

**SISTEM KLASIFIKASI TINGKAT KEPARAHAN KATARAK**

**MENGGUNAKAN METODE HYBRID SVM DAN**

**HARMONY SEARCH PADA CITRA FUNDUS**

**PROPOSAL SKRIPSI**



**Oleh :**

**Firdausa Yasmin Hermadiputri**

**20081010105**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"**

**JAWA TIMUR**

**2023**

# DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan.....	3
1.4. Manfaat.....	4
1.5. Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terdahulu.....	6
2.2. Katarak.....	9
2.3. Citra Fundus.....	10
2.4. Katarak Insipien - Stadium Satu.....	11
2.5. Katarak Imatur - Stadium Dua.....	12
2.6. Katarak Matur - Stadium Tiga.....	12
2.7. Katarak Hipermetur - Stadium Empat.....	13
2.8. Machine Learning .....	13
2.9. Computer Vision.....	14
2.10. Pengolah Citra Digital.....	14
2.11. Support Vector Machine (SVM).....	15
2.12. Pendekatan OvO - Support Vector Machine (SVM).....	17

2.13. Harmony Search.....	18
BAB III METODOLOGI.....	22
3.1. Deskripsi Studi Kasus.....	22
3.2. Kebutuhan Hardware dan Software.....	22
3.3. Jenis dan Sumber Data.....	23
3.4. Studi Pustaka.....	23
3.5. Tahapan Penelitian.....	23
3.6. Pengumpulan Dataset.....	24
3.7. Labelling Data.....	25
3.8. Pre-Processing Data.....	25
3.9. Feature Extraction.....	26
3.10. Split Data kedalam Train dan Test Data dengan K-Fold Cross Validation .....	29
3.11. Pencarian Vektor Bobot w Optimal Yang Dapat Meminimumkan Atau Menimimalkan Fungsi Tujuan SVM dengan HS.....	30
3.12. Menghitung Hyperplane (Decision Boundary SVM) Berdasarkan atau Dengan Vektor Bobot w Optimal yang Didapatkan Dengan HS.....	32
3.13. Prediksi Kelas (Pelatihan dan Pengujian).....	33
3.14. Evaluasi Performa.....	34
DAFTAR PUSTAKA.....	37

## **DAFTAR GAMBAR**

Gambar 2.1 Perbedaan Mata Normal dengan Mata Katarak.....	9
Gambar 2.2 Katarak Insipien.....	9
Gambar 2.3 Katarak Immatur.....	10
Gambar 2.4 Katarak Hiper matur.....	11
Gambar 2.5 Pengolahan Citra Digital.....	13
Gambar 3.1 Flowchart Alur Penelitian.....	21
Gambar 3.2 Confusion Matrix.....	35

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Katarak adalah kondisi dimana lensa mata menjadi keruh sehingga menghalangi masuknya cahaya ke dalam mata (Kumar Behera et al., 2020). Gejala awal dari penyakit katarak adalah penglihatan penderita menjadi buram, silau, dan gangguan penglihatan warna karena terjadinya koagulasi protein lensa mata yang disebabkan oleh reaksi biokimia dan pembelahan proteolitik cystalins (lensa protein) sehingga lensa menjadi berawan (Vaughan et al., 1990). Katarak dapat diklasifikasikan atau grading berdasarkan tingkat keparahan serta kematangan, yaitu katarak ringan - insipiens, katarak sedang - imatur, dan katarak berat - matur serta hiper matur (Mayer et al., 2013). Meskipun umumnya terkait dalam proses penuaan, katarak juga dapat terjadi pada anak-anak sejak lahir atau muncul setelah cedera mata, peradangan, trauma, dan beberapa penyakit mata lainnya (Abel, 2018).

Berdasarkan data World Health Organization (WHO), katarak adalah penyebab utama gangguan penglihatan terbanyak di dunia dan menjadi salah satu penyebab faktor kebutaan (Abel, 2018). Di Indonesia, angka kebutaan yang diakibatkan oleh katarak menempati peringkat tertinggi di Asia Tenggara mencapai 1.5% dari total 17 juta penduduk dunia (Kompas Cyber Media, 2019). Hal ini disebabkan karena kondisi geografis Indonesia yang merupakan daerah subtropis. Survei kebutaan, yaitu Rapid Assessment of Avoidable Blindness (RAAB) yang diadakan pada 15 wilayah di Indonesia pada tahun 2014-2016, menunjukkan bahwa provinsi Jawa Timur memiliki angka kebutaan tertinggi mencapai 4,4% dengan penyebab utama katarak sebesar 73.8%. Dinas Kesehatan Kabupaten/Kota Jawa Timur secara rutin mendata kasus katarak setiap tahunnya (Wicitra et al., 2023). Faktanya menunjukkan bahwa setiap tahun jumlah penderita katarak semakin bertambah melampaui 17.000 kasus baru (Wicitra et al., 2023).

Diagnosis katarak dapat ditegakkan melalui observasi pada citra fundus mata. Citra fundus mata adalah gambar bagian belakang lensa mata yang

mencakup retina, macula, optic disc, fovea, dan blood vessels(Data Science Center, 2023)(Turbert, 2020). Pengambilan citra fundus dilakukan dengan cara memberikan pantulan cahaya kedalam mata sehingga dapat memperlihatkan bagian mata tersebut. Pantulan cahaya yang diterima mewakili intensitas jumlah gelombang pantulan RGB dan jumlah kuantitas cahaya yang dipantulkan (Abràmoff & Kay, 2013). Terdapat beberapa alat pengambilan citra fundus, antara lain dengan funduscopy, direct ophthalmoscopy, indirect ophtalmoscopy, digital fundus photo, dan fundus autofluorescence. Ilustrasi fundus mata dapat digunakan sebagai indikasi adanya berbagai penyakit mata karena dapat memberikan detail dari area yang terdampak(Kumar Behera et al., 2020)(Abràmoff & Kay, 2013). Namun, observasi pada citra fundus mata dalam rangka penegakan diagnosis katarak memerlukan para ahli optalmologis untuk mengklasifikasikan atau grading tingkat keparahan katarak. Observasi yang dilakukan oleh ahli optalmologi ini dapat dipengaruhi dari pengalaman pribadi mereka, sehingga berpeluang menghasilkan akurasi pemeriksaan yang tidak standar antar ahli optalmologi. Oleh karena itu, kehadiran computer-aided-system dapat berfungsi sebagai piranti dukung ahli optalmologis dalam pemeriksaan katarak melalui citra fundus, yang diharapkan dapat menghasilkan akurasi standar dalam mengklasifikasikan atau grading tingkat keparahan katarak.

Pada penelitian ini diusulkan sistem klasifikasi atau grading penyakit katarak sebagai computer-aided-system yang mendukung ahli optalmologi dalam penegakan diagnosis katarak melalui citra fundus. Sistem klasifikasi katarak yang diusulkan ini mengimplementasikan bidang pengolahan citra digital, pengenalan pola, dan pembelajaran mesin. Metode yang diusulkan pada penelitian ini adalah Support Vector Machine (SVM) dan algoritma Harmony Search. Metode SVM digunakan sebagai metode klasifikasi karena metode ini dapat berkerja dengan baik dan lebih efektif ketika terdapat batas pemisahan yang jelas antara kelas-kelas yang ada, seperti dalam menangani masalah klasifikasi dengan dataset yang kompleks. Algoritma harmony search merupakan algoritma metaheuristik yang menjadi komponen penting dalam penelitian ini. Dalam sistem klasifikasi tingkat keparahan katarak, algoritma harmony search berperan dalam mencapai vektor bobot  $w$  yang optimal dari hyperlane atau decision boundary SVM sehingga dapat mengoptimalkan penimbangan fitur (weighting feature) pada citra

fundus mata. Vektor bobot yang optimal adalah vektor bobot yang meminimumkan atau meminimalkan fungsi tujuan SVM. Pada usulan penelitian ini, multi SVM untuk menyelesaikan kasus klasifikasi multi-class diaplikasikan melalui skema One-Versus-One (OVO). Pendekatan multi SVM diimplementasikan agar memungkinkan sistem dapat mengklasifikasikan berbagai tingkatan keparahan katarak yang mencakup katarak ringan - insipiens, katarak sedang - imatur, dan katarak berat - matur serta hiper matur dengan lebih tepat, efisien, dan efektif.

Melalui pengembangan sistem klasifikasi tingkat keparahan katarak sebagai aided-system menggunakan metode Support Vector Machine (SVM) dan algoritma Harmony Search, diharapkan dapat menciptakan sistem yang dapat mengklasifikasikan atau grading penyakit katarak dengan efektif dan efisien serta sebagai alat bantu bagi para ahli optalmologi dalam mendiagnosa penyakit katarak.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang permasalahan yang telah dijelaskan maka rumusan masalah pada penelitian ini adalah

1. Bagaimana cara mengembangkan sistem yang dapat melakukan klasifikasi penyakit katarak berdasarkan tingkat keparahan dengan menggunakan metode Support Vector Machine (SVM) dan algoritma optimasi Harmony Search?
2. Bagaimana performa Support Vector Machine (SVM) dan algoritma Harmony Search dalam melakukan klasifikasi penyakit katarak berdasarkan tingkat keparahan?

## **1.3. Tujuan**

Adapun tujuan yang ingin dicapai dari penelitian ini adalah

1. Mengembangkan sistem klasifikasi penyakit katarak berdasarkan tingkat keparahan dengan metode Support Vector Machine (SVM) dan algoritma optimasi Harmony Search.
2. Mengetahui dan mengevaluasi performa metode Support Vector Machine (SVM) dan algoritma optimasi Harmony Search dalam melakukan klasifikasi penyakit katarak berdasarkan tingkat keparahan.

#### **1.4. Manfaat.**

Manfaat penelitian ini terdiri dari manfaat teoritis dan manfaat praktis dengan penjabaran sebagai berikut:

##### **1.4.1 Manfaat Teoritis**

1. Mengembangkan Teorema dalam pengolahan citra digital, khususnya yang menggunakan citra fundus mata dalam melakukan grating penyakit katarak dengan menggunakan metode Support Vector Machine (SVM) dan algoritma optimasi Harmony Search. Namun, tidak menutup kemungkinan metode SVM dan Harmony Search untuk melakukan grating pada penyakit mata lainnya.
2. Mengetahui integrasi teknologi yang mengimplementasikan metode Support Vector Machine (SVM) dan algoritma Harmony Search dalam penanganan penyakit mata, khususnya pada penyakit katarak sehingga dapat memberikan landasan dalam pengembangan teknologi lebih lanjut di bidang oftalmologi.

##### **1.4.2 Manfaat Praktis**

1. Mengetahui penerapan metode Support Vector Machine (SVM) dan algoritma optimasi Harmony Search dalam melakukan grating penyakit katarak.
2. Algoritma Harmony Search yang digunakan dapat mengoptimalkan feature weighting pada citra fundus mata sehingga harapannya dapat memberikan efisiensi, keefektifan, dan keakuratan yang tinggi dalam melakukan grating penyakit katarak.

#### **1.5. Batasan Masalah**

Adapun Batasan Masalah dari penelitian ini adalah:

1. Penelitian berfokus pada tingkatan keparahan dan kematangan katarak, meliputi ringan - insipiens, katarak sedang - imatur, dan katarak berat - matur serta hiper matur.
2. Penelitian menggunakan metode Support Vector Machine (SVM) dan algoritma Harmony Search.
3. Data yang digunakan dalam penelitian ini berasal dari Rumah Sakit mata



undaan surabaya sebagai data primer dan Kaggle sebagai data sekunder.

4. Bahasa pemrograman yang digunakan adalah Python.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Terdahulu**

Dalam melakukan penelitian dibutuhkan acuan-acuan atau kajian pustaka sebagai pedoman selama berlangsungnya proses penelitian serta mencegah adanya duplikasi.

Penelitian oleh Zhiqiang Qiao, Qinyan Zhang, Yanyan Dong, dan Ji-Jiang Yang dengan judul “Application of SVM Based on Genetic Algorithm in Classification of Cataract Fundus Images” pada tahun 2017. Penelitian ini membahas tentang katarak yang merupakan penyakit mata yang umumnya menyebabkan kebutaan. Tujuan dari penelitian ini adalah untuk mengembangkan metode klasifikasi katarak kedalam empat kelas berbeda, meliputi mata sehat atau normal, katarak ringan (mild), katarak sedang (moderate), dan katarak parah (severe) serta meningkatkan akurasi dalam melakukan klasifikasi penyakit katarak. Penelitian ini menggunakan dataset berupa citra fundus. Tahapan dari penelitian ini mencakup segmentasi dataset dengan membagi citra menjadi 17 sub-gambar, ekstraksi fitur GLCM dan Wavelet, pemberian bobot optimal dengan menggunakan algoritma genetika, dan klasifikasi citra fundus dengan metode SVM. Hasil dari penelitian menunjukkan bahwa alur dan metode yang dilakukan dapat mengenali dan mengklasifikasi katarak kedalam empat kelas dengan tingkat akurasi sebesar 87.52%. Implementasi segmentasi dan pemberian bobot optimal dalam penelitian ini terbukti dapat meningkatkan akurasi sebesar 2.97%.

Penelitian pada tahun 2019 yang dilakukan oleh Muhammad Attique Khan, M. Ikramullah Lali, Muhammad Sharif, Kashif Javed, Khursheed Aurangzeb, Syed Irtaza Haider, Abdulaziz Saud Altamrah, dan Talha Akram dengan judul “An Optimized Method for Segmentation and Classification of Apple Diseases based on Strong Correlation and Genetic Algorithm based Feature Selection”. Penelitian ini dilakukan karena tanaman yang merupakan bagian besar dari ekonomi dunia terkena

berbagai penyakit yang menyebabkan kerugian besar dalam ekonomi dibidang industri pertanian. Maka dari itu diperlukan adanya sistem yang dapat mendeteksi penyakit pada tanaman apel. Metode yang digunakan akan mengklasifikasikan kedalam 4 tipe penyakit tanaman apel, yaitu daun sehat, blackrot, rust, dan scab. Alur dari penelitian ini meliputi pengumpulan dataset Plant Village, preprocessing dengan metode hybrid 3D box filtering, de-correlation, 3D-Gaussian filter, dan 3D-Media filter, segmentasi spot lesi dengan metode Expectation Maximization (EM), ekstraksi fitur dengan Local Binary Pattern (LBP), optimasi fitur dengan genetic algorithm, dan klasifikasi menggunakan metode One-vs-All Multi-SVM. Dari penelitian yang dilakukan menunjukkan bahwa metode dan alur dan dilakukan dapat mengklasifikasikan penyakit apel kedalam empat kelas dengan akurasi rata-rata sebesar 97.20% dengan waktu 0.6511 detik sehingga menjadi solusi dalam mengurangi ekonomi dalam industri pertanian yang menjanjikan. Selain itu, peneliti juga membandingkan metode M-SVM dengan metode lainnya seperti Complex Tree, Ensemble Boosted Tree, Fine-K-Nearest Neighbors, dan lain-lain yang hasilnya membuktikan bahwa penggunaan M-SVM memberikan akurasi tertinggi.

Penelitian yang dilakukan oleh Liaqat Ali, Iram Wajahat, Noorbakhsh Amiri Golilarz, Fazel Keshtkar, dan Syed Ahmad Chan Bukhari dengan judul “LDA-GA-SVM: improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine”. Penelitian ini berfokus pada pengembangan sistem dalam mendeteksi penyakit Hepatocellular Carcinoma (HCC) yang merupakan jenis kanker hati yang umum di seluruh dunia agar dapat mendeteksi tahap awal penyakit HCC karena dengan begitu peluang penderita HCC dalam bertahan hidup lebih besar. Dataset yang digunakan pada penelitian ini didapatkan dari opensource UCI Machine Learning repository berisikan 165 subjek. Metode yang diusulkan dengan menggabungkan metode Linear Discriminant Analysis (LDA) untuk mengurangi dimensi, Support Vector Machine (SVM) untuk metode klasifikasi, dan Genetic Algorithm (GA) untuk optimasi SVM. Dari penelitian yang dilakukan memberikan hasil akurasi yang tinggi dalam melakukan mendeteksi penyakit HCC yaitu

sebesar 90.30% dengan penggunaan kernel RBF. Penelitian ini juga membandingkan metode-metode lainnya, seperti SVM tunggal (kernel linear dan rbf), Random Forest, Extra Tree, dan Adaboost. Dari perbandingan metode-metode tersebut, terbukti bahwa mengabungkan SVM dengan Genetic Algorithm dapat meningkat akurasi.

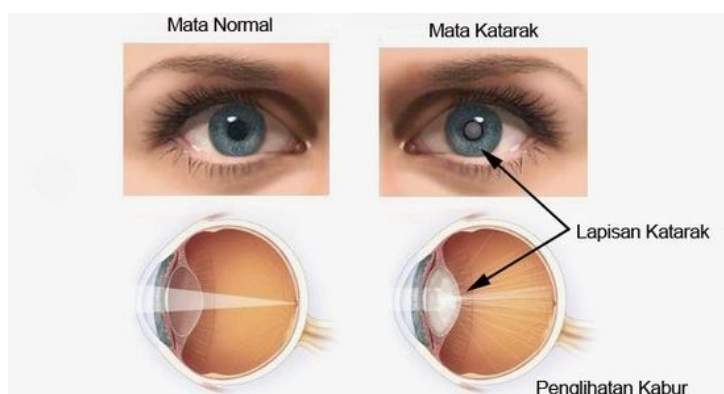
Penelitian dengan judul “Feature Selection Based on Modified Harmony Search Algorithm” yang dilakukan oleh Ani Dijah Rahajoe, Rifki Fahrial Zainal, Budi Mukhamad Mulyo, Boonyang Plangkang, dan Rahmawati Febrifyaning Tias pada tahun 2020 berfokus pada pengembangan metode harmony search untuk seleksi fitur sehingga dapat meningkatkan kinerja dari seleksi fitur, lebih efisien dan akurat, serta mengurangi biaya dan waktu komputasi. Penelitian ini mengusulkan metode harmony search yang diintegrasikan dengan metode Support Vector Machine (SVM). Hasil penelitian menunjukkan bahwa metode yang diusulkan yaitu harmony search setara dengan algoritma genetika untuk SVM tetapi memiliki waktu akses yang lebih cepat sehingga dapat mengurangi biaya dan waktu komputasi dengan akurasi sebesar 96.6%.

Penelitian yang dilakukan oleh A Sumarudin, A. Puspaningrum, dan A. Suheryadi dengan judul “Least-Square Support Vector Machine (LS-SVM) Parameters Optimization using Hybrid Cuckoo Search and Harmony Search Algorithm for Pre Collision Warning on Driver Assistance System (DAS)”. Pertumbuhan teknologi telah mendorong sektor transportasi untuk menyajikan berbagai keunggulan, salah satunya adalah pre collision warning. Penelitian ini fokus pada pengoptimalan parameter LS-SVM menggunakan algoritma hybrid Cuckoo Search dan Harmony Search dalam mendorong sektor keunggulan transportasi yaitu pre collision warning yang dapat mengklasifikasi dan memprediksi tabrakan. Dengan menggunakan dataset berisi 8.437 rekaman transportasi, metode ini berhasil meningkatkan kinerja dengan mencapai tingkat akurasi sebesar 84.513%. Selain itu, metode ini dibandingkan dengan algoritma optimasi metaheuristik lainnya, seperti Cuckoo Search Algorithm (CS-SVM), Bat Algorithm (BA-SVM), dan Firefly Algorithm (FA-SVM), yang menghasilkan hasil eksperimental yang memuaskan.

## 2.2. Katarak

Katarak adalah kondisi dimana lensa mata menjadi keruh sehingga menghalangi masuknya cahaya ke dalam mata yang dapat dilihat pada gambar 1 (Kumar Behera et al., 2020). Penyebab lensa mata menjadi keruh adalah adanya penambahan berat dan penebalan yang terjadi karena modifikasi kimia dan pembelahan proteolitik *cystallins* (lensa protein) sehingga nukleus mata menjadi mengeras (*nuclear sclerosis*) dan menyebabkan munculnya hamburan cahaya serta mengurangi tingkat kejernihan lensa (Vaughan et al., 1990). Gejala-gejala yang dialami antara lain penglihatan kabur yang bersifat perlahan dan layaknya melihat dari kaca yang buram, pengelihatn terasa silau dikarenakan adanya gangguan pembiasan lensa, dan mengalami gangguan diskriminasi warna karena lensa penderita semakin menguning (Abel, 2018).

Berdasarkan data World Health Organization (WHO), katarak menjadi penyebab kebutaan dan gangguan penglihatan terbanyak di dunia (Abel, 2018). Penderita katarak yang rata-rata adalah orang tua dikarenakan penyakit ini berkaitan dengan proses penuaan. Akan tetapi tidak menutup kemungkinan jika anak kecil bahkan bayi tidak akan terkena. Terdapat beberapa faktor yang dapat mempercepat munculnya risiko katarak, antara lain merokok, alkohol, diabetes, sinar ultraviolet, cedera mata, infeksi kehamilan yang dapat menyebabkan bayi mengalami katarak, trauma, dan mengkonsumsi obat-obatan (Abel, 2018). Penyakit katarak dapat dikelompokkan berdasarkan penyebab, umur, letak, dan tingkat stadium. Untuk katarak yang berdasarkan tingkat stadium atau kematangan, katarak dibedakan menjadi stadium satu atau *insipien*, stadium dua atau *imatur*, stadium tiga atau *mature* dan stadium empat atau *hipermatur* (Mayer et al., 2013).



Gambar 2.1 Perbedaan Mata Normal dengan Mata Katarak

### 2.3. Citra Fundus

Citra fundus mata adalah gambaran yang memperlihatkan bagian belakang lensa mata, termasuk struktur seperti retina, macula, diskus optikus (optic disc), fovea, dan pembuluh darah di dalam mata (Data Science Center, 2023; Turbert, 2020). Ini adalah bagian vital dari mata yang memberikan informasi tentang kondisi kesehatan mata seseorang. Pengambilan citra fundus dilakukan dengan memantulkan cahaya ke dalam mata menggunakan alat khusus sehingga memungkinkan visualisasi struktur internal mata (Abramoff & Kay, 2013).

Proses pengambilan citra fundus ini memperoleh pantulan cahaya yang mencerminkan intensitas gelombang pantulan RGB (merah, hijau, dan biru) serta jumlah cahaya yang dipantulkan oleh struktur-struktur dalam mata (Abramoff & Kay, 2013). Dengan menganalisis cahaya yang dipantulkan, dokter mata dapat melihat detail dan kondisi dari berbagai struktur mata secara lebih jelas. Ada beberapa metode yang digunakan untuk mengambil citra fundus, termasuk funduscopy, direct ophthalmoscopy, indirect ophthalmoscopy, digital fundus photo, dan fundus autofluorescence (Abramoff & Kay, 2013). Setiap metode memiliki keunggulan dan kelemahan tersendiri dalam memvisualisasikan berbagai struktur mata.

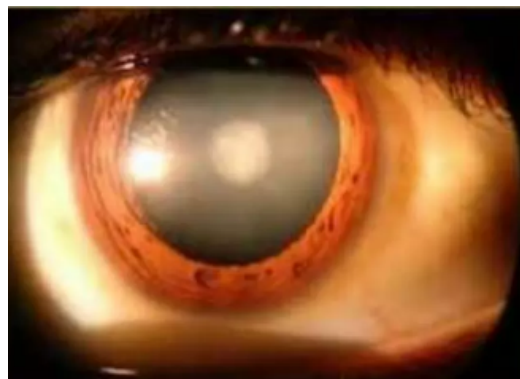
Ilustrasi fundus mata memberikan informasi yang sangat berharga dalam diagnosis penyakit mata. Dengan melihat citra fundus, dokter mata dapat mengidentifikasi tanda-tanda penyakit mata seperti retinopati diabetik, degenerasi makula terkait usia (age-related macular degeneration), glaukoma, dan katarak (Kumar Behera et al., 2020; Abramoff & Kay,

2013). Namun, perlu diingat bahwa untuk diagnosis katarak, observasi citra fundus mata memerlukan evaluasi yang cermat oleh ahli optalmologi.

Katarak adalah kondisi di mana lensa mata menjadi keruh, menghalangi cahaya masuk ke dalam mata. Evaluasi katarak dalam citra fundus bisa memerlukan penggunaan pencahayaan khusus dan penilaian yang terlatih untuk mengklasifikasikan tingkat keparahan katarak. Penting untuk dicatat bahwa interpretasi citra fundus mata dapat bervariasi antara ahli optalmologi karena pengalaman dan interpretasi pribadi mereka. Variabilitas ini dapat memengaruhi akurasi diagnosis dan penentuan rencana pengobatan. Oleh karena itu, penting untuk melakukan interpretasi citra fundus mata oleh ahli optalmologi yang berpengalaman dan terlatih agar diagnosis dan penanganan penyakit mata dapat dilakukan dengan tepat.

#### **2.4. Katarak Insipien - Stadium Satu**

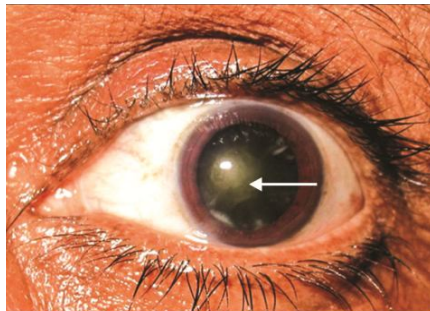
Stadium insipien merupakan fase awal dari katarak atau stadium satu, di mana terjadi kekeruhan lensa sektoral menyerupai spokes of wheel dapat dilihat pada gambar 2 (mirip biji atau roda)(Bachtiar, 2022). Pada tahap ini, degenerasi belum mencapai cairan mata dalam lensa, sehingga bilik mata depan tetap tampak normal dengan kedalaman yang biasa. Iris berada dalam posisi normal dan terdapat kekeruhan ringan pada lensa, tetapi ketajaman penglihatan masih terjaga (Ilyas 2005). Kekeruhan katarak ini dimulai dari tepi ekuator menuju korteks anterior dan posterior. Kondisi ini dapat menyebabkan poliopia (objek yang dilihat penderita tampak ganda) karena adanya perbedaan indeks refraksi di seluruh bagian lensa dalam waktu yang cukup lama (Ilyas dan Yulianti 2011).



Gambar 2.2 Katarak Insipien

## 2.5. Katarak Imatur - Stadium Dua

Katarak immatur pada gambar 3 merupakan katarak stadium atau tingkat kedua dengan kekeruhan lensa mata baru sebagian (Bachtiar, 2022). Pada stadium ini, akan terlihat iris shadow dan visus  $> 1/60$ . Terdapat bagian dari mata penderita yang masih jernih tetapi penderita akan tetap mengalami kesulitan dalam membaca dan membutuhkan cahaya terang. Letak katarak immatur ini berada di tengah lensa mata. Penderita katarak immature disarankan untuk menggunakan kacamata baru dan lensa anti silau.



Gambar 2.3 Katarak Immatur

## 2.6. Katarak Matur - Stadium Tiga

Katarak matur atau katarak matang adalah kondisi dimana lensa mata penderita mengalami kekeruhan total dan pupil berwarna putih susu. Dikarenakan letaknya yang menyebar hingga ke seluruh tepi lensa mata seperti pada gambar 4, penderita semakin sulit dalam membaca bahkan hingga tidak dapat mengerjakan aktivitas atau pekerjaannya. Spesialis dokter mata akan menyarankan kepada penyandang katarak matur untuk segera melakukan operasi pengangkatan katarak karena jika tidak, tekanan pada bola mata akan meningkat (Kenali 4 Stadium Katarak Dan Cara Pencegahannya, 2021).

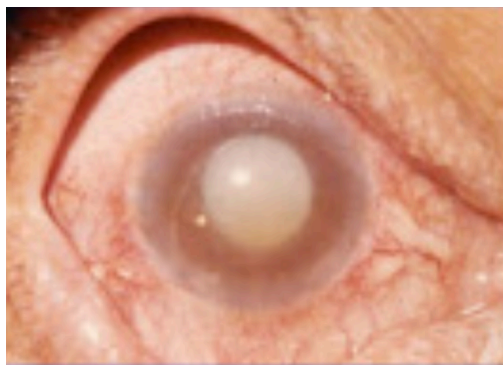


Gambar 2.4 Katarak Matur



## 2.7. Katarak Hiper matur - Stadium Empat

Katarak hiper matur adalah tahap keempat (fase lanjut yang parah), di mana kekeruhan melibatkan seluruh bagian lensa mata dan telah mengalami degenerasi yang signifikan, menjadikannya lembek dan cair (Kenali 4 Stadium Katarak Dan Cara Pencegahannya, 2021). Lensa mata sudah mencair dan telah melewati kapsul lensa (seperti pada gambar 5), yang dapat menyebabkan peradangan pada struktur mata lainnya. Pada tahap ini, kemampuan penglihatan sudah sangat terpengaruh dan bisa berlanjut ke tahap glaukoma atau kebutaan.



Gambar 2.5 Katarak Hiper matur

## 2.8. Machine Learning

Machine Learning (ML) adalah suatu kategori dari sebuah algoritma yang mengizinkan perangkat lunak menjadi lebih akurat dalam memprediksi hasil tanpa diprogram secara explicit (terang-terangan) (Stefanus, 2019). Terdapat beberapa proses mengolah machine learning, yaitu menentukan atau menanyakan sebuah pertanyaan, mengumpulkan data yang dibutuhkan, menyiapkan data, menganalisis data eksplor, membangun model machine learning, mengevaluasi serta mengoptimasi model, mengubah-ubah parameter atau variabel, dan menguji akurasi model (Lateef, 2019).

Machine Learning dapat dibagi menjadi 3 pendekatan yang meliputi Supervised Learning, Unsupervised Learning, dan Reinforcement Learning (Lateef, 2019). Perbedaan mendasar dari supervised dan unsupervised learning adalah cara pengumpulan datanya. Untuk pendekatan supervised, data input dan output akan memiliki label. Sedangkan unsupervised, data tidak memiliki label. Sementara itu, reinforcement

merupakan bagian machine learning akan mempelajari dan mengamati state lingkungan(Lateef, 2019).

## **2.9. Computer Vision**

Computer Vision atau Visi Komputer adalah sebuah bagian dari kecerdasan buatan (AI) yang memungkinkan komputer dan sistem untuk memperoleh informasi dari sumber gambar, video, atau input visual lainnya dan mengambil tindakan atau rekomendasi berdasarkan informasi tersebut. Visi komputer akan memungkinkan komputer untuk melihat, mengamati, dan memahami informasi serta data yang dimiliki.

Visi komputer membutuhkan data dalam jumlah banyak berupa gambar yang nantinya akan memudahkan dalam menganalisis berulang kali sampai berhasil menemukan dan mengenali input nya. Untuk mencapai tujuannya, visi komputer membutuhkan teknologi yang disebut Deep Learning. Machine learning menggunakan model algoritma akan memungkinkan komputer dalam melatih dirinya tentang konteks data visual dengan memecah gambar menjadi piksel dan diberi label (Supervised Learning).

Visi komputer telah diimplementasikan ke hampir seluruh bidang, seperti aplikasi manufaktur, robot industri, asisten pelayanan restoran, analisis di bidang kesehatan maupun kedokteran, kendaraan otonom, dan masih banyak lagi. Tugas umum dari visi komputer adalah pengenalan (recognition), analisis gerakan, rekonstruksi adegan, dan restorasi gambar. Metode dari CV meliputi akuisisi gambar, pra-pemrosesan, ekstraksi fitur, deteksi/segmentasi, pemrosesan tingkat tinggi, dan pengambilan keputusan.

## **2.10. Pengolahan Citra Digital**

Pengolahan Citra Digital adalah proses mengolah piksel-piksel pada gambar untuk mencapai tujuan yang diinginkan(Riadi, 2016). Seiring berjalannya waktu, pengolahan citra digital tidak hanya sebatas dalam memperbaiki kualitas citra tetapi dapat mengambil informasi dari gambar. Tahapan pengolahan citra digital dapat dikategorikan berdasarkan peringkat pengolahannya yaitu pengolahan tingkat rendah (Low-Level Processing), pengolahan tingkat menengah (Mid-Level Processing), dan pengolahan tingkat tinggi (High-Level Processing)(Riadi, 2016). Teknik-teknik yang

dapat digunakan adalah image enhancement atau memperbaiki kualitas baik kontras atau kecerahan citra, image restoration atau memperbaiki bentuk model, memperbaiki warna citra (color image processing), teknik wavelet dan multiresolution processing dimana citra akan dinyatakan dalam beberapa resolusi, mengubah ukuran data (image compression), mempelajari informasi yang memiliki kesamaan dalam bentuk organ (morphological processing), memisahkan atau mensegmentasi seluruh objek yang ada di gambar, dan mengenali objek yang ada pada citra (object recognition)(Riadi, 2016).



Gambar 2.6 Pengolahan Citra Digital

### 2.11. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan "hyperplane" optimal yang berfungsi untuk memisahkan ("mengklasifikasikan") observasi yang termasuk dalam satu kelas dari yang lain berdasarkan pola informasi tentang observasi tersebut yang disebut sebagai fitur, baik dalam pola linear maupun nonlinear yang dapat dipisahkan(X. Yang, 2019). Maksud dari hyperplane optimal adalah yang "memaksimalkan margin" antara kelas-kelas yang ada(X. Yang, 2019). SVM sangat cocok untuk menangani berbagai masalah klasifikasi, seperti diagnosis dan prognosis penyakit, seperti Alzheimer (AD), schizophrenia, depresi, katarak, glukoma, diabetes, dan lain-lain(Pisner & Schnyer, 2020).

Terdapat beberapa jenis SVM, seperti SVM linier untuk data terpisah secara linier dan SVM non-linier yang menggunakan kernel untuk menangani data yang tidak dapat dipisahkan secara linier(X. Yang, 2019).

Dibandingkan dengan jenis metode klasifikasi lainnya, kelebihan dan popularitas SVM adalah kemampuannya dalam mencapai kinerja seimbang dan akurasi tinggi yang dapat digeneralisasi, bahkan dalam kasus dengan dimensi yang tinggi.

Proses perhitungan SVM mencakup optimasi hyperplane dengan memaksimalkan margin antara kelas-kelas. Rumus perhitungan hyperplane atau decision boundary untuk SVM linear adalah

$$f(x) = w \cdot x + b = 0 \quad (2.1)$$

Dimana:

- $f(x)$  adalah prediksi kelas untuk data  $x$
- $w$  adalah vektor bobot
- $b$  adalah bias

$w$  dan  $b$  pada rumus tersebut digunakan untuk mendefinisikan posisi hyperplane yang terpisah yang mencakup (Fattahi, 2015):

$$y_i f(x_i) = y_i(w x_i + b) \geq 1, \quad i = 1, 2, \dots, n \quad (2.2)$$

Atau pendekatan yang menggunakan variable slack non negatif(Yang, 2019):

$$y_i f(x_i) = y_i(w x_i + b) \geq 1 - \eta_i, \quad i = 1, 2, \dots, n \quad (2.3)$$

Selanjutnya pada permasalahan optimasi pada Support Vector Machine adalah

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1} \eta_i, \quad (2.4)$$

Sesuai dengan:

$$y_i(w x_i + b) \geq 1 - \eta_i, \quad (2.5)$$

$$\eta_i \geq 0 \quad (i = 1, 2, \dots, n) \quad (2.6)$$

Pada SVM non-linier dengan menggunakan fungsi kernel dengan fungsi sebagai berikut(Fattahi, 2015):

$$f(x) = \text{sign}(n \sum_{i=1} a_i y_i K(x_i, x_j) + b) \quad (2.7)$$

dimana

$$K(x_i, x_j) \text{ adalah fungsi kernel} \quad (2.8)$$

Kernel SVM dapat disesuaikan berdasarkan permasalahan, untuk kernel polynomial menggunakan

$$K(x_i, x_j) = (x_i \cdot x_j)^d \quad (2.9)$$

Untuk neural networks dapat menggunakan fungsi

$$K(x_i, x_j) = \tanh[k(x_i, x_j) + \theta] \quad (2.10)$$

Selanjutnya untuk kernel RBF menggunakan rumus

$$\begin{aligned} K(x_i, x_j) &= \exp[-\|x_i - x_j\|^2 / (2\sigma^2)] \\ &= \exp[-\gamma\|x_i - x_j\|^2] = \exp[-\gamma r^2] \end{aligned} \quad (2.11)$$

Terdapat juga pendekatan One-vs-One (OvO) dan One-vs-All (OvA) dalam implementasi SVM, dimana OvO membangun model untuk setiap pasangan kelas dan OvA membangun model terpisah untuk setiap kelas. Pemilihan jenis SVM dan penyesuaian parameter perlu diperhatikan sesuai dengan karakteristik dataset yang digunakan.

## 2.12. Pendekatan OvO - Support Vector Machine (SVM)

Support Vector Machine (SVM) dengan pendekatan One-vs-One (OvO) adalah algoritma machine learning yang digunakan untuk klasifikasi. Dalam konteks OvO, SVM membangun model untuk setiap pasangan kelas yang mungkin ada dalam dataset, sehingga jika ada  $n$  kelas, akan ada  $n(n-1)/2$  model SVM. Misalnya, untuk tiga kelas (A, B, dan C), SVM akan membuat model untuk A vs B, A vs C, dan B vs C.

Rumus matematis yang mendasari Support Vector Machine (SVM) dalam pendekatan One-vs-One (OvO) melibatkan proses optimasi untuk mencari hyperplane yang memaksimalkan margin antara pasangan kelas (seperti pada rumus  $y(x)$  subbab Support Vector Machine). Di mana  $x$  adalah vektor fitur input. Dalam kasus OvO, untuk setiap pasangan kelas  $i$  dan  $j$ , optimasi melibatkan fungsi tujuan yang mencari nilai minimum dari norma vektor bobot ( $\|w\|$ ) dengan batasan terkait margin dan ketidaksetaraan kelas:

$$\text{minimize } \frac{1}{2}\|w\|^2 = \frac{1}{2}\|w\|_2^2 = \frac{1}{2}(w \cdot w) \quad (2.12)$$

dengan batasan:

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, \dots, n \quad (2.13)$$

di mana  $y_i$  adalah label kelas,  $x_i$  adalah vektor fitur untuk data ke-i, dan n adalah jumlah data dalam satu pasangan kelas [buku svm 2].

### 2.13. Harmony Search

Harmony Search (HS) adalah algoritma optimasi metaheuristik yang terinspirasi oleh proses penciptaan harmoni dalam musik, seperti saat seorang musisi melakukan improvisasi nada instrumen dalam menemukan kombinasi nada yang ideal (Pereira et al., 2013). Algoritma ini digunakan untuk menemukan solusi optimal dalam ruang pencarian yang kompleks yang merupakan himpunan nilai-nilai parameter potensial untuk solusi optimal, dan kemudian melakukan penyelarasan dan improvisasi untuk meningkatkan nilai-nilai tersebut seiring waktu (Oliva et al., 2013).

Solusi-solusi yang didapatkan dari Harmony Search disebut dengan “harmony” dan direpresentasikan oleh vektor nyata dengan dimensi n dan disimpan kedalam harmony memory (HM). Perhitungan dalam Harmony Search melibatkan elemen-elemen dari harmony memori (HM) dengan proses pertimbangan memori, improvisasi, penyelarasan, evaluasi dan memperbaiki harmoni terburuk dengan harmoni yang dihasilkan (merupakan solusi yang lebih atau paling baik). Proses Harmony Search akan terus berulang atau iterasi hingga solusi optimal didapatkan (Oliva et al., 2013).

Terdapat tiga fase utama dalam algoritma Harmony Search, antara lain:

1. Inisialisasi Masalah dan Parameter Algoritma

$$\text{minimize } f(x), \quad x = (x(1), x(2), \dots, x(n)) \in R^n,$$

berlandaskan pada:

$$x(j) \in [l(j), u(j)], \quad j = 1, 2, \dots, n, \quad (2.14)$$

dimana  $f(x)$  adalah fungsi objektif,

$x = (x(1), x(2), \dots, x(n))$  adalah himpunan variable, n adalah banyaknya variable, dan  $l(j)$  dan  $u(j)$  adalah batas

bawah dan batas atas untuk variable  $x(j)$ .

Parameter yang digunakan dan mempengaruhi dalam harmony search adalah ukuran harmony memory (jumlah vektor solusi dalam harmony memory), tingkat pertimbangan harmoni-memori (Harmony-Memory Consideration Rate - HMCR), tingkat penyesuaian nada (Pitch Adjusting Rate - PAR), bandwidth jarak (Distance Bandwidth - BW) dan jumlah improvisasi (NI) yang merupakan total iterasi (Lobo et al., 2007).

## 2. Inisialisasi Harmony Memory

Pada tahapan ini, metrik Harmony Search (HM) dibentuk dan diisi secara acak dengan nilai variabel desain dan mencakup vektor solusi yang memungkinkan.

Misalkan  $x_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$  menyatakan harmony vector ke- $i$  yang dihasilkan secara acak:  $x_i(j) = l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1)$  untuk  $j = 1, 2, \dots, n$  dan  $i = 1, 2, \dots, \text{HMS}$ ,  $\text{rand}(0, 1)$  adalah bilangan acak antara angka 0 hingga 1 dan  $l(j)$  dan  $u(j)$  untuk menentukan batas atas serta bawah dari ruang pencarian masing-masing. Sehingga matrix HM berisikan dengan HMS vector seperti:

$$\text{HM} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{\text{HMS}} \end{bmatrix}$$

## 3. Improvisasi Vektor Harmony Baru

Vektor harmoni baru didapatkan dari harmony memory berdasarkan tingkat pertimbangan harmoni-memori (Harmony-Memory Consideration Rate - HMCR), tingkat penyesuaian nada (Pitch Adjusting Rate - PAR), random reinitization (Harmony-Memory Consideration Rate - IHMCR), dan pitch adjustment (Oliva et al., 2013). Tahapan ini disebut dengan “improvisasi”.

Pada tahapan tingkat pertimbangan harmoni memori atau HMCR,

nilai variabel pertama  $x_{new}(1)$  untuk vektor yang baru diambil secara acak dari nilai yang sudah ada dalam Harmony Memory yang didapatkan dari  $x_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$  (2.15).

Selanjutnya didapatkan angka acak  $r1$  atau  $\text{rand}(0,1)$  dalam rentang  $[0,1]$ . Jika  $r1 < \text{HMCR}$ , maka nilai variabel keputusan  $x_{new}(1)$  didapatkan dari HMCR. Jika  $r1$  tidak memenuhi kondisi tersebut, maka dilakukan inisialisasi ulang acak antara batas  $[l(1), u(1)]$  sehingga akan mendapatkan  $x_{new}(1)$ . Proses ini akan terus berulang untuk mencari nilai-nilai keputusan lainnya, atau dapat direpresentasikan kedalam:

$$x_{new}(j) \quad (2.16)$$

$$= \begin{cases} x_i(j) \in \{x_1(j), x_2(j), \dots, x_{\text{HMS}}(j)\}, \\ \quad \text{with probability HMCR}, \\ l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1), \\ \quad \text{with probability } 1 - \text{HMCR}. \end{cases}$$

Setiap elemen yang dihasilkan melalui pertimbangan memori (HMCR) kemudian diperiksa lebih lanjut untuk menentukan apakah penyesuaian nada diperlukan. Dalam konteks ini, operasi penyesuaian nada melibatkan penggunaan tingkat penyesuaian nada (Pitch Adjusting Rate - PAR), yang didefinisikan untuk menetapkan frekuensi penyesuaian, dan (Distance Bandwidth - BW) untuk mengendalikan pencarian lokal di sekitar elemen terpilih dari Harmony Memory (HM). Oleh karena itu, perhitungan penyesuaian nada untuk keputusan dilakukan seperti berikut:

$$x_{new}(j) \quad (2.17)$$

$$= \begin{cases} x_{new}(j) = x_{new}(j) \pm \text{rand}(0, 1) \cdot \text{BW}, \\ \quad \text{with probability PAR}, \\ x_{new}(j), \quad \text{with probability } (1 - \text{PAR}) \end{cases}$$

Penyesuaian nada memiliki tanggung jawab untuk menciptakan potensi harmoni baru dengan melakukan sedikit modifikasi pada



posisi variabel asli. Proses ini dapat dibandingkan dengan operasi mutasi dalam algoritma evolusi. Oleh karena itu, keputusan variabel tersebut terpengaruh oleh angka acak dalam rentang 0 hingga BW, atau dibiarkan tidak berubah. Untuk menjaga operasi penyesuaian nada, penting untuk memastikan bahwa titik-titik tersebut berada di luar rentang yang valid  $[l, u]$  atau harus sesuai dengan nilai maksimum atau minimum dari interval.

4. Improvisasi Vektor Harmony Baru

Langkah ini dilakukan ketika vektor harmoni baru  $x_{new}$  lebih baik dari harmoni terburuk  $x_w$  yang ada di Harmony Memory yang ditinjau dari nilai fungsi tujuan.

5. Prosedur Komputasi Harmony Search

Jika Harmony search digunakan untuk memaksimalkan fungsi tujuan, maka ada modifikasi dari langkah keempat menjadi

$$(x_w = x_{new} \text{ jika } f(x_{new}) > f(x_w)) \quad (2.18).$$

## **BAB III**

### **METODOLOGI**

#### **3.1. Deskripsi Studi Kasus**

Studi kasus ini bertujuan untuk mengembangkan sebuah sistem klasifikasi atau grading tingkat keparahan katarak sebagai computer-aided-system, melibatkan penggunaan metode Support Vector Machine (SVM) dan algoritma Harmony Search. Latar belakang studi ini didorong oleh fakta bahwa katarak merupakan penyebab utama gangguan penglihatan dan kebutaan di dunia, dengan tingkat kejadian tertinggi di Indonesia, khususnya di provinsi Jawa Timur. Pengembangan sistem ini menjadi penting karena observasi citra fundus mata dalam menegakkan diagnosis katarak seringkali bergantung pada keahlian individu ahli optalmologi, yang dapat menyebabkan variasi dan kurangnya standarisasi. Oleh karena itu, studi ini akan memanfaatkan teknologi Machine Learning, khususnya SVM, dan algoritma Harmony Search untuk mengoptimalkan penimbangan fitur pada citra fundus mata. Data mencakup berbagai tingkatan katarak dan akan dievaluasi menggunakan metrik akurasi. Hasil dari studi ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan akurasi dan efisiensi dalam diagnosis katarak, serta menjadi landasan untuk pengembangan teknologi computer-aided-system di bidang optalmologi.

#### **3.2. Kebutuhan Hardware dan Software**

Kebutuhan perangkat keras (hardware) yang akan digunakan pada penelitian ini dijelaskan sebagai berikut:

- Laptop Asus FHVE7RQ2 dengan spesifikasi Processor Intel Core I7-1165G7, Nvidia, RAM 8 GB, @ 2.80GHz 2.80 GHz.
- Alat pengambilan citra fundus mata, seperti funduscopy.

Kebutuhan perangkat lunak (software) yang akan digunakan pada penelitian ini dijelaskan sebagai berikut:

- Sistem Operasi Windows 10
- Google Colaboratory
- Python

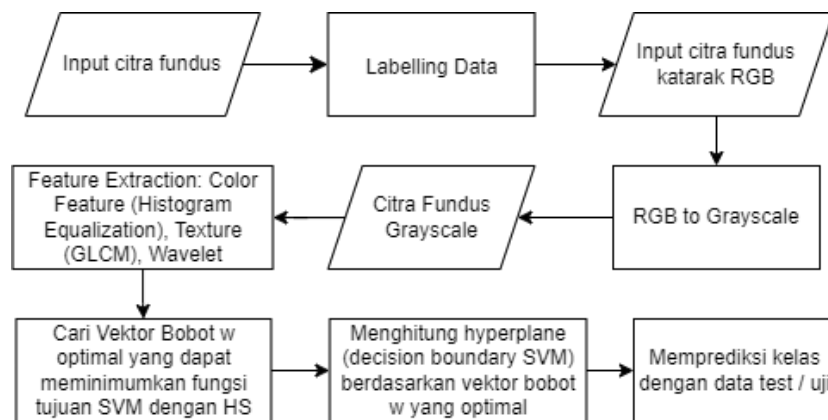
### 3.3. Jenis dan Sumber Data

Penelitian ini menggunakan 2 data yaitu citra fundus yang didapatkan dari Rumah Sakit Mata Undaan Surabaya sebagai data primer dan opensource Kaggle sebagai data sekunder. Untuk data primer didapatkan total citra fundus mata katarak dan non katarak sebanyak 300 citra serta data sekunder berjumlah 1074 citra fundus normal dan 1038 citra fundus katarak. Penelitian ini menggunakan data primer dan sekunder guna untuk mengetahui keefektifan dan efisien jika digunakan pada data real.

### 3.4. Studi Pustaka

Sebelum memulai penelitian, dilakukan pencarian informasi dan landasan teori dari beberapa sumber pustaka agar latar belakang penelitian serta metode yang digunakan memiliki teori yang kuat. Sumber pustaka yang dikumpulkan berupa jurnal maupun buku yang relevan dengan penelitian, yaitu mengenai penyakit mata katarak, metode Support Vector Machine (SVM), dan algoritma Harmony Search.

### 3.5. Tahapan Penelitian



Gambar 3.1 Flowchart Alur Penelitian

Penelitian ini melibatkan beberapa tahapan, yaitu:

1. Pengumpulan data: Data primer yang didapatkan dari Rumah Sakit Mata Undaan Surabaya dengan total 300 citra fundus katarak dan non katarak serta data sekunder yang didapatkan dari opensource Kaggle berjumlah 1074 citra fundus normal dan 1038 citra fundus katarak.
2. Labelling data: Pada tahapan ini dilakukan labelling pada keseluruhan data kedalam 4 kategori yaitu non katarak / normal, ringan - insipiens,

katarak sedang - imatur, dan katarak berat - matur serta hiper matur.

3. Pre-processing data: Pada tahapan ini dilakukan konversi atau merubah format dataset dari RGB (Red Green Blue) ke Grayscale.
4. Feature Extraction: Setelah dataset diubah menjadi skala grayscale, data diekstrak menggunakan histogram equalization sebagai fitur ekstraksi warna, GLCM sebagai fitur ekstraksi tekstur, dan transformasi wavelet untuk memberikan detail yang lebih jelas.
5. Split data menjadi train dan test data menggunakan k-fold cross validation: Pembagian data menjadi train dan test data menggunakan k-fold cross-validation untuk seluruh dataset sebagai tahapan pelatihan dan pengujian model secara bergantian dapat mengurangi risiko overfitting dan memperoleh estimasi performa yang lebih konsisten.
6. Pencarian vektor bobot  $w$  optimal yang dapat meminimumkan atau meminimalkan fungsi tujuan SVM dengan HS: Tahapan ini dilakukan untuk mencari nilai optimal dari vektor bobot ( $w$ ) dalam metode Support Vector Machine (SVM) dikarenakan  $w$  menjadi pengaruh utama dalam melakukan klasifikasi dan dapat meningkatkan akurasi lebih tinggi.
7. Menghitung hyperplane (decision boundary SVM) berdasarkan vektor bobot  $w$  optimal yang didapatkan dengan HS: Tahapan ini merupakan tahapan yang dilakukan dalam membangun hyperplane untuk setiap pasangan kelas dalam klasifikasi multikelas.
8. Prediksi kelas dari data baru atau test data: Tahapan terakhir adalah menggunakan model SVM yang telah dibangun untuk memprediksi kelas dari data baru.

### **3.6. Pengumpulan Dataset**

Tahap pertama adalah pengumpulan data primer yang dilakukan dari Rumah Sakit Mata Undaan Surabaya, yang menghasilkan total 300 citra fundus yang terbagi antara kasus katarak dan non-katarak. Kemudian, tahap kedua melibatkan pengumpulan data sekunder dari sumber terbuka Kaggle yang melibatkan pengunduhan dataset, di mana data berjumlah 1074 citra fundus normal dan 1038 citra fundus katarak. Kedua set data ini akan digunakan dalam analisis dan pengembangan model untuk mendukung penelitian atau aplikasi di bidang oftalmologi.

### 3.7. Labelling Data

Setelah tahapan pengumpulan data, langkah selanjutnya adalah labelling data, di mana keseluruhan dataset akan diberikan label atau kategori tertentu sesuai dengan tingkat keparahan katarak. Proses ini bertujuan untuk mempersiapkan dataset agar dapat digunakan dalam pengembangan model klasifikasi yang dapat membedakan berbagai tingkat katarak. Labelling dilakukan dalam empat kategori utama: (1) non-katarak atau normal, yang mencakup citra fundus mata tanpa tanda-tanda katarak; (2) katarak ringan-insipiens, yang menunjukkan gejala awal atau ringan dari katarak; (3) katarak sedang-imatur, mencerminkan katarak dalam tahap perkembangan yang lebih lanjut namun belum mencapai tingkat kematangan yang penuh; dan (4) katarak berat-matur serta hiper-matur, yang menggambarkan kondisi katarak yang sudah mencapai tahap kematangan atau bahkan hiper-maturitas. Proses labelling ini memerlukan pemahaman yang mendalam tentang karakteristik visual dari citra fundus serta pengetahuan tentang progresivitas katarak. Dengan labelling yang tepat, dataset akan menjadi lebih bermanfaat dalam pengembangan model yang dapat membantu dalam diagnosis dan penanganan katarak.

### 3.8. Pre-Processing Data

Tahapan pre-processing yang dilakukan adalah konversi atau mengubah format dataset dari RGB (Red Green Blue) ke Grayscale.

Berikut merupakan langkah-langkahnya konversi RGB ke Grayscale dengan pengambilan nilai intensitas dari satu channel:

1. Memperoleh Citra RGB:
  - Dapatkan citra berwarna dalam format Red, Green, dan Blue (RGB).
2. Pilih Channel:
  - Pilih satu channel dan hitung intensitas luminansinya dengan menggunakan rumus
$$Y = \text{intensitas luminasi dari suatu channel} \quad (3.1)$$
3. Mengevaluasi Kualitas:
  - Tinjau hasil citra grayscale yang dihasilkan dari saluran yang dipilih hingga hasilnya sesuai yang diinginkan atau dapat memperjelas detail dari dataset yang digunakan.

#### 4. Pilih Saluran Lainnya:

- Langkah ini dilakukan ketika hasil yang didapatkan sebelumnya tidak memuaskan dan cara perhitungannya sama seperti pada rumus (3.1) dan disesuaikan channelnya.

#### 5. Evaluasi Kualitas:

- Tinjau hasil citra grayscale yang dihasilkan dari saluran yang baru dipilih.

Jika langkah-langkah tersebut belum memuaskan, maka akan dilakukan pengambilan rata-rata nilai intensitas dari ketiga channel tersebut dengan menggunakan rumus

$$grayscale = (R + G + B)/3 \quad (3.2)$$

Tujuan tahapan konversi RGB ke Grayscale adalah untuk mengurangi kompleksitas data dan meningkatkan efisiensi pemrosesan serta memperjelas detail gambar. Selain itu, dengan citra Grayscale dapat menghilangkan informasi warna yang mungkin tidak relevan untuk deteksi katarak dan fokus pada tingkat kecerahan yang dapat menjadi fitur penting. Citra Grayscale juga memiliki ukuran file yang lebih kecil, sehingga dapat mempercepat proses komputasi dan menghemat ruang penyimpanan.

Melalui tahapan pre-processing ini, dataset yang sebelumnya berupa citra RGB telah disiapkan secara optimal untuk tahapan-tahapan ekstraksi fitur yang dilakukan selanjutnya, mencakup histogram equalization, GLCM (Grey-Level Co-occurrence Matrix), dan transformasi wavelet. Proses ini membentuk dasar yang terstruktur dan efisien dalam pengembangan model klasifikasi katarak berdasarkan tingkat keparahannya yang optimal.

### 3.9. Feature Extraction

Setelah tahap konversi dataset ke skala grayscale, langkah selanjutnya adalah ekstraksi fitur. Proses ini melibatkan tiga ekstraksi fitur, yaitu histogram equalization, GLCM (Grey-Level Co-occurrence Matrix), dan transformasi wavelet.

Pertama-tama, histogram equalization digunakan untuk meningkatkan distribusi intensitas piksel dalam citra dengan tujuan untuk memperbaiki kontras dan membuat fitur pada citra lebih terlihat. Proses dari histogram equalization dimulai dapat dijelaskan dengan langkah-langkah sebagai berikut

1. Hitung Histogram Gambar:

- Buat histogram (grafik yang menunjukkan jumlah piksel untuk setiap nilai intensitas warna) yang menunjukkan sebaran intensitas warna dalam gambar.

2. Normalisasi Histogram:

- Normalisasikan nilai histogram agar total jumlah piksel sama dengan jumlah piksel gambar. Ini dilakukan dengan membagi setiap nilai histogram dengan jumlah total piksel dengan rumus

$$P_r(r_k) = (P_r(r_k))/MN \quad (3.3)$$

dimana  $P_r(r_k)$  adalah nilai normalisasi dari intensitas warna ( $r_k$ ),  $M$  dan  $N$  adalah lebar dan tinggi gambar, dan  $MN$  adalah total jumlah piksel.

3. Hitung Fungsi Kumulatif (Cumulative Distribution Function - CDF):

- Hitung fungsi kumulatif dari histogram normalisasi dengan rumus

$$CDF(r_k) = \sum_{j=0}^k P_r(r_j) \quad (3.4)$$

dimana  $CDF(r_k)$  adalah nilai CDF untuk intensitas warna  $r_k$

4. Equalization Function:

- Tentukan fungsi equalization menggunakan rumus:

$$s_k = \text{round} \left( \frac{L-1}{MN} \cdot CDF(r_k) \right) \quad (3.5)$$

dimana  $L$  adalah level intensitas,  $MN$  adalah total jumlah piksel,  $r_k$  adalah intensitas warna, dan  $s_k$  adalah nilai baru setelah equalization.

5. Terapkan Fungsi Equalization:

- Terapkan fungsi equalization pada setiap piksel gambar. Gantilah nilai piksel  $r$  dengan  $s$  sesuai dengan fungsi equalization sebagai berikut

$$s = \text{round} \left( \frac{L-1}{MN} \cdot CDF(r) \right) \quad (3.6)$$

6. Hasil Akhir:

- Gambar yang telah dihasilkan memiliki distribusi intensitas warna yang merata, sehingga kontrasnya meningkat.

Langkah-langkah ini membantu dalam memperbaiki distribusi intensitas warna pada gambar dan dapat meningkatkan kualitas visualnya.

Selanjutnya, GLCM digunakan sebagai ekstraksi fitur tekstur. Langkah-langkah penghitungan GLCM meliputi:

1. Hitung GLCM:
  - Pilih arah (misalnya,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ) dan jarak tertentu.
  - Pindai gambar dan hitung matriks GLCM untuk setiap pasangan piksel dengan intensitas  $i$  dan  $j$  pada arah dan jarak yang ditentukan.
2. Normalisasi GLCM:
  - Normalisasikan matriks GLCM dengan membagi setiap elemen dengan jumlah total elemen dalam matriks.
3. Hitung Fitur GLCM (opsional):
  - Menghitung dari matriks GLCM yang telah dinormalisasi, seperti kontras, energi, homogenitas, dan lainnya.
4. Ulangi untuk Berbagai Arah dan Jarak:
  - Ulangi langkah-langkah tersebut untuk pada arah dan jarak lainnya.

Dengan demikian, GLCM memberikan gambaran tentang sejauh mana dan seberapa sering piksel dengan nilai keabuan tertentu berhubungan dengan piksel lainnya dalam suatu arah tertentu. Pada tahapan ini, GLCM memberikan informasi tentang tekstur pada citra, yang dapat digunakan sebagai fitur ekstraksi penting dalam membedakan kelas.

Terakhir, transformasi wavelet digunakan untuk mendapatkan detail yang lebih jelas pada citra. Langkah-langkahnya dari transformasi wavelet antara lain:

1. Dekomposisi (Forward DWT):
  - Pilih filter wavelet  $h$  dan  $g$  (misalnya, Haar, Daubechies, dll.)
  - Lakukan konvolusi sinyal asli ( $x$ ) dengan filter wavelet  $h$  dan  $g$  untuk mendapatkan koefisien wavelet.
  - Proses ini diulangi untuk setiap level dekomposisi ( $j$ ) hingga mencapai level tertentu atau resolusi terendah.
2. Normalisasi Koefisien:



- Normalisasi dapat dilakukan untuk memastikan bahwa energi sinyal tetap setelah dekomposisi.

### 3. Rekonstruksi (Inverse DWT):

- Lakukan rekonstruksi dari koefisien wavelet dan aproksimasi untuk mendapatkan sinyal asli.

Tahapan transformasi wavelet ini membantu dalam menangkap fitur-fitur pada berbagai tingkat skala, memperkaya representasi fitur citra, dan meningkatkan kemampuan model untuk memahami struktur kompleks dalam citra.

Tujuan dari tahapan ini adalah untuk menghasilkan representasi fitur yang lebih informatif dan mendalam dari setiap citra fundus mata. Dengan menggabungkan informasi warna, tekstur, dan detail pada tingkat frekuensi yang berbeda, model klasifikasi katarak yang dibangun akan memiliki landasan teori yang kuat sehingga kemampuan model dalam membedakan berbagai kelas katarak dengan tingkat ketepatan yang tinggi.

### 3.10. Split Data kedalam Train dan Test Data dengan K-Fold Cross

#### **Validation:**

Tahapan k-fold cross-validation dengan  $k=1$  hingga  $k=10$  yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Pembagian Data ( $k=1$  hingga  $k=10$ ): Dataset awal dibagi menjadi  $k$  subset yang disebut fold. Ketika  $k=1$ , seluruh dataset digunakan sebagai data uji, dan tidak ada data latih yang terpisah. Ketika  $k=2$ , dataset dibagi menjadi dua bagian yang sama besar. Proses ini berlanjut hingga  $k=10$ , di mana dataset dibagi menjadi 10 subset yang sama besar.

2. Iterasi Pembelajaran: Pada setiap iterasi, satu fold diambil sebagai data uji (test set), sementara  $k-1$  fold lainnya digunakan sebagai data latih (training set). Misalnya, pada  $k=5$ , terdapat lima iterasi di mana setiap fold akan menjadi data uji satu kali, dan sisa fold akan digunakan sebagai data latih.

3. Pelatihan Model: Model machine learning dipelajari dari data latih pada setiap iterasi. Ini melibatkan proses training di mana model belajar pola dan hubungan dari data latih. Pada tahapan ini dilakukan menggunakan metode Support Vector Machine.

4. Validasi Model: Model yang telah dilatih akan diuji menggunakan data

uji yang terpisah pada setiap iterasi. Hal ini dilakukan untuk mengukur seberapa baik model berkinerja pada data.

5. Perhitungan Performa: Performa model dievaluasi menggunakan metrik yang relevan seperti akurasi, presisi, recall, atau F1-score. Setiap iterasi menghasilkan nilai performa yang dapat digunakan untuk mengevaluasi kinerja model.

6. Iterasi: Proses diulangi sebanyak  $k$  kali, dengan setiap fold menjadi data uji satu kali. Dalam kasus  $k=10$ , tiap fold akan menjadi data uji satu kali, dan evaluasi model dilakukan sepuluh kali.

Dengan demikian,  $k$ -fold cross-validation memberikan estimasi yang lebih obyektif tentang kinerja model machine learning, meminimalkan bias yang mungkin muncul dari pembagian data yang spesifik. Proses ini membantu memastikan bahwa model yang akan dikembangkan dapat secara konsisten menggeneralisasi pola dari data, sesuai dengan tahapan atau alur penelitian yang dilakukan.

### **3.11. Pencarian Vektor Bobot $W$ Optimal Yang Dapat Meminimumkan atau Meminimalkan Fungsi Tujuan SVM dengan HS**

Pengoptimalan vektor bobot ( $w$ ) dengan Harmony Search merupakan tahapan untuk mengoptimalkan kinerja model klasifikasi. Langkah-langkah pada tahap ini melibatkan proses pencarian nilai optimal untuk vektor bobot ( $w$ ) dalam metode Support Vector Machine (SVM). Harmony Search digunakan untuk mencapai nilai  $w$  yang dapat memberikan hasil akurasi tertinggi dalam melakukan klasifikasi kelas-kelas katarak berdasarkan tingkat keparahan.

Metode Harmony Search adalah algoritma optimasi yang terinspirasi oleh proses pencarian harmoni dalam musik. Untuk mencari bobot vektor ( $w$ ) dengan Harmony Search, berikut langkah-langkahnya:

1. Inisialisasi Masalah dan Parameter Algoritma:

Melakukan inisialisasi masalah dengan menentukan fungsi tujuan  $f(x)$  yang ingin dioptimalkan dengan rumus:

$$\text{minimize } f(x), x = (x(1), x(2), \dots, x(n)) \in R^n, \quad (3.7)$$

berlandaskan pada:

$$x(j) \in [l(j), u(j)], j = 1, 2, \dots, n, \quad (3.8)$$

dengan parameter harmony search yaitu tingkat pertimbangan

harmoni-memori (Harmony-Memory Consideration Rate - HMCR), tingkat penyesuaian nada (Pitch Adjusting Rate - PAR), random reinitization (Harmony-Memory Consideration Rate - I HMCR), dan pitch adjustment.

2. Inisialisasi Populasi Harmoni dan Harmony Memory:

Menentukan jumlah harmoni atau individu dalam populasi dan inisialisasi nilai bobot ( $w$ ) untuk setiap harmoni secara acak dalam batasan yang ditentukan dengan rumus

$$x_{i,j} = rand(w_{min}, w_{max}) \quad (3.9)$$

Selanjutnya, inisialisasi metrik Harmony Search (HM) dengan dibentuk dan diisi secara nilai acak sehingga matrix HM berisikan akan berisikan dengan HMS vector sesuai yang telah didapatkan.

3. Improvisasi Vektor Harmony Baru

Langkah "improvisasi" dalam Harmony Search melibatkan pembentukan vektor harmoni baru dari harmony memory dengan mempertimbangkan tingkat pertimbangan harmoni-memori (HMCR), tingkat penyesuaian nada (PAR), dan random reinitialization (IHMCR):

1). Pertimbangan Harmoni-Memori (HMCR):

- Ambil nilai variabel pertama  $f(x_{new})$  dari Harmony Memory secara acak.
- Generate angka acak  $r1$  dalam rentang  $[0,1]$ .
- Jika  $r1 < HMCR$  maka  $x_{new}(1)$  didapatkan dari Harmony Memory. Jika sebaliknya, maka proses ini akan terus berulang hingga mendapatkan  $x_{new}$  dengan batas  $[l(1), u[1]$  Proses ini disebut dengan Random Reinitialization (IHMCR).

2). Penyesuaian Nada (Pitch Adjustment):\*\*

- Proses ini dilakukan untuk setiap elemen  $x_{new}(j)$  yang dihasilkan.
- Hitung penyesuaian nada menggunakan rumus  $x_{new(j)} = x_{new(j)} + rand(-1, 1) \cdot PAR \cdot BW$  dan pastikan nilai  $x_{new(j)}$  tetap dalam batas  $[l(1), u[1]$ .

4. Evaluasi Harmoni:

- Hitung fungsi tujuan untuk mendapatkan nilai fitness pada setiap harmoni dengan fungsi tujuan svm yaitu 
$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1} = \max(0, 1 - y(i)(wTx(i) + b))$$
 (3.10)

#### 5. Pembaruan Harmoni Baru:

- Setelah mendapatkan  $w$  pada fungsi tujuan SVM, maka pembaruan pada harmony terburuk  $x_w$  yang ada di Harmony Memory dengan vektor harmoni baru  $x_{new}$  lebih baik dari harmoni terburuk yang ditinjau dari nilai fungsi tujuan.

Tujuan dari pengoptimalan vektor bobot adalah menemukan kombinasi bobot yang optimal untuk setiap fitur dalam model SVM. Dengan melakukan pengoptimalan vektor bobot ( $w$ ), kemampuan model akan meningkat dalam menyesuaikan dengan data latih, mengidentifikasi pola yang rumit, dan meningkatkan tingkat akurasi klasifikasi pada data uji. Melalui tahapan pengoptimalan vektor bobot ( $w$ ) dengan harmony search, model SVM akan memiliki kinerja optimal dalam memisahkan kedalam multi kelas katarak.

#### 3.12. Menghitung Hyperplane (Decision Boundary SVM) Berdasarkan atau Dengan Vektor Bobot $w$ Yang Optimal Didapatkan Dengan HS.

Setelah mendapatkan vektor bobot optimal ( $w$ ) dengan Harmony Search dalam konteks fungsi tujuan Support Vector Machine (SVM), langkah selanjutnya adalah menggunakan vektor bobot tersebut untuk menghitung hyperplane atau decision boundary.

Rumus perhitungan hyperplane atau decision boundary untuk SVM linear adalah

$$f(x) = w \cdot x + b = 0 \quad (3.11)$$

Dimana:

- $f(x)$  adalah prediksi kelas untuk data  $x$
- $w$  adalah vektor bobot
- $b$  adalah bias

Decision boundary dicapai saat  $f(x) = 0$ . Dengan kata lain, dapat menetapkan nilai  $wTx + b = 0$ . Langkah-langkah perhitungan hyperplane atau decision boundary secara detail adalah sebagai berikut:

- a. Menggunakan vektor bobot optimal ( $w$ ) yang sebelumnya didapatkan menggunakan harmony search
- b. Menghitung Hyperplane:
  - Perhitungan hyperplane dalam SVM adalah  $f(x) = w \cdot x + b$  dengan mengatur  $f(x) = 0$  akan mendapatkan decision boundary atau hyperplane SVMnya.
- c. Selesaikan persamaan  $x$ , yang merupakan vektor fitur dan harus disesuaikan dengan jumlah fiturnya.

Perhitungan hyperplane SVM menggunakan vektor pembobotan optimal merupakan tahapan terpenting dalam membangun model klasifikasi yang mampu memisahkan antara kelas-kelas yang berbeda atau multikelas mencakup katarak ringan, sedang, dan berat.

### 3.13. Prediksi Kelas (Pelatihan dan Pengujian)

Tahapan selanjutnya adalah memprediksi kelas dari data test menggunakan model SVM yang telah dibangun sebelumnya. Langkah-langkah klasifikasi katarak berdasarkan tingkat keparahannya (mild, moderate, dan severe) menggunakan pendekatan SVM (Support Vector Machine) One-vs-One (OvO) dengan vektor bobot ( $w$ ) yang telah didapatkan dari tahapan Harmony Search dapat dijelaskan sebagai berikut:

1. Persiapkan Data Latih dan Data Uji
2. Latih SVM untuk Setiap Kombinasi Kelas (OvO):
  - Menggunakan vektor bobot optimal ( $w$ ) yang didapatkan dari Harmony Search untuk melatih SVM dengan pendekatan OvO.
  - Dalam pendekatan OvO, model SVM akan dilatih untuk setiap kombinasi dua kelas, seperti, mild vs. moderate, mild vs. severe, dan moderate vs. severe.
3. Klasifikasi Data Uji:
  - Gunakan model SVM yang telah dilatih dengan data test untuk mengklasifikasikan data uji dengan berbagai parameter kernel svm. Pada SVM kernel polynomial menggunakan

$$K(x_i, x_j) = (x_i \cdot x_j)^d \quad (3.12)$$

Untuk neural networks dapat menggunakan fungsi

$$K(x_i, x_j) = \tanh[k(x_i, x_j) + \Theta] \quad (3.13)$$

Selanjutnya untuk kernel RBF menggunakan rumus

$$\begin{aligned} K(x_i, x_j) &= \exp[-\|x_i - x_j\|^2 / (2\sigma^2)] \\ &= \exp[-\gamma\|x_i - x_j\|^2] = \exp[-\gamma r^2] \end{aligned} \quad (3.14)$$

- Untuk rumus klasifikasi dengan pendekatan OvO dapat dihitung menggunakan rumus

$$\operatorname{argmax}_k \sum_{l \neq k} f_{k,l}(x) \quad (3.15)$$

Dari tahapan pengujian model SVM, akan didapatkan akurasi berdasarkan confusion matrix dan perhitungan accuracy, precision, recall, F1-Score, dan specificity yang berbeda-beda dari pemilihan kernel dan parameter C serta Gamma dari SVM

### 3.14. Evaluasi Performa

Evaluasi performa model klasifikasi pada tahapan ini melibatkan beberapa metrik penting, yang memberikan wawasan mendalam tentang sejauh mana model mampu mengklasifikasikan citra fundus mata dengan katarak berdasarkan tingkat keparahan yang berbeda. Metrik-metrik tersebut termasuk confusion matrix, accuracy, precision, recall, dan F1-Score.

Confusion matrix adalah suatu tabel yang digunakan untuk mengevaluasi kinerja suatu model klasifikasi pada dataset yang diketahui labelnya. Confusion matrix memberikan gambaran holistik tentang sejauh mana model mampu mengklasifikasikan data dengan benar atau dengan kesalahan tertentu. Confusion matrix umumnya digunakan dalam evaluasi performa model klasifikasi, dan hasilnya dapat digunakan untuk menghitung berbagai metrik evaluasi seperti akurasi, presisi, recall, F1-score, dan lainnya.

Confusion matrix umumnya memiliki empat sel utama:

- True Positive (TP): Jumlah observasi yang sebenarnya termasuk dalam kelas positif dan diberi label positif oleh model.
- True Negative (TN): Jumlah observasi yang sebenarnya termasuk dalam kelas negatif dan diberi label negatif oleh model.
- False Positive (FP): Jumlah observasi yang sebenarnya termasuk dalam kelas negatif, tetapi diberi label positif oleh model (kesalahan jenis I).
- False Negative (FN): Jumlah observasi yang sebenarnya termasuk dalam

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 3.2 Confusion Matrix

Langkah-langkah dalam membuat confusion matrix:

1. Melakukan prediksi menggunakan model klasifikasi pada dataset uji.
2. Membandingkan dengan Label Sebenarnya pada dataset uji.
3. Menghitung True Positives, True Negatives, False Positives, dan False Negatives untuk mengetahui seberapa baik model mampu mengklasifikasikan setiap kelas, termasuk kelas katarak ringan, sedang, dan berat..
  - TP = Jumlah prediksi positif yang benar
  - TN = Jumlah prediksi negatif yang benar
  - FP = Jumlah prediksi positif yang salah
  - FN = Jumlah prediksi negatif yang salah
4. Membuat tabel yang berikan nilai TN, FP, FN, dan TP.

Metrik evaluasi yang didapatkan dari confusion matrix meliputi:

- Akurasi (Accuracy):

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+F} \quad (3.16)$$

Accuracy mengukur sejauh mana model benar-benar mengklasifikasikan seluruh citra secara tepat

- Presisi (Precision):

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (3.17)$$

precision mengevaluasi seberapa akurat model dalam mengklasifikasikan citra fundus mata dengan katarak, menghindari false positive

- Recall (Sensitivitas atau True Positive Rate):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3.18)$$

Recall untuk mengukur kemampuan model untuk mengidentifikasi seluruh citra katarak yang seharusnya diklasifikasikan sebagai positif, menghindari false negative.

- F1-Score:

$$\text{F1-Score} = \frac{2 \times \text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \quad (3.19)$$

merupakan harmonic mean dari precision dan recall, memberikan nilai yang seimbang antara kedua metrik tersebut.

Tujuan dari tahapan evaluasi performa ini adalah untuk memberikan pemahaman tentang seberapa baik model klasifikasi dapat membedakan citra fundus mata dengan katarak pada tingkat keparahan yang berbeda. Dengan mengevaluasi berbagai aspek performa, penelitian ini dapat memberikan kepercayaan dan wawasan yang diperlukan terkait dengan keefektifan dan keandalan model dalam mendukung proses diagnosis katarak.



## DAFTAR PUSTAKA

- Abel, R. (2018). Cataracts. *Integrative Medicine*, 830-837.e1.  
<https://doi.org/10.1016/b978-0-323-35868-2.00084-0>
- Abràmoff, M. D., & Kay, C. N. (2013). Image Processing. *Elsevier EBooks*, 151–176.  
<https://doi.org/10.1016/b978-1-4557-0737-9.00006-0>
- Agustina, N. (2022, October 18). *Direktorat Jenderal Pelayanan Kesehatan*.  
Yankes.kemkes.go.id.  
[https://yankes.kemkes.go.id/view\\_artikel/1697/mata-adalah-jendela-dunia#:~:text=Mata%20merupakan%20alat%20indra%20yang](https://yankes.kemkes.go.id/view_artikel/1697/mata-adalah-jendela-dunia#:~:text=Mata%20merupakan%20alat%20indra%20yang)
- Ali, L., Wajahat, I., Golilarz, N., Keshtkar, F., & Bukhari, S. (2020). LDA–GA–SVM: improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine. *Neural Computing and Applications*, 33(7), 2783–2792. <https://doi.org/10.1007/s00521-020-05157-2>
- Bachtiar, A. (2022). FAKTOR YANG BERHUBUNGAN DENGAN KEJADIAN KATARAK PADA RUMAH SAKIT UNIVERSITAS HASANUDDIN. *Repository Universitas Hasanuddin*.
- Budiono, S., Saleh, T., Moestidjab, & Eddyanto (Eds.). (2019). *Buku Ajar Ilmu Kesehatan Mata*. Airlangga University Press.
- Caplan, L. R. (2009). Nonatherosclerotic Vasculopathies. *Elsevier EBooks*.  
<https://doi.org/10.1016/b978-1-4160-4721-6.50013-4>
- Data Science Center. (2023). *Pengembangan Kecerdasan Buatan Untuk Identifikasi Penyakit Mata Kronis Dengan Menggunakan Citra Fundus – Pusat Data Sains*.  
Ui.ac.id.  
<https://dsc.ui.ac.id/pengembangan-kecerdasan-buatan-untuk-identifikasi-penyakit-mata-kronis-dengan-menggunakan-citra-fundus/#:~:text=Citra%20fundus%20ada>

lah%20gambar%20yang

Fattahi, H. (2015). PREDICTION OF SLOPE STABILITY STATE FOR CIRCULAR FAILURE: A HYBRID SUPPORT VECTOR MACHINE WITH HARMONY SEARCH ALGORITHM. *Iran University of Science & Technology*, 5(1), 103–115.

*Kenali 4 Stadium Katarak dan Cara Pencegahannya*. (2021, March 18). Ners.unair.ac.id. <https://ners.unair.ac.id/site/index.php/news-fkp-unair/30-lihat/1925-kenali-4-stadium-katarak-dan-cara-pencegahannya>

Khan, M., Lali, M. I., Sharif, M., Javed, K., Aurangzeb, K., Haider, S., Altamrah, A., & Akram, T. (2019). An Optimized Method for Segmentation and Classification of Apple Diseases Based on Strong Correlation and Genetic Algorithm Based Feature Selection. *IEEE Access*, 7, 46261–46277. <https://doi.org/10.1109/access.2019.2908040>

Kompas Cyber Media. (2019, October 13). *Kasus Katarak, Indonesia Tempati Peringkat Pertama di Asia Tenggara*. KOMPAS.com. <https://sains.kompas.com/read/2019/10/13/170300723/kasus-katarak-indonesia-tempati-peringkat-pertama-di-asia-tenggara>

Kumar Behera, M., Chakravarty, S., Gourav, A., & Dash, S. (2020). Detection of Nuclear Cataract in Retinal Fundus Image using RadialBasis FunctionbasedSVM. *Parallel, Distributed and Grid Computing (PDGC)*. <https://doi.org/10.1109/pdgc50313.2020.9315834>

Lateef, Z. (2019, April 22). *Introduction To Machine Learning: All You Need To Know About Machine Learning*. Edureka; Edureka. <https://www.edureka.co/blog/introduction-to-machine-learning/>

Lobo, F. J., Lima, C. F., & Michalewicz, Z. (2007). *Parameter Setting in Evolutionary*

*Algorithms*. Springer.

- Mayer, J., Donnelly, T., & Saunders, W. B. (Eds.). (2013). Cataracts. *Elsevier EBooks*, 435–437. <https://doi.org/10.1016/b978-1-4160-3969-3.00196-7>
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Pérez-Cisneros, M. (2013). Multilevel Thresholding Segmentation Based on Harmony Search Optimization. *Journal of Applied Mathematics*, 2013, 1–24. <https://doi.org/10.1155/2013/575414>
- Pereira, L., Papa, J., & Cristina, A. (2013). *Harmony search applied for support vector machines training optimization*. <https://doi.org/10.1109/eurocon.2013.6625103>
- Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. *Machine Learning*, 101–121. <https://doi.org/10.1016/b978-0-12-815739-8.00006-7>
- Qiao, Z., Zhang, Q., Dong, Y., & Yang, J.-J. (2017, October 1). *Application of SVM based on genetic algorithm in classification of cataract fundus images*. IEEE Xplore. <https://doi.org/10.1109/IST.2017.8261541>
- Rahajoe, A., Zainal, R., Mulyo, B., Plangkang, B., & Tias, R. (2020). Feature Selection Based on Modified Harmony Search Algorithm. *International Conference on Smart Technology and Applications (ICoSTA)*. <https://doi.org/10.1109/icosta48221.2020.1570615299>
- Riadi, M. (2016, April 21). *Pengolahan Citra Digital*. Kajian Pustaka. <https://www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html?m=1>
- Rokom. (2021, October 12). *Katarak Penyebab Terbanyak Gangguan Penglihatan di Indonesia*. Sehat Negeriku. <https://sehatnegeriku.kemkes.go.id/baca/umum/20211012/5738714/katarak-penyebab-terbanyak-gangguan-penglihatan-di-indonesia/>
- Sirajudeen, A., Balasubramaniam, A., & Karthikeyan, S. (2022). Novel angular binary pattern (NABP) and kernel based convolutional neural networks classifier for

- cataract detection. *Multimedia Tools and Applications*, 81.  
<https://doi.org/10.1007/s11042-022-13092-8>
- Stefanus, R. (2019, May 28). *Conventional Programming VS Machine Learning*.  
Medium.  
<https://rstefanus16.medium.com/conventional-programming-vs-machine-learning-a3b7b3425531>
- Sumarudin, A., Puspaningrum, A., & Suheryadi, A. (2020). Least-Square Support Vector Machine (LS-SVM) Parameters Optimization using Hybrid Cuckoo Search and Harmony Search Algorithm for Pre Collision Warning on Driver Assistance System (DAS). *Journal of Physics: Conference Series*, 1577(1), 012009.  
<https://doi.org/10.1088/1742-6596/1577/1/012009>
- Turbert, D. (2020, January 14). *Fundus*. American Academy of Ophthalmology.  
<https://www.aao.org/eye-health/anatomy/fundus>
- Vaughan, D., Asbury, T., Schaubert, L., Walibon, & Hariono, B. (1990). *Oftalmologi Umum* (14th ed). Jakarta: Penerbit Widya Media.
- Wicitra, R., Zamzam, M., & Cahyani, F. (2023). Studi Deskriptif Gambaran Mengenai Kasus Katarak dan Refraksi di Provinsi Jawa Timur Tahun 2022. *Jengala Jurnal Riset Pengembangan Dan Pelayanan Kesehatan*, 2, 12.
- Yang, X. (2019). Support vector machine and regression. *Elsevier EBooks*.  
<https://doi.org/10.1016/b978-0-12-817216-2.00014-4>
- Yang, X.-S. (2019). *Introduction to Algorithms for Data Mining and Machine Learning*. Elsevier. Copyright.