MAKALAH LAPORAN TUGAS AKHIR

RANCANG BANGUN APLIKASI *MOBILE ANDROID* UNTUK KLASIFIKASI CITRA LEUKOSIT DENGAN METODE *SUPPORT VECTOR MACHINE* (*SVM*) BERBASIS *PYTHON*

DESIGN AND DEVELOPMENT OF AN ANDROID MOBILE APPLICATION FOR LEUKOCYTE IMAGE CLASSIFICATION USING SUPPORT VECTOR MACHINE (SVM) METHOD BASED ON PYTHON

Fajrul Iman Giat Koentjoro[*1], Retno Supriyanti[2], Muhammad Syaiful Aliim[3]

Email[*]: fajrul.koentjoro@mhs.unsoed.ac.id

[1,2,3]Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

Abstrak— Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi *mobile Android* menggunakan metode *Support Vector Machine (SVM)* untuk klasifikasi citra leukosit, khususnya *Lymphoblast* dan *Myeloblast*. Penelitian ini didorong oleh tingginya tingkat mortalitas akibat leukemia dan tantangan dalam mendeteksi dini kelainan sel darah putih di daerah dengan fasilitas kesehatan terbatas. Metode yang digunakan meliputi segmentasi citra menggunakan teknik *threshold*, pengembangan model *SVM*, dan implementasi skrip *Python* ke dalam *Android Studio* dengan menggunakan *Chaquopy*. Hasil penelitian ini adalah aplikasi *Android* yang bernama *"LIFE (Leukocyte Identifier From Everywhere)"*. Hasil penelitian menunjukkan segmentasi citra yang efektif, model *SVM* dengan akurasi yang sangat baik, dan aplikasi yang berfungsi dengan lancar. Evaluasi kinerja aplikasi menunjukkan akurasi 98% dalam klasifikasi citra. Namun, aplikasi ini tidak sensitif terhadap variasi data, sehingga diperlukan kontrol terhadap kondisi pengambilan citra. Penelitian ini berkontribusi dalam meningkatkan efisiensi skrining awal penyakit kanker darah dan mendukung daerah dengan fasilitas kesehatan terbatas dalam pemantauan kesehatan.

Kata kunci — *Leukemia, Threshold, Support Vector Machine, Chaquopy,* Aplikasi *Mobile Android,* Pembelajaran Mesin

Abstract— *This study aims to design and develop an Android mobile application using the Support Vector Machine (SVM) method for leukocyte image classification, specifically Lymphoblast and Myeloblast. The research is motivated by the high mortality rate caused by leukemia and the challenges in early detection of white blood cell abnormalities in regions with limited healthcare facilities. The employed methods include image segmentation using the threshold technique, SVM model development, and the implementation of Python* skrips *into Android Studio using Chaquopy. The outcome of this research is an Android application named 'LIFE (Leukocyte Identifier From Everywhere)'. The results demonstrate effective image segmentation, a SVM model with impeccable accuracy, and a seamlessly functioning application. Performance evaluation of the application revealed a 98% accuracy in image classification. However, the application is not sensitive to data variations, necessitating control over the captured image conditions. This research contributes to enhancing the efficiency of early blood cancer screening and supports areas with limited healthcare facilities in health monitoring.*

Keywords — *Leukemia, Threshold, Support Vector Machine, Chaquopy, Android Mobile Apps, Machine learning*

I. PENDAHULUAN

A. Latar Belakang

Dalam kehidupan, manusia tidak lepas dari berbagai penyakit baik itu penyakit dengan tingkat mortalitas tinggi, rendah atau bahkan tidak berbahaya. Sebagian banyak penyakit yang diderita oleh manusia sudah ditemukan obat dan penawarannya namun tidak menutup fakta bahwa sebagian penyakit belum ditemukan obat dan penawarnya atau sudah ditemukan namun dalam penanganannya tidak boleh terlambat. Salah satu penyakit yang proses pengobatannya sudah ditemukan namun tingkat mortalitas tinggi karena

penanganan yang terlambat adalah kanker terutama kanker darah (Leukimia)[1].

Kanker darah (Leukimia) adalah jenis kanker yang diakibatkan pertumbuhan tidak terkontrol dari sel induk pembentuk sel darah dalam sum-sum tulang[1]. Secara umum, leukimia terjadi pada sel darah putih oleh karena itu disebut leukimia yang berasal dari kata Yunani yaitu *leukos* yang berarti putih dan *haima* yang berarti darah[1]. Leukimia terjadi pada dua jenis induk sel darah putih yaitu myeloid dan limfoid[2]. Kanker darah (leukimia) dapat dideteksi dengan melihat pertumbuhan yang tidak normal dari sel myeloid dan limfoid. Terjadinya kelainan darah ini mempengaruhi pertumbuhan sel darah merah sehingga mengakibatkan kanker darah putih (Leukimia). Sel darah yang tidak normal pertumbuhannya biasa disebut sebagai sel *blast*. Sedangkan sel limfoid abnormal disebut limfoblas *(Lymphoblast)* dan sel myeloid abnormal disebut mieloblas *(Myeloblast)* [1].

Kanker darah (Leukosit) akut dibagi menjadi dua berdasarkan sel induk yang terinfeksi yaitu *Acute Lymphostic Leukemia (ALL)* dan *Acute Myelostic Leukemia (AML)*[3]. Berdasarkan data yang penulis dapat dari berbagai sumber diketahui bahwa leukimia menempati peringkat ke-9 berdasarkan prevalensi kejadiannya yaitu sebesar 3.7% dari seluruh kanker di United State[1]. Menurut data yang berasal dari *Surveillance, Epidemology, and End Result Program National Cancer Institute* prevalensi leukimia adalah sebesar 13.7 per 100.000 populasi per tahun. Sedangkan kematian karena leukimia sebesar 6.8 per 100.000 populasi per tahun[3][4]. Pada tahun 2017 diperkirakan ada 62.130 kasus baru leukimia dan 24.500 orang meninggal karena mengidap leukimia di United State[1]. Bahkan kebanyakan kasus kanker pada anak adalah kanker leukimia. Dilihat pada data dari *American Childhood Cancer Organization* pada tahun 2016 prevalensi leukimia pada anak dengan usia 0-14 tahun adalah 40%[5].

Oleh karena itu, dalam penelitian ini penulis mengusulkan perancangan dan pembangunan aplikasi *mobile* untuk *screening* awal kelainan pertumbuhan sel *Myeloblast* dan *Lymphoblast* dengan menggunakan metode *SVM (Support Vector Machine)*. Hal ini dapat memudahkan daerah daerah-dengan fasilitas kesehatan yang terbatas untuk dapat mendeteksi secara dini adanya kelainan pertumbuhan sel *Lymphoblast* dan *Myeloblast* sehingga segera dapat dirujuk menuju fasilitas kesehatan yang lebih memadai untuk melakukan pengecekan kesehatan yang lebih spesifik lagi. Fasilitas terbatas hanya perlu menyediakan mikroskop manual dan perangkat berkamera yang sudah terinstal aplikasi penelitian ini. Maka dari itu penulis akan membuat penelitian dengan judul **"Rancang Bangun Aplikasi *Mobile Android* untuk Klasifikasi Citra Leukosit dengan Metode *Support Vector Machine (SVM)* berbasis *Python*"**.

B. Rumusan Masalah

Berdasarkan latar belakang di atas dapat diperoleh beberapa rumusan masalah sebagai berikut.

1. Bagaimana perancangan model *machine learning Support Vector Machine* untuk klasifikasi *Lymphoblast* dan *Myeloblast*?
2. Bagaimana uji unjuk kerja dari model *machine learning Support Vector Machine* dalam klasifikasi *Lymphoblast* dan *Myeloblast*?
3. Bagaimana implementasi model *machine learning Support Vector Machine* ke dalam aplikasi *mobile* berbasis *Android*?
4. Bagaimana uji unjuk kerja dari aplikasi *Android* untuk mengklasifikasi *Lymphoblast* dan *Myeloblast*?

C. Batasan Masalah

Agar penelitian tugas akhir didapatkan hasil yang optimal maka permasalahan dibatasi sebagai berikut:

1. Model *machine learning* dibuat untuk klasifikasi 2 (dua) jenis sel darah putih (Leukosit) yaitu *Lymphoblast* dan *Myeloblast*
2. Model klasifikasi yang dipilih untuk mengklasifikasi *Lymphoblast* dan *Myeloblast* adalah *Support Vector Machine* dengan menggunakan pustaka sumber terbuka *Scikit-learn* dalam bahasa *Python*
3. Pembuatan aplikasi *Android* sederhana menggunakan *Android Studio* dengan bahasa *Python* dan *Java*
4. Jenis *Software Development LIFE Cycle (SDLC)* yang digunakan adalah *Prototype*

D. Tujuan Penelitian

Tujuan penelitian pada tugas akhir ini adalah sebagai berikut.

1. Merancang dan membangun model *machine learning* untuk klasifikasi dua jenis sel darah putih yaitu *Lymphoblast* dan *Myeloblast*
2. Melakukan pengujian unjuk kerja dari model *machine learning* untuk klasifikasi dua jenis sel darah putih yaitu *Lymphoblast* dan *Myeloblast*

3. Merancang dan membangun aplikasi *Android* sederhana untuk klasifikasi dua jenis sel darah putih yaitu *Lymphoblast* dan *Myeloblast*

E. Manfaat Penelitian

Manfaat yang diharapkan pada penelitian ini adalah sebagai berikut.

1. Menambah khazanah keilmuan Teknik Elektro terutama dalam implementasi *Support Vector Machine* dalam memecahkan masalah
2. Membantu pengembangan keilmuan dalam bidang teknologi medis
3. Menghasilkan perangkat yang dapat membantu pemantauan kesehatan bagi daerah daerah-dengan fasilitas kesehatan terbatas
4. Menghasilkan perangkat yang dapat membantu meningkatkan efisiensi waktu dan biaya dalam *screaning* awal penyakit kanker sel darah putih (*Leukemia*)

II. TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Dalam penelitian ini penulis membawa referensi-referensi penelitian terdahulu tentang identifikasi dan klasifikasi jenis sel darah putih menggunakan *machine learning* diantarnya sebagai berikut ini.

1. Referensi pertama yaitu penelitian yang dilakukan oleh Jajat Eka Rahayu dengan judul "Implementasi *SVM (Support Vector Machine)* dalam Identifikasi Morfologi Sel Darah Putih (Leukosit)" yang menghasilkan sebuah aplikasi berbasis *software MATLAB* untuk mendeteksi jenis sel darah putih *Lymphoblast* dan *Myeloblast* menggunakan algoritma *SVM*. Menurut peneliti, *machine learning* dapat membantu dalam *screaning* awal kanker *leukemia*. Hasil penelitian menunjukkan bahwa *SVM* yang diimplementasikan dalam pemrograman *MATLAB* dapat mengklasifikasikan kedua jenis sel darah putih dengan akurasi sekitar $\pm 96{,}5\ \%$ [6].
2. Referensi kedua adalah sebuah jurnal penelitian yang ditulis oleh Aris Budiyanto dengan judul "Klasifikasi Citra Sel Darah Putih Berdasarkan Ciri Struktur Morfologi, Tekstur dan Jumlah *Hole* dengan Metode *K-Nearest Neighbors (KNN)*". Penelitian ini bertujuan untuk membangun model klasifikasi jenis sel darah putih, yaitu *Lymphoblast* dan *Myeloblast*, menggunakan algoritma *K-Nearest Neighbors (KNN)* dengan menggunakan pemrograman *MATLAB*. Hasil penelitian menunjukkan bahwa model yang digunakan berhasil mencapai akurasi sekitar $\pm 98\%$ [7].

B. Kanker Sel Darah Putih *(Leukemia)*

Leukemia adalah jenis kanker yang menyerang sel-sel darah putih. Ada beberapa jenis *leukemia*, seperti *leukemia* limfostik akut *(ALL)*, *leukemia* myeloid akut *(AML)*, *leukemia* limfostik kronis *(CLL)*, dan *leukemia* myeloid kronis *(CML)* [8].

Leukemia limfostik akut *(ALL)* adalah jenis *leukemia* yang paling umum pada anak-anak. Gejala *ALL* meliputi kelelahan, kehilangan nafsu makan, demam, dan memar atau perdarahan yang mudah. Pengobatan untuk *ALL* biasanya melibatkan kemoterapi dan terapi radiasi, dan dapat mencakup transplantasi sumsum tulang [8].

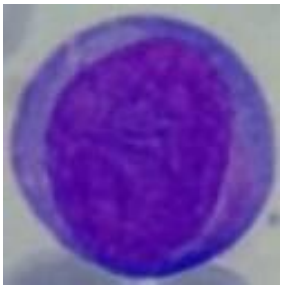
Leukemia myelostik akut *(AML)* adalah jenis *leukemia* yang lebih umum pada orang dewasa daripada pada anak-anak. Gejala *AML* meliputi kelelahan, sesak napas, pusing, dan perdarahan atau memar yang mudah. Pengobatan untuk *AML* biasanya melibatkan kemoterapi dan transplantasi sumsum tulang [9].

Leukemia kronis adalah jenis *leukemia* yang tumbuh dan berkembang secara lebih lambat daripada *leukemia* akut. *Leukemia* limfostik kronis *(CLL)* dan *leukemia* mielostik kronis *(CML)* adalah dua jenis *leukemia* kronis yang paling umum. Gejala *leukemia* kronis dapat termasuk pembengkakan kelenjar getah bening, kelelahan, dan demam. Pengobatan untuk *leukemia* kronis dapat mencakup kemoterapi, terapi biologis, dan transplantasi sumsum tulang [10].

Maka, dari penjelasan tersebut dapat diketahui bahwa *Leukemia* dibedakan menjadi dua berdasarkan sel induk pembentuk sel darah putih (*blast*) yang mengalami kelainan yaitu sel *Lymphoblast* dan sel *Myeloblast*.

a. *Lymphoblast*

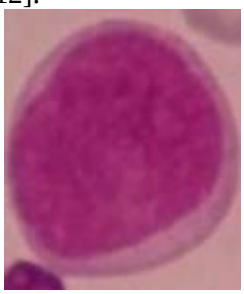
Lymphoblast adalah sel muda dari sistem kekebalan yang berasal dari sel punca *hematopoietik* dan merupakan *prekursor* limfosit. Mereka memiliki inti besar, *sitoplasma* sedikit, dan dapat membelah cepat. Terdapat di sumsum tulang, kelenjar getah bening, dan jaringan limfoid lainnya. *Lymphoblast* dapat menjadi sel kanker, seperti *leukemia* limfostik akut *(ALL)* atau limfoma. Terapi radiasi dan kemoterapi dapat digunakan untuk pencegahan dan pengobatan kanker tersebut [11].



Gambar-1. Sel Darah Putih *Lymphoblast*

b. *Myeloblast*

Myeloblast adalah sel darah putih muda yang berasal dari sel punca *hematopoietik* di sumsum tulang dan menjadi *prekursor* bagi jenis sel darah putih *granulosit* seperti *neutrofil*, *eosinofil*, dan *basofil*. *Myeloblast* memiliki inti besar, *nukleolus* jelas, dan sedikit *sitoplasma*, serta mampu membelah cepat. Mereka biasanya ditemukan di sumsum tulang dan dapat juga ditemukan di darah pada beberapa penyakit. *Leukemia* mielostik akut *(AML)* adalah kondisi di mana sel *Myeloblast* yang abnormal dan ganas tumbuh secara tidak terkendali. Pencegahan dan pengobatan kanker tersebut dapat dilakukan dengan berbagai cara, termasuk terapi radiasi dan kemoterapi, transplantasi sumsum tulang, dan terapi gen [12].



Gambar-2. Sel Darah Putih *Myeloblast*

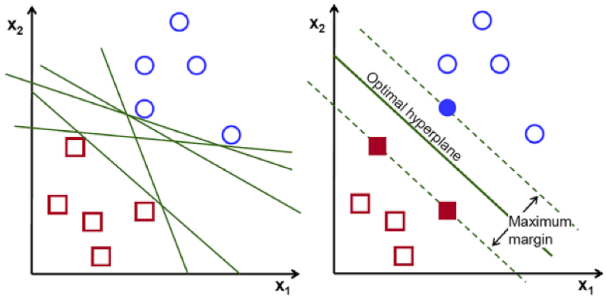
C. *Machine learning*

Machine learning (pembelajaran mesin) adalah bidang ilmu komputer yang berkaitan dengan pengembangan teknologi yang dapat belajar dari data[13]. Lebih spesifik, *machine learning* mengacu pada penggunaan algoritma dan model statistik untuk melatih komputer agar dapat mengambil keputusan atau melakukan tugas tertentu dengan memanfaatkan data yang telah diberikan. Dalam *machine learning*, komputer dilatih dengan memberikan contoh data dan memberikan umpan balik untuk membantu mesin belajar dan meningkatkan kinerjanya[13]. *Machine learning* telah digunakan dalam berbagai bidang, termasuk pengenalan wajah, pengenalan suara, analisis sentimen, dan deteksi anomali[14].

Metode *machine learning* yang paling umum digunakan adalah *supervised learning, unsupervised learning*, dan *reinforcement learning*[13]. *Supervised learning* adalah metode di mana mesin dilatih dengan memberikan contoh data yang telah dilabeli. Contoh-contoh ini kemudian digunakan untuk melatih mesin agar dapat mengklasifikasikan data baru yang belum diberi label. *Unsupervised learning*, di sisi lain, melibatkan penggunaan data yang tidak dilabeli dan berfokus pada pencarian pola dan hubungan dalam data. Metode ini berguna dalam analisis *clustering* dan reduksi dimensi. *Reinforcement learning* adalah metode di mana mesin belajar dari tindakan yang dilakukan dalam lingkungan tertentu dan menerima umpan balik positif atau negatif[13].

D. *Support Vector Machine (SVM)*

Support Vector Machine (SVM) atau Mesin *Vector* Pendukung adalah algoritme pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. *SVM* mengambil pendekatan linier untuk memisahkan kelas-kelas dalam data dengan mencari *hyperplane* yang optimal, atau pembatas kelas, yang memaksimalkan jarak antara data dari kelas yang berbeda[15].



Gambar-3. Ilustrasi Cara Kerja *SVM*

Konsep dasar dari *SVM* adalah mencari *hyperplane* yang memaksimalkan *margin*, yaitu jarak terdekat antara dua kelas pada *dataset*. *Margin* didefinisikan sebagai jarak antara *hyperplane* dan titik terdekat dari kedua kelas. Pada klasifikasi dengan *SVM*, setiap sampel diwakili oleh vektor fitur di ruang fitur dan diplot dalam koordinat dimensi n, di mana n adalah jumlah fitur yang digunakan dalam klasifikasi.

SVM adalah algoritme yang fleksibel dan dapat menangani klasifikasi linear dan non-linear. Untuk kasus klasifikasi non-linear, *SVM* dapat memetakan data ke dimensi yang lebih tinggi menggunakan fungsi *kernel*, yang memungkinkan *SVM* untuk menemukan pembatas kelas yang kompleks dan lebih akurat. *SVM* juga dapat digunakan dalam

masalah regresi dengan memodifikasi fungsi tujuan dan pembatas[15].

Kelebihan dari *SVM* adalah kemampuannya dalam menangani *dataset* yang berukuran besar dan kompleks, serta memiliki performa yang baik dalam memprediksi kelas data baru. *SVM* juga dapat menangani data yang tidak seimbang dengan baik. Namun, kelemahan dari *SVM* adalah kurangnya interpretasi makna yang jelas dari model, serta waktu komputasi yang memakan waktu jika digunakan pada *dataset* yang sangat besar.

Beberapa aplikasi dari *SVM* termasuk pengenalan wajah, pengenalan tulisan tangan, klasifikasi dokumen teks, pengenalan objek dalam citra, dan klasifikasi genom[16].

E. *Scikit Learn*

Gambar-4. Logo *Scikit Learn*

Scikit-learn atau *sklearn* adalah sebuah perpustakaan *open source* yang digunakan untuk keperluan *machine learning* pada bahasa pemrograman *Python*. Perpustakaan ini menyediakan berbagai fungsi dan algoritma untuk tugas-tugas seperti klasifikasi, regresi, pengelompokan, pengurutan, dan reduksi dimensi[17].

Scikit-learn juga menyediakan berbagai *dataset* dan fitur seperti data *preprocessing*, *cross-validation*, *hyperparameter tuning*, dan evaluasi model untuk membantu proses pemodelan. *Scikit-learn* mudah digunakan, terintegrasi dengan baik, stabil dan handal, dan *open source*. Beberapa algoritma *machine learning* yang disediakan oleh *scikit-learn* antara lain *SVM*, *k-NN*, *Naive Bayes*, *Decision Trees*, *Random Forests* untuk klasifikasi, *Linear Regression*, *Logistic Regression* untuk regresi, *K-Means*, *Hierarchical Clustering* untuk pengelompokan, dan *PCA*, *LDA* untuk reduksi dimensi dan *ANOVA*, *Recursive Feature Elimination* untuk seleksi fitur. *Scikit-learn* juga mendukung berbagai teknik untuk mengukur performa model seperti akurasi, presisi, *recall*, *F1 score*, dan *area under ROC curve*[18].

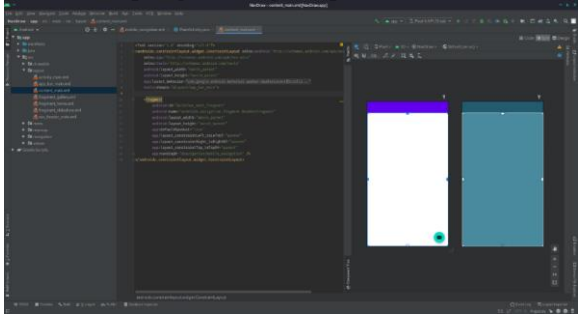
F. *Google Colaboratory*

Gambar-5. Logo *Google Collaboratory*

Google Colab adalah layanan *notebook Jupyter* yang di-*hosting* oleh *Google Research* yang memungkinkan pengguna untuk menulis dan mengeksekusi kode *Python* melalui *browser*. *Colab* sangat cocok untuk *machine learning*, analisis data, dan pendidikan karena memberikan akses gratis ke sumber daya komputasi termasuk *GPU*. *Colab* bekerja pada *cloud* sehingga tidak memerlukan ruang penyimpanan untuk meng-*install Colab*. Pengguna hanya perlu memiliki akun *Google* untuk mengakses *Colab* dan membuat *notebook* baru dengan format *Python 3*. Selain itu, *Colab* juga dapat dihubungkan dengan situs *GitHub* yang digunakan oleh *programmer* dalam mengelola kode atau proyek mereka. [19]

G. *Android Studio*

Android Studio adalah *IDE* untuk mengembangkan perangkat lunak *Android* yang dipublikasikan oleh *JetBrains' IntelJ IDEA*. Awalnya hanya menggunakan bahasa pemrograman *Java*, tetapi sekarang sudah mendukung *Java* dan *Kotlin*. *Android Studio* memiliki fasilitas yang baik dan menggabungkan *code editor* dengan *developer tools*. Fitur seperti *Gradle*, *emulator*, *template* kode, dan integrasi *Github* digunakan untuk mendukung pengembangan aplikasi *Android*. Aplikasi yang dibangun di *Android Studio* dikompilasi menjadi *APK* untuk diunggah ke *Google Play Store*. *Android Studio* diumumkan pada Mei 2013, dirilis pada Desember 2014, dan tersedia untuk *Mac*, *Windows*, dan *Linux*. *Android Studio* menggantikan *Eclipse Android Development Tools (ADT)* sebagai *IDE* utama untuk pengembangan aplikasi *Android* [20].



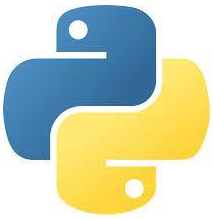
Gambar-6. *Interface* dari *Android Studio*

H. *Chaquopy*

Gambar-7. Logo *Chaquopy*

Chaquopy adalah sebuah alat pengembangan perangkat lunak yang memungkinkan pengembang untuk menggunakan *Python* untuk mengembangkan aplikasi *Android* dengan mudah, termasuk untuk memanipulasi file, mengambil gambar dan mengimplementasikan model *machine learning* ke dalam aplikasi *Android* tanpa API. Alat ini dapat diintegrasikan dengan *Android Studio* dan memungkinkan pengembangan aplikasi menjadi lebih cepat dan fleksibel dengan memanfaatkan sintaks yang mudah dipahami serta banyak pustaka dan modul yang tersedia di *Python*.

I. *Python Programming Language*



Gambar-8. Logo *Python*

Python adalah bahasa pemrograman tingkat tinggi yang diinterpretasikan, digunakan untuk berbagai aplikasi, termasuk pengembangan web, komputasi ilmiah, dan kecerdasan buatan. Bahasa ini pertama kali dibuat pada akhir tahun 1980-an oleh Guido van Rossum dan sejak itu menjadi salah satu bahasa pemrograman paling populer di dunia karena kemudahan dan kepraktisannya. Populeritas *Python* juga didorong oleh komunitas besar dan aktif, yang telah berkontribusi pada banyak perpustakaan dan kerangka kerja untuk berbagai tujuan.

Sebastian Raschka dan Vahid Mirjalili dalam makalah *"Python and Machine learning: A Practical Approach"* [22] memberikan gambaran tentang kemampuan *Python* untuk pembelajaran mesin dengan menggunakan pustaka *Numpy, pandas,* dan *scikit-learn.* Mereka menekankan pentingnya fleksibilitas dan modularitas *Python* untuk membuat prototipe dan bereksperimen dengan algoritma dan teknik yang berbeda.

Sementara itu, Mohit Sewak dan Rama Krishna Sai Gorthi dalam artikel *"Python for Machine learning: A Survey"* [23] menjelajahi penggunaan *Python* dalam pembelajaran mesin dan memberikan survei komprehensif tentang berbagai pustaka dan alat yang tersedia, termasuk *Numpy*, *pandas*, *matplotlib*, dan *TensorFlow*. Penulis menekankan pentingnya kesederhanaan dan kemudahan penggunaan *Python* yang telah membuatnya menjadi pilihan populer di antara praktisi pembelajaran mesin.

III. METODE PENELITIAN

A. Tempat Penelitian

Penelitian dan penyusunan tugas akhir ini dilaksanakan di Laboratorium Biomedis Teknik Elektro Universitas Jenderal Soedirman di Jalan Mayjen Sungkono KM. 5, Blater, Kalimanah, Purbalingga.

B. Alat dan Bahan

Alat dan bahan yang akan digunakan sebagai penunjang penelitian tugas akhir ini adalah sebagai berikut.

1. *Smartphone Android* dengan kamera
2. Tablet *Android* dengan kamera
3. *Laptop* atau komputer
4. *Microsoft Visual Studio Code*
5. *Google Collaboratory*
6. *Android Studio*
7. *Dataset* berupa citra *Lymphoblast* dan *Myeloblast* dari rumah sakit Prof. Dr. Margono Soekarjo di Purwokerto.

C. Tahap Penelitian

Tahapan-tahapan yang akan dilakukan pada pelaksanaan penelitian adalah sebagai berikut

1) Tahap Studi Literatur

Dilakukannya kegiatan studi literatur ini bertujuan untuk melakukan identifikasi dari teori-teori dasar seperti: tinjauan literasi tentang algoritma *Support Vector Machine* (*SVM)* untuk *image classification*, membuat aplikasi *Android* dengan *software Android Studio, framework-framework* yang digunakan dan hubungan antara fitur atau ciri yang satu dengan yang lain.

2) Tahap Persiapan

Tahap persiapan merupakan tahapan awal dalam melakukan persiapan penelitian yang meliputi latar belakang masalah, identifikasi masalah, merumuskan masalah, dan menyusun pra-proposal. Data yang telah didapat dari studi literatur lalu dipergunakan untuk merancang sebuah model.

3) Tahap Perancangan

Tahap perancangan merupakan tahap pengumpulan/pengambilan *dataset* untuk kemudian digunakan dalam merancang dan menyusun model klasifikasi gambar dengan menggunakan arsitektur *Support Vector Machine (SVM)*. Dalam tahap penelitian ini pula model klasifikasi gambar dirancang dan dibangun serta diimplementasikan ke dalam aplikasi *Android* sederhana.

4) Tahap Pengujian

Tahap pengujian merupakan tahap di mana semua model dan aplikasi sudah siap digunakan kemudian dilakukan pengujian untuk mengetahui unjuk kerja dari model dan aplikasi yang dibuat. *Output* dari tahap ini dapat berbentuk tabel atau *statement* yang menunjukkan berhasil atau tidaknya model dan aplikasi yang dibuat.

5) Tahap Akhir

Tahap akhir merupakan tahap penyusunan laporan penelitian yang berisi hasil dari analisis data pada tahap sebelumnya yang sudah diuji. Lalu diakhiri dengan memberi kesimpulan dan saran.

D. Waktu dan Jadwal Penelitian

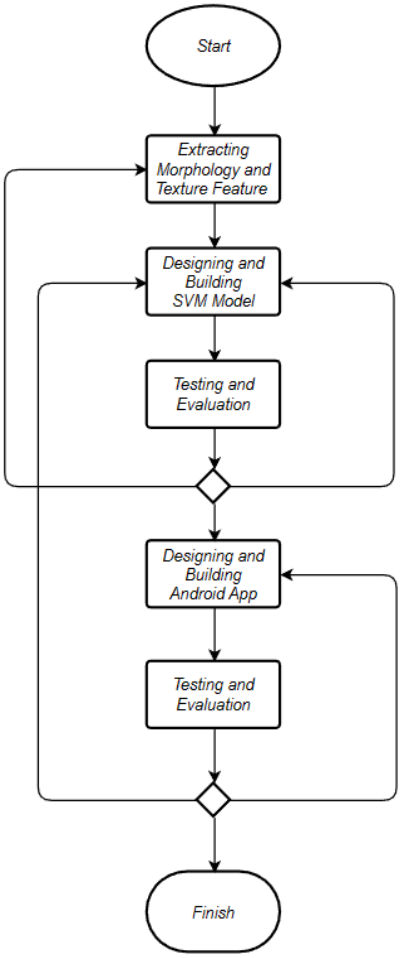
Penelitian tugas akhir ini akan dilaksanakan selama 4 bulan dari bulan Desember 2022 – Maret 2022. Adapun rincian rencana jadwal kegiatan dapat dilihat pada tabel berikut.

Tabel-1. Rencana Jadwal Kegiatan

Nama Kegiatan			Studi Literarur	Tahap Persiapan	Tahap Perancangan	Tahap Pengujian	Tahap Akhir
Bulan Ke-	I	1	■	■			
		2	■	■			
		3	■	■			
		4	■	■			
	II	1	■		■		
		2	■		■		
		3	■		■		
		4	■		■		
	III	1	■		■	■	
		2	■		■	■	
		3	■			■	
		4	■			■	
	IV	1	■			■	■
		2	■			■	■
		3	■				■
		4	■				■

E. Alur Penelitian

Alur penelitian tugas akhir ini digambarkan dalam sebuah diagram alir sebagai berikut.

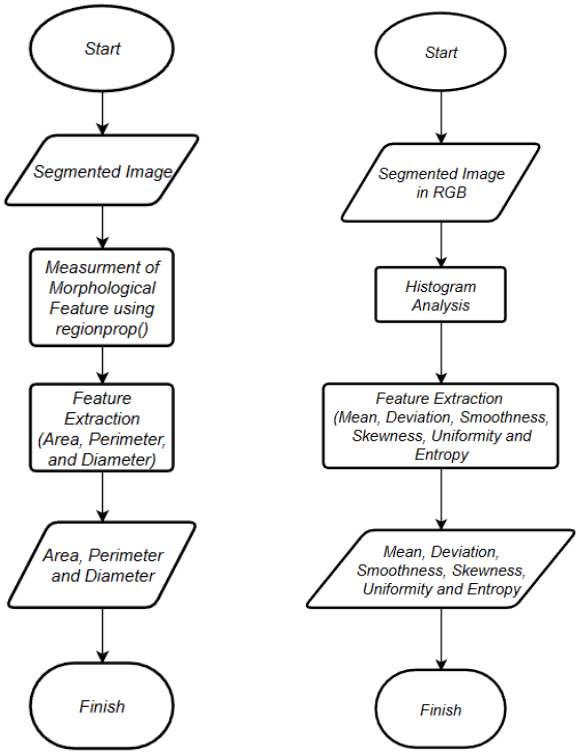


Gambar-9. *Flowchart* Penelitian

Setiap proses pada *flowchart* tersebut dapat dilihat secara lebih detail melalui *flowchart* berikut.

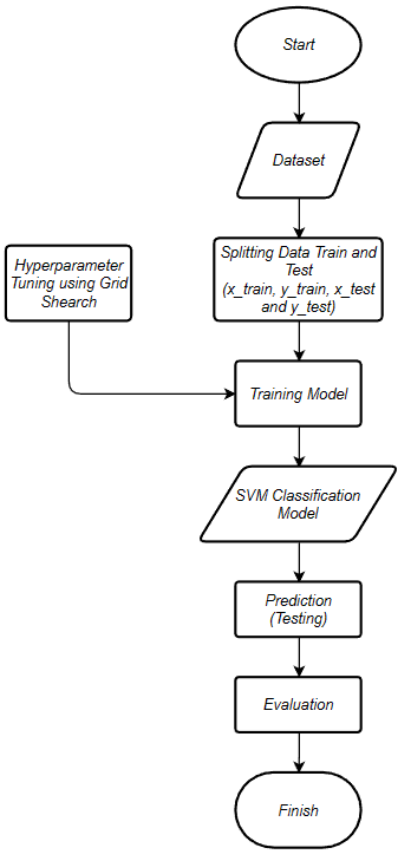
a) Ekstraksi Fitur

Ekstraksi fitur dibagi menjadi dua bagian yaitu ekstraksi fitur morfologi dan fitur tekstur.



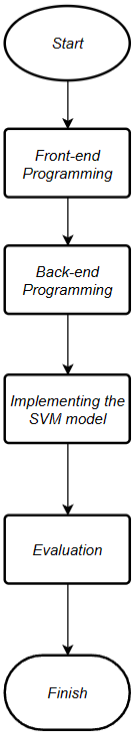
Gambar-10. Ekstraksi Fitur Morfologi (kiri) dan Fitur Tekstur (kanan)

b) Rancang Bangun Model Klasifikasi *Support Vector Machine (SVM)*



Gambar-10. Rancang Bangun Model *SVM*

c) Rancang Bangun Aplikasi *Mobile Android*

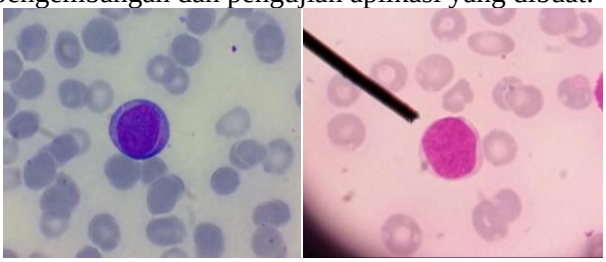


Gambar-11. Rancang Bangun Aplikasi *Mobile Android*

IV. HASIL DAN PENELITIAN

A. *Data Collection* (Pengambilan Data)

Pada sub-bab ini, peneliti menjelaskan teknik pengambilan data untuk *dataset* yang digunakan dalam pembuatan model *machine learning*. *Dataset* ini terdiri dari citra mikroskopis *Lymphoblast* (11 citra) dan *Myeloblast* (44 citra) yang diambil dari Laboratorium Hematologi Rumah Sakit Dr. Margono. Dalam penelitian ini, sebanyak 11 citra *Lymphoblast* dan 15 citra *Myeloblast* akan digunakan sebagai data latih dan data uji dalam pembuatan model *machine learning SVM*. Sementara sisanya akan digunakan untuk pengembangan dan pengujian aplikasi yang dibuat.



Gambar-12. Citra Mikroskopis *Lymphoblast* (kiri) dan *Myeloblast* (kanan)

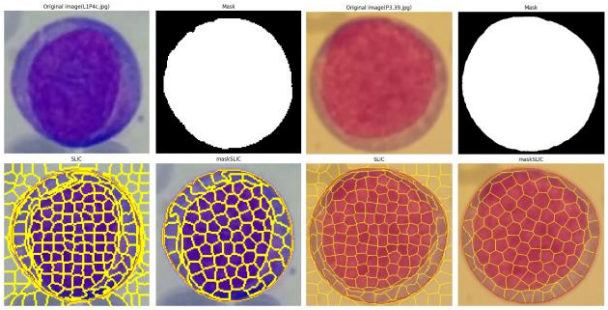
Setelah *dataset* citra terkumpul, dilakukan pengambilan data *(data collection)* untuk membedakan antara dua kelas, *Lymphoblast* dan *Myeloblast*. Proses ini melibatkan dua tahap utama: segmentasi citra *(image segmentation)* dan ekstraksi fitur *(feature extraction)*.

1) *Image Segmentation* (Segmentasi Citra)

Proses segmentasi citra adalah langkah awal dalam pengambilan data, di mana objek pada citra dipisahkan berdasarkan warna, tekstur, dan bentuk. Metode *threshold* digunakan dalam penelitian ini, dengan penentuan nilai ambang batas secara manual untuk menghasilkan citra tersegmentasi yang optimal. Algoritma *threshold* satu layer digunakan dalam penelitian ini untuk memberikan kontrol manual terhadap nilai *threshold*. Keputusan ini didasarkan pada evaluasi karakteristik *dataset*, di mana algoritma satu layer lebih sesuai daripada *multilayer threshold*. Selain itu, algoritma *mask SLIC* digunakan untuk visualisasi akurasi hasil segmentasi dengan memanfaatkan *mask* yang dihasilkan dari algoritma *threshold*. Bahasa pemrograman *Python* dan pustaka *open-source* seperti *Scikit-image* dan *Numpy* digunakan dalam program segmentasi citra ini.

Pada kode program untuk segmentasi citra dengan metode *threshold* ini, citra berwarna dikonversi menjadi citra *grayscale* menggunakan fungsi *"rgb2gray"* dari pustaka *Scikit-image*. Citra *grayscale* tersebut kemudian diproses untuk menghasilkan *mask* biner dengan menghapus objek kecil dan lubang kecil menggunakan fungsi *"remove_small_objects"* dan *"remove_small_holes"*. Nilai ambang batas *"lum"* dapat disesuaikan untuk mendapatkan hasil segmentasi yang diinginkan. Setelah itu, dilakukan operasi pembukaan morfologi pada *mask* biner menggunakan elemen struktural berbentuk *disk* dengan perintah *"morphology.disk()"*.

Selanjutnya, dilakukan visualisasi segmentasi citra menggunakan algoritma *SLIC* dengan segmentasi *mask*. Algoritma *SLIC* membagi citra menjadi beberapa segmen dengan jumlah segmen yang ditentukan oleh *"n_segments"*. "*Mask*" digunakan untuk membatasi piksel yang termasuk dalam setiap segmen. *"Start_label"* adalah label awal untuk segmen pertama yang dihasilkan. Hasil akhirnya adalah matriks label *"m_SLIC"* yang menunjukkan segmen mana setiap piksel pada citra masuk ke dalamnya. Berikut adalah contoh hasil segmentasi menggunakan algoritma *SLIC*.



Gambar 13. Segmentasi Citra *Lymphoblast* (kiri) dan *Myeloblast* (kanan)

2) *Feature Extraction* (Ekstraksi Fitur)

Setelah segmentasi citra, dilakukan ekstraksi fitur untuk mengambil informasi penting dari citra leukosit yang telah disegmentasi dengan metode *threshold*. Hasil ekstraksi fitur ini disusun dalam file *spreadsheet* berformat **xlsx* sebagai *dataset*. *Dataset* tersebut akan digunakan sebagai *input* dalam pemodelan *machine learning* untuk klasifikasi. Berikut adalah hasil dari tahap ekstraksi fitur.

a. *Morphology Feature Extraction* (Ekstraksi Fitur Morfologi)

Pada tahap ekstraksi fitur morfologi, dilakukan perhitungan fitur-fitur morfologi dari citra leukosit yang telah disegmentasi, seperti area, perimeter dan diameter. Proses ini membantu dalam mengekstraksi informasi tentang bentuk dan ukuran dari citra leukosit.

Peneliti menggunakan pustaka *open-source Scikit-image* dalam bahasa pemrograman *Python* dan *IPython* untuk memudahkan pengkodean. Fungsi *"regionprops"* dari *Scikit-image* digunakan untuk menghitung berbagai properti objek dalam citra biner atau label, seperti *area*, *perimeter*, *centroid*, dan *convex hull*. Fungsi ini mengambil citra biner atau label sebagai *input* dan mengembalikan objek *"RegionProperties"* yang berisi properti dari setiap objek dalam citra.

Dalam ekstraksi fitur morfologi, langkah-langkah yang dilakukan adalah sebagai berikut: Pertama, citra biner "*mask*" dihitung menggunakan fungsi *"label"* dari modul *"measure"* di *Scikit-image* untuk mengidentifikasi komponen terhubung. Selanjutnya, fungsi *"regionprops"* digunakan pada citra terlabel *"labels"* untuk menghitung properti wilayah dari setiap komponen terhubung, seperti luas, *centroid*, dan *bounding box*. Kemudian, indeks komponen terhubung terbesar ditentukan dengan mencari nilai

maksimum dari atribut *"area"* menggunakan fungsi *"argmax"* dari *Numpy*. Properti dari komponen terhubung terbesar disimpan dalam variabel *"biggest_props"*, yang mencakup informasi seperti area, diameter, dan perimeter. Sehingga dapat didapat nilai-nilai fitur morfologi yang dibutuhkan yaitu area, perimeter dan diameter.

b. *Texture Feature Extraction* (Ekstraksi Fitur Tekstur)

Ekstraksi fitur tekstur melibatkan perhitungan beberapa fitur tekstur dari citra leukosit yang telah disegmentasi, seperti korelasi, kontras, dan energi. Tujuan dari proses ini adalah untuk mengungkap informasi tentang tekstur dari citra leukosit tersebut.

Untuk melakukan ekstraksi fitur tekstur, digunakan diagram histogram dari citra *grayscale* tersegmentasi. Histogram ini digunakan untuk mendapatkan nilai-nilai yang kemudian dihitung menggunakan formula matematis, sehingga mendapatkan nilai fitur yang diperlukan.

Dalam penelitian ini digunakan beberapa fitur tekstur *yaitu mean, deviation, smoothness, skewness, uniformity* dan *entropy*. Di mana dalam mendapatkan data ini peneliti melakukan kalkulasi dengan beberapa rumus terhadap nilai histogram yang didapat. Berikut beberapa rumus tersebut.

- *Mean*

$$mean = \sum_{i=0}^{L-1} i \cdot hist_i \quad (1)$$

- *Deviation*

$$d = \sqrt{\sum_{i=0}^{L-1} (i - mean)^2 \cdot hist_i} \quad (2)$$

- *Smoothness*

$$smoothness = 1 - \frac{1}{1 + d^2} \quad (3)$$

- *Skewness*

$$skewness = \frac{\sum_{i=0}^{L-1} (i - mean)^3 \cdot hist_i)}{d^3} \quad (4)$$

- *Uniformity*

$$U = \sum_{i=0}^{L-1} hist_i^2 \quad (5)$$

- *Entropy*

$$Entropy = -\sum_{i=1}^{L} hist_i * \log_2 hist_i \quad (6)$$

3) *Data Normalization* (Normalisasi Data)

Normalisasi data pada *dataset machine learning* adalah proses transformasi data *input* agar nilainya berada dalam rentang atau skala tertentu yang mempermudah proses pembelajaran oleh algoritma. Hal ini dilakukan untuk menghindari masalah numerik dan meningkatkan akurasi serta kecepatan proses pembelajaran.

Terdapat beberapa teknik normalisasi yang umum digunakan, seperti *min-max scaling, z-score scaling*, dan *unit vector scaling*. Selain itu, normalisasi dapat dilakukan secara manual dengan menentukan faktor normalisasi sesuai karakteristik *dataset*. Pilihan teknik normalisasi tergantung pada jenis data dan tujuan analisis.

Dalam penelitian ini, peneliti menggunakan normalisasi manual dengan menentukan faktor normalisasi berdasarkan penelitian sebelumnya. Data akan dinormalisasi agar setiap fitur memiliki rentang nilai antara 0 hingga 1. Faktor normalisasi ini didapatkan dengan membagi nilai asli fitur tertinggi dengan suatu bilangan tertentu. Berikut faktor normalisasi yang digunakan.

- *Area*

$$Area = \frac{n}{10000000} \quad (7)$$

- *Diameter*

$$Diameter = \frac{n}{10000} \quad (8)$$

- *Perimeter*

$$Perimeter = \frac{n}{10000} \quad (9)$$

- *Mean*

$$Mean = \frac{n}{1000} \quad (10)$$

- *Deviation*

$$Deviation = \frac{n}{100} \quad (11)$$

- *Smoothness*

$$Smoothness = \frac{n}{1} \quad (12)$$

- *Skewness*

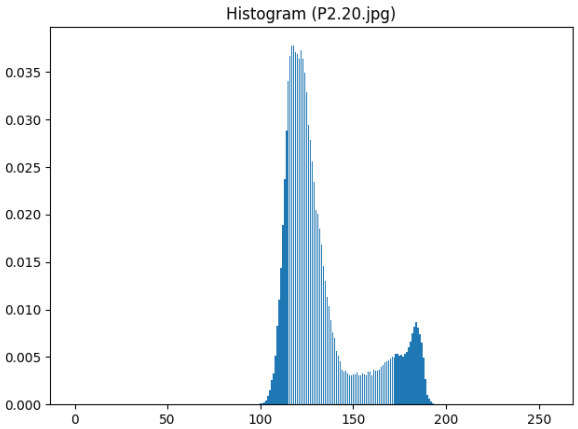
$$Skewness = \frac{n}{1} \quad (13)$$

- *Uniformity*

$$Uniformity = \frac{n}{1} \quad (14)$$

- *Entropy*

$$Entropy = \frac{n}{10} \quad (15)$$



Gambar-14. Sampel Histogram (P2.20.jpg)

Nama	Area	Perimeter	Diameter	Mean	Deviation	Smoothness	Skewness	Uniformity	Entropy	Kategori
L1P3c.jpg	0,0008906	0,038326198	0,0106487	0,026819	0,2500147	0,001597257	0,4593562	0,1611885	0,451253219	Lymphoblast
L1P5c.jpg	0,0009175	0,036819091	0,0108083	0,035177	0,3301233	0,000916747	0,540638	0,14195388	0,484045405	Lymphoblast
L1P7c.jpg	0,0007381	0,034804877	0,0096942	0,037357	0,2766463	0,001304918	-0,2921359	0,10663586	0,48971526	Lymphoblast
L1P9c.jpg	0,0007296	0,033029141	0,0096382	0,031699	0,2692836	0,001377151	0,0407426	0,14402851	0,466024209	Lymphoblast
L1P4c.jpg	0,0010698	0,038974726	0,011671	0,037826	0,3508864	0,000811548	0,443313	0,14448716	0,487045077	Lymphoblast
L1P8c.jpg	0,0010175	0,03820904	0,0113821	0,037673	0,3224241	0,000961009	0,2380822	0,13189632	0,471356444	Lymphoblast
L1P1c.jpg	0,0006592	0,030697771	0,0091614	0,029219	0,2727079	0,00134283	0,5057351	0,16221295	0,443342695	Lymphoblast
L1P10c.jpg	0,0005652	0,030214928	0,0084831	0,033772	0,3409093	0,000859704	0,6870062	0,17928577	0,46438555	Lymphoblast
P2.20.jpg	0,0000448	0,011253301	0,0023883	0,094509	0,6207874	0,000259419	-0,6656529	0,09279332	0,502335891	Myeloblast
P3.39.jpg	0,0183437	0,168131493	0,048328	0,070481	0,4887712	0,000418415	-0,5512057	0,10492169	0,485817323	Myeloblast
P3.35.jpg	0,000045	0,009397666	0,0023937	0,053285	0,352716	0,000803158	-0,6697724	0,10075421	0,45763258	Myeloblast
P3.45.jpg	0,0204511	0,182075353	0,0510286	0,076436	0,4619828	0,000468323	-0,7282233	0,06912785	0,53353375	Myeloblast
F0.JPG	0,111031	0,532040274	0,1188987	0,057405	0,4306183	0,00053899	-0,2240262	0,10185135	0,522258689	Myeloblast
P1S3-4s.jp	0,0019239	0,054197265	0,0156512	0,053277	0,462042	0,000468202	0,5200088	0,10878755	0,501044999	Myeloblast
P3.32.jpg	0,016304	0,158933218	0,0455619	0,078913	0,5004432	0,000399132	-0,6704794	0,08093542	0,520892314	Myeloblast
P3.28.jpg	0,0172908	0,169481241	0,0469205	0,08538	0,5475544	0,000333427	-0,7505373	0,08815919	0,506425525	Myeloblast
P1S2s.jpg	0,0041086	0,088992092	0,0228719	0,072435	0,527052	0,000359863	-0,1057597	0,08712394	0,528948099	Myeloblast
P3.43.jpg	0,0175157	0,202942258	0,0472247	0,064404	0,4098508	0,000594963	-0,6831765	0,08683578	0,484361423	Myeloblast
P3.38.jpg	0,0000766	0,012739087	0,003123	0,064146	0,3915025	0,000652	-0,7242294	0,07463504	0,515676677	Myeloblast

Gambar-15. Sampel *Dataset* yang Didapatkan

B. *Machine learning Modeling*

1) Implementasi *Support Vector Machine* (*SVM*)

*Support Vector Machine*s (*SVM*) adalah metode klasifikasi pada *machine learning* yang memprediksi kelas data dengan membagi data menjadi dua kelas berdasarkan margin dari *hyperplane* atau *decision boundary*. *SVM* dapat diimplementasikan menggunakan pustaka *Scikit-learn* di *Python* dengan fungsi 'SVC' untuk klasifikasi dua kelas biner.

SVM memiliki *hyperparameter* yang dapat diatur untuk mempengaruhi performa model. Beberapa *hyperparameter* yang digunakan dalam penelitian ini adalah:

- *Kernel*: *SVM* dapat menggunakan *kernel* seperti *linear*, *polynomial*, *rbf*, dan *sigmoid*. Pilihan *kernel* ini mempengaruhi pemetaan data dari ruang *input* ke ruang fitur yang lebih tinggi.
- C: *Hyperparameter* ini mengontrol seberapa besar margin yang diterima oleh *SVM*. Nilai C yang lebih besar memungkinkan margin yang lebih besar, tetapi dapat menyebabkan *overfitting*.
- *Gamma*: *Hyperparameter* ini mempengaruhi pengaruh data *point* pada *SVM*. Nilai *gamma* yang lebih besar memberikan pengaruh yang lebih besar pada data *point* yang lebih dekat.

Penyetelan *hyperparameter* dapat dilakukan secara manual atau otomatis. *Tunning* manual melibatkan coba-coba untuk menemukan nilai *hyperparameter* yang baik, sementara *tunning* otomatis dapat dilakukan dengan menggunakan teknik *"grid search"* yang disediakan oleh *Scikit-learn*. *Grid search* mencoba semua kombinasi parameter yang ditentukan pada *grid* untuk menemukan kombinasi terbaik yang memberikan performa optimal.

```
Best Score: 1.0
Best Parameters:
        C:0.1
        gamma:10
        kernel:poly
```

Gambar-16. Akurasi dan Kombinasi *hyperparameter* terbaik

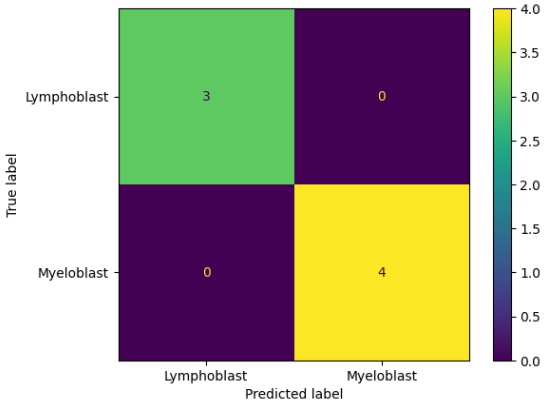
Setelah dilakukan pelatihan menggunakan algoritma *SVM* dan *grid search* didapati hasil *training* seperti pada gambar 16 di atas. Dari gambar di atas, dapat disimpulkan bahwa kombinasi terbaik dari *hyperparameter SVM* adalah *kernel polynomial*, *gamma* dengan nilai 10, dan C dengan nilai 0,1. Dengan menggunakan kombinasi *hyperparameter* tersebut, model *SVM* mencapai akurasi pelatihan sebesar 1, yang menunjukkan kualitas yang sangat baik. Model *SVM* ini dapat langsung digunakan untuk melakukan prediksi.

2) *Model Evaluation* (Evaluasi Model)

Evaluasi model dilakukan untuk mengukur kinerja model *machine learning* dalam memprediksi atau mengklasifikasikan data. Evaluasi ini melibatkan perhitungan berbagai metrik performa seperti akurasi, presisi, *recall*, *F1-score*, dan *confusion matrix*. Hasil evaluasi tersebut digunakan untuk menentukan sejauh mana model telah mencapai kualitas yang diinginkan dan apakah perlu dilakukan optimasi lebih lanjut. Evaluasi model juga membantu dalam menilai kegunaan model di lingkungan produksi.

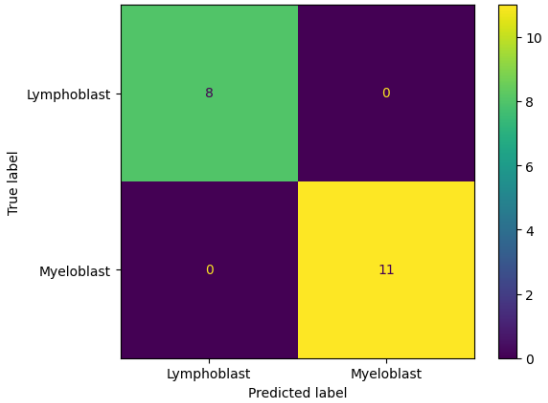
Dalam penelitian ini, terdapat dua metode evaluasi model yang digunakan: analisis *confusion matrix* dan *classification report*. Evaluasi dilakukan pada data uji menggunakan model yang telah dibuat. Data uji terdiri dari 3 data *Lymphoblast* dan 4 data *Myeloblast*. Untuk melakukan prediksi pada data uji, dapat digunakan fungsi *"prediction"* yang disediakan oleh *Scikit-learn* melalui variabel *"grid_search"*.

a. *Confusion matrix*



Gambar-17. *Confusion matrix* Data Uji

Berdasarkan *confusion matrix* tersebut, dapat disimpulkan bahwa terdapat 3 data uji *Lymphoblast* yang berhasil diklasifikasikan dengan benar (*True Positive*/*TP*) dan 4 data uji *Myeloblast* yang juga berhasil diklasifikasikan dengan benar (*True Negative*/*TN*). Tidak ada data yang salah diklasifikasikan sebagai *Lymphoblast* atau *Myeloblast* (*False Positive*/*FP* dan *False Negative*/*FN* bernilai 0). Dengan demikian, akurasi model berdasarkan *confusion matrix* tersebut adalah 1 (sempurna).



Gambar-18. *Confusion matrix* Data Latih

Berdasarkan *confusion matrix* tersebut, dapat disimpulkan bahwa model berhasil mengklasifikasikan semua data latih *Lymphoblast* dan *Myeloblast* dengan benar, dengan nilai *True Positive* (TP) sebesar 8 dan *True Negative* (TN) sebesar 11. Tidak ada data yang salah diklasifikasikan sebagai *Lymphoblast* atau *Myeloblast* (*False Positive*/FP dan *False Negative*/FN bernilai 0). Dengan demikian, akurasi pelatihan model *SVM* ini adalah 1 (sempurna).

Berdasarkan hasil evaluasi tersebut, dapat disimpulkan bahwa model *SVM* yang dibuat bekerja dengan sangat baik dan memiliki akurasi yang sangat tinggi. Model ini siap untuk diimplementasikan dalam aplikasi yang relevan.

b. *Classification report*

	precision	recall	f1-score	support
Lymphoblast	1.00	1.00	1.00	3
Myeloblast	1.00	1.00	1.00	4
accuracy			1.00	7
macro avg	1.00	1.00	1.00	7
weighted avg	1.00	1.00	1.00	7

Gambar-19. *Classification report* Data Uji

Hasil evaluasi dari model *SVM* terhadap data uji menunjukkan performa yang sangat baik. Nilai *precision*, *recall*, dan *F1-score* untuk kedua kelas ("*Lymphoblast*" dan "*Myeloblast*") adalah 1.00, menandakan bahwa model telah mengklasifikasikan semua data dengan benar tanpa ada kesalahan dalam prediksi. Kedua kelas memiliki jumlah data yang berbeda, dengan "*Lymphoblast*" memiliki 3 data dan "*Myeloblast*" memiliki 4 data. Akurasi model

adalah 1.00, menunjukkan bahwa model mampu mengklasifikasikan semua data dengan benar.

	precision	recall	f1-score	support
Lymphoblast	1.00	1.00	1.00	8
Myeloblast	1.00	1.00	1.00	11
accuracy			1.00	19
macro avg	1.00	1.00	1.00	19
weighted avg	1.00	1.00	1.00	19

Gambar-20. *Classification report* Data Latih

Model klasifikasi *SVM* berhasil mencapai performa yang sangat baik dalam pelatihan dan evaluasi. Dalam kedua *classification report*, nilai *precision*, *recall*, dan *F1-score* untuk kedua kelas ("*Lymphoblast*" dan "*Myeloblast*") adalah 1.00, menunjukkan bahwa model dapat mengklasifikasikan semua data dengan benar tanpa ada kesalahan. Akurasi model juga mencapai 1.00, yang menunjukkan bahwa model berhasil mengklasifikasikan seluruh data latih dengan benar.

3) Penyimpanan Model *Machine learning*

Setelah melatih dan menguji performa model *machine learning*, langkah selanjutnya adalah menyimpan model agar dapat digunakan di waktu dan tempat lain tanpa harus melatih ulang. Dalam penelitian ini, peneliti menggunakan modul *Python* yang disebut *"Pickle"* untuk menyimpan model *SVM* dalam format file dengan ekstensi **.pkl*.

Pickle adalah modul *Python* yang digunakan untuk serialisasi dan deserialisasi objek. Dengan *Pickle*, objek dapat diubah menjadi bentuk yang dapat disimpan atau ditransmisikan, dan kemudian dikembalikan ke bentuk aslinya. Dalam hal ini, peneliti mengimpor modul *pickle* dan menggunakan fungsi *"pickle.dump()"* untuk menyimpan objek *"grid_search"*, yang merupakan model *machine learning* yang telah dilatih menggunakan algoritma *grid search.*

Fungsi *"pickle.dump()"* digunakan untuk menyimpan objek serialisasi ke dalam file "model.pkl" dengan mode tulis biner *('wb')*. File ini dapat disimpan, ditransmisikan, atau digunakan untuk memuat kembali objek serialisasi ke dalam memori pada waktu yang berbeda menggunakan fungsi *"pickle.load()"*.

Dengan menyimpan model dalam file "model.pkl", model dapat diakses kembali dan digunakan dalam program *Python* lainnya, termasuk dalam aplikasi *Android* yang akan diimplementasikan.

C. Pengembangan Aplikasi *Mobile Android*

Peneliti mengembangkan aplikasi *mobile Android* menggunakan *Android Studio* untuk melakukan klasifikasi citra sel darah putih *Lymphoblast* dan *Myeloblast* dengan metode *machine learning Support Vector Machine* (*SVM*). Model *SVM* yang dibuat oleh peneliti menggunakan bahasa pemrograman *Python* dan disimpan dalam format file dengan ekstensi .pkl menggunakan modul atau pustaka "*Pickle*". Untuk mengimplementasikan skrip *Python* ke dalam aplikasi *Android*, peneliti menggunakan *Chaquopy* sebagai *plug-in* eksternal untuk *Android Studio*. Karena bahasa pemrograman yang didukung secara *native* oleh *Android Studio* adalah *Java* dan *Kotlin*. Aplikasi *mobile Android* yang dibuat dinamakan "*LIFE (Leukocyte Identifier From Everywhere)"*.

1) Pengembangan *UI* dan *Logical*

Dalam pengembangan *UI (front-end)* dan *logical (back-end)* ini dibuat 4 *activity layout XML* dan 4 kelas *activity Java* yang saling berkesinambungan. 4 *activity* tersebut yaitu:

a. *Activity Splash Screen*

Layout activity_splash_screen ini adalah sebuah *layout* yang akan menjadi initial page dari aplikasi yang digunakan. Dalam *Splash screen* hanya memunculkan logo dari aplikasi *LIFE* dengan backgroun berwarna *coastal breeze*. Berikut merupakan tangkapan layar dari *Splash screen layout*.

Gambar-21. *Activity Splash Screen*

b. *Activity Pick Image*

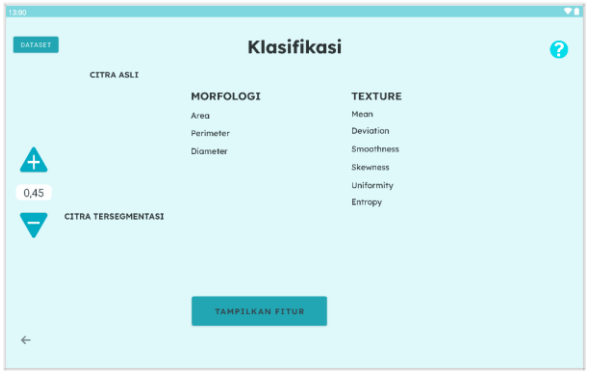
Layout activity_pick_image adalah halaman utama dari aplikasi *LIFE*. Pengguna dapat mengimpor atau memilih citra yang akan diklasifikasikan dan melakukan *preprocessing* citra seperti *cropping*. Citra yang telah di-*cropping* akan ditampilkan pada *ImageView* di sebelah kanan tombol impor. Di *layout* ini juga terdapat tombol bantuan untuk membuka menu

bantuan, tombol "Keluar" untuk keluar dari aplikasi, dan tombol "Lanjut" untuk memindahkan citra yang dipilih ke halaman klasifikasi.

Gambar-22. *Activity Pick Image*

c. *Activity Classification*

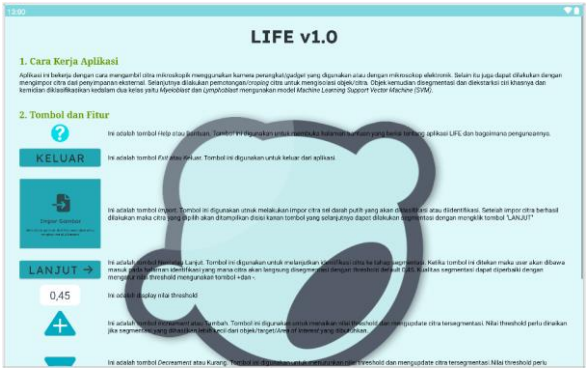
Layout classification berisi fitur untuk segmentasi citra, ekstraksi ciri, dan klasifikasi citra. Segmentasi citra dilakukan dengan mengatur nilai *threshold* menggunakan tombol "+" untuk meningkatkan nilai dan "-" untuk mengurangi nilai. Setelah segmentasi citra dilakukan, pengguna dapat melakukan ekstraksi ciri dan klasifikasi citra dengan menekan tombol "Tampilkan Fitur". Di halaman ini juga terdapat tombol bantuan untuk mengakses halaman bantuan.



Gambar-23. *Activity Classification*

d. *Activity Help*

Layout activity_help ini berisikan halaman bantuan/*help*. Di mana dalam halaman ini terdapat konten berupa informasi yang dapat membantu *user* lebih memahami tentang dan bagaimana cara kerja dari aplikasi *LIFE* ini.



Gambar-24. *Activity Help*

2) Implementasi Skrip *Python* Model *SVM* pada *Android Studio*

Dalam tahap ini, peneliti mengimplementasikan skrip model *machine learning SVM* yang ditulis dalam bahasa *Python* ke dalam *Android Studio* yang menggunakan bahasa *native Java*. Karena *Android Studio* hanya mendukung bahasa *Java* dan *Kotlin* secara *native*, peneliti menggunakan *plug-in* eksternal bernama *Chaquopy*. *Chaquopy* memungkinkan pengembang untuk melakukan pemrograman *Python* dalam bahasa *Java* dengan membuat fungsi-fungsi *Python* yang mengembalikan nilai.

Dengan menggunakan *Chaquopy*, pengembang dapat memanggil fungsi-fungsi *Python* tersebut dari dalam pemrograman *Java* dan menyimpan hasil yang dikembalikan oleh *Python* ke dalam variabel *Java*. Ini memungkinkan integrasi antara skrip *Python* yang mengimplementasikan model *machine learning SVM* dengan kode *Java* dalam proyek *Android Studio*.

a. Pemrograman *Python* untuk Segmentasi dan Klasifikasi

Peneliti membuat program *Chaquopy* untuk segmentasi dan klasifikasi. Di mana, program *Chaquopy* yang dibuat merupakan adaptasi dari program *Python machine learning* yang sudah ada sebelumnya. Program ini disesuaikan dengan kebutuhan dan bagian yang diperlukan. Algoritma yang digunakan dapat dijelaskan seperti berikut.

Fungsi "main" menerima *input* berupa *URI* gambar dan nilai ambang (*threshold*). Prosesnya melibatkan unduhan gambar dari *URI* dan penyimpanannya dalam file sementara "temp.jpg". Gambar kemudian dibuka, diubah menjadi citra *grayscale*, dan dilakukan proses *threshold*ing serta operasi morfologi untuk mendapatkan *mask*. *Mask* hasil *threshold*ing kemudian disaring dengan menghapus lubang

dan objek kecil yang memiliki luas di bawah 500 piksel. Dilakukan juga operasi *opening* pada *mask* untuk menghilangkan detail yang tidak diperlukan.

Selanjutnya, dilakukan pengambilan objek terbesar dari *mask* menggunakan perhitungan properti wilayah. Jika tidak ada objek dalam *mask*, maka *mask_best* akan kosong. Proses selanjutnya melibatkan pengolahan *mask* dengan algoritma MSLIC untuk visualisasi segmentasi citra. Hasil MSLIC dan *mask_best* disimpan dalam format *image string base64* menggunakan fungsi "*saveImage*" yang menggunakan modul *pillow (PIL)* dan *base64*.

Fitur morfologi dan tekstur diekstraksi dari objek tersegmentasi menggunakan operasi pada *array Numpy* dan *Open*CV. Setelah fitur diekstraksi, data dimasukkan ke dalam *DataFrame* dan dimasukkan ke dalam model terlatih yang telah disimpan sebelumnya menggunakan *pickle*. Hal ini akan mengklasifikasikan citra menjadi dua kelas, yaitu *Lymphoblast* dan *Myeloblast*. Jika *mask_best* memiliki nilai 0, maka kelas yang dihasilkan adalah *Unknown*. Kelas prediksi bersama dengan nilai fitur morfologi dan tekstur serta dua citra dalam format base64 dikembalikan sebagai *output* fungsi tersebut.

b. Mengimplementasikan Skrip *Python*

Untuk melakukan ini digunakan *library Chaquopy* untuk mengintegrasikan *Python* ke dalam *Java*. Pertama, objek *Python* diinisialisasi menggunakan *Python.getInstance()*. Kemudian, modul "*threshold*" diimpor menggunakan getModule().

Selanjutnya, fungsi main dalam modul "*threshold*" dipanggil menggunakan *callAttr()* dengan parameter *imageUri.toString()* dan *threshold*. Hasil pemanggilan fungsi tersebut berupa *PyObject* yang kemudian dikonversi menjadi *List<PyObject>* menggunakan *asList()*.

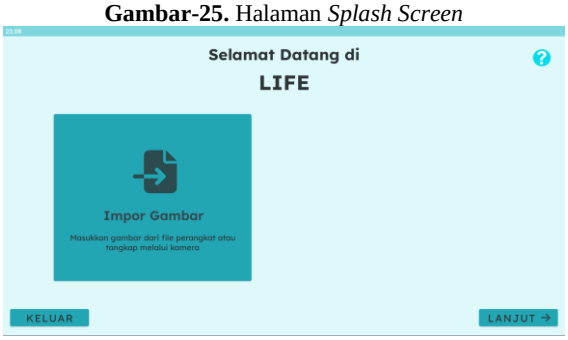
Fungsi main dalam modul "*threshold*" menerima dua argumen, yaitu *URI* gambar dan nilai ambang (*threshold*). Fungsi ini melakukan pengolahan citra sesuai dengan algoritma yang telah dijelaskan sebelumnya. Hasil akhir dari fungsi main berupa *list* yang berisi kelas prediksi, nilai fitur morfologi dan tekstur, serta dua citra dalam format base64.

Setelah nilai *return* yang diharapkan dari fungsi *Python* disimpan dalam *list Java*, nilai-nilai tersebut dapat diakses dan digunakan dalam pemrograman *Java* sesuai kebutuhan.

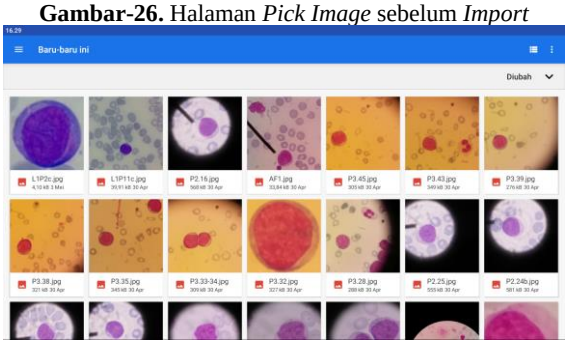
3) Uji Coba dan Evaluasi Aplikasi

a. Uji Coba Aplikasi

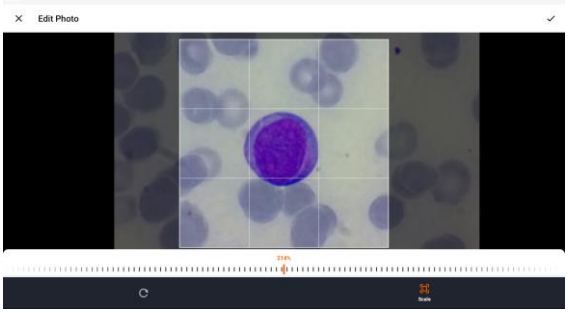
Gambar-25. Halaman *Splash Screen*



Gambar-26. Halaman *Pick Image* sebelum *Import*

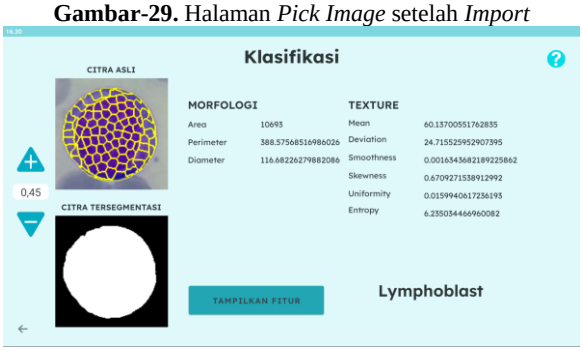


Gambar-27. Menu *Default File Explorer Device*

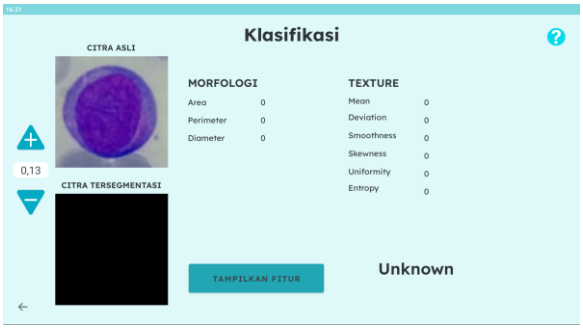


Gambar-28. *Cropping Image*

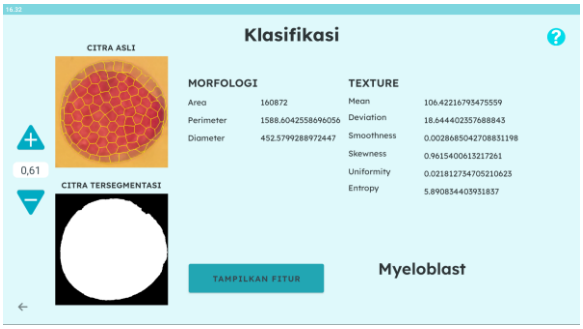
Gambar-29. Halaman *Pick Image* setelah *Import*



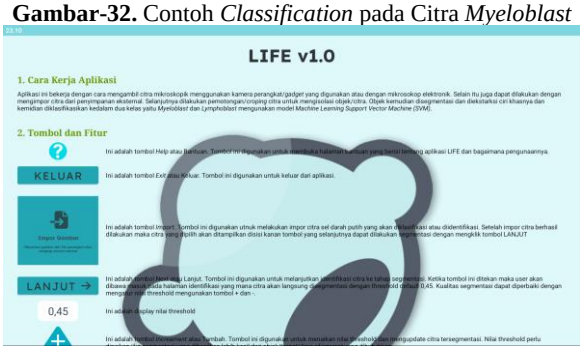
Gambar-30. Halaman *Classification* dengan *Threshold* Sesuai



Gambar-31. Halaman *Classification* saat *Threshold* Terlalu Rendah



Gambar-32. Contoh *Classification* pada Citra *Myeloblast*

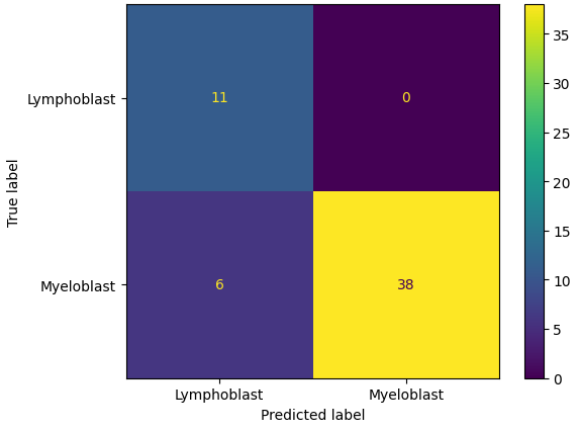


Gambar-33. Halaman Bantuan (*Help*)

Gambar-34. Tampilan Keluar Aplikasi

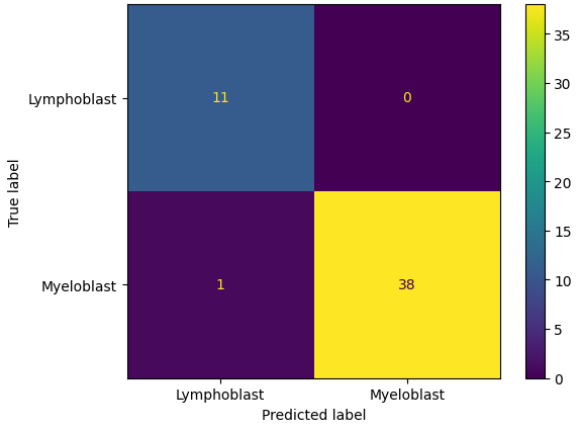
b. Evaluasi Aplikasi

Evaluasi aplikasi dilakukan dengan membuat tabel klasifikasi citra dan *confusion matrix*. Evaluasi ini bertujuan untuk mengetahui akurasi aplikasi dalam mengklasifikasikan citra *Lymphoblast* dan *Myeloblast*. Evaluasi dilakukan dalam dua kondisi: tanpa memperhatikan perbedaan nilai perbesaran mikroskop pada 55 data uji, dan dengan memperhatikan perbedaan nilai perbesaran mikroskop pada 50 data uji.



Gambar-35. *Confusion matrix* Kondisi 1

Pada kondisi pertama ini, dilakukan evaluasi terhadap 55 data uji dengan tanpa memedulikan adanya 5 data yang perbesaran mikroskopnya berbeda dengan data latih. Dari *confusion matrix* di atas diketahui bahwa dari 11 data *Lymphoblast* yang diuji semuanya diklasifikasikan dengan benar. Namun, dari 44 data *Myeloblast*, terdapat 6 data yang salah diklasifikasikan sebagai *Lymphoblast*. Dengan menggunakan rumus akurasi, didapatkan nilai akurasi sebesar 89%. Meskipun akurasi ini cukup baik untuk mengklasifikasikan citra, perlu diperhatikan bahwa aplikasi ini sebaiknya tidak digunakan secara eksklusif, tetapi hanya untuk pemeriksaan pencegahan.



Gambar-36. *Confusion matrix* Kondisi 2

Pada kondisi kedua ini, dilakukan evaluasi terhadap 50 data uji dengan memedulikan nilai perbesaran mikroskopnya. Di mana 50 data ini memiliki perbesaran mikroskop yang sama dengan data latih. Dari *confusion matrix* di atas diketahui bahwa dari 11 data *Lymphoblast* yang diuji semuanya diklasifikasikan dengan benar. Hanya terdapat 1 data *Myeloblast* yang salah diklasifikasikan sebagai *Lymphoblast*. Dengan menggunakan rumus akurasi, didapatkan nilai akurasi sebesar 98%. Akurasi ini sangat baik dan mendekati sempurna (100%).

Maka dari hasil uji coba dan evaluasi yang dilakukan dapat dikatakan bahwa aplikasi ini siap digunakan untuk melakukan klasifikasi citra *Lymphoblast* dan *Myeloblast*. Namun, dalam praktiknya, perlu diperhatikan agar nilai perbesaran mikroskop selalu sama dengan data latih.

D. Perbandingan Akurasi Aplikasi *LIFE* dengan Penelitian Terdahulu

Penelitian sebelumnya menggunakan sistem aplikasi berbasis *Matlab* untuk mengklasifikasikan jenis sel darah putih menggunakan arsitektur *Support Vector Machine* (*SVM*) dan metode *Threshold*. Pada tahap ini, peneliti membandingkan akurasi hasil penelitian sebelumnya dengan akurasi hasil aplikasi *LIFE* yang dibuat menggunakan arsitektur *SVM* dan *Threshold* berbasis *Python* dan *Android*. Pembandingan akan dilakukan pada dua kondisi yang sama seperti pada evaluasi aplikasi yaitu dengan 55 data uji tanpa memedulikan perbedaan perbesaran mikroskop dan dengan 50 data uji dengan memperhatikan nilai perbesaran mikroskop. Berikut merupakan tabel perbandingan yang didapat.

Tabel-1. Tabel Perbandingan Aplikasi *Matlab* dan Aplikasi *LIFE*

Pembanding	Kondisi 1		Kondisi 2	
	Matlab	*Python/LIFE*	*Matlab*	*Python/LIFE*
Jumlah Data	55	55	50	50
True Positive (TP)	11	11	11	11
True Negative (TN)	31	38	31	38
False Positive (FP)	13	1	8	6
False Negative (FN)	0	0	0	0
Accuracy	76%	89%	84%	98%

Dari tabel di atas dapat dipahami bahwa nilai akurasi aplikasi *Matlab* pada kondisi pertama sebesar 76% dan pada kondisi kedua sebesar 84%. Jika dibandingkan dengan nilai akurasi Aplikasi *LIFE* yang dibuat dalam penelitian ini, maka nilai akurasi aplikasi *Matlab* yang diperoleh jauh di bawah dengan nilai akurasi Aplikasi *LIFE* yaitu sebesar 89% untuk kondisi pertama dan 98% untuk kondisi kedua. Selain itu, dapat dipahami bahwa kedua sistem aplikasi yang dibuat belum mampu mengklasifikasikan data dengan benar jika memiliki nilai perbesaran yang berbeda dengan data latih.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini memiliki empat kesimpulan utama. Pertama, metode *threshold* sangat efektif dalam segmentasi citra, terbukti dengan hasil segmentasi yang hampir sempurna dalam memisahkan *area of interest (AOI)* pada citra *Lymphoblast* dan *Myeloblast*. Kedua, model *machine learning* dengan arsitektur *Support Vector Machine* (*SVM*) mampu bekerja dengan sangat baik dalam klasifikasi data penelitian, terlihat dari akurasi yang mencapai 100%

baik pada pelatihan maupun pengujian, menunjukkan representasi fitur citra yang baik. Ketiga, implementasi skrip *Python* ke dalam *Android Studio* menggunakan *Chaquopy* berjalan lancar, memungkinkan segmentasi, ekstraksi ciri, dan klasifikasi citra yang akurat pada aplikasi. Keempat, aplikasi *LIFE* untuk *Android* memiliki performa yang baik, dengan fitur dan fungsi yang bekerja tanpa *bug* atau *crash*, serta mencapai akurasi 98% pada klasifikasi citra dengan perbesaran mikroskop yang sama dengan data latih. Namun, penting untuk mencatat bahwa aplikasi ini tidak sensitif terhadap variasi perbesaran mikroskop, sehingga kontrol terhadap kondisi citra yang diambil perlu diperhatikan dalam penggunaan aplikasi tersebut.

B. Saran

Saran yang dapat dikembangkan untuk tugas akhir ini agar lebih baik lagi adalah sebagai berikut.

1. Meningkatkan kualitas model klasifikasi sehingga dapat lebih sensitif terhadap variasi data. Hal ini dapat dilakukan dengan memanfaatkan jarum penunjuk mikroskop sebagai objek pembanding dan normalisasi data.
2. Mengembangkan aplikasi agar dapat berjalan lebih cepat dan memiliki *interface* yang menarik. Hal ini dapat digunakan dengan melakukan analisa penggunaan memori dan melakukan manajemen memori dengan baik
3. Menambahkan fitur tambahan pada aplikasi seperti variasi model segmentasi, variasi model klasifikasi, menyimpan aktivitas, *update* model dan lain-lain.
4. Mengembangkan aplikasi agar dapat digunakan secara *real-time* dengan memanfaatkan teknologi pengindraan komputer *(computer vision).*
5. Mengimplementasikan model *machine learning* modern yang berbasis pengindraan komputer *(computer vision)* seperti *CNN, YOLO, DNN* dan lain-lain, sehingga tidak perlu adanya tahap ekstraksi ciri

DAFTAR PUSTAKA

[1]. *American Cancer Society*. (2022). *Leukemia [Online]*. *Available*: https://www.cancer.org/cancer/leukemia.html [Diakses pada 17 Januari 2023]

[2]. *National Cancer Institute*. (2021). *Adult Acute Myeloid Leukemia Treatment (PDQ) – Patient Version [Online]*. *Available*: https://www.cancer.gov/types/leukemia/patient/adult-aml-treatment-pdq [Diakses pada 17 Januari 2023]

[3]. *National Cancer Institute*. (2021). *SEER Cancer Stat Facts: Leukemia - Acute Lymphocytic Leukemia (ALL) [Online]*. *Available*: https://seer.cancer.gov/statfacts/html/alyl.html [diakses pada 17 Januari 2023]

[4]. *National Cancer Institute*. (2021). *SEER Cancer Stat Facts: Leukemia - Acute Myeloid Leukemia (AML) [Online]*. *Available*: https://seer.cancer.gov/statfacts/html/amyl.html [diakses pada 17 Januari 2023]

[5]. *American Childhood Cancer Organization*. (2016). *Childhood Cancer Facts and Figures [Online]*. *Available*: https://www.acco.org/wp-content/uploads/2016/10/2016-Childhood-Cancer-Facts-Figures.pdf [diakses pada 17 Januari 2023]

[6]. Rahayu, Jajat Eka. "Implementasi *SVM (Support Vector Machine)* dalam Identifikasi Morfologi Sel Darah Putih (Leukosit)", Universits Jenderal Soedirman, 2019.

[7]. Budiyanto, Aris. "Klasifikasi Citra Sel Darah Putih Berdasarkan Ciri Struktur Morfologi, Tekstur dan Jumlah *Hole* dengan Metode *K-Nearest Neighbors (KNN)*", Universits Jenderal Soedirman, 2018.

[8]. Pui, C.-H., Yang, J. J., & Hunger, S. P. (2015). *Childhood acute Lymphoblastic leukemia: progress through collaboration. Journal of Clinical Oncology*, 33(27), 2938–2948. doi: 10.1200/JCO.2014.59.1636

[9]. Döhner, H., Weisdorf, D. J., & Bloomfield, C. D. (2015). *Acute Myeloid Leukemia. New England Journal of Medicine*, 373(12), 1136–1152. doi: 10.1056/nejmra1406184

[10]. *National Cancer Institute, "Chronic Lymphocytic Leukemia Treatment (PDQ®)–Patient Version,"* Jan. 28, 2022. [Online]. Available: https://www.cancer.gov/types/leukemia/patient/cll-treatment-pdq. [Diakses pada 30 Januari 2023]

[11]. L. Zhang, X. Wu, and Z. Xu, "*Lymphoblast*," in Encyclopedia of Cancer, Berlin, Germany: Springer Berlin Heidelberg, 2011, pp. 1953–1954.

[12]. S. Zhou, L. Zuo and D. M. Morris, *"Acute Myeloid Leukemia (AML) in Adults: Treatment Options," Oncology (Williston Park)*, vol. 35, no. 10, pp. 714-9, 2019

[13]. E. Alpaydin, *Introduction to machine learning, 2nd ed. Cambridge, MA: MIT Press*, 2010

[14]. M. I. Jordan and T. M. Mitchell, *"Machine learning: Trends, perspectives, and prospects," Science*, vol. 349, no. 6245, pp. 255-260, Jul. 2015, doi: 10.1126/science.aaa8415

[15]. C. M. Bishop, *Pattern recognition and machine learning*, vol. 4. New York: Springer, 2006

[16]. S. Wang and X. Yao, *"A survey on application of Support Vector Machines in bioinformatics,"* Neurocomputing, vol. 74, no. 1-3, pp. 142-167, 2011, doi: 10.1016/j.neucom.2010.07.005

[17]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). *Scikit-learn: Machine learning in Python.*

Journal of machine learning research, 12(Oct), 2825-2830.

[18]. R. Pedregosa et al., "*Scikit-learn: Machine learning in Python," Scikit-learn*, 2021. [*Online*]. *Available*: https://*scikit-learn*.org/stable/*user*_guide.html. [Diakses pada 30 Januari 2023]

[19]. Google Corps. 2022. "*Frequently Asked Question"*[*Online*]. Available: https://research.google.com/colaboratory. [Diakses pada 31 Januari 2023]

[20]. Tech Target Contributor. 2018. "Definition *Android Studio*"[*Online*]. *Available*: https://www.techtarget.com/search*mobile*computing/ definition/ *Android-Studio*. [Diakses pada 31 Januari 2023]

[21]. Chaquo.com.(2023) "*Chaquopy* Version 14.0.2," [*Online*]. *Available*: https://chaquo.com/*chaquopy*/*chaquopy*-version-14-0-2/. [Diakses pada 31 Januari 2023]

[22]. S. Raschka and V. Mirjalili, "*Python* and *Machine learning*: A Practical Approach," IEEE Access, vol. 6, pp. 55,242-55,266, 2018

[23]. M. Sewak and R. K. S. Gorthi, "*Python* for *Machine learning*: A Survey," IEEE Access, vol. 7, pp. 17,012-17,032, 2019.