

SKRIPSI

**SISTEM PENGHITUNG JUMLAH POSTLARVA UDANG
VANAME BERBASIS IMAGES MENGGUNAKAN MASK R-
CNN**

Disusun dan diajukan oleh:

MUHAMMAD GHALIB S.M

D121 17 1005



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**



Optimized using
trial version
www.balesio.com

LEMBAR PENGESAHAN SKRIPSI

SISTEM PENGHITUNG JUMLAH POSTLARVA UDANG VANAME BERBASIS IMAGES MENGGUNAKAN MASK R-CNN

Disusun dan diajukan oleh

MUHAMMAD GHALIB S.M
D121 17 1005

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 15 Juli 2024 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

Dr. Ir. Zahir Zainuddin, M.Sc.
NIP 196404271989101002

Dr. Eng. Ir. Muhammad Niswar, S.T., M.IT.
NIP 197309221999031001

Ketua Program Studi,



Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng.
NIP 197507162002121004



PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Muhammad Ghalib S.M
NIM : D121171005
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Sistem Penghitung Jumlah Postlarva Udang Vaname Berbasis Images Menggunakan Mask R-CNN.

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 15 Juli 2024

Menyatakan




Muhammad Ghalib S.M



ABSTRAK

MUHAMMAD GHALIB S.M. *Sistem Penghitung Jumlah Postlarva Udang Vaname Berbasis Images Menggunakan Mask R-CNN.* (dibimbing oleh Zahir Zainuddin dan Muhammad Niswar)

Udang Vaname (*Litopenaues vannamei*), atau sering disebut juga udang putih diperkenalkan secara komersial pada tahun 2001 salah satu jenis udang yang habitat aslinya berasal dari pantai Pasifik Barat Amerika Latin.

Budidaya udang vaname terbagi menjadi dua kegiatan utama: pembenihan dan budidaya. Budidaya udang vaname, terutama di fase pembenihan, telah menjadi kegiatan yang semakin populer karena efisiensi dan adaptabilitas tinggi dari *postlarva* udang. ini mendorong banyak petambak untuk beralih fokus dari aktivitas ke pembenihan.

Penelitian ini bertujuan untuk mempermudah pembudidaya dalam menghitung keakuratan jumlah *postlarva* udang vaname dalam dalam skala kecil pada proses pembenihan saat perpindahan dari tempat pembenihan menuju tambak atau *hatchery*. Penerapan Metode *Masking R-CNN (Region Based Convolutional Neural Networks)* Sebagai sistem deteksi objek yang banyak diharapkan mampu dalam proses deteksi dan menganalisis tumpang tindih objek. Model didapatkan dengan menggunakan *backbone resnet50* menghasilkan akurasi 93%. Pada pengujian 1 Analisis kinerja model menggunakan *Confusion Matrix* menggunakan *dataset_testing* pada pengujian *backbone Resnet-50* memberikan hasil dengan akurasi sebesar 96.04% dengan dan *backbone Resnet-101* memberikan hasil dengan akurasi 97.08%. Pada pengujian 2 dilakukan secara langsung di BPBAP (Balai Perikanan Budidaya Air Payau Takalar) pengujian *backbone Resnet-50* menunjukkan efektivitas Data 1 49 *postlarva*, Data 2 100 *postlarva*, Data 3 150 *postlarva*, Data 4 200 *postlarva*, dan Data 5 250 *postlarva* menghasilkan akurasi 95%, akurasi 93%, akurasi 87%, akurasi 91%, dan akurasi 78.5% menunjukkan bahwa akurasi deteksi cenderung menurun seiring dengan peningkatan jumlah *postlarva* yang perlu dideteksi.

Kata Kunci : Udang Vaname, *postlarva*, *Confusion Matrix*, *backbone Resnet-50*, *backbone Resnet-101*, *Mask R-CNN*,.



ABSTRACT

MUHAMMAD GHALIB S.M. Images-based Vaname Shrimp *Postlarvae* Count System Using *Mask R-CNN*. (supervised by Zahir Zainuddin and Muhammad Niswar)

Vaname shrimp (*Litopenaues vannamei*), also known as white shrimp, was commercially introduced in 2001 as one of the shrimp species whose natural habitat originates from the West Pacific coast of Latin America.

Vaname shrimp cultivation is divided into two main activities: hatching and cultivation. Vaname shrimp cultivation, especially in the hatchery phase, has become an increasingly popular activity due to the high efficiency and adaptability of these shrimp *postlarvae*, prompting many farmers to shift their focus to hatchery activities.

This research aims to facilitate farmers in the process of accuracy in calculating the number of vaname shrimp *postlarvae* on a small scale in the hatchery process when moving to a pond or hatchery. The application of R-CNN (*Region Based Convolutional Neural Networks*) *Masking Method* as an object *detection* system that is expected to be able to detect and analyze overlapping objects. The model obtained using the *resnet50 backbone* produces 93% accuracy. In test 1, analysis of model performance using *Confusion Matrix* using *dataset_testing* on the *Resnet-50 backbone* test gave results with an accuracy of 96.04% and the *Resnet-101 backbone* gave results with an accuracy of 97.08%. In test 2 conducted directly at BPBAP (Takalar Brackish Water Aquaculture Fisheries Center), the *Resnet-50 backbone* test showed the effectiveness of Data 1 49 *postlarvae*, Data 2 100 *postlarvae*, Data 3 150 *postlarvae*, Data 4 200 *postlarvae*, and Data 5 250 *postlarvae* in producing accuracy 95%, 93% accuracy, 87% accuracy, 91% accuracy, and 78.5% accuracy indicate that *detection* accuracy tends to decrease as the number of *postlarvae* that need to be detected increases.

Keywords: Vaname shrimp, *Postlarvae*, *Confusion Matrix*, *Resnet-50 backbone*, *Resnet-101 backbone*, *Mask R-CNN*.



DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
DAFTAR SINGKATAN DAN ARTI SIMBOL	x
DAFTAR LAMPIRAN.....	xi
KATA PENGANTAR	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Udang Vannamei (Litopenaeus Vannamei).....	5
2.2 Citra.....	7
2.3 Pengolahan Citra Digital.....	8
2.3.1 Citra Warna	10
2.3.2 Segmentasi Citra	12
2.4 Kecerdasan Buatan.....	13
2.2 Visi Komputer.....	13
2.4 Deep Learning.....	14
2.6.1 CNN	15
2.6.2 R-CNN	20
2.6.3 Fast R-CNN.....	22
2.6.4 Faster R-CNN	23
2.6.5 Mask R-CNN	26
2.2 Confusion Matrix	30
METODE PENELITIAN.....	33
pan Penelitian.....	33
tu dan Lokasi Penelitian	34



3.3 Instrumen Penelitian	34
3.3.1 Hardware.....	34
3.3.2 Software.....	35
3.4 Teknik Pengambilan Data.....	35
3.5 Perancangan Sistem	36
3.5.1 Input Images	37
3.5.2 Preprocessing.....	38
3.5.3 Implementasi Mask-RCNN	42
3.5.4 Counting	46
3.6 Matriks Pengujian	47
BAB VI HASIL DAN PEMBAHASAN	49
4.1 Hasil Penelitian	49
4.2 Pembahasan.....	57
BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan	63
5.2 Saran	64
DAFTAR PUSTAKA	65
LAMPIRAN.....	67



DAFTAR GAMBAR

Gambar 1 Udang <i>Vannamei</i> (<i>Litopenaeus vannamei</i>) (Siregar 2018).....	6
Gambar 2 Operasi pengolahan citra digital.....	8
Gambar 3 Citra Kontinyu dan Citra Diskrit.....	9
Gambar 4 Citra Digital.....	10
Gambar 5 Warna 8 bit <i>Truecolor</i>	11
Gambar 6 Warna 16 bit <i>Highcolor</i>	12
Gambar 7 Citra warna RGB, Citra <i>grayscale</i> dan Citra biner.....	12
Gambar 8 Segmentasi Citra pada Citra Foto Harimau.....	13
Gambar 9 <i>Deep learning</i>	15
Gambar 10 Arsitektur CNN.....	15
Gambar 11 RGB Matriks.....	16
Gambar 12 <i>Feature Extraction (multiplying input matrix with filters)</i>	17
Gambar 13 <i>Activation Function</i>	17
Gambar 14 <i>Max-pooling</i>	18
Gambar 15 <i>Flattening</i>	18
Gambar 16 Block Diagram CNN.....	19
Gambar 17 <i>Softmax Activation</i>	19
Gambar 18 <i>Flow Chart CNN Algorithm</i>	20
Gambar 19 Arsitektur R-CNN.....	21
Gambar 20 Block Diagram R-CNN.....	21
Gambar 21 Blok Diagram <i>Fast R-CNN</i>	22
Gambar 22 Blok Diagram <i>Faster R-CNN</i>	24
Gambar 23 <i>Region Proposal Network (RPN)</i>	24
Gambar 24 Diagram RPN.....	25
Gambar 25 Struktur Model <i>Faster R-CNN</i>	26
Gambar 26 Struktur Model FCN.....	27
Gambar 27 Ilustrasi teknik <i>computer vision</i> taks.....	28
Gambar 28 Ilustrasi <i>Mask R-CNN</i> (Abdulhah, 2018).....	28
Gambar 29 Arsitektur <i>Mask R-CNN</i> (aws.amazon.com.2021).....	29
Gambar 30 Model <i>Mask R-CNN</i>	30
Gambar 31 Diagram Tahapan Penelitian.....	33
Gambar 32 Penempatan Cahaya dan Tabung Mika.....	35
Gambar 33 Gambaran Umum Sistem.....	37
Gambar 34 <i>postlarva</i> udang vaname.....	38
Gambar 35 Labeling data.....	39
Gambar 36 Augmentasi <i>flip</i>	39
Gambar 37 Augmentasi <i>grayscale</i>	40
Gambar 38 Augmentasi <i>saturation</i>	40
Gambar 39 Augmentasi <i>brightness</i>	40
40 Augmentasi <i>blur</i>	41
41 Augmentasi <i>noise</i>	41
42 Gambar sebelum <i>resize</i> dan setelah <i>resize</i>	42
43 Arsitektur <i>Mask R-CNN</i>	42
44 Implementasi <i>Resnet50</i> dan <i>Resnet101</i>	43



Gambar 45 Ilustrasi <i>Backbone</i>	43
Gambar 46 Proses RPN melakukan <i>refinement</i>	44
Gambar 47 Ilustrasi ROI <i>align</i>	45
Gambar 48 Ilustrasi <i>ROI Classifier & Bounding Box</i>	46
Gambar 49 Ilustrasi <i>segmentation mask</i>	46
Gambar 50 Hasil <i>postlarva</i> setelah dideteksi	49
Gambar 51 Hasil deteksi <i>postlarva</i> dengan nilai <i>confidence</i>	52
Gambar 52 Grafik perbandingan Hasil <i>testing confusion matrix</i>	59



DAFTAR TABEL

Tabel 1. Warna dan nilai penyusunan warna.	11
Tabel 2. <i>Precicion</i>	31
Tabel 3. <i>Recall</i>	31
Tabel 4. <i>F1- Score</i>	31
Tabel 5. Akurasi.	32
Tabel 6. <i>Confussion matrix</i>	47
Tabel 7. Akurasi.	47
Tabel 8. <i>F1- Score</i>	47
Tabel 9. Hasil Pengujian <i>Resnet-50</i> dengan FPN.	50
Tabel 10. Hasil Pengujian <i>Resnet-101</i> dengan FPN.	51
Tabel 11. Hasil uji nilai <i>confidence</i> pada objek <i>detection</i>	52
Tabel 12. Hasil Pengujian Data.....	53
Tabel 13. Hasil Pengujian gambar yang bukan <i>postlarva</i> vaname.	55



DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
pl	<i>postlarva</i>
CNN	<i>Convolutional Neural Network</i>
RGB	<i>Red, Green and Blue</i>
FPN	<i>Feature Pyramid Network</i>
NMS	<i>Non-Maximum Suppression</i>
VGG	<i>Visual Geometry Group</i>
SVM	<i>Support Vector Machine</i>



DAFTAR LAMPIRAN

Lampiran 1. Hasil Pengujian Resnet-50 dengan FPN.....	67
Lampiran 2. Hasil Pengujian Resnet-101 dengan FPN.....	69
Lampiran 3. Source code.....	71



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa, yang telah memberikan rahmat dan karunia-Nya, sehingga dapat menyelesaikan tugas akhir dengan judul “**Sistem Penghitung Jumlah Postlarva Udang Vaname Berbasis Images Menggunakan Mask R-CNN**” sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa penyusunan dan penulisan tugas akhir ini tidak terselesaikan dengan baik tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. Oleh karena itu, penulis dengan senang hati menyampaikan terimah kasih kepada:

1. Kedua orang tua penulis, ayahanda Alm. Drs. Syahrir Mappatakka dan Ibunda Kasmawati S.H yang selalu memotivasi penulis dari kecil hingga berada pada tahap ini dan menyelesaikan penyusunan tugas akhir ini;
2. Bapak Dr. Ir. Zahir Zainuddin, M.Sc. selaku pembimbing I yang senantiasa memberikan saran dan arahan serta bantuan selama proses pengambilan data hingga selesainya sistem ini dibuat, Bapak Dr. Eng. Ir. Muhammad Niswar, S.T., M.IT. selaku pembimbing II yang selalu menyediakan waktu untuk mengingatkan penulis untuk selalu semangat menyelesaikan penulisan penyusunan tugas akhir ini, Selaku Penguji saya ucapkan banyak Terimah Kasih kepada Prof. Dr. Eng. Intan Sari Areni, S.T., M.T. dan Dr. Amil Ahmad Ilham, S.T., M.IT telah memberikan masukan dan saran dengan terus mengingatkan atas penulisan dalam penyusunan tugas akhir ini.
3. Bapak Prof. Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng. selaku ketua departemen teknik informatika Universitas Hasanuddin yang telah membimbing penulis selama masa perkuliahan.
4. Segenap Dosen dan Staf Departemen Teknik Informatika, Fakultas Teknik Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir.



5. Kakak penulis Gamal abd nasser dan Siti meilani adik penulis Rughaya Salsabila yang telah memberikan dukungan kepada penulis untuk tetap semangat dalam menyusun tugas akhir.
6. Penghuni Lab UBICON dan Lab CBS yang senantiasa membantu dan mendengar keluhan penulis selama menyelesaikan tugas akhir ini
7. Teman-teman RECOGNIZER atas dukungan, bantuan, dan semangat yang diberikan selama ini.
8. Para pembudidaya *postlarva* undang di Galesong Kab. Takalar telah menyempatkan waktu dan tenaganya dalam membantu penulis dalam proses pengambilan data.
9. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.
10. Orang-orang berpengaruh lainnya yang tanpa sadar telah memotivasi dan membantu penulis.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karenanya diharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat memberikan sumbangsih dan manfaat besar bagi kepentingan bersama.

Makassar, 15 Juli 2024

Penulis,
Muhammad Ghalib S.M.



BAB I PENDAHULUAN

1.1 Latar Belakang

Litopenaeus vannamei untuk pertama kali diperkenalkan secara komersial pada tahun 2001 dalam perkembangannya induk dan benur yang dikembangkan di *hatchery*. Udang Vaname (*Litopenaeus vannamei*), atau kita sering sebut juga udang putih adalah salah satu jenis udang yang habitat aslinya berasal dari pantai Pasifik Barat Amerika Latin, seperti *Mexico* dan *Puertorico*. Produksi budidaya udang vaname, yaitu *Penaeus vannamei* terus meningkat dan menyumbang lebih dari setengah produksi *Krustasea* dunia (Guenard 2021).

Udang vaname sebagai produk udang utama saat ini sangat mudah untuk ditemukan dibandingkan dengan udang windu. Produksi udang vaname yang terbesar dibudidayakan oleh masyarakat daerah pesisir Galesong Kabupaten Takalar. Udang ini dianggap mampu menggantikan produksi udang windu yang mengalami penurunan produksi. Kurangnya pengetahuan tentang jumlah dan kepadatan udang dalam sistem produksi dapat menyebabkan pemberian pakan yang berlebihan. Oleh karena itu tidak dapat dihindari tingkat pemberian pakan akan berbanding terbalik dengan yang seharusnya. Berdasarkan pengukuran control awal *postlarva* udang (misalnya panjang rata-rata atau berat sampel udang) dapat digunakan untuk perhitungan, agar mencapai tingkat pertumbuhan dan tingkat kelangsungan hidup yang optimal. (Kementerian Kelautan dan Perikanan Direktorat Jenderal Perikanan Budidaya, 2020).

Dalam kegiatan budidaya udang vaname secara garis besar dibagi menjadi dua kegiatan utama yaitu pembenihan dan budidaya. Sebagai produk utama udang saat ini yang dibudidayakan, *postlarva* udang vaname diharapkan dapat beradaptasi dan mempengaruhi lingkungan menjadi lebih baik. Hasilnya banyak petambak dan pembudidaya saat ini beralih ke kegiatan pembenihan. Dengan waktu pembenihan selama 2-3 minggu perkembangan bibit dari telur, *naupli*, *protozoe*, *mysis*, akan mengalami proses perpindahan tempat dari tambak dan dikirim ke daerah.



Kegiatan pembenihan terdapat proses perhitungan benih udang, pada saat akan dijual ke konsumen dan proses perpindahan saat *postlarva* akan siap dibudidayakan pada kolam tarpal. Hingga saat ini penggunaan pendekatan manual telah mengalami beberapa kelemahan seperti tidak akurat, memakan waktu dan melelahkan. Campur tangan manusia dalam proses sekuensial dan perkiraan yang berlebihan dalam perhitungan manual menyebabkan kehilangan bibit benur menjadikannya masalah tersendiri. Tingginya kebutuhan penggunaan alat bantu komputer dalam menghitung *postlarva* diperlukan dalam mengetahui tingkat kelangsungan hidup *postlarva*. Pendekatan pembelajaran mesin telah diusulkan pada tahun 2018 menggunakan *Local Binary Pattern* (LBP) dengan pengklasifikasi acak, menghasilkan akurasi 98,5%. Namun jumlah larva udang yang akan diprediksi yaitu udang windu yang berkisar 64 *postlarva* hingga 150 ekor. Penelitian menggunakan teknik pengolahan citra telah banyak dilakukan dalam penghitungan larva baik itu benih kepiting, benih ikan dan banyak bentuk lainnya dalam bidang kelautan dan perikanan. Namun belum ada penelitian yang terkait pada perhitungan *postlarva* udang vaname menggunakan *Instance Segmentasi* yang dapat membantu dalam perhitungan *postlarva* khususnya pada peningkatan jumlah objek yang padat dan menumpuk untuk segmentasi citra khususnya pada *postlarva* udang vaname.

Setelah berhasil ditenakkan, maka *postlarva* udang vaname disebarkan dan dikembangkan di daerah-daerah terkhusus daerah pesisir Galesong Kabupaten Takalar. Tingginya kebutuhan produksi *postlarva* udang vaname juga berbanding lurus dengan banyaknya waktu yang digunakan dalam proses pengemasan ditambah perhitungan secara manual yang memakan waktu yang dilakukan oleh pembudidaya.

Maka dari itu, Tugas akhir ini menggunakan metode yang telah diusulkan dengan menghitung jumlah *postlarva* udang vaname berbasis *images* menggunakan *Mask R-CNN*. Penelitian ini juga diharapkan dapat membantu dalam mengenalisis *postlarva* yang tumpang tindih saat akan dihitung.



1.2 Rumusan Masalah

Dari Latar belakang yang dijelaskan oleh penulis. Maka rumusan masalah skripsi ini adalah:

1. Bagaimana sistem mendeteksi *postlarva* udang vaname menggunakan *Mask R-CNN*?
2. Bagaimana tingkat akurasi penghitung jumlah *postlarva* udang vaname menggunakan *Mask R-CNN*?

1.3 Tujuan Penelitian

Tujuan penelitian dalam masalah skripsi ini adalah:

1. Mengetahui hasil akurasi bentuk *postlarva* udang vaname menggunakan *Mask R-CNN*.
2. Mengetahui tingkat akurasi perhitungan jumlah *postlarva* udang vaname menggunakan *Mask R-CNN*.

1.4 Manfaat Penelitian

Adapun Manfaat penelitian ini adalah:

1. Manfaat bagi Masyarakat, Sebagai materi informasi teknologi untuk mempermudah dalam mengenali permasalahan perikanan perkembangan teknologi di masa depan.
2. Manfaat bagi pemerintah, penelitian ini diharapkan dapat menjadi sumbangan pemikiran dan pertimbangan dalam penyusunan kebijakan kesejahteraan yang mendukung usaha pembenihan tradisional terkhusus di Kabupaten Takalar.
3. Manfaat bagi pembudidaya udang vaname, semoga dapat mempermudah proses keakuratan dalam menghitung jumlah *postlarva* udang vaname dalam skala kecil.
4. Manfaat bagi peneliti, Sebagai bahan mengembangkan bidang keilmuan dalam elajari metode *Mask R-CNN* dan meningkatkan kemampuan berpikir logis stematika dibidang kelautan dan perikanan.



1.5 Ruang Lingkup

Ruang lingkup dalam penelitian ini adalah:

1. Penelitian ini menggunakan metode *Deep Learning* dengan algoritma *Mask R-CNN*.
2. Penelitian ini mendeteksi *postlarva* vaname lalu di *counting*.
3. Pengambilan data dilakukan menggunakan kamera Fuji Film X-T20 dan kamera HP.
4. Data *training* berupa gambar yang diambil langsung dari pembenihan tradisional dan BPBAP Kab. Takalar.
5. Menganalisis *postlarva* yang saling tumpang tindih
6. Matriks Evaluasi pengukuran performa menggunakan *Confusion Matrix* yang terdiri dari:
 - a. *Precision*
 - b. *Recall*
 - c. *F1-Score*



BAB II

TINJAUN PUSTAKA

2.1 Udang *Vannamei* (*Litopenaeus Vannamei*)

Udang *Vannamei* termasuk dalam *Family Penaidae* yaitu udang laut. Udang *Vannamei* berasal dari perairan Amerika Tengah seperti Venezuela, Brasil, Ekuador, Meksiko dan Panama. Indonesia juga telah lama membudidayakan jenis udang yang dikenal dengan nama *pacific white shrim*. Udang *Vannamei* (*L. Vannamei*) adalah salah satu jenis udang yang memiliki pertumbuhan yang cepat dan memiliki nafsu makan yang tinggi, tetapi ukuran pada saat dewasa lebih kecil dibandingkan dengan udang windu. Habitat aslinya berada di perairan Amerika tetapi spesies ini dapat hidup dan tumbuh diperairan Indonesia dengan baik (Siregar 2018).

Indonesia pertama kali memperkenalkan udang *vannamei* ditahun 2001 pada saat itu produksi udang windu sedang menurun. Udang *vannamei* memiliki beberapa keunggulan di bandingkan dengan spesies lain, inilah yang menyebabkan orang Indonesia memproduksi jenis tersebut, antara lain ketahanan tubuh udang yang lebih baik terhadap serangan berbagai penyakit, kebiasaan hidup di kolam air sehingga udang dapat dibudidayakan dalam kepadatan tinggi, dan juga adaptif terhadap perubahan kondisi lingkungan yang merupakan keunggulan dari udang *vannamei* (Wulandari 2020).

Menurut Boone, 1931 dalam *Marinespecies.org*, Adapun klasifikasi dari udang *vannamei* (*L. vannamei*) sebagai berikut :

Animalia : kingdom

Arthropoda : Phylum

Crustacea : Subphylum

Multicrustacea : Superclass

Malacostraca : Class

Eumalacostraca : Subclass

Eucarida : Superorder

Decapoda : Order



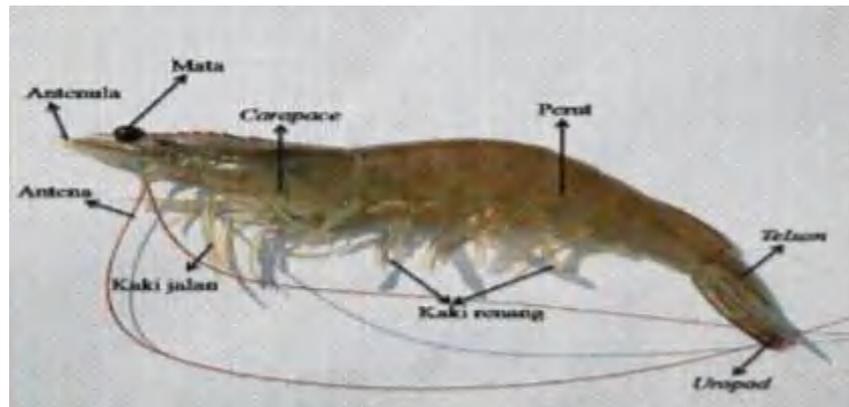
Dendrobranchiata : Suborder

Penaeoidea : Subfamily

Penaidae : Family

Litopenaeus : Genus

L. Vannamei : Spesies



Gambar 1 Udang *Vannamei* (*Litopenaeus vannamei*) (Siregar 2018).

Pada Gambar 1 dalam segi *morfologi*, udang *vannamei* mempunyai tubuh beruas-ruas seperti udang *panaeid* lainnya, dimana pada tiap ruasnya terdapat sepasang anggota badan. Udang *vannamei* termasuk *ordo decapoda* yang dicirikan memiliki sepuluh kaki terdiri dari lima kaki jalan dan lima kaki renang. Tubuh udang *vannamei* dibedakan menjadi dua bagian yaitu *cephalothorax* atau bagian kepala dan dada serta bagian *abdomen* atau perut. Bagian *cephalothorax* terlindung oleh kulit *chitin* yang tebal yang disebut *carapace* secara anatomi *cephalothorax* dan *abdomen* terdiri dari segmen-segmen atau ruas-ruas, dimana masing-masing segmen tersebut memiliki anggota badan yang mempunyai fungsi sendiri-sendiri udang *vannamei* juga memiliki ciri khusus yaitu adanya gigi pada *rostrum* bagian atas dan bawah, serta mempunyai antena panjang (Siregar 2018).

Warna Udang *vannamei* adalah putih transparan dengan warna biru yang terdapat dekat dengan bagian *telson* dan *urupoda* alat kelamin udang jantan disebut *petasma*, yang terletak pada kaki bagian pangkal pertama. Sedangkan alat kelamin

tina disebut *thelcyum*, terbuka dan terletak dibagian antara pangkal kaki 1 dan ke-5. Pada udang jantan dewasa *petasma* adalah simetri, semi *open*, bertudung. Bentuk dari *spermatophore*-nya sangat kompleks, terdiri dari



bagian struktur gumpalan sperma yang *encapsulated* oleh suatu pelindung. Ujung betina dewasa mempunyai *thelycum* terbuka ini merupakan salah satu perbedaan yang paling mencolok pada udang vaname betina (Panjaitan 2012).

Seperti udang dewasa, Pertumbuhan larva udang sangat dipengaruhi oleh temperatur. Larva berkembang menjadi *postlarva* pada temperatur 27-29°C yang prosesnya sekitar sepuluh hari pada kondisi optimal. Di taiwan pada umumnya untuk pemeliharaan larva udang *penaeid optimum* pada suhu 33-35°C. Pada temperatur yang tinggi, perkembangan stadia larva akan berlangsung cepat dan *postlarva* dapat dicapai dalam waktu tujuh hari sejak telur menetas. Larva akan berkembang menjadi beberapa stadia dan setiap stadia akan dibedakan menjadi beberapa substadia sesuai dengan perkembangan morfologinya. Stadia *nauplius* ukuran 0,31-0,33 mm larva stadia ini berbentuk seperti kutu air. Stadia *Protozoa* yang terdiri dari ukuran satu sampai tiga dengan ukuran 1,05-3,30 mm masing-masing akan berkembang dalam selang waktu 2 hari menjadi *mysis*. Setelah 5 hari ukuran larva *mysis* 3,50-4,80 mm larva stadia ini kelihatan lebih dewasa dari dua stadia sebelumnya. Pada stadia *postlarva* sudah aktif bergerak lurus kedepan dan mempunyai sifat cenderung karnivora perubahan dari planktonik menjadi *postlarva* terlihat seperti udang dewasa. Ukuran *postlarva*(pl)1-20 memiliki rentang waktu umur sekitar 25 hari dari *mysis* (Panjaitan 2012).

2.2 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Citra merupakan *output* alat perekaman, seperti kamera yang bersifat *analog* maupun *digital*. Citra *analog* berupa sinyal-sinyal video seperti gambar pada televisi, sedangkan citra digital dapat langsung disimpan pada suatu media penyimpanan (Bagian and Interpretasi 2021).

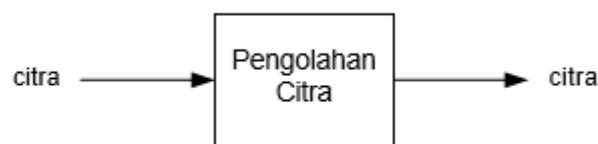
Secara harafiah citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi) yang dihasilkan dari citra *analog* yang *kontinu* dan kemudian dapat diubah diskrit melalui proses *sampling*. *Sampling* adalah pemilihan titik-titik yang mewakili citra digital yang diberikan. Pada umumnya citra digital berbentuk empat persegi panjang dengan lebar dan panjang tertentu. Ukuran citra



biasanya diukur dalam jumlah titik atau piksel, dimana setiap piksel mempunyai koordinat menurut letaknya di dalam citra digital. Koordinat ini biasanya dinyatakan dalam bilangan *integer* yang besarnya dari 0 sampai dengan 1 tergantung dari sistem yang digunakan. Setiap piksel berisi nilai yang mewakili informasi apa yang ada dalam piksel. Citra (*image*) juga dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$ di mana x dan y adalah koordinat spasial dan amplitudo f pada setiap pasang (x,y) disebut intensitas (*gray level*) citra pada titik tersebut. Jika x dan y berhingga (*finite*) dan diskrit (tdk *kontinyu*) maka disebut citra digital. Citra digital terdiri dari sejumlah elemen berhingga yang masing-masing mempunyai lokasi dan nilai, Elemen-elemen x dan y disebut elemen citra atau pels atau piksel (Munir 2013).

2.2 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra / gambar dilakukan secara digital menggunakan *computer*. Teknik-teknik pengolahan citra mentransformasikan citra ke citra yang lain. Jadi masukannya adalah citra dan keluarannya juga citra, namun citra keluaran atau hasil mempunyai kualitas lebih baik dari pada citra masukan (Putra 2010).



Gambar 2 Operasi pengolahan citra digital.

Dalam operasi Pengolahan citra digital bertujuan antara lain :

1. Memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin (dalam hal ini komputer).

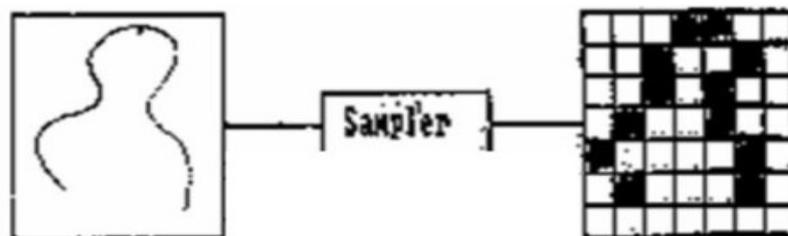


melakukan proses penarikan informasi atau deskripsi obyek atau pengenalan objek yang terkandung pada citra.

3. Melakukan pemeliharaan citra (*feature images*) yang optimal untuk tujuan analisis
4. Melakukan kompresi atau reduksi data yang bertujuan penyimpanan data, transmisi data, dan waktu proses data.
5. Aspek *radiometric* yang terdiri dari peningkatan kontras, restorasi citra, transformasi warna dan aspek *goemetric* terdiri dari rotasi, skala, translasi, transformasi geometric.

Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih banyak dari pada informasi tersebut disajikan dalam bentuk kata-kata (tekstual). Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan *bit* tertentu. Citra yang disimpan dalam memori komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel tersebut (Pratama 2021).

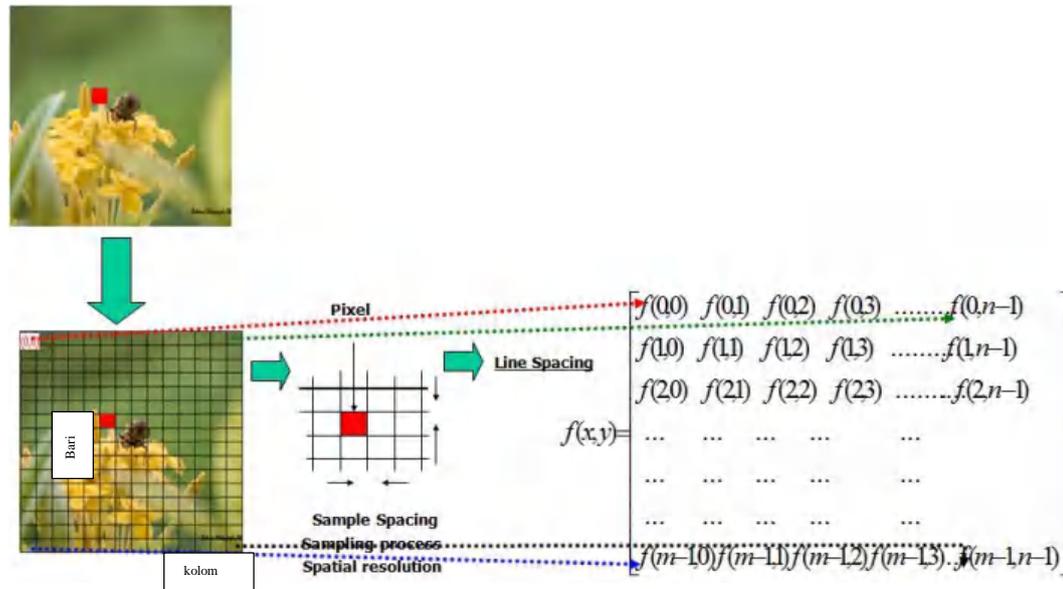
Pada Gambar 2 Sebuah citra digital $a[m, n]$ dapat dijelaskan sebagai ruang diskrit 2D yang berasal dari sebuah citra analog $a(x, y)$ menjadi citra kontinyu 2D melalui proses sampling yang sering disebut digitalisasi. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinyu.



Gambar 3 Citra Kontinyu dan Citra Diskrit.

Contoh hasil ditunjukkan pada Gambar 3 jika citra 2D kontinyu $a(x,y)$ dibagi menjadi N baris dan M kolom, maka perpotongan baris dan kolom disebut piksel.





Gambar 4 Citra Digital.

Citra yang ditunjukkan pada Gambar 4 telah dibagi menjadi $N = 16$ baris dan $M = 16$ kolom. Suatu citra berukuran $N \times M$ *pixel* yang memiliki intensitas beragam pada tiap *pixel*-nya, dipresentasikan secara *numerik* dalam bentuk matriks yang berukuran N_{baris} dan M_{kolom} Nilai yang diberikan untuk setiap piksel (kecerahan rata-rata) dalam piksel dibulatkan ke nilai integer terdekat. Citra digital atau digital *image* merupakan sekumpulan titik yang membentuk matriks berisi informasi warna. Proses ini mewakili amplitudo dari sinyal 2D pada koordinat sebagai nilai integer dengan tingkat berbeda dan biasanya disebut sebagai proses kuantisasi amplitudo atau disebut kuantisasi (Pratama 2021).

2.3.1 Citra Warna

Citra warna atau biasa dinamakan citra RGB atau model warna adalah jenis citra yang menyajikan warna dalam bentuk komponen R(*red*), G(*green*), B(*blue*). Setiap komponen warna menggunakan 8 bit (nilai berkisar antara 0 sampai dengan 255). Pada citra digital, setiap titik atau *pixel* berisi informasi warna yang direpresentasikan dengan model warna tertentu sehingga representasi warna dapat ditampilkan dengan benar di layar komputer. Dengan demikian warna yang dapat mencapai $255 \times 255 \times 255$ atau 16.581.375 warna (Setiawan 2018).

Tabel 1. Warna dan nilai penyusunan warna

Warna	Nilai RGB
-------	-----------



Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum, jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Secara umum jangkauannya adalah 0-255. Menurut (Munir 2013) Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya :

1. Citra *biner* adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner hanya membutuhkan satu bit untuk mewakili nilai setiap piksel dari citra biner.
2. Citra *grayscale* adalah citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *red = green = blue*. Citra *grayscale* memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).
3. Setiap piksel dari citra warna (8 bit) dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Setiap titik (piksel) pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah, hijau, dan biru yang biasa disebut citra RGB (*Red, Green, Blue*). Ada dua jenis citra warna 8 bit, pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Dapat dilihat pada gambar dibawah ini :



Gambar 5 Warna 8 bit *Truecolor*.

4. Citra warna 16 bit biasanya disebut sebagai citra *highcolor*. Setiap pikselnya diwakili dengan 2 byte *memory* (16 bit). Warna 16 bit memiliki 65.536 warna.



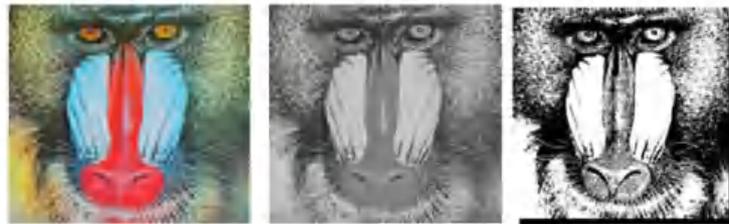
m formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih

sensitive terhadap warna hijau. Gambar dibawah ini menampilkan deret warna yang dihasilkan dari warna 16 bit.



Gambar 6 Warna 16 bit *Highcolor*.

5. Setiap piksel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini tentunya sudah sangat memvisualisasikan seluruh warna yang dapat dilihat oleh penglihatan manusia. Setiap poin informasi piksel (RGB) disimpan ke dalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.



Gambar 7 Citra warna RGB, Citra *grayscale* dan Citra biner.

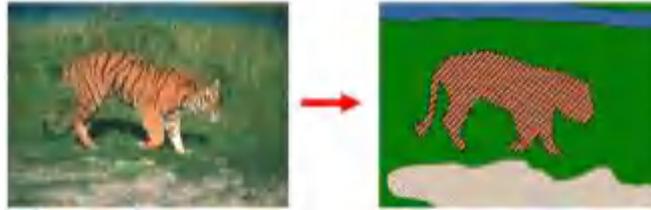
Model warna yang banyak digunakan pada saat ini adalah model warna RGB, CMY, CMYK, YcbCr, HSI, XZY, dan HSV. Masing-masing memiliki kasus penggunaan yang cocok untuk membantu penyelesaian masalah tertentu baik dari sisi penempatan, ukuran, penyimpanan citra, teknik pengenalan citra dan segmentasi citra (Pratama 2021).

2.3.2 Segmentasi Citra

Segmentasi citra untuk menyederhanakan dan mengubah penyajian gambar sesuatu yang lebih bermakna sehingga lebih mudah untuk dianalisis. Ini membagi citra menjadi beberapa bagian berbeda setelah itu menyajikan sebuah citra ke dalam kumpulan *region pixel* yang



direpresentasikan dalam sebuah *mask* atau label. Dengan membagi citra ke dalam segmen-segmen agar pemrosesan dapat dilakukan pada segmen tertentu saja pada Gambar 8 dapat dilihat contoh segmentasi citra pada foto harimau.



Gambar 8 Segmentasi Citra pada Citra Foto Harimau.

Segmentasi citra dilakukan dengan cara mencari kedekatan atau kemiripan antar *pixel*. Pendekatan yang umum digunakan dalam mencari kemiripan antar *pixel* adalah *region growing*, *split and merge*, *clustering*, dan *thresholding*. Teknik segmentasi citra dengan *thresholding* dibagi menjadi 3 yaitu :

1. *Global thresholding*, nilai ambang bergantung pada keseluruhan nilai-nilai *pixel*.
2. *Local thresholding*, nilai ambang berubah secara dinamis bergantung pada *pixel-pixel* segmentasi, hanya sekelompok *pixel* saja.
3. *Adaptive thresholding*, nilai ambang berubah secara dinamis bergantung pada perubahan pencahayaan dalam citra.

2.4 Kecerdasan Buatan

Menurut pengertian yang mengatakan bahwa *Artificial Intelligence* sebuah pembelajaran bagaimana membuat komputer melakukan hal-hal yang dimana saat ini masih lebih baik dilakukan oleh manusia. Hal ini terkait dengan tugas yang sama dengan menggunakan komputer untuk memahami kecerdasan manusia, tetapi kecerdasan buatan tidak harus membatasi dirinya terhadap metode yang diamati secara biologis sebuah ilmu dan teknik pembuatan mesin cerdas khususnya program komputer cerdas. (Cholissodin and Soebroto 2021)

2.4 Visi Komputer



urut Rinaldi Munir dosen Teknik Informatika ITB tahun 2021 *computer* u visi komputer adalah proses meniru cara kerja visual manusia, manusia lihat objek dengan indera penglihatan (mata), lalu citra objek diteruskan

ke otak untuk diinterpretasikan sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra klasifikasi, pengenalan (*recognition*), dan membuat membuat keputusan hasil interpretasi mungkin digunakan untuk pengambilan keputusan.

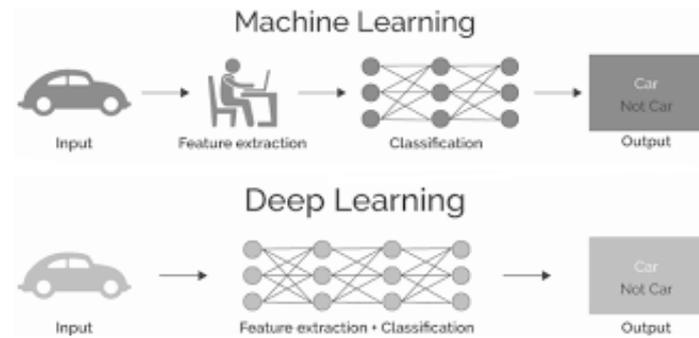
Computer vision menjadikan sebuah komputer “*act like human sight*”, sehingga komputer mempunyai kemampuan mendeteksi manusia dalam mendapatkan informasi secara *visual* (Syarifah 2018). Pada sebuah bidang ilmu pengetahuan yang mempelajari dalam pembuatan sistem untuk mengambil keputusan dalam mengenali objek fisik nyata dan sesuai keadaan berdasarkan sebuah gambar atau citra adalah *computer vision*. Contohnya :

1. *Object Detection*: Mengenali sebuah objek ada pada *scene* dan mengetahui dimana batasannya.
2. *Recognition*: Menempatkan label pada objek.
3. *Description*: Menugaskan properti kepada objek.
4. *3D Infrence*: Menafsirkan adegan 3D dari 2D yang dilihat.
5. *Interpreting motion*: Menafsirkan gerakan.

2.4 Deep Learning

Deep learning adalah salah satu bidang *machine learning* yang memanfaatkan banyak *layer* pengolahan informasi nonlinier untuk melakukan klasifikasi, pengenalan pola, dan ekstraksi fitur. Menurut Goodflow, dkk (2016). Pada Gambar 9 *deep learning* adalah sebuah pendekatan dalam menyelesaikan masalah pada sistem pembelajaran komputer yang menggunakan konsep yang kompleks dengan menggabungkan dari konsep-konsep yang lebih sederhana. Jika digambarkan sebuah graf bagaimana konsep tersebut dibangun di atas konsep yang lain. Graf ini akan dalam dengan banyak *layer*, hal tersebut menjadi alasan disebut sebagai *deep learning* (pembelajaran mendalam) (He et al. 2020).



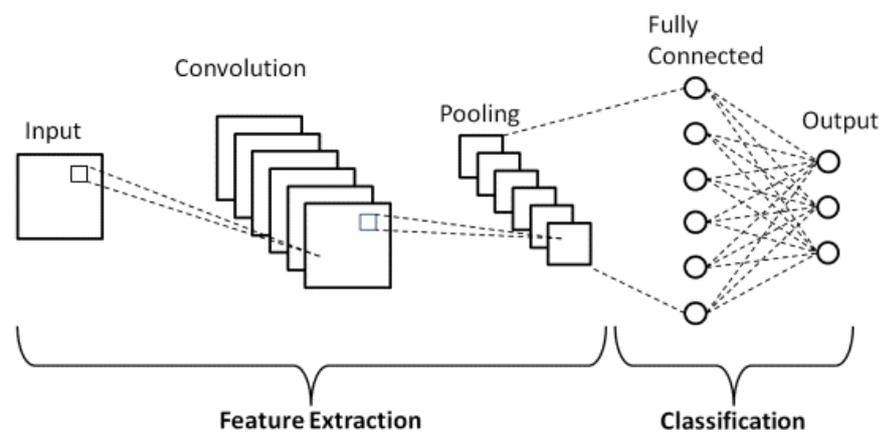


Gambar 9 *Deep learning*.

2.6.1 CNN

Convolutional Neural Network (CNN) adalah algoritma berbasis *Deep learning* yang dapat mengambil gambar sebagai *input*, menetapkan kelas untuk objek dalam gambar ini membedakan satu dari yang lain. *Preprocessing* CNN jauh lebih rendah jika dibandingkan dengan algoritma klasifikasi lainnya. CNN digunakan untuk memperkecil gambar tanpa kehilangan fiturnya yang membantu mendapatkan prediksi yang tepat. Untuk berhasil menangkap ketergantungan spasial dan temporal dalam gambar melalui penerapan *filter* yang relevan. Model ini dipilih karena tiga keunggulan utama, yaitu akurasi deteksi yang baik, parameter pelatihan yang lebih sedikit dan berjalan pada *hardware* komputasi yang minimal. (Sumahasan 2020)

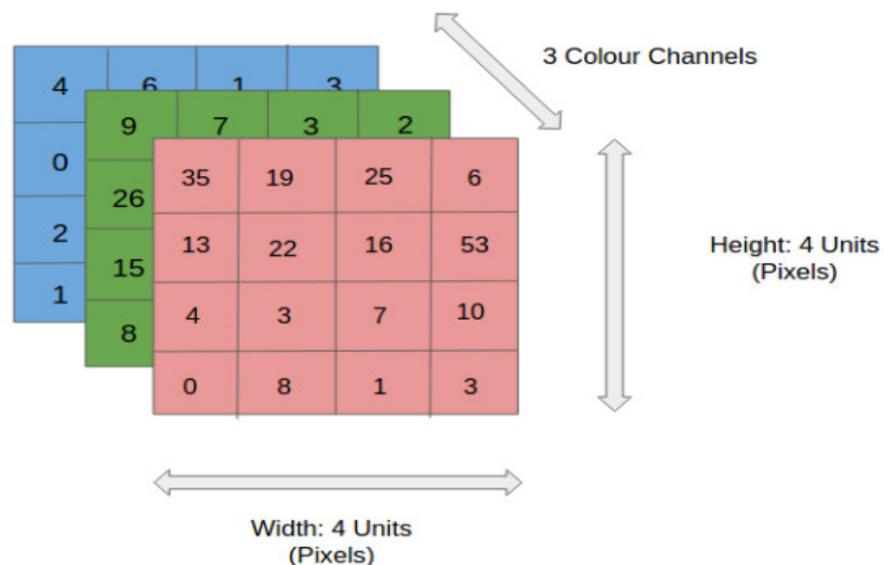
Gambar 10 menunjukkan arsitektur Algoritma CNN. Proses dari setiap lapisan dijelaskan sebagai berikut :



Gambar 10 Arsitektur CNN.



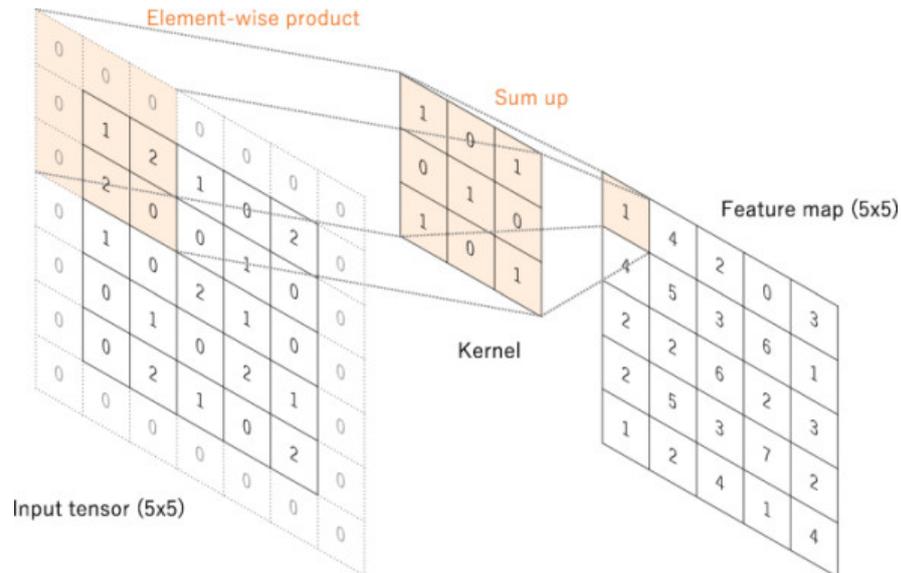
Lapisan *Input* data *training* dimulai dengan gambar *input* yang disediakan oleh pengguna. Seperti yang ditunjukkan pada Gambar 11 *input* diubah menjadi matriks berbentuk $128 \times 128 \times 3$ dimana 3 mewakili RGB yang merupakan bidang tiga warna merah, hijau dan biru.



Gambar 11 RGB Matriks.

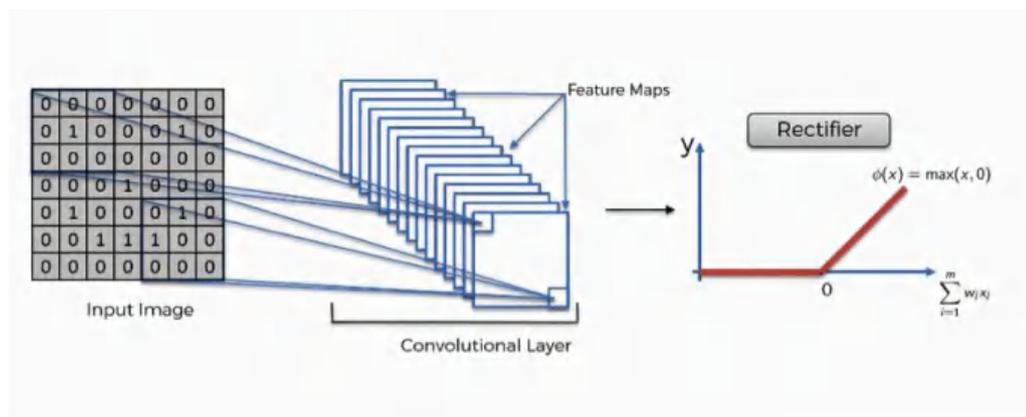
Selanjutnya *Convolutional Layer* bertujuan untuk mengekstraksi fitur seperti *edge*, *color* dan orientasi *gradien* dari gambar masukan. Operasi konvolusi dilakukan dengan bantuan elemen yang disebut kernel/filter, seperti yang ditunjukkan pada Gambar 12 memiliki banyak saluran (RGB), kernel memiliki kedalaman yang sama dengan gambar masukan. Dalam proses ini telah mengimplementasikan fungsi Conv2D karena gambar hanyalah *array* 2 dimensi. Kita dapat menggunakan *Convolution3D* jika kita perlu mengimplementasikan video, dimana dimensi ketiganya adalah waktu operasi konvolusi menghasilkan *feature map*.





Gambar 12 *Feature Extraction (multiplying input matrix with filters).*

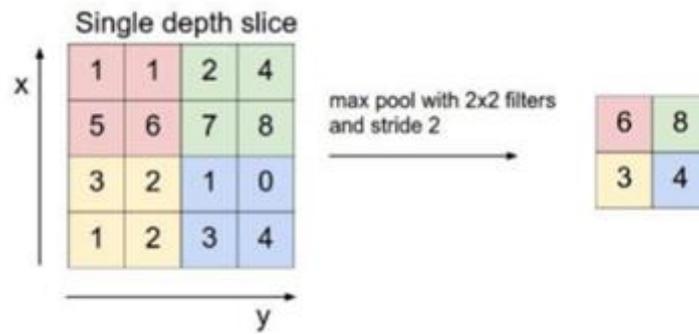
ReLU sebagai *function activation* adalah singkatan dari *Rectified Linear Unit* untuk operasi *non-linear*. *Outputnya* adalah $f(x) = \max(0, x)$. Ini diterapkan pada *feature maps* yang dihasilkan untuk mengubah nilai negatif menjadi positif (0), seperti yang ditunjukkan pada Gambar 13 *Activation Function*.



Gambar 13 *Activation Function.*

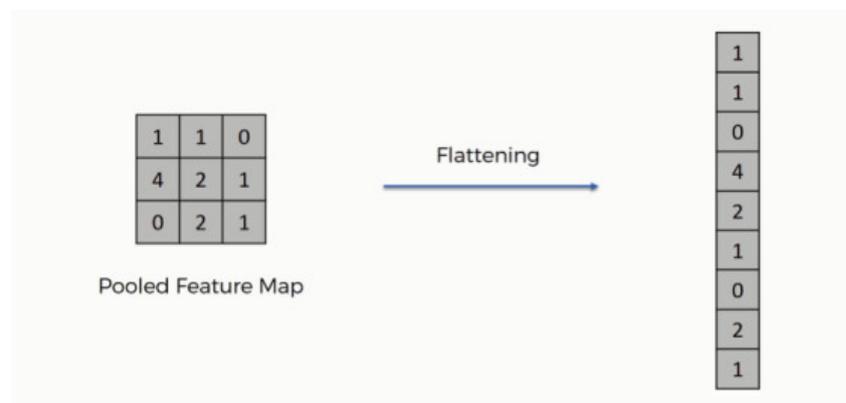
Berbagai jenis *pooling layer* adalah *max pooling*, *min pooling*, *mean pooling* dan *average pooling*. Dalam penggunaan *max-pooling* karena membutuhkan nilai piksel maksimum dari *region* yang diinginkan. Seperti yang ditunjukkan pada Gambar 14 telah menerapkan fungsi *MaxPooling2D* karena untuk mencapai piksel minimum dan mengurangi kompleksitas model tanpa mengorbankan *performance*. *Max Pooling* mengurangi aktivasi *noise* sama sekali dan mengurangi *noise* bersama dengan dimensinya.





Gambar 14 Max-pooling.

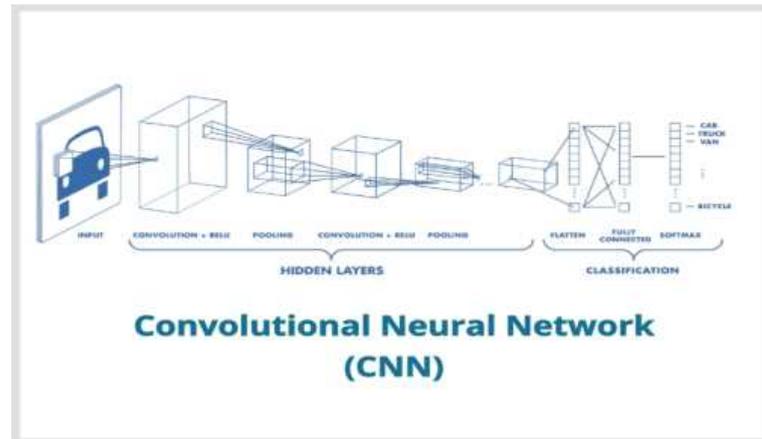
Flattening atau perataan seperti yang ditunjukkan pada Gambar 15 merupakan proses konversi *array* 2 dimensi yang dihasilkan dari operasi *convolution pooling* menjadi *single long* vektor linier kontinu. Untuk melakukan proses *flattening*, diterapkan fungsi matriks menjadi vektor dan memasukkan ke lapisan *fully connectoed layer* seperti *neural network*.



Gambar 15 Flattening.

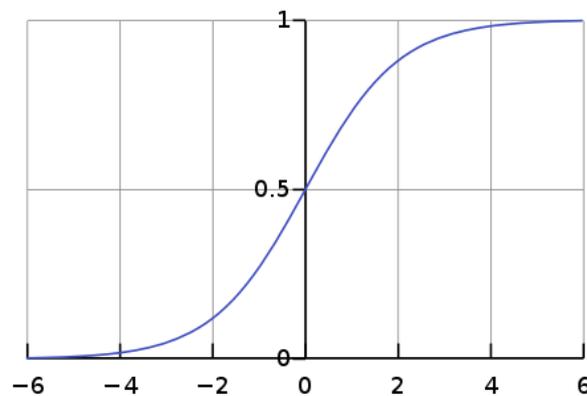
Fully Connected Layer yang diambil dari vektor *flattening* digunakan untuk mengklasifikasi gambar ke dalam label dan meneruskannya ke lapisan berikutnya yang ditunjukkan pada Gambar 16 Setelah digabungkan fitur untuk membuat sebuah model. Setelah *features extracted* oleh lapisan *convolution layer* lalu dikurangi oleh lapisan *pooling layer*, gambar yang direduksi dipetakan dengan *subgroup* dari lapisan *fully connected layer*. Seperti probabilitas dari setiap kelas dalam tugas klasifikasi. *Outputnya* dari *fully connected layer* memiliki jumlah *node* yang sama dengan jumlah kelasnya.





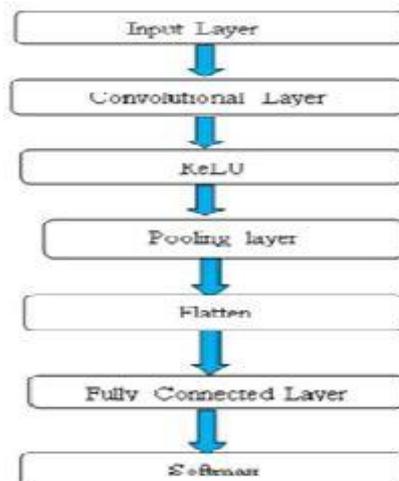
Gambar 16 Block Diagram CNN.

Softmax sebuah fungsi aktivasi yang diterapkan dalam beberapa kelas klasifikasi. Tujuan utama dari fungsi ini adalah menormalkan nilai *output* dari lapisan terakhir yang terhubung sepenuhnya ke probabilitas target kelas, dimana setiap nilai antara 0 dan 1, seperti yang ditunjukkan pada Gambar 17 Pada serangkaian *epoch*, model dapat membedakan antara dominasi dan level fitur terendah pada gambar dan mengklasifikasikannya menggunakan fungsi aktivasi *Softmax*.



Gambar 17 Softmax Activation.





Gambar 18 *Flow Chart CNN Algorithm*.

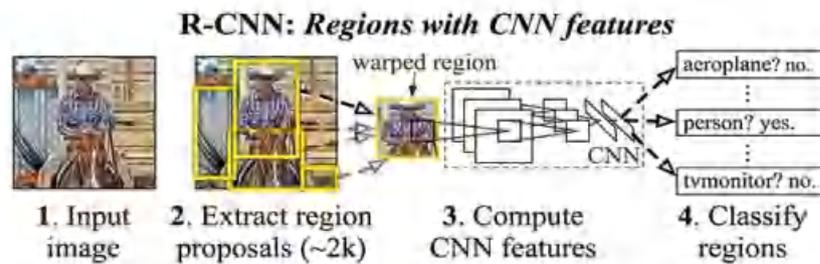
Arsitektur *convolutional neural network* pada Gambar 18 standarnya menumpuk beberapa lapisan konvolusional (masing-masing umumnya diikuti oleh *layer ReLU*), kemudian *pooling layer*, lalu beberapa lapisan konvolusional lainnya (+ *ReLU*), lalu *pooling layer* lain, dan seterusnya. Gambar semakin kecil dan semakin kecil seiring banyaknya jaringan, dan juga akan semakin dalam dengan kata lain lebih banyak *layer* (dengan lebih banyak peta fitur) berkat lapisan konvolusional. Dibagian atas CNN, terdapat *feed forward neural network* reguler ditambahkan, terdiri dari beberapa *fully connected layer* (+ *ReLU*s), dan lapisan akhir menampilkan prediksi (lapisan *softmax* yang menghasilkan perkiraan probabilitas kelas) (Sumahasan 2020).

2.6.2 R-CNN

Convolutional neural network untuk deteksi objek seperti yang telah dibahas pada sebelumnya, CNN dapat digunakan sebagai pendekatan untuk *problem-problem* dibidang *computer vision* selain klasifikasi gambar permasalahan dengan menggunakan teknik *sliding window* banyaknya potongan-potongan gambar yang harus diproses oleh CNN. Banyaknya lokasi gambar serta ukuran *sliding window* yang digunakan, komputasi dari keseluruhan proses tersebut akan sangat berat.

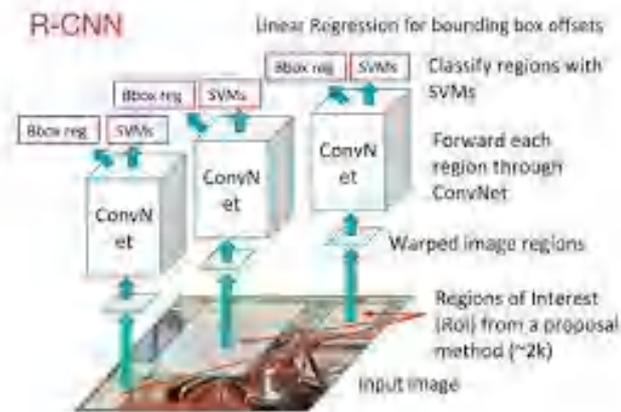


mul teknik *Regions* dengan CNN *feature* atau *Region based CNN* yang ulkan oleh peneliti dari UC Berkeley, Ross Girshik, et al. Pada tahun 2014 san 2020).



Gambar 19 Arsitektur R-CNN.

Seperti yang mungkin dilihat pada gambar diatas, alih-alih memproses sejumlah besar gambar yang dipangkas jadi *overfeat*. Pencarian selektif mengusulkan sekitar dua ribu wilayah per gambar, kemudian menyebar melalui CNN untuk mengekstraksi fitur dan kemudian melatih menggunakan pengklasifikasi SVM, selama waktu pengujian. Teknik *non-maximum suppression* digunakan untuk menolak kandidat yang daerahnya tumpang tindih dengan skor lebih tinggi. R-CNN akan mengurangi jumlah beberapa proses dengan begitu akan meningkatkan performa dari *overfeat* hingga hampir 50% (B. S 2020).



Gambar 20 Block Diagram R-CNN

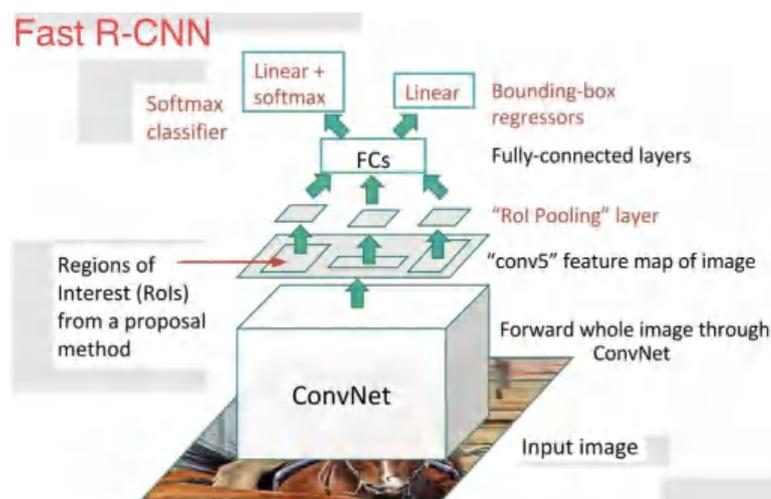
Mencari *region* atau bagian gambar yang mungkin merupakan sebuah objek, dengan metode *region proposal*. Salah satu contoh teknik *region* untuk setiap *region* tersebut kemudian dijadikan *input* untuk CNN sebagai *feature extractors* dari tiap *region* tersebut. Proposal dimana setiap fitur-fitur yang dihasilkan, menjadi *input* untuk SVM (yang akan menghasilkan kelas dari region dan *linear regression* (yang akan menghasilkan *bounding box*). Meskipun R-CNN menggunakan CNN yang telah dilatih sebelumnya untuk



mengekstrak fitur gambar secara efektif, namun lambat. Bayangkan kita memilih ribuan proposal wilayah dari satu gambar *input*: ini membutuhkan ribuan CNN *forward propagation* untuk melakukan deteksi objek. Beban komputasi yang besar ini membuat R-CNN tidak dapat digunakan secara luas dalam aplikasi dunia nyata (Czum 2020).

2.6.3 Fast R-CNN

Meskipun peningkatannya cukup signifikan, proses *training* dari R-CNN memiliki banyak kendala. Kita perlu menghasilkan *region* sebelum dapat memulai proses *training*. Selain itu, tidak hanya kita perlu melatih CNN itu sendiri, kita juga perlu melakukan *training* untuk SVM yang digunakan. Oleh karena itu, setahun berikutnya, yaitu pada tahun 2015, Ross Girshick mengembangkan R-CNN menjadi *Fast R-CNN*. Alur kerja dari *Fast R-CNN* sedikit berbeda dari sebelumnya, dimana bila sebelumnya tiap *region-region* dari *region proposal method* (*Selective Search*) akan memiliki CNN untuk *feature extraction* masing-masing, *Fast R-CNN* hanya menggunakan 1 CNN. *Feature map* yang dihasilkan kemudian “dicocokkan” dengan *Region of Interest* (ROI) yang didapat dari *Selective Search* tadi, untuk kemudian diklasifikasi kelasnya dan dideteksi *bounding box*. Dengan kata lain, *Fast R-CNN* melakukan *feature extraction* sebelum mengajukan *regions* (Czum 2020).



Gambar 21 Blok Diagram *Fast R-CNN*.

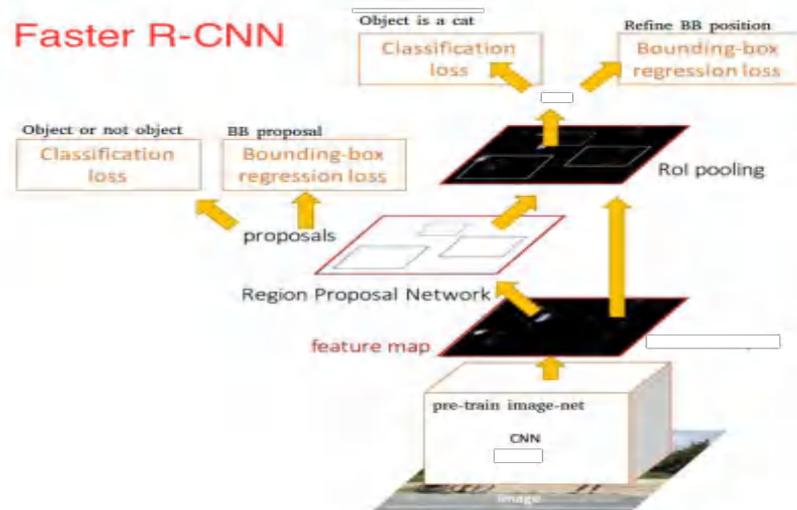


Pada gambar diatas *Fast R-CNN* juga meninggalkan penggunaan SVM sebagai *classifiers*, menggantinya dengan *ROI Pooling* dan *fully-connected layers*. Pendekatan dengan 1 CNN, *ROI pooling layer* dan *feed forward network* ini tidak hanya mempercepat performa R-CNN menjadi lebih cepat, tetapi juga menambah kapabilitas R-CNN menjadi *end-to-end differentiable* dan juga mempermudah proses *training* (karena tidak perlu lagi melakukan *training* untuk SVMs dan hanya melatih 1 arsitektur CNN ketimbang banyak CNN seperti sebelumnya). Alasan "*Fast R-CNN*" lebih cepat daripada R-CNN adalah karena Anda tidak perlu memasukkan 2000 *region proposal* ke *neural network* konvolusional setiap saat. Alih-alih, operasi konvolusi dilakukan hanya sekali per gambar dan peta fitur dihasilkan darinya. (Czum 2020).

2.6.4 *Faster R-CNN*

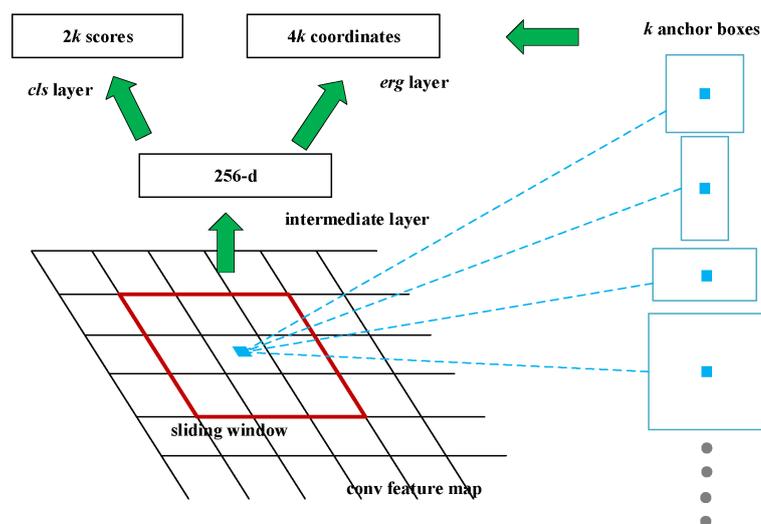
Faster Region-Based Convolutional Neural Network (Faster R-CNN) merupakan salah satu metode deteksi objek yang terdiri dari beberapa bagian yaitu CNN dan RPN sebagai *region proposer*, dan *Fast R-CNN* sebagai *detector*. Secara umum arsitektur dari *Faster R-CNN* diilustrasikan seperti gabungan dari metode *Fast R-CNN* dan *Region Proposal Network (RPN)*. Terdiri dari 5 bagian utama yaitu *deep convolutional network*, *Region Proposal Network*, *ROI Pooling Layer* dan *Fully Connected Layer*, *bounding box regressor* dan *classifier*. Melalui *region proposal networks* dan *deep fully convolutional* banyak objek yang diajukan untuk menjadi proposer lalu dinormalisasi melalui *ROI Pooling Layer*. Kemudian, *Fully Connected layer* mengekstrak fitur yang baik untuk melakukan klasifikasi dan regresi tahapan tersebut dapat diilustrasikan seperti pada Gambar 22 Blok Diagram *Faster R-CNN* (PARDEDE and HARDIANSAH 2022).





Gambar 22 Blok Diagram *Faster R-CNN*.

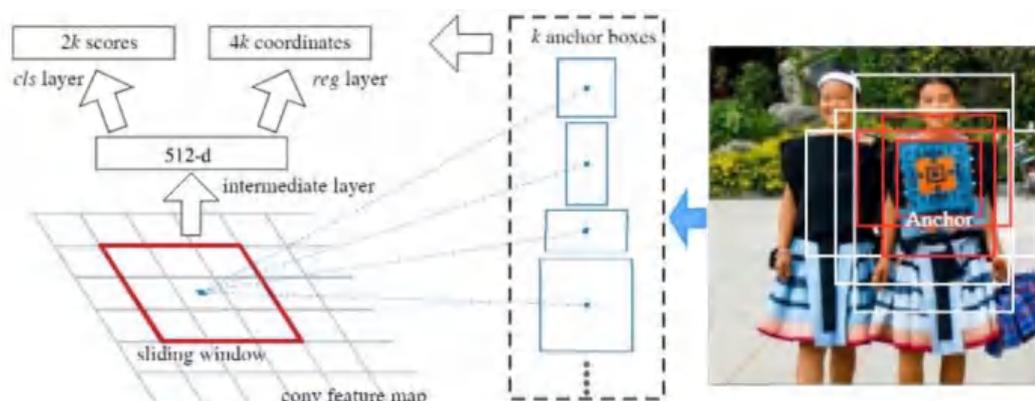
Metode terpenting dari *Faster R-CNN* sendiri adalah RPN, dimana RPN adalah sebuah *neural network* yang menggantikan peran *selective search* pada metode sebelumnya, yang berguna untuk mengajukan *region*. RPN menghasilkan beberapa *bounding box*, setiap *box* memiliki 2 nilai probabilitas apakah pada lokasi tersebut terdapat objek atau tidak. Untuk mendapatkan nilai probabilitas tersebut RPN menggunakan metode *anchor*. *Anchor* adalah sebuah area yang berfungsi untuk menilai apakah pada daerah tersebut ada objek atau tidak dengan cara menghitung nilai *Intersect of Union* (IoU). Proses RPN dapat diilustrasikan seperti pada Gambar 23 *Region Proposal Network* (RPN) (Charli et al. 2020).



Gambar 23 *Region Proposal Network* (RPN).



RPN mengambil gambar dari berbagai ukuran sebagai *input* dan *ouput* sekumpulan proposal objek persegi panjang, masing-masing dengan skor objektivitas. Gambar *input* diteruskan ke jaringan *convolutional layer* terakhir yang menampilkan *feature map*. *Sliding window* ditempatkan pada setiap bagian dari *feature map*. *Sliding window mask* biasanya diambil dari ukuran *mask* $n \times n$. Sesuai dengan setiap *sliding window*, secara bersamaan memprediksi beberapa *region* proposal, dimana jumlah proposal maksimum yang mungkin untuk setiap lokasi dilambangkan sebagai k . *Layer reg* memiliki *output* $4k$ yang mengkodekan koordinat kortak k , dan *layer class* menghasilkan $2k$ yang memperkirakan probabilitas objek atau tidak objek untuk setiap proposal. Setiap *anchor* diposisikan di tengah dari *sliding windows*. Secara standar menggunakan 3 skala dan 3 rasio aspek yang menghasilkan $k=9$ anchor pada setiap *sliding windows*. Klasifikasi menunjukkan probabilitas 0 atau 1 yang menunjukkan apakah wilayah tersebut berisi objek atau tidak dengan $p^* = 1$ jika $\text{IoU} > 0.7$, $p^* = 0$ jika $\text{IoU} < 0.3$, $p^* = 0$ jika selain itu (Charli et al. 2020).

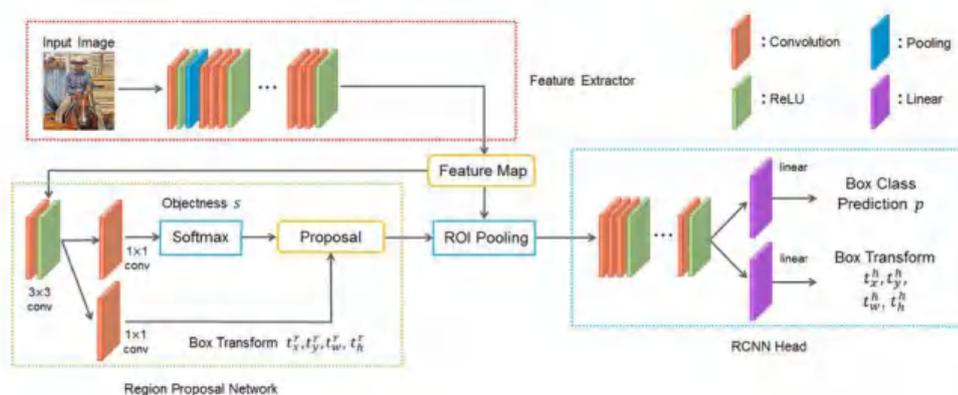


Gambar 24 Diagram RPN.

Keluaran pada proses ini terdapat 6 bagian yaitu probabilitas objek, probabilitas bukan objek, x koordinat, y koordinat, lebar, dan tinggi. Bagian – bagian tersebut yang akan membentuk *bounding box*. Proses *train* RPN dapat dilihat seperti Gambar 24 diatas. *Input* yang digunakan pada *layer* RoI adalah *feature maps* yang merupakan *output* dari CNN dengan *multiple convolution layers pooling layers*. Sebuah $N \times N$ matriks dibuat dengan membagi ruang *iap* menjadi RoI. Kolom pertama merupakan indeks dari citra dan sisa innya merupakan koordinat dari RoI, dimulai dari koordinat paling kiri



atas hingga kiri bawah. RoI yang sudah ditentukan disebut sebagai *region proposal*. Pada RPN, awalnya citra *input* diproses dalam jaringan saraf konvolusi untuk menghasilkan *feature map*. *Feature map* terdiri atas enam bagian, yaitu penentuan *object* dan *non-object* dengan nilai 0–1, koordinat nilai x dan y, serta nilai *weight* dan *height* dari *bounding box*. *Sliding window* ditempatkan pada setiap *feature map* dengan ukuran $N \times N$, sesuai dengan setiap *anchor sliding window* yang dibentuk. Setiap *anchor* memiliki titik pusat yang sama, tetapi memiliki *aspect ratios* dan *scaling factor* yang berbeda (Czum 2020).



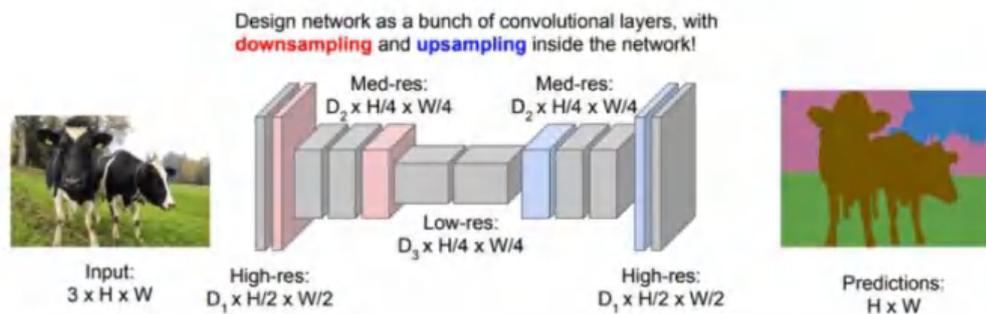
Gambar 25 Struktur Model *Faster R-CNN*.

2.6.5 *Mask R-CNN*

Para peneliti dengan cepat berlomba-lomba dalam meningkatkan kualitas sistem *object* deteksi yang bertujuan dalam menyelesaikan masalah di bidang *semantic segmentation* dan *instace segmentation* dalam waktu singkat. Dengan adanya metode sebelumnya *Faster R-CNN* menghasilkan objek deteksi *bounding box* dan *Fully Convolutional Network* (FCN) digunakan deteksi objek dan segmentasi *semantik*. FCN Segmentasi dapat dicapai dengan menggunakan arsitektur yang mirip dengan masalah klasifikasi dengan sedikit modifikasi. Satu prediksi untuk setiap piksel sehingga menemukan kelas yang berbeda dalam



gambar.



Gambar 26 Struktur Model FCN

FCN juga dikenal sebagai *dense prediction*, segmentasi *semantik* memberi label pada setiap piksel gambar *input* dengan kelas masing-masing yang mewakili objek/tubuh tertentu. FCN menggunakan CNN (*Convolutional Neural Network*) dalam proses menghasilkan piksel tiap piksel antar objek pada *dense prediction* ilustrasi FCN dapat dilihat pada gambar 26 Solusi populer untuk masalah yang dihadapi oleh arsitektur sebelumnya adalah dengan menggunakan *Downsampling* dan *Upsampling* adalah *Fully Convolutional Network*. Pada paruh pertama model, menurunkan resolusi *spasial* gambar yang mengembangkan kompleks *feature mappings*. Pada Gambar 27 Menurut Fei-Fei Li dan Justin Johnson Ilustrasi teknik *computer vision* taks dapat dijelaskan metode segmentasi gambar yang menetapkan setiap piksel dalam gambar pada suatu objek. Itu hanya berurusan dengan piksel, mengklasifikasikan setiap piksel dalam gambar dengan label kelas seperti anjing, orang, kucing, dll.:

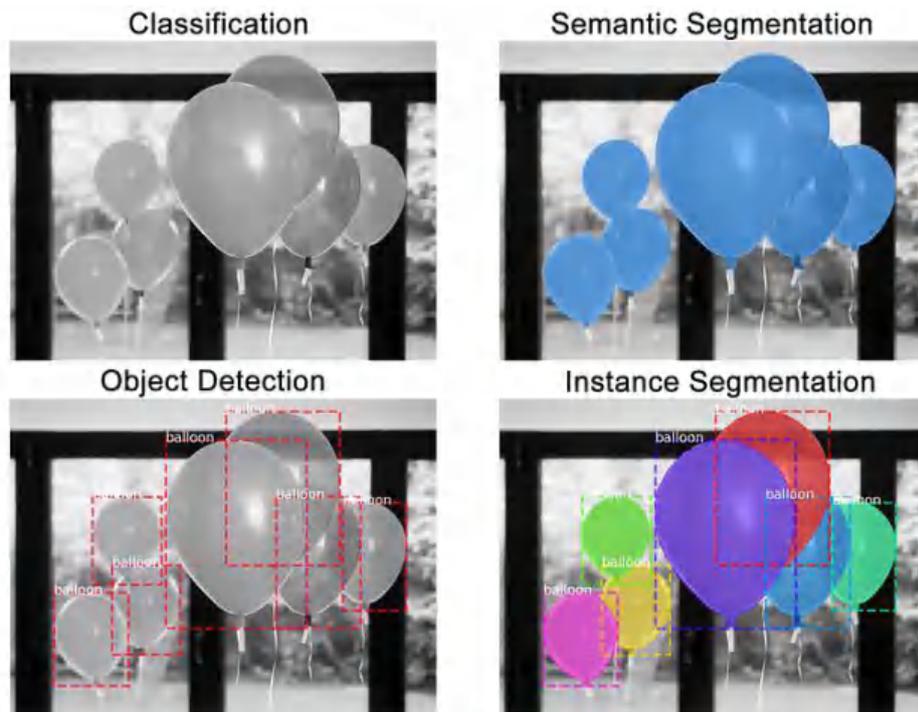
- Semantic segmentation* memperlakukan banyak objek yang termasuk dalam kelas yang sama sebagai satu kesatuan. Jadi, misalnya, ini bertujuan untuk melabeli semua anjing dalam sebuah gambar sebagai “Anjing”
- Instane Segmentation* membedakan beberapa *instance* dari kelas yang sama. Ini memberikan label yang berbeda seperti “Anjing 1”, “Anjing 2”, dll. Untuk setiap anjing.





Gambar 27 Ilustrasi teknik *computer vision* taks.

Mask R-CNN ini merupakan *framework deep learning* yang diperkenalkan pada tahun 2017 oleh tim *Facebook AI Research (FAIR)*. Metode ini dapat melakukan objek deteksi pada gambar yang juga menghasilkan sebuah *mask segmentation* pada setiap objeknya atau biasa disebut dengan *instance segmentation* (He et al, 2018). Berdasarkan Gambar dibawah dapat dilihat bahwa *instance segmentation* mirip dengan *object detection* di mana masing-masing objek dideteksi satu per satu namun kemudian dikombinasikan dengan *semantic segmentation* sehingga setiap objek dapat diketahui kelas, lokasi serta dapat dibedakan pikselnya.



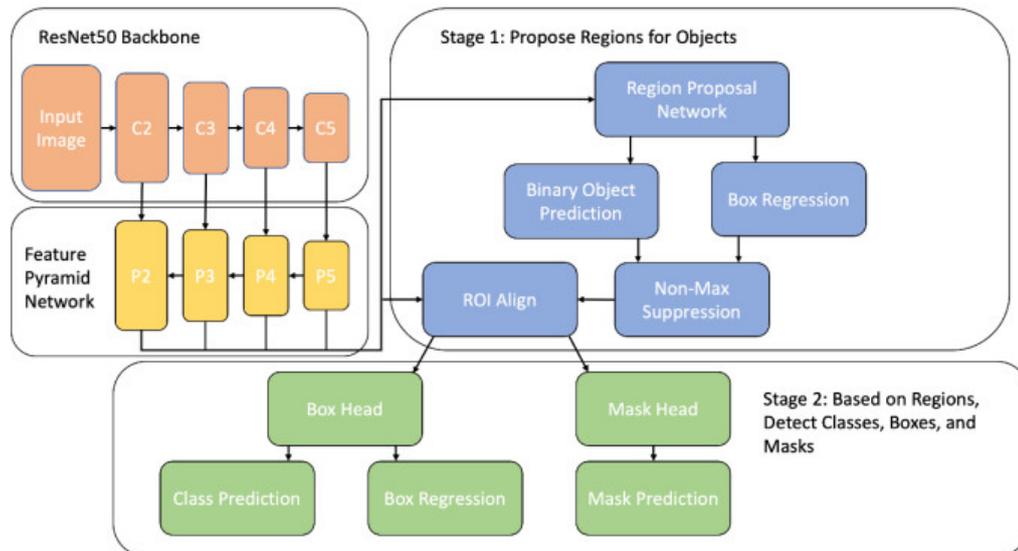
Gambar 28 Ilustrasi *Mask R-CNN* (Abdulhah, 2018).



ifikasi: Ada balon di gambar ini.

entasi *Semantik*: Ini semua adalah piksel balon.

- Deteksi Objek: Ada 7 balon di gambar ini. Mulai memperhitungkan objek yang tumpang tindih.
- Segmentasi *Instance*: Ada 7 balon pada gambar ini, dan memiliki piksel masing-masing pada setiap balon.

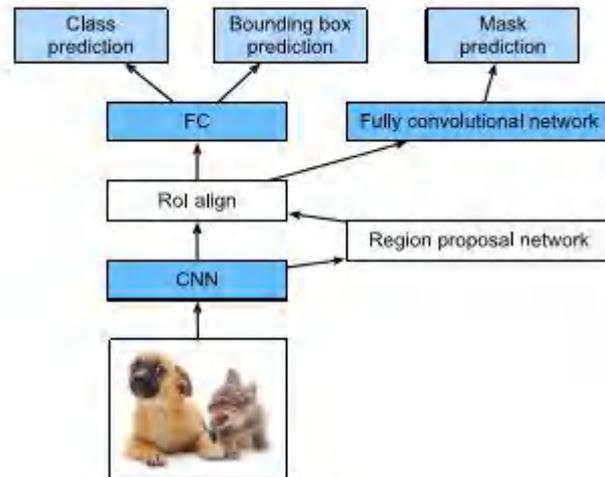


Gambar 29 Arsitektur *Mask R-CNN* (aws.amazon.com.2021).

Salah satu contoh model *Convolutional Layer* yang cukup populer adalah *ResNet*. *ResNet50* adalah Jaringan *Residual* yang berisi 50 layer. Seperti VGG, ukuran lapisan konvolusi adalah *filter* 3×3 , ukuran *input* model ini ditetapkan sebagai 224×224 . Model ini memiliki lebih sedikit filter dan kompleksitas yang lebih rendah dari pada jaring VGG dan ada varian lain seperti *ResNet101* dan *ResNet152* (Czum 2020). Gambar 29 Menunjukkan lapisan konfigurasi jaringan *ResNet50*. Model dari *Mask R-CNN* dibagi menjadi 2 bagian yaitu Melakukan *scan* terhadap citra dan menghasilkan proposal (area yang kemungkinan mengandung objek) pada citra tersebut. Mengklasifikasikan proposal dan menghasilkan *bounding box* dan *mask* pada setiap objek dalam citra tersebut.

Mask R-CNN adalah sebuah metode *Deep Neural Network* yang memiliki dua tahap kerangka kerja pertama scan gambar dan menghasilkan *proposal* (area si objek) kedua klasifikasi proposal dan menghasilkan *bounding box* dan dalam data set *training* tingkat piksel dari objek juga diberi label pada *Mask r-cnn* secara efektif memanfaatkan label terperinci untuk lebih tkan keakuratan deteksi objek. (Czum 2020)





Gambar 30 Model *Mask R-CNN*.

Seperti yang ditunjukkan pada gambar diatas, *Mask R-CNN* modifikasi *Faster R-CNN*. Secara khusus, *Mask R-CNN* menggantikan *layer region of interest pooling* dengan *layer alignment region of interest (ROI)*. Lapisan *layer ROI alignment* ini menggunakan menggunakan *interpolasi bilinear* untuk menyimpan informasi pada *feature maps*, yang lebih cocok untuk memprediksi tingkat pixelnya. Keluaran lapisan ini berisi peta fitur dengan bentuk sama untuk lapisan pada wilayah *interest*. Proses ini digunakan untuk memprediksi tidak hanya kelas dan *bounding box* untuk semua *region interest*, tetapi juga posisi tingkat piksel objek melalui jaringan *convolutinal network*. Rincian lebih lanjut dalam penggunaan *fully convolutional networks* untuk memprediksi tingkat *pixel-level semantic* dari suatu gambar yang akan diberikan. Berdasarkan *Faster R-CNN*, *Mask R-CNN* juga memperkenalkan jaringan yang konvolusional, sehingga dapat memanfaatkan label tingkat piksel untuk meningkatkan akurasi deteksi objek. (Czum 2020).

2.4 Confusion Matrix

Confusion matrix biasa juga disebut *error matrix* sebuah ringkasan prediksi pada permasalahan klasifikasi. Jumlah klasifikasi yang benar dan yang salah akan dengan nilai hitung kemudian dipecah oleh setiap kelas. Sehingga *n matrix* tidak hanya memberikan informasi kesalahan yang *classifier* a kesalahannya (Amwin 2021).



Confusion matrix dapat dilihat dengan beberapa ketentuan sebagai berikut :

- a. *Positive* (P): aktual bernilai positif.
- b. *Negative* (N): aktual bernilai negatif.
- c. *True Positive* (TP): aktual bernilai positif, dan diprediksi positif.
- d. *True Negative* (TN): aktual bernilai negatif, dan diprediksi negatif.
- e. *False Positive* (FP): aktual bernilai negatif, tetapi diprediksi positif.
- f. *False Negative* (FN): aktual bernilai positif, tetapi diprediksi negatif.

Precision adalah pecahan elemen *True Positive* dibagi dengan jumlah total unit yang diprediksi secara positif (jumlah kolom dari prediksi positif). Secara khusus, *True Positive* adalah elemen yang telah diberi label positif oleh model dan sebenarnya positif, sedangkan *False Positive* adalah elemen yang telah diberi label positif oleh model, tetapi sebenarnya negatif.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall adalah pecahan elemen *True Positive* dibagi dengan jumlah total unit yang diklasifikasikan secara positif (jumlah baris yang sebenarnya bernilai positif). Khususnya *False Negative* adalah elemen yang telah diberi label negatif oleh model, tetapi sebenarnya positif.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-Score menilai kinerja model klasifikasi dari *confusion matrix*. Rumus *F1-score* dapat diartikan sebagai rata-rata antara *Precision* dan *Recall*, di mana *F1-score* mencapai nilai terbaiknya pada 1 dan skor terburuk pada 0. Kontribusi relatif dari *precision* dan *recall* adalah sama pada *F1-score* dan rata-rata harmonik berguna untuk menemukan *trade-off* terbaik antara dua kuantitas.

$$F_1 \text{ Score} = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (4)$$



asi adalah salah satu metrik yang paling populer dalam klasifikasi *multi-* mus Akurasi mempertimbangkan jumlah elemen *True Positive* dan *True* pada pembilang dan jumlah semua entri *confusion matrix* pada penyebut.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$

(5)

Akurasi mengembalikan ukuran keseluruhan tentang seberapa banyak model memprediksi dengan benar pada seluruh kumpulan data. Elemen dasar metrik adalah individu tunggal dalam kumpulan data. Setiap unit memiliki bobot yang sama dan mereka berkontribusi sama pada nilai akurasi. Ketika kita berpikir tentang kelas dari pada individu tunggal (keseluruhan isi tabel), akan ada kelas dengan jumlah unit yang tinggi dan yang lain hanya dengan sedikit. Dalam situasi ini, kelas yang padat akan memiliki bobot yang lebih tinggi dibandingkan dengan kelas yang terkecil. Oleh karena itu, akurasi paling cocok ketika kita hanya peduli pada individu tunggal dari pada beberapa kelas (Amwin 2021).

