ALGORITMA PENDETEKSI DAN PENGENALAN OBJEK PADA MEDIA EDUKASI LKS SECARA OTOMATIS



Disusun Oleh:

N a m a : Fionna Saphira Farhani

NIM : 16523018

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

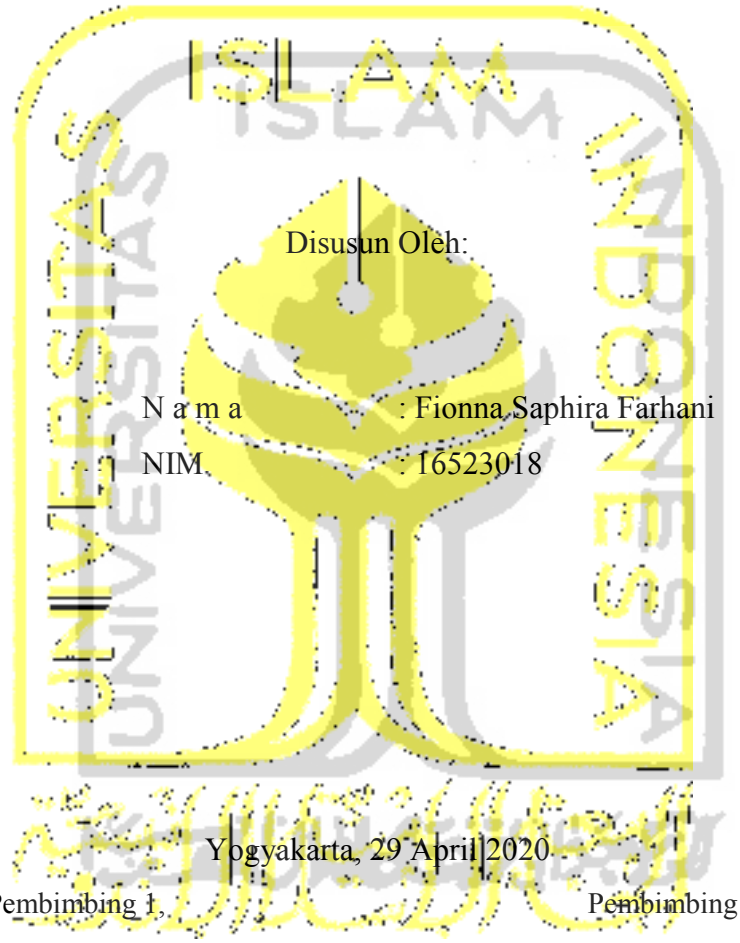
UNIVERSITAS ISLAM INDONESIA

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

ALGORITMA PENDETEKSI DAN PENGENALAN OBJEK PADA MEDIA EDUKASI LKS SECARA OTOMATIS

TUGAS AKHIR



Disusun Oleh:

N a m a : Fionna Saphira Farhani

NIM : 16523018

Yogyakarta, 29 April 2020

Pembimbing 1, Pembimbing 2,

(Arrie Kurniawardhani, S.Si., M.Kom.) (Sheila Nurul Huda, S.Kom., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

ALGORITMA PENDETEKSI DAN PENGENALAN OBJEK PADA MEDIA EDUKASI LKS SECARA OTOMATIS

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 29 April 2020

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom. _____

Anggota 1

Taufiq Hidayat, S.T., M.C.S. _____

Anggota 2

Septia Rani, S.T., M.Cs. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fionna Saphira Farhani

NIM : 16523018

Tugas akhir dengan judul:

ALGORITMA PENDETEKSI DAN PENGENALAN OBJEK PADA MEDIA EDUKASI LKS SECARA OTOMATIS

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 29 April 2020

(Fionna Saphira Farhani)

HALAMAN PERSEMBAHAN

Segala puji dan syukur bagi Allah SWT. Karena tugas akhir yang merupakan bagian dari perjuangan dan ibadah saya kepada Allah SWT akhirnya telah selesai. Dengan segala kendala yang dihadapi, Alhamdulillah tugas ini selesai. Tugas akhir ini saya persembahkan untuk:

1. Bapak Nudkhi Farhani Zuharon, ibu Tutik Welas Ismiyati dan eyang Surati yang saya cintai, saya kasihi dan juga saya hormati. Terimakasih atas segala bentuk perhatian, dukungan dan doa yang telah diberikan karenanya saya dapat menyelesaikan tugas akhir saya. Mungkin ini belum cukup untuk membalas segala usaha dan kebaikan yang diberikan pada saya. Tugas akhir ini saya persembahkan kepada ibu, bapak dan eyang saya sebagai wujud cinta dan terima kasih saya kepada para beliau.
2. Kedua adik saya Malva Nabilla Farhani dan Muhammad Axell Rizqee Farhani yang selalu mendukung, mendoakan dan menyayangi saya. Terimakasih atas segala perhatian dan dukungannya serta semangatnya, waktu berharga dalam hidup saya adalah waktu ketika saya dengan kalian.
3. Rekan seperjuangan Sabika Amalina, Sari Kurnia, Puspita Dewi, Assyifa Narulita, Ulfa Amalia, Sabila Amalia, Veda Murti, Mirotus Solekhah, Rizal Hamdan, Kahfiara, Amalia Citra, Lantang, Mufti dan Laili terimakasih atas dukungan yang luar biasa, telah menjadi teman seperjuangan yang berarti bagi saya dan selalu ada ketika saya berada dalam kesulitan. Kalian keluarga kedua saya dan terima kasih telah ada di hidup saya selama perkuliahan ini.
4. Ibu Arrie Kurniawardhani, S.Si., M.Kom. dan ibu Sheila Nurul Huda, S.Kom., M.Cs. selaku dosen pembimbing, terimakasih untuk semua nasihat, bimbingan, ilmu, dan dukungan yang luar biasa yang selama ini telah diberikan.
5. Jurusan Informatika Universitas Islam Indonesia yang telah mewadahi saya dalam menuntut ilmu dan sebagai wadah untuk saya mengembangkan diri.
6. Serta seluruh pihak yang tidak bisa saya sebutkan satu-persatu, terimakasih atas segala perhatian dan dukungan yang luar biasa.

HALAMAN MOTO

“Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras untuk urusan yang lain, dan hanya kepada Tuhanmulah engkau berharap”

(Q.S Al-Insyirah : 7-8)

“Hai orang-orang beriman apabila kamu dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: "Berdirilah kamu", maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan.”

(Q.S Al-Mujadillah : 11)

“If you love everything you do to reach your dream even though it needs more time, your dream will come true”

(Wendy Shon)

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji dan syukur saya panjatkan atas kehadirat Allah SWT. Yang telah memberikan ridho dan rahmatnya sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul "Algoritma Pengenalan Marker Augmented Reality pada Media Edukasi LKS Secara Otomatis". Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Starta-1 (S1) dan sebagai sarana untuk mengimplementasikan ilmu-ilmu yang telah dipelajari selama masa studi di Jurusan Informatika Universitas Islam Indonesia. Penulis sadar, tanpa dukungan serta motivasi laporan ini tidak akan dapat selesai tepat pada waktunya. Oleh karena itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya, Bapak Nudkhi Farhani Zuharon dan Ibu Tutik Welas Ismiyati untuk doa, dukungan, semangat dan rasa percaya terhadap diri saya untuk menyelesaikan studi ini.
2. Kedua saudara, Malva Nabilla Farhani dan Muhammad Axell Rizqee Farhani atas dukungan, doa dan semangatnya yang telah diberikan.
3. Bapak Fathul Wahid, S.T., M.Sc., Ph.D., selaku selaku Rektor Universitas Islam Indonesia.
4. Bapak Prof. Dr. Ir. Hari Purnomo selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Ibu Arrie Kurniawardhani, S.Si., M.Kom. dan Ibu Sheila Nurul Huda, S.Kom., M.Cs. selaku dosen pembimbing, untuk semua nasihat, bimbingan, ilmu, dan dukungan yang luar biasa yang selama ini telah diberikan.
7. Segenap keluarga besar teman-teman di Fakultas Teknologi Industri terutama dari Jurusan Informatika Universitas Islam Indonesia yang telah memberikan bantuan dan dukungannya.
8. Rekan seperjuangan FTI48 (Sabika Amalina, Sari Kurnia, Puspita Dewi, Assyifa Narulita, Ulfa Amalia, Sabila Amalia), Veda Murti, Mirotus Solekhah, Rizal Hamdan, dan Pabji Learning (Kahfiara, Amalia Citra, Febriana, Dwitiya) atas dukungan yang luar biasa, telah menjadi teman seperjuangan yang berarti bagi saya dan selalu ada ketika saya berada dalam kesulitan. Kalian keluarga kedua saya dan terimakasih telah ada di hidup saya selama perkuliahan ini.

9. Lantang, Mufti, Laili dan Dwitiya sebagai teman seperjuangan dalam konsentrasi Visi Komputer.
10. Teman-teman Hexadecima Angkatan 2016, keluarga Kosmik dan keluarga Marcomm FTI.
11. Semua pihak yang tidak bisa saya sebutkan satu per satu, terimakasih atas semua bentuk dukungannya.
12. Tidak lupa, terima kasih kepada diri saya sendiri karena selalu bersemangat dan pantang menyerah dalam proses pengerjaan Tugas Akhir ini.

Semoga segala kebaikan, bantuan, dukungan, bimbingan dan pengajaran yang telah diberikan kepada penulis mendapatkan kebaikan dari Allah SWT. Penulis memohon maaf apabila dalam proses penyelesaian Tugas Akhir terdapat kesalahan yang disengaja maupun tidak disengaja. Penulis menyadari atas kemampuan yang dimiliki, semoga kedepannya laporan ini dapat berguna dan bermanfaat bagi semua yang membacanya.

Yogyakarta, 29 April 2020



(Fionna Saphira Farhani)

SARI

Perkembangan teknologi informasi kini semakin pesat. Penyampaian informasi semakin cepat dan berkembang variasinya. Penyampaian informasi yang semakin cepat ini dimanfaatkan pada bidang keilmuan yang berbeda salah satunya dalam bidang Pendidikan. Media pembelajaran saat ini sangatlah banyak kita temukan variasinya, dari membuat buku khusus untuk latihan soal, media edukasi secara daring, pembelajaran melalui permainan, dan lain-lain. Media pembelajaran LKS merupakan salah satu media pembelajaran yang paling umum digunakan di sekolah-sekolah yang ada di Indonesia. Namun dalam penggunaannya, buku LKS menggunakan kertas buram untuk menekan biaya produksinya. Sehingga ini membuat para tenaga pengajar kesulitan untuk menjelaskan atau menerangkan objek gambar yang ada pada buku LKS. Dengan permasalahan yang dihadapi penulis mencari sebuah penyelesaian dengan mengaitkan pengenalan objek pada citra. Pada penelitian ini, penulis akan mencoba teknik pengenalan objek dengan menggunakan metode *Faster Region based Convolutional Neural Network* (R-CNN) yang akan diterapkan pada pendeteksi dan pengenalan gambar pada buku LKS. Citra yang digunakan pada penelitian ini yaitu, sel hewan, sel tumbuhan, otot polos, otot lurik, otot jantung, monokotil, dikotil,syaraf dan jantung. Dengan harapan kedepannya dapat diaplikasikan untuk pengenalan *marker* pada teknologi *Augmented Reality* sebagai media pembelajaran. Data diambil oleh penulis menggunakan kamera *smartphone* dengan jumlah citra *training* 495 gambar, testing 90 gambar dan evaluasi 45 gambar. *Faster* R-CNN telah terbukti mampu memproses citra yang akan dideteksi dengan akurasi mencapai 87% .

Kata kunci: Augmented Reality, Media pembelajaran LKS, Object Recognition, Faster Region based Convolutional Neural Network,

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Pertanyaan Penelitian 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 3
1.6 Metodologi Penelitian 3
1.7 Sistematika Laporan 4
BAB II LANDASAN TEORI 6
2.1 Tinjauan Pustaka 6
2.2 Landasan Teori 7
2.2.1 Anatomi 7
2.2.2 Lembar Kerja Siswa (LKS) 8
2.2.3 Citra 9
2.2.4 Pengolahan Citra Digital (*Digital Image Processing*) 10
2.2.5 Pembelajaran Mesin (*Machine Learning*) 11
2.2.6 Deteksi Objek (*Object Detection*) 11
2.2.7 Convolutional Neural Network (CNN) 12
2.2.8 Region based Convolutional Neural Network (R-CNN) 12
2.2.9 *Fast* R-CNN 13
2.2.10 *Faster* R-CNN 14
2.2.11 Training Neural Network 14
2.2.12 Konvolusi 15
2.2.13 Padding 15
2.2.14 Pooling 16
2.2.15 Activation Function 18
2.2.16 Fully Connected Layer 18
2.2.17 Learning Rate 19
2.2.18 Batch Size 19
2.2.19 Confusion Matrix 19
2.2.20 *Python* 20
2.2.21 *Tensorflow* 20
BAB III METODOLOGI PENELITIAN 21
3.1 Perancangan 21
3.1.1 Perancangan Penelitian 21
3.1.2 Studi Pendahuluan 21

3.1.3 Eksperimen Pendahuluan 21
3.1.4 Perumusan Masalah 22
3.1.5 Studi Pustaka 22
3.1.6 Pengumpulan Data 22
3.2 *Labelling* Data 24
3.2.1 Pelabelan citra 24
3.2.2 Konversi XML ke CSV 25
3.2.3 Konversi CSV ke TFRecord 25
3.2.4 Pembuatan Labelmap 25
3.3 *Training Process* 25
3.3.1 Konfigurasi 25
3.3.2 *Load* Data 25
3.3.3 Deteksi Objek 25
3.4 Pengujian Data 26
3.5 Detection Output 26
BAB IV HASIL DAN PEMBAHASAN 27
4.1 *Pre-processing* Data 27
4.1.1 Pelabelan Citra 27
4.1.2 Konversi Data XML ke CSV 28
4.1.3 Konversi *Dataset* CSV ke TFRecord 30
4.1.4 Label Map 31
4.2 *Training* Data 33
4.2.1 Konfigurasi 33
4.2.2 Load Data 34
4.2.3 Deteksi Objek 35
4.2.4 Total *Loss* 35
4.2.5 Export Model 36
4.2.6 Model 37
4.3 *Testing* 37
4.4 Evaluasi 40
BAB V KESIMPULAN DAN SARAN 41
5.1 Kesimpulan 41
5.2 Saran 41
DAFTAR PUSTAKA 42
LAMPIRAN 45

DAFTAR TABEL

Tabel 2. 1 Tabel Judul Penelitian Sebelumnya 6
Tabel 4. 1 Tabel Pengujian *Confusion Matrix* 40

DAFTAR GAMBAR

Gambar 2. 1 Buku LKS yang digunakan 8
Gambar 2. 2 *Object detection with bounding box* 11
Gambar 2. 3 Gambaran proses CNN 12
Gambar 2. 4 Gambaran proses pada R-CNN 13
Gambar 2. 5 *Fast* R-CNN 14
Gambar 2. 6 *Faster* R-CNN 14
Gambar 2. 7 Ilustrasi proses konvolusi 15
Gambar 2. 8 Ilustrasi proses konvolusi menggunakan padding 16
Gambar 2. 9 pooling 3x3 pada citra 5x5 17
Gambar 2. 10 Jenis pada pooling 17
Gambar 2. 11 Grafik pada ReLU 18
Gambar 2. 12 Ilustrasi *fully connected layer* 19
Gambar 3. 1 Alur pada penelitian 21
Gambar 3. 2 Contoh manipulasi data 23
Gambar 3. 3 Citra Media Edukasi LKS 24
Gambar 3. 4 Tahap pada *Labelling* data 24
Gambar 3. 5 Proses labeling pada citra 24
Gambar 3. 6 Diagram alir *training process* 26
Gambar 4. 1 Proses Pelabelan Objek Citra 27
Gambar 4. 2 Isi dari file XML setelah pelabelan 28
Gambar 4. 3 Program Konversi dari XML ke CSV 30
Gambar 4. 4 Hasil konversi dari XML ke CSV 30
Gambar 4. 5 Mendefinisikan label pada TFRecord 31
Gambar 4. 6 Isi pada *labelmap.*pbtxt 32
Gambar 4. 7 Bagian yang dapat diganti dari konfigurasi *ResNet* 34
Gambar 4. 8 Pemanggilan data pada proses *training* 34
Gambar 4. 9 Proses *training* pada *Tensorboard* 35
Gambar 4. 10 Total *loss* pada saat melakukan *training* 36
Gambar 4. 11 Program yang digunakan *extract* model yang telah di *training.* 37
Gambar 4. 12 Hasil citra yang berhasil dideteksi 38
Gambar 4. 13 Hasil citra yang berhasil dideteksi 38
Gambar 4. 14 Hasil deteksi pada citra LKS gelap 39

Gambar 4. 15 Hasil deteksi citra pada LKS pada sudut pandang samping 39

BAB I
PENDAHULUAN

1.1 Latar Belakang

Media pembelajaran kini semakin banyak variasinya, salah satunya adalah Lembar Kerja Siswa. Salah satu media pembelajaran yang sering digunakan di kelas adalah Lembar Kerja Siswa (LKS). LKS berisi soal-soal yang cukup banyak, sehingga sering dimanfaatkan oleh tenaga pengajar untuk memberikan latihan kepada peserta didik. Selain banyak soal latihan, LKS juga memuat ringkasan materi sehingga siswa dapat lebih mudah untuk memahami materi yang disampaikan. Buku LKS memiliki sedikit halaman tidak seperti buku paket, sehingga siswa merasa lebih mudah untuk membawa buku LKS jika dibandingkan dengan membawa buku paket yang lebih tebal. Selain dari kelebihan diatas, LKS juga memiliki kekurangan. Umumnya LKS menggunakan kertas buram karena dirasa lebih ringan dan untuk menekan biaya produksi, buku LKS di cetak dengan hanya menggunakan tinta hitam, banyak terjadi cacat cetak pada LKS sehingga sulit untuk siswa mengerti gambar yang dijelaskan di dalam buku.

Pada masa ini, banyak sekali dijumpai kolaborasi antara teknologi informasi dengan media pembelajaran, salah satunya Augmented Reality. Augmented Reality adalah sebuah teknologi menggabungkan benda maya dengan lingkungan nyata dalam bentuk 3 dimensi dalam waktu yang nyata (Biswas, 2019). Teknologi ini memungkinkan untuk menampilkan informasi secara visual dengan 3 dimensi yang dirasa dapat membantu pengguna dalam memahaminya. Teknologi ini telah banyak dikolaborasikan dengan media pemasaran, permainan dan media pembelajaran. Dalam pengembangan Augmented Reality, diperlukan pengenalan penanda agar dapat memunculkan objek 3 dimensi. Untuk pengenalan penanda dilakukan sebuah proses yang biasa disebut deteksi objek atau *object detection*. Deteksi objek adalah proses pengenalan sebuah objek yang ada pada citra digital untuk menentukan keberadaan dan mengenalinya. Pada teknologi Augmented Reality proses deteksi objek sangat penting dilakukan agar dapat menampilkan informasi visual dalam bentuk 3 dimensi yang diinginkan.

Dengan tujuan untuk mengembangkan variasi model pembelajaran dan memberikan kemudahan dalam melakukan pengajaran di sekolah, penulis menawarkan sebuah solusi untuk membangun teknologi *augmented reality* pada lembar buku LKS. Objek yang akan dikenali

antara lain struktur tumbuhan dikotil, monokotil, otot polos, otot lurik, otot jantung, syaraf, jantung, sel hewan dan tumbuhan. Setelah melakukan studi pustaka, penulis melakukan sebuah eksperimen pendahuluan untuk mengetahui masalah yang ada pada pendeteksi dan pengenalan objek dalam teknologi AR di salah satu *framework*. Dalam eksperimen pendahuluan yang telah dilakukan penulis, Augmented Reality gagal mengenali objek penanda dikarenakan kertas cetak buram yang mengakibatkan kurang jelasnya gambar objek penanda. Pada kesempatan ini, penulis ingin mencoba menggunakan salah satu algoritma dari *machine learning* yaitu *Faster* R-CNN untuk membantu mengenali objek penanda pada media pembelajaran LKS . Dengan LKS yang memiliki karakter citra keabuan dan tidak tercetak dengan baik dengan.

1.2 Pertanyaan Penelitian

Berdasarkan latar belakang masalah yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Apakah algoritma yang disusun dengan memanfaatkan *deep learning* dapat mengenali objek struktur organ makhluk hidup pada media edukasi LKS?
b. Bagaimana tingkat akurasi pada pendeteksian objek organ makhluk hidup menggunakan *Faster R-Convolutional Neural Network?*
c. Bagaimana hasil pengenalan dari organ makhluk hidup pada media edukasi LKS pada kertas buram?

1.3 Batasan Masalah

Batasan penelitian yang diberikan agar lebih terarah dan sesuai dengan yang dimaksudkan adalah sebagai berikut:

a. Data yang digunakan hanya terbatas pada media pembelajaran LKS Biologi dengan karakter citra RGB dengan latar belakang keabuan.
b. Objek citra yang digunakan adalah organ makhluk hidup yaitu struktur dikotil, monokotil, otot polos, otot lurik, otot jantung, syaraf, jantung, sel hewan dan sel tumbuhan.
c. Jumlah *dataset training* yang digunakan adalah 495 citra yang dipisah menjadi 9 kelas, yakni 55 citra di setiap kelasnya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

a. Program dapat mendeteksi dan mengenali objek citra LKS Biologi menggunakan metode *Faster R-CNN.*
b. Tingkat akurasi yang tinggi dihasilkan pada deteksi objek citra menggunakan *Faster R-Convolutional Neural Network.*
c. Program dapat digunakan pada pengembangan pendeteksian dan pengenalan objek pada citra buram.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

a. Dapat dimanfaatkan dalam pengembangan teknologi *Augmented Reality* untuk mengenali penanda berupa objek pada media edukasi LKS.
b. Memudahkan tenaga pengajar untuk menyampaikan materi bergambar buram.
c. Memudahkan peserta didik untuk memahami materi yang disampaikan di kelas.
d. Hasil penelitian dapat dikembangkan menjadi media pembelajaran baru yang dapat dengan mudah dipahami oleh siswa.

1.6 Metodologi Penelitian

Langkah-langkah yang diterapkan pada penelitian ini agar dapat mencapai tujuan yang diharapkan adalah sebagai berikut:

a. Tahap Pengumpulan Data

Pengumpulan data dilakukan untuk menunjang pelaksanaan penelitian. Data yang dikumpulkan ini nantinya yang akan dijadikan bahan dalam pembuatan program pengenalan objek makhluk hidup. Baik itu data untuk *training*/*training,* validasi, maupun pengujian.

b. Perancangan Sistem

Perancangan merupakan tahap penggambaran perencanaan sistem yang akan dibuat agar dapat dibuat dengan lebih terstruktur dan memudahkan dalam pembangunan sistem. Rancangan meliputi diagram alur proses pada sistem menggunakan *flowchart*

c. Implementasi Sistem

Implementasi adalah langkah untuk mengimplementasi hasil studi literatur kepada sebuah sistem terhadap permasalahan yang ditemukan. Langkah implementasi meliputi data citra, pemrosesan data, dengan keluaran berupa data citra yang telah berhasil

dikenali dengan box beserta nama kelas sebagai keluaran dari proses deteksi yang dilakukan.

d. Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan sistem sesuai dengan tujuan dari penelitian dilakukan, yaitu mendeteksi organ makhluk hidup pada media edukasi LKS.

1.7 Sistematika Laporan

Dalam penyusunan laporan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

Bab I Pendahuluan

Berisi latar belakang permasalahan yang mendasari dari penelitian, pentingnya melakukan deteksi objek organ makhluk hidup pada media edukasi LKS, dan solusi yang ditawarkan terkait dengan permasalahan tersebut. Berdasarkan latar belakang, kemudian disusun rumusan masalah atau pertanyaan penelitian sebagai dasar dari perencanaan penyelesaian masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Landasan Teori

Berisi uraian literatur yang terkait dengan topik penelitian sebagai dasar dalam melakukan penelitian yang dilakukan dan beberapa penelitian sejenis yang dilakukan. Teori-teori yang diuraikan dalam bab ini antaranya menggunakan jurnal, buku, dan artikel sebagai bahan referensi yang terkait dengan penelitian. Adapun permasalahan yang diambil pada referensi merupakan deteksi objek, *machine learning, Convolutional neural network,* beserta *tools* yang digunakan.

Bab III Metodologi Penelitian

Berisi tahapan dan kebutuhan dari penelitian yang dilakukan sebagai acuan untuk mencapai solusi yang ditawarkan. Bab ini terdiri dari pengumpulan data, karakter data, perancangan sistem dengan menggunakan diagram alir.

Bab IV Hasil dan Pembahasan

Berisi hasil penelitian dan pembahasannya dari setiap proses yang dilakukan dalam pembangunan system, pengujian sistem, kelebihan dan kekurangan system. Pengujian dari kinerja sistem menggunakan uji validasi.

Bab V Kesimpulan dan Saran

Berisi kesimpulan mengenai hasil dari penelitian yang telah dilakukan sesuai dengan tujuan penelitian beserta saran yang mendukung untuk penelitian kedepannya yag mungkin akan dilanjutkan oleh peneliti lain.

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Referensi dari penelitian terdahulu dibutuhkan untuk menjadi landasan dari penelitian ini berdasarkan Tabel 2.1.

Tabel 2. 1 Tabel Judul Penelitian Sebelumnya

Judul	Metode	Penulis
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks	Faster R-CNN	Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun
Fast Point R-CNN	R-CNN	Yilun Chen, Shu Liu, Xiaoyong Shen, Jiaya Jia
Fast R-CNN	Fast R-CNN	Ross Girshick
Implementation of Training Convolutional Neural Networks	Convolutional Neural Networks	Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, Jun Zhang
Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition	Faster R-CNN	Vipul Sharma , Roohie Naaz Mir

Berdasarkan Tabel 2.1, pada penelitian dengan judul "*Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*" dijelaskan bahwa pada penelitian tersebut dijelaskan bahwa penulis melakukan perbandingan terhadap RPN yang paling efektif dan akurat. dengan membagi fitur konvolusi dengan down-stream jaringan deteksi, Iangkah dalam region proposal mendekati cost-free. Metode yang digunakan oleh penulis memungkinkan penyatuan, basis deep learning deteksi objek untuk berjalan mendekati frame rates real-time. RPN yang telah di training juga memperbaiki kualitas dari region proposal dan keseluruhan akurasi dari deteksi objek (Shaoqing Ren, 2016).

Pada penelitian dengan judul "*Fast Point R-CNN*" disebutkan Pada paper ini, peneliti

mengusulkan framework yang generik, efektif dan cepat untuk deteksi objek 3D. Metode yang digunakan oleh penulis membuat kedua voxel representasi dan raw point cloud untuk mendapat keuntungan dari keduanya. tahap pertama mengambil voxes representation menjadi data masukan dan dikenakan proses konvolusi untuk memperoleh set inisial prediksi. selanjutnya memurnikan prediksi berdasarkan raw point cloud dan ekstraksi fitur konvolusi. konsep dari metode ini dinilai simpel namun pada prakteknya merupakan desain yang *powerfull* untuk mengatur kecepatan deteksi tertinggi (Yilun Chen, 2019).

Pada penelitian dengan judul "*Fast R-CNN*" dijelaskan bahwa penelitian tersebut menggunakan modifikasi dari CNN yaitu *Faster* R-CNN. Peneliti melakukan pembaruan dari *fast* R-CNN pada R-CNN dan SPPnet. Peneliti tersebut meningkatkan kualitas dari detektor. Pada penelitian ini, peneliti menyampaikan bahwa permasalahan pada CNN sangat mahal pada waktu prosesnya, dengan menggunakan *fast* R-CNN proses komputasi pada deteksi citra menjadi lebih praktis dan lebih cepat (Girshick, 2015).

Pada penelitian dengan judul "*Implementation of Training Convolutional Neural Networks*" disebutkan bahwa peneliti tersebut melakukan pengenalan wajah menggunakan *deep learning*. Peneliti menerapkan algoritma CNN untuk menggali informasi mendalam dalam proses pengenalan wajah dan memanfaatkan algoritma untuk komputasi paralel pada platform awan untuk mempercepat proses pengenalan wajah, menganalisis rasio perceptan teoritis dan verifikasi eksperimental (Tianyi Liu, 2015).

Pada penelitian dengan judul "*Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition"* disebutkan pada penelitian ini, penulis berfokus pada deteksi objek dan pengenalan menggunakan faster RCNN berdasar model. model konvensional melakukan hasil deteksi yang buruk, jadi penulis melakukan integrasi model deteksi yang dapat membantu mendefinisikan primary region dari gambar untuk meningkatkan presisi dari deteksi. langkah selanjutnya framework proposal generation di aplikasikan pada model regresi bounding box yang digunakan untuk mengurangi error pada deteksi, fungsi Rol Pooling mengkalkulasikan pembelajaran dari parameter pengenalan. penilaian komparatif demonstrasi meningkat secara signifikan pada efisiensi dari deteksi dan pengenalan (Vipul Sharma, 2019).

2.2 Landasan Teori

2.2.1 Anatomi

Anatomi berarti memotong. Anatomi adalah ilmu biologi yang mengulik dan mempelajari struktur dan organisasi dari sebuah makhluk hidup . Anatomi terdiri dari 3 jenis

yaitu Zootomi, Fitotomi dan anatomi manusia (TURNEY, 2007). Struktur anatomi pada hewan dan manusia pada umumnya sama mencakup otak, mata, jantung, paru-paru, lambung, limpa, pankreas, usus, kulit, ginjal, hati, uterus, saluran urin, tulang, dan lain-lain. Sedang pada tumbuhan mencakup akar, batang, daun, bunga, dan buah beserta biji .

2.2.2 Lembar Kerja Siswa (LKS)

Lembar kerja siswa (LKS) merupakan kumpulan lembaran yang berisi pengayaan yang harus dikerjakan oleh siswa. LKS dilengkapi dengan petunjuk, langkah-langkah dalam menyelesaikan permasalahan yang ada pada lembar tersebut. Pengayaan yang diberikan dalam lembar kerja harus sesuai dengan kompetensi dasar yang ingin dicapai (Majid, 2007).

Lembar kerja siswa adalah buku panduan bagi siswa sekolah dalam melatih diri untuk memecahkan suatu permasalahan. LKS berupa panduan latihan siswa dalam aspek kognitif maupun pengembangan aspek pembelajaran dalam bentuk refleksi dan eksperimen atau demonstrasi. LKS memuat kegiatan mendasar seperti soal-soal yang harus dikerjakan oleh siswa untuk memaksimalkan pemahaman siswa dalam menerima sebuah materi yang disampaikan. Pertanyaan-pertanyaan yang ada pada LKS haruslah sesuai dengan indikator-indikator dari pencapaian hasil pembelajaran yang ingin dicapai (Trianto 2010).

Buku LKS yang digunakan pada penelitian ini adalah buku LKS Biologi semester 1 yang diperuntukkan untuk kelas 11 Sekolah Menengah Atas (Wigawati Hadi, 2017). Sampul buku LKS tersebut ditunjukkan pada Gambar 2.1. Buku ini memiliki 218 halaman yang membahas seputar struktur organ makhluk hidup.



Gambar 2. 1 Buku LKS yang digunakan

Sumber. (Wigawati Hadi, 2017).

Struktur Lembar Kerja Siswa

1. Ringkasan materi

Pada buku Lembar Kerja Siswa, biasanya dilengkapi dengan materi-materi ringkas dan contoh pengerjaan soal yang dapat membantu siswa dalam mengerjakan latihan soal-soal yang ada pada buku LKS tersebut.

2. Soal-soal latihan

Lembar Kerja Siswa umumnya berisi soal-soal yang dapat digunakan siswa untuk melatih pengetahuan mereka. Pada LKS terdapat 2 jenis didalamnya yakni soal subyektif dan soal obyektif. Soal subyektif biasa disebut juga soal uraian yang memberikan kebebasan kepada siswa untuk menjawab pertanyaan yang diajukan. Adanya soal ini bertujuan untuk melatih siswa agar dapat memilih fakta yang relevan dengan persoalan dan melatih kemampuan analitik, sintetik, dan evaluatif. Soal obyektif atau biasa disebut soal pilihan ganda adalah soal yang dilengkapi dengan alternatif jawaban, sehingga siswa dapat memilih salah satu dari jawaban tersebut (Trianto 2010).

2.2.3 Citra

Citra merupakan gambar dua dimensi yang dihasilkan dari gambar analog yang kontinu menjadi gambar diskrit yang dihasilkan melalui proses *sampling*. Citra merupakan representasi dari sebuah biner pada informasi visual. Citra dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit (Diego Olivia, 2019). Hasil persilangan antara baris dan kolom disebut sebagai piksel. Gambar atau titik diskrit pada baris (n) dan kolom (m) disebut dengan piksel [n,m]. Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M\text{-}1$$

$$0 \leq y \leq N\text{-}1$$

$$0 \leq f(x,y) \leq G\text{-}1$$

dimana : M = jumlah piksel baris (row) pada array citra

N = jumlah piksel kolom (column) pada array citra

G = nilai skala keabuan (gray level)

Jenis Citra

Pada pengolahan citra umumnya citra dibagi menjadi tiga jenis yaitu, *color image, black and white* dan *binary image.*

1. *Color Image* atau RGB (*Red, Green, Blue*)

   Pada citra warna, masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*), biru (*Blue*). Jika setiap warna memiliki rentang kedalaman warna dari 0 sampai 255, maka rentang warna yang dimiliki citra warna atau RGB adalah 255 x 255 x 255 = 16.581.375 variasi warna (Diego Olivia, 2019).

2. *Black and White*

   Citra *black and white* atau citra hitam dan putih adalah citra yang setiap pikselnya mempunyai warna gradasi dari putih sampai hitam. Rentang warna tersebut berarti bahwa setiap piksel dapat diwakili 8 bit atau 1 byte.citra keabuan sangat cocok untuk digunakan pada pengolahan citra. Sebelum melakukan ekstraksi, citra warna akan diubah menjadi citra keabuan untuk mempermudah proses pengolahan citra. Citra keabuan salah satu fungsinya diterapkan dalam dunia kedokteran adalah citra X-ray.

3. *Binary Image*

   Setiap piksel hanya terdiri dari warna hitam dan putih, maka hanya perlu 1 bit per piksel antara 0 dan 1. Citra biner sangat cocok pada citra berisi teks, sidik jari atau gambar arsitektur. Citra biner juga sangat efisien dalam hal penyimpanan sehingga lebih mudah untuk diproses. Sehingga setelah citra warna diubah menjadi citra keabuan, maka langkah selanjutnya untuk memudahkan komputasi dengan mengubah citra keabuan menjadi citra biner (T.M Zaini, 2014).

2.2.4 Pengolahan Citra Digital (*Digital Image Processing*)

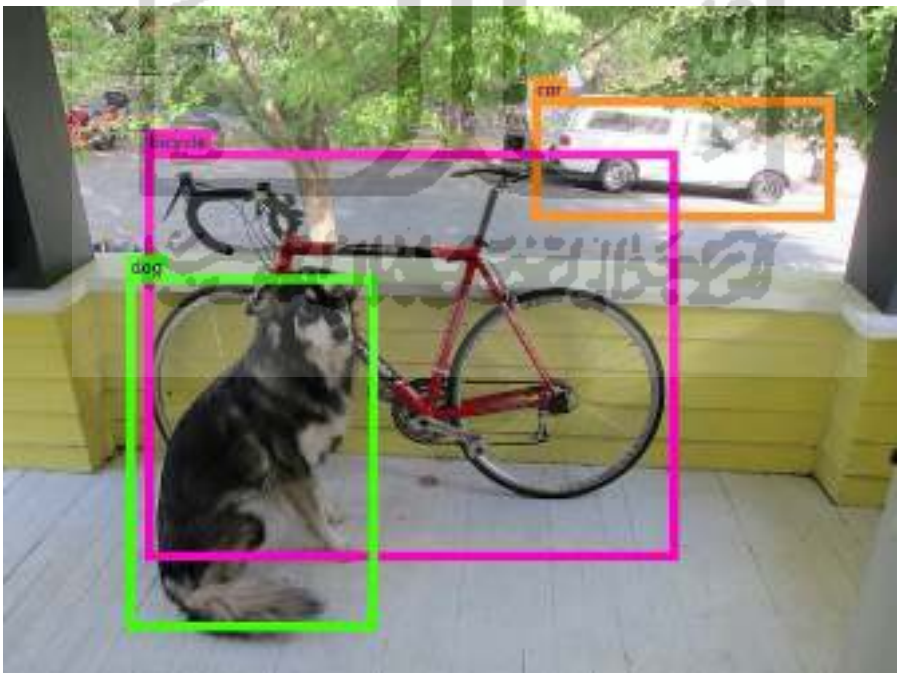
Pengolahan citra merupakan cabang ilmu yang mempelajari teknik-teknik dalam mengolah citra. Teknik dalam pengolahan citra diantaranya adalah memeriksa keadaan citra, deteksi objek dan lokalisasi, identifikasi dan verifikasi. Citra yang diolah adalah gambar diam (foto) maupun gambar bergerak (video) secara digital dengan menggunakan komputer. Pada teknik pengolahan citra terdapat beberapa bagian untuk pengolahannya yaitu, pengenalan pola, ekstraksi fitur, deteksi objek, dan proyeksi grafik (T.M Zaini, 2014).

2.2.5 Pembelajaran Mesin (*Machine Learning*)

Machine learning atau pembelajaran mesin adalah teknologi atau algoritma yang biasa digunakan diberbagai bidang ilmu komputer. *Machine learning* sangat memungkinkan pengguna mengambil berbagai pendekatan berdasar situasi sekitar dimana pekerjaan yang biasa dilakukan secara manual oleh manusia. Dalam penerapannya, pengguna membuat sebuah model dengan bantuan algoritma *machine learning* yang secara otomatis dapat mengenali pola, mengklasifikasikan data yang dimasukkan dan membuat prediksi berdasarkan data masukan tersebut (Naveen Sundar Govindarajulu, 2018).

2.2.6 Deteksi Objek (*Object Detection*)

Deteksi objek atau *object detection* adalah proses yang dilakukan untuk mengenali dan menentukan keberadaan dari sebuah objek tertentu yang ada pada suatu citra digital seperti pada Gambar 2.2. Proses dilakukannya deteksi dengan menggunakan metode yang umum digunakan untuk membaca fitur-fitur dari seluruh objek yang ada pada citra digital. Fitur yang didapat dari proses tadi kemudian dibandingkan dengan fitur objek referensi atau template. Dari perbandingan tersebut dapat digunakan untuk menentukan apakah suatu objek terdeteksi sebagai template atau objek referensi yang dimaksud atau tidak (Huang J. , 2019).

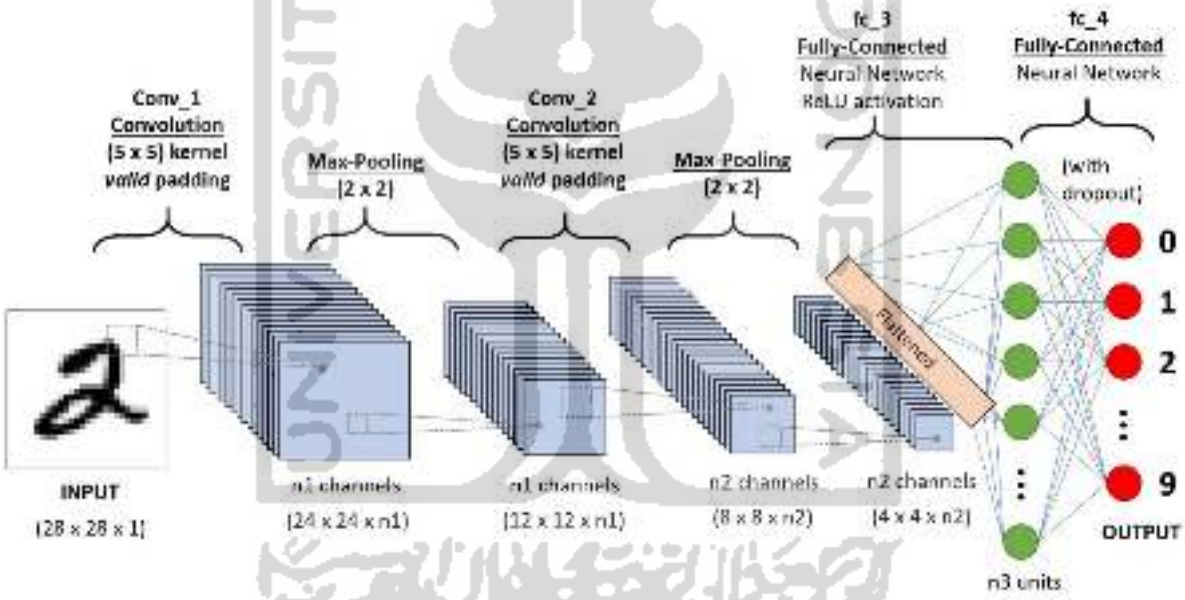


Gambar 2. 2 *Object detection with bounding box*

Sumber. (Rahyagara, 2018)

2.2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network adalah jenis lain dari jaringan saraf tiruan. CNN adalah bagian dari *deep neural network* yang biasa digunakan untuk menganalisis gambar visual. Penggunaan terkait CNN biasanya pada citra, video, pengenalan suara dan pengenalan Bahasa natural (NLP). CNN terdiri dari beberapa tahap *training. Input* dan *output* dari masing-masing tahap yang ada pada CNN adalah *array* yang biasa disebut *feature map. Feature map* atau peta fitur adalah fitur hasil pengolahan citra yang di *input.* Arsitektur pada CNN seperti yang digambarkan pada Gambar 2.3 umumnya meniru bagaimana kerja otak manusia dalam mencerna informasi visual yang diterimanya dari mata menuju *Visual Cortex.* Tahap yang terdiri dari tiga *layer* yaitu *convolutional layer, activation layer* dan *pooling layer* (Saad Albawi, 2017).



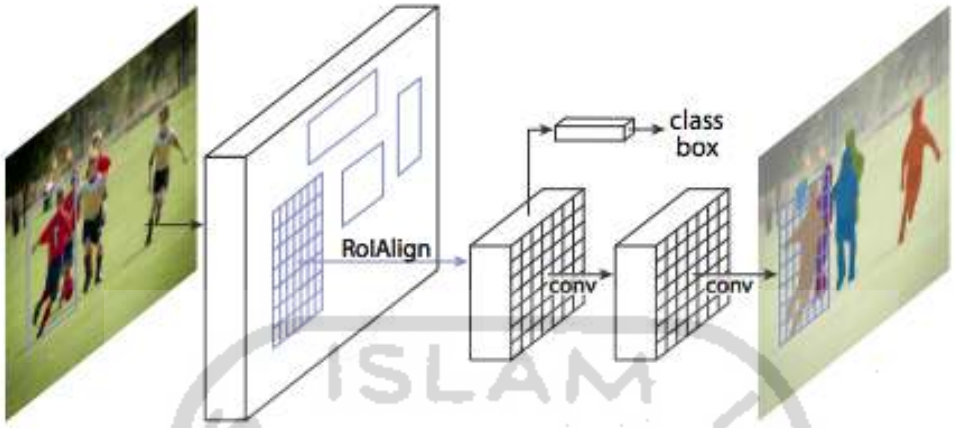
Gambar 2. 3 Gambaran proses CNN

Sumber. (Saha, 2018)

2.2.8 Region based Convolutional Neural Network (R-CNN)

Region-based Convolutional Neural Network adalah sebuah algoritma terusan dari *Convolutional Neural Network* yang biasa digunakan dalam *object detection.* R-CNN mengkombinasikan wilayah citra yang kaya akan fitur dengan *convolutional neural network.* Seperti pada Gambar 2.4, R-CNN memiliki dua jaringan yaitu *Region Proposal Network* yang

digunakan untuk menghasilkan sebuah proposal wilayah pada citra dan jaringan satunya digunakan untuk mendeteksi sebuah objek pada citra (Girshick, 2015).

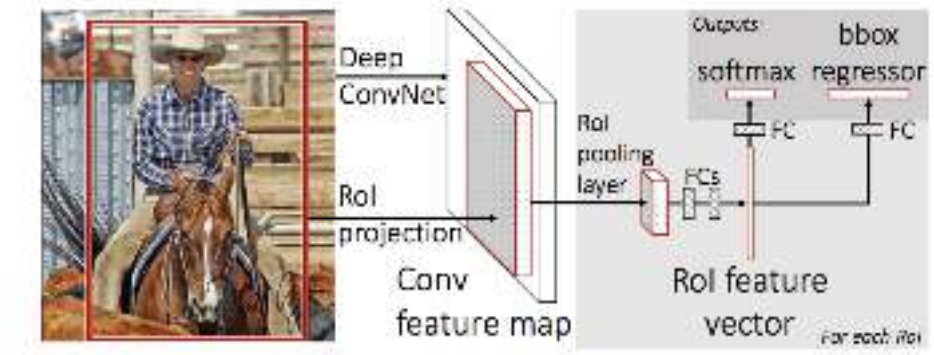


Gambar 2. 4 Gambaran proses pada R-CNN

Sumber. (Danilyuk, 2018)

2.2.9 *Fast* R-CNN

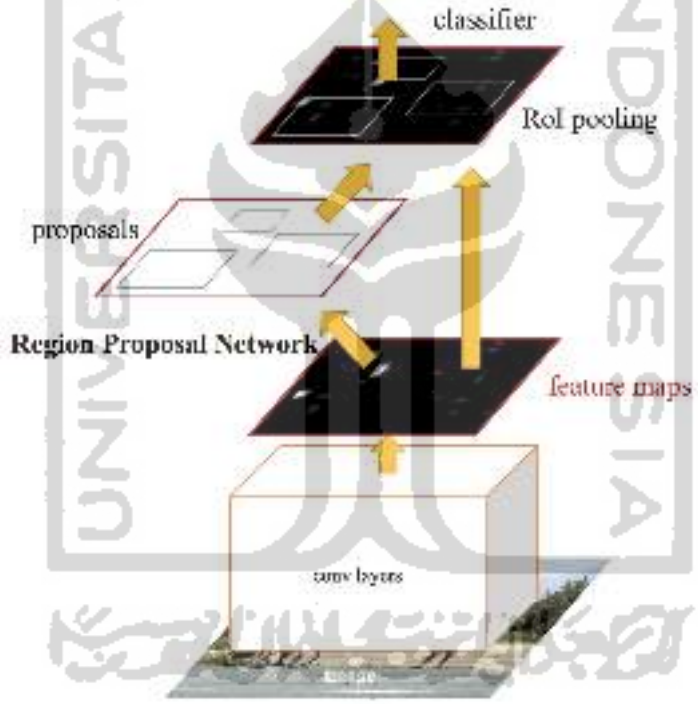
Fast R-CNN adalah sebuah metode yang sama dengan R-CNN dengan beberapa perbaikan yang telah dilakukan seperti pada Gambar 2.5 . Fast R-CNN mengubah input yang dimasukkan menjadi image untuk diolah di layer konvolusi yang menghasilkan feature map. Dari feature map ini kemudian diidentifikasi region proposal dan membungkusnya menjadi sebuah square menggunakan RoI pooling dan diserahkan ke fully connected layer. Fast R-CNN juga bekerja lebih cepat dari R-CNN. Namun Fast R-CNN juga memiliki kekurangan yaitu, pada saat melakukan pengujian region proposal memperlambat pekerjaan algoritma jika dibandingkan dengan tidak menggunakan region proposal. Oleh karena itu, region proposal menjadi hambatan dalam algoritma Fast R-CNN (Girshick, 2015).



Gambar 2. 5 *Fast* R-CNN
Sumber. (Girshick, 2015)

2.2.10 *Faster* R-CNN

Faster R-CNN adalah sebuah metode yang sama dengan Fast R-CNN yang telah disempurnakan. Sama dengan Fast R-CNN, citra merupakan input yang dimasukkan kedalam convolution layer untuk menghasilkan feature map seperti pada Gambar 2.6. Faster R-CNN menggunakan jaringan lain pada feature map untuk mengidentifikasi region proposal, jaringan terpisah digunakan untuk memprediksi region proposal. Region proposal yang dihasilkan nanti akan diproses menggunakan RoI pooling layer yang kemudian dilakukan klasifikasi pada citra dengan region proposal final dan memprediksi nilai pada bounding box (Shaoqing Ren, 2016).



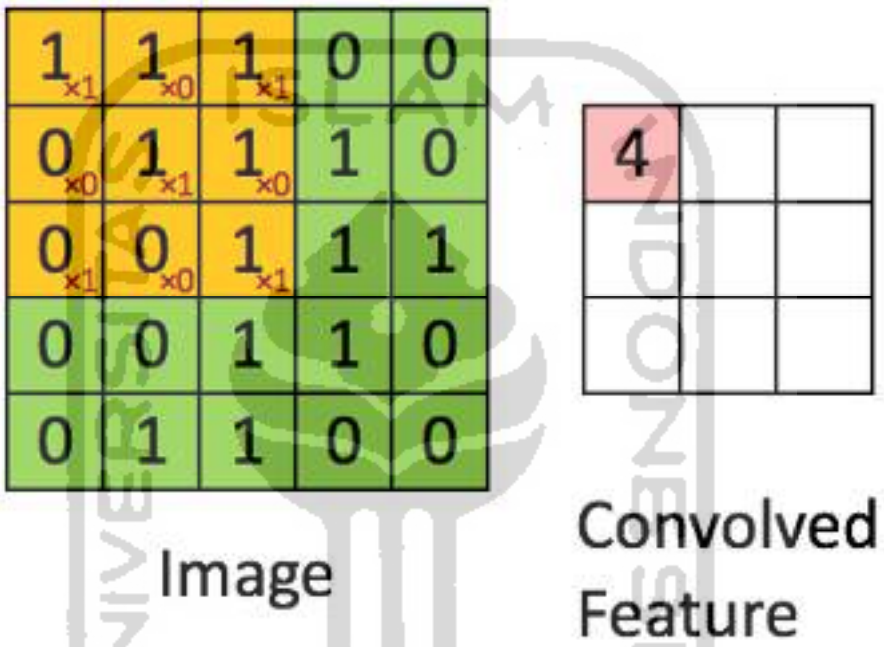
Gambar 2. 6 *Faster* R-CNN
Sumber. (Shaoqing Ren, 2016)

2.2.11 Training Neural Network

Tahap *training Neural Network* adalah tahap dimana sebuah jaringan dilatih untuk mempelajari suatu pola yang diharapkan dapat menghasilkan suatu model yang nantinya dapat diaplikasikan pada pengenalan deteksi objek sesuai dengan yang diharapkan dengan tingkat error yang rendah.

2.2.12 Konvolusi

Proses selanjutnya setelah gambar berhasil dimasukkan pada model adalah proses konvolusi. Proses konvolusi adalah proses utama yang mendasari sebuah CNN. Konvolusi merupakan teknik untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang ketiga. Pada Gambar 2. 7 merupakan contoh dalam proses konvolusi citra.



Gambar 2. 7 Ilustrasi proses konvolusi

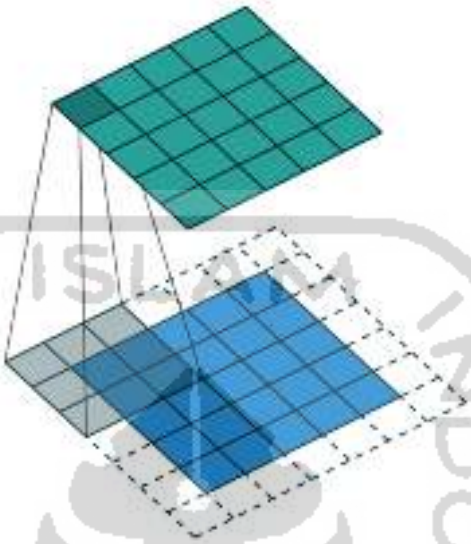
Sumber. (Saha, 2018)

Pada penelitian ini dua deret yang dimaksudkan terdapat pada citra masukan/*input* dan pada filter yang keduanya memiliki deret angka berupa matriks. Pada *input* memiliki deret angka yang didapat dari kedalaman dari warna yang terdapat pada masing-masing pixels. Sedang pada filter deret angka disesuaikan dengan kebutuhan penelitian. Terdapat beberapa jenis filter diantaranya operasi *identify, edge detection, Gaussian blur, sharpen box blur,* dan sebagainya (T.M Zaini, 2014).

2.2.13 Padding

Padding adalah sebuah Teknik konvolusi yang menambahkan beberapa piksel diluar dari citra. Teknik ini bertujuan agar citra masukan yang dikonvolusi tidak mengecil ukurannya

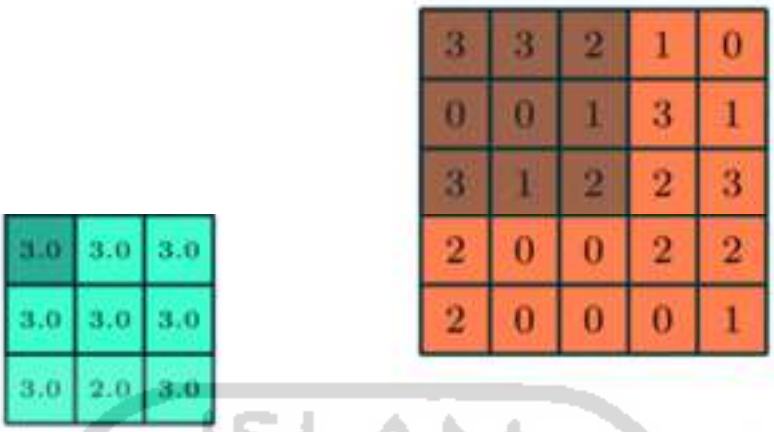
setelah dilakukan Teknik konvolusi pada citra. Jika terdapat citra dengan ukuran 5x5 piksel dengan ditambah padding disekitarya 1 piksel maka ukuran hasil konvolusi dari citra tersebut tetap 5x5 piksel. Apabila tidak menggunakan padding maka citra masukan dengan ukuran 5x5 piksel akan direduksi menjadi 3x3 piksel seperti pada Gambar 2.8 (Saha, 2018).



Gambar 2. 8 Ilustrasi proses konvolusi menggunakan padding
Sumber. (Saha, 2018)

2.2.14 Pooling

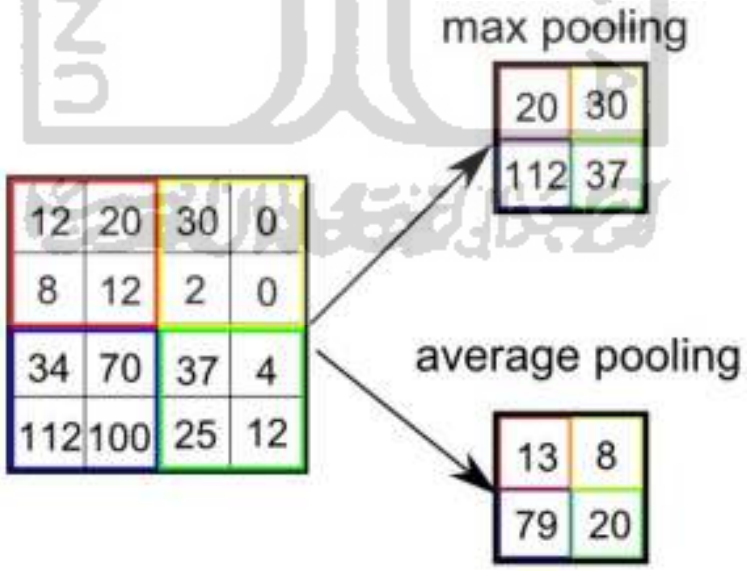
Pooling adalah *layer* yang bertanggung jawab untuk mengurangi ukuran spasial dari fitur konvolusi. *Pooling* dilakukan untuk mengurangi komputasi yang diperlukan untuk memperoses data melalui pengurangan dimensi citra yang diilustrasikan pada Gambar 2.9. Untuk kebutuhan lebih lanjut, *pooling* berguna untuk mengekstraksi fitur dominan yang rotasi dan invariant posisi, sehingga proses *training* model berjalan dengan efektif.



Gambar 2. 9 pooling 3x3 pada citra 5x5

Sumber. (Saha, 2018)

Ada dua jenis *pooling* yaitu *Max pooling* dan *average pooling*. *Max pooling* mengembalikan nilai maksimum dari bagian gambar yang dicakup oleh filter. Sedang *average pooling* mengembalikan rata-rata semua nilai dari bagian gambar yang dicakup oleh filter seperti pada Gambar 2.10. Pada penelitian ini digunakan *max pooling* dengan ukuran filter/kernel 3x3 (Saha, 2018).



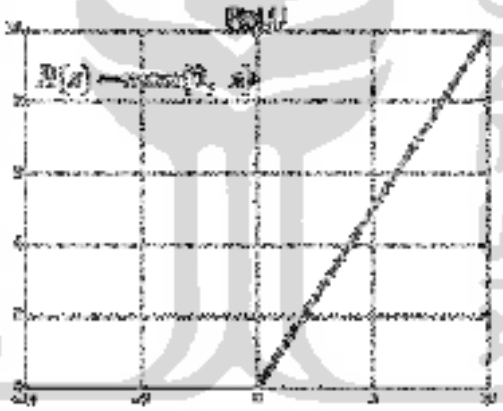
Gambar 2. 10 Jenis pada pooling

Sumber. (Saha, 2018)

2.2.15 Activation Function

Activation Function atau fungsi aktifasi adalah sebuah fungsi yang digunakan untuk mendapatkan output berupa *node. Activation Function* biasa juga dikenal sebagai *Transfer Function. Activation Function* biasa digunakan untuk menentukan keluaran dari *neural network* seperti *yes* atau *no. Activation Function* menghasilkan *values* antara 0 sampai 1 atau -1 sampai 1 tergantung jenis fungsi yang digunakan.

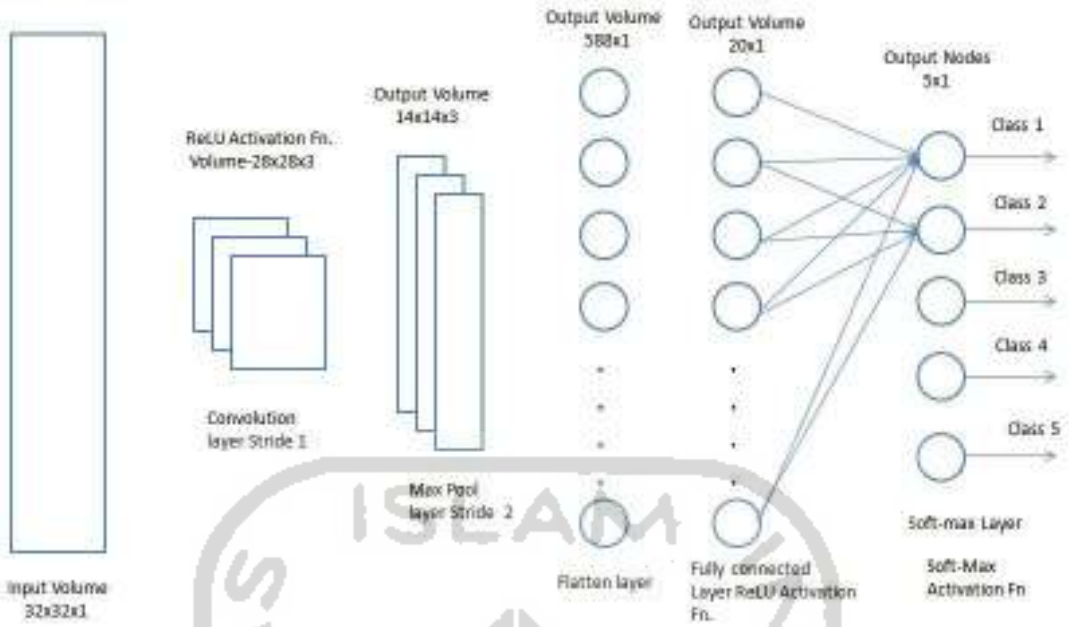
Activation function dibedakan menjadi 2 tipe yaitu *linear* dan *non-linear. Linear activation function* merupakan fungsi yang membentuk garis lurus atau *linear.* Keluaran dari fungsi ini tidak terbatas di antara rentang apapun. *Non-linear activation function* merupakan fungsi yang paling sering digunakan. Fungsi ini membuat model lebih mudah menyamaratakan atau adaptasi model dengan data yang bervariasi dan untuk membedakannya dengan hasil keluaran. ReLU merupakan fungsi aktivasi yang paling sering digunakan dan memiliki grafik seperti pada Gambar 2.11. ReLU digunakan di hampir semua proyek *convolutional neural network* atau *deep learning* (Saha, 2018).



Gambar 2. 11 Grafik pada ReLU
Sumber. (Saha, 2018)

2.2.16 Fully Connected Layer

Fully connected layer merupakan layer terakhir pada jaringan. Layer ini menghasilkan keluaran yang mengambil representasi yang telah dipelajari pada konvolusi dan *max pooling* yang kemudian akan dihasilkan sebuah prediksi seperti yang digambarkan pada Gambar 2.12. Secara singkat *fully connected layer* merupakan layer pada jaringan yang memuat *nodes* yang "menyala" ketika pola tertentu diamati (Saha, 2018).



Gambar 2. 12 Ilustrasi *fully connected layer*

Sumber. (Saha, 2018)

2.2.17 Learning Rate

Learning rate merupakan salah satu *hyper-parameter* yang penting dalam melakukan *training* pada *neural networks*. Pada *neural networks* proses *training* dilakukan dengan mengoptimalkan penurunan gradien. Parameter *learning rate* memberi tahu system optimasi seberapa jauh dalam bergerak. Jika *learning rate* rendah, maka proses *training* menjadi lebih realistis tetapi optimasi membutuhkan waktu yang lebih lama karena *steps* terhadap fungsi *loss* sangat kecil (Surmenok, 2017).

2.2.18 Batch Size

Batch size adalah salah satu pemeran utama dalam *training neural network. Batch size* merupakan jumlah sampel yang digunakan untuk melatih model sebelum memperbarui model yang telah dilatih. *Batch size* memiliki dampak pada akurasi model yang dibangun, serta berdampak pada kinerja dari sebuah proses *training.* Ketika jaringan semakin besar, ukuran *batch* yang dapat dijalankan pada GPU menjadi lebih kecil (Haleva, 2020).

2.2.19 Confusion Matrix

Confusion matrix adalah sebuah metode pengukuran klasifikasi dengan menggunakan tabel probabilitas. Tabel probabilitas umumnya dijalankan untuk menguji sebuah dataset. Pada

kolom mewakili dari data benar sedang pada baris mewakili data yang diuji (Sverdlov, 2018). Dalam penggunaannya, terdapat istilah dalam *confusion matrix* yaitu, *True Positive, False Positive, True Negative, False Positive.* TP merupakan nilai yang terdeteksi benar, FP merupakan nilai salah yang terdeteksi benar. TN merupakan nilai salah yang terdeteksi benar, sedangkan FN merupakan nilai benar yang terdeteksi salah. Persamaan *confusion matrix* ditunjukkan pada Persamaan (2.1).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2.2.20 *Python*

Python adalah sebuah bahasa pemrograman interpretatif multiguna yang berorientasi objek, dengan semantik yang dinamis. Kapabilitas dan kemampuan struktur data yang tinggi dikombinasi dengan fungsionalitas *dynamic typing* dan *dynamic binding* membuat *python* sangat cocok digunakan dalam *RAD. Python* juga cocok untuk bahasa *scripting* atau *glue* untuk menghubungkan komponen yang ada (python.org, n.d.).

2.2.21 *Tensorflow*

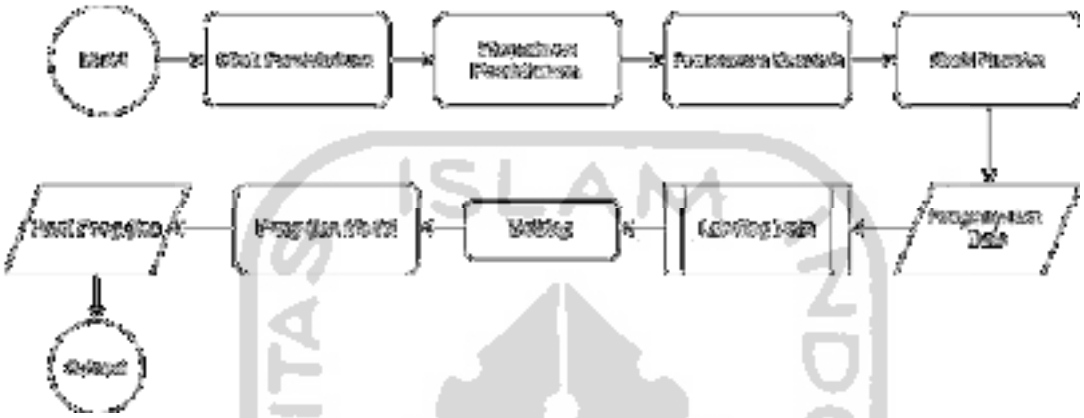
Tensorflow adalah platform terbuka yang *end-to-end* untuk *Machine Learning.* Platform ini memiliki alat yang fleksibel, terdapat banyak *library* yang bisa dimanfaatkan dan komunitas yang memungkinkan bagi para peneliti untuk membangun dan mengembangkan *Machine Learning. Tensorflow* dibangun oleh Google dalam organisasi peneliti Mesin Cerdas Google. *Tensorflow* menggunakan aljabar komputasi pengoptimalan kompilasi yang mempermudah perhitungan matematis (tensorflow.org, n.d.).

BAB III
METODOLOGI PENELITIAN

3.1 Perancangan

3.1.1 Perancangan Penelitian

Gambar 3.1 menunjukan tahapan yang dilakukan pada penelitian ini.



Gambar 3. 1 Alur pada penelitian

Proses-proses yang dilakukan pada penelitian ini seperti pada Gambar 3.1 akan dijelaskan pada sub bab selanjutnya secara lebih mendetail.

3.1.2 Studi Pendahuluan

Studi pendahuluan dilakukan untuk mencari referensi-referensi yang terkait dengan penelitian. Dari studi penelitian dan membaca website referensi terkait, didapatkan bahwa teknologi AR memanfaatkan *image processing* untuk mengenali citra penanda sebelum menampilkan objek 3D. Pada salah satu *library* pengembang teknologi AR, Vuforia. Vuforia menggunakan cloud recognition dan menggunakan *target star rating* untuk mengenali objek pada citra. Tidak dijelaskan secara spesifik metode apa yang digunakan oleh pengembang tersebut. Teknik *target star rating* merupakan Teknik pengenalan fitur natural pada objek yang kemudian akan dikomparasikan dengan citra yang diambil dari kamera secara *real time. Star rating* memiliki skala dari 1 sampai 5. Dimana 1 *rate star* mewakili citra yang rendah akan fitur dan 5 *star rate* mewakili citra yang kaya kanan detail atau fitur.

3.1.3 Eksperimen Pendahuluan

Sebelum dilakukan penelitian, penulis telah melakukan eksperimen pendahuluan dengan menggunakan Vuforia untuk menguji apakah citra LKS dapat dideteksi dengan baik oleh

alatnya. Setelah dilakukan pengujian, didapatkan hasil dari 10 gambar yang diujikan hanya 1 citra yang dapat dikenali dengan *star rate 3.* Sedang 9 citra lainnya mendapat 0 *star rate,* yang artinya tidak mampu dikenali.

3.1.4 Perumusan Masalah

Dari permasalahan yang didapat dari studi pendahuluan dan eksperimen pendahuluan didapat beberapa rumusan masalah yaitu, metode yang digunakan pada pengembangan AR pada Vuforia belum bisa mengenali citra penanda pada LKS. Ini dipengaruhi karena kertas buram tidak mencetak gambar dengan baik dan terlihat samar, sehingga objek citra sulit dikenali.

3.1.5 Studi Pustaka

Setelah merumuskan masalah yang ditemui, penulis melakukan studi Pustaka terkait dengan penelitian ini. Dari studi pustaka yang telah dilakukan, terdapat beberapa metode yang digunakan untuk mengenali sebuah objek citra. Dalam masa studi pustaka peneliti menemukan metode deep learning yang bisa diterapkan untuk pengenalan citra, yakni *Convolutional Neural Network* (CNN). Tidak berhenti disitu, penulis Kembali melakukan studi pustaka dan menemukan bahwa terdapat metode yang lebih cepat dalam pengenalan objek citra yaitu, *Faster Region based Convolutional Network* (*Faster* R-CNN). Metode ini menggunakan konsep yang hampir sama dengan CNN, namun dengan tambahan pengenalan wilayah objek citra dan beberapa modifikasi arsitektur yang memungkinkan *faster* R-CNN bekerja lebih cepat.

3.1.6 Pengumpulan Data

3.1.6.1 Karakter Data

Jenis data yang digunakan pada penelitian ini adalah data primer. Pada penelitian ini menggunakan data yang diperoleh dari buku LKS yang difoto setiap objeknya menggunakan kamera *smartphone* dengan resolusi 13 Mega piksel dengan hasil seperti pada Gambar 3.3. Dalam pengambilan gambar dilakukan tanpa menggunakan filter apapun dan dilakukan pada ruangan yang cukup penerangan.

Data pada penelitian kali ini dikelompokkan menjadi 9 jenis citra struktur organ pada makhluk hidup yakni sel hewan, sel tumbuhan, tumbuhan monokotil, tumbuhan dikotil, jantung, syaraf, otot polos, otot lurik, dan otot jantung. Kesembilan jenis citra yang digunakan, diambil dari buku LKS Biologi kelas XI semester 1 dengan penerbit Intan Pariwara (Wigawati

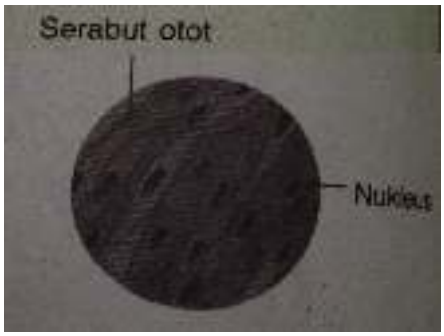
Hadi, 2017). Dari citra yang diambil tersebut dilakukan beberapa manipulasi data dengan memutar citra dan melakukan proses blurring pada citra seperti pada Gambar 3.2.



(a) (b) (c)

Gambar 3. 2 Contoh manipulasi data. (a) Gambar asli, (b) Manipulasi melalui proses blurring, (c) Manipulasi dengan merotasi citra

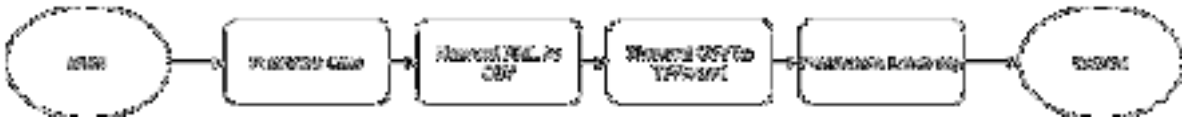
Gambar-gambar yang diambil ini akan dijadikan data masukan pada proses *training* dan *testing* data yang nantinya akan menghasilkan sebuah model yang dapat digunakan pada deteksi objek pada citra. Data citra dibagi menjadi 3 kelompok yaitu data *train,* data *test,* dan data evaluasi. Perbandingan dari jumlah data citra yaitu 85% data *train*, 15% data *test,* dan 7% data evaluasi. Data *train* berisi 495 citra, data *test* berisi 90 citra, dan data evaluasi 45 citra. Pada setiap kelas yang ada di data *train* memiliki 55 citra, di data *test* terdapat 10 citra di setiap kelasnya, dan pada data evaluasi terdapat 5 citra di setiap kelasnya. Data *train* dan *test* nantinya akan dipanggil pada saat akan melakukan *training* untuk menghasilkan sebuah model. Selanjutnya model yang telah di *training* tadi di evaluasi dengan menggunakan data evaluasi.





Gambar 3. 3 Citra Media Edukasi LKS

3.2 *Labelling* Data

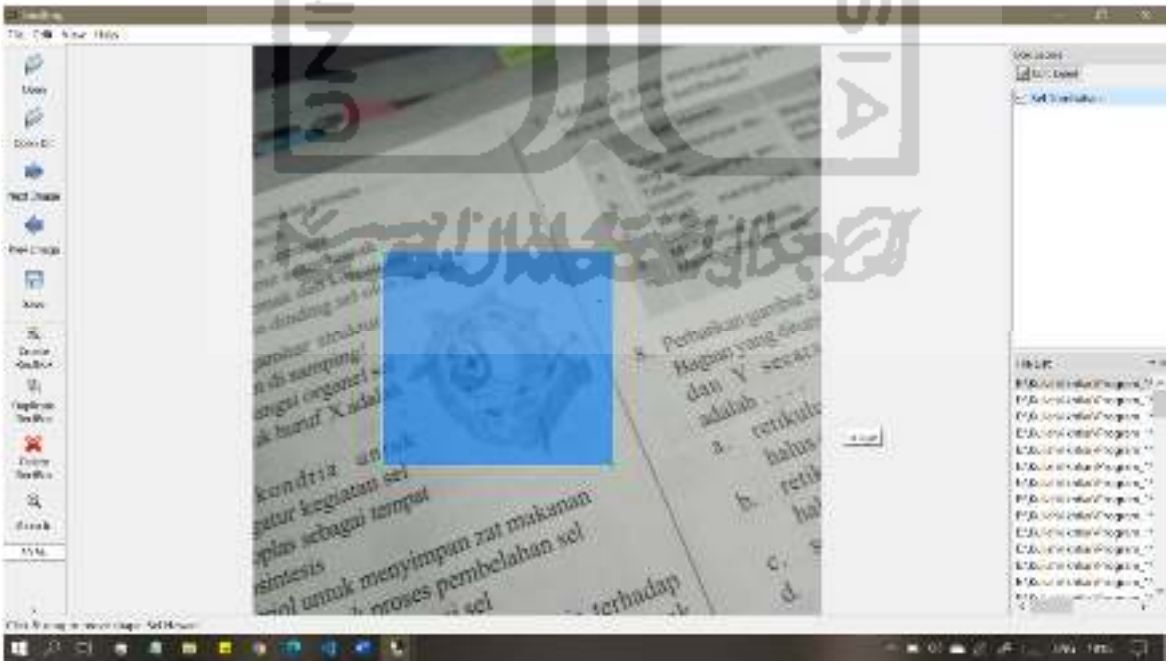


Gambar 3. 4 Tahap pada *Labelling* data

Pada proses pre-processing, urutan dari tahap-tahap yang dilakukan dapat dilihat pada Gambar 3.4 yang kemudian akan dijelaskan pada sub bab selanjutnya.

3.2.1 Pelabelan citra

Pelabelan citra merupakan tahap awal dalam sebuah pre-processing. Pada tahap ini dataset diberi label berupa 9 kelas yaitu dikotil, monokotil, otot polos, otot lurik, otot jantung, sel hewan, sel tumbuhan, jantung, dan syaraf. Di tahap ini dilakukan seleksi citra berdasar kelas dengan menggunakan aplikasi LabelImg seperti pada Gambar 3.5. Keluaran dari pelabelan citra ini adalah data letak objek citra beserta labelnya yang disimpan pada file XML.



Gambar 3. 5 Proses labeling pada citra

3.2.2 Konversi XML ke CSV

Setelah label dan posisi objek pada citra disimpan pada file XML, Langkah selanjutnya adalah mengkonversi file XML ke CSV. Ini perlu dilakukan mengingat file XML tadi hanya berisi data dari satu citra. Pengubahan file XML ke file CSV dilakukan agar data XML yang banyak tadi dijadikan satu dalam bentuk tabel agar lebih mudah untuk di proses.

3.2.3 Konversi CSV ke TFRecord

Langkah selanjutnya adalah konversi data CSV menjadi TFRecord. Tahap ini perlu dilakukan agar data tabel yang ada pada file CSV tadi dapat dibaca oleh *tensorflow*. Data TFRecord ini lah yang nantinya akan di proses untuk dilatih membentuk sebuah model yang dapat mengenali sebuah objek yang terdapat pada citra LKS.

3.2.4 Pembuatan Labelmap

Langkah selanjutnya sebelum melakukan tahap training adalah menyiapkan labelmap. Labelmap berfungsi mirip seperti id pada label yang digunakan pada saat konfigurasi jaringan sebelum memasuki tahap *training.*

3.3 *Training Process*

3.3.1 Konfigurasi

Memasuki tahap pertama dalam proses *training* adalah tahap konfigurasi. Pada tahap ini penulis akan mengatur parameter-parameter seperti *learning rate* dan *batch size* yang menunjang dalam proses *training.* Disini penulis memperkirakan konfigurasi yang tepat untuk membentuk model yang handal dalam mengenali objek pada citra LKS.

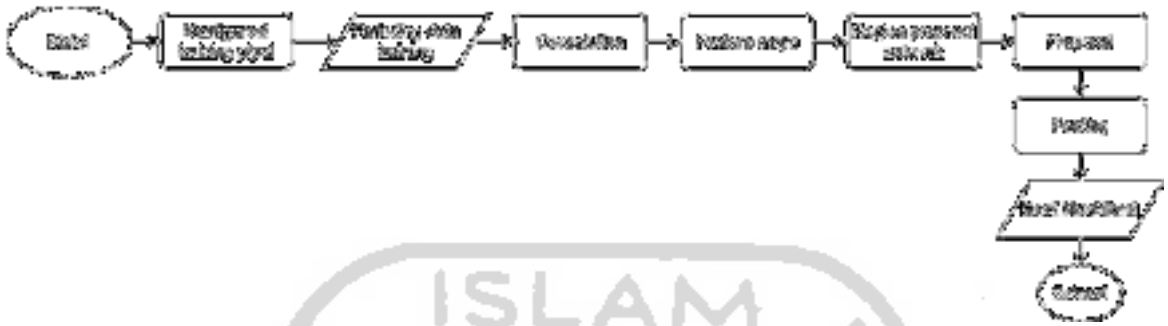
3.3.2 *Load* Data

Tahap selanjutnya adalah memanggil data. Pada tahap *pre-processing* dijelaskan bahwa dataset yang telah diberi label disimpan dan di konversikan hingga menjadi data TFRecord. Data yang telah dikonversi menjadi TFRecord dipanggil kemudian dilatih sedemikian rupa dengan konfigurasi yang telah di set agar menghasilkan model pengenalan objek yang baik.

3.3.3 Deteksi Objek

Pada tahap ini akan ditentukan bagian mana saja pada tiap piksel yang memiliki pola sesuai dengan citra pada kelas yang telah ditentukan ditentukan. Dalam penelitian ini, peneliti

menguji citra dari masing-masing kelas yang ada yaitu kelas monokotil,dikotil, otot polos, otot lurik, otot jantung, jantung, syaraf, sel hewan dan sel tumbuhan. Urutan dari tahap yang dilakukan pada proses training dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Diagram alir *training process*

3.4 Pengujian Data

Setelah masa *training* telah selesai, Langkah selanjutnya adalah proses pengujian citra. Seperti yang telah dijelakan pada bab 2, pengujian dalam penelitian ini menggunakan metode *confusion matrix.* Dengan menggunakan metode ini, pengujian dilakukan dengan menggunakan tabel prediksi yang kemudian dihitung akurasinya berdasarkan dengan hasil uji ini. Pada penelitian ini, dilakukan pengujian data dengan menggunakan total 45 citra yang dibagi 5 citra untuk setiap kelasnya.

3.5 Detection Output

Detection Output merupakan hasil akhir dari deteksi organ makhluk hidup pada citra buku LKS. Untuk menjalankan dan menguji hasil *training* model yang dilakukan, diperlukan program untuk menjalankan model (*frozen inference graph*) dan menyimpan hasil gambar yang telah diberikan deteksi berupa kotak warna beserta kelas dan nilai presentasinya.

BAB IV
HASIL DAN PEMBAHASAN

4.1 *Pre-processing* Data

Setelah pengumpulan data dilakukan, tahap yang dilakukan selanjutnya adalah tahap *preprocessing* data. Tahap ini terdiri dari pelabelan, konversi, dan pembuatan Label map.

4.1.1 Pelabelan Citra

Pelabelan citra merupakan tahap awal pada *preprocessing* data, yang dimana data masukan diberikan label atau pengenal dengan tujuan untuk menyimpan informasi. Pelabelan citra juga digunakan untuk membantu dalam proses *training.* Karena citra diambil dengan mengambil foto buku LKS, maka perlu dilakukan pelabelan. Pelabelan dilakukan dengan memberi tanda posisi objek berada, lalu memberi label kelas atau jenis pada objek tersebut. Alat yang digunakan dalam pelabelan adalah LabelImg seperti pada Gambar 4.1 . Pelabelan dilakukan secara manual pada 495 citra *training* dan 90 citra *test*.



Gambar 4. 1 Proses Pelabelan Objek Citra

Hasil pemberian tanda dan label pada objek citra disimpan dalam bentuk file XML. Setiap citra akan memiliki file XML nya masing-masing dan file ini menyimpan informasi posisi objek pada citra beserta dengan label nya seperti pada Gambar 4.2.

```
<annotation>
   <folder>train</folder>
   <filename>20191104_200205.jpg</filename>
   <path>E:\Kuliah\Ikhtiar\Program_1\images\train\2019
1104_200205.jpg</path>
   <source>
      <database>Unknown</database>
   </source>
   <size>
      <width>800</width>
      <height>600</height>
      <depth>3</depth>
   </size>
   <segmented>0</segmented>
   <object>
      <name>Sel Tumbuhan</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
      <bndbox>
         <xmin>171</xmin>
         <ymin>71</ymin>
         <xmax>731</xmax>
         <ymax>550</ymax>
      </bndbox>
   </object>
</annotation>
```

Gambar 4. 2 Isi dari file XML setelah pelabelan

4.1.2 Konversi Data XML ke CSV

Setelah melakukan pelabelan pada semua citra di dataset, langkah selanjutnya adalah mengubah atau mengkonversi file XML terpisah tadi menjadi satu data kesatuan dengan format CSV. Dengan kode program seperti pada Gambar 4.3.

```
import os
import glob
import pandas as pd
import xml.etree.ElementTree as ET

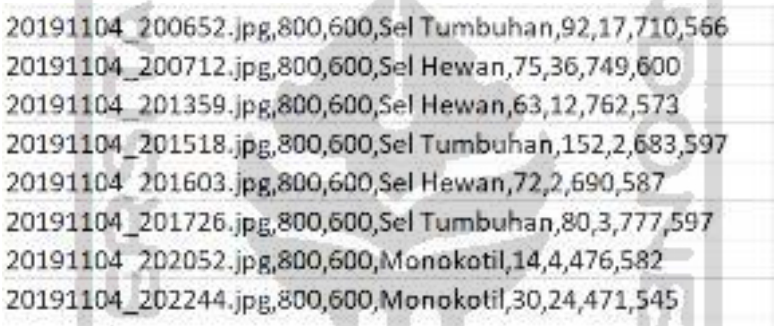
def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                     int(root.find('size')[0].text),
                     int(root.find('size')[1].text),
                     member[0].text,
                     int(member[4][0].text),
                     int(member[4][1].text),
                     int(member[4][2].text),
                     int(member[4][3].text)
                     )
            xml_list.append(value)
    column_name = ['filename', 'width', 'height',
'class', 'xmin', 'ymin', 'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list,
columns=column_name)
    return xml_df

def main():
    for folder in ['train', 'test']:
        image_path = os.path.join(os.getcwd(),
('images/' + folder))
        xml_df = xml_to_csv(image_path)
```

```
xml_df.to_csv(('images/'+folder+'_labels.csv'),
index=None)
        print('Successfully converted xml to csv.')
main()
```

Gambar 4. 3 Program Konversi dari XML ke CSV

Tahap konversi dari file XML ke CSV init perlu dilakukan karena file XML yang terpisah-pisah di gabungkan menjadi satu tabel pada file CSV seperti pada Gambar 4.4. Jadi dengan konversi ini memudahkan komputer untuk mengambil data posisi dan label dari objek citra yang sebelumnya telah diberikan label.



Gambar 4. 4 Hasil konversi dari XML ke CSV

4.1.3 Konversi *Dataset* CSV ke TFRecord

Setelah proses konversi dataset XML ke CSV berhasil dilakukan, langkah selanjutnya adalah mengkonversi dataset CSV ke TFRecord agar dataset yang diubah menjadi CSV tadi dapat dibaca oleh tensorflow. Pada konversi TFRecord telah dilengkapi oleh TF-API sehingga pada penelitian perlu untuk mengganti label data sesuai dengan label yang digunakan pada penelitian seperti yang ditunjukkan pada Gambar 4.5.

```
def class_text_to_int(row_label):
    if row_label  == 'Sel Hewan':
        return 1
    elif row_label  == 'Sel Tumbuhan':
        return 2
```

```
    elif row_label  == 'Monokotil':
        return 3
    elif row_label  == 'Dikotil':
        return 4
    elif row_label  == 'Otot Polos':
        return 5
    elif row_label  == 'Otot Lurik':
        return 6
    elif row_label  == 'Otot Jantung':
        return 7
    elif row_label  == 'Saraf':
        return 8
    elif row_label  == 'Jantung':
        return 9
    else:
        return 0
```

Gambar 4. 5 Mendefinisikan label pada TFRecord

Hasil dari proses ini adalah file train.record dan test.record. File ini akan dibaca oleh *tensorflow* pada saat melakukan *training* dan *test* pada data citra buku LKS.

4.1.4 Label Map

Pada proses ini, dibuat petlabel yang sesuai dengan pelabelan pada citra. Pada peta label terdiri 9 kelas yang digunakan pada penelitian ini yakni dikotil, monokotil, otot polos, otot lurik, otot jantung, syaraf, sel hewan, sel tumbuhan dan jantung seperti pada Gambar 4.6 yang akan disimpan pada file "labelmap.pbtxt". Label map memiliki fungsi seperti seperti id pada label kelas yang nantinya akan dibutuhkan pada saat konfigurasi pipeline.

```
item {
    id: 1
    name: 'Sel Hewan'
}
item {
    id: 2
```

```
    name: 'Sel Tumbuhan'
}
item {
    id: 3
    name: 'Monokotil'
}
item {
    id: 4
    name: 'Dikotil'
}
item {
    id: 5
    name: 'Otot Polos'
}
item {
    id: 6
    name: 'Otot Lurik'
}
item {
    id: 7
    name: 'Otot Jantung'
}
item {
    id: 8
    name: 'Saraf'
}
item {
    id: 9
    name: 'Jantung'
}
```

Gambar 4. 6 Isi pada *labelmap.*pbtxt

4.2 *Training* Data

Tahap *training* data dilakukan untuk melatih jaringan yang nantinya akan menghasilkan sebuah model jaringan. Pada penelitian ini menggunakan Bahasa pemrograman *Python* versi 3.6.8 dan *Tensorflow* versi 1.14.0. Algoritma pengenalan objek menggunakan *faster* R-CNN.

4.2.1 Konfigurasi

Tahap pertama dalam proses *training* adalah melakukan konfigurasi pada *hyper parameter* yang dapat diubah seperti *learning rate* dan *batch size.* Pada penelitian ini, penulis mengatur hyper parameter learning rate 0,0001 atau 0,1%. Pada hyper parameter lainnya yaitu *batch size,* penulis mengaturnya menjadi 1. Setelah kedua *hyper parameter* ini diatur dalam file konfigurasi seperti pada Gambar 4.7, selanjutnya adalah tahap *load data.*

```
train_config: {
  batch_size: 1
  optimizer {
    momentum_optimizer: {
      learning_rate: {
        manual_step_learning_rate {
          initial_learning_rate: 0.0001
          schedule {
            step: 900000
            learning_rate: .0001
          }
          schedule {
            step: 1200000
            learning_rate: .0001
          }
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
```

Gambar 4. 7 Bagian yang dapat diganti dari konfigurasi *Faster R-CNN*

4.2.2 Load Data

Tahap load data akan dipanggil dataset citra yang telah dilabeli sebelumnya. Dataset citra tersebut sebelumnya disimpan dalam format XML dijadikan satu dalam satu file CSV yang kemudian diubah menjadi format TFRecord agar bisa dibaca oleh *tensorflow*. File TFRecord yang dipanggil adalah train.record yang berisi 495 data citra dan test.record yang berisi 90 data citra. Langkah selanjutnya adalah melakukan *training* seperti pada Gambar 4.8.

```
train_input_reader: {
  tf_record_input_reader {
    input_path: "E:/Kuliah/Ikhtiar/Program_1/train.record"
  }
  label_map_path:
"E:/Kuliah/Ikhtiar/Program_1/Training/labelmap.pbtxt"
}

eval_config: {
  metrics_set: "coco_detection_metrics"
  num_examples: 38
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "E:/Kuliah/Ikhtiar/Program_1/test.record"
  }
  label_map_path:
"E:/Kuliah/Ikhtiar/Program_1/Training/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
}
```

Gambar 4. 8 Pemanggilan data pada proses *training*

4.2.3 Deteksi Objek

Tahap klasifikasi akan ditentukan bagian mana saja pada tiap piksel yang memiliki pola sesuai dengan citra pada kelas yang telah ditentukan ditentukan. Dalam penelitian ini, peneliti menguji citra dari masing-masing kelas yang ada yaitu kelas monokotil, dikotil, otot polos, otot lurik, otot jantung, jantung, syaraf, sel hewan dan sel tumbuhan. Setelah didapat proposal target layer dan fitur map dari proses konvolusi pada proses training, fitur map melewati layer dari ResNet mengikuti rata-rata dari pooling dari dimensi spasial. Fitur vector kemudian melewati fully connected layer, bbox_pred_net dan cls_score_net. Cls_score_net layer memproduksi skor kelas dari setiap bounding box. Bbox_pred_net layer menghasilkan kelas spesifik dari koefisien regresi bounding box yang dikombinasikan dengan bounding box original untuk menghasilkan proposal target layer untuk membuat bounding box final.



Gambar 4. 9 Proses *training* pada *Tensorboard*

4.2.4 Total *Loss*

Pada Gambar 4.10 ditunjukkan grafik dari total loss yang didapat dari proses training sampai dengan selesai sesuai dengan jumlah epoch yang dilakukan. Pada Gambar 4.9 ditunjukkan grafik yang naik yang menggambarkan kenaikan akurasinya, sedang pada Gambar 4.10 ditunjukkan grafik menurun yang menggambarkan *total loss* dari model yang dilatih. *Total loss* yang diperoleh dari proses ini adalah 30%.



Gambar 4. 10 Total *loss* pada saat melakukan *training*

4.2.5 Export Model

Pada saat melakukan proses *training*, akan terbentuk file *checkpoint* yang dibuat secara otomatis oleh *Tensorflow* yang bertujuan untuk menyimpan informasi proses *training* yang dilakukan. Setelah *loss* yang rendah dan proses *training* selesai dilakukan, maka proses selanjutnya adalah mengubah file *checkpoint* menjadi model yang siap untuk dipakai pada pengenalan dan deteksi objek citra dengan kode program seperti pada Gambar 4.11.

```
def main(_):
  pipeline_config                                    =
pipeline_pb2.TrainEvalPipelineConfig()
  with tf.gfile.GFile(FLAGS.pipeline_config_path, 'r') as
f:
    text_format.Merge(f.read(), pipeline_config)
  text_format.Merge(FLAGS.config_override,
pipeline_config)
  if FLAGS.input_shape:
    input_shape = [
        int(dim) if dim != '-1' else None
        for dim in FLAGS.input_shape.split(',')
    ]
```

```
  else:
    input_shape = None
  exporter.export_inference_graph(
      FLAGS.input_type,                  pipeline_config,
FLAGS.trained_checkpoint_prefix,
      FLAGS.output_directory, input_shape=input_shape,
      write_inference_graph=FLAGS.write_inference_graph)


if __name__ == '__main__':
  tf.app.run()
```

Gambar 4. 11 Program yang digunakan *extract* model yang telah di *training.*

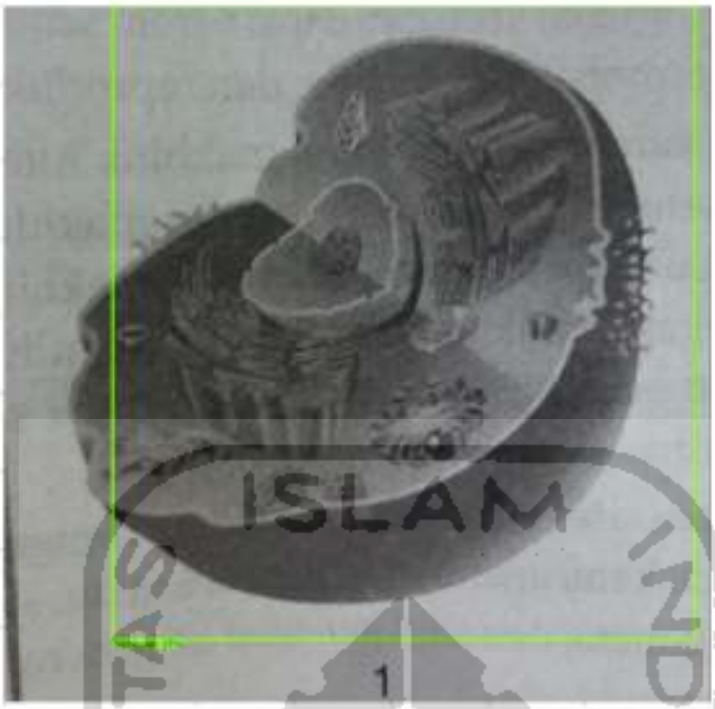
4.2.6 Model

Hasil akhir dari prose *training* adalah terbentuknya model yang siap dipakai untuk pendeteksian yang akan dilakukan pada tahap *testing*. Model yang digunakan adalah file *checkpoint* hasil dari proses *training* yang telah di export dengan nama file "frozen_inference_graph.pb". File ini nanti akan dipanggil ketika menjalankan program deteksi objek yang ada.

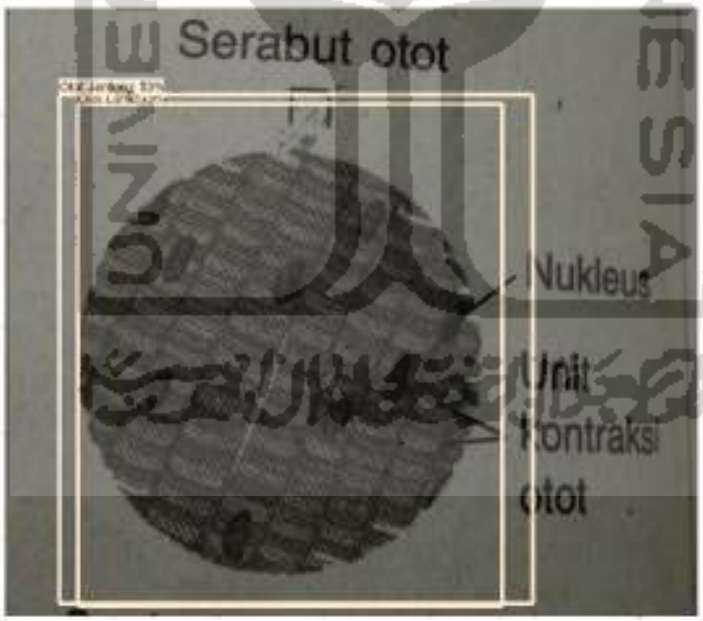
4.3 *Testing*

Hasil uji coba yang telah dilakukan pada model yang telah di *training* akan siap dipakai untuk dilakukan pendeteksian terhadap objek citra buku LKS dengan kotak berwarna yang menandakan bahwa objek tersebut terdeteksi beserta dengan nama kelas dan presentase akurasinya.

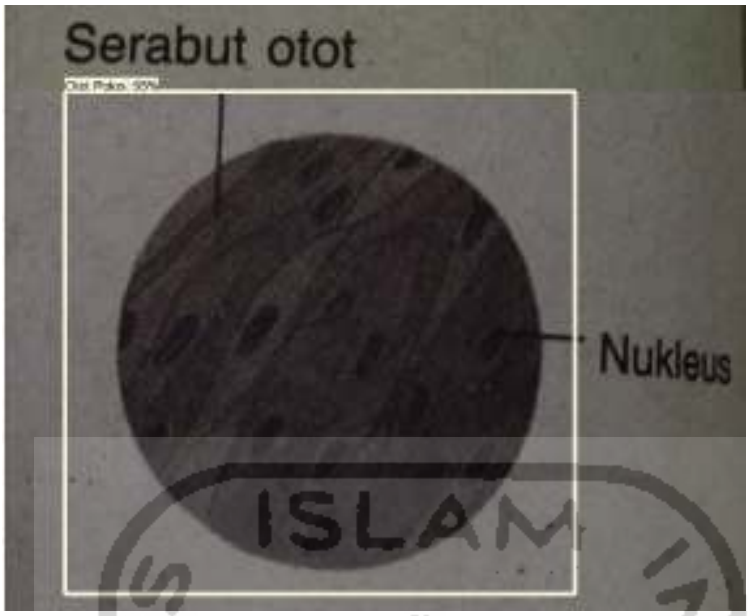
Pada box pada deteksi objek, terdapat nilai akurasi yang menyatakan kebenaran sebuah objek yang dideteksi dengan nama kelasnya. Nilai tersebut didapat dari sebuah kelas yang bernama *graph(),* kelas tersebut bertugas untuk mengkomputasikan nilai keluaran pada *neural network* yang mempresentasikan *dataflow. Graph* yang dimaksud adalah grafik yang telah dilatih sebelumnya berupa file *checkpoint* yang diekspor ke *graph inference.* Selanjutnya kelas akan memanggil *tensor* berdasarkan variabel yang mengembalikan data tersebut "*detection_scores:0",* skor ini diinisialisasikan dengan angka 0 agar hasil presentase dimulai dari 0% hingga 100%.



Gambar 4. 12 Hasil citra yang berhasil dideteksi



Gambar 4. 13 Hasil citra yang berhasil dideteksi



Gambar 4. 14 Hasil deteksi pada citra LKS gelap



Gambar 4. 15 Hasil deteksi citra pada LKS pada sudut pandang samping

Berdasarkan Gambar 4.12 dan Gambar 4.13, diperoleh probabilitas kemiripan sebesar 97% pada citra sel hewan dan diperoleh probabilitas kemiripan sebesar 79% pada citra otot lurik yang diambil dengan sudut pandang lurus.Sedangkan pada Gambar 4.14 didapat probabilitas kemiripan sebesar 95% pada citra otot polos dengan kondisi citra gelap dan blur.

Pada Gambar 4.15 didapatkan probabilitas kemiripan sebesar 99% pada citra jantung yang diambil dari sudut samping dari buku LKS. Kecepatan deteksi dari setiap image membutuhkan waktu sekitar 36 detik. Hasil deteksi yang dilakukan ada citra masukan menghasilkan nilai probabilitas kemiripan yang cukup tinggi.

4.4 Evaluasi

Pada tahap evaluasi, jaringan yang telah dilatih selanjutnya diuji pada citra organ makhluk hidup yang ada pada LKS dengan menggunakan tabel seperti pada Tabel 4.1. Tahap evaluasi menggunakan 45 data citra yang setiap kelasnya terdapat 5 citra.

Tabel 4. 1 Tabel Pengujian

Aktual	Prediksi									
	Dikotil	Monokotil	O. Polos	O. Lurik	O. Jantung	S. Hewan	S. Tumb	Jantung	Syaraf	Tdk Terdeteksi
Dikotil	5	-	-	-	-	-	-	-	-	-
Monokotil	-	5	-	-	-	-	-	-	-	-
O.Polos	-	-	5	-	-	-	-	-	-	-
O.Lurik	-	-	-	4	-	-	-	-	-	1
O.Jantung	-	-	-	-	5	-	-	-	-	-
S.Hewan	-	-	-	-	-	5	-	-	-	-
S.Tumb	-	-	-	-	-	-	5	-	-	-
Jantung	-	-	-	-	-	-	-	5	-	-
Syaraf	-	1	-	-	-	-	-	-	-	4

$$\frac{7(5)+4}{9(5)} \times 100\% = \frac{39}{45} \times 100\% = 86{,}66\% = 87\% \quad (4.1)$$

Berdasarkan dari hasil pengujian pada Tabel 4.1 yang telah dilakukan terhadap 9 kelas yang ada, didapat hasil akurasi sebesar 87%. Hal ini terjadi pada gambar syaraf 4 diantaranya tidak dapat dikenali, 1 gambar lainnya dikenali sebagai kelas lain. Sedang pada gambar Otot Lurik terdapat 1 gambar yang tidak dapat dikenali.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Algoritma *Faster* R-CNN dapat diaplikasikan dengan baik pada deteksi dan pengenalan objek citra media edukasi LKS Biologi.
2. Tingkat akurasi yang didapat dari hasil deteksi dan pengenalan objek citra organ makhluk hidup LKS Biologi menggunakan *Faster* R-CNN cukup baik dengan akurasi sebesar 87%.
3. Deteksi dan pengenalan objek pada program ini berjalan dengan cukup baik pada kertas buram sehingga mungkin untuk diaplikasikan pada pengenalan objek teknologi AR.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, dapat diberikan beberapa saran sebagai berikut:

1. Menambahkan jumlah *dataset* yang kaya akan ragam objek untuk melatih objek dan agar akurasi yang didapat cukup tinggi
2. Melakukan *training* dengan step yang lebih banyak dan menambah jumlah *batch size* agar menghasilkan hasil akurasi model yang lebih tinggi.
3. Mengembangkan kembali pengenalan objek organ makhluk hidup pada citra digital LKS secara *realtime* dengan menggunakan webcam.
4. Menggunakan spesifikasi perangkat keras yang lebih tinggi dengan *Random Access Memory (RAM)* dengan standar 8GB atau diatasnya dan menggunakan *Graphic Processing Unit (GPU)* untuk mempercepat proses *training.*

DAFTAR PUSTAKA

Adam, A. G. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Biswas, B. (2019, 4 21). *What Is Augmented Reality*. Dipetik 11 2019, dari https://towardsdatascience.com/fundamentals-of-augmented-reality-ar-89ce986f70f8

Chris Wendler, D. A. (2019). Powerset Convolutional Neural Networks.

David Nagataries1), S. H. (t.thn.). Deteksi Objek pada Citra Digital Menggunakan Algoritma Genetika untuk Studi Kasus Sel Sabit.

Diego Olivia, M. A. (2019, 10). Metaheuristic Algorithms for Image Segmentation: Theory and Application. Switzerland: Springer. Dipetik 01 2020, dari https://en.wikipedia.org/wiki/Image

Fengliang Xu, K. F. (2003). Human Detection Using Depth and Gray Images.

Girshick, R. (2015). Fast R-CNN.

Haleva, R. (2020, January). *How to Break GPU Memory Boundaries Even with Large Batch Sizes*. Diambil kembali dari towardsdatascience.com: https://towardsdatascience.com/how-to-break-gpu-memory-boundaries-even-with-large-batch-sizes-7a9c27a400ce

Huang, J. (2019). Object Detection.

Huang, X. (2018). Convolutional Neural Networks In Convolution.

Jyoti Rao, S. J. (2019). GREYSCALE IMAGE AUTHENTICATION AND REPAIRING .

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification.

Kluwer, W. (2020). Artificial Intelligence. *Journal of Clinical Ophthalmology and Research*.

LABS, S. (2019, March 22). *Understanding Deep Learning: DNN, RNN, LSTM, CNN and R-CNN*. Diambil kembali dari Medium: https://medium.com/@sprhlabs/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff

Majid, A. (2007). Perencanaan Pembelajaran. Dalam A. Majid, *Perencanaan Pembelajaran* (hal. 177). Bandung: Remaja Rosdakarya.

Miguel A. Jaramillo Morán, J. Á. (1999). APPLICATIONS OF CELLULAR NEURAL NETWORKS (CNN) TO GREY SCALE IMAGE FILTERING.

Naveen Sundar Govindarajulu, J. H. (2018). Do Machine-Learning Machines Learn?

Nehme, A. (2018, May 25). *Understanding Convolutional Neural Networks*. Diambil kembali dari Toward Data Science: https://towardsdatascience.com/understanding-convolutional-neural-networks-221930904a8e

Parker, J. (2010). *Algorithms for Image Processing and Computer Vision Second Edition.* Canada: Wiley Publishing, Inc.

Penousal Machado, A. D. (2002). Learning to Colour Greyscale Images.

Python. (t.thn.). *What is Python?* Diambil kembali dari Python: https://www.python.org/doc/essays/blurb/

python.org. (t.thn.). *What is Python? Executive Summary*. Diambil kembali dari www.python.org.

RD. Kusumanto, A. N. (2011). PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB.

Saad Albawi, T. A. (2017). Understanding of a Convolutional Neural Network .

Saha, S. (2018, December 16). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Diambil kembali dari Toward Data Science: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Shaoqing Ren, K. H. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

SHARMA, S. (2017, September 6). *Activation Functions in Neural Networks*. Diambil kembali dari Towards Data Science: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Shuiwang Ji, W. X. (2013). 3D Convolutional Neural Networks for Human Action Recognition.

Steve Lawrence, M. I. (1997). Face Recognition: A Convolutional Neural-Network Approach.

Surmenok, P. (2017, 11 13). *Estimating an Optimal Learning Rate For a Deep Neural Network*. Diambil kembali dari towardsdatascience.com: https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0

Sverdlov, A. (2018). Confusion Matrix.

T.M Zaini, M. ,. (2014). Pengolahan Citra DIgital. Bandar Lampung: Aura Publishing.

Tensorflow. (t.thn.). *Why TensorFlow*. Diambil kembali dari Tensorflow: https://www.tensorflow.org/about

tensorflow.org. (t.thn.). *https://www.tensorflow.org/about*. Diambil kembali dari www.tensorflow.org.

Tianyi Liu, S. F. (2015). Implementation of Training Convolutional Neural Networks.

Trianto. (2010). Mendesain Model Pembelajaran Inovatif. Progresif Konsep, Landasan, dan Implementasi Pada Kurikulum Tingkat Satuan (KTSP) (Cetakan ke-5). Dalam Trianto, *Mendesain Model Pembelajaran Inovatif. Progresif Konsep, Landasan, dan Implementasi Pada Kurikulum Tingkat Satuan (KTSP) (Cetakan ke-5)* (hal. 111). Jakarta: Kencana Prenada Media Group.

TURNEY, B. (2007). Anatomy in a modern medical curriculum.

Vipul Sharma, R. N. (2019). Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition.

Wigawati Hadi, H. P. (2017). Biologi. Penerbit Intan Pariwra.

Yilun Chen, S. L. (2019). Fast Point R-CNN.

LAMPIRAN

Lampiran Script Konversi XML ke CSV

```
import os
import glob
import pandas as pd
import xml.etree.ElementTree as ET


def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                     int(root.find('size')[0].text),
                     int(root.find('size')[1].text),
                     member[0].text,
                     int(member[4][0].text),
                     int(member[4][1].text),
                     int(member[4][2].text),
                     int(member[4][3].text)
                     )
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class',
'xmin', 'ymin', 'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_df


def main():
    for folder in ['train', 'test']:
```

```
        image_path = os.path.join(os.getcwd(), ('images/' +
folder))
        xml_df = xml_to_csv(image_path)
        xml_df.to_csv(('images/'+folder+'_labels.csv'),
index=None)
        print('Successfully converted xml to csv.')

main()
```

Lampiran Script Konversi CSV ke TFRecord

```
from __future__ import division
from __future__ import print_function
from __future__ import absolute_import

import os
import io
import pandas as pd
import tensorflow as tf

from PIL import Image
from object_detection.utils import dataset_util
from collections import namedtuple, OrderedDict
import argparse

parser = argparse.ArgumentParser(description="GENERATE TF
RECORDS")
parser.add_argument('-i', '--image_dir', type=str,
required=True, help="Image Dir")
parser.add_argument('-o', '--output_path', type=str,
required=True, help='Output Path')
parser.add_argument('-c', '--csv_input', type=str,
required=True, help='CSV Input')
```

```
FLAGS = parser.parse_args()

#flags = tf.app.flags
#flags.DEFINE_string('csv_input', '', 'Path to the CSV input')
#flags.DEFINE_string('output_path', '', 'Path to output
TFRecord')
#flags.DEFINE_string('image_dir', '', 'Path to images')
#FLAGS = flags.FLAGS


# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label  == 'Sel Hewan':
        return 1
    elif row_label  == 'Sel Tumbuhan':
        return 2
    elif row_label  == 'Monokotil':
        return 3
    elif row_label  == 'Dikotil':
        return 4
    elif row_label  == 'Otot Polos':
        return 5
    elif row_label  == 'Otot Lurik':
        return 6
    elif row_label  == 'Otot Jantung':
        return 7
    elif row_label  == 'Saraf':
        return 8
    elif row_label  == 'Jantung':
        return 9
    else:
        return 0
```

```
def split(df, group):
    data = namedtuple('data', ['filename', 'object'])
    gb = df.groupby(group)
    return [data(filename, gb.get_group(x)) for filename, x in
zip(gb.groups.keys(), gb.groups)]


def create_tf_example(group, path):
    with                tf.io.gfile.GFile(os.path.join(path,
'{}'.format(group.filename)), 'rb')   as fid:
        encoded_jpg = fid.read()
    encoded_jpg_io = io.BytesIO(encoded_jpg)
    image = Image.open(encoded_jpg_io)
    width, height = image.size

    filename = group.filename.encode('utf8')
    image_format = b'jpg'
    xmins = []
    xmaxs = []
    ymins = []
    ymaxs = []
    classes_text = []
    classes = []

    for index, row in group.object.iterrows():
        xmins.append(row['xmin'] / width)
        xmaxs.append(row['xmax'] / width)
        ymins.append(row['ymin'] / height)
        ymaxs.append(row['ymax'] / height)
        classes_text.append(row['class'].encode('utf8'))
        classes.append(class_text_to_int(row['class']))
```

```
    tf_example                                                    =
tf.train.Example(features=tf.train.Features(feature={
        'image/height': dataset_util.int64_feature(height),
        'image/width': dataset_util.int64_feature(width),
        'image/filename':
dataset_util.bytes_feature(filename),
        'image/source_id':
dataset_util.bytes_feature(filename),
        'image/encoded':
dataset_util.bytes_feature(encoded_jpg),
        'image/format':
dataset_util.bytes_feature(image_format),
        'image/object/bbox/xmin':
dataset_util.float_list_feature(xmins),
        'image/object/bbox/xmax':
dataset_util.float_list_feature(xmaxs),
        'image/object/bbox/ymin':
dataset_util.float_list_feature(ymins),
        'image/object/bbox/ymax':
dataset_util.float_list_feature(ymaxs),
        'image/object/class/text':
dataset_util.bytes_list_feature(classes_text),
        'image/object/class/label':
dataset_util.int64_list_feature(classes),
    }))
    return tf_example


def main():
    writer = tf.io.TFRecordWriter(FLAGS.output_path)
    path = os.path.join(FLAGS.image_dir)
    examples = pd.read_csv(FLAGS.csv_input)
    grouped = split(examples, 'filename')
```

```
    for group in grouped:
        tf_example = create_tf_example(group, path)
        writer.write(tf_example.SerializeToString())

    writer.close()
    output_path = os.path.join(os.getcwd(),
FLAGS.output_path)
    print('Successfully created the TFRecords:
{}'.format(output_path))


if __name__ == '__main__':
    main()
```

Lampiran Script ResNet configuration

```
model {
  faster_rcnn {
    num_classes: 9
    image_resizer {
      keep_aspect_ratio_resizer {
        min_dimension: 600
        max_dimension: 1024
      }
    }
    feature_extractor {
      type: 'faster_rcnn_resnet50'
      first_stage_features_stride: 16
    }
    first_stage_anchor_generator {
      grid_anchor_generator {
        scales: [0.25, 0.5, 1.0, 2.0]
        aspect_ratios: [0.5, 1.0, 2.0]
```

```
    height_stride: 16
    width_stride: 16
  }
}
first_stage_box_predictor_conv_hyperparams {
  op: CONV
  regularizer {
    l2_regularizer {
      weight: 0.0
    }
  }
  initializer {
    truncated_normal_initializer {
      stddev: 0.01
    }
  }
}
first_stage_nms_score_threshold: 0.0
first_stage_nms_iou_threshold: 0.7
first_stage_max_proposals: 300
first_stage_localization_loss_weight: 2.0
first_stage_objectness_loss_weight: 1.0
initial_crop_size: 14
maxpool_kernel_size: 2
maxpool_stride: 2
second_stage_box_predictor {
  mask_rcnn_box_predictor {
    use_dropout: false
    dropout_keep_probability: 1.0
    fc_hyperparams {
      op: FC
      regularizer {
        l2_regularizer {
```

```
              weight: 0.0
            }
          }
          initializer {
            variance_scaling_initializer {
              factor: 1.0
              uniform: true
              mode: FAN_AVG
            }
          }
        }
      }
    }
    second_stage_post_processing {
      batch_non_max_suppression {
        score_threshold: 0.0
        iou_threshold: 0.6
        max_detections_per_class: 100
        max_total_detections: 300
      }
      score_converter: SOFTMAX
    }
    second_stage_localization_loss_weight: 2.0
    second_stage_classification_loss_weight: 1.0
  }
}

train_config: {
  batch_size: 1
  optimizer {
    momentum_optimizer: {
      learning_rate: {
        manual_step_learning_rate {
```

```
          initial_learning_rate: 0.0001
          schedule {
            step: 900000
            learning_rate: .0001
          }
          schedule {
            step: 1200000
            learning_rate: .0001
          }
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
  gradient_clipping_by_norm: 10.0
  fine_tune_checkpoint:
"E:/Kuliah/Ikhtiar/Program_1/faster_rcnn_resnet50_coco_2018_0
1_28/model.ckpt"
  from_detection_checkpoint: true
  load_all_detection_checkpoint_vars: true

  num_steps: 200000
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  max_number_of_boxes: 50
}

train_input_reader: {
  tf_record_input_reader {
    input_path: "E:/Kuliah/Ikhtiar/Program_1/train.record"
```

```
  }
  label_map_path:
"E:/Kuliah/Ikhtiar/Program_1/Training/labelmap.pbtxt"
}

eval_config: {
  metrics_set: "coco_detection_metrics"
  num_examples: 38
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "E:/Kuliah/Ikhtiar/Program_1/test.record"
  }
  label_map_path:
"E:/Kuliah/Ikhtiar/Program_1/Training/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
}
```