ANALISIS PERFORMASI PADA PENGAMBILAN URL BERBASIS *WEB CRAWLING* DENGAN MENGGUNAKAN TEKNOLOGI PENGENALAN WAJAH YOLOv3

PERFORMANCE ANALYSIS IN GENERATING INFORMATION OF URL BASED ON WEB CRAWLING USING YOLOv3 FACE RECOGNITION TECHNOLOGY

Lulud Annisa Ainun Mahmuddah[1], Suryo Adhi Wibowo[2], Gelar Budiman[3]
[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung
[1]luludaam@student.telkomuniversity.ac.id, **[2]suryoadhiwibowo@telkomuniversity.co.id**, **[3]gelarbudiman@telkomuniversity.ac.id**

Abstrak

Artifcial Intelligence (AI) adalah sistem yang dikembangkan untuk mempelajari dan menirukan kecerdasan manusia. Beberapa teknologi yang dihasilkan dari pengembangan AI diantaranya adalah teknologi *web crawling* dan *face recognition.* Teknologi *web crawling* digunakan untuk mendapatkan informasi pada halaman *web,* sedangkan teknologi *face recognition* digunakan untuk mengidentifikasi wajah manusia. Penelitian ini mengimplementasikan sistem *web crawling* dengan menggunakan teknologi *face recognition.* Penelitian ini merancang sistem yang akan memindai wajah manusia secara *real-time* dengan menggunakan metode deteksi objek *You Only Look Once* (YOLO), bahasa pemrograman *Python*, *library OpenCV*, dan *library Requests* untuk *web crawling*. Kemudian hasil dari *face recognition* akan digunakan sebagai *keyword* untuk proses *web crawling.* Penelitian ini menggunakan 8000 *dataset* yang terbagi menjadi 5 kelas, 6000 citra digunakan sebagai data uji dan 2000 citra sebagai data latih. Konfigurasi sistem diuji menggunakan *learning rate* dan *step training*. Parameter performansi yang dianalisis yaitu akurasi, *Intersection over Union* (IoU), *recall,* presisi, dan *precison rate*. Total *web* yang di-*crawl* pada proses *web crawling* sebanyak 45 *web*. Dari hasil penelitian, didapatkan konfigurasi terbaik pada *learning rate* 0.0001 dan *step training* 10K. Akurasi yang didapatkan sebesar 94.6%, IoU 0.75, *recall* 0.94, presisi 0.99, dan *precision rate* 0.87.
Kata kunci : ***face recognition, you only look once, object detection, web crawling.***

Abstract

Artificial Intelligence (AI) is a system developed to learn and apply human intelligence. Some technologies produced from the development of Al are web crawling and face recognition. Web crawling is used to get information on webpages, while face recognition is used to identify human faces. This research is implementing web crawling system using face recognition technology. This research designed a system that will scan human faces in real-time using the You Only Look Once (YOLO) object detection method, Python language programming, OpenCV library, and Requests library for web crawling. Afterward, the result from face recognition will be used as a keyword for web crawling system. This research using 8000 datasets which are divided into 5 classes, 6000 images used as data train and 2000 images as data test. The system configuration is tested using learning rate and step training. Performance parameter to be analyzed are accuracy, Intersection over Union (IoU), recall, precision, and precision rate. The total of crawled web in web crawling process is 45 web pages. From the test results, the best configuration was obtained at learning rate 0.0001 and step training 10K. The accuracy obtained is 94.6%, IoU 0.75, recall 0.94, precision 0.99, and precision rate 0.87.
Keywords: ***face recognition, you only look once, object detection, web crawling.***

1. Pendahuluan

Teknologi informasi ikut berkembang secara pesat seiring dengan berkembangnya zaman. Teknologi informasi yang sedang populer saat ini adalah *Artificial Intelligence* (AI) atau teknologi kecerdasan buatan. AI adalah sistem yang dikembangkan untuk mempelajari dan menirukan kecerdasan manusia [1]. *Face recognition* merupakan teknologi pengembangan dari AI yang digunakan untuk mengidentifikasi wajah manusia. Saat ini sudah banyak metode yang dapat digunakan untuk teknologi *face recognition* diantaranya adalah *Convolutional Neural Network* (CNN), *Faster* R-CNN, dan *You Only Look Once* (YOLO).

Selain teknologi *face recognition,* salah satu bidang ilmu dari AI adalah *web crawling. Web crawling* adalah program yang dapat mengekstrak data dari sebuah halaman *web* [2]. *Tools* yang dapat digunakan untuk melakukan *web crawling* di antaranya adalah *Scrapy, BeautifulSoup, Selenium,* dan *Requests*-HTML. Penelitian ini merancang sistem *web crawling* berdasarkan hasil identifikasi wajah manusia. Sistem ini membutuhkan sistem *web crawling* yang dapat mengambil informasi yang relevan serta dilengkapi dengan metode deteksi objek yang memiliki tingkat akurasi tinggi dan dapat bekerja secara *real-time*. Metode deteksi objek *Region*

Based CNN (R-CNN), dinilai lambat dalam mendeteksi objek karena proses *convolutional network* dilakukan untuk setiap objek proposalnya tanpa melakukan *sharing computation* [3]. *Fast* R-CNN dan *Faster* R-CNN, dapat dilatih untuk melakukan *share convolutional network* sehingga dapat meningkatkan kecepatan dan tingkat akurasi dari deteksi objek. Namun keduanya masih kurang optimal dalam mendeteksi objek secara *real-time* [4]. Sedangkan untuk sistem *web crawling,* dibutuhkan *tools* yang dapat bekerja secara efektif dalam mendapatkan informasi yang berada di internet. *Selenium* merupakan sebuah *framework* yang digunakan untuk *software testing* yang juga memiliki fungsi sebagai *crawler*, khususnya pada *website* yang bersifat dinamis [5]. Namun, proses *crawling* menggunakan *Selenium* membutuhkan waktu yang lebih lama dibandingkan mengirim *web requests* dengan *Requests* HTTP [6]. Sedangkan *BeautifulSoup,* untuk melakukan *web scraping* perlu dibantu dengan *request module* untuk mengirimkan *web request* [7].

Penelitian ini mengusulkan untuk menggunakan *library Requests*-HTML untuk sistem *web crawling* dan metode YOLO sebagai metode untuk deteksi objek pada teknologi *face recognition*. *Library Requests*-HTML dapat mengirimkan *web request* sekaligus mengekstrak informasi dari *web* yang dituju. YOLO menerapkan *single neural network* di seluruh citra selama *training* dan *testing time* [4], hal ini yang membuat YOLO memiliki kecepatan untuk deteksi objek lebih tinggi dibanding metode lain sehingga sangat cocok untuk diterapkan pada sistem deteksi objek secara *real-time*. Untuk menganalisis kinerja dari sistem yang dirancang, maka akan dilakukan pengujian terhadap parameter akurasi, *intersection over union* (IoU), *recall,* presisi, dan *precision rate*.

2. Konsep Dasar

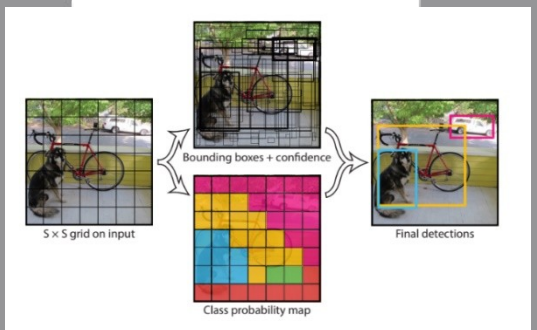
Pada bagian ini, akan dijelaskan konsep dasar dan tinjauan pustaka mengenai *You Only Look Once* (YOLO), Arsitektur *Network*, *Web Crawler, Hyper Text Markup Language* (HTML), dan *Requests*-HTML.

1.

2.

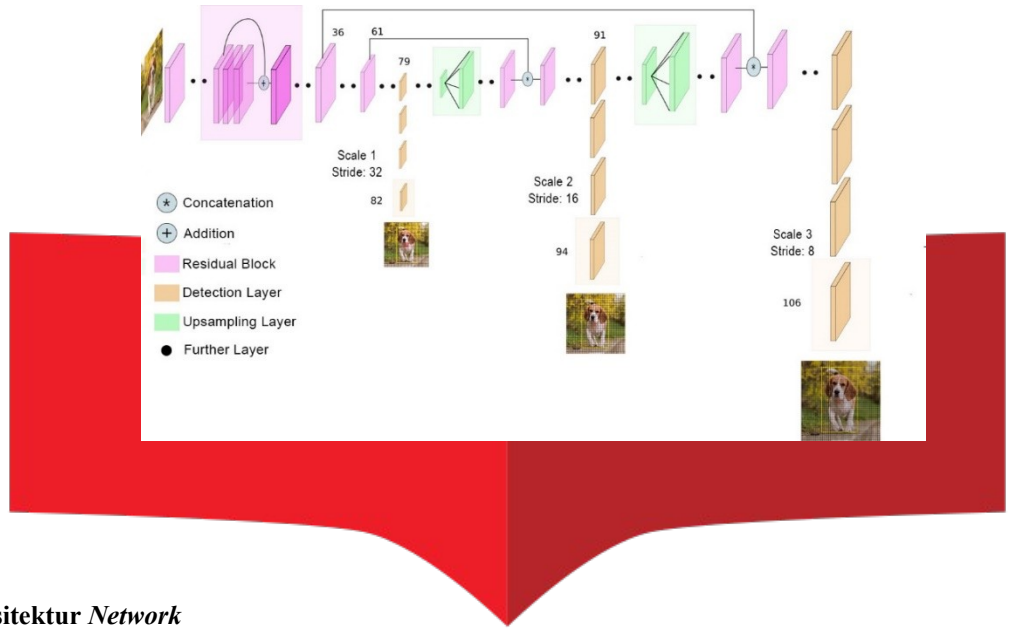
2.1. *You Only Look Once* (YOLO)

Algoritma deteksi objek *You Only Look Once* (YOLO) merupakan detektor *one-stage* pertama yang berbasis CNN. YOLO adalah algoritma deteksi objek yang ditargetkan untuk pemrosesan secara *real-time.* YOLO menggunakan *single neural network* untuk memprediksi *bounding box* dan probabilitas kelas langsung dari citra *input*. YOLO membagi citra input menjadi *grid cell* berukuran 7×7, setiap *cell* menghasilkan 2 *predicted bounding boxes,* maka akan terdapat 98 *predicted bounding boxes.* Kemudian, setiap *predicted bounding box* akan menghasilkan dua nilai *confidence score*, yaitu *confidence score* yang menggambarkan keberadaan objek dan *confidence score* dari prediksi *class* yang terdeteksi. YOLO melakukan proses deteksi objek pada setiap *grid cell* secara bersamaan, hal ini yang membuat YOLO menjadi algoritma yang sangat cepat dalam mendeteksi objek [4]. Dari 98 *predicted bounding boxes*, terdapat banyak *predicted bounding boxes* yang



memiliki *confidence score* rendah. Agar deteksi objek bekerja secara maksimal, perlu dilakukan penyesuaian pada nilai *threshold*. YOLO akan mengambil *predicted bounding boxes* dengan *confidence scores* terbaik dan menghilangkan *predicted bounding boxes* yang *confidence score*-nya berada di bawah nilai *threshold*. Proses menghilangkan *predicted bounding boxes* dengan *confidence score* yang rendah disebut *Non-Max Suppression* (NMS). Gambar 1 merupakan ilustrasi dari algoritma deteksi objek YOLO.

Gambar 1. Ilustrasi Algoritma Deteksi YOLO [4].



1.

2.

2.1.

2.2. Arsitektur *Network*

Penelitian ini menggunakan arsitektur YOLOv3. YOLOv3 merupakan pengembangan dari YOLOv1. Arsitektur YOLOv3 ditunjukkan pada Gambar 2. YOLOv3 menggunakan Darknet53 sebagai *feature extractor.* Darknet53 memiliki 53 *convolutional layers.* Namun untuk mendeteksi objek, Darknet53 menambahkan 53 *convolutional layers* sehingga terdapat 106 *fully convolutional layers*. YOLOv3 melakukan deteksi objek pada 3 skala pada 3 *layer* yang berbeda. Hal ini dilakukan untuk menyelesaikan masalah pada versi sebelumnya, YOLOv1, yang kesulitan dalam mendeteksi objek kecil. 3 *layer* yang digunakan untuk mendeteksi objek yaitu *layer* 82, *layer* 94, dan *layer* 106. *Network* akan melakukan *downsample* pada setiap *layer*-nya.

Gambar 2. Arsitektur YOLOv3 [8].

2.3. *Web Crawler*

Web crawler adalah program komputer yang melintasi jaringan internet *World Wide Web* (WWW) secara sistematis dengan maksud untuk mengumpulkan data [9]. *Web crawler* mengunduh data dari internet dan kemudian meletakannya ke dalam *local backup*. *Crawler* mulai melintasi dari satu *Uniform Resource Locator* (URL) atau lebih, kemudian mengunduh konten dari halaman URL tersebut dan mengesktrak halaman URL lain yang dibutuhkan di halaman web dan memasukkannya ke dalam antrian [10]. Proses ini diulangi hingga *crawler* memutuskan untuk berhenti apabila sudah memenuhi kondisi yang diperlukan.

1.

2.

2.1.

2.2.

2.3.

2.4. *Requests*-HTML

Request-HTML merupakan sebuah *library* pada *Python* yang mengizinkan *user* untuk melakukan *web scraping*. *Library* ini akan mengirimkan *request* pada halaman *web* mendapatkan informasi pada halaman *web* tersebut dengan cara mem-*parsing* HTML [11].

3. Desain Model Sistem

Penelitian ini membuat model sistem *face recognition* yang bekerja secara *real-time* dengan menggunakan algoritma YOLOv3. Identitas yang dihasilkan dari proses *face recognition* kemudian dijadikan sebagai *keyword* untuk pencarian di internet menggunakan teknik *web crawling*. Sistem ini menggunakan *library requests* untuk proses *crawling*. Penelitian ini memiliki skema seperti pada Gambar 3.

1.

1.1.

1.2.

1.3.

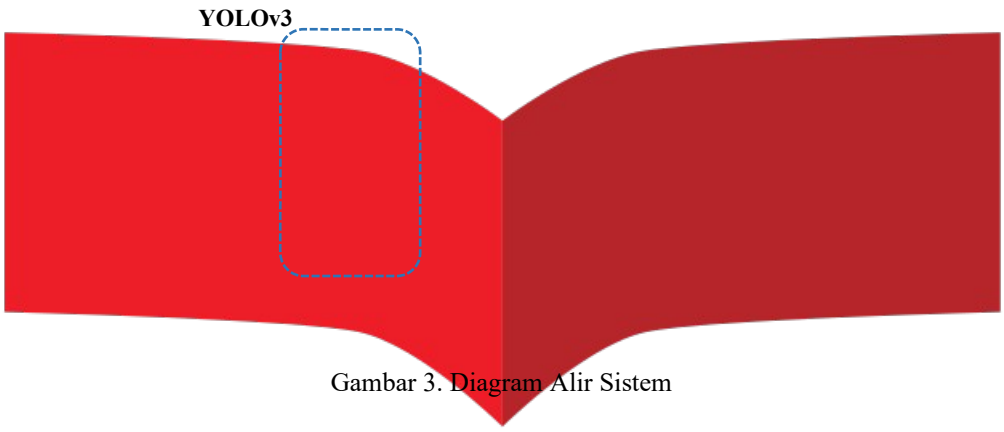
1.

2.

2.1.

2.2.

2.3.



Gambar 3. Diagram Alir Sistem

1.

2.

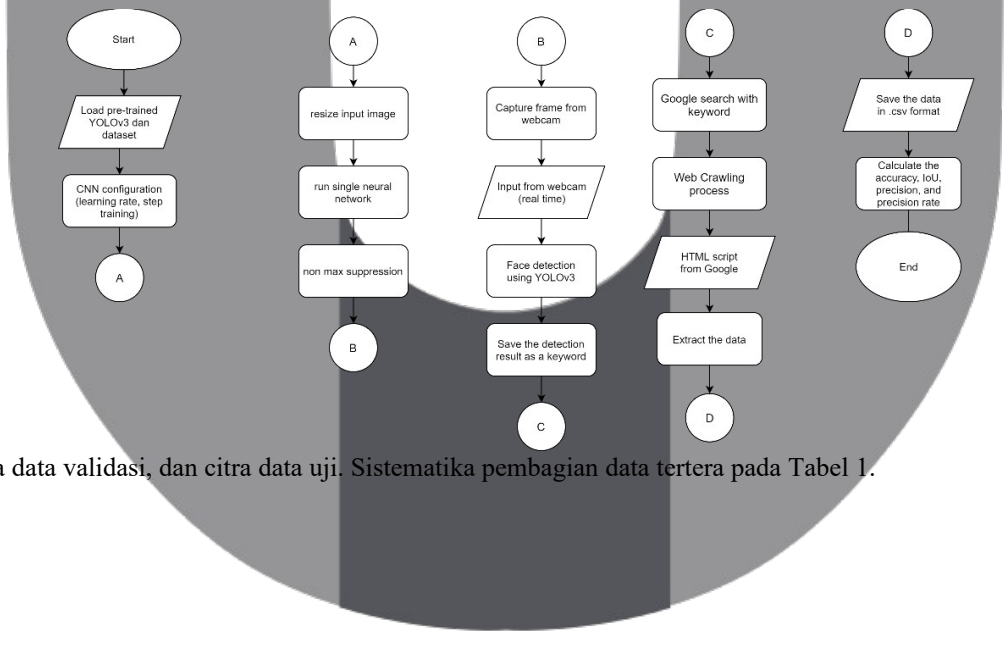
3.

3.1. *Load Pre-Trained Weights* YOLOv3 dan *Dataset*

YOLOv3 merupakan arsitektur CNN yang telah dilatih menggunakan *dataset* COCO yang memiliki 80 kelas. *Dataset* yang digunakan pada Tugas Akhir ini menggunakan 5 kelas yaitu wajah dari 5 orang yang berbeda. Proses *load pre-trained weights* YOLOv3 dan *dataset* dilakukan hanya satu kali pada saat memulai sistem.

3.2. *Dataset*

Penelitian ini menggunakan *dataset* berupa citra dengan 5 *class*, yaitu wajah dari orang yang berbeda. Citra tersebut diambil dari jarak 60 cm dengan berbagai kondisi wajah. *Dataset* dibagi menjadi tiga, yaitu citra data latih, citra data validasi, dan citra data uji. Sistematika pembagian data tertera pada Tabel 1.



Tabel 1. Sistematika data.

Jarak (cm)	Data Latih	Data Validasi	Data Uji	Kondisi Wajah
60	1500	750	500	Tanpa Penghalang
60	1500	750	500	Menggunakan Kacamata
60	1500	750	500	Menggunakan Masker
60	1500	750	500	Menggunakan Kacamata dan Masker

3.3. Konfigurasi Sistem

Konfigurasi yang digunakan pada Penelitian ini berupa *learning rate* dan *step training*. *Learning rate* yang digunakan yaitu 0.001 dan 0.0001. Sedangkan jumlah *step training* yang digunakan sebesar 10K dan 5K. Konfigurasi yang bervariasi digunakan untuk membandingkan performansi sistem dan mencari sistem dengan konfigurasi terbaik untuk diterapkan pada teknologi *face recognition*.

3.4. Proses YOLO

YOLOv3 mengubah ukuran citra menjadi 416×416 pada saat melakukan proses deteksi objek. Kemudian, YOLO menjalankan *single neural network* pada citra. YOLO menggunakan *Non-maximum suppression* (NMS) untuk memilih *predicted bounding box* terbaik. Proses deteksi YOLO diilustrasikan pada Gambar 4.

Gambar 4. Ilustrasi Proses Deteksi YOLO

3.5. Proses Deteksi YOLO

Webcam digunakan untuk menangkap video dalam ruang warna RGB secara *real-time* yang selanjutkan akan digunakan untuk proses deteksi wajah. Data *input* sistem bergantung pada kondisi wajah yang telah dijelaskan pada Tabel 1. Pada proses ini dibutuhkan cahaya yang terang agar wajah dapat mudah terdeteksi. Adapun jarak *input* antara *webcam* dengan wajah adalah 60 cm. Setelah *webcam* menangkap video, selanjutnya dilakukan proses *capture frame.* Tahap ini dilakukan untuk mengambil *sampling* dari video agar memudahkan sistem saat mendeteksi wajah. Proses *capture frame* ini mengubah *input* video menjadi citra.

Setelah data *input* video diubah menjadi citra, maka sistem akan melakukan deteksi wajah dengan YOLO. *Ouput* dari deteksi wajah berupa *bounding box.* Parameter dari *bounding box* yang akan digunakan untuk proses

web crawling adalah nama kelas. Tahap ini merupakan tahap paling penting untuk diproses pada tahap *web crawling.* Gambar 5 adalah hasil deteksi wajah dengan YOLO.

Gambar 5. Hasil deteksi dari *sub-sampling* data uji.

3.6. Pencarian di Google Menggunakan *Keyword*

Setelah wajah terdeteksi oleh sistem, nama kelas pada wajah yang terdeksi akan disimpan menjadi *keyword.* Kemudian sistem akan melakukan pencarian informasi di google dengan menggunakan *keyword* tersebut. Pencarian informasi dilakukan secara *real-time*.

3.

3.1.

3.2.

3.3.

3.4.

3.5.

3.6.

3.7. Proses *Web Crawling*, HTML *parsing*, dan Mengambil Data

Proses *crawling* akan dimulai dari alamat URL www.google.com. Kemudian sistem akan melakukan *parsing* konten *webpage* untuk mendapatkan informasi yang spesifik dengan menggunakan *library Requets* HTML. Sistem yang dirancang hanya akan mengambil 9 URL dari *search result* google untuk setiap *keyword*-nya. Sehingga total *crawled* URL pada Penelitian ini berjumlah 45 *crawled* URL. Penelitian ini mengambil informasi berupa nama *website* beserta *link* dari hasil pencarian yang telah dilakukan. Setelah data berhasil diekstrak, data kemudian disimpan dalam bentuk tabel terstruktur dengan .csv.

Gambar 6. Hasil *web crawling*.

1.

2.

3.

3.1.

3.2.

3.3.

3.4.

3.5.

3.6.

3.7.

3.8. Analisis Parameter Kinerja

Desain sistem dirancang untuk melakukan *web crawling* berdasarkan hasil dari *face recognition* dengan metode YOLO. Parameter kinerja dianalisis untuk menilai kinerja sistem. Penelitian ini menguji 4 parameter kinerja, yaitu akurasi, *Intersection over Union* (IoU), presisi, dan *precission rate* dengan keterangan sebagai berikut:

1. Akurasi
   Parameter akurasi digunakan untuk menghitung jumlah data benar dalam mendeteksi wajah. Akurasi dapat dihitung dengan menggunakan persamaan 3.1,

$$A = \frac{T}{N} \times 100\,\% \tag{3.1}$$

   dengan *T* dan *N* didefinisikan sebagai jumlah data yang dideteksi dengan benar dan total keseluruhan data.
2. IoU
   IoU digunakan untuk menghitung akurasi ketepatan prediksi dari *bounding box* dengan *groundtruth box* dari objek. IoU dapat dihitung menggunakan persamaan 3.2,

$$D = \frac{B_i \cap B_{AT}}{} \tag{3.2}$$

title	link
Kiki Widianto Profil | Facebook	https://id-id.facebook.com/public/Kiki-Wi
Kiki Widianto | Facebook	https://id-id.facebook.com/people/Kiki-W
Kiki Widianto | Facebook	https://id-id.facebook.com/people/Kiki-W

dengan B_i dan B_{AT} didefinisikan sebagai *groundtruth box* dan *boundary box actual*.

3. *Recall*
   Parameter *recall* digunakan untuk menghitung sesnsitifitas dari model sistem yang dirancang Untuk menghitung *recall* dapat menggunakan *confusion matrix* dengan persamaan 3.3,

$$R = \frac{TP}{TP + FN} \tag{3.3}$$

   dengan *TP* dan *FN* didefinisikan sebagai hasil deteksi *true positive* dan *false negative*.

4. Presisi
   Parameter presisi digunakan untuk menghitung ketepatan sistem dalam mendeteksi objek. Presisi menggunakan *confusion matrix* dengan persamaan 3.4,

$$P = \frac{TP}{TP + FP} \tag{3.4}$$

   dengan *TP* dan *FP* didefinisikan sebagai hasil deteksi *true positive* dan *false positive*.

5. *Precision Rate*
   Parameter *precission rate* menunjukkan tingkat keterkaitan antara *crawled web* dengan topik yang dicari. *Precission rate* dapat dihitung dengan menggunakan persamaan 3.5,

$$Pr = \frac{R}{N} \tag{3.5}$$

   dengan R dan N didefinisikan sebagai jumlah *crawled web* yang relevan dan total *crawled web.*

4. Hasil dan Analisis

Bagian ini berisi hasil pengujian sistem terhadap parameter akurasi, IoU, presisi, dan *precision rate*. Pengujian menggunakan 2 perbedaan konfigurasi sistem yaitu *learning rate* dan *step*.

4.1. Skenario Pengujian

Penelitian ini menggunakan 2000 citra data uji sesuai dengan Tabel 1. Terdapat 4 skenario pengujian yang dilakukan untuk mendapatkan konfigurasi terbaik dari model sistem yang telah dirancang. Berikut skenario yang diujikan:

1. Skenario 1: Pengujian terhadap parameter Akurasi
   Pengujian parameter akurasi bertujuan untuk mengetahui kecerdasan sistem yang telah dirancang dalam mengenali objek pada citra yang belum pernah dilatih oleh sistem. Suatu objek dapat dikatakan akurat apabila memiliki nilai IoU ≥ 0.5. Pada tahap ini, sistem yang telah dilatih akan diuji dengan 2000 citra uji dengan ruang warna RGB untuk mengenali objek pada citra data uji tersebut.
2. Skenario 2: Pengujian terhadap parameter IoU
   Pengujian parameter IoU bertujuan untuk mengetahui seberapa akurat letak *predicted bounding boxes* terhadap *bounding boxes groundtruth.* Pada umumnya, sistem yang baik adalah sistem yang memiliki nilai IoU ≥ 0.5. Nilai IoU yang semakin mendekati angka 1 menandakan bahawa jarak antara *predicted bounding boxes* dengan *bounding boxes groundtruth* semakin dekat.
3. Skenario 3: Pengujian terhadap parameter *Recall*
   Pengujian parameter *recall* bertujuan untuk mengetahui tingkat sensitifitas dari model sistem dalam mendeteksi objek. Parameter *recall* menghitung keberhasilan sistem dalam menemukan kembali sebuah informasi (*true positive rate*).
4. Skenario 4: Pengujian terhadap parameter Presisi
   Pengujian parameter presisi dilakukan untuk mengetahui tingkat keakuratan sistem dalam mendeteksi objek pada citra data uji. Nilai presisi yang ideal adalah ketika mendekati angka 1. Di mana sistem berhasil mendeteksi objek dengan kategori *true positive* lebih banyak dibanding kategori *false positive*.
5. Skenario 5: Pengujian terhadap parameter *Precision Rate*
   Pada skenario 4, pengujian dilakukan apabila proses *face recognition* telah selesai. Sistem akan mencari informasi di internet dengan menggunakan *output* dari proses *face recognition* sebagai *keyword*. Pengujian dilakukan dengan 9 *crawled web* untuk setiap kelasnya, sehingga total *crawled web* sebanyak 45 *web*. Pengujian ini bertujuan untuk mengetahui tingkat ketepatan sistem dalam mengumpulkan informasi yang berada di internet. *Web* dapat dikatakan relevan apabila memiliki informasi yang langsung mengarah kepada *keyword* yang digunakan.

4.

5. Data Hasil Pengujian Sistem

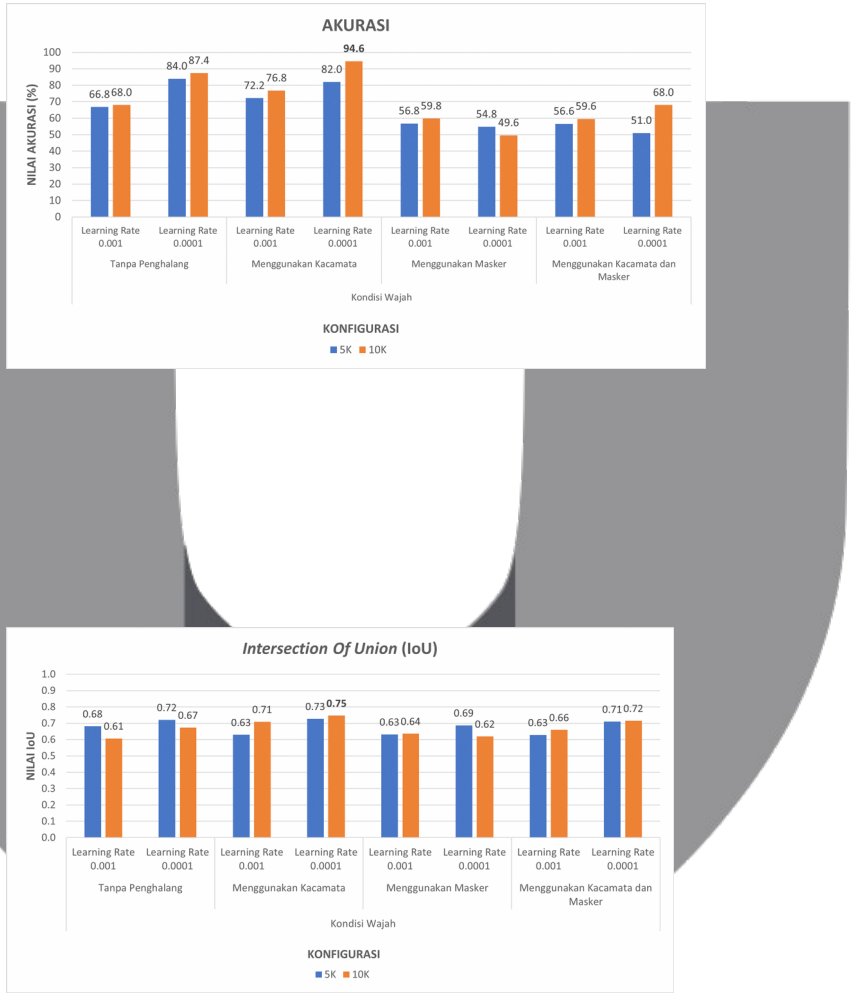
Penelitian ini memiliki 2 konfigurasi. Konfigurasi pertama dilatih menggunakan *learning rate* 0.001 sedangkan konfigurasi kedua dilatih menggunakan *learning rate* 0.0001. Kemudian setiap konfigurasi diuji pada *step* 5K dan 10K.

Grafik hasil pengujian sistem terhadap parameter akurasi dapat dilihat pada Gambar 7. Nilai akurasi terbaik yang didapat yaitu 94.6%. Akurasi tertinggi didapat pada konfigurasi *learning rate* 0.0001 pada *step* 10K dengan kondisi wajah menggunakan kacamata. Pada grafik tersebut, dapat dilihat nilai akurasi mengalami kenaikan dari *learning rate* 0.001 ke *learning rate* 0.0001 pada kondisi wajah tanpa penghalang, wajah dengan kacamata, serta wajah dengan kacamata dan masker. Nilai akurasi mengalami penurunan dari *learning rate* 0.001 ke *learning rate* 0.0001 pada kondisi wajah menggunakan masker.

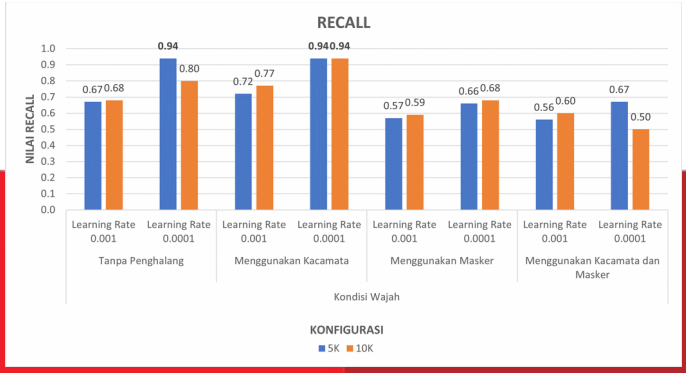
Gambar 7. Hasil Skenario 1.

Grafik hasil pengujian sistem terhadap parameter IoU ditunjukkan pada Gambar 8. Nilai IoU terbaik yang didapatkan yaitu 0.75. IoU terbaik didapat pada konfigurasi *learning rate* 0.0001 dan *step training* 10K dengan kondisi wajah menggunakan kacamata. Hal ini menandakan bahwa pada konfigurasi tersebut sistem memiliki jarak terdekat antara *predicted bounding boxes* dengan *bounding boxes groundtruth*. Pada grafik tersebut, terlihat bahwa nilai IoU mengalami kenaikan dari *learning rate* 0.001 ke *learning rate* 0.0001. Nilai IoU berkisar antara 0 sampai dengan 1.

Gambar 8. Hasil Skenario 2.



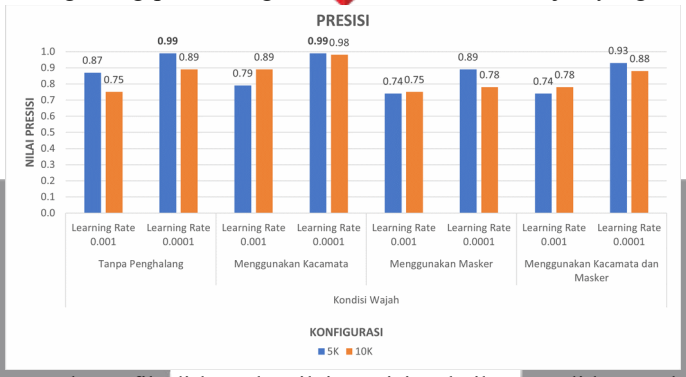
Grafik hasil pengujian sistem terhadap parameter *recall* ditunjukkan pada Gambar 9. Pengujian *recall* dilakukan dengan menghitung perbandingan antara hasil deteksi objek yang dikategorikan sebagai *true positive*



dan *false negative*. Pada grafik di bawah, nilai *recall* terbaik yang didapat adalah 0.94. *Recall* terbaik didapat pada konfigurasi *learning rate* 0.0001 dan *step* 5K dan 10K dengan kondisi wajah tanpa penghalang dan wajah menggunakan kacamata.

Gambar 9. Hasil Skenario 3.

Grafik hasil pengujian sistem terhadap parameter presisi ditunjukkan pada Gambar 10. Pengujian presisi dilakukan dengan menghitung perbandingan antara hasil deteksi objek yang dikategorikan sebagai *true*



positive dan *false positive*. Pada grafik di bawah, nilai presisi terbaik yang didapat adalah 0.99. Presisi terbaik didapat pada konfigurasi *learning rate* 0.0001 dan *step* 5K dengan kondisi wajah tanpa penghalang dan wajah menggunakan kacamata. Nilai presisi yang ideal adalah ketika mendekati angka 1.

Gambar 10. Hasil Skenario 4.

Setelah didapatkan nilai *recall* dan presisi, maka dapat diketahui pulai nilai *mean average precision* (mAP). mAP adalah sebuah *metrics* yang digunakan untuk mengevaluasi kinerja pada model deteksi objek. Tugas Akhir ini menggunakan nilai *threshold* IoU = 0.5, maka mAP yang dihitung adalah mAP@0.5. Nilai mAP didapatkan dengan mencari rata-rata dari nilai *Average Precision* (AP). mAP dapat dihitung dengan persamaan 5.1,

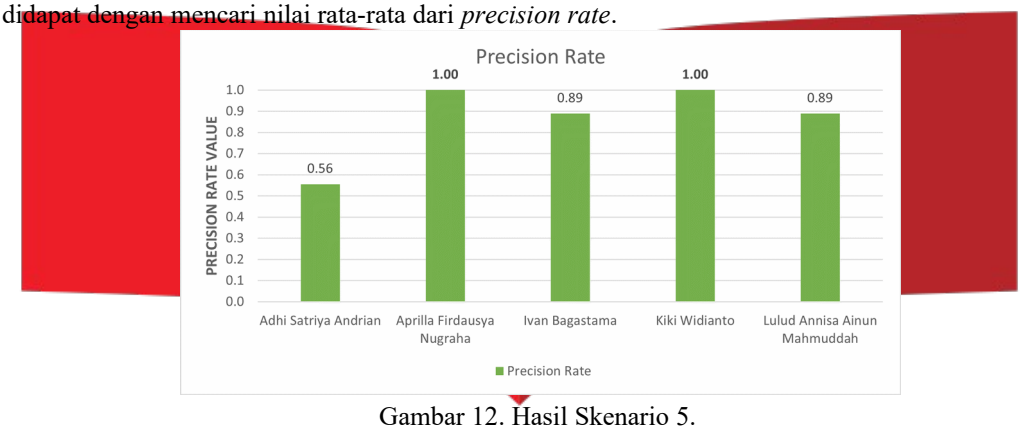
$$mAP = \frac{1}{n}\sum_{k=1}^{k-n} AP_k \qquad (5.1)$$

$$AP = \int_0^1 p(r)\,dr \tag{5.2}$$

dimana *n, p,* dan *r* didefinisikan sebagai jumlah *class*, presisi, dan *recall*. AP merupakan area di bawah kurva presisi-*recall*. Grafik mAP untuk setiap konfigurasi sistem ditunjukkan pada Gambar 11.

Gambar 11. Hasil mAP.

Pengujian *precision rate* dilakukan dengan dengan mengambil informasi berdasarkan hasil pencarian di google dengan menggunakan *keyword* berupa nama kelas dari hasil proses *face recognition*. Sistem hanya mengambil informasi dari 9 URL pada halaman hasil pencarian google. Suatu *web* dikatakan relevan apabila informasi yang terdapat di dalamnya sesuai dengan *keyword* yang digunakan. Jika terdapat kesalahan penulisan pada *web* di hasil pencarian, maka *web* tersebut dikatakan tidak relevan. Pada Gambar 12, nilai *precision rate* terbaik adalah 1 dengan *keyword* Aprilla Firdausya Nugraha dan Kiki Widianto. Sedangkan *precision rate* terburuk adalah 0.56 dengan *keyword* Adhi Satriya Andrian. Nilai *precision rate* keseluruhan sistem adalah 0.87, nilai ini didapat dengan mencari nilai rata-rata dari *precision rate*.



Gambar 12. Hasil Skenario 5.

6. Kesimpulan

Sistem *web crawling* berbasis teknologi *face recognition* dengan metode YOLO telah berhasil diimplementasikan dan mendapatkan akurasi 94.6%, IoU 0.75, *recall* 0.94, presisi 0.99, dan *precision rate* 0,87. Konfigurasi terbaik untuk sistem *face recognition* didapatkan saat *learning rate* 0.0001, *step training* 10K, dan dengan kondisi wajah menggunakan kacamata. Berdasarkan hasil pengujian, performansi sistem membaik ketika nilai *learning rate* 0.001 diubah menjadi 0.0001. Hal ini terjadi karena semakin besar nilai *learning rate* dapat menyebabkan *loss* yang naik turun. Selain itu, nilai akurasi pada sistem juga dipengaruhi oleh kondisi wajah dan kualitas dari citra data uji. Semakin banyak bagian wajah yang tertutupi, maka akurasinya akan semakin mengecil.

7. Referensi:

[1] X. Liu, "Artificial intelligence and modern sports education technology," *Proc. - 2010 Int. Conf. Artif. Intell. Educ. ICAIE 2010*, pp. 772–776, 2010, doi: 10.1109/ICAIE.2010.5641441.

[2] J. Cho, "Crawling the web: discovery and maintenance of large-scale web data," *Plant Physiol.*, vol. 154, no. November, pp. 1–17, 2001.

[3] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[5] G. S. Kalra, R. S. Kathuria, and A. Kumar, "YouTube Video Classification based on Title and Description Text," *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, vol. 2019-Janua, pp. 74–79, 2019, doi: 10.1109/ICCCIS48478.2019.8974514.

[6] I. Savinkin, "Web Scraping & data mining," 2020. [Online]. Available: https://webscraping.pro/using-selenium-webdriver-for-website-scraping/. [Accessed: 01-Mar-2020].

[7] J. Grimes, "BestProxy Reviews," 2020. [Online]. Available: https://www.bestproxyreviews.com/scrapy-vs-selenium-vs-beautifulsoup-for-web-scraping/. [Accessed: 01-Mar-2021].

[8] K. Alderliesten, "YOLOv3 — Real-time object detection," 2018. [Online]. Available: https://medium.com/analytics-vidhya/yolov3-real-time-object-detection-54e69037b6d0. [Accessed: 27-Feb-2021].

[9] A. G. K. Leng, R. Kumar P, A. K. Singh, and R. K. Dash, "PyBot: An algorithm for web crawling," *2011 Int. Conf. Nanosci. Technol. Soc. Implic. NSTSI11*, 2011, doi: 10.1109/NSTSI.2011.6111993.

[10] Y. Ren, "web scraping in python using SCRAPY," *2018 15th Int. Conf. Serv. Syst. Serv. Manag.*, pp. 1–6, 2018.

[11] K. Reitz, "Request-HTML: HTML Parsing for Humans (writing Python 3)!," 2017. [Online]. Available: https://requests.readthedocs.io/projects/requests-html/en/latest/. [Accessed: 10-Feb-2021].