PREDIKSI STUNTING PADA BALITA DENGAN MENGGUNAKAN ALGORITMA KLASIFIKASI NAÏVE BAYES

STUNTING PREDICTION IN CHILDREN USING NAÏVE BAYES CLASSIFICATION ALGORITHM

Vega Herliansyah[1], Roswan Latuconsina[2], Ashri Dinimaharawati[3]

[1, 2,3] Universitas Telkom, Bandung
vegavga@students.telkomuniversity.ac.id [1], roswan@telkomuniversity.ac.id[2], ashridini@telkomuniversity.ac.id[3]

Abstrak

Stunting merupakan kondisi gagal pertumbuhan pada anak yaitu terhambatnya pertumbuhan pada balita yang diakibatkan dari kekurangan gizi dalam waktu yang lama. Permasalahan tersebut menjadi perhatian khusus untuk dapat mengurangi jumlah tingginya angka stunting pada balita, dengan memanfaatkan teknologi pembelajaran mesin.

Pembelajaran mesin atau *machine learning* pada pengerjaan Tugas Akhir ini, diharapkan mampu dalam memberikan solusi untuk menangani kasus tingginya angka stunting pada balita. *Machine learning* ini akan diimplementasikan ke dalam pembuatan aplikasi berbasis *mobile*. Aplikasi *mobile* yang dibangun, menggunakan algoritma *naïve bayes classification*. *Naïve bayes classification* adalah sebuah algoritma metode klasifikasi menggunakan teori probabilitas dan statistik.

Algoritma Naïve Bayes yang diimplementasikan dalam Tugas Akhir ini, memiliki performansi nilai rata-rata yaitu akurasi sebesar 64.02%, *precision* sebesar 62.16%, dan *recall* sebesar 71.86% dari hasil pengujian *confusion matrix* dengan 10% data *testing* dan 90% data *training*.

Kata Kunci: ***Machine Learning, Mobile, Naïve Bayes, Stunting.***

Abstract

Stunting is a condition of growth failure in children, namely stunted growth in toddlers caused by malnutrition for a long time. These problems are of particular concern to be able to reduce the high number of stunting in toddlers, by utilizing machine learning technology.

Machine learning in this final project, is expected to be able to provide solutions to deal with cases of increased stunting. This machine learning will be implemented into making mobile-based applications. The mobile application is built, using the naïve Bayes classification algorithm. Naïve Bayes classification is a classification method algorithm using probability theory and statistics.

The Naïve Bayes algorithm which is implemented in a mobile application in this Final Project, has an average performance value of 64.02% accuracy, 62.16% precision, and 71.86% recall from the results of the confusion matrix test with 10% data. testing and 90% of training data.

Keywords: Machine Learning, Mobile, Naïve Bayes, Stunting

1. Pendahuluan

Stunting merupakan masalah gizi kronis dengan kondisi tinggi badan anak lebih pendek dari tinggi badan umum anak seusianya. Balita dinyatakan stunting apabila standar pertumbuhan mencapai kurang dari -2 standar deviasi [1]. Saat ini angka stunting di Indonesia lebih tinggi dari ambang batas WHO. Berdasarkan hasil survei pada tahun 2019, prevalensi stunting di Indonesia mencapai angka 27.67 persen [2].

Untuk menangani kasus tingginya angka stunting pada balita, peran *machine learning* dapat digunakan dalam kasus tersebut. *Machine learning* yang dibuat akan diimplementasikan kedalam aplikasi *mobile* guna mempermudah pengguna dalam melakukan pengecekan stunting dan sebagai langkah awal dalam pencegahan stunting. Algoritma prediksi yang digunakan yaitu algoritma *Naïve Bayes Classification*.

Naïve Bayes Classification merupakan algoritma perhitungan probabilitas pada data – data yang telah didapat sebelumnya, sehingga dapat mengklasifikasikan keluaran hasil prediksi stunting atau normal pada kondisi

pertumbuhan balita. Diharapkan dari aplikasi prediksi yang dibangun, bisa menjadikan solusi terhadap tingginya kasus stunting yang terjadi.

2. Dasar Teori

2.1 Stunting

Stunting adalah suatu gambaran dari status gizi kurang pada anak balita (bayi di bawah lima tahun) yang bersifat kronik pada masa pertumbuhan dan perkembangan sejak awal kehidupan, sehingga anak terlalu pendek dibanding anak seusianya. Faktor – faktor yang berhubungan dengan stunting yaitu panjang badan dengan status pendek atau sangat pendek yang diperoleh dari hasil perhitungan nilai *Z-score*. Terdapat 2 kategori yang dapat dikategorikan dari hasil perhitungan nilai *Z-score*, yaitu kategori pendek dengan ambang batas -3 SD sampai <-2 SD dan kategori sangat pendek dengan ambang batas *Z-score* <-3SD [3]. Rumus perhitungan nilai Z score (TB/U):

$$\text{Z score} = \frac{TB\ hitung - median\ baku\ rujukan}{simpangan\ baku\ rujukan} \quad \ldots(2.1)$$

Keterangan dari formula 2.1 tersebut adalah:
Z-Score = Deviasi nilai individu dari nilai rata-rata median dibagi dengan standar deviasi referensi
TB hitung = Nilai umur dari hasil pengukuran tinggi badan balita
MBR = Hasil nilai median dari kasus yang dihitung
SBR = Hasil selisih nilai standar deviasi dari kasus yang dihitung

2.2 Preprocessing Data

Data preprocessing adalah proses manipulasi dataset sebelum di input kedalam model. Proses data preprocessing berfungsi untuk melakukan treatment awal terhadap data demi menghilangkan permasalahan-permasalahan yang dapat mengganggu hasil perediksi [4].

2.3 Naïve Bayes Classifier

Naïve bayes classifier merupakan metode pengklasifikasian suatu probabilitas dan statistik[5]. Algoritma *naïve bayes classifier* beroperasi atas dasar teori probabilitas yang melihat seluruh atribut dari data sebagai bukti dalam probabilitas. Untuk klasifikasi data kontinyu menggunakan rumus Densitas Gauss [6].

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad \ldots(2.2)$$

Keterangan dari formula (2.2) tersebut adalah:
P = Peluang
X_i = Atribut ke i
x_i = Nilai atribut ke i
Y = Kelas yang dicari
y_j = Sub kelas Y yang dicari
μ = Mean, menyatakan rata rata dari seluruh atribut
σ = Deviasi standar, menyatakan varian dari seluruh atribut

Rumus untuk menghitung mean, dapat dilihat pada rumus 2.3 berikut.

$$mean = \frac{jumlah\ data}{banyaknya\ data} \quad \ldots(2.3)$$

Rumus menghitung standar deviasi, dapat dilihat pada rumus 2.4 berikut.

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \quad \ldots(2.4)$$

Keterangan dari formula 2.4 adalah:
S = standar deviasi

N = jumlah data
i = nomor data (i =1,2,3,….N)
x_i = data ke-i (i =1,2,3,….N)
$\bar{x}$ = rata-rata sampel

2.4 Evaluasi Performansi

Terdapat metode yang dapat digunakan untuk mengevaluasi kinerja dari model yaitu dengan *confusion matrix* dan *k-fold cross validation*.

a. *Confusion Matrix*

Confusion Matrix merupakan tahap analisis dan evaluasi terhadap performansi sistem yang dirancang. Performansi diukur dengan nilai akurasi, presisi, dan *recall*[7]. Berikut merupakan tabel *confusion matrix*:

Tabel 2. 1 Kategori kombinasi *confusion matrix*.

	Predicted Class No (0)	**Predicted Class Yes Yes (1)**
Actual Class No (0)	*True Negative (TN)*	*False Positive (FP)*
Actual Class Yes (1)	*False Negative (FN)*	*True Positive (TP)*

Keterangan:
True Positive (TP) = Data positif yang diprediksi benar
True Negative (TN) = Data negatif yang diprediksi benar
False Positive (FP) = Data negatif yang diprediksi sebagai data positif
False Negative (FN) = Data positif yang diprediksi sebagai data negatif

b. *K-Fold Cross Validation*

Pengujian ini berfungsi untuk menilai kinerja proses dengan teknik membagi sampel asli secara acak menjadi sub sampel *k*. Kemudian, satu sub sampel dianggap sebagai data validasi untuk pengujian model, dan sub sampel *k-1* yang tersisa digunakan sebagai data latih. Proses ini diulang sebanyak k kali dan masing – masing sub sampel k digunakan tepat satu sebagai data validasi. Hasil *k* dari lipatan kemudian dirata rata untuk menghasilkan estimasi tunggal[8].

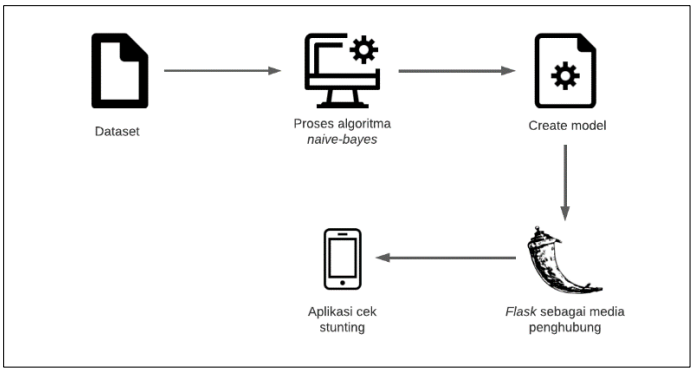
2.5 Python

Python merupakan bahasa pemrograman populer yang mendukung paradigma pemrograman berbasi objek. Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Python merupakan salah satu produk yang *opensource* juga *multiplatform* [9].

a. Scikit Learn
Scikit learn adalah modul *machine learning* yang ditulis dengan bahasa pemrograman Python dengan mengintegrasikan berbagai algoritma pembelajaran mesin. [10].
b. Flask
Flask merupakan *microweb framework* yang ditulis dengan bahasa pemrograman Python. *Flask* menggunakan dependensi werkzeug dan jinja2. *Flask* mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam *Flask* itu sendiri [11].

3. Analisis dan Perancangan Sistem

3.1 Gambaran Umum Sistem

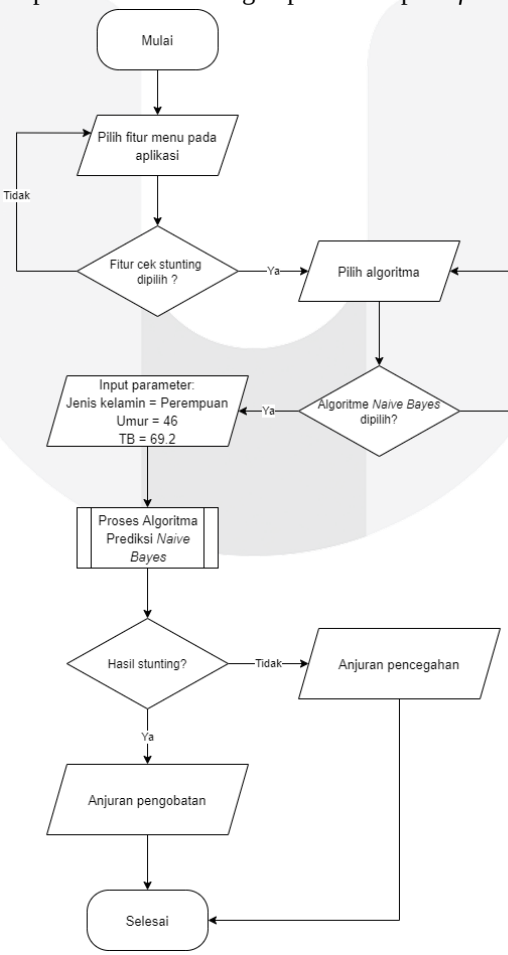


Gambar 3. 1 Gambaran umum sistem prediksi.

Proses sistem yang dibangun dari penjelasan Gambar 3.1 mulai awal hingga akhir selesainya sistem dibangun, berawal dari pengolahan dataset. Dalam proses pengolahan dataset terdapat hal – hal yang dilakukan yaitu rekap data, pembersihan data, peneyeimbangan data, dan pelabelan data. Setelah dataset selesai, selanjutnya akan dibuat *learning* sistem prediksi menggunakan algoritma perhitungan *naïve-bayes*. Algoritma tersebut akan disimpan dalam file menggunakan pemodelan *pickle*. Flask yang digunakan, berfungsi sebagai media penghubung antara aplikasi *mobile* dengan pemodelan *pickle.* Aplikasi akan meminta pengguna untuk memberikan masukkan parameter stunting berupa jenis kelamin, umur, dan tinggi badan. Setelah hasil prediksi keluar, pengguna akan dapat mengetahui saran pencegahan dan penanganan yang diberikan sistem sesuai hasil pengujian stunting.

3.2 Perancangan Sistem

Perancangan dari sistem aplikasi prediksi cek stunting dapat dilihat pada *flowchart* berikut.

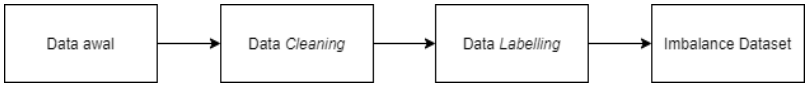


Gambar 3. 2 Flowchart sistem aplikasi prediksi.

3.3 Pemodelan *Naïve Bayes*

3.3.1 Data *Preprocessing*

Masukkan sistem yang dibangun akan diproses menggunakan algoritma *naïve bayes*. Berikut diagram model pengolahan data pada Gambar 3.3.



Gambar 3. 3 Diagram alur *preprocessing data.*

1. Data Awal

Data awal adalah data mentah yang belum melalui proses pembersihan. Data awal yang didapat berjumlah 22855 data.

2. Data *Cleaning*

Proses pembersihan (*cleaning*) data, berfungsi untuk menghapus data yang mempunyai nilai kosong atau *missing value* agar tidak mengganggu dalam proses perhitungan.

3. Data *Labelling*

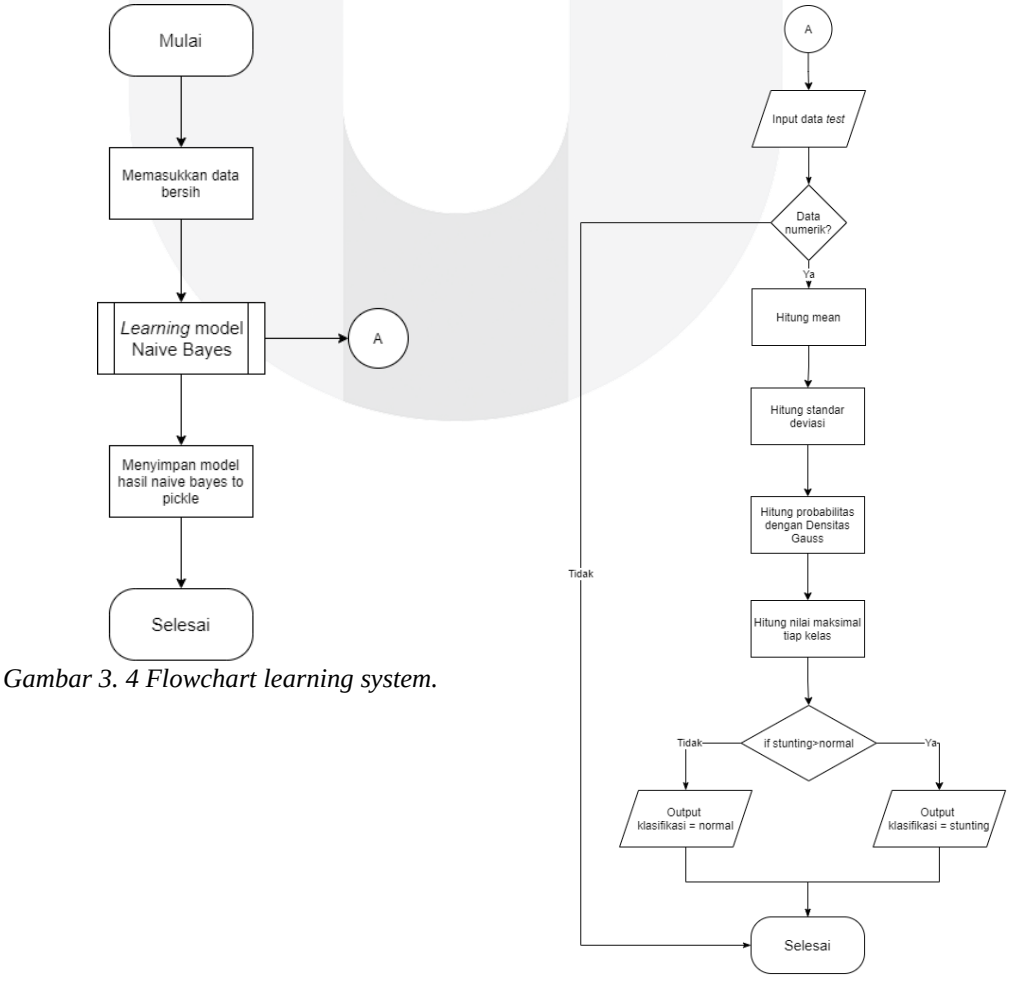
Proses pemberian label digunakan untuk mengelompokan data dengan label kelas hasil prediksi stunting atau normal. Label yang diberikan berdasarkan dari hasil perhitungan TB/U dengan memberikan kondisi jika nilai TB/U <-2 maka akan diberi label 1 (stunting) dan apabila jika nilai TB/U ≠< -2 maka diberi label 0 (normal).

4. *Imbalance* Dataset

Pada proses ini data akan diseimbangkan antara jumlah kedua kelas dari label tersebut. Teknik yang digunakan dalam menyeimbangkan data yaitu menggunakan Teknik *Undersampling*. Pada teknik ini data dari kelas mayoritas akan dibuang, sehingga jumlah data kelas mayoritas akan mendekati jumlah pada kelas minoritas.

3.3.2 Proses Prediksi

Proses prediksi dengan algoritma *naïve bayes* dapat dilihat pada diagram alir berikut.



Gambar 3. 4 Flowchart learning system.

Gambar 3. 5 Lanjutan *flowchart learning system.*

4. Implementasi dan Pengujian

4.1 Dataset

Dataset yang digunakan berasal dari UPT Puskesmas Pitu yang berlokasi di Ngawi. Data pengecekan balita pada puskesmas tersebut, merupakan data informasi lengkap mengenai pemeriksaan informasi gizi pada balita. Informasi data yang direkap berjumlah sebanyak 22.855 data.

4.2 Pengujian Fungsional

Pengujian ini berfokus pada sisi fungsionalitas, yang melihat kesesuaian masukan dan keluaran aplikasi yang dibuat dengan perencanaan.

Tabel 4. 1 Hasil pengujian membuka aplikasi.

Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Membuka Aplikasi	Dapat masuk ke halaman menu	Aplikasi dapat terbuka dengan menampilkan animasi logo dan tulisan pada halaman *splash screen,* dan setelah itu masuk halaman menu	[√] Berhasil [X] Tidak Berhasil

Tabel 4. 2 Hasil pengujian membuka menu cek stunting.

Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Membuka Menu Cek Stunting	Menampilkan menu cek stunting	Aplikasi dapat menampilkan pilihan algoritma dan pengguna dapat memilih pilihan algoritma yang tersedia.	[√] Berhasil [X] Tidak Berhasil

Tabel 4. 3 Hasil pengujian *input* prediksi *naïve bayes.*

Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Input Prediksi *Naïve Bayes*: Jenis Kelamin = Perempuan Umur = 46 Tinggi Badan = 96.2	Dapat memvalidasi *input* sesuai ketentuan sistem.	Muncul pesan *error* pada parameter masukan, karena tidak sesuai dengan ketentuan *input,* dan sistem tidak dapat memproses hasil prediksi.	[√] Berhasil [X] Tidak Berhasil

4.3 Pengujian Partisi Data

Berikut tabel hasil perhitungan pada pengujian partisi 1 sampai 3 dapat dilihat pada Tabel 4.4.

Tabel 4.4 tabel hitung performa uji partisi data.

Pengujian	Data *Training*	Data *Testing*	Akurasi	Presisi	*Recall*	*F1 Score*
1	90%	10%	64.02%	62.16%	71.86%	66.67%
2	70%	30%	60.67%	58.63%	70.39%	64%
3	50%	50%	60.37%	58.60%	71.25%	64%

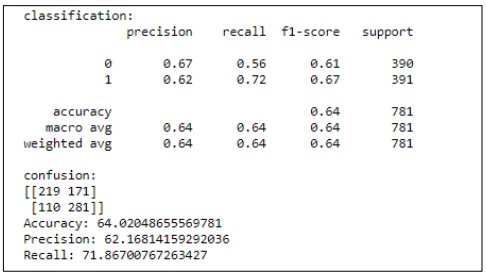
Berdasarkan hasil hitung performa uji partisi data pada Tabel 4.4, dapat ditarik kesimpulan bahwa rasio partisi data *training* dan data *testing* yang paling optimal, yaitu pada data *training* 90% dan data *testing* 10%.

4.4 Pengujian Sistem Prediksi

Pengujian sistem prediksi yang dilakukan, yaitu dengan memberikan *input* parameter yaitu jenis kelamin perempuan (2), umur 46 bulan, dan tinggi badan 96.2Cm.

4.5.2 Uji Evaluasi Performa *Naïve Bayes*

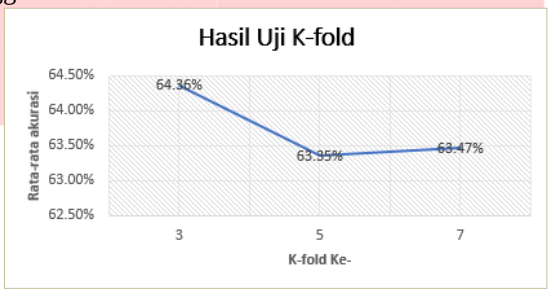
1. Uji Evaluasi Performa Menggunakan *Confusion Matrix*



Gambar 4. 1 Uji evaluasi performa menggunakan *confusion matrix.*

Pada Gambar 4.1 diatas, hasil evaluasi berdasarkan nilai *accuracy*, *precision*, dan *recall* dengan kasus data yang digunakan dalam algoritma tersebut, didapatkan nilai *accuracy* yang diperoleh yaitu sebesar 64.02%, *precision* sebesar 62.16%, dan *recall* sebesar 71.86% dengan 90% data *training* dan 10% data *testing*

2. Uji Evaluasi Performa Menggunakan *K-Fold Cross Validation*



Gambar 4. 2 Hasil uji *K-fold.*

Pada Gambar 4.2 nilai akurasi hasil pengujian *k-fold cross validation* yang paling optimal yaitu pada pengujian ke 1 dengan rata – rata akurasi sebesar 64.36%.

4.5.3 Hasil Uji Validitas

Hasil uji validitas merupakan hasil dari pengujian keluaran prediksi dari aplikasi dengan membandingkan dengan nilai *z-score.* Kesimpulan dari hasil uji validitas ini yaitu untuk mengetahui hasil prediksi dari sistem valid atau tidak valid.

Tabel 4.5 Hasil uji validitas.

No	SEX	Umur	TB	Aplikasi Prediksi	Nilai *Z-Score*	Keterangan
1	2	8	66.7	0	-1.77273	Valid
2	1	29	87.4	1	-1.11765	Tidak Valid
3	2	56	101.3	0	-1.44444	Valid
4	1	9	69.8	1	-0.95652	Tidak Valid
5	1	51	99.8	0	-1.2093	Valid
6	2	39	93.4	0	-1.21053	Valid
7	1	40	94.9	0	-0.94872	Valid
8	2	55	100.8	0	-1.45455	Valid
9	2	54	100.5	0	-1.40909	Valid
10	1	6	65.6	0	-0.95238	Valid
11	2	57	101.8	0	-1.44444	Valid
12	1	11	71.6	1	-1.26087	Tidak Valid
13	1	41	95.4	0	-0.97436	Valid
14	1	13	74.2	1	-1.125	Tidak Valid
15	1	36	91.4	0	-1.27027	Valid
16	2	43	94.7	0	-1.425	Valid
17	2	55	101	0	-1.40909	Valid
18	2	33	88.2	1	-1.63889	Tidak Valid
19	2	47	96.6	0	-1.47619	Valid
20	2	16	75.8	1	-1.69231	Tidak Valid

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari pembahasan yang telah diuraikan sebelumnya, dapat diambil suatu kesimpulan, yaitu:

1. Aplikasi ini dapat digunakan sebagai langkah awal pencegahan kondisi stunting dengan cara memprediksi dari hasil input pengguna. Berdasarkan hasil pengujian fungsional, aplikasi prediksi stunting pada balita telah 100% berjalan sesuai dengan rencana.
2. Metode *Naïve Bayes Classification* yang digunakan dalam mengklasifikasikan status pertumbuhan balita, berdasarkan dari hasil pengujian *confusion matrix* mendapatkan nilai akurasi yaitu sebesar 64.02% dan pada pengujian *K-fold cross validation* mendapatkan nilai rata – rata akurasi sebesar 64.36%.

5.2 Saran

Saran untuk pengembangan aplikasi ini yaitu:

1. Menggunakan algoritma selain *Naïve Bayes Classification* pada sistem prediksi stunting pada balita.
2. Menambahkan fungsi *database* pada aplikasi, untuk menyimpan data informasi pengguna, dan *history* pengecekan stunting.

Referensi

[1] A. D. N. Yadika, K. N. Berawi, and S. H. Nasution, "Pengaruh Stunting terhadap Perkembangan Kognitif dan Prestasi Belajar," *J. Major.*, vol. 8, no. 2, pp. 273–282, 2019.

[2] F. C. Farisa, "Terawan: Angka Stunting di Indonesia Lebih Tinggi dari Ambang Batas WHO," *www.kompas.com*, 2020. https://nasional.kompas.com/read/2020/11/19/17020401/terawan-angka-stunting-di-indonesia-lebih-tinggi-dari-ambang-batas-who (accessed Mar. 05, 2021).

[3] Kemenkes RI, "KEPMENKES RI Tentang Standar Antropometri Penilaian Status Gizi Anak," *Jornal de Pediatria*, vol. 95, no. 4. p. 41, 2011.

[4] S. Garcia, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining. Intelligent Systems Reference Library. 2015*, vol. 10. 2015.

[5] A. A. S. Maulana Aditya Rahman, Nurul Hidayat, "Tampilan Komparasi Metode Data Mining K-Nearest Neighbor Dengan Naïve Bayes Untuk Klasifikasi Kualitas Air Bersih (Studi Kasus PDAM Tirta Kencana Kabupaten Jombang).pdf." 2018.

[6] Bustami, "Penerapan Algoritma Naive Bayes untuk Mengklasifikasi Data Nasabah," *TECHSI J. Penelit. Tek. Inform.*, vol. 4, pp. 127–146, 2010.

[7] S. Agarwal, *Data mining: Data mining concepts and techniques*. 2012.

[8] A. Rohani, M. Taki, and M. Abdollahpour, "A novel soft computing model (Gaussian process regression with K-fold cross validation) for daily and monthly solar radiation forecasting (Part: I)," *Renew. Energy*, vol. 115, pp. 411–422, 2018, doi: 10.1016/j.renene.2017.08.061.

[9] A. Wahhah, "Pengenalan Python," *Teknik*, vol. 1, no. 1, pp. 1–22, 2000.

[10] A. Abraham *et al.*, "Machine learning for neuroimaging with scikit-learn," *Front. Neuroinform.*, vol. 8, no. FEB, pp. 1–10, 2014, doi: 10.3389/fninf.2014.00014.

[11] G. Prehandayana, W. Yahya, and H. Nurwasito, "Implementasi Struktur Data Dictionary Untuk Sistem Monitoring Perangkat Internet of Things," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3466–3473, 2018.