

KLASIFIKASI STUNTING MENGGUNAKAN METODE *HYPERPARAMETER TUNING* BERBASIS *PROGRESSIVE WEB APPLICATION (PWA)*

Rayhan Zulfitri Dwi Cahyo

Program Studi Teknik Informatika, Universitas PGRI Madiun

email: rayhan_1905101025@mhs.unipma.ac.id

Abstract: *Toddler growth is an important health issue that requires serious attention, especially in the context of stunting which can have long-term impacts on children's physical and cognitive development. In an effort to improve the effectiveness of stunting detection in toddlers, this study aims to create an E-Stunting application using the Neural Network (NN) artificial intelligence model with a hyperparameter tuning method using the grid search technique. This application uses Progressive Web Application (PWA) technology which has the advantages of being accessible offline, easy installation, and automatic updates. The results of hyperparameter tuning optimization using the grid search technique produced an accuracy of 98% with a hyperparameter batch size of 32, dropout 0.2, and ADAM optimizer. The E-Stunting application is expected to be an effective tool in early detection of stunting in toddlers, allowing for faster and more targeted interventions. Thus, this application is expected to contribute to efforts to address stunting problems and support improving the quality of children's health.*

Keywords: *Machine Learning, Hyperparameter tuning, Progressive Web Application, Stunting*

Abstrak: Pertumbuhan Balita merupakan isu kesehatan penting yang memerlukan perhatian serius, terutama dalam konteks stunting yang dapat memiliki dampak jangka panjang terhadap perkembangan fisik dan kognitif anak. Dalam upaya untuk meningkatkan efektivitas pendeteksian stunting pada balita, penelitian ini bertujuan untuk membuat aplikasi *E-Stunting* menggunakan model kecerdasan buatan *Neural Network (NN)* dengan metode *hyperparameter tuning* menggunakan teknik *grid search*. Aplikasi ini menggunakan teknologi *Progressive Web Application (PWA)* yang memiliki kelebihan bisa diakses secara *offline*, instalasi mudah, dan pembaruan otomatis. Hasil dari optimasi *hyperparameter tuning* menggunakan teknik *grid search* menghasilkan akurasi sebesar 98% dengan *hyperparameter batch size 32, dropout 0.2, dan optimizer ADAM*. Aplikasi *E-Stunting* diharapkan dapat menjadi alat yang efektif dalam pendeteksian dini stunting pada balita, sehingga memungkinkan intervensi yang lebih cepat dan tepat sasaran. Dengan demikian, aplikasi ini diharapkan dapat berkontribusi dalam upaya penanganan masalah stunting dan mendukung peningkatan kualitas kesehatan anak-anak.

Kata kunci: *Machine Learning, Hyperparameter tuning, Progressive Web Application, Stunting*

Pendahuluan

Dalam era digital saat ini, teknologi informasi telah menjadi bagian tak terpisahkan dari berbagai aspek kehidupan manusia. Hal ini juga berdampak pada sektor kesehatan, dimana teknologi kecerdasan buatan mulai digunakan untuk meningkatkan efisiensi dalam pemantauan kesehatan masyarakat, terutama pada tahap pertumbuhan balita. Anak-anak balita memerlukan perhatian khusus dalam pemantauan pertumbuhan mereka, terutama untuk mencegah masalah *stunting* yang merupakan masalah serius di banyak negara, termasuk Indonesia.

Posyandu Balita Rambutun Kelurahan Oro-oro Ombo Kecamatan Kartoharjo Kota Madiun menjadi pusat penting dalam pelayanan kesehatan masyarakat di daerah tersebut. Melalui Posyandu ini, data pertumbuhan anak balita terkumpul secara teratur. Namun dalam pengecekan status pertumbuhan anak balita masih diperlukan inovasi teknologi yang lebih maju dan efisien.

Dalam menanggapi permasalahan yang sudah diterangkan sebelumnya, peneliti melakukan studi dengan merujuk pada berbagai penelitian sebelumnya. Pada penelitian pertama mengenai penerapan metode *hyperparameter tuning* menggunakan teknik *grid search* dan *cross-validation* pada klasifikasi diabetes menggunakan model *XGBoost* yang memperoleh nilai akurasi 0,772 dan model *Decision tree* yang memiliki nilai terendah yaitu 0,701. Meski demikian teknik *grid search* memakan waktu lebih lama jika dilakukan penambahan pada *hyperparameter* dan nilainya (Nugraha & Sasongko, 2022).

Pada penelitian kedua penggunaan teknologi website terbaru menggunakan *Progressive Web Application (PWA)* menjadi fokus utama untuk mengimplementasikan hasil dari pelatihan model. Pada penelitian kedua testing dilakukan pada sebuah situs uji performa aplikasi *website* menggunakan *Lighthouse* dan *GTMetrix*. Performa website meningkat sebesar 23% dari 66% sebelum mengimplementasikan PWA menjadi 89% setelah mengimplementasikan PWA. Terlebih lagi website juga dapat *diinstall* layaknya aplikasi *native* dan dapat diakses secara *offline* (Phie Joarno et al., 2022).

Pada penelitian ketiga ini didasarkan pada studi sebelumnya yang mengusulkan model Intrusion Detection System (IDS) berbasis deep learning (DL) yang menggunakan Denoising Autoencoder (DAE) dan Deep Neural Network (DNN). Studi tersebut menunjukkan bahwa optimasi hiperparameter otomatis secara signifikan dapat meningkatkan performa dalam klasifikasi serangan pada sistem keamanan siber. Hasil yang diperoleh dari studi tersebut, khususnya terkait pentingnya pemilihan struktur jaringan dan hiperparameter yang optimal (Kunang et al., 2021).

Dari penelitian sebelumnya, dapat diketahui bahwa belum ada penelitian yang membahas perancangan aplikasi berbasis *Progressive Web Application (PWA)* yang mengimplementasikan model kecerdasan buatan *Neural Network (NN)* dengan metode *hyperparameter tuning* menggunakan teknik *grid search* dalam aplikasi website. *Neural Network* adalah model matematis yang meniru cara kerja jaringan saraf di otak manusia dalam memproses informasi dan sering digunakan untuk metode kecerdasan buatan *Deep Learning* (Karno et al., 2020). Pendekatan ini terutama digunakan untuk menyelesaikan masalah yang berkaitan dengan prediksi dan pengambilan keputusan (Ramadhan et al., 2020).

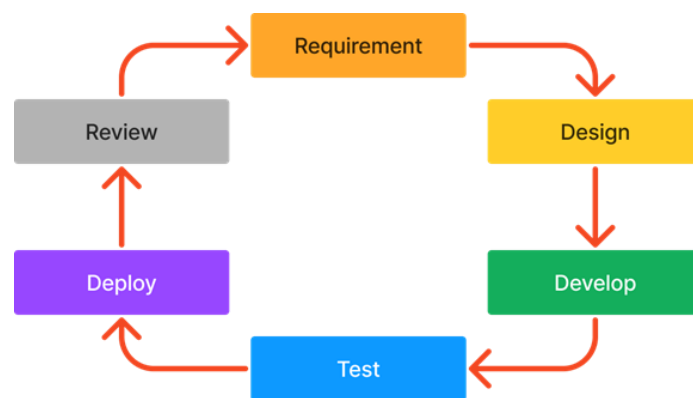
Penerapan teknologi *Progressive Web Application (PWA)* pada sistem klasifikasi stunting, yang didukung oleh model kecerdasan buatan, menawarkan berbagai keuntungan. Fitur akses *offline* pada PWA memungkinkan pengguna untuk mengunjungi website hanya satu kali untuk memperoleh akses ke fitur *offline*. Dengan cara ini, pengguna dapat melakukan klasifikasi stunting dengan lebih mudah tanpa perlu khawatir tentang kualitas jaringan yang kurang stabil.

Metode

Metode Pengembangan Sistem

Penelitian ini mengadopsi *Agile* untuk merancang sistem yang lebih mudah ditinjau dan direvisi. Pendekatan ini terbukti lebih efektif dalam pengembangan proyek dibandingkan metode desain terstruktur (Setiawan et al., 2022). Proses pengembangan *Agile* melibatkan enam tahapan. *Agile* adalah suatu metodologi pengembangan perangkat lunak yang mengedepankan fleksibilitas dan kolaborasi tim dalam menghadapi perubahan kebutuhan pengguna (Susilo & Azimah, 2023). Metodologi ini menekankan pada pengembangan iteratif dan inkremental dari metode tradisional, di mana perangkat lunak dikembangkan dalam siklus pendek yang disebut sprint (Habel & Levitar, 2023). Setiap sprint menghasilkan peningkatan fungsionalitas sehingga membuat tingkat keberhasilan pengembangan proyek jadi lebih baik (Puspitasari & Anardani, 2023).

Tahapan penelitian *Agile* untuk pengembangan aplikasi klasifikasi stunting dapat dilihat pada gambar 1.



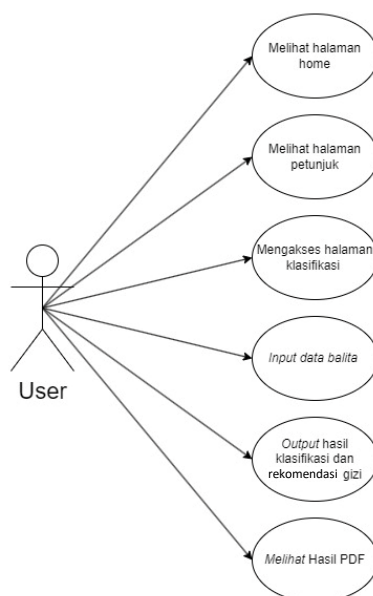
Gambar 1. Metode *agile*

Hasil dan Pembahasan

Pada tahap membangun dan mengembangkan sistem *E-Stunting*, diperlukan beberapa persyaratan penting agar implementasinya dapat berjalan dengan baik sesuai dengan kebutuhan yang ada. Sistem ini dibuat menggunakan bahasa pemrograman *Typescript* dengan *framework Next.js*, *package Tensorflow Js*, dan menggunakan teknologi *Progressive Web Application* yang dapat diakses melalui *browser* dan dapat berjalan secara *offline*. Untuk melatih model kecerdasan buatan yang akan dipasang pada aplikasi nanti dibutuhkan bahasa pemrograman *python* sebagai bahasa pemrograman untuk merancang model kecerdasan buatan dan menggunakan *text editor Jupyter Notebook* yang dapat diakses melalui *google colab* untuk mendukung keterbatasan *device* dalam membuat model kecerdasan buatan. Untuk menjalankan sistem ini, diperlukan persyaratan fungsional dan non-fungsional yang harus dipenuhi agar sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan. Adapun kebutuhan fungsional dan non-fungsional pada penelitian ini adalah sebagai berikut.

Perancangan

Use case diagram adalah sebuah uraian dari sekelompok yang berkaitan dan membentuk sistem teratur yang dilakukan dan diawasi oleh seorang aktor. Berikut merupakan fungsi-fungsi aplikasi *E-Stunting* yang dimuat dalam *use case* diagram yang dapat dilihat pada gambar 2.



Gambar 2. *Use case* pengguna

Pada gambar diatas menyampaikan bahwa *user* atau pengguna dapat melihat halaman *home*, melihat halaman petunjuk, mekakses halaman klasifikasi, input data balita mulai dari tanggal lahir balita, jenis kelamin, dan tinggi badan (cm), melihat *output* hasil klasifikasi dan rekomendasi gizi, dan melihat hasil PDF.

Datasets

Data yang digunakan dalam penelitian ini berasal dari dua sumber. Pertama, data primer dikumpulkan langsung dari Posyandu Rambutan, meliputi pengukuran tinggi badan, jenis kelamin, tanggal lahir balita. Kedua, data sekunder diperoleh dari dataset yang tersedia pada *platform* Kaggle berjudul, "*Stunting Toddler (Balita) Detection (121 rows)*", yang disedian oleh Rendi Putra Pradana dataset ini berisi informasi 121 ribu balita meliputi umur (bulan), jenis kelamin, tingi badan (cm), dan status gizi yang berisikan 4 klasifikasi *severly stunting*, *stunting*, *normal*, dan *tinggi*. Dapat dilihat pada tabel 1 adalah data dari Posyandu Rambutan dan pada tabel 2 adalah *dataset* "*Stunting Toddler (Balita) Detection (121 rows)*".

Tabel 1. Data balita posyandu balita

No	Umur (bulan)	Jenis Kelamin	Tinggi Badan	Status Gizi
1	60	laki-laki	105	normal
2	60	laki-laki	107	normal
3	60	laki-laki	115	normal
4	60	perempuan	102	normal
5	49	perempuan	89	<i>severely stunted</i>
...
39	0	perempuan	51	normal

Tabel 2. Tabel dataset

No	Umur (bulan)	Jenis Kelamin	Tinggi Badan	Status Gizi
1	0	laki-laki	44.59197329	stunted
2	0	laki-laki	56.70520337	tinggi
3	0	laki-laki	46.8633576	normal
4	0	laki-laki	47.50802563	normal
5	0	laki-laki	42.74349389	<i>severely stunted</i>
...
120000	60	perempuan	109.8	normal

Preprocessing Data

Sebelum melakukan training pada model, dilakukan *preprocessing* untuk membersihkan *dataset* yang berjumlah 121.000 data. Proses ini mencakup pengecekan kolom label menggunakan *Z-Score* tinggi badan per umur (dalam bulan) untuk menemukan *Standard Deviation (SD)*. Rumus *Z-Score* dapat dilihat pada rumus 1 sebagai berikut.

$$Z - Score = \frac{(TB \text{ anak} - \text{median } TB \text{ standar})}{\text{standar deviasi (SD) } TB \text{ standar}}$$

Setelah mengetahui hasil dari *Z-Score* data akan dicek kesesuaian dengan deskripsi pada *dataset*. Jika tidak sesuai dengan deskripsi label dihapus dan menyisakan 120.373 baris. Setelah itu, dilakukan pelabelan dan penyatuan dataset dengan data posyandu yang telah dikumpulkan. Dataset yang telah dibersihkan dapat dilihat pada tabel 3.

Tabel 3. Hasil penggabungan data

Index	Umur (bulan)	Jenis Kelamin	Tinggi Badan (cm)	Status Gizi
0	0	laki-laki	44.591973	1
1	0	laki-laki	56.705203	3
2	0	laki-laki	46.863358	2
3	0	laki-laki	47.508026	2
4	0	laki-laki	42.743494	0
...
120372	60	perempuan	109.800000	2

Perancangan Hyperparameter

Sebelum membangun arsitektur model dan melatihnya, langkah krusial yang perlu dilakukan adalah merancang strategi *hyperparameter tuning*. *Hyperparameter* adalah parameter-parameter yang nilainya tidak dipelajari oleh model selama pelatihan, tetapi harus ditentukan sebelumnya. Tujuan dari *tuning* adalah menemukan kombinasi nilai *hyperparameter* yang optimal untuk mencapai performa model terbaik. Perancangan *Hyperparameter Tuning* dapat dilihat pada tabel 4.

Tabel 4. Tabel perancangan *hyperparameter*

<i>Batch size</i>	<i>dropout</i>	<i>optimizer</i>
32	0.2	<i>ADAM</i>
64	0.5	<i>SGD</i>

Dapat dilihat pada gambar 3 ruang pencarian *hyperparameter tuning* yang divisualkan menggunakan *scatter plot* dimana terdapat 2 visual *ADAM* dan *SGD* dimana pada masing-masing visual terdapat 4 titik yang merepresentasikan 2 nilai dari *dropout* (0.2, dan 0.5) dan 2 nilai dari *batch size* (32 dan 64). Dari keterangan visual sebelumnya dapat diketahui berapa model yang akan tercipta dari optimasi yang sudah divisualkan. Berikut rumus *hyperparameter tuning* dengan teknik *grid search* pada rumus 2.

$$\text{Total model} = H_1 \times H_2 \times H_3 \times \dots$$

Rumus 2 adalah rumus *hyperparameter tuning* dengan teknik *grid search* untuk mencari total model yang akan tercipta dari optimasi pengaturan *hyperparameter* dan nilainya. Dimana H_1 mewakili jumlah nilai untuk *hyperparameter* ke- i .

Pelatihan

Arsitektur model sangat penting untuk menentukan hasil akurasi yang tinggi ini juga bergantung pada optimasi *hyperparameter*. Arsitektur *Neural Network* yang digunakan adalah arsitektur piramida. Tujuannya adalah untuk mengidentifikasi optimasi dengan akurasi yang paling tinggi yang akan digunakan pada sistem. Pada penelitian ini diketahui bahwa optimasi *hyperparameter* yang paling bagus berada pada *batch size* dengan nilai 32, *dropout* dengan nilai 0.2, dan menggunakan *optimizer ADAM* dengan akurasi 98%. Dan optimasi dengan tingkat akurasi yang paling rendah berada pada *batch size* dengan nilai 64, *dropout* dengan nilai 0.5, dan menggunakan *optimizer SGD* dengan tingkat akurasi sebesar 55%. Model yang sudah dilatih dan memiliki akurasi tinggi akan digunakan pada sistem yang akan dirancang.

Hasil Optimasi Hyperparameter

Saat melakukan pelatihan dan validasi pada model dan *hyperparameter* yang sudah dirancang sebelumnya, terdapat 8 model hasil dari perancangan *hyperparameter* dan total model ditentukan pada rumus 2. Berikut adalah akurasi dari tiap-tiap model yang sudah dilatih pada tabel 5.

Tabel 5. Hasil akurasi optimasi *hyperparameter*

<i>batch_size</i>	<i>dropout</i>	<i>optimizer</i>	<i>accuracy</i>
32	0.2	<i>ADAM</i>	0.982726
64	0.2	<i>ADAM</i>	0.956565
64	0.5	<i>ADAM</i>	0.952413
32	0.5	<i>ADAM</i>	0.895773
32	0.2	<i>SGD</i>	0.682086
32	0.5	<i>SGD</i>	0.558093
64	0.2	<i>SGD</i>	0.558093
64	0.5	<i>SGD</i>	0.558093

Hasil Uji Model Terbaik

Dalam tes atau evaluasi kinerja model terbaik dalam penelitian ini, dilakukan pengukuran menggunakan *classification report*. Pada *classification report*, beberapa metrik seperti *precision*, *recall*, *f1-score*, dan *accuracy* dihitung untuk memberikan gambaran yang lebih komprehensif mengenai performa model dalam mengklasifikasikan data. Dapat dilihat pada tabel 6.

Tabel 6. *Classification report*

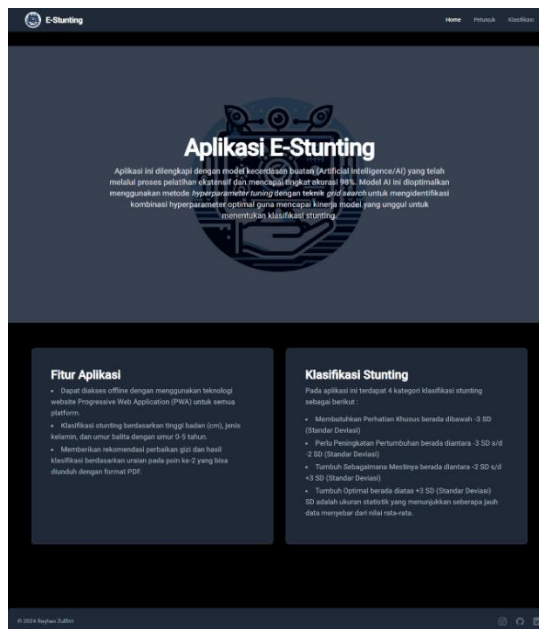
<i>Class</i>	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
0	0.9655	0.99949	0.982201	1960
1	0.986933	0.923077	0.953938	1391
2	0.986688	0.992708	0.989689	6720
3	0.984103	0.974112	0.979082	1970

<i>accuracy</i>	0.982726	0.982726	0.982726	0.982726
<i>macro avg</i>	0.980806	0.972347	0.976227	12041
<i>weighted avg</i>	0.982845	0.982726	0.982605	12041

Tabel 6 menerangkan laporan klasifikasi menilai kinerja model dengan menunjukkan *precision*, *recall*, *f1-score*, dan *support* untuk setiap kelas (0, 1, 2, 3). *Precision* mengukur akurasi prediksi untuk kelas tertentu, *recall* menunjukkan kemampuan model dalam mendeteksi semua contoh dari kelas tersebut, dan *f1-score* adalah keseimbangan antara *precision* dan *recall*. *Support* adalah jumlah data aktual untuk setiap kelas. Selain itu, tabel menyertakan *accuracy* yang mencerminkan persentase prediksi benar secara keseluruhan, *macro average* yang memberikan rata-rata metrik di seluruh kelas tanpa mempertimbangkan ukuran kelas, dan *weighted average* yang menghitung rata-rata metrik dengan mempertimbangkan ukuran kelas. *Macro avg* dan *weighted avg support* mencerminkan total jumlah data dalam dataset uji, yaitu 12041, yang merupakan jumlah agregat dari semua kelas.

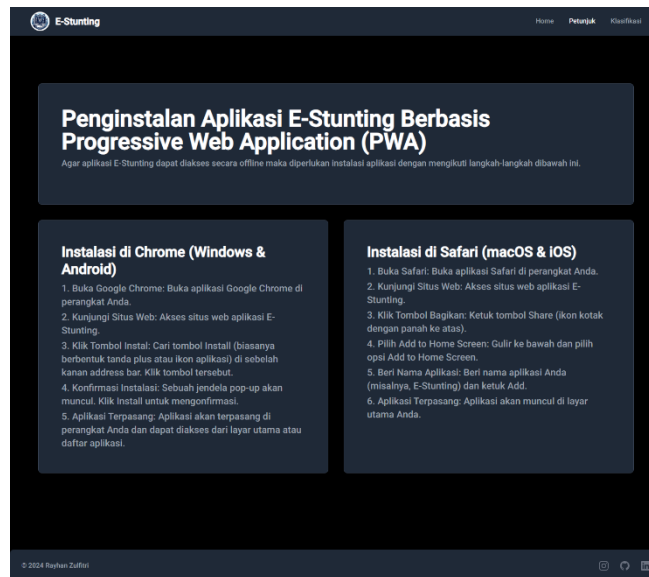
Antarmuka sistem

Setelah menganalisis dan merancang aplikasi maka tahap selanjutnya adalah implementasi. Pada tahap ini peneliti akan menjelaskan hasil eksekusi rancangan aplikasi web. Pertama adalah halaman *home* yang dapat dilihat pada gambar 3.



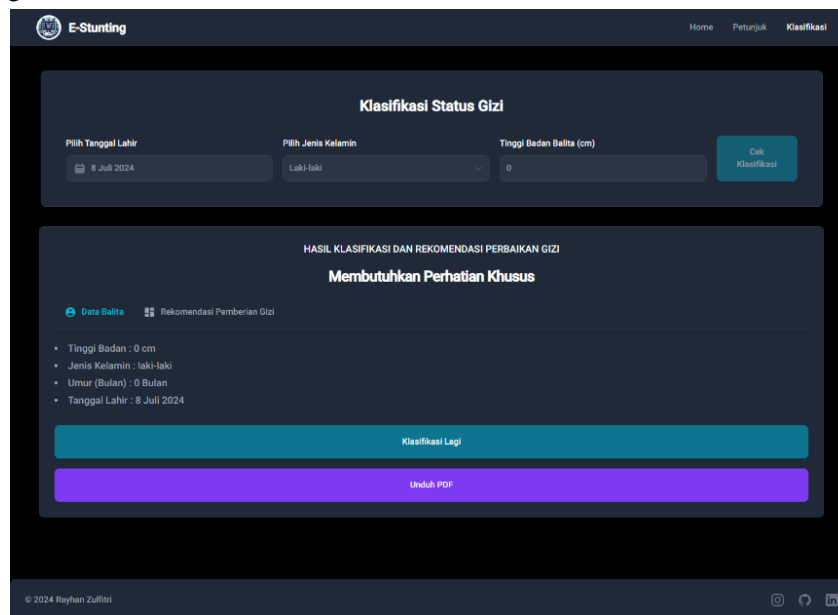
Gambar 3. Halaman *home*

Pada halaman *home* akan menampilkan informasi terkait aplikasi web berupa deskripsi singkat, fitur aplikasi, dan informasi hasil klasifikasi. Untuk fitur aplikasi terdapat tiga poin yaitu aplikasi bisa berjalan secara *offline*, menerangkan batasan umur yang bisa diklasifikasikan oleh aplikasi, dan memberikan rekomendasi gizi yang dapat diunduh menjadi file PDF. Pada informasi klasifikasi ini memberitahukan kepada pengguna deviasi standar dari 4 klasifikasi yang ada. Berikut merupakan implementasi halaman *home* pada penelitian ini. Berikutnya adalah implementasi halaman petunjuk dapat dilihat pada gambar 4.



Gambar 4. Halaman petunjuk

Halaman ini bertujuan untuk memberitahu kepada pengguna bagaimana cara pemasangan aplikasi pada empat *device* yang berbeda dengan 2 aplikasi *browser* yang berbeda juga, untuk *windows* dan *android* diberikan petunjuk pemasangan melalui *Chrome* dan untuk *iOS* dan *macOS* diberikan petunjuk pemasangan melalui *Browser Safari*. Berikut adalah implementasi halaman petunjuk pada penelitian ini. Yang terakhir adalah halaman paling inti dari aplikasi web yaitu halaman klasifikasi yang dapat dilihat pada gambar 5.



Gambar 5. Halaman klasifikasi

Halaman ini adalah halaman inti dari aplikasi *E-Stunting* karena proses klasifikasi dan rekomendasi gizi akan diproses dan ditampilkan pada halaman ini. Sebelum pengguna disajikan komponen hasil klasifikasi dan rekomendasi gizi pengguna akan diminta untuk memasukkan data balita mulai dari tanggal lahir, jenis kelamin, dan tinggi badan. Setelah data balita pengguna akan menekan tombol `Cek Klasifikasi` yang nantinya akan diproses oleh model kecerdasan buatan untuk mengeluarkan klasifikasi dan sistem akan memberikan rekomendasi gizi. Berikut adalah implementasi halaman klasifikasi.

Simpulan

Penelitian ini menghasilkan aplikasi web dengan fitur *Progressive Web Application (PWA)* yang digunakan untuk klasifikasi dan rekomendasi gizi pada balita, dengan model *Neural Network* dengan

metode *Hyperparameter Tuning*. Dalam prosesnya, aplikasi ini menggunakan teknik *Grid Search* untuk menentukan *hyperparameter* terbaik dan dirancang dengan metode *Agile* yang berorientasi pada pemrograman berorientasi objek. Aplikasi ini dikembangkan menggunakan bahasa pemrograman *TypeScript* dan *Python*, serta berbagai *tool* seperti *Visual Studio Code* dan *Google Colab*. *Framework* yang digunakan termasuk *Next Js*, *TensorFlow*, dan *TensorFlow.js*, didukung oleh sejumlah *package* dan *library* lain. Aplikasi ini tidak hanya memudahkan petugas posyandu dalam menentukan klasifikasi stunting pada balita, tetapi juga dapat digunakan oleh masyarakat umum, meskipun implementasinya dibatasi pada area penelitian tertentu. Model *Neural Network* yang diterapkan pada aplikasi ini dirancang dengan arsitektur jaringan saraf piramida, yang terbukti mampu mengklasifikasikan stunting dengan tingkat akurasi sebesar 98%, serta memiliki keunggulan dalam memprediksi data baru.

Daftar Pustaka

- Ahmad, G. N., Fatima, H., Shafiullah, Salah Saidi, A., & Imdadullah. (2022). Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques with and Without GridSearchCV. *IEEE Access*, 10(March), 80151–80173. <https://doi.org/10.1109/ACCESS.2022.3165792>
- Andriana, Dickya, Jefikaa, M., Kosasia, H., Prayogia, G., Williama, & Dharmaa, A. (2020). Prediksi Gelombang Corona dengan Metode Neural Network. *JIKOMSI (Jurnal Ilmu Komputer Dan Sistem Informasi)*, 3(2), 102–107.
- Bartz, E., Bartz-Beielstein, T., Zaefferer, M., & Mersmann, O. (2023). Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide. In *Hyperparameter Tuning for Machine and Deep Learning with R: a Practical Guide*. <https://doi.org/10.1007/978-981-19-5170-1>
- Birchha, V., & Nigam, B. (2022). Data Splitting Techniques to Reduce False-Positive and False-Negative Cases in Breast Cancer Prediction. 20(8), 9410–9424. <https://doi.org/10.14704/NQ.2022.20.8.NQ44961>
- Daqiqil, I. (2021). *MACHINE LEARNING: Teori, Studi Kasus, dan Implementasi Menggunakan Python*. Badan Penerbit Universitas Riau.
- Decastro-García, N., Muñoz Castañeda, Á. L., Escudero García, D., & Carriegos, M. V. (2019). Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm. *Complexity*, 2019. <https://doi.org/10.1155/2019/6278908>
- Fanny, F. (2023). Sistem Pengenalan Buah menggunakan Metode Artificial Neural Network BackPropagation Studi Kasus: Mangga Gedong Gincu. 11(4), 591–595. <https://doi.org/10.26418/justin.v11i4.68234>
- Habel, D., & Levitar, S. (2023). Seminar Nasional Teknologi Informasi dan Komunikasi-2023 "Exploring the Intersection of Big Data, Cyber Security, Pengembangan Game Media Pembelajaran Peta Menggunakan Adobe Flash Di SD Negeri 03 Kanigoro. 285–294.
- Hikmah, N., Cahyo, B. T., Rianto, H., & Dewi, S. (2020). Rancang Bangun Pembuatan Program Kamus Plesetan Berbasis Pwa (Progressive Web Application). *JISAMAR (Journal of Information System, Applied, Management, Accounting and Research)*, 4(4), 1–8. <https://repository.bsi.ac.id/index.php/repo/viewitem/27231>
- Karno, A. S. B., Hastomo, W., & Wardhana, I. S. K. (2020). Validation of Monthly Rainfall Prediction Taken from the Output of Canonical Correlation Analysis Using Area Topographical Scenarios in Centra Java. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 3, 483–490. <http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1474>
- Kunang, Y. N., Nurmaini, S., Stiawan, D., & Suprpto, B. Y. (2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications*, 58. <https://doi.org/10.1016/j.jisa.2021.102804>
- Li, A. (2021). *Allan Li PROGRESSIVE WEB APPS : FACTORS Faculty of Information Technology and Communication Sciences Allan Li : Progressive Web Apps : Factors for Consideration in Development*. May.
- Nugraha, W., & Sasongko, A. (2022). Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search. *SISTEMASI: Jurnal Sistem Informasi*, 11(2), 391–401. <http://sistemasi.ftik.unisi.ac.id>
- Phie Joarno, R. J., Mohammad Fajar, & Arfan Yunus. (2022). Implementasi Progressive Web Apps

