



**KLASIFIKASI DAN DETEKSI ELEMEN PADA SKETSA MOCKUP
WEBSITE DENGAN CONVOLUTIONAL NEURAL NETWORK**

TUGAS AKHIR

Iwan Darmawan
41515120075

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA**

**JAKARTA
2021**



**KLASIFIKASI DAN DETEKSI ELEMEN PADA SKETSA MOCKUP
WEBSITE DENGAN CONVOLUTIONAL NEURAL NETWORK**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Iwan Darmawan
41515120075

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA

2021
MERCU BUANA

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41515120075

Nama : Iwan Darmawan

Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 13 November 2020



Iwan Darmawan

UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Iwan Darmawan
NIM : 41515120075
Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA SKETSA MOCKUP WEBSITE DENGAN CONVOLUTIONAL NEURAL NETWORK

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 20 Desember 2020



Iwan Darmawan

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Iwan Darmawan
 NIM : 41515120075
 Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA SKETSA MOCKUP WEBSITE DENGAN CONVOLUTIONAL NEURAL NETWORK

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi		Diajukan	v
		Jurnal Nasional Terakreditasi	v		
		Jurnal International Tidak Bereputasi		Diterima	
		Jurnal International Bereputasi			
Disubmit/dipublikasikan di :	Nama Jurnal	: JEPIN (Jurnal Edukasi dan Penelitian Informatika)			
	ISSN	: 2548-9364			
	Link Jurnal	: https://jurnal.untan.ac.id/index.php/jepin/index			
	Link File Jurnal Jika Sudah di Publish	:			

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 20 Desember 2020



Iwan Darmawan

LEMBAR PERSETUJUAN PENGUJI 1

NIM : 41515120075
Nama : Iwan Darmawan
Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 9 Februari 2021



(Sabar Rudiarto, M.Kom)

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI 2

NIM : 41515120075
Nama : Iwan Darmawan
Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 9 Februari 2021



(Wawan Gunawan, S.Kom, MT)

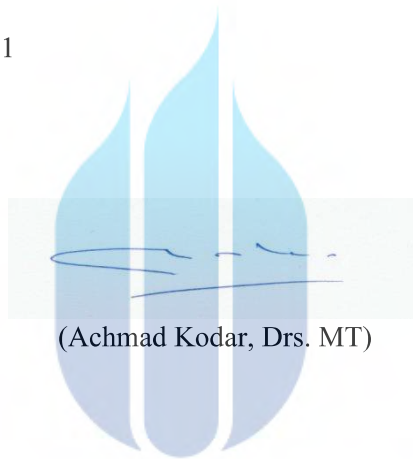
UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI 3

NIM : 41515120075
Nama : Iwan Darmawan
Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 9 Februari 2021



UNIVERSITAS
MERCU BUANA

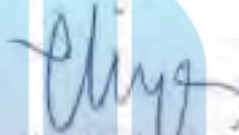
LEMBAR PENGESAHAN

NIM : 41515120075
Nama : Jwan Darmawan
Judul Tugas Akhir : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 23 Februari 2021

Menyetujui,



(Dr. Ir. Eliyani)
Dosen Pembimbing

Mengetahui,



UNIVERSITAS
MERCU BUANA

(Diky Firmus, S.Kom, MM)
Koord. Tugas Akhir Teknik Informatika

(Dest Ramayanti, S.Kom, MT)
Ka. Prodi Teknik Informatika

ABSTRAK

Nama : Iwan Darmawan
NIM : 41515120075
Pembimbing TA : Dr.Ir.Eliyani
Judul : KLASIFIKASI DAN DETEKSI ELEMEN PADA
SKETSA MOCKUP WEBSITE DENGAN
CONVOLUTIONAL NEURAL NETWORK

Abstrak— Pengembangan user-interface pada proses perubahan design berupa mockup website ke bahasa markup yang dilakukan developer membutuhkan pekerjaan yang bersifat manual dan memakan waktu yang tidak sedikit. Sehingga terdapat potensi solusi otomatisasi pada proses tersebut menggunakan pendekatan machine learning, proses otomatisasi tersebut bisa dicapai dengan mendeteksi dan klasifikasi elemen-elemen website yang terdapat pada mockup dengan algoritma YOLO yang memiliki basis Convolutional Neural Network. Penelitian ini memiliki tahapan preprocessing yang terdapat proses pelabelan objek beserta augmentasi data. Proses evaluasi dilakukan menggunakan metric mAP, precision dan recall dengan hasil 0.97 mAP, 0.99 untuk precision dan 0.98 untuk hasil recall, sehingga hasil metrik evaluasi menunjukkan bahwa algoritma YOLO dapat digunakan untuk proses deteksi dan klasifikasi elemen-elemen mockup. Kata kunci:

Machine Learning, CNN, Pengolahan Citra, Wireframe, Mockup, YOLO, Data Augmentation.



UNIVERSITAS
MERCU BUANA

ABSTRACT

Name : Iwan Darmawan
Student Number : 41515120075
Counsellor : Dr.Ir.Eliyani
Title : Classification and Detection

Abstract- The Process of translate from mockup design into markup language in user-interface development by developer need large amount of manually and time consuming tasks. Therefore there are some potential solutions to automatize that process trough machine learning approach. That automatitation process can be reach by detect the website elements on the mockup trough classification and detection method with CNN based algorithm ,YOLO. The proposed system arranged with some steps such as image preprocessing with image labelling and annotating in it and data augmentation. Evaluation of this system composed with mAP wich results 97% ,99% for precision and 98% for recall calculation.

Key words:

Machine Learning, CNN, Image Processing, Wireframe, Mockup ,YOLO, Data Augmentation



KATA PENGANTAR

Puji syukur kita panjatkan kepada Allah S.W.T, karena berkat rahmat-Nya penulis bisa menyelesaikan penelitian ini.

Penulis menyadari bahwa tanpa bantuan dan bimbingan Ibu Dr.Ir.Eliyani dan rekan-rekan yang telah membantu saya dalam menyelesaikan penelitian saya mungkin saya tidak dapat menyelesaikan penelitian ini tepat waktu, Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibu Dr.Eliyani selaku Dosen Pembimbing Tugas Akhir Teknik Informatika.
2. Ibu Desi Ramayanti, S.Kom., MT selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Mercu Buana.
3. Bapak Diky Firdaus, S.Kom, MM selaku Koordinator Tugas Akhir Teknik Informatika.
4. Orang tua yang telah memberikan dukungan doa dan semangat.
5. Teman-teman rekan mahasiswa Universitas Mercubuana yang tidak bisa disebut satu persatu yang turut membantu memberikan support dan dukungan kepada penulis selama proses penelitian ini.

Akhir kata, penulis menyadari terdapat ketidaksempurnaan dan kekurangan dalam penelitian tugas akhir ini. Penulis berharap semoga penelitian tugas akhir ini dapat memberikan manfaat.

UNIVERSITAS
MERCU BUANA

Jakarta, 20 Desember 2020
Penulis

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR... ..	iii
SURAT PERNYATAAN LUARAN TUGAS AKHIR.....	iv
LEMBAR PERSETUJUAN PENGUJI 1	v
LEMBAR PERSETUJUAN PENGUJI 2	vi
LEMBAR PERSETUJUAN PENGUJI 3	vii
LEMBAR PENGESAHAN	viii
ABSTRAK	ix
ABSTRACT	x
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xii
NASKAH JURNAL	1
KERTAS KERJA.....	12
BAB 1. LITERATUR REVIEW	13
BAB 2. ANALISIS DAN PERANCANGAN.....	15
BAB 3. SOURCE CODE	30
BAB 4. DATASET.....	39
BAB 5. TAHAPAN EKSPERIMEN.....	44
BAB 6. HASIL SEMUA EKSPERIMEN.....	52
DAFTAR PUSTAKA	55
LAMPIRAN DOKUMEN HAKI.....	57
LAMPIRAN KORESPONDENSI	60
LAMPIRAN TANDA TERIMA TATA USAHA	61
CURICULLUM VITAE	62

NASKAH JURNAL

Klasifikasi dan Deteksi elemen mockup dengan Convolutional Neural Network

Iwan Darmawan^{#1}, Eliyani^{*2}

[#]Universitas Mercu Buana, Jakarta

Universitas Mercu Buana, Meruya, (021) 5840816

¹41515120075@student.mercubuana.ac.id

¹ elijani@pmmercubuana.ac.id

Abstrak— Pengembangan user-interface pada proses perubahan design berupa mockup website ke bahasa markup yang dilakukan developer membutuhkan pekerjaan yang bersifat manual dan memakan waktu yang tidak sedikit. Sehingga terdapat potensi solusi otomatisasi pada proses tersebut menggunakan pendekatan machine learning, proses otomatisasi tersebut bisa dicapai dengan mendeteksi dan klasifikasi elemen-elemen website yang terdapat pada mockup dengan algoritma YOLO yang memiliki basis Convolutional Neural Network. Penelitian ini memiliki tahapan preprocessing yang terdapat proses pelabelan objek beserta augmentasi data. Proses evaluasi dilakukan menggunakan metric mAP, precision dan recall dengan hasil 0.97 mAP, 0.99 untuk precision dan 0.98 untuk hasil recall, sehingga hasil metrik evaluasi menunjukkan bahwa algoritma YOLO dapat digunakan untuk proses deteksi dan klasifikasi elemen-elemen mockup.

Kata kunci— Machine Learning, CNN, Pengolahan Citra, Wireframe, Mockup, YOLO, Augmentasi Data.

I. PENDAHULUAN

Pengembangan *user interface* untuk aplikasi berbasis *web* biasanya dilakukan secara manual dan memakan banyak waktu. Dari survey yang dilakukan dari lebih 5700 *developer* setiap bulannya mereka mengalokasikan sekitar 13% waktu mereka untuk pengembangan *user interface*[1]. Disamping itu study lain menyatakan bahwa rata-rata sekitar 48% *source code* dalam sebuah aplikasi memiliki kaitan dengan pengembangan *user interface*[2].

Alur kerja yang sering digunakan untuk mengembangkan *user interface* berbasis web adalah pengembangan *user interface* berbasis

mockup. Pada proses ini *graphic designer* membuat gambaran kasar yang merepresentasikan elemen yang kedepannya akan menjadi *design user interface*. Proses ini menghasilkan suatu *output* yang disebut *mockup* atau *wireframe*[3]. Suatu *mockup* atau *wireframe* terdiri dari beberapa komponen atau elemen. Tugas dari *Developer* adalah melakukan konversi *mockup* atau *wireframe* menjadi kode *markup* yang umumnya berformat HTML, pekerjaan ini dilakukan dengan menentukan jenis elemen apa saja yang terdapat pada *mockup* atau *wireframe* beserta posisinya[4].

Selain itu dalam proses pengembangan *user interface* terdapat istilah *reusable component* yaitu suatu *component* yang memiliki code standar dan memiliki fungsi yang spesifik, tujuan dari penggunaan *reusable component* yaitu memudahkan penulisan code pada elemen yang sering digunakan[5].

Sehingga dibutuhkan sebuah system yang bisa melakukan proses automasi *mockup* ke code *markup* yang pada penelitian ini akan melakukan pendekatan dengan Metode *Deep Learning* dengan menggunakan algoritma CNN (*Convolutional Neural Network*) yang diharapkan memiliki *output* hasil klasifikasi dan deteksi elemen atau komponen yang terdapat pada *mockup* yang kedepannya bisa dikembangkan menjadi sistem yang dapat melakukan proses konversi *mockup* menjadi bahasa *markup* yaitu kode HTML.

Terdapat beberapa penelitian terkait yang mendukung untuk menggunakan metode CNN untuk proses klasifikasi dan deteksi citra adalah jurnal berjudul *Generating Reusable Web Components from Mockups*

Universitas Mercu Buana

oleh Mohammad Bajamma dengan yang memiliki hasil nilai precision 94% dan nilai recall 75%[6].

Studi lain yang mendukung adalah jurnal yang berjudul *pix2code: Generating Code from a Graphical User Interface Screenshot* oleh Tony Beltramelli yang berhasil melakukan proses klasifikasi elemen *mockup* atau *wireframe* dengan Metode CNN yang memiliki hasil nilai akurasi sebesar 77%[7].

Proses klasifikasi dan deteksi citra juga akan menggunakan metode You Only Look Once (YOLO). Untuk mendukung adalah jurnal berjudul *Pengembangan Deteksi Citra Mobil untuk mengetahui jumlah tempat parkir menggunakan cuda dan modified YOLO* oleh Sisco Jupiyandi yang berhasil mendeteksi jumlah mobil melalui citra yang didapat dari CCTV dengan akurasi 100%[8].

Untuk Proses *pre-processing* dan penggunaan label dan anotasi terhadap dataset menggunakan *tool* labelImg[9]. Penelitian yang mendukung terhadap proses augmentasi data adalah penelitian yang berjudul *Cycle GAN-Based Data Augmentation For Multi-Organ Detection In CT Images Via Yolo* yang menyatakan bahwa penelitian tersebut mendapatkan perbedaan hasil deteksi yang signifikan lebih baik dibandingkan tanpa menggunakan proses augmentasi data[10].

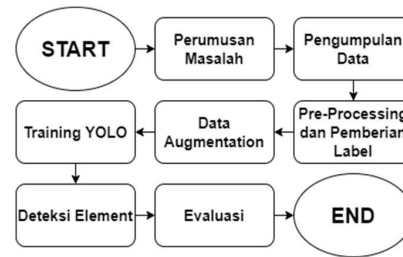
II. METODE PENELITIAN

A. Metodologi Penelitian

Metode penelitian yang digunakan adalah metode penelitian kuantitatif yang dilakukan dengan mengumpulkan data dari studi dokumen. Studi dokumen adalah proses dimana mempelajari dokumen dilakukan dengan tujuan mendapatkan data. Dokumentasi didapatkan dari sebuah penyedia *software* UXPin yang juga menyediakan buku referensi untuk melakukan proses pengembangan *design user interface* yang berjudul *UX Design The Definitive Beginner's Guide*[11].

B. Tahap Penelitian

Tahapan pertama yaitu perumusan masalah secara rinci dan melakukan pembatasan masalah secara rinci, literatur terkait juga dikumpulkan untuk menambah referensi dalam penelitian ini.



Gambar. 1 Tahapan Penelitian

Tahapan berikutnya adalah pengumpulan data menggunakan beberapa elemen website yang termasuk kedalam *reusable component* terdiri dari 2.595 data yang dibagi menjadi 6 kelas. Data gambar didistribusikan untuk menjadi sebanyak pada 70% data training, 20% data validasi, dan 10% data tester. Tahapan *pre-processing* pada penelitian ini melalui tahap *resize* sehingga data input dapat melewati proses training dan pemberian label serta anotasi terhadap setiap input dataset. Sebelum ketahapan training data terlebih dahulu melewati proses augmentasi. Tahapan deteksi elemen merupakan tahapan yang berisi beberapa percobaan pada penelitian ini. Evaluasi akan dipresentasikan dengan tabel dari perhitungan *precision* khususnya mAP, sehingga dari semua tahapan-tahapan diatas mampu menghasilkan sebuah kesimpulan terhadap penelitian ini.

III. PERANCANGAN SISTEM

A. Mockup atau Wireframe

Wireframe atau *Mockup* adalah hasil dari proses kreatif *graphic designer* untuk memulai pengembangan *user interface* yang menjadi representasi visual dari elemen-elemen website dan interaksi antar elemen beserta fungsi dari setiap elemen-elemen tersebut[12].

Pengembangan *user-interface* dikembangkan secara non-digital dan digital. Pembuatan *wireframe* secara digital dikembangkan dengan bantuan software seperti adobe, balsamic dan sebagainya. Sedangkan pengembangan *wireframe* secara non-digital dibuat dengan cara menggambar secara manual pada media seperti kertas atau papan tulis. Pada penelitian ini dataset yang dihasilkan maupun citra yang akan diuji merupakan *wireframe* atau *mockup* yang didapatkan secara manual dengan cara menggambar dan didokumentasikan dengan kamera sehingga menjadi citra yang dapat diolah.

Universitas Mercu Buana

Elemen-elemen yang dipilih menjadi dataset training merupakan elemen yang dikategorikan *reusable component* yang merupakan elemen yang bisa digunakan secara berulang dan memiliki fungsi secara spesifik. Walaupun tidak ada standar yang baku tentang bentuk yang merepresentasikan setiap elemen, pada kasus *reusable elemen* kebanyakan *designer* menggunakan bentuk yang *relative* sama. Bentuk elemen-elemen *mockup* ini dipelajari dari dokumentasi pembuatan *wireframe* yang berjudul *The Guide to Wireframing:UXpin*.

B. Preprocessing

Preprocessing adalah proses memanipulasi data input sehingga citra yang menjadi data input memiliki format yang dibutuhkan untuk proses tahapan training.

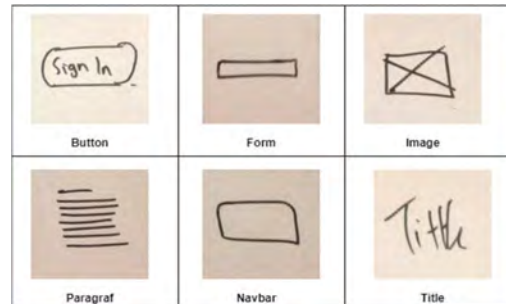
1) Dataset

Pengumpulan data kumpulkan secara manual dengan menggambar 6 elemen *mockup* yang berbeda secara manual. Elemen-elemen yang dipilih sebagai data input adalah sebagai berikut *button*, *form*, *image*, *navbar*, *paragraph* dan *title*. Dataset mentah berjumlah 2.595 gambar terbagi menjadi 6 kelas yang direpresentasikan dengan 6 label yaitu *button* oleh label 0, *form* oleh label 1, *image* oleh label 2, *navbar* oleh label 3, *paragraph* oleh label 4, dan *title* oleh label 5.

TABEL 1
RINCIAN DATASET DAN DISTRIBUSI SETIAP KELAS

Kelas	Jumlah
Button	464
Form	428
Image	433
Navbar	419
Paragraf	440
Title	411

Data train akan dialokasikan sebanyak 70%, sedangkan Data Validasi sebanyak 20%, dan data test sebanyak 10%.



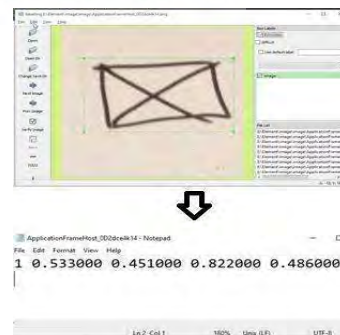
Gambar. 2 Contoh bentuk dari elemen-elemen mockup yang dipilih menjadi dataset

2) Resize

Resize diperlukan agar setiap data input dapat mengikuti karakteristik dari model YOLO. Proses training model YOLO yang melakukan klasifikasi dan deteksi disebut *box* yang disebut *anchor* yang membagi gambar menjadi *grid* 13 x 13 oleh sebab itu data input disesuaikan melalui proses *resize* ke ukuran 416 x 416 terlepas dari berapapun ukuran asli dari data input. Hal ini juga menyebabkan perbedaan rasio yang tinggi citra sebagai data input akan memunculkan distorsi sehingga bentuk objek yang direpresentasikan pada citra yang telah diresize tidak sesuai dengan bentuk objek pada citra yang asli.[13].

3) Labelling

Pemberian label dilakukan untuk menghasilkan *ground truth* pada dataset training yang berformat txt sesuai format darknet YOLO. Pemberian label dilakukan menggunakan *tool* bernama LabelImg.



Gambar. 3 Proses labelling dan informasi yang terdapat pada output dari proses labelling

C. Data Augmentation

Data Augmentation dilakukan untuk melakukan pengayaan variasi terhadap dataset tanpa kehilangan inti atau esensi dari data tersebut. Sehingga diharapkan dapat menghasilkan performa model yang lebih optimal[13].

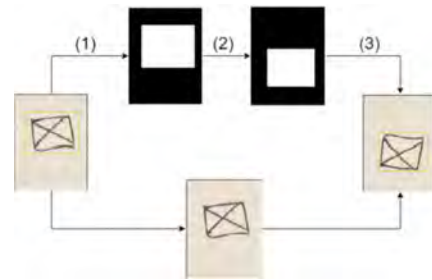
Langkah-langkah yang dipilih pada proses augmentasi data ini menyesuaikan kebutuhan dan jenis dataset, pada penelitian ini Langkah-langkah augmentasi yang dipilih adalah *horizontal flip*, *vertical flip*, dan *average blurring*. Proses ini menggunakan *tool CloDSA* yaitu *tool* yang berfungsi untuk membuat variasi citra yang dibutuhkan untuk *object detection* dan *object localization*[14]. Tidak seperti augmentasi data yang dipersiapkan untuk model klasifikasi citra, augmentasi data pada deteksi objek mengharuskan *ground truth* yang dihasilkan dari proses *labelling* terupdate sesuai dengan objek yang terdapat pada citra hasil augmentasi data, dengan kata lain augmentasi data pada deteksi objek harus menghasilkan transformasi baik untuk gambar maupun label yang dimiliki oleh setiap data input.



Gambar. 4 Transformasi Augmentasi Data yang dilakukan (1)Vertical Flip, (2)Horizontal Flip, (3)Average Bluring

1) Horizontal Flip dan Vertical Flip

Proses transformasi *Horizontal Flip* dan *Vertical flip* merupakan upaya membalikan gambar baik secara horizontal maupun vertikal, tidak hanya gambar yang mengalami *flip augmentation*, label yang dimiliki oleh setiap gambar juga harus mengikuti objek pada gambar hasil dari proses flip augmentation ini.



Gambar. 5 Alur yang terjadi pada proses vertical flip (1) Generate Masking, (2) Transformasi dari label dan transformasi citra. (3) Hasil dari transformasi label dan transformasi citra

2) Average Blurring

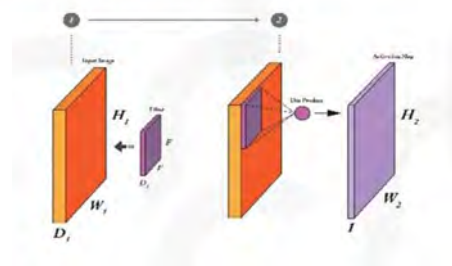
Proses transformasi *Average Bluring* adalah Langkah augmentasi untuk mengantisipasi data citra yang *blur*, karena citra yang *blur* menyebabkan ketidakakuratan *feature abstraction* pada *network training*.

D. Convolutional Neural Network

Searah dengan perkembangan teknologi *Neural Network* sudah berkembang *Neural Network* yang memiliki *multi layer* yang menjadikan semakin banyaknya *hidden layer* yang berakibat pada buruknya akurasi. CNN dikomposisikan terbagi menjadi beberapa set layer yang sesuai fungsinya[15].

1) Convolutional Layer

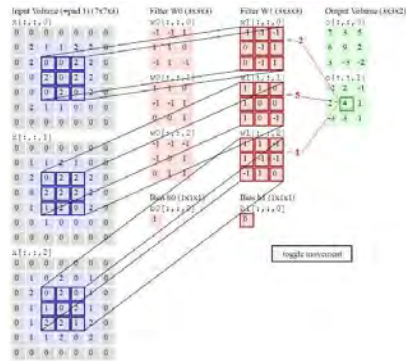
Convolutional Layer adalah layer yang mengekstraksi feature dari gambar input. *Convolutional layer* memanfaatkan fitur yang disebut filter. Filter memiliki ukuran tinggi dan lebar serta ketebalan tertentu. Biasanya filter diinisialisasi dengan nilai random atau dengan teknik tertentu.



Gambar. 6 Ilustrasi Filter pada Convolutional Layer

Pada gambar diatas dihasilkan sebuah angka yang merupakan *dot product* antara bagian gambar dengan filter yang digunakan. Dengan menggeser filter dengan jarak

tertentu yang disebut *stride* yang kemudian menghasilkan *activation map*.



Gambar. 7 Pergeseran filter pada Convolutional Layer

Ukuran spasial dari output proses pada gambar diatas dapat dihitung berdasarkan formula:

$$(N - F + 2P)/S + 1$$

dimana N adalah ukuran spasial filter, P adalah jumlah zero padding dan S adalah nilai stride pada setiap proses komputasi.

2) Stride

Stride adalah jarak pergerakan filter keseluruhan gambar input dengan satuan *pixel*. Ketika *stride* memiliki nilai 1 maka filter bergerak sebanyak 1 *pixel*, ketika *stride* memiliki nilai 2 maka filter juga bergerak sebanyak 2 *pixel* dan seterusnya.

3) Padding

Padding atau *zero padding* dilakukan karena pada beberapa kasus filter tidak sesuai dengan output yang dihasilkan oleh *convolutional layer* maka harus disesuaikan dengan memanipulasi dimensi dari output tersebut melalui penambahan jumlah *pixel* disekeliling gambar input dengan nilai 0.

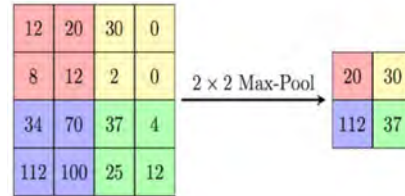
4) Non-Linearity (ReLU)

ReLU merupakan kepanjangan dari Rectified Linear Unit untuk operasi non-linear. Output dari ReLU adalah $f(x) = \max(0, x)$. ReLU bertujuan memanfaatkan operasi non-linear dalam jaringan ConvNet sehingga tidak memiliki nilai negative.

5) Pooling Layer

Pooling layer berfungsi untuk mengurangi dimensi dari *feature map* yang dilakukan dengan operasi down-sampling. Metode *down-sampling* yang digunakan

adalah *max-pooling* yaitu mengambil nilai terbesar pada bagian tersebut.



Gambar. 8 Max-Pool dengan filter 2 x 2

6) Fully Connected Layer

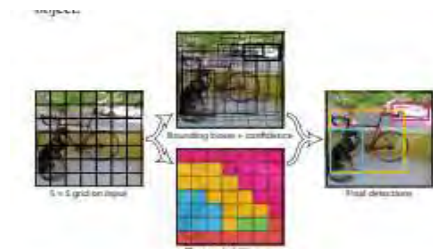
Activation map yang dihasilkan pada proses *feature extraction layer* berbentuk multidimensional array, sehingga dibutuhkan sebuah upaya agar sebuah vector dapat digunakan menjadi input dari *fully-connected layer* melalui *reshape activation map*.

E. You Look Only Once (YOLO)

Saat ini system untuk deteksi citra memanfaatkan klasifikasi untuk sebuah objek dan mengevaluasinya pada lokasi dan skala yang bervariasi. Sistem ini menggunakan *sliding window* dimana proses klasifikasi berjalan dengan rentang jarak sehingga melingkupi keseluruhan gambar.

Pendekatan *region proposal* seperti ini digunakan pada R-CNN untuk menghasilkan *bounding box* dan melakukan klasifikasi pada setiap box tersebut. Hal ini mengakibatkan data *pipeline* menjadi kompleks dan lambat serta sulit dioptimisasi karena setiap komponen harus melewati proses training secara terpisah.

YOLO menawarkan solusi yang lebih sederhana dimana suatu *convolutional network* memprediksi banyak *bounding box* dan probabilitas setiap classnya. YOLO melihat keseluruhan gambar saat training dan pengujian.



Gambar. 9 Model YOLO membagi gambar menjadi grid $s \times s$ dan setiap sel gridnya memprediksi bounding box dan probabilitas kelas.

YOLO membagi data input menjadi $S \times S$ grid, bila titik sentral dari object terdapat pada suatu sel grid, maka sel grid tersebut berfungsi untuk mendeteksi object tersebut. Setiap sel grid memprediksi B bounding box dan *confidence* dari box itu. Nilai dari *confidence* ini merefleksikan bagaimana keyakinan dari model apakah sebuah box itu memiliki objek didalamnya atau tidak dan bagaimana keakuratan model terhadap box tersebut [16].

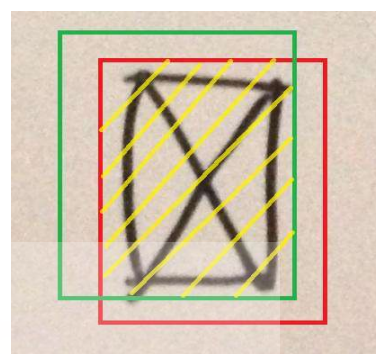
Nilai *confidence* didefinisikan sebagai $Pr(\text{object}) * IOU$. IOU adalah irisan dari union antara *predicted box* dan *ground truth*, bila tidak ada objek didalam sel grid maka nilai *confidence* menjadi nol.

Setiap *bounding box* yang ada terdiri dari 5 jenis prediksi yaitu x, y, w, h dan nilai *confidence*, (x, y) : adalah *coordinate* yang merepresentasikan titik sentral dari box, coordinate ini dikalkulasikan dengan ikatan dari sel grid, (w, h) adalah lebar dan tinggi dari *bounding box*.

Setiap sel grid juga memprediksi nilai C yaitu kondisi probabilitas kelas yang didefinisikan sebagai $Pr(\text{Class}|\text{Object})$ yang memprediksi probabilitas yang dimiliki setiap kelas per satuan sel grid, terlepas dari seberapa banyak box B . Selama masa pengujian kondisi probabilitas kelas ini dikalikan dengan nilai *confidence* setiap box prediction yang dimana memberikan nilai *confidence* kelas yang spesifik pada setiap box. Nilai ini menunjukkan probabilitas dari kelas dan bagaimana box mengikat objek yang didefinisikan dengan $Pr(\text{Class}|\text{Object}) * Pr(\text{Object}) * IOU = Pr(\text{Class}) * IOU$.

1) Intersection Over Union

IoU adalah suatu metric yang biasa digunakan untuk mengevaluasi algoritma object detection yang secara sederhana adalah irisan dari *ground truth* dan *predicted bounding box*. IOU mengkalkulasi kemiripan *predicted box* dengan *ground truth*. IOU didefinisikan sebagai $IoU = \frac{\text{Area Irisan}}{\text{Area Union}}$.



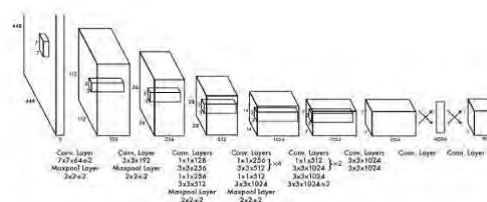
Gambar. 10 Kotak merah menunjukkan Ground truth, Kotak hijau adalah prediction box dan wilayah arsis kuning adalah irisan dari keduanya

2) Non-Max Supression

Pada beberapa kasus ada kemungkinan ditemukannya lebih dari satu deteksi pada objek yang sama. Teknik *Non-Max Supression* adalah teknik dimana algoritma hanya memilih satu deteksi. Untuk menghapus duplikasi prediksi akan dipilih prediksi bounding box dengan probabilitas tertinggi dan menjadikan keluaran tersebut menjadi prediksi akhir, lalu mengeliminasi bounding box yang memiliki nilai $IoU > 0.5$ ataupun nilai *threshold* lain yang dikehendaki.

3) Arsitektur Jaringan

Model dasar dari YOLO memiliki 24 *convolutional layer* yang diikuti *fully connected layer* yang menggunakan reduksi 1×1 diikuti dengan 3×3 *convolutional Layer*.



Gambar. 11 Arsitektur dasar jaringan YOLO

F. YOLO V3

YOLO V3 merupakan improvisasi dari versi jaringan deteksi YOLO sebelumnya,

Universitas Mercu Buana

versi ini memiliki fitur *multi-scale detection*, jaringan *feature extraction* yang lebih kuat, dan beberapa perubahan pada loss function. YOLO versi ini menyediakan *inference* yang lebih cepat dan bersifat *real-time*[17].

1) Arsitektur Jaringan



Gambar. 12 Diagram high level komponen-komponen YOLO V3

Secara sederhana system dari YOLO v3 dibagi menjadi dua komponen yaitu *Feature Extractor* dan *Detector* kedua komponen memiliki fitur multi-scale. Ketika data input masuk melalui *feature extractor* maka menghasilkan fitur-fitur yang melekat pada gambar input pada 3 (atau lebih) skala yang berbeda. Fitur yang telah dihasilkan dari komponen *Feature Extractor* diproses menjadi suatu inputan ke proses *detector* untuk menghasilkan *bounding box* dan informasi class.

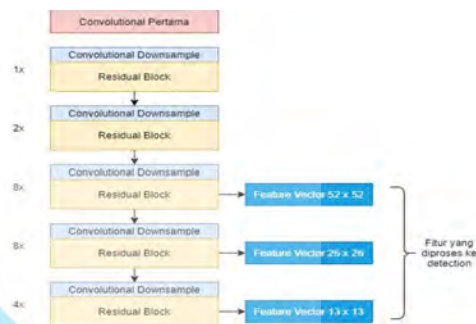
2) Darknet-53

Darknet-53 adalah jaringan yang digunakan pada *Feature Extractor* yang digunakan pada YOLO V3. Seperti namanya Darknet-53 memiliki 53 layer convolutional dibandingkan dengan versi Darknet yang dimiliki YOLO V1.

Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
Convolutional	32	1 × 1	
Convolutional	64	3 × 3	
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
Convolutional	64	1 × 1	
Convolutional	128	3 × 3	
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
Convolutional	128	1 × 1	
Convolutional	256	3 × 3	
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
Convolutional	256	1 × 1	
Convolutional	512	3 × 3	
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
Convolutional	512	1 × 1	
Convolutional	1024	3 × 3	
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

Gambar. 13 Arsitektur YOLO V3

Berdasarkan gambar diagram Darknet-53 diatas terdapat 5 block disebut *residual block*. Setiap *residual block* dihubungkan dengan layer convolutional yang memiliki 2 stride untuk mereduksi dimensi. Pada setiap residual block memiliki struktur *bottleneck* (1x1 dan diikuti 3x3). *Multi-scale detector* terdapat pada 3 residual block terakhir. Tiga skala yang digunakan adalah 52x52, 26x26, dan 13x13 lalu input diasumsikan 416x416.



Gambar. 14 Tiga skala feature vector yang terdapat pada 3 residual block terakhir.

3) Multi-Scale Detector

Feature vector dengan skala yang berbeda-beda akan melewati *detector*. *Detector* ini memiliki struktur yang terdiri atas beberapa *convolutional layer* 1x1 dan 3x3 yang digunakan sebelum *convolutional layer* 1x1 terakhir. Untuk skala medium dan kecil juga menggabungkan *feature* pada skala sebelumnya.

4) Anchor Box

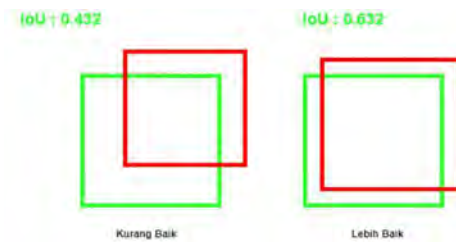
Bounding box biasanya direpresentasikan dengan nilai-nilai yang dinormalisasi yang berformat Xmin, Ymin, Xmax, Ymax. Contohnya jika Xmin adalah 0.5 dan Ymin juga 0.5 maka titik kiri atas berada ditengah-tengah gambar.

IV. HASIL DAN PEMBAHASAN

A. Parameter Kinerja

Proses evaluasi performance dari model *object detection* dilakukan dengan perhitungan yang disebut *mean Average Precision* khususnya evaluasi yang digunakan untuk model YOLO. *Mean average precision* adalah nilai rata-rata AP. Dalam beberapa konteks perhitungan AP untuk setiap kelas label dan mencari nilai rata-ratanya. Untuk melakukan perhitungan mAP maka nilai dari *threshold* untuk memastikan model *object detection* valid atau tidak, yang dalam hal ini nilai *threshold* yang digunakan adalah nilai dari IoU.

Universitas Mercu Buana



Gambar. 15 hasil perhitungan IoU

Dari gambar diatas dapat disimpulkan bahwa area *overlap* atau irisan dari box *predicted* dan box *ground truth* merupakan nilai yang digunakan untuk mengetahui IoU. Semakin besar area irisan kedua box tersebut maka berbanding lurus dengan seberapa dekatnya prediksi yang dihasilkan dengan ground truth.

Asumsi jika nilai IoU adalah 0,5 yang dimana nilai tersebut umum digunakan dan merupakan nilai IoU default dari YOLO untuk melakukan perhitungan mAP maka akan dihasilkan beberapa premis bahwa :

- Jika $\text{IoU} < 0.5$ maka akan diklasifikasikan sebagai *False Positive* (FP).
- Jika IoU memiliki nilai ≥ 0.5 maka hasil dari *object detection* akan diklasifikasikan sebagai *True Positive* (TP).
- Sebuah *ground truth* merepresentasikan objek namun model *object detection* gagal dalam menghasilkan prediksi dari objek tersebut maka akan diklasifikasikan sebagai *False Negative* (FN).
- Sementara *False Negative* (FN) adalah dimana setiap bagian gambar yang tidak memiliki prediksi akan sebuah objek, metrik ini tidak digunakan dalam model *object detection* sehingga metrik ini akan diabaikan.

Precision adalah rasio antara gambar yang terprediksi oleh model dan jumlah total data gambar[19]. Disisi lain *Recall* adalah representasi dari kemampuan model menolak prediksi objek yang tidak relevan terhadap gambar sebenarnya[18]. *Precision* dan *Recall* digunakan sebagai metrik untuk mengevaluasi performance menggunakan true positives (TP), false positives (FP) dan false negatives (FN).

Precision

$$= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Setelah itu nilai dari precision dan recall akan diplot pada *Precision Recall* (PR), grafik PR ini secara garis besar akan menurun dikarenakan adanya tarik menarik antara *precision* dan *recall*, menaikkan salah satu akan menurunkan yang lainnya, dan akan membuat plot grafik menjadi *zig-zag* yang akan dibuat *smooth* dengan persamaan matematika yang dimana *precision* (p) dengan *recall* (r) yang selevel akan disamakan dengan nilai precision maksimum pada *recall* yang berikutnya (r') yang dapat didefinisikan dengan persamaan dibawah ini:

$$P_{inter}(r_{n+1}) = \max p(r'); r' \geq r_{n+1}$$

Perhitungan AP menggunakan teknik yang dinamakan *11 interpolation technique* yaitu 11 titik *precision* yang berdasarkan 11 level yang dibagi sama rata antara 0.0, 0.1, 0.2, 0.3 ...0.9, 1.0. Dengan demikian *point interpolation* yang didapat lalu diaplikasikan ke persamaan berikut ini :

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \dots, 1\}} P_{inter}(r)$$

Untuk menemukan nilai mAP dari semua kelas ditemukan dengan persamaan matematika berikut ini:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Dimana N merupakan jumlah kelas dari dataset yang telah dilatih.

B. Training Model

Proses training dilakukan dengan menggunakan *file pre-trained weight* dari YOLO yang telah diunduh yang berformat .weight .

Dataset training yang telah melewati proses augmentasi beserta label *ground truth* berformat .txt yang dimana telah disesuaikan dengan transformasi gambar yang dihasilkan dari proses augmentasi.

YOLO memiliki beberapa *hyperparameter* yang dapat disesuaikan, yaitu *batch-size*, *subdivision* dan *learning rate*, dimana *batch-size* merupakan jumlah gambar yang diproses per

Universitas Mercu Buana

iteration, *subdivision* adalah pembagian dari *batch-size* untuk diproses ke GPU.

Jumlah maksimal dari *batch* atau *iteration* untuk proses training ditentukan dari jumlah kelas, dan setiap kelas membutuhkan setidaknya 2000 *batch* atau *iteration* maka setidaknya dibutuhkan 12000 *batch* atau *iteration* untuk proses training.

C. Evaluasi

Proses Evaluasi akan dilakukan menggunakan file weight YOLO yang telah dihasilkan dan disimpan dalam *directory framework* darknet, setidaknya terdapat 13 *file* berformat *.weight* yang dihasilkan selama 12000 *iteration* training. File *weight* yang dihasilkan tersebut dihasilkan dan disimpan setiap 1000 *iteration* beserta *file weight final* yaitu hasil training dari keseluruhan 12000 *iteration*.

Berdasarkan tabel dibawah terdapat perhitungan mAP setiap 1000 *iteration* sehingga menghasilkan 12 nilai mAP beserta nilai AP dari masing-masing kelas pada setiap 1000 *iteration*, nilai mAP yang tertinggi adalah saat 9000 *iteration* dengan nilai 0.97 dengan nilai *precision* 0.99 dan *recall* 0.98 sehingga model dengan 9000 *iteration* dipilih sebagai model yang paling baik untuk proses klasifikasi dan deteksi.

TABEL III
HASIL EVALUASI PER 1000 ITERATION

Iteration	precision	recall	f1-score	map(0.5)
1000	0.36	0.46	0.4	0.32454
2000	0.94	0.94	0.94	0.899864
3000	0.69	0.7	0.69	0.53407
4000	0.91	0.91	0.91	0.839782
5000	0.82	0.82	0.82	0.690502
6000	0.94	0.93	0.93	0.889343
7000	0.97	0.97	0.97	0.94943
8000	0.95	0.95	0.95	0.917493
9000	0.99	0.98	0.98	0.976096
10000	0.99	0.98	0.98	0.970985
11000	0.94	0.95	0.95	0.923574
12000	0.99	0.98	0.98	0.975207

D. Pengujian

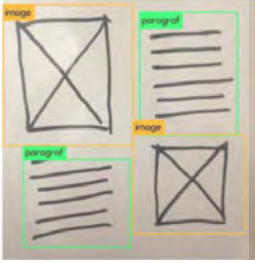
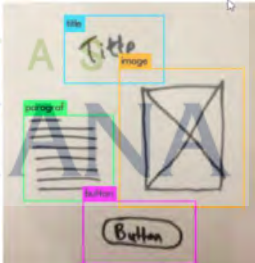
Berdasarkan pengujian nilai mAP pada setiap *iteration* pada proses evaluasi telah diketahui bahwa nilai mAP yang paling

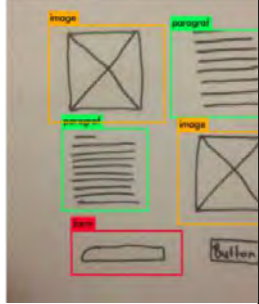
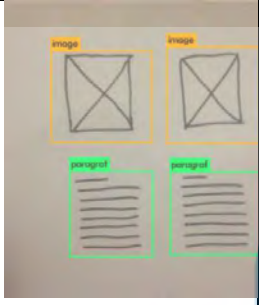
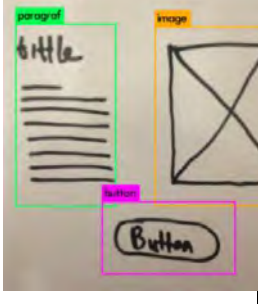
tinggi adalah model dengan weight yang memiliki 9000 *iteration*.

Model tersebut telah tersimpan pada *direktori framework* Darknet untuk digunakan untuk mengimplementasi model terhadap gambar yang telah dipersiapkan di *direktori darknet* untuk pemrosesan *detection*.

Preview pada gambar dibawah ini merupakan hasil *detection* telah berhasil mengklasifikasi objek sehingga menghasilkan informasi kelas dan *bounding box* dengan sebagai representasi posisi dari prediksi yang dihasilkan.

TABEL IIIII
HASIL PERCOBAAN DENGAN GAMBAR TESTER

No	Hasil Deteksi	Confidence Score
1		Image : 75% Paragraf : 48% Image : 78% Paragraf : 55%
2		Paragraf : 61% Title : 55% Button : 73% Image : 71%

3		<p>Image : 68%</p> <p>Paragraph :46%</p> <p>Form : 53%</p> <p>Paragraph : 44%</p> <p>Image : 60%</p>
4		<p>Image : 39%</p> <p>Paragraph : 51%</p> <p>Image : 74%</p> <p>Paragraph : 82%</p>
5		<p>Paragraph : 40%</p> <p>Button : 85%</p> <p>Image : 79%</p>

V. KESIMPULAN

Proses deteksi dan klasifikasi objek yang pada penelitian ini adalah elemen *website* pada *wireframe* pada citra dua dimensi telah mengimplementasikan algoritma YOLO yang berbasis CNN untuk menghasilkan model deteksi objek yang dilatih selama 12000 iteration sudah cukup baik dengan menghasilkan nilai mAP maksimal sebesar 0.97 dengan nilai precision 0.99 dan nilai recall 0.98.

Proses training model ini juga dibantu dengan proses augmentasi data sehingga menambah variasi dari setiap data training yang telah dipersiapkan.

DAFTAR PUSTAKA

- [1] IDC, "Mobile Trends Report,," 2015. <https://www.appcelerator.com/resource-center/research/2015-mobile-trends-report/> (accessed Nov. 02, 2020).
- [2] B. Rosson, "Survey on user interface," *Most*, pp. 195–202, 1992.
- [3] F. K. Ozenc, M. Kim, J. Zimmerman, S. Oney, and B. Myers, "How to support designers in getting hold of the immaterial material of software," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 4, pp. 2513–2522, 2010, doi: 10.1145/1753326.1753707.
- [4] React, "Thinking in React." <https://reactjs.org/docs/thinking-in-react.html> (accessed Nov. 02, 2020).
- [5] A. C. Fulop, "United States Patent (19)," no. 19, 2000.
- [6] M. Bajammal, D. Mazinanian, and A. Mesbah, "Generating reusable web components from mockups," *ASE 2018 - Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng.*, pp. 601–611, 2018, doi: 10.1145/3238147.3238194.
- [7] T. Beltramelli, "pix2code: Generating Code from a Graphical User Interface Screenshot," May 2017, [Online]. Available: <http://arxiv.org/abs/1705.07962>.
- [8] S. Jupiyandi *et al.*, "Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda Dan Modified Yolo Development of Car Image Detection To Find Out the Number of Parking Space Using Cuda and Modified Yolo," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, pp. 413–419, 2019, doi: 10.25126/jtiik.201961275.
- [9] N. Yee, D. Cline, and D. Edgington, "Workflows for Automated Detection and Classification of Unlabeled Deep Sea Imagery," *Monterey Bay Aquarium Res. Inst.*, pp. 1–11, 2017.
- [10] M. Hammami, D. Friboulet, and R. Kechichian, "Cycle GAN-Based Data Augmentation For Multi-Organ Detection In CT Images Via Yolo," pp. 390–393, 2020, doi: 10.1109/icip40778.2020.9191127.
- [11] Priyono, *Metode Penelitian Kuantitatif*. Sidoarjo, 2008.
- [12] C. Bank, "The Guide To Wireframing Call To Action," p. 114, 2016.

Universitas Mercu Buana

- [13] H. J. Jeong, K. S. Park, and Y. G. Ha, "Image Preprocessing for Efficient Training of YOLO Deep Learning Networks," *Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. BigComp 2018*, pp. 635–637, 2018, doi: 10.1109/BigComp.2018.00113.
- [14] Á. Casado-García *et al.*, "Clodsa: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks," *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–14, 2019, doi: 10.1186/s12859-019-2931-1.
- [15] F. Chollet, *Deep Learning with Python*. New York, 2018.
- [16] 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.
- [17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.
- [18] I. Nurhaida, H. Wei, R. A. M. Zen, R. Manurung, and A. M. Arymurthy, "Texture fusion for batik motif retrieval system," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 3174–3187, 2016, doi: 10.11591/ijece.v6i6.12049.
- [19] Umniy Salamah, "Application of Logistic Regression Methods to Retinal Damage Detection on Digital Fundus Images", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN : 2456- 3307, Volume 6, Issue 4, pp.103-109, July-August2020. Available at doi : <https://doi.org/10.32628/CSEIT206217> Journal URL : <http://ijsrseit.com/CSEI>



KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan: literature review, dataset yang digunakan, source code, dan hasil eksperimen secara keseluruhan.



Universitas Mercu Buana