

PENDETEKSIAN PLAT NOMOR KENDARAAN MENGUNAKAN ALGORITMA YOU ONLY LOOK ONCE V3 DAN TESSERACT

Muhamad Rizky Fauzan¹, Ari Purno Wahyu W²

Universitas Widyatama, Universitas Widyatama
Universitas Widyatama

Jl. Cikutra No.204A, Sukapada, Cibeunying Kidul, Kota Bandung
mrfauzannn.99@gmail.com¹, ari.purno@widyatama.ac.id²

Abstrak

Perkembangan teknologi saat ini sangat berkembang pesat. Teknologi yang saat ini sedang dilakukan pengembangan secara besar-besaran yaitu *Artificial Intelligence*. *Artificial Intelligence* atau AI memiliki berbagai macam fungsi dan tujuan tergantung dari sistem yang akan dibuat. Salah satunya yaitu pendeteksi objek dan teks dari gambar atau video. Contoh dari pemanfaatan teknologi ini yaitu pada pendeteksi objek dan teks pada plat nomor kendaraan. Pada penelitian ini dilakukan perancangan sistem dengan menggunakan algoritma *You Only Look Once V3* sebagai algoritma pendeteksi objek dan *Tesseract Optical Character Recognition* sebagai pendeteksi teks dalam gambar. Perancangan ini akan dibantu dengan *library OpenCV* pada bahasa pemrograman python dan menggunakan *dataset* gambar yang sudah tersedia. Penelitian ini bertujuan untuk mengetahui tingkat keakurasian algoritma *You Only Look Once V3* yang dikombinasikan dengan *Tesseract Optical Character Recognition*.

Kata kunci :

Artificial Intelligence, Object Detection, You Only Look Once V3, Tesseract Optical Character Recognition, OpenCV.

Abstract

The development of technology is currently growing very rapidly. One technology that is currently being developed on a large scale is Artificial Intelligence. Artificial Intelligence or AI has various functions and purposes depending on the system created. One of them is to detect objects and text from images or videos. An example of the

application of this technology is the detection of objects and text on vehicle number plates. In this study, a system design was carried out using the You Only Look Once V3 algorithm as an object detection algorithm and Tesseract Optical Character Recognition as a text in image detector. This design is assisted by the OpenCV library in the python programming language and uses an image dataset that is already available. This study aims to determine the level of accuracy of the You Only Look Once V3 algorithm combined with Tesseract Optical Character Recognition.

Keywords :

Artificial Intelligence, Object Detection, You Only Look Once V3, Tesseract Optical Character Recognition, OpenCV.

I. PENDAHULUAN

Pesatnya berkembang teknologi saat ini membuat peningkatan teknologi dan penemuan teknologi baru dikembangkan dan ditemukan dengan cepat. Hal ini dilakukan untuk membantu mempermudah pekerjaan manusia dengan biaya yang cukup efisien. Salah satu teknologi yang sedang dilakukan pengembangan secara besar-besaran yaitu *Artificial Intelligence*.

Artificial Intelligence atau biasa dikenal AI merupakan teknologi yang dikembangkan untuk bisa belajar, berpikir dan bekerja seperti manusia berdasarkan suatu data (Samek et al., 2017). Dengan kata lain, AI dapat belajar dari data-data yang ada yang selanjutnya akan dilakukan train data sebagai tahap pembelajaran. *Object detection* atau

pendeteksian objek merupakan salah satu contoh sistem yang nantinya bisa mendeteksi objek yang ada.

Object detection merupakan suatu teknik agar komputer bisa melihat dan mengidentifikasi objek yang ada baik dalam gambar ataupun video atau dapat diartikan juga sebagai perancangan suatu sistem agar sistem dapat melihat dan mengidentifikasi objek seperti manusia (Zhao et al., 2019). Dalam pembuatan ini diperlukan data pendukung yang nantinya data-data tersebut akan dilatih agar sistem dapat mengenali suatu objek.

Pemanfaatan dari *object detection* dapat diimplementasikan diberbagai aspek. Saat ini banyak perusahaan yang menggunakan sistem ini untuk nantinya dikombinasikan dengan disiplin ilmu lain. Hal yang paling sering digunakan dalam implementasi *object detection* yaitu pada sistem *self-driving car* atau mobil tanpa awak ataupun dapat digunakan untuk pendeteksian plat nomor kendaraan secara real time.

Selain itu, beberapa penelitian melakukan pengkombinasian antara pendeteksian objek dengan pendeteksian teks pada gambar. Hal ini diimplementasikan pada pendeteksian objek plat nomor kendaraan sekaligus dapat melakukan pendeteksian teks plat nomor yang ada pada kendaraan tersebut. Maka dari itu penulis tertarik untuk melakukan penelitian terkait *object detection* dan *optical character recognition* dalam mendeteksi plat nomor kendaraan.

II. LANDASAN TEORI

Membahas berbagai macam referensi yang sudah difokuskan oleh penulis dan akan digunakan sebagai acuan dalam proses penelitian dengan maksud untuk memberikan informasi kepada pembaca.

1. Pendeteksian, dalam Kamus Besar Bahasa Indonesia pendeteksian adalah suatu proses, cara, perbuatan untuk mendeteksi atau pelacakan (Kamus Besar Bahasa Indonesia, n.d.).
2. *Image Processing*, merupakan salah satu bentuk teknik pengolahan sinyal oleh komputer dengan masukan berupa citra (gambar) yang akan ditransformasikan sebagai keluran yang memiliki kualitas yang lebih baik (Waskito et al., 2019).

3. Python, merupakan salah satu bahasa pemrograman berorientasi objek tingkat tinggi, diciptakan oleh Guido Van Rossum. Python didesain agar mudah dimengerti, mudah digunakan dan disebut sebagai bahasa pemrograman yang *user-friendly* dan mudah digunakan oleh pemula (Srinath, 2017).
4. Pycharm, salah satu lingkungan pengembangan terintegrasi (IDE) untuk bahasa pemrograman python yang dikembangkan oleh JetBrains. Kelebihan dari IDE ini adalah dapat mengakses *command line*, terhubung ke *database*, membuat lingkungan *virtual* dan juga dapat melakukan *control* terhadap sistem dalam satu tempat.
5. OpenCV, merupakan salah satu *library open source* yang biasanya dimanfaatkan untuk mengolah citra secara *real-time* yang dibuat dan dikembangkan oleh Intel (Kumari et al., 2013). *Library* ini memiliki lebih dari 2500 algoritma didalamnya dan sudah termasuk algoritma untuk *computer vision* dan *machine learning*. OpenCV tidak berbayar dan memiliki lisensi *open source* dari *BSD License* (OpenCV team, n.d.).
6. Tesseract OCR, merupakan salah satu *open source engine* untuk mengenali karakter yang ada pada gambar (Google Open Source, n.d.). Tesseract mulai populer pada tahun 2016 dan memiliki lebih dari 100 bahasa. Data dari karakter juga dapat dilatih untuk menghasilkan pengenalan karakter yang lebih akurat atau untuk mengenali karakter khusus.
7. YOLO v3, merupakan pengembangan dari versi YOLO sebelumnya. Jika dibandingkan dengan versi sebelumnya, YOLO v3 memiliki jairngan ekstraktor yang lebih kuat dan memiliki fitur deteksi *multi-scale*. Selain itu juga dapat mendeteksi target yang lebih banyak dari target yang kecil hingga besar (Redmon, n.d.).
8. *Black-box testing*, merupakan salah satu teknik pengujian pada perangkat lunak yang berfokus pada bagian kebutuhan fungsional dari perangkat lunak tersebut (Bansal, 2014). Dalam pengujian *black-box* ini, penguji hanya memahami arsitektur dari sistem yang dibangun. Teknik ini dilakukan untuk memastikan bahwa semua masukan yang dibutuhkan oleh sistem diterima dan dapat memberikan keluaran yang baik (Jan et al., 2016).

9. Teknik fuzzing, merupakan teknik pengujian *black-box* yang dikembangkan oleh Barton Miller di University of Wisconsin pada tahun 1989. Teknik ini akan memberikan masukan yang acak ke aplikasi. Hal ini dilakukan untuk menemukan kesalahan implementasi (Bansal, 2014).

III. METODE PENELITIAN

III.1 Metodologi Penelitian

Metodologi yang digunakan dalam perancangan sistem yang berorientasi aliran data dengan model *waterfall*. *Waterfall* atau biasa disebut sebagai *Classic Life Cycle* merupakan metode dengan melakukan pendekatan sistematis pada pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pelanggan dan perkembangan melalui tahap perencanaan, pemodelan, konstruksi, penerapan dan terus berlanjut hingga pemeliharaan dari perangkat lunak tersebut (Buchori et al., 2017).

III.2 Analisis Permasalahan

Pendeteksian objek dilakukan untuk mendeteksi objek secara otomatis oleh sistem yang kemudian sistem dapat mengenali objek yang terdeteksi sesuai dengan *class* yang sudah dibuat sebelumnya. Salah satunya mengenai pendeteksian plat nomor kendaraan. Pendeteksian ini dilakukan untuk mempermudah pengecekan plat nomor dengan menggunakan sistem dengan menggunakan data yang sudah dilatih. Penelitian ini dilakukan untuk memodifikasi penelitian sebelumnya mengenai pendeteksian teks dan plat nomor kendaraan (Agbemenu et al., 2018) . Pendeteksian pada penelitian ini akan menggunakan *dataset* plat nomor kendaraan Indonesia agar dapat mendeteksi teks dan plat nomor Indonesia..

III.3 Perancangan Sistem

Pada proses perancangan ini, penulis membaginya terhadap perancangan *hardware* dan perancangan *software* seperti pada berikut ini :

1. *Hardware*, dalam penelitian ini tidak ada perancangan khusus untuk *hardware* yang digunakan. Alat yang akan digunakan yaitu *webcam* sebagai alat untuk merekam objek dan laptop untuk menjalankan sistem pendeteksian plat nomor dan teks.

2. *Software*, pada bagian ini penulis melakukan beberapa perancangan pengkodean sistem sebelum nantinya diimplementasikan. Beberapa perancangan *software* yang dilakukan yaitu :

- *Data training*, hal yang dipersiapkan yaitu pengumpulan gambar, *labelling* gambar, *pretrained weights* YOLOv3 dan *pretrained weights convolutional layers*, *custom configuration file* YOLOv3, *file* obj.data, obj.names dan train.txt.
- Pendeteksian objek, pada perancangan sistem ini penulis memanfaatkan beberapa *library* yang sudah tersedia dalam lingkup python seperti OpenCV dan Numpy. OpenCV berfungsi untuk mengolah video secara *real-time* dan meng-ekstrak informasi yang ada di dalam video tersebut dan numpy berfungsi untuk mengolah data yang bersifat *array*.
- Pendeteksian teks, dirancang dengan menggunakan Tesseract OCR. Dalam pengkodeannya, Tesseract OCR akan dipanggil dengan menggunakan *pytesseract* yang satu file program dengan pendeteksian objek. Pada penelitian ini, peneliti tidak melakukan *custom* karakter pada pendeteksian teks ini.

IV. IMPLEMENTASI

Implementasi mengenai sistem yang sudah dibuat oleh penulis akan dijelaskan pada bagian ini. Implementasi ini meliputi data training dan pengujian *black box testing* pada sistem. Berikut penjelasannya :

IV.1 Data Training

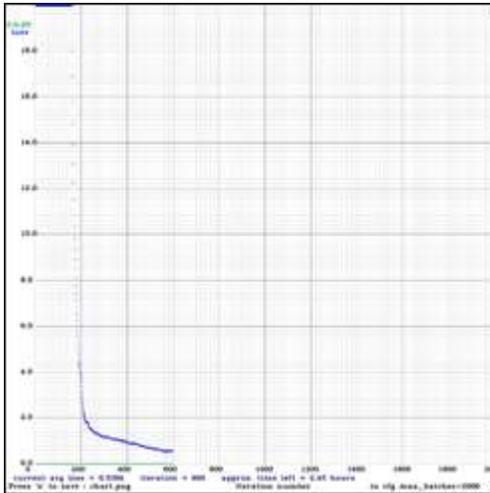
1. Pengumpulan gambar, gambar yang digunakan pada penelitian ini menggunakan gambar pada dataset yang sudah tersedia pada *website* Kaggle (Gunawan, n.d.). Gambar yang dimaksud disini yaitu berupa gambar kendaraan lengkap dengan plat nomornya. Gambar yang akan dijadikan acuan ini berjumlah 235 gambar.
2. *Labelling* gambar, proses ini berfungsi untuk melakukan penandaan pada gambar, yang akan digunakan dalam pembelajaran mesin untuk melatih kumpulan data yang akan membantu sistem memahami objek yang akan dideteksi. Hasil akhir dari proses ini berupa *file* txt yang berisi angka *annotation image*. *File* txt ini lah

yang akan digunakan pada proses pelatihan data nanti.



Gambar 4.1 Proses Labelling Gambar

- Proses *data train*, dilakukan secara *cloud* dengan memanfaatkan penggunaan *cloud notebook* yaitu Google Colaboratory. Selain itu, proses ini juga memanfaatkan fitur *cloning* dan *building data darknet* yang sudah tersedia di *online repository* (Redmon, n.d.). Berikut hasil perhitungan dari proses pelatihan data.



Gambar 4.2 Chart Perhitungan Pelatihan Data

IV.2 Pengujian Black Box Testing

Black box testing dilakukan untuk menguji fungsi yang ada pada sistem apakah dapat berjalan atau tidak sesuai dengan requirement yang ada dan disajikan dalam bentuk tabel (Kholipa et al., 2021).

Pengujian sistem ini akan dimulai dengan membuat tabel perencanaan pengujian yang berisi kelas uji dan butir uji. Tabel 1 merupakan tabel perencanaan dari sistem ini.

Tabel 1. Rencana Pengujian

No	Kelas Uji	Butir Uji
1.	Perekaman video secara <i>real-time</i> (KF-01).	Pengujian perekaman video secara <i>real-time</i> menggunakan <i>webcam</i> (U1-01), Pengujian meneruskan video <i>real-time</i> ke dalam sistem (U1-02).
2.	Deteksi plat nomor kendaraan Indonesia (KF-02).	Mendeteksi plat nomor kendaraan Indonesia dengan menampilkan <i>bounding box</i> dan presentase tingkat akurasi (U2-01), Pengujian menampilkan <i>Region of Interest</i> (U2-02).
3.	Deteksi teks pada plat nomor kendaraan Indonesia (KF-03).	Pengujian untuk mendeteksi teks plat nomor kendaraan Indonesia (U3-01), Pengujian untuk menampilkan teks yang terdeteksi pada layar (U3-02).

Selanjutnya akan dilakukan pengujian pada setiap butir uji yang sudah direncanakan pada tabel 1. Tabel pengujian ini terdiri dari butir uji, deskripsi pengujian, keluaran yang diharapkan, dan hasil uji.

Tabel 2. Hasil Pengujian

No	Butir Uji	Deskripsi Pengujian	Keluaran yang Diharapkan	Hasil Uji
1.	U1-01	Pengujian perekaman video secara <i>real-time</i> menggunakan <i>webcam</i> .	Muncul <i>pop up windows</i> dengan menampilkan rekaman video <i>real-time</i> .	Diterima
2.	U1-02	Pengujian meneruskan video <i>real-time</i> ke dalam sistem. presentase tingkat akurasi, Pengujian menampilkan <i>Region of Interest</i>	Rekaman video <i>real-time</i> diteruskan kedalam sistem untuk diproses	Diterima
3.	U2-01	Mendeteksi plat nomor kendaraan Indonesia dengan	Muncul <i>pop up windows</i> dengan menampilkan rekaman video <i>real-time</i> .	Diterima

		menampilkan bounding box dan presentase tingkat akurasi.		
4.	U2-02	Pengujian menampilkan <i>Region of Interest</i> .	Muncul <i>pop up windows</i> dengan menampilkan video yang ada pada <i>bounding box</i> .	Diterima
5.	U3-01	Pengujian untuk mendeteksi teks plat nomor kendaraan Indonesia	Muncul <i>pop up windows</i> dengan menampilkan rekaman video <i>real-time</i> .	Diterima, dengan catatan memerlukan waktu.
6.	U3-02	Pengujian untuk menampilkan teks yang terdeteksi pada layar.	Teks yang terdeteksi akan tertulis pada layar.	Diterima, dengan catatan memerlukan waktu.

Hasil dari pengujian sistem diatas tidak ada yang gagal tetapi dalam pendeteksian teks diperlukan waktu yang lebih karena pendeteksian dilakukan secara *real-time*.

V. KESIMPULAN DAN SARAN

V.1 Kesimpulan

Berdasarkan hasil yang didapat dalam menyusun tugas akhir ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Hasil dari pendeteksian objek dengan menggunakan YOLO v3 dapat ditampilkan pada layar laptop secara langsung. Kemudian pendeteksian teks menggunakan *tesseract* akan dilakukan dengan merekam objek melalui *webcam* dan menampilkan hasilnya secara *real-time*.
2. Proses pelatihan data dilakukan secara *cloud* dengan bantuan GPU agar tidak memakan waktu yang lama. Hasil dari pelatihan data ini digunakan sebagai *file* acuan sistem untuk mendeteksi plat nomor kendaraan.
3. Proses pengkombinasian antara pendeteksian objek dengan pendeteksian teks pada penelitian ini dapat mendeteksi objek dan teks pada plat nomor kendaraan. Hanya saja saat melakukan

pendeteksian teks memerlukan waktu yang cukup lama.

4. Cara kerja dari algoritma *You Only Look Once V3* sudah berjalan cukup baik dilihat dari hasil pelatihan datanya dan hasil dari pendeteksiannya. Hasil dari pelatihan datanya menampilkan *average loss* dibawah 2.0 yang mana bisa dikatakan memiliki akurasi yang baik dengan penggunaan data yang terbatas. Sedangkan untuk *Optical Character Recognition* pada penelitian ini memiliki tingkat akurasi yang belum begitu baik.

V.2 Saran

Adapun dibawah ini merupakan saran penulis yang dapat digunakan untuk pengembangan sistem kedepannya agar lebih baik lagi, sebagai berikut :

1. Disarankan untuk menggunakan gambar untuk pelatihan data diatas 500 atau 1000 gambar dengan kondisi yang berbeda agar mendapatkan tingkat akurasi yang jauh lebih baik lagi.
2. Disarankan untuk melakukan proses kustomisasi pendeteksian teks pada *tesseract* agar tingkat akurasi pendeteksian lebih akurat.
3. Disarankan untuk pengembangan selanjutnya menggunakan algoritma versi terbaru.
4. Disarankan untuk menggunakan gambar pada deteksi teks plat nomor kendaraan agar waktu pendeteksian menjadi lebih singkat dan akurat.
5. Disarankan untuk melakukan pengembangan terhadap *file* ekstraksi hasil deteksi plat nomor kendaraan beserta teksnya.

REFERENSI

- Agbemenu, A. S., Yankey, J., & Addo, E. O. (2018). An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine. *International Journal of Computer Applications*, 180(43), 1–5. <https://doi.org/10.5120/ijca2018917150>
- Bansal, A. (2014). A Comparative Study of Software Testing Techniques. *International Journal of Computer Science and Mobile Computing*, 3(6), 579–584.
- Buchori, A., Setyosari, P., Dasna, I. W., & Ulfa, S.

- (2017). Mobile augmented reality media design with waterfall model for learning geometry in college. *Int. J. Appl. Eng. Res.*, 12(13), 3773–3780.
- Google Open Source. (n.d.). *Tesseract OCR*. Retrieved May 10, 2021, from <https://opensource.google/projects/tesseract>
- Gunawan, K. (n.d.). *Labelled Indonesian Car and Plate Number Kaggle*. Retrieved May 10, 2021, from <https://www.kaggle.com/gunawan26/labelled-indonesian-car-and-plate-number>
- Jan, S. R., Shah, S. T. U., Johar, Z. U., Shah, Y., & Khan, F. (2016). An Innovative Approach to Investigate Various Software Testing Techniques and Strategies. *International Journal of Scientific Research in Science, Engineering and Technology*, 2(2), 682–689.
- Kamus Besar Bahasa Indonesia. (n.d.). *Arti Kata Deteksi*. Retrieved May 10, 2021, from <https://kbbi.web.id/deteksi>
- Kumari, S., Gupta, L., Gupta, P., & Abdul, A. P. J. (2013). Automatic License Plate Recognition Using OpenCV and Neural Network. *International Journal of Computer Science Trends and Technology (IJCT)*, 5(3), 114–118. www.ijctjournal.org
- OpenCV team. (n.d.). *Open CV*. Retrieved March 23, 2021, from <https://opencv.org/about/>
- Redmon, J. (n.d.). *YOLO*. Retrieved May 10, 2021, from <https://pjreddie.com/darknet/yolo/>
- Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ArXiv Preprint ArXiv:1708.08296*.
- Srinath, K. R. (2017). Python – The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*, 354–357. https://d1wqtxts1xzle7.cloudfront.net/55458585/IRJET-V4I1266.pdf?1515226715=&response-content-disposition=inline%3B+filename%3DPython_The_Fastest_Growing_Programming_L.pdf&Expires=1593202307&Signature=HBD7oa85wDxqRzTwx01uVRBIMacGX5mkGk1b~SVVTTkE
- NJ6cf5diKz
- Waskito, T. B., Setianingsih, C., Elektro, F. T., Telkom, U., & Network, C. N. (2019). *Kendali Robot Beroda Dengan Gerak Isyarat Tangan Berbasis Image Processing Wheeled Robot Control With Hand Gesture*. 6(3), 10052–10059.
- Zhao, Z.-Q., Zheng, P., Xu, S., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.