

## PERANCANGAN APLIKASI DETEKSI KANKER SERVIKS MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

### *APPLICATION DESIGN OF SERVIKS CANCER DETECTOR BASED USING CONVOLUTIONAL NETWORK*

Dhira Nurriszka Agustyawati<sup>1</sup>, Hilman Fauzi<sup>2</sup>, Agus Pratondo<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

<sup>1</sup>dhiranurriszka@student.telkomuniversity.ac.id, <sup>2</sup>hilmanfauzitsp@telkomuniversity.ac.id,

<sup>3</sup>agus@tass.telkomuniversity.ac.id

---

#### Abstrak

Kanker serviks merupakan salah satu penyakit yang mematikan. Menurut WHO (*world health organization*), kanker serviks menempati urutan ke-2 angka kematian yang tinggi setelah kanker payudara dan diperkirakan bahwa 50.000 wanita menderita kanker serviks setiap tahun. Saat ini banyak wanita penderita kanker serviks yang baru menyadari bahwa dirinya memiliki penyakit kanker serviks pada saat sudah stadium lanjut atau stadium akhir, hal ini yang menyebabkan angka kematian penderita kanker serviks sangat tinggi. Salah satu upaya untuk mengurangi penderita kanker serviks adalah dengan mendeteksi potensi kanker serviks sejak dini.

Terdapat tiga metode deteksi dini kanker serviks diantaranya: pap smear, pemeriksaan HPV dan IVA test. Salah satu cara deteksi dini kanker serviks guna menurunkan resiko kanker serviks adalah dengan melalui pemeriksaan IVA (*Inspeksi Visual Asam Asetat*). Pemeriksaan IVA banyak dilakukan karena cenderung lebih murah, serta pemeriksaan dan hasil test diolah langsung, tanpa harus menunggu hasil laboratorium.

Pada tugas akhir ini dilakukan pengolahan citra untuk mengetahui bentuk pola lesi putih berdasarkan data yang di dapat dari hasil pemeriksaan IVA test. Pengolahan citra dilakukan dengan metode *Canny Edge Detection*, kemudian data akan di klasifikasikan menggunakan *Convolutional Neural Network* untuk mengetahui diagnosis pra-kanker serviks. Hasil yang didapatkan pada penelitian ini menghasilkan akurasi deteksi pra-kanker serviks mencapai 96%.

**KATA KUNCI:** kanker serviks, Iva test, pengolahan citra digital, *Canny Edge Detection*, *Convolutional Neural Network*.

---

#### Abstract

*Cervical cancer is a deadly disease. According to the WHO (world health organization), cervical cancer ranks the 2nd highest mortality rate after breast cancer and it is estimated that 50,000 women suffer from cervical cancer every year. Currently, many women with cervical cancer only realize that they have cervical cancer at an advanced or late stage, which causes the death rate of cervical cancer sufferers to be very high. One of the efforts to reduce cervical cancer sufferers is to detect the potential for cervical cancer early.*

*There are three methods of early detection of cervical cancer including: pap smear, HPV examination and IVA test. One way of early detection of cervical cancer in order to reduce the risk of cervical cancer is through an IVA (Acetic Acid Visual Inspection) examination. IVA examinations are often performed because they tend to be cheaper, and the examinations and test results are processed directly, without having to wait for laboratory results.*

*In this final project, image processing is performed to determine the shape of the white lesion pattern based on the data obtained from the results of the IVA test. Image processing was performed using the Canny Edge Detection method, then the data would be classified using a Convolutional Neural Network to determine pre-cervical cancer diagnosis. The results obtained in this study resulted in pre-cervical cancer detection accuracy reaching 96 %*

**KEY WORD:** cervical cancer, Iva test, digital image processing, *Canny Edge Detection*, *Convolutional Neural Network*.

---

## 1. Pendahuluan

Kanker serviks merupakan salah satu penyakit yang mematikan. Menurut WHO (*World Health Organization*), kanker serviks menempati urutan ke-2 angka kematian yang tinggi setelah kanker payudara dan diperkirakan bahwa 50.000 wanita menderita kanker serviks setiap tahun [1]. Saat ini banyak wanita penderita kanker serviks yang baru menyadari bahwa dirinya terdiagnosis kanker serviks pada saat sudah stadium lanjut atau stadium akhir, hal ini yang menyebabkan angka kematian penderita kanker serviks sangat tinggi. Salah satu upaya untuk mengurangi penderita kanker serviks adalah dengan mendeteksi potensi kanker serviks sejak dini. Salah satu cara deteksi dini kanker serviks guna menurunkan resiko kanker serviks adalah dengan melalui pemeriksaan IVA (Inspeksi Visual Asam Asetat). Pemeriksaan IVA banyak dilakukan karena cenderung lebih murah serta pemeriksaan tidak harus menunggu hasil laboratorium [2].

Pada penelitian sebelumnya, pengolahan citra digital dilakukan hanya mengkuantifikasi karakter warna lesi putih yang muncul pada sambungan skuamosa kolumnar (SSK) dengan menggunakan proses deteksi tepi zero crossing dan kuantifikasi nilai RGB [3]. Namun demikian dalam menentukan positif atau negatifnya hasil test IVA, dengan mengetahui karakter warna lesi saja tidak cukup akurat, perlu diketahui parameter penting dalam mendiagnosis kanker serviks yaitu bentuk pola lesi putih yang muncul pada jaringan epitel SSK.

Berdasarkan uraian tersebut untuk mengetahui pola lesi kanker serviks yang terbentuk pada jaringan epitel SSK, maka disusun penelitian ini menggunakan metode segmentasi citra *Canny Edge Detection* untuk memisahkan antar region dalam suatu citra berdasarkan pada perbedaan nilai intensitas piksel dari citra *greyscale* dengan jumlah nilai threshold lebih dari satu [9]. Input dimulai dengan memasukan gambar lalu dilakukan histogram 8 Bin, kemudian dilakukan *thresholding* dan *gaussian filter* guna membantu proses *Canny Edge Detection*, lalu diklasifikasikan menjadi CNN. Kemudian data diklasifikasikan menggunakan *Convolutional Neural Network* untuk pengelompokan data dari berbagai macam karakteristik yang berbeda-beda. Oleh penelitian ini sistem deteksi pra-kanker serviks mencapai akurasi optimal 96%.

## 2. Tinjauan Pustaka

### 2.1 Kanker Serviks

Kanker serviks atau kanker leher rahim adalah penyakit yang tumbuh pada sel-sel dibagian leher rahim disebabkan oleh Infeksi *Human Papilloma Virus* (HPV) tipe 16 dan 18 [4]. Kanker serviks terbentuk sangat perlahan. Umumnya, kanker serviks tidak menunjukkan gejala pada tahap awal. Gejala baru muncul saat kanker sudah mulai menyebar luas, serta terdeteksi awalnya dari kelainan pada sel-sel serviks yang disebut displasia [5]. Beberapa faktor penyebab terjadinya resiko kanker serviks diantaranya:

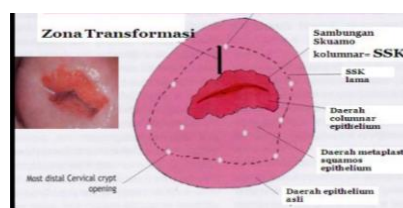
- Hubungan seksual pada usia terlalu dini [4].
- Berganti-ganti pasangan untuk berhubungan seksual, hal ini disebabkan terpaparnya sel-sel mulut rahim yang mempunyai pH tertentu dengan sperma yang mempunyai pH yang berbeda-beda pada *multipatner*, sehingga dapat merangsang terjadinya perubahan kearah *dysplasia* [4].
- Wanita merokok, memungkinkan nikotin (zat racun) masuk ke dalam darah mampu meningkatkan kemungkinan terjadinya kondisi *cervical neoplasia* atau tumbuhnya sel-sel abnormal pada rahim [4].

### 2.2 Inspeksi Visual Asam Asetat (IVA) test

Inspeksi Visual Asam Asetat (IVA) test merupakan metode yang dapat dilakukan untuk deteksi kanker serviks dengan cara yang mudah, murah dan cepat. Dilakukan dengan langkah skrining yaitu dengan memulas serviks menggunakan asam asetat 3-5% hanya pada area serviks saja didalam jaringan epitel SSK, kemudian diinspeksi secara kasat mata oleh tenaga medis yang terlatih tanpa menggunakan alat pembesar apapun. Setelah serviks diulas dengan asam asetat, akan terjadi perubahan warna atau bercak putih yang muncul pada area serviks yang dapat diamati secara langsung dan dapat dibaca sebagai normal atau abnormal [3].

### 2.3 Sambungan Skuamosa Kolumnar

SSK (sambungan skuamosa kolumnar) merupakan perbatasan epitel skuamous berlapis dari ekto serviks dan epitel kolumnar endoserviks. Pada SSK terdapat epitel kolumnar dan epitel skuamous. Diantara epitel kolumnar dan epitel skuamous terdapat sambungan Skuamokolumnar yang ditandai dengan metaplasia skuamosa [3]. Pada gambar 2.1 merupakan letak SSK berada.



**Gambar 2.1** posisi SSK  
(Sumber: materi pelatihan TOT PTM)

Pada gambar 2.1 Sambungan Skuamosa kolumnar terlihat sebagai garis yang berbeda diantara daerah kolumnar epitel dan skuamosa. Letak sambungan skuamosa kolumnar yang berhubungan dengan os eksternal (SSK asli) beragam sesuai dengan faktor yang mempengaruhinya seperti usia, perkembangan hormon, trauma kelahiran, penggunaan kontrasepsi oral [3].

#### 2.4 Pengolahan citra

Pengolahan citra (*image processing*) merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer. Beberapa contoh bidang kehidupan yang membutuhkan pengolahan citra digital diantaranya adalah bidang kesehatan: segmentasi untuk membedakan bagian - bagian sel darah, deteksi kerusakan organ tubuh, deteksi keberadaan tumor [7]. Pengolahan citra dilakukan untuk melakukan transformasi suatu gambar/citra menjadi informasi yang dapat dipahami oleh manusia dengan menggunakan Teknik tertentu diawali dengan akuisisi citra atau proses menangkap citra analog sehingga diperoleh citra digital menggunakan kamera digital. Kemudian dilakukan proses *pre-processing* untuk menghilangkan noise, dalam hal ini noise yang ada pada permasalahan akan disaring atau dihilangkan sehingga tidak merusak citra serta menentukan bagian citra yang akan di observasi. Selanjutnya metode segmentasi dilakukan untuk membagi citra menjadi beberapa bagian-bagian pokok yang mengandung informasi penting. Misalnya memisahkan objek dan latar belakang. ekstraksi ciri dan seleksi direpresentasikan menjadi suatu persamaan yang dapat membedakan kelas-kelas objek citra nantinya.

#### 2.5 Citra Digital

Berdasarkan bentuk sinyal penyusunnya, citra dapat digolongkan menjadi dua jenis yaitu citra analog dan citra digital. Citra analog adalah citra yang dibentuk dari sinyal analog yang bersifat kontinu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reperesentasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris, dimana perpotongan antara kolom dan baris disebut piksel (pixel = picture element) atau elemen terkecil dari sebuah citra [8].

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Gambar 2.2 bentuk matriks

Berdasarkan gambar 2.2, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas  $f(x,y)$ , dimana harga  $x$  (baris) dan  $y$  (kolom) merupakan koordinat posisi dan  $f(x,y)$  adalah nilai fungsi pada setiap titik  $(x,y)$  yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (sampling dan kuantitas) diperoleh besar baris  $M$  dan kolom  $N$  hingga citra membentuk matriks  $M \times N$  dan jumlah tingkat keabuan piksel  $G$ .

#### 2.6 Histogram 8 Bin

Visualisasi selalu menjadi cara yang efisien untuk merepresentasikan dan menjelaskan banyak detail statistik. Dalam pemrosesan gambar, histogram digunakan untuk menggambarkan banyak aspek mengenai gambar yang sedang kita kerjakan. Seperti: *Exposure*, *Contrast*, *Dynamic Range*, *Saturation*, dan masih banyak lagi. Dengan memvisualisasikan histogram, kita dapat meningkatkan tampilan visual suatu citra dan juga dapat mengetahui jenis pemrosesan citra apa yang dapat diterapkan dengan membandingkan histogram suatu citra.

Gambar disimpan sebagai nilai piksel, setiap nilai piksel mewakili nilai intensitas warna. Histogram adalah distribusi frekuensi dari nilai intensitas yang terjadi pada gambar.

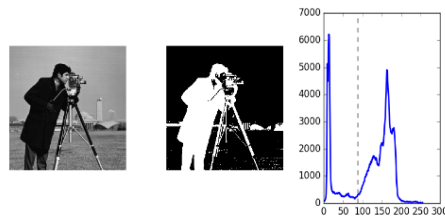
$h(i)$  = banyaknya pixel pada  $I$  (*image*) dengan nilai intensitas  $i$

Misalnya, jika  $i = 0$ , maka  $h(0)$  adalah banyaknya piksel dengan nilai 0.

Biasanya, kisaran nilai intensitas gambar adalah dari  $[0-255]$  dalam representasi 8 bit ( $2^8$ ). Tetapi gambar juga dapat direpresentasikan dengan menggunakan  $2^{16}$ ,  $2^{32}$  bit dan seterusnya. Dalam kasus seperti ini, kisaran intensitasnya tinggi dan sulit untuk merepresentasikan setiap nilai intensitas dalam histogram, menggunakan *binning* untuk mengatasi masalah di atas. Di sini menghitung rentang menjadi beberapa kelompok.

#### 2.7 Thresholding

*Thresholding* merupakan suatu algoritma segmentasi yang sederhana dengan melakukan pendekatan piksel citra pada ruang warna dengan toleransi tertentu [13]. Setiap ruang warna merepresentasikan suatu citra dengan nilai unik yang telah dihitung oleh setiap *channel* pada setiap piksel [8]. Keunikan nilai tersebut memungkinkan adanya partisi ruang warna menggunakan batasan linear [8]. Misalnya, kita dapat memilih suatu piksel dengan karakteristik yang sama dengan karakteristik yang telah ditetapkan di awal [8]. Ide tersebut dapat digunakan untuk mensegmen objek yang kita inginkan dari suatu citra [8]. Berikut gambar 2.3 merupakan contoh hasil segmentasi objek kulit menggunakan ruang warna HSV:



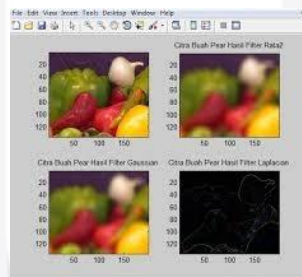
Gambar 2.3 Thresholding

### 2.8 Canny Edge Detection

Algoritma *Canny edge detection* dinamai penemunya, John F. Canny, yang menemukan algoritma pada tahun 1986. *Canny edge detection* biasanya mengambil gambar skala abu-abu sebagai input dan menghasilkan gambar yang menunjukkan lokasi diskontinuitas intensitas sebagai output (yaitu tepi). Saya tidak ingin menggunakan matematika di sini, tetapi saya akan menjelaskan apa yang terjadi di balik layar dalam algoritma *Canny edge detection* dari sudut pandang tingkat tinggi. Hal pertama yang dilakukan oleh Canny edge detection adalah menggunakan *Gaussian convolution* untuk menghaluskan gambar input dan menghilangkan *noise*. Operator turunan pertama kemudian diterapkan pada gambar yang dihaluskan untuk menyoroti daerah-daerah gambar dengan turunan spasial pertama yang tinggi.

### 2.9 Gaussian

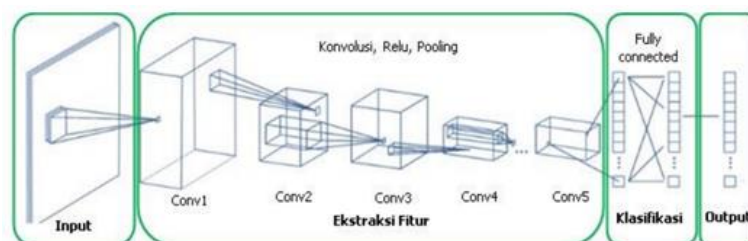
*Gaussian filter* adalah *linear filter* yang biasanya digunakan sebagai pengolah citra agar dapat lebih halus. *Gaussian filter* yang banyak digunakan dalam memproses gambar. *Gaussian filter* bertujuan untuk menghilangkan *noise* pada citra dan meningkatkan kualitas detail citra. Gaussian merupakan model *noise* yang mengikuti distribusi normal standard dengan rata-rata nol dan standard deviasi 1. Efek dari gaussian ini, pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan presentase *noise*. *Noise speckle* merupakan model *noise* yang memberikan warna hitam pada titik yang terkena *noise*. Sedangkan *noise salt & pepper* seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena *noise*. Gambar 2.4 dibawah merupakan contoh hasil dari proses gaussian filter.



Gambar 2.4 Contoh Gaussian Filter

### 2.10 Convolutional Neural Network

*Convolutional Neural Network* (CNN) termasuk dalam jenis deep learning karena kedalaman jaringannya. Deep learning adalah cabang dari machine learning yang dapat mengajarkan komputer untuk melakukan pekerjaan selayaknya manusia, seperti komputer dapat belajar dari proses *training*. Pada CNN setiap *neuron* dipresentasikan dalam bentuk 2 dimensi, sehingga metode ini cocok untuk pemrosesan dengan input berupa citra. Arsitektur jaringan dengan menggunakan CNN ditunjukkan pada Gambar 2.5. Struktur CNN terdiri dari input, proses ekstraksi fitur, proses klasifikasi dan *output*. Proses ekstraksi dalam CNN terdiri dari beberapa lapisan tersembunyi atau *hidden layer*, yaitu lapisan konvolusi, fungsi aktivasi (*Relu*), dan *Pooling*. CNN bekerja secara hierarki, sehingga *output* pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari *fully-connected* dan fungsi aktivasi (*softmax*) yang *output*nya berupa hasil klasifikasi.



Gambar 2.5 arsitektur CNN

### 2.11 Deep Learning

*Deep learning* merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning* [7]. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *machine learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara [7]. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi. Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditanggihkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar [7].

Prinsip ini terus berkembang hingga *Deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *computer vision*, *speech recognition*, dan *Natural Language Processing* [7]. *Feature Engineering* adalah salah satu fitur utama dari *Deep learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga.

## 2.12 Python

*Python* adalah salah satu bahasa pemrograman komputer yang populer dan sering digunakan terutama dalam melakukan perhitungan. Bahasa pemrograman ini dapat dijalankan hampir di semua platform, seperti *Linux*, *Windows*, dan *Machintos*. Pada bahasa pemrograman *Python*, deklarasi suatu variabel dapat dilakukan secara langsung tanpa menyebutkan tipe datanya.

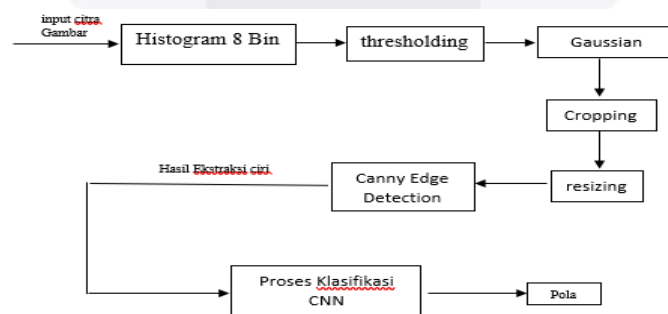
Bahasa pemrograman *Python* ini pertama kali dibuat oleh Guido van Rossum pada awal tahun 1990 di negeri Belanda sebagai pengganti bahasa pemrograman yang disebut ABC. Bahasa pemrograman *Python* merupakan bahasa pemrograman yang dapat dikembangkan oleh siapa saja karena bersifat Open Source, dapat digunakan tanpa lisensi, dan dapat dikembangkan semampu yang dapat dilakukan. Bahasa pemrograman *Python* ini memiliki efisiensi tinggi untuk struktur data level tinggi, pemrograman berorientasi objek lebih sederhana tetapi efektif, dapat bekerja pada multi-platform, dan dapat digabungkan dengan bahasa pemrograman lain untuk menghasilkan aplikasi yang diinginkan [11].

## 2.13 Android

Android adalah sebuah sistem operasi berbasis linux yang dirancang untuk perangkat mobile layar sentuh seperti telepon pintar dan komputer tablet. Android bersifat terbuka atau *open source* yang memungkinkan pengembang menciptakan aplikasinya sendiri. Android adalah sistem operasi berbasis Linux untuk perangkat seluler seperti smartphone dan tablet yang bersifat open source. Pada tahun 2005 Google membeli android Inc., dan dijadikan bisnis ponsel cerdas untuk perangkat lunak dan menarik perhatian HTC, yang menggunakan platform untuk ponsel Android pertama, HTC Dream, pada tahun 2008 [12]. Android hadir dengan sekumpulan aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan berbagai fitur lain. Semua aplikasi menggunakan Bahasa pemrograman Java. Android memberikan platform terbuka untuk pengembang agar membuat aplikasi mereka sendiri [13].

## 3. Pemodelan sistem

### 3.1 Gambaran umum



**Gambar 3.1** Diagram Blok tahapan permodelan sistem

Gambar 3.1 merupakan tahapan-tahapan yang ada pada sistem yang dibangun dalam melakukan deteksi kanker serviks. Tahap pertama adalah tahap ekstraksi citra. Pada tahap ini dilakukan *feature extration* atau ekstraksi citra pada dataset yang digunakan. *feature extration* pada tahap ini berupa dilakukannya proses *histogram 8 bin* untuk mencari *intensitas value* dari setiap citra dataset, hasil dari histogram 8 Bin adalah intensitas value dari warna *Red Green Blue*. Setelah proses *histogram 8 bin*, maka akan dilakukan proses *thresholding* dan *gaussian filter* untuk mengubah kualitas gambar. Hal ini dilakukan agar proses klasifikasi lebih mudah. Selanjutnya dilakukan proses *cropping* dan *resizing* lalu deteksi tepi atau deteksi pola untuk menghasilkan pola dari setiap gambar positif dan negatif.

Setelah proses ekstraksi citra selesai, tahap berikutnya adalah pembentukkan model *Convolutional Neural Network*. Tahap ini diawali dengan menentukan layers yang digunakan, jumlah dense pada setiap layers, mencari

nilai relu dan menghasilkan *output layers*. Setelah model terbentuk, maka akan dilakukan pengujian, lalu untuk menghitung tingkat akurasi, selanjutnya akan digunakan *confusion matrix* atau matriks konfusi.

### 3.1 Data Masukan

Data masukan pada penelitian ini di dapat dari gambar yang diambil dari pemeriksaan IVA Test di Rumah Sakit Hasan Sadikin, Bandung. Pengambilan gambar menggunakan kamera dengan resolusi minimal 8 MP pada OS Android minimal 4.0 Kitkat dengan pengambilan gambar secara langsung dilakukan pada jarak 10 – 15 cm tepat didepan mulut Rahim, untuk intensitas cahaya pada area sekitar pengambilan gambar sebesar 200 – 500 lux dan kemudian data disimpan dalam format .jpg dan .png. Data masukan yang ada pada penelitian ini terbagi kedalam dua class yaitu *class* positif dan *class* negatif. Berikut gambar 3.2 merupakan contoh citra data masukan:



**Gambar 3.2** Citra serviks

Masukan Pada citra tersebut, akan dideteksi apakah pola tersebut data masukannya berupa positif ataupun negatif. Masing-masing *class* tersebut mempunyai ciri tersendiri dalam polanya. Adapun jumlah *class* dan masing-masing jumlah citra serviks positif sebanyak 77 dan citra serviks negatif sebanyak 82.


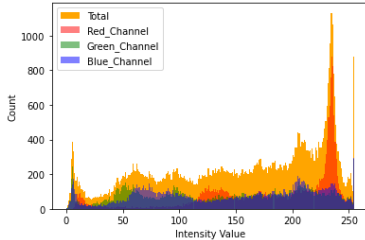

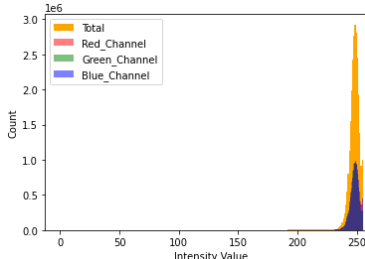
### 3.2 Ekstraksi Citra

Hal yang pertama kali dilakukan pada dataset adalah tahapan *preprocessing*. Tujuan dari *preprocessing* ini adalah untuk mendapatkan data yang baik dan sesuai dengan apa yang dibutuhkan sistem.

#### 1. Tahap Histogram 8 Bin

Histogram 8 Bin Visualisasi selalu menjadi cara yang efisien untuk merepresentasikan dan menjelaskan banyak detail statistik. Dalam pemrosesan gambar, histogram digunakan untuk menggambarkan banyak aspek mengenai gambar yang sedang dianalisis. Dengan memvisualisasikan histogram, kita dapat meningkatkan tampilan visual suatu citra dan juga dapat mengetahui jenis pemrosesan citra apa yang dapat diterapkan dengan membandingkan histogram suatu citra. Gambar disimpan sebagai nilai piksel, setiap nilai piksel mewakili nilai intensitas warna. Histogram adalah distribusi frekuensi dari nilai intensitas yang terjadi pada gambar. Dalam gambar berwarna, terdapat 3 saluran warna yang mewakili RGB. Dalam Histogram Warna Gabungan, jumlah intensitas adalah jumlah dari ketiga saluran warna. Biasanya, kisaran nilai intensitas gambar adalah dari [0-255] dalam representasi 8 bit ( $2^8$ ). Tetapi gambar juga dapat dipresentasikan menggunakan  $2^{16}$ ,  $2^{32}$  bit dan seterusnya. Dalam kasus seperti ini, kisaran intensitasnya tinggi dan sulit untuk merepresentasikan setiap nilai intensitas dalam histogram. Penelitian ini menggunakan binning untuk mengatasi masalah tersebut. Berikut merupakan hasil histogram 8 Bin pada Data Masukan. Pada Tabel 3.1 merupakan contoh hasil dari histogram pada citra kanker serviks.

**Tabel 3.1** Hasil Histogram

Class	Citra	Hasil Histogram
Positif		
Negatif		

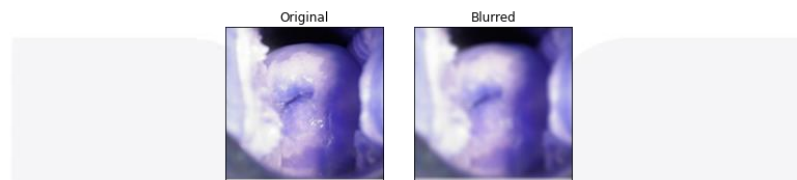
Tahapan ini dilakukan agar mendapatkan gambar atau citra yang telah diubah menggunakan thresholding. Berikut ini gambar 3.3 merupakan hasil citra thresholding. Parameter thresholdingnya yaitu Binary, Binary\_inv, trunc, tozero dan tozero\_inv. digunakan parameter untuk mendapatkan derajat keabuan. adapun batas bawah pada IVA negatif yaitu 110 dan batas atas 140. Sedangkan untuk IVA positif batas bawahnya 75 dan batas atasnya 150. sedangkan batas bawah untuk IVA positif dan negatif yang dipakai secara umum untuk seluruh parameter thresholding bernilai 127 dan 255.



Gambar 3.3 Hasil citra Thresholding

3. Gaussian Filter

Gaussian filter digunakan untuk menghaluskan citra guna membantu proses canny edge detection lebih mudah. parameter yang digunakan dalam melakukan gaussian yaitu berukuran 5x5. Berikut gambar 3.4 hasil gaussian filter.



Gambar 3.4 Hasil gaussian filter

3.3 Canny Edge Detection

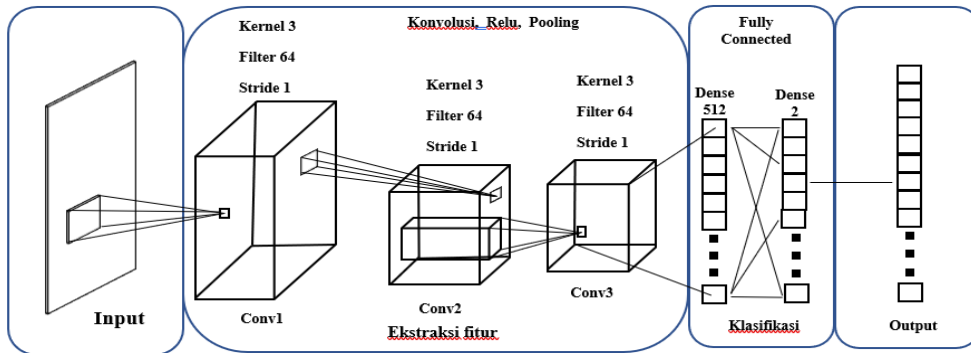
Ada beberapa kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh operator canny:

- a. Mendeteksi dengan baik (kriteria deteksi) Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.
- b. Melokalisasi dengan baik (kriteria lokalisasi) Dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
- c. Respon yang jelas (kriteria respon) Hanya ada satu respon untuk tiap tepi, sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pemilihan parameter deteksi tepi Canny sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut adalah nilai standar deviasi Gaussian dan nilai ambang (threshold). Berikut Tabel 3.2 merupakan hasil deteksi tepi Canny pada penelitian ini.

Tabel 3.2 Hasil deteksi tepi

Class	Citra	Hasil deteksi tepi
Positif		
Positif		

Pelatihan *Convolutional Neural Network* yang dibangun seperti pada Gambar 3.5 berikut.



Gambar 3.5 Model CNN

Proses klasifikasi menggunakan metode *Convolutional Neural Network*. pada penelitian ini, digunakan 3 *convolutional* layer pada proses klasifikasi dimana masing-masing *convolutional* memiliki kernel dengan masing-masing filter dan dropout sebesar 0,2. Pada proses *training*, diterapkan beberapa kernel yang berbeda untuk mendapatkan hasil akurasi yang paling baik. Proses input merupakan gambar IVA positif maupun negatif yang telah dilakukan proses *resizing*. Adapun proses tersebut dilakukan dengan *me-resize* gambar menjadi 50%. Klasifikasi dilakukan dengan membagi data latih dan data uji dengan komposisi masing-masing 80% dan 20%. Pada setiap *convolutional* menghasilkan nilai *max pooling* dan *relu* untuk selanjutnya diproses pada *convolutional* berikutnya. Setelah didapatkan hasil *max pooling* dan *relu* selanjutnya dicari nilai *Fully Connected (FC)*. nilai FC akan dikalikan dengan bobot pada proses klasifikasi untuk menentukan bobot setiap *class*.

### 3.6 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dibagi menjadi dua kebutuhan, yang pertama adalah kebutuhan non-fungsional sedangkan yang kedua adalah kebutuhan fungsional. Untuk lebih jelasnya, tertera pada Tabel 3.3 dan 3.4 di bawah ini. Berikut merupakan spesifikasi kebutuhan perangkat lunak fungsional pada pembuatan aplikasi ini. Ini menjelaskan tentang fitur apa saja yang dapat digunakan user.

Tabel 3.3 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem dapat melakukan tambah data menggunakan kamera
SKPL-F-002	Sistem dapat melakukan tambah data menggunakan canvas gambar
SKPL-F-003	Sistem dapat melakukan deteksi

Tabel 3.4 Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem yang dibangun menggunakan bahasa pemrograman <i>python 3.6</i> , sehingga dibutuhkan interpreter python yang sudah terpasang
SKPL-NF-002	Tampilan antarmuka dibuat menggunakan <i>Bahasa Pemrograman Java</i>
SKPL-NF-003	Dibutuhkan <i>anaconda, library pandas, dan Keras, tensorflow</i>

#### 3.6.1 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan *non-fungsional* merupakan analisis kebutuhan di luar fungsi dari sistem, berisi hal-hal mengenai kebutuhan sekunder dari sistem yang dibangun. Analisis ini terdiri dari analisis perangkat keras, analisis perangkat lunak, dan analisis pengguna. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan. Berikut adalah penjelasan masing-masing analisis kebutuhan *non-fungsional* dari sistem yang dibangun:

##### 3.6.1.1 Analisis Kebutuhan Perangkat Lunak



Analisis kebutuhan perangkat lunak adalah analisis sebuah perangkat lunak yang digunakan dalam proses pembangunan sebuah sistem. Untuk analisis kebutuhan perangkat lunak dari sistem yang dibangun adalah sebagai berikut:

- *Operating System*: Windows 10 pro 64 bit
- *Integrated Development Environment (IDE)*: Android Studio, Jupyter Notebook.
- *Virtual Environment*: Anaconda, pandas, Tensorflow, Keras.
- *Modelling Tools*: Draw.io

### 3.6.1.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras adalah analisis sebuah perangkat keras yang digunakan dalam proses pembangunan sistem. Untuk analisis kebutuhan perangkat keras dari sistem yang dibangun adalah sebagai berikut:

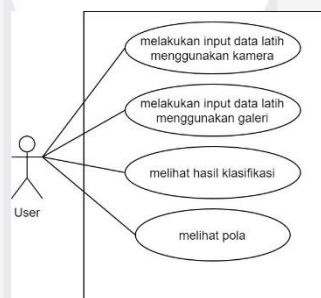
- Processor: AMD Quad Core Processor A8-7410 APU @2.2GHz
- RAM: DDR3 12GB
- Storage: 1TB Harddisk
- Monitor dengan resolusi minimal 800 x 600.
- Keyboard
- *Mouse atau touchpad.*

### 3.6.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang terjadi pada sebuah sistem. Pembangunan perangkat lunak pada penelitian ini menggunakan konsep OO (*Object-Oriented*). Sehingga analisis kebutuhan fungsional yang digunakan pada penelitian ini dimodelkan menggunakan *Unified Modeling Language (UML)*. Analisis kebutuhan fungsional pada penelitian ini terdiri dari *use case diagram* dan *activity diagram*.

#### 1. Use case diagram.

*use case diagram* sebagai berikut:

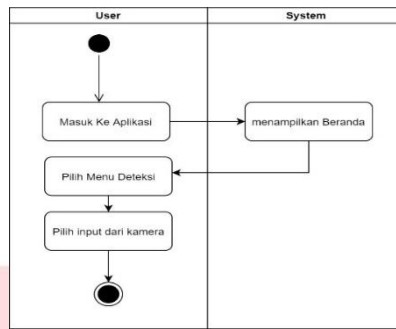


**Gambar 3.6** Use Case Diagram

Pada aplikasi deteksi ini, seperti yang terlihat pada gambar 3.6 aplikasi hanya digunakan oleh 1 user dimana user dapat melakukan 4 kegiatan di aplikasi, yaitu melakukan input data latih menggunakan kamera, melakukan input data menggunakan foto dari galeri, lalu melihat hasil klasifikasi dan melihat pola.

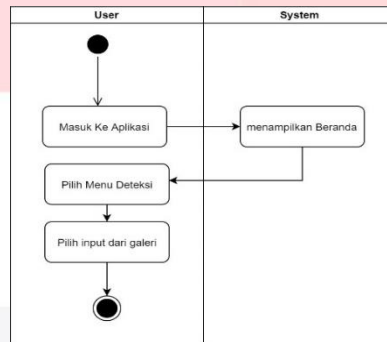
#### 2. Activity Diagram

*Activity Diagram* Melakukan Input Data Latih Menggunakan kamera terlihat pada gambar 3.7 pada proses ini ketika akan memasukan citra untuk data latih, user harus masuk ke aplikasi dan memilih menu input dari kamera.



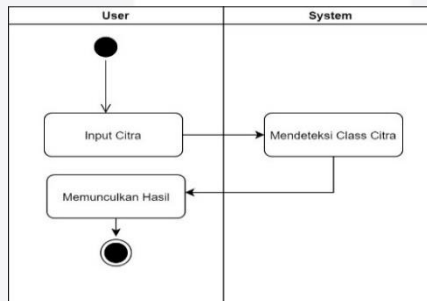
**Gambar 3.7** Activity Diagram Melakukan Input  
Kemudian ada Input Data Latih

Data Latih Menggunakan kamera  
Activity Diagram Melakukan Menggunakan Gallery terlihat



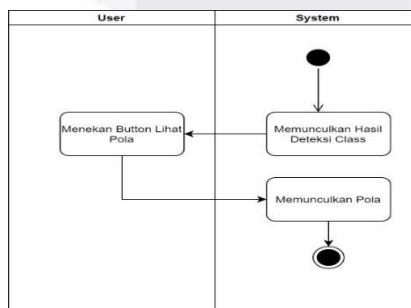
**Gambar 3.8** Activity Diagram Melakukan Input Data Latih Menggunakan galeri

Kemudian untuk Activity Diagram Melihat Hasil Klasifikasi dapat dilihat pada gambar 3.9 dibawah ini. pada proses ini ketika akan melihat hasil klasifikasi, citra diinputkan terlebih dahulu lalu muncul hasil klasifikasi apakah negative atau positif.



**Gambar 3.9** Activity Diagram melihat hasil klasifikasi.

Kemudian pada deteksi pra-Diagram gambar 3.10 dibawah

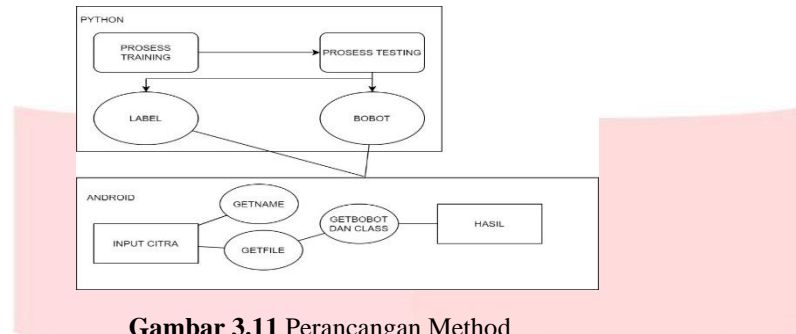


untuk mengetahui bentuk pola kanker serviks ada Activity memunculkan Pola seperti pada ini.

**Gambar 3.10** Activity Diagram melihat pola

### 3.6.3 Perancangan Method / Alur Program

Perancangan *method* menggambarkan alur program yang berjalan. Berikut pada gambar 3.11 merupakan gambaran perancangan method.



**Gambar 3.11** Perancangan Method

Setelah dilakukan proses *training* dan proses *testing* akan menghasilkan bobot dan label untuk di proses pada program aplikasi android. Pada android untuk melihat hasil klasifikasi dilakukan proses input citra kemudian pilih file dan jenis file nya, lalu program akan membaca bobot yang sebelumnya telah diketahui, kemudian akan menghasilkan klasifikasi dan yang terakhir akan mengeluarkan hasil akurasi.

### 3.7 Pengujian Sistem

Pengujian sistem dilakukan setelah sistem pendeteksi lesi pra kanker telah dibuat. Pengujian sistem dilakukan dengan tujuan:

1. Mengetahui kinerja aplikasi sistem dari aplikasi pendeteksi.
2. Mengetahui hasil pengujian pola/bentuk lesi putih

### 3.8 Parameter Pengujian Sistem

Parameter yang digunakan untuk pengujian sistem aplikasi:

1. Tingkat akurasi sistem

Tingkat akurasi sistem didapatkan dari perbandingan jumlah data uji benar dengan data uji. Data uji benar didapatkan dari kecocokan hasil yang dikeluarkan sistem dengan penilaian spesialis ginekologi. Data yang diuji sebanyak 159 citra serviks: 82 citra serviks iva negatif dan 77 gambar serviks iva positif. Berikut skema pengujian untuk mendapatkan tingkat akurasi sistem:

- a. Pengujian dengan gambar asli.
- b. Pengujian dengan gambar yang telah dilakukan tresholding.

Rumus yang digunakan untuk mendapatkan tingkat akurasi sistem (dilakukan setiap sekenario pengujian):

$$\text{Akurasi (\%)} = \frac{\text{total data benar}}{\text{total data uii}} \times 100\% \quad (3.1)$$

2. Tingkat sensitivitas dan spesifisitas sistem

Tingkat sensitivitas dan spesifisitas sistem didapat melalui rumus dari hasil pengujian data oleh sistem. Kecocokan hasil pengujian didapat berdasarkan pendapat spesialis ginekologi. Data yang diuji sebanyak 159 citra serviks: 82 citra serviks iva negatif dan 77 gambar serviks iva positif. Rumus yang digunakan untuk mendapatkan tingkat sensitivitas dan spesifisitas sistem:

$$\text{Sensitivitas(\%)} = \frac{\text{data positif benar}}{\text{data positif benar} + \text{data negatif palsu}} \times 100\% \quad (3.2)$$

$$\text{Spesifisitas(\%)} = \frac{\text{data negatif benar}}{\text{data negatif benar} + \text{data negatif palsu}} \times 100\% \quad (3.3)$$

#### 4. Hasil dan Analisis

Pada tahap analisis hasil, hasil evaluasi didapatkan dari data uji dan data latih. Data yang digunakan berjumlah 159 baik class positif maupun negatif, setiap class memiliki deteksi masing-masing. Namun karena data yang tidak seimbang, perlu dilakukan oversampling sehingga tidak ada informasi dari data latih asli yang terbangun, karena dengan teknik ini akan menyimpan seluruh data minoritas maupun mayoritas. Setelah dilakukan oversampling secara acak, total data berjumlah 159 dengan masing-masing kelas positif dan negatif sebanyak 77 dan 82. Karena melalui ekstraksi citra dan untuk menghilangkan data kosong/terbuang. Untuk membagi data, digunakan 30% secara acak sebagai data uji dan sisanya sebagai data latih. Kemudian ketika pelatihan data, pada data latih dibagi kembali sebanyak 30% menjadi data uji dan data latih sebagai validasi model. Pada pembahasan evaluasi ini di bawah ini, digunakan data citra sebanyak 159. Berikut pembahasan evaluasi dari setiap skenario:

##### 4.1 Analisis Skenario 1

Analisis Skenario 1 merupakan skenario dimana pelatihan dilakukan perubahan jumlah epoch. Besaran akurasi pada proses *testing* dan *training* dipengaruhi dengan besaran jumlah epoch yang dilakukan. Epoch merupakan jumlah data citra dilakukan pelatihan. Adapun pengujian akurasi dengan merubah jumlah epoch dilakukan perbandingan seperti tabel 4.1 berikut:

**Tabel 4.1** Akurasi dengan epoch

No	Jumlah Epoch	Akurasi
1	10	65%
2	20	74%
3	30	96%
4	40	96%

sesuai tabel 4.1 didapatkan akurasi dengan beberapa perubahan jumlah epoch, *epoch* yang paling optimal untuk menghasilkan akurasi yang baik adalah *epoch* 30 dengan menghasilkan akurasi 96%. sedangkan akurasi yang lain yaitu 10 *epoch* menghasilkan 65% dan *epoch* 20 menghasilkan akurasi 74%, dengan demikian dilihat dari perubahan yang signifikan semakin besar jumlah epoch maka akan semakin besar akurasi yang dihasilkan sampai titik yang optimal, karna dapat dilihat ketika pada epoch 40 akurasi tidak lagi bertambah melainkan sudah pada titik yang optimal yakni epoch 30.

##### 4.2 Analisis Skenario 2

Setelah diketahui bahwa akurasi yang optimal adalah menggunakan 30 epoch. maka pada skenario 2 akan dilakukan pengujian dengan 30 epoch dan beberapa jumlah kernel dan filter untuk diketahui jumlah kernel terbaik untuk menghasilkan akurasi terbaik untuk hasil pengujian akurasi dengan beberapa parameter tersebut dapat dilihat pada tabel 4.2 dibawah ini.

**Tabel 4.2** Skenario 2

Skenario	Epoch	Metode	Kernel	Filter	Dropout	Hidden Dims	Akurasi
2	30	CNN	1	64	0.25	128	56%
	30		1	128	0.25	128	76%
	30		1	192	0.25	128	65%
	30		1	256	0.25	128	59%
	30		1	320	0.25	128	68%
	30		1	384	0.25	128	77%
	30		1	448	0.25	128	60%
	30		1	512	0.25	128	57%
	30		1	576	0.25	128	77%
	30		1	640	0.25	128	84%

30	3	64	0.25	128	96%
30	3	128	0.25	128	52%
30	3	192	0.25	128	52%
30	3	256	0.25	128	96%
30	3	320	0.25	128	83%
30	3	384	0.25	128	53%
30	3	448	0.25	128	60%
30	3	512	0.25	128	86%
30	3	576	0.25	128	52%
30	3	640	0.25	128	60%
30	5	64	0.25	128	62%
30	5	128	0.25	128	60%
30	5	192	0.25	128	84%
30	5	256	0.25	128	80%
30	5	320	0.25	128	61%
30	5	384	0.25	128	60%
30	5	448	0.25	128	64%
30	5	512	0.25	128	63%
30	5	576	0.25	128	79%
30	5	640	0.25	128	60%
30	7	64	0.25	128	75%
30	7	128	0.25	128	79%
30	7	192	0.25	128	64%
30	7	256	0.25	128	63%
30	7	320	0.25	128	82%
30	7	384	0.25	128	72%
30	7	448	0.25	128	70%
30	7	512	0.25	128	61%
30	7	576	0.25	128	60%
30	7	640	0.25	128	60%
30	9	64	0.25	128	46%
30	9	128	0.25	128	59%
30	9	192	0.25	128	61%
30	9	256	0.25	128	58%
30	9	320	0.25	128	74%

30	9	384	0.25	128	72%
30	9	448	0.25	128	73%
30	9	512	0.25	128	49%
30	9	576	0.25	128	89%
30	9	640	0.25	128	54%

Setelah didapatkan akurasi yang optimal, maka pada deteksi parameter dengan metode CNN, dilakukan percobaan perubahan beberapa parameter seperti kernel dan *filter* dengan nilai *dropout* dan *hidden dims* yang tetap. *Dropout* merupakan jumlah filter yang dibuang secara random pada proses klasifikasi sedangkan *hidden dims* adalah jumlah *hidden layers* pada proses klasifikasi. Pada kernel 1 dan 64 filter dengan *dropout* dan *hidden dims* yang tetap, didapatkan akurasi sebesar 56%. Hal ini tentunya belum menunjukkan akurasi yang optimal sehingga dilakukan percobaan dengan merubah kernel dan filter seperti yang ditunjukkan pada Tabel 4.2. terdapat penurunan akurasi pada perubahan filter dari 128 ke 192 lalu ke 256 filter sebanyak 76% dan 59%. hal ini dapat disebabkan karena ketidakefektifan jumlah filter, dimana pada kondisi filter di titik tersebut belum mencapai akurasi yang optimal/ kondisi tertinggi. Setelah itu akurasi meningkat pada filter 384 dengan akurasi sebesar 77% akan tetapi turun kembali pada 448 filter dikarenakan masih belum optimalnya jumlah filter. jumlah filter yang teroptimal pada kernel satu adalah 640 dengan akurasi yang didapat sebesar 84%.

Dilakukan Kembali perubahan jumlah kernel guna mendapatkan akurasi paling optimal. Maka selanjutnya dilakukan perubahan kernel dengan pengujian kernel 3 dan diuji dengan angka filter yang sama dengan kernel 1. Hasilnya terjadi penurunan akurasi pada filter 128, 384 dan 576. Jumlah filter paling optimal pada kernel 3 adalah 64 dan 256 dengan menghasilkan akurasi sebesar 96%, dibandingkan dengan filter 320 dan 512 yang belum mencapai akurasi yang optimal. Hal ini menunjukkan besaran filter tidak mempengaruhi besaran akurasi pada proses klasifikasi dengan CNN. Dikarenakan akurasi tidak meningkat dengan signifikan ketika besaran filter atau jumlah filter yang lebih besar, tetapi akurasi yang optimal ada pada beberapa filter tertentu.

Kemudian dilakukan perubahan kernel sebanyak 5 dengan parameter filter yang masih sama dengan pengujian filter sebelumnya. akurasi menurun pada filter sejumlah 384 dan naik kembali sampai filter 576. jumlah filter yang paling optimal pada kernel 5 adalah 192 filter dengan akurasi sebesar 84%. hal ini juga membuktikan filter kecil dapat menghasilkan akurasi yang tinggi pada klasifikasi CNN. Angka kernel selanjutnya yaitu 7. pada kernel 7, akurasi menurun pada filter 64,192 dan naik kembali sampai filter 512. filter paling optimal pada kernel 7 adalah filter 320 menghasilkan akurasi sebesar 82%. Angka kernel terakhir yang diuji adalah sejumlah 9 kernel. pada jumlah kernel ini, akurasi terus menaik dan berada diposisi terendah pada filter 512 dan akurasi tertinggi pada kernel 9 terletak pada jumlah filter sebanyak 576 filter dengan akurasi sebesar 89%.

Dari scenario 2 dapat disimpulkan bahwa jumlah kernel dan filter optimal yang menghasilkan akurasi paling tinggi adalah sebanyak 3 kernel dan 64 serta 256 filter dengan jumlah *dropout* sebanyak 0.2 dan *hidden dims* sebanyak 128 menghasilkan akurasi sebesar 96% juga dicapai dengan *epoch* 30. Kemudian akurasi terendah terdapat pada kernel 9 dengan jumlah filter 64 menghasilkan akurasi sebesar 42%. Maka pada klasifikasi menggunakan CNN ketika jumlah filter meningkat tidak diikuti dengan akurasi yang semakin meningkat tetapi akurasi yang optimal ada pada beberapa kondisi filter tertentu. Begitupun dengan jumlah kernel yang digunakan, akurasi yang optimal ada pada kondisi kernel tertentu.

### 4.3 Skenario 3

Skenario 3 merupakan analisis hasil pengujian aplikasi pendeteksi lesi pra kanker pada tes IVA berdasarkan parameter – parameter dan skema pengujian terkait.

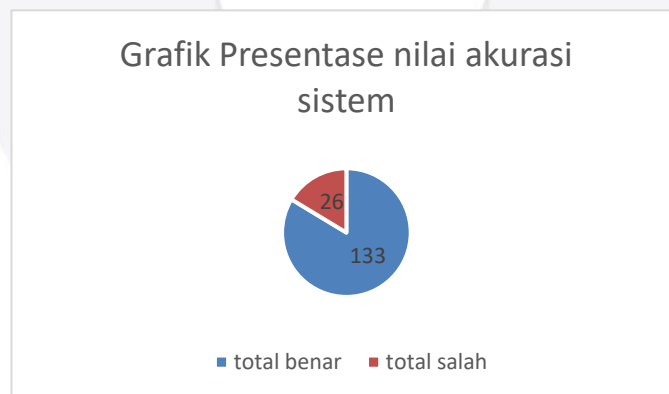
#### 1. Pengujian pada Citra Asli

Berikut merupakan hasil dari pengujian pada citra uji:

Tabel 4.3 Data Hasil Uji pada Citra Asli

No.	Citra	keterangan		Keterangan
		IVA Tutor	Hasil Sistem	
1	data positif 1	IVA positif	IVA positif	(v) benar
2	data positif 2	IVA positif	IVA negatif	(x) salah
3	data positif 3	IVA positif	IVA positif	(v) benar
4	data positif 4	IVA positif	IVA negatif	(x) salah
5	data positif 5	IVA positif	IVA positif	(v) benar
6	data positif 6	IVA positif	IVA positif	(v) benar
7	data positif 7	IVA positif	IVA positif	(v) benar
8	data positif 8	IVA positif	IVA positif	(v) benar
9	data positif 9	IVA positif	IVA positif	(v) benar
10	data positif 10	IVA positif	IVA positif	(v) benar

Setelah melakukan pengujian terhadap 159 citra uji, didapatkan hasil berupa 133 citra benar. Dari data tersebut didapatkan nilai akurasi sebesar  $133/159 \times 100\% = 83,64\%$ . Berikut grafik dari data yang telah diperoleh dapat dilihat pada gambar 4.1:



Gambar 4.1 Grafik presentase nilai akurasi sitem

Dari 159 data (82 IVA negatif dan 77 IVA positif) didapatkan hasil: 68 citra positif benar, 9 positif palsu, 65 citra negatif benar, dan 17 negatif palsu. Sehingga dapat dihitung sensitivitas dan spesifisitasnya. Berikut nilai sensitivitas dan spesifisitas yang diperoleh:

$$\text{Sensitivitas (\%)} = \frac{68}{68+17} \times 100\% = 80\%$$

$$\text{Spesifisitas (\%)} = \frac{65}{65+9} \times 100\% = 87,83\%$$

#### 4.4 Analisis umum

Pada Skenario I, II, dan III didapatkan hasil performansi dari sistem.pada skenario 1 pengujian akurasi berdasarkan epoch menghasilkan akurasi yang optimal pada saat epoch 30 berada pada akurasi 96%. Kemudian pada epoch 30 karna sudah menghasilkan akurasi yang optimal maka menjadi parameter tetap pada skenario ke-2 saat pengujian klasifikasi CNN dengan parameter kernel dan filter. Kemudian pada skenario ke-3 di dapatkan akurasi pada pengujian citra uji sebesar 83,64% dapat dikatakan sudah cukup baik dalam perancangan sistem ini. Hasil dari performansi tersebut juga dipengaruhi dari banyaknya data yang diolah,semakin banyak data masukan yang diolah maka akan semakin tinggi nilai akurasi dan nilai akurasi rendah didapatkan karena jumlah data yang terbatas dan tipe citra yang bervariasi. Hal tersebut menandakan, bahwa jumlah data dan tipe citra yang bervariasi berpengaruh pada tingkat akurasi dan berpengaruh pada hasil peromansi sistem aplikasi.

### 5. Kesimpulan dan Saran

#### 5.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada sistem program pendeteksi Lesi Pra Kanker Serviks pada Test IVA, dapat diambil kesimpulan sebagai berikut:

1. Tugas akhir ini berhasil mendeteksi pola lesi putih dalam pemeriksaan IVA melalui citra digital dengan metode klasifikasi *Convolutional Neural Network* dengan optimal akurasi sistem maksimum pada epoch 30 sebesar 96% dengan kernel 3 serta pada filter 64 dan 256.
2. Tugas akhir ini berhasil merancang aplikasi android deteksi kanker serviks menggunakan python dan dapat mengklasifikasikan citra kanker serviks hasil test IVA menjadi positif dan negatif dibuktikan pada Sensitivitas dan spesifisitas sistem optimal pada rancangan aplikasi ini menghasilkan sebesar 80% dan 87,83%.
3. Tingkat akurasi sistem yang didapatkan dari jumlah data uji benar pada data uji menghasilkan sebesar 84%. maka dengan demikian menunjukkan bahwa sistem pada rancangan aplikasi ini berjalan dengan cukup baik karna akurasi pengujian sistem, termasuk sensitivitas dan juga spesifisitas yang sudah mencapai nilai diatas 80%.

#### 5.2 Saran

Saran untuk pengembangan aplikasi ini adalah:

1. Diharapkan pada penelitian selanjutnya terkumpul variasi data dan jumlah data yang lebih banyak, agar akurasi sistem lebih maksimal lagi dikarenakan proses training data membutuhkan banyak data sebagai inputnya.
2. Diharapkan adanya pengembangan ke platform lain seperti i-phone, windows phone dan terintegrasi dengan sistem *database*. Kemudian diharapkan untuk pengembangan aplikasi menggunakan metode lainnya untuk akurasi yang maksimal.

### REFERENSI

- [1] N. Anggraeni and M. Muhartati, "the correlation between cervix cancer knowledge level and wus behavior in performing iva examination at banguntapan 1 primary health center of bantul in 2015 1" vol.1 no.1 ,2015.
- [2] Kurnia Suci Nurani, Sri Ratnaningsih "hubungan pengetahuan wanita usia subur tentang kanker serviks dengan keikutsertaan iva test di puskesmas umbur harjo II yogyakarta naskah publikasi"jurnal./ vol.1 no.1, 2017.
- [3] D. S. Bestry, B. Hidayat, and H. Fauzi, "Deteksi Dini Kanker Serviks Metode Iva Berbasis Pengolahan Sinyal Digital Menggunakan Deteksi Tepi Zero Crossin," *e-Proceeding Eng.*, vol. 3, no. 3, pp. 4816–4822, 2016.
- [4] C. Arisusilo, "Kanker Leher Rahim (Cancer Cervix) Sebagai Pembunuh Wanita Terbanyak Di Negara Berkembang," *Sainstis*, pp. 112–123, 2012, doi: 10.18860/sains.v0i0.1862.
- [5] Nila Anggriyani "klasifikasi kanker serviks menggunakan jaringan syaraf tiruan *backropagation* dengan GUI."sains.,vol.1,2015
- [6] S. Wiyono and T. Mirza Iskandar, "Media Medika Indonesiana Inspeksi Visual Asam Asetat (IVA) untuk Deteksi Dini Lesi Prakanker Serviks," vol 43, no 03, 2008.
- [7] N. B. Tsani and A. Rachman, "Implementasi Deteksi Tepi Canny Dengan Transformasi Powerlaw Dalam Mendeteksi Stadium Kanker Serviks," *J. Ilm. INTECH (Information Technol. Journal) UMUS*, vol. 01, no. 01, pp. 22–33, 2019.
- [8] O. N. Shpakov and G. V. Bogomolov, "Technogenic activity of man and local sources of environmental pollution," *Stud. Environ. Sci.*, vol. 17, no. C, pp. 329–332, 1981, doi: 10.1016/S0166-1116(08)71924-1.
- [9] R. J. Sari *et al.*, "Identifikasi Obyek Bergerak Berbasis Segmentasi Warna RGB," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 1, 2018.
- [10] A. Tri Fadhila, "Simki-Techsain Vol. 01 No. 01 Tahun 2017 " *Simki-Techsin*, vol. 01, no. 01, pp. 1–7, 2017.
- [11] Halim Kurniawan, Budi Setiyono, R. Rizal Isnanto, "Aplikasi Penjawab Pesan Singkat Otomatis Dengan Bahasa Python", 2011.
- [12] O. H. Alliance, "Android," 2010.
- [13] A. Developers, "What is android?" developer. android. com/.../what-isandroid. html. Diunduh tanggal, vol. 14, 2011.



