

DETEKSI PENYAKIT PTERIGIUM MENGGUNAKAN FORWARD CHAINING DAN ALGORITMA VIOLA JONES

DETECTION OF PTERIGIUM DISEASE USING FORWARD CHAINING AND VIOLA JONES ALGORITHM

Umu Hanifah¹, Ir. Purba Daru Kusuma S.T., M.T² Casi Setianingsih, S.T., M.T..³

^{1,2,3}Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹umuhnfh@student.telkomuniversity.ac.id, ²purbadaru@telkomuniversity.ac.id, ³setiacasie@telkomuniversity.ac.id

Abstrak

Mata merupakan salah satu panca indra yang paling penting bagi kehidupan manusia. Karena dengan mata, kita dapat melihat dan mengetahui situasi dan kondisi yang terjadi di sekitar kita. Jika terdapat masalah atau kelainan yang terjadi pada mata kita, maka kita akan merasakan hal yang tidak nyaman. Mata mempunyai beberapa penyakit membahayakan yang mengakibatkan menurunnya kualitas penglihatan serta bisa menyebabkan kebutaan.

Pada penelitian ini akan dibuat sebuah aplikasi khusus untuk mendeteksi penyakit mata *Pterigium* berdasarkan adanya gejala-gejala dini yang telah dirasakan oleh pasien dan mengetahui seberapa parah pasien terkena penyakit *Pterigium* dengan tingkatan yang berbeda-beda. Tahapan yang digunakan untuk mengetahui tingkatan penyakit *Pterigium* yaitu dengan cara mengisi semua gejala-gejala yang dirasakan oleh pasien di dalam aplikasi menggunakan metode *Forward Chaining* serta dilakukan proses segmentasi citra dengan menggunakan *Algoritma Viola Jones*.

Kata Kunci : Pterigium, Algoritma Viola Jones, Forward Chaining

Abstract

Eye is one of the most important senses for human life. Because with the eyes, we can see and know the situations and conditions that occur around us. If there are problems or disorders that happen in our eyes, then we will feel uncomfortable. The eye has several diseases that can reduce the quality of vision and can cause blindness.

In this Final Project will be make an application to detect Pterygium eye disease based on the early symptoms that have been felt by the patient and find out how severely the patient affected by Pterygium disease with different levels. The stages used to determine the level of Pterygium disease is by filling all the symptoms by the patient in the application using Forward Chaining method and using an image segmentation process with Viola Jones Algorithm.

Keyword : Pterygium, Algoritma Viola Jones, Forward Chaining.

1. Pendahuluan

Seiring dengan berkurangnya kualitas dan gaya hidup seperti pola makan, olahraga, bekerja, tingkat stress dan usia, jumlah individu dengan keluhan penyakit mata semakin bertambah. Perbandingan jumlah medis yang sedikit menyebabkan kurangnya masyarakat memahami penyakit yang diderita. Saat ini, ada sekitar 45 juta orang buta diseluruh dunia dengan peningkatan 1-2 juta setiap tahun dari seluruh populasi di dunia. Populasi dunia diperkirakan tumbuh dari 6 hingga 7,9 miliar pada tahun 2020, dengan 6 miliar orang di Afrika dan Asia saja[1].

Indonesia menempati posisi tertinggi kedua setelah Ethiopia untuk angka kebutaan yang terjadi di dunia, dilaporkan pada pertemuan Asia Pasific Academy of Ophthalmology di Sydney 2010. Dengan angka kebutaan Indonesia yang diatas 1% menjadikan angka kebutaan di Indonesia tidak hanya menjadi masalah kesehatan, tetapi sudah menjadi masalah sosial[2].

Menurut dr.Johan Hutaaruk, spesialis mata dari Jakarta Eye Center menekankan terdapat lima penyakit mata yang sering dialami oleh masyarakat Indonesia, dan salah satu dari lima penyakit tersebut terdapat penyakit mata yang disebut Pterigium. Pterigium sendiri adalah penyakit dengan kondisi mata yang ditandai oleh tumbuhnya selaput yang menutupi bagian berwarna putih pada bola mata. Prevalensi Pterigium sesuai dengan Panduan Manajemen Klinis Perdami, insidens Pterigium ini cukup tinggi di Indonesia, pada daerah equator mencapai hingga 13,1%. [3].

Dari latar belakang diatas penulis membuat suatu sistem yang berguna untuk mendeteksi penyakit pterygium pada mata seseorang yang akan dilakukan melalui proses pengambilan keputusan dengan menggunakan sistem pakar yang diambil oleh seorang atau beberapa pakar dengan metode *Forward Chaining* dan menerapkan pengolahan citra digital dengan *Algoritma Viola-Jones*. Dengan adanya aplikasi ini diharapkan dapat memudahkan para dokter pada fasilitas kesehatan tingkat 1 untuk dapat mengetahui lebih lanjut mengenai tingkat penyakit Pterigium yang diderita oleh para pasien.

2. Dasar Teori

2.1 Pterigium

Pterigium berasal dari bahasa Yunani yaitu *Pterigion* yang berarti sayap kecil (*small wing*) merupakan suatu kelainan pada permukaan mata yang ditandai oleh perkembangan jaringan fibrovascular dari konjungtiva yang terdapat dicelah kelopak mata berbentuk segitiga di mana puncaknya mengarah kebagian tengah dari kornea. Jaringan pterigium yang berkembang ke daerah pupil dapat menyebabkan penglihatan terganggu dan memberi perasaan tidak nyaman serta penyakit ini dapat mengenai kedua mata dengan derajat pertumbuhannya yang berbeda[4].

Hasil survei didapatkan dari Departemen Kesehatan Republik Indonesia tahun 1993-1996, angka kejadian pterigium sebesar 13,9% dan menempati urutan kedua penyakit mata. Di Sulawesi Selatan pterigium menduduki peringkat kedua dari sepuluh macam penyakit utama dengan insiden sekitar 8,2% [5].

Pada tahap awal penderita penyakit ini sering merasakan panas di daerah mata dan menganjat, lalu mata sering memerah. Pada proses ini dokter melakukan pendekatan konservatif dengan menyarankan pasien untuk menggunakan kacamata yang tepat agar bisa melindungi mata dari radiasi sinar UV. Pada pterigium tingkat lanjut akan mengakibatkan penderita menjadi kurang nyaman karena biasanya pada proses ini terjadi pertumbuhan yang semakin membesar serta mengarah ke daerah kornea, sehingga bisa menutup kornea sampai ke pupil. Jika sampai menutup pupil maka penglihatan akan terganggu bahkan bisa menyebabkan kebutaan. Pada saat itu tindakan lanjut harus dilakukan dengan cara pembedahan mata[2].

2.2 Algoritma Viola Jones

Metode Algoritma *Viola Jones* adalah metode yang paling banyak digunakan untuk pengolahan objek karena metode ini memiliki algoritma yang efisien dan tingkat keakuratannya cukup tinggi dibandingkan dengan metode deteksi objek lainnya. Metode ini, diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001. Terdapat empat kunci utama di dalam metode pendeteksian objek *Viola Jones*, yaitu:

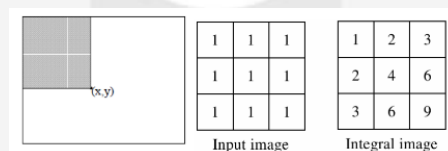
1. Fitur segi empat sederhana disebut sebagai fitur *Haar*.
2. *Integral image* (citra *integral*) untuk pendeteksian fitur secara cepat.
3. Metode *machine learning AdaBoost* (algoritma *boosting*).
4. *Cascade Classifier* (Klasifier bertingkat) untuk menghubungkan dan mengelompokkan banyak fitur secara efisien.

2.3 Haar Like Feature

Haar-like feature memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa pixel. Perkotak itu pun kemudian diproses dan didapatkan perbedaan nilai (*threshold*) yang menandakan daerah gelap dan terang dengan cara mengurangi nilai piksel pada daerah berwarna putih dengan nilai piksel yang berada pada daerah hitam. Algoritma ini membagi gambar berdasarkan nilai fitur-fitur yang dideteksi. Terdapat empat fitur pada *haar cascade*. Terlihat pada gambar bahwa algoritma ini memiliki dua fitur yang terdiri dari dua persegi panjang (A dan B), satu fitur dengan tiga persegi panjang (C), dan satu fitur dengan empat persegi panjang (D). Fitur-fitur ini diciptakan oleh Viola dan Jones dengan memodifikasi metode *Haar wavelets*, sehingga fitur-fitur tersebut dinamakan *Haar-like features*[6].

2.4 Integral Image

Untuk mempermudah proses penghitungan nilai fitur, algoritma *Viola-Jones* menggunakan sebuah media berupa *Integral Image*. *Integral Image* digunakan sebagai cara yang cepat dan efektif untuk menghitung sebuah citra yang nilai tiap pikselnya merupakan penjumlahan dari nilai piksel kiri atas hingga kanan bawah. Contoh citra *integral* dapat dilihat pada **Gambar 2.2**:



Gambar 2.1 : Contoh perhitungan citra integral

Dapat dilihat pada gambar diatas perhitungan jumlah dari *pixel* dalam persegi, dapat dihitung dengan cara menambahkan nilai *pixel* pada setiap persegi. Tiap nilai pada citra *integral* sudah ditambahkan dengan nilai *pixel* persegi sebelumnya. Proses pencarian nilai fitur ini dilakukan secara iteratif mulai dari ujung kiri atas *image* hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy . Semakin kecil nilai Δx dan Δy , maka semakin akurat pula proses deteksi[7]. Dari nilai-nilai *pixel* yang didapatkan pada fitur tersebut, maka akan dihitung nilai *integral image* pada fitur tersebut dengan rumus:

$$s(x,y) = i(x,y) + s(x,y) + s(x,y-1) + s(x-1,y) - s(x-1,y-1) \dots\dots\dots (1)$$

2.5 Algoritma Boosting

Untuk memilih fitur yang spesifik dan mengatur nilai ambangnya (*threshold*), *Viola Jones* menggunakan sebuah metode *machine learning* yang disebut *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* lemah untuk membuat

sebuah classifier kuat. *AdaBoost* memilih sejumlah classifier lemah untuk disatukan dan menambahkan bobot pada setiap classifier, sehingga akan menjadi classifier yang kuat. *Viola-Jones* menggabungkan beberapa *AdaBoost* classifier sebagai rangkaian filter yang cukup efisien untuk menggolongkan daerah image[8].

Dikatakan *boosting* karena algoritma ini dapat mengurangi kesalahan pada *weak classifier* dan meningkatkan akurasi dari setiap algoritma pembelajaran yang diberikan. Pada dasarnya *Adaboost* digunakan untuk mampu mengurangi kesalahan pada proses pembelajaran algoritma tersebut. Algoritma ini disusun menggunakan persamaan[9]:

1. Normalisasikan bobot,

$$W_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \dots \dots \dots (2)$$

2. Untuk setiap fitur j latih sebuah classifier h_j yang dibatasi agar menggunakan sebuah fitur tunggal. Kesalahan dievaluasi sehubungan dengan

$$W_{t,\epsilon_j} = \sum_{i=1}^n w_i |h_j(x_i) - y_i| \dots \dots \dots (3)$$

3. Pilih classifier, h_t dengan kesalahan terendah ϵ_t .

4. Perbarui bobot:

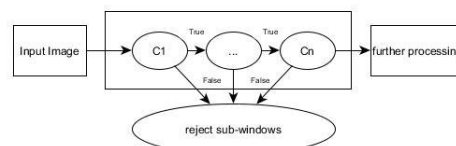
$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \dots \dots \dots (4)$$

Dimana $e_i = 0$ jika data x_i diklasifikasikan dengan benar, $e_i = 1$ jika sebaliknya dan $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

2.6 Cascade Classifier

Kombinasi *Cascade of Classifier* merupakan tahap terakhir dalam metode *Viola Jones*. Metode ini merupakan sebuah cara untuk mendeteksi objek berdasarkan klasifikasi, dimana kumpulan fungsi klasifikasi disusun secara bertahap untuk mengklasifikasi sebuah citra atau potongan citra sebagai objek atau non objek. Dengan cara ini, kecepatan pada proses pendeteksian dapat meningkat yaitu dengan cara memusatkan perhatian pada daerah-daerah dalam *image* yang berpeluang saja. Hal ini dilakukan untuk menentukan di mana letak objek yang dicari pada suatu *image*.

Karakteristik dari algoritma *Viola-Jones* adalah adanya klasifikasi bertingkat. Klasifikasi pada algoritma ini terdiri dari tiga tingkatan di mana tiap tingkatan mengeluarkan *subimage* yang diyakini bukan objek. Hal ini dilakukan karena lebih mudah untuk menilai *subimage* tersebut bukan objek yang ingin dideteksi ketimbang menilai apakah *subimage* tersebut merupakan objek yang ingin dideteksi. Di bawah ini adalah alur kerja dari klasifikasi bertingkat.



Gambar 2.2: Proses Cascade Classifier

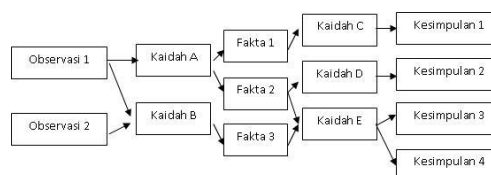
2.7 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar[10]. Pakar adalah seorang yang memiliki pengetahuan khusus, pengalaman, serta metode yang dapat digunakan sebagai acuan dalam menyelesaikan suatu permasalahan. Sistem pakar ini juga dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai asisten yang berpengalaman serta mempunyai pengetahuan yang dibutuhkan[10].

2.8 Forward Chaining

Metode *forward chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Merupakan pemberi alasan dari fakta untuk kesimpulan hasil dari fakta. Solusi untuk beberapa masalah secara alami dimulai dengan pengumpulan informasi. Penalaran diterapkan untuk untuk memperoleh kesimpulan yang logis dari informasi yang didapat[12].

Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan konklusi[12].



Gambar 2.3: Diagram Forward Chaining

2.9 Certainty Factor

Certainty Factor (CF) merupakan salah satu teknik yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan. Metode ini mirip dengan fuzzy logic, karena ketidakpastian direpresentasikan dengan derajat kepercayaan sedangkan perbedaannya adalah pada *fuzzy logic* saat perhitungan untuk rule yang premisnya lebih dari satu, *fuzzy logic* tidak memiliki nilai keyakinan untuk rule tersebut sehingga perhitungannya hanya melihat nilai terkecil untuk operator AND atau nilai terbesar untuk operator OR dari setiap premis yang pada rule tersebut berbeda dengan certainty factor yaitu setiap rule memiliki nilai keyakinannya sendiri tidak hanya premis-premisnya saja yang memiliki nilai keyakinan[14].

$$CF[h,e] = MB[h,e] - MD[h,e] \dots\dots\dots (5)$$

Keterangan :

CF[h,e] = Faktor kepastian

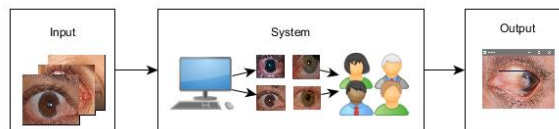
MB[h,e] = Measure of belief, ukuran kepercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1

MD[h,e] = Measure of disbelief, ukuran ketidakpercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1

3. Pembahasan

3.1 Gambaran Umum Sistem

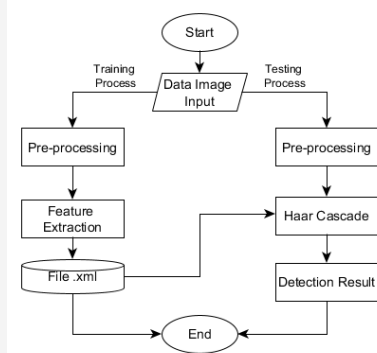
Pada penelitian ini akan dibuat sebuah aplikasi berbasis web yang dapat mendeteksi penyakit Pterigium pada mata pasien menggunakan metode *Viola Jones* dan metode Sistem Pakar. Aplikasi ini menggunakan bahasa pemrograman Python yang telah diprogram untuk mengetahui tingkat kerusakan mata akibat terserang penyakit Pterigium. Aplikasi ini menggunakan citra untuk menjalankan program dan sebagai *input* untuk pengklasifikasian seberapa parah tingkat kerusakan yang diderita oleh user dengan menggunakan metode *Forward Chaining*.



Gambar 3.1 : Skema Diagram Pembuatan Aplikasi

3.2 Perancangan Sistem

Data yang akan digunakan dalam tugas akhir ini adalah data yang diambil dari internet yang telah diverifikasi oleh Pakar. Total data sebanyak 113 data Berikut adalah flowchart pengolahan citra pada sistem:



Gambar 3.2 : Flowchart Pengolahan Citra

Gambar 3.2 menjelaskan tentang proses pengolahan citra pada system yang telah dibuat. Mulai dari pengguna yang akan memasukan citra yang sudah disediakan pada galeri, lalu citra masuk ke tahap pre-preprocessing dan pada akhirnya sistem akan melakukan ekstraksi fitur pada citra yang telah dipilih dengan menggunakan fitur cascade yang telah dibuat sebelumnya. Hasil akhir yang akan dikeluarkan adalah gambar citra yang terdeteksi penyakit pterigium.

3.3 Perancangan

Tahapan ini adalah tahapan perancangan sistem pada aplikasi yang telah dibuat. Perancangan antara lain meliputi sistem deteksi, penyusunan tabel keputusan dan pohon keputusan.



Gambar 3.3 : Flowchart Sistem Pakar

Pada **gambar 3.3** diatas pengguna akan memulai dengan masuk pada aplikasi dan aplikasi langsung memberikan menu select image untuk menentukan gambar mana yang akan diujikan pada sistem ini. Lalu pengguna membuka galeri gambar dan melakukan pemilihan citra. Lalu citra tersebut diproses melalui pre-processing dan melalui tahap ekstraksi fitur. Lalu setelah semua telah diproses pengguna harus menjawab semua gejala yang telah disediakan dan sistem akan mengkalkulasi berdasarkan peraturang yang telah dibuat. Hasil akhir yang didapatkan adalah deteksi pterigium pada citra berdasarkan tingkat keparahannya.

Pre-processing Citra

➤ Ekstraksi Fitur

Langkah selanjutnya setelah dilakukan pre-processing pada gambar adalah ekstraksi fitur. Dimana dari ekstraksi fitur ini sistem dapat mendeteksi objek yang diinginkan. Pada ekstraksi fitur ini terdapat beberapa langkah yang harus dilakukan, antara lain:

- Tahap Training Data

Pada proses *Haar like feature* atau yang dikenal pada *OpenCV* dengan *Haarcascade* ini membutuhkan data training berupa gambar positif dan gambar negatif. Terdapat beberapa tahap yang akan dilakukan pada proses training, berikut adalah flowchart yang menggambarkan proses training:



Gambar 3.4: Flowchart Tahap Training

Penjelasan alur dapat dijelaskan sebagai berikut :

- a. Persiapan Data *Training*
 - Mengumpulkan 113 citra positif yang berukuran 300x200 pixel yang berisikan objek yang akan dideteksi. Beberapa sampel ditunjukkan pada gambar 3.3(a).
 - Mengumpulkan 587 citra negatif yang berukuran 320x240 pixel berisikan citra latar belakang dimana objek yang diinginkan tidak ditemukan dalam kumpulan citra tersebut. Beberapa sampel ditunjukkan pada gambar 3.3(b).
- b. Membuat Sampel Positif
 - Mengambil sampel citra mata yang terkena penyakit pterigium sebesar 54 dari total keseluruhan sebanyak 113 citra dan melakukan proses pemilihan objek yang ingin dideteksi dengan menggunakan *object_marker.py* secara manual lalu data gambar dimasukkan ke dalam file positif.txt.
 - Lalu proses selanjutnya menggunakan bantuan *library OpenCV* yaitu *opencv_createsample.exe* pada terminal untuk melihat hasil objek yang sebelumnya telah dilakukan pemilihan secara manual lalu dibentuk menjadi sample. Sample tersebut selanjutnya dikonversi dalam bentuk vector dengan menggunakan bantuan *library OpenCV* yaitu *opencv_createsample.exe*.
- c. Membuat sample negatif

- Mengambil sampel yang tidak berisi objek yang dideteksi sebanyak 587 citra dan data gambar dimasukkan ke dalam file bg.txt
- d. Ekstraksi Fitur
Melakukan proses ekstraksi fitur dengan menggunakan *library OpenCV* yaitu *opencv_traincascade.exe* pada terminal. Pada pembuatan ekstraksi fitur ini menghasilkan keluaran berbentuk file .xml. Berikut adalah proses ekstraksi fitur menggunakan *library OpenCV*
 - e. Membuat file .xml
Setelah melakukan pelatihan data dan kemudian menjadi file berformat .xml yang berfungsi untuk memanggil beberapa fitur haar dalam suatu gambar dan selanjutnya file tersebut akan digunakan untuk proses testing.

➤ Input Gejala

Pada tahap ini setelah pengguna melakukan pengambilan gambar pada galeri, maka selanjutnya memaukan data berupa jawaban dari gejala- gejala penyakit pterigium yang telah disetujui oleh Pakar. Pengguna hanya memilih Ya/Tidak pada aplikasi sesuai dengan apa yang dirasakan mengenai penyakit ini. Proses ini menghasilkan suatu kesimpulan mengenai tingkat keparah yang diderita pasien mengenai penyakit pterigium.

Tabel 3.1 Tabel Gejala

No.	Gejala-gejala	Jawaban
1.	Apakah mata Anda sering terlihat memerah?	YA/TIDAK
2.	Apakah mata Anda sering berair?	YA/TIDAK
3.	Apakah mata Anda peka terhadap cahaya(silau)?	YA/TIDAK
4.	Apakah mata Anda sering langsung terkena sinar radiasi?	YA/TIDAK
5.	Apakah Anda bekerja diluar ruangan	YA/TIDAK
6.	Apakah mata Anda terasa seperti ada benda asing yang masuk?	YA/TIDAK
7.	Apakah penglihatan mata Anda menurun?	YA/TIDAK

4. Implementasi dan Pengujian Sistem

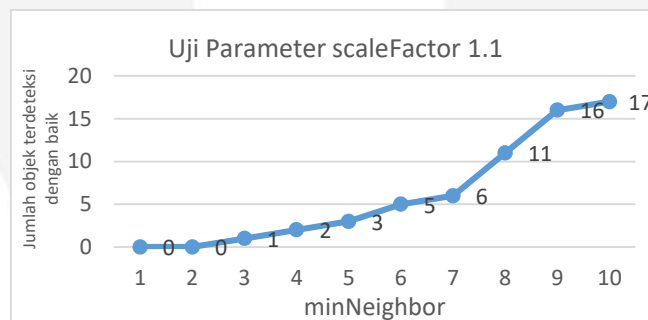
4.1 Pengujian Kinerja Sistem

Pengujian sistem deteksi yang akan dilakukan menggunakan data pengujian 59 dari 113 data trainin dan data testing. Pengujian dilakukan pada beberapa kombinasi nilai dua parameter berbeda untuk sistem deteksi. Parameter-parameter ini adalah nilai faktor untuk memperbesar jendela pemindaian yaitu *scaleFactor* dan jumlah anggota minimum yang harus dimiliki suatu kelompok jendela deteksi positif bertetangaan yaitu *minNeighbor*. Adapun nilai parameter yang digunakan sebagai berikut:

- *scaleFactor* : { 1.1, 1.2, 1.3, 1.4 }
- *minNeighbor* : { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

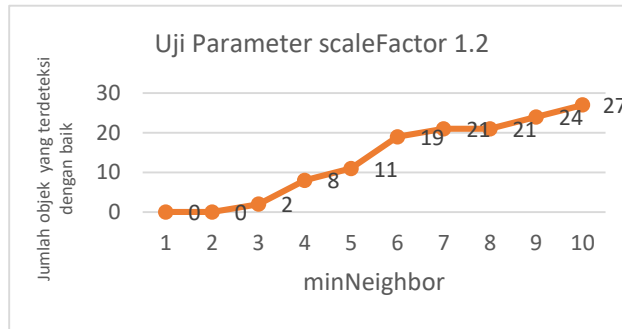
4.2.1 Hasil Pengujian

Berikut adalah hasil pengujian sistem deteksi berdasarkan parameter yang sudah dipilih:



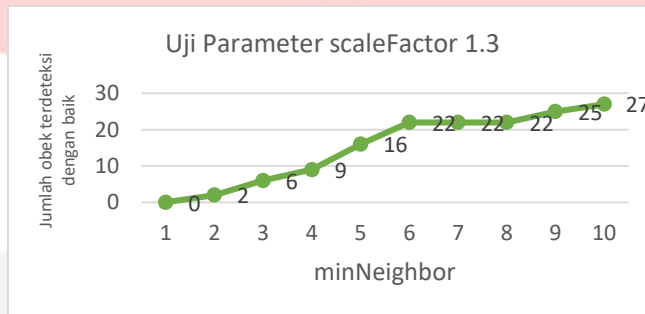
Gambar 4.1: Grafik Hasil Pengujian Parameter scaleFactor 1.1

Pada gambar 4.1 diatas penulis menguji dengan menggunakan parameter scaleFactor terlebih dahulu, dan yang diuji pada parameter 1.1. Hasil pada gambar diatas menunjukan bahwa pada semakin tinggi parameter minNeighbor semakin bagus dan akurat objek yang terdeteksi. Pada parameter scaleFactor 1.1 dan minNeighbor 10 jumlah objek yang terdeteksi sebanyak 17 citra dari seluruh citra yang diujikan sebanyak 50.



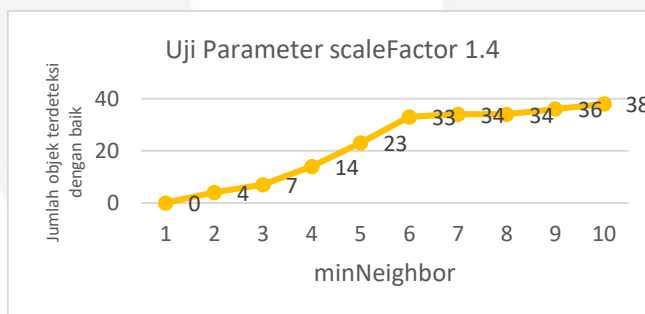
Gambar 4.2: Grafik Hasil Pengujian Parameter scaleFactor 1.2

Pada gambar 4.2 diatas penulis menguji pada parameter 1.2. Hasil pada gambar diatas menunjukan bahwa berbeda dengan scaleFactor 1.1, terdapat beberapa parameter minNeighbor yang membuat jumlah objek yang terdeteksi berkurang. Pada parameter scaleFactor 1.2 dan minNeighbor 10 jumlah objek yang terdeteksi sebanyak 27 citra dari seluruh citra yang diujikan sebanyak 50.



Gambar 4.3: Grafik Hasil Pengujian Parameter scaleFactor 1.3

Pada gambar 4.3 diatas penulis menguji pada parameter 1.3. Hasil pada gambar diatas menunjukan bahwa berbeda dengan scaleFactor 1.1, terdapat beberapa parameter minNeighbor yang membuat jumlah objek yang terdeteksi berkurang. Pada parameter scaleFactor 1.3 dan minNeighbor 10 jumlah objek yang terdeteksi menurun menjadi 27 citra dari seluruh citra yang diujikan sebanyak 50.



Gambar 4.4: Grafik Hasil Pengujian Parameter scaleFactor 1.4

Pada gambar 4.4 diatas penulis menguji pada parameter 1.4. Hasil pada gambar diatas menunjukan bahwa berbeda dengan scaleFactor 1.4, terdapat beberapa parameter minNeighbor yang membuat jumlah objek yang terdeteksi berkurang. Pada parameter scaleFactor 1.4 dan minNeighbor 10 jumlah objek yang terdeteksi sebanyak 38 citra dari seluruh citra yang diujikan sebanyak 50. Dengan menggunakan 10 layer *cascade detector* akan dilatih dengan algoritma *adaboost* berdasarkan gambar sampel positif dan gambar sampel negatif. Jumlah sampel gambar positif adalah 54 gambar dan jumlah sampel gambar negatif adalah 587. Adapun setting parameter lain yang dilakukan pada algoritma *adaboost*, yaitu *boosting = gentle adaboost*, *min hit rate = 0.995*, *max false alarm = 0.5*, *mode = BASIC*, *subwindow size = 48x32*. Hasil yang didapatkan bahwa parameter yang terbaik untuk pengujian sistem deteksi adalah parameter *scaleFactor* 1,4 dan *minNeighbor* 10.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian tugas akhir ini adalah sebagai berikut:

1. Hasil keakuratan terbaik dalam sistem deteksi yang telah dibuat sebanyak 76% dengan parameter scaleFactor dan parameter minNeighbor yang berubah-ubah.
2. Hasil keakuratan pada sistem deteksi untuk proses testing citra yang tidak berpenyakit (mata sehat) sebesar 100% dari banyaknya total 30 citra dengan parameter yang digunakan sama seperti proses testing pada mata berpenyakit pterigium. Ini menandakan bahwa sistem benar hanya mendeteksi selaput pterigium pada citra mata.
3. Pengujian pada sistem pakar sebanyak 100% karena sudah dibuktikan menggunakan perhitungan manual dengan hasil yang sesuai.
4. Hasil survey yang didapatkan kepada 3 orang dokter fasilitas tingkat satu menyetujui bahwa mereka membutuhkan aplikasi pterigium ini.

5.1 Saran

Berdasarkan dari Penelitian Tugas akhir ini maka, penulis menyarankan untuk penelitian selanjutnya adalah sebagai berikut :

1. Ada baiknya penulis berharap adanya penggunaan metode lain dalam penerapan yang akan dilakukan selanjutnya.
2. Diharapkan sistem deteksi pada viola jones ini dapat lebih ditingkatkan lagi dilihat dari sisi terbatasnya citra yang digunakan.

DAFTAR PUSTAKA

- [1] S. West and A. Sommer, "Prevention of blindness and priorities for the future," vol. 79, no. 3, 2001.
- [2] J. Chui, M. T. Coroneo, L. T. Tat, R. Crouch, D. Wakefield, and N. Di Girolamo, "Ophthalmic pterygium: A stem cell disorder with premalignant features," *Am. J. Pathol.*, vol. 178, no. 2, pp. 817–827, 2011.
- [3] Errly, U. A. Mulyani, and D. Susilowati, "Distribusi Dan Karakteristik Pterigium Di Indonesia," *Bul. Penelit. Sist. Kesehatan.*, vol. 14, no. 23, p. 85, 2011.
- [4] S. R. Abdani, W. M. D. W. Zaki, A. Hussain, and A. Mustapha, "An adaptive nonlinear enhancement method using sigmoid function for iris segmentation in pterygium cases," *Proc. - 2015 Int. Electron. Symp. Emerg. Technol. Electron. Information, IES 2015*, pp. 53–57, 2016.
- [5] S. Djajakusli, S. Rukiah, S. Junaedi, and N. Syamsu, "The profile of tear mucin layer and impression cytology in pterygium patient," *J. Oftalmol. Indones.*, vol. 7, no. 4, p. 139, 2010.
- [6] D. Kajaree and R. . Behera, "A Survey on Web Crawler Approaches," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 2, pp. 1302–1309, 2017.
- [7] D. Dan, R. Fitur, and M. Pada, "Citra Wajah Menggunakan Haar Cascade," pp. 298–305, 2015.
- [8] D. Suprianto, "Sistem Pengenalan Wajah Secara Real-Time," *Sist. Pengenalan Wajah Secara Real-Time dengan Adab. Eig. PCA MySQL*, vol. 7, no. 2, pp. 179–184, 2013.
- [9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, p. I-511-I-518, 2001.
- [10] M. Marimin, "Teori dan Aplikasi Sistem Pakar dalam Teknologi Manajerial," no. January 2009, 2007.
- [11] C. F. Tan, L. S. Wahidin, S. N. Khalil, N. Tamaldin, J. Hu, and G. W. M. Rauterberg, "THE APPLICATION OF EXPERT SYSTEM : A REVIEW OF RESEARCH," vol. 11, no. 4, pp. 2448–2453, 2016.
- [12] A. Al-Ajlan, "The Comparison between Forward and Backward Chaining," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 2, pp. 106–113, 2015.