

**KLASIFIKASI STUNTING DAN STATUS GIZI BALITA DENGAN
METODE SVM (*SUPPORT VECTOR MACHINE*)**

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



Disusun Oleh:

AKMAD FAIZ FILDAN NUGROHO

32602000103

**PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2024

FINAL PROJECT

**CLASSIFICATION OF STUNTING AND NUTRITIONAL STATUS OF
TODDLERS USING SUPPORT VECTOR MACHINE (SVM) METHOD.**

*Proposed to complete the requirement to obtain a bachelor's degree (SI)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University*



**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2024**

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **"Klasifikasi Stunting Dan Status Gizi Balita Dengan Metode SVM (Support Vector Machine)"** ini disusun oleh :

Nama : Akmad Faiz Fildan Nugroho

NIM : 32602000103

Program Studi : Teknik Informatika

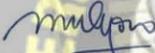
Telah disahkan oleh dosen pembimbing pada :

Hari : Senin

Tanggal : 13 Mei 2024

Mengesahkan,

Pembimbing I


Ir. Sri Mulyono .Eng
NIDN. 0626066601

Pembimbing II


Moch Taufik S.T., M.IT
NIDN. 0622037502

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung


Ir. Sri Mulyono .Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “**Klasifikasi Stunting Dan Status Gizi Balita Dengan Metode SVM (Support Vector Machine)**” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Rabu

Tanggal : 29 Mei 2024

TIM PENGUJI

Anggota I

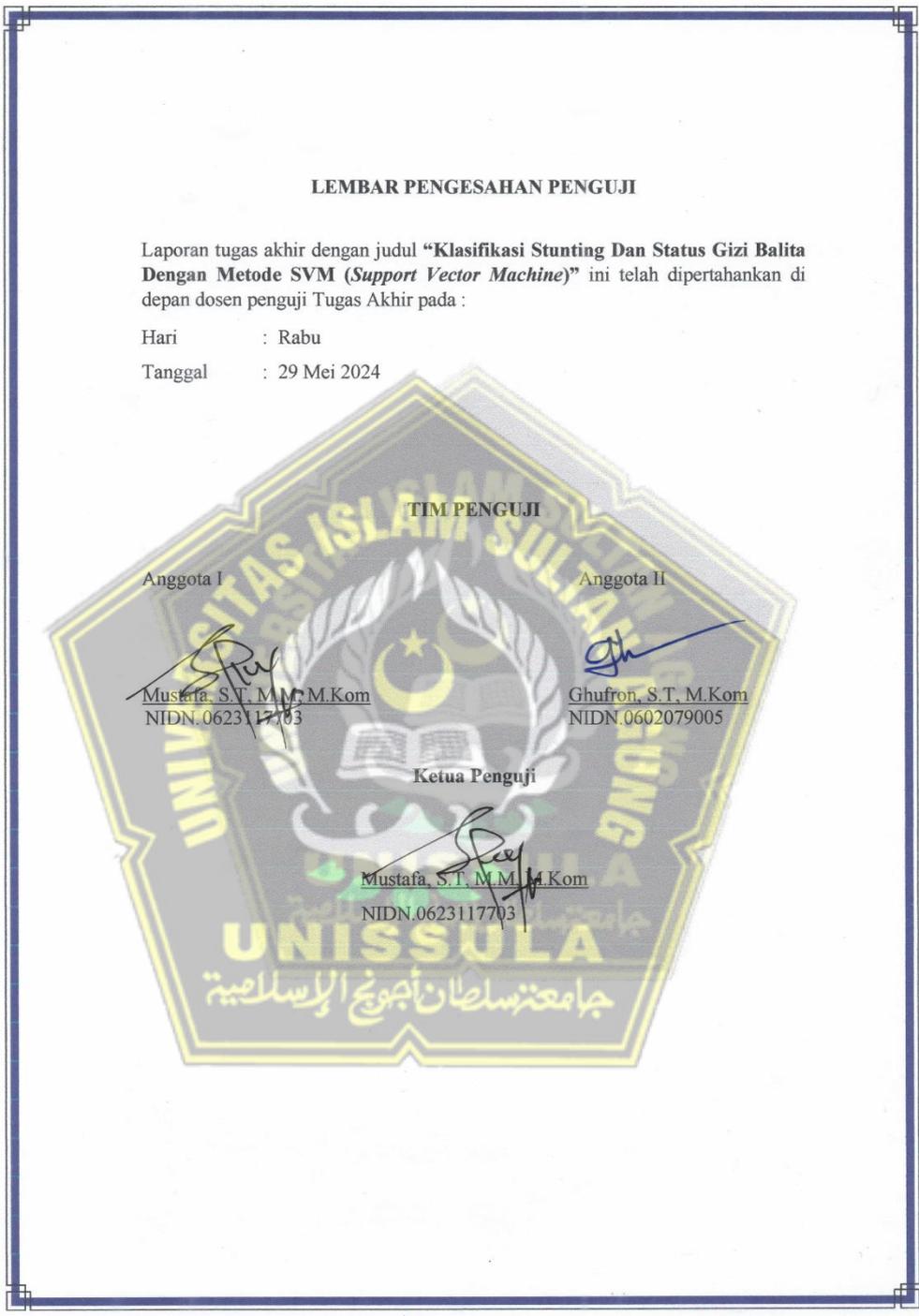

Mustafa, S.T., M.M., M.Kom
NIDN.0623117703

Anggota II


Ghufro, S.T., M.Kom
NIDN.0602079005

Ketua Penguji


Mustafa, S.T., M.M., M.Kom
NIDN.0623117703



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Akmad Faiz Fildan Nugroho

NIM : 32602000103

Judul Tugas Akhir : Klasifikasi Stunting Dan Status Gizi Balita Dengan Metode SVM (*Support Vector Machine*)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 13 Mei 2024

Yang Menyatakan,



Akmad Faiz Fildan Nugroho

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Akmad Faiz Fildan Nugroho

NIM : 32602000103

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul:

Klasifikasi Stunting Dan Status Gizi Balita Dengan Metode SVM (*Support Vector Machine*)

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantunkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 13 Mei 2024

Yang menyatakan,



Akmad Faiz Fildan Nugroho

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayahnya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul “Klasifikasi Stunting dan Status Gizi Balita Dengan Metode SVM (*Support Vector Machine*)” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis. Oleh karena itu, saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H., yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana S.T., M.T., IPU., ASEAN.Eng.
3. Dosen Pembimbing I penulis Bapak Ir. Sri Mulyono, M.Eng yang telah membimbing dan memberikan banyak nasehat dan saran.
4. Dosen Pembimbing II penulis Bapak Moch Taufik, S.T, M.IT yang telah membimbing dan memberikan banyak nasehat dan saran.
5. Orang tua dan keluarga penulis yang menjadi *support system* dan mengizinkan untuk menyelesaikan laporan ini.
6. Untuk AM yang selalu mengingatkan dan memberikan semangat kepada penulis dalam menyelesaikan laporan ini.
7. Anak-anak kontrakan AK-Basecamp dan kepada semua pihak yang tidak dapat penulis sebutkan satu persatu atas doa, *support* serta bantuannya.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 13 Mei 2024

Akmad Faiz Fildan Nugroho

DAFTAR ISI

| | |
|---|-------------|
| COVER | i |
| LEMBAR PENGESAHAN PEMBIMBING | iii |
| LEMBAR PENGESAHAN PENGUJI..... | iii |
| SURAT PERNYATAAN KEASLIAN TUGAS AKHIR..... | v |
| KATA PENGANTAR..... | vii |
| DAFTAR ISI..... | viii |
| DAFTAR GAMBAR | x |
| DAFTAR TABEL | xi |
| ABSTRAK | xii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan Penelitian..... | 4 |
| 1.5 Manfaat..... | 4 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB II TINJAUAN PUSTAKA DAN DASAR TEORI..... | 6 |
| 2.1 Tinjauan Pustaka..... | 6 |
| 2.2 Dasar Teori | 7 |
| 2.2.1 Stunting | 7 |
| 2.2.2 Standar Antropometri | 8 |
| 2.2.3 <i>Supervised Learning</i> | 10 |
| 2.2.4 <i>Support Vector Machine (SVM)</i> | 11 |
| 2.2.5 Kernel SVM | 13 |
| BAB III METODE PENELITIAN | 14 |
| 3.1 Metode Penelitian..... | 14 |
| 3.1.1 Studi Literatur | 14 |
| 3.1.2 Pengumpulan Data | 15 |
| 3.1.3 Pra-Pemrosesan Data | 17 |
| 3.1.4 Pemilihan Fitur..... | 19 |

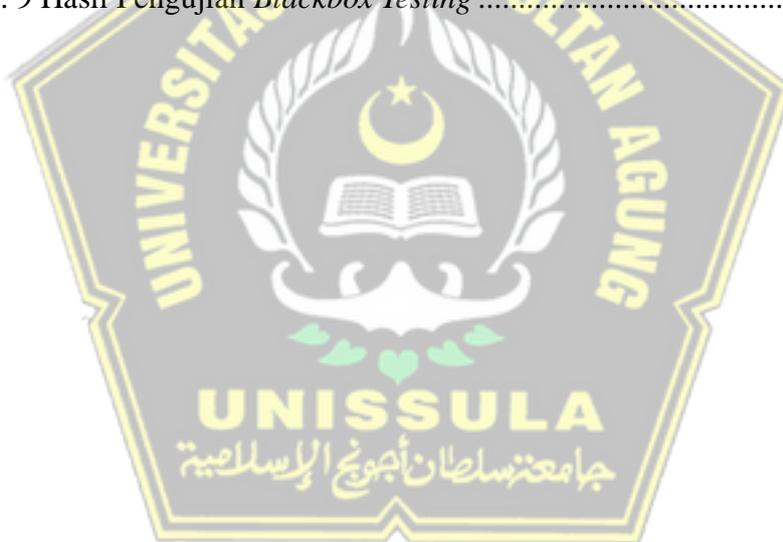
| | | |
|--|--|-----------|
| 3.1.5 | Pembangunan dan Pelatihan Model | 19 |
| 3.1.6 | Evaluasi Model..... | 24 |
| 3.1.7 | <i>Deployment</i> Model | 26 |
| 3.2 | Perancangan Sistem..... | 26 |
| BAB IV HASIL DAN ANALISIS PENELITIAN..... | | 28 |
| 4.1 | Hasil dan Analisis | 28 |
| 4.1.1 | Uji Kernel Pada Model..... | 28 |
| 4.1.2 | Pengujian Model Dengan Kernal RBF | 30 |
| 4.1.3 | Validasi Dengan <i>Statified K-Fold Cross-Validation</i> | 33 |
| 4.1.4 | Evaluasi Model Dengan <i>Confusion Matrix</i> Secara Manual..... | 36 |
| 4.2 | <i>Deployment</i> | 41 |
| 4.2.1 | Serialisasi Model..... | 42 |
| 4.2.2 | Membangun Sistem..... | 43 |
| 4.2.3 | <i>Deployment</i> Sistem..... | 44 |
| 4.3 | Pengujian Sistem | 47 |
| BAB V KESIMPULAN DAN SARAN | | 49 |
| 5.1 | Kesimpulan..... | 49 |
| 5.2 | Saran..... | 50 |
| DAFTAR PUSTAKA..... | | 51 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1. 1 Prevalensi stunting di 34 provinsi tahun 2022 | 2 |
| Gambar 2. 1 Alternatif bidang pemisah terbaik | 11 |
| Gambar 3. 1 Flowchart pembangunan model SVM..... | 14 |
| Gambar 3. 2 Dataset status gizi balita berdasarkan tinggi badan/umur | 15 |
| Gambar 3. 3 Dataset status gizi balita berdasarkan berat badan/umur | 16 |
| Gambar 3. 4 Proses split dataset | 16 |
| Gambar 3. 5 Label encoding jenis kelamin dan status gizi dataset tinggi balita .. | 18 |
| Gambar 3. 6 Label encoding jenis kelamin dan status gizi dataset tinggi balita .. | 18 |
| Gambar 3. 7 Alur klasifikasi SVM..... | 19 |
| Gambar 3. 8 Pembangunan dan training model | 21 |
| Gambar 3. 9 Visualisasi data 3D model deteksi stunting..... | 22 |
| Gambar 3. 10 Visualisasi data 2D model deteksi stunting..... | 23 |
| Gambar 3. 11 Visualisasi data 3D model deteksi status berat badan | 23 |
| Gambar 3. 12 Visualisasi data 2D model deteksi status berat badan | 24 |
| Gambar 3. 13 Flowchart Streamlit aplikasi deteksi stunting dan status gizi | 27 |
| Gambar 4. 1 Stratified k-fold cross-validation model deteksi stunting | 34 |
| Gambar 4. 2 Stratified k-fold cross-validation model deteksi berat badan..... | 34 |
| Gambar 4. 3 Hasil klasifikasi benar model deteksi stunting..... | 35 |
| Gambar 4. 4 Hasil klasifikasi salah model deteksi stunting..... | 35 |
| Gambar 4. 5 Hasil klasifikasi benar model deteksi status berat badan | 36 |
| Gambar 4. 6 Hasil klasifikasi salah model deteksi status berat badan..... | 36 |
| Gambar 4. 7 Grafik confusion matrix model deteksi stunting..... | 37 |
| Gambar 4. 8 Grafik confusion matrix model deteksi status berat badan | 39 |
| Gambar 4. 9 Serialisasi model deteksi stunting | 42 |
| Gambar 4. 10 Serialisasi model deteksi status berat badan | 42 |
| Gambar 4. 11 File hasil serialisasi model | 43 |
| Gambar 4. 12 Project sistem | 43 |
| Gambar 4. 13 Tampilan sistem..... | 44 |
| Gambar 4. 14 Repository project di Github | 45 |
| Gambar 4. 15 File requirment.txt..... | 45 |
| Gambar 4. 16 Deployment sistem pada streamlit cloud | 46 |
| Gambar 4. 17 Tampilan sistem yang telah dilakukan deployment | 46 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2. 1 Kategori dan ambang batas status gizi anak..... | 10 |
| Tabel 3. 1 Hasil split dataset tinggi balita | 17 |
| Tabel 3. 2 Hasil split dataset berat balita..... | 17 |
| Tabel 4. 1 Hasil uji kernel dengan <i>confusion matrix</i> model deteksi stunting | 28 |
| Tabel 4. 2 Hasil uji kernel dengan <i>confusion matrix</i> model deteksi status berat badan | 29 |
| Tabel 4. 3 Evaluasi model deteksi stunting dengan parameter gamma..... | 31 |
| Tabel 4. 4 Evaluasi model deteksi status berat badan dengan parameter gamma . | 32 |
| Tabel 4. 5 TP, FP, FN, dan TN dari model deteksi stunting | 37 |
| Tabel 4. 6 Perhitungan <i>precision</i> , <i>recall</i> , dan <i>f1-Score</i> model deteksi stunting. ... | 38 |
| Tabel 4. 7 TP, FP, FN, dan TN dari model deteksi status berat badan | 40 |
| Tabel 4. 8 Perhitungan <i>precision</i> , <i>recall</i> , dan <i>f1-Score</i> model deteksi status berat badan | 40 |
| Tabel 4. 9 Hasil Pengujian <i>Blackbox Testing</i> | 47 |



ABSTRAK

Stunting adalah keadaan di mana anak mengalami kekurangan asupan gizi yang cukup dalam jangka waktu yang lama, yang menyebabkan masalah dalam pertumbuhan mereka. Stunting dapat menimbulkan dampak serius pada perkembangan anak dan kualitas hidupnya di masa depan. Deteksi dini stunting sangat penting untuk dilakukan guna mencegah terjadinya kasus stunting pada balita. Dengan deteksi dini, kondisi dan kesehatan balita akan terpantau dan apabila ada balita yang menunjukkan gejala stunting dapat ditangani dengan segera sehingga kondisi balita tidak bertambah parah dan peluang untuk kembali tumbuh normal lebih besar. Dengan sistem yang memanfaatkan AI yaitu menggunakan SVM (*Support Vector Machine*) dengan kernel RBF (*Radial Basis Function*) dapat memberikan hasil yang akurat dalam mengklasifikasikan balita stunting dan status gizi balita. Pada sistem diterapkan dua model untuk mengklasifikasikan balita stunting dan status gizi balita. Model deteksi stunting konfigurasi terbaik yaitu pada percobaan ke enam yaitu dengan $C = 10$ dan $\gamma = 5$. Menghasilkan *accuracy* sebesar 99.50% dengan rata-rata nilai *precision* 0.994, *recall* sebesar 0.991, dan *f1-score* sebesar 0.994 pada keempat kelas. Model deteksi status berat badan konfigurasi terbaik yaitu pada percobaan ke lima yaitu dengan $C = 10$ dan $\gamma = 1$. Menghasilkan *accuracy* sebesar 94.92% dengan rata-rata nilai *precision* 0.912, *recall* sebesar 0.913, dan *f1-score* sebesar 0.912 pada keempat kelas.

Kata kunci: Stunting, Status gizi balita, *Support Vector Machine*, RBF

ABSTRACT

Stunting is a condition where a child experiences prolonged insufficient nutritional intake, leading to growth problems. Stunting can have serious impacts on a child's development and quality of life in the future. Early detection of stunting is crucial to prevent stunting cases in toddlers. Through early detection, the condition and health of toddlers can be monitored, and if any toddlers show symptoms of stunting, they can be promptly addressed, preventing the condition from worsening and increasing the chances of returning to normal growth. A system utilizing AI, specifically using SVM (Support Vector Machine) with an RBF (Radial Basis Function) kernel, can provide accurate results in classifying stunted toddlers and their nutritional status. The system applies two models to classify stunting and the nutritional status of toddlers. The best configuration for the stunting detection model was found in the sixth experiment with $C = 10$ and $\gamma = 5$, achieving an accuracy of 99.50%, with an average precision of 0.994, recall of 0.991, and F1-score of 0.994 across all four classes. The best configuration for the weight status detection model was found in the fifth experiment with $C = 10$ and $\gamma = 1$, achieving an accuracy of 94.92%, with an average precision of 0.912, recall of 0.913, and F1-score of 0.912 across all four classes.

Keywords: Stunting, Nutritional Status of Toddlers, Support Vector Machine, RBF

BAB I

PENDAHULUAN

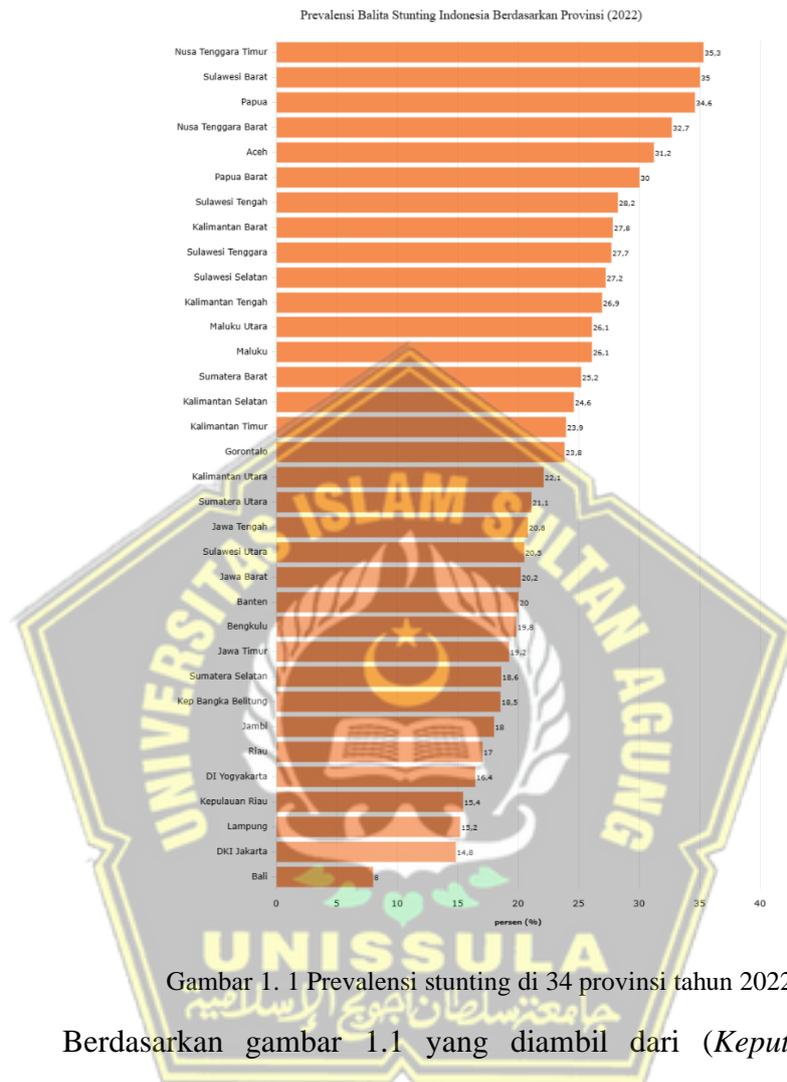
1.1 Latar Belakang

Stunting adalah salah satu masalah Kesehatan pada anak yang sering ditemui di Posyandu. Menurut (Chafidin dkk., 2022) stunting adalah kondisi di mana anak-anak mengalami kekurangan asupan gizi yang cukup dalam jangka waktu lama, yang dimana ini dapat menyebabkan masalah dan gangguan dalam pertumbuhan mereka. Dalam kasus ini, masalah pertumbuhan yang dimaksud adalah dimana tinggi badan seorang anak dibawah standar ukuran tubuh anak pada umumnya yang ditandai dengan ukuran tubuh yang lebih kecil atau kerdil.

Stunting merupakan suatu kondisi yang dapat menimbulkan dampak serius pada perkembangan anak dan kualitas hidupnya di masa depan. Data yang dihimpun Organisasi Kesehatan Dunia (WHO) mengenai prevalensi (jumlah individu yang mengalami gangguan, penyakit, atau kondisi tertentu di dalam suatu populasi) stunting pada anak di bawah usia lima tahun menunjukkan bahwa Indonesia menempati peringkat ketiga negara dengan prevalensi stunting tertinggi di wilayah Asia Tenggara (SEAR). Pada tahun 2005 hingga 2017 rata-rata angka prevalensi bayi stunting di Indonesia sangat tinggi yaitu berada di angka 36,4%.

Data Riset Kesehatan Dasar (Riskesdas) yang dilakukan oleh Kementerian Kesehatan Republik Indonesia (Keputusan Menteri Kesehatan Republik Indonesia Nomor Hk.01.07/Menkes/1928/2022 Tentang Pedoman Nasional Pelayanan Kedokteran Tata Laksana Stunting, 2022) prevalensi balita pendek dan sangat pendek di Indonesia mengalami penurunan dari 37,2% di tahun 2013 turun ke angka 30,8% pada tahun 2018. Dan berdasarkan Survei Status Gizi Indonesia (SSGI) tahun 2021 dan 2022 yang mencakup 34 provinsi, pada tahun 2021 prevalensi stunting di Indonesia berada di angka 24,4% dan kembali mengalami penurunan menjadi 21,6% pada tahun 2022. Prevalensinya menurun namun masih dalam kategori

tinggi (>20%) menurut kriteria WHO. Berikut adalah grafik daftar prevalensi stunting setiap provinsi di Indonesia pada tahun 2022.



Gambar 1. 1 Prevalensi stunting di 34 provinsi tahun 2022

Berdasarkan gambar 1.1 yang diambil dari (*Keputusan Menteri Kesehatan Republik Indonesia Nomor Hk.01.07/Menkes/1928/2022 Tentang Pedoman Nasional Pelayanan Kedokteran Tata Laksana Stunting, 2022*), dapat disimpulkan bahwa sebagian besar provinsi di Indonesia prevalensi balita stuntingnya masih cukup tinggi. Bahkan ada beberapa provinsi yang masih memiliki prevalensi diatas 30%. Meskipun ada beberapa provinsi yang memiliki prevalensi cukup rendah seperti provinsi Bali yaitu sebesar 8%. Dengan masih tingginya rata-rata prevelansi balita stunting di Indonesia, beberapa program seperti deteksi dini stunting, pemberian makanan tambahan untuk balita/ MPASI (Makanan Pendamping Asi), dan

program yang lainnya telah dilakukan pemerintah guna menurunkan tingkat prevalensi stunting di Indonesia.

Deteksi dini stunting sangat penting untuk dilakukan guna mencegah terjadinya kasus stunting pada balita. Dengan deteksi dini, kondisi dan kesehatan balita akan terpantau dan apabila ada balita yang menunjukkan gejala stunting dapat ditangani dengan segera sehingga kondisi balita tidak bertambah parah dan peluang untuk kembali tumbuh normal lebih besar.

Saat ini, pendeteksian balita stunting di beberapa posyandu masih dilakukan secara konvensional dengan melakukan perhitungan secara manual. Dengan perkembangan kemajuan teknologi dalam komputasi dan program pengolahan data memberikan kesempatan untuk dapat digunakan sebagai bahan penelitian dalam melakukan proses klasifikasi terkait penderita stunting dan status gizi balita. Metode SVM (*Support Vector Machine*) dipilih karena kemampuan generalisasinya. Pola yang tidak memuat data yang digunakan pada tahap pembelajaran metode dapat diklasifikasikan dengan fungsi generalisasi. Oleh karena itu, dalam penelitian ini, SVM (*Support Vector Machine*) digunakan sebagai metode untuk melakukan proses klasifikasi stunting dan status gizi balita.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang, maka rumusan masalah dari penelitian ini adalah bagaimana klasifikasi stunting dan status gizi balita dengan menggunakan metode SVM (*Support Vector Machine*).

1.3 Batasan Masalah

Batasan masalah yang akan dibuat berdasarkan rumusan masalah tersebut adalah sebagai berikut:

1. Variabel yang di gunakan adalah parameter antropometri (tinggi badan, berat badan), data demografis (usia, jenis kelamin).
2. Subjek penelitian adalah balita usia 0 - 5 tahun karena merupakan usia yang rentan mengalami stunting.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Mengimplementasikan metode SVM (*Support Vector Machine*) untuk klasifikasi stunting dan status gizi balita guna membantu dalam pendeteksian dini stunting.
2. Menemukan kernel dan konfigurasi terbaik untuk model deteksi stunting dan status gizi balita.

1.5 Manfaat

Berdasarkan masalah dan tujuan penelitian, dapat diambil kesimpulan bahwa penelitian ini diharapkan dapat membantu orang tua, kader posyandu, dan tenaga kesehatan dalam pendeteksian dini stunting dan status gizi balita.

1.6 Sistematika Penulisan

Sistem penulisan yang akan penulis gunakan dalam laporan akhir tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan dari penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Mencakup tinjauan pustaka yang dilakukan penulis, dan landasan teori yang dipakai untuk mendukung dalam menganalisis masalah sebagai pedoman dalam penyusunan Tugas Akhir.

BAB III : METODE PENELITIAN

Memuat tentang analisis proses dari sistem yang akan digunakan untuk klasifikasi stunting dan status gizi balita yang berisi mengenai perancangan model, perancangan sistem, dan implementasi model pada sistem.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Memuat tentang hasil klasifikasi dari penelitian yang menggunakan SVM (*Support Vector Machine*), pengimplementasian model pada sistem, serta pembahasan mengenai sistem atau prosedur – prosedur kinerja dari sistem, serta pengujian sistem.

BAB V : KESIMPULAN DAN SARAN

Pada bab V ini memuat tentang kesimpulan serta saran dari penelitian yang telah dilakukan oleh penulis.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Salah satu penelitian yang dilakukan oleh (Kusumaningrum dkk., 2020) melihat tolak ukur dari algoritma *multi-class* terhadap pengklasifikasian yang berkaitan dengan penyakit stunting. Tujuan dilakukannya penelitian ini guna mengevaluasi kinerja dari masing-masing algoritma, dengan tujuan untuk memberikan tolak ukur algoritma dalam pengolahan klasifikasi *multi-class*. Penelitian ini menggunakan 900 sampel data untuk tiga kelas dengan algoritma *regression*, *random forest*, dan *support vector machine* (SVM). Hasil penelitian menunjukkan bahwa metode SVM mampu menghasilkan hasil yang lebih baik daripada kedua algoritma lainnya yang digunakan dalam pengolahan klasifikasi *multi-class*. Pada penelitian ini, metode SVM menghasilkan nilai akurasi sebesar 98% dengan standar deviasi sebesar 0.03.

Penelitian sebelumnya yang dilakukan oleh (Nababan, 2021) hasil klasifikasi menggunakan metode *support vector machine* (SVM) Model prediksi untuk klasifikasi penderita stunting menggunakan data lima puskesmas di Kota Bandar Lampung menghasilkan rata-rata hasil presisi kernel linier sebesar 80,8%, rata-rata hasil *recall* adalah sebesar 86,2%, rata-rata hasil presisi diangka 82,8%, dan rata-rata hasil skor F1 sebesar adalah 84,6%. Kemudian rata-rata presisi kernel Gaussian sebesar 80,4%, rata-rata hasil *recall* sebesar 82%, rata-rata hasil presisi sebesar 86%, dan rata-rata skor F1 hasil hasil sebesar 83,4%. Terakhir, rata-rata hasil presisi pada kernel polinomial sebesar 69,1%, rata-rata hasil *recall* sebesar 90,3%, rata-rata hasil presisi sebesar 69,6%, dan rata-rata hasil skor F1 sebesar 78,3%.

Penelitian berjudul “Perbandingan Metode Naïve Bayes Dan *Support Vector Machine* Dalam Klasifikasi Penyakit Diabetes Melitus” yang dilakukan oleh (Apriyani, 2020). Setelah data pelatihan dan pengujian digunakan, tingkat akurasi diukur dan dievaluasi menggunakan *confusion*

matrix terhadap percobaan yang terbaik. Untuk menghasilkan nilai akurasi dengan algoritma SVM, digunakan dua jenis kernel, yaitu kernel polinomial dan kernel RBF. Akurasi kernel polinomial adalah sebesar 96.27%, yang membuatnya lebih akurat daripada algoritma Naive Bayes, yang memiliki akurasi 92.07%. Sedangkan akurasi model dengan kernel RBF adalah sebesar 80.88%.

Penelitian yang berjudul “Perbandingan Kernel *Support Vector Machine* (SVM) Dalam Penerapan Analisis Sentimen Vaksinasi Covid-19” yang dilakukan oleh (Aulia dkk., 2021). Pada penelitian tersebut, algoritma SVM dapat bekerja dan digunakan dengan baik untuk kasus klasifikasi teks. SVM digunakan untuk melakukan klasifikasi terhadap sentimen mengenai vaksinasi Covid-19 dari media sosial Twitter dengan data sebanyak 1977 data. Sentimen analisis dilakukan dengan menggunakan dan membandingkan hasil kinerja dari beberapa kernel SVM yaitu kernel *linear*, *polynomial*, RBF, dan *sigmoid*. Percobaan terhadap empat kernel SVM menunjukkan bahwa kernel *sigmoid* dan *linear* memiliki tingkat akurasi yang lebih tinggi, yaitu sebesar 87% dibandingkan dengan kernel RBF dan *polynomial* yang hanya memiliki akurasi 86%.

2.2 Dasar Teori

2.2.1 Stunting

Stunting adalah sebuah kondisi malnutrisi di mana anak mengalami masalah pertumbuhan akibat kekurangan asupan gizi dalam jangka waktu yang lama. Dampaknya adalah anak mengalami pertumbuhan fisik yang lebih kecil atau kerdil dari ukuran standar pada umumnya, terutama dalam hal tinggi badan (Chafidin dkk., 2022).

Menurut (Sutarto dkk., 2018), stunting atau perawakan pendek, atau yang juga dikenal sebagai *shortness*, adalah kondisi dimana tinggi badan (TB) yang tidak sesuai dengan usianya. Kondisi ini dianggap mengalami stunting jika *Z-indeks* dari tinggi badan menurut usia (TB/U) seseorang berada di bawah angka -2 SD, atau standar deviasi. Kurangnya asupan gizi,

baik dari segi kualitas maupun kuantitas, buruknya sistem sanitasi, serta tingginya tingkat kesakitan dapat menjadi penyebab terjadinya kasus stunting. Negara-negara dengan ekonomi yang kurang baik sering mengalami kondisi ini.

2.2.2 Standar Antropometri

Untuk menilai atau menentukan status gizi seorang anak adalah dengan menggunakan standar antropometri. Penilaian status gizi seorang anak yaitu dilakukan dengan melakukan perbandingan dari hasil pengukuran berat badan dengan panjang atau tinggi badan berdasarkan standar antropometri anak. Status gizi akan dikasifikasikan menurut indeks dari standar antropometri yang mengikuti kategori status gizi dari *WHO Child Growth Standards* untuk anak usia 0-5 tahun dan *The WHO Reference 2007* untuk anak usia 5-18 tahun.

Indeks panjang badan atau tinggi badan menurut umur (PB/U atau TB/U) merupakan gambaran dari pertumbuhan panjang atau tinggi badan anak yang didasarkan pada umurnya. Dengan indeks ini, anak-anak yang mengalami kondisi pendek (*stunted*) atau sangat pendek (*severely stunted*) yang diakibatkan karena kekurangan gizi jangka panjang atau sering sakit dapat diidentifikasi. Anak-anak yang termasuk kategori tinggi di usianya juga dapat teridentifikasi. Anak-anak dengan tinggi badan di atas normal (sangat tinggi) biasanya disebabkan karena adanya gangguan endokrin, meskipun kasus ini jarang ditemukan di Indonesia (*Peraturan Menteri Kesehatan RI Nomor 2 Tahun 2020, 2020*).

Menurut (*Peraturan Menteri Kesehatan RI Nomor 2 Tahun 2020, 2020*) Standar Antropometri Anak terdiri dari empat indeks yang menentukan panjang/tinggi badan dan berat badan anak.

1. Indeks berat badan menurut umur (BB/U)

Indeks ini digunakan dalam melakukan penilaian status gizi anak dengan kategori berat badan kurang (*underweight*) atau yang sangat kurang (*severely underweight*). Penting untuk diingat bahwa anak dengan indeks BB/U rendah bisa saja mengalami masalah dalam

pertumbuhannya. Oleh karena itu, intervensi harus dilakukan sebelum konfirmasi dengan indeks BB/PB, BB/TB, atau IMT/U.

2. Indeks panjang atau tinggi badan menurut umur (PB/U atau TB/U)

Indeks ini digunakan dalam menggambarkan pertumbuhan panjang atau tinggi anak yang didasarkan pada umurnya. Anak-anak yang pendek (*stunted*) atau sangat pendek (*severely stunted*), yang disebabkan akibat kurangnya gizi atau karena sering sakit dapat diklasifikasikan dengan penggunaan indeks ini. Anak-anak dengan kategori tinggi pada umurnya juga dapat teridentifikasi dengan indeks ini. Di Indonesia, gangguan endokrin adalah salah satu penyebab umum yang membuat anak-anak memiliki tinggi badan di atas rata-rata anak pada umumnya, atau tinggi sekali.

3. Indeks berat badan berdasarkan panjang/tinggi badan (BB/PB atau BB/TB).

Indeks ini menggambarkan berat badan anak yang dibandingkan dengan pertumbuhan panjang atau tingginya yang ditunjukkan oleh indeks BB/PB atau BB/TB. Dengan menggunakan indeks ini, dapat diidentifikasi anak yang kekurangan nutrisi (*wasted*), anak dengan status gizi buruk (*severely wasted*), serta anak yang memiliki kemungkinan besar mengalami obesitas. Adanya penyakit dan kurangnya asupan gizi dalam jangka yang lama (kronis) atau baru (akut) menjadi salah satu penyebab utama anak mengalami gizi buruk.

4. Indeks masa tubuh menurut umur (IMT/U)

Indeks ini digunakan dalam penentuan kategori gizi buruk, gizi kurang, gizi baik, berisiko gizi lebih, gizi lebih, ataupun obesitas pada anak. Indeks IMT/U dan indeks BB/PB atau BB/TB biasanya akan memiliki hasil yang sama, tetapi indeks IMT/U bersifat lebih sensitif dalam menapis anak dengan kategori lebih gizi dan obesitas. Anak dengan ambang batas IMT/U $>+1SD$ berisiko mengalami kondisi kelebihan gizi, sehingga perlu adanya perawatan tambahan untuk mencegah hal tersebut terjadi.

Tabel 2. 1 Kategori dan ambang batas status gizi anak

| Indeks | Kategori Status Gizi | Ambang Batas (Z-Score) |
|---|--|------------------------|
| Berat Badan menurut Umur (BB/U) anak usia 0 – 60 bulan | Berat badan sangat kurang (<i>Severely Underweight</i>) | < -3 |
| | Berat badan kurang (<i>Underweight</i>) | -3 SD sd < -2 SD |
| | Berat badan normal | -2 SD sd +1 SD |
| | Resiko berat badan lebih | > +1 SD |
| Panjang Badan atau Tinggi Badan menurut Umur (PB/U atau TB/U) anak usia 0 – 60 bulan | Sangat pendek (<i>Severely Stunted</i>) | < -3 |
| | Pendek (<i>Stunted</i>) | -3 SD sd < -2 SD |
| | Normal | -2 SD sd +3 SD |
| | Tinggi | > +3 SD |
| Berat Badan menurut Panjang Badan atau Tinggi Badan (BB/PB atau BB/TB) anak usia 0 – 60 bulan | Gizi buruk (<i>Severely Wasted</i>) | < -3 |
| | Gizi kurang (<i>Wasted</i>) | -3 SD sd < -2 SD |
| | Gizi Baik (Normal) | -2 SD sd +1 SD |
| | Berisiko gizi lebih (<i>Possible risk of overweight</i>) | > +1 SD sd +2 SD |
| | Gizi lebih (<i>Overweight</i>) | > +2 SD sd +3 SD |
| | Obesitas (<i>Obese</i>) | > +3 SD |
| Indeks Massa Tubuh menurut Umur (IMT/U) anak usia 0 – 60 bulan | Gizi buruk (<i>Severely Wasted</i>) | < -3 |
| | Gizi kurang (<i>Wasted</i>) | -3 SD sd < -2 SD |
| | Gizi Baik (Normal) | -2 SD sd +1 SD |
| | Berisiko gizi lebih (<i>Possible risk of overweight</i>) | > +1 SD sd +2 SD |
| | Gizi lebih (<i>Overweight</i>) | > +2 SD sd +3 SD |
| | Obesitas (<i>Obese</i>) | > +3 SD |
| Indeks Massa Tubuh menurut Umur (IMT/U) anak usia 5 – 18 tahun | Gizi buruk (<i>Severely Thinness</i>) | < -3 |
| | Gizi kurang (<i>Thinness</i>) | -3 SD sd < -2 SD |
| | Gizi baik (Normal) | -2 SD sd +1 SD |
| | Gizi lebih (<i>Overweight</i>) | > +1 SD sd +2 SD |
| | Obesitas (<i>Obese</i>) | > +2 SD |

Tabel 2.1 merupakan tabel kategori status gizi dengan menggunakan beberapa indeks yang diperoleh dari (*Peraturan Menteri Kesehatan RI Nomor 2 Tahun 2020, 2020*).

2.2.3 Supervised Learning

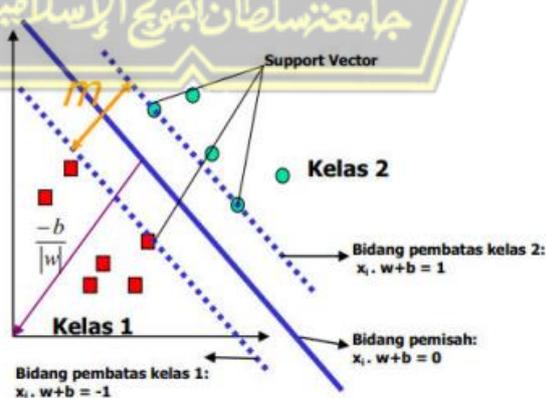
Salah satu metode pembelajaran mesin yang disebut *supervised learning* yang memanfaatkan dataset (data pembelajaran) yang telah memiliki label dalam melakukan pembelajaran pada mesin. Ini memungkinkan mesin untuk menggunakan fiturnya untuk mengidentifikasi label *input* dan kemudian melakukan prediksi dan klasifikasi. Algoritma

Naïve Bayes, *K-Nearest Neighbor (KNN)*, *Decision Tree*, *Regresi*, dan *Support Vector Machine (SVM)* adalah algoritma yang termasuk dalam teknik pembelajaran *supervised* (Retnoningsih dan Pramudita, 2020).

2.2.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah metode yang menggabungkan teori perhitungan matematis dan komputasi dari yang telah ada. Metode SVM berprinsip *Structural Risk Minimization (SRM)* dan cara kerjanya adalah dengan berusaha mencari *hyperplane* yang terbaik yang menjadi pemisah antar dua buah *class* dari ruang *input* dan ini yang membedakan SVM dengan metode *neural network* lainnya yang hanya mencoba menemukan *hyperplane* yang memisahkan antar *class*. *Classifier linear* adalah prinsip dasar dari SVM yang kemudian dikembangkan dalam penanganan masalah *non-linear* dengan penambahan konsep kernel *trick* pada ruang berdimensi yang lebih tinggi. Saat ini, SVM telah banyak digunakan untuk mengatasi permasalahan-permasalahan di dunia nyata dan umumnya solusi yang dihasilkan memiliki hasil lebih baik daripada metode konvensional seperti jaringan saraf tiruan (Nugroho dkk., 2003).

SVM digunakan untuk menghasilkan model klasifikasi seperti pada fungsi $\text{sign}(x)$, $f(x) = y$. Ini dimaksudkan untuk memungkinkan data diklasifikasikan selama proses pengujian.



Gambar 2. 1 Alternatif bidang pemisah terbaik

Gambar 2.1 menunjukkan bahwa dua bidang pembatas (*hyperplane*) sejajar dapat memisahkan dua kelas dengan bidang pembatas 1 dan 2

merupakan bidang pembatas dari masing-masing kelas. Sehingga persamaannya adalah sebagai berikut:

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \dots \dots \dots (1)$$

$$x_i \cdot w + b \geq -1 \text{ for } y_i = -1 \dots \dots \dots (2)$$

Keterangan:

w : Bidang normal

b : Posisi relatif bidang dengan koordinat pusat

SVM akan menemukan jarak terjauh dari sebuah *hyperplane* dengan kedua *class*. Penentuan jarak terjauh ini dilakukan dengan proses yang berulang sampai ditemukan *hyperplane* yang terbaik. Untuk itu optimasi dalam SVM sangat diperlukan guna menemukan jarak maksimum dari *hyperplane* dengan kedua *class* tersebut (Jumeilah, 2017).

Misalkan data yang ada di data *training* dinotasikan dengan $x_i \in R^d$ sedangkan label dari suatu kelas dinotasikan dengan $y_i \in \{-1, +1\}$. Maka, model linier SVM yang menghasilkan *hyperplane* dapat dinotasikan dengan persamaan adalah sebagai berikut (Fitriyah dkk., 2020)

$$y = w^T x_i + b, i = 1, 2, \dots, l \dots \dots \dots (3)$$

Keterangan:

x_i = $[x_1, x_2 \dots x_k]$ vector baris yang memiliki dimensi k (jumlah fitur)

$y_i \in \{-1, +1\}$ = nilai target yang berasal dari himpunan data x_i

l = jumlah dari data

w = $[w_1, w_2 \dots w_k]$ vector baris yang digunakan sebagai parameter bobot

b = bias / error

Pada SVM, jenis kernel yang biasanya umum dan sering digunakan yaitu:

1. Kernel *Linear*

$$K(x_i, x_j) = x_i^T x_j \dots \dots \dots (4)$$

2. Kernel *Polynomial*

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \dots \dots \dots (5)$$

3. Kernel RBF (*Radial Basis Function*)

$$K(x_i, x_j) = \exp\{-\gamma \|x_i - x_j\|^2\}, \gamma > 0 \dots\dots\dots(6)$$

4. Kernel Sigmoid

$$K(x, x_k) = \tanh[\gamma x_i^T x_j + r] \dots\dots\dots(7)$$

2.2.5 Kernel SVM

Kernel dalam konteks *Support Vector Machine* (SVM) merujuk pada fungsi matematis yang dipergunakan dalam mengubah data input ke dalam bentuk yang lebih dapat diolah dalam ruang fitur yang lebih tinggi. Kernel memungkinkan SVM menangani kasus di mana pemisahan secara *linear* pada data tidak dapat dilakukan dalam ruang dimensi rendah. Beberapa kernel yang umumnya digunakan dalam SVM adalah:

1. *Linear*: Jika data terpisah secara linear, kernel linear dapat menghasilkan model SVM dengan batas keputusan linear.
2. *Polynomial*: Kernel ini mengubah data ke dalam ruang dengan dimensi yang lebih tinggi menggunakan fungsi polinomial. Parameter tambahan seperti derajat polinomial dapat diatur untuk mengontrol kompleksitas model.
3. RBF (*Radial Basis Function*): Kernel ini merupakan salah satu yang paling umum digunakan. Kernel RBF bekerja dengan cara menggunakan fungsi Gaussian dalam mentransformasikan data ke dalam ruang fitur tak terbatas. Kernel RBF memiliki parameter khusus yaitu gamma yang digunakan untuk mengontrol penyebaran dari fungsi Gaussian.
4. *Sigmoid*: Untuk mengubah data ke dalam ruang fitur yang lebih besar, kernel ini menggunakan fungsi sigmoid, tetapi lebih jarang digunakan daripada kernel lainnya.

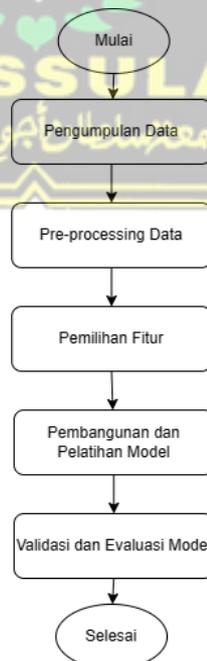
Pemilihan kernel dalam SVM sangat penting karena dapat mempengaruhi kinerja model secara signifikan, terutama ketika menangani data yang kompleks atau tidak linier. Pemilihan kernel biasanya didasarkan pada karakteristik data dan tujuan masalah klasifikasi yang ingin diselesaikan.

BAB III METODE PENELITIAN

3.1 Metode Penelitian

3.1.1 Studi Literatur

Metode studi literatur merupakan serangkaian aktivitas terkait dengan cara pengumpulan data dari sumber-sumber pustaka, meliputi membaca, mencatat, dan mengelola bahan penelitian. Dengan demikian, para peneliti dapat mengumpulkan dan menggunakan berbagai referensi pustaka yang relevan dalam bidang penelitiannya. Studi literatur memberikan pemahaman yang lebih mendalam dan luas terhadap masalah yang akan diteliti. Dalam penelitian ini, teori tentang *machine learning*, *supervised learning*, pemrograman dengan bahasa Python, dan SVM (*Support Vector Machine*) diperoleh dari berbagai penelitian sebelumnya, jurnal, artikel, dan berbagai situs web. Tujuan dari tinjauan ini adalah untuk mengetahui model arsitektur yang dapat memberikan hasil akurasi terbaik dalam menilai status gizi balita dan klasifikasi balita stunting. Untuk tahapan pembangungan model SVM pada penelitian ini dapat dilihat pada diagram alir dibawah.

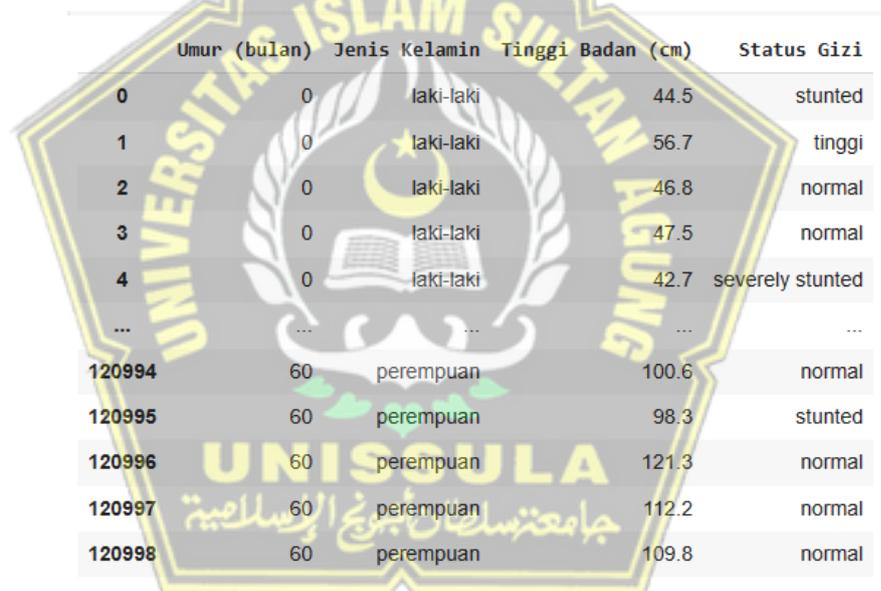


Gambar 3. 1 *Flowchart* pembangunan model SVM

3.1.2 Pengumpulan Data

Dataset yang dipergunakan untuk melakukan penelitian ini diperoleh dari *platform* Kaggle yang berupa *dataset* status gizi berdasarkan tinggi badan (cm) / umur (bulan) dan jenis kelamin. Serta *dataset* status gizi berdasarkan berat badan (kg) / umur (bulan) dan jenis kelamin. Dengan rentang usia balita adalah 0 – 60 bulan atau 0 – 5 tahun.

Kedua *dataset* tersebut kemudian disimpan ke dalam Google Drive sebagai berikut: https://bit.ly/dataset_status_gizi_balita dengan nama ‘dataset_tinggi_balita.csv’ dan ‘dataset_berat_balita.csv’. Bentuk dari *dataset* yang nantinya dipergunakan pada penelitian ini dapat dilihat pada gambar 3.2 dan gambar 3.3.



| | Umur (bulan) | Jenis Kelamin | Tinggi Badan (cm) | Status Gizi |
|--------|--------------|---------------|-------------------|------------------|
| | 0 | laki-laki | 44.5 | stunted |
| | 1 | laki-laki | 56.7 | tinggi |
| | 2 | laki-laki | 46.8 | normal |
| | 3 | laki-laki | 47.5 | normal |
| | 4 | laki-laki | 42.7 | severely stunted |
| ... | ... | ... | ... | ... |
| 120994 | 60 | perempuan | 100.6 | normal |
| 120995 | 60 | perempuan | 98.3 | stunted |
| 120996 | 60 | perempuan | 121.3 | normal |
| 120997 | 60 | perempuan | 112.2 | normal |
| 120998 | 60 | perempuan | 109.8 | normal |

Gambar 3. 2 Dataset status gizi balita berdasarkan tinggi badan/umur

Gambar 3.2 merupakan bentuk dari *dataset* ‘dataset_tinggi_balita.csv’ yang memiliki atribut umur (bulan), jenis kelamin, tinggi badan/Panjang badan (cm), dan atribut status gizi yang nantinya menjadi kelas target klasifikasi yang berisi 120.999 data.

| | Umur (bulan) | Berat Badan (kg) | Jenis Kelamin | Status Gizi | |
|--|--------------|------------------|---------------|-------------|----------------------|
| | 0 | 0 | 1 | perempuan | severely underweight |
| | 1 | 0 | 6 | perempuan | overweight |
| | 2 | 0 | 1 | perempuan | severely underweight |
| | 3 | 0 | 6 | perempuan | overweight |
| | 4 | 0 | 4 | perempuan | normal |
| | ... | ... | ... | ... | ... |
| | 122117 | 60 | 18 | laki-laki | normal |
| | 122118 | 60 | 21 | laki-laki | normal |
| | 122119 | 60 | 25 | laki-laki | overweight |
| | 122120 | 60 | 19 | laki-laki | normal |
| | 122121 | 60 | 24 | laki-laki | overweight |

Gambar 3. 3 Dataset status gizi balita berdasarkan berat badan/umur

Gambar 3.3 merupakan bentuk dari dataset ‘dataset_berat_balita.csv’ yang memiliki atribut umur (bulan), jenis kelamin, berat badan (kg), dan atribut status gizi yang nantinya menjadi kelas target klasifikasi dengan data sebanyak 122.122.

Dataset yang diperoleh kemudian dilakukan *split* data. *Split* data adalah untuk membagi dataset menjadi dua set data yaitu data *training*, dan data *testing*, dengan rasio sebesar 80% : 20%. Untuk data testing sebesar 80% dari dataset dan sisanya untuk data *training* sebesar 20% dan ini dilakukan untuk kedua dataset. Untuk proses dan *coding* untuk melakukan *split* data adalah seperti pada gambar 3.4.

Membagi dataset menjadi data training dan testing

```
[ ] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 3. 4 Proses *split* dataset

Split dilakukan dengan menggunakan fungsi *train_test_split* yang ada pada *library* Scikit-learn atau biasa disebut Sklearn. Pada fungsi *trian_test_split*, pada parameter *test_size* diisikan dengan nilai 0.2 yang artinya porsi untuk data *testing* adalah sebesar 20% dari keseluruhan data.

Setelah dilakukan *split* data, berikut adalah rincian jumlah data tiap kelas untuk kedua dataset yaitu dataset_tinggi_balita.csv dan dataset_berat-balita.csv.

Tabel 3. 1 Hasil *split dataset* tinggi balita

| Kelas | Data Training | Data Testing | Total |
|-------------------------|---------------|--------------|----------------|
| <i>Severely Stunted</i> | 15.739 | 4.130 | 19.869 |
| <i>Stunted</i> | 11.025 | 2.790 | 13.815 |
| Normal | 54.373 | 13.382 | 67.755 |
| Tinggi | 15.662 | 3.898 | 19.560 |
| Total Keseluruhan Data: | | | 120.999 |

Tabel 3.1 Merupakan hasil *split dataset* yang dilakukan pada dataset_tinggi_balita dengan perbandingan sebesar 80% dialokasikan untuk data *training* dan 20% sisanya untuk data *testing* dari total keseluruhan data.

Tabel 3. 2 Hasil *split dataset* berat balita

| Kelas | Data Training | Data Testing | Total |
|-----------------------------|---------------|--------------|----------------|
| <i>Severely Underweight</i> | 25.836 | 6.402 | 32.238 |
| <i>Underweight</i> | 6.842 | 1.700 | 8.542 |
| Normal | 27.435 | 6.928 | 34.363 |
| <i>Overweight</i> | 37.584 | 9.395 | 46.979 |
| Total Keseluruhan Data: | | | 122.122 |

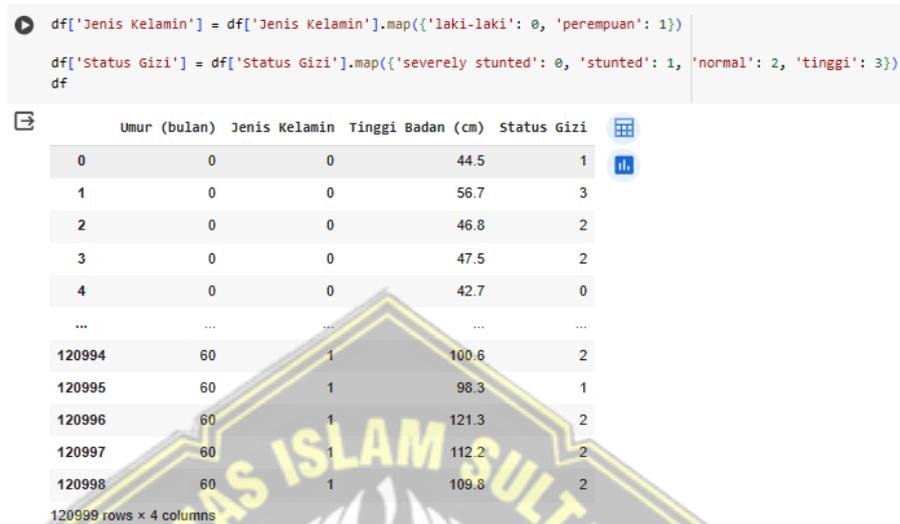
Tabel 3.2 Merupakan hasil *split dataset* yang dilakukan pada dataset_berat_balita dengan perbandingan sebesar 80% dialokasikan untuk data *training* dan 20% sisanya adalah untuk data *testing* dari total keseluruhan data.

3.1.3 Pra-Pemrosesan Data

Pada tahapan ini, dataset yang telah diperoleh akan diidentifikasi dan dilakukan pembersihan jika terdapat nilai yang hilang atau kosong, data tidak valid, *outlier*, atau kesalahan lainnya yang ada dalam data. Hal – hal tersebut dapat diatasi dengan mengisi dengan nilai yang sesuai atau dengan menghapus data tersebut. Selain itu, dilakukan juga *label encoding* dengan menggunakan fungsi *map* pada kolom jenis kelamin dan status gizi pada

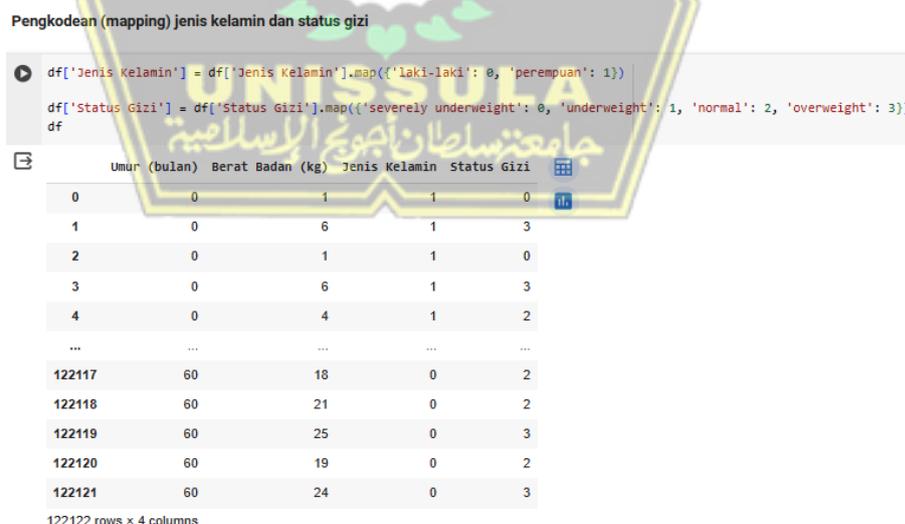
kedua dataset. Proses dan *coding label encoding* kolom jenis kelamin dan status gizi adalah sebagai berikut:

Pengkodean (mapping) jenis kelamin dan status gizi



Gambar 3. 5 Label encoding jenis kelamin dan status gizi dataset tinggi balita

Pada gambar 3.5 dilakukan *label encoding* pada kolom jenis kelamin menjadi nilai *integer* dengan laki-laki = 0, dan perempuan = 1. *Label encoding* juga dilakukan pada kolom status gizi dengan 'severely stunted' menjadi 0, 'stunted' menjadi 1, 'normal' menjadi 2, 'tinggi' menjadi 3.



Gambar 3. 6 Label encoding jenis kelamin dan status gizi dataset tinggi balita

Pada gambar 3.6 dilakukan *label encoding* pada kolom jenis kelamin menjadi nilai *integer* dengan laki-laki = 0, dan perempuan = 1. *Label*

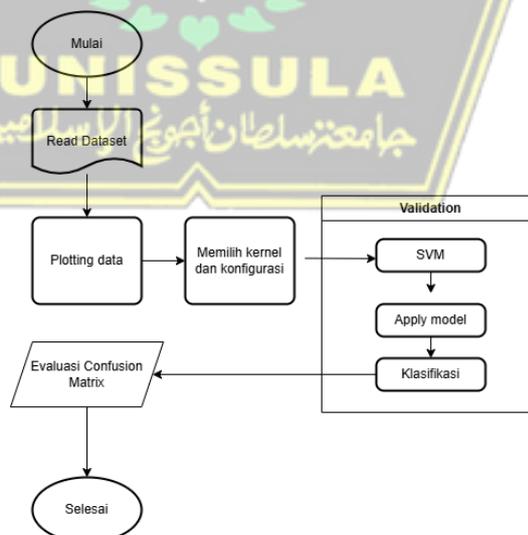
encoding juga dilakukan pada kolom status gizi dengan ‘*severely underweight*’ menjadi 0, ‘*underweight*’ menjadi 1, ‘*normal*’ menjadi 2, ‘*overweight*’ menjadi 3. *label encoding* berfungsi untuk mengkonversi label kelas menjadi representasi numerik yang diperlukan oleh model. Pada umumnya SVM memerlukan label kelas dalam bentuk nilai-nilai *integer* untuk melakukan pelatihan dan prediksi.

3.1.4 Pemilihan Fitur

Salah satu langkah penting dalam proses merancang model SVM (*Support Vector Machine*) adalah pemilihan fitur. Pemilihan fitur yang cermat membantu meningkatkan akurasi prediksi, mengurangi dimensi dataset, dan meningkatkan efisiensi komputasi. Fitur sendiri mengacu pada variabel atau atribut yang digunakan untuk memprediksi kelas atau label dari suatu data. Fitur-fitur ini memberikan informasi tentang entitas yang diamati, dan menjadi komponen penting dalam proses pembelajaran mesin.

3.1.5 Pembangunan dan Pelatihan Model

Pada tahapan ini, model dibangun dengan menggunakan *platform* Google Colab. Adapun alur dari klasifikasi dengan SVM adalah seperti pada gambar 3.7.



Gambar 3. 7 Alur klasifikasi SVM

Gambar 3.7 Merupakan alur dari proses klasifikasi menggunakan metode SVM (*Support Vector Machine*).

Untuk membangun model SVM klasifikasi stunting dan status gizi balita juga digunakan beberapa *library* dari Python diantaranya:

1. Numpy

Numpy merupakan salah satu dari *library* Python yang dipergunakan untuk komputasi numerik dalam bahasa pemrograman Python. Numpy tidak hanya menyediakan objek *array* multidimensi yang kuat, tetapi juga menyediakan berbagai fungsi yang dapat digunakan dalam melakukan berbagai operasi matematika pada *array*.

2. Pandas

Pandas adalah *library* Python yang sering dipergunakan dalam melakukan manipulasi dan analisis data. Pandas memiliki struktur data yang kuat dan mudah digunakan, terutama DataFrame, yang mirip dengan *spreadsheet* dalam Excel. Pandas juga memungkinkan untuk melakukan berbagai operasi analisis data, seperti *indexing*, *filtering*, *grouping*, dan visualisasi.

3. Scikit-learn / Sklearn

Scikit-learn juga dikenal sebagai sklearn, adalah salah satu *library* bahasa pemrograman Python yang paling banyak digunakan untuk pembelajaran mesin. Berbagai algoritma pembelajaran mesin yang efektif dan mudah digunakan disertakan dengan berbagai fungsi untuk pra-pemrosesan data, validasi model, dan evaluasi kinerja model.

4. Seaborn

Seaborn adalah salah satu *library* visualisasi data yang kuat dan populer yang dibangun di atas Matplotlib. Fungsi utama dari *library* Seaborn adalah untuk membuat visualisasi data yang menarik dan informatif dengan mudah dan cepat.

5. Matplotlib

Matplotlib adalah salah satu *library* visualisasi data yang paling populer dan kuat dalam Python. Fungsi utama dari Matplotlib adalah untuk membuat berbagai jenis plot dan visualisasi data dengan cara yang fleksibel dan kuat.

Ketika mentraining data, digunakan GPU T4 yang telah disediakan oleh Google Colab, yang memiliki memori GDDR6 16 GB dan 2.560 CUDA *core* untuk meningkatkan kinerja, yang membuatnya cocok untuk aplikasi pembelajaran mesin. Pada peelitian ini dilakukan percobaan dengan mencoba beberapa kernel SVM yang sering digunakan yaitu kernel *linear*, *polynomial*, RBF (*Radial Basis Function*), dan *sigmoid* untuk mengetahui jenis kernel mana yang cocok dengan kedua model yang memiliki akurasi paling tinggi dan mencari konfigurasi terbaik untuk model stunting dan model deteksi status berat badan. Untuk proses pembangunan dan pelatihan model adalah seperti pada gambar 3.8.



```

#Training Model dan Test Akurasi

from sklearn.svm import SVC

def train_and_evaluate_svm(kernel):
    svm_model = SVC(kernel=kernel, C=1, decision_function_shape='ovo')
    svm_model.fit(X_train, y_train)

    y_pred = svm_model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Akurasi model dengan kernel: {kernel}, Akurasi: {accuracy * 100:.2f}%")

#Coba Berbagai kernel
kernels = ['linear', 'poly', 'rbf', 'sigmoid']

for kernel in kernels:
    train_and_evaluate_svm(kernel)

```

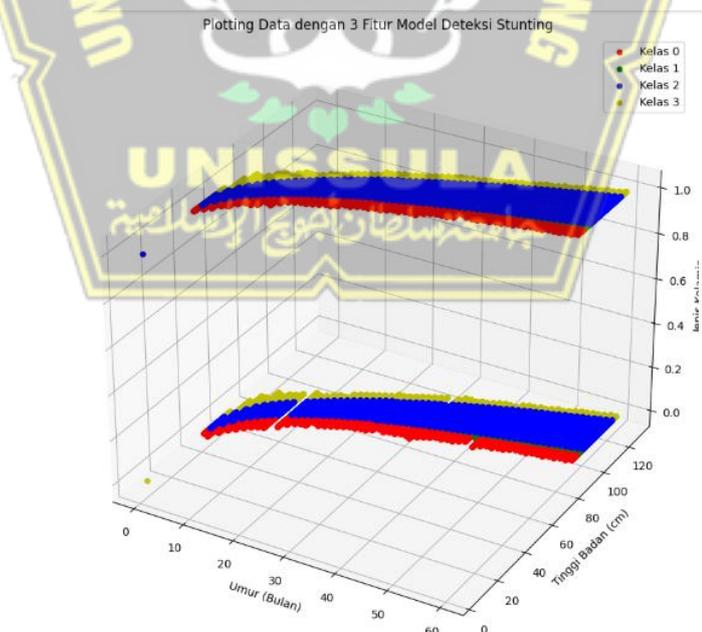
Akurasi model dengan kernel: linear, Akurasi: 78.45%
 Akurasi model dengan kernel: poly, Akurasi: 87.81%
 Akurasi model dengan kernel: rbf, Akurasi: 95.23%
 Akurasi model dengan kernel: sigmoid, Akurasi: 48.03%

Gambar 3. 8 Pembangunan dan *training* model

Pada gambar 3.8 model dilatih dengan menggunakan beberapa macam kernel dengan parameter $C = 1$, dan *decision_function_shape* = 'ovo'. Parameter C sendiri adalah parameter yang bertanggung jawab atas *trade-off* antara *margin* dan kesalahan klasifikasi. Sedangkan *decision_fuction_shape* menentukan bentuk fungsi keputusan yang dihasilkan oleh model. Parameter ini biasanya digunakan dalam SVM dengan tiga kelas atau lebih. Pendekatan OVO (*One-vs-One*) dipilih untuk mengatasi masalah klasifikasi

multikelas dengan membuat pemodelan biner untuk setiap pasangan kelas yang mungkin. Ini menunjukkan bahwa model SVM akan dilatih untuk membedakan setiap pasangan kelas dalam dataset. Karena kedua dataset masing- masing memiliki target kelas sebanyak 4, maka masing masing akan terbentuk 6 model SVM sesuai dengan rumus $\frac{n \cdot (n-1)}{2}$ dengan n adalah banyaknya kelas dalam target. Model SVM yang berbeda yang akan dibuat untuk membedakan antara pasangan kelas. Kemudian, saat melakukan prediksi, semua model ini akan digunakan untuk menghasilkan suara dan kelas yang mendapatkan suara terbanyak akan dipilih sebagai prediksi akhir. Beberapa perbandingan dengan mengubah parameter yang sesuai digunakan untuk mendapatkan hasil model yang memiliki akurasi tinggi tanpa *overfitting* atau *underfitting*.

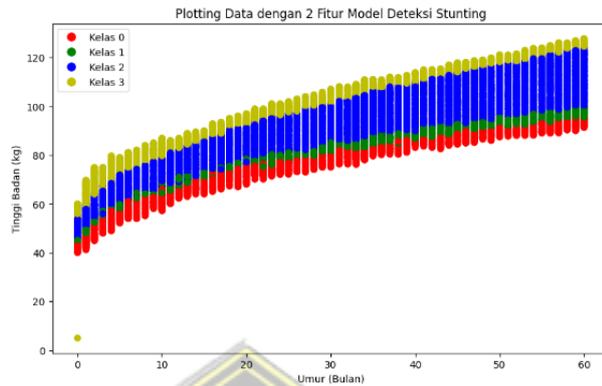
Setelah model dilatih, dilakukan visualisasi data pada dimensi yang dipilih untuk memahami distribusi dari data dan bagaimana model memisahkan data antar kelas. Pada kedua model dilakukan visualisasi data pada bidang 3D dan 2D.



Gambar 3. 9 Visualisasi data 3D model deteksi stunting

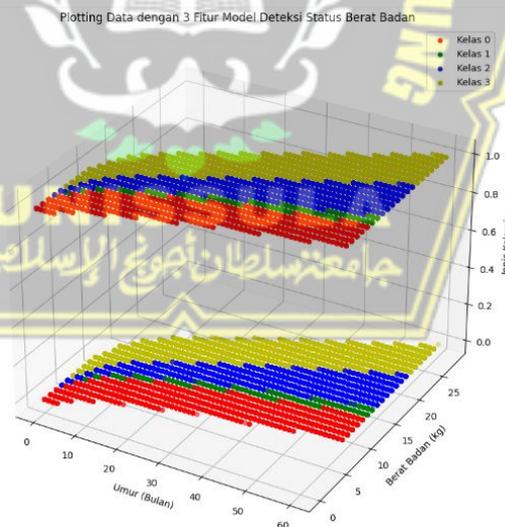
Gambar 3.9 merupakan hasil *plotting* data pada bidang 3D setelah dilakukan *training* pada model deteksi stunting. Data yang di-*plotting*-kan

merupakan data dari ketiga fitur yaitu umur (bulan), tinggi badan (cm), dan jenis kelamin.



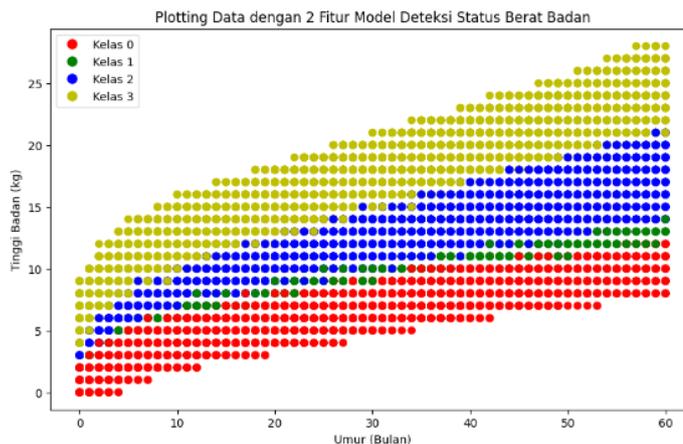
Gambar 3. 10 Visualisasi data 2D model deteksi stunting

Gambar 3.10 merupakan hasil *plotting* data pada bidang 2D setelah dilakukan *training* pada model deteksi stunting. Data yang di-*plotting*-kan merupakan data dari fitur umur (bulan) dan tinggi badan (cm) tanpa jenis kelamin. Dengan kelas 0 adalah “*severely stunted*”, kelas 1 “*stunted*”, kelas 2 “*normal*”, dan kelas 3 merupakan “*tinggi*”.



Gambar 3. 11 Visualisasi data 3D model deteksi status berat badan

Gambar 3.11 merupakan hasil *plotting* data pada bidang 3D setelah dilakukan *training* pada model deteksi status berat badan. Data yang di-*plotting*-kan merupakan data dari ketiga fitur yaitu umur (bulan), berat badan (kg), dan jenis kelamin.



Gambar 3. 12 Visualisasi data 2D model deteksi status berat badan

Gambar 3.12 merupakan hasil *plotting* data pada bidang 2D setelah dilakukan *training* pada model deteksi status berat badan. Data yang di-*plotting*-kan merupakan data dari fitur umur (bulan) dan berat badan (kg) tanpa jenis kelamin. Dengan kelas 0 adalah “*severely underweight*”, kelas 1 “*underweight*”, kelas 2 “*normal*”, dan kelas 3 merupakan “*overweight*”.

3.1.6 Evaluasi Model

Proses evaluasi model dilakukan untuk menentukan kombinasi model yang paling efektif untuk mengklasifikasikan status gizi balita dengan hasil yang akurat. Jadi, untuk membandingkan model yang dilatih sebelumnya, hasil akurasi dan *confusion matrix* digunakan. Akurasi adalah rasio yang benar dari prediksi terhadap data yang dilatih secara keseluruhan. Dalam masalah klasifikasi biner dan *multiclass*, *confusion matrix* adalah alat pengukur performa model yang sering digunakan di dalam *machine learning*. *Confusion matrix* adalah metode evaluasi yang digunakan untuk mengidentifikasi masalah klasifikasi dalam pembelajaran mesin, dengan keluaran yang terdiri dari dua atau lebih kelas. Tabel *confusion matrix* terdiri dari empat kombinasi nilai prediksi dan nilai aktual (Hovi dkk., 2022)

Baris pada *confusion matrix* menunjukkan hasil *class* sebenarnya, sedangkan kolom *confusion matrix* menunjukkan hasil *class* prediksi. Oleh karena itu, keduanya dapat digunakan untuk menghitung seluruh kemungkinan pada masalah klasifikasi. *Precision*, *recall*, dan *f1-score*

adalah tiga kategori *confusion matrix*. Untuk menunjukkan akurasi dan *confusion matrix* menggunakan persamaan rumus seperti pada (8), (9), (10), dan (11).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \dots\dots\dots(8)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(9)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(10)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \dots\dots\dots(11)$$

A. Accuracy

Accuracy adalah seberapa akurat model dapat memprediksi kelas data pengujian. Secara keseluruhan, akurasi menunjukkan seberapa baik model dapat membuat prediksi yang benar.

B. Confusion Matrix

Confusion matrix adalah tabel yang menyimpulkan seberapa sukses model dalam memprediksi data uji. Ada beberapa istilah yang mempresentasikan hasil dari model yang dilatih dalam *confusion matrix*:

- TP (*True Positives*): Jumlah kasus di mana model memprediksi positif dan data aktual juga positif. Contohnya adalah ketika model memprediksi seseorang sakit (positif) dan orang tersebut memang sakit (positif).
- TN (*True Negatives*): Jumlah kasus di mana model memprediksi negatif dan data aktual juga negative. Contohnya adalah model memprediksi seseorang tidak sakit (negatif) dan orang tersebut memang tidak sakit (negatif).
- FP (*False Positives*): Jumlah kasus di mana model memprediksi positif tetapi data aktual negatif. Contohnya adalah model memprediksi seseorang sakit (positif) tetapi orang tersebut sebenarnya tidak sakit (negatif). Ini juga dikenal sebagai "*type I error*".
- FN (*False Negatives*): Jumlah kasus di mana model memprediksi negatif tetapi data aktual positif. Contohnya adalah model

memprediksi seseorang tidak sakit (negatif) tetapi orang tersebut sebenarnya sakit (positif). Ini juga dikenal sebagai "*type II error*".

Sedangkan tiga jenis *Confusion matrix* yang digunakan dalam menghitung prediksi antara lain :

1. *Precision* adalah rasio model dengan melakukan prediksi benar dengan jumlah hasil prediksi positif dan negatif.
2. *Recall* adalah rasio model melakukan prediksi benar dengan jumlah nilai aktual positif atau negatif.
3. *F1 Score* adalah nilai perbandingan rata – rata *precision* dan *recall* yang telah dilakukan pembobotan.

3.1.7 Deployment Model

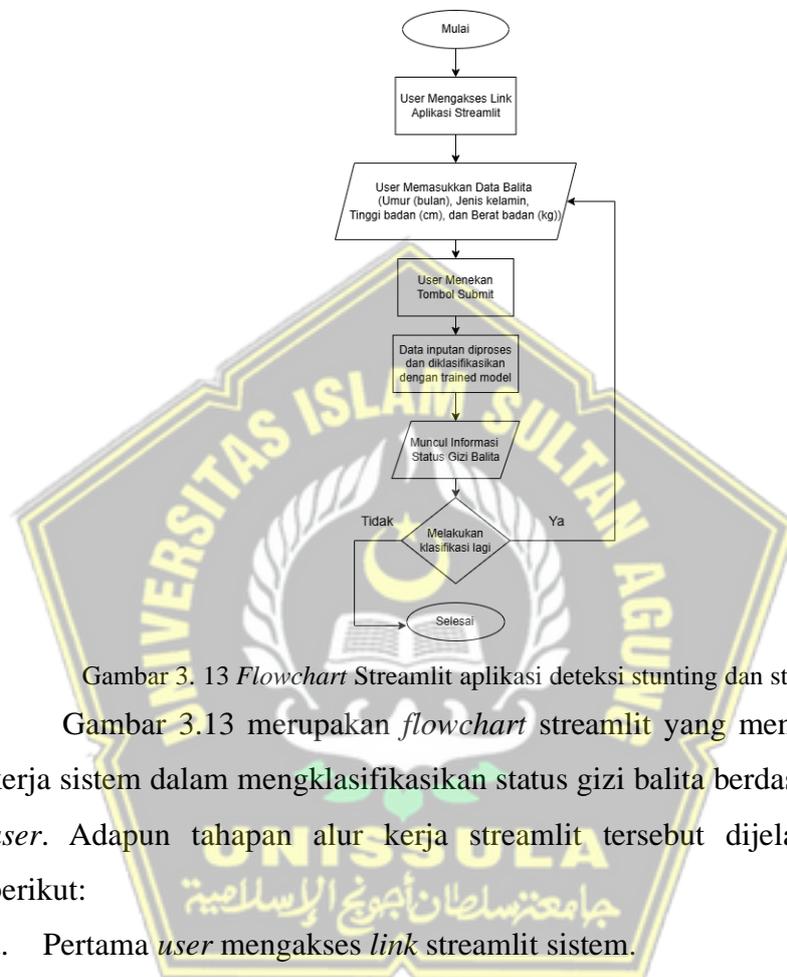
Untuk menjalankan proses *deployment* pada penelitian ini, *software* yang digunakan adalah Streamlit. Streamlit sendiri merupakan sebuah *framework* Python yang digunakan untuk membangun aplikasi web di bidang ilmu data. Streamlit memiliki berbagai fitur yang dapat digunakan untuk mengontrol aplikasi web yang akan dibuat, seperti *checkbox*, *text input*, dan *selectbox*, antara lain. Untuk penelitian ini, penggunaan Streamlit sendiri karena memiliki banyak manfaat karena termasuk *framework* yang sederhana, mudah dipelajari, *rapid prototyping*, interaktif, responsif, mudah dipasang, *open source*, dan mendukung *machine learning* (ML).

Selain itu, siklus pengembangan yang cepat, dokumentasi yang baik, fleksibilitas dan kustomisasi, *widget* dan elemen antarmuka bawaan, *update* otomatis, peningkatan dengan *library* eksternal, dan deklaratif membuatnya lebih mudah untuk mengembangkan. Oleh karena itu, penelitian ini menggunakan Streamlit yang memungkinkan pengembangan dan pengujian yang lebih mudah.

3.2 Perancangan Sistem

Pada tahap ini, peneliti akan menentukan alur kerja sistem melalui *flowchart*, yang akan memberikan gambaran tentang alur dari sistem yang akan dibuat. Tahap pertama dimulai dengan *user* mengakses *link deploy* dari

aplikasi Streamlit yang telah dibuat. Setelah itu, *user* dapat melakukan prediksi status gizi balita dengan memasukkan inputan berupa umur balita (bulan), jenis kelamin, tinggi badan (cm), dan berat badan (kg). *Flowchart* dari Streamlit dapat dilihat pada gambar 3.6.



Gambar 3. 13 *Flowchart* Streamlit aplikasi deteksi stunting dan status gizi

Gambar 3.13 merupakan *flowchart* streamlit yang menerangkan alur kerja sistem dalam mengklasifikasikan status gizi balita berdasarkan inputan *user*. Adapun tahapan alur kerja streamlit tersebut dijelaskan sebagai berikut:

- a. Pertama *user* mengakses *link* streamlit sistem.
- b. Lalu *user* memasukkan data-data balita berupa umur balita (bulan), jenis kelamin, tinggi badan (cm), dan berat badan (kg).
- c. Setelah *user* menekan tombol submit, maka data inputan akan diproses dan diklasifikasikan dengan menggunakan model yang telah dilatih sebelumnya. Kemudian akan menampilkan hasil klasifikasi status gizi balita. Apakah balita masuk kategori ‘*severely stunted*’, ‘*stunted*’, ‘*normal*’, ataukah ‘*tinggi*’. Dan status berat badannya apakah ‘*severely underweight*’, ‘*underweight*’, ‘*normal*’, ataukah ‘*overweight*’.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil dan Analisis

4.1.1 Uji Kernel Pada Model

Dilakukan uji menggunakan ke-empat kernel yaitu kernel *linear*, *polynomial*, RBF, dan *sigmoid* dengan menggunakan pendekatan OVO (*One-vs-One*) dan nilai $C = 1$ pada kedua model yaitu model deteksi stunting yang menggunakan *dataset_tinggi_balita.csv* dan model deteksi status berat badan dengan *dataset_berat_balita.csv* sehingga diperoleh hasil evaluasi model dengan menggunakan *confusion matrix* sebagai berikut:

Tabel 4. 1 Hasil uji kernel dengan *confusion matrix* model deteksi stunting

| Kernel | Class | Precision | Recall | F1 score | Accuracy |
|-------------------|-------------------------|-----------|--------|----------|-------------|
| <i>Linear</i> | <i>Severely Stunted</i> | 0.76 | 0.86 | 0.81 | 0.78 |
| | <i>Stunted</i> | 0.63 | 0.32 | 0.42 | |
| | Normal | 0.80 | 0.88 | 0.84 | |
| | Tinggi | 0.79 | 0.69 | 0.74 | |
| <i>Polynomial</i> | <i>Severely Stunted</i> | 0.82 | 0.93 | 0.87 | 0.88 |
| | <i>Stunted</i> | 0.80 | 0.64 | 0.71 | |
| | Normal | 0.91 | 0.93 | 0.92 | |
| | Tinggi | 0.89 | 0.83 | 0.86 | |
| RBF | <i>Severely Stunted</i> | 0.93 | 0.96 | 0.94 | 0.95 |
| | <i>Stunted</i> | 0.87 | 0.81 | 0.84 | |
| | Normal | 0.97 | 0.98 | 0.97 | |
| | Tinggi | 0.98 | 0.96 | 0.97 | |
| <i>Sigmoid</i> | <i>Severely Stunted</i> | 1.00 | 0.01 | 0.02 | 0.48 |
| | <i>Stunted</i> | 0.00 | 0.00 | 0.00 | |
| | Normal | 0.64 | 0.79 | 0.71 | |
| | Tinggi | 0.13 | 0.27 | 0.18 | |

Berdasarkan hasil evaluasi model deteksi stunting pada tabel 4.1 yang telah dilakukan dengan menggunakan beberapa macam kernel disimpulkan bahwa kernel yang paling cocok dipakai untuk model deteksi stunting adalah dengan menggunakan kernel RBF. Kernel RBF memiliki tingkat akurasi paling tinggi dibandingkan dengan kernel yang lain yaitu sebesar 95%. Dengan rata-rata *precision* sebesar 0.94, *recall* sebesar 0.93, dan *f1-score* sebesar 0.95 untuk keempat kelas.

Tabel 4. 2 Hasil uji kernel dengan *confusion matrix* model deteksi status berat badan

| Kernel | Class | Precision | Recall | F1 score | Accuracy |
|------------|----------------------|-----------|--------|----------|-------------|
| Linear | Severely Underweight | 0.93 | 0.96 | 0.95 | 0.91 |
| | Underweight | 0.70 | 0.63 | 0.66 | |
| | Normal | 0.89 | 0.89 | 0.89 | |
| | Overweight | 0.96 | 0.95 | 0.95 | |
| Polynomial | Severely Underweight | 0.79 | 0.98 | 0.87 | 0.86 |
| | Underweight | 0.64 | 0.24 | 0.35 | |
| | Normal | 0.84 | 0.84 | 0.84 | |
| | Overweight | 0.96 | 0.91 | 0.94 | |
| RBF | Severely Underweight | 0.96 | 0.97 | 0.96 | 0.94 |
| | Underweight | 0.77 | 0.72 | 0.74 | |
| | Normal | 0.92 | 0.94 | 0.93 | |
| | Overweight | 0.98 | 0.97 | 0.97 | |
| Sigmoid | Severely Underweight | 0.38 | 0.58 | 0.46 | 0.40 |
| | Underweight | 0.00 | 0.00 | 0.00 | |
| | Normal | 0.32 | 0.45 | 0.38 | |
| | Overweight | 0.58 | 0.31 | 0.41 | |

Berdasarkan hasil evaluasi model deteksi status berat badan pada tabel 4.2 yang telah dilakukan dengan menggunakan beberapa macam kernel disimpulkan bahwa kernel yang paling cocok dipakai untuk model deteksi status berat badan adalah dengan menggunakan kernel RBF. Kernel RBF memiliki tingkat akurasi paling tinggi dibandingkan kernel yang lain yaitu sebesar 94%. Dengan rata-rata *precision* sebesar 0.90, *recall* sebesar 0.90, dan *f1-score* sebesar 0.90 untuk keempat kelas.

Dari kedua hasil evaluasi uji kernel dengan *confusion matrix* pada model deteksi stunting dan deteksi status berat badan dengan parameter $C = 1$ dan dengan pendekatan OVO (*One-vs-One*) didapatkan bahwa kernel yang paling cocok dipakai dan memiliki akurasi paling tinggi untuk kedua model adalah kernel RBF. Selain itu, hasil dari *confusion matrix* dengan menggunakan kernel RBF nilai *precision*, *recall*, dan *f1-score* nya cenderung lebih stabil antara kelas yang satu dengan kelas yang lain.

4.1.2 Pengujian Model Dengan Kernal RBF

Untuk memaksimalkan kinerja model dengan menggunakan kernel RBF, dilakukan pengujian dengan menambahkan parameter uji gamma (γ). Parameter gamma berfungsi untuk mengatur luasnya dampak setiap titik data terhadap pembentukan batas keputusan. Nilai gamma yang tinggi menunjukkan jangkauan pengaruh yang lebih terlokalisasi, sedangkan nilai gamma yang rendah menunjukkan jangkauan pengaruh yang lebih global. Selain menggunakan parameter gamma juga digunakan parameter C dan pendekatan OVO untuk melakukan evaluasi dan pemaksimalan kinerja model dengan menggunakan kernel RBF.

Tabel 4. 3 Evaluasi model deteksi stunting dengan parameter gamma

| C | γ | Class | Precision | Recall | F1- score | Accuracy |
|------|----------|-------------------------|-----------|--------|-----------|------------------|
| 0.01 | 1 | <i>Severely Stunted</i> | 0.85 | 0.98 | 0.91 | 0.92 (92.36%) |
| | | <i>Stunted</i> | 0.95 | 0.61 | 0.74 | |
| | | Normal | 0.93 | 0.98 | 0.95 | |
| | | <i>Tinggi</i> | 0.98 | 0.90 | 0.94 | |
| 0.1 | 1 | <i>Severely Stunted</i> | 0.95 | 0.99 | 0.97 | 0.97 (97.13%) |
| | | <i>Stunted</i> | 0.97 | 0.88 | 0.92 | |
| | | Normal | 0.98 | 0.99 | 0.98 | |
| | | <i>Tinggi</i> | 0.98 | 0.96 | 0.97 | |
| 1 | 1 | <i>Severely Stunted</i> | 0.98 | 0.99 | 0.99 | 0.99 (98.80%) |
| | | <i>Stunted</i> | 0.98 | 0.95 | 0.96 | |
| | | Normal | 0.99 | 1.00 | 0.99 | |
| | | <i>Tinggi</i> | 0.99 | 0.99 | 0.99 | |
| 1 | 5 | <i>Severely Stunted</i> | 0.99 | 1.00 | 0.99 | 0.99 (99.12%) |
| | | <i>Stunted</i> | 0.98 | 0.97 | 0.97 | |
| | | Normal | 0.99 | 1.00 | 0.99 | |
| | | <i>Tinggi</i> | 0.99 | 0.99 | 0.99 | |
| 10 | 1 | <i>Severely Stunted</i> | 0.99 | 0.99 | 0.99 | 0.99 (99.31%) |
| | | <i>Stunted</i> | 0.98 | 0.98 | 0.98 | |
| | | Normal | 1.00 | 1.00 | 1.00 | |
| | | <i>Tinggi</i> | 0.99 | 0.99 | 0.99 | |
| 10 | 5 | <i>Severely Stunted</i> | 0.99 | 1.00 | 1.00 | 0.99 (99.50%) |
| | | <i>Stunted</i> | 0.99 | 0.98 | 0.99 | |
| | | Normal | 1.00 | 1.00 | 1.00 | |
| | | <i>Tinggi</i> | 1.00 | 0.99 | 0.99 | |

Berdasarkan tabel 4.3 hasil evaluasi model dengan penambahan parameter γ (gamma) dapat meningkatkan akurasi model dengan melakukan beberapa kali percobaan dengan nilai C dan γ yang berbeda. Untuk konfigurasi menggunakan kernel RBF yang menghasilkan akurasi

paling tinggi yaitu dengan parameter nilai $C = 10$ dan nilai $\gamma = 5$ dapat meningkatkan akurasi model dari yang sebelumnya sebesar 94% seperti pada tabel 4.1 meningkat menjadi 99.50%.

Tabel 4. 4 Evaluasi model deteksi status berat badan dengan parameter gamma

| C | γ | Class | Precision | Recall | F1- score | Accuracy |
|------|----------|-------------|-----------|--------|-----------|------------------|
| 0.01 | 1 | Severely | 0.91 | 0.98 | 0.95 | 0.92 (92.48%) |
| | | Underweight | | | | |
| | | Underweight | 0.84 | 0.41 | 0.55 | |
| | | Normal | 0.88 | 0.94 | 0.91 | |
| | | Overweight | 0.97 | 0.97 | 0.97 | |
| 0.1 | 1 | Severely | 0.95 | 0.97 | 0.96 | 0.94 (94.47%) |
| | | Underweight | | | | |
| | | Underweight | 0.79 | 0.71 | 0.75 | |
| | | Normal | 0.93 | 0.94 | 0.93 | |
| | | Overweight | 0.98 | 0.97 | 0.97 | |
| 1 | 1 | Severely | 0.96 | 0.97 | 0.97 | 0.95 (94.85%) |
| | | Underweight | | | | |
| | | Underweight | 0.78 | 0.74 | 0.76 | |
| | | Normal | 0.94 | 0.94 | 0.94 | |
| | | Overweight | 0.98 | 0.97 | 0.98 | |
| 1 | 5 | Severely | 0.97 | 0.97 | 0.97 | 0.95 (94.77%) |
| | | Underweight | | | | |
| | | Underweight | 0.76 | 0.75 | 0.76 | |
| | | Normal | 0.93 | 0.94 | 0.94 | |
| | | Overweight | 0.98 | 0.97 | 0.98 | |
| 10 | 1 | Severely | 0.97 | 0.97 | 0.97 | 0.95 (94.92%) |
| | | Underweight | | | | |
| | | Underweight | 0.76 | 0.77 | 0.77 | |
| | | Normal | 0.94 | 0.94 | 0.94 | |
| | | Overweight | 0.98 | 0.98 | 0.98 | |

| | | | | | | |
|----|---|--------------------|------|------|------|----------|
| | | <i>Severely</i> | 0.97 | 0.97 | 0.97 | |
| | | <i>Underweight</i> | | | | 0.95 |
| 10 | 5 | <i>Underweight</i> | 0.77 | 0.77 | 0.77 | (94.88%) |
| | | <i>Normal</i> | 0.94 | 0.93 | 0.94 | |
| | | <i>Overweight</i> | 0.97 | 0.98 | 0.98 | |

Berdasarkan tabel 4.4 hasil evaluasi model deteksi status berat badan dengan penambahan parameter γ (gamma) dapat meningkatkan akurasi model dengan melakukan beberapa kali percobaan dengan nilai C dan γ yang berbeda. Untuk itu disimpulkan bahwa konfigurasi parameter C dan γ yang menghasilkan akurasi model terbaik untuk model deteksi status berat badan adalah pada konfigurasi ke lima yaitu dengan nilai C = 10 dan $\gamma = 1$ yang menghasilkan akurasi sebesar 94.92%.

4.1.3 Validasi Dengan *Stratified K-Fold Cross-Validation*

Dilakukan validasi pada kedua model yaitu model deteksi stunting dan model deteksi status berat badan dengan menggunakan *Stratified K-Fold Cross-Validation*. *Stratified K-Fold Cross-Validation* bekerja dengan memastikan distribusi kelas pada keseluruhan dataset serupa dengan distribusi kelas pada setiap *fold*. Hal ini diperlukan karena SVM biasanya memerlukan penyesuaian parameter (seperti C dan gamma) yang dapat dioptimalkan dengan *cross-validation*. Dengan *Stratified K-Fold Cross-Validation*, setiap kelas dalam dataset akan diwakili dengan proporsi yang seimbang di setiap *fold*, sehingga memberikan perkiraan yang lebih stabil. Berikut adalah hasil validasi kedua model dengan menggunakan *Stratified K-Fold Cross-Validation*.

```

Hasil Stratified K-Fold Cross Validation Model Deteksi Stunting

Fold 1 Accuracy: 99.60%
Fold 2 Accuracy: 99.64%
Fold 3 Accuracy: 99.33%
Fold 4 Accuracy: 99.59%
Fold 5 Accuracy: 99.50%
Fold 6 Accuracy: 99.46%
Fold 7 Accuracy: 99.49%
Fold 8 Accuracy: 99.39%
Fold 9 Accuracy: 99.55%
Fold 10 Accuracy: 99.49%

Rata-rata accuracy: 99.50%

```

Gambar 4. 1 *Stratified k-fold cross-validation* model deteksi stunting

Pada gambar 4.1 merupakan hasil validasi model deteksi stunting dengan konfigurasi terbaik ($C = 10$, $\text{gamma} = 5$) menggunakan *Stratified K-Fold Cross-Validation*. Dalam kasus ini menggunakan 10 folds, yang berarti data dibagi menjadi 10 bagian yang masing-masing digunakan sebagai data uji satu kali, sementara bagian lainnya digunakan sebagai data latih. Hasil akurasi yang diperoleh dari setiap *fold* sangat konsisten, dengan nilai akurasi berkisar antara 99.33% hingga 99.64%. Ini menunjukkan bahwa model memiliki performa yang stabil pada berbagai subset data dan juga didapat rata-rata untuk 10 *fold* adalah dengan akurasi sebesar 99.50%.

```

Hasil Stratified K-Fold Cross Validation Model Deteksi Status Berat Badan

Fold 1 Accuracy: 94.64%
Fold 2 Accuracy: 94.70%
Fold 3 Accuracy: 95.01%
Fold 4 Accuracy: 94.80%
Fold 5 Accuracy: 94.47%
Fold 6 Accuracy: 94.81%
Fold 7 Accuracy: 94.89%
Fold 8 Accuracy: 95.44%
Fold 9 Accuracy: 95.06%
Fold 10 Accuracy: 94.77%

Rata-rata accuracy: 94.86%

```

Gambar 4. 2 *Stratified k-fold cross-validation* model deteksi berat badan

Pada gambar 4.2 merupakan hasil validasi model deteksi status berat badan dengan konfigurasi terbaik ($C = 10$, $\text{gamma} = 1$) menggunakan *Stratified K-Fold Cross-Validation*. Dalam kasus ini adalah menggunakan 10 *folds*. Hasil akurasi yang diperoleh berkisar antara 94.47% hingga 95.44%

dan juga didapat rata-rata untuk 10 *fold* adalah dengan akurasi sebesar 94.86%.

Hasil klasifikasi pada model deteksi tunting dan model deteksi status berat badan adalah sebagai berikut:

Data yang Benar klasifikasi Model Deteksi Stunting

| Indeks | Data Testing | Hasil Prediksi |
|--------|--------------|----------------|
| 25283 | 2 | 2 |
| 69608 | 2 | 2 |
| 120963 | 2 | 2 |
| 43133 | 2 | 2 |
| 11811 | 0 | 0 |
| 104915 | 2 | 2 |
| 33165 | 3 | 3 |
| 92442 | 3 | 3 |
| 117138 | 3 | 3 |
| 118280 | 2 | 2 |

Gambar 4. 3 Hasil klasifikasi benar model deteksi stunting

Data yang Salah Klasifikasi Model Deteksi Stunting

| Indeks | Data Testing | Hasil Prediksi |
|--------|--------------|----------------|
| 116430 | 3 | 2 |
| 41569 | 1 | 0 |
| 27036 | 1 | 0 |
| 43447 | 1 | 2 |
| 48813 | 1 | 2 |
| 44904 | 3 | 2 |
| 95052 | 2 | 1 |
| 49191 | 1 | 0 |
| 60276 | 0 | 1 |
| 930 | 1 | 0 |

Gambar 4. 4 Hasil klasifikasi salah model deteksi stunting

Gambar 4.3 merupakan contoh sampel data *testing* yang diklasifikasikan benar oleh model deteksi stunting. Sedangkan gambar 4.4 merupakan contoh sampel data *testing* yang diklasifikasikan salah oleh model deteksi stunting.

Data yang Benar klasifikasi Model Deteksi Status Berat Badan

| Indeks | Data Testing | Hasil Prediksi |
|--------|--------------|----------------|
| 54981 | 2 | 2 |
| 39862 | 3 | 3 |
| 13289 | 2 | 2 |
| 21321 | 3 | 3 |
| 67025 | 1 | 1 |
| 101295 | 3 | 3 |
| 71383 | 3 | 3 |
| 57312 | 3 | 3 |
| 60879 | 2 | 2 |
| 74068 | 3 | 3 |

Gambar 4. 5 Hasil klasifikasi benar model deteksi status berat badan

Data yang Salah Klasifikasi Model Deteksi Status Berat Badan

| Indeks | Data Testing | Hasil Prediksi |
|--------|--------------|----------------|
| 41358 | 2 | 3 |
| 97546 | 1 | 0 |
| 81633 | 2 | 1 |
| 12562 | 3 | 2 |
| 11472 | 2 | 3 |
| 100966 | 2 | 3 |
| 64733 | 1 | 2 |
| 8005 | 1 | 0 |
| 94405 | 0 | 1 |
| 7350 | 1 | 2 |

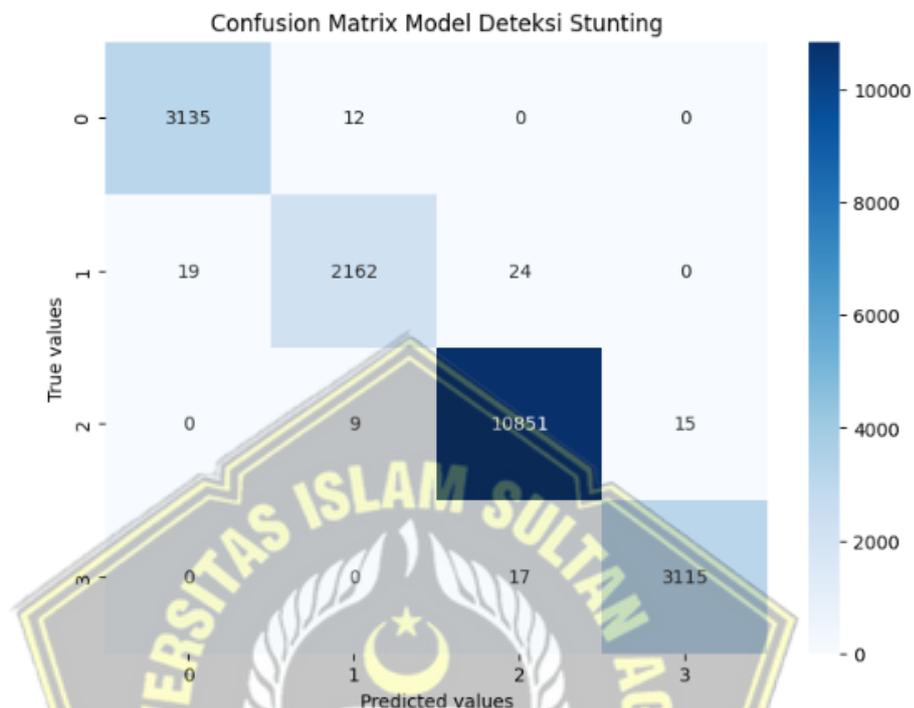
Gambar 4. 6 Hasil klasifikasi salah model deteksi status berat badan

Gambar 4.5 merupakan contoh sampel data *testing* yang diklasifikasikan benar oleh model deteksi status berat badan. Sedangkan gambar 4.6 merupakan contoh sampel data *testing* yang diklasifikasikan salah oleh model deteksi status berat badan.

4.1.4 Evaluasi Model Dengan *Confusion Matrix* Secara Manual

Setelah menemukan konfigurasi terbaik pada setiap model. Dilakukan lagi evaluasi model dengan *confusion matrix* tetapi dengan perhitungan secara manual untuk memastikan kesesuaian dengan hasil *confusion matrix* secara otomatis. Dengan melakukan perhitungan ini, berbagai metrik evaluasi dapat diterapkan, termasuk *accuracy*, *presision*, *recall*, dan *F1-*

score. Keakuratan model dapat dinilai dengan membandingkan prediksi positif dan negatif yang benar dengan keseluruhan populasi sampel.



Gambar 4. 7 Grafik *confusion matrix* model deteksi stunting

Gambar 4.7 menunjukkan hasil grafik *confusion matrix* pada model deteksi stunting dengan konfigurasi $C = 10$ dan $\gamma = 5$. Untuk memudahkan analisis dan perhitungan maka dibuat tabel seperti dibawah. Dengan TP adalah nilai *True Positif*, FP (*False Positif*), FN (*False Negatif*), dan TN (*True Negatif*).

Tabel 4. 5 TP, FP, FN, dan TN dari model deteksi stunting

| Kelas | TP | FP | FN | TN |
|-------------------------|--------|----|----|--------|
| <i>Severely Stunted</i> | 3.135 | 19 | 12 | 16.186 |
| <i>Stunted</i> | 2.162 | 21 | 43 | 17.155 |
| Normal | 10.851 | 41 | 24 | 8.426 |
| Tinggi | 3.115 | 15 | 17 | 16.214 |

*Keterangan:

- a. SS = *Severely Stunted*
- b. S = *Stunted*

c. N = Normal

d. T = Tinggi

Untuk percobaan perhitungan *confusion matrix* pada model deteksi stunting secara manual dapat dilihat pada tabel 4.6.

Tabel 4. 6 Perhitungan *precision*, *recall*, dan *f1-Score* model deteksi stunting.

| | Perhitungan | Hasil |
|---|---|-------|
| Precision (P) = $\frac{TP}{TP + FP}$ | | |
| P_{SS} | $= \frac{TP_{SS}}{TP_{SS} + FP_{SS}} = \frac{3.135}{3.135 + 19} = \frac{3.135}{3.154} = 0.993$ | 0.99 |
| P_S | $= \frac{TP_S}{TP_S + FP_S} = \frac{2.162}{2.162 + 10} = \frac{2.162}{2.172} = 0.995$ | 1.00 |
| P_N | $= \frac{TP_N}{TP_N + FP_N} = \frac{10.851}{10.851 + 41} = \frac{10.851}{10.892} = 0.996$ | 1.00 |
| P_T | $= \frac{TP_T}{TP_T + FP_T} = \frac{3.115}{3.115 + 15} = \frac{3.115}{3.130} = 0.995$ | 1.00 |
| Recall (R) = $\frac{TP}{TP + FN}$ | | |
| R_{SS} | $= \frac{TP_{SS}}{TP_{SS} + FN_{SS}} = \frac{3.135}{3.135 + 12} = \frac{3.135}{3.147} = 0.996$ | 1.00 |
| R_S | $= \frac{TP_S}{TP_S + FN_S} = \frac{2.162}{2.162 + 43} = \frac{2.162}{2.205} = 0.980$ | 0.98 |
| R_N | $= \frac{TP_N}{TP_N + FN_N} = \frac{10.851}{10.851 + 24} = \frac{10.851}{10.875} = 0.997$ | 1.0 |
| R_T | $= \frac{TP_T}{TP_T + FN_T} = \frac{3.115}{3.115 + 17} = \frac{3.115}{3.132} = 0.994$ | 0.99 |
| F1 – Score = $\frac{2 * Precision * Recall}{Precision + Recall}$ | | |
| $F1_{SS}$ | $= \frac{2 * P_{SS} * R_{SS}}{P_{SS} + R_{SS}} = \frac{2 * 0.99 * 1.00}{0.99 + 1.00} = \frac{1.98}{1.99} = 0.994$ | 0.99 |
| $F1_S$ | $= \frac{2 * P_S * R_S}{P_S + R_S} = \frac{2 * 1.00 * 0.98}{1.00 + 0.98} = \frac{1.96}{1.98} = 0.989$ | 0.99 |
| $F1_N$ | $= \frac{2 * P_N * R_N}{P_N + R_N} = \frac{2 * 1.00 * 1.00}{1.00 + 1.00} = \frac{2.00}{2.00} = 1.00$ | 1.00 |

| | | |
|--------|---|------|
| $F1_T$ | $\frac{2 * P_T * R_T}{P_T + R_T} = \frac{2 * 1.00 * 0.99}{1.00 + 0.99} = \frac{1.98}{1.99} = 0.994$ | 0.99 |
|--------|---|------|

Pada tabel 4.6 merupakan perhitungan *precision*, *recall*, dan *f1-score* pada model deteksi stunting dengan konfigurasi $C = 10$, $\gamma = 5$, dan dengan pendekatan OVO dengan perhitungan secara manual berdasarkan jumlah kelas, jumlah data, hasil *precision* setiap kelas, hasil *recall* setiap kelas, dan hasil *f1-score* setiap kelas. Sehingga didapat rata-rata nilai *precision* 0.994, *recall* sebesar 0.991, dan *f1-score* sebesar 0.994 pada keempat kelas.



Gambar 4. 8 Grafik *confusion matrix* model deteksi status berat badan

Gambar 4.8 menunjukkan hasil grafik *confusion matrix* pada model deteksi status berat badan dengan konfigurasi $C = 10$ dan $\gamma = 1$. Untuk memudahkan analisis dan perhitungan maka dibuat tabel seperti dibawah. Dengan TP adalah nilai *True Positif*, FP (*False Positif*), FN (*False Negatif*), dan TN (*True Negatif*).

Tabel 4. 7 TP, FP, FN, dan TN dari model deteksi status berat badan

| Kelas | TP | FP | FN | TN |
|-----------------------------|-------|-----|-----|--------|
| <i>Severely Underweight</i> | 5.009 | 165 | 158 | 14.200 |
| <i>Underweight</i> | 1.060 | 327 | 308 | 17.825 |
| Normal | 5.143 | 329 | 344 | 13.738 |
| <i>Overweight</i> | 7.334 | 172 | 183 | 11.861 |

*Keterangan:

- a. SU = *Severely Underweight*
- b. U = *Underweight*
- c. N = Normal
- d. O = *Overweight*

Pada tabel 4.7 dilakukan pemetaan pada hasil grafik *confusion matrix* model deteksi status berat badan agar mempermudah dalam analisis dan perhitungan secara manual.

Tabel 4. 8 Perhitungan *precision*, *recall*, dan *f1-Score* model deteksi status berat badan

| | Perhitungan | Hasil |
|----------|---|-------|
| | $Precision (P) = \frac{TP}{TP + FP}$ | |
| P_{SU} | $= \frac{TP_{SU}}{TP_{SU} + FP_{SU}} = \frac{5.009}{5.009 + 165} = \frac{5.009}{5.174} = 0.968$ | 0.97 |
| P_U | $= \frac{TP_U}{TP_U + FP_U} = \frac{1.060}{1.060 + 327} = \frac{1.060}{1.387} = 0.764$ | 0.76 |
| P_N | $= \frac{TP_N}{TP_N + FP_N} = \frac{5.143}{5.143 + 329} = \frac{5.143}{5.472} = 0.939$ | 0.94 |
| P_O | $= \frac{TP_O}{TP_O + FP_O} = \frac{7.334}{7.334 + 172} = \frac{7.334}{7.506} = 0.977$ | 0.98 |
| | $Recall (R) = \frac{TP}{TP + FN}$ | |
| R_{SU} | $= \frac{TP_{SU}}{TP_{SU} + FN_{SU}} = \frac{5.009}{5.009 + 158} = \frac{5.009}{5.167} = 0.969$ | 0.97 |
| R_U | $= \frac{TP_U}{TP_U + FN_U} = \frac{1.060}{1.060 + 308} = \frac{1.060}{1.368} = 0.774$ | 0.77 |

| | | |
|---|---|------|
| R_N | $= \frac{TP_N}{TP_N + FN_N} = \frac{5.143}{5.143 + 344} = \frac{5.143}{5.487} = 0.937$ | 0.94 |
| R_O | $= \frac{TP_O}{TP_O + FN_O} = \frac{7.334}{7.334 + 183} = \frac{7.334}{7.517} = 0.975$ | 0.98 |
| F1 – Score = $\frac{2 * Precision * Recall}{Precision + Recall}$ | | |
| $F1_{SU}$ | $\frac{2 * P_{SU} * R_{SU}}{P_{SU} + R_{SU}} = \frac{2 * 0.97 * 0.97}{0.97 + 0.97} = \frac{1.88}{1.94} = 0.969$ | 0.97 |
| $F1_U$ | $\frac{2 * P_U * R_U}{P_U + R_U} = \frac{2 * 0.76 * 0.77}{0.76 + 0.77} = \frac{1.17}{1.53} = 0.764$ | 0.76 |
| $F1_N$ | $\frac{2 * P_N * R_N}{P_N + R_N} = \frac{2 * 0.94 * 0.94}{0.94 + 0.94} = \frac{1.76}{1.88} = 0.936$ | 0.94 |
| $F1_O$ | $\frac{2 * P_O * R_O}{P_O + R_O} = \frac{2 * 0.98 * 0.98}{0.98 + 0.98} = \frac{1.92}{1.96} = 0.979$ | 0.98 |

Pada tabel 4.8 merupakan perhitungan *precision*, *recall*, dan *f1-score* pada model deteksi status berat badan dengan konfigurasi $C = 10$, $\gamma = 1$, dan dengan pendekatan OVO dengan perhitungan secara manual berdasarkan jumlah kelas, jumlah data, hasil *precision* setiap kelas, hasil *recall* setiap kelas, dan hasil *f1-score* setiap kelas. Sehingga didapat rata-rata nilai *precision* 0.912, *recall* sebesar 0.913, dan *f1-score* sebesar 0.912 pada keempat kelas

4.2 Deployment

Streamlit digunakan sebagai *platform* untuk *deployment* dikarenakan kemudahan penggunaan, kecepatan pengembangan dan kebutuhan spesifik penelitian. Selain itu, streamlit juga bersifat *open source* dan *free deployment*, dan menggunakan bahasa Python sebagai bahasa bawaannya. Sehingga tidak memerlukan konfigurasi yang rumit dalam menghubungkan dan mengimplementasikan model kedalam sistemnya.

4.2.1 Serialisasi Model

Model yang telah di-*training* kemudian dilakukan serialisasi model. Serialisasi model adalah proses mengubah keadaan model yang telah dilatih menjadi format yang dapat disimpan kedalam *disk* sehingga suatu saat ketika model akan digunakan tidak perlu melakukan *training* model kembali. Proses dan *coding* serialisasi model adalah sebagai berikut:

```
import joblib

model_path = '/content/drive/My Drive/Deteksi Stunting/model_deteksi_stunting.pkl'

joblib.dump(svm_model, model_path)

['/content/drive/My Drive/Deteksi Stunting/model_deteksi_stunting.pkl']
```

Gambar 4. 9 Serialisasi model deteksi stunting

Pada gambar 4.9 merupakan proses serialisasi model yang sebelumnya telah melalui proses *training* dan *testing*. Proses serialisasi dilakukan dengan menggunakan *library* joblib. Model deteksi stunting disimpan dengan nama “model_deteksi_stunting.pkl”.

```
import joblib

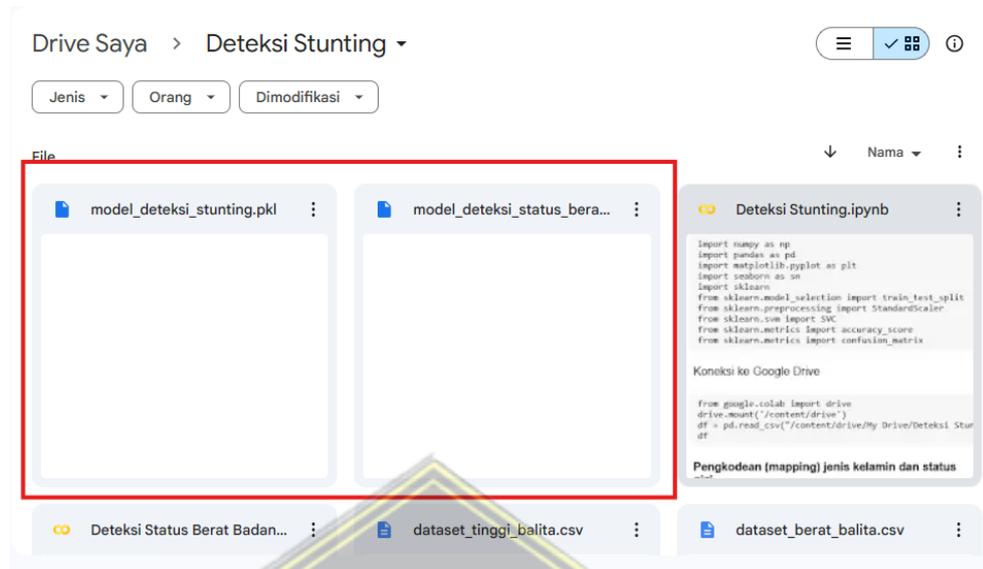
model_path = '/content/drive/My Drive/Deteksi Stunting/model_deteksi_status_berat_badan.pkl'

joblib.dump(svm_model, model_path)

['/content/drive/My Drive/Deteksi Stunting/model_deteksi_status_berat_badan.pkl']
```

Gambar 4. 10 Serialisasi model deteksi status berat badan

Pada gambar 4.10 merupakan proses serialisasi model yang sebelumnya telah melalui proses *training* dan *testing*. Proses serialisasi dilakukan dengan menggunakan *library* joblib. Model deteksi status berat badan disimpan dengan nama “model_deteksi_berat_badan.pkl”.

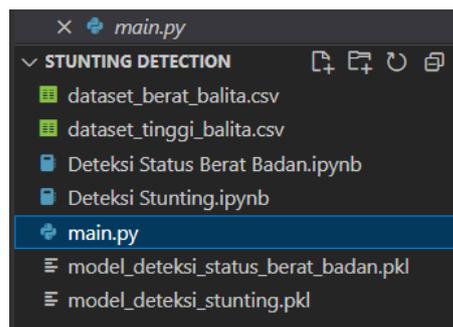


Gambar 4. 11 File hasil serialisasi model

Gambar 4.11 merupakan file hasil serialisasi pada model_deteksi stunting dan model_deteksi_berat_badan. File hasil serialisasi dari model inilah yang nantinya penulis gunakan untuk membangun sistem deteksi stunting dan status gizi balita.

4.2.2 Membangun Sistem

Pembangunan sistem dilakukan dengan menggunakan *framework* streamlit, dan untuk pengkodean penulis menggunakan *text editor visual studio code* (VS-Code) dengan versi python adalah 3.10.1. Setelah itu, penulis melakukan installasi streamlit dan beberapa *library* python yang digunakan pada saat pembangunan model pada VS-Code. Kemudian melakukan *import file dataset, file training model, dan file* serialisasi model kedalam *folder project*.



Gambar 4. 12 Project sistem

Pada gambar 4.12 terdapat file dengan nama `main.py` yang dimana file tersebut adalah tempat pengkodean dari aplikasi streamlit yang telah dibuat. Untuk tampilan dari sistem adalah sebagai berikut.

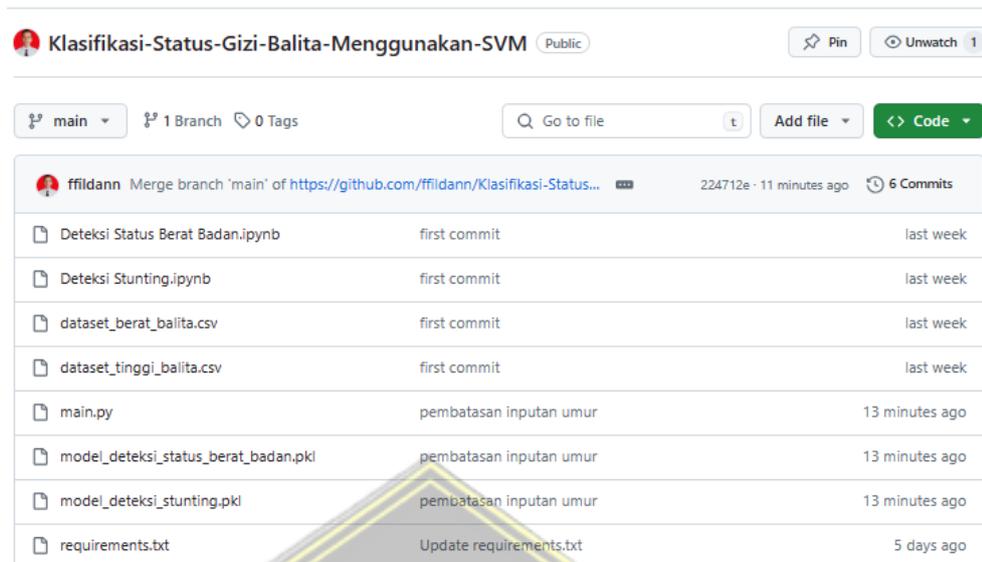
The image shows a web browser window displaying a Streamlit application. The title of the application is "Klasifikasi Stunting dan Status Gizi Balita Dengan SVM". Below the title, there is a form with four input fields: "Inputkan umur balita (bulan)", "Jenis Kelamin" (with radio buttons for "laki-laki" and "perempuan"), "Masukkan tinggi badan (cm)", and "Masukkan berat badan (kg)". A "Submit" button is located at the bottom of the form. The browser's address bar shows "localhost:8501".

Gambar 4. 13 Tampilan sistem

Pada sistem terdapat beberapa *form inputan* yang harus diisi oleh *user* ketika akan melakukan proses klasifikasi yaitu umur balita (bulan), jenis kelamin, tinggi badan (cm), dan berat badan (kg) dan terdapat tombol *submit* untuk melihat hasil prediksi.

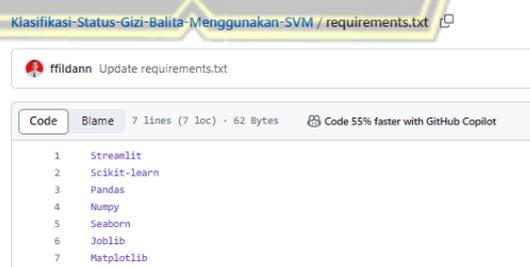
4.2.3 Deployment Sistem

Pada gambar 4.13 merupakan tampilan dari sistem klasifikasi stunting dan status gizi balita dengan SVM yang masih berada di *localhost*. Agar sistem dapat digunakan dan diakses oleh orang lain, maka perlu dilakukan *deployment* sistem. Untuk men-*deploy* sistem digunakan streamlit *cloud*. Sebelum melakukan *deployment* sistem, *project* terlebih dahulu harus di *push* ke akun github karena nantinya streamlit *cloud* akan mengambil *source code project* yang ada di github untuk dilakukan *deployment*. Berikut adalah *link* repositori dari *project* yang telah dilakukan *push* ke *platform* github: <https://github.com/ffildann/Klasifikasi-Status-Gizi-Balita-Menggunakan-SVM>.

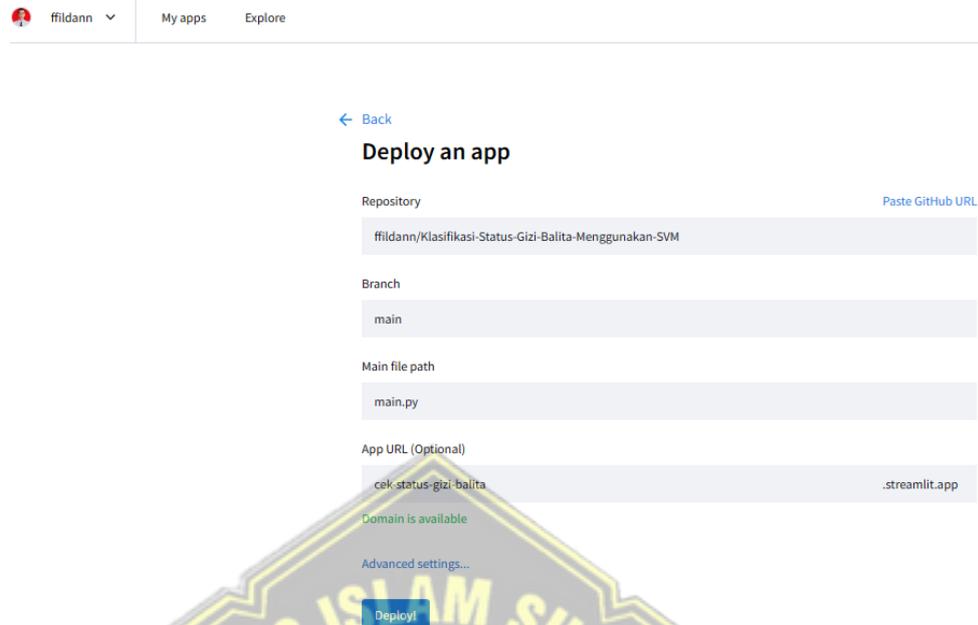


Gambar 4. 14 *Repository project* di Github

Gambar 4.14 merupakan tampilan dari *repository* tempat *project* di-*upload*. Setelah *project* berhasil di-*push* pada *platform* Github, selanjutnya membuat *file* *requirement.txt* yang berisi *library* apa saja yan dipakai pada *project* karena ini merupakan pemberitahuan ke *streamlit cloud*, *library* apa saja yang harus di-*install* pada saat *deployment* nanti. Kita dapat menuliskan versi dari *library* yang dipakai disamping nama *library*. Karena pada saat pengembangan *project* ini menggunakan *library* Python semuanya versi terbaru, maka tidak perlu menuliskan versi *library*nya. Berikut adalah tampilan dari *file* *requirement.txt* pada gambar 4.15.

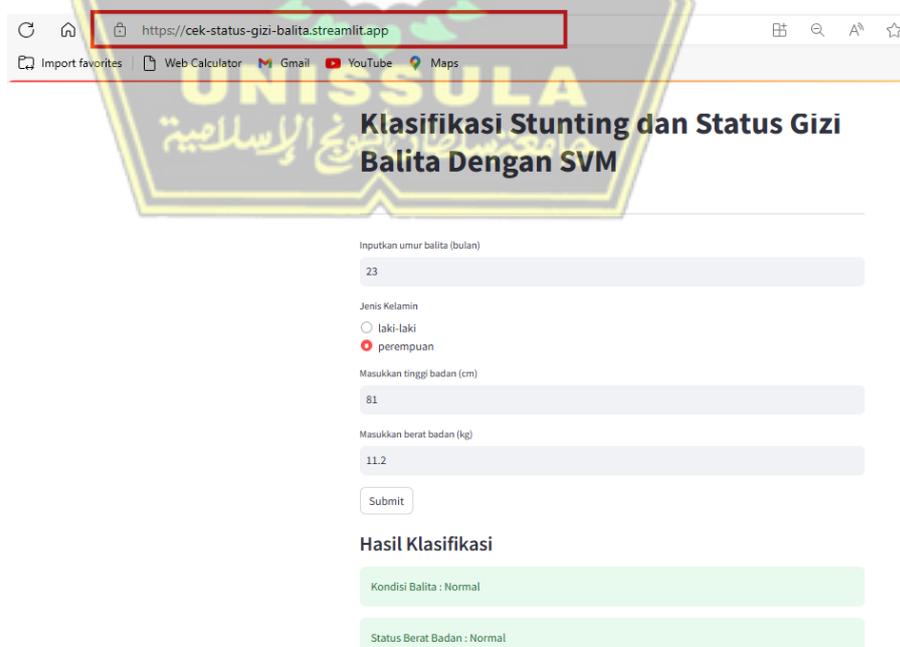


Gambar 4. 15 File requirment.txt



Gambar 4. 16 *Deployment* sistem pada streamlit cloud

Gambar 4.16 adalah proses *deployment* sistem pada *platform* streamlit *cloud*. Dimana kita harus mengisikan informasi terkait *project* yang telah kita lakukan *pushing* ke github. Kita juga dapat mengatur alamat *domain* dari sistem yang kita buat. Untuk alamat domain sistem kita *setting* dengan menggunakan nama “cek-status-gizi-balita.streamlit.app”.



Gambar 4. 17 Tampilan sistem yang telah dilakukan *deployment*

Gambar 4.17 adalah tampilan sistem yang telah berhasil dilakukan *deployment* pada *streamlit cloud*. Untuk mengakses sistem, *user* dapat mengakses *link* berikut: <https://cek-status-gizi-balita.streamlit.app/>. Kemudian *user* dapat mengisi informasi balita dan melakukan prediksi status gizi balita dengan menekan tombol '*submit*'. Setelah menekan tombol *submit*, maka akan muncul informasi status gizi dari balita.

4.3 Pengujian Sistem

Metode pengujian sistem yang digunakan pada penelitian ini adalah menggunakan metode *blackbox testing*. Metode *blackbox testing* merupakan suatu pendekatan yang menguji fungsionalitas sebuah sistem tanpa memperhatikan struktur internal atau kode sumbernya dan hanya meninjau *input* dan *output* dari sistem. Pada pengujian sistem ini digunakan *functional testing* yang merupakan salah satu jenis dari *blackbox testing*. Proses pengujian fungsional dilakukan untuk memastikan bahwa fungsi dan fitur sistem atau aplikasi berjalan sesuai dengan spesifikasi fungsional yang telah ditentukan.

Tabel 4. 9 Hasil Pengujian *Blackbox Testing*

| Input | Kasus Pengujian | Output | Kesimpulan |
|----------------------|---|--------|------------|
| <i>Link</i> aplikasi | Mengakses <i>link</i> aplikasi | Sesuai | Normal |
| Form data balita | Memasukkan umur, jenis kelamin, tinggi badan, dan berat badan | Sesuai | Normal |
| Tombol <i>submit</i> | Menekan tombol <i>submit</i> | Sesuai | Normal |
| Testing klasifikasi | Mengecek hasil klasifikasi stunting | Sesuai | Normal |

| Input | Kasus Pengujian | Output | Kesimpulan |
|------------------------|---|---------------|-------------------|
| Testing klasifikasi | Mengecek hasil klasifikasi status berat badan | Sesuai | Normal |

Dari tabel 4.9 diterangkan bahwa pengembangan aplikasi dapat menjalankan semua fungsinya dengan baik. Mulai dari mengakses *link* aplikasi, memasukkan data balita, tombol *submit* yang berfungsi untuk perintah melakukan klasifikasi, hingga hasil *output* dari proses klasifikasi.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat diambil kesimpulan bahwa metode SVM (*Support Vector Machine*) dapat diimplementasikan dan dapat menghasilkan akurasi yang tinggi dalam mengklasifikasikan balita stunting dan status gizi balita. Hal tersebut telah dilakukan pengujian dengan melakukan perbandingan terhadap beberapa percobaan dan dengan parameter yang berbeda. Berikut beberapa poin kesimpulan yang penulis dapat selama penelitian ini:

1. Metode SVM (*Support Vector Machine*) dapat diimplementasikan pada pengklasifikasian balita stunting dan status gizi balita.
2. Kernel RBF (*Radial Basis Fuction*) memiliki akurasi paling tinggi dibandingkan dengan kernel yang lain pada model deteksi stunting dan model deteksi status berat badan.
3. Pada model deteksi stunting konfigurasi terbaik yaitu pada percobaan keenam yaitu dengan $C = 10$ dan $\gamma = 5$. Menghasilkan *accuracy* sebesar 99.50% dengan rata-rata nilai *precision* 0.994, *recall* sebesar 0.991, dan *f1-score* sebesar 0.994 pada keempat kelas.
4. Pada model deteksi status berat badan konfigurasi terbaik yaitu pada percobaan ke tiga yaitu dengan $C = 10$ dan $\gamma = 1$. Menghasilkan *accuracy* sebesar 94.92% dengan rata-rata nilai *precision* 0.912, *recall* sebesar 0.913, dan *f1-score* sebesar 0.912 pada keempat kelas.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis menyarankan untuk penelitian yang akan datang adalah:

1. Dari hasil penelitian ini, diharapkan penelitian selanjutnya dapat melakukan modifikasi penelitian yang telah dilakukan dengan mengubah dataset, fitur, atau bahkan metode yang berbeda yang menghasilkan akurasi yang lebih baik.
2. Penelitian selanjutnya dapat mengembangkan sistem menjadi lebih baik dan kompleks dengan menambahkan fitur yang belum ada atau menerapkannya dengan basis yang berbeda seperti *mobile*.



DAFTAR PUSTAKA

Apriyani, H.;Kurniati (2020) “Perbandingan Metode Naïve Bayes Dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes Melitus,” *Journal of Information Technology Ampera*, 1, hal. 133–143.

Aulia, T. M. P. *dkk.* (2021) “Perbandingan Kernel Support Vector Machine (Svm) Dalam Penerapan Analisis Sentimen Vaksinisasi Covid-19,” *SINTECH (Science and Information Technology) Journal*, 4(2), hal. 139–145.

Chafidin, A. N. *dkk.* (2022) “Sistem Pendeteksi Gejala Stunting pada Anak dengan Metode Certainty factor Berbasis Website,” *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 6(3), hal. 366–377. doi: 10.35870/jtik.v6i3.434.

Fitriyah, N. *dkk.* (2020) “Analisis Sentimen Gojek Pada Media Sosial Twitter Dengan Klasifikasi Support Vector Machine (SVM),” *Jurnal Gaussian*, 9(3), hal. 376–390.

Hovi, H. S. W. *dkk.* (2022) “Prediksi Penyakit Diabetes MengguChafidin, A. N., Triayudi, A. dan Andrianingsih, A. (2022) ‘Sistem Pendeteksi Gejala Stunting pada Anak dengan Metode Certainty factor Berbasis Website,’ Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 6(3), hal. ,” *Informatics and Digital Expert (INDEX)*, 4(1), hal. 40–45.

Jumeilah, F. S. (2017) “Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian,” *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 1(1), hal. 19–25.

Keputusan Menteri Kesehatan Republik Indonesia Nomor Hk.01.07/Menkes/1928/2022 Tentang Pedoman Nasional Pelayanan Kedokteran Tata Laksana Stunting (2022).

Kusumaningrum, R. *dkk.* (2020) “Benchmarking of multi-class algorithms for classifying documents related to stunting,” *Applied Sciences*, 10(23), hal. 8621.

Nababan, J. F. (2021) “KLASIFIKASI PENDERITA STUNTING DENGAN METODE SUPPORT VECTOR MACHINE (STUDI KASUS: LIMA

PUSKESMAS DI KOTA BANDAR LAMPUNG).”

Nugroho, A. S. *dkk.* (2003) “Support vector machine,” *Proceeding Indones. Sci. Meeting Cent. Japan.*

Peraturan Menteri Kesehatan RI Nomor 2 Tahun 2020 (2020).

Retnoningsih, E. dan Pramudita, R. (2020) “Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python,” *Bina Insani Ict Journal*, 7(2), hal. 156–165.

Sutarto, S. T. T. *dkk.* (2018) “Stunting, Faktor Resiko dan Pencegahannya,” *Agromedicine Unila*, 5(1), hal. 540–545.

