumbu kuning sampai biru, dan L^* merupakan *lightness* atau kecerahan warna. Nilai untuk L^* , yaitu

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 \left(\frac{y}{y_n}\right)^{1/3} & , \left(\frac{y}{y_n}\right) \leq \left(\frac{29}{3}\right)^3 \\ 116 \left(\frac{29}{3}\right)^{1/3} - 16 & , otherwise \end{cases} \tag{2.36}$$

Nilai y , u , dan v pada persamaan 2.35 dan 2.36 merupakan besaran stimuli warna dan nilai y_n merupakan *luminance*. Nilai konstanta pada persamaan 2.35 menghasilkan rentang nilai antara -134 dan 220 untuk nilai u^* dan dari -140 sampai 122 untuk v^* . Persamaan 2.36 membuat persepsi kecerahan menjadi lebih seragam.

Model warna CIE LAB merupakan alternatif dari model LUV. CIE LAB menggunakan transformasi yang sama untuk kecerahannya, tapi mengubah cara warna dinormalisasi dengan memerhatikan titik referensi. Definisi dari model warna CIE LAB adalah sebagai berikut.

$$\begin{aligned}
L^* &= 116f\left(\frac{y}{y_n}\right) - 16 \\
a^* &= 500\left(f\left(\frac{x}{x_n}\right) - f\left(\frac{y}{y_n}\right)\right) \\
b^* &= 200\left(f\left(\frac{y}{y_n}\right) - f\left(\frac{z}{z_n}\right)\right)
\end{aligned} \tag{2.37}$$

di mana $s = \frac{y}{y_n}$, maka

$$f(s) = \begin{cases} \frac{1}{3} \left(\frac{29}{6}\right)^2 s + \frac{16}{116} & , s \leq \left(\frac{6}{29}\right)^3 \\ s^{1/3} & , otherwise \end{cases} \tag{2.38}$$

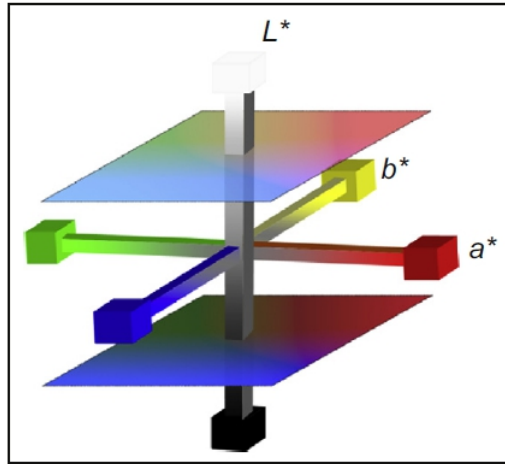
Pada persamaan 2.37 dan 2.38, x_n , y_n , dan z_n merupakan titik referensi warna putih dari CIE XYZ. Dengan menggunakan *Standard Illuminant* D65 dan dengan normalisasi $y = 100$, maka nilai dari x_n , y_n , dan z_n adalah

$$x_n = 95,0489, y_n = 100, z_n = 108,8840 \tag{2.39}$$

Dan nilai x_n , y_n , dan z_n dengan menggunakan *Standard Illuminant* D50, yaitu

$$x_n = 96,4212, y_n = 100, z_n = 82,5188 \tag{2.40}$$

Pendefinisian dari L^* sama dengan model LUV, namun normalisasi dengan menggunakan titik referensi pada LUV dan LAB berbeda. CIE LUV menggunakan pengurangan sedangkan LAB membagi koordinat warna dengan titik referensi. Nilai a^* disebut krominan merah atau hijau dan b^* disebut krominan kuning atau biru, keduanya memiliki nilai positif dan negatif, tidak memiliki batas, dan meluas ke warna-warna yang tidak terlihat oleh mata manusia. Namun, untuk representasi digital kisarannya dibatasi antara -127 dan 127.



Gambar 2.3: Model warna CIE LAB

Gambar 2.3 mengilustrasikan diagram LAB. Diagram LAB merupakan diagram tiga dimensi yang ditentukan oleh luminositas a^* , b^* , dan terang-gelap warna. Pada gambar ini, kecerahan L^* ditunjukkan oleh sumbu atas-bawah dan berkisar dari hitam hingga putih. Dua sumbu lainnya mewakili nilai merah atau hijau dan kuning atau biru. Nilai negatif pada a^* menunjukkan warna hijau dan nilai positif menunjukkan warna magenta (merah). Untuk b^* , nilai negatif menunjukkan warna kuning dan nilai positif menunjukkan warna biru.

Komponen warna pada model warna XYZ dari nilai LUV dan LAB dapat diperoleh dengan meng-*inverse* persamaan yang mendefinisikan transformasi. Untuk mendapatkan koordinat kromatik suatu warna dari koordinat LUV, dapat dilakukan dengan meng-*inverse* persamaan 2.32, 2.33 dan 2.35, yaitu

$$\bar{x} = \frac{9u}{6u - 16v + 12}, \quad \bar{y} = \frac{4v}{6u - 16v + 12} \quad (2.41)$$

Dan

$$u = \frac{u^*}{13L^*} + u_n, \quad v = \frac{v^*}{13L^*} + v_n \quad (2.42)$$

Untuk model warna LAB, koordinat XYZ dapat ditentukan dengan meng-

inverse persamaan 2.37 dan 2.38. Persamaannya adalah sebagai berikut.

$$\begin{aligned} y &= y_n f^{-1} \left(\frac{L^* + 16}{116} \right) \\ x &= x_n f^{-1} \left(\frac{L^* + 16}{116} + \frac{a^*}{500} \right) \\ b^* &= y_n f^{-1} \left(\frac{L^* + 16}{116} + \frac{b^*}{200} \right) \end{aligned} \quad (2.43)$$

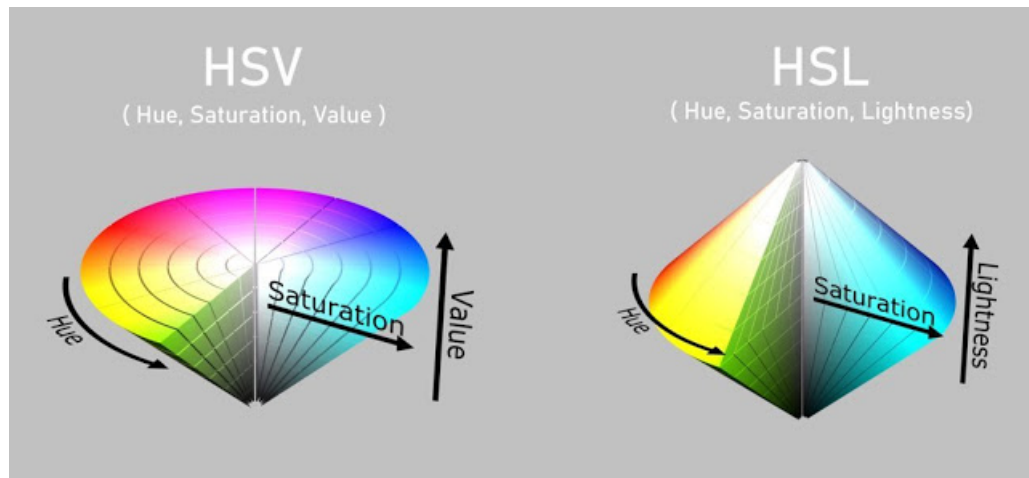
Untuk

$$f^{-1}(s) = \begin{cases} 3 \left(\frac{29}{6} \right)^2 \left(s - \frac{16}{116} \right) & , s \leq \frac{6}{29} \\ s^3 & , otherwise \end{cases} \quad (2.44)$$

Model LUV, LAB, dan XYZ memiliki warna yang sama dan ketiganya merepresentasikan warna yang dapat dilihat oleh mata manusia. Namun, model warna LUV dan LAB memiliki linieritas yang lebih baik, memberikan pengaturan yang mendekati cara kita melihat perbedaan warna, dan lebih cocok dengan persepsi manusia tentang cahaya (Nixon and Aguado, 2020). Antara model LUV dan LAB, model LAB lebih sederhana dibandingkan dengan model LUV karena transformasinya lebih mudah dihitung dan hasil dari transformasinya pun lebih sederhana.

2.2.3 Model warna HSV dan HLS

HSV (*Hue, Saturation, Value*) dan HLS (*Hue, Lightness, Saturation*) merupakan model warna perseptual yang paling umum. HSV disebut juga sebagai HSI (*Hue, Saturation, Intensity*) atau sebagai HSB (*Hue, Saturation, Brightness*). HSV dan HLS menggunakan dua komponen untuk mendefinisikan *hue* (warna) dan *saturation* (saturasi atau kejenuhan) dari warna, namun mereka menggunakan konsep yang berbeda untuk mendefinisikan komponen yang mewakili *brightness* (kecerahan) (Nixon and Aguado, 2020).



Gambar 2.4: Model warna HSV dan HLS

Hue merupakan karakteristik warna berdasarkan cahaya yang dipantulkan. Dalam *color wheel* atau lingkaran warna, *hue* memiliki rentang dari 0° sampai 360° . 0° adalah warna merah, 60° adalah warna kuning, 120° adalah warna hijau, 180° adalah warna *cyan*, 240° adalah warna biru, dan 300° adalah warna magenta. *Saturation* merupakan intensitas warna atau sederhananya merupakan kepekatan atau lemah-kuatnya sebuah warna. Rentang intensitas suatu warna dinyatakan dalam bentuk persen (%), yaitu dari 0% sampai 100%. 0% menunjukkan warna *grey* dan 100% menunjukkan bahwa warna tersebut sangat cerah (*full saturation*).

Perbedaan antara HSV dan HLS terletak pada *Value* dan *Lightness*, seperti yang ditunjukkan pada gambar 2.4. *Value* dan *Lightness* merupakan nilai yang memberitahu seberapa gelap atau cerahnya suatu warna. Rentang dari keduanya dinyatakan dalam bentuk persen (%), yaitu dari 0% sampai 100%. Perbedaan di antara keduanya, yaitu *lightness* mempercerah setiap warna (*hue*) yang dipilih dari tingkat paling bawah (hitam) sampai ke tingkat yang paling terang (putih). Sedangkan *value* hanya mempercerah *hue* dari posisi bawah (hitam) sampai ke titik maksimal *Hue* atau warna yang dipilih.

Model warna HSV dan HLS menggunakan metode yang lebih sederhana di

mana nilai *hue* dan saturasi dihitung langsung dari koordinat RGB tanpa harus di-transformasi terlebih dahulu ke model XYZ. Kedua model warna ini berguna untuk operasi yang mengategorikan warna dan pergantian warna otomatis.

Model warna HSV dan HLS didefinisikan menggunakan geometri *hexagonal* dan segitiga. Geometri *hexagonal* mendefinisikan HSV dan geometri segitiga mendefinisikan HLS. Kedua geometri ini mendefinisikan kecerahan dengan menggunakan bidang normal di sepanjang garis yang mendefinisikan abu-abu. Pada model *hexagonal*, bidang didefinisikan sebagai proyeksi dari sub-kubus dalam kubus warna RGB sedangkan pada model segitiga bidang ditentukan oleh tiga titik dalam sumbu RGB.

Untuk mentransformasi nilai dari RGB ke HLS, pertama tentukan masing-masing nilai warna RGB yaitu *R*, *G*, dan *B*, dan nilai yang dimasukkan harus berada dalam rentang 0 sampai 1. Selanjutnya, carilah nilai maksimum dan nilai minimum dari nilai RGB yang dimasukkan. Sudut *hue* pada HLS yang berada dalam rentang 0° sampai 360° , dapat diperoleh dengan menggunakan rumus sebagai berikut (Swedia and Cahyanti, 2010):

$$h(hue) = \begin{cases} 0 & , \text{jika } max = min \\ 60^\circ \times \left(\frac{G - B}{max - min} \bmod 6 \right) & , \text{jika } max = R \\ 60^\circ \times \left(\frac{B - R}{max - min} + 2 \right) & , \text{jika } max = G \\ 60^\circ \times \left(\frac{G - B}{max - min} + 4 \right) & , \text{jika } max = B \end{cases} \quad (2.45)$$

Nilai *saturation* dan *lightness* (kecerahan cahaya) pada HLS dengan koordinat *s* dan *l* dalam rentang 0 sampai 1, dapat diperoleh dengan menggunakan rumus sebagai berikut (Fishkin, 1990):

$$l(lightness) = \frac{1}{2}(max + min) \quad (2.46)$$

$$s(\text{saturation}) = \begin{cases} 0 & , \text{jika } \max = \min \\ \frac{\max - \min}{2L} & , \text{jika } l \leq 0.5 \\ \frac{\max - \min}{2 - (2L)} & , \text{jika } l \geq 0.5 \end{cases} \quad (2.47)$$

Rumus 2.46 dan 2.47 menghasilkan nilai *lightness* dan *saturation* dalam rentang 0 sampai 1. Untuk memperoleh nilai dalam rentang 0 sampai 255, maka kalikan hasil dari nilai *hue*, *saturation*, dan *lightness* dengan 255.

Untuk mentransformasi nilai RGB ke HSV, langkah-langkah yang dilakukan dan rumus yang digunakan untuk memperoleh sudut *hue* sama seperti saat mentransformasi nilai dari RGB ke HLS. Perbedaannya terletak pada pendefinisian nilai *s* dan *v*. Pendefinisian dari nilai *s* dan *v* pada HSV ditunjukkan pada persamaan berikut (Fishkin, 1990):

$$s(\text{saturation}) = \begin{cases} 0 & , \text{jika } \max = \min \\ \frac{\max - \min}{V} & , \text{otherwise} \end{cases} \quad (2.48)$$

$$V(\text{value}) = \max \quad (2.49)$$

Rumus 2.48 dan 2.49 menghasilkan nilai *value* dan *saturation* dalam rentang RGB 0 sampai 1. Untuk memperoleh nilai dalam rentang RGB 0 sampai 255, maka kalikan hasil dari nilai *hue*, *saturation*, dan *value* dengan 255.

Secara umum, model *hexagonal* (HSV) lebih mudah digunakan dan lebih sederhana karena transformasinya mudah dihitung. Namun dalam hal kecerahan, model HLS lebih fleksibel dan pendistribusian warnanya pun lebih baik dibandingkan dengan model *hexagonal* khususnya dalam pengaplikasian pemrosesan gambar.

2.3 Segmentasi Citra (*Image Segmentation*)

Segmentasi citra merupakan proses mempartisi gambar digital menjadi beberapa segmen (set piksel, disebut juga superpiksel) yang memiliki fitur atau atribut yang sama. Segmentasi bertujuan untuk menyederhanakan atau mengubah representasi gambar menjadi sesuatu yang lebih bermakna dan lebih mudah untuk dianalisis. Segmentasi citra biasa digunakan untuk menemukan objek dan batas (garis, kurva, dan lain-lain) dalam gambar (Shmmla and Ashour, 2013).

Segmentasi citra akan menghasilkan objek-objek dalam citra, atau menghasilkan objek yang sudah dipisah dengan latar belakang. Dalam proses segmentasi, citra digital dipartisi menjadi sekumpulan piksel berdasarkan wilayah piksel yang memiliki karakteristik yang sama seperti kesamaan pada warna, intensitas, tekstur, dan bentuk. Kesamaan karakteristik ini dapat digunakan untuk menemukan dan mengidentifikasi objek dan batas dalam suatu citra (Mohanty et al., 2013). Terdapat empat metode yang digunakan dalam proses segmentasi citra, yaitu:

1. *Thresholding*, yaitu pengubahan citra menjadi citra biner dengan mengelompokkan derajat keabuan menjadi dua, yaitu hitam dan putih. *Thresholding* dilakukan dengan memilih nilai T pada lembah terdalam pada histogram yang mampu membagi dua puncak.
2. *Region growing*, yaitu metode segmentasi dengan menumbuhkan *region* yang kecil menjadi *region* yang besar. Penumbuhan *region* dimulai dengan memilih sebuah piksel sebagai umpan untuk satu *region*. Kemudian *region* berkembang dengan menambahkan piksel yang mirip dengan piksel umpan. Perkembangan *region* berhenti apabila sudah tidak ada piksel yang mirip dengan piksel umpan.
3. *Split and merge*, yaitu metode segmentasi yang menerapkan algoritma *divide and conquer*. Pada metode ini, citra dibagi menjadi sejumlah *region* yang tidak

terkait. Kemudian, *region* bertetangga yang homogen digabung. Pembagian *region* dilakukan terus sampai tidak ada *region* yang dapat digabung.

4. *Clustering*, yaitu metode segmentasi yang mengelompokkan nilai-nilai piksel ke dalam k *cluster*. Pengelompokan dilakukan berdasarkan kesamaan dari nilai-nilai piksel (Teresa, 2019). Pembahasan lebih lanjut mengenai *clustering* akan dibahas pada sub-bab 2.4.

2.4 *Clustering*

Clustering adalah partisi dari kumpulan data atau objek menjadi kelompok atau *cluster*. *Cluster* dapat didefinisikan sebagai kumpulan benda yang menyerupai satu sama lain dan berbeda dengan objek lainnya (Shmmla and Ashour, 2013). Tujuan dari *clustering* adalah mengelompokkan data yang memiliki ciri yang sama dan memisahkan data yang memiliki ciri yang berbeda ke dalam *cluster* yang berbeda.

Terdapat banyak algoritma dan metode *clustering* dalam beberapa literatur. Secara umum, metode *clustering* dapat dibagi menjadi dua, yaitu *hierarchial methods* dan *partitioning methods*. *Hierarchial methods* merupakan metode yang mengelompokkan set data ke dalam bentuk hirarki atau pohon *cluster*, di mana di setiap iterasinya dilakukan penggabungan dua kelompok terdekat (Han et al., 2011). Beberapa algoritma yang termasuk ke dalam metode *hierarchial clustering*, yaitu *Single Linkage*, *Complete Linkage*, *Average Linkage*, dan *Average Group Linkage*.

Sedangkan *partitioning methods* merupakan metode *clustering* yang mengelompokkan suatu set data menjadi beberapa *cluster* di mana setiap *cluster* memiliki titik pusat (*centroid*) (Han et al., 2011). Metode ini bertujuan untuk meminimalkan jarak dari seluruh data ke pusat masing-masing *cluster*. Beberapa algoritma yang termasuk ke dalam metode *partitionial clustering*, yaitu *K-Means*, *Fuzzy K-Means*, dan *Mixture Modelling*.

2.5 *K-means*

K-means merupakan metode pengelompokan dari sejumlah *cluster* yang terpisah. "*K*" mengacu pada jumlah *cluster* yang ditentukan (Yadav et al., 2013). Metode *k-means* mempartisi *n*-set input data menjadi *k-cluster* di mana setiap set input data termasuk ke dalam *cluster* dengan *mean* terdekat (Zheng and Liu, 2012). Metode ini akan mempartisi data yang berkarakteristik sama ke dalam suatu kelompok yang sama dan data yang lainnya ke kelompok yang berbeda (Gustientiedina et al., 2019).

Diberikan set input data pengamatan (x_1, x_2, \dots, x_n) , di mana setiap set input data merupakan vektor berdimensi-*d*, *k-means* akan mempartisi *n*-set input data menjadi himpunan-*k*, yaitu $S = \{S_1, S_2, \dots, S_k\}$, di mana $k \leq n$, dan untuk mencari nilai minimum pada *cluster*, yaitu (Zheng and Liu, 2012):

$$\arg \min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - u_i\|^2 \quad (2.50)$$

di mana $\|x_j - u_i\|^2$ adalah jarak *Euclidean* antara x_j dan u_i , x_j adalah data ke-*j* dalam *cluster* S_i , dan u_i adalah nilai *mean* atau pusat dari data pada *cluster* S_i .

Algoritma *k-means* dilakukan secara berulang. Pertama, secara acak ditetapkan *cluster* untuk setiap set input data dan kemudian dihitung jarak *Euclidean* dari setiap kumpulan data. Jarak *Euclidean* merupakan jarak geometris antar dua objek data dan digunakan untuk mengukur jarak dari objek data ke pusat *cluster*. Data akan semakin mirip apabila jarak antara kedua objek data tersebut semakin dekat (Johnson and Wichern, 2002).

Algoritma standar untuk menghitung rata-rata awal menjadi pusat *cluster* yang ditetapkan secara acak dan menempatkan semua data dekat dengan pusat kumpulan data adalah sebagai berikut:

Diberikan himpunan awal rata-rata k $m_1^{\{1\}}, \dots, m_k^{\{1\}}$, algoritma dijalankan se-

cara bergantian ke dua langkah, yaitu (MacKay, 2003):

1. **Assignment Step:** Menetapkan setiap set input data ke *cluster* dengan *mean* terdekat.

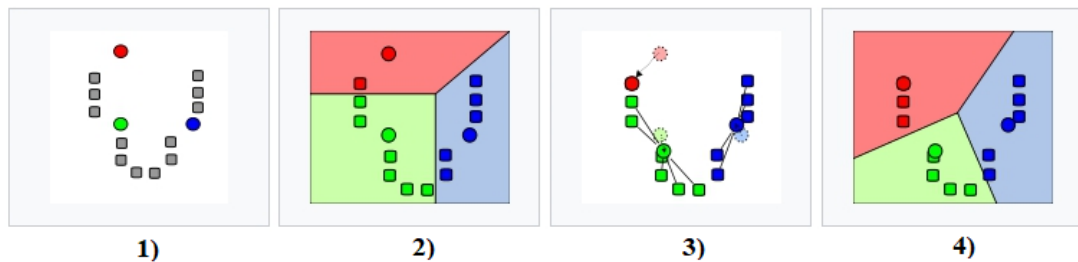
$$S_i^{(t)} = \{x_j : \|x_j - \mathbf{m}_i^{(t)}\| \leq \|x_j - \mathbf{m}_{i^*}^{(t)}\|, i^* = 1, \dots, k\} \quad (2.51)$$

di mana setiap x_j ditetapkan ke tepat satu $S^{(t)}$, meskipun jika x_j dapat ditetapkan ke dua atau lebih dari $S^{(t)}$.

2. **Update Step:** Menghitung rata-rata baru untuk menjadi pusat dari data input dalam *cluster*.

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (2.52)$$

Ulangi *assignment step* dan *update step* hingga *assignments* tidak berubah lagi dan algoritma dianggap telah selesai. *Assignment step* disebut sebagai langkah ekspektasi, sedangkan *update step* merupakan langkah pemaksimalan. Demonstrasi dari algoritma standar pada *K-means* dengan $k = 3$ ditunjukkan pada gambar 2.5 (Zheng and Liu, 2012).



Gambar 2.5: 1) k inisial "means" dipilih secara acak dari data set, 2) k cluster dibuat dengan mengaitkan setiap observasi dengan *mean* terdekat, 3) *centroid* (pusat) dari masing-masing k cluster menjadi *mean* baru, 4) ulangi tahap 2 dan 3 sampai konvergensi tercapai

Langkah pertama yang dilakukan dalam proses segmentasi menggunakan k -

means, yaitu menentukan jumlah *cluster* atau nilai k pada citra hasil *pre-processing* dan nilai *centroid* atau pusatnya secara acak. Selanjutnya, jarak piksel ke *centroid* dihitung dan piksel dikelompokkan berdasarkan jarak terdekat. Setelah itu, nilai *centroid* dihitung kembali sebagai *centroid* baru dan piksel dikelompokkan kembali sesuai dengan *centroid* baru tersebut. Proses *clustering* akan berakhir apabila sudah tidak ada piksel yang berpindah *cluster* (Ferdinanto et al., 2018).

Permasalahan utama dari *k-means* adalah ketergantungannya pada inisialisasi k . Jika k awal tidak dipilih dengan hati-hati, komputasi akan menjalankan kemungkinan konvergen ke minimum lokal daripada solusi minimum global. Hal ini dikarenakan algoritma *k-means* tidak menjamin konvergensi ke nilai optimal global dan hasilnya mungkin saja tergantung pada *cluster* awal. Jadi, menginisialisasi k dengan tepat dapat memberikan efek besar pada *k-means*.

Terdapat beberapa metode yang dapat digunakan untuk menentukan nilai k terbaik pada algoritma *k-means*, di antaranya adalah metode *Elbow* dan metode *Silhouette*. Metode *Elbow* menentukan jumlah *cluster* (k) terbaik dengan melihat presentase hasil perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Perbandingan didapatkan dengan menghitung SSE (*Sum of Square Error*) dari masing-masing nilai *cluster*. Nilai *cluster* terbaik diambil dari nilai SSE yang mengalami penurunan yang signifikan dan berbentuk siku pada grafik (Rahman et al., 2017). Nilai SSE dapat diperoleh dengan menggunakan persamaan berikut.

$$SSE = \sum_{k=1}^k \sum_{x_i \in S_k} \|x_i - C_k\|^2 \quad (2.53)$$

dimana k merupakan banyaknya *cluster* yang digunakan pada algoritma *k-means*, x_i adalah data k -i dan C_k adalah pusat dari *cluster*.

Metode *Silhouette* menghitung *silhouette coefficient* dari setiap titik yang meng-

ukur seberapa mirip suatu titik dengan *cluster*-nya sendiri dibandingkan dengan *cluster* lainnya dengan memberikan representasi grafis tentang seberapa baik setiap objek telah diklasifikasikan. *Silhouette* merupakan ukuran dari bagaimana algoritmanya telah bekerja, yang memiliki rentang dari -1 hingga $+1$. Nilai *cluster* terbaik ditentukan oleh nilai rata-rata *silhouette*, dimana *cluster* dengan nilai rata-rata *silhouette* tertinggi yang merupakan nilai *cluster* terbaik (Kumar, 2020). *Silhouette coefficient* dapat diperoleh dengan menggunakan persamaan berikut (Dewi and Pramita, 2019).

$$SI_i^j = \frac{b_i^j - a_i^j}{\max(a_i^j, b_i^j)} \quad (2.54)$$

dimana SI_i^j merupakan *silhouette coefficient* data ke- i pada *cluster*- j , b_i^j merupakan rata-rata jarak data ke- i terhadap semua data yang tidak satu *cluster* dengan data ke- i , dan a_i^j merupakan rata-rata jarak ke- i terhadap semua data yang satu *cluster* dengan data ke- i .

2.6 Mean shift

Mean shift adalah teknik analisis *non-parametric feature space* untuk mencari nilai maksimum dari fungsi kerapatan atau kepadatan yang diberikan dari data diskrit yang ada di fungsi tersebut. *Mean shift* merupakan prosedur berulang (iteratif) sederhana yang menggeser setiap titik data ke rata-rata (*mean*) titik data di daerahnya. Algoritma *mean shift* disebut juga sebagai algoritma pencarian mode (Cheng, 1995).

Langkah pertama untuk menerapkan *mean shift* adalah merepresentasikan data secara matematis, yang berarti data direpresentasikan sebagai titik. *Mean shift* dibangun dengan konsep *kernel density estimation* (KDE). KDE merupakan metode untuk memperkirakan distribusi yang mendasari (disebut juga fungsi kepadatan probabilitas) untuk sekumpulan data. KDE bekerja dengan menempatkan kernel pada

setiap titik dalam kumpulan data (Nedrich, 2015).

Fungsi kepadatan probabilitas (*probability density function*) merupakan fungsi kepadatan pada variabel acak kontinu yang nilainya pada suatu sampel atau titik tertentu dalam ruang sampel (kumpulan nilai yang mungkin diambil oleh variabel acak) dapat dijelaskan sebagai nilai yang memberikan kemungkinan relatif bahwa nilai variabel acak akan sama dengan sampel tersebut. Fungsi kepadatan probabilitas dari variabel acak kontinu X dalam himpunan S adalah fungsi integral $f(x)$ yang memenuhi (Eberly, 2021):

1. $f(x)$ selalu bernilai positif di setiap S , dimana $f(x) > 0$ untuk semua x dalam himpunan S
2. Area di bawah kurva $f(x)$ dalam himpunan S bernilai 1, yaitu:

$$\int_S f(x)dx = 1 \quad (2.55)$$

3. Jika $f(x)$ adalah suatu fungsi kepadatan probabilitas, maka probabilitas bahwa x adalah anggota A di mana A adalah suatu interval, diberikan fungsi integral $f(x)$ pada interval tersebut, yaitu:

$$P(X \in A) = \int_A f(x)dx \quad (2.56)$$

Dimungkinkan untuk mewakili suatu variabel acak diskrit tertentu serta variabel acak yang melibatkan bagian kontinu dan bagian diskrit dengan fungsi kepadatan probabilitas, dengan menggunakan fungsi delta Dirac. Fungsi delta Dirac merupakan fungsi atau distribusi umum atas bilangan riil yang nilainya selalu 0 kecuali di titik 0, dan nilai integralnya di seluruh garis riil sama dengan satu. Sebagai contoh, anggaplah sebuah variabel acak diskrit biner memiliki distribusi Rademacher, yaitu

suatu distribusi yang menggunakan -1 dan 1 sebagai *value* dengan probabilitas dari masing-masing *value* yaitu $\frac{1}{2}$. Maka, kepadatan probabilitas dari variabel tersebut adalah

$$f(t) = \frac{1}{2}(\delta(t + 1) + \delta(t - 1)) \quad (2.57)$$

Umumnya, jika suatu variabel diskrit bisa mengambil n nilai yang berbeda sepanjang bilangan riil, maka fungsi kepadatan probabilitasnya adalah

$$f(t) = \sum_{i=1}^n p_i \delta(t - x_i) \quad (2.58)$$

dimana x_1, \dots, x_n adalah nilai diskrit dari variabel dan p_1, \dots, p_n adalah probabilitas yang terkait dengan nilai tersebut. Hal ini secara substansial menyatukan penggunaan dari distribusi probabilitas diskrit dan kontinu.

Misalkan dalam beberapa kepadatan probabilitas $p(x)$ yang tidak diketahui yang berada di dalam ruang berdimensi- D (dianggap sebagai *Euclidean*), yaitu R , diambil vektor pengamatan x . Maka probabilitas bahwa di dalam suatu daerah R akan terdapat vektor x adalah

$$P = \int_R p(x) dx \quad (2.59)$$

Misalkan dari distribusi probabilitas $p(x)$ diambil satu set data yang terdiri dari N pengamatan. Karena di dalam daerah R setiap titik memiliki probabilitas P , maka jumlah titik K yang berada di dalam daerah R akan didistribusikan sesuai dengan distribusi binomial.

$$Bin(K|N, P) = \frac{N!}{K!(N - K)!} P^K (1 - P)^{1-K} \quad (2.60)$$

Dari persamaan 2.60, dapat diperoleh bahwa rata-rata (*mean*) dari titik yang

berada di daerah tersebut adalah $E[K/N] = P$ dan variansi di sekitar *mean* tersebut adalah $var[K/N] = P(1-P)/N$. Untuk nilai N besar, distribusi ini akan memuncak tajam di sekitar *mean*, sehingga

$$K \simeq NP \quad (2.61)$$

Jika diasumsikan bahwa probabilitas $p(x)$ konstan di daerah R yang disebabkan karena daerah R cukup kecil, maka didapatkan

$$P \simeq p(x)V \quad (2.62)$$

di mana V adalah volume dari daerah R . Dengan mensubstitusikan persamaan 2.62 ke persamaan 2.61, maka estimasi dari fungsi distribusi probabilitas diperoleh dalam bentuk

$$\begin{aligned} K &\simeq NP \\ K &= Np(x)V \\ p(x) &= \frac{K}{NV} \end{aligned} \quad (2.63)$$

Persamaan 2.63 dapat dikembangkan dengan dua cara. Pertama, dengan menetapkan nilai K dan kemudian menentukan nilai V dari data, hal ini yang mendasari metode *k-nearest-neighbour*. Kedua, dengan menetapkan V dan kemudian menentukan nilai K dari data, hal inilah yang mendasari metode *kernel density estimation* (KDE).

Diasumsikan daerah R sebagai *hypercube* kecil berdimensi- D yang berpusat pada titik x . Untuk menghitung jumlah K titik yang termasuk ke dalam wilayah R , ditentukan menggunakan fungsi sebagai berikut.

$$k(u) = \begin{cases} 1 & , |u_i| \leq \frac{1}{2} \\ 0 & , otherwise \end{cases} \quad (2.64)$$

dengan $i = 1, \dots, D$. Fungsi ini merupakan contoh dari fungsi kernel, yang disebut juga sebagai *Parzen window*. Dari persamaan 2.62, $k((x - x_n)/h)$ akan bernilai 1 jika data x_n berada dalam *cube* yang bersisi h dan berpusat pada x , dan akan bernilai 0 jika tidak. Maka, jumlah total data yang berada dalam *cube* ini adalah

$$K = \sum_{n=1}^N k\left(\frac{x - x_n}{h}\right) \quad (2.65)$$

Substitusikan persamaan 2.65 ke persamaan 2.63, maka didapatkan persamaan an fungsi distribusi dari KDE, yaitu

$$P_{KDE}(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^D} k\left(\frac{x - x_i}{h}\right) \quad (2.66)$$

di mana N adalah banyaknya sampel pengamatan, k adalah fungsi kernel non-negatif, D adalah dimensi variabel pengamatan, x adalah nilai data, dan $h > 0$ adalah parameter penghalusan yang disebut dengan *bandwidth* (Bishop, 2006).

Bandwidth atau lebar pita merupakan parameter yang digunakan untuk menghitung sebuah data akan berada pada *cluster* tersendiri atau bergabung dengan *cluster* yang ada, sedangkan kernel merupakan istilah matematika untuk mendefinisikan fungsi pembobotan. Ada berbagai macam jenis kernel, namun yang paling terkenal adalah Gaussian kernel. Dalam matematika, fungsi Gaussian merupakan fungsi dari bentuk:

$$f(x) = a \cdot \exp\left(-\frac{(x - b)^2}{2c^2}\right) \quad (2.67)$$

untuk konstanta riil sembarang a , b , dan $c \neq 0$. Grafik dari Gaussian merupakan sebuah kurva berbentuk lonceng. Parameter a adalah ketinggian puncak kurva, b adalah posisi puncak, dan c adalah standar deviasi yang disebut juga dengan lebar Gaussian RMS. Fungsi gaussian sering digunakan untuk merepresentasikan fungsi

kepadatan probabilitas dari variabel acak yang terdistribusi normal dengan nilai $\mu = b$ dan varian $\sigma^2 = c^2$. Maka, fungsi Gaussian berbentuk (Squires, 2001):

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right) \quad (2.68)$$

Misalkan S adalah data dari himpunan berhingga dalam ruang *Euclidean* berdimensi- n , yaitu X . Diberikan fungsi kernel $K(s - x)$, yang menentukan bobot dari titik terdekat untuk menghitung ulang nilai *mean*. Gaussian kernel yang digunakan pada jarak ke perkiraan saat ini, yaitu fungsi gaussian pada persamaan 2.68. Maka, nilai *mean* dari kerapatan pada jendela (*window*) yang ditentukan oleh K adalah:

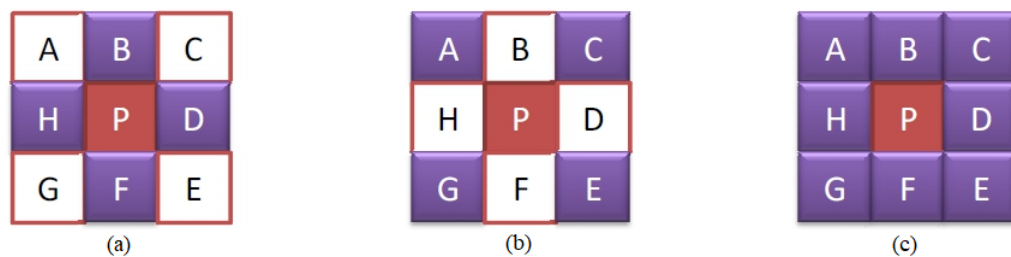
$$m(x) = \frac{\sum_{s \in S} K(s - x)s}{\sum_{s \in S} K(s - x)} \quad (2.69)$$

Persamaan 2.69 merupakan rumus paling sederhana dari *mean shift*. $m(x)$ adalah nilai *mean* dari sampel dan merupakan fungsi X yang menerima data sampel. Terdapat beberapa parameter pada persamaan 2.69, yaitu X , x , S , dan s . X merupakan keseluruhan data, x merupakan data sampel, dan S merupakan sub set dari X . Karena persamaan 2.69 bersifat iteratif, maka akan terjadi perubahan nilai x setiap *looping* atau pengulangan. Hal ini menandakan bahwa *mean shift* dihitung atau jumlah iterasi yang dilakukan sesuai dengan banyak datanya. Nilai s merupakan data tetangga yang terdekat atau terjauh dari nilai *mean*. Jarak dari data tetangga dengan nilai *mean* disebut *range* interval konvolusi atau lebar *window*.

Dalam penerapannya, *mean shift* memanfaatkan ide dari konsep KDE dengan membayangkan apa yang terjadi saat titik mendaki ke puncak terdekat di permukaan KDE. Hal ini dilakukan dengan menggeser setiap titik menanjak secara berulang hingga mencapai puncak. Hasil pengelompokkan akhir KDE akan berbeda, tergantung besarnya *bandwidth* yang digunakan. Sebagai contoh, apabila kita menggunakan

kernel yang sempit dan sangat tinggi dengan *bandwidth* kernel yang kecil, maka hasil yang didapatkan adalah setiap titik akan memiliki puncak dan ditempatkan ke *cluster*-nya sendiri. Sedangkan apabila kita menggunakan kernel yang luas dan sangat pendek dengan *bandwidth* yang besar, hasil yang didapatkan adalah setiap titik akan mendaki satu puncak dan menghasilkan satu *cluster*. Kernel di antara kedua titik tersebut akan menghasilkan pengelompokan yang lebih baik (Nedrich, 2015).

Pada pengolahan citra, kernel adalah matriks angka berukuran kecil yang digunakan dalam konvolusi gambar. Kernel dengan ukuran dan isi pola angka yang berbeda akan memberikan hasil yang berbeda di bawah suatu konvolusi (Fisher et al., 2003). Kernel menggunakan konsep tetangga (*neighbouring pixels*), di mana matriks kernel dibuat dengan asumsi bahwa nilai sebuah piksel bisa dipengaruhi oleh piksel-piksel tetangga. Piksel tetangga sendiri adalah sejumlah piksel yang bersebelahan langsung (*adjacent*) dengan sebuah piksel pusat (Mahastama, 2013). Konsep inilah yang digunakan dalam mengukur jarak tetangga pada *mean shift*.



Gambar 2.6: Jenis piksel tetangga: (a) 4-tetangga +, (b) 4-tetangga x, (c) 8-tetangga

Pada kernel, terdapat operasi yang digunakan untuk menghitung jarak tetangga, yang disebut operasi konvolusi. Dalam pengolahan citra, konvolusi merupakan teknik untuk menghaluskan atau memperjelas suatu citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya tanpa merubah ukuran dari citra tersebut.

Pendefinisian dari dua fungsi konvolusi, yaitu $f(x)$ dan $g(x)$ adalah sebagai

berikut (Gazali et al., 2012):

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha \quad (2.70)$$

Tanda $*$ menyatakan operator konvolusi dan variabel α merupakan peubah bantu (*dummy* variabel). Dalam pengolahan citra, karena koordinat piksel berbentuk diskrit, maka operasi konvolusi dilakukan secara diskrit. Bentuk diskrit dari operasi konvolusi didefinisikan sebagai berikut (Gazali et al., 2012):

$$h(x) = f(x) * g(x) = \sum_{-\infty}^{\infty} f(\alpha)g(x - \alpha) \quad (2.71)$$

Citra merupakan sebuah sinyal dwimatra, yaitu sebuah sinyal yang memiliki fungsi dengan dua peubah. Operasi konvolusi untuk fungsi dwimatra didefinisikan sebagai berikut (Gazali et al., 2012):

Untuk fungsi integral:

$$h(x, y) = f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)g(x - a, y - b)dad b \quad (2.72)$$

Untuk fungsi diskrit:

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x - a, y - b) \quad (2.73)$$

Fungsi $g(x)$ dan $g(x, y)$ disebut sebagai kernel konvolusi atau kernel penapis (filter). Pada fungsi diskrit, kernel konvolusi dinyatakan dalam bentuk matriks (umumnya matriks 3×3). Operasi konvolusi dilakukan dengan menggeser konvolusi kernel piksel per piksel sepanjang permukaan citra dan hasilnya akan disimpan dalam matriks yang baru. Apabila semua piksel citra input sudah terkena perhitungan

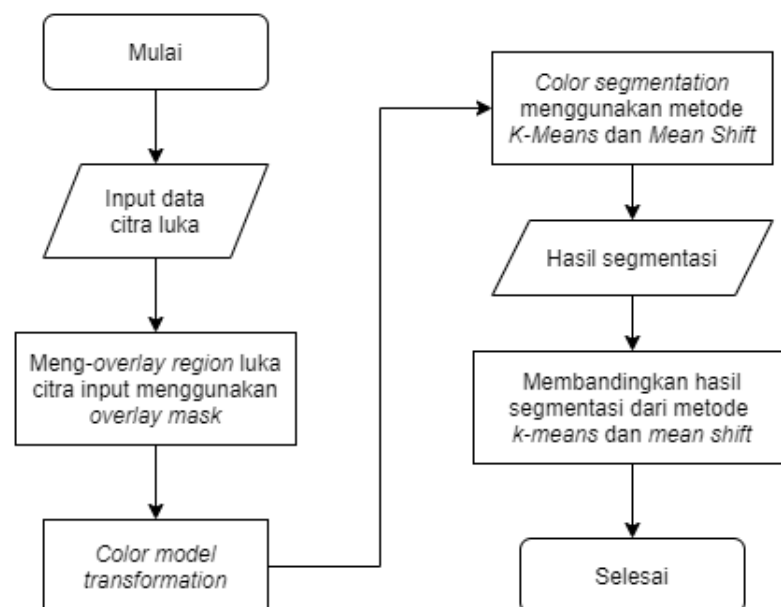
konvolusi, maka operasi konvolusi akan dihentikan. Jika masih terdapat piksel yang dibutuhkan namun bukan merupakan nilai dari piksel input yang ada, dapat dilakukan *padding* dengan menyalin nilai piksel terdekat agar nilai tersebut dapat terpenuhi.

Tidak seperti pada *k-means*, *mean shift* tidak memerlukan penentuan jumlah *cluster* terlebih dahulu. *Mean shift* meng-*cluster* data tanpa parameter buatan, dan hasilnya mengandung beberapa kelas yang ditentukan oleh konten gambar. *Mean shift* juga menyediakan satu parameter, yaitu parameter *bandwidth* kernel, yang dapat dengan mudah disetel dengan tepat untuk aplikasi yang berbeda.

BAB III

METODOLOGI PENELITIAN

Metode yang diusulkan pada penelitian ini dirancang untuk melihat pengaruh ruang warna (*color space*) LAB terhadap segmentasi warna *red*, *yellow*, dan *black* pada citra luka kronis. *Overlay mask* yang didapat dari dari *database* publik <https://github.com/mezas/InjuryDetection> akan dijadikan sebagai filter yang akan di-*overlay* pada citra input sehingga segmentasi hanya berjalan pada region yang dilingkupi oleh citra ini. Proses segmentasi dilakukan menggunakan dua metode, yaitu *k-means* dan *mean shift*. Hasil segmentasi dari kedua metode tersebut akan dibandingkan untuk menentukan metode apa yang dapat mensegmentasi warna luka lebih baik. Berikut diagram dari alur penelitian yang akan dilakukan.



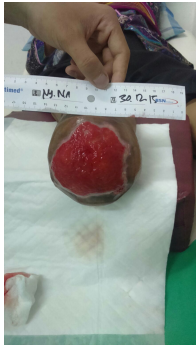









Gambar 3.1: Diagram alur penelitian

3.1 Data











Dataset citra luka yang diujikan didapat dari *database* publik <https://github.com/mekas/InjuryDetection>. *Dataset* yang terdapat pada *database* tersebut berjumlah 185 buah citra luka yang dikategorikan berdasarkan warna luka, yaitu luka campur, luka hitam, luka kuning, dan luka merah. Pada *database* tersebut, terdapat sebuah *ground truth* yang dinamakan *overlay mask*. *Overlay mask* ini merupakan *mask* yang akan dijadikan sebagai filter yang akan di-*overlay* pada citra input sehingga segmentasi hanya berjalan pada *region* yang dilingkupi citra itu.

Untuk mendapatkan hasil segmentasi yang baik, penulis memisahkan data yang layak untuk diuji dan yang tidak. Data yang layak untuk diuji adalah data yang memiliki sedikit *noise*, memiliki *ground truth*, dan ukuran dari *region* lukanya sama dengan ukuran *ground truth* yang ada pada *database*. Dari 185 citra luka, diambil 15 citra luka yang terdiri dari 5 buah citra luka merah, 5 buah citra luka kuning, dan 5 buah citra luka hitam yang di mana semua citra ini memiliki resolusi atau ukuran piksel yang berbeda-beda. Berikut rincian dari data citra luka yang akan digunakan.

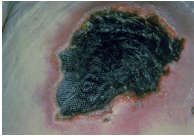







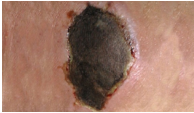

Tabel 3.1: Rincian data luka merah

Citra Luka	Resolusi (piksel)	Overlay Mask
	873x1552	
	309x231	
	600x464	
	300x168	
	260x194	

Tabel 3.2: Rincian data luka kuning

Citra Luka	Resolusi (piksel)	Overlay Mask
	205x246	
	266x190	
	265x190	
	500x333	
	242x208	

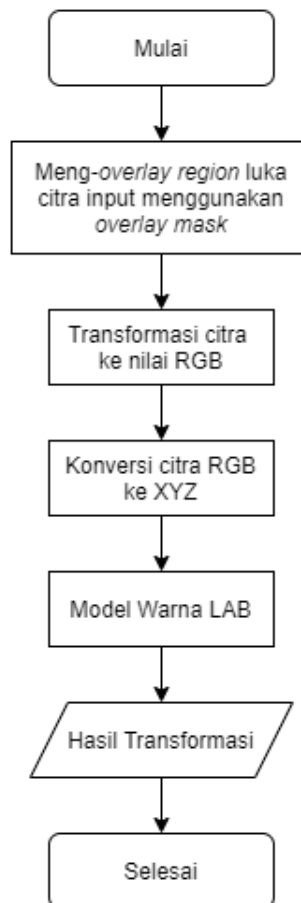
Tabel 3.3: Rincian data luka hitam

Citra Luka	Resolusi (piksel)	Overlay Mask
	620x430	
	640x480	
	234x216	
	350x263	
	650x375	

3.2 Color Model Transformation

Color model transformation merupakan tahapan yang dilakukan untuk mentransformasi warna dari citra yang dipilih ke dalam model warna LAB. Sebelum dilakukan proses transformasi warna, region luka dari citra input terlebih dahulu di-*overlay* dengan menggunakan *overlay mask* sebagai filter. Hasil dari proses *overlay* ini yang akan dijadikan sebagai citra input. Proses transformasi dimulai dengan mentransformasi warna citra ke dalam nilai RGB. Untuk mentransformasi warna ke model warna LAB, warna yang telah ditransformasi menjadi nilai RGB dikonversi terlebih dahulu menjadi model warna XYZ. Proses konversi ini dilakukan dengan menerap-

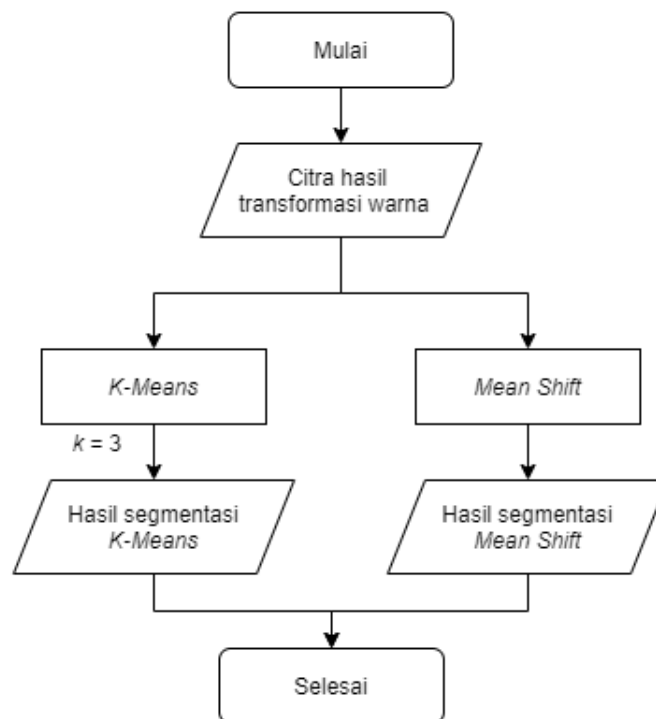
kan persamaan 2.29. Selanjutnya, warna yang telah dikonversi menjadi model warna XYZ akan ditransformasikan ke dalam model warna LAB. Proses ini dilakukan dengan menerapkan persamaan 2.37-2.38.



Gambar 3.2: Proses *color model transformation*

3.3 Color Segmentation

Setelah dilakukan proses *color model transformation*, selanjutnya citra luka akan disegmentasi menjadi warna merah, kuning dan hitam (RYB). Metode yang akan digunakan adalah metode *K-Means* dan *Mean Shift*.



Gambar 3.3: Proses *color segmentation*

Proses segmentasi yang pertama dilakukan adalah proses segmentasi menggunakan metode *k-means* dengan menerapkan persamaan 2.50. Pertama, tentukan terlebih dahulu jumlah *cluster* atau nilai k dan nilai *centroid* atau pusat dari tiap *cluster*. Pada penelitian ini, jumlah *cluster* atau nilai k yang digunakan adalah $k = 3$ dan nilai *centroid*-nya di *hardcode* ke nilai merah, kuning, dan hitam (RYB). Selanjutnya, jarak piksel ke *centroid* dihitung dengan menerapkan persamaan 2.51, dan piksel dikelompokkan berdasarkan jarak terdekat. Setelah itu, nilai *centroid* dihitung kembali sebagai *centroid* baru dengan menerapkan persamaan 2.52, dan piksel dikelompokkan kembali sesuai dengan *centroid* baru tersebut. *Clustering* akan berakhir setelah tidak ada piksel yang berpindah *cluster*.

Setelah dilakukan proses segmentasi menggunakan metode *k-means*, selanjutnya dilakukan proses segmentasi menggunakan metode *mean shift* dengan mene-

rapkan persamaan 2.69. Pertama, tentukan nilai *mean* awal dan *window* dari *dataset*. *Mean* awal ini merupakan titik awal yang akan digunakan sebagai *center* dari *window*, dan *window* merupakan wilayah di sekitar titik *center*. *Window* bisa disebut juga sebagai kernel yang besarnya ditentukan dari nilai *bandwidth*. Pada penelitian ini, kernel yang digunakan adalah gaussian kernel (persamaan 2.68). Selanjutnya, hitung jarak dari setiap data dan *mean* awal. Setelah itu, dicari data mana saja yang termasuk ke dalam *window* dengan membandingkan nilai jarak yang sudah didapat dengan nilai *bandwidth*. Hitung nilai rata-rata dari data yang termasuk ke dalam *window* dan nilai rata-rata ini yang akan dijadikan sebagai nilai *mean (center)* baru. *Window* akan bergeser sesuai dengan nilai *mean*. Proses *clustering* akan berhenti apabila nilai *mean* sudah tidak bergeser dan data sudah *convergence*.

3.4 Evaluasi

Setelah dilakukan proses segmentasi, selanjutnya dilakukan proses evaluasi. Pada tahap ini, citra hasil segmentasi menggunakan metode *k-means* dan *mean shift* dibandingkan untuk menentukan metode apa yang dapat mensegmentasi warna luka lebih baik.

BAB IV

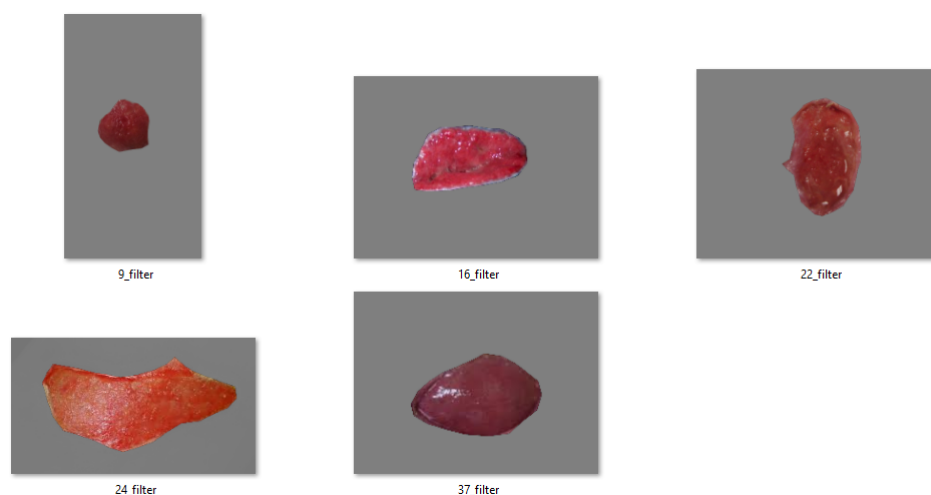
HASIL DAN PEMBAHASAN

4.1 Pengujian

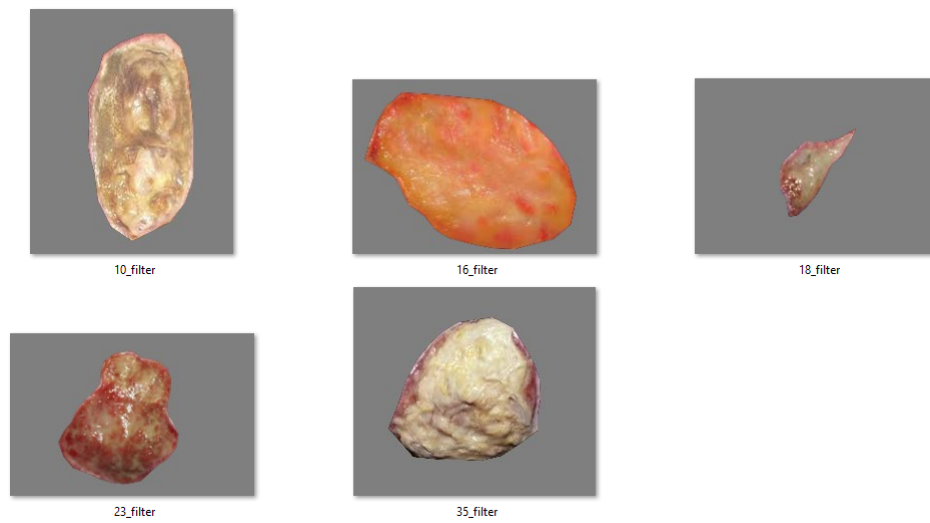
Proses pengujian pada penelitian ini dilakukan terhadap 15 buah citra luka yang terdiri dari 5 buah citra luka merah, 5 buah citra luka kuning, dan 5 buah citra luka hitam. Ke-15 buah citra luka ini akan diujikan menggunakan dua metode, yaitu *k-means* dan *mean shift* yang disusun menggunakan bahasa pemrograman Python.

4.2 Implementasi *Color Model Transformation*

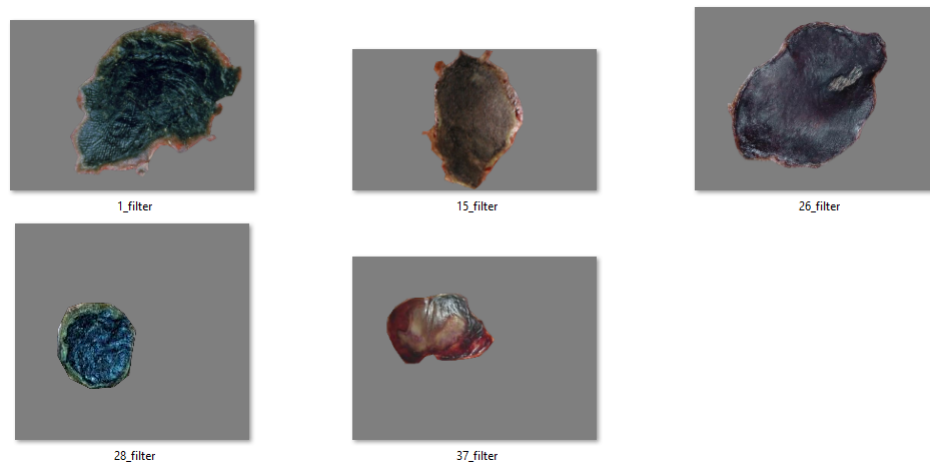
Region dari data citra yang dipilih terlebih dahulu akan di-overlay menggunakan *overlay mask* sebagai filter. *Mask* ini digunakan agar proses segmentasi hanya berjalan pada region luka yang dilingkupi oleh citra ini. Implementasi dari proses *overlay* dilampirkan pada lampiran A. Hasil dari proses *overlay* ditunjukkan pada gambar 4.1, 4.2, dan 4.3



Gambar 4.1: Hasil *overlay* pada citra luka merah

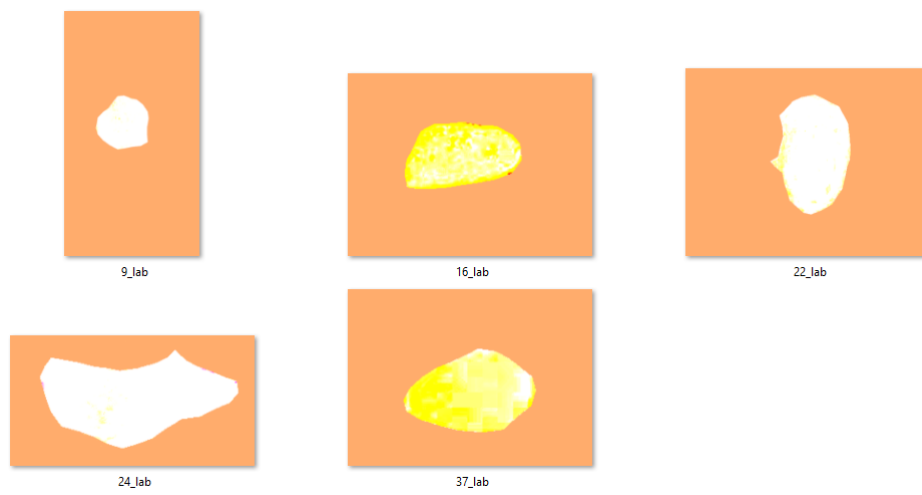


Gambar 4.2: Hasil *overlay* pada citra luka kuning

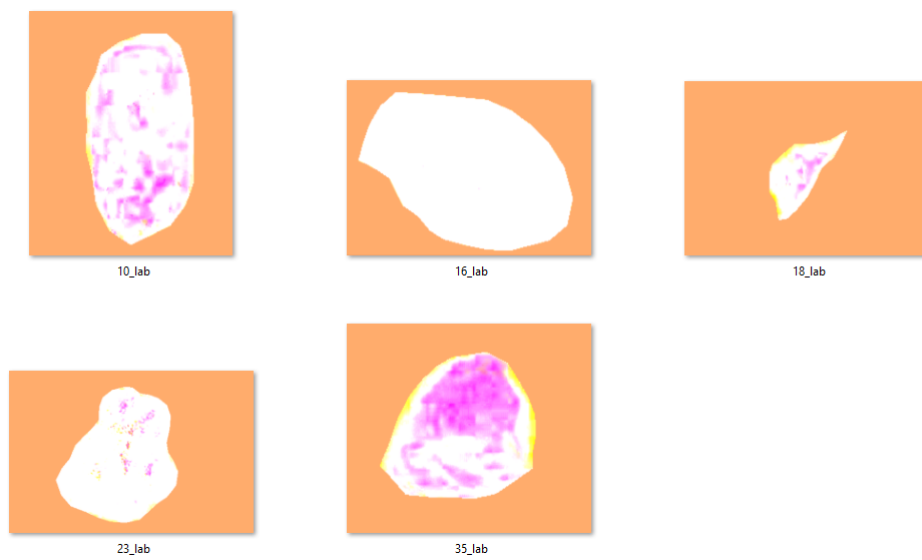


Gambar 4.3: Hasil *overlay* pada citra luka hitam

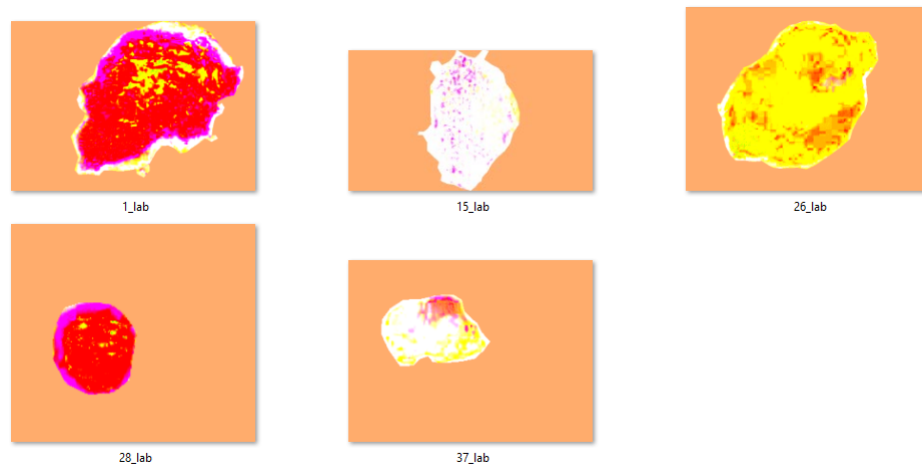
Citra luka yang sudah di-*overlay* menggunakan *overlay mask* kemudian di-transformasi menjadi model warna LAB. Implementasi transformasi ke model warna LAB dilampirkan pada lampiran B. Hasil implementasi dari proses transformasi warna menjadi model warna LAB ditunjukkan pada gambar 4.4, 4.5, dan 4.6



Gambar 4.4: Hasil transformasi warna LAB pada citra luka merah



Gambar 4.5: Hasil transformasi warna LAB pada citra luka kuning



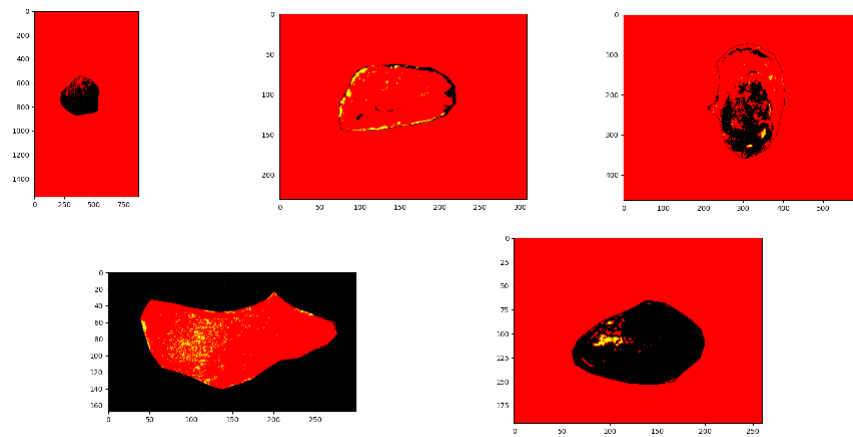
Gambar 4.6: Hasil transformasi warna LAB pada citra luka hitam

4.3 Implementasi *Color Segmentation*

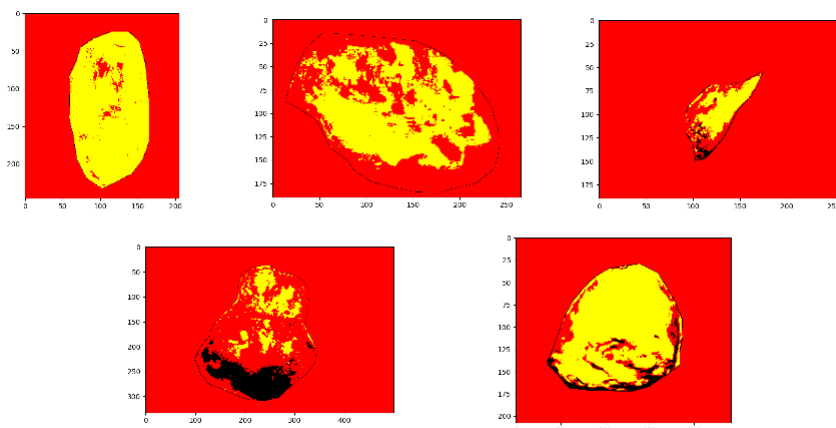
Hasil dari proses *color model transformation* kemudian akan disegmentasi menggunakan dua metode, yaitu metode *k-means* dan *mean shift*.

4.3.1 Implementasi Metode *K-Means*

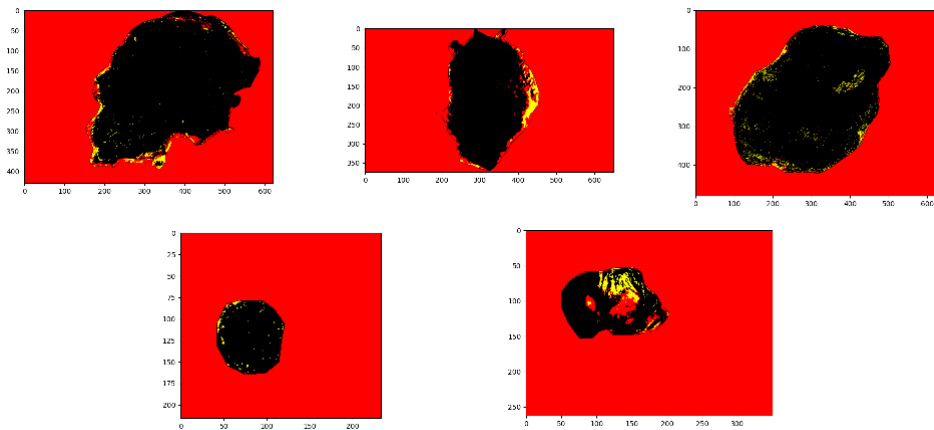
Proses segmentasi dilakukan untuk nilai $k = 3$ dan nilai *centroid*-nya di *hard-code* ke nilai merah, kuning, dan hitam (RYB) dalam rentang 0 sampai 1 atau 0 sampai 255. Implementasi menggunakan metode *k-means* dilampirkan pada lampiran C. Hasil implementasi menggunakan metode *k-means* ditunjukkan pada gambar 4.7, 4.8, dan 4.9



Gambar 4.7: Hasil segmentasi metode *k-means* pada citra luka merah



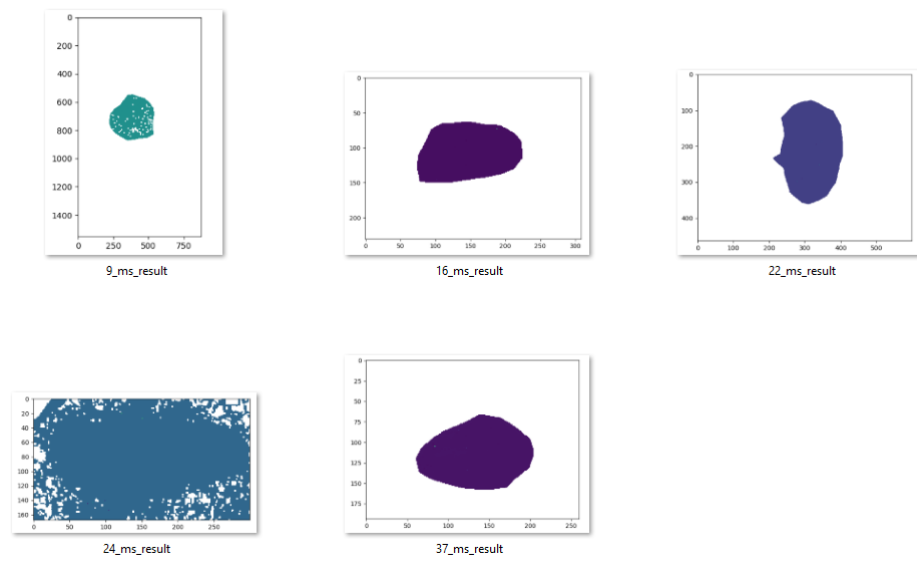
Gambar 4.8: Hasil segmentasi metode *k-means* pada citra luka kuning



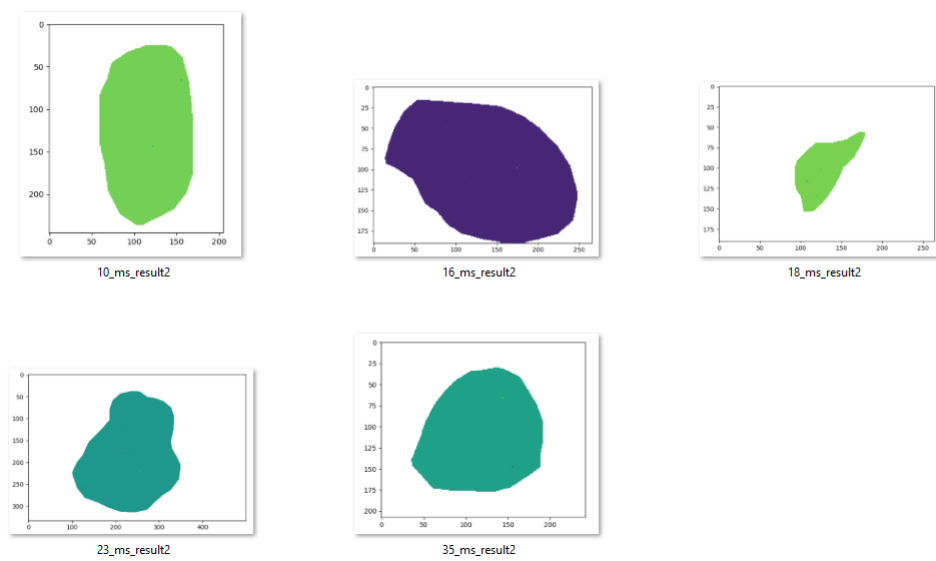
Gambar 4.9: Hasil segmentasi metode *k-means* pada citra luka hitam

4.3.2 Implementasi Metode *Mean Shift*

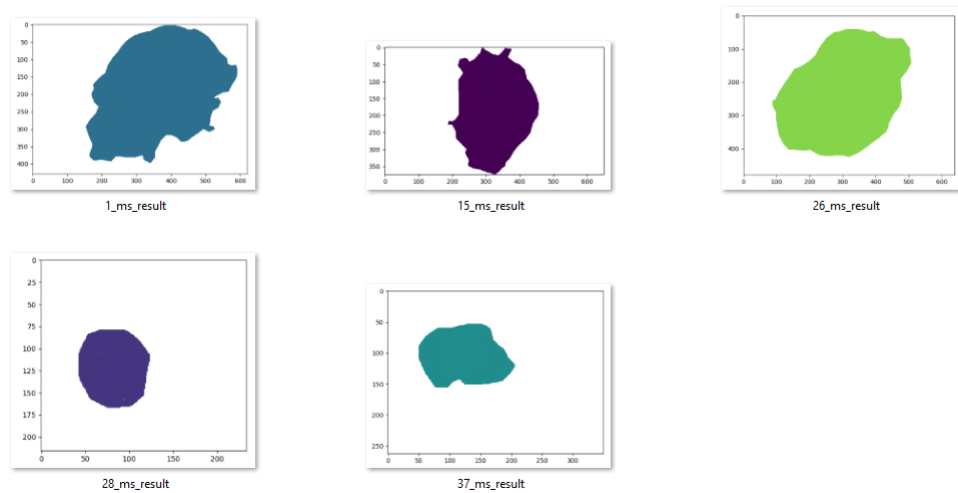
Proses segmentasi pada metode *mean shift* menggunakan dua parameter utama, yaitu w dan σ . w merupakan parameter yang menentukan besarnya *window* dan σ merupakan parameter yang digunakan pada fungsi gaussian kernel. Nilai w dan σ yang digunakan pada penelitian ini adalah $w = 5$ dan $\sigma = 30$. Implementasi menggunakan metode *mean shift* dilampirkan pada lampiran D. Hasil implementasi menggunakan metode *mean shift* ditunjukkan pada gambar 4.10, 4.11, dan 4.12.



Gambar 4.10: Hasil segmentasi metode *mean shift* pada citra luka merah



Gambar 4.11: Hasil segmentasi metode *mean shift* pada citra luka kuning



Gambar 4.12: Hasil segmentasi metode *mean shift* pada citra luka hitam

4.4 Evaluasi

Berdasarkan hasil segmentasi dari metode *k-means* dan *mean shift* pada citra luka, menunjukkan bahwa metode *k-means* dapat mensegmentasi warna merah, kuning, dan hitam dengan cukup baik seperti yang dapat dilihat pada gambar 4.7, 4.8, dan 4.9. Namun, pada metode *mean shift*, masih terdapat beberapa kelemahan dalam melakukan segmentasi, di antaranya citra input harus di-*convert* menjadi citra *grayscale* terlebih dahulu, di mana hal ini menyebabkan hasil segmentasi warna tidak mengarah pada warna merah, kuning, dan hitam seperti yang dapat dilihat pada gambar 4.10, 4.11, dan 4.12. Selain itu, metode *mean shift* juga memiliki kompleksitas yang cukup tinggi yang menyebabkan *running time* dengan metode ini memakan waktu yang cukup lama.

Namun, hasil segmentasi tersebut merupakan hasil segmentasi pada citra RGB dan bukan pada citra LAB. Hasil segmentasi pada citra LAB, baik itu pada metode *k-means* maupun *mean shift*, belum dapat memperlihatkan pengaruh yang signifikan dari penggunaan model warna LAB pada proses segmentasi. Pada metode *k-means*,

hasil segmentasi pada citra LAB tidak menunjukkan adanya perbedaan warna dan citra yang dihasilkan hanya menampilkan satu warna, yaitu warna hitam. Sedangkan pada metode *mean shift*, hasil segmentasi pada citra LAB sama dengan hasil segmentasi menggunakan citra RGB.

Penentuan nilai k dan *centroid* awal yang tepat pada metode *k-means* serta penentuan nilai w dan *sigma* yang tepat pada metode *mean shift* sangat berpengaruh pada hasil segmentasi dari kedua metode ini, di mana nilai k , w , dan *sigma* inilah yang akan menentukan banyaknya jumlah *cluster* yang dihasilkan.

Salah satu kekurangan dari penelitian ini, yaitu penelitian yang dilakukan masih kurang komprehensif, di mana hasil dari penelitian ini belum dapat memperlihatkan pengaruh dari penggunaan model warna LAB pada proses segmentasi. Komprehensif yang dimaksud di sini, yaitu masih belum dapat dilakukannya pengembangan lebih lanjut dari algoritma yang digunakan, namun untuk pengimplementasian dari algoritma yang digunakan sudah selesai dan kebenaran dari algoritmanya sudah di cek oleh Pembimbing melalui *source code* yang di-upload ke platform Github. Beberapa proses pengembangan yang dapat dilakukan agar penelitian yang dilakukan bisa lebih komprehensif di antaranya adalah menentukan nilai *centroid* awal yang tepat pada metode *k-means* agar hasil segmentasinya tidak hanya menampilkan satu warna saja, yaitu warna hitam. Selain itu, dapat juga dilakukan pengembangan lebih lanjut terhadap metode *mean shift*, seperti menentukan nilai w dan *sigma* yang tepat agar hasil segmentasinya bisa mengarah ke warna merah, kuning, dan hitam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian terhadap 15 buah citra luka yang terdiri dari 5 buah citra luka merah, 5 buah citra luka kuning, dan 5 buah citra luka hitam didapatkan kesimpulan sebagai berikut:

1. Implementasi pada penelitian ini menggunakan dua metode, yaitu *k-means* dan *mean shift* yang disusun menggunakan bahasa pemrograman Python.
2. Penentuan nilai *k* dan *centroid* awal pada metode *k-means* serta penentuan nilai *w* dan *sigma* pada metode *mean shift* sangat berpengaruh pada hasil dari segmentasi kedua metode ini.
3. Hasil segmentasi pada citra RGB menunjukkan bahwa metode *k-means* dapat mensegmentasi warna merah, kuning, dan hitam lebih baik dibandingkan dengan metode *mean shift*.
4. Hasil segmentasi pada citra LAB belum dapat menunjukkan pengaruh dari penggunaan model warna LAB pada proses segmentasi, baik pada metode *k-means* maupun *mean shift*.
5. Salah satu kekurangan dari penelitian ini, yaitu penelitian yang dilakukan masih kurang komprehensif, di mana hasil dari penelitian ini belum dapat memperlihatkan pengaruh dari penggunaan model warna LAB pada proses segmentasi.
6. Komprehensif yang dimaksud, yaitu masih belum dapat dilakukannya pengembangan lebih lanjut terhadap algoritma yang digunakan, namun untuk pengim-

plementasian dari algortimanya sudah selesai dan telah di cek kebenarannya oleh Pembimbing.

5.2 Saran

Adapun berikut merupakan beberapa saran untuk pengembangan penelitian selanjutnya:

1. Melakukan pengembangan lebih lanjut terhadap algoritma yang digunakan agar dapat mengetahui pengaruh dari penggunaan model warna LAB pada proses segmentasi.
2. Menggunakan model warna selain model warna LAB dan RGB sebagai citra input pada proses segmentasi.
3. Menggunakan lebih dari dua metode dan objek yang lebih banyak untuk mendapatkan hasil segmentasi yang lebih baik.

DAFTAR PUSTAKA

- Amalia, I. (2015). Identifikasi fase penyembuhan luka berbasis citra. *Jurnal Sains dan Teknologi ISTP*, 4(1).
- Andy (2020). Perbedaan rgb dan cmyk dalam desain grafis.
- Aryani, R., Yusro, M., Suryana, M. E., and Firmansyah, I. (2019). *Buku Panduan : Rancang Bangun Aplikasi Mobile Android Sebagai Alat Deteksi Warna Dasar Luka dalam Membantu Prose Pengkajian Luka Kronis dengan Nekrosis*. Kementrian Kesehatan Republik Indonesia.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Brown, M. S., Ashley, B., and Koh, A. (2018). Wearable technology for chronic wound monitoring: Current dressings, advancements, and future prospects. *Frontier in Bioengineering and Biotechnology*, 6(47):1–21.
- Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799.
- Dewi, D. A. I. C. and Pramita, D. A. K. (2019). Analisis perbandingan metode elbow dan silhouette pada algoritma clustering k-medoids dalam pengelompokan produksi kerajinan bali. *Jurnal Matrix*, pages 102–119.
- Eberly, P. (2021). Introduction to probability theory.
- Ferdinanto, F. G., Dewi, C., and Wiratno, A. T. (2018). Implementasi algoritme k-means sebagai metode segmentasi citra dalam identifikasi penyakit daun jeruk. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11):5375–5383.
- Fisher, R., Perkins, S., Walker, A., and Wolfart, E. (2003). Kernel.

- Fishkin, K. (1990). *A Fast HSL-to-RGB Transform*, pages 448–449. Academic Press.
- Gazali, W., Soeparno, H., and Ohliati, J. (2012). Penerapan metode konvolusi dalam pengolahan citra digital. *Jurnal Mat Stat*, 12(2):103–113.
- Gupta, A. (2017). Real time wound segmentation/management using image processing on handheld devices. *Journal of Computational Methods in Sciences and Engineering*, 17(2):321–329.
- Gustientiedina, Adiya, M. H., and Desnelita, Y. (2019). Penerapan algoritma k-means untuk clustering data obat-obatan pada rsud pekanbaru. *Jurnal Nasional Teknologi dan Sistem Informasi*, 5(1):17–24.
- Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concept and Techniques*. Morgan Kaufmann - Elsevier.
- HSE (2007). *Wound Management Guidelines*. USA.
- Huda, N., Febriyanti, E., and de Laura, D. (2018). Edukasi berbasis nutrisi dan budaya pada penderita luka kronis. *Jurnal Pendidikan Keperawatan Indonesia*, 4(1):1–12.
- Johnson, R. A. and Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall, Inc.
- Kesehatan, D. (2013). Riset kesehatan dasar (riskesdas) 2013. *Jakarta: Badan Penelitian dan Pengembangan Kesehatan Departemen Kesehatan, Republik Indonesia*.
- Khattab, D., Ebied, H. M., Hussein, A. S., and Tolba, M. F. (2014). Color image segmentaton based on different color space models using automatic grabcut. *The Scientific World Journal*, 2014:1–10.

- Kolesnik, M. and Fexa, A. (2004). Segmentation of wounds in the combined color-texture feature space. *Proceedings of SPIE - The International Society of Optical Engineering*, 5370:549–556.
- Kumar, S. (2020). Silhouette method — better than elbow method to find optimal clusters.
- Landén, N. X., Li, D., and Ståhle, M. (2016). Transition from inflammation to proliferation: a critical step during wound healing. *Cellular and Molecular Life Science*, 73(20):3861–3885.
- MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
- Mahastama, A. W. (2013). Pengolahan citra digital.
- Mekkes, J. R. and Westerhof, W. (1995). Image processing in the study of wound healing. *Clinics in Dermatology*, 13(4):401–407.
- Mohanty, A., Soundrapandiyam, R., Mir, Z. M., and Puja, B. (2013). Analysis of color images using cluster based segmentation techniques. *International Journal of Computer Applications*, 79(2):42–47.
- Mukherjee, R., Manohar, D. D., Das, D. K., Achar, A., Mitra, A., and Chakraborty, C. (2014). Automated tissue classification framework for reproducible chronic wound assessment. *BioMed Research International*, 2014:1–9.
- Nedrich, M. (2015). Mean shift clustering.
- Nixon, M. S. and Aguado, A. S. (2020). *Feature Extraction and Image Processing for Computer Vision*, chapter 11. Elsevier Ltd.
- Pamungkas, A. (2017). Pengolahan citra digital.

- Panjaitan, J. H. (2016). Analisis segmentasi citra menggunakan algoritma *k-means*, *mean shift*, dan *normalized cut*. *Bachelor Thesis, Universitas Multimedia Nusantara*.
- Pascale, D. (2002–2003). *A Review of RGB Color Spaces ...from xyY to R'G'B'*. The BabelColor Company.
- Paschotta, D. R. (2020). Color spaces.
- Phil, C. (2021). Introduction to colour models ('spaces').
- Purnama, H., Sriwidodo, and Ratnawulan, S. (2017). Review sistematis: Proses penyembuhan dan perawatan luka. *Jurnal Ilmiah Farmasi Indonesia Fakultas Farmasi Universitas Padjajaran*, 15(2):251–258.
- Rahman, A. T., Wiranto, and Anggrainingsih, R. (2017). Coal trade data clustering using k-means (case study pt. global bangkit utama). *ITS MART: Jurnal Ilmiah Teknologi dan Informasi*, 6(1):24–31.
- Rahmansyah, G., Mahmudy, W. F., and Perdana, R. S. (Maret 2016). Optimasi model segmentasi citra metode *fuzzy divergence* pada citra luka kronis menggunakan algoritma genetika. *Jurnal Teknik Informasi dan Ilmu Komputer*, 3(1):66–74.
- Ramya, R. and Jenitta, A. (2017). Foot injury detection using k-means clustering and mean shift segmentation algorithm. *International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)*, 3(24):323–329.
- Shmmla, F. A. and Ashour, W. (2013). Color based image segmentation using different version of k-means in two spaces. *Global Advanced Research Journal of Engineering, Technology and Innovation*, 1(9):30–41.
- Squires, G. L. (2001). *Practical Physics*. Cambridge University Press.

- Swedia, E. R. and Cahyanti, M. (2010). *Algoritma Transformasi Ruang Warna*.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- Teresa (2019). Pewarnaan citra *Grayscale* ke dalam citra berwarna dengan menggunakan *Pseudocoloring* berbasis palet warna. *Journal of Physics: Conference series*.
- Van Poucke, S., Haeghen, Y. V., Vissers, K., Meert, T., and Jorens, P. (2010). Automatic colorimetric calibration of human wounds. *BMC Medical Imaging*, 10(7):1–11.
- Wang, L., Pedersen, P. C., Agu, E., Strong, D. M., and Tulu, B. (2017). Area determination of diabetic foot ulcer images using a cascaded two-stage svm-based classification. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 64(9):2098–2109.
- Wang, L., Pedersen, P. C., Strong, D. M., Tulu, B., Agu, E., and Ronald, I. (2015). Smartphone-based wound assessment system for patients with diabetes. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 62(2):477–488.
- Wannous, H., Lucas, Y., and Treuillet, S. (2011). Enhanced assessment of the wound-healing process by accurate multiview tissue classification. *IEEE Transactions on Medical Imaging*, 30(2):315–326.
- Wolfcrow (2021). What is the difference between cie lab, cie rgb, cie xyy and cie xyz?
- Yadav, M. K., Manohar, D. D., Mukherjee, G., and Chakraborty, C. (2013). Segmentation of chronic wound areas by clustering techniques using selected color space. *Journal of Medical Imaging and Health Informatics*, 3(1):22–29.

Zakaria, A. (2020). Mengenal color gamut srgb, adobe rgb, ntsc, dci-p3 dalam desain grafis.

Zheng, X. and Liu, N. (2012). Color recognition of clothes based on k-means and mean shift. *2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment (ICADE)*, pages 49–53.

LAMPIRAN A

Sampel Kode Proses *Overlay* Menggunakan *Overlay Mask*

```
image = imread('../skripsi-april/luka_kuning/35.jpg',  
as_gray=False)  
height, width, channel = image.shape  
  
mask = imread('../skripsi-april/luka_kuning/35_region.png',  
as_gray=True)  
mask_resize = resize(mask, (height, width, 3))  
  
hasil1 = compare_images(mask_resize, image, method='diff')  
hasil2 = compare_images(hasil1, image, method='blend')
```

LAMPIRAN B

Sampel Kode Proses *Color Model Transformation*

```
for i in range(height):
    for j in range(width):
        for k in range(channel):
            R = hasil2.item(i, j, 0)
            G = hasil2.item(i, j, 1)
            B = hasil2.item(i, j, 2)

            X = (R*0.49000 + G*0.31000 + B*0.20000)/0.17697
            Y = (R*0.17697 + G*0.81240 + B*0.01063)/0.17697
            Z = (R*0.00000 + G*0.01000 + B*0.99000)/0.17697

            x = X/(X+Y+Z)
            y = Y/(X+Y+Z)
            z = Z/(X+Y+Z)

            image_xyz.itemset((i, j, 0), float(x))
            image_xyz.itemset((i, j, 1), float(y))
            image_xyz.itemset((i, j, 2), float(z))

x_n = 95.0489
y_n = 100
z_n = 108.8840
```

```

x_ = x/x_n
y_ = y/y_n
z_ = z/z_n

def t(s):
    if s > (6/29)**3 :
        return s**(1/3)
    else :
        return (1/3)*((29/6)**2)*s + 16/116

fx = t(x_)
fy = t(y_)
fz = t(z_)

L = (116*fy) - 16
a = 500*(fx - fy)
b = 200*(fy - fz)

image_lab1.itemset((i, j, 0), float(L))
image_lab1.itemset((i, j, 1), float(a))
image_lab1.itemset((i, j, 2), float(b))}

```

LAMPIRAN C

Sampel Kode Segmentasi Menggunakan Metode *K-Means*

```
X = hasil2
Y = deepcopy(X)

class Kmeans:
    def __init__(self, K, max_iter=500):
        self.K = K
        self.max_iter = max_iter

    def compute_centroids(self, X, labels):
        centroids = np.zeros((self.K, X.shape[2]))
        with warnings.catch_warnings():
            warnings.simplefilter("ignore", category=RuntimeWarning)
        for k in range(self.K):
            centroids[k, :] = np.mean(X[labels == k, :], axis=0)
        #print(centroids)
        return centroids

    def compute_distance(self, X, centroids):
        D = np.zeros((X.shape[0], X.shape[1], self.K))
        for k in range(self.K):
            dist = np.linalg.norm(X - centroids[k, :], axis=2)
            D[:, :, k] = np.square(dist)
        #print(D.shape)
```



```

#print(D)

return D

# untuk mencari jarak terdekat
def find_closest_cluster(self, D):
    a = np.argmin(D, axis=2)
    for i in range(X.shape[0]):
        for j in range(X.shape[1]):
            X[i,j] = self.centroids[a[i,j]]
    return a

def compute_sse(self, X, labels, centroids):
    distance = np.zeros((X.shape[0], X.shape[1], self.K))
    for k in range(self.K):
        distance[labels == k, :] = \
            np.linalg.norm(X[labels == k, :] - centroids[k, :], axis=0)
    return np.sum(np.square(distance))

def fit(self, X):
    #self.centroids = np.array([[255, 0, 0],
    #                           [255, 255, 0],
    #                           [0, 0, 0]])
    self.centroids = np.array([[1, 0, 0],
    #                           [1, 1, 0],
    #                           [0, 0, 0]])
    #print(self.centroids.shape)

```

```

for i in range(self.max_iter):
    old_centroids = self.centroids
    distance = self.compute_distance(X, old_centroids)
    self.labels = self.find_closest_cluster(distance)
    self.centroids = self.compute_centroids(X, self.labels)

    if np.all(old_centroids == self.centroids):
        break

self.error = self.compute_sse(X, self.labels, self.centroids)

def predict(self, X):
    old_centroids = self.centroids
    distance = self.compute_distance(X, old_centroids)
    return self.find_closest_cluster(distance)

kmeans = Kmeans(K=3, max_iter=500)
kmeans.fit(X)
centroids = kmeans.centroids

fig = plt.figure()
ax1 = fig.add_subplot(121), imshow(Y)
ax2 = fig.add_subplot(122), imshow(X)
plt.show()

```

LAMPIRAN D

Sampel Kode Segmentasi Menggunakan Metode *Mean Shift*

```
def gaussian_filter_2d(w, sigma):  
    # we must be odd, or we enforce it  
    w = w + (w % 2 == 0)  
  
    # create an array of size w x w  
    F = np.zeros([w, w])  
    # window w must be odd if it isn't add 1 to it  
    w = w + ((w % 2) == 0)  
    mid = w // 2  
    X = np.arange(w) - mid  
    denom = 1 / (2 * np.pi * sigma ** 2)  
    denom_s = 1 / (2 * sigma ** 2)  
    for i in X:  
        for j in X:  
            power_val = (i ** 2 + j ** 2) * denom_s  
            F[i + mid, j + mid] = np.exp(-power_val) * denom  
    return F  
  
def convolve_2d_global(X, K):  
    # pad X  
    (m, n) = X.shape  
    w1 = K.shape[0]
```

```

w = wl // 2

Xp = np.pad(X, w, 'symmetric')
Z = np.zeros(X.shape)
for i in range(m):
    for j in range(n):
        W = Xp[i:i + wl, j:j + wl]
        Z[i, j] = np.sum(np.dot(W, K))
return Z

def gaussian_kernel(X, F):
    Z = np.sum(np.dot(X, F))
    return Z

def get_neighbors(X, w, i, j):
    X = np.pad(X, w, 'symmetric')
    return X[i:i + w:, j:j + w]

def meanshift(X, w, sigma):
    # store original data
    Xc = np.array(X)
    F = gaussian_filter_2d(w, sigma)

    # get all neighbors

```

```

(m, n) = X.shape
for i in range(m):
    for j in range(n):
        with warnings.catch_warnings():
            warnings.simplefilter('ignore')
            Xn = get_neighbors(X, w, i, j)
            upper_par = (Xn - X[i, j]) * Xn
            lower_par = (Xn - X[i, j])

            new_x =
            gaussian_kernel(upper_par, F) / gaussian_kernel(lower_par, F)
            Xc[i, j] = new_x
        X = Xc
    return X

imgrey = 0.3*hasil2[:, :, 0] +
0.59*hasil2[:, :, 1] +
0.11*hasil2[:, :, 2]

w = 5
sigma = 30
F = gaussian_filter_2d(w, sigma)
Xk = meanshift(imgrey, w, sigma)

plt.imshow(imgrey)
plt.imshow(Xk)
plt.show()

```

DAFTAR RIWAYAT HIDUP



APRILIA KHAIRUNNISA. Lahir di Jakarta, 14 April 1999. Anak pertama dari pasangan Alm. Bapak Nursyahbuddin dan Ibu Susilawati. Saat ini beralamatkan di Jl. Sangingi No. 26 A, Cideng, Gambir, Jakarta Pusat.

No. Ponsel : 085717763132

Email : aprilnisa1491@gmail.com

Riwayat Pendidikan : Penulis mengawali pendidikan di SDN Cideng 07 Pagi pada tahun 2005 - 2010. Setelah itu, penulis melanjutkan studi ke SMPN 72 Jakarta hingga tahun 2013. Kemudian melanjutkan ke SMAN 4 Jakarta pada tahun 2013-2016. Di tahun 2016, penulis melanjutkan ke Universitas Negeri Jakarta (UNJ), Program Studi Ilmu Komputer, melalui jalur SNMPTN. Pada tahun 2021, penulis telah memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta.

Riwayat Organisasi : Selama di bangku perkuliahan, penulis berpartisipasi dalam organisasi keilmiah di Program Studi Ilmu Komputer yaitu DEFAULT, di mana penulis tergabung sebagai anggota divisi *web* dan menjabat sebagai Bendahara I. Penulis juga kerap mengikuti kepanitiaan kegiatan yang diadakan oleh lembaga DEFAULT.