

LAPORAN TUGAS AKHIR

PERBANDINGAN TEKNIK *RESAMPLING* PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT

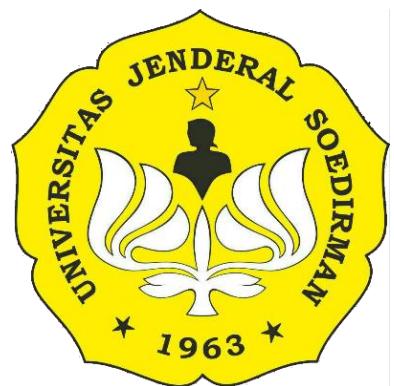
Disusun untuk memenuhi persyaratan memperoleh gelar sarjana di Jurusan
Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Ali Rohman
H1A018021

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN
TEKNOLOGI
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2021**



LAPORAN TUGAS AKHIR

PERBANDINGAN TEKNIK *RESAMPLING* PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT

Disusun untuk memenuhi persyaratan memperoleh gelar sarjana di Jurusan
Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Ali Rohman
H1A018021

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN
TEKNOLOGI
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2021**

HALAMAN PENGESAHAN

Tugas Akhir dengan Judul:

PERBANDINGAN TEKNIK RESAMPLING PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT

Disusun oleh:

Ali Rohman
H1A018021

Diajukan untuk memenuhi salah satu persyaratan
memperoleh gelar Sarjana Teknik pada
Jurusan/Program Studi Teknik Elektro
Fakultas Teknik
Universitas Jenderal Soedirman

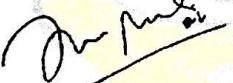
Diterima dan disetujui

Pada Tanggal : 25 Oktober 2021

Pembimbing I

25 OCT 2021

Pembimbing II


Imron Rosyadi, S.T., M.Sc.
NIP 197909242003121003


Muhammad Syaiful Aliim, S.T., M.T.
NIP 199009052019031021

Mengetahui:

Dekan Fakultas Teknik


Prof. Dr. Eng. Suroso, S.T., M.Eng.
NIP 197812242001121002



HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Laporan Tugas Akhir dengan judul "**Perbandingan Teknik Resampling Pada Dataset HAM10000 Terhadap Performa Model Klasifikasi Kanker Kulit**" ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Purbalingga, 25 Oktober 2021



Ali Rohman
NIM. H1A018021

HALAMAN MOTTO DAN PERSEMPAHAN

MOTTO

“Lebih baik gagal karena mencoba daripada menyesal karena tak melakukan apa-apa.”

PERSEMPAHAN

Penulis menyadari bahwa selesainya pembuatan tugas akhir hingga pembuatan laporan tugas akhir ini tidak terlepas dari bantuan berbagai pihak, oleh karena itu penulis mengucapkan terimakasih kepada:

1. Allah SWT, atas berkat rahmat, hidayah, karunia serta petunjuk-Nya, penulis dapat menyelesaikan tugas akhir hingga pembuatan laporan dengan lancar.
2. Orang tua dan saudara-saudara penulis yang senantiasa memberikan dukungan dan do'a selama penulis melaksanakan tugas akhir hingga pembuatan laporan,
3. Ibu Farida Asriani, S.Si., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jenderal Soedirman,
4. Bapak Imron Rosyadi, S.T., M.Sc. selaku Dosen Pembimbing I,
5. Bapak Muhammad Syaiful Aliim, S.T., M.T. selaku Dosen Pembimbing II dan Dosen Pembimbing Akademik,
6. Bapak dan Ibu dosen Teknik Elektro Universitas Jenderal Soedirman,
7. Teman-teman seperjuangan ketika menjalani program Bangkit 2021,
8. Teman-teman Teknik Elektro Universitas Jenderal Soedirman angkatan 2018,
9. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam pelaksanaan dan penulisan laporan tugas akhir.

RINGKASAN

PERBANDINGAN TEKNIK *RESAMPLING* PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT

Ali Rohman

Kanker kulit adalah jenis kanker yang tumbuh di jaringan kulit yang ditandai dengan adanya perubahan kondisi kulit yang abnormal (lesi kulit). Selain dapat menyebabkan kematian, kanker kulit tentu akan merusak penampilan seseorang. Dan yang menambah permasalahan adalah banyak orang biasa tidak dapat membedakan antara tahi lalat dan kanker kulit *melanoma*, padahal *5-year survival rate* untuk *melanoma* jika dapat terdeteksi dini bisa mencapai 99%. Karena itulah penulis ingin membuat sistem klasifikasi kanker kulit menggunakan *dataset* HAM10000 yang mengklasifikasikan kanker kulit berdasarkan kondisi lesi kulit berpigmen menjadi 7 kelas kanker kulit, yaitu akiec, bcc, bkl, nv, mel, vasc, dan df. Yang menjadi permasalahan adalah *dataset* yang digunakan memiliki distribusi data antar kelas yang tidak seimbang, sehingga perlu dilakukan penyeimbangan kualitas data.

Model arsitektur klasifikasi dibuat sendiri menggunakan metode CNN. Untuk melakukan penyeimbangan dataset, dilakukan dengan teknik *resampling* dengan bantuan *library Python imbalanced-learn*. Pada penelitian ini dilakukan uji coba beberapa teknik *resampling* untuk mendapatkan hasil yang paling optimal. Teknik *resampling* yang digunakan antara lain, *random undersampler*, *cluster centroids*, *random oversampler*, SMOTE, ADASYN, gabungan SMOTE dengan *Tomek Link*, dan gabungan SMOTE dengan ENN.

Berdasarkan hasil pelatihan dan pengujian model arsitektur yang telah dirancang dan melewati *trial and error* hingga mencapai kualitas maksimal yang diharapkan. Di antara tiga teknik umum *resampling*, yaitu *undersampling*, *oversampling*, dan gabungan (*hybrid*), secara umum teknik *resampling* gabungan merupakan teknik yang memberikan hasil paling optimal. Dari beberapa teknik *resampling* *dataset* HAM10000, teknik *resampling* gabungan antara *oversampling* SMOTE dan *undersampling* ENN memberikan hasil paling optimal dengan nilai *validation accuracy* sebesar 0.986843, *validation precision* sebesar 0.958295, dan *validation recall* sebesar 0.949213.

Kata kunci : *resampling dataset*, CNN, *deep learning*, kanker kulit.

SUMMARY

COMPARISON OF RESAMPLING TECHNIQUES IN HAM10000 DATASET ON THE PERFORMANCE OF SKIN CANCER CLASSIFICATION MODEL

Ali Rohman

Skin cancer is a type of cancer that grows in the skin tissue which is characterized by abnormal changes in skin conditions (skin lesions). Besides being able to cause death, skin cancer will certainly damage a person's appearance. And what adds to the problem is that many ordinary people can't tell the difference between moles and melanoma skin cancer, whereas the 5-year survival rate for melanoma if detected early can reach 99%. That's why the author wants to create a skin cancer classification system using the HAM10000 dataset which classifies skin cancer based on the condition of pigmented skin lesions into 7 classes of skin cancer, namely akiec, bcc, bkl, nv, mel, vasc, and df. The problem is that the dataset used has an unbalanced distribution of data between classes, so it is necessary to balance the quality of the data.

The classification architecture model was created using the CNN method. To balance the dataset, a resampling technique was used with the help of the imbalanced-learn Python library. In this study, several resampling techniques were tested to get the most optimal results. The resampling techniques used include random undersampler, cluster centroids, random oversampler, SMOTE, ADASYN, combined SMOTE with Tomek Link, and combined SMOTE with ENN.

Based on the results of training and testing of architectural models that have been designed and gone through trial and error to achieve the expected maximum quality. Among the three general resampling techniques, namely undersampling, oversampling, and hybrid, in general the combined resampling technique is the technique that gives the most optimal results. From several resampling techniques for the HAM10000, the combined resampling technique between SMOTE oversampling and ENN undersampling gave the most optimal results with a validation accuracy value is 0.986843, a validation precision is 0.958295, and a validation recall is 0.949213.

Keywords: resampling dataset, CNN, deep learning, skin cancer.

PRAKATA

Puji syukur kehadirat Allah S.W.T. yang telah melimpahkan berkah dan rahmat-Nya dokumen **“PERBANDINGAN TEKNIK RESAMPLING PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT”** ini dapat disusun. Adapun maksud dan tujuan penyusunan laporan tugas akhir ini adalah untuk melengkapi salah satu syarat dalam menempuh Pendidikan sarjana pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jenderal Soedirman. Penyusunan laporan tugas akhir ini telah melibatkan dukungan dan bantuan dari berbagai pihak, oleh karena itu penulis ingin mengucapkan terima kasih kepada :

1. Allah SWT, atas berkat rahmat, hidayah, karunia serta petunjuk-Nya, penulis dapat menyelesaikan tugas akhir hingga pembuatan laporan dengan lancar.
2. Orang tua dan saudara-saudara penulis yang senantiasa memberikan dukungan dan do'a selama penulis melaksanakan tugas akhir hingga pembuatan laporan tugas akhir,
3. Ibu Farida Asriani, S.Si., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jenderal Soedirman,
4. Bapak Imron Rosyadi, S.T., M.Sc. selaku Dosen Pembimbing I,
5. Bapak Muhammad Syaiful Aliim, S.T., M.T. selaku Dosen Pembimbing II dan Dosen Pembimbing Akademik,
6. Bapak dan Ibu dosen Teknik Elektro Universitas Jenderal Soedirman,
7. Teman-teman seperjuangan ketika menjalani program Bangkit 2021,
8. Semua teman - teman Teknik Elektro Universitas Jenderal Soedirman angkatan 2018,
9. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam pelaksanaan dan penulisan laporan tugas akhir.

Purbalingga, 25 Oktober 2021

Ali Rohman

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN	iii
HALAMAN MOTTO DAN PERSEMBAHAN.....	iv
RINGKASAN	v
<i>SUMMARY</i>	vi
PRAKATA.....	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xv
DAFTAR ISTILAH DAN SINGKATAN	xvi
DAFTAR SIMBOL.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan dan Manfaat.....	5
1.4.1 Tujuan	5
1.4.2 Manfaat.....	6
1.5 Sistematika Penulisan.....	6
BAB 2 TINJAUAN PUSTAKA.....	8
2.1 Penelitian Terdahulu	8
2.2 Kanker Kulit	9
2.2.1 Pengertian Kanker Kulit	9
2.2.2 Faktor Risiko Kanker Kulit	10
2.2.3 Fakta Tentang Kanker Kulit.....	10
2.2.4 <i>Dataset</i> Kanker Kulit MNIST: HAM10000	11
2.3 <i>Deep Learning</i>	15
2.4 <i>Convolutional Neural Network</i> (CNN)	18
2.4.1 Konsep <i>Convolutional Neural Network</i> (CNN)	18
2.4.2 Arsitektur Jaringan <i>Convolutional Neural Network</i> (CNN)	20

2.5 Teknik <i>Resampling</i> Data.....	23
2.5.1 Teknik <i>Undersampling</i>	24
2.5.2 Teknik <i>Oversampling</i>	26
2.5.3 Teknik <i>Resampling</i> Gabungan (<i>Hybrid Resampling</i>)	28
2.6 <i>Google Colaboratory</i>	28
2.7 <i>Framework TensorFlow</i> dan <i>Keras</i>	30
2.7.1 <i>Framework TensorFlow</i>	30
2.7.2 <i>Framework Keras</i>	31
BAB 3 METODE PENELITIAN.....	33
3.1 Waktu dan Tempat	33
3.2 Alat dan Bahan	33
3.3 Alur dan Tahap Penelitian	34
3.3.1 Tahap Persiapan	34
3.3.2 Tahap Pelaksanaan	35
3.3.3 Tahap Pengujian.....	37
3.3.4 Tahap Evaluasi.....	38
3.3.5 Tahap Pembuatan Laporan.....	38
3.4 Waktu dan Jadwal Penelitian.....	39
BAB 4 HASIL DAN PEMBAHASAN.....	40
4.1 Perancangan Sistem dan <i>Dataset</i> Penelitian	40
4.1.1 Sumber <i>Dataset</i>	40
4.1.2 Pengunggahan <i>Library</i> dan <i>Dataset</i>	41
4.1.3 <i>Dataset</i> Hasil <i>Resampling</i>	43
4.1.4 Pelatihan dan Pengujian Pada <i>Google Colaboratory</i>	45
4.2 Hasil Pelatihan dan Pengujian Pada <i>Google Colaboratory</i>	47
4.2.1 Model Menggunakan <i>Dataset</i> Hasil Teknik <i>Random Undersampler</i> ..50	50
4.2.2 Model Menggunakan <i>Dataset</i> Hasil Teknik <i>Cluster Centroids</i>55	55
4.2.3 Model Menggunakan <i>Dataset</i> Hasil Teknik <i>Random Oversampler</i>60	60
4.2.4 Model Menggunakan <i>Dataset</i> Hasil Teknik <i>SMOTE</i>	65
4.2.5 Model Menggunakan <i>Dataset</i> Hasil Teknik <i>ADASYN</i>	70
4.2.6 Model Menggunakan <i>Dataset</i> Hasil Teknik Gabungan <i>SMOTE</i> dan <i>Tomek Link</i>	75
4.2.7 Model Menggunakan <i>Dataset</i> Hasil Teknik Gabungan <i>SMOTE</i> dan <i>ENN</i>	80
4.3 Perbandingan Unjuk Kerja Model Dengan Teknik <i>Resampling Dataset</i>84	84
4.4 <i>Website</i> Antarmuka Pengguna Sistem Klasifikasi Kanker Kulit.....	87
BAB 5 PENUTUP	91
5.1 Kesimpulan.....	91
5.2 Saran	92
DAFTAR PUSTAKA	93

LAMPIRAN	96
Lampiran 1. <i>Source Code Program Model Klasifikasi Kanker Kulit di Google Colaboratory Menggunakan Bahasa Pemrograman Python</i>	96
Lampiran 2. <i>Source Code Program Wesbite di Visual Studio Code</i>	111
Lampiran 3. Simulasi Prediksi Kanker Kulit di <i>Website</i>	130
BIODATA PENULIS	133

DAFTAR TABEL

Tabel 3.1 Waktu dan Jadwal Penelitian.....	39
Tabel 4.1 Distribusi Data Tiap Kelas Setelah Dilakukan Resampling.....	44
Tabel 4.2 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Random Undersampler.....	53
Tabel 4.3 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Cluster Centroids.....	58
Tabel 4.4 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Random Oversampler.....	63
Tabel 4.5 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE	68
Tabel 4.6 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik ADASYN	73
Tabel 4.7 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE + Tomek.....	78
Tabel 4.8 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE + ENN.....	83
Tabel 4.9 Perbandingan Metrik Klasifikasi Model Terbaik Tiap Teknik Resampling.....	85
Tabel 4.10 Perbandingan Nilai Prediksi Tiap Kelas Menggunakan Berbagai Teknik Resampling.....	86

DAFTAR GAMBAR

Gambar 2.1 Skema Aliran metode kerja dataset HAM10000.....	12
Gambar 2.2 Contoh penyakit kanker kulit jenis akiec	13
Gambar 2.3 Contoh penyakit kanker kulit jenis bcc	13
Gambar 2.4 Contoh penyakit kanker kulit jenis bkl	14
Gambar 2.5 Contoh penyakit kanker kulit jenis df	14
Gambar 2.6 Contoh penyakit kanker kulit jenis nv.....	14
Gambar 2.7 Contoh penyakit kanker kulit jenis mel.....	15
Gambar 2.8 Contoh penyakit kanker kulit jenis vasc	15
Gambar 2.9 Proses Perancangan dan Deployment dari Sistem Berbasis Deep Learning	16
Gambar 2.10 Representasi deep learning dalam model klasifikasi digit	18
Gambar 2.11 Ilustrasi Arsitektur Metode CNN	19
Gambar 2.12 Contoh nput dengan pixel ukuran 4x4 dan 3 channel warna RGB .	20
Gambar 2.13 Ilustrasi Proses Konvolusi Pada Sebuah Citra	21
Gambar 2.14 Macam-macam fungsi aktivasi.....	22
Gambar 2.15 Tipe Pooling	22
Gambar 2.16 Fully Connected Layer	23
Gambar 2.17 Ilustrasi Teknik Resampling Data	24
Gambar 2.18 Tampilan Awal Google Colaboratory	29
Gambar 2.19 Popularitas framework deep learning pada tahun 2014-2017	30
Gambar 3.1 Diagram Alir Tahapan Penelitian	34
Gambar 3.2 Tahapan Pelaksaan Penelitian	35
Gambar 3.3 Diagram Alir Tahap Pembuatan Desain Arsitektur	36
Gambar 4.1 Grafik Distribusi Kelas Kanker Kulit Pada Dataset HAM10000	41
Gambar 4.2 Distribusi Data Jenis Kelamin Pada Dataset HAM10000.....	41
Gambar 4.3 Distribusi Usia Pada Dataset HAM10000.....	42
Gambar 4.4 Lokasi Penyakit Kanker Kulit Pada Dataset HAM10000	42
Gambar 4.5 Arsitektur Model CNN Klasifikasi Kanker Kulit.....	45
Gambar 4.6 Visualisasi Arsitektur Model CNN Klasifikasi Kanker Kulit	46
Gambar 4.7 Ilustrasi Metrik Evaluasi Terhadap Suatu Prediksi	49
Gambar 4.8 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	50
Gambar 4.9 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Random Undersampler	50
Gambar 4.10 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Undersampler.....	51
Gambar 4.11 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Undersampler.....	52
Gambar 4.12 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Random Undersampler.....	54
Gambar 4.13 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	55
Gambar 4.14 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Cluster Centroids.....	55

Gambar 4.15 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Cluster Centroids.....	56
Gambar 4.16 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Cluster Centroids.....	57
Gambar 4.17 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Cluster Centroids.....	59
Gambar 4.18 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	60
Gambar 4.19 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Random Oversampler	60
Gambar 4.20 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Oversampler.....	61
Gambar 4.21 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Oversampler.....	62
Gambar 4.22 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Random Oversampler	64
Gambar 4.23 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	65
Gambar 4.24 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE	65
Gambar 4.25 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE	66
Gambar 4.26 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE	67
Gambar 4.27 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE	69
Gambar 4.28 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	70
Gambar 4.29 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik ADASYN	70
Gambar 4.30 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling ADASYN	71
Gambar 4.31 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling ADASYN	72
Gambar 4.32 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan ADASYN	74
Gambar 4.33 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	75
Gambar 4.34 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE + Tomek Link	75
Gambar 4.35 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + Tomek Link	76
Gambar 4.36 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + Tomek Link	77
Gambar 4.37 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE + Tomek Link	79
Gambar 4.38 Grafik Distribusi Dataset Sebelum Dilakukan Resampling	80
Gambar 4.39 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE + ENN.....	80

Gambar 4.40 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + ENN.....	81
Gambar 4.41 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + ENN	82
Gambar 4.42 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE + ENN.....	84
Gambar 4.43 Kurva Metrik Evaluasi Model Terbaik Tiap Teknik Resampling ...	85
Gambar 4.44 Tampilan Program app.py di Visual Studio Code	89
Gambar 4.45 Tampilan Program HTML di Visual Studio Code.....	89
Gambar 4.46 Tampilan Website Sebagai Antarmuka Pengguna Sistem Klasifikasi Kanker Kulit.....	90

DAFTAR LAMPIRAN

Lampiran 1.	<i>Source Code Program Model Klasifikasi Kanker Kulit di Google Colaboratory Menggunakan Bahasa Pemrograman Python</i>	96
Lampiran 2.	<i>Source Code Program Website di Visual Studio Code</i>	111
Lampiran 3.	Simulasi Prediksi Kanker Kulit di <i>Website</i>	130

DAFTAR ISTILAH DAN SINGKATAN

- Resampling* : Proses *sampling* ulang terhadap sampel atau data.
- Dataset* : Kumpulan objek data yang merepresentasikan data dan relasinya di memori.
- HAM10000* : *Human Against Machine with 10000 training images*, merupakan *dataset* kanker kulit berisi 10015 citra yang mengklasifikasikan kanker kulit menjadi 7 kelas/tipe kanker kulit.
- Lesi Kulit* : jaringan kulit yang tumbuh abnormal, baik di permukaan atau di bawah permukaan kulit.
- Pigmen* : Pigmen adalah zat pewarna tubuh pada makhluk hidup. Banyak atau sedikitnya pigmen memengaruhi warna kulit seseorang.
- CNN* : *Convolutional Neural Network*, merupakan salah satu metode dari *deep learning* yang digunakan untuk mengklasifikasikan data.
- RNN* : *Recurrent neural networks (RNN)* adalah jenis arsitektur jaringan saraf tiruan yang cocok dipakai untuk pemrosesan data sekuensial.
- Deep learning* : subbidang dari *machine learning* yang banyak digunakan untuk klasifikasi objek.
- SCC* : *Squamous Cell Carcinoma* adalah jenis kanker kulit yang sering ditemukan mengenai lapisan epidermis bagian bawa.
- Akiec* : *Actinic keratoses and intraepithelial carcinoma / bowen's disease*
- BCC* : *Basal Cell Carcinoma*
- BKL* : *Benign Keratosis-Like Lesions*
- Df* : *Dermatofibroma*
- Nv* : *Melanocytic Nevi*
- Mel* : *Melanoma*
- Vasc* : *Vascular Lesions*
- Python* : Bahasa pemrograman interpretatif yang bersifat *open source* yang mudah dibaca dan mudah dipahami.
- CPU* : *Central Processing Unit*
- GPU* : *Graphical Processing Unit*
- API* : *Application Programming Interface*
- Pixel Array* : Baik pada monitor komputer atau keluaran digital, kisi-kisi *pixels* dua dimensi yang menentukan bentuk dan warna gambar digital.
- SMOTE* : *Synthetic Minority Over-sampling Technique*
- ENN* : *Edited Nearest Neighbour*

DAFTAR SIMBOL

BAB 1 **PENDAHULUAN**

1.1 Latar Belakang

Kulit adalah organ tubuh makhluk hidup yang paling terlihat karena terletak paling luar dari tubuh manusia karena itulah kulit memiliki fungsi sebagai lapisan pelindung untuk organ-organ di dalam tubuh. Kulit berfungsi untuk melindungi otot, pembuluh darah, sel saraf serta organ tubuh lainnya dari kuman dan zat berbahaya. Maka dari itu, kulit sangat risikan terhadap adanya penyakit karena kulit menerima segala zat berbahaya bagi tubuh.

Kanker kulit adalah jenis kanker yang tumbuh di jaringan kulit yang ditandai dengan adanya perubahan kondisi kulit yang abnormal (lesi kulit). Kanker kulit dapat terjadi karena disebabkan oleh mutasi pada DNA sel yang menyebabkan kelainan pada sel kulit. Mutasi ini membuat sel tumbuh lebih cepat dan hidup lebih lama, serta sel kehilangan sifat dasarnya. Penyebab utama kanker kulit diduga karena disebabkan oleh paparan sinar ultraviolet dari matahari yang kemudian menyebabkan rusaknya sel pada kulit, hingga menimbulkan kanker kulit. Kanker kulit biasanya terjadi pada area kulit yang sering terpapar sinar matahari, seperti kulit kepala, wajah, bibir, telinga, leher, dan kaki [1].

Ada dua jenis utama kanker kulit, yaitu *melanoma* dan *nonmelanoma*. Tumor *nonmelanoma* yang paling umum adalah jenis *Basal Cell Carcinoma* (BCC) dan *Squamous Cell Carcinoma* (SCC). Kanker kulit *melanoma* adalah jenis kanker ke-19 yang paling sering terjadi pada pria dan wanita di dunia. Ada hampir 300.000 kasus baru pada tahun 2018 dan kanker kulit *nonmelanoma* adalah jenis kanker ke-

5 yang paling sering terjadi pada pria dan wanita di dunia, dengan lebih dari 1 juta diagnosis pada tahun 2018, walaupun mungkin angka ini tidak terlalu besar namun hal ini tetap harus menjadi perhatian mengingat urutannya merupakan urutan teratas pada jenis kanker yang terjadi di dunia [2].

Di Jakarta, pada tahun 2000-2009, Poliklinik Departemen Ilmu Kesehatan Kulit dan Kelamin (IKKK) Rumah Sakit dr. Cipto Mangunkusumo (RSCM) melaporkan 261 kasus jenis kanker kulit BCC, 69 kasus jenis kanker kulit SCC, dan 22 kasus jenis kanker kulit *melanoma*. Berlainan dengan data sebelumnya, Rumah Sakit Kanker Dharmais (RSKD) selama tahun 2005-2007 mencatat bahwa kanker kulit *nonmelanoma* yang paling sering adalah jenis SCC, diikuti oleh BCC dan *melanoma* [3].

Melihat tingkat kasus kanker kulit dan dengan kondisi sinar ultraviolet dari matahari yang semakin mengkhawatirkan, diperlukan deteksi dini kondisi kanker kulit dengan memanfaatkan teknologi yang juga semakin maju. Dalam bidang teknologi, *computer vision* merupakan bidang teknologi yang mampu menciptakan sistem klasifikasi kanker kulit berdasarkan kondisi lesi kulit. *Computer vision* merupakan serangkaian teknologi yang memungkinkan komputer untuk menangkap, menganalisis, dan mengolah data gambar maupun objek lainnya. Banyak pengembangan sistem yang menerapkan *computer vision*, seperti *face detection*, *image recognition*, maupun pengenalan pola sehingga mempermudah pekerjaan di berbagai bidang karena fungsionalitasnya yang tinggi dalam hal menginterpretasikan objek yang tampak melalui komputer. Hal ini sesuai dengan

namanya, yaitu “*computer*” yang berarti komputer dan “*vision*” yang berarti penglihatan.

Pada masa sekarang, *deep learning* menjadi teknik yang populer di bidang *computer vision*. Yang menjadi salah satu bentuk kemajuan dalam bidang *computer vision* dengan penggunaan metode *deep learning* adalah adanya algoritma yang telah dibangun dan disempurnakan seiring waktu untuk mengolah data dua dimensi, yaitu *Convolutional Neural Network* (CNN). CNN bekerja dengan cara mengambil gambar input, kemudian memilih dan menganalisis karakteristiknya agar dapat dibedakan satu dari yang lain. Sehingga, penerapannya dalam sistem klasifikasi kanker kulit adalah algoritma CNN akan menganalisis karakteristik gambar kondisi lesi kulit seseorang yang ingin diprediksi, kemudian mengklasifikasikannya sesuai dengan jenis kanker kulit sesuai kondisi lesi kulit berpigmennya.

Model klasifikasi kanker kulit berdasarkan kondisi lesi kulit berpigmen dalam penelitian ini dibuat menggunakan *dataset* yang populer untuk permasalahan klasifikasi kanker kulit, yaitu *dataset* HAM10000. *Dataset* HAM10000 berisi 10015 gambar yang dibagi menjadi 7 kelas kanker kulit, yaitu *actinic keratoses and intraepithelial carcinoma* (akiec), *basal cell carcinoma* (bcc), *benign keratosis-like lesions* (bkl), *dermatofibroma* (df), *melanoma* (mel), *melanocytic nevi* (nv), dan *vascular lesions* (vasc) [4].

Dataset HAM10000 yang digunakan sayangnya memiliki masalah, yaitu distribusi data antar kelas yang ada sangat tidak seimbang, sehingga akan menyebabkan model klasifikasi menjadi cenderung ke salah satu kelas/jenis kanker kulit. Sehingga pada tugas akhir ini, penulis mengambil topik tentang optimasi

teknik atau metode untuk menyeimbangkan *dataset* pada model klasifikasi kanker kulit dengan judul “PERBANDINGAN TEKNIK RESAMPLING PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana merancang klasifikasi kanker kulit dengan algoritma *deep learning*?
2. Bagaimana mengukur unjuk kerja *deep learning* dalam sistem klasifikasi kanker kulit?
3. Bagaimana meningkatkan unjuk kerja klasifikasi kanker kulit dengan metode teknik *resampling dataset*?
4. Bagaimana merealisasikan sistem klasifikasi kanker kulit pada *in-browser app*?

1.3 Batasan Masalah

Agar permasalahan tidak meluas, maka penulis menentukan batasan masalah penelitian tugas akhir sebagai berikut.

1. Model klasifikasi kanker kulit berdasarkan kondisi lesi kulit berpigmen dibuat hanya menggunakan *dataset* HAM10000 yang berisikan 10015 citra yang terbagi menjadi 7 kelas.
2. Arsitektur model klasifikasi dibuat sendiri menggunakan metode *deep learning* khususnya algoritma *Convolutional Neural Network (CNN)*.

3. Program klasifikasi kanker kulit dibuat dengan menggunakan data masukan citra dengan Bahasa Pemrograman *Python* di platform *Google Colaboratory*.
4. Pembuatan model klasifikasi kanker kulit menggunakan *framework* *TensorFlow* dan *Keras*.
5. Teknik *resampling* yang dipakai untuk menyeimbangkan *dataset* menggunakan metode dari *python library package imbalanced-learn (imblearn)*.
6. Uji banding unjuk kerja antar teknik *resampling dataset* dilakukan dengan membandingkan nilai metrik klasifikasi, seperti *accuracy*, *loss*, *true positive*, *false positive*, *true negative*, *false negative*, *precision*, dan *recall* dari data uji menggunakan metode *model.evaluate* dan menggunakan metode *classification report* dari *library scikit-learn* untuk mengukur nilai *precision*, *recall*, dan *F1-score* dari tiap kelas yang diklasifikasikan.
7. *Website* untuk antarmuka pengguna dibuat menggunakan Bahasa Pemrograman *HTML*, *CSS*, dan *Javascript* dengan menggunakan platfrom *Visual Studio Code*.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan pembuatan laporan tugas akhir ini adalah sebagai berikut.

1. Merancang arsitektur model CNN untuk klasifikasi kanker kulit.
2. Mengukur unjuk kerja *deep learning* dalam sistem klasifikasi kanker kulit.
3. Membandingkan pengaruh berbagai teknik *resampling dataset* HAM10000 terhadap performa model klasifikasi kanker kulit.

4. Meningkatkan untuk kerja sistem klasifikasi kanker kulit dengan teknik *resampling dataset*.
5. Merealisasikan sistem klasifikasi kanker kulit dalam *in-browser app*.

1.4.2 Manfaat

Manfaat yang diharapkan dari pelaksanaan tugas akhir ini adalah sebagai berikut.

1. Mampu menerapkan ilmu yang didapat selama kuliah untuk menyelesaikan tugas akhir.
2. Mempermudah masyarakat dan tenaga kesehatan dalam melakukan *screening* klasifikasi lesi kanker kulit dan untuk pengembangan lebih lanjutnya dapat diintegrasikan dengan sistem atau alat pada rumah sakit.
3. Mengetahui teknik *resampling dataset* HAM10000 yang menghasilkan unjuk kerja optimal terhadap performa model klasifikasi kanker kulit.
4. Mengetahui pengaruh berbagai teknik *resampling dataset* terhadap unjuk kerja model klasifikasi lesi kanker kulit, sehingga bisa dijadikan referensi dalam membuat model lainnya.

1.5 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan dalam laporan tugas akhir ini adalah sebagai berikut.

BAB 1 Pendahuluan

Bab Pendahuluan berisi tentang judul penelitian, latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian.

BAB 2 Tinjauan Pustaka

Bab Tinjauan Pustaka berisi tentang dasar kajian pustaka yang mendasari gagasan penelitian tugas akhir ini, antara lain pembahasan tentang penelitian terkait yang terdahulu serta teori-teori pendukung terkait penelitian tugas akhir ini, seperti penjelasan tentang kanker kulit, *dataset HAM10000, deep learning, Convolutional Neural Network (CNN), teknik resampling dataset, Google Colaboratory, framework keras dan TensorFlow.*

BAB 3 Metode Penelitian

Bab Metode Penelitian berisi tentang langkah-langkah atau metode yang digunakan dalam penelitian, meliputi waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, tahapan-tahapan dalam melakukan penelitian, sumber data, alur penelitian, dan jadwal penelitian.

BAB 4 Hasil dan Pembahasan

Bab Hasil dan Pembahasan berisi tentang pemaparan hasil dan pembahasannya tentang pengaruh penerapan berbagai teknik *resampling* pada *dataset HAM10000* terhadap performa model klasifikasi kanker kulit.

BAB 5 Penutup

Bab Penutup berisi tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam penyusunan penelitian tugas akhir “Perbandingan Teknik *Resampling* Pada *Dataset HAM10000* Terhadap Performa Model Klasifikasi Kanker Kulit” terdapat beberapa penelitian terdahulu yang berkaitan, yaitu tentang pembuatan model klasifikasi kanker kulit dan tentang pengaruh teknik *resampling* pada suatu model. Penelitian tersebut dapat dilihat sebagai berikut.

1. Teresia R. Savera, Winsya H. Suryawan, dan Agung W. Setiawan dengan judul “Deteksi Dini Kanker Kulit Menggunakan K-NN dan *Convolutional Neural Network*”. Penelitian ini melakukan klasifikasi kanker kulit menjadi 2 jenis, yaitu *malignant* dan *benign* dengan menggunakan metode klasifikasi k-NN dengan 80 citra latih *benign*, 80 citra latih *malignant*, 20 citra uji *benign*, dan 20 citra uji *malignant* dan menggunakan metode klasifikasi CNN dengan 320 citra latih *benign*, 320 citra latih *malignant*, 80 citra uji *benign*, dan 80 citra uji *malignant* [5].
2. Zeyhan Aliyah, Anditya Arifianto, dan Febryanti Sthevanie dengan judul “*Classifying Skin Cancer in Digital Images Using Convolutional Neural Network with Augmentation*”. Penelitian ini membahas tentang penggunaan *augmentation* seperti *feature wise center*, *feature wise std normalization*, *rotation range*, *zoom range*, *width shift range*, *weight shift range*, *horizontal flip*, dan *vertical flip* dalam meningkatkan akurasi model klasifikasi kanker kulit [6].

3. Akhmad Syukron dan Agus Subekti dengan judul “Penerapan Metode *Random Over-Under Sampling* dan *Random Forest* untuk Klasifikasi Penilaian Kredit”. Penelitian ini membahas tentang perbandingan metode *Random Over-Undersampling* dan *Random Forest* untuk klasifikasi penilaian kredit yang menggunakan *dataset German Credit* [7].
4. Rimbun Siringoringo dengan judul “Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan *k-Nearest Neighbor*”. Penelitian ini membahas tentang perbandingan metode *Synthetic Minority Over-sampling Technique* (SMOTE) dan *k-Nearest Neighbor* yang menggunakan *dataset Credit Card Fraud* yang bersumber dari *UCI Repository* yang terdiri dari 29.976 data, dengan pembagian 23.347 data positif (label 0) dan 6.629 data negatif (label 1) [8].

2.2 Kanker Kulit

2.2.1 Pengertian Kanker Kulit

Kanker kulit adalah jenis kanker yang tumbuh di jaringan kulit yang ditandai dengan adanya perubahan kondisi kulit yang abnormal (lesi kulit). Kanker kulit dapat terjadi karena disebabkan oleh mutasi pada DNA sel yang menyebabkan kelainan pada sel kulit. Mutasi ini membuat sel tumbuh lebih cepat dan hidup lebih lama, serta sel kehilangan sifat dasarnya. Penyebab utama kanker kulit diduga karena disebabkan oleh paparan sinar ultraviolet dari matahari yang kemudian menyebabkan rusaknya sel pada kulit, hingga menimbulkan kanker kulit. Kanker kulit biasanya terjadi pada area kulit yang sering terpapar sinar matahari, seperti kulit kepala, wajah, bibir, telinga, leher, dan kaki [1].

2.2.2 Faktor Risiko Kanker Kulit

Beberapa faktor risiko yang menyebabkan timbulnya penyakit kanker kulit, antara lain kulit sering terpapar sinar ultraviolet dari matahari, daya tahan tubuh rendah, memiliki kulit berwarna cerah atau putih, karena orang berkulit putih lebih rawan terkena penyakit kanker kulit, orang dengan usia lanjut, terpapar oleh bahan kimia tertentu yang dapat merusak kondisi kulit, mengidap kondisi lesi kulit *actinic keratosis* (akiec), yaitu bercak-bercak berupa penebalan kulit yang bersisik pada daerah wajah, tangan, dan kepala. Kanker kulit juga dapat disebabkan oleh faktor keturunan, yaitu memiliki hubungan darah dengan anggota keluarga yang memiliki riwayat penyakit kanker kulit, atau dapat juga karena memiliki riwayat mendapatkan terapi dengan metode radiasi, karena radiasi dapat mengganggu kondisi kulit [9].

2.2.3 Fakta Tentang Kanker Kulit

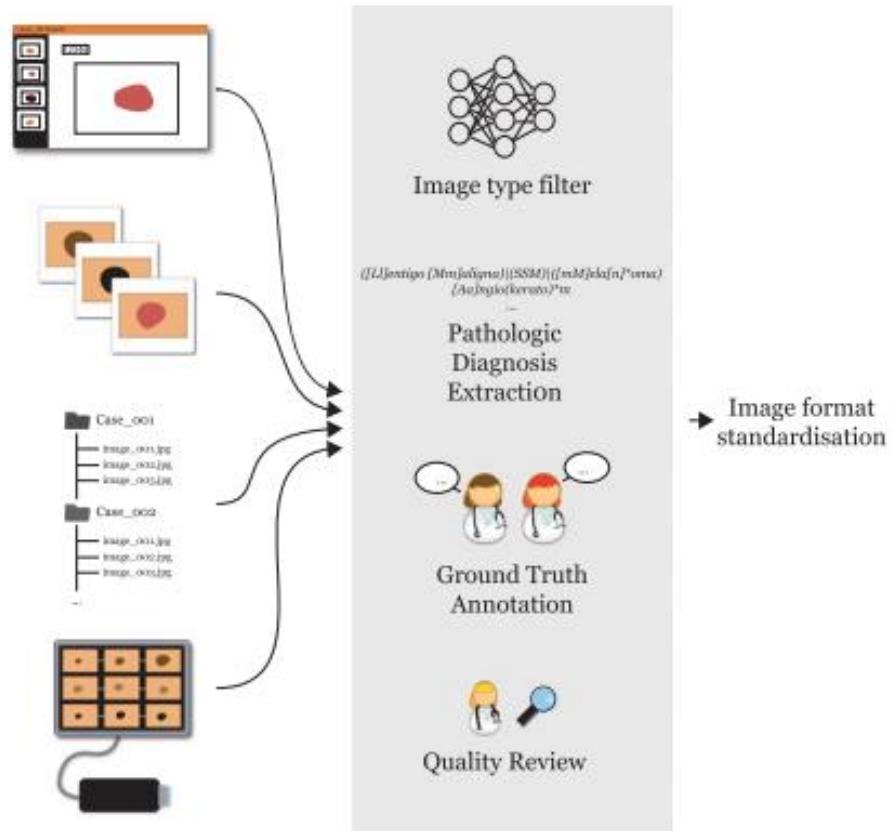
Sebagian besar *melanoma* disebabkan oleh sinar matahari. Satu penelitian di Inggris menemukan bahwa sekitar 86% *melanoma* dapat dikaitkan dengan paparan radiasi ultraviolet (UV) dari matahari [10]. Kanker kulit jenis *melanoma* adalah jenis kanker ke-19 yang paling sering terjadi pada pria dan wanita di dunia. Ada hampir 300.000 kasus baru pada tahun 2018 dan kanker kulit *nonmelanoma* adalah jenis kanker ke-5 yang paling sering terjadi pada pria dan wanita di dunia[2]. Kanker kulit adalah kanker paling umum di Amerika Serikat. Perkiraan saat ini adalah bahwa 1 dari 5 orang Amerika akan terdampak kanker kulit dalam hidup mereka [11]. Diperkirakan sekitar 9.500 orang di Amerika Serikat didiagnosis menderita kanker kulit setiap hari [12]. Actinic keratosis adalah kanker kulit yang paling umum yang mempengaruhi lebih dari 58 juta orang Amerika [13].

Kanker kulit bisa menyerang siapa saja, apapun warna kulitnya. Insiden kanker kulit di antara orang kulit putih *non-Hispanic* (*Hispanic* mengacu pada orang-orang yang berbicara bahasa Spanyol atau yang merupakan keturunan dari garis keturunan berbahasa Spanyol) hampir 30 kali lebih tinggi daripada orang kulit hitam *non-Hispanic* atau orang Asia/Pasifik [14]. *Melanoma* pada orang kulit hitam, orang Asia, dan penduduk asli Hawaii paling sering terjadi pada kulit yang tidak terpapar matahari dengan pigmen yang lebih sedikit, dengan 60 hingga 75% tumor muncul di telapak tangan, telapak kaki, selaput lendir, dan daerah kuku [15]. *5-year relative survival rate* untuk orang-orang yang *melanoma*-nya terdeteksi dini dan diobati sebelum menyebar ke kelenjar getah bening adalah 99%. *5-year relative survival rate* untuk *melanoma* yang sudah menyebar ke kelenjar getah bening di dekatnya adalah 66%. *5-year relative survival rate* untuk penyakit kanker kulit jenis *melanoma* yang sudah menyebar ke kelenjar getah bening yang jauh dan organ lain adalah 27% [16].

2.2.4 Dataset Kanker Kulit MNIST: HAM10000

Dataset HAM10000 (Human Against Machine with 10000 training images) merupakan *dataset* kanker kulit yang mengklasifikasikan kanker kulit menjadi 7 kelas/tipe kanker kulit berdasarkan kondisi lesi kulit berpigmen. *Dataset HAM10000* terdiri dari 10015 gambar dermatoskopik yang dikumpulkan selama 20 tahun dari dua lokasi berbeda, yaitu *Department of Dermatology at the Medical University of Vienna*, Austria, dan *the skin cancer practice of Cliff Rosendahl in Queensland*, Australia. Situs Australia menyimpan gambar dan meta-data dalam *file*

PowerPoint dan *database Excel* [4]. Proses peneliti mendapatkan citra kanker kulit ditunjukkan pada Gambar 2.1 di bawah ini.



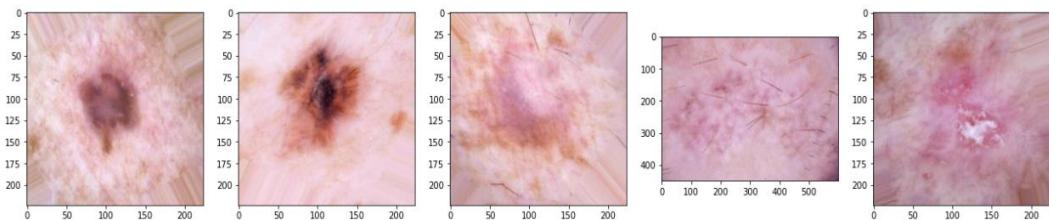
Gambar 2.1 Skema Aliran metode kerja dataset HAM10000.

Berikut ini adalah penjelasan 7 kelas kanker kulit yang diklasifikasikan dalam *dataset* HAM10000 [4].

Actinic Keratoses (akiec)

Actinic Keratoses (Solar Keratoses) and Intraepithelial Carcinoma (Bowen's disease) atau yang disingkat akiec adalah varian non-invasif yang umum dari *squamous cell carcinoma* (SCC) yang dapat diobati tanpa operasi. Kulit di sekitarnya biasanya ditandai dengan kerusakan parah akibat sinar matahari (sinar

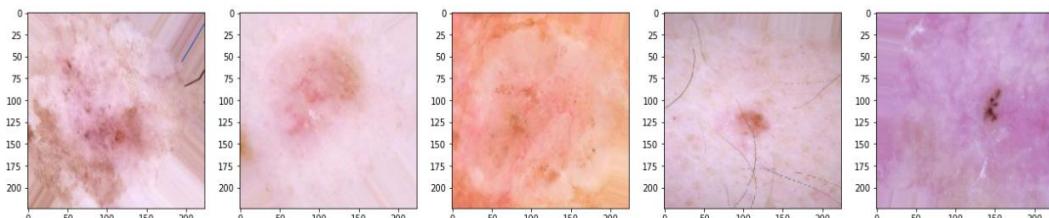
UV). Akiec lebih sering terjadi pada wajah, sering terlihat seperti bercak kecil, kasar, bersisik pada kulit.



Gambar 2.2 Contoh penyakit kanker kulit jenis akiec

Basal Cell Carcinoma (bcc)

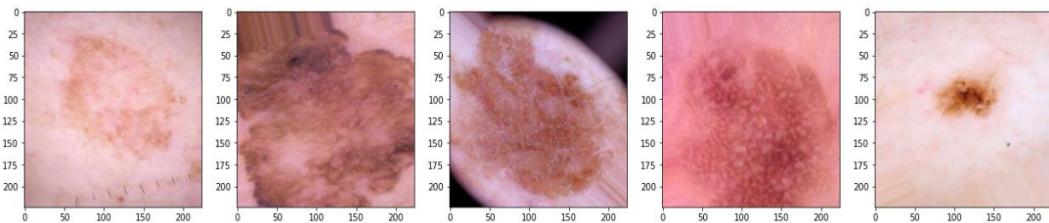
Bcc adalah varian umum dari kanker kulit epitel yang jarang bermetastasis (menyebar). Biasanya dimulai sebagai benjolan berbentuk mutiara di bagian wajah, telinga atau leher. Jika berkembang di bagian dada atau punggung, kanker ini terlihat seperti bagian kulit yang bersisik atau kering.



Gambar 2.3 Contoh penyakit kanker kulit jenis bcc

Benign Keratosis-Like Lesions (bkl)

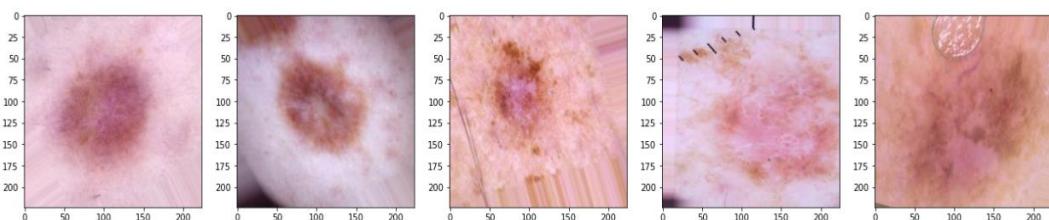
Bkl adalah kelas generik jinak yang termasuk seborrheic keratosis ("kutil pikun"). Dari pandangan dermatoskopik, bkl dapat menunjukkan bentuk yang meniru *melanoma*. Tampilan dermatoskopiknya bervariasi menurut lokasi anatomi dan jenisnya.



Gambar 2.4 Contoh penyakit kanker kulit jenis bkl

Dermatofibroma (df)

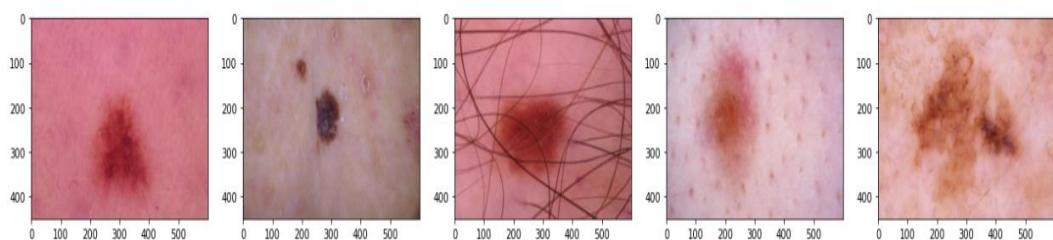
Df adalah lesi kulit jinak yang dianggap sebagai proliferasi jinak atau reaksi inflamasi terhadap minimal trauma. Persentasi dermatoskopis yang paling umum adalah garis retikuler di perifer dengan *patch* putih berpusat yang menunjukkan fibrosis.



Gambar 2.5 Contoh penyakit kanker kulit jenis df

Melanocytic Nevi (nv)

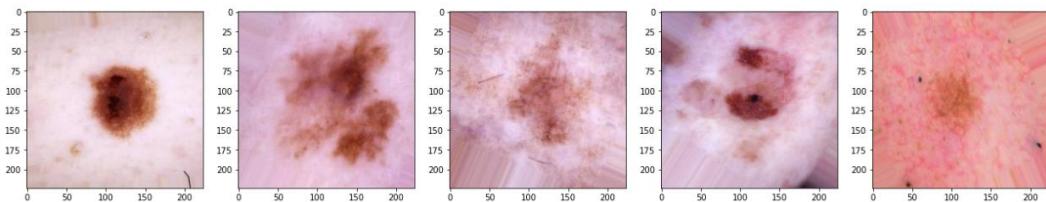
Nv adalah tumor jinak dan muncul dalam berbagai varian, yang semuanya termasuk dalam seri nv. Varian ini mungkin berbeda secara signifikan dari sudut pandang dermatoskopik. Berbeda dengan *melanoma*, mereka biasanya simetris dalam hal distribusi warna dan struktur bentuk.



Gambar 2.6 Contoh penyakit kanker kulit jenis nv

Melanoma (mel)

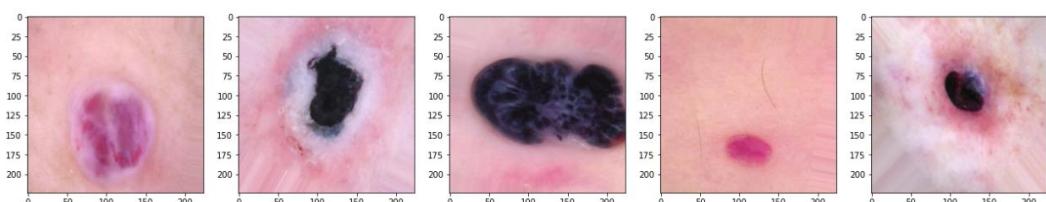
Mel adalah salah satu bentuk kanker kulit yang lebih serius dan terkadang muncul dari tahi lalat yang ada di kulit. Lebih umum *melanoma* muncul sebagai tanda atau tahi lalat baru pada kulit normal. *Melanoma* dapat muncul di mana saja di tubuh tetapi paling sering terjadi pada pria di punggung, dan pada wanita di kaki.



Gambar 2.7 Contoh penyakit kanker kulit jenis mel

Vascular Lesions (vasc)

Vasc dalam kumpulan data dari *angioma ceri* hingga *angiookeratoma* dan *granuloma piogenik*. *Angioma* secara dermatoskopis ditandai dengan warna merah atau ungu dan struktur padat berbatas yang dikenal sebagai gumpalan merah (*lakuna*). Vasc adalah kelainan yang relatif umum pada kulit dan jaringan di bawahnya, lebih dikenal sebagai tanda lahir.

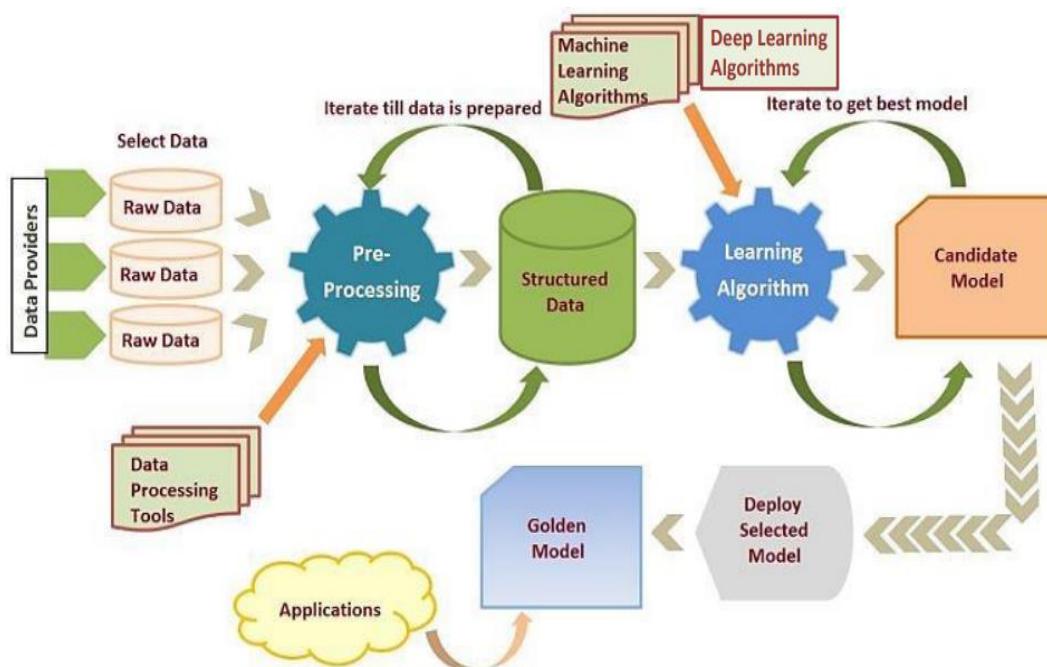


Gambar 2.8 Contoh penyakit kanker kulit jenis vasc

2.3 Deep Learning

Deep learning adalah jenis pembelajaran mesin (*machine learning*) dengan model belajar dengan cara melakukan pengklasifikasikan data seperti gambar, tes, atau suara berdasarkan karakteristik data. *Deep learning* biasanya

diimplementasikan menggunakan jaringan saraf. Istilah “*deep*” merepresentasikan pembelajaran dari data yang menekankan pada jumlah lapisan-lapisan (*layers*) jaringan saraf, semakin banyak lapisan maka semakin dalam jaringan (*deeper network*). Jaringan saraf tradisional hanya mengandung 2 atau 3 lapisan, sedangkan *deep network* dapat memiliki ratusan *network* [17]. *Deep learning* sangat cocok diimplementasikan pada aplikasi identifikasi seperti pengenalan wajah, terjemah teks, pengenalan suara, dan sistem bantuan pengemudi tingkat lanjut. Proses alur kerja dari *deep learning* ditunjukkan pada Gambar 2.9.

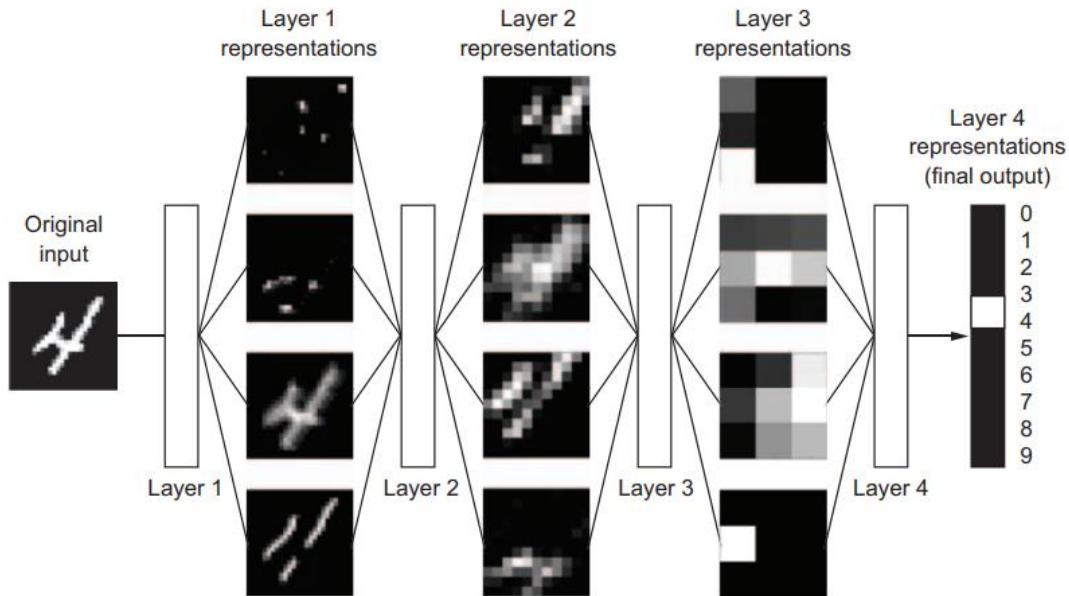


Gambar 2.9 Proses Perancangan dan Deployment dari Sistem Berbasis Deep Learning

Deep learning pada hakikatnya merupakan proses mendekripsi atau mengenali karakteristik dari sebuah objek yang merupakan representasi yang sederhana tetapi dengan konsep *deep learning* komputer dapat membangun suatu sistem yang kompleks. *Deep learning* yang merupakan pengembangan dari jaringan saraf biasa yang memiliki sedikit jaringan, kemudian dikembangkan

menjadi jaringan saraf yang terdiri dari banyak jaringan yang membuat jaringan menjadi lebih dalam (*deep*) dan oleh karena itu diberi nama *deep learning*. *Deep learning* dapat melakukan representasi terhadap objek berupa data gambar, video, atau teks. Representasi algoritma *deep learning* dapat dilihat contohnya pada Gambar 2.10, bisa dilihat bagaimana proses jaringan saraf mengubah gambar digit menjadi representasi dari gambar asli dengan mempelajari karakteristik data dari gambar digit dan meningkatkan informasi di hasil akhirnya [18].

Dari gambar 2.10, dapat dilihat contoh proses algoritma *deep learning* untuk klasifikasi *handwritten digits*. Pada *layer input*, menggunakan data masukan berupa *handwritten digits* atau tulisan angka menggunakan tulisan tangan yang dibuat secara digital, kemudian metode *deep learning* membuat representasi dari data yang kemudian diteruskan ke lapisan selanjutnya untuk dideteksi lagi karakteristik bagian lainnya, begitu seterusnya sehingga komputer dapat merepresentasikan data masukan secara keseluruhan, dan menampilkan nilai hasil klasifikasi data tersebut, yang dalam hal ini lapisan terakhir menampilkan *final output* berupa nilai dari angka hasil klasifikasi. Dan begitu seterusnya hingga algoritma *deep learning* dapat melatih dirinya sendiri untuk meningkatkan akurasi dalam melakukan klasifikasi, seperti halnya sistem saraf otak manusia yang sangat kompleks dapat belajar dari data masukan yang kita lihat atau rasakan, kemudian mengolahnya dengan menganalisis karakteristiknya, hingga akhirnya memunculkan hasil keluaran yang tepat.



Gambar 2.10 Representasi deep learning dalam model klasifikasi digit

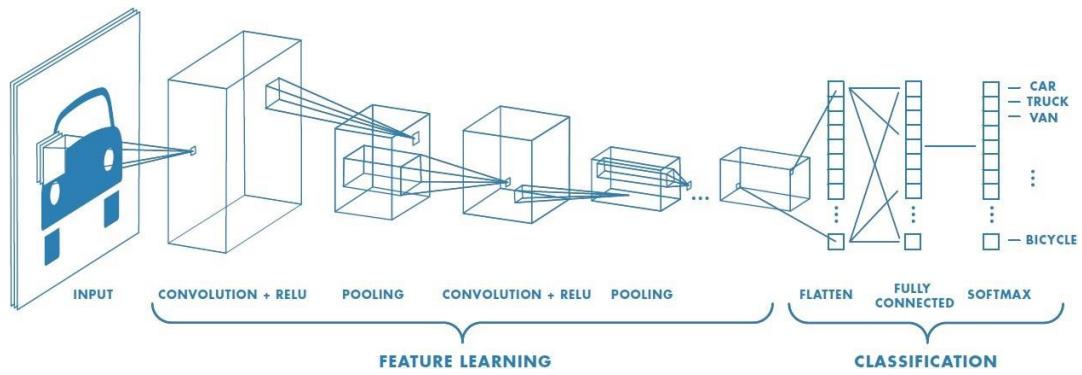
2.4 Convolutional Neural Network (CNN)

2.4.1 Konsep Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu bentuk kemajuan dalam bidang *computer vision* dengan penggunaan metode *deep learning* yang telah dibangun dan disempurnakan seiring waktu untuk mengolah data dua dimensi. CNN bekerja dengan cara mengambil gambar input, memilih dan menetapkan kepentingan karakteristik (bobot dan bias yang dapat dipelajari) untuk berbagai aspek/objek dalam gambar dan dapat membedakan satu dari yang lain. Pra-pemrosesan yang diperlukan dalam CNN jauh lebih rendah dibandingkan dengan algoritma klasifikasi lainnya. Sementara dalam metode tradisional sebelumnya, filter direkayasa dengan manual, dengan pelatihan yang cukup untuk mendapatkan hasil yang maksimal [19].

CNN bekerja dengan memanfaatkan proses konvolusi dengan menggerakan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah data

masukan, komputer mendapatkan informasi representatif baru dari hasil konvolusi data masukan dengan filter yang digunakan [20].



Gambar 2.11 Ilustrasi Arsitektur Metode CNN

Secara garis besar, alur proses metode CNN dibagi menjadi 2 bagian, yaitu bagian *Feature Learning/Feature Detection* dan bagian *Classification* yang ilustrasinya dapat dilihat pada Gambar 2.4. Bagian *feature learning* merupakan bagian yang berfungsi untuk melakukan ekstraksi karakteristik (*feature*) dari input yang diberikan dengan tujuan untuk mendeteksi fitur dari input yang merepresentasikan data input tersebut. Pada bagian ini dilakukan operasi-operasi pada data input, yaitu *convolution*, *pooling*, atau *activation* seperti *activation ReLU*.

Kemudian, bagian selanjutnya adalah bagian *classification*. Bagian ini merupakan lanjutan setelah data input dilakukan deteksi atau ekstraksi fitur. Bagian ini terhubung sebagai *fully connected layer* yang menghasilkan dimensi K di mana K adalah jumlah kelas yang dapat diprediksi/diklasifikasi oleh jaringan saraf CNN. Pada lapisan ini, arsitektur CNN menggunakan fungsi *activation softmax* untuk memberikan hasil klasifikasi berupa vektor yang berisi probabilitas untuk setiap kelas dari input terhadap label yang diklasifikasikan [17].

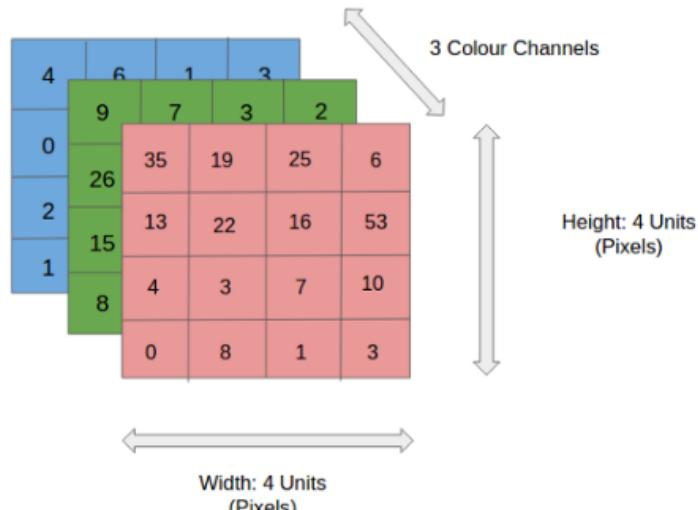
2.4.2 Arsitektur Jaringan *Convolutional Neural Network (CNN)*

Jaringan saraf CNN dibedakan dari jaringan saraf lainnya karena kinerja superiornya dengan input sinyal gambar atau audio. CNN memiliki lima komponen *layer* utama, yaitu:

1. Lapisan Masukan (*Input Layer*)

Input layer menampung nilai *pixel* dari citra yang menjadi masukan.

Misalnya, untuk citra dengan ukuran *pixel* 64×64 dengan 3 *channel* warna, RGB (*Red, Green, Blue*) maka yang menjadi data masukan adalah *pixel array* berukuran $64 \times 64 \times 3$ [21].



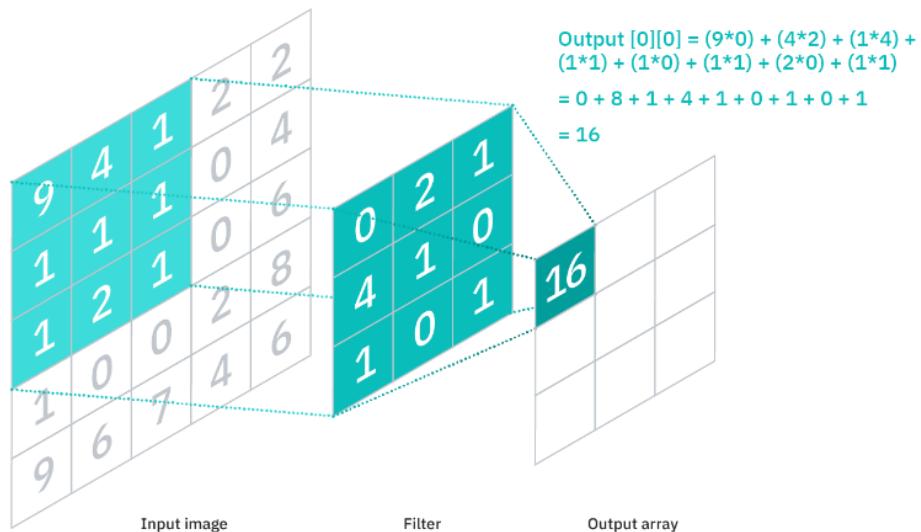
Gambar 2.12 Contoh input dengan pixel ukuran 4×4 dan 3 channel warna RGB

Peran CNN adalah untuk mereduksi gambar masukan menjadi bentuk yang lebih mudah untuk diproses tanpa kehilangan fitur yang penting untuk mendapatkan hasil prediksi yang maksimal.

2. Lapisan Konvolusi (*Convolutional layer*)

Convolution Layer adalah inti dari metode CNN. *Convolution Layer* menghasilkan citra baru yang merepresentasikan fitur dari gambar

masukan dengan menerapkan proses konvolusi. Dalam proses ini, akan dilakukan operasi konvolusi (perkalian “dot”) antara citra masukan dan nilai dari filter tersebut sehingga menghasilkan sebuah keluaran yang biasa disebut *feature map* yang kemudian akan digunakan pada *activation layer*.



Gambar 2.13 Ilustrasi Proses Konvolusi Pada Sebuah Citra

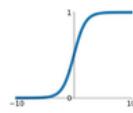
3. Lapisan Aktivasi (*Activation Layer*)

Activation Layer adalah *layer* dimana *feature map* dimasukkan ke dalam fungsi aktivasi. Fungsi aktivasi digunakan untuk mengubah nilai-nilai pada *feature map* pada rentang nilai tertentu dengan tujuan untuk menampilkan fitur dominan dari gambar yang masuk ke *layer* selanjutnya. Fungsi aktivasi yang paling umum digunakan terhadap *feature map* adalah fungsi aktivasi *ReLU* karena sifatnya yang bekerja dengan baik pada bagian *Convolutional Layer*.

Activation Functions

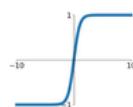
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



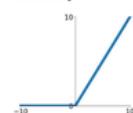
tanh

$$\tanh(x)$$



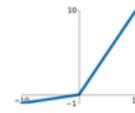
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

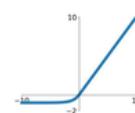


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

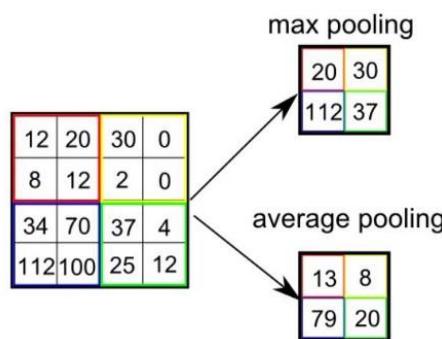
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Gambar 2.14 Macam-macam fungsi aktivasi

4. Lapisan Pooling (Pooling layer)

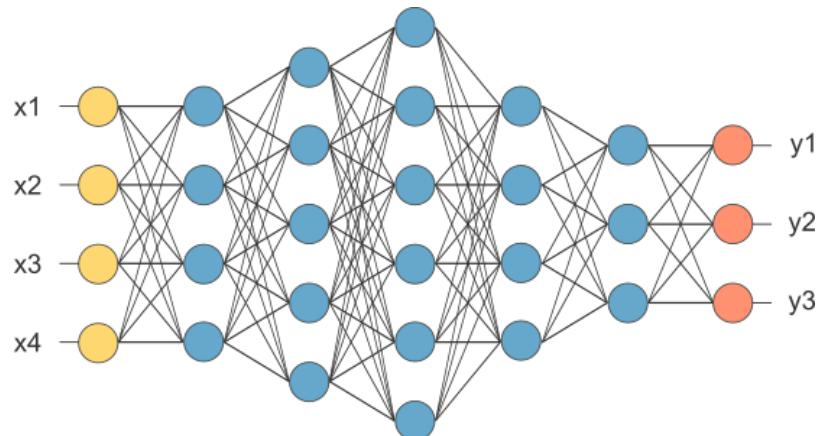
Mirip dengan *Convolutional Layer*, *Pooling Layer* juga dikenal sebagai *downsampling* karena berfungsi untuk mengurangi ukuran jumlah dari parameternya untuk mengurangi daya dan mempercepat komputasi yang diperlukan untuk memproses data dengan mengurangi dimensi dari *feature map*. Selain itu, berguna untuk mengekstraksi fitur dominan yang invariant rotasi dan posisinya, sehingga mempertahankan proses pelatihan model secara efektif [19]. Ada dua jenis *pooling*, yaitu *max pooling* dan *average pooling*. *Max pooling* bekerja dengan cara mengambil nilai maksimum dari bagian gambar yang dicakup oleh kernel. Di sisi lain, *average pooling* mengambil nilai rata-rata semua nilai dari bagian gambar yang dicakup oleh kernel.



Gambar 2.15 Tipe Pooling

5. Lapisan Terhubung Sepenuhnya (*Fully-connected layer*)

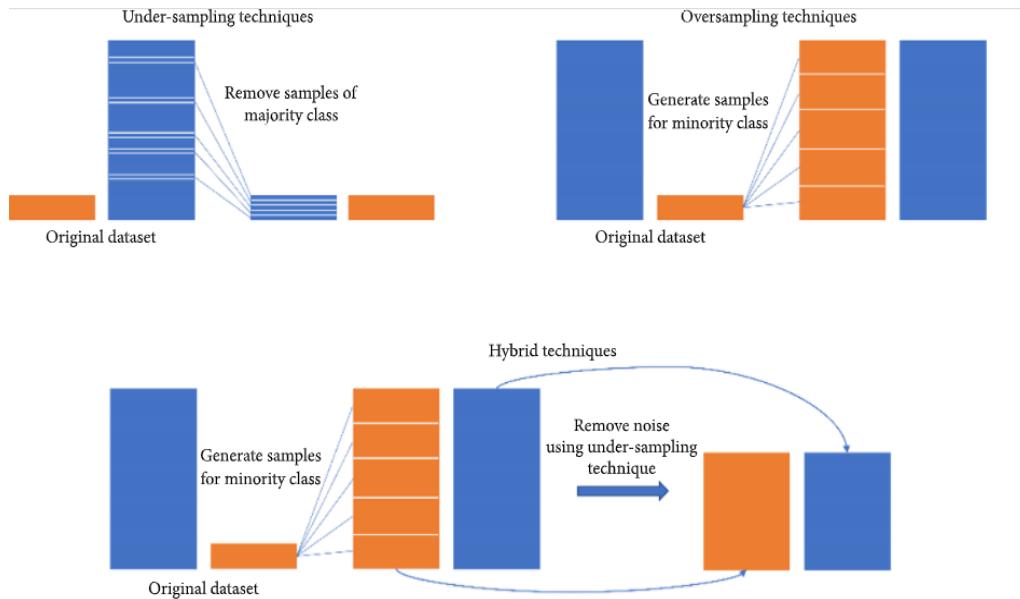
Fully Connected layer merupakan *layer* di mana semua jaringan saraf dari lapisan sebelumnya terhubung dengan jaringan saraf di lapisan selanjutnya. Lapisan ini berfungsi untuk melakukan klasifikasi berdasarkan fitur yang sudah diekstraksi pada lapisan sebelumnya dan filter yang berbeda. Pada layer ini, terdapat fungsi aktivasi *softmax* untuk mengklasifikasikan input dengan tepat sesuai dengan label sebenarnya yang ingin diklasifikasikan, dengan menghasilkan probabilitas dari 0 hingga 1.



Gambar 2.16 Fully Connected Layer

2.5 Teknik *Resampling* Data

Dalam menangani data yang tidak seimbang dapat diatasi dengan teknik *resampling* yang bertujuan untuk membuat kumpulan sampel yang baru. Teknik *resampling* akan memanipulasi distribusi data dengan memperbaiki jumlah data, sehingga didapatkan *dataset* dengan distribusi data antar kelas yang seimbang. Terdapat beberapa teknik *resampling*, secara garis besar teknik *resampling* dibagi menjadi tiga macam, yaitu *undersampling*, *oversampling*, dan *hybrid sampling*. Untuk ilustrasi teknik *resampling* dapat dilihat pada Gambar 2.10 di bawah ini.



Gambar 2.17 Ilustrasi Teknik Resampling Data

2.5.1 Teknik *Undersampling*

Metode *undersampling* ini digunakan ketika jumlah data mencukupi.

Undersampling menghasilkan jumlah sampel yang sama di setiap kelas dengan cara mengurangi ukuran kelas yang mayoritas sampai semua kelas memiliki jumlah sampel yang sama. Meskipun mungkin tidak tampak intuitif, ada beberapa kasus bahwa dalam beberapa situasi, *undersampling* lebih baik daripada *oversampling*.

Kerugian yang signifikan dari metode ini adalah membuang sebagian data yang tersedia. Untuk mengatasi kekurangan ini, beberapa modifikasi diperkenalkan pendekatan yang lebih hati-hati memilih sampel yang akan dihapus. Misalnya, seleksi satu sisi mengurangi sampel berlebihan yang posisinya dekat dengan batas antar kelas. Pendekatan yang lebih umum dari *undersampling* adalah dekontaminasi (menghilangkan) data yang dapat melibatkan pelabelan ulang beberapa sampel [22].

Ada beberapa metode dalam melakukan *resampling dataset* menggunakan teknik *undersampling*, antara lain sebagai berikut.

1. *Random Under Sampler*

Random Under Sampler adalah pendekatan sederhana untuk *resampling* dengan menghilangkan data sampel secara acak sampai rasio antara kelas minoritas dan mayoritas berada pada tingkat yang diinginkan. Metode ini merupakan cara cepat dan mudah untuk menyeimbangkan data dengan memilih subset data secara acak untuk kelas yang ditargetkan [23].

2. *Cluster Centroids*

Metode *Cluster Centroids* mengurangi kelas mayoritas dengan mengganti *cluster* sampel mayoritas dengan *cluster centroid* dari algoritma *KMeans*. Algoritma ini menyimpan N sampel mayoritas dengan menyesuaikan algoritma *KMeans* dengan N *cluster* ke kelas mayoritas dan menggunakan koordinat N *cluster centroid* sebagai sampel mayoritas baru [23].

3. *Tomek Link*

Tomek Links merupakan salah satu modifikasi dari teknik *undersampling Condensed Nearest Neighbors* yang dikembangkan oleh Tomek (1976). Berbeda dengan metode *Condensed Nearest Neighbors* yang hanya secara acak memilih sampel dengan *K-nearest neighbor* dari kelas mayoritas yang ingin dihilangkan, metode *Tomek Links* menggunakan aturan untuk memilih pasangan observasi (katakanlah, a dan b) yang memenuhi sifat: Pengamatan tetangga terdekat a adalah b. Tetangga terdekat dari pengamatan b adalah a.

Pengamatan a dan b termasuk dalam kelas yang berbeda. Artinya, a dan b masing-masing milik kelas minoritas dan mayoritas (atau sebaliknya) [24].

4. *Edited Nearest Neighbours (ENN)*

Metode ENN bekerja dengan cara mencari *K-nearest neighbor* (tetangga terdekat) dari setiap observasi terlebih dahulu, kemudian mengecek apakah kelas mayoritas dari *K-nearest neighbor* observasi tersebut sama dengan kelas observasinya atau tidak. Jika kelas mayoritas dari *K-nearest neighbor* observasi dan kelas observasi berbeda, maka observasi dan *K-nearest neighbor* dihapus dari *dataset* [25].

2.5.2 Teknik *Oversampling*

Oversampling digunakan ketika jumlah data tidak mencukupi. *Oversampling* berusaha untuk meningkatkan jumlah anggota kelas minoritas dalam *dataset*. Keuntungan dari *oversampling* adalah tidak ada informasi dari *dataset* asli yang hilang. Namun, kerugiannya adalah meningkatkan ukuran data. Dengan demikian, akan meningkatkan waktu pelatihan dan jumlah memori yang diperlukan untuk melakukan pelatihan model [26].

Ada beberapa metode dalam melakukan *resampling dataset* menggunakan teknik *oversampling*, antara lain sebagai berikut [23].

1. *Random Over Sampler*

Random over sampler menyeimbangkan data dengan melakukan *oversampling* secara acak pada kelas minoritas. *Oversampling* memang terbukti efektif dalam menangani *dataset* yang tidak seimbang, namun karena sampel data diambil

secara acak untuk menyamakan jumlah data antar kelas, hal itu dapat menyebabkan model mengalami *overfitting*.

2. SMOTE (*Synthetic Minority Over-sampling Technique*)

Algoritma SMOTE sederhana namun efektif, mengungguli *random oversampler* dalam beberapa masalah dimensi rendah. SMOTE bekerja dengan cara membuat data sintetis (buatan) dari data minoritas dengan mencari *k-nearest neighbors* (yaitu ketetanggaan terdekat data sebanyak k) untuk setiap data di kelas minoritas [8]. Jumlah *k-nearest neighbors* ditentukan dengan mempertimbangkan kemudahan dalam melakukannya ataupun dapat dipilih secara acak.

3. *Borderline* SMOTE

Tidak seperti metode *over-sampling* yang ada, *Borderline* SMOTE hanya melakukan *oversample* atau memperkuat sampel-sampel pada *borderline* kelas minoritas. Data sintetis baru dihasilkan di sepanjang garis antara contoh garis batas minoritas dan tetangga terdekat mereka dari kelas yang sama, sehingga memperbanyak sampel *borderline* [27].

4. *Adaptive Synthetic* (ADASYN)

ADASYN melakukan *oversampling* dengan cara menghasilkan sampel data kelas minoritas secara adaptif sesuai dengan bobot distribusi tingkat kesulitan belajar, data sintetis dibuat lebih banyak untuk sampel kelas minoritas yang sulit belajar dibandingkan sampel kelas minoritas yang lebih mudah belajar dalam proses pelatihan [28].

2.5.3 Teknik *Resampling* Gabungan (*Hybrid Resampling*)

Ada beberapa metode dalam melakukan *resampling dataset* menggunakan teknik *resampling* gabungan (*hybrid*), antara lain sebagai berikut: [23]

1. Gabungan SMOTE dengan *Tomek Link*

Metode ini menggabungkan metode *oversampling* SMOTE untuk memperbanyak sampel data buatan untuk kelas minoritas dan metode *undersampling* *Tomek Links* untuk menghapus sampel data dari kelas mayoritas yang paling dekat dengan data kelas minoritas atau yang bisa juga disebut sebagai *Tomek Links*.

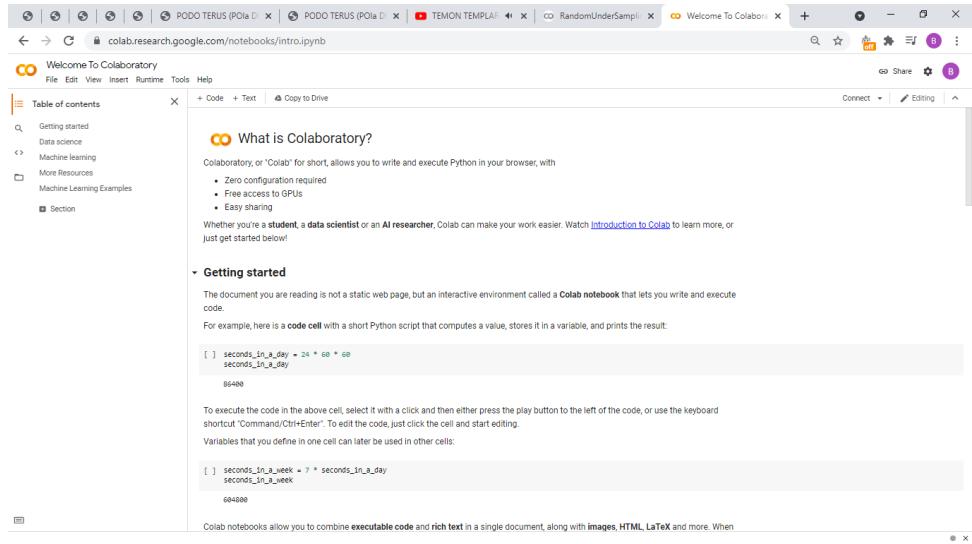
2. Gabungan SMOTE dengan ENN

Metode ini menggabungkan metode *oversampling* SMOTE untuk memperbanyak sampel data buatan untuk kelas minoritas dan metode *undersampling* ENN untuk menghapus beberapa observasi dari kedua kelas yang diidentifikasi memiliki kelas yang berbeda antara kelas observasi dan kelas mayoritas *K-nearest neighbor*-nya.

2.6 *Google Colaboratory*

Google Colaboratory adalah salah satu produk *Google* berbasis *cloud* yang bisa digunakan secara gratis [29]. *Colaboratory*, atau dapat disingkat "Colab", adalah produk dari *Google Research*. *Colab* adalah *coding environment* bahasa pemrograman *Python* dengan format "*Jupyter Notebook (.ipynb)*" yang di-hosting yang tidak memerlukan persiapan untuk digunakan, sekaligus memberikan akses gratis ke sumber daya komputasi termasuk GPU. *Colab* memungkinkan siapa saja untuk menulis/membuat dan mengeksekusi kode *jupyter notebook* dengan Bahasa

pemrograman *python* melalui *browser*, dan sangat cocok untuk pembelajaran mesin, analisis data, dan pendidikan.



Gambar 2.18 Tampilan Awal Google Colaboratory

Banyak sekali kelebihan yang diberikan oleh *Google Colaboratory*, antara lain sebagai berikut:

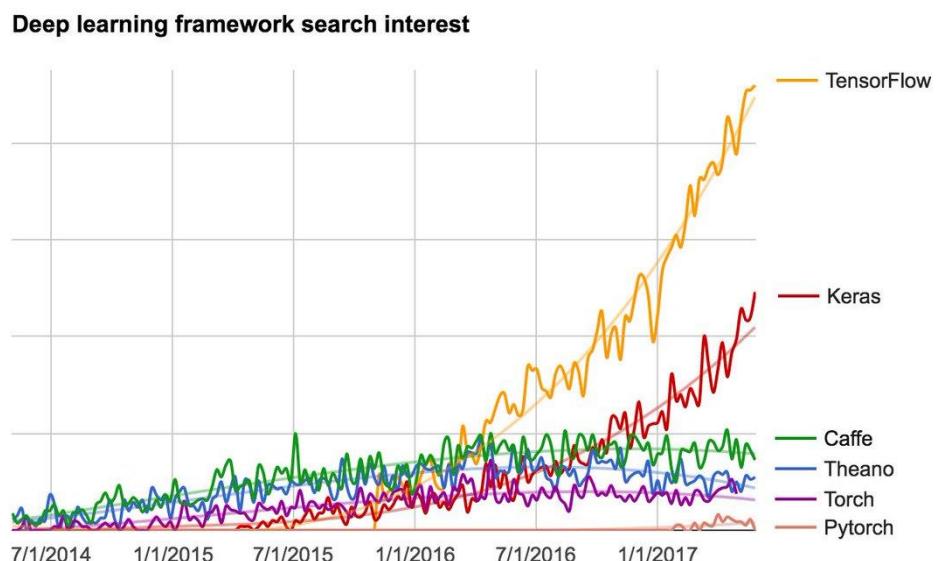
1. GPU Gratis. *Google Colab* memberikan komputer dengan spesifikasi tinggi berupa GPU Tesla, RAM ~12GB, disk ~300GB yang masih bisa disambungkan dengan *Google Drive*, dan akses internet cepat untuk mengunduh data besar. Hal ini akan sangat membantu bagi orang-orang yang memiliki komputer dengan spesifikasi kurang mumpuni.
2. Kolaborasi. *Google Colab* memudahkan proses kolaborasi dengan orang lain karena *coding* yang dibuat di *Google Colab* dapat dikerjakan dengan berkolaborasi, sehingga tidak perlu memindah-mindahkan data karena *file notebook* yang dibuat tersimpan di *Cloud* dan dapat langsung digunakan.

3. Mudah berintegrasi. *Google Colab* sangat fleksibel dalam hal ini karena dapat diintegrasikan dengan *Jupyter Notebook*, *Google Drive*, atau dengan *Github*. Misalnya, integrasi ke *Google Drive* dapat membuat pengguna membagikan, memberi komentar, dan kolaborasi pada *file notebook*.
4. Fleksibel. *Google Colab* sangat fleksibel karena pada dasarnya hanya perlu dijalankan pada browser dan tidak perlu melakukan konfigurasi yang rumit. Bahkan, *Google Colab* dapat dijalankan di *smartphone* yang tentunya hal ini akan sangat membantu jika ingin digunakan dalam keadaan mendesak dan tidak ada komputer.

2.7 Framework TensorFlow dan Keras

2.7.1 Framework TensorFlow

Saat ini cukup banyak *framework* yang digunakan dalam mengolah *deep learning*, di antaranya adalah *TensorFlow*, *Keras*, *MXNet*, dan *PyTorch*. Gambar 2.11 menampilkan grafik tingkat kepopuleran *framework deep learning* dari 2014 hingga 2017 [30].



Gambar 2.19 Popularitas framework deep learning pada tahun 2014-2017

TensorFlow adalah *framework open source end-to-end* dan gratis yang dikembangkan oleh Google Brain Team untuk pembelajaran mesin yang paling terkenal di dunia saat ini. *TensorFlow* memberikan kemudahan bagi *developer* yang perlu melakukan *debug* dan melakukan introspeksi ke dalam aplikasi *TensorFlow*. *TensorFlow* dapat digunakan dalam berbagai tugas tetapi memiliki fokus khusus pada pelatihan dan inferensi *deep neural network*. *TensorFlow* memungkinkan peneliti dan pengembang dengan mudah membangun dan menerapkan aplikasi yang didukung *machine learning*.

Berikut adalah beberapa fitur dari *TensorFlow*, yaitu: [17]

1. Dapat melakukan *debug* lebih cepat dengan *tool Python*.
2. Model dinamis dengan aliran kontrol *Python*.
3. Menawarkan berbagai tingkat abstraksi, yang membantu dalam membuat dan melatih model.
4. Memungkinkan untuk melatih dan menggunakan model dengan cepat.
5. Menyediakan fleksibilitas dan kontrol dengan fitur seperti API dan *model keras functional*.

2.7.2 Framework Keras

Keras adalah *High Neural Network Application Programming Interface* (API), dalam bahasa pemrograman *Python* dan mampu berjalan di atas *TensorFlow*, CNTK, atau *Theano*. *Keras* memiliki antarmuka yang ramah dan sangat produktif untuk memecahkan permasalahan yang menggunakan *machine learning* dengan fokus pada *deep learning* modern. *Framework* ini memiliki fokus agar memungkinkan proses eksperimen yang cepat dan mudah [18].

Keras memiliki fitur-fitur yang dapat digunakan pengguna untuk kebutuhan *deep learning library*, antara lain:

1. Memungkinkan pembuatan prototipe yang mudah dan cepat (melalui keramahan pengguna, modularitas, dan ekstensibilitas) [17].
2. Dukungan *built-in* untuk *Convolutional Neural Networks* (CNN) untuk keperluan *computer vision*, *Recurrent Neural Networks* (RNN) untuk keperluan *sequence processing*, dan kombinasi keduanya.
3. Mampu berjalan lancar di CPU dan GPU.

BAB 3

METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian dilaksanakan dalam waktu 4 bulan dari bulan Juli 2021 hingga bulan Oktober 2021 dilakukan secara daring di lingkungan tempat tinggal penulis.

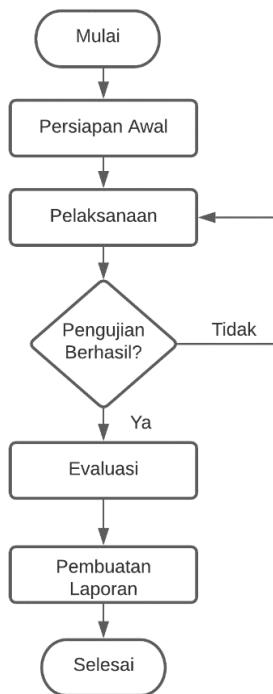
3.2 Alat dan Bahan

Dalam penelitian tugas akhir ini, daftar alat dan bahan yang digunakan adalah sebagai berikut.

1. Perangkat Keras
 - a. *Laptop HP* dengan spesifikasi *processor AMD A9-9425, RAM 4 GB.*
2. Perangkat Lunak
 - a. Sistem operasi *Windows 10 64 bit.*
 - b. Peramban internet *Google Chrome* versi 92.0.4515.131.
 - c. *Google Colaboratory*
 - d. *Microsoft Visual Studio Code* versi 1.58.2.
 - e. *Python for Windows* versi 3.9.6 64 bit.
3. *Dataset* kanker kulit MNIST: HAM10000 yang didapatkan dari *Kaggle* yang diunggah oleh K. Scott Mader untuk pelatihan dan pengujian berupa 10015 gambar kanker kulit berdasarkan kondisi lesi kulit berpigmen yang terbagi menjadi 7 kelas dengan pembagian kelas, yaitu *Melanocytic nevi, melanoma, benign keratosis-like lesions, basal cell carcinoma, actinic keratoses, vascular lesions, dan dermatofibroma.*

3.3 Alur dan Tahap Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan, dimulai dari tahap persiapan awal, yaitu menyiapkan alat dan bahan serta studi pustaka, yang dilanjutkan dengan tahap pelaksanaan dengan membuat desain arsitektur, tahap pengujian dan tahap evaluasi untuk membandingkan teknik *resampling dataset* yang paling baik pada model klasifikasi kanker kulit, hingga tahap terakhir adalah pembuatan laporan. Untuk diagram alir tahapan penelitian ditunjukkan pada Gambar 3.1 di bawah ini.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Penelitian tugas akhir ini dilaksanakan melalui beberapa tahapan sebagai berikut.

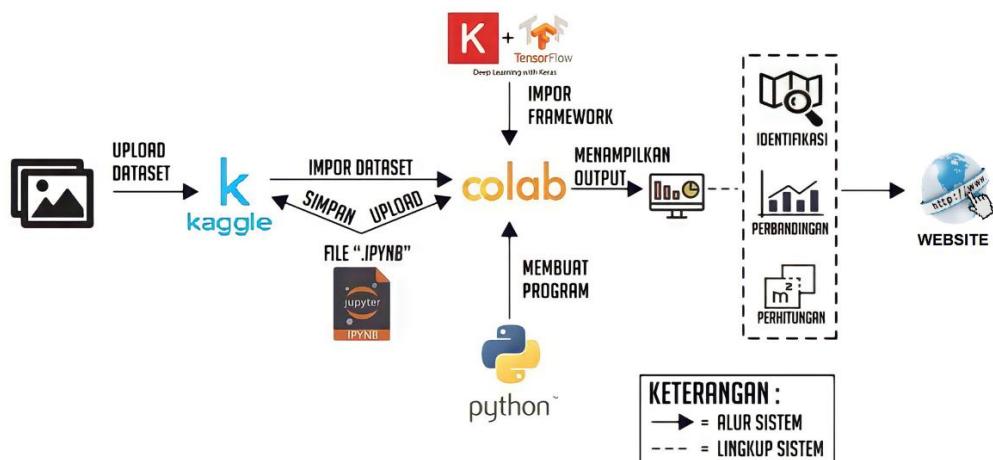
3.3.1 Tahap Persiapan

Pada tahap ini merupakan persiapan awal yang dilakukan oleh penulis. Pada tahap ini, penulis melakukan studi pustaka melalui sumber *website*, jurnal,

buku dan diskusi dengan teman. Penulis mencari materi-materi terkait pembuatan model klasifikasi kanker kulit, *deep learning*, *Convolutional Neural Network* (CNN), cara menyeimbangkan *dataset* dengan teknik *resampling dataset*, dan cara membuat tampilan *website* untuk antarmuka pengguna. Dan juga penulis menyiapkan alat dan bahan yang diperlukan dalam melakukan penelitian.

3.3.2 Tahap Pelaksanaan

Pada tahap ini penulis melakukan persiapan untuk mengolah data agar bisa mengklasifikasikan kanker kulit sesuai yang diharapkan. Pada tahap ini dilakukan persiapan dan pra-pemrosesan *dataset*, mendesain model arsitektur klasifikasi yang dipakai, melakukan pengujian, dan melakukan perbandingan hasil.



Gambar 3.2 Tahapan Pelaksanaan Penelitian

Persiapan

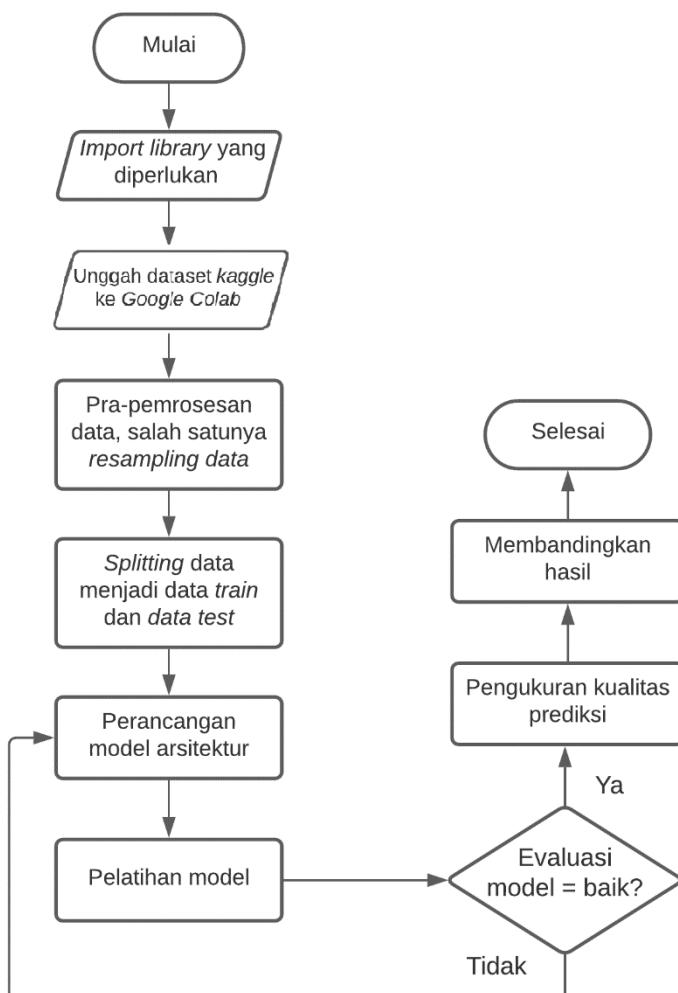
Pada tahap ini, penulis melakukan pengumpulan dataset dan menyiapkan beberapa penelitian terkait *dataset* HAM10000 dan pembuatan model arsitektur klasifikasi kanker kulit sebagai panduan untuk melakukan penelitian ini. Dalam hal

ini penulis menggunakan *dataset* HAM10000 yang disediakan di *Kaggle*. Setelah itu, melakukan analisa isi *dataset* tersebut.

Perancangan Model Arsitektur

Pada tahap ini, penulis membuat model klasifikasi kanker kulit dengan metode *Convolutional Neural Network* (CNN) dengan menggunakan *Google Colab* sebagai platform untuk menjalankan file *Jupyter Notebook* (*.ipynb*) dan menggunakan *framework TensorFlow* dan *keras* sebagai *framework* utamanya.

Diagram alir pembuatan desain model arsitektur ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram Alir Tahap Pembuatan Desain Arsitektur

Pada tahap ini, penulis memilih merancang sendiri arsitektur model klasifikasi kanker kulit tanpa menggunakan *transfer learning*. Model yang dirancang menggunakan jaringan *Convolutional Neural Network* (CNN). Sebelum merancang model, *dataset* HAM10000 harus dilakukan penyeimbangan data. Untuk menyeimbangkan *dataset*, penulis melakukan uji coba beberapa teknik *resampling*, yaitu teknik *random under sampler*, *cluster centroids undersampling*, *random over sampler*, SMOTE, ADASYN, Gabungan SMOTE dengan *Tomek Link*, dan Gabungan SMOTE dengan ENN.

Dataset citra pelatihan dan pengujian yang digunakan dilakukan *image augmentation* untuk meningkatkan kualitas data dan menghindari model *overfitting*. Pelatihan model dilakukan dengan batas iterasi (*epoch*) sebanyak 70 kali dan dengan memanfaatkan metode *callbacks* dari *framework keras*, antara lain *callbacks Model Checkpoint* untuk menyimpan model atau *weights* (dalam *checkpoint file*) dalam kualitas terbaik setiap interval tertentu yang ditentukan, kemudian *callbacks ReduceLROnPlateau* untuk mengurangi nilai koreksi bobot (*learning rate*) ketika metrik yang ditinjau saat pelatihan tidak mengalami perubahan yang baik, dan *callbacks Early Stopping* untuk menghentikan pelatihan sebelum semua iterasi (*epoch*) selesai jika metrik yang ditinjau tidak lagi mengalami perubahan yang baik.

3.3.3 Tahap Pengujian

Setelah arsitektur model dirancang, tahap selanjutnya adalah pengujian model. Pada tahap ini, penulis melakukan pengujian keseluruhan terhadap sistem yang telah dirancang untuk mengetahui kualitas performa model. Seluruh *dataset*

pelatihan dilatih terhadap masing-masing *dataset* yang telah dilakukan penyeimbangan kualitas distribusi datanya dengan teknik *resampling dataset*. Kemudian, setelah pelatihan dilakukan pengujian atau evaluasi kualitas prediksi menggunakan metode *model.evaluate* untuk menguji dan mengukur kualitas prediksi dari data *test*, seperti nilai *accuracy, loss, true positive, false positive, true negative, false negative, precision, dan recall*. Kemudian, menggunakan metode *classification_report* dari *library scikit learn* untuk mengukur nilai *precision, recall, dan F1-score* dari tiap kelas yang diprediksi oleh model tersebut.

3.3.4 Tahap Evaluasi

Pada tahap ini penulis melakukan evaluasi terhadap hasil dari pengujian yang selanjutnya dilakukan identifikasi, perbandingan, dan perbaikan sistem, mulai dari arsitektur model klasifikasi, teknik *resampling* yang digunakan, dan parameter yang digunakan pada metode *callbacks* saat pelatihan hingga menghasilkan model klasifikasi kanker kulit yang memiliki kualitas prediksi baik dan dapat digunakan oleh banyak orang.

3.3.5 Tahap Pembuatan Laporan

Tahap pembuatan laporan adalah tahap terakhir dalam penelitian ini. Setelah sistem klasifikasi kanker kulit menghasilkan kualitas prediksi yang baik sesuai yang diharapkan, maka disusunlah laporan tugas akhir berjudul “PERBANDINGAN TEKNIK RESAMPLING PADA DATASET HAM10000 TERHADAP PERFORMA MODEL KLASIFIKASI KANKER KULIT”

3.4 Waktu dan Jadwal Penelitian

Penelitian dilaksanakan dalam waktu 4 bulan dimulai dari bulan Juli 2021 sampai dengan bulan Oktober 2021 dengan rincian jadwal kegiatan penelitian tugas akhir ditampilkan pada Tabel 3.1 berikut ini.

Tabel 3.1 Waktu dan Jadwal Penelitian

BAB 4

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem dan *Dataset* Penelitian

Pada bagian perancangan sistem dan *dataset* penelitian mulai dari pengunggahan *dataset* hingga mengevaluasi model dilakukan pada *Google Colaboratory*.

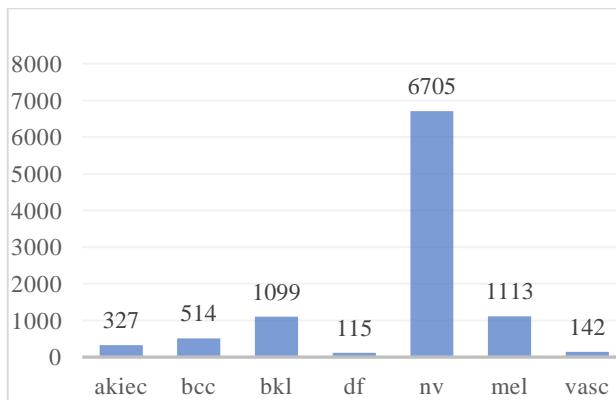
4.1.1 Sumber *Dataset*

Dataset yang digunakan dalam penelitian ini adalah *dataset* kanker kulit HAM10000 yang merupakan koleksi gambar dermatoskopik dari lesi berpigmen yang terdiri dari 10015 gambar dermatoskopik yang dikumpulkan selama 20 tahun dari dua lokasi berbeda, yaitu *Department of Dermatology at the Medical University of Vienna*, Austria, dan *the skin cancer practice of Cliff Rosendahl in Queensland*, Australia. Situs Australia menyimpan gambar dan meta-data dalam *file PowerPoint* dan *database Excel*. Lebih dari 50% lesi sudah dikonfirmasi melalui histopatologi (*histo*), sementara uji kebenaran untuk sisanya juga sudah dilakukan pemeriksaan lanjutan (*followup*), konsensus ahli (*konsensus*), atau dikonfirmasi dengan mikroskop *confocal in-vivo* [4].

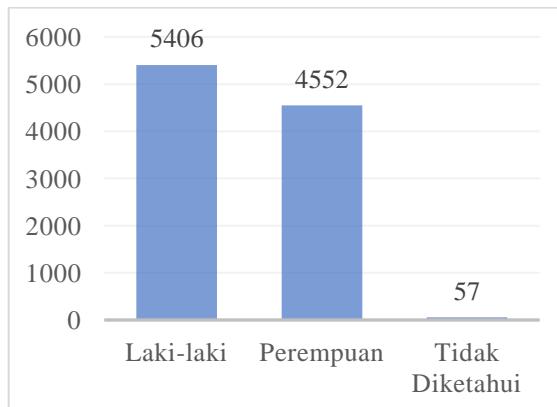
Dataset yang digunakan peneliti diambil dari platform *Kaggle* yang diunggah oleh K. Scott Mader yang berisi folder berisi 10015 citra lesi kulit, metadata lesi kulit dalam file bernama HAM10000_metadata.csv, dan file berisi data tentang nilai *pixel* tiap citra lesi kulit dengan dimensi warna *red, green, blue* (RGB) dalam file bernama hmnist_28_28_RGB.csv.

4.1.2 Pengunggahan *Library* dan *Dataset*

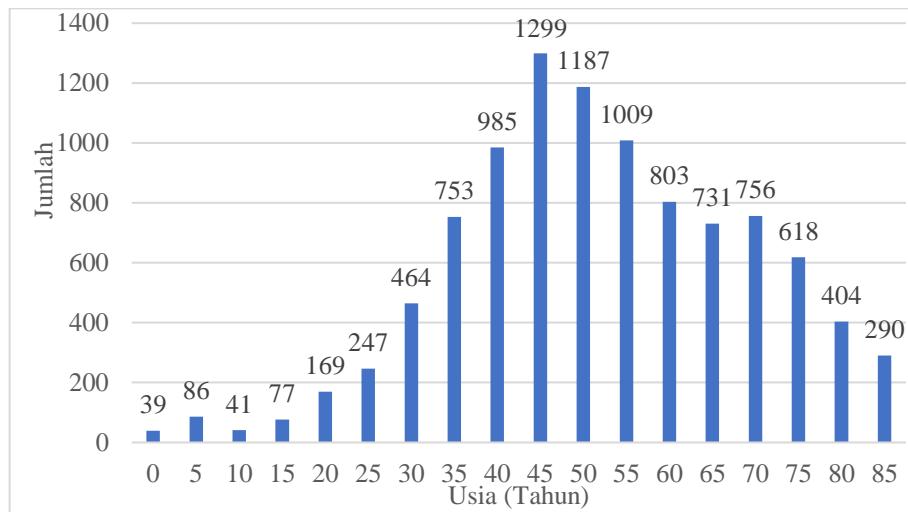
Pada tahap awal membuat program klasifikasi kanker kulit, pertama-tama penulis mengunggah *library* yang dibutuhkan seperti *TensorFlow*, *Keras*, *imbalanced-learn*, *matplotlib*, *numpy*, *pandas*, dan lainnya. Setelah itu mengunggah *dataset* dari *Kaggle* ke *Google Colaboratory* menggunakan bahasa pemrograman *python*. Kemudian, menganalisis terhadap isi data dari *dataset* HAM10000. Setelah dianalisis maka dapat diketahui detail dari isi dataset yang digunakan. Di bawah ini akan ditampilkan grafik distribusi kelas kanker kulit, jenis kelamin pengidap kanker kulit, usia pengidap kanker kulit, *localization* (lokasi kanker kulit di tubuh) yang ada pada *dataset* HAM10000.



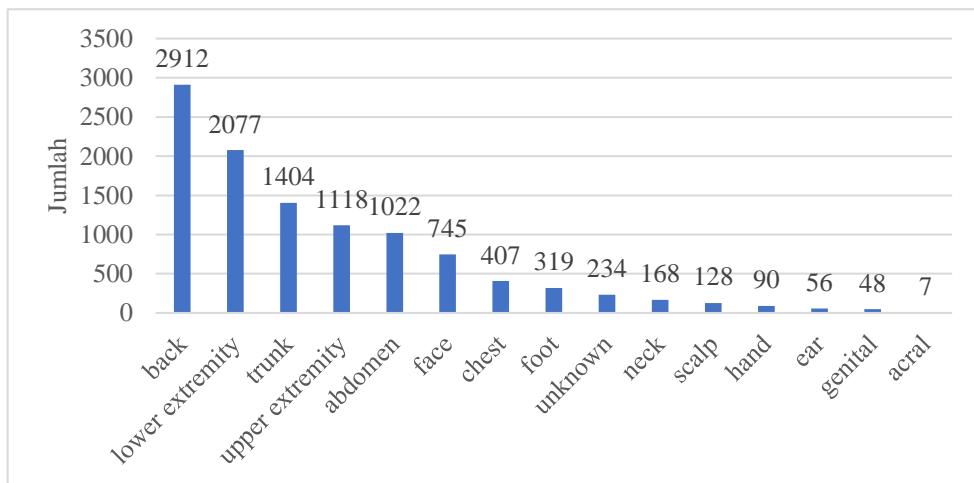
Gambar 4.1 Grafik Distribusi Kelas Kanker Kulit Pada Dataset HAM10000



Gambar 4.2 Distribusi Data Jenis Kelamin Pada Dataset HAM10000



Gambar 4.3 Distribusi Usia Pada Dataset HAM10000



Gambar 4.4 Lokasi Penyakit Kanker Kulit Pada Dataset HAM10000

Dari grafik distribusi data tipa kelas kanker kulit pada Gambar 4.1, kita bisa melihat bahwa *dataset* yang digunakan memiliki distribusi yang sangat tidak seimbang kelas *Melanocytic Nevi* (nv) memiliki jumlah data 6705 citra lesi kulit, sementara kelas *dermatofibroma* (df) hanya memiliki jumlah data 115 citra, jika model yang dirancang tetap menggunakan *dataset* awal ini, maka model yang didapatkan akan mengalami *overfitting* (kecenderungan) memprediksi kelas nv karena pada saat pelatihan, program memiliki data latih yang sangat banyak untuk

kelas tersebut. Pada Gambar 4.2, menampilkan grafik distribusi data jenis kelamin yang menunjukkan bahwa yang lebih banyak mengidap kanker kulit adalah laki-laki. Dan dari Gambar 4.3, dapat disimpulkan bahwa yang lebih banyak mengidap kanker kulit adalah orang dengan umur 30 tahun ke atas. Dan pada Gambar 4.4, dapat dilihat distribusi data lokasi tampaknya lesi kulit yang menunjukkan bahwa paling banyak adalah terkena di bagian punggung.

4.1.3 Dataset Hasil *Resampling*

Dari berbagai cara untuk menyeimbangkan distribusi data antar kelas pada suatu dataset, penulis memilih menggunakan teknik *resampling* dengan bantuan *library Python imbalanced-learn (imblearn)*. Teknik *resampling* yang digunakan antara lain, *random undersampler* dan *cluster centroids* yang termasuk dalam ke dalam metode *undersampling*, kemudian, *random oversampler* dan SMOTE yang termasuk ke dalam metode *oversampling*, serta gabungan SMOTE dengan *Tomek Link* dan gabungan SMOTE dengan ENN yang termasuk ke dalam metode *resampling gabungan (hybrid)*.

Dari tiap teknik akan menghasilkan perbedaan distribusi data sesuai dengan metodenya masing-masing dalam mengolah sampel yang baru seperti yang telah dijelaskan pada BAB 2. Karena perbedaan metode penanganan tersebut, maka akan mengakibatkan perbedaan kualitas data juga pada *dataset* yang digunakan.

Sebelum dilakukan *resampling*, penulis membagi dataset menjadi dua bagian, yaitu “Data” yang merepresentasikan karakteristik nilai *pixel* dari tiap citra dan “Label” yang merepresentasikan label atau jenis kanker kulit lesi berpigmen. Setelah itu, barulah dilakukan *resampling* menggunakan masing-masing teknik

resampling. Setelah itu, dilakukan *reshaping*, yaitu mengubah nilai *pixel array* citra masukan menjadi nilai 28x28x3 yang berarti berukuran 28x28 *pixel* dengan dimensi warna *Red Green, Blue* (RGB).

Tabel 4.1 di bawah ini menunjukkan distribusi data antar kelas pada dataset setelah dilakukan *resampling* dengan tiap metode.

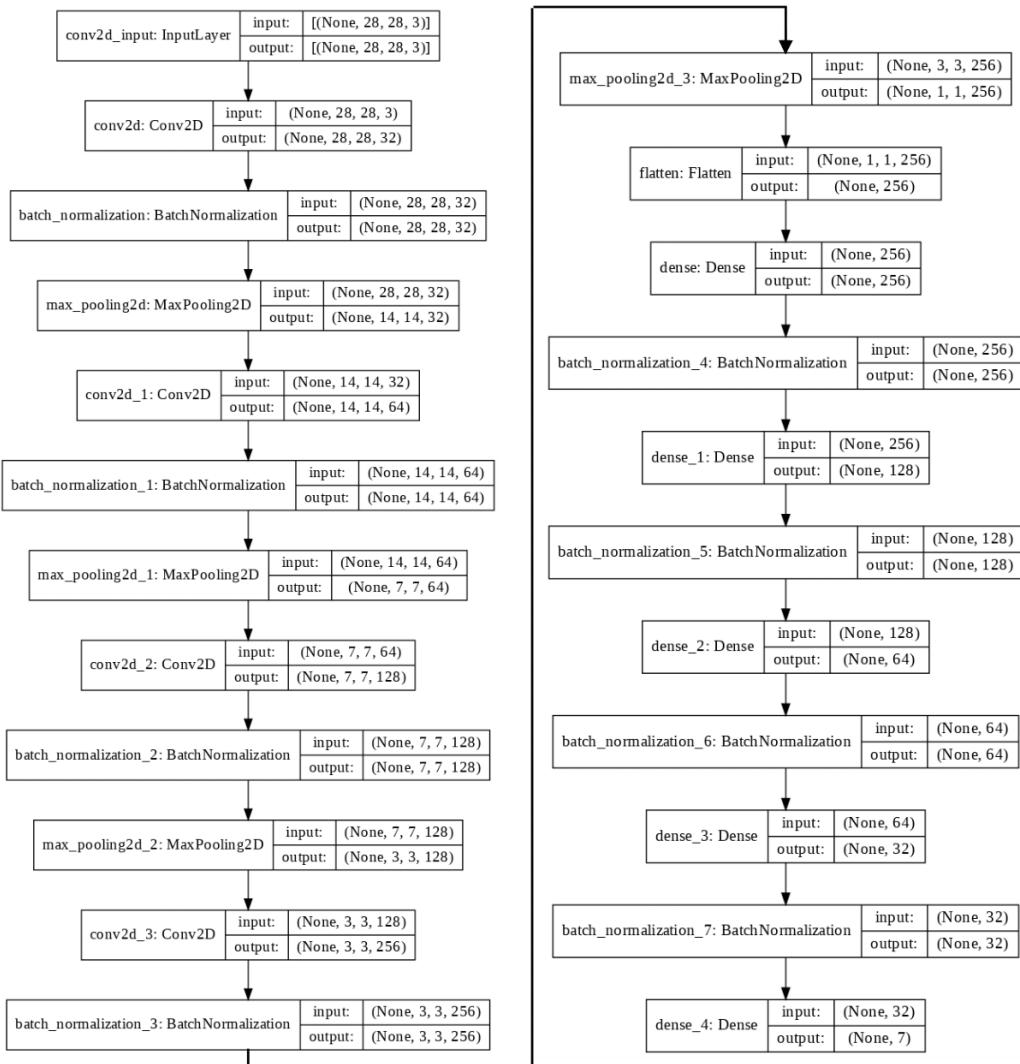
Tabel 4.1 Distribusi Data Tiap Kelas Setelah Dilakukan Resampling

Teknik Resampling	Total Data	Jumlah Data Tiap Kelas						
		akiec	bcc	blk	df	nv	vasc	mel
<i>Random Undersampler</i>	805	115	115	115	115	115	115	115
<i>cluster centroids</i>	805	115	115	115	115	115	115	115
<i>random oversampler</i>	46.935	6705	6705	6705	6705	6705	6705	6705
SMOTE	46.935	6705	6705	6705	6705	6705	6705	6705
ADASYN	47.187	6724	6641	6886	6691	6705	6730	6810
Gabungan SMOTE dengan <i>Tomek Link</i>	46.933	6705	6705	6705	6705	6704	6705	6704
Gabungan SMOTE dengan ENN	43.473	6705	6702	6597	6705	3464	6705	6595

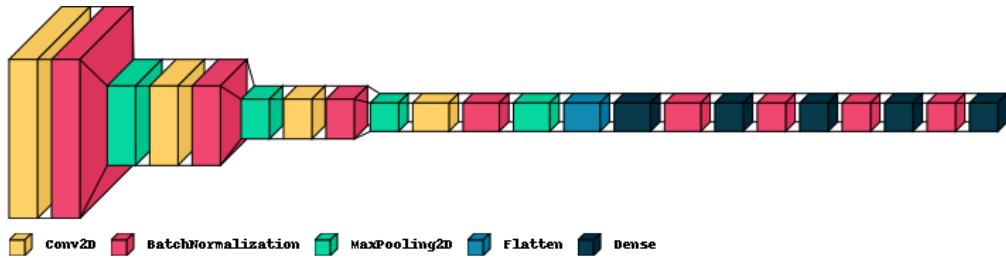
Setelah dilakukan *resampling*, kemudian *dataset* dibagi menjadi dua bagian menggunakan fungsi *train test split* dari library *sklearn model selection* dengan perbandingan 75% data latih (*train*) dan 25% data uji (*test*) yang akan digunakan dalam pelatihan model klasifikasi kanker kulit dengan arsitektur *Convolutional Neural Network*.

4.1.4 Pelatihan dan Pengujian Pada Google Colaboratory

Untuk melakukan klasifikasi dibutuhkan model jaringan saraf *deep learning* untuk dapat mengklasifikasikan objek. Pada pelatihan ini, model dibuat menggunakan arsitektur *Convolutional Neural Network* (CNN) tanpa menggunakan *transfer learning* (bantuan model yang sudah dilatih sebelumnya). Arsitektur model klasifikasi kanker kulit yang dibuat dapat dilihat pada Gambar 4.5 dan visualisasinya ditampilkan pada Gambar 4.6.



Gambar 4.5 Arsitektur Model CNN Klasifikasi Kanker Kulit



Gambar 4.6 Visualisasi Arsitektur Model CNN Klasifikasi Kanker Kulit

Arsitektur model ini sudah dirancang sebaik-baiknya dan melewati *trial and error* berkali-kali hingga didapatkan arsitektur model klasifikasi kanker kulit paling baik yang saat ini dapat penulis buat.

Pada arsitektur model digunakan *layer* yang sebelumnya belum dijelaskan di BAB 2. *Layer batch normalization* berfungsi untuk menormalisasikan aktivasi pada *layer* sebelumnya. *Batch normalization* ini digunakan untuk mengurangi pergeseran kovarian atau menyamakan distribusi setiap nilai input yang selalu berubah karena perubahan pada *layer* sebelumnya selama proses training. Menggunakan *batch normalization* memungkinkan akurasi menjadi jauh lebih tinggi, meningkatkan kecepatan pada saat pelatihan model CNN, dan mengurangi tingkat *overfitting* model.

Pada arsitektur model yang digunakan terdapat juga *dense layer*. *Dense layer* ini berfungsi untuk menambahkan *layer* yang *fully connected*, sebagaimana diketahui secara garis besar, CNN memiliki dua bagian besar, yaitu *feature extraction* dan *fully connected layer*. *Dense layer* dengan *activation softmax* berfungsi untuk melakukan klasifikasi sesuai dengan jumlah kelas keluaran sesuai yang diklasifikasi (dalam hal ini, *dense layer* membuat keluarannya 7 yang berarti ada 7 kelas kanker kulit yang diklasifikasikan [31].

Dataset citra pelatihan dan pengujian yang digunakan dilakukan *image augmentation* untuk meningkatkan kualitas data dan menghindari model *overfitting*. Pelatihan model dibuat dengan batas iterasi (*epoch*) yang ditentukan sebanyak 70 kali, dengan memanfaatkan *metode callbacks* dari *framework Keras*, antara lain *callbacks Model Checkpoint* digunakan untuk menyimpan model atau *weights* (dalam *checkpoint file*) dalam kualitas terbaik setiap interval tertentu yang ditentukan, kemudian *callbacks ReduceLROnPlateau* digunakan untuk mengurangi nilai koreksi bobot/tingkat belajar (*learning rate*) ketika metrik yang ditinjau saat pelatihan tidak mengalami perubahan yang baik, dan *callbacks Early Stopping* digunakan untuk menghentikan pelatihan sebelum semua iterasi (*epoch*) selesai jika metrik yang ditinjau tidak lagi mengalami perubahan yang baik.

4.2 Hasil Pelatihan dan Pengujian Pada *Google Colaboratory*

Proses pelatihan dan pengujian dilakukan pada *Google Colaboratory* menggunakan arsitektur CNN yang telah dibuat, namun yang membedakan adalah perbedaan komposisi *dataset* yang digunakan tiap pelatihan karena akan dilakukan uji banding perbedaan komposisi *dataset* terhadap kualitas model.

Jika kurva akurasi yang dihasilkan dari hasil pelatihan model sangatlah baik, presisinya baik, dan tidak banyak *noise* atau bias pada kurvanya, maka dapat dikatakan hasil pelatihan tersebut dalam kondisi *goodfit (low loss and high accuracy)*. Jika nilai akurasi rendah dan nilai validasi akurasi jauh lebih rendah lagi daripada nilai akurasi pelatihannya maka dikatakan model tersebut dalam kondisi *underfitting (high loss and low accuracy)*. Jika model yang dibuat terlalu fokus pada data pelatihan namun tidak bisa melakukan prediksi tepat jika diberikan data yang

baru (data pengujian), nilai akurasi pelatihan tinggi namun nilai validasi variasi bias pada kurva *loss* dan *accuracy* selama pembelajaran karena model menangkap data *noise* yang seharusnya diabaikan, maka dikatakan model tersebut dalam kondisi *overfitting (low loss and low accuracy)*.

Menggunakan akurasi saja dalam mengukur kinerja dari sebuah sistem/metode bisa menimbulkan bias yang sangat fatal. Sehingga, harus dilakukan evaluasi terhadap metrik lain seperti *recall*, *precision*, atau *F1-score*. Metrik evaluasi yang dipakai dalam penelitian ini untuk melihat apakah kualitas prediksi model baik atau tidak dan dilakukan perbandingan untuk tiap modelnya dijelaskan sebagai berikut [32].

1. Akurasi (*accuracy*)

Merupakan rasio prediksi benar (*true positif* (TP) dan *true negative* (TN) dengan keseluruhan data. Perhitungan secara matematis dapat dilihat pada persamaan 4.1 berikut.

$$\text{Akurasi} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN}) \quad (4.1)$$

2. Presisi (*precision*)

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Presisi dapat dikatakan juga sebagai tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem [33]. Perhitungan secara matematis dari *precision* dapat dilihat pada persamaan 4.2 berikut.

$$\text{Presisi} = \text{TP} / (\text{TP} + \text{FP}) \quad (4.2)$$

3. Sensitifitas (*recall*)

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. sebuah Sensitifitas dapat dikatakan juga sebagai tingkat keberhasilan sistem dalam menemukan kembali informasi. Perhitungan secara matematis dapat dilihat pada persamaan 4.3 berikut.

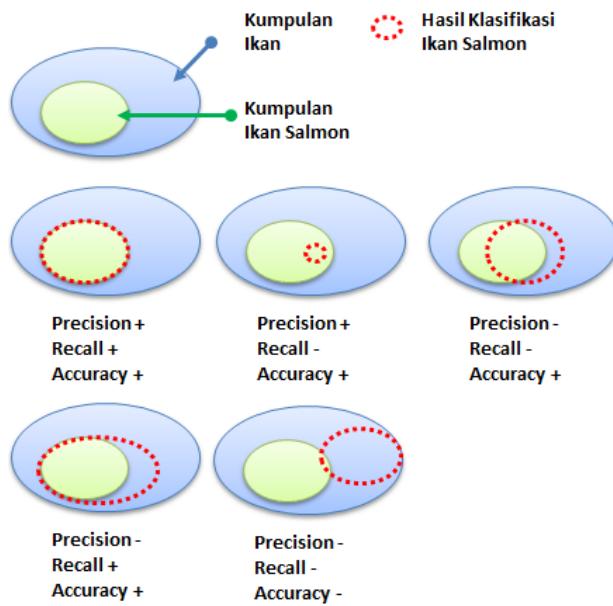
$$\text{Sensitifitas} = (\text{TP}) / (\text{TP} + \text{FN}) \quad (4.3)$$

4. *F1-Score*

Merupakan rata-rata harmonic dari *precision* dan *recall*. Akurasi tepat digunakan sebagai acuan performa model klasifikasi jika *dataset* yang digunakan memiliki jumlah data FN dan FP jumlahnya mendekati. Namun, jika jumlahnya tidak mendekati, sebaiknya menggunakan *F1 Score* sebagai acuan.

Perhitungan secara matematis dapat dilihat pada persamaan 4.4 berikut.

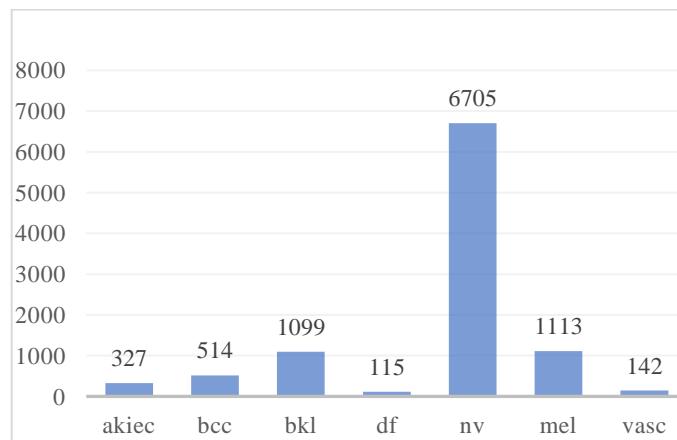
$$F1\text{-Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (4.4)$$



Gambar 4.7 Ilustrasi Metrik Evaluasi Terhadap Suatu Prediksi

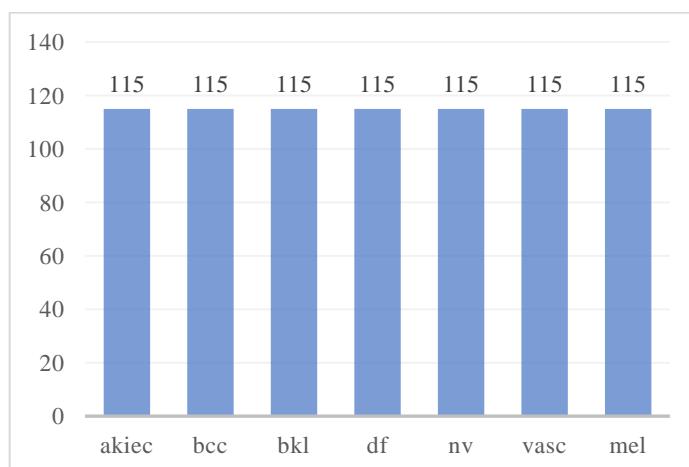
4.2.1 Model Menggunakan *Dataset Hasil Teknik Random Undersampler*

Gambar 4.8 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi data sangat tidak seimbang, di mana kelas nv memiliki data yang sangat mendominasi sebanyak 6705 citra lesi kulit, sedangkan kelas df hanya memiliki data sebanyak 115 citra.



Gambar 4.8 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Setelah dilakukan setelah dilakukan *resampling* menggunakan metode *random undersampler*, komposisi data menjadi lebih sebanding di mana *random undersampler* mengurangi jumlah data menjadi sebanyak jumlah kelas terendahnya, yaitu 115 citra lesi kulit yang ditunjukkan pada Gambar 4.9.



Gambar 4.9 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Random Undersampler

Berikut ini, ditampilkan kurva akurasi hasil pelatihan dan validasi akurasinya pada Gambar 4.10.



Gambar 4.10 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Undersampler

Pada Gambar 4.10 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik *random undersampler*. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* dihentikan saat mencapai *epoch* ke-47, sehingga diperoleh nilai terakhir akurasi pelatihan adalah 0.9161 dan akurasi validasi akurasinya adalah 0.8911.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Namun, validasi akurasi pelatihannya lebih rendah dan memiliki bentuk kurva kurang baik yang menandakan model dalam kondisi *underfitting*.



Gambar 4.11 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Undersampler

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.11, dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang kurang cukup baik dengan nilai akhir *loss* adalah 0.8296 dan nilai validasi *loss* pelatihan adalah 1.3286. Validasi *loss* pelatihannya memiliki nilai lebih tinggi daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *underfitting*.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik

dari pelatihan ketika mencapai model terbaiknya yang hasil unjuk kerjanya ditunjukkan pada Tabel 4.2.

Tabel 4.2 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Random Undersampler

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	1.2691818475723267
2.	<i>Validation Accuracy</i>	0.8910890817642212
3.	<i>True Positive</i>	80.0
4.	<i>False Positive</i>	32.0
5.	<i>True Negative</i>	1180.0
6.	<i>False Negative</i>	122.0
7.	<i>Precision</i>	0.7142857313156128
8.	<i>Recall</i>	0.39603960514068604

Berdasarkan Tabel 4.2 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai cukup besar dari total keseluruhannya. Dari model ini, didapat nilai *validation accuracy* sebesar 0.891, *precision* bernilai 0.714, dan *recall* yang bernilai 0.396 menandakan model ini tidak dapat memprediksi dengan baik dan benar.

Kemudian dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.12 di bawah ini.

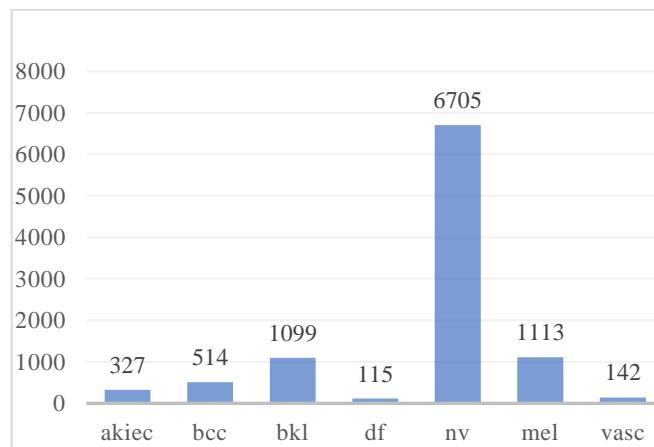
	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.50	0.39	0.44
('bcc', ' basal cell carcinoma')	0.67	0.25	0.36
('bkl', 'benign keratosis-like lesions')	0.70	0.24	0.36
('df', 'dermatofibroma')	0.80	0.13	0.22
('nv', ' melanocytic nevi')	1.00	0.41	0.59
('vasc', ' vascular lesions')	0.88	0.88	0.88
('mel', 'melanoma')	0.62	0.50	0.55
	micro avg	0.71	0.40
	macro avg	0.74	0.40
	weighted avg	0.73	0.40
	samples avg	0.40	0.40

Gambar 4.12 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Random Undersampler

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik dan hanya kelas *vascular lesions* (vasc) yang memiliki nilai rata-rata cukup baik dan itupun hanya memiliki nilai *precision* sebesar 0,88, nilai *recall* sebesar 0.85, dan nilai *F1-score* adalah 0.86 yang menunjukkan kualitas prediksi model ini belum terlalu baik. Dan nilai *F1-score* untuk kelas lainnya, kelas df memiliki nilai *precision* yang baik sebesar 1.00, tetapi memiliki nilai *recall* dan *F1-score* yang sangat buruk.

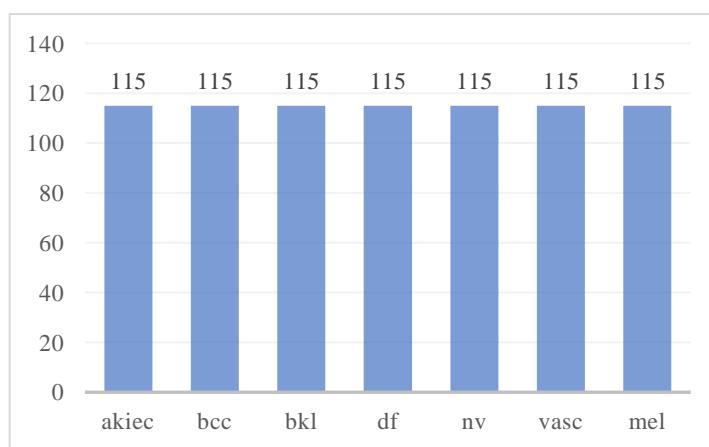
4.2.2 Model Menggunakan *Dataset Hasil Teknik Cluster Centroids*

Gambar 4.13 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi data sangat tidak seimbang, di mana kelas nv memiliki data yang sangat mendominasi sebanyak 6705 citra lesi kulit.



Gambar 4.13 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Setelah dilakukan *resampling* menggunakan metode *cluster centroids*, komposisi data menjadi lebih sebanding di mana *cluster centroids* mengurangi jumlah data menjadi sejumlah kelas terendahnya, yaitu 115 citra lesi kulit yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Cluster Centroids

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan validasinya pada Gambar 4.15.



Gambar 4.15 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Cluster Centroids

Pada Gambar 4.15 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik *cluster centroids*. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* dihentikan saat mencapai *epoch* ke-70, sehingga diperoleh nilai terakhir akurasi pelatihan adalah 0.9370 dan akurasi validasi akurasinya adalah 0.8918.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Namun, validasi akurasi pelatihannya lebih rendah dan

memiliki bentuk kurva kurang baik yang menandakan model dalam kondisi *underfitting*.



Gambar 4.16 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Cluster Centroids

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.16, dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang kurang cukup baik dengan nilai akhir *loss* adalah 0.6145 dan nilai validasi *loss* pelatihan adalah 1.0431. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih tinggi daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *underfitting*.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik dari pelatihan ketika mencapai model terbaiknya yang hasil unjuk kerjanya ditunjukkan pada Tabel 4.3.

Tabel 4.3 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Cluster Centroids

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	1.0431325435638428
2.	<i>Validation Accuracy</i>	0.891796350479126
3.	<i>True Positive</i>	103.0
4.	<i>False Positive</i>	54.0
5.	<i>True Negative</i>	1158.0
6.	<i>False Negative</i>	99.0
7.	<i>Precision</i>	0.656050980091095
8.	<i>Recall</i>	0.5099009871482849

Berdasarkan Tabel 4.3 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai cukup besar dari total keseluruhannya, *precision* yang bernilai 0.656 dan *recall* bernilai 0.509. Nilai yang didapatkan ini menunjukkan bahwa kualitas unjuk kerja model ini masih sangat buruk dan belum dapat memprediksi dengan baik dan benar.

Kemudian dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.17 di bawah ini.

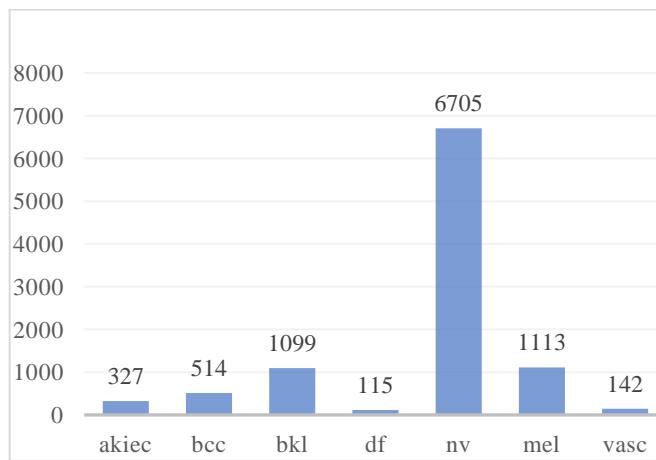
	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.55	0.55	0.55
('bcc', ' basal cell carcinoma')	0.33	0.21	0.26
('bkl', 'benign keratosis-like lesions')	0.52	0.41	0.46
('df', 'dermatofibroma')	0.62	0.58	0.60
('nv', ' melanocytic nevi')	0.94	0.59	0.72
('vasc', ' vascular lesions')	0.95	0.81	0.88
('mel', 'melanoma')	0.68	0.41	0.51
micro avg	0.66	0.51	0.57
macro avg	0.66	0.51	0.57
weighted avg	0.66	0.51	0.57
samples avg	0.51	0.51	0.51

Gambar 4.17 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Cluster Centroids

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Namun nilai ini sudah mengalami peningkatan nilai pada kelas lainnya. Kini, nilai klasifikasi dari *vascular lesions* (vasc) memiliki nilai *precision* sebesar 0.95, nilai *recall* sebesar 0.81, dan nilai *F1-score* adalah 0.88 dan kini untuk kelas *melanocytic nevi* (nv) mengalami peningkatan signifikan dengan nilai *precision* sebesar 0.94, nilai *recall* sebesar 0.59, dan nilai *F1-score* adalah 0.72. Hal ini menunjukkan teknik *resampling cluster centroids* masih belum memiliki kualitas unjuk kerja yang baik dalam membuat model klasifikasi 7 kelas kanker kulit berdasarkan lesi kulit, namun teknik ini mampu meningkatkan kualitas unjuk kerja pada klasifikasi kelas lainnya dibandingkan jika menggunakan metode *random undersampler*.

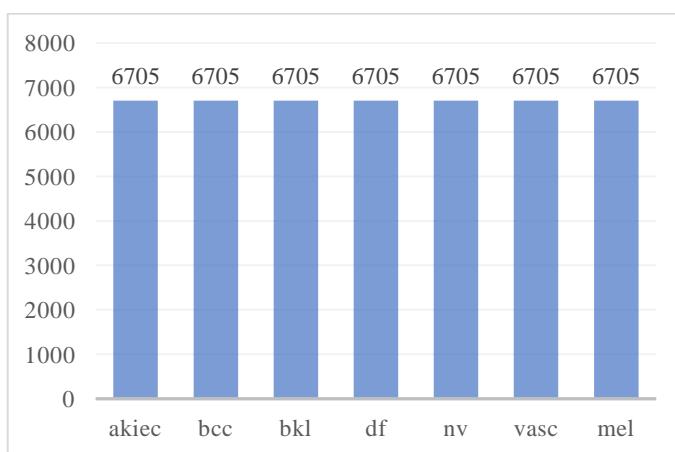
4.2.3 Model Menggunakan *Dataset Hasil Teknik Random Oversampler*

Gambar 4.18 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi dataset sangat tidak seimbang, di mana kelas nv memiliki data yang sangat mendominasi sebanyak 6705 citra lesi kulit, sementara kelas lainnya sangat jauh di bawahnya.



Gambar 4.18 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Setelah dilakukan *resampling* menggunakan metode *random oversampler*, komposisi data menjadi lebih sebanding di mana *random oversampler* memperbanyak jumlah data menjadi sebanyak jumlah kelas tertingginya, yaitu 6705 citra lesi kulit yang ditunjukkan pada Gambar 4.19.



Gambar 4.19 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik Random Oversampler

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan akurasi validasinya pada Gambar 4.20.



Gambar 4.20 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Oversampler

Pada Gambar 4.20 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik *Random Oversampler*. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) adalah sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* dihentikan saat mencapai *epoch* ke-50, sehingga diperoleh nilai terakhir akurasi pelatihan adalah 0.9786 dan akurasi validasinya adalah 0.9802.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Nilai akurasi yang didapat sangat tinggi dan nilai akurasi

pada data validasi lebih tinggi sedikit yang menunjukkan model ini dalam keadaan *goodfit* karena nilai akurasi validasinya sudah lebih tinggi dibanding akurasi pelatihannya.



Gambar 4.21 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling Random Oversampler

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.21 dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang baik dengan nilai akhir *loss* rendah, yaitu 0.2079 dan nilai validasi *loss* pelatihan adalah 0.1998. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih rendah daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *goodfit*.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik dari pelatihan ketika mencapai model terbaiknya yang hasil unjuk kerjanya ditunjukkan pada Tabel 4.4.

Tabel 4.4 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik Random Oversampler

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	0.1748005747795105
2.	<i>Validation Accuracy</i>	0.9827849864959717
3.	<i>True Positive</i>	10927.0
4.	<i>False Positive</i>	607.0
5.	<i>True Negative</i>	69797.0
6.	<i>False Negative</i>	807.0
7.	<i>Precision</i>	0.9473729729652405
8.	<i>Recall</i>	0.9312254786491394

Berdasarkan Tabel 4.4 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai sedikit dibanding total keseluruhannya, *precision* bernilai 0.947, dan *recall* bernilai 0.931. Nilai yang didapatkan ini sudah jauh lebih baik dibandingkan menggunakan metode *resampling dataset* sebelumnya. Dan hasil diprediksi kanker kulit yang dihasilkan juga cukup baik.

Kemudian dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.22 di bawah ini.

	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.98	1.00	0.99
('bcc', ' basal cell carcinoma')	0.97	0.98	0.98
('bkl', 'benign keratosis-like lesions')	0.91	0.88	0.89
('df', 'dermatofibroma')	1.00	1.00	1.00
('nv', ' melanocytic nevi')	0.92	0.79	0.85
('vasc', ' vascular lesions')	1.00	1.00	1.00
('mel', 'melanoma')	0.86	0.88	0.87
	micro avg	0.95	0.93
	macro avg	0.95	0.93
	weighted avg	0.95	0.93
	samples avg	0.93	0.93

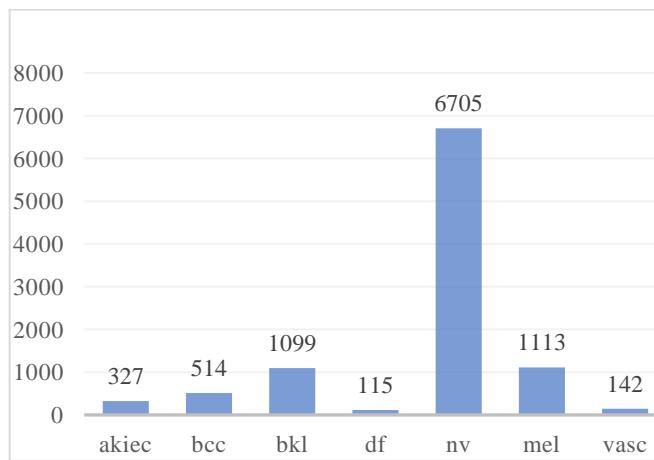
Gambar 4.22 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan Random Oversampler

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Namun, nilai ini sudah mengalami peningkatan dibandingkan penggunaan metode sebelumnya. Pada model ini, karena nilai *precision*, *recall*, dan *F1-score* memiliki nilai yang bisa dikatakan baik karena hampir semua nilainya sudah di atas 0.80, maka dipilih *F1-score* sebagai acuan perbandingan, didapatkan nilai *F1-score* untuk kelas akiec adalah 0.99, bcc adalah 0.98, bkl adalah 0.89, df adalah 1.00, nv adalah 0.85, vasc adalah 1.00, dan untuk mel adalah 0.87.

Dapat disimpulkan, bahwa model klasifikasi 7 kelas kanker kulit berdasarkan kondisi lesi kulit dengan menggunakan dataset yang telah dilakukan *resampling dataset* menggunakan teknik *oversampling* SMOTE menghasilkan kualitas klasifikasi tiap kanker kulit yang baik. Teknik *random oversampler* ini jauh lebih baik dibanding dengan model dengan teknik *undersampling* sebelumnya dalam melakukan klasifikasi 7 kelas kanker kulit.

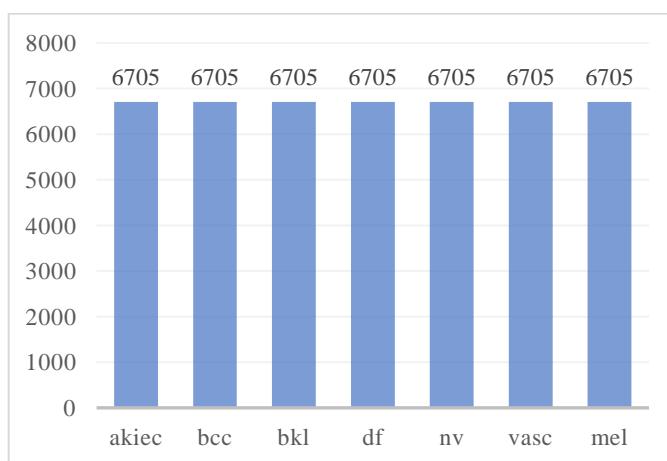
4.2.4 Model Menggunakan *Dataset Hasil Teknik SMOTE*

Gambar 4.23 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi dataset sangat tidak seimbang, di mana kelas nv memiliki data yang sangat mendominasi sebanyak 6705 citra lesi kulit, sementara kelas lainnya sangat jauh di bawahnya.



Gambar 4.23 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Setelah dilakukan *resampling* menggunakan metode SMOTE, komposisi data menjadi lebih sebanding di mana SMOTE memperbanyak jumlah data dengan membuat data sintetis menjadi sebanyak jumlah kelas tertingginya, yaitu 6705 citra lesi kulit yang ditunjukkan pada Gambar 4.24.



Gambar 4.24 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan akurasi validasinya pada Gambar 4.25.



Gambar 4.25 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE

Pada Gambar 4.25 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik SMOTE. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) adalah sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* berhenti saat mencapai *epoch* ke-66 dan diperoleh nilai terakhir akurasi pelatihan adalah 0.9766 dan akurasi validasinya adalah 0.9790.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Nilai akurasi pelatihan dan nilai akurasi pada data

validasi menunjukkan model ini dalam keadaan *goodfit*, *noise* yang ada pada kurva hanya sedikit, sehingga bisa dikatakan model ini memiliki kurva akurasi yang baik.



Gambar 4.26 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.26, dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang baik dengan nilai akhir *loss* yang rendah, yaitu 0.2321 dan nilai validasi *loss* pelatihan adalah 0.2087. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih rendah daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *goodfit* dan memiliki bentuk kurva yang baik.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik klasifikasi dari pelatihan ketika mencapai model terbaiknya yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	0.20775383710861206
2.	<i>Validation Accuracy</i>	0.9793394804000854
3.	<i>True Positive</i>	10784.0
4.	<i>False Positive</i>	747.0
5.	<i>True Negative</i>	69657.0
6.	<i>False Negative</i>	950.0
7.	<i>Precision</i>	0.935218095779419
8.	<i>Recall</i>	0.919038712978363

Berdasarkan Tabel 4.5 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai sangat sedikit dibanding total keseluruhannya, *precision* yang bernilai 0.935, dan *recall* bernilai 0.919. Hasil prediksi kanker kulit yang dihasilkan cukup baik. Nilai yang didapatkan ini sudah baik karena *validation accuracy*, *precision*, dan *recall* berada di atas nilai 0.9 yang berarti model dapat melakukan klasifikasi dengan cukup baik.

Kemudian, dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.27 di bawah ini.

	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.95	0.98	0.96
('bcc', ' basal cell carcinoma')	0.95	0.94	0.94
('bkl', 'benign keratosis-like lesions')	0.92	0.85	0.88
('df', 'dermatofibroma')	0.99	1.00	1.00
('nv', ' melanocytic nevi')	0.91	0.81	0.86
('vasc', ' vascular lesions')	1.00	1.00	1.00
('mel', 'melanoma')	0.84	0.86	0.85
micro avg	0.94	0.92	0.93
macro avg	0.94	0.92	0.93
weighted avg	0.93	0.92	0.93
samples avg	0.92	0.92	0.92

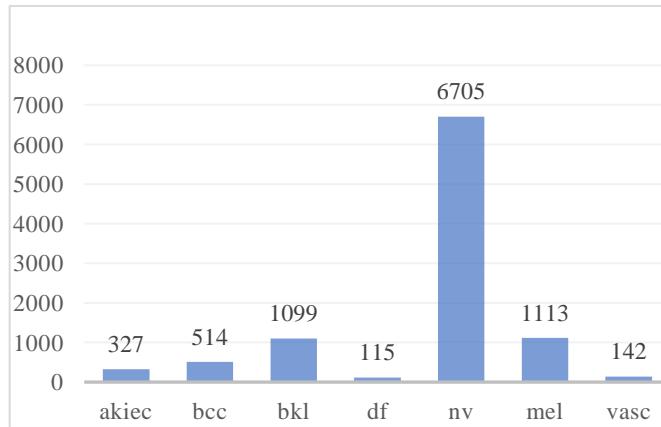
Gambar 4.27 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Nilai ini sudah mengalami peningkatan dibandingkan penggunaan metode sebelumnya. Dari model ini, digunakan *F1-score* sebagai acuan perbandingan, didapatkan nilai *F1-score* untuk kelas akiec adalah 0.96, bcc adalah 0.94, bkl adalah 0.88, df adalah 1.00, nv adalah 0.86, vasc adalah 1.00, dan untuk mel adalah 0.85.

Dapat disimpulkan, bahwa model klasifikasi 7 kelas kanker kulit berdasarkan kondisi lesi kulit dengan menggunakan *dataset* yang telah dilakukan *resampling dataset* menggunakan teknik *oversampling* SMOTE menghasilkan kualitas klasifikasi tiap kanker kulit yang cukup baik dan memiliki nilai yang lebih baik dari percobaan sebelumnya, karena pada model ini nilai *precision*, *recall*, dan *F1-score* dari model ini sudah berada di atas nilai 0.8.

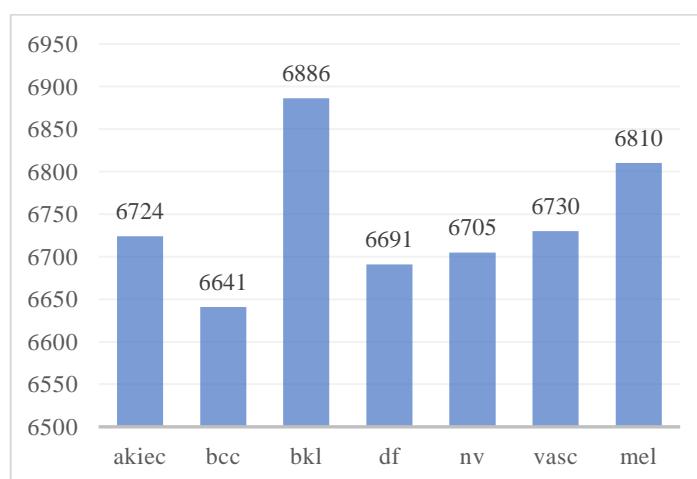
4.2.5 Model Menggunakan *Dataset Hasil Teknik ADASYN*

Gambar 4.28 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi data antar kelas sangat tidak seimbang.



Gambar 4.28 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Setelah dilakukan *resampling* menggunakan metode ADASYN, di mana ADASYN memperbanyak jumlah data dengan cara menghasilkan sampel data kelas minoritas secara adaptif sesuai dengan bobot distribusi tingkat kesulitan belajar menjadi komposisi data dengan jumlah yang bervariatif yang ditunjukkan pada Gambar 4.29.



Gambar 4.29 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik ADASYN

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan akurasi validasinya pada Gambar 4.30.



Gambar 4.30 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling ADASYN

Pada Gambar 4.30 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik ADASYN. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) adalah sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks ReduceLRonplateau* dengan melakukan *monitoring* terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* berhenti saat mencapai *epoch* ke-70 dan diperoleh nilai terakhir akurasi pelatihan adalah 0.9774 dan akurasi validasi pelatihan adalah 0.9802.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1 yang signifikan. Nilai akurasi pelatihan dan nilai akurasi

pada data validasi menunjukkan model ini dalam keadaan *goodfit*, walaupun masih terdapat *noise* pada kurva akurasinya, namun akurasi validasinya mengalami perubahan yang signifikan, sehingga memiliki bentuk kurva yang cukup baik.



Gambar 4.31 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling ADASYN

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.31, dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang baik dengan nilai akhir *loss* rendah, yaitu 0.2173 dan nilai validasi *loss* pelatihan adalah 0.2015. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih rendah daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *goodfit*.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik dari pelatihan ketika mencapai model terbaiknya yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik ADASYN

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	0.2015235424041748
2.	<i>Validation Accuracy</i>	0.9802249670028687
3.	<i>True Positive</i>	10889.0
4.	<i>False Positive</i>	725.0
5.	<i>True Negative</i>	70057.0
6.	<i>False Negative</i>	908.0
7.	<i>Precision</i>	0.9375753402709961
8.	<i>Recall</i>	0.9230312705039978

Berdasarkan Tabel 4.6 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai sangat sedikit dibanding total keseluruhannya, *validation accuracy* bernilai 0.98, *precision* yang bernilai 0.938, dan *recall* bernilai 0.923. Hasil prediksi kanker kulit yang dihasilkan menggunakan *dataset* yang telah dilakukan *resampling dataset* dengan metode ADASYN ini memiliki unjuk kerja yang baik.

Kemudian, dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.32 di bawah ini.

	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.95	0.99	0.97
('bcc', ' basal cell carcinoma')	0.96	0.96	0.96
('bkl', 'benign keratosis-like lesions')	0.90	0.85	0.88
('df', 'dermatofibroma')	1.00	1.00	1.00
('nv', ' melanocytic nevi')	0.92	0.81	0.86
('vasc', ' vascular lesions')	1.00	1.00	1.00
('mel', 'melanoma')	0.85	0.87	0.86
micro avg	0.94	0.92	0.93
macro avg	0.94	0.92	0.93
weighted avg	0.94	0.92	0.93
samples avg	0.92	0.92	0.92

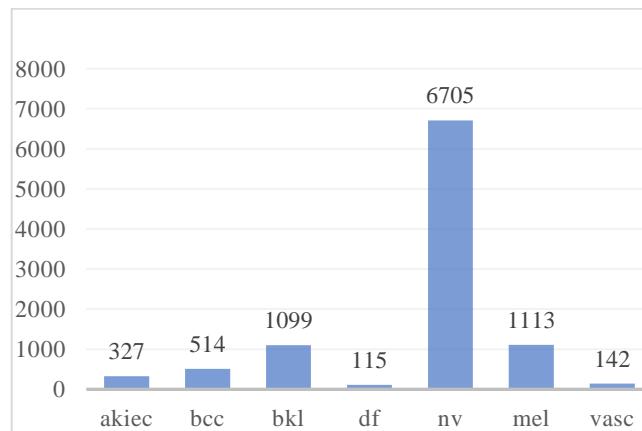
Gambar 4.32 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan ADASYN

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Nilai ini sudah mengalami peningkatan dibandingkan penggunaan metode sebelumnya. Dari model ini, digunakan *F1-score* sebagai acuan perbandingan, didapatkan nilai *F1-score* untuk kelas akiec adalah 0.97, bcc adalah 0.96, bkl adalah 0.88, df adalah 1.00, nv adalah 0.86, vasc adalah 1.00, dan untuk mel adalah 0.86.

Dapat disimpulkan, bahwa model klasifikasi 7 kelas kanker kulit berdasarkan kondisi lesi kulit dengan menggunakan *dataset* yang telah dilakukan *resampling dataset* menggunakan teknik *oversampling* ADASYN sudah menghasilkan kualitas klasifikasi tiap kelas kanker kulit yang baik.

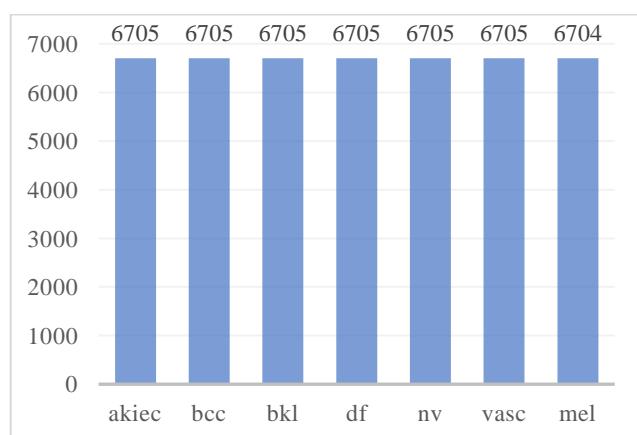
4.2.6 Model Menggunakan *Dataset* Hasil Teknik Gabungan SMOTE dan Tomek Link

Gambar 4.33 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset* HAM10000. Dapat dilihat bahwa distribusi data antar kelas sangat tidak seimbang.



Gambar 4.33 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Gambar 4.34 di bawah ini, menunjukkan komposisi distribusi *dataset* antar kelas kanker kulit setelah dilakukan *resampling* menggunakan metode gabungan teknik *oversampling* SMOTE dan *undersampling* TOMEK Link menghasilkan komposisi data dengan jumlah data untuk kelas akiec, bcc, bkl, df, nv, dan mel berjumlah 6705 citra dan untuk kelas vasc berjumlah 6704 citra lesi kulit.



Gambar 4.34 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE + Tomek Link

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan akurasi validasinya pada Gambar 4.35.



Gambar 4.35 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + Tomek Link

Pada Gambar 4.35 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik gabungan *oversampling* SMOTE dan *undersampling* Tomek Link. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) adalah sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks* *ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* berhenti saat mencapai *epoch* ke-63 dan diperoleh nilai terakhir akurasi pelatihan adalah 0.9777 dan akurasi validasi pelatihan adalah 0.9778.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Nilai akurasi pelatihan dan nilai akurasi pada data

validasi menunjukkan model ini dalam keadaan *goodfit*, namun memiliki bentuk kurva yang kurang baik dalam kurva akurasi validasinya pada iterasi awal pelatihan.



Gambar 4.36 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + Tomek Link

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.30, dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang baik dengan nilai akhir *loss* rendah, yaitu 0.2132 dan nilai validasi *loss* pelatihan adalah 0.2214. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih rendah daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *goodfit*, namun memiliki kurva validasi yang kurang baik pada iterasi awal pelatihan.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik dari pelatihan ketika mencapai model terbaiknya yang ditunjukkan pada Tabel 4.7.

Tabel 4.7 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE + Tomek

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	0.20816180109977722
2.	<i>Validation Accuracy</i>	0.9790721535682678
3.	<i>True Positive</i>	10772.0
4.	<i>False Positive</i>	757.0
5.	<i>True Negative</i>	69647.0
6.	<i>False Negative</i>	962.0
7.	<i>Precision</i>	0.9343394637107849
8.	<i>Recall</i>	0.9180160164833069

Berdasarkan Tabel 4.7 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai sangat sedikit dibanding total keseluruhannya, *validation accuracy* bernilai 0.979, *precision* yang bernilai 0.934, dan *recall* bernilai 0.918. Hasil prediksi kanker kulit yang dihasilkan dengan metode ini sudah menghasilkan unjuk kerja yang baik.

Kemudian, dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.37 di bawah ini.

	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.95	0.98	0.96
('bcc', ' basal cell carcinoma')	0.97	0.96	0.96
('bkl', 'benign keratosis-like lesions')	0.90	0.84	0.87
('df', 'dermatofibroma')	1.00	1.00	1.00
('nv', ' melanocytic nevi')	0.90	0.81	0.85
('vasc', ' vascular lesions')	1.00	1.00	1.00
('mel', 'melanoma')	0.83	0.84	0.84
micro avg	0.93	0.92	0.93
macro avg	0.93	0.92	0.93
weighted avg	0.93	0.92	0.93
samples avg	0.92	0.92	0.92

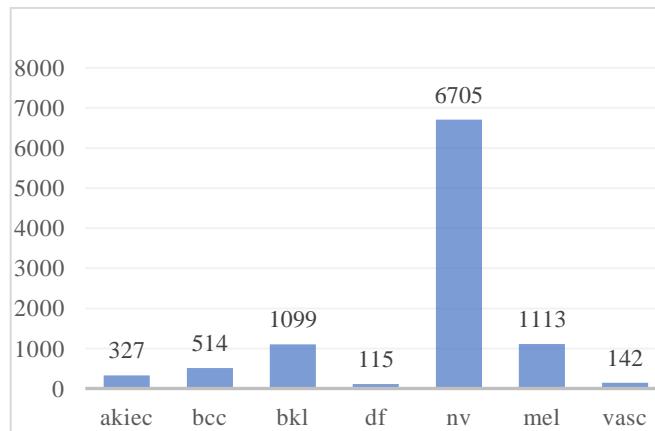
Gambar 4.37 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE + Tomek Link

Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Namun, nilai ini sudah mengalami peningkatan dibandingkan penggunaan metode sebelumnya. Dari model ini, digunakan *F1-score* sebagai acuan perbandingan, mendapatkan nilai *F1-score* untuk kelas akiec adalah 0.96, bcc adalah 0.96, bkl adalah 0.87, df adalah 1.00, nv adalah 0.85, vasc adalah 1.00, dan untuk mel adalah 0.84.

Dapat disimpulkan, bahwa model klasifikasi 7 kelas kanker kulit berdasarkan kondisi lesi kulit dengan menggunakan *dataset* yang telah dilakukan *resampling dataset* menggunakan teknik *oversampling* SMOTE dan *undersampling* Tomek Link menghasilkan kualitas klasifikasi tiap kelas kanker kulit yang baik. Namun, jika dibandingkan nilai *F1-score* dengan metode ADASYN sebelumnya, metode ini memiliki penurunan kualitas unjuk kerja dalam mengklasifikasikan tiap kelas kanker kulit.

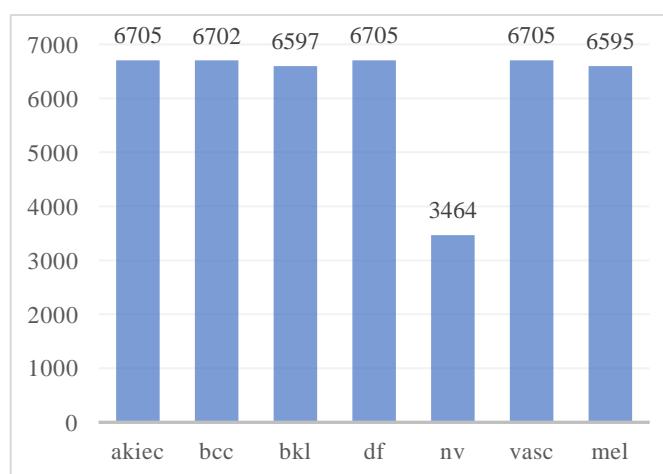
4.2.7 Model Menggunakan *Dataset Hasil Teknik Gabungan SMOTE dan ENN*

Gambar 4.38 di bawah ini menampilkan komposisi distribusi data 7 kelas kanker kulit pada *dataset HAM10000*. Dapat dilihat bahwa distribusi data antar kelas sangat tidak seimbang.



Gambar 4.38 Grafik Distribusi Dataset Sebelum Dilakukan Resampling

Gambar 4.39 di bawah ini, menunjukkan komposisi distribusi *dataset* antar kelas kanker kulit setelah dilakukan *resampling* menggunakan metode gabungan teknik *oversampling SMOTE* dan *undersampling Edited Nearest Neighbour (ENN)* menghasilkan komposisi data dengan jumlah data seperti yang ditunjukkan pada grafik pada Gambar 4.39 di bawah ini.



Gambar 4.39 Grafik Distribusi Dataset Setelah Dilakukan Resampling dengan Teknik SMOTE + ENN

Berikut ini ditampilkan kurva akurasi hasil pelatihan dan akurasi validasinya pada Gambar 4.40.



Gambar 4.40 Kurva Accuracy Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + ENN

Pada Gambar 4.40 di atas, memperlihatkan kondisi kurva akurasi pelatihan dari model yang dibuat menggunakan *dataset* yang telah dilakukan *resampling* menggunakan teknik gabungan *oversampling* SMOTE dan *undersampling* ENN. Pada saat pelatihan, pelatihan dibuat dengan batas maksimal iterasi (*epoch*) adalah sebanyak 70 kali. Dikarenakan saat pelatihan digunakan metode *callbacks* *ReduceLRonplateau* dengan melakukan monitor terhadap metrik *val_recall* dan *callbacks Early Stopping* dengan melakukan monitor terhadap metrik *val_loss*. Ketika dilakukan pelatihan, *epochs* berhenti saat mencapai *epoch* ke-66 dan diperoleh nilai terakhir akurasi pelatihan adalah 0.9848 dan akurasi validasi pelatihan adalah 0.9869.

Dapat dilihat pada kurva akurasi, nilai akurasi dari pelatihan mengalami peningkatan mendekati 1. Nilai akurasi pelatihan dan nilai akurasi pada data validasi menunjukkan model ini dalam keadaan *goodfit* dan bentuk kurva akurasinya pun sangat baik karena tidak terlalu banyak *noise* atau bias pada kurva akurasi validasinya, noise hanya terjadi pada iterasi awal pelatihan dan bentuknya membaik secara signifikan.



Gambar 4.41 Kurva Loss Pada Pelatihan dan Validasi Dengan Dataset Menggunakan Teknik Resampling SMOTE + ENN

Dari kurva kesalahan (*loss*) pelatihan pada Gambar 4.41 dapat dilihat bahwa kurva *loss* tersebut memiliki bentuk kurva yang baik dengan nilai akhir *loss* rendah, yaitu 0.1483 dan nilai validasi *loss* pelatihan adalah 0.1412. Kesalahan (*loss*) validasi pelatihannya memiliki nilai lebih rendah daripada nilai kurva kesalahannya yang menandakan model berada dalam keadaan *goodfit*.

Dengan *callbacks Model Checkpoint*, kita dapat menyimpan model dan/atau bobot hasil penelitiannya dalam keadaan terbaiknya. Dan ketika dilakukan

evaluasi dengan metode *model evaluate* dari *Keras* dapat terlihat nilai tiap metrik dari pelatihan ketika mencapai model terbaiknya yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Metrik Evaluasi Model Terbaik Menggunakan Dataset dengan Teknik SMOTE + ENN

No.	Metrik Evaluasi	Nilai
1.	<i>Validation loss</i>	0.13576099276542664
2.	<i>Validation Accuracy</i>	0.9868435859680176
3.	<i>True Positive</i>	10317.0
4.	<i>False Positive</i>	449.0
5.	<i>True Negative</i>	64765.0
6.	<i>False Negative</i>	552.0
7.	<i>Precision</i>	0.9582946300506592
8.	<i>Recall</i>	0.9492133855819702

Berdasarkan Tabel 4.8 di atas, menunjukkan bahwa hasil prediksi yang salah atau yang biasa disebut *false positive* (FP) maupun *false negative* (FN) dari model terbaik yang didapatkan pada pelatihan kali ini bernilai sangat sedikit dibanding total keseluruhannya, *validation accuracy* bernilai 0.987, *precision* bernilai 0.958, dan *recall* bernilai 0.949. Hasil prediksi kanker kulit yang dihasilkan sudah baik.

Kemudian, dengan menggunakan metode *classification report* dari *library scikit learn* kita dapat melihat nilai *precision*, *recall*, dan *F1-score* tiap kelas kanker dari hasil prediksi terhadap data uji menggunakan model klasifikasi kanker kulit yang dibuat. Untuk hasil dari klasifikasi terhadap prediksi terhadap data uji ditunjukkan pada Gambar 4.35 di bawah ini.

	precision	recall	f1-score
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	0.95	0.97	0.96
('bcc', ' basal cell carcinoma')	0.98	0.95	0.96
('bkl', 'benign keratosis-like lesions')	0.93	0.87	0.90
('df', 'dermatofibroma')	0.99	1.00	1.00
('nv', ' melanocytic nevi')	0.96	0.93	0.95
('vasc', ' vascular lesions')	1.00	1.00	1.00
('mel', 'melanoma')	0.91	0.92	0.91
micro avg	0.96	0.95	0.95
macro avg	0.96	0.95	0.95
weighted avg	0.96	0.95	0.95
samples avg	0.95	0.95	0.95

Gambar 4.42 Classification Report Dari Model Menggunakan Dataset Hasil Resampling Dataset Menggunakan SMOTE + ENN

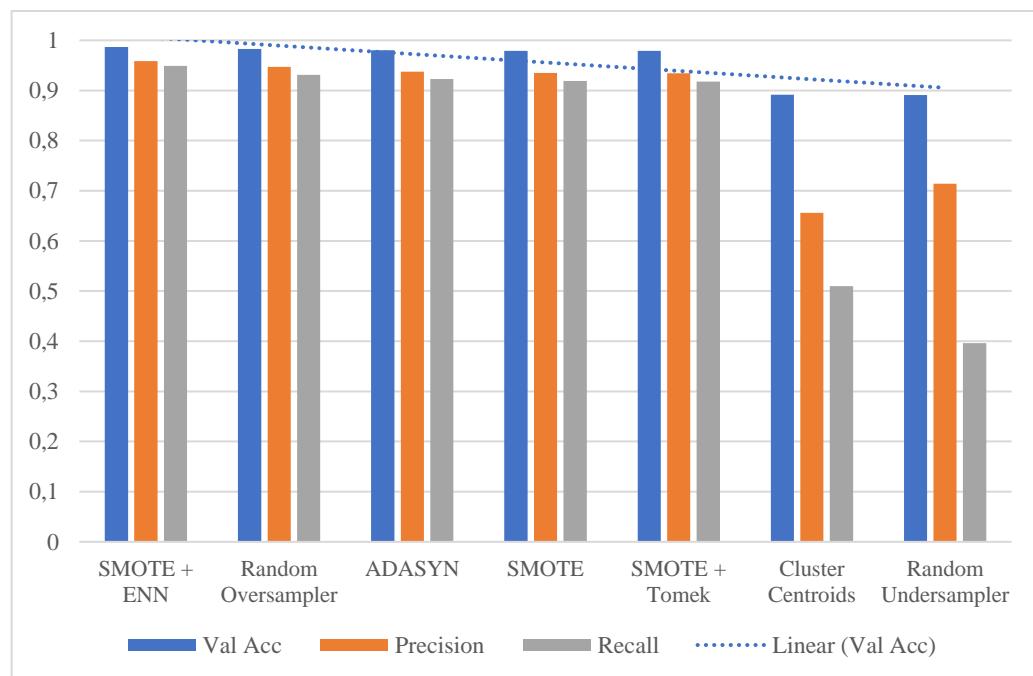
Dari *classification report* di atas, dapat dilihat bahwa keseluruhan tiap kelas kanker kulit belum memiliki nilai *precision*, *recall*, dan *F1-score* yang baik. Namun, nilai ini sudah mengalami peningkatan dibandingkan penggunaan metode sebelumnya. Dari model ini, digunakan *F1-score* sebagai acuan perbandingan, mendapatkan nilai *F1-score* untuk kelas akiec adalah 0.96, bcc adalah 0.96, bkl adalah 0.90, df adalah 1.00, nv adalah 0.95, vasc adalah 1.00, dan untuk mel adalah 0.91. Dari nilai evaluasi model dengan teknik *resampling* gabungan metode SMOTE dan ENN menghasilkan model klasifikasi dengan kualitas *precision*, *recall*, dan *F1-score* yang sangat baik dan dengan bentuk kurva akurasi dan kesalahan yang sangat baik juga.

4.3 Perbandingan Unjuk Kerja Model Dengan Teknik *Resampling Dataset*

Pada bagian ini akan dilakukan uji banding unjuk kerja model klasifikasi kanker kulit menggunakan *dataset* HAM10000 yang dilakukan *resampling dataset* dengan berbagai teknik *resampling dataset*. Tabel 4.9 berikut ini akan menampilkan perbandingan unjuk kerja dari model evaluasi pada kondisi terbaik model yang didapat dengan menggunakan metode *callbacks Model Checkpoint*.

Tabel 4.9 Perbandingan Metrik Klasifikasi Model Terbaik Tiap Teknik Resampling

No.	Teknik Resampling	Val Acc	Precision	Recall
1.	Gabungan SMOTE dan ENN	0.986843	0.958295	0.949213
2.	Random Oversampler	0.982785	0.947373	0.931225
3.	ADASYN	0.980225	0.937575	0.923031
4.	SMOTE	0.979339	0.935218	0.919039
5.	Gabungan SMOTE dan Tomek Link	0.979072	0.934339	0.918016
6.	Cluster Centroids	0.891796	0.656051	0.509901
7.	Random Undersampler	0.891089	0.71428	0.39604



Gambar 4.43 Kurva Metrik Evaluasi Model Terbaik Tiap Teknik Resampling

Dari Tabel 4.9 yang berisikan data metrik klasifikasi model terbaik ketika pelatihan model yang sudah diurutkan berdasarkan nilai tertinggi pada metrik Val Acc (*Validation Accucary*), dapat dilihat bahwa nilai *validation accuracy*, *precision*, dan *recall* untuk model menggunakan *dataset* yang dilakukan *resampling*

menggunakan teknik gabungan *oversampling* SMOTE dengan *undersampling* ENN memiliki nilai yang paling tinggi untuk tiap metrik yang dibandingkan. Dari grafik pada Gambar 4.36 pun dapat dilihat bentuk grafik dan *trendline* dari grafik tersebut mengalami arah dengan nilai tertinggi (puncak) berada pada model menggunakan gabungan SMOTE dengan ENN.

Untuk mengetahui kualitas prediksi klasifikasi kanker kulit tiap kelasnya, Tabel 4.10 di bawah ini menampilkan perbandingan nilai *F1-score* dari hasil prediksi tiap kelas kanker kulit menggunakan metode *model.predict* yang ditampilkan menggunakan metode *classification report* dari *scikit-learn library*.

Tabel 4.10 Perbandingan Nilai Prediksi Tiap Kelas Menggunakan Berbagai Teknik Resampling

Teknik Resampling	<i>F1-score</i>						
	akiec	bcc	bkl	df	nv	vasc	mel
<i>Random Undersampler</i>	0.44	0.36	0.36	0.22	0.59	0.88	0.55
<i>Cluster Centroids</i>	0.55	0.26	0.46	0.60	0.72	0.88	0.51
<i>Random Oversampler</i>	0.99	0.98	0.89	1.00	0.85	1.00	0.87
SMOTE	0.96	0.94	0.88	1.00	0.86	1.00	0.85
ADASYN	0.97	0.96	0.88	1.00	0.86	1.00	0.86
Gabungan SMOTE + Tomek Link	0.96	0.96	0.87	1.00	0.85	1.00	0.84
Gabungan SMOTE + ENN	0.96	0.96	0.90	1.00	0.95	1.00	0.91

Dari Tabel 4.10, kita bisa melihat perbandingan nilai *F1-score* dari klasifikasi tiap kelas kanker kulit dengan menggunakan berbagai teknik resampling dataset yang diamati pada penelitian ini. Dari tabel diberikan indikator warna kuning untuk memudahkan pembacaan tabel dalam melihat nilai *F1-score* tertinggi untuk tiap kelas kanker kulit. Dapat dilihat bahwa teknik *resampling dataset*

menggunakan teknik gabungan SMOTE dengan ENN memiliki nilai *F1-score* tertinggi yang paling dari keseluruhan klasifikasi 7 kelas kanker kulit. Dari 7 kelas kanker kulit yang diklasifikasikan, teknik ini memiliki nilai *F1-score* tertinggi untuk 5 kelas, yaitu kelas bkl, df, nv, vasc, dan mel.

Berdasarkan dua tabel perbandingan unjuk kerja yang ditampilkan oleh Tabel 4.9 dan Tabel 4.10, dapat disimpulkan bahwa unjuk kerja dari *resampling* pada *dataset* HAM10000 terhadap performa model klasifikasi kanker kulit berdasarkan kondisi lesi kulit yang memiliki hasil klasifikasi paling baik adalah teknik *resampling dataset* menggunakan gabungan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dengan *undersampling Edited Nearest Neighbour* (ENN).

4.4 Website Antarmuka Pengguna Sistem Klasifikasi Kanker Kulit

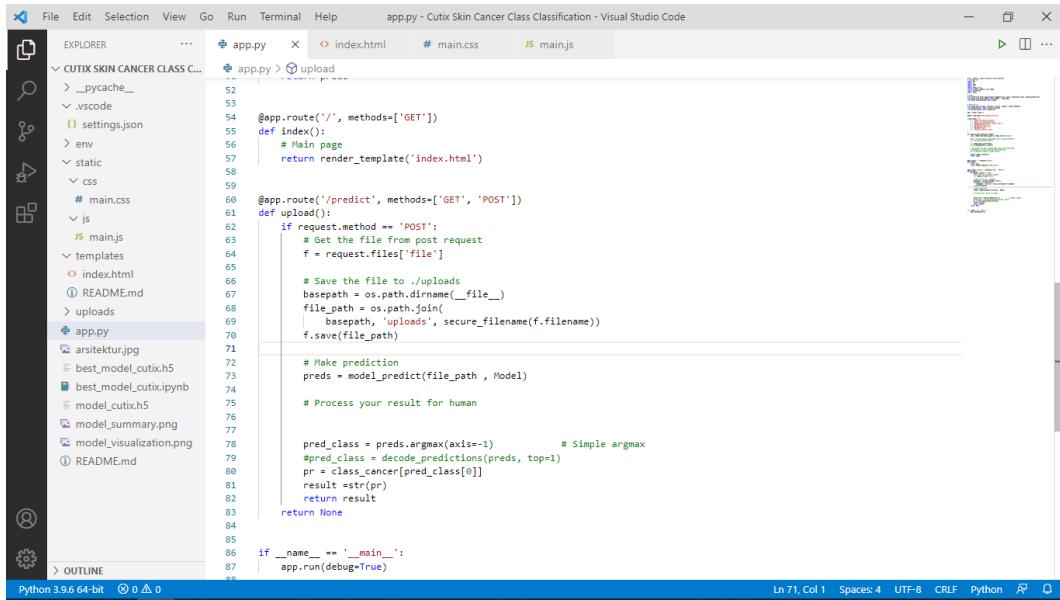
Setelah dilakukan uji banding unjuk kerja model klasifikasi kanker kulit dengan menggunakan *dataset* HAM10000 yang telah dilakukan resampling menggunakan beberapa teknik *resampling dataset*. Akhirnya, penulis memilih model *deep learning* untuk mengklasifikasikan kanker kulit berdasarkan kondisi lesi kulit berpigmen yang digunakan dalam *in-browser app* adalah model yang dilakukan *resampling dataset* menggunakan gabungan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dengan *undersampling Edited Nearest Neighbour* (ENN). Model ini dipilih karena dinilai memiliki unjuk kerja yang paling baik dalam mengklasifikasikan tiap kelas kanker kulit pada penelitian ini.

Perancangan program *website* dilakukan setelah program pada *Google Colaboratory* selesai dibuat dan didapatkan hasil model klasifikasi kanker kulit

yang disimpan dalam format “.h5”. Setelah mendapatkan file model, kemudian penulis membuat *website* yang akan digunakan sebagai antarmuka pengguna dengan sistem klasifikasi kanker kulit. Mekanismenya adalah pengguna mengunggah foto dari kondisi lesi kulit yang ingin dideteksi ke dalam *website*, kemudian sistem akan memprediksi foto tersebut ke dalam kelas kanker kulit yang sesuai.

Program *website* dibuat menggunakan platform *Visual Studio Code*. Program tampilan *website* dibuat menggunakan bahasa pemrograman HTML untuk membuat struktur halaman *website*, bahasa pemrograman CSS yang digunakan untuk mengatur tampilan elemen *website*, dan bahasa pemrograman *javascript* yang digunakan pada *website* agar *website* tersebut menjadi lebih interaktif dan dinamis, serta untuk mengambil data hasil klasifikasi kanker kulit untuk ditampilkan ke halaman *website*.

Untuk menghubungkan file model berformat “.h5” dengan antarmuka *website*, dibuat program menggunakan bahasa pemrograman *python* yang digunakan untuk pengembangan aplikasi *website* dengan menarik data foto dari *website* kemudian terhubung dengan file model berformat “.h5” yang kemudian diklasifikasikan. Setelah itu, program *javascript* akan menarik data tersebut dan setelah itu ditampilkan ke halaman *website*. Untuk memberikan kemudahan dalam membuat program *website* ini, semua file yang dibutuhkan dalam program ini disimpan di dalam satu folder yang sama. Untuk tampilan pembuatan *website* di *Visual Studio Code* ditampilkan pada Gambar 4.44 dan *interface website* ditunjukkan pada Gambar 4.45.



```

File Edit Selection View Go Run Terminal Help app.py - Cutix Skin Cancer Class Classification - Visual Studio Code
EXPLORER ... app.py > upload
CUTIX SKIN CANCER CLASS C...
> __pycache__ 52
> .vscode 53
settings.json 54 @app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

    # Make prediction
    preds = model_predict(file_path, Model)

    # Process your result for human

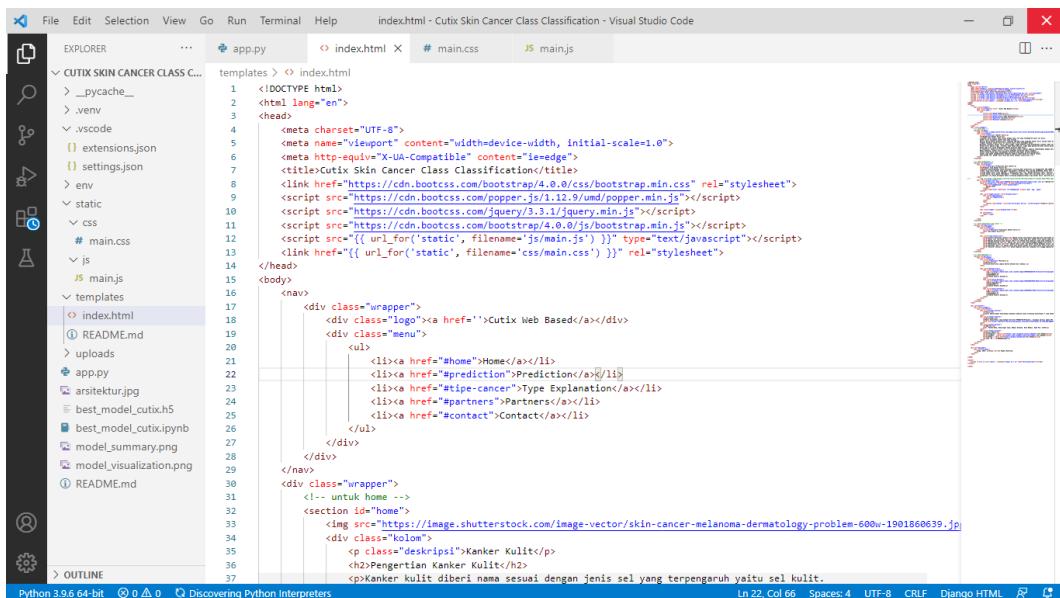
    pred_class = preds.argmax(axis=-1)           # Simple argmax
    pred_class = decode_predictions(preds, top=1)
    pr = class_cancer[pred_class[0]]
    result = str(pr)
    return result
return None

if __name__ == '__main__':
    app.run(debug=True)

```

Python 3.9.6 64-bit ⚡ 0 ▲ 0 OUTLINE Ln 71, Col 1 Spaces: 4 UTF-8 CRLF Python ⚡

Gambar 4.44 Tampilan Program app.py di Visual Studio Code



```

File Edit Selection View Go Run Terminal Help index.html - Cutix Skin Cancer Class Classification - Visual Studio Code
EXPLORER ... index.html # main.css JS main.js
CUTIX SKIN CANCER CLASS C...
templates > index.html
> __pycache__ 1 <!DOCTYPE html>
> .env 2 <html lang="en">
> .vscode 3 <head>
settings.json 4     <meta charset="UTF-8">
extensions.json 5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
.JS settings.json 6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
.env 7     <title>Cutix Skin Cancer Class Classification</title>
> static 8     <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
> static # main.css 9     <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
> static JS main.js 10    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
templates > index.html 11    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
12    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
13
14 </head>
15 <body>
16     <nav>
17         <div class="wrapper">
18             <div class="logo"><a href='>Cutix Web Based</a></div>
19             <div class="menu">
20                 <ul>
21                     <li><a href="#">Home</a></li>
22                     <li><a href="#">Prediction</a></li>
23                     <li><a href="#">Type Explanation</a></li>
24                     <li><a href="#">Partners</a></li>
25                     <li><a href="#">Contact</a></li>
26                 </ul>
27             </div>
28         </div>
29     </nav>
30     <div class="wrapper">
31         <!-- untuk home -->
32         <section id="home">
33             
34             <div class="kolom">
35                 <p class="deskripsi">Kanker Kulit</p>
36                 <h2>Pengertian Kanker Kulit</h2>
37                 <p>Kanker kulit diberi nama sesuai dengan jenis sel yang terpengaruh yaitu sel kulit.</p>

```

Python 3.9.6 64-bit ⚡ 0 ▲ 0 Discovering Python Interpreters OUTLINE Ln 22, Col 66 Spaces: 4 UTF-8 CRLF Django HTML ⚡

Gambar 4.45 Tampilan Program HTML di Visual Studio Code

Gambar 4.46 Tampilan Website Sebagai Antarmuka Pengguna Sistem Klasifikasi Kanker Kulit

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka diperoleh kesimpulannya sebagai berikut.

1. *Convolutional Neural Network* (CNN) yang merupakan algoritma dari *deep learning* yang cocok digunakan untuk melakukan klasifikasi objek dengan cara melakukan ekstraksi karakteristik (*feature*) data masukan menjadi data baru yang merepresentasikan data masukan.
2. Meninjau nilai akurasi dalam mengukur performa model klasifikasi terkadang dapat menimbulkan bias yang fatal, maka dari itu penting dilakukan evaluasi terhadap metrik klasifikasi lain, seperti *recall*, *precision*, atau *F1-score*.
3. Satu jenis teknik *resampling* belum tentu cocok untuk setiap *dataset* yang digunakan.
4. Ditinjau dari metrik *validation accuracy* dari hasil pelatihan yang dilakukan, teknik *resampling dataset* yang menghasilkan unjuk kerja paling baik adalah teknik Gabungan SMOTE dengan ENN dengan nilai 0.986843, diikuti teknik *random oversampler* dengan nilai 0.982785, dan ADASYN dengan nilai 0.980225.
5. Ditinjau dari metrik *precision* dari hasil pelatihan yang dilakukan, teknik *resampling dataset* yang menghasilkan unjuk kerja paling baik adalah teknik Gabungan SMOTE dengan ENN dengan nilai 0.958295, diikuti teknik *random oversampler* dengan nilai 0.947373, dan ADASYN dengan nilai 0.937575.

6. Ditinjau dari metrik *recall* dari hasil pelatihan yang dilakukan, teknik *resampling dataset* yang menghasilkan unjuk kerja paling baik adalah teknik Gabungan SMOTE dengan ENN dengan nilai 0.949213, diikuti teknik *random oversampler* dengan nilai 0.931225, dan ADASYN dengan nilai 0.923031.
7. Dalam uji banding unjuk kerja secara keseluruhan, pengaruh teknik *resampling dataset* HAM10000 terhadap performa model klasifikasi kanker kulit, teknik *resampling* gabungan *oversampling* SMOTE dengan *undersampling* ENN menghasilkan unjuk kerja yang paling baik dibandingkan teknik *resampling* lainnya yang digunakan dalam penelitian ini.

5.2 Saran

Berdasarkan penelitian yang telah dikerjakan penulis, penulis memberikan beberapa saran guna pengembangan penelitian selanjutnya.

1. Mengembangkan implementasi sistem klasifikasi kanker kulit berdasarkan lesi berpigmen pada aplikasi berbasis *smartphone* atau *standalone machine*.
2. Menggunakan *dataset* kanker kulit lain yang lebih terbaru atau melakukan kombinasi dengan *dataset* kanker kulit lain.
3. Melakukan *resampling dataset* menggunakan algoritma lainnya, seperti K-*Means*, SVM, atau lainnya.
4. Melakukan pengembangan arsitektur model CNN, antara lain mengganti teknik *pooling* atau *activation* pada *convolutional layer* atau dapat juga dilakukan percobaan menggunakan metode *transfer learning*.
5. Melakukan *pre-processing dataset* dan *image augmentation* lain sebelum dilakukan pelatihan model.

DAFTAR PUSTAKA

- [1] R. Halodoc, “Kenali 9 Gejala Kanker Kulit yang Jarang Disadari,” halodoc, 7 Mei 2021. [Online]. Available: <https://www.halodoc.com/artikel/kenali-9-gejala-kanker-kulit-yang-jarang-disadari>. [Diakses 5 Agustus 2021].
- [2] J. Ferlay, M. Colombet, . I. Soerjomataram, C. Mathers, D. M. Parkin, M. Piñeros, A. Znaor dan . F. Bray, “Estimating the global cancer incidence and mortality in 2018: GLOBOCAN sources and methods,” *International Journal of Cancer*, vol. 144, no. 8, pp. 1941-1953, 2019.
- [3] A. S. Suriadiredja, “Epidemiologi kanker kulit,” *MDVI*, vol. 38, no. 2, p. 61, 2011.
- [4] P. Tschandl, C. Rosendahl dan H. Kittler, “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,” *Scientific Data*, vol. 5, no. 1, 2018.
- [5] T. R. Savera, W. H. Suryawan dan A. W. Setiawan, “Deteksi Dini Kanker Kulit Menggunakan K-NN Dan Convolutional Neural Network,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 2, pp. 373-378, 2020.
- [6] Z. Aliyah, A. Arifianto dan F. Sthevanie, “Classifying Skin Cancer in Digital Images Using Convolutional Neural Network with Augmentation,” *Ind. Journal on Computing*, vol. 5, no. 2, pp. 55-66, 2020.
- [7] A. Syukron dan A. Subekti , “Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit,” *JURNAL INFORMATIKA*, vol. 5, no. 2, pp. 175-185, 2018.
- [8] R. Siringoringo, “Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor,” *Jurnal ISD*, vol. 3, no. 1, 2018.
- [9] R. Halodoc, “Kanker Kulit,” HALODOC, 2021. [Online]. Available: <https://www.halodoc.com/kesehatan/kanker-kulit>. [Diakses 6 Agustus 2021].
- [10] D. M. Parkin, D. Mesher dan P. Sasieni, “Cancers attributable to solar (ultraviolet) radiation exposure in the UK in 2010,” *Br J Cancer*, vol. 105, no. 2, pp. S66-S69, 2011.
- [11] R. S. Stern, “Prevalence of a history of skin cancer in 2007: results of an incidence-based model,” *Arch Dermatol*, vol. 146, no. 3, pp. 279-282, 2010.
- [12] H. W. Rogers, M. A. Weinstock, S. R. Feldman dan B. M. Coldiron, “Incidence Estimate of Nonmelanoma Skin Cancer (Keratinocyte Carcinomas) in the U.S. Population,” *JAMA Dermatol*, vol. 151, no. 10, pp. 1081-1086, 2015.
- [13] I. The Lewin Group, The Burden of Skin Diseases 2005, Washington, D.C.: The Society for Investigative Dermatology and The American Academy of Dermatology Association, 2005.

- [14] A. C. Society, “Cancer Facts & Figures 2021,” 2021. [Online]. Available: <https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2021/cancer-facts-and-figures-2021.pdf>. [Diakses 6 Agustus 2021].
- [15] H. M. Gloster Jr dan K. Neal, “Skin cancer in skin of color,” *J Am Acad Dermatol*, vol. 55, no. 5, pp. 741-760, 2006.
- [16] A. A. o. D. Association, “Skin Cancer,” American Academy of Dermatology Association, 2021. [Online]. Available: <https://www.aad.org/media/stats-skin-cancer>. [Diakses 6 Agustus 2021].
- [17] A. B. Mutiara, “Implementasi Deep Learning: Matlab dan Python-Keras-Tensorflow,” 2020. [Online]. Available: <https://mooc.aptikom.or.id/mod/resource/view.php?id=1095>. [Diakses 6 Agustus 2021].
- [18] F. Chollet, Deep Learning with Python, Shelter Island: Deep Learning with Python, 2018.
- [19] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,” 16 Desember 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Diakses 6 Agustus 2021].
- [20] Q. Lina, “Apa itu Convolutional Neural Network?,” 2019. [Online]. Available: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>. [Diakses 7 Agustus 2021].
- [21] Sofyan, “<http://sofyantandungan.com/pengenalan-convolutional-neural-network-part-1/>,” 2019. [Online]. Available: <http://sofyantandungan.com/pengenalan-convolutional-neural-network-part-1/>. [Diakses 7 Agustus 2021].
- [22] M. Buda, A. Maki dan M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, p. 249–259, 2018.
- [23] G. Lemaitre, F. Nogueira dan C. K. Aridas , “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning,” *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1-5, 2017.
- [24] G. E. A. P. A. B. A. L. C. a. M. M. A. Batista, “Balancing Training Data for Automated Annotation of Keywords: Case Study,” *Proceedings of the Second Brazilian Workshop on Bioinformatics*, p. 35–43, 2003.
- [25] G. E. A. P. A. P. R. C. a. M. M. C. Batista, “A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20-29, 2004.
- [26] B. Alexander Yun-chung Liu, Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets, Austin: The University of Texas, 2004.
- [27] e. a. H. Han, “Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning,” *Advances in intelligent computing*, pp. 878-887, 2005.

- [28] H. He, Y. Bai, E. . A. Garcia dan S. Li, “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322-1328, 2008.
- [29] R. Adam, “Mengenal Google Colab,” 2019. [Online]. Available: <https://structilm.com/2019/05/mengenal-google-colab/>. [Diakses 7 Agustus 2021].
- [30] E. Tjioe, “Klasifikasi Gambar menggunakan Keras,” 2019. [Online]. Available: <https://rpubs.com/enlik/keras>. [Diakses 7 Agustus 2021].
- [31] “Konsep Convolutional Neural Network (CNN),” 2020. [Online]. Available: <https://iglab.tech/konsep-convolutional-neural-network-cnn/>. [Diakses 10 Agustus 2021].
- [32] <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>, “Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning,” 2019. [Online]. Available: <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>. [Diakses 10 Agustus 2021].
- [33] DATAQ, “Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning,” 2013. [Online]. Available: <https://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/>. [Diakses 11 Agustus 2021].

LAMPIRAN

Lampiran 1. *Source Code Program Model Klasifikasi Kanker Kulit di Google Colaboratory Menggunakan Bahasa Pemrograman Python*

```
# -*- coding: utf-8 -*-
"""SKIN_CANCER_CLASSIFICATION.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/16nvPsAjsqmHW
CGoHVGvGpaN8xVI3cjdA

# Skin Lesion Analyser (Skin Cancer Classification
Based on Lesion Pigment)
---
"""

""" Prepare all required library """
!pip install -q kaggle
!pip install visualkeras
# !pip install keras==2.3.0
# !pip install tensorflow
# !pip install scikit-learn

# Data Preparation
import os
import tensorflow as tf
from tensorflow import keras
import numpy as np # for linear algebra

# Colab library to upload files to notebook
from google.colab import files

# Visualization lib
import pandas as pd # for data processing, CSV file
I/O
import seaborn as sn
import json
import glob
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
from collections import Counter
```

```
from sklearn.datasets import make_classification
from matplotlib import pyplot
from numpy import where

# Report Metrics
from sklearn import metrics

# Model Development
from sklearn.model_selection import train_test_split
from tensorflow import keras
from keras.preprocessing.image import
ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout,
Flatten, Conv2D, MaxPool2D, BatchNormalization,
LeakyReLU, GlobalMaxPooling2D, Activation
from keras.callbacks import ModelCheckpoint,
EarlyStopping
from keras.optimizers import Adam
from glob import glob
from PIL import Image
import visualkeras

""" Download Dataset """
# Upload kaggle API key file
uploaded = files.upload()

!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/

# This permissions change avoids a warning on Kaggle
tool startup.
!chmod 600 ~/.kaggle/kaggle.json

!kaggle datasets download -d kmader/skin-cancer-
mnist-ham10000

!unzip skin-cancer-mnist-ham10000.zip -d dataset

!ls /content/dataset

print(len(os.listdir("dataset/HAM10000_images_part_1"
)))
print(len(os.listdir("dataset/HAM10000_images_part_2"
)))

""" Showing metadata of HAM10000 dataset"""
```

```
df_data =
pd.read_csv('dataset/HAM10000_metadata.csv')

print('Metadata dataset')
df_data.head(10)

# Data distribution visualization
fig = plt.figure(figsize=(16,12))

class_plot = fig.add_subplot(221)
df_data['dx'].value_counts().plot(kind='bar',
ax=class_plot)
class_plot.set_ylabel('Count', size=12)
class_plot.set_title('Cancer Class')

sex_plot = fig.add_subplot(222)
df_data['sex'].value_counts().plot(kind='bar',
ax=sex_plot)
sex_plot.set_ylabel('Count', size=12)
sex_plot.set_title('Sex')

age_plot = fig.add_subplot(223)
df_data['age'].value_counts().plot(kind='bar')
age_plot.set_ylabel('Count', size=12)
age_plot.set_title('Age')

loc_plot = fig.add_subplot(224)
df_data['localization'].value_counts().plot(kind='bar')
loc_plot.set_ylabel('Count', size=12)
loc_plot.set_title('Localization')

plt.tight_layout()
plt.show()

# crating one hot encoding

# Class Cancer yang terdapat pada dataset
class_cancer = {
    'nv': 'Melanocytic nevi',
    'mel': 'Melanoma',
    'bkl': 'Benign keratosis ',
    'bcc': 'Basal cell carcinoma',
    'akiec': 'Actinic keratoses',
    'vasc': 'Vascular lesions',
    'df': 'Dermatofibroma'
```

```

}

sex_dict = {
    'female' : 0,
    'male' : 1
}
location_dict = {
'scalp' : 0 , 'ear' : 1, 'face' : 2, 'back' : 3,
'trunk' : 4, 'chest' : 5,
    'upper extremity' : 6, 'abdomen' : 7,
'unknown' : 8, 'lower extremity' : 9,
    'genital' : 10, 'neck' : 11, 'hand' : 12,
'foot' : 13, 'acral' : 14}

#create a modification dataframe
df_data['sex'] = df_data['sex'].map(sex_dict.get)
df_data['localization'] =
df_data['localization'].map(location_dict.get)
df_data['lesion']= df_data.dx.map(class_cancer)
df_data['lesion_idx'] =
pd.Categorical(df_data['lesion']).codes
df_data.sample(5)

print('Kelas yang terdapat pada dataset awal :',
df_data['dx'].unique())
print(df_data.lesion.value_counts())

""" Checking null dataset"""

df_data.isna().sum()

# # Cleaning the data
# df_data['age'].fillna((df_data['age'].mean()),
inplace=True)
#
df_data['sex'].fillna(df_data['sex'].value_counts().i
ndex[0], inplace=True)
# df_data.isnull().any()

# Merging images from both folders
HAM10000_images_part1.zip and
HAM10000_images_part2.zip into one dictionary
imageid_path_dict =
{os.path.splitext(os.path.basename(x))[0]: x
            for x in
glob(os.path.join('/content/dataset/', '*',
'*.*'))}

```

```

df_data['path'] =
df_data['image_id'].map(imageid_path_dict.get)
df_data.sample(10)

# from skimage.io import imread
# df_data['image'] = df_data['path'].map(imread)

# Loading and resize images
df_data['image'] = df_data['path'].map(lambda x:
np.asarray(Image.open(x).resize((64, 64)))) # pixels
of image

# Checking the image size distribution
df_data['image'].map(lambda x:
x.shape).value_counts()

df_data['image'].head()

# Show image for each class cancer (dx)
n_samples = 5
fig, m_axs = plt.subplots(7, n_samples, figsize =
(4*n_samples, 3*7))
for n_axs, (type_name, type_rows) in zip(m_axs,
df_data.sort_values(['lesion']).groupby('lesion')):
    n_axs[0].set_title(type_name)
    for c_ax, (_, c_row) in zip(n_axs,
type_rows.sample(n_samples,
random_state=10).iterrows()):
        c_ax.imshow(c_row['image'])
        c_ax.axis('off')

df_data.info()

df_data.to_csv ('metadata_skincancer.csv', index =
False, header=True)

""" Mengolah Model Menggunakan data
hmnist_28_28_RGB.csv"""

import seaborn as sns
import imblearn

# Load file CSV contains pixel values
Path = "/content/dataset/hmnist_28_28_RGB.csv"
DataSet = pd.read_csv(Path)
Label = DataSet["label"]
Data = DataSet.drop(columns=["label"])

```

```
plt.figure(figsize = (10,10))
sns.set_style("darkgrid")
plot_data = sns.countplot(Label)

# PILIH SALAH SATU TEKNIK RESAMPLING YANG INGIN
# DIGUNAKAN (UNCOMMENT CODES TEKNIK YANG INGIN
# DIGUNAKAN)
# Resampling Dataset Using Random Under Sampler
from imblearn.under_sampling import
RandomUnderSampler
RUS = RandomUnderSampler(random_state=0)
Data_balanced,Label_balanced =
RUS.fit_resample(Data,Label)
Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using Cluster Centroids
# from imblearn.under_sampling import
RandomUnderSampler
# from imblearn.under_sampling import
ClusterCentroids
# cc = ClusterCentroids(random_state=0)
# Data_balanced,Label_balanced =
cc.fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using Random Over Sampler
# ROS = RandomOverSampler(random_state=42)
# Data_balanced,Label_balanced =
ROS.fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using SMOTE
# from imblearn.over_sampling import SMOTE
# sm = SMOTE(random_state=42)
# Data_balanced,Label_balanced =
sm.fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using ADASYN
```

```

# from imblearn.over_sampling import ADASYN
# Data_balanced,Label_balanced =
ADASYN().fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using Oversampling SMOTE +
Undersampling Tomek Link
# from imblearn.combine import SMOTETomek
# smt = SMOTETomek(random_state=0)
# Data_balanced,Label_balanced =
smt.fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

# # Resampling Dataset Using Oversampling SMOTE +
Undersampling ENN
# from imblearn.combine import SMOTEENN
# smtn = SMOTEENN(random_state=42)
# Data_balanced,Label_balanced =
smtn.fit_resample(Data,Label)
# Data_balanced = np.array(Data_balanced).reshape(-
1,28,28,3)
# print('Shape of Data :', Data_balanced.shape)

DataSet["label"].value_counts().sort_index()

plt.figure(figsize = (10,10))
sns.set_style("darkgrid")
sns.countplot(Label_balanced)

pd.Series(Label_balanced).value_counts().sort_index()

# Show scatter plot of balanced data
counter = Counter(Label_balanced)
print(counter)
# scatter plot of examples by class label
for label, _ in counter.items():
    row_ix = where(Label_balanced == label)[0]
    pyplot.scatter(Data_balanced[row_ix, 0],
Data_balanced[row_ix, 1], label=str(label))
pyplot.legend()
pyplot.show()

Label_balanced = np.array(Label_balanced)

```

```
Label_balanced

""" Splitting data into 75% train data and 25% test
data"""

from sklearn.model_selection import train_test_split
x_train , x_test , y_train , y_test =
train_test_split(Data_balanced, Label_balanced ,
test_size = 0.25 , random_state = 42)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

""" Convert labels into One-hot encoding"""

num_classes = 7
y_train = tf.keras.utils.to_categorical(y_train,
num_classes)
y_test = tf.keras.utils.to_categorical(y_test,
num_classes)

print("y_train")
print(y_train)
print("\n")
print("y_test")
print(y_test)

img = x_train[0]
print(img)
print(img.shape)
plt.imshow(img)

# Image Augmentation
from tensorflow.keras.preprocessing.image import
ImageDataGenerator

aug = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode="nearest")
```

```

""" Make a Model Architecture """

METRICS = [
    keras.metrics.TruePositives(name='tp'),
    keras.metrics.FalsePositives(name='fp'),
    keras.metrics.TrueNegatives(name='tn'),
    keras.metrics.FalseNegatives(name='fn'),
    keras.metrics.BinaryAccuracy(name='accuracy'),
    keras.metrics.Precision(name='precision'),
    keras.metrics.Recall(name='recall'),
    keras.metrics.AUC(name='auc'),
    keras.metrics.AUC(name='prc', curve='PR'), #
precision-recall curve
]

# Reference Material :
https://www.tensorflow.org/tutorials/structured\_data/imbalanced\_data#define\_the\_model\_and\_metrics
# Reference Material :
https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8
# Generally, the use of ROC curves and precision-recall curves are as follows:
# ROC curves should be used when there are roughly equal numbers of observations for each class.
# Precision-Recall curves should be used when there is a moderate to large class imbalance.

def make_model(metrics=METRICS, output_bias=None):
    if output_bias is not None:
        output_bias =
tf.keras.initializers.Constant(output_bias)
    model = keras.models.Sequential([
        # model = Sequential()
        keras.layers.Conv2D(32, kernel_size = (3,3),
input_shape = (28, 28, 3), activation = 'relu',
padding = 'same'),
        keras.layers.BatchNormalization(),
        # keras.layers.Dropout(0.2),
        keras.layers.MaxPool2D(pool_size = (2,2)),

        keras.layers.Conv2D(64, kernel_size = (3,3),
activation = 'relu', padding = 'same'),
        # model.add(Conv2D(128, kernel_size = (3,3),
activation = 'relu'))
    ])

```

```

keras.layers.BatchNormalization(),
# keras.layers.Dropout(0.3),
keras.layers.MaxPool2D(pool_size = (2,2)),

    keras.layers.Conv2D(128, kernel_size = (3,3),
activation = 'relu', padding = 'same'),
    keras.layers.BatchNormalization(),
# keras.layers.Dropout(0.2),
keras.layers.MaxPool2D(pool_size = (2,2)),

    keras.layers.Conv2D(256, kernel_size = (3,3),
activation = 'relu', padding = 'same'),
    keras.layers.BatchNormalization(),
# keras.layers.Dropout(0.2),
keras.layers.MaxPool2D(pool_size = (2,2)),

keras.layers.Flatten(),
keras.layers.Dense(256, activation = 'relu'),
# keras.layers.Dropout(0.25),
keras.layers.BatchNormalization(),
keras.layers.Dense(128, activation = 'relu'),
# keras.layers.Dropout(0.2),
keras.layers.BatchNormalization(),
keras.layers.Dense(64, activation = 'relu'),
# keras.layers.Dropout(0.15),
keras.layers.BatchNormalization(),
keras.layers.Dense(32, activation = 'relu',
kernel_regularizer=keras.regularizers.L1L2()),
    keras.layers.BatchNormalization(),
    keras.layers.Dense(7, activation='softmax')))

model.compile(loss='categorical_crossentropy',
optimizer='Adam', metrics=METRICS)

return model

model = make_model()
# model.summary()

from keras.utils.vis_utils import plot_model
keras.utils.plot_model(model, to_file='model.png',
show_shapes = True, show_layer_names = True,
rankdir="TB", expand_nested = True, dpi = 96)

from PIL import ImageFont
font = ImageFont.load_default()

```

```

visualkeras.layered_view(model, legend=True,
font=font, to_file='visualkeras.png')

""" Set for callbacks"""

from tensorflow.keras.callbacks import
ReduceLROnPlateau, ModelCheckpoint, EarlyStopping

# https://keras.io/api/callbacks/model_checkpoint/
checkpoint = ModelCheckpoint("best_model_cutix.h5",
                             monitor='val_fn',
                             verbose=1,
                             save_best_only=True,
                             save_weights_only=False,
                             mode='min'
                             )

#
# https://keras.io/api/callbacks/reduce_lr_on_plateau/
learning_rate_reduction =
ReduceLROnPlateau(monitor='val_recall',
                  patience=10,
                  verbose=1,
                  # factor=0.35, #awal
                  factor=0.45,
                  min_lr=1e-10)

# https://keras.io/api/callbacks/early_stopping/
early_stopping = EarlyStopping(
    monitor='val_loss',
    verbose=1,
    patience=10,
    mode='min',
    restore_best_weights=True)

""" Training Model """
history = model.fit(
    x = aug.flow(x_train, y_train),
    epochs=70,
    batch_size = 64,
    validation_data=(x_test, y_test),
    # steps_per_epoch=len(x_train) // batch_size,
)

```

```

# shuffle=True, # testing shuffle
    callbacks=[learning_rate_reduction,
early_stopping, checkpoint])

import matplotlib as mpl
mpl.rcParams['figure.figsize'] = (12, 10)
colors =
plt.rcParams['axes.prop_cycle'].by_key()['color']

def plot_metrics(history):
    metrics = ['loss', 'prc', 'precision', 'recall']
    for n, metric in enumerate(metrics):
        name = metric.replace("_", " ").capitalize()
        plt.subplot(2,2,n+1)
        plt.plot(history.epoch, history.history[metric],
color=colors[0], label='Train')
        plt.plot(history.epoch,
history.history['val_'+metric],
color=colors[0], linestyle="--",
label='Val')
        plt.xlabel('Epoch')
        plt.ylabel(name)
        if metric == 'loss':
            plt.ylim([0, plt.ylim()[1]])
        elif metric == 'auc':
            plt.ylim([0.8,1])
        else:
            plt.ylim([0,1])

    plt.legend()

plot_metrics(history)

""" Evaluate Model Accuracy"""

score = model.evaluate(x_test, y_test)
print('loss :', score[0])
print('true positive : ', score[1])
print('false positive : ', score[2])
print('true negative : ', score[3])
print('false negative : ', score[4])
print('accuracy : ', score[5])
print('precision : ', score[6])
print('recall : ', score[7])
print('auc : ', score[8])

```

```

"""# Visualization Accuracy and Validation Accuracy
Model"""

import matplotlib.pyplot as plt
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.figure(figsize=(8,6))
plt.plot(epochs, acc, 'bo', label='Training
accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation
accuracy')
plt.title('Training and validation accuracy')

plt.figure(figsize=(8,6))
plt.plot(epochs, loss, 'bo', label='Training Loss')
plt.plot(epochs, val_loss, 'b', label='Validation
Loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()

"""# Using Function of Model Predict"""

# Prediction on test data
y_pred = model.predict(x_test)
# Convert predictions classes to one hot vectors
y_pred_classes = np.argmax(y_pred, axis = 1)
# Convert test data to one hot vectors
y_true = np.argmax(y_test, axis = 1)

""" Save Model """

import time
saved_model_path = "./model_cutix.h5"
model.save(saved_model_path)

""" Classification Report """

from sklearn.metrics import classification_report

```

```

Y_pred = model.predict(x_test).round()
classes = {4: ('nv', 'melanocytic nevi'),
            6: ('mel', 'melanoma'),
            2 :('bkl', 'benign keratosis-like
lesions'),
            1:('bcc' , 'basal cell carcinoma'),
            5: ('vasc', 'vascular lesions'),
            0: ('akiec', 'Actinic keratoses and
intraepithelial carcinomae'),
            3: ('df', 'dermatofibroma')}

target_names = [f"{classes[i]} " for i in range(7)]
print(classification_report(y_test, Y_pred,
target_names = target_names))

from sklearn.metrics import confusion_matrix
f, ax = plt.subplots(figsize=(10,10))
cm = confusion_matrix(y_test.argmax(axis = 1) ,
y_pred.argmax(axis = 1))
cm = pd.DataFrame(cm , index = [i for i in
range(7)] , columns = [i for i in range(7)])
cm_labels = ['akiec', 'bcc', 'bkl', 'df', 'nv',
'vasc','mel']
sns.heatmap(cm,cmap= "Blues", linecolor = 'black' ,
linewidth = 1 , annot = True, fmt='',
            yticklabels = cm_labels,
            xticklabels = cm_labels, ax=ax)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.show()

""" Upload Photo for Testing Classification """
from tensorflow.keras.preprocessing import image
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():

    # predicting images
    path = fn
    img = image.load_img(path, target_size=(28, 28))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

```

```
images = np.vstack([x])
classes = model.predict(images, batch_size=10)
print(fn)
print(classes)

cancer_class = np.argmax(classes)

if cancer_class == 0:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
akiec')

elif cancer_class == 1:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
bcc')

elif cancer_class == 2:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
bkl')

elif cancer_class == 3:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
df')

elif cancer_class == 4:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
nv')

elif cancer_class == 5:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
vasc')

elif cancer_class == 6:
    print('Penyakit yang anda alami diprediksi
adalah Kanker Kulit yang termasuk dalam kelas (jenis)
mel')
```

Lampiran 2. Source Code Program Wesbite di Visual Studio Code

1. Source code program index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
            <title>Cutix Skin Cancer Class Classification</title>
            <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
            <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
            <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
            <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
                <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
                <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
            </head>
            <body>
                <nav>
                    <div class="wrapper">
                        <div class="logo"><a href='>Cutix Web Based</a></div>
                        <div class="menu">
                            <ul>
                                <li><a href="#home">Home</a></li>
                                <li><a href="#prediction">Prediction</a></li>
                                    <li><a href="#tipe-cancer">Type Explanation</a></li>
                                        <li><a href="#partners">Partners</a></li>
```

```

<li><a href="#contact">Contact</a></li>
</ul>
</div>
</div>
</nav>
<div class="wrapper">
    <!-- untuk home -->
    <section id="home">
        
        <div class="kolom">
            <p class="deskripsi">Kanker Kulit</p>
            <h2>Pengertian Kanker Kulit</h2>
            <p>Kanker kulit diberi nama sesuai dengan jenis sel yang terpengaruh yaitu sel kulit. Kanker kulit merupakan kanker yang umum terjadi.
            Menurut World Health Organization, sebanyak 160.000 orang mengidap kanker kulit setiap tahun di dunia.
            Tingkat kejadian kanker kulit di seluruh dunia telah meningkat pesat.
            Walaupun insidensi kanker kulit cukup tinggi, angka kematian yang diakibatkan olehnya tidak terlalu tinggi.
            Karena kulit adalah organ terbesar pada tubuh manusia, organ yang paling terlihat secara kasat mata, mudah bagi kita untuk mendeteksi adanya kelainan pada kulit.
            Di Indonesia, penderita kanker kulit terbilang relatif sedikit apabila dibandingkan dengan ketiga negara tersebut.
            Namun demikian, kanker kulit ini harus mendapat perhatian sedini mungkin, karena tidak hanya dapat menyebabkan kecacatan yang dapat merusak penampilan, tetapi juga dapat mengakibatkan kematian apabila sudah masuk ke stadium lanjut.
            Oleh karena itu, kanker kulit bisa dan harus diobati secara dini.</p>
        </div>
    </section>

```

```

<!-- untuk prediction -->
<section id="prediction">
    <div class="kolom">
        <p class="deskripsi">Prediction Skin Cancer</p>
            <h2>Prediksi Tipe Kanker Kulit</h2>
            <p>Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.</p>
                <p>Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya, Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.</p>
                    <p>Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).</p>
                    <p>Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).</p>
                    <p>Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter kulit.</p>
                </div>
<!-- -->
    <div class="predict-section">
        <p style="font-size: 20px; color: #364f6b; font-weight: bold; font-family: comic sans ms;">Upload foto kulit untuk diprediksi!</p>
            <form id="upload-file" method="post" enctype="multipart/form-data">
                <label for="imageUpload" class="upload-label">
                    Upload Photo
                </label>
                <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
            </form>

```

```

<div class="image-section"
style="display:none;">
    <div class="img-preview">
        <div id="imagePreview">
        </div>
    </div>
    <div>
        <button type="button"
class="btn btn-primary btn-lg " id="btn-
predict">Predict!</button>
    </div>
</div>

    <div class="loader"
style="display:none;"></div>

    <h3 id="result">
        <span> </span>
    </h3>

    </div>
</section>

<!-- untuk penjelasan tipe cancer -->
<section id="tipe-cancer">
    <div class="tengah">
        <div class="kolom">
            <p class="deskripsi">Tipe/Jenis
Kanker Kulit</p>
            <h2>7 Tipe Kanker Kulit</h2>
        </div>

        <div class="tipe-cancer-list">
            <p><b>"Actinic Keratoses
(akiec)"</b> adalah varian non-invasif yang umum dari
karsinoma sel skuamosa yang dapat diobati tanpa
operasi. Kulit di sekitarnya biasanya ditandai dengan
kerusakan parah akibat sinar matahari (sinar UV).
Akiec lebih sering terjadi pada wajah, sering
terlihat seperti bercak kecil, kasar, bersisik pada
kulit.</p>
            <p><b>"Basal cell carcinoma
(bcc)"</b> adalah varian umum dari kanker kulit
epitel yang jarang bermetastasis (menyebar). biasanya
dimulai sebagai benjolan berbentuk mutiara di bagian
wajah, telinga atau leher. Jika berkembang di bagian

```

dada atau punggung, kanker ini terlihat seperti bagian kulit yang bersisik atau kering.</p>
 <p>"Benign keratosis-like lesions (bkl)" adalah kelas generik jinak yang termasuk seborrheic keratosis ("kutil pikun"). Dari pandangan dermatoscopic, bkl dapat menunjukkan bentuk yang meniru melanoma. Tampilan dermatoskopiknya bervariasi menurut lokasi anatomi dan jenisnya.</p>
 <p>"Dermatofibroma (df)" adalah lesi kulit jinak yang dianggap sebagai proliferasi jinak atau reaksi inflamasi terhadap minimal trauma. Presentasi dermatoskopik yang paling umum adalah garis retikuler di perifer dengan patch putih berpusat yang menunjukkan fibrosis.</p>
 <p>"Melanocytic nevi (nv)" adalah tumor jinak dan muncul dalam berbagai varian, yang semuanya termasuk dalam seri nv. Varian ini mungkin berbeda secara signifikan dari sudut pandang dermatoskopik. Berbeda dengan melanoma, mereka biasanya simetris dalam hal distribusi warna dan struktur bentuk.</p>
 <p>"Melanoma (mel)" adalah salah satu bentuk kanker kulit yang lebih serius dan terkadang muncul dari tahi lalat yang ada di kulit. Lebih umum melanoma muncul sebagai tanda atau tahi lalat baru pada kulit normal. Melanoma dapat muncul di mana saja di tubuh tetapi paling sering terjadi pada pria di punggung, dan pada wanita di kaki.</p>
 <p>"Vascular lesions (vasc)" dalam kumpulan data dari angioma ceri hingga angiokeratoma dan granuloma piogenik. Angioma secara dermatoskopis ditandai dengan warna merah atau ungu dan struktur padat berbatas yang dikenal sebagai gumpalan merah (lakuna). Vasc adalah kelainan yang relatif umum pada kulit dan jaringan di bawahnya, lebih dikenal sebagai tanda lahir.</p>
 </div>
 </div>
</section>

<!-- untuk partners -->
<section id="partners">
 <div class="tengah">
 <div class="kolom">
 <p class="deskripsi">Partners</p>
 <h2>Partners</h2>

```
<p>Terima kasih atas segala  
bentuk bantuan dan ilmunya.</p>  
</div>  
  
<div class="partner-list">  
    <div class="kartu-partner">  
          
        <p>Ali Rohman</p>  
        <p>H1A018021</p>  
        <p>Teknik Elektro Unsoed</p>  
    </div>  
    <div class="kartu-partner">  
          
        <p>Arif Fahrudin</p>  
        <p>H1A018069</p>  
        <p>Teknik Elektro Unsoed</p>  
    </div>  
    <div class="kartu-partner">  
          
        <p>Nur Ardli</p>  
        <p>H1A018045</p>  
        <p>Teknik Elektro Unsoed</p>  
    </div>  
    </div>  
    </div>  
</section>  
</div>  
  
<div id="contact">  
    <div class="wrapper">  
        <div class="footer">  
            <div class="footer-section">  
                <h3>Cutix</h3>  
                <p>Cutix adalah media klasifikasi  
berbasis website untuk screening klasifikasi 7 tipe
```

```

kanker kulit berdasarkan kondisi lesi kulit
berpigmen.</p>
        </div>
        <div class="footer-section">
            <h3>Disclaimer</h3>
            <p>Hasil klasifikasi yang didapat
bersifat PREDIKTIF(Prediksi), walaupun akurasi model
99% ini hanya sebagai media screening dan tetap harus
dilakukan pemeriksaan lebih lanjut oleh dokter kulit
untuk kepastian.</p>
            <p><a
href="https://dataverse.harvard.edu/dataset.xhtml?per
sistentId=doi:10.7910/DVN/DBW86T">Detail Penjelasan
Kanker Kulit</a></p>
        </div>
        <div class="footer-section">
            <h3>Address</h3>
            <p>Jl. Patuha Raya, Kayuringin
Jaya, Bekasi Selatan, Kota Bekasi. Kode Pos:
17144</p>
        </div>
        <div class="footer-section">
            <h3>Social Media</h3>
            <p><b>Instagram : </b><a
href="https://www.instagram.com/alirohman57/">Ali
Rohman</a></p>
            <p><b>LinkedIn : </b><a
href="https://www.linkedin.com/in/alirohman/">Ali
Rohman</a></p>
            <p><b>Github : </b><a
href="https://github.com/batslay">Ali Rohman<a/></p>
            <p><b>No. HP :
</b>089669232481</p>
        </div>
    </div>
<div id="copyright">
    <div class="wrapper">
        &copy; 2021. <b>Cutix.</b> All Rights
Reserved.
    </div>
</div>

</body>

```

```

<footer>
    <script src="{{ url_for('static',
filename='js/main.js') }}"
type="text/javascript"></script>
</footer>

</html>

```

2. Source code program main.css

```

* {
    text-decoration: none;
    margin: 0px;
    padding: 0px;
}

body {
    margin: 0px;
    padding: 0px;
    font-family: 'Open Sans', sans-serif;
}

.wrapper {
    width: 1100px;
    margin: auto;
    position: relative;
}

.logo a {
    font-size: 50px;
    font-weight: 800;
    float: left;
    font-family: courier;
    color: #364f6b;
}

.footer-section a {
    color: #364f6b;
}

.menu {
    float: right;
}

nav {

```

```
width: 100%;  
margin: auto;  
display: flex;  
line-height: 55px;  
position: sticky;  
position: -webkit-sticky;  
top: 0;  
background: #FFFFFF;  
z-index: 1;  
border-bottom:1px solid #364f6b;  
}  
  
nav ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
}  
  
nav ul li {  
    float: left;  
}  
  
nav ul li a {  
    color: black;  
    font-weight: bold;  
    text-align: center;  
    padding: 0px 16px 0px 16px;  
    text-decoration: none;  
}  
  
nav ul li a:hover {  
    text-decoration: underline;  
}  
  
section {  
    margin: auto;  
    display: flex;  
    margin-bottom: 50px;  
}  
  
.kolom {  
    margin-top: 50px;  
    margin-bottom: 0px;  
}  
  
.predict-section {
```

```
margin-top: 50px;
margin-bottom: 50px;
margin-left: 100px;
}

.kolom .deskripsi {
    font-size: 20px;
    font-weight: bold;
    font-family: 'comic sans ms';
    color: #364f6b;
}

h2 {
    font-family: 'comic sans ms';
    font-weight: 800;
    font-size: 45px;
    margin-bottom: 20px;
    margin-right: 50px;
    color: #364f6b;
    width: 100%;
    line-height: 50px;
}

a.tbl-biru {
    background: #3f72af;
    border-radius: 20px;
    margin-top: 20px;
    padding: 15px 20px 15px 20px;
    color: #FFFFFF;
    cursor: pointer;
    font-weight: bold;
}

a.tbl-biru:hover {
    background: #fc5185;
    text-decoration: none;
}

a.tbl-pink {
    background: #fc5185;
    border-radius: 20px;
    margin-top: 20px;
    padding: 15px 20px 15px 20px;
    color: #FFFFFF;
    cursor: pointer;
    font-weight: bold;
}
```

```
a.tbl-pink:hover {
    background: #3f72af;
    text-decoration: none;
}

p {
    margin: 0px 0px 5px 0px;
    padding: 0px 0px 5px 0px;
    font-family: 'Cambria';
    text-align: justify;
}

.tengah {
    text-align: center;
    width: 100%;
}

.tutor-list {
    width: 100%;
    position: relative;
    display: flex;
    flex-wrap: wrap;
}

.tipe-cancer-list p{
    border-radius: 10px;
    background-color: #E8F6EF;
    border: 3px #E8F6EF groove;
    padding: 5px;
    text-align: justify;
}

.kartu-partner {
    width: 20%;
    margin: 0 auto;
}

.kartu-partner img {
    width: 80%;
    border-radius: 50%;
    margin-bottom: 20px;
}

.kartu-partner p {
    font-family: 'comic sans ms';
```

```
    font-weight: 800;
    font-size: 15px;
    text-align:center;
    color: #364f6b;
    /* margin-left: 55px; */
}

.partner-list {
    width: 100%;
    position: relative;
    display: flex;
    flex-wrap: wrap;
}

#contact {
    background: #dedede;
    padding:50px 0px 50px 0px;
}

.footer {
    width: 100%;
    position: relative;
    display: flex;
    /* flex-wrap: wrap; */
    margin: 0px 20px 0px 20px;
}

.footer-section {
    width: 75%;
    margin: 0px 45px 0px 45px;
}

h3 {
    font-family: 'comic sans ms';
    font-weight: 800;
    font-size: 30px;
    margin-bottom: 20px;
    color: #364f6b;
    width: 100%;
    line-height: 50px;
}

#copyright {
    text-align: center;
    width: 100%;
    padding: 50px 0px 50px 0px;
    margin-top: 50px;
```

```
}
```

```
@media screen and (max-width: 991.98px) {
```

```
    .wrapper {
```

```
        width: 90%;
```

```
    }
```

```
    .logo a {
```

```
        display: block;
```

```
        width: 100%;
```

```
        text-align: center;
```

```
    }
```

```
    nav .menu {
```

```
        width: 100%;
```

```
        margin: 0;
```

```
    }
```

```
    nav .menu ul {
```

```
        text-align: center;
```

```
        margin: auto;
```

```
        line-height: 60px;
```

```
    }
```

```
    nav .menu ul li {
```

```
        display: inline-block;
```

```
        float: none;
```

```
    }
```

```
    section {
```

```
        display: block;
```

```
    }
```

```
    section img {
```

```
        display: block;
```

```
        width: 100%;
```

```
        height: auto;
```

```
    }
```

```
    .kartu-tutor {
```

```
        width: 50%;
```

```
    }
```

```
    .kartu-partner {
```

```
        width: 50%;
```

```
    }
```

```
}
```

```
/*for prediction*/

.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 0px;
    margin-bottom: 20px;
    margin-left: 0px;

}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

/**/
input[type="file"] {
/*    display: none; */
    width: 550px;
/*    padding: 15px 20px 15px 20px; */
    visibility:hidden;
}

.upload-label{
    display: inline-block;
/*    padding: 12px 30px;*/
/*    background: #39D2B4;*/
/*    color: #fff;*/
    font-size: 1em;
    transition: all .4s;
/*    font-family: sans-serif;*/
    font-family: 'Cambria';
    border-radius: 10px;
    margin-top: 10px;
    margin-left: 0px;
    padding: 15px 20px 15px 20px;
    background: #3f72af;
    color: #FFFFFF;
```

```

        cursor: pointer;
        font-weight: bold;
    }

.upload-label:hover{
/*     background: #34495E;*/
/*     color: #39D2B4;*/
    background: #fc5185;
    text-decoration: none;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

```

3. *Source code* program main.js

```

from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
from PIL import Image as pil_image
import tensorflow
import keras

# Keras
from tensorflow.keras.applications.imagenet_utils
import preprocess_input, decode_predictions
from tensorflow.keras.models import Model ,
load_model
from keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request,
render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

app = Flask(__name__)

```

```

Model= load_model('best_model_cutix.h5')

class_cancer = {
    0 : 'Actinic keratoses (akiec)',
    1 : 'Basal cell carcinoma (bcc)',
    2 : 'Benign keratosis-like lesions (bkl) ',
    3 : 'Dermatofibroma (df)',
    4 : 'Melanocytic nevi (nv)',
    5 : 'Vascular lesions (vasc)',
    6 : 'Melanoma (mel)'
}

def model_predict(img_path, Model):
    img = image.load_img(img_path,
target_size=(28,28,3))

    #img =
np.asarray(pil_image.open('img').resize((120,90)))
    #x = np.asarray(img.tolist())

    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    # Be careful how your trained model deals with
    the input
    # otherwise, it won't make correct prediction!
    #x = preprocess_input(x, mode='caffe')

    preds = Model.predict(x)
    return preds

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(

```

```

        basepath, 'uploads',
secure_filename(f.filename))
f.save(file_path)

# Make prediction
preds = model_predict(file_path, Model)

# Process your result for human

pred_class = preds.argmax(axis=-1)
# Simple argmax
#pred_class = decode_predictions(preds,
top=1)
pr = class_cancer[pred_class[0]]
result = str(pr)
return result
return None

if __name__ == '__main__':
    app.run(debug=False) # Production Environment, if
debug=True == Development Environment

```

4. *Source code* program app.py

```

from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
from PIL import Image as pil_image
import tensorflow
import keras

# Keras
from tensorflow.keras.applications.imagenet_utils
import preprocess_input, decode_predictions
from tensorflow.keras.models import Model ,
load_model
from keras.preprocessing import image

```

```

# Flask utils
from flask import Flask, redirect, url_for, request,
render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

app = Flask(__name__)

Model= load_model('best_model_cutix.h5')

class_cancer = {
    0 : 'Actinic keratoses (akiec)',
    1 : 'Basal cell carcinoma (bcc)',
    2 : 'Benign keratosis-like lesions (bkl) ',
    3 : 'Dermatofibroma (df)',
    4 : 'Melanocytic nevi (nv)',
    5 : 'Vascular lesions (vasc)',
    6 : 'Melanoma (mel)'
}

def model_predict(img_path, Model):
    img = image.load_img(img_path,
target_size=(28,28,3))

    #img =
np.asarray(pil_image.open('img').resize((120,90)))
    #x = np.asarray(img.tolist())

    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    # Be careful how your trained model deals with
the input
    # otherwise, it won't make correct prediction!
    #x = preprocess_input(x, mode='caffe')

    preds = Model.predict(x)
    return preds

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

```

```
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads',
        secure_filename(f.filename))
        f.save(file_path)

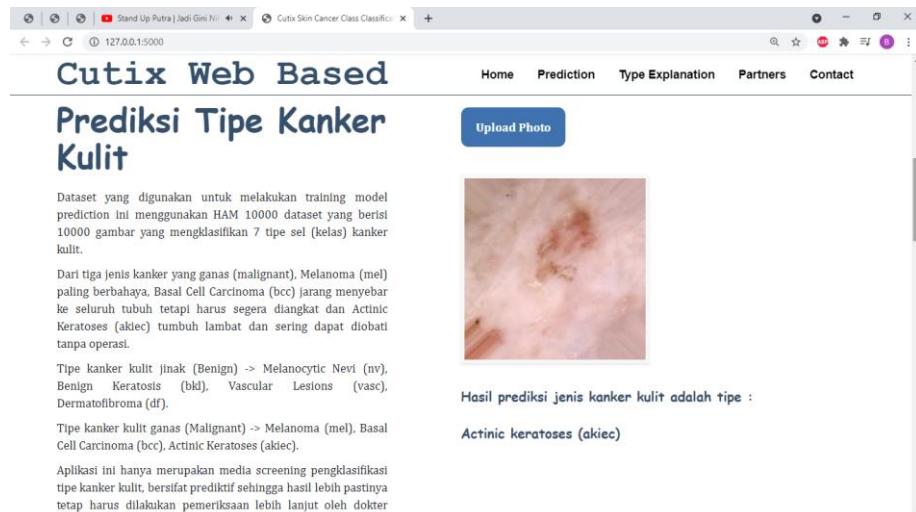
        # Make prediction
        preds = model_predict(file_path, Model)

        # Process your result for human

        pred_class = preds.argmax(axis=-1)
    # Simple argmax
        #pred_class = decode_predictions(preds,
    top=1)
        pr = class_cancer[pred_class[0]]
        result = str(pr)
        return result
    return None

if __name__ == '__main__':
    app.run(debug=False) # Production Environment, if
debug=True == Development Environment
```

Lampiran 3. Simulasi Prediksi Kanker Kulit di Website



Cutix Web Based
Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

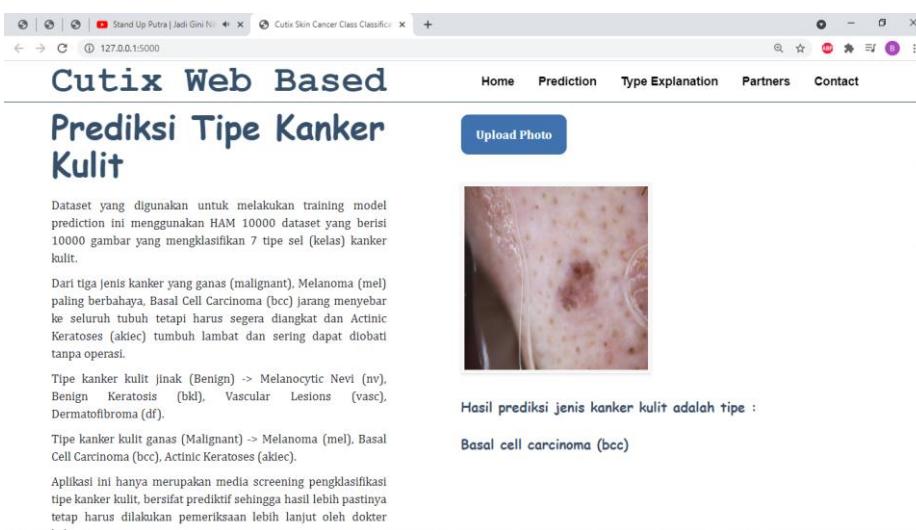
Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya, Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :
Actinic keratoses (akiec)



Cutix Web Based
Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

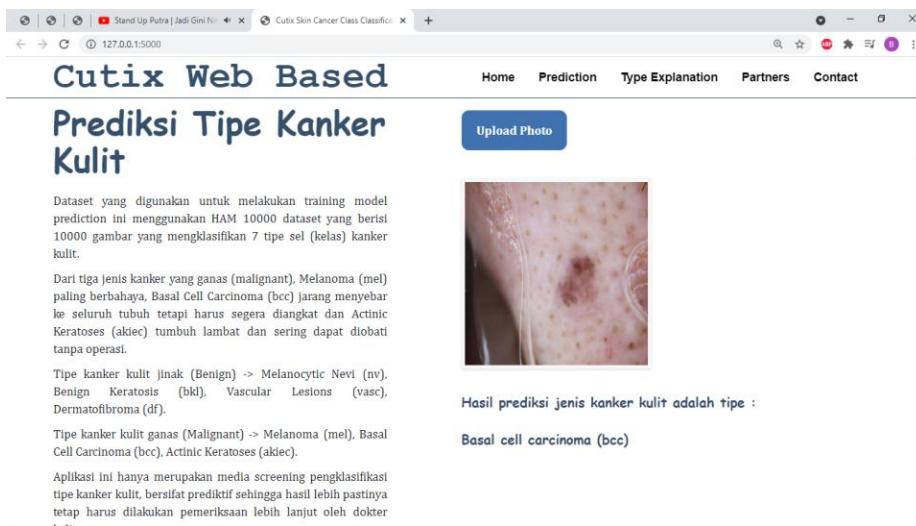
Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya, Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :
Basal cell carcinoma (bcc)



Cutix Web Based
Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya, Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :
Basal cell carcinoma (bcc)

Cutix Web Based

Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya. Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :

Dermatofibroma (df)



Cutix Web Based

Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya. Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :

Melanocytic nevi (nv)



Cutix Web Based

Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya. Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

Tipe kanker kulit jinak (Benign) -> Melanocytic Nevi (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter

Hasil prediksi jenis kanker kulit adalah tipe :

Melanoma (mel)



ALDO_Paling Males Norton < x SKIN_CANCER_CLASSIFICATION x Cutix Skin Cancer Class Classification x 127.0.0.1:5000

Cutix Web Based Prediksi Tipe Kanker Kulit

Dataset yang digunakan untuk melakukan training model prediction ini menggunakan HAM 10000 dataset yang berisi 10000 gambar yang mengklasifikasikan 7 tipe sel (kelas) kanker kulit.

Dari tiga jenis kanker yang ganas (malignant), Melanoma (mel) paling berbahaya, Basal Cell Carcinoma (bcc) jarang menyebar ke seluruh tubuh tetapi harus segera diangkat dan Actinic Keratoses (akiec) tumbuh lambat dan sering dapat diobati tanpa operasi.

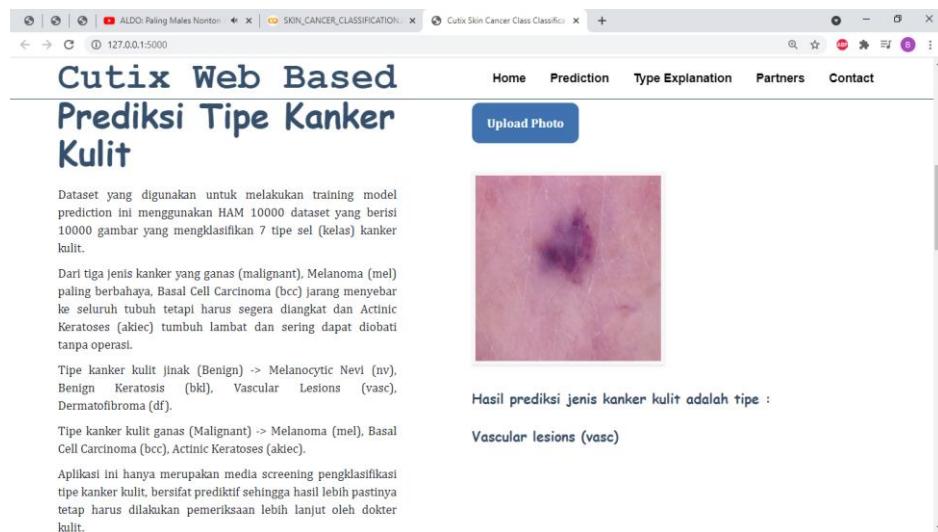
Tipe kanker kulit jinak (Benign) -> Melanocytic Nevus (nv), Benign Keratosis (bkl), Vascular Lesions (vasc), Dermatofibroma (df).

Tipe kanker kulit ganas (Malignant) -> Melanoma (mel), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec).

Aplikasi ini hanya merupakan media screening pengklasifikasi tipe kanker kulit, bersifat prediktif sehingga hasil lebih pastinya tetap harus dilakukan pemeriksaan lebih lanjut oleh dokter kulit.

Home Prediction Type Explanation Partners Contact

Upload Photo



Hasil prediksi jenis kanker kulit adalah tipe :

Vascular lesions (vasc)

BIODATA PENULIS



A. Identitas

Nama	:	Ali Rohman
NIM	:	H1A018021
Tempat, tanggal lahir	:	Bekasi, 5 Juli 2000
Alamat	:	Jalan Patuha Raya RT 04/RW 16, Kelurahan Kayuringin Jaya, Kecamatan Bekasi Selatan, Kota Bekasi, Jawa Barat 17144
No. Telp.	:	089669232481
Alamat e-mail	:	alirohman56@gmail.com

B. Riwayat Pendidikan Akademik

Periode	Jenjang	Institusi
2018 – 2021	S1	Teknik Elektro Universitas Jenderal Soedirman
2015 – 2018	SMA	SMAN 2 Bekasi
2012 – 2015	SMP	SMPN 4 Bekasi

C. Riwayat Pendidikan Non Formal (jika ada)

Tahun	Keahlian	Penyelenggara	Kota

D. Prestasi

Tahun	Tingkat	Prestasi

E. Keahlian (tuliskan secara diskriptif)

Memiliki minat di bidang pengembangan *machine learning* dan otomasi industri. Mampu merancang program *machine learning* seperti klasifikasi objek dan pemrograman PLC. Terlibat secara aktif dalam kegiatan asisten praktikum Fisika Elektro, praktikum Konsep Telekomunikasi, praktikum Rangkaian Listrik, dan praktikum Instrumentasi.