CESS (Journal of Computer Engineering System and Science) **p-ISSN :2502-7131**
Vol. 6 No. 2 Juli 2021 **e-ISSN :2502-714x**

DETEKSI DAN PENGENALAN OBJEK DENGAN MODEL *MACHINE LEARNING*: MODEL YOLO

Qurotul Aini[1], Ninda Lutfiani[2], Hendra Kusumah[3], Muhammad Suzaki Zahran[4]

[1, 2] *Program Magister Departemen Informatika, Fakultas Sains dan Teknologi, Universitas Raharja*
Jl. Jenderal Sudirman No.40, RT.002/RW.006, Cikokol, Kec. Tangerang, Kota Tangerang, Banten 15117

[3] *Program Magister Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Raharja*
Jl. Jenderal Sudirman No.40, RT.002/RW.006, Cikokol, Kec. Tangerang, Kota Tangerang, Banten 15117

[4] *Program Studi Sistem Komputer, Fakultas Sains dan Teknologi, Universitas Raharja*
Jl. Jenderal Sudirman No.40, RT.002/RW.006, Cikokol, Kec. Tangerang, Kota Tangerang, Banten 15117
[1]aini@raharja.info, [2]ninda@raharja.info, [3]hendra.kusumah @raharja.info, [4]m.suzaki@raharja.info

Abstrak— **Ranah pengenalan dan pendeteksian objek telah diminati oleh banyak pihak sejak ditemukannya *Computer Vision* pada 1960-an, baik di bidang industri maupun medis. Sejak saat itu, mulai banyak penelitian yang berfokus pada ranah pengenalan dan pendeteksian objek dengan berbagai jenis model algoritma yang mampu mengenali dan mendeteksi objek pada suatu gambar. Namun tidak semua model algoritma ini efisien dan efektif dalam penerapannya. Kebanyakan dari model-model algoritma yang ada sebelumnya memiliki tingkat kerumitan yang cukup tinggi. Di sini, penulis berusaha menjelaskan dan memperkenalkan model algoritma YOLO (*You only look once*) yang memiliki kemampuan kecepatan pemrosesan pendeteksian gambar yang cukup tinggi dan dengan akurasi yang mampu menyaingi model-model algoritma yang ada sebelumnya. Namun, selain akurasinya yang cukup tinggi, YOLO juga masih memiliki banyak kekurangan, seperti YOLO v3 yang masih kesulitan mengenali objek-objek gambar yang berukuran medium dan besar, serta YOLO v5 yang masih belum ada penjelasan saintifik resmi sehingga masih belum dapat dijelaskan lebih lanjut. Pada penelitian ini, Penulis juga menggunakan metode literatur review dengan membandingkan tiap jurnal ilmiah yang ada sehingga mencukupi informasi yang dibutuhkan.**

Kata Kunci— **deteksi objek, YOLO, pengenalan objek, algoritma, efisiensi.**

Abstract— O**bject recognition and detection have been in demand by many parties since Computer Vision in the 1960s, both in the industrial and medical fields. Since then, many studies have focused on object recognition and detection with various types of algorithm models that can recognize and detect objects in an image. However, not all of these algorithm models are efficient and effective in their application. Most of the previous algorithm models have a relatively high level of complexity. Here, the author tries to explain and introduce the YOLO (You only look once) algorithm model, which has a fairly high image detection processing speed capability and accuracy that can compete with previous algorithm models. However, YOLO also still has many shortcomings in addition to its high accuracies, such as YOLO v3, which still has difficulty recognizing medium and large-sized image objects, and YOLO v5, which still has no official scientific explanation so that it cannot be explained further. In this study, the author also uses the literature review method by comparing each existing scientific journal to provide the required information.**

Keywords— **object detection, YOLO, object recognition, algorithm, efficiency**

I. PENDAHULUAN

Ranah pengenalan dan pendeteksian objek [1]–[3], pada *Computer Vision* saat ini sedang berkembang pesat dan mulai diterapkan di berbagai bidang, dari industri hingga medis. Hal ini dapat dilihat dari berbagai macam riset yang meluas, sebagian berfokus pada penerapan dan penyesuaian model *machine learning* [4]–[6], sedangkan sebagian lainnya berfokus pada pengembangan model baru guna menjawab permasalahan dan tantangan permasalahan spesifik [7], [8], [9], [10] pada *Computer Vision*, terutama pada efisiensi model. Oleh karena itu, penelitian ini dilakukan untuk membantu para pengembang baru untuk dapat mengetahui perbedaan dan perkembangan model YOLO, baik dari segi efisiensi maupun kecepatan model. Selain itu, penelitian ini juga membantu pengembang pemula untuk dapat membantu memahami model YOLO dengan mudah sehingga model YOLO ini dapat ditingkatkan dan dikembangkan serta dapat bersaing dengan lebih kompetitif dengan model lain.

Ranah pengenalan dan pendeteksian objek telah berkembang dari waktu ke waktu. Perkembangan pengenalan dan pendeteksian objek secara garis besar dapat dibagi menjadi dua era [11], yaitu era tradisional

deteksi objek di mana proses pendeteksian objeknya masih dilakukan secara manual yang mana manusia masih sangat terlibat dalam memberikan masukan ke sistem mengenai apa saja yang perlu dideteksi oleh sistem. Selanjutnya, yaitu era *deep learning*, yang merupakan bagian dari metode *machine learning* yang memungkinkan algoritma sistem mampu untuk belajar dan berkembang dengan sendirinya hanya dengan data yang disediakan dan pengalaman yang dialami tanpa peran manusia yang signifikan [12], [13], [14]. Seiring berkembangnya *deep learning*, muncul lah model-model baru pada ranah pengenalan dan pendeteksian objek pada *Computer Vision*. Mulai dari *Region based Convolutional Neural networks* (R-CNN) [15], [16] yang ditemukan oleh Ross Girshick, *Spatial Pyramid Pooling Network* (SPP-Net) oleh K. He et al, *fast* R-CNN, *Faster* R-CNN [17], [18], [19], *You only look once* (YOLO) [20]–[23] hingga RetinaNet [24], [25]. Model-model pengenalan dan pendeteksian objek ini dapat diterapkan di berbagai bidang, mulai dari pengenalan karakter optik, mobil *self-driving*, *object tracking*, pengenalan dan pendeteksi wajah, verifikasi identitas melalui kode iris, ekstraksi objek dari citra ataupun video, deteksi pejalan kaki, pencitraan medis, dan lain sebagainya.

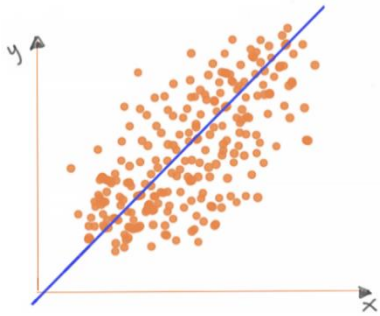
Susunan penelitian ini yaitu bagian kedua akan memperkenalkan beberapa hal penting untuk dapat memahami model yang akan dibahas, bagian ketiga akan meneliti mengenai beberapa model, bagian keempat akan membandingkan mengenai model-model yang telah diteliti sebelumnya, dan bagian kelima akan menyimpulkan topik bahasan.

II. TINJAUAN PUSTAKA

A. *Overfitting vs. Underfitting*

Data yang menyebar secara tidak teratur pada suatu grafik dapat menyebabkan suatu keadaan yang disebut *overfitting* dan *underfitting* [26], [27]. *Overfitting* merupakan keadaan di mana algoritma yang buat dapat bekerja dengan baik pada tahapan *training*, namun memiliki performa yang buruk pada tahapan *testing* saat melakukan pengujian data baru. *Overfitting* terjadi akibat model yang dibuat terlalu banyak belajar pada dataset *training*. *Overfitting* juga disebut sebagai *'problem of high variance'* (permasalahan varian tinggi). Sedangkan *underfitting* merupakan suatu keadaan yang terjadi ketika algoritma yang dibuat memiliki performa yang tidak baik atau tidak dapat dijalankan sama sekali baik pada tahapan *training* ataupun *testing*. *Underfitting* dapat terjadi akibat dua hal, (i) model yang dibuat terlalu sederhana, ataupun (ii) model yang dibuat memiliki regulasi yang terlalu banyak. Kedua faktor *underfitting* ini dapat membuat model kesulitan dalam menyesuaikan performa pada dataset yang ada.

B. *Analisis Regresi*

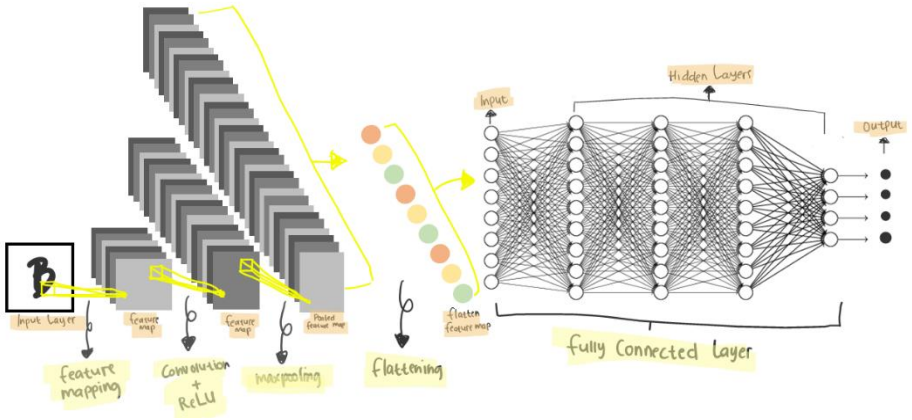


Gbr. 1 Contoh grafik *Simple Linear Regression*

Regresi merupakan teknik statistika yang diadaptasikan untuk membuat model *machine learning*. Regresi digunakan dalam bentuk analisis sehingga lebih dikenal sebagai analisis regresi. Regresi sederhananya merupakan teknik analisis yang menggambarkan hubungan antara suatu variabel terhadap variabel lainnya yang mana satu diantara variabel tersebut merupakan variabel dependen. Umumnya dirumuskan sebagai:

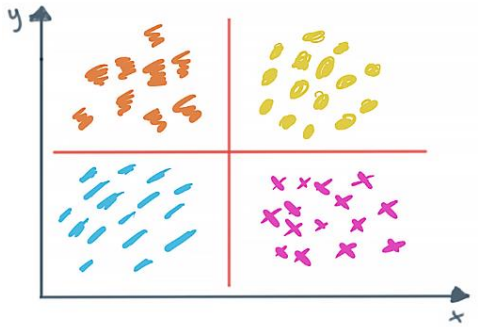
$$Y_i = f(X_i, \beta) + e_i \qquad (1)$$

Umumnya regresi digunakan untuk mengolah data kontinyu. Kegunaan regresi yaitu untuk memprediksi perubahan variabel dependen berdasarkan perubahan pada satu atau lebih variabel dependen berdasarkan faktor tertentu sehingga didapatkan hubungan antara keduanya. Regresi secara garis besar dibagi menjadi dua kategori, yaitu regresi linear dan regresi nonlinear. Regresi linear memiliki beberapa teknik penyelesaian seperti *simple linear regression*, *multi linear regression*, *polynomial regression*, *support vector machine*, *decision tree*, *random forest*, dan sebagainya yang tentu saja ditujukan untuk menyelesaikan data kontinyu yang bersifat linear. Sedangkan regresi nonlinear dapat diterapkan pada data nonlinear dengan beberapa metode yaitu seperti metode bayesian, *percentage regression*, dan *least absolute deviations*. Pada penjelasan kali ini, penulis tidak akan berfokus pada teori-teori mendetail mengenai regresi karena teori tersebut masuk ke dalam pembahasan statistika. Analisis regresi sangat baik untuk menganalisis serta mengukur hubungan antara dua atau lebih variabel minat (*variable of interest*) karena metode statistiknya yang sangat kuat sehingga menjadikannya tepat dan akurat. Namun, dibalik keakuratannya ini, regresi masih memiliki batasan yang cukup serius, yaitu seperti hubungan antar variabel-variabelnya yang dianggap tetap padahal pada kenyataannya hubungan-hubungan tiap variabel bersifat dinamis. Selain itu, prosedur kalkulasi dan analisis yang cukup rumit juga menjadi halangan bagi regresi selain tidak dapat diterapkannya pada kasus permasalahan kualitatif.



Gbr. 3 Langkah-langkah pemrosesan *machine learning*

C. Analisis Klasifikasi

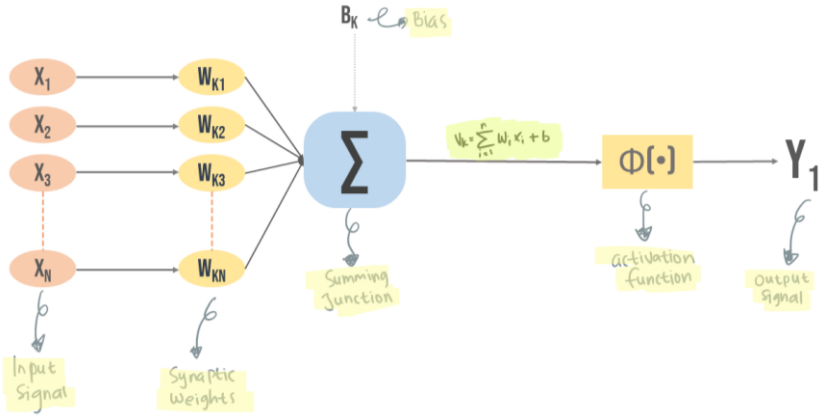


Gbr. 2 Contoh grafik *Simple Classification*

Klasifikasi [28], [29] merupakan suatu metode analisis yang digunakan untuk mengidentifikasikan dan menetapkan suatu data ke dalam kategori tertentu untuk dapat melakukan penganalisisan yang lebih akurat. Umumnya, analisis klasifikasi dapat diterapkan pada memprediksi suatu kejadian, ataupun memutuskan suatu permasalahan tertentu. Analisis klasifikasi dapat diterapkan pada dataset yang bersifat diskrit yang pada dasarnya jenis analisis ini dapat dibagi menjadi dua jenis, yaitu *binary classifier* dan *multiclass classifier*. Metode-metode yang dapat digunakan pada analisis klasifikasi ada berbagai macam, yaitu seperti *binary and multiclass classification* dan *feature vector* [30]–[33]. Ada beberapa model algoritma yang telah dibuat oleh peneliti sebelumnya guna menjawab permasalahan-permasalahan yang umum dihadapi seperti algoritma *naive bayes classifier* dan *perceptron* pada kasus klasifikasi linear. Selain itu ada juga *least squares support vector machine* yang ditemukan oleh Suykens. Penerapan analisis klasifikasi ini cukup luas, diantaranya yaitu pada ranah *Computer Vision* dapat diterapkan pada *medical imaging* dan *optical character recognition*. Selain itu bisa juga diterapkan pengenalan tulisan tangan, identifikasi biometris, dan klasifikasi biologis. Analisis klasifikasi dapat mengetahui hubungan tiap-tiap grup data serta membantu mempelajari asal-usul dan evolusi data.

D. Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) [28], [34]–[36], spesifik dikenal sebagai *Shift Invariant atau Space Invariant Artificial Neural networks* (SIANN), merupakan sebuah model *machine learning* yang penerapannya menggunakan arsitektur *shared-weight* dengan mengandalkan konvolusi [37]. Konvolusi merupakan operasi matematis [38] yang bertujuan untuk menghasilkan suatu fungsi sehingga didapatkan gambaran rupa fungsi baru dari hasil perubahan fungsi awal tadi. Umumnya dirumuskan sebagai:



Gbr. 4 Langkah-langkah pemrosesan perceptron

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau, \quad (2)$$

sebenarnya CNN merupakan pengembangan model perceptron yang diregulasi sehingga semua layer pada perceptron terhubung satu sama lain [39] yang mengakibatkan *multilayer perceptron*. Regulasi pada CNN, misal *penalizing parameter* saat *training* dan *trimming connectivity*, ditujukan agar tidak terjadi *overfitting*.

CNN diperkenalkan oleh W. Zhang, dkk., pada 1988 dengan tujuan untuk mengenali karakter gambar. Namun, arsitektur dan algoritma pelatihan CNN baru dikembangkan dan dimodifikasi pada 1991 dan kemudian diterapkan pada pemrosesan gambar medis (*medical image processing*) dan pendeteksi kanker payudarah otomatis pada Mammogram. Pengenalan gambar dengan CNN awalnya diperkenalkan oleh Yann Lecun, dkk. pada 1989. Saat itu, Yann Lecun, dkk., menggunakan *backpropagation* untuk mempelajari *convolution kernel coefficient* langsung dari gambar angka tulisan tangan. Lalu berkembang dengan ditemukannya LeNet-5 oleh Lecun pada 1998 yang diterapkan untuk mengenali tulisan tangan pada cek di bank-bank. Kemudian, dengan tujuan untuk mempercepat dan meningkatkan performa pada tahap *training* agar didapatkan *error rate* yang lebih sedikit, maka pemrosesan CNN beralih dari penggunaan CPU ke penggunaan Multi GPU. Penggunaan GPU mulai dilakukan sejak 2004 oleh K. S. Oh dan K. Jung yang berhasil menunjukkan hasil performa yang luar biasa pada papernya yang berjudul '*GPU implementation of neural networks*'. Pada papernya dinyatakan bahwa *Standard Neural networks* memiliki kecepatan 20x lipat lebih cepat dengan GPU dibandingkan dengan CPU [40]. Kemudian, pada 2006, K. Chellapilla, dkk., berhasil mengimplementasikan CNN pada GPU yang diketahui memiliki kecepatan 4x lebih cepat dibandingkan dengan CPU.

CNN bekerja dengan menggunakan prinsip neuron pada otak manusia. Terdapat tiga tahapan utama yang digunakan [41] di CNN pada umumnya. Pertama, input gambar pada CNN akan melalui proses konvolusi dan ReLU layer untuk diproses secara konvolusi. Proses ini bertujuan untuk memperkecil ukuran gambar dengan cara mengalikan informasi array yang didapat dari input gambar dengan *feature detector* pada array tersebut [41]–[43] yang akhirnya akan didapatkan sebuah *feature map* yang ukuran arraynya lebih kecil dibandingkan informasi array pada gambar namun tidak mengurangi informasi dari array input sebelumnya. Lalu *feature map* akan akan masuk ke dalam ReLU layer (*Rectified Linear Unit*) untuk meningkatkan nonlinearitas pada objek-objek gambar dengan menghilangkan *block value* pada *shadow image* sehingga didapatkan informasi gambar yang lebih baik untuk diproses [44]. Proses pada ReLU layer sebenarnya opsional, namun baik dilakukan untuk meningkatkan kualitas hasil pada *feature map*. Kedua, *feature map* yang dihasilkan akan dimasukkan ke *max pooling layer* di mana akan terjadi proses pengurangan ukuran array lagi sebanyak hampir 75% serta pengurangan parameter yang dimiliki agar tidak terjadi *overfitting* informasi pada *neural networks*. Proses *max poo*ling ini tidak mengutamakan *special invariant* seperti perbedaan arah, rotasi, dan sebagainya. Proses *max pooling* bekerja dengan mengambil nilai-nilai terbesar dari tiap-tiap array matriks 2x2 ataupun 3x3 pada *feature map* sehingga didapatkan pooled *feature map*. Selain *max pooling*, dapat juga dilakukan *averaging/mean pooling* (*subsampling*) yang mengganti nilai terbesar dengan nilai rerata dari tiap array matriks bagian *feature map*. Proses pooling ini lebih umum dikenal sebagai *downsampling* karena fungsi utamanya yaitu untuk mengurangi ukuran pada informasi gambar yang dimiliki. Selanjutnya melalui tahap terakhir, yaitu *flattening* di mana *pooled feature map* akan diratakan menjadi array 1 kolom (*flatten*) untuk mempermudah proses input data ke *neural networks*. Lalu proses *neural networks* akan dijalankan sehingga dihasilkan output klasifikasi sesuai yang diinginkan.

Penggunaan CNN sangat menguntungkan bagi manusia karena dapat mengurangi peran manusia dalam improvisasi fungsi-fungsinya. Selain itu dependensi pada tahap *pre-processing*-nya relatif lebih sedikit serta memiliki akurasi tertinggi diantara algoritma-algoritma pemrosesan gambar lainnya. Namun, CNN tidak dapat memprediksikan lokasi objek pada gambar serta kekurangan kemampuan sebagai spatial invariant untuk input data. Selain itu, untuk mendapatkan performa yang bagus, perlu menggunakan data pelatihan yang banyak. Penerapan CNN secara spesifik ada pada pemrosesan gambar, seperti pengenalan video, sistem rekomendasi, klasifikasi dan segmentasi gambar.

E. Bounding Boxes Regression

Bounding boxes regression, umum dikenal sebagai *bounding box* atau *b-box* saja, merupakan teknik yang digunakan untuk menandai lokasi objek pada gambar dengan menggambarkan persegi pada area objek gambar tersebut. Komponen pada *bounding box* tersebut yang diperlukan yaitu P_x dan P_y yang bertindak sebagai titik tengah objek serta P_w dan P_h yang bertindak sebagai jarak sisi objek pada gambar. *Bounding box* digunakan untuk menggabungkan area objek yang diajukan atau *anchor box* sesuai dengan target objek yang telah didefinisikan sebelumnya pada kelas objek tertentu.

F. Anchor box

Anchor box merupakan kumpulan dari *predefined bounding box* dengan berbagai macam bentuk. Jadi *bounding box* merupakan *anchor box* yang telah disatukan dan *anchor box* merupakan *box* yang mendeteksi bagian-bagian objek dengan skala tertentu. Penggunaan *anchor box* ini bertujuan untuk meningkatkan akurasi saat menandai objek.

G. AlexNet

AlexNet pertama kali diperkenalkan oleh Alex Krizhevsky, dkk. , pada 2010 di ajang ImageNet *Large Scale Recognition Challenge* (ILSVRC), namun masih dianggap kurang menarik saat itu karena model arsitekturnya mulai mengimplementasikan GPU daripada CPU. pada 2012, Alex, dkk., berhasil membuktikan bahwa AlexNet dapat mengalahkan model-model sebelumnya dengan memanfaatkan kemampuan GPU serta berhasil memenangkan ajang ILSVRC di kategori *image recognition*. AlexNet berhasil mendapatkan 15.4% *error rate* pada ILSVRC 2012 dan memiliki top-1 dan top-5 *error rate* sebesar 37.5% dan 17%. AlexNet memiliki 5 *convolutional layers* dan 3 *fully-connected layers*. Masalah utama yang dihadapi oleh AlexNet yaitu *overfitting* karena memiliki parameter yang sangat banyak, yaitu lebih dari 60 juta parameter dengan ukuran input tetap di gambar RGB. Alex, dkk., mengatasi masalah *overfitting* ini dengan augmentasi data dan teknik baru yang mereka kembangkan yang disebut *dropout*. Yang membuat AlexNet spesial ada pada fitur-fiturnya, seperti ReLU nonlinearitas yang lebih cepat saat *training*, multi GPU, dan *overlapping pooling*.

Selain itu, AlexNet sangat bergantung dengan layer konvolusinya, kehilangan salah satu layer konvolusinya dapat menyebabkan turunnya performa *network*-nya. Performa AlexNet juga masih terlampau jauh oleh model yang lebih kompleks seperti GoogleNet dan ResNet.

H. VGG-16

Diperkenalkan oleh Simonyan dan Zisserman dari *University of Oxford* pada 2014 untuk kompetisi ImageNet (ILSVRC). VGG-16 lebih berfokus pada *convolutional layer* dengan 3x3 *filter* pada *stride* 1 dan 2x2 *padding* serta *max pool layer* pada *stride* 2. VGG-16 dibuat untuk dapat menciptakan sebuah model yang merepresentasikan suatu susunan berdasarkan layer konvolusi dan *layer pooling*. VGG-16 memiliki parameter yang sangat banyak yang menyebabkan performanya sangat lambat dibandingkan dengan AlexNet.

IV. METODE

Di sini, peneliti akan menjabarkan hal-hal penting mengenai model *machine learning* yang berperan dalam pemrosesan gambar. Model-model yang ada seperti YOLO v1, YOLO 9000, YOLO v3, dan YOLO v4 akan dijabarkan karena dinyatakan bahwa YOLO merupakan model *machine learning* dengan performa tercepat saat penelitian ini dibuat, serta penulis akan berusaha menjelaskan dan membandingkan model YOLO dengan tiap-tiap versinya sesuai dengan literatur-literatur yang ada saat ini. Dataset yang digunakan yaitu COCO dan ImageNet yang disediakan untuk publik yang umum digunakan untuk penelitian dan academia.

TABEL I
PERBANDINGAN YOLOv3 DENGAN YOLOv4

Method	Backbone	Size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4	CSPDarknet-53	416	38 (M)	41,20%	62,80%	44,30%	20,40%	44,40%	56,00%
	CSPDarknet-53	512	31 (M)	43,00%	64,90%	46,50%	24,30%	46,10%	55,20%
	CSPDarknet-53	608	23 (M)	43,50%	65,70%	47,30%	26,70%	46,70%	53,30%
YOLOV3	Darknet-53	320	45 (M)	28,20%	51,50%	29,70%	11,90%	30,60%	43,40%
	Darknet-53	416	35 (M)	31,00%	55,30%	32,30%	15,20%	33,20%	42,80%
	Darknet-53	608	20 (M)	33,00%	57,90%	34,40%	18,30%	35,40%	41,90%
YOLOv3-SPP	Darknet-53	608	20 (M)	36,20%	60,60%	38,20%	20,60%	37,40%	46,10%

III. HASIL DAN PEMBAHASAN

Saat ini, para peneliti sedang berlomba-lomba untuk menemukan serta mengembangkan model algoritma pengenalan dan deteksi objek yang cepat dan akurat dikarenakan dampak dari semakin berkembangnya teknologi dan semakin banyaknya kebutuhan manusia. Model-model pengenalan dan deteksi objek seperti CNN (*Convolutional Neural Networks*), R-CNN (*Regional* CNN), *Fast* R-CNN, dan *Faster* R-CNN, serta DPM (*Deformable Part Model*) memiliki kompleksitas yang cukup tinggi dan biaya penggunaan yang tidak murah. Namun, model-model ini memiliki akurasi yang cukup tinggi walaupun sangat kompleks.

Akhirnya pada 2015, J. Redmon, dkk., mulai memperkenalkan model algoritma pemrosesan gambar baru bernama 'YOLO' (*You only look once*) dalam papernya. YOLO v1 ini ditujukan untuk dapat memproses gambar dengan cepat dan memiliki akurasi yang tinggi. Redmon, dkk., berusaha menggabungkan proses-proses identifikasi gambar yang pada model sebelumnya (misal pada R-CNN) dilakukan secara terpisah. Proses-proses terpisah ini dirasa sangat berpengaruh dalam kecepatan dan performa dari model algoritma tersebut. Akhirnya, Redmon, dkk., dengan menggunakan *single neural networks* berhasil menggabungkan proses-proses tersebut menjadi satu kesatuan proses. Hasil dari penggabungan proses ini

ternyata menghasilkan model algoritma yang sangat cepat melebihi model-model yang ada saat itu. Namun, kecepatan pemrosesannya ini belum sebanding dengan akurasinya karena saat itu YOLO v1 hanya mampu memiliki akurasi sebesar 88% pada top-5 *accuracy* dengan lama waktu pelatihan sekitar kurang lebih satu minggu.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]$$
$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$$
$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}(C_i-\hat{C}_i)^2$$
$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}(C_i-\hat{C}_i)^2$$
$$+\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\,\epsilon\,classes}(p_i(c)-\hat{p}_i(c))^2 \quad (3)$$

Pada proses pelatihannya, YOLO v1 fokus pada penggunaan *multi-part loss function* (dirumuskan seperti rumus di atas) yang dapat merespon langsung terhadap performa dan keseluruhan model deteksi secara bersamaan. Selain itu, YOLO v1 juga mengalami kesulitan dalam mengenali objek-objek kecil yang berdekatan yang diakibatkan oleh adanya *spatial constraints* pada prediksi *bounding box*-nya. YOLO v1 juga mengalami kesulitan dalam mengenali objek-objek baru dari gambar yang memiliki konfigurasi maupun aspek rasio yang berbeda karena YOLO v1 hanya mempelajari prediksinya dari data-data yang diberikan saja.

Untuk memperbaiki batasan-batasan tersebut, Redmon, dkk., akhirnya merilis YOLO v2 dan YOLO9000 pada paper keduanya pada tahun 2016. Pada dasarnya YOLO v2 memfokuskan pengembangan pada perbaikan *recall* dan lokalisasi bersamaan dengan mempertahankan akurasi klasifikasinya. Yolo v2 menggunakan kustomisasi jaringan berbasis arsitektur GoogleNet yang memiliki kecepatan yang lebih cepat dari VGG-16, namun dengan akurasi yang lebih rendah. Selain itu, penggunaan algoritma *open-source* Darknet-19 juga mampu meningkatkan kinerja dan performa dari YOLO v2. Darknet-19 memiliki prinsip kerja seperti *Network In Network* (NIN) yang memanfaatkan *global average pooling* untuk melakukan prediksi. Berdasarkan YOLO v2 ini, Redmon, dkk., juga mengajukan model yang lebih kuat yang diberi nama YOLO9000. YOLO9000 menggunakan *Hierarchical View of Object Classification* untuk dapat menyatukan dataset-dataset yang berbeda. Selain itu, YOLO9000 juga menggunakan algoritma pelatihan gabungan yang memungkinkan untuk menggabungkan pelatihan data deteksi dan klasifikasi pada detektor. YOLO9000 menggunakan dataset label dari WordNet yang mampu mengenali 9000 kelas objek. Dikarenakan WordNet berbentuk grafik berarah yang susunannya cukup kompleks, maka untuk menyederhanakannya dibentuklah bentuk pohon hirarki yang diberi nama WordTree yang konsepnya berasal dari ImageNet. Kemudian WordTree diisi dengan dataset gabungan dari COCO dan ImageNet. YOLO9000 mampu mempelajari jenis objek baru namun masih kesulitan untuk dapat mempelajari kategori objek baru.

Selanjutnya, pada tahun 2018, Redmon, dkk., berhasil meluncurkan versi terbaru dari YOLO. Berdasarkan laporan teknologi yang dirilisnya, YOLO v3 memiliki ukuran yang lebih besar dari versi-versi sebelumnya. Namun, YOLO v3 ini masih memiliki performa yang lebih cepat serta akurasi yang lebih baik dibandingkan versi-versi sebelumnya. YOLO v3 menerapkan regresi logistik pada *bounding box*-nya untuk dapat mendeteksi keobjekan lebih baik. Selain itu penggunaan softmax digantikan oleh *Independent Logistic Classifier* karena softmax dinyatakan tidak berpengaruh langsung terhadap performa. Selain itu, *binary cross-entropy loss* juga digunakan saat *training* untuk memprediksi *class object*. Dibanding dengan Darknet-19, Darknet-53 digunakan pada versi ini dikarenakan memiliki ukuran tertinggi pada *floating point operation* per detiknya yang berarti dapat memanfaatkan GPU lebih baik sehingga lebih efisien dan lebih cepat. Namun, dibalik performa yang lebih tinggi dari versi sebelumnya ini, YOLO v3 masih mengalami beberapa kesulitan. Satu diantaranya yaitu kesulitan dalam mengenali objek-objek berukuran medium dan besar. Selain itu, YOLO v3 juga sulit untuk dapat mensejajarkan *bounding box* dengan objek pada gambarnya. Oleh karena itu, walaupun YOLO v3 memiliki kecepatan tertinggi daripada versi-versi sebelumnya, namun YOLO v3 lebih disarankan untuk dijalankan pada matriks deteksi lama dengan 0,5 IOU.

YOLO v4 selanjutnya dikembangkan oleh Alexey Bochkovskiy, dkk., pada tahun 2020. Alexey Bochkovskiy melanjutkan pengembangan YOLO karena Redmon menghentikan penelitiannya di bidang *Computer Vision* (CV) karena mengetahui bahwa karya ilmiahnya dapat disalahgunakan oleh pihak-pihak tertentu. YOLO v4 lebih menekankan pada pengoptimisasian komputasi paralel dan kecepatan operasi. YOLO v4 ditujukan untuk dapat menggunakan GPU (*Graphical Processing Unit*) konvensional tunggal tertentu (misal NVIDIA GeForce GTX 1080 Ti dan RTX 2080 Ti). Hal yang berbeda dari versi sebelumnya yaitu YOLO v4 menggunakan teknik pengembangan baru yang dinamakan *Bag-of-Freebies* (BoF) dan *Bag-of-Special* (BoS) pada tahap *training*-nya dengan tujuan untuk dapat meningkatkan performa model serta akurasi tanpa mempengaruhi lama waktu pemrosesan saat *training*.

Oleh karena itu, Alexey, dkk., berhasil memodifikasi YOLO v4 sehingga tidak lagi memerlukan GPU mahal untuk dapat melakukan pemrosesan pengenalan objek lagi. Alexey, dkk., menambahkan beberapa metode yang dimodifikasi untuk melengkapi kemampuan YOLO v4, seperti PAN (*Path Aggregation Network)*, CBN (*Cross mini-Batch Normalization*), SAM (*Spatial Attention Module*), dan lain sebagainya. Saat jurnal penelitian ini disusun, YOLO v5 sudah diperkenalkan oleh Glenn Jocher, dkk., yang dirilis beberapa hari setelah YOLO v4

diperkenalkan. YOLO v5 masih belum memiliki sumber karya ilmiah yang memadai sehingga cukup sulit untuk mencari informasi yang relevan. Glenn, dkk., mempublikasikan YOLO v5 pada *platform* Github. YOLO v5 menggunakan *framework* PyTorch dalam berbagai versi, yaitu *small*, *medium*, *large*, dan *Xlarge* yang masing-masing memiliki spesifikasi dan kegunaannya masing-masing.

V. PENUTUP

Tujuan awal dari model YOLO yaitu untuk mendesain suatu model algoritma yang mampu mengenali dan mendeteksi objek dengan cepat tanpa mengurangi akurasinya. Hal ini mampu diwujudkan dan dikembangkan dari waktu ke waktu dengan berbagai kekurangan di tiap-tiap versinya. Namun seiring dengan perkembangannya, kekurangan dari tiap-tiap versi model YOLO ini berhasil digarap dan ditingkatkan ke taraf yang lebih baik. Selain itu, keberhasilan dalam menerapkan model YOLO pada GPU konvensional juga memberikan keuntungan yang besar bagi berbagai pihak. Namun, tidak semua GPU konvensional mampu digunakan dalam menjalankan model YOLO ini. Diharapkan penelitian selanjutnya mampu mengembangkan penelitian yang dapat diterapkan pada berbagai GPU konvensional atau bahkan ke tingkat komputer papan tunggal seperti Raspberry Pi atau semacamnya.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Alphabet Incubator yang telah membantu, mendukung, dan memfasilitasi serta membimbing hingga terbentuklah jurnal ilmiah ini. Terima kasih ditujukan kepada Universitas Raharja yang telah membantu mendukung pembuatan jurnal ilmiah ini.

REFERENSI

[1] D. A. Prabowo and D. Abdullah, “Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking,” *Pseudocode*, vol. 5, no. 2, pp. 85–91, 2018, doi: 10.33369/pseudocode.5.2.85-91.

[2] “FACE RECOGNITION MENGGUNAKAN METODE ALGORITMA VIOLA JONES DALAM PENERAPAN COMPUTER VISION | Jurnal Processor.” http://ejournal.stikom-db.ac.id/index.php/processor/article/view/120 (accessed May 31, 2021).

[3] R. D. Kusumanto, W. S. Pambudi, and A. N. Tompunu, “Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang,” 2012.

[4] A. K. Syarifuddin Fakultas Tarbiyah IAIN Raden Fatah Palembang Jl Zainal Abidin Fikri No, “BELAJAR DAN FAKTOR-FAKTOR YANG MEMPENGARUHINYA,” 2011. doi: 10.19109/TD.V16I01.57.

[5] Y. Eka Achyani STMIK Nusa Mandiri Jakarta, “Penerapan Metode Particle Swarm Optimization Pada Optimasi Prediksi Pemasaran Langsung,” *J. Inform.*, vol. 5, no. 1, pp. 1–11, Apr. 2018, Accessed: May 31, 2021. [Online]. Available: https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/2736.

[6] A. Wanto, “Penerapan Jaringan Saraf Tiruan Dalam Memprediksi Jumlah Kemiskinan Pada Kabupaten/Kota Di Provinsi Riau,” *Klik - Kumpul. J. Ilmu Komput.*, vol. 5, no. 1, p. 61, 2018, doi: 10.20527/klik.v5i1.129.

[7] D. A. NOVIANA, “ANALISIS KESULITAN SISWA DALAM PEMECAHAN MASALAH MATEMATIKA DITINJAU DARI GAYA KOGNITIF SISWA PADA POKOK BAHASAN TURUNAN FUNGSI KELAS XII TKJ SMK NEGERI TEMAYANG SEMESTER GENAP TAHUN PELAJARAN 2018/2019,” 2019.

[8] J. Falakhi Mawaza, A. Khalil, and I. Negeri Sunan Kalijaga Yogyakarta, “Masalah Sosial dan Kebijakan Publik di Indonesia (Studi Kasus UU ITE No. 19 Tahun 2016),” *J. Gov. Innov.*, vol. 2, no. 1, pp. 657–1714, doi: 10.36636/jogiv.v2i1.386.

[9] “PEMBELAJARAN SEJARAH KESIAPANNYA MENGHADAPI TANTANGAN ZAMAN.” https://repository.unej.ac.id/handle/123456789/83960 (accessed May 31, 2021).

[10] A. A. Nasrulloh, “PENGEMBALIAN FUNGSI BAITUL MAL WA TAMWIL MELALUI STRATEGI PENYELESAIAN MASALAH RENTENIR DI TASIKMALAYA,” *Amwaluna J. Ekon. dan Keuang. Syariah*, vol. 4, no. 1, pp. 75–95, Feb. 2020, doi: 10.29313/amwaluna.v4i1.5271.

[11] N. O’Mahony *et al.*, “Deep Learning vs. Traditional Computer Vision,” *Adv. Intell. Syst. Comput.*, vol. 943, no. April, pp. 128–144, 2020, doi: 10.1007/978-3-030-17795-9_10.

[12] P. R. Sihombing and A. M. Arsani, “COMPARISON OF MACHINE LEARNING METHODS IN CLASSIFYING POVERTY IN INDONESIA IN 2018,” *J. Tek. Inform.*, vol. 2, no. 1, pp. 51–56, Jan. 2021, doi: 10.20884/1.jutif.2021.2.1.52.

[13] “Belajar Dan Pembelajaran - Dina Gasong - Google Books.” https://books.google.co.id/books?hl=en&lr=&id=3rljDwAAQBAJ&oi=fnd&pg=PA162&dq=mampu+untuk+belajar+dan+berkembang+dengan+sendirinya+machine+learning&ots=tlA9Bzkeu1&sig=tf-rWiJjmXlLW_iA_-PRGVWjGP8&redir_esc=y#v=onepage&q&f=false (accessed May 31, 2021).

[14] “TEORI BELAJAR & PEMBELAJARAN - Dr. Yenny Suzana , M.Pd., Imam Jayanto, S.Farm., M.Sc. - Google Books.” https://books.google.co.id/books?hl=en&lr=&id=cyYvEAAAQBAJ&oi=fnd&pg=PP1&dq=mampu+untuk+belajar+dan+berkembang+dengan+sendirinya+machine+learning&ots=zh9CHaMCX5&sig=aiIX2c_uDjJFjhUSkXfcSWrCsp4&redir_esc=y#v=onepage&q&f=false (accessed May 31, 2021).

[15] Z. Zhang, K. Liu, F. Gao, X. Li, and G. Wang, “Vision-based vehicle detecting and counting for traffic flow analysis,” in *Proceedings of the International Joint Conference on Neural Networks*, Oct. 2016, vol. 2016-October, pp. 2267–2273, doi: 10.1109/IJCNN.2016.7727480.

[16] J. S. Asri and G. Firmansyah, “Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang,” 2018.

[17] W. Wu, Y. Yin, X. Wang, and D. Xu, “Face detection with different scales based on faster R-CNN,” *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 4017–4028, Nov. 2019, doi: 10.1109/TCYB.2018.2859482.

[18] Z. Zuo, K. Yu, Q. Zhou, X. Wang, and T. Li, “Traffic Signs Detection Based on Faster R-CNN,” in *Proceedings - IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW 2017*, Jul. 2017, pp. 286–288, doi: 10.1109/ICDCSW.2017.34.

[19] J. Li *et al.*, “Facial Expression Recognition with Faster R-CNN,” in *Procedia Computer Science*, Jan. 2017, vol. 107, pp. 135–140, doi: 10.1016/j.procs.2017.03.069.

[20] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[21] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[23] 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.

[24] M. Zlocha, Q. Dou, and B. Glocker, "Improving RetinaNet for CT Lesion Detection with Dense Masks from Weak RECIST Labels," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Oct. 2019, vol. 11769 LNCS, pp. 402–410, doi: 10.1007/978-3-030-32226-7_45.

[25] J. M. Walz *et al.*, "Treated cases of retinopathy of prematurity in Germany: 5-year data from the Retina.net ROP registry," *Ophthalmologe*, vol. 115, no. 6, pp. 476–488, Jun. 2018, doi: 10.1007/s00347-018-0701-5.

[26] R. Budiarto, "Kinerja Algoritme Pengenalan Wajah untuk Sistem Penguncian Pintu Otomatis Menggunakan Raspberry-Pi," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, p. 80, Jan. 2018, doi: 10.23917/khif.v3i2.5160.

[27] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery," *Remote Sens.*, vol. 11, no. 5, p. 531, Mar. 2019, doi: 10.3390/rs11050531.

[28] I. W. S. E. Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," 2016.

[29] "Komparasi Algoritma Klasifikasi Machine Learning dan Feature Selection pada Analisis Sentimen Review Film - Neliti." https://www.neliti.com/publications/243750/komparasi-algoritma-klasifikasi-machine-learning-dan-feature-selection-pada-anal (accessed May 31, 2021).

[30] G. Garraux *et al.*, "Multiclass classification of FDG PET scans for the distinction between Parkinson's disease and atypical parkinsonian syndromes," *NeuroImage Clin.*, vol. 2, no. 1, pp. 883–893, Jan. 2013, doi: 10.1016/j.nicl.2013.06.004.

[31] T. Takenouchi and S. Ishii, "Binary classifiers ensemble based on Bregman divergence for multi-class classification," *Neurocomputing*, vol. 273, pp. 424–434, Jan. 2018, doi: 10.1016/j.neucom.2017.08.004.

[32] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *Proceedings of the International Joint Conference on Neural Networks*, 2011, pp. 1453–1460, doi: 10.1109/IJCNN.2011.6033395.

[33] D. Smith *et al.*, "Behavior classification of cows fitted with motion collars: Decomposing multi-class classification into a set of binary problems," *Comput. Electron. Agric.*, vol. 131, pp. 40–50, Dec. 2016, doi: 10.1016/j.compag.2016.10.006.

[34] H. Afrisal, "Metode Pengenalan Tempat Secara Visual Berbasis Fitur CNN untuk Navigasi Robot di Dalam Gedung," *J. Teknol. dan Sist. Komput.*, vol. 7, no. 2, pp. 47–55, Apr. 2019, doi: 10.14710/jtsiskom.7.2.2019.47-55.

[35] N. Hidayasari, I. Riadi, and Y. Prayudi, "Steganalisis Blind dengan Metode Convolutional Neural Network (CNN) Yedroudj- Net terhadap Tools Steganografi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, p. 787, Aug. 2020, doi: 10.25126/jtiik.2020703326.

[36] 14611135 Tutut Furi Kusumaningrum, "IMPLEMENTASI CONVOLUTION NEURAL NETWORK (CNN) UNTUK KLASIFIKASI JAMUR KONSUMSI DI INDONESIA MENGGUNAKAN KERAS," Universitas Islam Indonesia, May 2018. Accessed: May 31, 2021. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/7781.

[37] Q. Aini, N. Lutfiani, N. P. L. Santoso, S. Sulistiawati, and E. Astriyani, "Blockchain For Education Purpose: Essential Topology," *Aptisi Trans. Manag.*, vol. 5, no. 2, pp. 112–120, 2021, doi: 10.33050/atm.v5i2.1506.

[38] N. Lutfiani, P. Harahap, Q. Aini, A. Dimas, A. R. Ahmad, and U. Rahardja, "InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan Attribution-NonCommercial 4.0 International. Some rights reserved Inovasi Manajemen Proyek I-Learning Menggunakan Metode Agile Scrumban," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 96–101, Sep. 2020, doi: 10.30743/infotekjar.v5i1.2848.

[39] U. Rahardja, N. Lutfiani, A. Setiani Rafika, and E. Purnama Harahap, "Determinants of Lecturer Performance to Enhance Accreditation in Higher Education," Oct. 2020, doi: 10.1109/CITSM50537.2020.9268871.

[40] T. Nurhaeni, N. Lutfiani, A. Singh, W. Febriani, and M. Hardini, "International Journal of Cyber and IT Service Management (IJCITSM) p-ISSN: ******* Vol. 1 No. April 2021 e-ISSN: ******* The Value of Technological Developments Based on An Islamic Perspective International Journal of Cyber and IT Service Management (IJCITSM) p-ISSN: ******* Vol. 1 No. April 2021 e-ISSN: *******," Apr. 2021. Accessed: May 31, 2021. [Online]. Available: https://pandawan.aptisi.or.id/index.php/att/article/view/59.

[41] U. Rahardja, A. N. Hidayanto, N. Lutfiani, D. A. Febiani, and Q. Aini, "Immutability of Distributed Hash Model on Blockchain Node Storage," *Sci. J. Informatics*, vol. 8, no. 1, pp. 137–143, May 2021, doi: 10.15294/sji.v8i1.29444.

[42] E. Guustaaf, U. Rahardja, Q. Aini, H. W. Maharani, and N. A. Santoso, "Blockchain-based Education Project," *Aptisi Trans. Manag.*, vol. 5, no. 1, pp. 46–61, 2021, doi: 10.33050/atm.v5i1.1433.

[43] "TMJ (Technomedia Journal) Vol. 4 No.2 Februari 2020 - TMJ (Technomedia Journal), Dr. Ir. Untung Rahardja, M.T.I., MM - Google Books." https://books.google.co.id/books?hl=en&lr=&id=qUMZEAAAQBAJ&oi=fnd&pg=PA223&ots=3tR8yqm6AT&sig=JdOh4cHeZWh17gDRdHgfpCfkOLo&redir_esc=y#v=onepage&q&f=false (accessed May 31, 2021).

[44] iketut gunawan and M. Suzaki Zahran, "International Journal of Cyber and IT Service Management (IJCITSM) p-ISSN: ******* Vol. 1 No. April 2021 e-ISSN: ******* Blockchain Technology: Can Data Security Change Higher Education Much Better?," May 2021. Accessed: May 31, 2021. [Online]. Available: https://pandawan.aptisi.or.id/index.php/att/article/view/59.