

Sistem Estimasi Tinggi Badan dengan Menggunakan Model Regresi dalam Aplikasi Mobile

Sulthan Dzaky Alfaro
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
dzakysultan30@gmail.com

Rinaldi Munir
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@staff.itb.ac.id

Abstrak— Pada saat ini, mengukur tinggi badan merupakan hal yang sangat penting, terutama dalam bidang Kesehatan maupun olahraga. Tinggi badan juga dijadikan tanda pertumbuhan bagi anak-anak. Namun, dalam mengukur tinggi tentu memerlukan alat khusus. Selain itu, diperlukan juga waktu untuk mengukur tinggi tersebut. Untuk mengatasi masalah tersebut dibutuhkan sebuah sistem yang mampu mengestimasi tinggi hanya dengan sebuah gambar yang berisi objek yang ingin diukur dan sebuah smartphone. Pengembangan sistem estimasi tinggi badan manusia dilakukan dengan mengkombinasikan dan mengintegrasikan Mediapipe, model estimasi kedalaman, model regresi, dan aplikasi mobile. Untuk model regresi terdapat tiga model, yaitu Random Forest, SVR (Support Vector Regression), dan KNN (K-Nearest Neighbors) yang dibandingkan hasil pengujian estimasinya untuk digunakan pada modul regresi (estimasi tinggi badan). Berdasarkan analisis hasil pengujian terhadap data uji, terdapat 3 model regresi yang diuji dimana yang paling baik hasilnya adalah Random Forest dengan error MAE adalah 2.94 cm dan RMSE adalah 5.33 cm. Aplikasi mobile yang telah dikembangkan dan diintegrasikan dengan modul-modul yang dibutuhkan, yaitu modul MiDaS, YOLO, modul regresi, dan Mediapipe berhasil memberikan fleksibilitas, namun hasil yang diberikan kurang baik.

Kata Kunci— estimasi tinggi; regresi; MiDaS; YOLO; Mediapipe

I. PENDAHULUAN

Citra digital merupakan sebuah objek yang ditangkap oleh sensor perangkat citra lalu direpresentasikan menjadi dua dimensi. Citra digital ini memiliki informasi yang sangat banyak yang bisa kita manfaatkan untuk keperluan tertentu. Namun tentu banyak tantangan dalam memproses gambar, terutama karena keterbatasan manusia dalam hal kecepatan pemrosesan, akurasi, dan faktor lainnya. Salah satu pengaplikasian pemrosesan gambar adalah dalam mengukur tinggi badan. Pengukuran tinggi badan sangatlah penting baik dalam bidang kesehatan, olahraga, dan masih banyak lagi. Namun, untuk melakukan pengukuran badan tentu perlu bantuan orang lain untuk mengukur tinggi badan kita. Selain itu diperlukan juga alat untuk mengukur tinggi badan kita, baik dengan penggaris, meteran, atau alat ukur lain. Hal ini tidak efisien karena akan memerlukan waktu yang lama dengan kisaran 2 menit untuk mengukur tinggi manusia.

Dalam salah satu kasus, terdapat anak yang bergerak-gerak pada pengukuran, yang mana kaki anak harus lurus untuk diukur tingginya. Apabila pengukuran tidak dilakukan dengan benar, tidak menutup kemungkinan tenaga kesehatan dapat salah mendiagnosa seorang anak yang sehat, masuk dalam kategori yang mengalami stunting. Seperti pada penelitian di posyandu [17] memperlihatkan bahwa terdapat tahap pengukuran yang dilakukan oleh kader posyandu di Yogyakarta dengan benar hanya 27% pada salah satu tahap. Hasil observasi yang telah dilakukan menunjukkan bahwa kader pada tahap melakukan tinggi badan untuk anak kurang dari 24 bulan tanpa melakukan koreksi (ditambah 0.7). Kesalahan ini berpengaruh terhadap presisi, akurasi, dan validitas hasil pengukuran. Beberapa kesalahan biasanya berhubungan dengan keterampilan kader yang tidak cukup.

Terdapat beberapa penelitian sebelumnya yang telah melakukan hal yang serupa, yaitu mengestimasi tinggi badan dengan teknologi. Berikut beberapa penelitian yang serupa.

1. Dengan menggunakan deteksi tepi, memiliki kelemahan tidak akurat jika terdapat *noise* dan tidak adaptif dalam jarak tertentu, sehingga akurasi akan berkurang [6].
2. Dengan *stereo vision*, hasil yang diberikan akurat, namun harus memiliki 2 kamera untuk mengimplementasikan *stereo vision* [7].
3. Dengan *depth 3D conversion*, memberikan akurasi yang baik, namun memerlukan kamera khusus, yaitu *depth camera* [5].

Untuk melengkapi keterbatasan studi sebelumnya, diusulkan dengan menggunakan pendekatan model regresi, yang akan menghubungkan beberapa variabel untuk mengestimasi tinggi badan. Selain itu, digunakan juga Mediapipe dalam pengukuran tinggi badan serta deteksi kedalaman berbasis MiDaS. Mediapipe digunakan untuk mengetahui koordinat kepala dan kaki pada gambar secara akurat. Penggunaan Mediapipe agar penentuan koordinat kepala dan kaki diharapkan tidak terlalu terpengaruh dalam gambar yang memiliki *noise*. Selain itu, metode tersebut akan diintegrasikan ke dalam aplikasi *mobile*. Dengan mengkombinasikan metode tersebut diharapkan dapat menghasilkan solusi yang efisien dan lebih fleksibel dalam mengukur tinggi badan.

II. TINJAUAN PUSTAKA

A. Digital Image Processing

Sebuah gambar dapat didefinisikan sebagai 2 dimensi fungsi, $f(x,y)$ dimana x dan y adalah koordinat dan amplitudo dari f dengan pasangan x dan y adalah intensitas atau gray level. Ketika nilai x , y , dan intensitas adalah diskrit, bisa disebut bahwa gambar tersebut adalah gambar digital [2]. *Digital Image Processing* merupakan bidang yang memproses gambar digital dengan menggunakan komputer digital. Gambar digital terdiri dari finite angka dari element. Element tersebut biasa disebut pixel. Pixel ini sudah banyak dikenal sebagai elemen dari gambar digital.

Ada beberapa representasi dari sebuah gambar digital. Namun yang paling simpel dan sudah diketahui banyak orang adalah berupa nilai numerik dari $f(x,y)$ dalam sebuah matrix. Hal ini akan sangat berguna dalam membuat algoritma yang hanya memakai bagian gambar tertentu. Dan representasi ini sangat berguna dalam menganalisis sebuah gambar karena gambar direpresentasikan sebagai nilai numerik dalam sebuah matrix.

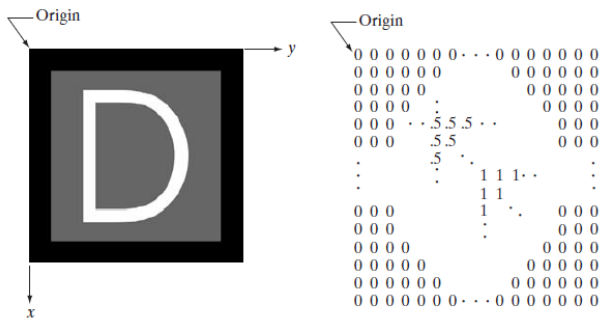


Fig. 1. Representasi gambar sebagai matrix

Ada beberapa operasi dalam memproses suatu citra digital. Terdapat operasi linear, operasi non-linear, operasi aritmatika, operasi spasial, operasi vektor, dan masih banyak lagi. Secara umum, operasi tersebut merupakan operasi spasial. Operasi spasial ini akan dilakukan langsung pada pixel pada sebuah gambar. Operasi ini akan dibagi menjadi 3 kategori, yaitu operasi 1 pixel (*single pixel operation*), operasi tetangga (*neighborhood operation*), dan *geometric spatial transformation*.

B. Mediapipe

Mediapipe merupakan salah satu pustaka yang dikembangkan oleh Google untuk membangun alur kerja pemrosesan media yang memanfaatkan *machine learning*. *Mediapipe* memungkinkan memproses berbagai data sensorik (video, audio, gambar, dan lain-lain). *Mediapipe* merupakan salah satu solusi *machine learning* untuk mendeteksi pose dalam sebuah gambar atau video. *Mediapipe* Pose akan mengembalikan 33 *landmarks* prediksi pose. *Mediapipe* akan memvisualisasikan graf pose *landmarks* dengan submodul pose *landmarks*. *Mediapipe* menyediakan konfigurasi *crossplatform* yang dapat digunakan di beberapa situasi, termasuk mode gambar statik, kompleksitas model, *minimum detection confidence*, dan *minimum tracking confidence*. Mode gambar

bisa diaktifkan untuk input gambar, atau dinonaktifkan untuk video stream.

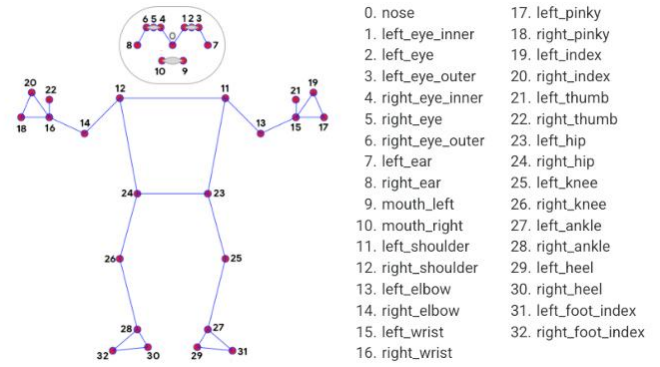


Fig. 2. Landmark yang didapat dari Mediapipe

C. Monocular Depth Estimation (MiDaS)

Monocular depth estimation mengacu pada proses regresi kedalaman padat hanya pada satu citra masukan. Terdapat banyak aplikasi dalam beberapa tugas, seperti *generative AI*, *3D reconstruction*, dan *autonomous driving*. Namun, sangat menantang untuk menyimpulkan informasi kedalaman pada piksel individual hanya dengan satu citra, karena *monocular depth estimation* adalah masalah yang kurang dibatasi. Sejak awal pengembangan dari model ini, telah ada beberapa model MiDaS yang menawarkan model-model baru dengan backbones yang lebih kuat dan varian ringan untuk aplikasi mobile.

Terdapat nilai kedalaman per pixel yang bisa didapatkan dari model MiDaS, diantaranya physical unit (cm, meter) atau relative unit. Dalam physical unit, model dapat memberikan nilai yang sebenarnya. Hal ini dapat memberikan beberapa aplikasi yang dapat digunakan untuk menentukan keputusan dalam fungsinya. Untuk relative unit, peta kedalaman (*depth map*) berisi nilai kedalaman per pixel yang hanya konsisten relatif satu sama lain. Hal ini berarti tidak ada skala yang pasti, nilai kedalaman hanya menunjukkan seberapa dekat atau jauh objek dibandingkan satu sama lain, dan tidak memberikan informasi nyata tentang jarak sebenarnya objek dari kamera.

Salah satu model yang dapat digunakan untuk *monocular depth estimation* yaitu MiDaS. Model ini sangat populer dalam mengestimasi kedalaman dalam sebuah gambar. Implementasi dari MiDaS dengan menggunakan *convolutional backbone*. Untuk versi terbarunya, MiDaS ini menggunakan model transformer atau disebut model DPT (*Dense Prediction Transformer*).

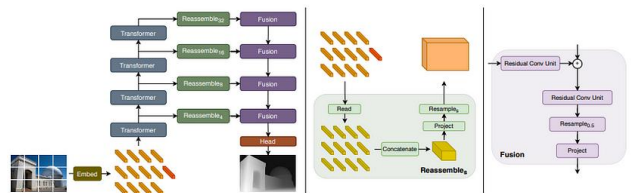


Fig. 3. Arsitektur DPT (Dense Prediction Transformer) pada MiDaS

D. Regresi

Regresi merupakan salah satu model yang menganalisis pengaruh antar variabel. Ada beberapa pemodelan regresi yang ada. Salah satunya adalah pemodelan dengan *gaussian regression analysis*. Pemodelan dengan cara ini dilakukan dengan menggunakan beberapa asumsi sebagai dasarnya. Seperti *linear equation models*, *normally distributed data*, *homogeneous residual variants*, dan *absence of multicollinearity symptoms*. Dalam kenyataannya, tidak semua asumsi dalam pemodelan regresi bisa bertemu. Oleh karena itu, pemodelan terus berkembang hingga pemodelan dengan prinsip *machine learning*. Pemodelan ini tidak berdasarkan pola linearitas data dan asumsi sebaran data.

Beberapa penggunaan metode regresi berbasis *machine learning* adalah pemodelan dengan menggunakan teknik *decision tree*, *random forest*, *support vector regression*, dan lain-lain. Beberapa penelitian telah membandingkan berbagai model regresi dengan teknik *machine learning*. Salah satunya terdapat penelitian untuk memprediksi kasus COVID-19. Hasilnya, *random forest* memiliki hasil yang lebih baik dari pada regresi linear dengan nilai error yang lebih kecil.

1. K-Nearest Neighbor

K-Nearest Neighbors merupakan algoritma klasifikasi non-parametrik, yang berarti tidak membuat asumsi pada sebuah *dataset*. Algoritma ini termasuk *Supervised Learning* dimana algoritma ini terkenal karena kesederhanaan dan keefektifannya. KNN sebagian besar digunakan untuk pengklasifikasian. Untuk data kontinu, algoritma ini menggunakan jarak Euclidian untuk menghitung tetangga terdekatnya [9].

Untuk input baru, diperlukan K tetangga terdekat untuk menentukan prediksi input baru. Nilai 'K' ini sangat penting dalam menentukan prediksi sebuah data baru. Penentuan nilai 'K' dilakukan dengan menjalankan algoritma beberapa kali dengan nilai yang berbeda untuk melihat nilai 'K' mana yang memberikan nilai yang paling efektif. KNN tidak melakukan generalisasi pada data latih. Oleh karena itu, seluruh data latih digunakan dalam tahap pengujian. Dalam regresi, KNN memprediksi nilai kontinu. Nilai ini adalah rata-rata dari nilai tetangganya. Ketika sebuah data baru tidak berlabel ditemukan, KNN akan melakukan 2 operasi, pertama, menganalisis K titik terdekat dengan titik data baru, yaitu K tetangga terdekat. Kedua, menggunakan kelas tetangga, KNN menentukan kedalam kelas mana data baru harus diklasifikasikan.

Untuk mengetahui tetangga terdekat, dapat menggunakan formula jarak Euclidian (d) dengan formula pada Persamaan 1 dengan x_i merupakan data yang ingin diprediksi dan y_i merupakan data latih.

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

2. Random Forest

Random forest merupakan algoritma *machine learning* yang digunakan untuk pemodelan regresi dan klasifikasi. Dalam konteks regresi, Random Forest akan membangun banyak *independent decision trees* dan mengkombinasikan hasil prediksi tiap tree untuk menghasilkan prediksi regresi akhir. Random Forest menghasilkan prediksi regresi dengan merata-ratakan prediksi yang dihasilkan dari seluruh *decision tree*. Dalam kasus regresi, prediksi akhir adalah rata-rata nilai dari prediksi seluruh *decision tree*.

Kelebihan dari Random Forest dalam pemodelan regresi adalah penanganan kumpulan data yang tidak terstruktur dan pola yang kompleks. Pemodelan yang dihasilkan stabil dan tidak sensitif terhadap perubahan kecil serta dapat mengatasi *overfitting*. Terdapat juga kelemahan pemodelan ini, yaitu ketahanan terhadap *overfitting* pada data yang sangat kompleks sangat terbatas. Selain itu, model ini juga sensitif terhadap kumpulan data yang tidak seimbang.

3. Support Vector Regression (SVR)

Model SVR adalah algoritma *machine learning* yang memprediksi nilai kontinu dari variabel masukan. SVR ini mengadaptasi prinsip *Support Vector Machine* ke dalam konteks regresi. SVR memiliki beberapa kelebihan, yaitu model ini dapat menangani data non-linear dengan menghubungkan variabel masukan dengan target menggunakan fungsi kernel. Dengan ini, SVR dapat menyesuaikan diri dengan pola yang kompleks dalam data. SVR juga tahan terhadap *overfitting* karena pendekatan *regularization* yang dapat membantu dalam mengendalikan *overfitting*.

Kelemahan SVR adalah dalam pengaturan parameter. Performa SVR sangat dipengaruhi oleh parameter yang digunakan, seperti C dan parameter kernel. Pengaturan parameter yang tidak tepat dapat menghasilkan model yang buruk. SVR juga tidak efisien untuk data yang besar. Hal ini dapat mengakibatkan waktu komputasi SVR menjadi lambat.

III. USULAN SOLUSI

Pada bagian ini akan ditampilkan desain dari sistem estimasi tinggi badan dalam aplikasi *mobile* dengan menggunakan model regresi. Sistem yang diusulkan akan mengukur tinggi badan dari gambar masukan yang berisi objek manusia yang ingin diukur. Estimasi tinggi dilakukan dengan menggunakan model regresi dengan menghubungkan beberapa variabel, antara lain estimasi kedalaman (MiDaS) dan *Mediapipe*.

A. Diagram Alir Solusi

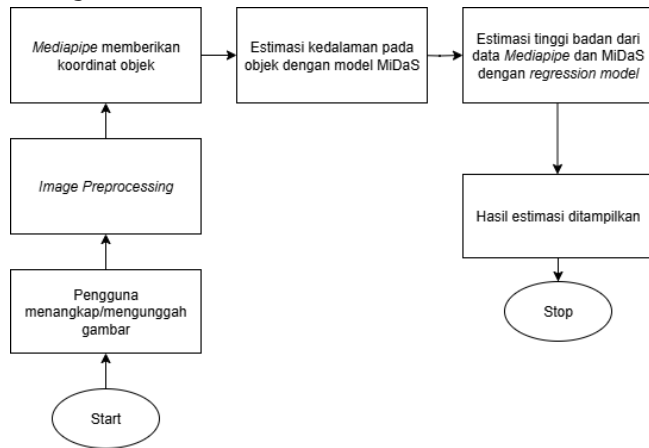


Fig. 4. Diagram alir solusi

Sistem estimasi terdiri dari beberapa tahapan. Pertama, pengguna akan memasukkan sebuah gambar pada aplikasi. Selanjutnya, aplikasi akan melakukan *image preprocessing* yang terdiri dari penyesuaian ukuran menjadi 256 x 256 pixel dan *denoising* untuk mengurangi derau pada gambar. Selanjutnya, gambar hasil *preprocessing* akan diolah oleh *Mediapipe* untuk didapatkan koordinat kepala, kaki, dan titik tengah badan objek. Setelah mendapatkan koordinat, gambar hasil *preprocessing* akan digunakan kembali untuk diolah oleh MiDaS (estimasi kedalaman). Dari MiDaS didapat *depth map* yang akan digunakan untuk mengetahui kedalaman objek. Kedalaman objek didapat dari mengekstrak pixel *depth map* dengan koordinat titik tengah badan objek. Setelah mendapat estimasi kedalaman dan koordinat, selanjutnya data tersebut akan diolah oleh model regresi untuk diestimasi tinggi badan. Setelah sistem mengestimasi tinggi dengan model regresi, selanjutnya akan ditampilkan ke layar ponsel beserta *bounding box* YOLO agar mudah mengetahui lokasi objek.

B. Dataset

Dataset citra yang digunakan untuk melatih model dan pengujian adalah gabungan dari *dataset* publik, yaitu People Snapshot, serta data tambahan yang diambil secara mandiri. Data tambahan yang diambil mandiri diperoleh dari citra-citra pasien yang akan berobat di UPTD Puskesmas Susunan 1. Selanjutnya, *dataset* akan dilakukan pemilihan dan pemisahan antara yang sesuai dengan kebutuhan dengan yang tidak secara manual. Hal ini dilakukan agar mendapat citra yang sesuai, yaitu citra manusia berdiri yang terlihat seluruh tubuh, agar mendapat data yang akurat. Selain itu, dipilih juga citra yang mengandung hanya 1 manusia, agar tidak terjadi kekeliruan pada saat ekstraksi data pada gambar. Adapun total *dataset* yang diperoleh untuk proses pengujian dan melatih model sebanyak 526 citra.

Dataset yang telah dikumpulkan akan dilakukan penyaringan data, dengan gambar yang kurang sesuai seperti buram atau objek badan manusia terpotong pada gambar akan disingkirkan. Didapat dari data 526 gambar, data akhir yang digunakan adalah 508 data. Dari 508 data ini, memiliki rata rata tinggi badan 168.74 cm dan standar deviasi 16.50 cm yang menunjukkan variasi ukuran tinggi badan cukup besar di antara

sampel. Nilai tinggi badan terendah tercatat 105 dan tertinggi adalah 192 cm. Untuk kuartil pertama didapat 159.26, median 173.58 dan kuartil ketiga didapat 179.93. Distribusi ini menjelaskan bahwa *dataset* yang dikumpulkan berisi tinggi badan di rentang sekitar 159 hingga 180 cm. Data yang dicatat hanyalah tinggi manusia.

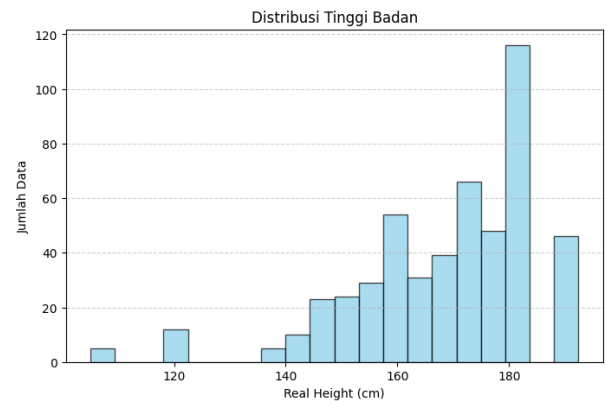


Fig. 5. Diagram persebaran data tinggi pada *dataset*

C. Arsitektur Sistem

Sistem solusi yang diusulkan terdiri atas tiga komponen utama, yaitu *Mediapipe*, modul estimasi kedalaman, modul regresi, dan aplikasi *mobile*. Secara garis besar, *Mediapipe* digunakan untuk mengetahui koordinat kepala dan kaki serta titik tengah badan manusia pada gambar. Modul estimasi kedalaman digunakan untuk mengetahui kedalaman suatu objek pada gambar. Modul regresi digunakan untuk estimasi tinggi badan dengan menggunakan informasi yang didapat dari modul deteksi objek dan modul estimasi kedalaman. Sedangkan aplikasi *mobile* bertujuan untuk mendapatkan gambar yang berisi objek manusia yang ingin dilakukan estimasi tinggi badan, dan akan menampilkan hasil keluaran estimasi tinggi badan, dan menyediakan pilihan fitur apakah pengguna ingin memilih foto pada penyimpanan atau menangkap langsung gambar pada kamera.

Sistem estimasi tinggi badan yang diusulkan akan menggunakan bahasa Kotlin dan terdiri dari beberapa komponen. Komponen tersebut antara lain: (i) pustaka *Mediapipe*, (ii) estimasi kedalaman MiDaS dengan model EfficientNet-Lite3, (iii) model regresi dengan model terbaik yaitu Random Forest, (iv) model YOLO untuk mengetahui objek pada gambar.

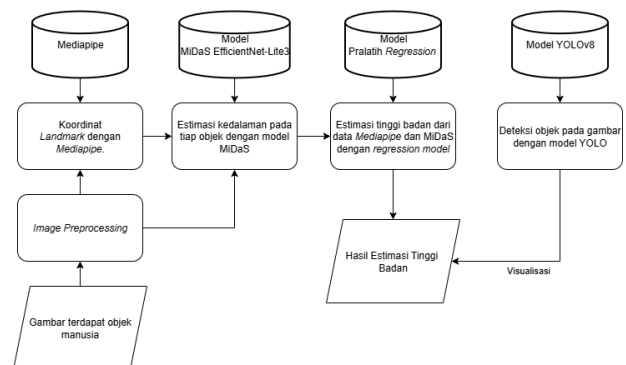


Fig. 6. Diagram arsitektur sistem

IV. PENGUJIAN DAN EVALUASI

A. Tujuan Evaluasi

Evaluasi dilakukan untuk mengukur seberapa bagus model mengestimasi tinggi badan serta kinerja aplikasi pada beberapa ponsel. Pengujian dibagi menjadi 3 bagian

1. Pengujian untuk mengukur model, dengan menggunakan metrik *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE).
2. Pengujian aplikasi *mobile* dengan menggunakan beberapa ponsel dengan mengukur waktu eksekusi aplikasi
3. Pengujian kuantitatif dengan beberapa kondisi, mulai dari kondisi ketinggian kamera hingga jarak kamera dengan objek.

B. Pengujian Model

Pengujian dilakukan pada model hasil *tuning hyperparameter*. Model-model yang akan dibandingkan untuk dipilih yang terbaik ada 3, yaitu SVR, KNN, dan Random Forest. *Tuning hyperparameter* dilakukan dengan menggunakan GridSearch dengan *k-fold cross-validation* dengan k adalah 3. Didapat hasil *tuning* masing masing model adalah sebagai berikut.

1. Random Forest
 - a. *n_estimators*: 50
 - b. *max_depth*: 10
2. SVR
 - a. *C*: 10
 - b. *kernel*: "scale"
 - c. *gamma*: "rbf"
3. KNN
 - a. *n_neighbors*: 5
 - b. *weights*: "distance"

Dari hasil *tuning*, sebelum pengujian dilakukan pelatihan pada model dengan menggunakan 80% dari *dataset* dengan parameter yang telah didapat. selanjutnya akan dilakukan pengujian dengan menggunakan 20% dari *dataset* yang telah disiapkan. Didapat hasil akurasi dari tiap model

TABLE I. AKURASI TIAP MODEL

Metrik	SVR	KNN	Random Forest
MAE	4.08 cm	3.25 cm	2.94 cm
RMSE	6.1 cm	5.95 cm	5.33 cm

Dari hasil pengujian dengan menggunakan metrik MAE dan RMSE, didapat bahwa akurasi terbaik dari ketiga model adalah Random Forest dengan MAE 2.94 cm dan RMSE 5.33 cm. Hal ini juga termasuk baik karena model tersebut lebih

baik dari model *baseline* dengan menggunakan Dummy Regressor dengan MAE sebesar 11.4 cm dan RMSE sebesar 13.56 cm.

C. Pengujian Aplikasi Mobile

Pengujian aplikasi mobile bertujuan untuk memastikan keberhasilan integrasi antara seluruh komponen yang dibutuhkan sistem, diantaranya modul deteksi objek berbasis YOLO, modul estimasi kedalaman berbasis MiDaS, *Mediapipe*, dan modul regresi untuk estimasi tinggi. Selain itu, pengujian juga memastikan aplikasi memiliki keseluruhan fitur yang berhasil berjalan dengan lancar tanpa suatu bug atau error. Fitur tersebut yaitu *Capture* (menangkap gambar langsung untuk proses estimasi) dan *Upload* (mengunggah sebuah gambar untuk proses estimasi). Selain untuk memastikan sistem hasil integrasi berhasil berjalan, pengujian ini dilakukan untuk mengetahui performa aplikasi pada beberapa smartphone. Performa dihitung dari seberapa lama waktu eksekusi program dijalankan pada smartphone untuk menyelesaikan sistem estimasi tinggi

Pengujian performa, akan dilakukan di dua *smartphone* yang berbeda. Untuk spesifikasi tiap *smartphone* sebagai berikut.

1. Samsung A12 (Low end)

CPU: Mediatek MT6765 Helio P35 (12 nm)

Android: 11

Kamera: 48 MP, f/2.0,

2. Samsung A35 (Mid-range)

CPU: Exynos 1380 (5 nm)

Android: 15

Kamera: 50 MP, f/1.8, (wide)

TABLE II. PENGUJIAN PERFORMA APLIKASI

Smartphone	Run 1	Run 2	Run 3
Samsung A12	3.373 s	3.392 s	3.204 s
Samsung A35	1.316 s	1.314 s	0.696 s

D. Pengujian Kuantitatif

Pada pengujian ini, citra yang digunakan terdapat manusia dari berbagai jarak dengan kamera. Manusia yang dijadikan uji coba akan difoto dari berbagai jarak, mulai dari jarak dekat hingga jauh dan dari berbagai ketinggian kamera. Pengujian ini untuk menilai kemampuan aplikasi dalam mengestimasi tinggi dari berbagai jarak pengambilan gambar.

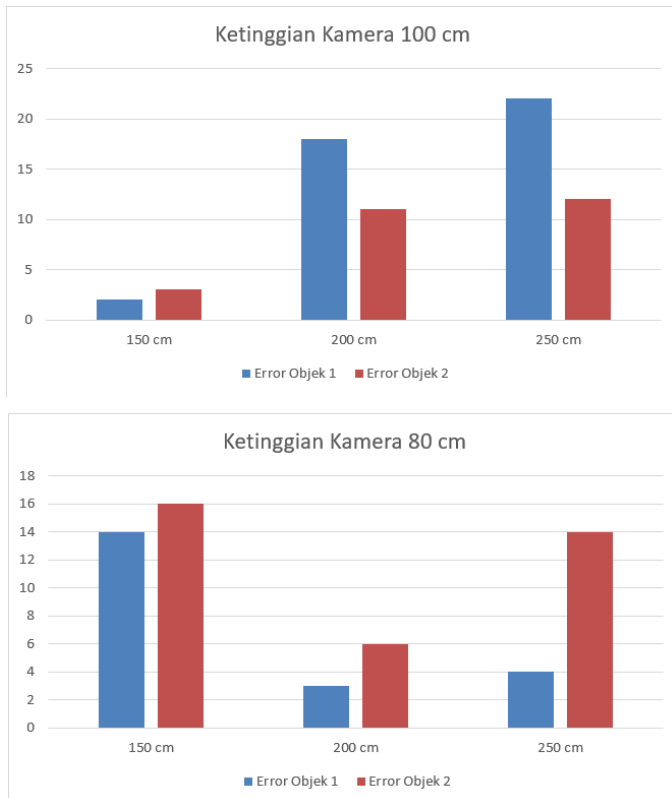


Fig. 7. Hasil pengujian berbagai kondisi ketinggian kamera dan jarak kamera dengan objek

Dari hasil pengujian seperti pada Fig. 7., pengambilan gambar pada jarak tertentu memiliki pengaruh terhadap akurasi sistem. Selain itu, ketinggian pengambilan gambar juga mempengaruhi. Dari pengujian ini, yang paling optimal dalam pengukuran di sistem ini dai pengujian ini adalah pada ketinggian kamera 100 cm dengan jarak antara objek dan kamera 150 cm, dan pada ketinggian kamera 80 cm dengan jarak antara objek dan kamera 200 cm.

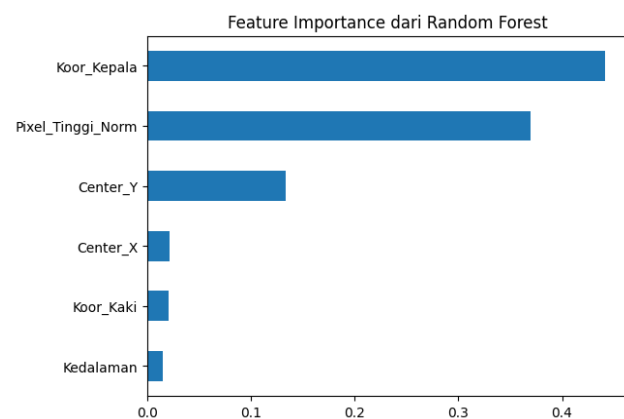


Fig. 8. Fitur penting dalam model Random Forest

Dalam model Random Forest, terdapat fitur penting yang dapat mempengaruhi model dalam mengestimasi. Dalam Fig. 8., “Kedalaman” memiliki pengaruh yang sangat kecil dibandingkan variabel lain. “Kedalaman” merupakan jarak antara kamera dengan objek. Hal ini tidak sesuai harapan dimana “Kedalaman” diharapkan memiliki peranan penting,

agar akurasi sistem dapat terjaga walaupun dari jarak objek dengan kamera tidak ditentukan. Hal ini bisa terjadi karena kesalahan pemilihan model estimasi kedalaman, dimana MiDaS mengembalikan nilai kedalaman yang relatif. Diharapkan, untuk penelitian selanjutnya, dapat menggunakan model yang memberikan nilai absolut seperti ZoeDepth atau Depth-Pro agar jauh lebih akurat.

V. KESIMPULAN DAN PENGEMBANGAN SELANJUTNYA

Dari *paper* ini, penulis mengajukan sebuah metode untuk mengestimasi tinggi badan. Metode tersebut merupakan metode regresi dengan menggunakan beberapa variabel yaitu *Mediapipe* dan estimasi kedalaman (MiDaS). Model-model akan dilatih dan diuji lalu dibandingkan untuk mendapatkan model terbaik.

Setelah melakukan perbandingan dari ketiga model, didapat bahwa Random Forest memiliki akurasi yang paling baik dengan MSE sebesar 2.94 cm dan RMSE sebesar 5.33 cm. Dan model dapat digunakan setelah diintegrasikan dengan aplikasi *mobile*. Namun, hasil akurasi setelah diintegrasikan menjadi lebih buruk. Hal ini dikarenakan fitur kedalaman dalam model tidak memiliki peranan penting karena kesalahan dalam pemilihan model estimasi kedalaman.

Untuk penelitian selanjutnya, dapat menggunakan model estimasi kedalaman dengan mengembalikan nilai absolut agar akurasi lebih baik dalam segala kondisi. Karena model ini memiliki komputasi yang sangat besar, dapat menggunakan sistem *backend* agar tidak membebani ponsel. Selain itu, untuk menambah keakuratan dalam mengestimasi tinggi, dalam menangkap gambar dapat menggunakan *accelerometer* agar posisi pengambilan gambar tetap terjaga.

REFERENCES

- [1] Russell, S., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.
- [2] Rafael C. Gonzales, & Richard E.Woods (2008). Digital Image Processing (3th ed.). Pearson..
- [3] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience, 2018, 1–13. <https://doi.org/10.1155/2018/7068349>
- [4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. ArXiv. <https://doi.org/10.48550/arXiv.1506.02640>
- [5] Dong-seok Lee, Jong-soo Kim, Seok Chan Jeong, Soon-kak Kwon (2020). Human Height Estimation by Color Deep Learning and Depth 3D Conversion. Appl. Sci. 2020. <https://doi.org/10.3390/app10165531>
- [6] Ida U., Eko S., Eni D. W. (2017). Height Measurement System Based on Edge Detection Technique and Analysis of Digital Image Processing. JAICT.
- [7] Henry O., Vulgarin, J., Boris X. (2023). Deep Learning-based Human Height Estimation from a Stereo Vision System. IEEE. <https://doi.org/10.1109/ICPRS58416.2023.10179079>
- [8] Nateski, V. (2017). An Overview of the Supervised Machine Learning Methods. ResearchGate.
- [9] Taunk, K., Sanjukta De, Verma, S., Swetapadma, A. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. IEEE.
- [10] Dhikhi, T., Allagada Naga Suhas, Gosula Ramakanth Reddy, Kanadam Chandu Vardhan. Measuring Size of Object Using Computer Vision. IJITEE.

- [11] Plevris, V., Solorzano, G., Bakas, N., Ben Seghier, M. (2022). Investigation of performance metrics in regression analysis and machine learning-based prediction models. Scipedia. 2022. <http://dx.doi.org/10.23967/eccomas.2022.155>
- [12] Szeliski R. (2010). Computer Vision: Algorithm and Application. Springer.
- [13] Sihombing P. R., Budiantono S., Arsani A. M., Aritonang T. M, Kurniawan M.A. (2023). Comparasion of Regression Analysis with Machine Learning Supervised Predictive Model Techniques. ResearchGate. 2023. <http://dx.doi.org/10.11594/jesi.03.02.03>
- [14] ANTARA. 2022. BKKBN ingatkan posyandu tak sepelekan ukuran, berat, tinggi badan anak. Diakses pada 14 Juli 2025 dari <https://www.antaranews.com/berita/3108437/bkkbn-ingatkan-posyandu-tak-sepelekan-ukuran-berat-tinggi-badan-anak>
- [15] Medium. 2023. Monocular Depth Estimation using ZoeDepth : Our Experience. Diakses pada 18 Juli 2025 dari <https://medium.com/@bhaskarbose1998/monocular-depth-estimation-using-zoedepth-our-experience-42fa5974cb59>
- [16] Kim, J. W., Choi, J. Y., Ha, E. J., Choi, J. H. (2023). Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. MDPI. <https://doi.org/10.3390/app13042700>
- [17] Syagata, A. S., Rohmah, F., N., Khairani, K., Arifah, S. (2021). Evaluasi pelaksanaan pengukuran tinggi badan oleh kader Posyandu di Wilayah Yogyakarta. eJournal.
- [18] Gzar, D. A., Mahmood, A. M., Abbas, M. K. (2022). A Comparative Study of Regression Machine Learning Algorithms: Tradeoff Between Accuracy and Computational Complexity. IIETA. <https://doi.org/10.18280/mmep.090508>