BAB II

LANDASAN TEORI

A. Kajian Pustaka

1. Kamera *Smartphone*

a. Kamera

Kamera adalah alat yang sangat sering digunakan dalam dunia *fotografi*. Kamera digunakan untuk membentuk dan merekam suatu bayangan potret pada lembaran film. (Riyantomo, 2015)

b. Smartphone

Smartphone adalah telephone yang memiliki fitur-fitur dengan kemampuan melebihi telephone pada ummumnya, hal ini bisa dilihat dengan keberadaan fitur tambahan selain untuk komunikasi, seperti fasilitas pendukung tambahan aplikasi. Kemajuan teknologi dan informasi *dapat* dilihat degan semakin banyaknya pengunaan smartphone sebagai alat bantu yang muthakhir, yang bertujuan untuk mempermudah pekerjaan manusia, sehingga waktu yang digunakan semakin cepat, dan mudah. (Solikin, 2018)

2. *Android*

Android merupakan sistem operasi yang dikembangkan untuk perangkat mobile berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2005.Dalam usaha mengembangkan Android, pada tahun 2007 dibentuklahOpen Handset Alliance (OHA), sebuah konsorsium dari beberapa perusahaan,yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel,LG,Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat mobile. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek Android, termasuk PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc (Hermawan S, 2011). (Maiyana, 2018)

3. *Android Studio*

Android studio adalah *IDE (Integrated Development Environment)* resmi untuk pengembangan aplikasi *Android* dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio mengantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi *Android. Android studio* sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). *Android studio* memiliki fitur:

a. Projek 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.

Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan. (Andi, 2015).

4. *Java*

Bahasa pemprograman *Java* merupakan salah satu dari sekian banyak bahasa pemprograman yang dapat dijalankan di berbagai sistem operasi termasuk telepon gengam. Bahasa pemprograman ini pertama kali dibuat oleh James Gosling saat masih bergabung Sun Microsystem.

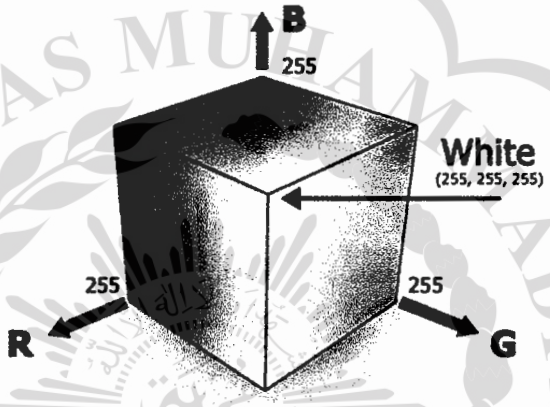
Bahasa pemprograman ini merupakan pengembangan C++, saat ini Java merupakan bahasa pemprograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Kelebihan java dari bahasa pemprograman yang lain adalah bisa dijalankan di berbagai jenis sistem operasi sehingga dikenal juga bahasa pemprograman multiplatform, bersifat pemprograman berorientasi object (PBO), memiliki library yang lengkap.

5. Ruang RGB

Ruang warna RGB *(Red, Green, Blue*) adalah kombinasi warna primer yaitu merah, hijau, dan biru, yang biasa digunakan oleh monitor komputer atau televisi. Warna yang dihasilkan berasal dari kombinasi tiga warna dan masing – masing memiliki nilai 8 bit merah, 8 bit hijau, dan 8 bit biru. Campuran ketiga warna primer tersebut dengan porposi seimbang akan menghasilkan nuansa warna kelabu. Jika ketiga warna ini disaturasikan penuh, maka akan menghasilkan warna putih. (Rulaningtyas et al., 2015)

Gambar warna RGB seperti pada gambar 1,



Gambar 1. Citra warna RGB.

6. *Grayscale*

Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu (Sen, 2018). Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra keabuan hanya perlu menyatakan nilai intensitas untuk tiap pixsel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pixelnya.

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

$Iy = 0{,}333Fr + 0{,}5Fg + 0{,}1666Fb$ (1)

Fr = intensitas R

Fg = intensitas G

Fb = intensitas B

Iy = intensitas abu-abu yang setara gambar level RGB

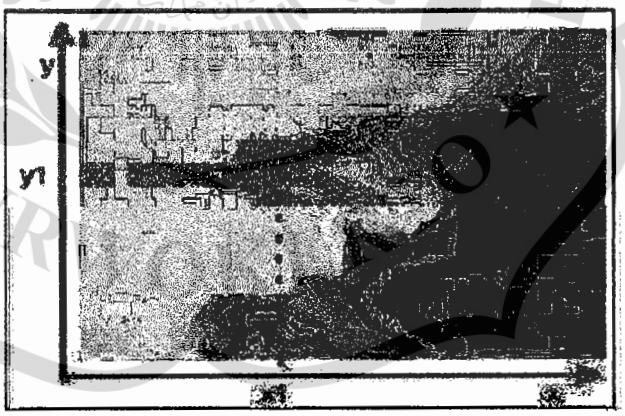
7. *Fotografi*

Fotografi berasal dari dua istilah yunani photo dari phos (cahaya) dan graphy dari graphe (tulisan atau gambar). Maka makna harfiah *fotografi* adalah menulis atau menggambar dengan cahaya. Dengan ini maka identitas fotografi bisa digabungkan menjadi kombinasi dari sesuatu yang terjadi secara alamiah (cahaya) dengan kegiatan yang diciptakan oleh manusia dengan budaya (menulis dan menggambar/melukis). Pada dasarnya *fotografi* adalah kegiatan merekam dan memanipulasi cahaya untuk mendapatkan hasil yang kita inginkan. Fotografi dapat dikategorikan sebagai teknik dan seni (Tanjung, 2019).

8. Citra

Menurut (Fahmi, 2007), pemrosesan citra dengan menggunakan komputer membutuhkan citra digital sebagia masukannya. Citra digital merupakan gambar dua dimensi yang dapat ditampilkan pada layar monitor computer sebagai himpunan berhingga (diskrit) nilai digital yang disebut dengan *Pixel (picture element)*. Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit, baik koordinat ruang maupun intensitas cahayanya.

Citra digital dapat dinyatakan dalam bentuk matriks dua dimensi *f(x,y)* dimana 'x' dan 'y' merupakan koordinat *pixel* dan 'f' merupakan derajat intensitas *pixel* tersebut, hal tersebut diilustrasikan pada gambar dibawah ini.



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

9. *MSE (Mean Squared Error)*

Mean Square Error (MSE) mengukur tingkat perbedaan kuadrat rata-rata. Nilai MSE didapatkan dari nilai selisih citra awal dengan citra hasil dengan posisi piksel yang sama. Semakin tinggi nilai MSE menunjukkan perbedaan besar antara citra awal dengan citra hasil. Perhitungan MSE menggunkan persamaan . Nilai *m*

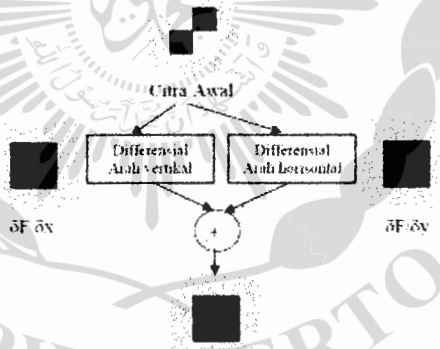
dan n adalah nilai ukuran panjang dan lebar citra. $I(i,j)$ adalah citra awal dan $K(i,j)$ adalah citra hasil deteksi tepi.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

Ketika citra diproses dan dilakukan restorasi, rekonstruksi dan kompresi, nilai MSE akan semakin berkurang. Tetapi, dalam deteksi tepi citra apabila MSE memiliki nilai yang tinggi menunjukkan bahwa lebih banyak tepi gambar yang terdeteksi serta mampu mendeteksi titik-titik tepi gambar yang lemah

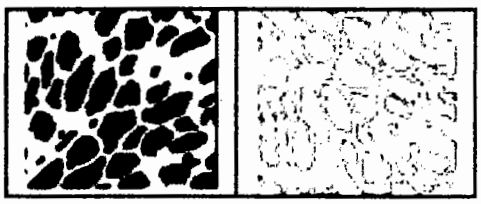
10. Deteksi Tepi

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek gambar. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga (Laxmi et al., 2017). Pada Gambar 3 dapat dilihat proses yang dilakukan untuk memperoleh tepi gambar dari suatu citra yang ada.



Gambar 3. Proses Deteksi Tepi.

Pada Gambar 3 terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu citra terletak pada titik- titik yang memiliki perbedaan intensitas piksel yang tinggi.



Gambar 4. Hasil Deteksi Tepi.

11. Metode *Canny*

Canny adalah algoritma deteksi tepi yang banyak digunakan dalam berbagai penelitian karena dinilai sebagai algoritma deteksi tepi yang paling optimal. Langkah awal pada algoritma *Canny* adalah mengimplementasikan tapis *Gaussian* pada citra untuk menghilangkan derau. Kemudian dilanjutkan dengan melakukan deteksi tepi pada citra dengan salah satu algoritma deteksi tepi yang ada, misalnya *Sobel* atau *Prewitt*. Langkah pertama adalah menghilangkan derau yang ada pada citra dengan menerapkan tapis *Gaussian*. Seperti ditunjukan pada gambar 5.

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

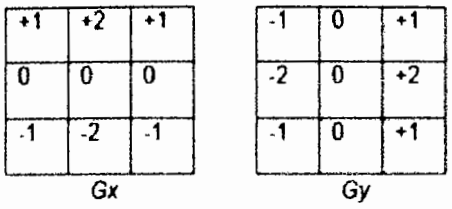
2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Gambar 5. Tapis Gaussian

Selanjutnya melakukan deteksi tepi dengan salah satu operator deteksi tepi (Operator *Sobel* dan *Prewitt*) dengan melakukan pencarian secara horizontal (GX) dan secara vertical (Gy) melalui rumus

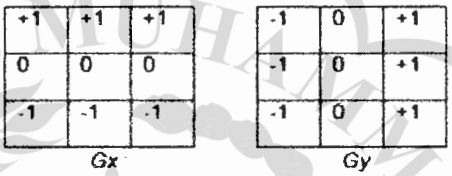
$$|G| = |G \begin{cases} Gx = Pencarian\ secara\ horizontal \\ Gy = Pencarian\ secara\ vertikal \end{cases}$$

Operator Sobel yaitu sepasang kernel berupa matriks berukuran 3x3 untuk mendeteksi tepi vertical dan horizontal.



Gambar 6. Deteksi Tepi Vertikal dan Horisontal I

Prewitt merupakan algoritma deteksi tepi yang hampir serupa dengan Sobel, tetapi algoritma ini menggunakan Operator Prewitt yang nilainya agak berbeda dengan Operator Sobel.



Gambar 7. Deteksi Tepi Vertikal dan Horisontal II

Untuk menghitung jarak gradient, digunakan persamaan berikut :

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

Persamaan gradient 1

Kemudian untuk menghitung arah dari garis tepi yang dihasilkan, digunakan persamaan berikut :

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

Perasamaan gradient 2

Langkah berikutnya adalah membagi garisgaris yang ada menjadi 4 warna terpisah dengan sudut masing-masing dengan ketentuan yaitu :

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.

Kemudian memperkecil masing-masing garis tepi agar menjadi tipis (*non maximum surpression*). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai *low & high threshold* yang diberikan.

B. Studi Pendahuluan

Pada penelitian sebelumnya yang dilakukan oleh (Ahmed, 2018) penulis melakukan penelitian tentang studi perbandingan antara *Sobel*, *Prewitt* dan operator deteksi tepi *Canny* digunakan dalam *Image Processing*. Penelitian ini menggunakan lima gambar sampel daintaranya adalah gambar patung Liberty, gambar menara Eifel, gambar mobil, gambar rumah Jepang dan gambar katedral Saint Basil. Pada penelitian ini ditemukan hasil bahwa metode *Prewitt* memberikan kinerja yang baik dalam hal kompleksitas rata-rata waktu pemrosesan dibanding metode lainnya.

Lalu pada penelitian yang dilakukan oleh (İÇER & TÜRK, 2016) mereka melakukan penelitian tentang implementasi algoritma deteksi tepi terutama digunakan pada FPGA (*Field Programmable Gate Arrays*). Peneltian ini menggunakan satu sampel gambar seseorang sedang memotret yang diproses menggunakan FPGA dan MATLAB. Dari penelitian ini dapat diambil kesimpulan bahwa hasil kinerja FPGA lebih baik dari pada MATLAB dalam kasus deteksi tepi. Pada kasus ini Algoritma *Canny* memiliki kinerja deteksi tepi terbaik daripada operator *Sobel* dan *Prewitt* karena memiliki MSE dan nilai PSNR terendah dibandingkan kedua operator *Sobel* dan *Prewitt*.

Menurut (Ivansyah, 2015) melakukan penelitian tentang implementasi deteksi tepi *canny* pada cita mammografi. Pada penelitian ini dapat di ambil kesimpulan, Deteksi tepi citra dengan metode Canny yang dikembangkan dalam bahasa program (MATLAB) ini dapat mendeteksi tepi citra dengan baik dan dapat diaplikasikan pada citra mammografi. Kualitas tepi yang dihasilkan sangat mendekati dengan tepi pada citra mammografi asli dan waktu yang dibutuhkan

dalam menentukan tepi citra di sini kurang dari 1 detik.

(Tsani & Rachman, 2019) melakukan penelitian tntang Implementasi Deteksi Tepi *Canny* Dengan Transformasi Powerlaw Dalam Mendeteksi Stadium Kanker Serviks. Pada penelitian ini dapat di ambil ksimpulan bahwa diagnosa tingkat stadium kanker pada citra hasil kolposkopi dapat terdeteksi dengan menggunakan operator deteksi tepi canny yang akan menghasilkan tepi-tepi citra yang tajam sehingga lesi kanker dapat terlihat dengan jelas. Selain itu hasil pengujian menunjukkan bahwa operator canny dapat melakukan proses deteksi tepi dengan cepat yaitu dengan nilai running time 0,0589809 ms sementara untuk nilai rata – rata MSE (Mean Square Error) lebih kecil artinya tingkat kemiripan citra lebih besar yaitu 11296,3. Validasi sistem berdasarkan diagnosa dokter hanya 2 gambar yang tidak sesuai, sehingga persentasi akurasi sistem hanyalah 80%.

(Tanjungpura, 2014) melakukan Perbandingan Penggunaan Beberapa Metode Deteksi Tepi Pada Pengolahan Citra Radiologi Fraktur Tulang, pada penelitian ini menunjukkan bahwa metode *Canny* menghasilkan bentuk tepi citra yang terlihat jelas, tetapi dengan kualitas yang masih terbilang buruk karena nilai PSNR di bawah 40 dB.

(Haris & Prasetyo, n.d.) melakukan penelitian tentang implementasi algoritma *Canny* pada onjek sebagai model keamanan aplikasi pada *Smartphone* android. Pada penelitian ini dapat di ambil kesimpulan :

1. Pendeteksian Objek dapat dimanfaatkan sebagai metode alternatif untuk pengunci aplikasi pada smartphone Android
2. Metode Deteksi Tepi Canny dapat digunakan sebagai metode yang dapat diterapkan untuk keamanan pada aplikasi smartphone Android