

**KLASIFIKASI *STUNTING* DI KABUPATEN LAMPUNG BARAT  
BERDASARKAN DATA ANTROPOMETRI MENGGUNAKAN  
ANALISIS ALGORITMA *SUPPORT VECTOR MACHINE* DAN  
*XTREME GRADIENT BOOSTING***

Oleh

**NESA DWI CAHYANI  
2017051009**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA ILMU KOMPUTER**

Pada

**Program Studi Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

**KLASIFIKASI *STUNTING* DI KABUPATEN LAMPUNG BARAT  
BERDASARKAN DATA ANTROPOMETRI MENGGUNAKAN  
ANALISIS ALGORITMA *SUPPORT VECTOR MACHINE* DAN  
*XTREME GRADIENT BOOSTING***

**(Skripsi)**

**Oleh**

**NESA DWI CAHYANI  
2017051009**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

## ABSTRAK

### **KLASIFIKASI *STUNTING* DI KABUPATEN LAMPUNG BARAT BERDASARKAN DATA ANTROPOMETRI MENGGUNAKAN ANALISIS ALGORITMA *SUPPORT VECTOR MACHINE* DAN *XTREME GRADIENT BOOSTING***

Oleh

**NESA DWI CAHYANI**

Komponen penting dalam memastikan tumbuh kembang anak adalah pemenuhan gizi yang baik, salah satu konsekuensi yang didapatkan apabila seorang anak kekurangan asupan gizi adalah *stunting*. Diagnosis *stunting* pada saat ini masih dilakukan melalui pemeriksaan dalam yang membutuhkan waktu dan biaya yang relatif mahal. Terdapat sebuah gagasan yang dapat dilakukan dengan menggunakan bantuan *machine learning*, tepatnya pada bagian klasifikasi dengan menggunakan metode SVM dan XGBoost. Penelitian ini berfokus pada implementasi data *stunting* di Kabupaten Lampung Barat dan membandingkan kedua metode yang digunakan. Perbandingan yang dilakukan yaitu perbandingan hasil pengujian dengan menggunakan *confussion matrix*. Selain itu perbandingan juga dilakukan dengan membandingkan waktu komputasi yang dibutuhkan. Pada penelitian ini proses pembagian dataset akan dilakukan dengan menggunakan *k-fold cross validation* ( $k=10$ ). Berdasarkan hasil pengujian didapatkan bahwa hasil implementasi dengan metode XGBoost mendapatkan hasil yang lebih baik yaitu akurasi sebesar 99.79% dengan waktu komputasi rata-rata per pengujian yaitu 0.43 menit, sedangkan SVM mendapatkan hasil akurasi sebesar 99.35% dengan waktu komputasi rata-rata per pengujian yaitu 18.09 menit.

Kata kunci : *Stunting*, Klasifikasi, SVM, XGBoost

## **ABSTRACT**

### **STUNTING CLASSIFICATION IN WEST LAMPUNG REGENCY BASED ON ANTHROPOMETRIC DATA USING ALGORITHM ANALYSIS OF SUPPORT VECTOR MACHINE AND XTREME GRADIENT BOOSTING**

**By**

**NESA DWI CAHYANI**

An important component in ensuring children's growth and development is the fulfilment of good nutrition, one of the consequences obtained if a child lacks nutritional intake is stunting. The diagnosis of stunting at this time is still carried out through an internal examination which requires time and relatively expensive costs. There is an idea that can be done using the help of machine learning, precisely in the classification section using the SVM and XGBoost methods. This research focusses on the implementation of stunting data in West Lampung Regency and compares the two methods used. The comparison that is done is a comparison of the test results using a confusion matrix. In addition, the comparison is also carried out by comparing the required computational time. In this research, the dataset division process will be carried out using k-fold cross validation (k=10). Based on the test results, it was found that the implementation results with the XGBoost method got better results, namely 99.79% accuracy with an average computing time per test of 0.43 minutes, while SVM got an accuracy result of 99.35% with an average computing time per test of 18.09 minutes.

**Keywords :** Stunting, Classification, SVM, XGBoost

Judul Skripsi : **KLASIFIKASI *STUNTING* DI KABUPATEN LAMPUNG BARAT BERDASARKAN DATA ANTROPOMETRI MENGGUNAKAN ANALISIS ALGORITMA *SUPPORT VECTOR MACHINE* DAN *XTREME GRADIENT BOOSTING***

Nama Mahasiswa : Nesa Dwi Cahyani

Nomor Pokok Mahasiswa : 2017051009

Program Studi : S1 Ilmu Komputer

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

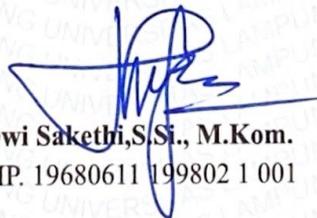


**Dr. Aristoteles, S.Si., M.Si.**  
NIP. 19810521 200604 1 002



**Ridho Sholehurrohmat, M.Mat.**  
NIP. 232111970128 101

2. Ketua Jurusan Ilmu Komputer



**Dwi Sakethi, S.Si., M.Kom.**  
NIP. 19680611 199802 1 001

**MENGESAHKAN**

1. Tim Penguji

Ketua : Dr. Aristoteles, S.Si., M.Si.



Sekretaris : Ridho Sholehurrohman, M.Mat.



Penguji Utama : Prof. Admi Syarif, Ph.D.



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi : 21 Juni 2024

## PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul **“Klasifikasi *Stunting* di Kabupaten Lampung Barat Berdasarkan Data Antropometri Menggunakan Analisis Algoritma *Support Vector Machine* dan *Xtreme Gradient Boosting*”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 21 Juni 2024

Penulis,



Nesa Dwi Cahyani

NPM. 2017051009

## RIWAYAT HIDUP



Penulis lahir di Way Mengaku, Liwa Lampung Barat pada tanggal 19 Juli 2002 sebagai anak kedua dari pasangan Bapak Eko Sujanarko, S.E dan Ibu Hartini. Penulis telah menyelesaikan Pendidikan formal di SD Negeri 1 Way Mengaku pada tahun 2014. Kemudian SMP N Sekuting Terpadu pada tahun 2017 dan SMA Negeri 1 Liwa pada tahun 2020. Penulis terdaftar sebagai mahasiswi program studi Ilmu Komputer di Universitas Lampung melalui jalur undangan atau SNMPTN.

Selama menjadi mahasiswi di Ilmu Komputer, penulis aktif di dalam berbagai kegiatan baik didalam maupun diluar Universitas Lampung. Kegiatan yang di lakukan adalah sebagai berikut,

1. Menjadi Asisten Dosen mata kuliah Logika di Jurusan Ilmu Komputer tahun 2021.
2. Menjadi Asisten Dosen mata kuliah Rekayasa Perangkat Lunak di Jurusan Ilmu Komputer pada tahun 2022.
3. Menjadi Asisten Dosen mata kuliah Analisis dan Desain Sistem Informasi di Jurusan Ilmu Komputer pada tahun 2023.
4. Menjadi Anggota Bidang Kaderisasi Himpunan Mahasiswa Ilmu Komputer pada tahun 2022.
5. Menjadi Bendahara Bidang Kaderisasi Himpunan Mahasiswa Ilmu Komputer pada tahun 2022.
6. Menjadi Announcer Chief Radio Kampus Unila pada tahun 2022.
7. Mengikuti Praktik Kerja Lapangan di PT. Bukit Asam Unit Pelabuhan Tarahan pada tahun 2023.
8. Menjadi Manager On Air Radio Kampus Unila pada tahun 2023.
9. Mengikuti Kuliah Kerja Nyata di Desa Agung Timur, Kecamatan Kalirejo, Kabupaten Lampung Tengah pada tahun 2023.

## **PERSEMBAHAN**

Segala puji saya panjatkan kepada Allah SWT atas segala Rahmat-Nya, serta shalawat dan salam senantiasa juga tercurahkan kepada Nabi Muhammad SAW sehingga saya dapat menyelesaikan skripsi ini.

### **Skripsi ini saya persembahkan kepada :**

Kedua orang tua penulis, juga kakak dan adik penulis yang telah kebersamai serta membantu penulis untuk menyelesaikan skripsi ini, juga untuk diri saya pribadi yang telah berhasil menyelesaikan pendidikan di Ilmu Komputer Universitas Lampung.

## MOTTO

“Aku membahayakan nyawa ibuku untuk lahir ke kedunia, jadi tidak mungkin aku tidak ada artinya”

-Ali bin Abi Thalib

"إِنَّمَا أَمْرُهُ إِذَا أَرَادَ شَيْئًا أَنْ يَقُولَ لَهُ كُنْ فَيَكُونُ"

“Sesungguhnya urusan-Nya apabila Dia menghendaki sesuatu hanyalah berkata kepadanya: '**Jadilah!**' maka terjadilah”

-Surah Yasin (Ayat 82)

“Cukuplah Allah SWT sebagai penolong, segala yang sulit akan menjadi mudah dengan campur tangan-Nya dan segala yang mustahil akan menjadi mungkin dengan ridho-Nya”

-Nesa

## SANCAWACANA

Puji syukur kehadiran Allah SWT. yang telah memberikan rahmat dan hidayat-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini tepat waktu. Sholawat serta salam penulis sanjungkan kepada Nabi dan Rasul Muhammad SAW yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul “ Klasifikasi *Stunting* di Kabupaten Lampung Barat Berdasarkan data Antropometri Menggunakan Analisis Algoritma *Support Vector Machine* dan *Xtreme Gradient Boosting* ” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana jurusan Ilmu Komputer Universitas Lampung.

Dalam proses penelitian ini, sangat banyak orang-orang yang terlibat dalam pelaksanaannya. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar besarnya kepada:

1. Allah SWT yang telah memberikan hidayah kesehatan dan kemampuan untuk menyelesaikan skripsi ini.
2. Kedua orang tua penulis, Bapak Eko Sujanarko, S.E dan Ibu Hartini yang tidak henti-hentinya selalu memberikan segala doa dan semangat serta mempercayai penulis untuk menyelesaikan skripsi ini. Untuk Kakak penulis Restu Mulyajansih, S.E yang sudah menjadi panutan penulis serta adik-adik penulis Dinara Safitri dan Wisnu Aldric Kayana yang telah kebersamai serta mendoakan penulis untuk tiap kesempatannya.
3. Bapak Dwi Sakethi, S.Si., M.Kom selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Bapak Dr. Aristoteles, S.Si., M.Si selaku pembimbing utama yang telah membimbing dan memberikan banyak arahan kepada penulis sehingga penelitian yang penulis lakukan dapat menjadi lebih baik.

5. Bapak Ridho Sholehurrohman, M.Mat sebagai pembimbing kedua yang telah memberikan bimbingan dalam penulisan skripsi serta arahan yang sangat membantu sehingga penulisan skripsi ini dapat terselesaikan.
6. Prof. Admi Syarif, Ph.D sebagai pembahas yang telah banyak memberikan masukan, dan saran yang bermanfaat bagi penulis dalam menyusun serta menyelesaikan skripsi.
7. Bapak, Ibu Dosen dan semua Staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan banyak ilmu dan membantu segala urusan administrasi yang penulis butuhkan.
8. Nasywa Nathania Wirawan yang telah kebersamai melalui 3 seminar dan 1 sidang bersama-sama.
9. Teman-teman Sabyan, JCC, BD, Lord RPL yang telah kebersamai dan membantu penulis dalam menyelesaikan skripsi ini.
10. Untuk diri saya pribadi Nesa Dwi Cahyani, terima kasih telah bertahan dan berhasil menyelesaikan pendidikan di prodi ilmu komputer.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari kata sempurna karena masih terbatasnya pengetahuan, pengalaman, dan kemampuan penulis. Oleh karena itu, penulis sangat menghargai dan mengharapkan saran dan kritik untuk penelitian ini sebagai bahan pertimbangan untuk karya tulis yang akan datang. Semoga skripsi ini dapat bermanfaat bagi semua pihak

Bandar Lampung, 21 Juni 2024

Penulis,



Nesa Dwi Cahyani

NPM. 2017051009

## DAFTAR ISI

	Halaman
DAFTAR TABEL.....	v
DAFTAR GAMBAR .....	viii
I. PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
II. TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu.....	5
2.2 Landasan Teori .....	8
2.2.1 <i>Stunting</i> .....	8
2.2.2 Antropometri.....	9
2.2.3 Klasifikasi .....	14
2.2.4 <i>Machine Learning</i> .....	14
2.2.5 <i>K-Fold Cross Validation</i> .....	15
2.2.6 <i>Support Vector Machine</i> .....	15
2.2.7 <i>Xtreme Gradient Boosing</i> .....	18
2.2.8 <i>Confussion Matrix</i> (Matriks Konfusi).....	21
III. METODOLOGI PENELITIAN.....	23
3.1 Tempat dan Waktu .....	23
3.1.1 Tempat.....	23
3.1.2 Waktu .....	23
3.2 Alat Pendukung .....	25
3.2.1 Perangkat Keras .....	25
3.2.1 Perangkat Lunak.....	25

3.3	Data.....	25
3.4	Tahapan Penelitian .....	27
3.4.1	Studi Literatur .....	27
3.4.2	Pengumpulan Data .....	28
3.4.3	<i>Pre-Processing</i> Data .....	28
3.3.4	<i>K-Fold Cross Validation</i> .....	28
3.3.5	Implementasi Model SVM dan <i>XGBoost</i> .....	29
3.3.6	Pengujian.....	29
3.3.7	Perbandingan Hasil Klasifikasi .....	29
IV.	PEMBAHASAN .....	30
4.1	Pengumpulan Data.....	30
4.2	<i>Pre-Processing</i> Data.....	30
4.2.1.	Data Cleaning.....	30
4.2.2.	Data Transforming .....	34
4.2.3.	Normalisasi Data.....	40
4.2.1.	Balancing Data .....	42
4.3	Pembagian Data.....	43
4.4	Pembentukan Model Klasifikasi.....	44
4.4.1	Model Klasifikasi <i>Support Vector Machine</i> (SVM) .....	45
4.4.2	Model Klasifikasi <i>Xtreme Gradient Boosting</i> (XGBoost).....	54
4.5	Perbandingan Algoritma SVM dan XGBoost .....	66
V.	PENUTUP .....	68
5.1	Kesimpulan.....	68
5.2	Saran .....	69
	DAFTAR PUSTAKA .....	71

## DAFTAR TABEL

Tabel 1. Penelitian Terdahulu .....	5
Tabel 2. Kategori dan Ambang Batas Status Gizi Anak .....	10
Tabel 3. Standar PB/U Anak Perempuan Umur 0-60 Bulan.....	10
Tabel 4. Standar PB/U Anak Laki-Laki Umur 0-60 Bulan.....	12
Tabel 5. Parameter pada XGBoost.....	19
Tabel 6. Matriks Konfusi .....	21
Tabel 7. Waktu Penelitian .....	24
Tabel 8. Jumlah Data Stunting Per Kelas.....	26
Tabel 9. Contoh Data Penelitian .....	26
Tabel 10. Keterangan Flowchart .....	29
Tabel 11. Jumlah rekap cleaning data 0 .....	32
Tabel 12. Jumlah rekap cleaning data kosong .....	33
Tabel 13. Perbedaan jumlah record data sebelum dan sesudah cleaning data	33
Tabel 14. Rekap perbedaan jumlah data sebelum dan sesudah cleaning data .	34
Tabel 15. Rekap jumlah perubahan TB/U awal dan TB/U baru .....	36
Tabel 16. Perubahan nama atribut.....	38
Tabel 17. Nilai minimum dan maksimum dari masing-masing data .....	41
Tabel 18. Hasil Normalisasi (Record data ilustrasi) .....	41
Tabel 19. Nilai minimum dan maksimum dari masing-masing data .....	42
Tabel 20. Hasil Normalisasi .....	42
Tabel 21. Jumlah data per kelas sebelum dilakukan balancing data.....	43
Tabel 22. Jumlah data per kelas setelah dilakukan balancing data .....	43
Tabel 23. Jumlah pembagian data .....	44
Tabel 24. Nilai Hyperparameter SVM .....	45
Tabel 25. Percobaan metode SVM tanpa parameter .....	45
Tabel 26. Confussion matriks tanpa parameter .....	46

Tabel 27. Hasil akurasi model SVM dengan menggunakan kernel .....	46
Tabel 28. Confussion matriks kernel polynomial .....	47
Tabel 29. Hasil akurasi model SVM dengan menggunakan Cost.....	47
Tabel 30. Confussion matriks SVM cost 1000 .....	48
Tabel 31. Hasil akurasi model SVM dengan menggunakan Gamma .....	48
Tabel 32. Confussion matriks SVM gamma 1000 .....	49
Tabel 33. Hasil akurasi model SVM dengan Iterasi Maksimum .....	49
Tabel 34. Confussion matriks SVM Iterasi Maksimum.....	50
Tabel 35. Nilai akurasi terbesar dari masing-masing parameter SVM .....	50
Tabel 36. Hasil akurasi model SVM dengan Gabungan 4 parameter .....	50
Tabel 37. Confussion matriks SVM menggunakan 4 parameter gabungan.....	51
Tabel 38. Hasil Perbandingan Pengujian SVM berdasarkan Accuracy .....	51
Tabel 39. Hasil Perbandingan Pengujian SVM berdasarkan Precision .....	52
Tabel 40. Hasil Perbandingan Pengujian SVM berdasarkan Recall .....	52
Tabel 41. Hasil Perbandingan Pengujian SVM berdasarkan F1-Score.....	53
Tabel 42. Nilai Hyperparameter XGBoost.....	54
Tabel 43. Percobaan metode XGBoost tanpa parameter .....	54
Tabel 44. Confussion matriks tanpa parameter.....	55
Tabel 45. Hasil Akurasi XGBoost menggunakan Learning rate.....	55
Tabel 46. Hasil Pengujian Learning rate.....	56
Tabel 47. Hasil Akurasi XGBoost menggunakan Max depth.....	56
Tabel 48. Hasil Pengujian dengan Max depth .....	57
Tabel 49. Hasil Akurasi XGBoost menggunakan Gamma .....	57
Tabel 50. Hasil Pengujian dengan Gamma .....	58
Tabel 51. Hasil Akurasi XGBoost menggunakan Min child weight .....	59
Tabel 52. Hasil Pengujian dengan Min child weight .....	60
Tabel 53. Hasil Akurasi XGBoost menggunakan Subsample .....	60
Tabel 54. Hasil Pengujian dengan Subsample .....	61
Tabel 55. Hasil Akurasi XGBoost menggunakan Colsample bytree .....	61
Tabel 56. Hasil Pengujian dengan Colsample bytree.....	62
Tabel 57. Nilai akurasi terbesar dari masing-masing parameter XGBoost.....	62
Tabel 58. Hasil Akurasi XGBoost menggunakan gabungan 6 parameter .....	62

Tabel 59. Confussion matriks dengan 6 Parameter Gabungan .....	63
Tabel 60. Hasil Perbandingan Pengujian XGBoost berdasarkan Accuracy ....	63
Tabel 61. Hasil Perbandingan Pengujian XGBoost berdasarkan Precision.....	64
Tabel 62. Hasil Perbandingan Pengujian XGBoost berdasarkan Recall.....	64
Tabel 63. Hasil Perbandingan Pengujian XGBoost berdasarkan F1-Score .....	65
Tabel 64. Perbandingan Algoritma SVM dan XGBoost.....	66
Tabel 65. Rata-rata waktu komputasi SVM dan XGBoost .....	66

## DAFTAR GAMBAR

Gambar 1. Penurunan angka stunting .....	9
Gambar 2. Kategori machine learning berdasarkan cara belajarnya.....	14
Gambar 3. Hyperplane terbaik yang memisahkan dua class.....	16
Gambar 4. Flowchart Tahapan Penelitian.....	27
Gambar 5. Ilustrasi data setelah cleaning data 0 .....	31
Gambar 6. Ilustrasi data 0 .....	31
Gambar 7. Ilustrasi data kosong.....	32
Gambar 8. Ilustrasi data setelah cleaning data kosong .....	33
Gambar 9. Ilustrasi data sebelum ditambah atribut tanggal data .....	34
Gambar 10. Ilustrasi data sesudah penambahan atribut tanggal data.....	35
Gambar 11. Ilustrasi data sebelum penambahan atribut usia .....	35
Gambar 12. Ilustrasi data sesudah penambahan atribut usia.....	35
Gambar 13. Ilustrasi data sebelum penambahan atribut TB/U Baru .....	36
Gambar 14. Ilustrasi data sesudah penambahan atribut TB/U Baru .....	36
Gambar 15. Ilustrasi perubahan data dalam format csv .....	37
Gambar 16. Ilustrasi data setelah dilakukan penghapusan beberapa .....	38
Gambar 17. Atribut yang akan digunakan .....	39
Gambar 18. Perubahan nilai data beberapa atribut .....	39
Gambar 19. Type data sebelum diubah.....	40
Gambar 20. Type data setelah diubah .....	40
Gambar 21. Grafik Perbandingan Pengujian Metode SVM.....	53
Gambar 22. Grafik Perbandingan Pengujian Metode XGBoost .....	65

## I. PENDAHULUAN

### 1.1 Latar Belakang

Salah satu komponen penting dalam memastikan tumbuh kembang anak adalah pemenuhan gizi yang baik, kesehatan yang baik tentunya akan sejalan dengan pemenuhan gizi yang tercukupi dengan baik. Kurangnya asupan gizi akan menyebabkan seseorang mengalami defisit dalam memenuhi kebutuhan tubuhnya, dan salah satu konsekuensinya adalah menjadi rentan terhadap serangan penyakit, yang apabila terjadi akan memperburuk status gizinya (Laswati, 2019). Salah satu masalah kesehatan yang timbul akibat kurangnya asupan gizi adalah *stunting*.

Berdasarkan Keputusan Menteri Kesehatan Republik Indonesia Nomor Hk.01.07/MENKES/1928/2022 *stunting* merupakan suatu kondisi pada anak yang ditandai panjang atau tinggi badan menurut umur dan jenis kelamin di bawah anak seusianya berdasarkan kurva pertumbuhan World Health Organization (WHO), yang disebabkan oleh kekurangan gizi kronis. Kemudian terdapat Kementerian Desa, Pembangunan Daerah Tertinggal, dan Transmigrasi (Kemendes PDTT) lembaga pemerintah Indonesia yang berkomitmen untuk mengurangi tingkat *stunting* di negara ini. Kemendes PDTT telah menerapkan berbagai program untuk mengatasi *stunting*, seperti program Desa Anak Sejahtera (DAS). Selain itu, mereka telah menerbitkan buku saku untuk desa-desa dalam mengatasi *stunting*, yang berisi informasi tentang faktor-faktor yang menyebabkan *stunting*, serta cara menemukan anak-anak yang mengalami *stunting*.

Berdasarkan data Survei Status Gizi Indonesia (SSGI) 2021 status gizi di Indonesia dibagi menjadi empat kategori yaitu kategori baik ( $\text{Stunted} < 20\%$  dan  $\text{Wasted} < 5\%$ ), kategori Akut ( $\text{Stunted} < 20\%$  dan  $\text{Wasted} \leq 5\%$ ), kategori kronis ( $\text{Stunted} \geq 20\%$  dan  $\text{Wasted} < 5\%$ ) dan kategori Kronis-Akut ( $\text{Stunted} \geq 20\%$  dan

Wasted  $\geq 5\%$ ). Provinsi Lampung sendiri masuk kedalam salah satu provinsi dengan kategori akut. Berdasarkan hasil SSGI yang intervensi spesifiknya difokuskan pada masa sebelum kelahiran dan anak usia 6-23 bulan, prevalensi *stunting* di Indonesia terus menurun dari tahun 2019 yaitu sebesar 27.7%, tahun 2021 sebesar 24.4% dan pada tahun 2022 menurun lagi hingga mencapai 21.6%. Sedangkan di Lampung, prevalensi *stunting* pada tahun 2022 sebesar 15,2 persen dan Lampung Barat berada di urutan 8 dari atas (urutan prevalensi *stunting* dari terbesar ke terkecil) dan juga urutan 8 dari bawah (urutan prevalensi dari terkecil ke terbesar) yaitu sebesar 16,4%. Prevalensi ini diartikan sebagai persentase anak di bawah usia 5 tahun yang tinggi badannya lebih rendah dari dua standar deviasi di bawah median untuk usianya (SSGI, 2022).

Diagnosis *stunting* masih banyak dilakukan melalui pemeriksaan biokimia atau pemeriksaan dalam yang melihat indikator gizi dari beberapa nutrisi (Nababan, 2021). Pemeriksaan ini menggunakan proses keperawatan metode berpikir kritis, dimana tiap perawat atau tenaga kesehatan gizi harus menganalisis satu persatu data yang terdapat pada pasien untuk dapat menentukan hasil yang tepat untuk pengelompokan *stunting* (Aristoteles et al., 2023). Cara tersebut sangat baik untuk mengetahui hasil diagnosis *stunting* secara lebih terpercaya tetapi terhambat oleh waktu dan biaya yang relatif mahal. Berdasarkan hal tersebut, terdapat sebuah gagasan untuk memanfaatkan bidang ilmu *machine learning* dengan menggunakan model klasifikasi menggunakan metode *support vector machine* (SVM) dan *xtreme gradient boosting* (XGBoost) dengan memanfaatkan data antropometri. Metode SVM dan XGBoost adalah dua algoritma yang dapat digunakan untuk klasifikasi dan regresi.

Terdapat penelitian terdahulu yang membahas mengenai klasifikasi menggunakan metode SVM dan XGBoost. Pada tahun 2021 dengan judul klasifikasi penderita *stunting* dengan metode SVM (studi kasus: lima puskesmas di kota Bandar Lampung) mendapatkan hasil terbesar dengan rata-rata *accuracy* pada *kernel* linear sebesar 80,8%. Penelitian kedua yang dilakukan oleh (Rahman, 2020) dengan menggunakan metode SVM, MLP dan XGBoost dengan hasil terbaik menggunakan metode SVM dan penelitian ketiga yang dilakukan oleh (Givari et al., 2022) dengan menggunakan metode SVM, random forest dan

XGBoost mendapatkan hasil berupa kinerja XGBoost memiliki akurasi yang lebih besar dibandingkan dengan metode lainnya.

Berdasarkan latar belakang tersebut, penelitian dengan tujuan klasifikasi *stunting* ini dilakukan dengan menggunakan dua metode algoritma sebagai perbandingan, yaitu *support vector machine* dan *xtreme gradient boosting* untuk mendapatkan hasil yang paling baik juga agar dapat membandingkan antara penggunaan metode yang sudah lama (*support vector machine*) dengan metode yang merupakan pengembangan dari metode sebelumnya (*xtreme gradient boosting*) dengan menggunakan data antropometri yang didapatkan dari dinas kesehatan Kabupaten Lampung Barat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas, rumusan masalah pada penelitian ini yaitu:

1. Bagaimana implementasi *support vector machine* dan *xtreme gradient boosting* dalam mengklasifikasikan balita penderita *stunting* yang ada di Kabupaten Lampung Barat.
2. Bagaimana kinerja metode *support vector machine* dan *xtreme gradient boosting* dalam mengklasifikasikan balita penderita *stunting* yang ada di Kabupaten Lampung Barat.

## 1.3 Batasan Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas, batasan masalah pada penelitian ini dibatasi pada:

1. Klasifikasi menggunakan dua algoritma yaitu, *support vector machine* dan *xtreme gradient boosting*.
2. Klasifikasi yang akan dilakukan berdasarkan 4 kelas yaitu, normal, pendek, sangat pendek, dan tinggi.
3. Dataset yang digunakan hanya berfokus pada balita usia 0-60 bulan.

#### **1.4 Tujuan**

Berdasarkan latar belakang yang telah dijabarkan diatas, tujuan pada penelitian ini sebagai berikut.

1. Implementasi model klasifikasi balita penderita *stunting* di Kabupaten Lampung Barat.
2. Mengukur dan mengevaluasi hasil kinerja metode *support vector machine* dan *xtreme gradient boosting*.

#### **1.5 Manfaat**

Berdasarkan latar belakang yang telah dijabarkan diatas, manfaat pada penelitian ini adalah sebagai berikut.

1. Mengetahui evaluasi kinerja metode *support vector machine* dan *xtreme gradient boosting*.
2. Mengetahui perbandingan hasil analisa model *support vector machine* dan *xtreme gradient boosting* dalam klasifikasi *stunting* di Kabupaten Lampung Barat berdasarkan data antropometri.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Pada penelitian ini, dibutuhkan penelitian terdahulu sebagai berikut:

Tabel 1. Penelitian Terdahulu

No	Peneliti	Judul	Objek	Metode	Hasil
1.	(Nababan, 2021)	Klasifikasi Penderita <i>Stunting</i> dengan Metode <i>Support vector machine</i> (Studi Kasus: Lima Puskesmas di Kota Bandar Lampung)	Balitas	<i>Support vector machine</i>	Hasil rata-rata <i>accuracy</i> terbesar dengan <i>kernel</i> linear
2.	(Lumbanraja et al., 2020)	Prediksi Jumlah Penderita Penyakit Tuberkulosis di Kota Bandar Lampung Menggunakan Metode SVM	Penyakit tuberkulosis	<i>Support vector machine</i>	$R^2$ terbesar pada <i>kernel</i> gaussian

3.	(Lumbanraja et al., 2022)	Implementasi <i>Support vector machine</i> (SVM) untuk Klasifikasi Penderita Diabetes Mellitus	Penderita Diabetes Mellitus	<i>Support vector machine</i>	Hasil <i>accuracy</i> dengan <i>kernel gaussian</i>	rata-rata terbesar
4.	(Rahman, 2020)	Implementasi Metode SVM, MLP dan <i>XGBoost</i> pada Data Ekspresi Gen	Data ekspresi Gen.	<i>Support vector machine, Multilayer Perceptron</i> dan <i>Xtreme gradient boosting</i>	Hasil dengan <i>Support Machine</i>	terbaik metode <i>Vector</i>
5.	(Andryan et al., 2022)	Komparasi Kinerja Algoritma <i>XGBoost</i> dan Algoritma <i>Support vector machine</i> (SVM) untuk Diagnosa Penyakit Kanker Payudara	Kanker payudara.	<i>Xtreme gradient boosting</i> dan <i>Support vector machine.</i>	Hasil dengan <i>XGBoost</i>	terbaik metode
6.	Penelitian yang akan dilakukan	Klasifikasi <i>Stunting</i> di Kabupaten Lampung Barat Berdasarkan Data Antropometri Menggunakan Analisis Algoritma <i>Support Vector Machine</i> dan <i>Xtreme Gradient Boosting</i>	<i>Stunting</i> di Kabupaten Lampung Barat	<i>Support vector machine, Xtreme gradient boosting</i>	Penelitian ini akan membahas tentang klasifikasi <i>stunting</i> di Kabupaten Lampung Barat dengan menggunakan metode SVM dan <i>XGBoost</i>	

- A. Penelitian oleh (Nababan, 2021) mengenai hasil kinerja metode *support vector machine* dalam mengklasifikasikan penderita *stunting* menggunakan data dari lima puskesmas di Kota Bandar Lampung. Data yang digunakan berupa 234 data sampel balita pada tahun 2019 (89 data kejadian *stunting* & 145 data dengan kejadian normal) dengan menggunakan 7 atribut yaitu jenis kelamin, usia, berat badan, panjang badan, berat badan lahir, panjang badan lahir, pemberian asi, serta satu kelas yaitu *stunting*. Hasil penelitian ini menunjukkan bahwa hasil terbesar menggunakan *kernel* linear yaitu sebesar 80.8%, sedangkan *kernel* gaussian 80,4% dan *kernel* polynomial sebesar 69,1%.
- B. Penelitian (R Lumbanraja et al., 2020) mengenai penderita tuberculosis dengan menggunakan metode *support vector machine*. Data yang digunakan berupa data tuberculosis di Kota Bandar Lampung pada tahun 2015 sampai tahun 2018 dan data cuaca yang didapatkan dari website Badan Pusat Statistik (BPS) dan matriks jarak antara lokasi penderita yang satu dengan penderita lainnya berdasarkan kecamatan dan letak *latirude* dan *longitude*. Hasil penelitian ini didapatkan  $R^2$  terbesar pada *kernel* gaussian yaitu sebesar 46.7% dengan tanpa menggunakan *feature selection* dan tanpa matriks jarak.
- C. Penelitian oleh (Lumbanraja et al., 2022) mengenai penderita diabetes mellitus dengan menggunakan metode *support vector machine*. Dataset yang digunakan pada penelitian ini adalah data dari *Diabetes 130-US Hospitals For Years 1999-2008* Dataset yang didapatkan dari situs *UCI Machine Learning Repository*. Dataset terdiri dari 101766 record dengan 1 label atau kelas dan 34 atribut lainnya. Pada penelitian ini didapatkan hasil rata-rata *accuracy* terbesar diperoleh dari *kernel* gaussian yaitu sebesar 82.76%.
- D. Penelitian oleh (Rahman, 2020) mengenai implementasi ekspresi gen dengan menggunakan tiga metode yaitu metode *support vector machine*, *multilayer perceptron* dan *xtreme gradient boosting*. Hasil dari penelitian ini didapatkan bahwa hasil terbaik didapatkan dengan menggunakan metode *support vector machine* yaitu sebesar 91.30% sedangkan dengan *multilayer*

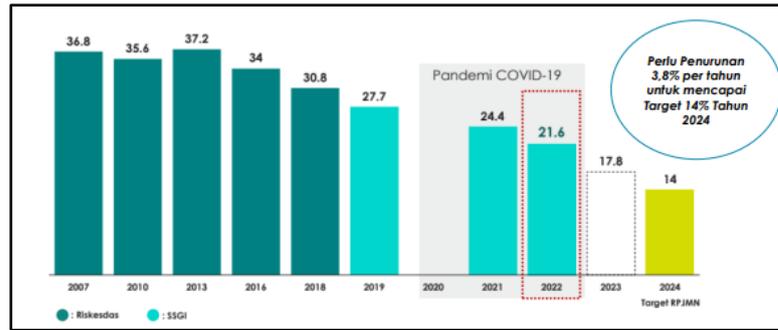
*perceptron* didapatkan hasil sebesar 78,26% dan pada *xtreme gradient boosting* didapatkan hasil sebesar 73,91%.

- E. Penelitian oleh (Andryan et al., 2022) mengenai diagnosa kanker payudara dengan menggunakan dua metode yaitu *support vector machine* dan *xtreme gradient boosting*. Data diperoleh dari data public yang berasal dari *Kaggle dataset Wisconsin Machine Breast Cancer Diagnostic* yang berisikan rata-rata, standard error dan “worst” atau terbesar (rata-rata dari lima nilai terbesar). Hasil penelitian dengan menggunakan dua metode tersebut mendapatkan hasil terbesar dengan menggunakan model *Xtreme gradient boosting* dengan nilai *accuracy* sebesar 82%, *recall* 70%, dan *precision* 92%.

## 2.2 Landasan Teori

### 2.2.1 Stunting

*Stunting* merupakan masalah gizi utama yang dihadapi Indonesia yang belum teratasi (Kemenkes RI, 2022). *Stunting* adalah kondisi gagal tumbuh pada bayi yang menggambarkan status gizi kurang pada masa pertumbuhan sejak awal kehidupan. *Stunting* ditandai dengan panjang atau tinggi tubuh yang tidak sesuai dengan anak seusianya dan diakibatkan oleh kurangnya gizi pada anak terutama pada periode 1000 hari pertama pertama yaitu sejak bayi dalam kandungan sampai usia kurang lebih 2 tahun. Seseorang yang terkena *stunting* akan cenderung mengalami perkembangan fisik yang juga terhambat dan akan berpengaruh pada tingkat kecerdasan bahkan produktivitasnya dimasa yang akan datang. Banyak faktor yang menyebabkan seseorang beresiko terkena *stunting* seperti kekurangan vitamin A, mengidap tuberculosis (TBC), buruknya sanitasi dan lain sebagainya (Fadilah et al., 2022). Hal ini biasa terjadi pada masa balita, balita adalah pengertian umum dari anak yang berusia 1-3 tahun (batita) dan anak prasekolah yang berumur 3-5 tahun, pada masa ini proses tumbuh kembang manusia sangat penting untuk menjadi penentu keberhasilan tumbuh kembang anak di masa yang akan datang (Sulut, 2017).



Sumber : (SSGI, 2022)

Gambar 1. Penurunan angka *stunting*

Penurunan angka *stunting* pada tahun 2022 ini disebabkan oleh peningkatan sumber gizi yaitu peningkatan IMD (Inisiasi Menyusu Dini), pemberian ASI, sumber protein hewani dan konseling gizi.

### 2.2.2 Antropometri

Menurut Widgnjosoebroto, antropometri berasal dari bahasa Yunani yaitu *Anthropos (man)* yang berarti manusia dan *metreinn (to measure)* yang berarti ukuran, sehingga dapat disimpulkan bahwa antropometri adalah ilmu yang berhubungan dengan pengukuran dimensi tubuh. Berdasarkan pengukurannya, antropometri dibagi menjadi dua macam, yaitu antropometri statis atau yang berhubungan dengan pengukuran tubuh manusia dalam posisi diam dan antropometri dinamis atau yang berhubungan dengan pengukuran tubuh manusia dalam posisi bergerak (Pattiasina et al., 2022).

Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia Nomor 2 Tahun 2020 tentang Standar Antropometri Anak menyatakan bahwa standar antropometri anak didasarkan pada parameter berat badan dan panjang/tinggi badan yang terdiri atas empat indeks yaitu, indeks berat badan menurut umur (BB/U), indeks panjang badan menurut umur atau tinggi badan menurut umur (PB/U atau TB/U), indeks berat badan menurut panjang badan/tinggi badan (BB/PB atau BB/TB) dan indeks masa tubuh menurut umur (IMT/U). Untuk *stunting* dapat diidentifikasi menggunakan indeks PB/U atau TB/U, indeks ini menggambarkan pertumbuhan panjang atau tinggi badan anak berdasarkan umurnya yang disebabkan oleh gizi kurang dalam waktu lama. Anak dengan PB/U atau TB/U di bawah minus dua standar deviasi merupakan anak dengan

perawakan pendek yang wajib ditindaklanjuti dengan tatalaksana *stunting* dan dirujuk (Kemenkes, 2020).

Menurut peraturan Menteri Kesehatan Republik Indonesia Nomor 2 Tahun 2020 tentang Standar Antropometri Anak, untuk indeks panjang badan atau tinggi badan menurut umur (anak usia 0-60 bulan) terdapat 4 kategori status gizi, yaitu:

Tabel 2. Kategori dan Ambang Batas Status Gizi Anak

Kategori Status Gizi	Ambang Batas (Z-Score)
Sangat Pendek ( <i>severely stunted</i> )	$< -3 SD$
Pendek ( <i>stunted</i> )	$-3 SD \text{ sd } < -2 SD$
Normal	$-2 SD \text{ sd } +3 SD$
Tinggi	$> +3 SD$

Tabel 3. Standar PB/U Anak Perempuan Umur 0-60 Bulan

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
0	43.6	45.4	47.3	49.1	51.0	52.9	54.7
1	47.8	49.8	51.7	53.7	55.6	57.6	59.5
2	51.0	53.0	55.0	57.1	59.1	61.1	63.2
3	53.5	55.6	57.7	59.8	61.9	64.0	66.1
4	55.6	57.8	59.9	62.1	64.3	66.4	68.6
5	57.4	59.6	61.8	64.0	66.2	68.5	70.7
6	58.9	61.2	63.5	65.7	68.0	70.3	72.5
7	60.3	62.7	65.0	67.3	69.6	71.9	74.2
8	61.7	64.0	66.4	68.7	71.1	73.5	75.8
9	62.9	65.3	67.7	70.1	72.6	75.0	77.4
10	64.1	66.5	69.0	71.5	73.9	76.4	78.9
11	65.2	67.7	70.3	72.8	75.3	77.8	80.3
12	66.3	68.9	71.4	74.0	76.6	79.2	81.7
13	67.3	70.0	72.6	75.2	77.8	80.5	83.1
14	68.3	71.0	73.7	76.4	79.1	81.7	84.4
15	69.3	72.0	74.8	77.5	80.2	83.0	85.7
16	70.2	73.0	75.8	78.6	81.4	84.2	87.0
17	71.1	74.0	76.8	79.7	82.5	85.4	88.2
18	72.0	74.9	77.8	80.7	83.6	86.5	89.4

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
19	72.8	75.8	78.8	81.7	84.7	87.6	90.6
20	73.7	76.7	79.7	82.7	85.7	88.7	91.7
21	74.5	77.5	80.6	83.7	86.7	89.8	92.9
22	75.2	78.4	81.5	84.6	87.7	90.8	94.0
23	76.0	79.2	82.3	85.5	88.7	91.9	95.0
24	76.0	79.3	82.5	85.7	88.9	92.2	95.4
25	76.8	80.0	83.3	86.6	89.9	93.1	96.4
26	77.5	80.8	84.1	87.4	90.8	94.1	97.4
27	78.1	81.5	84.9	88.3	91.7	95.0	98.4
28	78.8	82.2	85.7	89.1	92.5	96.0	99.4
29	79.5	82.9	86.4	89.9	93.4	96.9	100.3
30	80.1	83.6	87.1	90.7	94.2	97.7	101.3
31	80.7	84.3	87.9	91.4	95.0	98.6	102.2
32	81.3	84.9	88.6	92.2	95.8	99.4	103.1
33	81.9	85.6	89.3	92.9	96.6	100.3	103.9
34	82.5	86.2	89.9	93.6	97.4	101.1	104.8
35	83.1	86.8	90.6	94.4	98.1	101.9	105.6
36	83.6	87.4	91.2	95.1	98.9	102.7	106.5
37	84.2	88.0	91.9	95.7	99.6	103.4	107.3
38	84.7	88.6	92.5	96.4	100.3	104.2	108.1
39	85.3	89.2	93.1	97.1	101.0	105.0	108.9
40	85.8	89.8	93.8	97.7	101.7	105.7	109.7
41	86.3	90.4	94.4	98.4	102.4	106.4	110.5
42	86.8	90.9	95.0	99.0	103.1	107.2	111.2
43	87.4	91.5	95.6	99.7	103.8	107.9	112.0
44	87.9	92.0	96.2	100.3	104.5	108.6	112.7
45	88.4	92.5	96.7	100.9	105.1	109.3	113.5
46	88.9	93.1	97.3	101.5	105.8	110.0	114.2
47	89.3	93.6	97.9	102.1	106.4	110.7	114.9
48	89.8	94.1	98.4	102.7	107.0	111.3	115.7
49	90.3	94.6	99.0	103.3	107.7	112.0	116.4
50	90.7	95.1	99.5	103.9	108.3	112.7	117.1
51	91.2	95.6	100.1	104.5	108.9	113.3	117.7
52	91.7	96.1	100.6	105.0	109.5	114.0	118.4
53	92.1	96.6	101.1	105.6	110.1	114.6	119.1

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
54	92.6	97.1	101.6	106.2	110.7	115.2	119.8
55	93.0	97.6	102.2	106.7	111.3	115.9	120.4
56	93.4	98.1	102.7	107.3	111.9	116.5	121.1
57	93.9	98.5	103.2	107.8	112.5	117.1	121.8
58	94.3	99.0	103.7	108.4	113.0	117.7	122.4
59	94.7	99.5	104.2	108.9	113.6	118.3	123.1
60	95.2	99.9	104.7	109.4	114.2	118.9	123.7

Tabel 4. Standar PB/U Anak Laki-Laki Umur 0-60 Bulan

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
0	44.2	46.1	48.0	49.9	51.8	53.7	55.6
1	48.9	50.8	52.8	54.7	56.7	58.6	60.6
2	52.4	54.4	56.4	58.4	60.4	62.4	64.4
3	55.3	57.3	59.4	61.4	63.5	65.5	67.6
4	57.6	59.7	61.8	63.9	66.0	68.0	70.1
5	59.6	61.7	63.8	65.9	68.0	70.1	72.2
6	61.2	63.3	65.5	67.6	69.8	71.9	74.0
7	62.7	64.8	67.0	69.2	71.3	73.5	75.7
8	64.0	66.2	68.4	70.6	72.8	75.0	77.2
9	65.2	67.5	69.7	72.0	74.2	76.5	78.7
10	66.4	68.7	71.0	73.3	75.6	77.9	80.1
11	67.6	69.9	72.2	74.5	76.9	79.2	81.5
12	68.6	71.0	73.4	75.7	78.1	80.5	82.9
13	69.6	72.1	74.5	76.9	79.3	81.8	84.2
14	70.6	73.1	75.6	78.0	80.5	83.0	85.5
15	71.6	74.1	76.6	79.1	81.7	84.2	86.7
16	72.5	75.0	77.6	80.2	82.8	85.4	88.0
17	73.3	76.0	78.6	81.2	83.9	86.5	89.2
18	74.2	76.9	79.6	82.3	85.0	87.7	90.4
19	75.0	77.7	80.5	83.2	86.0	88.8	91.5
20	75.8	78.6	81.4	84.2	87.0	89.8	92.6
21	76.5	79.4	82.3	85.1	88.0	90.9	93.8
22	77.2	80.2	83.1	86.0	89.0	91.9	94.9

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
23	78.0	81.0	83.9	86.9	89.9	92.9	95.9
24	78.0	81.0	84.1	87.1	90.2	93.2	96.3
25	78.6	81.7	84.9	88.0	91.1	94.2	97.3
26	79.3	82.5	85.6	88.8	92.0	95.2	98.3
27	79.9	83.1	86.4	89.6	92.9	96.1	99.3
28	80.5	83.8	87.1	90.4	93.7	97.0	100.3
29	81.1	84.5	87.8	91.2	94.5	97.9	101.2
30	81.7	85.1	88.5	91.9	95.3	98.7	102.1
31	82.3	85.7	89.2	92.7	96.1	99.6	103.0
32	82.8	86.4	89.9	93.4	96.9	100.4	103.9
33	83.4	86.9	90.5	94.1	97.6	101.2	104.8
34	83.9	87.5	91.1	94.8	98.4	102.0	105.6
35	84.4	88.1	91.8	95.4	99.1	102.7	106.4
36	85.0	88.7	92.4	96.1	99.8	103.5	107.2
37	85.5	89.2	93.0	96.7	100.5	104.2	108.0
38	86.0	89.8	93.6	97.4	101.2	105.0	108.8
39	86.5	90.3	94.2	98.0	101.8	105.7	109.5
40	87.0	90.9	94.7	98.6	102.5	106.4	110.3
41	87.5	91.4	95.3	99.2	103.2	107.1	111.0
42	88.0	91.9	95.9	99.9	103.8	107.8	111.7
43	88.4	92.4	96.4	100.4	104.5	108.5	112.5
44	88.9	93.0	97.0	101.0	105.1	109.1	113.2
45	89.4	93.5	97.5	101.6	105.7	109.8	113.9
46	89.8	94.0	98.1	102.2	106.3	110.4	114.6
47	90.3	94.4	98.6	102.8	106.9	111.1	115.2
48	90.7	94.9	99.1	103.3	107.5	111.7	115.9
49	91.2	95.4	99.7	103.9	108.1	112.4	116.6
50	91.6	95.9	100.2	104.4	108.7	113.0	117.3
51	92.1	96.4	100.7	105.0	109.3	113.6	117.9
52	92.5	96.9	101.2	105.6	109.9	114.2	118.6
53	93.0	97.4	101.7	106.1	110.5	114.9	119.2
54	93.4	97.8	102.3	106.7	111.1	115.5	119.9
55	93.9	98.3	102.8	107.2	111.7	116.1	120.6
56	94.3	98.8	103.3	107.8	112.3	116.7	121.2
57	94.7	99.3	103.8	108.3	112.8	117.4	121.9

Umur (bulan)	Panjang Badan (cm)						
	-3SD	-2SD	-1SD	Median	+1SD	+2SD	+3SD
58	95.2	99.7	104.3	108.9	113.4	118.0	122.6
59	95.6	100.2	104.8	109.4	114.0	118.6	123.2
60	96.1	100.7	105.3	110.0	114.6	119.2	123.9

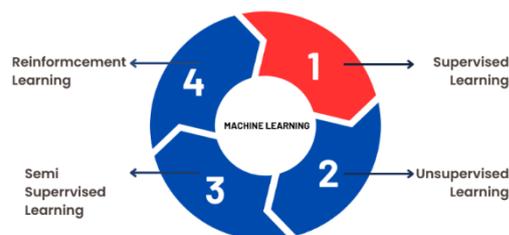
### 2.2.3 Klasifikasi

Menurut Sulistyio Basuki pada tahun 1991 klasifikasi berasal dari kata latin *classis* yang berarti pengelompokan atau pembagian. Jadi klasifikasi adalah pengelompokan benda atau objek yang sama serta memisahkan yang tidak sama untuk mendapatkan suatu keputusan. Klasifikasi merupakan suatu teknik yang dilakukan dengan tujuan untuk memudahkan pencarian suatu objek karna objeknya sudah dikelompokkan terlebih dahulu (Agustina et al., 2018).

### 2.2.4 Machine Learning

*Machine learning* menurut Samuel adalah sebuah cabang ilmu komputer yang mempelajari sebuah metode perancangan algoritma yang mampu beradaptasi terhadap pola data tanpa di program secara eksplisit. Pada tahun 2017 *expert system* mendefinisikan bahwa *machine learning* merupakan bagian dari kecerdasan buatan yang mampu belajar tanpa diprogram sehingga dapat meningkatkan kemampuan dari pengalaman sebelumnya. *Machine learning* secara operasional dapat diterapkan kedalam sebuah algoritma pembelajaran untuk menghasilkan model *machine learning* (Saputro, 2023).

Berdasarkan cara belajarnya, *machine learning* dibagi menjadi empat kategori utama yaitu:



Gambar 2. Kategori machine learning berdasarkan cara belajarnya

A. *Supervised Learning* (Pembelajaran Terstruktur)

*Supervised learning* adalah metode pembelajaran yang memiliki tujuan untuk prediksi variabel target menggunakan data baru dengan data lain pada lokasi terdekat (Hartawan et al., 2023). *Supervised learning* umumnya digunakan untuk menyelesaikan dua permasalahan yaitu klasifikasi (bertujuan mengklasifikasikan data baru dengan akurat) dan regresi (bertujuan memprediksi).

B. *Unsupervised Learning* (Pembelajaran Tidak Terstruktur)

*Unsupervised learning* adalah metode pembelajaran yang dilakukan menggunakan sejumlah data tanpa label. *Unsupervised learning* umumnya digunakan untuk menyelesaikan dua permasalahan yaitu *clustering* dan pengurangan dimensi data.

C. *Semi Supervised Learning* (Pembelajaran Semi Terstruktur)

*Semi supervised learning* adalah metode pembelajaran yang berdasarkan pada data yang berlabel dan tidak berlabel.

D. *Reinforcement Learning*

*Reinforcement Learning* adalah metode pembelajaran yang menggunakan beberapa data tidak berlabel yang berdasarkan pada pemberian penghargaan pada keputusan yang diinginkan dan pemberian tindakan pada keputusan yang tidak diinginkan.

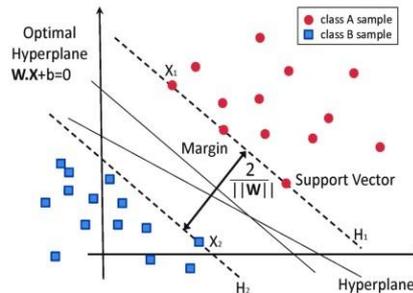
### **2.2.5 K-Fold Cross Validation**

*Cross Validation* adalah teknik yang digunakan untuk mengevaluasi kinerja model yang melibatkan pemecahan data menjadi beberapa subnet, dan model tersebut akan dilatih dan diuji menggunakan subnet tersebut secara bergantian. Pada *K-Fold Cross Validation*, dataset akan dibagi secara acak dengan ukuran yang sama menjadi k subset, model dilatih dan diuji sebanyak k kali yang dilakukan berulang-ulang (Tapikap et al., 2019).

### **2.2.6 Support Vector Machine**

Menurut Vapnik (1992) *Support vector machine* (SVM) adalah salah satu algoritma *machine learning* yang memiliki prinsip *Structural Risk Minimization*

(SRM) dengan tujuan mendapatkan pemisah dua buah *class* pada *input space* atau yang biasa disebut dengan *hyperplane* (Mudyaningasih, 2019).



Sumber : (Veronica & Tumanggor, 2022)

Gambar 3. *Hyperplane* terbaik yang memisahkan dua class

Adapun persamaannya adalah sebagai berikut:

$$\frac{1}{2} ||W||^2 = \frac{1}{2} (W_1^2 + W_2^2)$$

Dengan syarat:

$$Y_i (W \cdot X_1 + b) \geq 1$$

$$Y_i (W_1 \cdot X_1 + W_2 \cdot X_2 + b) \geq 1$$

Keterangan :

$X_i$  = Data ke -i

$W$  = Nilai bobot SV yang tegak lurus dengan *hyperplane*

$b$  = Nilai bias

$Y_i$  = *Class* data ke -i

Nilai bobot *Support Vector* (SV) adalah garis *vector* yang tegak lurus antara titik pusat kordinat dengan garis *hyperplane*, sedangkan nilai bias adalah kordinat garis relatif terhadap titik kordinat.

Menurut Liu, Shen & Wang (2014) terdapat empat *kernel* pada SVM, sebagaimana persamaan-persamaan berikut:

A. *Kernel* linear

$$K (X_i, X_j) = X_i \cdot X_j$$

B. *Kernel* polynomial

$$K (X_i, X_j) = (X_i \cdot X_j + 1)^h$$

C. *Kernel* gaussian

$$K (X_i, X_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

D. *Kernel* sigmoid

$$K (X_i, X_j) = \tanh (kX_i \cdot X_j + \beta)$$

Keterangan :

$X$  = Data ke- $i$

$K$  = Parameter kernel

$h$  = Derajat polynomial

$\beta$  = Bias

$\sigma$  = Lebar gaussian

Tingkat akurasi SVM sangat bergantung pada fungsi *kernel* dan parameter yang digunakan (Monika & Furqon, 2018).

Selain kernel, terdapat 3 parameter yang dapat digunakan dalam metode support vector machine untuk mendapatkan akurasi terbaik, yaitu

A. *Cost*

*Cost* akan digunakan untuk menentukan kemampuan model dalam mengelola kesalahan dalam klasifikasi (Pratiwi & Setyawan, 2021). Semakin besar nilai *cost* mengakibatkan model akan lebih sensitif terhadap kesalahan klasifikasi positif dan tidak sensitif terhadap kesalahan klasifikasi negatif.

### B. *Gamma*

*Gamma* digunakan untuk mengatur jarak antara dua titik dalam ruang efektif. Jika nilai *gamma* terlalu besar akan menyebabkan kurangnya ketelitian pada akurasi yang dihasilkan dan sebaliknya, sehingga nilai *gamma* harus disesuaikan untuk mencari nilai yang optimal untuk digunakan kedalam model sehingga akan meningkatkan kinerja dari model (Wijayanti et al., 2018).

### C. Iterasi Maksimum

Iterasi maksimum (*max iter*) digunakan untuk mengatur nilai iterasi yang dilakukan oleh suatu model algoritma. Semakin besar nilai *max iter* yang digunakan akan mengakibatkan waktu komputasi akan semakin lama, sehingga perlu didapatkan nilai *max iter* yang paling tepat sehingga kinerja model akan lebih baik tetapi waktu komputasi tidak terlalu lama.

## 2.2.7 *Xtreme Gradient Boosting*

Metode *Xtreme Gradient Boosting* (XGBoost) pertama kali dikenalkan oleh Friedman yang digunakan untuk klasifikasi dan regresi. XGBoost adalah metode lanjutan *gradient boosting* yang merupakan metode *ensemble* dari model yang digunakan pada metode *decision tree* yang dikembangkan dengan tujuan mendapatkan hasil *running time* yang lebih cepat dengan data yang lebih besar (Irawan, 2023). Algoritma ini memungkinkan melakukan optimasi 10 kali lebih cepat dibandingkan dengan *gradient boosting* lainnya (Givari et al., 2022).

Fungsi objektif digunakan untuk mengukur seberapa baik model tersebut sesuai dengan data latih (Hanif, 2019). Terdapat 2 bagian penting dalam *objective function* yaitu *training loss* yang digunakan untuk mengukur seberapa prediktif model tersebut sehubungan dengan data latih dan *regularization term* yang digunakan untuk mengontrol kompleksitas model dan membantu untuk menghindari overfitting (keadaan ketika model *machine learning* terlalu kompleks sehingga tidak dapat mempelajari pola latih yang mengakibatkan tidak dapat menggeneralisasi dengan baik data uji) (Rombe, 2021)

Nilai akurasi *XGBoost* tergantung pada parameter yang digunakan. Berikut parameter yang dapat digunakan untuk mendapatkan hasil akurasi terbaik.

Tabel 5. Parameter pada *XGBoost*

Parameter	Keterangan
<i>Eta</i>	<i>Learning rate</i> pada proses pelatihan
<i>Max_depth</i>	Tingkat kedalaman suatu pohon, semakin dalam pohon maka akan semakin kompleks
<i>Gamma</i>	Parameter penalty pada <i>regularization</i>
<i>Min_child_weight</i>	Nilai minimal bobot yang dibutuhkan <i>child node</i>
<i>Subsample</i>	Jumlah sampel yang digunakan untuk proses pelatihan. Misal 0.5 berarti menggunakan setengah dari data acak dalam membuat <i>tree</i> baru
<i>Colsample_bytree</i>	Jumlah sampel kolom untuk membuat <i>tree</i> baru

#### A. *Learning rate* (eta)

*Learning rate* memiliki fungsi untuk menentukan besar langkah yang diambil pada masing-masing iterasi dalam model. Semakin besar nilai eta yang digunakan maka proses pembelajaran dari sebuah model akan semakin cepat, hal ini dapat menyebabkan model melewati titik titik tertentu sehingga tidak mengeksplorasi sekitarnya secara mendetail, sedangkan nilai eta yang kecil akan mengakibatkan langkahnya semakin kecil dan eksplorasi nya akan lebih mendetail tetapi akan membutuhkan waktu komputasi yang lebih banyak (Saputri, 2021). Sehingga dibutuhkan nilai eta yang tepat untuk mendapatkan hasil kinerja terbaik dari model.

#### B. *Max depth*

*Max depth* pada *XGBoost* digunakan untuk menentukan batas kedalaman pohon dalam pemodelan (Shafila, 2020). Parameter ini digunakan untuk mengendalikan kompleksitas algoritma agar tidak terjadi overfitting (Saputri, 2021).

### C. *Gamma*

*Gamma* pada *XGBoost* adalah parameter yang menentukan batas minimum reduksi kehilangan yang dibutuhkan dalam membagi node pohon dalam pemodelan. Nilai *gamma* berkisar antara 0 sampai 1 (Syukron et al., 2020).

### D. *Min child weight*

*Min child weight* pada *XGBoost* digunakan untuk menentukan jumlah minimum bobot yang akan digunakan dalam masing-masing tree (Sudarman & Budi, 2023). Semakin besar nilai *min child weight* yang digunakan maka model akan lebih konservatif.

### E. *Subsample*

*Subsample* pada *XGBoost* digunakan untuk mengatur jumlah data yang akan digunakan dalam masing-masing pohon untuk pemodelan (Rachmi, 2020). Jika *subsample* yang digunakan adalah sebesar 0,75 maka pohon pertama akan dibentuk dari 75 persen data, dan 25 persen data akan digunakan untuk pohon sisanya, masing-masing pohon kedua akan berfokus pada penghilangan kesalahan yang dibuat pada pohon pertama dan seterusnya.

### F. *Colsample bytree*

*Colsample bytree* pada *XGBoost* merupakan sebuah parameter yang digunakan untuk mengatur jumlah persentase kolom yang digunakan untuk membangun masing-masing pohon dalam pemodelan (Agustin et al., 2023). *Colsample bytree* memiliki nilai antara 0 sampai 1, dan nilai defaultnya adalah 1. Jika *colsample bytree* yang digunakan adalah 0, maka algoritma akan menggunakan beberapa fitur dalam jumlah kecil untuk membangun masing-masing pohon, sedangkan jika *colsample bytree* diatur ke angka 1, maka algoritma akan menggunakan semua fitur untuk membangun masing-masing pohon. Penggunaan nilai yang lebih kecil dari 1 akan menyebabkan pengambilan fitur yang lebih rendah dan akan mengurangi *overfitting* terutama dalam dataset yang memiliki banyak fitur.

### 2.2.8 Confussion Matrix (Matriks Konfusi)

*Confussion matrix* atau matriks konfusi digunakan untuk mengukur hasil kinerja dari suatu metode klasifikasi. Dengan penjelasan matriks konfusi pada tabel 6.

Tabel 6. Matriks Konfusi

MATRIKS KONFUSI		PREDIKSI KELAS	
		Positif	Negatif
AKTUAL KELAS	Positif	TP	FN
	Negatif	FP	TN

Keterangan:

- TP : *True Positive* (jika hasil prediksi positif dan data sebenarnya positif)
- FP : *False Positive* (jika hasil prediksi positif dan data sebenarnya negatif)
- TN : *True Negative* (jika hasil prediksi negatif dan data sebenarnya negatif)
- FN : *False Negative* (jika hasil prediksi negatif dan data sebenarnya positif)

Setelah mengetahui nilai TP, FP, FN dan keseluruhan dataset, selanjutnya adalah mencari nilai dari confusion matrix. Adapun beberapa nilai yang dapat diperoleh dari matrix yang akan digunakan untuk menghitung kinerja model klasifikasi adalah sebagai berikut:

#### A. Accuracy

*Accuracy* ialah nilai ketepatan model dalam melakukan prediksi data dengan perbandingan data aktualnya dan sebagai pengukur model untuk menentukan keakuratan prediksi. *Accuracy* yang akan digunakan adalah akurasi keseluruhan dari sejumlah k pada fold yang digunakan, sehingga persamaannya adalah sebagai berikut:

$$Accuracy = \frac{TP\ 0 + TP\ 1 + TP\ 2 + TP\ 3}{Keseluruhan\ Dataset}$$

### B. *Precision*

*Precision* digunakan untuk melihat tingkat kecocokan data untuk keperluan prediksi. Adapun persamaannya adalah sebagai berikut:

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

Selanjutnya adalah rumus untuk mencari nilai dari keseluruhan *precision* adalah sebagai berikut:

$$Overall\ Precision = \frac{Precision\ 0 + Precision\ 1 + Precision\ 2 + Precision\ 3}{Jumlah\ Kelas}$$

### C. *Recall*

*Recall* digunakan untuk mengevaluasi seberapa besar keberhasilan suatu model dalam memprediksi kelas positif yang diklasifikasikan. Adapun persamaannya dapat dilihat sebagai berikut:

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

Selanjutnya adalah rumus untuk mencari nilai dari keseluruhan *recall* adalah sebagai berikut:

$$Overall\ Recall = \frac{Recall\ 0 + Recall\ 1 + Recall\ 2 + Recall\ 3}{Jumlah\ Kelas}$$

### D. *F-1 Score*

*F-1 Score* adalah nilai rata-rata dari *precision* dan *recall*, nilai terbaik untuk *F-1 Score* bernilai 1 dan nilai terburuknya bernilai 0.

Persamaannya dapat dilihat sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

### **III. METODOLOGI PENELITIAN**

#### **3.1 Tempat dan Waktu**

##### **3.1.1 Tempat**

###### **A. Tempat Pengumpulan Data**

Data pada penelitian ini diperoleh dari Dinas Kesehatan Kabupaten Lampung Barat.

###### **B. Tempat Penelitian**

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang terletak di Jalan Soemantri Brojonegoro No. 1 Gedung Meneng, Bandar Lampung.

##### **3.1.2 Waktu**

Penelitian ini dilaksanakan mulai bulan November 2023 hingga bulan Mei 2024. Rincian waktu penelitian dapat dilihat pada tabel 7 berikut:

Tabel 7. Waktu Penelitian

NO	Jadwal Kegiatan	BULAN PELAKSANAAN PENELITIAN 2023/2024						
		November	Desember	Januari	Februari	Maret	April	Mei
1.	Studi Literatur dan Penulisan Laporan Bab 1-3							
2.	Pengumpulan Data							
3.	Seminar Usul							
4.	<i>Pre-Processing</i> Data							
5.	Pembagian Data <i>Training</i> dan Data <i>Testing</i>							
6.	Model SVM & <i>XGBoost</i>							
7.	Pengujian							
8.	Perbandingan Hasil Klasifikasi							
9.	Penulisan Laporan Bab 4-5							
10.	Seminar Hasil Penelitian							

## 3.2 Alat Pendukung

### 3.2.1 Perangkat Keras

- A. Processor: Intel(R) Core(TM) i7-11800H
- B. Memori: 512 GB Solid State Drive, 16 GB SODIMM DDR4 SDRAM
- C. System Type: 64-bit operating system

### 3.2.1 Perangkat Lunak

- A. Sistem Operasi: Windows 11
- B. Jupyter Notebook version 7.1.0  
Pada penelitian ini menggunakan bahasa pemrograman python version 3.8.8.  
Adapun *packages* yang digunakan, yaitu:
  - 1. Pandas *version* 1.2.4
  - 2. Numpy *version* 1.20.1
  - 3. Scikit-learn *version* 0.24.1
  - 4. Matplotlib *version* 3.3.4
  - 5. Imbalanced-learn *version* 0.12.0
  - 6. Xgboost *version* 2.0.3
  - 7. Scikit-optimize *version* 0.9.0
  - 8. Jcopml *version* 1.2.4
  - 9. Time
- C. Microsoft Excel 2016
- D. Microsoft Word 2016
- E. Draw.io

## 3.3 Data

Data yang didapatkan pada penelitian ini sebanyak 18.372 record data balita yang terdapat di Kabupaten Lampung Barat pada tahun 2023 bulan timbang Oktober. Data ini diperoleh dari Dinas Kesehatan Kabupaten Lampung Barat. Adapun detail record data yang didapatkan dapat dilihat pada tabel berikut:

Tabel 8. Jumlah Data *Stunting* Per Kelas

No	Kelas	Jumlah Record data	Jumlah Record data Perempuan	Jumlah Record data Laki-Laki
1	Normal	17.752	8.156	9.596
2	Pendek	451	192	259
3	Sangat Pendek	169	59	110

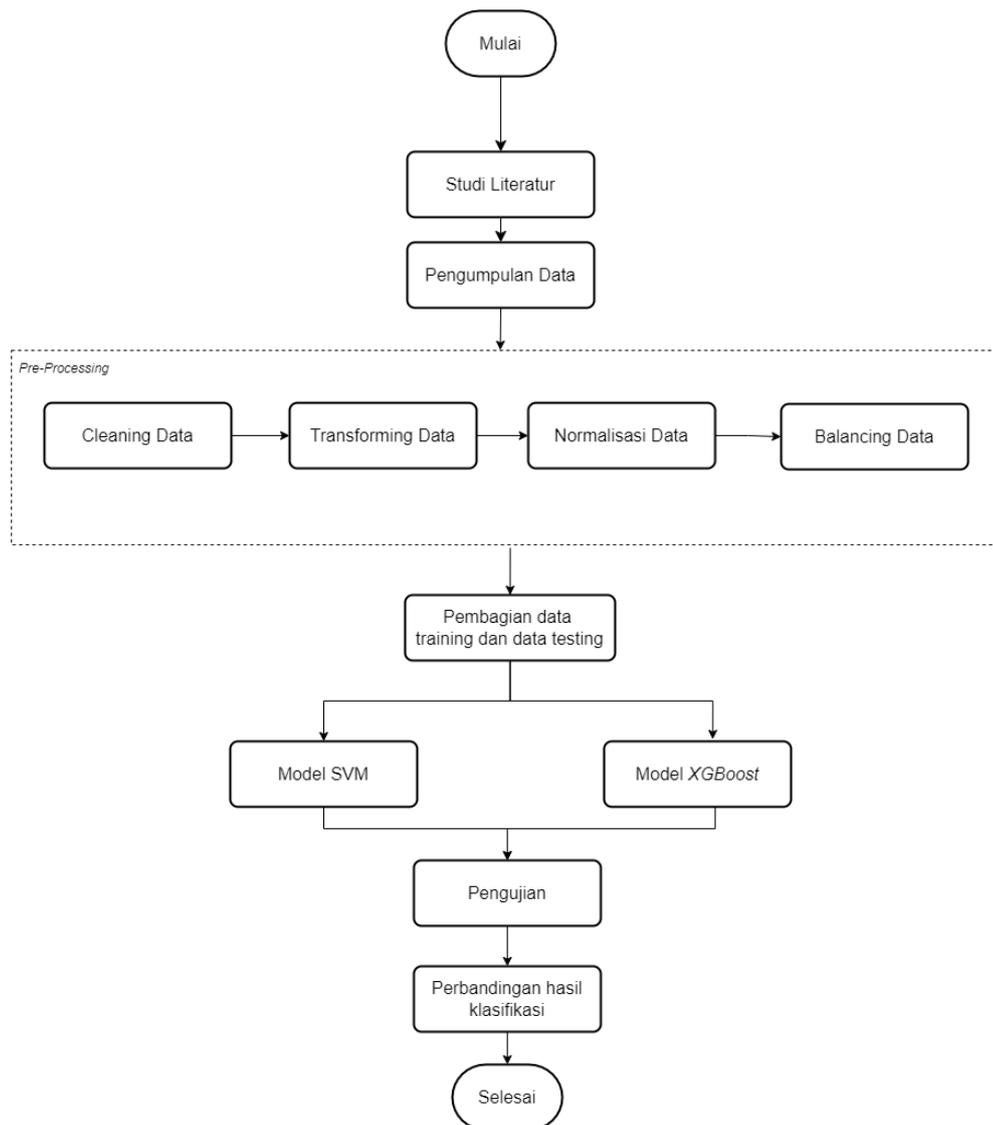
Data yang didapatkan terdiri dari 7 atribut yaitu jenis kelamin (JK), tanggal lahir (Tgl Lahir), berat badan lahir (BB Lahir) dalam kg, tinggi badan lahir (TB Lahir) dalam cm, berat badan (Berat) dalam kg, tinggi badan (Tinggi) dalam cm, dan tinggi badan/usia (TB/U). Contoh data penelitian adalah sebagai berikut :

Tabel 9. Contoh Data Penelitian

No	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	28/12/2018	2.2	48	18	98	Pendek
4	L	07/01/2019	3	48	13	97	Pendek
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek

### 3.4 Tahapan Penelitian

Tahapan penelitian dapat dilihat pada gambar 4.



Gambar 4. *Flowchart* Tahapan Penelitian

Berikut adalah penjelasan dari tahapan penelitian pada Gambar 4.

#### 3.4.1 Studi Literatur

Tahapan pertama pada penelitian ini adalah mencari referensi mengenai penelitian-penelitian terdahulu sesuai dengan objek ataupun metode yang digunakan sebagai referensi untuk penelitian yang akan dilakukan.

### 3.4.2 Pengumpulan Data

Pengumpulan data *stunting* di Kabupaten Lampung Barat berdasarkan data antropometri diperoleh dari Dinas Kesehatan Kabupaten Lampung Barat.

### 3.4.3 Pre-Processing Data

*Pre-Processing* data dilakukan guna meningkatkan kinerja dari data yang akan digunakan, tahapan ini meliputi proses berikut:

A. Menghapus data (*data cleaning*)

Tahapan ini bertujuan memperbaiki kesalahan yang terdapat di dalam data seperti data 0 (nol) dan data kosong (tidak terdapat *value* data) dengan menggunakan microsoft excel.

B. Mengubah data (*data transforming*)

Tahapan ini bertujuan untuk mengubah data menjadi format yang sesuai untuk analisis yaitu numerik, sehingga data yang berbentuk text harus diubah menjadi bentuk numerik dengan bantuan microsoft excel dan jupyter notebook versi 6.3.0.

C. Normalisasi data

Tahapan ini bertujuan untuk mengubah nilai data pada skala tertentu agar bersifat lebih umum untuk digunakan ke tahap selanjutnya, tahapan ini dilakukan dengan bantuan jupyter notebook versi 6.3.0.

D. *Balancing* data

Tahapan ini bertujuan untuk menyeimbangkan jumlah data yang berbeda pada masing-masing kelasnya, hal ini dilakukan agar sistem tidak memiliki kecenderungan pada kelas yang memiliki record data lebih banyak saja, tetapi semua kelas di prediksi secara rata.

### 3.3.4 K-Fold Cross Validation

Tahapan selanjutnya adalah pembagian dataset menggunakan *k-fold cross validation*, tahapan ini dilakukan dengan membagi data penelitian menjadi dua bagian yaitu data *training* dan data *testing*.

### 3.3.5 Implementasi Model SVM dan XGBoost

Tahapan selanjutnya adalah implementasi model SVM dan *XGBoost*, untuk proses klasifikasi. Model diperoleh dari data *training* dengan percobaan menggunakan beberapa parameter pendukung sampai mendapatkan hasil terbaik, kemudian, model yang sudah didapatkan akan digunakan untuk memprediksi data uji sehingga didapatkan nilai akurasi prediksi dari masing-masing metode yang telah digunakan untuk kemudian mendapatkan hasil klasifikasi data.

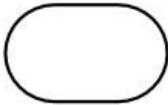
### 3.3.6 Pengujian

Selanjutnya adalah tahap pengujian. Model SVM dan *XGBoost* yang telah berhasil dibuat dengan data *training* selanjutnya diuji menggunakan data *testing*. Pengujian yang dilakukan menggunakan *confussion matrix*.

### 3.3.7 Perbandingan Hasil Klasifikasi

Tahapan terakhir adalah melakukan perbandingan hasil klasifikasi dari 2 metode yang telah digunakan yaitu *support vector machine* dan *xtreme gradient boosting*. Perbandingan dilakukan dengan melihat hasil nilai akurasi terbaik dalam memprediksi serta mengklasifikasikan data uji.

Tabel 10. Keterangan Flowchart

SIMBOL	KETERANGAN
	<i>Terminator</i> (menunjukkan awal atau akhir dari diagram alur)
	<i>Process</i> (mewakili langkah dalam suatu proses)
	<i>Flow Line</i> (menunjukkan arah proses dan menghubungkan dua blok)

## IV. PEMBAHASAN

Berikut adalah tahapan-tahapan yang dilakukan untuk mengolah data, tahapan ini akan dilakukan berurutan sesuai dengan tahapan penelitian yang telah dijelaskan pada sub bab tahapan penelitian.

### 4.1 Pengumpulan Data

Data yang dibutuhkan untuk melakukan klasifikasi *stunting* di Kabupaten Lampung Barat menggunakan metode *support vector machine* dan *xtreme gradient boosting* diperoleh dari Dinas Kesehatan Kabupaten Lampung Barat. Data yang didapatkan berjumlah 18.372 record data yang terdiri dari 7 atribut yaitu jenis kelamin, tanggal lahir, berat badan lahir, tinggi badan lahir, berat badan, tinggi badan dan tinggi badan per usia.

### 4.2 Pre-Processing Data

Setelah keseluruhan data yang akan digunakan telah dikumpulkan, langkah pertama yang dilakukan adalah *pre-processing*. Terdapat empat tahapan pada *pre-processing*, yaitu *data cleaning*, *data transforming*, normalisasi data dan *balancing data* yang digunakan untuk mempermudah, mempercepat komputasi dan menghasilkan hasil yang lebih akurat.

#### 4.2.1. Data Cleaning

Pada penelitian ini, jumlah record data awal berjumlah 18.372 record data yang terdiri dari kelas normal sebanyak 17.732 record data, kelas pendek 451 record data dan kelas sangat pendek 169 record data dengan jumlah jenis kelamin perempuan berjumlah 8407 record data dan jumlah jenis kelamin laki-laki berjumlah 9965 record data.

Terdapat beberapa hal yang dilakukan pada proses *cleaning data* yaitu

1. Menghapus data nol, data bernilai nol adalah jika terdapat nilai data pada suatu atribut bernilai nol (0), sehingga tidak dapat diidentifikasi.

Adapun contohnya adalah sebagai berikut:

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	16/04/2019	0	55	17.2	110.1	Normal
4	L	21/07/2019	0	0	14	102	Normal
5	P	07/02/2021	3	0	14.6	96	Normal
6	P	27/11/2018	2.8		16	110	Normal
7	P	18/06/2019		54	15	100	Normal
8	L	24/08/2019			16	99.7	Normal
9	P	28/12/2018	2.2	48	18	98	Pendek
10	L	07/01/2019	3	48	13	97	Pendek
11	P	15/10/2018	0	0	12.5	98.5	Pendek
12	P	22/01/2021	2.9	0	10.7	82	Pendek
13	L	16/09/2019			12.7	93.4	Pendek
14	L	31/01/2021	3.5		10.2	86	Pendek
15	P	01/12/2019		48	11.9	89.2	Pendek
16	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
17	L	24/11/2018	3	49	18	89.5	Sangat Pendek
18	L	13/08/2019			12	90	Sangat Pendek
19	P	26/09/2021		49	7.1	74	Sangat Pendek
20	L	13/07/2022	2.7		8.8	70.5	Sangat Pendek
21	P	30/05/2022	2.5	0	6.5	67	Sangat Pendek

Gambar 6. Ilustrasi data 0

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	27/11/2018	2.8		16	110	Normal
4	P	18/06/2019		54	15	100	Normal
5	L	24/08/2019			16	99.7	Normal
6	P	28/12/2018	2.2	48	18	98	Pendek
7	L	07/01/2019	3	48	13	97	Pendek
8	L	2019-09-16			12.7	93.4	Pendek
9	L	31/01/2021	3.5		10.2	86	Pendek
10	P	01/12/2019		48	11.9	89.2	Pendek
11	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
12	L	24/11/2018	3	49	18	89.5	Sangat Pendek
13	L	13/08/2019			12	90	Sangat Pendek
14	P	26/09/2021		49	7.1	74	Sangat Pendek
15	L	13/07/2022	2.7		8.8	70.5	Sangat Pendek

Gambar 5. Ilustrasi data setelah *cleaning data* 0

Data yang memiliki nilai 0 hanya terdapat pada berat badan lahir atau biasa disebut BBL dan tinggi badan lahir atau biasa disebut TBL dengan rincian sebagai berikut.

	<b>BBL 0</b>	<b>TBL 0</b>	<b>BBL &amp; TBL 0</b>	<b>Total Record data 0</b>
Normal	6	202	109	317
Pendek	-	6	1	7
Sangat Pendek	-	1	-	1
<b>TOTAL</b>	<b>6</b>	<b>209</b>	<b>110</b>	<b>325</b>

Tabel 11. Jumlah rekap *cleaning data 0*

Sehingga total data 0 yang dihapus berjumlah 325 record data.

- Menghapus data kosong, data kosong adalah jika terdapat data yang tidak terisi atau kosong pada atribut tertentu sehingga suatu record data menjadi tidak lengkap.

Adapun contohnya sebagai berikut:

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	27/11/2018	2.8		16	110	Normal
4	P	18/06/2019		54	15	100	Normal
5	L	24/08/2019			16	99.7	Normal
6	P	28/12/2018	2.2	48	18	98	Pendek
7	L	07/01/2019	3	48	13	97	Pendek
8	L	2019-09-16			12.7	93.4	Pendek
9	L	31/01/2021	3.5		10.2	86	Pendek
10	P	01/12/2019		48	11.9	89.2	Pendek
11	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
12	L	24/11/2018	3	49	18	89.5	Sangat Pendek
13	L	13/08/2019			12	90	Sangat Pendek
14	P	26/09/2021		49	7.1	74	Sangat Pendek
15	L	13/07/2022	2.7		8.8	70.5	Sangat Pendek

Gambar 7. Ilustrasi data kosong

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	28/12/2018	2.2	48	18	98	Pendek
4	L	07/01/2019	3	48	13	97	Pendek
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek

Gambar 8. Ilustrasi data setelah *cleaning data* kosong

Adapun data yang memiliki nilai kosong hanya terdapat pada berat badan lahir atau biasa disebut BBL dan tinggi badan lahir atau biasa disebut TBL dengan rincian sebagai berikut.

Tabel 12. Jumlah rekap *cleaning data* kosong

	BBL kosong	TBL kosong	BBL & TBL kosong	Total Record data kosong
Normal	121	318	89	528
Pendek	3	8	2	13
Sangat Pendek	2	5	1	8
<b>TOTAL</b>	<b>126</b>	<b>331</b>	<b>92</b>	<b>549</b>

Sehingga total data kosong yang dihapus berjumlah 549 record data.

Total jumlah record data yang terhapus adalah penjumlahan dari total *cleaning data* 0 (325) dan total *cleaning data* kosong (549), yaitu berjumlah 874 record data.

Berikut adalah perbedaan jumlah record data sebelum dan sesudah dilakukan *cleaning data*.

Tabel 13. Perbedaan jumlah record data sebelum dan sesudah *cleaning data*

	Sebelum <i>Cleaning data</i>		Setelah <i>Cleaning data</i>		Jumlah Perubahan	
	P	L	P	L	P	L
Normal	8156	9596	7763	9144	393	452

	Sebelum <i>Cleaning data</i>		Setelah <i>Cleaning data</i>		Jumlah Perubahan	
	P	L	P	L	P	L
Pendek	192	259	182	249	10	10
Sangat Pendek	59	110	54	106	5	4
<b>TOTAL</b>	<b>8407</b>	<b>9965</b>	<b>7999</b>	<b>2499</b>	<b>408</b>	<b>466</b>

Tabel 14. Rekap perbedaan jumlah data sebelum dan sesudah *cleaning data*

Jumlah record data sebelum <i>cleaning data</i>	Jumlah record data setelah <i>cleaning data</i>
18372	17498

Dengan total kelas normal berjumlah 16907 record data, kelas pendek berjumlah 431 record data dan kelas sangat pendek berjumlah 160 record data.

#### 4.2.2. Data Transforming

Pada penelitian ini, *transforming data* merupakan proses mengubah data menjadi bentuk numerik sehingga data dapat digunakan untuk proses selanjutnya, tahapan ini berupa:

A. Penambahan atribut baru, yaitu:

1. Atribut tanggal data (Tgl Data), penambahan atribut tanggal data digunakan untuk menambahkan atribut selanjutnya. Dikarenakan bulan timbang dari data yang didapatkan adalah Oktober 2023, sehingga Tanggal Data dari seluruh record data 1 sampai record data 17498 adalah sama yaitu 31 Oktober 2023 (31/10/2023). Penambahan atribut ini dilakukan dengan menggunakan microsoft excel.

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U
1	P	25/10/2018	3.3	50	20	113	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal
3	P	28/12/2018	2.2	48	18	98	Pendek
4	L	07/01/2019	3	48	13	97	Pendek
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek

Gambar 9. Ilustrasi data sebelum ditambah atribut tanggal data

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U	Tgl Data
1	P	25/10/2018	3.3	50	20	113	Normal	31/10/2023
2	L	04/11/2020	2.8	49	16.7	107	Normal	31/10/2023
3	P	28/12/2018	2.2	48	18	98	Pendek	31/10/2023
4	L	07/01/2019	3	48	13	97	Pendek	31/10/2023
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek	31/10/2023
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek	31/10/2023

Gambar 10. Ilustrasi data sesudah penambahan atribut tanggal data

2. Atribut usia, atribut ini ditambahkan dengan tujuan memeriksa dan memastikan bahwa data yang digunakan hanya terdiri dari balita dengan usia 0-60 bulan, atribut usia ini kemudian akan digunakan untuk memastikan bahwa atribut TB/U sudah sesuai dengan ketentuan standar panjang badan menurut umur yang telah dijelaskan pada sub bab antropometri. Penambahan atribut ini didapatkan dari atribut tanggal lahir (Tgl Lahir) dan tanggal data (Tgl Data). Penambahan atribut ini dilakukan dengan menggunakan microsoft excel.

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U	Tgl Data
1	P	25/10/2018	3.3	50	20	113	Normal	31/10/2023
2	L	04/11/2020	2.8	49	16.7	107	Normal	31/10/2023
3	P	28/12/2018	2.2	48	18	98	Pendek	31/10/2023
4	L	07/01/2019	3	48	13	97	Pendek	31/10/2023
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek	31/10/2023
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek	31/10/2023

Gambar 11. Ilustrasi data sebelum penambahan atribut usia

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U	Tgl Data	Usia
1	P	25/10/2018	3.3	50	20	113	Normal	31/10/2023	60
2	L	04/11/2020	2.8	49	16.7	107	Normal	31/10/2023	35
3	P	28/12/2018	2.2	48	18	98	Pendek	31/10/2023	58
4	L	07/01/2019	3	48	13	97	Pendek	31/10/2023	57
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek	31/10/2023	55
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek	31/10/2023	59

Gambar 12. Ilustrasi data sesudah penambahan atribut usia

3. Penambahan atribut TB/U yang sesuai dengan peraturan Menteri Kesehatan Republik Indonesia (Kemenkes R1) sebagaimana yang telah dijelaskan pada sub bab antropometri. Penambahan atribut ini dilakukan untuk memeriksa

apakah TB/U yang sudah ada sesuai dengan yang tertera pada peraturan kemenkes RI tersebut.

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U	Tgl Data	Usia
1	P	25/10/2018	3.3	50	20	113	Normal	31/10/2023	60
2	L	04/11/2020	2.8	49	16.7	107	Normal	31/10/2023	35
3	P	28/12/2018	2.2	48	18	98	Pendek	31/10/2023	58
4	L	07/01/2019	3	48	13	97	Pendek	31/10/2023	57
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek	31/10/2023	55
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek	31/10/2023	59

Gambar 13. Ilustrasi data sebelum penambahan atribut TB/U Baru

NO	JK	Tgl Lahir	BB Lahir	TB Lahir	Berat	Tinggi	TB/U	Tgl Data	Usia	TB/U Baru
1	P	25/10/2018	3.3	50	20	113	Normal	31/10/2023	60	Normal
2	L	04/11/2020	2.8	49	16.7	107	Normal	31/10/2023	35	Tinggi
3	P	28/12/2018	2.2	48	18	98	Pendek	31/10/2023	58	Pendek
4	L	07/01/2019	3	48	13	97	Pendek	31/10/2023	57	Pendek
5	P	07/03/2019	3	50	13.1	91.3	Sangat Pendek	31/10/2023	55	Sangat Pendek
6	L	24/11/2018	3	49	18	89.5	Sangat Pendek	31/10/2023	59	Sangat Pendek

Gambar 14. Ilustrasi data sesudah penambahan atribut TB/U Baru

Pada gambar 14. dapat dilihat bahwa tidak semua data TB/U sesuai dengan peraturan kemenkes RI seperti yang sudah dijabarkan pada BAB II. Adapun perubahan jumlah record data dari masing-masing kelas adalah sebagai berikut:

Tabel 15. Rekap jumlah perubahan TB/U awal dan TB/U baru

	TB/U Awal		TB/U Baru		TOTAL	
	P	L	P	L	P	L
Normal	7763	9144	7567	8905	-196	-239
Pendek	182	249	366	486	+184	+237
Sangat Pendek	54	106	60	107	+6	+1
Tinggi	-	-	6	1	+6	+1
<b>TOTAL</b>	<b>7999</b>	<b>9499</b>	<b>7999</b>	<b>9499</b>	<b>0</b>	<b>0</b>

Data yang didapatkan dari dinas Kesehatan kabupaten Lampung Barat hanya terbagi menjadi 3 kelas yaitu kelas normal, kelas pendek dan kelas sangat pendek. Tetapi setelah melakukan pemeriksaan ulang dengan penambahan atribut TB/U baru sesuai dengan peraturan kemenkes RI yang telah

dijelaskan pada sub bab antropometri, terdapat penambahan kelas baru yaitu kelas tinggi, penambahan kelas tersebut terjadi dikarenakan terdapat record data yang tidak termasuk kedalam kelas normal, kelas pendek ataupun kelas sangat pendek, tetapi ia masuk kedalam kelas tinggi yaitu memiliki tinggi  $> +3 SD$ . Sehingga terdapat 1 kelas tambahan yaitu kelas tinggi. Total kelas normal menjadi berjumlah 16472 record data dari 16907 record data, kelas pendek menjadi berjumlah 832 record data dari 431 record data, kelas sangat pendek menjadi berjumlah 167 record data dari 160 record data dan kelas tinggi berjumlah 7 record data.

#### B. Pengubahan format menjadi CSV

Pengubahan format data menjadi csv (*Comma Separated Value*) dari format xlsx, perubahan ini dilakukan dikarenakan data yang akan diolah harus menggunakan format csv. Data kemudian di *import* kedalam *tools* yang akan digunakan, pada penelitian ini saya menggunakan jupyter notebook dengan bahasa pemrograman python.

```
JK,Tgl Lahir,BB Lahir,TB Lahir,Berat,Tinggi,TB/U,Tgl Data,Usia,TB/U Baru
P,10/25/2018,3.3,50,20,113,Normal,10/31/2023,60,Normal
L,11/04/2020,2.8,49,16.7,107,Normal,10/31/2023,35,Tinggi
P,12/28/2018,2.2,48,18,98,Pendek,10/31/2023,58,Pendek
L,1/7/2019,3,48,13,97,Pendek,10/31/2023,57,Pendek
P,3/7/2019,3,50,13.1,91.3,Sangat Pendek,10/31/2023,55,Sangat Pendek
L,11/24/2018,3,49,18,89.5,Sangat Pendek,10/31/2023,59,Sangat Pendek
```

Gambar 15. Ilustrasi pengubahan data dalam format csv

#### C. Penghapusan beberapa atribut

Penghapusan beberapa atribut yaitu atribut tanggal lahir, atribut tanggal data dan atribut TB/U. Penghapusan beberapa atribut ini dilakukan karena atribut tersebut sudah tidak lagi digunakan. Atribut tanggal lahir dan tanggal data hanya digunakan untuk menghitung usia dari masing-masing balita, sedangkan atribut TB/U tidak lagi digunakan karena sudah terdapat atribut TB/U yang baru yang sesuai dengan peraturan kemenkes RI. Penghapusan beberapa atribut ini juga dilakukan agar mengurangi waktu

komputasi dan penyimpanan data. Tahap ini dilakukan dengan bantuan jupyter notebook.

	JK	BB Lahir	TB Lahir	Berat	Tinggi	Usia	TB/U Baru
0	P	3.3	50	20.0	113.0	60	Normal
1	L	2.8	49	16.7	107.0	35	Tinggi
2	P	2.2	48	18.0	98.0	58	Pendek
3	L	3.0	48	13.0	97.0	57	Pendek
4	P	3.0	50	13.1	91.3	55	SangatPendek
5	L	3.0	49	18.0	89.5	59	SangatPendek

Gambar 16. Ilustrasi data setelah dilakukan penghapusan beberapa atribut

#### D. Perubahan beberapa nama atribut

Perubahan beberapa nama atribut yang digunakan untuk mempermudah proses baca data pada jupyter, adapun beberapa nama atribut yang diubah yaitu sebagai berikut:

Tabel 16. Perubahan nama atribut

Nama atribut lama	Nama atribut baru
Tgl Lahir	TglLahir
BB Lahir	BBL
TB Lahir	TBL
TB/U Baru	Kelas

Sehingga atribut yang akan digunakan untuk tahapan selanjutnya terdiri dari 7 atribut yaitu jenis kelamin (JK), tanggal lahir (TglLahir), berat badan lahir (BBL) dalam kg, tinggi badan lahir (TBL) dalam cm, berat badan (Berat) dalam kg, tinggi badan (Tinggi) dalam cm, dan tinggi badan/usia (Kelas).

	JK	BBL	TBL	Berat	Tinggi	Usia	Kelas
0	P	3.3	50	20.0	113.0	60	Normal
1	L	2.8	49	16.7	107.0	35	Tinggi
2	P	2.2	48	18.0	98.0	58	Pendek
3	L	3.0	48	13.0	97.0	57	Pendek
4	P	3.0	50	13.1	91.3	55	SangatPendek
5	L	3.0	49	18.0	89.5	59	SangatPendek

Gambar 17. Atribut yang akan digunakan

#### E. Pengubahan value

Pengubahan value dalam record data yang masih berbentuk non numerik menjadi numerik, pada tahapan ini dilakukan dengan bantuan jupyter notebook menggunakan bahasa python dengan package pandas.

	JK	BBL	TBL	Berat	Tinggi	Usia	Kelas
0	0	3.3	50	20.0	113.0	60	2
1	1	2.8	49	16.7	107.0	35	3
2	0	2.2	48	18.0	98.0	58	0
3	1	3.0	48	13.0	97.0	57	0
4	0	3.0	50	13.1	91.3	55	1
5	1	3.0	49	18.0	89.5	59	1

Gambar 18. Perubahan nilai data beberapa atribut

Atribut JK yang terdiri dari P (perempuan) dan L (laki-laki) diubah menjadi 0 (perempuan) dan 1 (laki-laki). Lalu atribut Kelas yang terdiri dari normal, pendek, sangat pendek dan tinggi diubah menjadi 1(pendek), 2(sangat pendek), 3 (normal) dan 4 (sangat tinggi).

#### F. Pengubahan type data

Pengubahan type data pada atribut JK dan Kelas, hal ini dilakukan dengan tujuan agar keseluruhan data ber-type numerik dan dapat digunakan untuk klasifikasi.

Adapun perubahan type data nya sebagai berikut:

JK	object
BBL	float64
TBL	int64
Berat	float64
Tinggi	float64
Usia	int64
Kelas	object

Gambar 19. *Type* data sebelum diubah

JK	int64
BBL	float64
TBL	int64
Berat	float64
Tinggi	float64
Usia	int64
Kelas	int32

Gambar 20. *Type* data setelah diubah

#### 4.2.3. Normalisasi Data

Normalisasi data digunakan untuk mengubah bentuk data menjadi bentuk yang lebih pantas untuk model klasifikasi dengan skala tertentu yang bersifat lebih umum (Karim et al., 2023).

Pada penelitian ini akan dilakukan percobaan dengan *min-max scaler* dikarenakan nilai maksimum dan nilai minimum dari masing-masing data sudah diketahui.

Berdasarkan ilustrasi data pada gambar 18, diketahui nilai maksimum dan nilai minimum dari masing-masing data adalah sebagai berikut.

Tabel 17. Nilai minimum dan maksimum dari masing-masing data

	<b>BBL</b>	<b>TBL</b>	<b>Berat</b>	<b>Tinggi</b>	<b>Usia</b>
<b>Nilai Minimum</b>	2.2	48	13.0	89.5	35
<b>Nilai Maksimum</b>	3.3	50	20.0	113.0	59

Adapun rumus dari *min-max scaler* adalah sebagai berikut:

$$x_{new} = \frac{x + x_{min}}{x_{max} - x_{min}}$$

Keterangan:

$x_{new}$  = nilai yang telah dinormalisasi dari fitur

$x$  = nilai awal fitur

$x_{max}$  = nilai maksimal dari fitur

$x_{min}$  = nilai minimum dari fitur

Contoh perhitungannya dengan menggunakan record data ilustrasi pertama pada atribut BBL yang bernilai 3.3 adalah sebagai berikut:

$$x_{new} = \frac{3.3 - 2.2}{3.3 - 2.2}$$

$$x_{new} = 1$$

Sehingga didapatkan nilai normalisasi dari masing-masing data ilustrasi adalah sebagai berikut.

Tabel 18. Hasil Normalisasi (Record data ilustrasi)

	<b>BBL</b>	<b>TBL</b>	<b>Berat</b>	<b>Tinggi</b>	<b>Usia</b>
<b>0</b>	1.000000	1.000000	1.000000	1.000000	1.000000
<b>1</b>	0.545455	0.500000	0.528571	0.744681	0.000000
<b>2</b>	0.000000	0.000000	0.714286	0.361702	0.920000
<b>3</b>	0.727273	0.000000	0.000000	0.319149	0.880000
<b>4</b>	0.727273	1.000000	0.014286	0.076596	0.800000

	<b>BBL</b>	<b>TBL</b>	<b>Berat</b>	<b>Tinggi</b>	<b>Usia</b>
<b>5</b>	0.727273	0.500000	0.714286	0.000000	0.960000

Selanjutnya tahap normalisasi ini dilakukan secara menyeluruh terhadap data penelitian, dengan nilai minimum dan maksimum data keseluruhan sebagai berikut.

Tabel 19. Nilai minimum dan maksimum dari masing-masing data

	<b>BBL</b>	<b>TBL</b>	<b>Berat</b>	<b>Tinggi</b>	<b>Usia</b>
<b>Nilai Minimum</b>	1.5	5	1.9	45	0
<b>Nilai Maksimum</b>	4.5	60	27.5	120.4	60

Berikut adalah hasil normalisasi keseluruhan data menggunakan *min-max scaler* dengan bantuan jupyter notebook.

Tabel 20. Hasil Normalisasi

	<b>BBL</b>	<b>TBL</b>	<b>Berat</b>	<b>Tinggi</b>	<b>Usia</b>
<b>0</b>	0.566667	0.727273	0.378906	0.595491	0.983333
<b>1</b>	0.500000	0.818182	0.437500	0.614058	0.916667
<b>2</b>	0.500000	0.781818	0.433594	0.689655	0.950000
<b>3</b>	0.300000	0.763636	0.468750	0.729443	1.000000
<b>4</b>	0.233333	0.781818	0.628906	0.702918	0.966667
...	...	...	...	...	...
<b>17493</b>	0.500000	0.800000	0.042969	0.053050	0.116667
<b>17494</b>	0.666667	0.763636	0.062500	0.026525	0.000000
<b>17495</b>	0.566667	0.818182	0.234375	0.358090	0.100000
<b>17496</b>	0.500000	0.800000	0.136719	0.172414	0.050000
<b>17497</b>	0.766667	0.818182	0.074219	0.066313	0.000000

#### 4.2.1. Balancing Data

Data hasil cleaning menunjukkan jumlah kelas yang tidak seimbang (*imbalance*), yaitu sebagai berikut:

Tabel 21. Jumlah data per kelas sebelum dilakukan *balancing* data

	Jumlah data masing-masing kelas
<b>Kelas (0)</b>	167
<b>Kelas (1)</b>	832
<b>Kelas (2)</b>	16472
<b>Kelas (3)</b>	7

Kelas yang lebih banyak disebut dengan *majority class* dan kelas yang lebih sedikit disebut dengan *minority class*. Hal ini akan bermasalah pada proses pengklasifikasian dikarenakan system akan cenderung memprediksi kelas dengan jumlah data yang lebih banyak (Zhafirah, 2023).

Pada penelitian ini akan dilakukan percobaan *balancing* data dengan menggunakan *Random Oversampling* (ROS). ROS adalah suatu metode yang digunakan untuk menyeimbangkan jumlah data pada masing-masing kelas dengan melakukan duplikasi pada kelas minoritas sehingga jumlahnya sama atau mendekati kelas mayoritas (Firman et al., 2021).

Tabel 22. Jumlah data per kelas setelah dilakukan *balancing* data

	Jumlah data masing-masing kelas
<b>Kelas (0)</b>	16472
<b>Kelas (1)</b>	16472
<b>Kelas (2)</b>	16472
<b>Kelas (3)</b>	16472

Total record data setelah dilakukan *balancing* data adalah 65888.

### 4.3 Pembagian Data

Setelah dilakukan *pre-processing* pada data, tahapan selanjutnya adalah pembagian data, pembagian data dilakukan dengan menggunakan *k fold-cross validation* dengan k berjumlah 10 dengan perbandingan 9:1 (9 untuk data *training* dan 1 untuk data *testing*), sehingga akan dilakukan 10 kali iterasi yang akan menghasilkan 10 akurasi, dan kemudian akan diambil rata-rata dari 10 akurasi tersebut sehingga akurasi yang didapatkan akan bersifat lebih akurat.

Sehingga perbandingan jumlah data pada masing-masing iterasi adalah sebagai berikut:

- A. Data Latih (*training*): 592992 record data
- B. Data Uji (*testing*): 65888 record data

Dengan rincian sebagai berikut:

Tabel 23. Jumlah pembagian data

		<b>Total</b>	<b>Kelas 0</b>	<b>Kelas 1</b>	<b>Kelas 2</b>	<b>Kelas 3</b>
<b>Fold 1</b>	Data <i>Training</i>	59299	14824	14851	14818	14806
	Data <i>Testing</i>	6589	1666	1654	1648	1621
<b>Fold 2</b>	Data <i>Training</i>	59299	14793	14802	14869	14835
	Data <i>Testing</i>	6589	1679	1670	1637	1603
<b>Fold 3</b>	Data <i>Training</i>	59299	14793	14830	14843	14833
	Data <i>Testing</i>	6589	1679	1642	1639	1629
<b>Fold 4</b>	Data <i>Training</i>	59299	14882	14814	14798	14805
	Data <i>Testing</i>	6589	1590	1658	1674	1667
<b>Fold 5</b>	Data <i>Training</i>	59299	14861	14786	14830	14822
	Data <i>Testing</i>	6589	1611	1686	1642	1650
<b>Fold 6</b>	Data <i>Training</i>	59299	14858	14763	14863	14815
	Data <i>Testing</i>	6589	1614	1709	1609	1657
<b>Fold 7</b>	Data <i>Training</i>	59299	14813	14888	14818	14780
	Data <i>Testing</i>	6589	1659	1584	1654	1692
<b>Fold 8</b>	Data <i>Training</i>	59299	14797	14840	14823	14839
	Data <i>Testing</i>	6589	1675	1632	1649	1633
<b>Fold 9</b>	Data <i>Training</i>	59300	14848	14804	14773	14881
	Data <i>Testing</i>	6588	1630	1668	1699	1591
<b>Fold 10</b>	Data <i>Training</i>	59300	14785	14870	14825	14820
	Data <i>Testing</i>	6588	1687	1602	1647	1652

#### 4.4 Pembentukan Model Klasifikasi

Pada tahapan sebelumnya telah dilakukan *pre-processing data* dan pembagian data menjadi 2 bagian, yaitu data *training* dan data *testing*, tahapan selanjutnya adalah pemodelan untuk klasifikasi dengan menggunakan 2 algoritma yaitu *Support Vector Machine* (SVM) dan *Xtreme Gradient Boosting* (XGBoost).

#### 4.4.1 Model Klasifikasi *Support Vector Machine* (SVM)

Pada model SVM akan dilakukan beberapa percobaan, yaitu percobaan tanpa menggunakan parameter, dengan menggunakan kernel, dengan menggunakan parameter *cost*, dengan menggunakan parameter *gamma*, dengan menggunakan parameter iterasi maksimum dan dengan menggunakan 4 parameter yang digabungkan (dengan mengambil nilai terbaik dari masing-masing parameter). Terdapat beberapa nilai hyperparameter yang akan digunakan yaitu sebagai berikut:

Tabel 24. Nilai Hyperparameter SVM

<i>Hyperparameter</i>	Nilai
Kernel	Gaussian, Linear, Polynomial, Sigmoid
<i>Cost</i>	0.001, 1, 10, 100, 1000
<i>Gamma</i>	0.001, 0.01, 0.1, 1
Iterasi Maksimum	1000, 2000, 5000, 10000

##### A. Tanpa Parameter

Berdasarkan percobaan tanpa parameter, maka didapatkan hasil akurasi terbesar yaitu sebagai berikut:

Tabel 25. Percobaan metode SVM tanpa parameter

Iterasi	Akurasi
<i>Fold 1</i>	0.9433
<i>Fold 2</i>	0.9456
<i>Fold 3</i>	0.9417
<i>Fold 4</i>	0.9487
<i>Fold 5</i>	0.9468
<i>Fold 6</i>	0.9503
<i>Fold 7</i>	0.9432
<i>Fold 8</i>	0.9438
<i>Fold 9</i>	0.9479
<i>Fold 10</i>	0.9452
<b>Rata-Rata</b>	<b>0.9456</b>

Selanjutnya adalah pengujian menggunakan *confussion matriks*, berikut adalah hasil dari *confussion matrix* pada model SVM.

$$\begin{bmatrix} 16383 & 89 & 0 & 0 \\ 852 & 15620 & 0 & 0 \\ 0 & 2512 & 13835 & 125 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 26. Confussion matriks tanpa parameter

Hasil Pengujian SVM tanpa parameter	
<i>Accuracy</i>	0.9456
<i>Precision</i>	0.9500
<i>Recall</i>	0.9456
<i>F1-Score</i>	0.9477

## B. Kernel

Berikut adalah hasil dari percobaan penggunaan beberapa kernel.

Tabel 27. Hasil akurasi model SVM dengan menggunakan kernel

	Kernel			
	Gaussian	Linear	Polynomial	Sigmoid
<i>Fold 1</i>	0.9433	0.8203	0.9678	0.2695
<i>Fold 2</i>	0.9456	0.8165	0.9691	0.2770
<i>Fold 3</i>	0.9417	0.8310	0.9673	0.2759
<i>Fold 4</i>	0.9487	0.8248	0.9684	0.3332
<i>Fold 5</i>	0.9468	0.8251	0.9702	0.2828
<i>Fold 6</i>	0.9503	0.8291	0.9691	0.2787
<i>Fold 7</i>	0.9432	0.8239	0.9720	0.2797
<i>Fold 8</i>	0.9438	0.8107	0.9690	0.2777
<i>Fold 9</i>	0.9479	0.8114	0.9694	0.3811
<i>Fold 10</i>	0.9452	0.8263	0.9707	0.2730
<b>Rata-Rata</b>	<b>0.9456</b>	<b>0.8219</b>	<b>0.9693</b>	<b>0.2929</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai kernel *polynomial* mendapatkan hasil akurasi terbaik yaitu 0.9963. Berikut adalah hasil dari confusion matriks kernel *polynomial*.

$$\begin{bmatrix} 16370 & 102 & 0 & 0 \\ 491 & 15877 & 104 & 0 \\ 1 & 1269 & 15150 & 52 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 28. Confusion matriks kernel *polynomial*

Hasil Pengujian SVM dengan Kernel Polynomial	
<i>Accuracy</i>	0.9693
<i>Precision</i>	0.9703
<i>Recall</i>	0.9693
<i>F1-Score</i>	0.9697

### C. Cost

Berikut adalah hasil dari percobaan penggunaan beberapa *cost*.

Tabel 29. Hasil akurasi model SVM dengan menggunakan *Cost*

	<i>Cost</i>						
	<b>0.001</b>	<b>0.01</b>	<b>0.1</b>	<b>1</b>	<b>10</b>	<b>100</b>	<b>1000</b>
<i>Fold 1</i>	0.4915	0.7272	0.8940	0.9433	0.9722	0.9867	0.9921
<i>Fold 2</i>	0.4868	0.7189	0.8916	0.9456	0.9752	0.9895	0.9936
<i>Fold 3</i>	0.4970	0.7315	0.8967	0.9417	0.9723	0.9895	0.9946
<i>Fold 4</i>	0.4926	0.7313	0.8967	0.9487	0.9764	0.9893	0.9939
<i>Fold 5</i>	0.5026	0.7306	0.8914	0.9468	0.9740	0.9893	0.9937
<i>Fold 6</i>	0.4853	0.7378	0.9011	0.9503	0.9740	0.9892	0.9924
<i>Fold 7</i>	0.4956	0.7237	0.8948	0.9432	0.9729	0.9908	0.9943
<i>Fold 8</i>	0.4932	0.7180	0.8916	0.9438	0.9716	0.9902	0.9946
<i>Fold 9</i>	0.4740	0.7170	0.8914	0.9479	0.9740	0.9884	0.9937
<i>Fold 10</i>	0.5080	0.7213	0.8907	0.9452	0.9725	0.9875	0.9916
<b>Rata-Rata</b>	<b>0.4927</b>	<b>0.7257</b>	<b>0.8940</b>	<b>0.9456</b>	<b>0.9735</b>	<b>0.9891</b>	<b>0.9935</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *cost* 1000 mendapatkan hasil akurasi terbaik yaitu 0.9935. Berikut adalah hasil dari confusion matriks *cost* 1000.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 36 & 16397 & 39 & 0 \\ 0 & 350 & 16119 & 3 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 30. Confusion matriks SVM *cost* 1000

Hasil Pengujian SVM dengan <i>Cost</i> 1000	
<i>Accuracy</i>	0.9935
<i>Precision</i>	0.9935
<i>Recall</i>	0.9934
<i>F1-Score</i>	0.9934

#### D. *Gamma*

Berikut adalah hasil dari percobaan penggunaan beberapa *gamma*.

Tabel 31. Hasil akurasi model SVM dengan menggunakan *Gamma*

	<i>Gamma</i>			
	<b>0.001</b>	<b>0.01</b>	<b>0.1</b>	<b>1</b>
<i>Fold 1</i>	0.4915	0.7272	0.8940	0.9433
<i>Fold 2</i>	0.4868	0.7189	0.8916	0.9456
<i>Fold 3</i>	0.4970	0.7315	0.8967	0.9417
<i>Fold 4</i>	0.4926	0.7313	0.8967	0.9487
<i>Fold 5</i>	0.5026	0.7306	0.8914	0.9468
<i>Fold 6</i>	0.4853	0.7378	0.9011	0.9503
<i>Fold 7</i>	0.4956	0.7237	0.8948	0.9432
<i>Fold 8</i>	0.4932	0.7180	0.8916	0.9438
<i>Fold 9</i>	0.4740	0.7170	0.8914	0.9479
<i>Fold 10</i>	0.5080	0.7213	0.8907	0.9452
<b>Rata-Rata</b>	<b>0.4927</b>	<b>0.7257</b>	<b>0.8940</b>	<b>0.9456</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai  $\gamma$  1 mendapatkan hasil akurasi terbaik yaitu 0.9456. Berikut adalah hasil dari confusion matriks  $\gamma$  1000.

$$\begin{bmatrix} 16179 & 293 & 0 & 0 \\ 962 & 15510 & 39 & 0 \\ 0 & 2902 & 13401 & 169 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 32. Confusion matriks SVM  $\gamma$  1000

Hasil Pengujian SVM dengan $\gamma$ 1000	
<i>Accuracy</i>	0.9343
<i>Precision</i>	0.9406
<i>Recall</i>	0.9343
<i>F1-Score</i>	0.9374

#### E. Iterasi Maksimum

Berikut adalah hasil dari percobaan penggunaan beberapa iterasi maksimum.

Tabel 33. Hasil akurasi model SVM dengan Iterasi Maksimum

	Iterasi Maksimum			
	1000	2000	5000	10000
<i>Fold 1</i>	0.9921	0.9931	0.9942	0.9948
<i>Fold 2</i>	0.9936	0.9937	0.9949	0.9966
<i>Fold 3</i>	0.9946	0.9948	0.9960	0.9966
<i>Fold 4</i>	0.9939	0.9945	0.9954	0.9952
<i>Fold 5</i>	0.9937	0.9943	0.9963	0.9952
<i>Fold 6</i>	0.9924	0.9936	0.9945	0.9952
<i>Fold 7</i>	0.9943	0.9945	0.9960	0.9960
<i>Fold 8</i>	0.9946	0.9951	0.9959	0.9960
<i>Fold 9</i>	0.9937	0.9942	0.9959	0.9960
<i>Fold 10</i>	0.9916	0.9927	0.9937	0.9960
<b>Rata-Rata</b>	<b>0.9935</b>	<b>0.9940</b>	<b>0.9953</b>	<b>0.9959</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai iterasi maksimum 10000 mendapatkan hasil akurasi terbaik yaitu 0.9959. Berikut adalah hasil dari confusion matriks iterasi maksimum

$$\begin{bmatrix} 16383 & 89 & 0 & 0 \\ 852 & 15620 & 17 & 0 \\ 0 & 2511 & 13836 & 125 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 34. Confussion matriks SVM Iterasi Maksimum

<b>Hasil Pengujian SVM dengan Iterasi Maksimum</b>	
<i>Accuracy</i>	0.9457
<i>Precision</i>	0.9500
<i>Recall</i>	0.9456
<i>F1-Score</i>	0.9477

Berdasarkan percobaan menggunakan 4 parameter dengan nilai yang berbeda, didapatkan nilai terbesar dari masing-masing parameter adalah sebagai berikut

Tabel 35. Nilai akurasi terbesar dari masing-masing parameter SVM

<b>Parameter</b>	<b>Nilai</b>
<i>Kernel</i>	Polynomial
<i>Cost</i>	1000
<i>Gamma</i>	1
Iterasi Maksimum	10000

Percobaan terakhir pada metode SVM adalah dengan menggabungkan seluruh nilai terbesar dari 4 parameter.

Berikut adalah hasil akurasi dari penggabungan keempat parameter dari tabel 35.

Tabel 36. Hasil akurasi model SVM dengan Gabungan 4 parameter

<b>Iterasi</b>	<b>Akurasi</b>
<i>Fold 1</i>	0.8143
<i>Fold 2</i>	0.7984

<b>Iterasi</b>	<b>Akurasi</b>
<i>Fold 3</i>	0.8301
<i>Fold 4</i>	0.7060
<i>Fold 5</i>	0.7772
<i>Fold 6</i>	0.8436
<i>Fold 7</i>	0.8423
<i>Fold 8</i>	0.7946
<i>Fold 9</i>	0.6903
<i>Fold 10</i>	0.8633
<b>Rata-Rata</b>	<b>0.7960</b>

Selanjutnya adalah pengujian menggunakan confusion matriks, berikut adalah hasil dari *confusion matrix* pada model SVM.

$$\begin{bmatrix} 14421 & 2051 & 0 & 0 \\ 982 & 8253 & 7237 & 0 \\ 4 & 1967 & 14487 & 14 \\ 0 & 0 & 1182 & 15290 \end{bmatrix}$$

Tabel 37. Confusion matriks SVM menggunakan 4 parameter gabungan

<b>Hasil Pengujian SVM dengan 4 parameter gabungan</b>	
<i>Accuracy</i>	0.7960
<i>Precision</i>	0.8100
<i>Recall</i>	0.7960
<i>F1-Score</i>	0.8082

Berikut adalah hasil perbandingan pengujian metode SVM dengan beberapa perlakuan yang berbeda.

Tabel 38. Hasil Perbandingan Pengujian SVM berdasarkan *Accuracy*

	<i>Accuracy</i>
Tanpa Parameter	0.9456
<i>Best Kernel</i>	0.9693
<i>Best Cost</i>	<b>0.9935</b>

	<i>Accuracy</i>
<i>Best Gamma</i>	0.9343
<i>Best Iterasi Maksimum</i>	0.9457
Gabungan 4 Parameter	0.7960

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian XGBoost dengan hanya menggunakan parameter *cost* yang bernilai 1000 mendapatkan nilai akurasi terbesar yaitu 0.9935.

Tabel 39. Hasil Perbandingan Pengujian SVM berdasarkan *Precision*

	<i>Precision</i>
Tanpa Parameter	0.9500
<i>Best Kernel</i>	0.9703
<i>Best Cost</i>	<b>0.9935</b>
<i>Best Gamma</i>	0.9407
<i>Best Iterasi Maksimum</i>	0.9500
Gabungan 4 Parameter	0.8100

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian SVM dengan hanya menggunakan parameter *cost* yang bernilai 1000 mendapatkan nilai *precision* terbesar yaitu 0.9935.

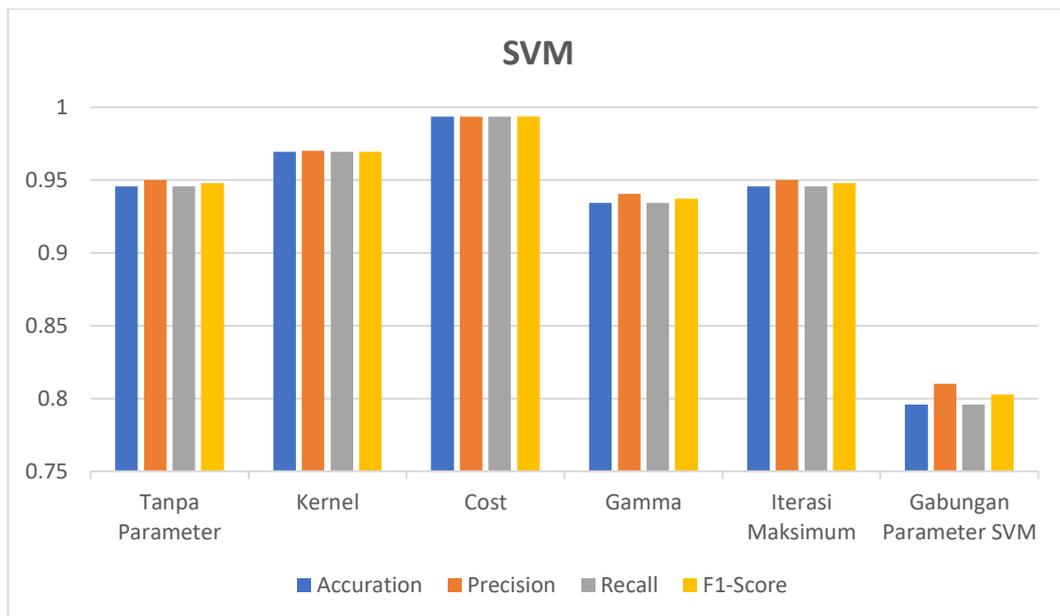
Tabel 40. Hasil Perbandingan Pengujian SVM berdasarkan *Recall*

	<i>Recall</i>
Tanpa Parameter	0.9456
<i>Best Kernel</i>	0.9693
<i>Best Cost</i>	<b>0.9934</b>
<i>Best Gamma</i>	0.9343
<i>Best Iterasi Maksimum</i>	0.9456
Gabungan 4 Parameter	0.7960

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian SVM dengan hanya menggunakan parameter *cost* yang bernilai 1000 mendapatkan nilai *recall* terbesar yaitu 0.9934.

Tabel 41. Hasil Perbandingan Pengujian SVM berdasarkan F1-Score

	F1-Score
Tanpa Parameter	0.9477
<i>Best Kernel</i>	0.9693
<i>Best Cost</i>	0.9934
<i>Best Gamma</i>	0.9374
<i>Best Iterasi Maksimum</i>	0.9477
Gabungan 6 Parameter	0.8028



Gambar 21. Grafik Perbandingan Pengujian Metode SVM

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian SVM dengan hanya menggunakan parameter *cost* yang bernilai 1000 mendapatkan nilai F1-score terbesar yaitu 0.9934.

Berdasarkan pengujian pada confusion matrix metode SVM dengan menggunakan *accuracy*, *precision*, *recall* dan F1-Score dapat dilihat bahwa pada dataset balita *stunting* di kabupaten Lampung Barat mendapatkan hasil akurasi terbesar dengan hanya menggunakan parameter *cost* (1000) yaitu dengan nilai *accuracy* 0.9935 serta nilai *precision* 0.9935, nilai *recall* 0.9934 dan F1-Score sebesar 0.9934. Hal ini membuktikan bahwa pada penelitian ini yaitu dengan

metode SVM penggabungan 4 parameter terbaik tidak mendapatkan hasil yang pengujian yang semakin baik (besar) juga.

#### 4.4.2 Model Klasifikasi *Xtreme Gradient Boosting (XGBoost)*

Pada model XGBoost akan dilakukan beberapa percobaan, yaitu percobaan tanpa menggunakan parameter, dengan menggunakan parameter *learning rate*, dengan menggunakan parameter *max depth*, dengan menggunakan parameter *gamma*, dengan menggunakan parameter *min child weight*, dengan menggunakan parameter *subsample*, dengan menggunakan parameter *colsample bytree* dan dengan menggunakan 6 parameter yang digabungkan (dengan mengambil nilai terbaik dari masing-masing parameter). Terdapat beberapa nilai hyperparameter yang akan digunakan yaitu sebagai berikut:

Tabel 42. Nilai Hyperparameter XGBoost

<i>Hyperparameter</i>	<b>Nilai</b>
<i>Learning rate</i>	0.1, 0.3, 0.5, 0.7, 0.9
<i>Max depth</i>	1, 3, 5, 7, 9
<i>Gamma</i>	0.01, 0.1, 0.5, 1
<i>Min child weight</i>	0.1, 0.5, 1
<i>Subsample</i>	0.1, 0.3, 0.5, 0.8, 1
<i>Colsample bytree</i>	0.1, 0.3, 0.5, 0.8, 1

##### A. Tanpa Parameter

Berdasarkan percobaan tanpa parameter, maka didapatkan hasil akurasi terbesar yaitu sebagai berikut:

Tabel 43. Percobaan metode XGBoost tanpa parameter

<b>Iterasi</b>	<b>Akurasi</b>
<i>Fold 1</i>	0.9960
<i>Fold 2</i>	0.9960
<i>Fold 3</i>	0.9978
<i>Fold 4</i>	0.9974
<i>Fold 5</i>	0.9966
<i>Fold 6</i>	0.9963

<b>Iterasi</b>	<b>Akurasi</b>
<i>Fold 7</i>	0.9977
<i>Fold 8</i>	0.9963
<i>Fold 9</i>	0.9966
<i>Fold 10</i>	0.9968
<b>Rata-Rata</b>	<b>0.9967</b>

Selanjutnya adalah pengujian menggunakan *confussion matriks*, berikut adalah hasil dari *confussion matrix* pada model XGBoost.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 210 & 16261 & 1 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 44. Confussion matriks tanpa parameter

<b>Hasil Pengujian tanpa Parameter</b>	
<i>Accuracy</i>	0.9967
<i>Precision</i>	0.9968
<i>Recall</i>	0.9967
<i>F1-Score</i>	0.9967

#### B. *Learning rate (eta)*

Berikut adalah hasil dari percobaan penggunaan beberapa *learning rate*.

Tabel 45. Hasil Akurasi XGBoost menggunakan *Learning rate*

	<i>Learning rate</i>				
	<b>0.1</b>	<b>0.3</b>	<b>0.5</b>	<b>0.7</b>	<b>0.9</b>
<i>Fold 1</i>	0.9863	0.9960	0.9966	0.9975	0.9977
<i>Fold 2</i>	0.9851	0.9960	0.9968	0.9974	0.9969
<i>Fold 3</i>	0.9887	0.9978	0.9981	0.9984	0.9986
<i>Fold 4</i>	0.9893	0.9974	0.9987	0.9987	0.9984
<i>Fold 5</i>	0.9867	0.9966	0.9977	0.9977	0.9977
<i>Fold 6</i>	0.9878	0.9963	0.9974	0.9980	0.9983

<i>Fold 7</i>	0.9881	0.9977	0.9984	0.9987	0.9992
<i>Fold 8</i>	0.9854	0.9963	0.9963	0.9971	0.9968
<i>Fold 9</i>	0.9881	0.9966	0.9974	0.9975	0.9978
<i>Fold 10</i>	0.9866	0.9968	0.9978	0.9981	0.9977
<b>Rata-Rata</b>	<b>0.9872</b>	<b>0.9967</b>	<b>0.9975</b>	<b>0.9979</b>	<b>0.9979</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *learning rate* 0.9 dan 0.8 mendapatkan hasil akurasi terbesar yaitu 0.9979, tetapi nilai akurasi terbesar pada masing-masing fold dari keseluruhan percobaan berada pada *learning rate* 0.9 yaitu sebesar 0.9992 (fold 7). Oleh karena itu pada penelitian ini, hasil *learning rate* terbaik adalah dengan nilai 0.9. Berikut adalah hasil dari confusion matriks *learning rate* 0.9.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 133 & 16337 & 2 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 46. Hasil Pengujian *Learning rate*

<b>Hasil Pengujian dengan <i>Learning rate</i></b>	
<i>Accuracy</i>	0.9979
<i>Precision</i>	0.9979
<i>Recall</i>	0.9979
<i>F1-Score</i>	0.9979

### C. *Max depth*

Berikut adalah hasil dari percobaan penggunaan beberapa *max depth*.

Tabel 47. Hasil Akurasi XGBoost menggunakan *Max depth*

	<b><i>Max depth</i></b>				
	<b>1</b>	<b>3</b>	<b>5</b>	<b>7</b>	<b>9</b>
<i>Fold 1</i>	0.7784	0.9699	0.9937	0.9969	0.9977
<i>Fold 2</i>	0.7688	0.9717	0.9925	0.9968	0.9972

	<i>Max depth</i>				
	<b>1</b>	<b>3</b>	<b>5</b>	<b>7</b>	<b>9</b>
<i>Fold 3</i>	0.7732	0.9717	0.9952	0.9981	0.9987
<i>Fold 4</i>	0.7738	0.9696	0.9946	0.9980	0.9977
<i>Fold 5</i>	0.7784	0.9707	0.9943	0.9977	0.9980
<i>Fold 6</i>	0.7864	0.9729	0.9942	0.9977	0.9981
<i>Fold 7</i>	0.7728	0.9728	0.9962	0.9984	0.9990
<i>Fold 8</i>	0.7620	0.9710	0.9943	0.9968	0.9962
<i>Fold 9</i>	0.7656	0.9748	0.9946	0.9972	0.9972
<i>Fold 10</i>	0.7830	0.9716	0.9928	0.9980	0.9978
<b>Rata-Rata</b>	<b>0.7742</b>	<b>0.9717</b>	<b>0.9943</b>	<b>0.9976</b>	<b>0.9978</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *max depth* 9 mendapatkan hasil akurasi terbaik yaitu 0.9978. Berikut adalah hasil dari confusion matriks *max depth* 9.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 142 & 16328 & 2 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 48. Hasil Pengujian dengan *Max depth*

<b>Hasil Pengujian dengan <i>Max depth</i></b>	
<i>Accuracy</i>	0.9978
<i>Precision</i>	0.9978
<i>Recall</i>	0.9978
<i>F1-Score</i>	0.9978

#### D. *Gamma*

Berikut adalah hasil dari percobaan penggunaan beberapa *gamma*.

Tabel 49. Hasil Akurasi XGBoost menggunakan *Gamma*

	<i>Gamma</i>			
	<b>0.01</b>	<b>0.1</b>	<b>0.5</b>	<b>1</b>
<i>Fold 1</i>	0.9962	0.9962	0.9954	0.9924

<i>Gamma</i>				
	<b>0.01</b>	<b>0.1</b>	<b>0.5</b>	<b>1</b>
<i>Fold 2</i>	0.9965	0.9960	0.9951	0.9931
<i>Fold 3</i>	0.9966	0.9971	0.9951	0.9940
<i>Fold 4</i>	0.9977	0.9975	0.9963	0.9937
<i>Fold 5</i>	0.9968	0.9969	0.9963	0.9942
<i>Fold 6</i>	0.9963	0.9966	0.9959	0.9948
<i>Fold 7</i>	0.9971	0.9978	0.9966	0.9948
<i>Fold 8</i>	0.9965	0.9955	0.9959	0.9925
<i>Fold 9</i>	0.9965	0.9962	0.9957	0.9955
<i>Fold 10</i>	0.9968	0.9969	0.9951	0.9931
<b>Rata-Rata</b>	<b>0.9967</b>	<b>0.9967</b>	<b>0.9957</b>	<b>0.9938</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *gamma* 0.01 dan 0.1 mendapatkan hasil akurasi yang sama yaitu 0.9967, tetapi nilai akurasi terbesar pada masing-masing fold dari keseluruhan percobaan berada pada parameter *gamma* 0.1 yaitu sebesar 0.9978 (fold 7). Berikut adalah hasil dari confusion matriks *gamma* 0.1.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 214 & 16256 & 2 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 50. Hasil Pengujian dengan *Gamma*

<b>Hasil Pengujian dengan <i>Gamma</i></b>	
<i>Accuracy</i>	0.9967
<i>Precision</i>	0.9967
<i>Recall</i>	0.9967
<i>F1-Score</i>	0.9967

### E. *Min child weight*

Berikut adalah hasil dari percobaan penggunaan beberapa *min child weight*.

Tabel 51. Hasil Akurasi XGBoost menggunakan *Min child weight*

<i>Min child weight</i>			
	<b>0.1</b>	<b>0.5</b>	<b>1</b>
<i>Fold 1</i>	0.9959	0.9959	0.9960
<i>Fold 2</i>	0.9962	0.9963	0.9960
<i>Fold 3</i>	0.9971	0.9971	0.9978
<i>Fold 4</i>	0.9977	0.9966	0.9974
<i>Fold 5</i>	0.9966	0.9965	0.9966
<i>Fold 6</i>	0.9974	0.9971	0.9963
<i>Fold 7</i>	0.9974	0.9978	0.9977
<i>Fold 8</i>	0.9959	0.9962	0.9963
<i>Fold 9</i>	0.9963	0.9966	0.9966
<i>Fold 10</i>	0.9972	0.9966	0.9968
<b>Rata-Rata</b>	<b>0.9967</b>	<b>0.9967</b>	<b>0.9967</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *min child weight* 0.1, 0.5 dan 1 mendapatkan hasil akurasi yang sama yaitu 0.9967, tetapi nilai akurasi terbesar pada masing-masing fold dari keseluruhan percobaan berada pada parameter *min child weight* 0.5 dan 1 yaitu sebesar 0.9978 (fold 7 pada *min child weight* 0.5 dan fold 3 pada *min child weight* 1). Selanjutnya jika dilihat dari kestabilan hasil akurasi *min child weight* dengan nilai 1 memiliki interval yang lebih stabil yaitu 0.0018. Berikut adalah hasil dari confusion matriks *min child weight* 1.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 1 & 209 & 16261 & 1 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 52. Hasil Pengujian dengan *Min child weight*

<b>Hasil Pengujian dengan <i>Min child weight</i></b>	
<i>Accuracy</i>	0.9967
<i>Precision</i>	0.9968
<i>Recall</i>	0.9967
<i>F1-Score</i>	0.9967

F. *Subsample*

Berikut adalah hasil dari percobaan penggunaan beberapa *subsample*.

Tabel 53. Hasil Akurasi XGBoost menggunakan *Subsample*

	<i>Subsample</i>				
	<b>0.1</b>	<b>0.3</b>	<b>0.5</b>	<b>0.8</b>	<b>1</b>
<i>Fold 1</i>	0.9921	0.9955	0.9966	0.9955	0.9960
<i>Fold 2</i>	0.9921	0.9954	0.9959	0.9959	0.9960
<i>Fold 3</i>	0.9936	0.9955	0.9968	0.9959	0.9978
<i>Fold 4</i>	0.9948	0.9963	0.9969	0.9963	0.9974
<i>Fold 5</i>	0.9928	0.9952	0.9965	0.9971	0.9966
<i>Fold 6</i>	0.9948	0.9963	0.9969	0.9966	0.9963
<i>Fold 7</i>	0.9934	0.9966	0.9969	0.9980	0.9963
<i>Fold 8</i>	0.9911	0.9954	0.9960	0.9954	0.9963
<i>Fold 9</i>	0.9937	0.9946	0.9965	0.9966	0.9963
<i>Fold 10</i>	0.9937	0.9957	0.9968	0.9962	0.9963
<b>Rata-Rata</b>	<b>0.9932</b>	<b>0.9957</b>	<b>0.9966</b>	<b>0.9965</b>	<b>0.9967</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *subsample* 1 mendapatkan hasil akurasi terbaik yaitu 0.9967. Berikut adalah hasil dari confusion matriks *subsample* 1.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 210 & 16261 & 1 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 54. Hasil Pengujian dengan *Subsample*

<b>Hasil Pengujian dengan <i>Subsample</i></b>	
<i>Accuracy</i>	0.9967
<i>Precision</i>	0.9968
<i>Recall</i>	0.9967
<i>F1-Score</i>	0.9967

G. *Colsample bytree*

Berikut adalah hasil dari percobaan penggunaan beberapa *colsample bytree*.

Tabel 55. Hasil Akurasi XGBoost menggunakan *Colsample bytree*

<b><i>Colsample bytree</i></b>					
	<b>0.1</b>	<b>0.3</b>	<b>0.5</b>	<b>0.8</b>	<b>1</b>
<i>Fold 1</i>	0.8344	0.8344	0.9855	0.9937	0.9960
<i>Fold 2</i>	0.8257	0.8257	0.9883	0.9946	0.9960
<i>Fold 3</i>	0.8289	0.8289	0.9896	0.9940	0.9978
<i>Fold 4</i>	0.8289	0.8277	0.9905	0.9948	0.9974
<i>Fold 5</i>	0.8339	0.8339	0.9896	0.9948	0.9966
<i>Fold 6</i>	0.8386	0.8386	0.9913	0.9954	0.9963
<i>Fold 7</i>	0.8459	0.8459	0.9884	0.9946	0.9977
<i>Fold 8</i>	0.8195	0.8195	0.9886	0.9946	0.9963
<i>Fold 9</i>	0.8292	0.8292	0.9886	0.9934	0.9966
<i>Fold 10</i>	0.8407	0.8407	0.9886	0.9951	0.9968
<b>Rata-Rata</b>	<b>0.8325</b>	<b>0.8325</b>	<b>0.9891</b>	<b>0.9945</b>	<b>0.9967</b>

Berdasarkan hasil diatas, didapatkan bahwa nilai *subsample 1* mendapatkan hasil akurasi terbaik yaitu 0.9967. Berikut adalah hasil dari confusion matriks *subsample 1*.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 210 & 16261 & 1 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 56. Hasil Pengujian dengan *Colsample bytree*

<b>Hasil Pengujian dengan <i>Colsample bytree</i></b>	
<i>Accuracy</i>	0.9967
<i>Precision</i>	0.9968
<i>Recall</i>	0.9967
<i>F1-Score</i>	0.9967

Berdasarkan percobaan menggunakan 6 parameter dengan nilai yang berbeda, didapatkan nilai terbesar dari masing-masing parameter adalah sebagai berikut

Tabel 57. Nilai akurasi terbesar dari masing-masing parameter XGBoost

<b>Parameter</b>	<b>Nilai</b>
<i>Learning rate</i>	0.9
<i>Max depth</i>	9
<i>Gamma</i>	0.1
<i>Min child weight</i>	1
<i>Subsample</i>	1
<i>Colsample bytree</i>	1

Percobaan terakhir pada metode XGBoost adalah dengan menggabungkan seluruh nilai terbesar dari 6 parameter.

Berikut adalah hasil akurasi dari penggabungan keenam parameter dari tabel 57.

Tabel 58. Hasil Akurasi XGBoost menggunakan gabungan 6 parameter

<b>Iterasi</b>	<b>Akurasi</b>
<i>Fold 1</i>	0.9969
<i>Fold 2</i>	0.9972
<i>Fold 3</i>	0.9984
<i>Fold 4</i>	0.9978
<i>Fold 5</i>	0.9978
<i>Fold 6</i>	0.9980

Iterasi	Akurasi
<i>Fold 7</i>	0.9987
<i>Fold 8</i>	0.9959
<i>Fold 9</i>	0.9972
<i>Fold 10</i>	0.9983
<b>Rata-Rata</b>	<b>0.9976</b>

Selanjutnya adalah pengujian menggunakan confusion matriks, berikut adalah hasil dari *confusion matrix* pada model *XGBoost*.

$$\begin{bmatrix} 16472 & 0 & 0 & 0 \\ 0 & 16472 & 0 & 0 \\ 0 & 150 & 16319 & 3 \\ 0 & 0 & 0 & 16472 \end{bmatrix}$$

Tabel 59. Confussion matriks dengan 6 Parameter Gabungan

Hasil Pengujian dengan 6 Parameter	
<i>Accuracy</i>	0.9976
<i>Precision</i>	0.9976
<i>Recall</i>	0.9976
<i>F1-Score</i>	0.9976

Berikut adalah hasil perbandingan pengujian metode *XGBoost* dengan beberapa perlakuan yang berbeda.

Tabel 60. Hasil Perbandingan Pengujian *XGBoost* berdasarkan *Accuracy*

	<i>Accuracy</i>
Tanpa Parameter	0.9967
<i>Best Learning rate</i>	<b>0.9979</b>
<i>Best Max depth</i>	0.9978
<i>Best Gamma</i>	0.9967
<i>Best Min child weight</i>	0.9967
<i>Best Subsample</i>	0.9967
<i>Best Colsample bytree</i>	0.9967
Gabungan 6 Parameter	0.9976

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian XGBoost dengan hanya menggunakan *learning rate* yang bernilai 0.9 mendapatkan nilai *accuracy* terbesar yaitu 0.9979.

Tabel 61. Hasil Perbandingan Pengujian XGBoost berdasarkan *Precision*

	<i>Precision</i>
Tanpa Parameter	0.9968
<i>Best Learning rate</i>	<b>0.9979</b>
<i>Best Max depth</i>	0.9978
<i>Best Gamma</i>	0.9967
<i>Best Min child weight</i>	0.9968
<i>Best Subsample</i>	0.9968
<i>Best Colsample bytree</i>	0.9968
Gabungan 6 Parameter	0.9976

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian XGBoost dengan hanya menggunakan *learning rate* yang bernilai 0.9 mendapatkan nilai *precision* terbesar yaitu 0.9979.

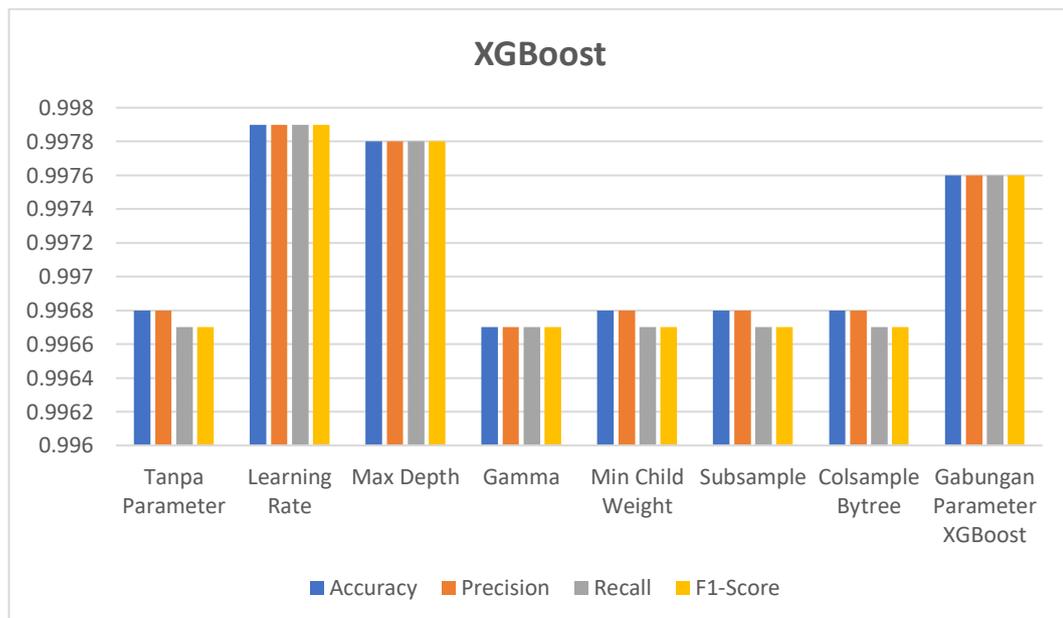
Tabel 62. Hasil Perbandingan Pengujian XGBoost berdasarkan *Recall*

	<i>Recall</i>
Tanpa Parameter	0.9967
<i>Best Learning rate</i>	<b>0.9979</b>
<i>Best Max depth</i>	0.9978
<i>Best Gamma</i>	0.9967
<i>Best Min child weight</i>	0.9967
<i>Best Subsample</i>	0.9967
<i>Best Colsample bytree</i>	0.9967
Gabungan 6 Parameter	0.9976

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian XGBoost dengan hanya menggunakan *learning rate* yang bernilai 0.9 mendapatkan nilai *recall* terbesar yaitu 0.9979.

Tabel 63. Hasil Perbandingan Pengujian XGBoost berdasarkan F1-Score

	F1-Score
Tanpa Parameter	0.9967
<i>Best Learning rate</i>	<b>0.9979</b>
<i>Best Max depth</i>	0.9978
<i>Best Gamma</i>	0.9967
<i>Best Min child weight</i>	0.9967
<i>Best Subsample</i>	0.9967
<i>Best Colsample bytree</i>	0.9967
Gabungan 6 Parameter	0.9976



Gambar 22. Grafik Perbandingan Pengujian Metode XGBoost

Berdasarkan tabel diatas, dapat dilihat bahwa hasil pengujian XGBoost dengan hanya menggunakan *learning rate* yang bernilai 0.9 mendapatkan nilai F1-Score terbesar yaitu 0.9979.

Berdasarkan pengujian pada confusion matrix metode XGBoost dengan menggunakan *accuracy*, *precision*, *recall* dan F1-Score dapat dilihat bahwa pada dataset balita *stunting* di kabupaten Lampung Barat mendapatkan hasil akurasi terbesar dengan hanya menggunakan parameter *learning rate* (0.9) yaitu dengan nilai *accuracy*, *precision*, *recall* dan F1-Score sebesar 0.9979. Hal ini

membuktikan bahwa pada penelitian ini yaitu dengan metode XGBoost penggabungan 6 parameter terbaik tidak mendapatkan hasil yang pengujian yang semakin baik (besar) juga.

#### 4.5 Perbandingan Algoritma SVM dan XGBoost

Setelah mendapatkan hasil pengujian dengan akurasi terbesar dari masing-masing metode yaitu SVM dan XGBoost, selanjutnya adalah melakukan perbandingan antara kedua metode tersebut. Akurasi terbesar dari masing-masing metode didapatkan setelah menggunakan beberapa parameter. Pada metode SVM didapatkan bahwa penggunaan parameter *cost* terbaik yaitu 1000 mendapatkan hasil pengujian yang lebih besar dibandingkan hanya dengan penggunaan kernel, parameter *gamma*, parameter iterasi maksimum, tanpa menggunakan parameter dan bahkan penggabungan beberapa parameter. Sama halnya pada metode XGBoost, penggunaan parameter *learning rate* terbaik yaitu 0.9 mendapatkan hasil pengujian yang lebih besar dibandingkan hanya dengan penggunaan parameter *max depth*, parameter *gamma*, parameter *min child weight*, parameter *subsample*, parameter *colsample bytree*, tanpa menggunakan parameter dan bahkan penggabungan beberapa parameter. Berikut adalah hasil akurasi terbesar dari metode SVM dan XGBoost.

Tabel 64. Perbandingan Algoritma SVM dan XGBoost

Metode	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
SVM	0.9935	0.9935	0.9934	0.9934
<i>XGBoost</i>	<b>0.9979</b>	<b>0.9979</b>	<b>0.9979</b>	<b>0.9979</b>

Tabel 65. Rata-rata waktu komputasi SVM dan XGBoost

Waktu Komputasi		
Metode	Iterasi	Pengujian
SVM	127.54 menit	18.09 menit
<i>XGBoost</i>	<b>3.7 menit</b>	<b>0.43 menit</b>

Berdasarkan tabel 64 dapat dilihat bahwa dari kedua algoritma yang digunakan yaitu *Support vector Machine (SVM)* dan *Xtreme Gradient Boosting (XGBoost)* masing-masing memiliki nilai akurasi terbesar sebesar 99,35% untuk SVM dan 99.79% untuk XGBoost, sehingga kedua metode tersebut dapat menghasilkan akurasi yang besar dengan selisih yang sangat kecil yaitu.

Selain itu, berdasarkan hasil penelitian yang telah dilakukan, didapatkan bahwa pada metode SVM hasil akurasi minimum dari masing-masing iterasi adalah sebesar 0.2695 (pada kernel *sigmoid* fold 1) dan hasil akurasi terbesar dari masing-masing iterasi adalah sebesar 0.9966 (pada parameter iterasi maksimum 10000 fold 2 dan fold 3). Sedangkan pada metode XGBoost hasil akurasi minimum dari masing-masing iterasi adalah sebesar 0.7620 (pada parameter *max depth* 1 fold 8) dan hasil akurasi terbesar dari masing-masing iterasi adalah sebesar 0.9992 (pada parameter *learning rate* 0.9 fold 7). Hal ini menunjukkan bahwa kestabilan hasil akurasi antar iterasi pada XGBoost memiliki interval yang lebih kecil apabila dibandingkan dengan metode SVM.

Tabel 65 menunjukkan bahwa metode SVM membutuhkan waktu yang lebih lama dari metode XGBoost yaitu 127.54 menit lama pencarian akurasi fold untuk masing-masing percobaan dan 18.09 menit lama waktu pengujian untuk masing-masing percobaan. Sedangkan metode XGBoost hanya membutuhkan waktu 5.38 menit lama pencarian akurasi pada fold untuk masing-masing percobaan dan 0.43 menit lama waktu pengujian untuk masing-masing percobaan.

## V. PENUTUP

### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, didapatkan dua kesimpulan yaitu sebagai berikut:

1. Model Klasifikasi SVM dan XGBoost berhasil diimplementasikan pada dataset balita penderita *stunting* di Kabupaten Lampung Barat berdasarkan data yang didapatkan dari Dinas Kesehatan Kabupaten Lampung Barat Provinsi Lampung.
2. Pada SVM dilakukan 6 kali percobaan pada data *training* dan data *testing*, yang mendapatkan hasil akurasi terbesar yaitu dari pengujian dengan hanya menggunakan *best cost* dengan akurasi sebesar 99.35%. Sedangkan pada XGBoost dilakukan 8 kali percobaan pada data *training* dan data *testing* yang mendapatkan hasil terbesar dari pengujian dengan hanya menggunakan *best learning rate* yaitu mendapatkan akurasi sebesar 99.79%. Sehingga metode XGBoost menghasilkan akurasi yang lebih besar jika dibandingkan dengan metode SVM. Selain nilai akurasi pengujian yang lebih besar, nilai akurasi yang dihasilkan pada masing-masing iterasi oleh XGBoost relatif lebih stabil pada masing-masing percobaannya jika dibandingkan dengan SVM. Keunggulan lain dari metode XGBoost adalah membutuhkan waktu komputasi yang lebih cepat jika dibandingkan dengan metode SVM.

## 5.2 Saran

Adapun saran dari penelitian yang telah dilakukan yaitu:

1. Pada penelitian selanjutnya dapat menggunakan metode *deep learning* atau metode terbaru untuk melakukan prediksi terhadap klasifikasi *stunting*.
2. Pada penelitian selanjutnya dapat melakukan penambahan data dengan memperbesar skala penelitian menjadi provinsi Lampung.

## DAFTAR PUSTAKA

- Agustin, Eviyanti, & Lutvi Azizah. 2023. Deteksi Penyakit Epilepsi Melalui Sinyal EEG Menggunakan Metode DWT Dan Extreme Gradient Boosting. *Jurnal Media Informatika Budidarma*. 7 : 117–127.
- Agustina, Furqon, & Rahayudi. 2018. Implementasi Metode Support Vector Machine (SVM) Untuk Klasifikasi Rumah Layak Huni (Studi Kasus: Desa Kidal Kecamatan Tumpang Kabupaten Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 2 : 3366–3372.
- Andryan, Fajri, & Sulistyowati. 2022. Komparasi Kinerja Algoritma Xgboost Dan Algoritma Support Vector Machine (Svm) Untuk Diagnosis Penyakit Kanker Payudara. *JIKO (Jurnal Informatika dan Komputer)*. 6 : 1.
- Aristoteles, Kusuma, Irawati, Sakethi, Suarni, Miswar, & Azhar. 2023. Development of Nursing Process Expert System for Android-Based Nursing Student Learning. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*,. 14 : 234–239.
- Fadilah, Pangestu, Lumbanbatu, & Defiyanti. 2022. Pengelompokan Kabupaten/Kota Di Indonesia Berdasarkan Faktor Penyebab Stunting Pada Balita Menggunakan Algoritma K-Means. *JIKO (Jurnal Informatika dan Komputer)*. 6 : 223.
- Firman, Masjkur, & Suhaeni. 2021. Pemodelan Klasifikasi Keterlambatan Pembayaran UKT Mahasiswa IPB Dengan Random Forest Dan AdaBoost. Givari, Sulaeman, & Umaidah. 2022. Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit. *Nuansa Informatika*. 16 : 141–149.
- Hartawan, Erkamim, Rachmawati, Santi, Legito, & Sepriano. 2023. Penerapan Algoritma Supervised Learning Untuk Klasifikasi Program Keluarga Harapan. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*. 3 : 83–91.
- Irawan. 2023. Komparasi Deteksi Kecurangan Pada Data Klaim Asuransi Pelayanan Kesehatan Menggunakan Metode Support Vector Machine (SVM) Dan Extreme Gradient Boosting (XGBoost). *Jurnal Sains Dan Seni*. 12
- Karim, Nurhadi, Setiawan, Rizky, & Br. Manurung. 2023. Pengaruh Normalisasi

- Data Pada Klasifikasi Harga Ponsel Berdasarkan Spesifikasi Menggunakan Klasifikasi Naive Bayes Dan Multinomial Logistic Regression. *Jurnal Rekayasa Elektro Sriwijaya*. 4 : 8–16.
- Kemenkes. 2020. Peraturan Menteri Kesehatan Republik Indonesia. 167 : 1–5.
- Kemenkes RI. 2022. Kemenkes RI No HK.01.07/MENKES/1928/2022 Tentang Pedoman Nasional Pelayanan Kedokteran Tata Laksana Stunting. 1–52.
- Laswati. 2019. Masalah Gizi Dan Peran Gizi Seimbang. *Agrotech : Jurnal Ilmiah Teknologi Pertanian*. 2 : 69–73.
- Lumbanraja, Lufiana, Heningtyas, & Muludi. 2022. Implementasi Support Vector Machine (Svm) Untuk Klasifikasi Penderita Diabetes Mellitus. *Jurnal Komputasi*. 10
- Lumbanraja, Sitepu, Kurniawan, & Aristoteles. 2020. Prediksi Jumlah Penderita Penyakit Tuberkulosis Di Kota Bandar Lampung Menggunakan Metode Svm (Support Vector Machine). *Klik - Kumpulan Jurnal Ilmu Komputer*. 7 : 320.
- Monika & Furqon. 2018. Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 2 : 3165–3166.
- Mudyaningasih. 2019. Prediksi Metilasi Pada Sequence Protein Arginine Menggunakan Random Forest. *Jurusan Teknik Kimia USU*. 3 : 18–23.
- Nababan. 2021. Klasifikasi Penderita Stunting Dengan Metode Support Vector Machine (Studi Kasus: Lima Puskesmas Di Kota Bandar Lampung). 1–23.
- Pattiasina, Markus, & Pattiselanno. 2022. Kajian Antropometri Pengrajin Tenun Ikat Khas Maluku. *Jurnal Simetrik*. 11 : 495–503.
- Pratiwi & Setyawan. 2021. Analisis Akurasi Dari Perbedaan Fungsi Kernel Dan Cost Pada Support Vector Machine Studi Kasus Klasifikasi Curah Hujan Di Jakarta. *Journal of Fundamental Mathematics and Applications (JFMA)*. 4 : 203–212.
- Rachmi. 2020. Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn. 1–101.
- Rahman. 2020. Implementasi Metode Svm , Mlp Dan Xgboost. *Implementasi Metode SVM, MLP dan XGboost pada Data Ekspresi Gen*.
- Rombe. 2021. Penggunaan Metode XGBoost Untuk Klasifikasi Status Obesitas Di Indonesia.
- Saputri. 2021. Optimalisasi Algoritma XGBoost Menggunakan Hyperparameter

Tuning Dan Multiple Preprocessing Untuk Prediksi Harga Mobil Bekas.

- Saputro. 2023. Penerapan Machine Learning Untuk Mengidentifikasi Faktor-Faktor Yang Mempengaruhi Kemampuan Komunikasi Matematis Pada Materi Program Linear. 1–254.
- Shafila. 2020. Implementasi Metode Extreme Gradient Boosting (Xgboost ) Untuk Klasifikasi Pada Data Bioinformatika (Studi Kasus : Penyakit Ebola , GSE 122692). 1–77.
- SSGI. 2022. Hasil Survei Status Gizi Indonesia. 77–77.
- Sudarman & Budi. 2023. Pengembangan Model Kecerdasan Mesin Extreme Gradient Boosting Untuk Prediksi Keberhasilan Studi Mahasiswa. *Jurnal STRATEGI ... 5* : 297–314.
- Sulut. 2017. Status Gizi Balita. *Profil Kesehatan Provinsi Sulawesi Utara 2016*.
- Syukron, Santoso, & Widiharih. 2020. Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*. 9 : 227–236.
- Tapikap, Djahi, & Widiastuti. 2019. Klasifikasi Spam E-Mail Menggunakan Metode Transformed Complement Naive Bayes. 7 : 21–26.
- Veronica & Tumanggor. 2022. Kinerja Teller Dalam Melakukan Standar Layanan Terkait Kepuasan Nasabah Studi Pada Kantor Kas World Trade Center Jakarta PT. Bank Negara Indonesia (Persero) Tbk. *Jurnal Sumber Daya Aparatur*. 4 : 53–72.
- Wijayanti, Furqon, & Adinugroho. 2018. Penerapan Algoritme Support Vector Machine Terhadap Klasifikasi Tingkat Risiko Pasien Gagal Ginjal. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 2 : 3500–3507.
- Zhafirah. 2023. *Penanganan Imbalance Data Dengan Random Oversampling (ROS) Pada Klasifikasi Penderita Diabetes Menggunakan Support Vector Machine (SVM)*. 1–23 hlm.