

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Kamera *Smartphone*

a. Kamera

Kamera merupakan alat utama yang sering digunakan dalam fotografi. Fungsinya adalah untuk membentuk dan merekam bayangan potret pada lembaran film. (Riyantomo, 2015)

b. *Smartphone*

Smartphone merupakan alat komunikasi yang memiliki kemampuan lebih dari alat komunikasi biasa. Selain digunakan untuk berkomunikasi, *smartphone* juga dilengkapi dengan fitur tambahan seperti aplikasi pendukung. Semakin majunya teknologi dan informasi, semakin banyak pula penggunaan *smartphone* sebagai alat bantu terbaru yang membantu mempermudah pekerjaan manusia dengan efisiensi waktu yang lebih cepat dan mudah. (Solikin, 2018)

2. Kulit Wajah

Kulit merupakan bagian dari tubuh manusia yang menutupi seluruh bagian, termasuk wajah. Penting untuk merawat kulit wajah karena selain menjadi simbol identitas, memiliki kulit wajah yang sehat adalah keinginan semua orang. Ciri-ciri kulit yang sehat meliputi warna kulit yang merata dan bersinar, tekstur kulit yang lembut dan kencang, serta bebas dari jerawat. (Isfianati, 2018)

3. *Android*

Awal diciptakannya *Android Inc*, *Android* adalah sebuah sistem operasi berbasis Linux yang dirancang khusus untuk perangkat mobile. Kemudian, pada tahun 2005, Google membeli *Android Inc* dan mengambil alih pengembangan sistem operasi tersebut. Pada tahun 2007, dibentuklah *Open Handset Alliance (OHA)* sebagai sebuah konsorsium yang terdiri dari beberapa perusahaan seperti *Texas Instruments*, *Broadcom Corporation*, *Google*, *HTC*, *Intel*, *LG*, *Marvell Technology Group*, *Motorola*, *Nvidia*, *Qualcomm*, *Samsung Electronics*, *Sprint Nextel*, dan *T-*

Mobile. Tujuan dibentuknya OHA adalah untuk mengembangkan standar terbuka yang berlaku untuk perangkat mobile dan terus mengembangkan *Android*. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 anggota baru bergabung dengan proyek *Android*, di antaranya *PacketVideo*, *ARM Holdings*, *Atheros Communications*, *Asustek Computer Inc*, *Garmin Ltd Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*. (Maiyana, 2018)

4. *Android Studio*

Android Studio adalah sebuah *IDE (Integrated Development Environment)* resmi untuk mengembangkan aplikasi *Android* yang tersedia secara gratis dan *open source*. Pada konferensi Google I/O tahun 2013 pada 16 Mei 2013, Google mengumumkan peluncuran *Android Studio* yang kemudian menggantikan *Eclipse* sebagai *IDE* resmi untuk pengembangan aplikasi *Android*. *Android Studio* dibangun berdasarkan *IntelliJ IDEA* dan dilengkapi dengan *plugin ADT (Android Development Tools)* yang mirip dengan *Eclipse*. *Android Studio* menawarkan berbagai fitur, di antaranya:

- a. Proyek berbasis pada *Gradle Build*.
- b. *Refactory* dan pembenahan bug yang cepat.
- c. Tools baru yang bernama "*Lint*" diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard And App-signing* untuk keamanan.
- e. Memiliki *GUI* aplikasi *android* lebih mudah.

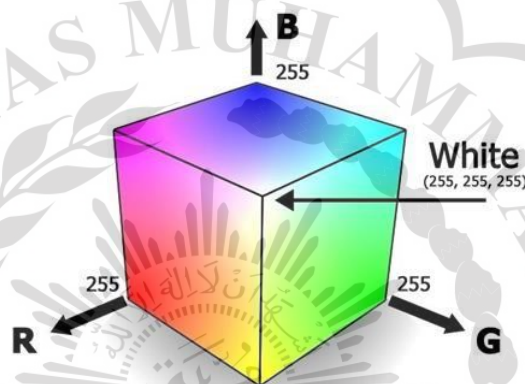
Setiap aplikasi yang dikembangkan didukung oleh *Google Cloud Platform*. (Andi, 2015).

5. *Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi, termasuk perangkat seluler. Bahasa pemrograman ini awalnya dikembangkan oleh James Gosling saat bekerja di *Sun Microsystems*. *Java* sendiri merupakan pengembangan dari bahasa pemrograman C++ dan sekarang menjadi salah satu bahasa pemrograman yang paling luas digunakan dan populer, baik dalam pengembangan aplikasi perangkat lunak maupun berbasis web. (Gosling, J., Joy, B., Steele, G., & Bracha, G., 2019)

6. Ruang RGB

Dalam bidang monitor komputer atau televisi, terdapat sebuah model warna yang disebut dengan RGB (Red, Green, Blue) yang terdiri dari warna dasar yaitu merah, hijau, dan biru. Dalam model ini, warna yang berbeda dihasilkan melalui percampuran tiga warna dasar ini dan setiap warna memiliki nilai intensitas 8 bit untuk masing-masing warna primer. Jika ketiga warna dicampur secara seimbang, maka akan menghasilkan warna abu-abu, sementara jika ketiga warna disaturasikan sepenuhnya, maka akan menghasilkan warna putih. (Rulaningtyas et al., 2015). Gambar warna RGB seperti pada Gambar 1,



Gambar 1. Citra warna RGB.

7. Grayscale

Citra *grayscale* mengacu pada gambar yang hanya menampilkan variasi warna abu-abu tanpa menggunakan warna lainnya. (Sen, 2018). Citra *grayscale* memiliki variasi warna abu-abu yang dihasilkan dari intensitas yang sama dari ketiga komponen warna utama yaitu merah, hijau, dan biru. Oleh karena itu, nilai intensitas pada setiap piksel pada citra grayscale hanya perlu diwakili oleh satu nilai tunggal, berbeda dengan citra berwarna yang memerlukan tiga nilai intensitas untuk setiap piksel.

Konversi dari citra RGB ke *grayscale* pada persamaan (1) :

$$I_y = 0,333F_r + 0,5F_g + 0,1666F_b \quad (1)$$

F_r = intensitas R

F_g = intensitas G

F_b = intensitas B

I_y = intensitas abu-abu yang setara gambar level RGB

8. *Fotografi*

Fotografi memiliki asal-usul dari bahasa Yunani, yaitu terdiri dari dua kata yaitu "*photo*" yang berarti cahaya dan "*graphy*" yang berarti menulis atau menggambar. Oleh karena itu, makna harfiah dari *fotografi* adalah membuat gambar atau tulisan dengan menggunakan cahaya. Konsep *fotografi* menggabungkan unsur alamiah (cahaya) dengan kegiatan manusia (menulis atau menggambar/melukis) untuk menciptakan identitasnya. *Fotografi* pada dasarnya melibatkan proses merekam dan memanipulasi cahaya untuk menghasilkan gambar yang diinginkan. Kegiatan *fotografi* dapat dikategorikan sebagai teknik dan seni. (Tanjung, 2019).

9. *Citra*

Menurut (Fahmi, 2017), untuk memproses citra menggunakan komputer, dibutuhkan citra digital sebagai input. Citra digital adalah gambar dua dimensi yang terdiri dari himpunan nilai digital diskrit yang disebut dengan *Pixel* (*picture element*), yang dapat ditampilkan pada layar monitor komputer.

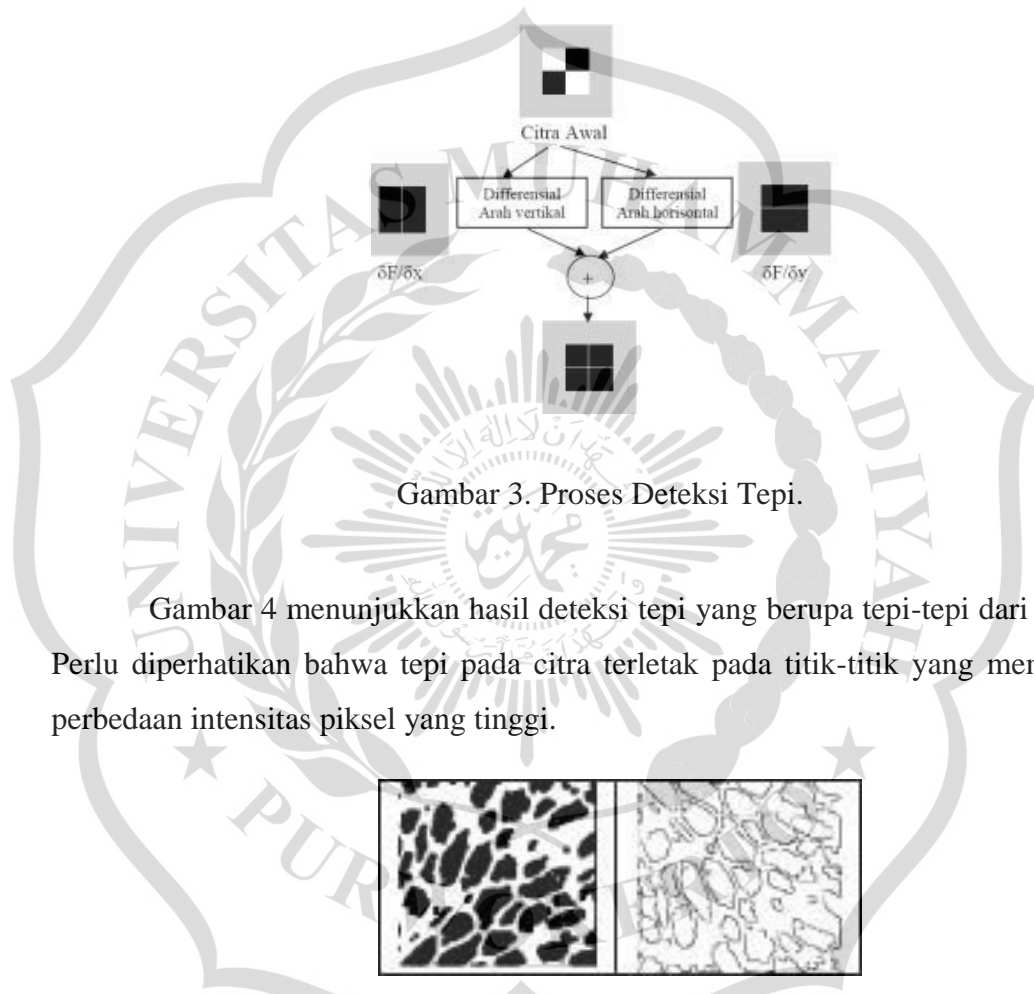
Citra digital awalnya bersifat kontinu namun diubah menjadi bentuk diskrit, termasuk koordinat ruang dan intensitas cahayanya. Citra digital dapat direpresentasikan dalam bentuk matriks dua dimensi dengan $f(x,y)$ yang merepresentasikan derajat intensitas dari tiap *pixel* pada koordinat (x,y) . Ilustrasi tersebut dapat dilihat pada Gambar 2.



Gambar 2. *Citra Digital* dalam Bentuk *Matriks* Dua Dimensi.

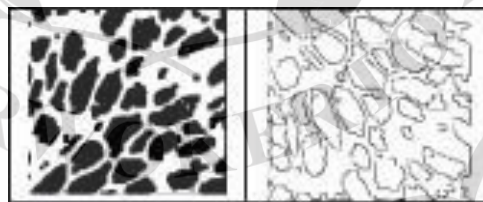
10. Deteksi Tepi

Deteksi tepi (*Edge Detection*) adalah proses yang menghasilkan tepi-tepi dari objek pada citra. Suatu titik pada citra (x,y) dianggap sebagai tepi (*edge*) apabila terdapat perbedaan yang signifikan dengan nilai piksel tetangganya. (Laxmi et al., 2017). Gambar 3 menunjukkan langkah-langkah yang dilakukan untuk mendapatkan tepi pada citra yang diberikan.



Gambar 3. Proses Deteksi Tepi.

Gambar 4 menunjukkan hasil deteksi tepi yang berupa tepi-tepi dari citra. Perlu diperhatikan bahwa tepi pada citra terletak pada titik-titik yang memiliki perbedaan intensitas piksel yang tinggi.



Gambar 4. Hasil Deteksi Tepi.

11. Metode *Canny*

Menurut (Panjaitan, 2020) Algoritma *Canny* banyak digunakan dalam penelitian deteksi tepi karena dianggap sebagai algoritma deteksi tepi yang paling optimal. Langkah pertama dalam algoritma *canny* adalah menerapkan filter Gaussian pada citra untuk menghilangkan derau, diikuti dengan melakukan deteksi tepi pada citra menggunakan salah satu algoritma deteksi tepi yang tersedia, seperti *Sobel* atau *Prewitt*. Ilustrasi langkah pertama dalam algoritma

Canny, yaitu penghilangan derau pada citra menggunakan filter Gaussian, ditunjukkan pada Gambar 5.

$$\frac{1}{115}$$

| | | | | |
|---|----|----|----|---|
| 2 | 4 | 5 | 4 | 2 |
| 4 | 9 | 12 | 9 | 4 |
| 5 | 12 | 15 | 12 | 5 |
| 4 | 9 | 12 | 9 | 4 |
| 2 | 4 | 5 | 4 | 2 |

Gambar 5. Tapis Gaussian

Langkah berikutnya adalah melakukan deteksi tepi menggunakan salah satu operator deteksi tepi (*Sobel* atau *Prewitt*) dengan menghitung nilai gradien secara horizontal (GX) dan vertikal (GY) menggunakan rumus :

$$|G| = |G| \begin{cases} G_x = \text{Pencarian secara horizontal} \\ G_y = \text{Pencarian secara vertikal} \end{cases}$$

Operator Sobel terdiri dari sepasang kernel berukuran 3x3 yang digunakan untuk mendeteksi tepi vertikal dan horizontal seperti pada Gambar 6.

| | | | | | |
|----------------|----|----|----------------|---|----|
| +1 | +2 | +1 | -1 | 0 | +1 |
| 0 | 0 | 0 | -2 | 0 | +2 |
| -1 | -2 | -1 | -1 | 0 | +1 |
| G _x | | | G _y | | |

Gambar 6. Deteksi Tepi Vertikal dan Horizontal I

Algoritma deteksi tepi *Prewitt* memiliki kesamaan dengan *Sobel*, namun menggunakan Operator *Prewitt* yang memiliki nilai yang sedikit berbeda dengan Operator *Sobel* seperti pada gambar 7.

| | | | | | |
|----------------|----|----|----------------|---|----|
| +1 | +1 | +1 | -1 | 0 | +1 |
| 0 | 0 | 0 | -1 | 0 | +1 |
| -1 | -1 | -1 | -1 | 0 | +1 |
| G _x | | | G _y | | |

Gambar 7. Deteksi Tepi Vertikal dan Horisontal II

Persamaan berikut digunakan untuk menghitung jarak gradien :

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Persamaan gradient Nomor 1

Persamaan berikut digunakan untuk menghitung arah garis tepi yang dihasilkan :

$$\theta = \text{artan} \left(\frac{Gy}{Gx} \right)$$

Persamaan gradient Nomor 2

Langkah selanjutnya adalah membagi garis-garis yang ada menjadi 4 bagian terpisah dengan warna yang berbeda-beda dengan sudut tertentu yang diatur sebagai berikut :

1. Derajat 0 - 22,5 dan 157,5 - 180 berwarna kuning.
2. Derajat 22,5 - 67,5 berwarna Hijau.
3. Derajat 67,5 - 157,5 berwarna Merah.

Setelah itu, garis tepi dikurangi agar menjadi tipis menggunakan metode *non maximum suppression*. Tahap akhirnya adalah melakukan binerisasi dengan menggunakan nilai threshold rendah dan tinggi yang telah ditentukan.

12. OpenCV

OpenCV menurut (Permana & Destriana, 2018), yang merupakan singkatan dari *Open Source Computer Vision*, adalah sebuah perpustakaan perangkat lunak gratis yang digunakan untuk membuat aplikasi pengolahan citra dan visi komputer. Perpustakaan ini menyediakan berbagai macam fitur dan algoritma untuk melakukan operasi seperti segmentasi citra, deteksi objek, pelacakan objek, dan pengenalan wajah. Selain itu, OpenCV mendukung berbagai platform dan bahasa pemrograman, termasuk C++, Python, Java, dan MATLAB, yang memungkinkan para pengembang untuk mengintegrasikan teknologi pengolahan citra ke dalam aplikasi mereka dengan mudah. Dalam penelitian ini, OpenCV digunakan untuk mengembangkan sistem deteksi tepi berbasis kamera yang bertujuan untuk melihat tepi dari sebuah permukaan kulit.

13. Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpretatif, artinya kode program Python dapat dieksekusi langsung oleh interpreter Python tanpa memerlukan proses kompilasi terlebih dahulu. Python dikembangkan oleh Guido van Rossum pada awal tahun 1990 dan dirilis pertama kali pada tahun 1991.

Python dirancang dengan fokus pada kemudahan pembacaan dan penulisan kode, Python digunakan dalam berbagai bidang seperti pengembangan web, kecerdasan buatan (artificial intelligence), pengolahan data (data science), analisis sistem, dan sebagainya. (J. Cutler & M. Dickenson, 2020).

14. PSNR (Peak Signal-to-Noise Ratio)

PSNR (Peak Signal-to-Noise Ratio) adalah sebuah metrik yang digunakan untuk mengukur kualitas citra atau video dengan membandingkan citra atau video asli dengan citra atau video hasil kompresi atau proses pengolahan lainnya. PSNR menghitung rasio antara puncak kekuatan sinyal dengan kekuatan noise pada citra atau video yang dibandingkan. Semakin tinggi nilai PSNR, semakin baik kualitas citra atau video yang dibandingkan. PSNR dihitung dalam satuan desibel (dB). (A. Hore & D. Ziou, 2020).

Namun, PSNR memiliki kelemahan yaitu tidak selalu konsisten dengan kualitas visual manusia dan tidak dapat mengidentifikasi jenis distorsi tertentu seperti distorsi blok pada hasil kompresi. Oleh karena itu, perlu dikombinasikan dengan metrik lain seperti Structural Similarity Index (SSIM) untuk menilai kualitas citra atau video secara lebih lengkap dan akurat.

15. SSIM (Structural Similarity Index)

SSIM (Structural Similarity Index) adalah metrik yang digunakan untuk mengukur kesamaan struktural antara dua citra atau video. SSIM memperhitungkan kesamaan struktural dalam citra atau video seperti kecerahan, kontras, dan tekstur menurut (A. Hore & D. Ziou, 2020). Nilai SSIM berkisar dari 0 hingga 1, dengan 1 menunjukkan kesamaan sempurna antara dua citra atau video. SSIM sering digunakan untuk menilai kualitas citra atau video dalam situasi di mana kualitas visual manusia penting, karena SSIM lebih konsisten dengan persepsi visual manusia daripada metrik lain seperti PSNR.

B. Hasil Penelitian Terdahulu

Dalam penelitian sebelumnya yang dilakukan oleh (Ahmed, 2018) telah dilakukan sebuah studi perbandingan antara tiga operator deteksi tepi, yaitu *Sobel*, *Prewitt*, dan *Canny* dalam *Image Processing*. Lima gambar sampel, termasuk gambar patung *Liberty*, gambar menara *Eiffel*, gambar mobil, gambar rumah Jepang, dan gambar katedral *Saint Basil*, digunakan dalam penelitian ini. Hasil penelitian menunjukkan bahwa metode *Prewitt* memiliki kinerja yang lebih baik dalam hal kompleksitas rata-rata waktu pemrosesan dibandingkan dengan metode lainnya.

Dalam penelitian sebelumnya yang dilakukan oleh (İÇER & TÜRK, 2016) algoritma deteksi tepi diimplementasikan pada *FPGA (Field Programmable Gate Arrays)*. Salah satu gambar contoh digunakan dalam penelitian ini adalah gambar seseorang yang sedang memotret dan diproses dengan menggunakan *FPGA* dan *MATLAB*. Hasil penelitian menunjukkan bahwa *FPGA* memberikan kinerja yang lebih baik daripada *MATLAB* dalam deteksi tepi, dan algoritma *Canny* terbukti memberikan kinerja deteksi tepi terbaik dibandingkan dengan operator *Sobel* dan *Prewitt* karena memiliki nilai *MSE* dan *PSNR* terendah dibandingkan dengan kedua operator tersebut.

Menurut (Ivansyah, 2015) melakukan penelitian eksplorasi implementasi deteksi tepi menggunakan metode *Canny* pada citra mammografi. Dari hasil penelitian ini, dapat disimpulkan bahwa metode deteksi tepi *Canny* yang diimplementasikan menggunakan bahasa pemrograman *MATLAB* mampu mendeteksi tepi citra dengan baik dan dapat diterapkan pada citra mammografi. Kualitas tepi yang dihasilkan sangat dekat dengan tepi pada citra mammografi asli, dan waktu yang dibutuhkan untuk proses deteksi tepi kurang dari 1 detik.

(Tsani & Rachman, 2019) melakukan penelitian mengenai penggunaan Transformasi *Powerlaw* dalam Implementasi Deteksi Tepi *Canny* untuk mendeteksi stadium kanker serviks. Berdasarkan hasil penelitian, dapat disimpulkan bahwa operator deteksi tepi *Canny* dapat menghasilkan tepi citra yang tajam sehingga stadium kanker dapat terlihat dengan jelas pada citra hasil kolposkopi. Pengujian menunjukkan bahwa operator *Canny* dapat melakukan deteksi tepi dengan cepat, yaitu dengan running time 0,0589809 ms, dan rata-rata *MSE* yang lebih kecil menunjukkan tingkat kemiripan citra yang lebih besar, yaitu 11296,3. Sistem berhasil divalidasi oleh dokter keculi untuk dua

gambar yang tidak sesuai, sehingga persentasi akurasi sistem mencapai 80%.

(Tanjungpura, 2019) melakukan penelitian Perbandingan Penggunaan Beberapa Metode Deteksi Tepi Pada Pengolahan Citra Radiologi Fraktur Tulang, ditemukan bahwa meskipun metode *Canny* menghasilkan bentuk tepi citra yang jelas, kualitasnya masih kurang baik dengan nilai PSNR di bawah 40 dB.

(Haris & Prasetyo, n.d.) melakukan studi tentang penerapan algoritma *Canny* pada objek sebagai model keamanan untuk aplikasi pada perangkat smartphone Android. Temuan dari penelitian ini adalah sebagai berikut:

1. Pendeteksian objek dapat menjadi metode alternatif untuk mengamankan aplikasi pada perangkat smartphone Android.
2. Metode deteksi tepi *Canny* dapat diadopsi sebagai metode yang efektif untuk meningkatkan keamanan aplikasi pada perangkat smartphone *Android*.

Terdapat beberapa tahap dalam pengambilan gambar menggunakan kamera mobile yang kami lakukan dalam penelitian ini.

1. Pertama, subjek diminta untuk membersihkan area kulit yang akan diambil gambarnya.
2. Selanjutnya, kamera mobile diarahkan ke area kulit tersebut dan gambar diambil dengan posisi yang tepat dan sesuai dengan standar yang ditentukan. (standar pengambilan gambar : gambar diambil dalam keadaan cukup cahaya, tidak disarankan untuk pengambilan gambar di area kurang cahaya).
3. Setelah gambar berhasil diambil, gambar tersebut kemudian diolah menggunakan algoritma *Canny* untuk deteksi tepi pada gambar kulit.

Tahap-tahap ini dilakukan secara berulang-ulang pada setiap subjek untuk memastikan akurasi dan kevalidan hasil penelitian. Dengan melalui tahap-tahap pengambilan gambar yang terstandarisasi, diharapkan dapat meminimalkan faktor-faktor yang mempengaruhi hasil deteksi tepi pada gambar kulit.