



PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG
JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2
IMAGERY MENGGUNAKAN METODE *OBJECT-BASED*
IMAGE ANALYSIS (OBIA)

DISERTASI

GUNTUR EKA SAPUTRA

99219009

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

2023



**Pengembangan Metode Deteksi Dan Menghitung Jumlah Pohon
Kelapa Sawit Dari Sentinel 2 Imagery Menggunakan Metode
*OBJECT-BASED IMAGE ANALYSIS (OBIA)***

DISERTASI

Untuk Memenuhi Salah Satu Syarat Meraih Gelar Doktor Teknologi
Informasi di bawah Pimpinan Rektor Universitas Gunadarma
Profesor Doktor E.S. Margianti, SE, MM

Laporan Rapat Komisi Pembimbing Dipertahankan dalam Sidang Terbuka
Senat Universitas Gunadarma
Pada Hari Rabu, 10 Mei 2023

GUNTUR EKA SAPUTRA

99219009

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

2023

PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG
JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2
IMAGERY MENGGUNAKAN METODE *OBJECT-BASED*
IMAGE ANALYSIS (OBIA)

DISERTASI

GUNTUR EKA SAPUTRA

Telah disetujui oleh:

Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.
Promotor

Profesor Doktor Sarifuddin Madenda, S.Si., DEA.
Ko-Promotor

Jakarta, 10 Mei 2023

Judul Disertasi : **PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2 IMAGERY MENGGUNAKAN METODE *OBJECT-BASED IMAGE ANALYSIS* (OBIA)**

Nama Mahasiswa : Guntur Eka Saputra
NIM : 99219009

Komite Pembimbing

Promotor : Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.
Ko-Promotor : Profesor Doktor Sarifuddin Madenda, S.Si., DEA.

Komite Penguji

Ketua : Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.

Anggota : Profesor Doktor Insinyur Sudrajat, M.Sc.
Profesor Doktor E. S. Margianti, S.E., M.M.
Profesor Suryadi Harmanto, S.Si., M.M.S.I.
Profesor Doktor Insinyur Bambang Suryawan, M.T.
Profesor Insinyur Busono Soerowirdjo, M.Sc., Ph.D.
Profesor Doktor Eri Prasetyo Wibowo, S.Si., M.M.S.I.
Doktor rer. nat. I Made Wiryan, S.Si., S.Kom.,
M.App.Sc.
Doktor Detty Purnamasari, S.Kom., M.M.S.I.,
M.I.Kom.
Profesor Doktor Sarifuddin Madenda, S.Si., D.E.A.

PERNYATAAN ORIGINALITAS DAN PUBLIKASI

Saya yang bertanda tangan di bawah ini:

Nama : Guntur Eka Saputra

NIM : 99219009

Judul Disertasi : PENGEMBANGAN METODE DETEKSI DAN
MENGHITUNG JUMLAH POHON KELAPA SAWIT
DARI SENTINEL 2 IMAGERY MENGGUNAKAN
METODE *OBJECT-BASED IMAGE ANALYSIS*
(OBIA)

Tanggal Sidang : 10 Mei 2023

Tanggal Lulus :

Menyatakan bahwa tulisan ini adalah merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apapun telah mengikuti kaidah dan etika yang berlaku. Mengenai sisi dan tulisan adalah merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, 10 Mei 2023

(Guntur Eka Saputra)

ABSTRAK

Penerapan teknologi informasi dan komunikasi sangat dibutuhkan, khususnya di bidang pertanian. Kelapa sawit merupakan produk pertanian yang terbesar di Indonesia dan produksi kelapa sawit sangat penting bagi perekonomian. Dalam menghasilkan produksi kelapa sawit yang baik dibutuhkan pengelolaan perkebunan kelapa sawit yang baik. Permasalahan utama dalam pengelolaan ini, yaitu luas area dalam skala besar (perusahaan perkebunan kelapa sawit memiliki luas minimum sebesar 6.000 hektar yang harus dikelola), wilayah perkebunan berada di *remote area*, akses infrastruktur yang terbatas, dan pemupukan presisi mengalami kesulitan untuk mendapatkan data secara akurat berdasarkan jumlah tegakkan pohon kelapa sawit pada suatu area lahan. Hal ini menyebabkan deteksi dan menghitung pohon kelapa sawit sangat dibutuhkan. Selama ini, penghitungan tradisional didasarkan pada catatan awal penanaman pohon kelapa sawit atau penghitungan teoritis berdasarkan jarak tanam antara pohon kelapa sawit dalam satu hektar atau blok. Metode tradisional ini lambat dan tidak akurat, serta tidak diketahui status pohon kelapa sawit yang rusak atau mati. Penggunaan teknologi dibutuhkan untuk dapat secara otomatis dan *real-time* dalam memonitoring data pohon kelapa sawit dan memperkirakan produktivitasnya. Penggunaan ini dibutuhkan dalam deteksi objek berupa citra.

Penggunaan citra untuk deteksi objek dibutuhkan dalam persiapan data, seperti menganalisa, memberikan anotasi atau kelas dari objek tersebut. Metode dalam melakukan anotasi dataset selama ini dilakukan secara manual, satu per satu dengan memberikan kotak batas. Persiapan data ini menghabiskan lebih dari 70% waktu dalam siklus hidup *deep learning* untuk menjadi dataset yang dapat digunakan sebagai data pelatihan, validasi, dan pengujian. Hal inilah yang menjadi tantangan bagi *stakeholders*.

Penelitian ini bertujuan untuk menghasilkan pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan label suatu kelas pada data citra. Metode yang dikembangkan menggunakan algoritma klasifikasi *template matching*. Algoritma ini sebagai template awal citra pohon kelapa sawit yang memiliki kunci nilai ambang batas dalam menentukan kelas dari objek di dalam citra. Algoritma BIRCH digunakan untuk mengurangi objek yang bukan terdeteksi ke dalam kelas pohon kelapa sawit. Hasil evaluasi performance pelatihan menunjukkan bahwa model dengan algoritma YOLOv7 lebih baik dengan akurasi best MAP sebesar 0,993 dan pada pengujian sebesar 0,997. Berdasarkan waktu pemrosesan DGX-A-100 Universitas Gunadarma lebih baik, pada pelatihan sebesar 2948 detik dibandingkan dengan Google Colab Pro sebesar 4847 detik.

Penelitian ini dihasilkan purwarupa sistem yang menggunakan model algoritma dari YOLOv7 untuk dapat mendeteksi dan menghitung pohon kelapa sawit pada area tertentu dari citra satelit yang terintegrasi dengan Google Maps API. Berdasarkan hasil pengujian 4 blok pada Kebun Pendidikan dan Pendidikan Kelapa Sawit IPB-Cargil bahwa hasil presentasi berhasil dideteksi

sebesar 97,67%, dan diketahui setiap pohon kelapa sawit yang terdeteksi diketahui letak titik koordinat untuk dapat dilakukan pemantauan, pengelolaan, dan estimasi produktivitas pohon kelapa sawit.

Kata kunci : OBIA, *Deep Learning*, YOLOv7, Kelapa Sawit.

ABSTRACT

The application of information and communication technology is needed, especially in agriculture. Palm oil is the largest agricultural product in Indonesia and palm oil production is very important for the economy. In producing good oil palm production, good oil palm plantation management is needed. The main problems in this management, namely large-scale areas (oil palm plantation companies have a minimum area of 6,000 hectares that must be managed), plantation areas in remote areas, limited infrastructure access, and precision fertilization have difficulty obtaining accurate data based on the number of standing oil palm trees in a land area. This makes detecting and counting oil palm trees necessary. So far, traditional counting has been based on early records of oil palm tree planting or theoretical counting based on the spacing between oil palm trees within a hectare or block. These traditional methods are slow and inaccurate, and the status of damaged or dead oil palm trees is unknown. The use of technology is needed to be able to automatically and real-time monitor palm oil tree data and estimate its productivity. This use is needed in object detection in the form of images.

The use of images for object detection is needed in data preparation, such as analyzing, annotating or classifying the object. The method of annotating datasets has been done manually, one by one by providing boundary boxes. This data preparation consumes more than 70% of the time in the deep learning lifecycle to become a dataset that can be used as training, validation, and testing data. This is the challenge for stakeholders.

This research aims to produce the development of object-based image analysis (OBIA) method to create datasets automatically by labeling a class on image data. The developed method uses a template matching classification algorithm. This algorithm as an initial template of palm tree image that has a key threshold value in determining the class of objects in the image. The BIRCH algorithm is used to reduce objects that are not detected into the palm tree class. The results of the training performance evaluation show that the model with the YOLOv7 algorithm is better with a best MAP accuracy of 0.993 and on testing of 0.997. Based on processing time, Gunadarma University's DGX-A-100 is better, at 2948 seconds of training compared to Google Colab Pro at 4847 seconds.

This research produced a prototype system that uses the algorithm model from YOLOv7 to be able to detect and count oil palm trees in a certain area from satellite images integrated with the Google Maps API. Based on the results of testing 4 blocks in the IPB-Cargil Oil Palm Education and Education Plantation that the presentation results were successfully detected by 97.67%, and it is known that each detected oil palm tree is known to the location of the coordinate point to be able to monitor, manage, and estimate the productivity of oil palm trees.

Key words: OBIA, Deep Learning, YOLOv7, Oil Palm.

KATA PENGANTAR

Bismillahhirrohmaanirrohim

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah robil'aalamin

Segala Puji, Kebesaran, Kemulian, dan apa yang dilangit dan di bumi milik Allah SWT. Puji Syukur saya panjatkan kehadirat Allah SWT atas segala rahmat serta nikmat-Nya yang telah memberikan kemudahan serta kelancaran kepada Saya dalam penyelesaian Disertasi yang berjudul "Pengembangan Metode Deteksi Dan Menghitung Jumlah Pohon Kelapa Sawit Dari Sentinel 2 Imagery Menggunakan Metode *OBJECT-BASED IMAGE ANALYSIS (OBIA)*". Disertasi ini merupakan syarat untuk memperoleh gelar Doktor dalam bidang Teknologi Informasi pada Program Doktor Teknologi Informasi, Program Pascasarjana, Universitas Gunadarma, dimana penulis telah menyelesaikan seluruh rangkaian proses studi Program Doktor sejak tahun 2019. Sepanjang proses penyusunan Disertasi ini, banyak pihak yang turut membantu baik secara moril maupun materil kepada saya. Untuk itu dengan segala kerendahan dan ketulusan hati, perkenankan saya mengucapkan terima kasih kepada:

1. Ibu Profesor Doktor E. S. Margianti, S.E., MM., selaku Rektor Universitas Gunadarma.
2. Bapak Profesor Doktor Insinyur Bambang Suryawan, MT., selaku Koordinator Program Pascasarjana Universitas Gunadarma.
3. Bapak Profesor Insinyur Busono Soerowirdjo, MSc., Ph.D., selaku Direktur Program Doktor Universitas Gunadarma.
4. Profesor Doktor Sarifuddin Madenda, selaku Ketua Program Doktor Teknologi Informasi Universitas Gunadarma sekaligus Ko-Promotor dis-

elasela kesibukannya dengan sabar membimbing, mengarahkan, memberi masukan dan memotivasi dalam menyelesaikan disertasi

5. Bapak Profesor Doktor Eri Prasetyo Wibowo, selaku Sekretaris Program Doktor Teknologi Informasi Universitas Gunadarma.
6. Bapak Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc. selaku Promotor yang dengan sabar memberikan membimbing, memotivasi, melakukan koreksi, memberi masukkan dan saran dalam menyelesaikan disertasi ini.
7. Bapak Profesor Doktor Insinyur Sudrajat, M.Sc., selaku Penguji Luar terima kasih atas waktu, kesediaan, dan masukkan, serta saran disertasinya bagi saya.
8. Bapak Profesor Suryadi Harmanto, SSi., MMSI, Bapak Doktor rer. nat. I Made Wiriana dan Ibu Doktor Detty Purnamasari selaku penguji dalam yang telah memberi banyak masukan dan saran perbaikan, sehingga disertasi ini semakin berkualitas.
9. Bapak Doktor Irwan Bastian yang telah memberikan dukungan kepada saya dapat melaksanakan kuliah Program Doktor Teknologi Informasi di Universitas Gunadarma.
10. Ayahanda tercinta Baikusnendro dan Ibunda tercinta Mennik Trihastuti, yang selalu memberikan doa yang terbaik dan motivasi, serta kedua adik saya Finsa Dwi Hestu Fikriansya, dan Hilmi Hestu Saputra yang selalu mendukung secara moril.
11. Rekan-rekan angkatan 25 Program Doktor Teknologi Informasi Universitas Gunadarma yang selalu memberikan semangat, dan Mas Bonang Waspadadi Ligar, serta Sumaiyah Fitriandini yang berjuang bersama, serta diskusi.

12. Ibu-ibu di Sekretariat Program Doktor Teknologi Informasi Universitas Gunadarma Ibu Doktor Diny Wahyuni, Ibu Doktor Reni Diah Kusumawati, Ibu Doktor Aini Suri Talita, dan Ibu Doktor Dety Purnamasari yang sangat membantu dalam administrasi penyelesaian disertasi ini.
13. Teman-teman Bidang Kemahasiswaan Universitas Gunadarma, komunitas GDSC UG, Gunadarma IO, UGTV, Tim Teknis UG dan sahabat saya Muhammad Rifqi Al Furqon, Manfred Michael, dan Evan Sakti Endi yang telah membantu dalam penyelesaian disertasi ini.

Semoga Allah SWT memberikan limpahan kebaikan dan pahala atas semua perhatian dan dukungan yang Bapak, Ibu, Saudara sekalian berikan kepada saya. Harapan saya agar Disertasi ini memberikan manfaat nyata bagi semua pihak yang berkepentingan. Saya mengharapkan kritik dan saran untuk perbaikan pada masa yang akan datang.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Jakarta, 10 Mei 2023

(Guntur Eka Saputra)

DAFTAR ISTILAH

Akronim

ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
EDA	Electronic Design Automation

Singkatan

AMS	Austria Micro System
CMOS	Complementary Metal Oxide Silicon
FPGA	Field Programmable Gate Array
GDSII	Graphics Data Station Information Interchang
HDL	Hardware Description Language
IP	Intellectual Property
IC	Integrated Circuit
μm	Mikrometer, 1×10^{-6}
RTL	Register Transfer Level
SOC	System On CHIP
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Interated Circuit
VLSI	Very Large Scale Integration

Simbol

AMS	Austria Micro System
CMOS	Complementary Metal Oxide Silicon
FPGA	Field Programmable Gate Array
GDSII	Graphics Data Station Information Interchang

HDL	Hardware Description Language
IP	Intellectual Property
IC	Integrated Circuit
μm	Mikrometer, 1×10^{-6}
RTL	Register Transfer Level
SOC	System On CHIP
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Interated Circuit
VLSI	Very Large Scale Integration

DAFTAR ISI

COVER	i
HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGUJI	iv
PERNYATAAN ORIGINALITAS DAN PUBLIKASI	v
ABSTRAK	vi
ABSTRACT	viii
KATA PENGANTAR	x
DAFTAR ISTILAH	xiii
DAFTAR ISI	xv
DAFTAR TABEL	xxi
DAFTAR GAMBAR	xxvii
DAFTAR ALGORITMA	xxvii
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Penelitian	13

1.3	Rumusan Masalah	14
1.4	Tujuan Penelitian	15
1.5	Kontribusi Hasil Penelitian	16
II	TELAAH PUSTAKA	17
2.1	<i>Computer Vision</i>	17
2.2	Pengertian Citra	18
2.3	Pengolahan Citra	19
2.4	<i>Artificial Intelligence</i> (AI)	20
2.5	<i>Object Detection</i>	22
2.6	Machine Learning	26
2.6.1	Supervised Learning	27
2.6.2	Unsupervised Learning	29
2.6.3	Semi-supervised learning	31
2.6.4	Reinforcement Learning	32
2.6.5	Deep Learning	34
2.6.5.1	Recurrent Neural Networks (RNN)	38
2.6.5.2	Long Short Term Memory (LSTM)	39
2.6.5.3	Convolutional Neural Network (CNN)	40
2.6.5.4	Deep Believe Networks (DBN)	47
2.6.5.5	Deep Stacking Networks (DSN)	48
2.7	You Only Look Once (YOLO)	49
2.7.1	Arsitektur YOLO	51
2.7.2	Cara kerja Deteksi Objek YOLO	52
2.7.3	Perkembangan YOLO	56
2.8	Object Based Image Analysis (OBIA)	66
2.9	Sentinel-2	68
2.10	Pertanian Presisi	69
2.11	Kelapa Sawit	71

2.12	Teknologi Informasi	73
2.13	Teknik Televisi Digital	74
2.14	Digital Video Broadcasting Terestrial 2 (DVB-T2)	77
2.15	Jenis Kompresi Transmisi Video Digital	81
2.15.1	Video H.222 MPEG-TS	81
2.15.2	Video H.264 MPEG-4	83
2.16	<i>Signal to Noise Ratio</i> (SNR)	84
2.17	<i>Signal Strength</i>	85
2.18	Metrik Kualitas Gambar	87
2.18.1	Penilaian Kualitas Gambar Referensi Penuh	89
2.18.2	Penilaian Kualitas Gambar Tanpa Referensi	90
2.18.2.1	Metrik NR-Bloking	90
2.18.2.2	Metrik NR-Blur	91
2.18.2.3	Metrik NR-Temporal	92
2.18.3	<i>Artificial Neural Network</i> (ANN)	93
2.19	Perangkat Lunak <i>tvheadend</i>	99
2.20	<i>Subjective Assessment</i>	100
2.20.1	Raspberry Pi + TV Hat	103
2.20.2	DVB-T2 TV Tunner	103
2.20.3	<i>Confusion Matrix</i>	103
III	METODE PENELITIAN	106
3.1	Tahapan Penelitian	106
3.2	Area Studi	108
3.2.1	Obyek Penelitian	108
3.2.2	Pemilihan Perangkat Keras	109
3.3	Pembentukan <i>Dataset</i>	112
3.3.1	Akuisisi <i>Dataset</i> Video	113
3.3.2	Metrik Obyektif Pengukuran Kualitas Gambar	113

3.3.2.1	Metrik Bloking	114
3.3.2.2	Metrik Blur	117
3.3.2.3	Metrik Temporal	121
3.3.3	Pengukuran Subyektif Kualitas Gambar	123
3.3.3.1	Single Stimulus Standar ITU-R 500-14	123
3.3.3.2	Tahap 1 - Pemilihan materi dan luaran	126
3.3.3.3	Tahap 2 - Sesi tes	128
3.3.3.4	Tahap 3 - Pengukuran hasil tes	129
3.3.3.5	Desain Aplikasi Subyektif <i>Assessment</i>	131
3.4	Pembuatan Model NN	132
3.5	Hasil dan Perhitungan Model	135
3.5.1	<i>Confusion</i> matriks	136
3.5.2	Pengukuran Korelasi	139
3.6	Sistem Pengukuran Waktu Nyata	141
3.6.1	Tahapan pembuatan <i>website</i> pemantauan	143
3.6.2	Arsitektur <i>Website Monitoring</i>	145
IV HASIL DAN PEMBAHASAN		147
4.1	Material yang digunakan	147
4.1.1	Proses akuisi material video	147
4.1.2	Koleksi data	153
4.1.3	Perangkat pengolahan data	154
4.2	Pengujian metrik obyektif	155
4.2.1	Pengujian metrik <i>blocking</i>	155
4.2.2	Pengujian metrik <i>blur</i>	158
4.2.3	Pengujian metrik <i>temporal</i>	162
4.2.4	Pengukuran obyektif pada keseluruhan <i>dataset</i>	167
4.3	Hasil pengambilan data pengukuran subyektif	168
4.3.1	Material pengukuran subyektif	168

4.3.2	Perangkat yang digunakan	169
4.3.3	Proses pengukuran subyektif	169
4.3.4	Hasil pengukuran subyektif	171
4.4	Pengujian dan hasil pada NN	175
4.4.1	Data masukan dan luaran model	176
4.4.2	Pengujian arsitektur NN	179
4.4.2.1	Hasil regresi multivariabel	179
4.4.2.2	Hasil ANN tanpa klasifikasi	181
4.4.2.3	Hasil ANN dengan klasifikasi	181
4.5	Aplikasi pemantauan kualitas gambar siaran DVB-T2	181
4.5.1	Arsitektur aplikasi	181
4.5.2	Pengujian dan implementasi aplikasi	182
V	PENUTUP	183
5.1	Kesimpulan	183
5.2	Saran	184
DAFTAR PUSTAKA		185
DAFTAR RIWAYAT HIDUP		200

DAFTAR TABEL

1.1	Statistik Areal dan Produksi Kelapa Sawit Menurut Provinsi dan Status Pengusahaan Tahun 2021 (Sumber: Ditjenbun, 2023)	4
2.1	Penerapan Arsitektur Deep Learning	38
2.2	Pengelompokan Berdasarkan Masa Berbuah	72
2.3	Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman dalam Kondisi Kebun Percobaan Balit Marihat	73
2.4	Status DTV membandingkan sistem penyiaran digital di selu- ruh dunia ATSC, DTMB, DVB-T/DVB-T2, dan ISDB-T (DTVs- tatus 2017)	77
2.5	Fitur DVB-S2, DVB-C2 dan DVB-T2 (SANDY 4.)	79
2.6	Spesifikasi DVB-T dan DVB-T2 (Ya'acob et al. 2019)	81
3.1	Versi bahasa pemrograman python dan librariya	111
3.2	Standar pengujian SS dengan <i>home-environment</i>	126
3.3	Skala ITU-R Quality dan Impairment	127
4.1	Pengambilan <i>dataset</i> dari layanan siaran DVB-T2 pada area studi	152
4.2	Perangkat pengolahan data yang digunakan	154
4.3	Waktu yang dibutuhkan untuk memproses satu video dalam satuan detik	165
4.4	Pengukuran metrik obyektif terhadap dataset	167
4.5	Hasil pengukuran subyektif terhadap dataset	172
4.6	Batasan masing-masing kelas berdasarkan pengukuran subyektif	175
4.7	Hasil pengukuran subyektif terhadap dataset	176

DAFTAR GAMBAR

1.1	Area Lahan Kelapa Sawit IPB Cargil Jonggol Blok 3 Jarak Tanam 9x9x9	5
1.2	Pola Tanam dan Jarak Tanam Kelapa Sawit yang Tepat (Sumber: GD Morganic, 2018)	6
1.3	Peta 10 Besar Provinsi Sentra Kelapa Sawit Tahun 2021 (Sumber: Ditjenbun, 2023)	7
1.4	Estimasi Luas Area dan Produksi Minyak Sawit Indonesia 2023 Sumber: (Ditjenbun, 2023)	7
2.1	(a) Hubungan Artificial Intelligence, Machine Learning, dan Deep Learning. (b) Jenis Machine Learning (Sumber: C. Thongprayoon et al, 2020)	21
2.2	Classification, Object Detection dan Segmentation Representation (Sumber: A. Patel, 2020)	22
2.3	Segmentation, Classification+Localization, Object Detection (Sumber: A. Patel, 2020)	23
2.4	Image Classification (Sumber: A. Patel, 2020)	23
2.5	Object Detection dengan Bounding Box (Sumber: A. Patel, 2020)	24
2.6	Intersection Over Union (IoU) (Sumber: A. Patel, 2020)	25
2.7	Persamaan IoU (Sumber: A. Patel, 2020)	25
2.8	Skema Supervised Learning (Sumber: M. Kozan, 2021)	27
2.9	Skema Unsupervised Learning (Sumber: M. Kozan, 2021)	29

2.10 Skema Semi-Supervised Learning (Sumber: A. B. Nassif et al, 2019)	32
2.11 Skema Reinforcement Learning (Sumber: R. Sutton, 1998)	33
2.12 Skema Umum <i>Deep Learning</i> (Sumber: A. Feizollah et al, 2022)	36
2.13 Skema <i>Deep Learning</i> dengan Penambahan beberapa <i>hidden layer</i> (Sumber: H. Kaur et al, 2021)	37
2.14 Diagram <i>Recurrent Neural Network</i> (RNN) (Sumber: K. Dass, 2020)	39
2.15 Diagram <i>Long Short Term Memory</i> (LSTM) (Sumber: S. Yan, 2016)	40
2.16 Array dari Matriks RGB (Sumber: R. Prabhu, 2018)	41
2.17 Neural Network dengan banyak Convoluti Layer (Sumber: R. Prabhu, 2018)	41
2.18 Gambar Matriks Multiplies Kernel atau Filter Matriks (Sumber: R. Prabhu, 2018)	42
2.19 Gambar matriks 5 x 5 dikalikan dengan Filter matiks 3 x 3	42
2.20 Output Matriks 3 x 3	43
2.21 Beberapa Filter Umum (Sumber: R. Prabhu, 2018)	44
2.22 Stride 2 Piksel (Sumber: R. Prabhu, 2018)	45
2.23 Operasi ReLU (Sumber: P. Ratan, 2021)	46
2.24 Max Pooling	46
2.25 Setelah Pooling Layer Diratakan sebagai FC Layer	47
2.26 Arsitektur CNN Lengkap	47
2.27 Skema Diagram DBN (Sumber: H. Liu & B. Lang, 2019)	48
2.28 Skema Diagram DSN (Sumber: L. Deng et al, 2012)	49
2.29 Kecepatan YOLO dibandingkan dengan detector Objek Lainnya (Sumber: S. A. S. Hernandez et al, 2020)	50
2.30 Arsitektur YOLO dari <i>Original Paper</i> (J. Redmon et al., 2015)	51

2.31 (A) Input Image dan (B) Hasil Algoritma YOLO (Sumber: Z. Kelta, 2022)	52
2.32 Residual Blocks (Sumber: Z. Kelta, 2022)	53
2.33 Grid dengan Probabilitas (Sumber: Z. Kelta, 2022)	54
2.34 Cara <i>Bounding Box</i> (Sumber: Z. Kelta, 2022)	54
2.35 IOU (Sumber: Z. Kelta, 2022)	55
2.36 Perkembangan YOLO (Sumber: Z. Kelta, 2022)	56
2.37 Komparasi YOLOv4 dengan YOLOv3 dan <i>state-of-the-art</i> Deteksi Objek Lain (Sumber: Z. Kelta, 2022)	59
2.38 Unified Network Architecture (Sumber: C. Y. Wang et al, 2021)	60
2.39 Performance YOLOR vs YOLOv4 dan Model Lainnya (Sumber: C. Y. Wang et al, 2021)	61
2.40 Perbandingan Waktu Pelatihan antara YOLOv4 dan YOLOv5 (Sumber: J. Nelson, 2020)	63
2.41 Comparison of state-of-the-art efficient object detectors. All models are tested with TensorRT 7 except that the quantized model is with TensorRT 8 (Sumber: C. Li et al, 2022)	64
2.42 Perbandingan YOLOv7 inference time dengan real-time object detector lainnya (Sumber: C. Y. Wang et al, 2022)	65
2.43 <i>Compound scaling up depth and width for concatenation-based model</i> (Sumber: C. Y. Wang et al, 2022)	66
2.44 OBIA Segmentasi Proses Pengelompokkan Pixel yang hampir sama ke dalam Objek (Sumber: GISGeography, 2020)	67
2.45 OBIA Klasifikasi menggunakan shape, size, dan spectral properties objek untuk klasifikasi setiap objek	67
2.46 Siklus Pertanian Presisi (Sumber: A. Comparetti, 2011)	70
2.47 Buah Kelapa Sawit(Inti dan Daging Sawit)	72
2.48 dasar telekomunikasi (El-Saba 2018)	74

2.49 Sistem penyiaran digital untuk televisi terestrial (DTVstatus 2017)	76
2.50 Sistem pemancar DVB-T2 (Hou 2010)	80
2.51 H.264/AVC Struktur GOP (Apple-Inc. 2012)	83
2.52 Contoh gambar dari Basis Data Penilaian Kualitas Gambar LIVE, yang menunjukkan ambiguitas metrik kualitas objektif (dalam hal ini, PSNR)(Sheikh et al. 2006)	89
2.53 Contoh gambar asli Lena dan versi terdistorsinya dengan artefak pemblokiran (Kusuma 2005)	91
2.54 Contoh video yang terganggu oleh pembekuan frame temporal. Setiap huruf yang berbeda mewakili frame yang unik. (Huynh-Thu & Gha 2009)	93
2.55 Struktur otak manusia dengan kemampuan operasional (SANDY 24)	95
2.56 Fungsi Sigmoid	97
2.57 Langkah training ANN (SANDY 25)	98
2.58 Antarmuka <i>website</i> perangkat lunak <i>tvheadend</i>	100
 3.1 Tahapan penelitian yang dilakukan	107
3.2 Contoh Gambar FHD dari layanan TVRI World	109
3.3 Fokus Penelitian	111
3.4 Rancangan Sistem Pengukuran Kualitas Gambar Siaran DVB-T2112	
3.5 Perbandingan gambar normal dan <i>blocking</i>	114
3.6 Perbandingan Gambar Blur dan tidak, dari kiri ke kanan: Tumbling Es, Lena, Street View (Xu et al. 2021)	118
3.7 Hasil Transformasi ke bentuk deteksi tepi menggunakan sobal pada gambar Lenna	119
3.8 Diagram alur algoritma pengukuran blur pada gambar (Marziliano et al. 2002)	120

3.9	Struktur presentasi dari subyektif tes	128
3.10	Interval antara <i>midgrey</i> dan materi uji	129
3.11	Rancangan tampilan aplikasi web subyektif <i>assessment</i>	132
3.12	Diagram parameter I/O untuk membangun model NN	133
3.13	Tahapan pembuatan arsitektur dan model NN	135
3.14	<i>Multiple Class Confusion Matrix</i> (Markoulidakis et al. 2021) . .	137
3.15	Diagram alur program python pada pengukuran kualitas gambar siaran DVB-T2 waktu nyata	142
3.16	Tahapan pembuatan <i>website monitoring real-time</i> kualitas gambar DVB-T2	144
3.17	Diagram arsitektur <i>website</i> pemantauan kualitas gambar siaran DVB-T2	145
4.1	Perangkat keras yang digunakan untuk akuisisi video siaran DVB-T2	148
4.2	Pengukuran kekuatan sinyal menggunakan <i>signal analyzer, TinySA</i>	149
4.3	Penempatan antena outdoor PX HDA-5000	149
4.4	Fitur melakukan perekaman video siaran DVB-T2 menggunakan <i>tvheadend</i>	150
4.5	Jarak stasiun pemancar terhadap lokasi pengambilan data siaran DVB-T2	152
4.6	Citra/frame video dengan resolusi FHD 1920×1080 piksel hasil rekaman dari siaran DVB-T2	153
4.7	Pengukuran metrik obyektif blok pada citra normal dan <i>blocking artifact</i>	156
4.8	Contoh pengukuran metrik obyektif blok pada beberapa citra siaran televisi DVB-T2	157
4.9	Pengukuran metrik obyektif blur pada citra normal dan blur . .	159

4.10 Contoh pengukuran metrik obyektif blur pada beberapa citra siaran televisi DVB-T2	160
4.11 Pengukuran metrik temporal <i>freeze</i> dengan MSE skor	162
4.12 Contoh pengukuran metrik obyektif temporal pada video <i>dataset</i> no.84 siaran televisi DVB-T, temporal skor : 90	164
4.13 Contoh pengukuran metrik obyektif temporal pada video <i>dataset</i> no.15 siaran televisi DVB-T, temporal skor : 65	166
4.14 Tampilan aplikasi pengukuran subyektif kualitas citra video siaran DVB-T2	170
4.15 Tampilan dashboard aplikasi pengukuran subyektif dari sisi administrator	171
4.16 Sebaran MOS pengukuran tes subyektif dari dataset video	173
4.17 Pembagian kelas MOS pengukuran subyektif dataset	173
4.18 Varian pengukuran subyektif dataset dengan skor terurut	174
4.19 Diagram scatter pengukuran metrik spatial terhadap MOS pada pengukuran subyektif (a) metrik blok, (b) metrik blur.	177
4.20 Diagram scatter pengukuran metrik temporal terhadap MOS pada pengukuran subyektif (a) metrik freeze, (b) metrik variansi gradien	178
4.21 Diagram hasil prediksi terhadap MOS	180
4.22 Diagram Scater hasil prediksi terhadap MOS	181

DAFTAR ALGORITMA

4.1	Metrik pengukuran blok	155
4.2	Metrik pengukuran blur	159
4.3	Metrik pengukuran blur	161
4.4	Metrik pengukuran blur	163

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang pesat telah membantu dalam menjalani kehidupan yang lebih sederhana dan praktis. Permintaan untuk pengembangan teknologi terus meningkat dan telah menjadi salah satu bidang pekerjaan dan studi yang paling popular(Abid Haleem 2022). Perkembangan teknologi, khususnya investasi teknologi informasi dikaitkan dengan pertumbuhan produktivitas yang signifikan di negara maju dan berkembang (Jason Dedrick 2014). Di negara-negara berkembang, teknologi informasi dan komunikasi sangat membutuhkan penerapan khususnya di bidang pertanian. Pertanian memainkan peran sentral dalam pembangunan di banyak negara. Akses pasar, pembiayaan, dan pengetahuan adalah landasan pertumbuhan pertanian. Teknologi Informasi dan Komunikasi (TIK) mendukung petani dengan harga pasar secara real-time, prakiraan cuaca, hama, varietas benih dan teknik penanaman, identifikasi dan penghitungan tanaman. Pertanian juga merupakan sumber pendapatan utama bagi penduduk pedesaan di sebagian besar negara. Sektor ini menghadapi banyak tantangan untuk meningkatkan produksi. TIK memiliki potensi untuk memenuhi tantangan yang dihadapi oleh petani dan atau pemangku kepentingan yang dapat meningkatkan taraf hidup masyarakat pedesaan(Havinal 2020).

Dalam kasus ini, Indonesia adalah salah satu produsen dan eksportir pertanian terbesar di dunia, yang memasok bahan mentah penting seperti

karet alam, kopi, kakao, beras, dan minyak kelapa sawit ke seluruh dunia. Dalam beberapa dekade terakhir, industri pertanian juga telah menjadi sektor yang paling banyak menyerap tenaga kerja di Indonesia (Statista Research, 2023). Indonesia juga sebagai produsen perkebunan terbesar di dunia, seperti karet alam dan kelapa sawit, produksi tanaman Indonesia sangat penting bagi perekonomian nasional. Dari 15 produk pertanian utama Indonesia, kelapa sawit merupakan yang terbesar. Produksi kelapa sawit sangat penting bagi perekonomian, karena Indonesia produsen dan konsumen terbesar di dunia dan menyumbang sekitar setengah dari pasokan dunia. Sebagian minyak kelapa sawit berasal dari perkebunan yang dikelola oleh petani kecil dengan luas 6,02 juta hektar pada tahun 2021. Petani kecil menghasilkan sekitar 34,36% produksi minyak kelapa sawit, menjadikan petani kecil sebagai kontributor penting dalam mempromosikan industri kelapa sawit yang berkelanjutan di Indonesia (Ditjenbun, 2023).

Dalam menghasilkan produksi kelapa sawit yang baik, maka dibutuhkan penerapan teknologi yang mendukung untuk dapat memonitoring pohon kelapa sawit tersebut yang dapat menghasilkan, karena luas lahan yang begitu besar tidak memungkinkan dapat dikerjakan secara konvensional.

Pertanian presisi membutuhkan informasi yang dapat diandalkan tentang situasi terkini pada waktu yang tepat. Manajer kelapa sawit biasanya mengukur kepadatan atau jumlah pohon kelapa sawit secara manual. Tanaman kelapa sawit juga memiliki lahan yang luas, maka diperlukan pemantauan lahan secara berkala untuk mengontrol produktivitas kelapa sawit dan juga sebagai data inventaris, oleh karena itu, identifikasi otomatis dan lokasi kelapa sawit merupakan cara alternatif bagi petani untuk mengelola sumber daya mereka dengan menggunakan teknologi, bukan dengan pendekatan manual.

Kelapa sawit yang memiliki nama ilmiah (*Elaeis guineensis Jacq.*) merupakan tanaman yang berasal dari daerah Benua Afrika dan negara di Amerika

Selatan. Pada awalnya tanaman ini tumbuh liar dan setengah liar di daerah tepi sungai. Di Indonesia, tanaman ini pertama kali diperkenalkan oleh pemerintah colonial Belanda pada tahun 1848 di Kebun Raya Bogor (I. Pahan, 2018). Perkebunan kelapa sawit di Indonesia berkembang pesat, pada tahun 1939, Indonesia menjadi negara produsen dan eksportir utama kelapa sawit dunia dengan volume mencapai 244 ribu ton atau sebesar 48% total ekspor minyak kelapa sawit dunia (S. Prayitno et al, 2008).

Besar volume dari kelapa sawit yang diproduksi oleh Indonesia menjadikan sector kelapa sawit membantu ekonomi Indonesia. Dalam perekonomian makro ekonomi Indonesia, industri atau sector kelapa sawit telah memiliki peran strategis, diantaranya penghasil devisa terbesar, lokomotif perekonomian nasional, kedaulatan energy, pendukung sector ekonomi kerakyatan, dan penyerapan tenaga kerja. Hal ini dilihat dari besarnya perkebunan sawit di Indonesia. Perkebunan kelapa sawit Indonesia berkembang di 22 provinsi dari 33 provinsi. Dua pulau utama sentra perkebunan kelapa sawit yaitu Sumatera dan Kalimantan. Sekitar 90% perkebunan kelapa sawit di Indonesia berada pada dua pulau tersebut, dan kedua pulau tersebut telah menghasilkan 95% produksi minyak sawit mentah (*crude palm oil/CPO*) Indonesia. Dalam perjalannya perkebunan kelapa sawit mengalami revolusi dan berkembang dengan cepat. Dalam kurun 1990-2015, tumbuh dan berkembangnya perkebunan rakyat dengan cepat, yakni, 24% per tahun selama periode 1990-2015. Pada tahun 2015, luas perkebunan sawit di Indonesia adalah 11,3 juta hektar (KPRI, 2015). Menurut Ketua Asosiasi Industri Minyak Makan Indonesia (AIMMI) Adi Wisoko Kasman menjelaskan bahwa peningkakan konsumsi minyak kelapa sawit dalam negeri terus mengalami peningkatan (L. Yuniartha, 2019). Pada tahun 2019, konsumsi minyak sawit tumbuh hingga 23,57% atau meningkat dari 13,49 juta ton di 2018 menjadi 16,67 juta ton di tahun 2019, kemudian konsumsi minyak sawit untuk kategori makanan (food) dalam negeri mencapai

9,86 juta ton atau naik hingga 49% tahun per tahun (L. Yuniartha, 2020). Pada tahun 2021, hasil reevaluasi luas areal dari direktorat jenderal perkebunan, diketahui bahwa perkebunan besar swasta sebesar 60,64%, diikuti perkebunan rakyat 34,36%, dan perkebunan negara 5% dari 14.621.690 ha. Pada tahun 2023, CPO Indonesia diprediksi mencapai 48,2 juta ton (Ditjenbun, 2023). Luas areal dan produksi kelapa sawit menurut provinsi dan status pengusahaan tahun 2021, seperti Tabel 1.1 berikut ini:

Tabel 1.1: Statistik Areal dan Produksi Kelapa Sawit Menurut Provinsi dan Status Pengusahaan Tahun 2021
(Sumber: Ditjenbun, 2023)

Perkebunan Rakyat Smallholders		Perkebunan Negara Government Estate		Perkebunan Swasta Private Estate		Jumlah (Total)	
Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)
6.029.749	15.503.840	550.333	2.256.134	8.041.068	27.361.506	14.621.690	45.121.480

Berdasarkan Tabel 1.1 bahwa saat ini areal dan produksi kelapa sawit masih dikuasai dan dikelola oleh Perkebunan Swasta yang tersebar dari 34 provinsi di Indonesia, yaitu luas areal sebesar 8.041.068 (Ha) dan dapat memproduksi 27.361.506 (Ton) dari 45.121.480 (Ton) (Ditjenbun, 2023).

Dari data yang diperoleh menurut Ditjenbun tahun 2023 bahwa dibutuhkan kelapa sawit dalam jumlah yang besar dan produksi minyak sawit yang dihasilkan dari kelapa sawit tersebut harus baik. Pohon kelapa sawit membutuhkan waktu sekitar 4 (empat) tahun untuk menghasilkan buah yang sesuai untuk panen. Setiap pohon kelapa sawit kemudian akan terus menghasilkan buah hingga usia 25 tahun. Menurut Smart Agribusiness and food dalam artikelnya menyebutkan bahwa pohon kelapa sawit ini memanfaatkan teknologi pertanian didukung oleh analisis satelit di seluruh area perkebunan, sehingga pemanfaatan hasil panen dapat dihasilkan hasil panen dapat digunakan secara optimal (Smart Agribusiness and Food, 2017). Dalam pengelolaan perkebunan kelapa sawit, populasi dalam satuan hektar (ha) sangat penting dan hal ini

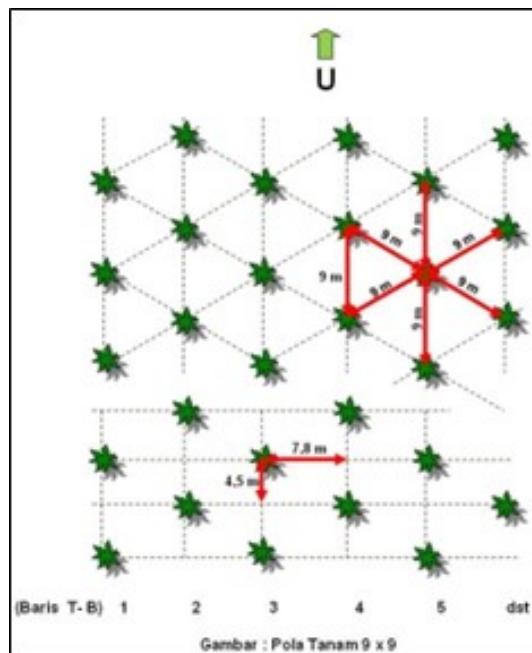
berhubungan dengan pengaruh jarak tanam. Jarak tanam merupakan faktor yang mempengaruhi pertumbuhan tanaman kelapa sawit. Pengaturan jarak tanam bertujuan untuk mendapatkan ruang tumbuh bagi pertumbuhan tanaman guna menghindari kompetisi unsur hara dan cahaya matahari dari setiap tanaman kelapa sawit pada jarak tanam yang berbeda. Jarak tanam 9x9x9 m baik untuk kelapa sawit (Hayata et al, 2020). Kepadatan pada jarak tanam normal biasanya bervariasi sesuai dengan jenis tanah yang ditanami kelapa sawit. Jumlah tanaman kelapa sawit pada pesisir dan di tanah mineral adalah antara 136 - 148 kelapa sawit/hektar seperti pada Gambar 1.1, sedangkan di tanah gambut, jarak tanamnya biasanya lebih padat, sekitar 150 kelapa sawit/hektar (J. Latif et al., 2003). Gambar 1.1 merupakan area lahan Kebun Pendidikan dan Penelitian IPB Cargil - Jonggol Blok 3 dengan tanah mineral (J. Albari et al., 2018). Tanah mineral adalah tanah yang terbentuk dan berkembang dari bahan mineral, melalui proses pelapukan, baik secara fisis maupun kimia, didominasi oleh pelapukan bebatuan (A. Ahmad et al., 2018).



Gambar 1.1: Area Lahan Kelapa Sawit IPB Cargil Jonggol Blok 3 Jarak Tanam 9x9x9

Dalam memenuhi kebutuhan minyak kelapa sawit, selain membutuhkan areal lahan, dibutuhkan pola tanam dan jarak tanam kelapa sawit yang tepat.

Pola tanam kelapa sawit yang baik dibutuhkan perhatikan yang lebih karena berkaitan dengan efektifitas penggunaan lahan. Pola tanam segitiga sama sisi merupakan pola tanam yang paling efektif di areal datar, sehingga untuk areal bergelombang atau berbukit perlu dilakukan "viol linning" untuk mempertahankan jumlah populasi per hektarnya dengan tetap memperhatikan tingkat kesuburan tanahnya (GD Morganic, 2018), ilustrasi jarak tanam kelapa sawit seperti pada Gambar 1.2.



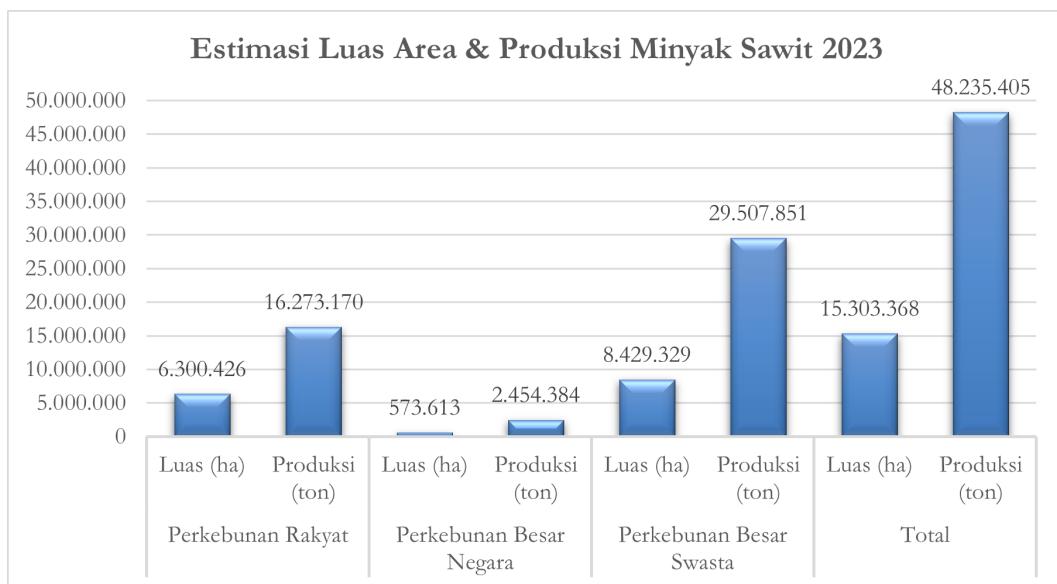
Gambar 1.2: Pola Tanam dan Jarak Tanam Kelapa Sawit yang Tepat
(Sumber: GD Morganic, 2018)

Direktorat Jenderal Perkebunan (Ditjenbun) memetakan memetakan produksi kelapa sawit menurut provinsi dan menunjukkan bahwa provinsi Riau memiliki produksi terbesar 2021 sebesar 8,96 juta ton dengan luas 3,49 juta hektar (Ditjenbun, 2023). Berikut ini Gambar 1.3. peta kelapa sawit Indonesia berdasarkan Direktorat Jenderal Perkebunan.



Gambar 1.3: Peta 10 Besar Provinsi Sentra Kelapa Sawit Tahun 2021
(Sumber: Ditjenbun, 2023)

Direktorat jenderal perkebunan juga mencatat estimasi luas area dan produksi minyak sawit Indonesia tahun 2023 mencapai 48,24 juta ton dan luas hektar mencapai 15,3 juta ton (Ditjenbun, 2023) seperti pada Gambar 1.4.



Gambar 1.4: Estimasi Luas Area dan Produksi Minyak Sawit Indonesia 2023
Sumber: (Ditjenbun, 2023)

Gambar 1.4 menyatakan bahwa perkebunan swasta (PBS) menguasasi sebanyak 61,2% dengan total produksi 29,5 juta ton, perkebunan rakyat (PR) sebesar 33,7% dengan total produksi sebanyak 16,2 juta ton, sedangkan perkebunan besar negara (PBN) hanya 5% dengan total 2,45 juta ton.

Jumlah estimasi luas area lahan pohon kelapa sawit yang besar juga menjadikan permasalahan bagi pengelolaan perkebunan kelapa sawit. Dari sisi agronomis, bahwa batasan minimum untuk perusahaan perkebunan kelapa sawit yaitu sebesar 6.000 (enam ribu) hektar yang harus dikelola (PPRI, 2021). Luas lahan untuk petani kelapa sawit yang didefinisikan sebagai petani kelapa sawit adalah tinggal di pedesaan/sekitar kebun yang dimana kelapa sawit sebagai mata pencaharian utama, dikerjakan dan dikontrol sendiri oleh keluar-ganya, dan mengalami kesulitan karena bibit yang digunakan disemai sendiri dan tidak bersertifikat, sulit mengakses area karena luas, serta produktivitas rendah menjadi permasalahan utama (SPKS, 2020). Di samping permasalahan luas area ini, lokasi perkebunan sawit juga berada pada remote area, yaitu area atau daerah yang terpencil, jauh dari peradaban (daerah pedalaman atau pelosok hutan) yang dimana karena lokasi perkebunan kelapa sawit yang luas, sehingga sulit dimonitoring tanpa teknologi informasi. Permasalahan yang ada bahwa di perkebunan kelapa sawit yang luas juga terkendala akses infrastruktur, seperti jalan yang belum memadai untuk akses kendaraan, seperti di Kebun Pendidikan dan Penelitian IPB-Cargil akses jalan bergelombang dan masih bebatuan, dan menurut dinas perkebunan provinsi Kalimantan timur bahwa infrastruktur dan aksesibilitas terhubung dengan baik, maka produktivitas dan kegiatan monitoring kawasan perkebunan kelapa sawit akan sangat membantu (Disbun, 2014). Produksi per individu tanaman pohon kelapa sawit berkontribusi besar terhadap produktivitas karena setiap pohon kelapa sawit akan menghasilkan buah kelapa sawit yang dapat diproduksi dan dipanen hasilnya untuk masyarakat, sehingga data pohon kelapa sawit sangat penting untuk dilakukan. Selain itu, salah satu upaya yang dilakukan dalam perkebunan kelapa sawit adalah masalah pemupukan. Adanya peningkatan produksi melalui intensifikasi dengan pemberian pupuk yang presisi, secara cepat, dan akurat, serta up-to-date pada suatu luasan lahan. Upaya

yang dapat dilakukan adalah dengan meningkatkan produktivitas melalui peningkatan efektivitas dan efisiensi penggunaan pupuk. Efektivitas dan efisiensi pemberian pupuk sangat penting dilakukan karena biaya pemupukan tanaman kelapa sawit sangat besar yaitu 50%-70% dari biaya pemeliharaan dan 25% dari seluruh biaya produksi (Fairhurst et al., 2006). Penambahan unsur hara dapat meningkatkan pertumbuhan tanaman, produksi tanaman, dan kualitas produk yang dihasilkan, seperti buah kelapa sawit (S. Parman, 2007).

Buah kelapa sawit yang dapat dihasilkan di Indonesia cukup besar dan harus diproduksi atau dipanen dengan baik, agar hasil dari buah kelapa sawit yang mayoritas menjadi minyak menghasilkan minyak yang berkualitas. Hal ini berkenaan dengan standar mutu kematangan buah kelapa sawit yang dikenal dengan standar kematangan panen. Buah sawit bergerombol dalam tandan yang muncul dari tiap pelapah. Minyak dihasilkan oleh buah. Kandungan minyak bertambah sesuai dengan tingkat kematangan buah. Penghitungan pohon kelapa sawit yang menghasilkan minyak dapat dilakukan secara manual di lapangan, tetapi ini menghabiskan waktu, membutuhkan pekerja yang tidak sedikit, dan mahal (N. A Mubin, et al, 2019). Disamping itu, Manajer perkebunan kelapa sawit biasanya mengukur kepadatan kelapa sawit secara manual setiap tahun. Data penting ini dapat digunakan untuk memperkirakan produktivitas kelapa sawit, jumlah pupuk yang dibutuhkan, biaya penyiraman berkala, dan jumlah pekerja yang dibutuhkan, dan terkait dengan kegiatan lain (J. W. Kiama, 2014).

Oleh karena itu, dibutuhkan suatu metode atau teknologi untuk membantu mengklasifikasikan atau mendeteksi jumlah dan estimasi usia pohon kelapa sawit untuk dapat diketahui tingkat kematangan buah kelapa sawit, maka dapat diproduksi (dipanen) dengan baik, sehingga produksi yang dihasilkan dapat mencapai optimum. Berikut ini penelitian-penelitian yang telah dilakukan penelitian sebelumnya.

Klasifikasi berbasis piksel sering digunakan untuk mengklasifikasikan kelas fitur dari gambar. Metode Object-Based Image Analysis (OBIA) telah berevolusi untuk menganalisis gambar resolusi tinggi dengan cepat (T. Blaschke, 2010), dengan semakin tersedianya resolusi tinggi dan citra skala besar, maka dihasilkan cara baru untuk memulai penelitian tentang interpretasi foto pohon berbasis komputer (P. Gong et al, 1999). OBIA dikembangkan oleh Hay dan Castilla merupakan disiplin ilmu spasial yang berfokus pada pengelompokan citra penginderaan jauh ke makna objek penuh melalui pemanfaatan spasial (G. J. Hay et al, 2008). Gagasan menganalisis gambar dalam ruang objek daripada ruang pixel dikembangkan karena kekurangan metode berbasis pixel, terutama pada gambar resolusi tinggi. Selain itu, ruang objek telah diperkuat dalam kapasitas komputasi dan ketersediaan untuk analisis gambar resolusi tinggi seperti IKONOS, GeoEye, dan WorldView (T. Blaschke, 2010).

Jusoff dan Pathan menggunakan airborne hyperspectral sensing linear analis campuran spectral, bersama dengan campuran ke konverter murni dan norma Euclidean teknik untuk memetakan masing-masing pohon kelapa sawit (K. Jusoff, 2009). Korom et all tahun 2014 mengelompokkan bentuk kanopi atau mahkota kelapa sawit menggunakan citra WorldView-2 berdasarkan segmentasi watershed dan mencapai akurasi sekitar 77% (A. Korom et al, 2014).

Wong-in et al. tahun 2015 telah mencapai akurasi 90% menggunakan gambar udara dengan beberapa langkah, seperti menghapus komponen non-pohon dari gambar, membedakan minyak sawit dari komponen lain menggunakan filter low-pass dan korelasi silang dinormalisasi, mengidentifikasi pohon kelapa sawit secara individu dan menghitung jumlah pohon kelapa sawit (Wong-in et al, 2015).

H. Santoso et al. pada tahun 2016 membangun dan mengembangkan metode yang mudah digunakan pengguna yang akan memungkinkan manager kelapa sawit untuk menghitung minyak pohon kelapa sawit menggunakan

teknik penginderaan jauh. Pohon kelapa sawit dianalisis dalam penilitian ini dengan melihat usia dan kepadatan yang berbeda. Penelitian ini menggunakan Citra QuickBird yang diaplikasikan dengan enam metode pansharpening. Hitam dan citra putih dari komposit warna palsu citra pansharpening diproses dalam tiga cara: (1) deteksi pohon kelapa sawit, (2) penggambaran area kelapa sawit, (3) penghitungan pohon kelapa sawit dan penilaian akurasi. Penelitian ini menggunakan ENVI 5.2, ERDAS Imagine 2015, dan ArcGIS 10.2.2. Hasil penelitian ini meningkat pada akurasi dari beberapa studi penelitian sebelumnya yang memiliki akurasi 90-95%. Hasil dalam penelitian ini menunjukkan (1) gabungan resolusi intensitas huesaturation (HIS) cocok untuk kelapa sawit berusia 16 tahun, pohon dan memiliki kepadatan agak tinggi dengan akurasi 100%, (2) untuk kelapa sawit berusia 21 tahun dan memiliki kepadatan rendah didapatkan hasil dengan akurasi 99,5%, (3) resolusi subtraktif penggabungan ini cocok untuk pohon kelapa sawit berusia 15-18 tahun dan memiliki kerapatan yang agak tinggi dengan akurasi 99,8%, (4) penajaman spectral PC dengan akurasi 99,3% cocok untuk pohon kelapa sawit berusia 10 tahun dan memiliki kepadatan rendah, dan (5) untuk semua kondisi objek studi, warna dinormalisasi (Brovey) dan penggabungan resolusi wavelet adalah dua metode pansharpening yang cocok untuk ekstraksi dan kelapa sawit dengan penghitungan masing-masing dengan akurasi 98,9% dan 98,4% (H. Santoso et al, 2016).

H. M. Rizeei et al. tahun 2018 meneliti bahwa pemantauan karakteristik kelapa sawit di area perkebunan sangat berharga bagi petani dan stakeholders untuk memaksimalkan produktivitas. Penelitian ini mengusulkan metode baru untuk estimasi dan penghitungan usia kelapa sawit. Algoritma Support Vector Machine (SVM) dari analisis gambar berbasis objek (OBIA) diterapkan untuk penghitungan kelapa sawit. Analisis sensitivitas dilakukan pada empat jenis kernel SVM (Sigmoid (SIG), Linear (LN), fungsi basis radial (RBF), dan

polynomial (PL)) dengan parameter terkait (Nilai ambang batas, Gamma (γ) dan faktor Penalti (c)) untuk mendapatkan pendekatan klasifikasi OBIA optimal untuk setiap blok perkebunan. Citra dengan resolusi sangat tinggi dari Worldview-3 (WV-3) digunakan untuk deteksi kelapa sawit. Hasil deteksi kelapa sawit memiliki akurasi keseluruhan 98,27%, 99,48%, 99,28%, 99,49%, dan 97,49% untuk blok A, B, C, D, dan E (H. M. Rizeei, 2018).

W Li et al tahun 2018 meneliti deteksi pohon kelapa sawit dalam skala besar dari gambar satelit beresolusi tinggi (QuickBird) menggunakan Two-Stage Convolutional *Neural Networks* (TS-CNN). TS-CNN terdiri dari satu CNN untuk klasifikasi tutupan lahan dan satu CNN untuk klasifikasi objek. Kedua CNN dilatih dan dioptimalkan secara independen berdasarkan pada 20.000 sampel yang dikumpulkan melalui interpretasi manusia. Skala area pohon kelapa sawit sebesar 55 km². Penelitian ini mengusulkan alur kerja yang efektif yang terdiri dari metode partisi yang tumpeng tindih untuk divisi gambar skala besar, metode *multi-scale sliding window* untuk memprediksi koordinat kelapa sawit, dan metode filter jarak untuk pasca-pemrosesan. Pendekatan yang diusulkan mencapai skor F1 rata-rata yang jauh lebih tinggi yaitu 94,99% di wilayah studi dibandingkan dengan metode deteksi kelapa sawit yang ada masing-masing sebesar (87,95%, 81,80%, 80,61%, dan 78,35% untuk CNN satu-tahap, *Support Vector Machine* (SVM), *Random Forest* (RF), dan *Artificial Neural Network* (JST)), dan jauh lebih sedikit kebingungan dengan vegetasi dan bangunan lain di seluruh hasil deteksi gambar (W. Li et al, 2019).

N. A. Mubin et al. tahun 2019 telah meneliti bahwa deteksi dan penghitungan kelapa sawit penting dalam pengelolaan perkebunan kelapa sawit. Dalam penelitian ini, digunakan pendekatan pembelajaran mendalam untuk memprediksi dan menghitung kelapa sawit dalam citra satelit. Deteksi kelapa sawit sebelumnya secara umum berfokus pada mendeteksi kelapa sawit

yang tidak memiliki tumpeng tindih mahkota. Selain itu, terdapat kurangnya penelitian yang membangun sistem deteksi terpisah untuk kelapa sawit muda dan dewasa, memanfaatkan pendekatan pembelajaran mendalam untuk mendeteksi kelapa sawit dan menggabungan sistem informasi geografis (SIG) dengan pendekatan pembelajaran mendalam. Penelitian ini mencoba untuk mengisi kesenjangan ini dengan memanfaatkan dua jaringan saraf convolution (CNN) yang berbeda untuk mendeteksi kelapa sawit muda dan matang secara terpisah dan menggunakan GIS selama pemrosesan data dan proses penyimpanan hasil. Arsitektur awal yang dikembangkan didasarkan pada CNN yang disebut LeNet. Proses pelatihan mengurangi kerugian dengan menggunakan algoritma gradient adaptif dengan batch mini ukuran 20 untuk semua set pelatihan yang digunakan. Kemudian, mengekspor hasil prediksi ke perangkat lunak GIS dan membuat peta prediksi kelapa sawit dewasa dan muda. Berdasarkan metode yang diusulkan, akurasi keseluruhan untuk kelapa sawit muda dan matang masing-masing adalah 95,11% dan 92,96%. Secara keseluruhan, pengklasifikasi bekerja dengan baik pada dataset yang sebelumnya tidak terlihat, dan mampu secara akurat mendeteksi kelapa sawit dari latar belakang, termasuk bayangan tanaman lainnya (N. A. Mubin, 2019).

1.2 Batasan Penelitian

Ruang lingkup yang menjadi penelitian adalah:

1. Dataset citra sebagai dataset primer berjumlah 69 citra dari Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil, Jonggol, Jawa Barat dan 205 citra dari Universitas Gunadarma, PPU Campus, Kalimantan Timur yang ditanami pohon kelapa sawit.
2. Dataset sekunder digunakan melalui akses roboflow secara free sebanyak 1795 citra dataset sekunder yang sudah memiliki label atau anotasi kelas

'oil palm'.

3. Pengujian penerapan sistem menghitung dan mendeteksi pohon kelapa sawit dilakukan pada lahan pohon kelapa sawit tanah mineral Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil, Jonggol Jawa Barat.

1.3 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini secara umum bahwa pengelolaan perkebunan kelapa sawit memiliki permasalahan yang utama, diantaranya luas area pengelolaan kebun kelapa sawit dalam skala besar (perusahaan perkebunan kelapa sawit memiliki luas minimum sebesar 6.000 (enam ribu) hektar dan petani 4 hektar yang harus dikelola). Perkebunan berada di *remote area*, area yang terpencil, daerah pedalaman atau pelosok hutan, yang sulit untuk dilakukan monitoring, dan akses infrastruktur terutama jalan yang terbatas dan sulit. Permasalahan yang dihadapi juga berupa pemupukan presisi yang mengalami kesulitan untuk mendapatkan secara akurat, dan cepat, serta *up-to-date* jumlah tegakkan pohon kelapa sawit pada suatu luasan lahan tertentu, sementara praktik yang ada adalah menggunakan perhitungan manual dan normatif berbasis jarak tanam pohon kelapa sawit. Hal ini menyebabkan pentingnya data jumlah pohon kelapa sawit sangat penting dibutuhkan karena produksi per individu tanaman berkontribusi besar terhadap produktivitas dengan penyelesaian teknologi informasi. Berdasarkan permasalahan tersebut, maka rumusan masalah secara khusus:

1. Bagaimana membuat pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas pada data citra?
2. Bagaimana mendapatkan model CNN yang baik dari algoritma YOLOv5,

YOLOv6, dan YOLOv7 untuk mendekripsi dan menghitung pohon kelapa sawit pada suatu area tertentu?

3. Bagaimana mengembangkan sistem berbasis web dengan menerapkan model CNN dari algoritma YOLO agar dapat mendekripsi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit dari citra satelit agar dapat dilakukan pemantauan, pengelolaan, dan estimasi produktivitas secara otomatis?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini secara umum adalah menghasilkan purwarupa sistem berbasis web dengan menerapkan model OBIA dan *Deep Learning* (DL) untuk mendekripsi, menghitung jumlah pohon dalam suatu luasan tertentu dan mendapatkan nilai koordinat latitude-longitude setiap pohon kelapa sawit dari citra satelit. Tujuan penelitian secara khusus sebagai berikut:

1. Menghasilkan metode object-based image analysis (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas, kelas 'oil palm' pada data citra.
2. Menghasilkan model CNN dari algoritma YOLOv5, YOLOv6, dan YOLOv7 yang dapat mendekripsi dan menghitung pohon kelapa sawit pada suatu area tertentu.
3. Menghasilkan purwarupa perangkat lunak aplikasi dengan model convolutional neural network dari algoritma YOLO yang terintegrasi dengan citra satelit dengan menggunakan Google Maps API yang dapat mendekripsi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit pada suatu area tertentu.

1.5 Kontribusi Hasil Penelitian

Kontribusi penting dari penelitian yang dilakukan dalam penelitian ini adalah:

1. Dari sisi keilmuan, berupa pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas pada data citra dengan menggunakan algoritma *template matching* dan BIRCH.
2. Dari sisi teknologi, penelitian ini menghasilkan dataset primer dengan kelas pohon kelapa sawit (*oil palm*) dan purwarupa perangkat lunak aplikasi dengan model *convolutional neural network* dari algoritma YOLO yang terintegrasi dengan citra satelit dengan menggunakan Google Maps API yang dapat mendeteksi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit yang berhasil didekksi.
3. Dari sisi pemanfaatan, menawarkan suatu cara alternatif untuk proses persiapan data citra untuk mendapatkan dataset dengan proses anotasi yang lebih cepat, serta bagi *stakeholder* dapat memantau, mengelola, dan melakukan estimasi produktivitas secara otomatis pohon kelapa sawit pada suatu area tertentu dengan penerapan teknologi informasi.

BAB II

TELAAH PUSTAKA

2.1 *Computer Vision*

Computer Vision atau visi komputer adalah bidang kecerdasan buatan (AI) yang memungkinkan komputer dan sistem memperoleh informasi bermakna dari gambar digital, video, dan input visual lainnya, dan mengambil tindakan atau membuat rekomendasi berdasarkan informasi tersebut. Jika kecerdasan buatan memungkinkan komputer untuk berpikir, visi komputer memungkinkan untuk melihat, mengamati, dan memahami (IBM, 2020).

Computer vision bekerja hamper sama dengan visi manusia, kecuali manusia memiliki permulaan. Penglihatan manusia memiliki keunggulan konteks seumur hidup untuk melatih cara membedakan objek, seberapa jauh jaraknya, apakah bergerak, dan apakah ada yang salah dalam sebuah gambar.

Computer vision melatih mesin untuk melakukan fungsi-fungsi melatih mesin untuk melakukan fungsi-fungsi ini, tetapi ia harus melakukannya dalam waktu yang jauh lebih singkat dengan kamera, data, dan algoritma daripada retina, saraf, optic, dan korteks visual karena sistem yang dilatih untuk memeriksa produk atau mengamati asset produksi dapat menganalisis ribuan produk atau proses dalam satu menit, memperhatikan cacat atau masalah yang tidak terlihat, sistem tersebut dapat dengan cepat melampaui kemampuan manusia.

Memahami dan menentukan tugas visi komputer tertentu dapat memfokuskan dan memvalidasi proyek dan aplikasi, serta mempermudah untuk memulai. Hal dasar untuk *computer vision* adalah *object detection*, beberapa tugas visi komputer lainnya, seperti *image classification*, *object detection*,

object tracking, dan *content-based image retrieval* (U. Arshad, 2021).

2.2 Pengertian Citra

Citra didefinisikan sebagai fungsi dari dua variabel misalnya $a(x,y)$ dimana a sendiri sebagai amplitudo (misalnya kecerahan) citra pada koordinat (x,y) (I. T. Young et al, 1995). Citra digital $a[m,n]$ merupakan citra dalam ruang diskrit 2D yang berasal dari citra analog $a(x,y)$ di ruang kontinyu 2D melalui proses sampling yaitu yang biasa disebut sebagai digitalisasi. Sedangkan, menurut Maria citra digital adalah citra $f(x,y)$ yang telah didiskritkan oleh pada koordinat spasial dan kecerahan. Citra digital direpresentasikan oleh *array* dua dimensi atau sekumpulan *array* dua dimensi dimana setiap *array* merepresentasikan satu kanal warna. Nilai kecerahan yang didigitalkan dinamakan nilai tingkat keabuan (A. McAndrew, 2004).

Setiap elemen *array* tersebut dinamakan piksel yang diambil dari istilah *picture element*. Dimensi citra biasanya ditulis dengan format panjang x tinggi (misalnya 640 x 480 piksel). Namun, perlu diperhatikan dengan seksama bahwa secara matematis, definisi citra terlihat seperti di bawah ini, dimana x menunjukkan baris dan y menunjukkan kolom:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Seperti pada layar monitor, koordinat citra dimulai dari pojok kiri atas. Secara matematis dimulai dari $(0,0)$ dan berakhir di $(M-1, N-1)$, dimana M

menunjukkan tinggi, dan N menunjukkan panjang.

2.3 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya menggunakan komputer menjadi citra yang kualitasnya lebih baik. Pengolahan citra adikembangkan bertujuan untuk (M. Petrou, 1999):

1. Untuk memperbaiki tampilan citra (*image enhancement*).
2. Untuk mengurangi ukuran file citra dengan tetap mempertahankan kualitas citra (*image compression*).
3. Untuk memulihkan citra ke kondisi semula (*image restoration*).
4. Untuk menyoroti ciri tertentu dari citra agar lebih mudah untuk di analisis.

Pengolahan citra adalah cabang ilmu informatika untuk memperbaiki kualitas citra agar kualitasnya lebih baik atau lebih mudah diinterpretasi oleh manusia maupun komputer. Input dari program pengolahan citra adalah citra dan outputnya pun citra pula.

Pengolahan citra digital digunakan dalam berbagai bidang untuk mempermudah manusia dalam melakukan analisis dan pekerjaan. Bentuk aplikasi pengolahan citra digital yang digunakan bidang militer, industry, medis, transportasi, hukum dan keamanan, pemetaan, robotika, fotografi, film, pencarian gambar berdasarkan kandungan citra, dan pemahaman kandungan citra. Salah satu pemanfaatan teknologi pengolahan citra digital yaitu bisa memahami maksud dari sebuah citra. Apabila aplikasi diberikan input berupa gambar yang mampu mendefinisikan bahwa dalam gambar tersebut terdapat gambar yang mendapatkan objek, seperti kendaraan, jalan, buah, dan lainnya.

2.4 *Artificial Intelligence (AI)*

Artificial intelligence atau kecerdasan buatan adalah studi tentang teori dan pengembangan sistem komputer agar mampu melakukan tugas-tugas yang dahulu hanya dapat dilakukan oleh manusia. Seperti membandakan berbagai gambar, menjawab pertanyaan, mengenali dan menerjemahkan bahasa, dan sebagainya (R. Primartha, 2018).

Komputer atau mesin cukup bagus untuk melakukan hal-hal berikut: menyelesaikan perhitungan aritmatika dengan cepat, mengerjakan secara akurat apa-apa yang sudah deprogram oleh komputer. Namun, komputer atau mesin memiliki kelemahan, seperti sulit berinteraksi dengan *noisy data* (data yang blur/bias), sulit memahami lingkungan, kurang toleran terhadap kesalahan (*fault tolerance*), sulit beradaptasi dengan situasi dan kondisi tertentu. Untuk mengatasi hal tersebut ada lima hal yang perlu dimiliki oleh mesin, yaitu:

1. Persepsi

Terkait dengan permasalahan pengindraan. Mesin harus memiliki indra untuk dapat mengenali dunia sekitarnya.

2. Pemrosesan bahasa alami (NLP)

Kemampuan untuk mengidentifikasi kalimat dan memahami perbedaan aksesn dan maknanya.

3. Menyampaikan pengetahuan

Menyampaikan berbagai informasi di dunia luar berdasarkan pemikirannya sendiri.

4. Pengambilan keputusan

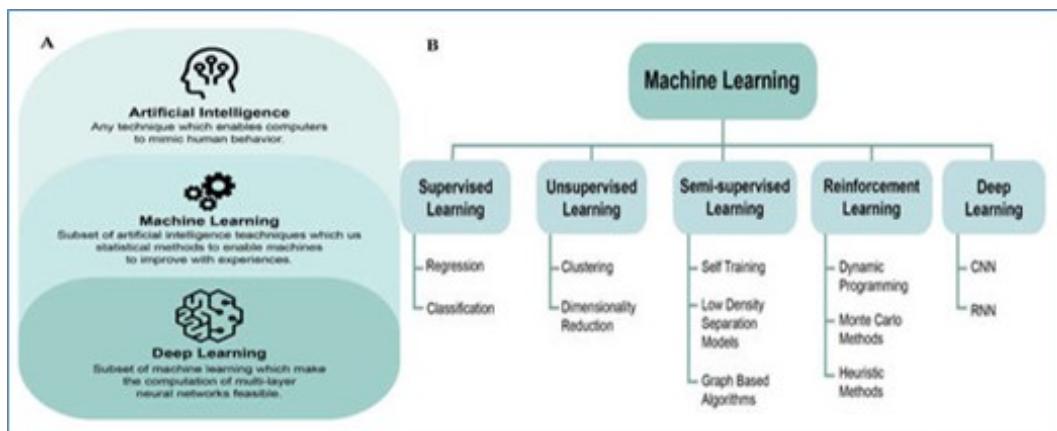
Mampu memecahkan berbagai permasalahan secara logis.

5. Perencanaan dan pemetaan

Memetakan dunia tiga dimensi dan merencanakan rute paling efektif.

Salah satu bagian penting dari AI adalah machine learning atau pembelajaran

Salah satu bagian penting dari AI adalah *machine learning* atau pembelajaran mesin, yaitu dicirikan sebagai studi tentang algoritma dan model statistik yang digunakan sistem komputer untuk belajar dari data sampel dan pengalaman sebelumnya tanpa diprogram secara eksplisit untuk mencapai tugas tertentu. Dengan kemampuan untuk mengidentifikasi pola yang tidak jelas dalam data, kita dapat menggunakan pembelajaran mesin untuk memecahkan banyak masalah, termasuk menilai hubungan dua variabel, membuat prediksi berdasarkan karakteristik dasar, mengidentifikasi objek dengan pola yang sebanding, dan menggabungkan subjek dengan kriteria tertentu. Baik *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning* merupakan tiga istilah yang popular, namun banyak yang salah mengira bahwa ketiganya menggambarkan hal yang sama, padahal tiga hal yang berbeda. Bagaimana hubungan tiga istilah ini, seperti pada Gambar 2.1



Gambar 2.1: (a) Hubungan Artificial Intelligence, Machine Learning, dan Deep Learning. (b) Jenis Machine Learning
(Sumber: C. Thongprayoon et al, 2020)

Disamping hal tersebut, *artificial intelligence* (AI) memungkinkan un-

tuk berfikir, dan salah satu bagian dari *artificial intelligence* untuk dapat melihat, mengamati, dan memahami adalah komputer visi atau *computer vision*. *Computer vision* memungkinkan untuk komputer dan sistem memberikan informasi berarti dari gambar digital, dan visual input.

2.5 Object Detection

Object detection atau deteksi objek dianggap sebagai salah satu bidang penting dalam pembelajaran mendalam dan visi komputer. Deteksi objek telah ditentukan oleh banyak aplikasi dalam visi komputer, seperti pelacakan objek, pengambilan, dan pengawasan video. Deteksi objek adalah teknologi *deep learning* dimana benda, manusia, bengunan, mobil, dapat dideteksi sebagai objek dalam gambar dan video (U. Arshad, 2021).

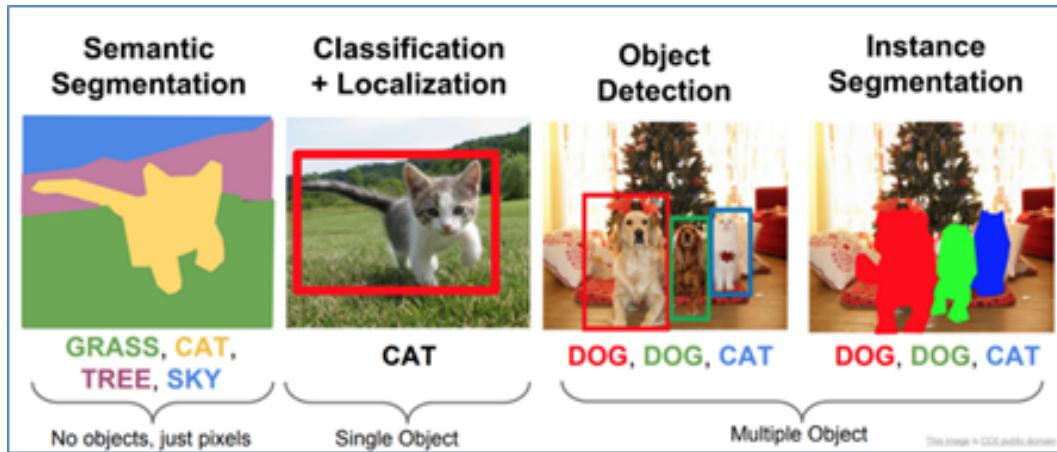
Deteksi objek untuk mengenali objek dengan kotak pembatas pada gambar, dimana dalam klasifikasi gambar, cukup mengkategorikan (mengklasifikasikan) objek pada gambar atau tidak dalam hal kemungkinan (*probability*), seperti contoh pada Gambar 2.2



Gambar 2.2: Classification, Object Detection dan Segmentation Representation
(Sumber: A. Patel, 2020)

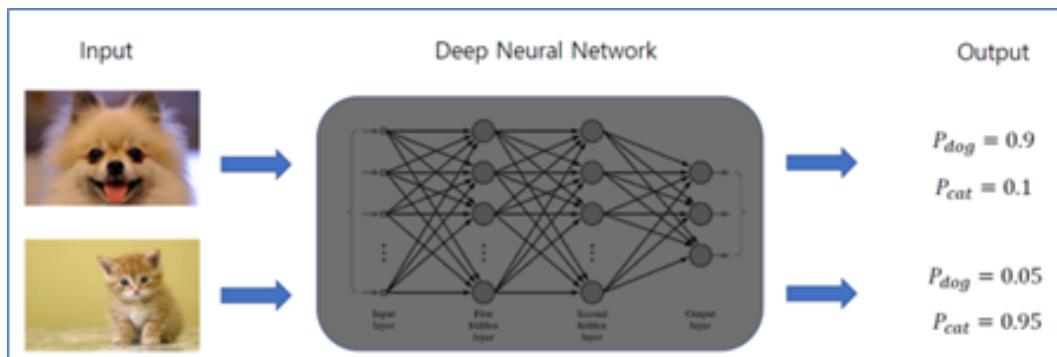
Pada Gambar 2.3, terlihat bahwa kucing (cat) dengan kotak pembatas dan tanpa kotak pembatas dapat membedakan mendasar antara klasifikasi

citra dan deteksi objek.



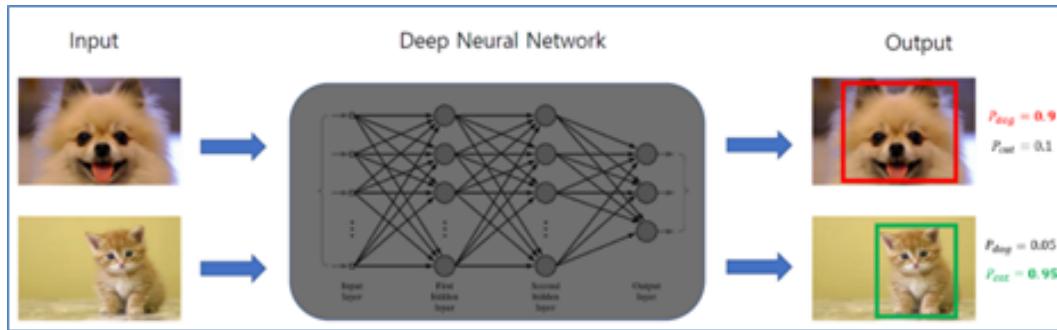
Gambar 2.3: Segmentation, Classification+Localization, Object Detection
(Sumber: A. Patel, 2020)

Dalam mempelajari deteksi objek, maka diperlukan mengetahui klasifikasi citra (*image classification*). Ketika gambar adalah input ke CNN, masalah mengklasifikasikan kelas yang sesuai dengan gambar dikenal sebagai klasifikasi gambar, dan seperti yang ditunjukkan pada Gambar 2.4, nilai probabilitas untuk semua kelas yang ditargetkan adalah keluaran.



Gambar 2.4: Image Classification
(Sumber: A. Patel, 2020)

Dapat juga dianggap bahwa deteksi objek sebagai masalah dimana tugas klasifikasi gambar memiliki tugas regresi yang memprediksi posisi objek menggunakan *bounding box* (kotak pembatas) pada Gambar 2.5.



Gambar 2.5: Object Detection dengan Bounding Box

(Sumber: A. Patel, 2020)

Masalah deteksi objek mengasumsikan bahwa beberapa kelas objek mungkin ada dalam gambar pada waktu yang sama. Dapat memvisualisasikan seperti dua jenis masalah, 1) klasifikasi multi label (beberapa kelas dalam satu gambar), 2) Bounding Box (masalah regresi) dimana harus memprediksi nilai koordinat kotak pembatas (*bounding box*) dalam bentuk x, y, w, h.

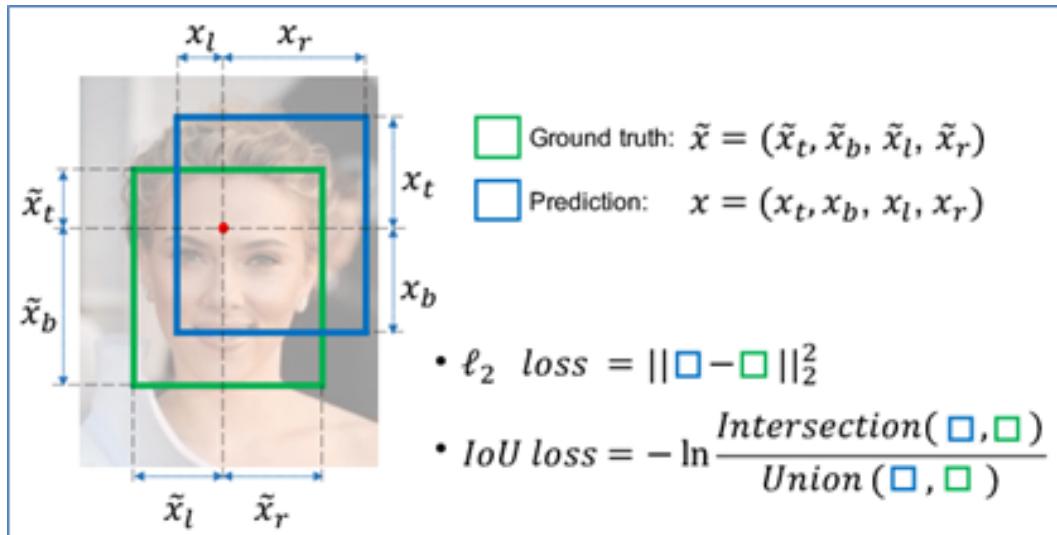
Dalam deteksi objek, terdapat *object localization* atau lokalisasi objek yang merupakan untuk memprediksi objek dalam sebuah citra serta batas-batasnya. Perbedaan antara lokalisasi objek dan deteksi objek tidak ken-tara. Sederhananya, lokalisasi objek bertujuan untuk menemukan objek utama (atau yang paling terlihat) dalam sebuah gambar, sedangkan deteksi objek mencoba untuk mengetahui semua objek dan batasannya.

Suatu klasifikasi citra atau model pengenalan citra hanya mendekripsi probabilitas suatu objek dalam suatu citra. Berbeda dengan ini, lokalisasi objek mengacu pada mengidentifikasi lokasi suatu objek dalam gambar. Algoritma lokalisasi objek akan menampilkan koordinat lokasi objek sehubungan dengan gambar. Dalam visi komputer, cara paling popular untuk melokalkan objek dalam gambar adalah dengan merepresentasikan lokasinya dengan bantuan kotak pembatas (*bounding box*).

Bounding box dapat diinisialisasi menggunakan parameter berikut:

- bx, by: koordinat pusat kotak pembatas (center of bounding box)

- bw: lebar kotak pembatas dengan lebar gambar (width)
- bh: tinggi kotak pembatas dengan tinggi gambar (height)



Gambar 2.6: Intersection Over Union (IoU)
(Sumber: A. Patel, 2020)

Dengan memprediksi ini, dapat menghitung Mean-IoU dan memprediksi kotak pembatas (*bounding box*) yang melokalkan objek di gambar.

- IoU adalah Intersection-Over-Union (IoU) disebut sebagai Indeks Jaccard (*Jaccard Index*) dianggap sebagai salah satu metrik kinerja yang paling banyak digunakan dalam deteksi objek.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Gambar 2.7: Persamaan IoU
(Sumber: A. Patel, 2020)

- IoU adalah area tumpang tindih (*overlap*) antara segmentasi yang diprediksi (*prediction*) dan kebenaran dasar (*ground truth*), seperti yang ditampilkan pada Gambar 2.7. Metrik ini bervariasi dari 0-1 (0-100%) dengan 0 menyiratkan tidak ada tumpeng tindih (sampah) dan 1 menandakan segmentasi yang tumpang tindih sempurna (*fat dub*).
- Mean IoU adalah segmentasi biner (dua kelas) atau multi-kelas, mean Io Udari gambar dihitung dengan mengambil IoU dari setiap kelas dan merata-ratakannya.

2.6 Machine Learning

Istilah *machine learning* mula-mula diperkenalkan oleh Arthur Samuel pada tahun 1959 melalui jurnalnya yang berjudul "Some Studies in Machine Learning Using the Game of Checkers". (IBM Journal of Research and Development). Samuel mencoba mengajari program komputer untuk bermain catur. Tujuannya adalah membuat agar komputer dapat bermain catur lebih baik dari dirinya. Pada tahun 1962 program buatannya dapat mengalahkan juara catur dari negara bagian Connecticut (R. Primartha, 2018).

Machine learning membutuhkan sebuah model yang didefinisikan berdasar parameter-parameter tertentu. Proses learning adalah eksekusi program komputer untuk mengoptimasi parameter-parameter dari model tersebut, dengan memanfaatkan data training atau *past experience*.

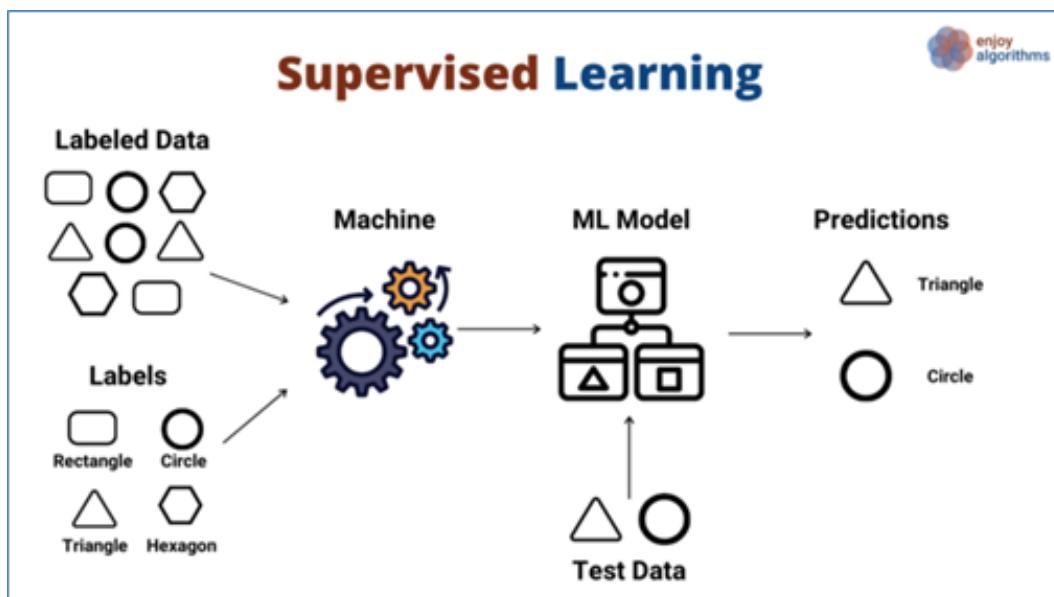
Jadi, secara sederhana dapat dijelaskan bahwa *machine learning* adalah pemrograman komputer untuk mencapai kriteria/performa tertentu dengan menggunakan sekumpulan data training atau pengalaman di masa lalu (*past experience*). *Machine learning* mempelajari teori agar komputer mampu "belajar" dari data.

Secara umum algoritma *machine learning* dapat dikelompokkan menjá-

di lima bagian, yaitu *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *reinforcement learning*, dan *deep learning*.

2.6.1 Supervised Learning

Sebagian besar praktik machine learning mengandalkan algoritma *supervised learning*. Algoritmanya dinamakan seperti ini karena training dataset (sekumpulan data untuk training) akan memandu dan mengajari komputer agar menghasilkan outcome sesuai harapan. Pada *supervised learning* menggunakan sebuah algoritma untuk mempelajari mapping function antara input dengan output. Berbagai kemungkinan output sudah diketahui dan data-data yang digunakan untuk latihan (training) sudah diberi label dengan jawaban yang benar. *Supervised learning* dapat bermanfaat untuk memprediksi sesuatu dengan bantuan training dataset. Berikut skema *supervised learning* pada Gambar 2.8 (R. Primartha, 2018).



Gambar 2.8: Skema Supervised Learning
(Sumber: M. Kozan, 2021)

Supervised learning menggunakan training data yang sudah diberi label untuk mempelajari *mapping function*, dari input variables (x) ke output

variables (y).

$$y = f(x)$$

Sebagai contoh, sebuah algoritma klasifikasi akan dapat mengidentifikasi berbagai bentuk bangun setelah melalui proses belajar dari sekumpulan bangun datar yang sudah ditandai atau diberi label dengan ciri tertentu seperti pada gambar 2.12.

Permasalahan-permasalahan yang terkait dengan *supervised learning* dapat dikategorikan menjadi dua jenis:

1. *Classification*

Klasifikasi bertujuan untuk memprediksi outcome dari input (sample yang diberikan), dimana output variabel berbentuk kategori-kategori. Contoh: pria/wanita, sakit/sehat, tinggi/rendah, dan sebagainya.

2. *Regression*

Regression bertujuan untuk memprediksi outcome dari input (sample yang diberikan), dimana output, variabel berbentuk nilai aktual (*real values*). Contoh: prediksi harga rumah, tinggi badan seseorang, curah hujan, dan sebagainya.

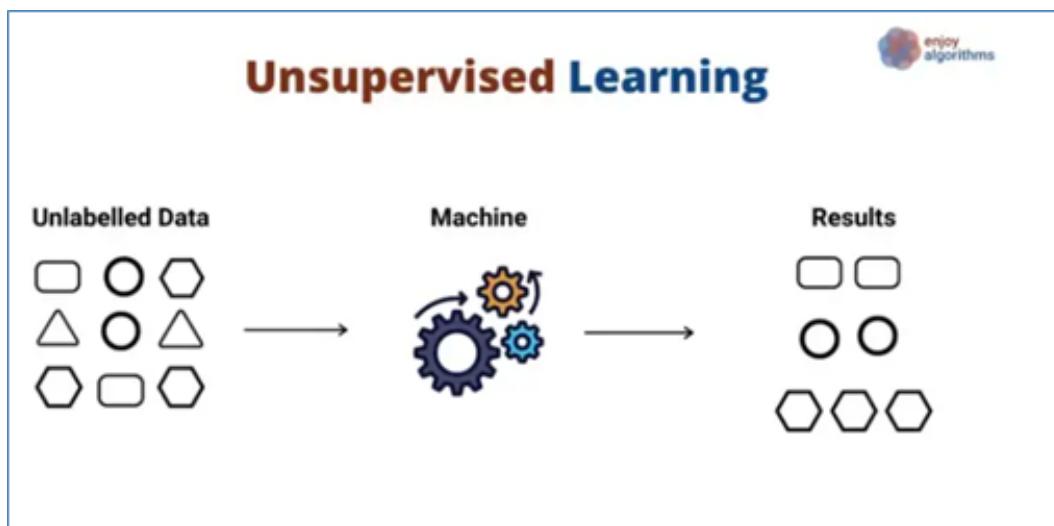
Ada beberapa algoritma yang sudah dikembangkan dan terkait dengan *supervised learning*, diantaranya:

1. Decision tree,
2. Naïve Bayes Classifier,
3. Artificial Neural Network,
4. Support Vector Machine,
5. Linear Regression,

6. Logistic Regression,
7. CART,
8. KNN (-KNearest Neighbor), dsb.

2.6.2 Unsupervised Learning

Berbeda dengan supervised learning, pada *unsupervised learning* persoalan diproses hanya mengandalkan data yang belum dilatih sebelumnya. *Unsupervised learning* menggunakan *unlabeled training dataset* untuk memodelkan struktur dari data, sehingga unsupervised learning bersifat lebih subjektif dibandingkan *supervised learning*. Berikut skema *unsupervised learning* pada Gambar 2.9 (R. Primartha, 2018).



Gambar 2.9: Skema Unsupervised Learning
(Sumber: M. Kozan, 2021)

Unsupervised learning bermanfaat untuk kasus-kasus dimana kita ingin menemukan relasi implisit (implicit relationships) dari *unlabeled dataset* yang disediakan. Jadi, pada *unsupervised learning* kita tidak memprediksi masa depan, sebab input variable (X) tidak memiliki relasi dengan output variabel

(Y).

$$f(x)$$

Untuk memudahkan memahaminya, dapat diasumsikan saat ini belum pernah membeli majalah sama sekali. Suatu ketika membeli beberapa buah majalah dan ingin membaginya menjadi beberapa kategori, dengan tujuan agar nantinya mudah dicari. Maka, dapat dimulai dengan mengidentifikasi majalah-majalah berdasarkan kemiripan. Misalnya, berdasarkan isi, penerbit, dan lain-lain yang bisa ditentukan sesuai kebutuhan.

Permasalahan seputar *unsupervised learning* dapat dikelompokkan menjadi tiga kategori, yaitu:

1. Association

Association bertujuan untuk menemukan peluang (probabilitas) berdasarkan keterkaitan (co-occurrence) dari item-item dalam sebuah kumpulan. Sebuah contoh, jika customer membeli teh celup, maka kemungkinan besar (sekitar 80

2. Clustering

Clustering bertujuan untuk mengelompokkan sample dalam cluster yang sama berdasarkan kemiripan (similarity).

3. Dimensionality Reduction

Dimensionality Reduction berarti mengurangi sejumlah variabel dari dataset namun tetap memastikan informasi yang penting masih tersedia. Dimensionality Reduction dapat diwujudkan menggunakan metode:

(a) Feature Extraction

Melakukan transformasi data dari dimensi tinggi (a high-dimensional space) ke dimensi yang lebih rendah (a low-dimensional space).

(b) Feature Selection

Memilih Sebagian saja (subset) dari variabel asal (original variabel).

Beberapa algoritma yang dikelompokkan dalam *unsupervised learning*, antara lain:

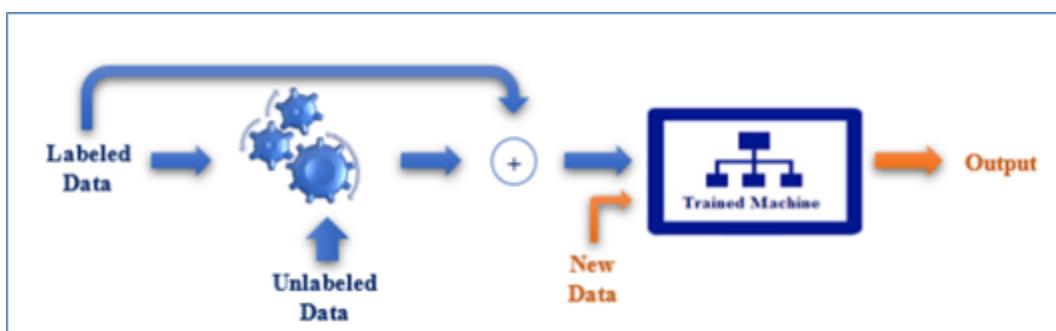
1. K-Means
2. Hierarchical Clustering
3. DBSCAN
4. Fuzzy C-Means
5. Self-Organizing Map, dan sebagainya.

2.6.3 Semi-supervised learning

Metode ini berada diantara yang *supervised* dan *unsupervised learning* di mana memiliki sejumlah besar masukan data, beberapa di antaranya diberi label dan sisanya tidak. Banyak masalah pembelajaran kehidupan nyata termasuk dalam bidang pembelajaran mesin ini. Alasannya adalah *semi-supervised* membutuhkan lebih sedikit intervensi manusia karena menggunakan data berlabel dalam jumlah yang sangat kecil dan data yang tidak berlabel dalam jumlah besar. Memanfaatkan kumpulan data yang kurang berlabel lebih menarik karena kumpulan data tersebut sangat sulit untuk dikumpulkan serta mahal dan mungkin memerlukan akses ke pakar domain. Dataset yang tidak berlabel di sisi lain lebih murah dan lebih mudah diakses (X. Zhu, 2018).

Kedua teknik pembelajaran *supervised* dan *unsupervised learning* bisa digunakan untuk melatih algoritma pembelajaran dalam pembelajaran *semi-supervised*. Teknik *unsupervised learning* dapat digunakan untuk mengungkap struktur dan pola tersembunyi dalam kumpulan data input. Sedangkan teknik

supervised learning dapat digunakan untuk membuat prediksi tebakan pada data yang tidak berlabel, memasukkan data kembali ke algoritma pembelajaran sebagai data pelatihan, dan menggunakan pengetahuan yang diperoleh untuk membuat prediksi pada kumpulan data baru. Dengan demikian, dapat mengatakan bahwa data yang tidak berlabel digunakan untuk memodifikasi atau memprioritaskan kembali prediksi atau hipotesis yang diperoleh dari data yang berlabel. Gambar 2.10 mengilustrasikan berbagai tahapan metode *semi-supervised learning*.



Gambar 2.10: Skema Semi-Supervised Learning
(Sumber: A. B. Nassif et al, 2019)

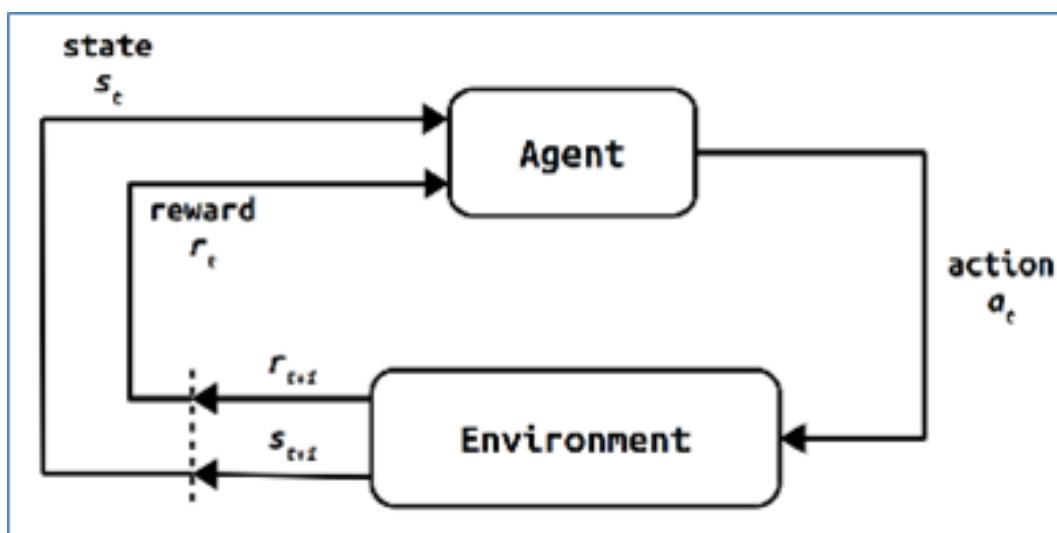
Untuk memanfaatkan data pelatihan yang tidak berlabel, semua algoritma *semi-supervised learning* melakukan setidaknya satu dari asumsi berikut asumsi kehalusan, asumsi cluster, dan asumsi manifold.

2.6.4 Reinforcement Learning

Reinforcement learning merupakan metode pembelajaran yang dipengaruhi oleh feedback dari lingkungan dengan Teknik pembelajaran yang iterative (berulang-ulang) dan adaptive (menyesuaikan). *Reinforcement learning* dipercaya mendekati cara manusia belajar (R. Primartha, 2018).

Reinforcement learning (RL) diinspirasi oleh kebiasaan makhluk hidup dalam belajar dan bertindak, khususnya manusia. Pada RL tidak ada dataset. Data-data diperoleh berdasarkan pengalaman. Algoritma *reinforcement learning*

ing mengijinkan agent untuk memutuskan aksi selanjutnya berdasarkan kondisi saat ini (*current state*). *Reinforcement learning* kadang disebut juga *credit assessment learning*, sebab learning difokuskan untuk memaksimalkan perolehan *rewards*. *Reinforcement learning* tergantung pada proses coba-coba untuk mengungkap rangkaian tindakan yang memaksimalkan metrik imbalan kumulatif, yang digunakan untuk membuat algoritma memahami apakah itu mengearah ke arah yang benar atau tidak. Berikut skema *reinforcement learning* pada Gambar 2.11.



Gambar 2.11: Skema Reinforcement Learning

(Sumber: R. Sutton, 1998)

Menurut R. Sutton, proses *reinforcement learning* dapat dipresentasikan dalam matematika sebagai *Markov Decision Process* (MDP), memperkenalkan 4 set S, A, P, R, dengan:

S - Kumpulan status tempat agen dapat berada, pada saat tertentu;

A - Serangkaian kemungkinan Tindakan yang dapat dilakukan agen dalam waktu tertentu;

P - Himpunan probabilitas, bahwa sebuah agen, yang berada dalam keadaan s, bertransisi ke keadaan s' dengan melakukan Tindakan A dalam waktu t+1;

Representasi *reinforcement learning* mirip dengan supervised learning.

Yang membedakan adalah pada reinforcement learning tidak hanya x, namun x dan z.

$$y = f(x) \text{ given } z$$

Tidak seperti *supervised* dan *unsupervised learning* dimana algoritma sudah memiliki tujuan (goal). Algoritma *reinforcement learning* tidak memiliki tujuan eksplisit, sebagai gantinya algoritma dipaksa untuk belajar menemukan nilai optimal melalui kegiatan trial dan error.

Reinforcement learning banyak diimplementasikan pada *game theory*, *control theory*, *operation research*, *information theory*, *simulation-based optimization*, *multi-agent systems*, *swarm intelligence*, *statistics*, dan *genetic algorithm*.

Contoh penerapan *reinforcement learning* yaitu pada bidang robotic. Sebuah robot dapat belajar untuk menghindari tabrakan dengan cara menerima feedback negative manakala robot tersebut menabrak halangan tertentu. Robot akan dibiarkan berjalan tanpa dipandu. Robot akan belajar dari pengalaman sebelumnya untuk menemukan rute paling optimal.

Beberapa algoritma yang dikelompokkan dalam *reinforcement learning* antara lain:

1. Genetic Algorithm (GA)
2. Dynamic Programming (DP)
3. Generalized Policy Iteration (GPI)
4. Monte Carlo Methods

2.6.5 Deep Learning

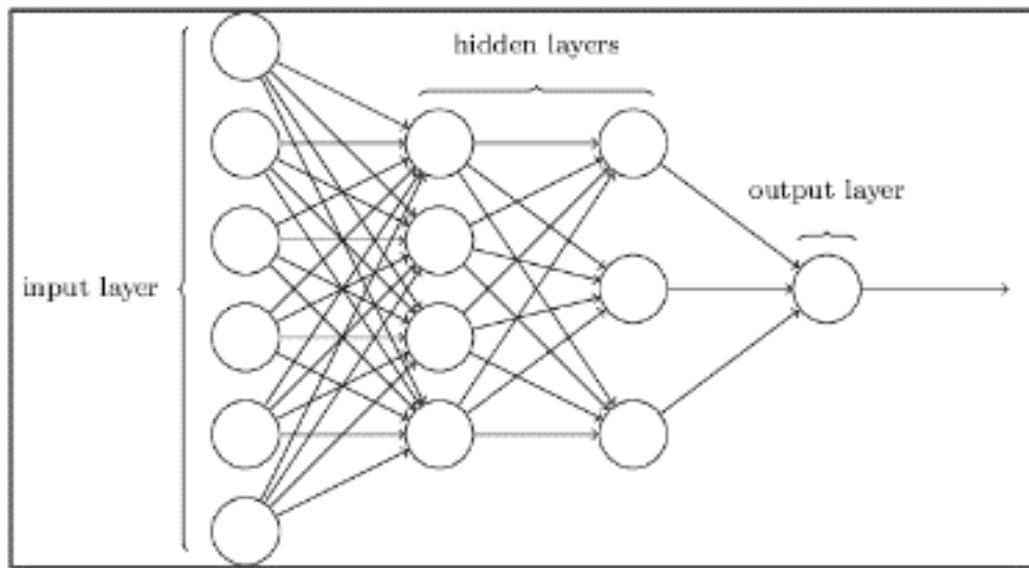
Deep learning merupakan metode pembelajaran yang memanfaatkan *artificial neural networks* yang berlapis-lapis (multi layer). *Artificial neu-*

ral networks ini dibuat mirip dengan otak manusia, di mana neuron-neuron terkoneksi satu sama lain, sehingga membentuk sebuah jaringan neuron yang sangat rumit (R. Primartha, 2018).

Deep learning atau *deep structured leaning* atau *hierarchical learning* atau *deep neural* merupakan metode pembelajaran yang memanfaatkan multiple non-linear transformation. *Deep learning* dapat dipandang sebagai gabungan machine learning dengan *artificial intelligence* (AI). *Deep learning* pada hakekatnya merupakan perluasan atau pengembangan dari *neural network* atau jaringan saraf tiruan (JST).

Jika dikembalikan kepada tujuan *machine learning* semula, yaitu komputer yang dapat belajar (dari data atau pengalaman), maka *deep learning* adalah apa yang selama ini dicari. *Deep learning* menirukan cara berpikir manusia. Pada *deep learning*, komputer harus memproses data yang sangat banyak, berlapis-lapis, dan output dari layer sebelumnya akan menjadi input bagi layer sesudahnya.

Struktur umum dan dasar dari skema *deep learning* ditunjukkan pada Gambar 2.12. Ini terdiri dari lapisan masukan, yang merupakan data masukan ke algoritma; lapisan tersembunyi, di mana algoritma membuat banyak perhitungan matematis, dan lapisan output, yang merupakan hasil dari perhitungan algoritma.

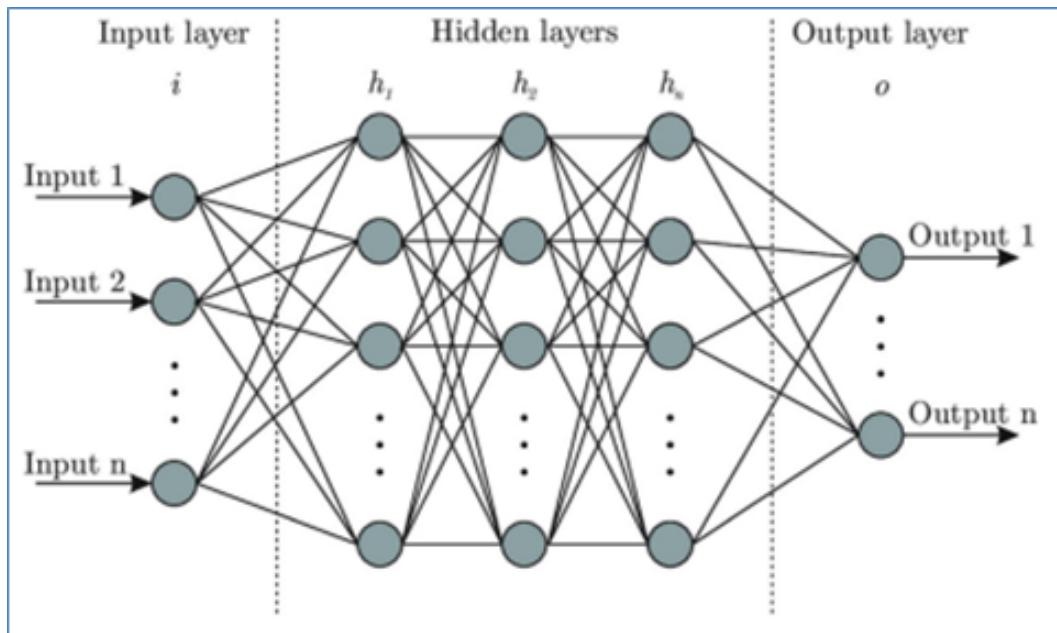


Gambar 2.12: Skema Umum *Deep Learning*
(Sumber: A. Feizollah et al, 2022)

Sejarah *deep learning* dimulai pada tahun 2006, yaitu setelah Geoffrey Hinton mempublikasikan paper yang memperkenalkan salah satu varian *neural networks* yang disebut *deep belief nets*. Paper ini merupakan awal kemunculan istilah *deep learning*, untuk membedakan arsitektur *neural network* konvensional (*single layer*) dengan arsitektur neural network multi/banyak layer. Dengan kata lain, *deep learning* adalah salah satu cabang machine learning yang menggunakan *deep neural network* untuk menyelesaikan permasalahan pada domain *machine learning*.

Pada tahun 2009, Andrew memperkenalkan penggunaan GPU untuk *deep learning* melalui paper yang berjudul *large-scale deep unsupervised learning using graphics processors*. Dengan menggunakan GPU, algoritma *deep learning* dapat dijalankan lebih cepat dibanding dengan tanpa GPU (hanya menggunakan CPU). Perkembangan *deep learning* maju pesat berkat keberadaan *hardware* yang memadai. Dan saat ini, *deep learning* sudah banyak diaplikasikan di berbagai area, seperti pengenal wajah, *self-driving car*, pengenal suara, dan sebagainya.

Deep learning merupakan jalan untuk mencapai apa yang sudah dicita-citakan sebelumnya oleh manusia, yaitu kecerdasan buatan bagi mesin. Bentuk diagram *network model deep learning* seperti pada Gambar 2.13. Perhatikan bahwa *hidden layer* hanya digambarkan tiga lapis saja, padahal kenyataannya bisa berjumlah sangat banyak, dapat diasumsikan seperti Gambar 2.13.



Gambar 2.13: Skema *Deep Learning* dengan Penambahan beberapa *hidden layer*

(Sumber: H. Kaur et al, 2021)

Pada Gambar 2.13 mengandung 3 layer, yaitu input, *hidden* dan output layer. Penambahan *layer* ini terjadi pada *hidden layer*. *Hidden layer* pada skema *deep learning* yang disebut dengan *Multi Layer Perceptron* (MLP) disebabkan jumlah neuron semakin banyak dan itu artinya semakin banyak juga perhitungan yang harus dikerjakan pada setiap *layer*. MLP merupakan pengembangan dari *Single Layer Perceptron* (SLP) yang merupakan model paling sederhana dari neural network dan sekaligus merupakan dasar bagi model-model tingkat lanjut yang digunakan pada *deep learning*. MLP kemudian menjadi cikal bakal metode *deep learning* atau *deep neural network* (DNN).

Deep learning sudah dikembangkan ke berbagai model atau arsitektur yang berbeda-beda. Berikut daftar beberapa model atau arsitektur untuk *deep learning*.

1. *Recurrent Neural Networks* (RNN)
2. *Long Short-Term Memory* (LSTM)
3. *Convolutional Neural Network* (CNN)
4. *Deep Believe Networks* (DBN)
5. *Deep Stacking Networks* (DSN)

Contoh penerapan masing-masing arsitektur deep learning dapat dipelajari pada Tabel 2.1

Tabel 2.1: Penerapan Arsitektur Deep Learning

No.	Arsitektur	Penerapan
1	RNN	<i>Speech recognition, handwriting recognition</i>
2	LSTM	<i>Natural language text compression, handwriting recognition, speech recognition, gesture recognition, image captioning</i>
3	CNN	<i>Image recognition, video analysis, natural language processing</i>
4	DBN	<i>Image recognition, information retrieval, natural language understanding, failure prediction</i>
5	DSN	<i>Information retrieval, continuous speech recognition</i>

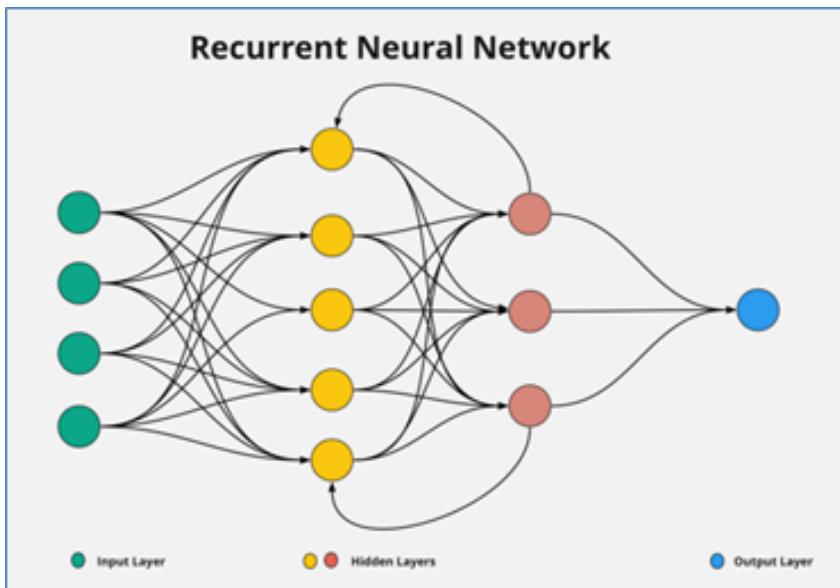
Masing-masing arsitektur pada Tabel 2.1 memiliki perbedaan, berikut penjelasan dan diagram network beberapa arsitektur deep learning yang umum.

2.6.5.1 Recurrent Neural Networks (RNN)

Recurrent Neural Network (RNN) merupakan arsitektur *deep learning* yang popular serta sangat menjanjikan untuk menyelesaikan berbagai persoalan yang terkait dengan *Natural Language Processing* (NLP). Model RNN digunakan agar mesin dapat memahami bahasa manusia. Mulai dari cara berkomunikasi, mendengarkan, mengenali percakapan, hingga memahami tata

bahasa dan aksen. RNN juga dapat diimplementasikan untuk mengenali gambar-gambar atau objek (R. Primartha, 2018).

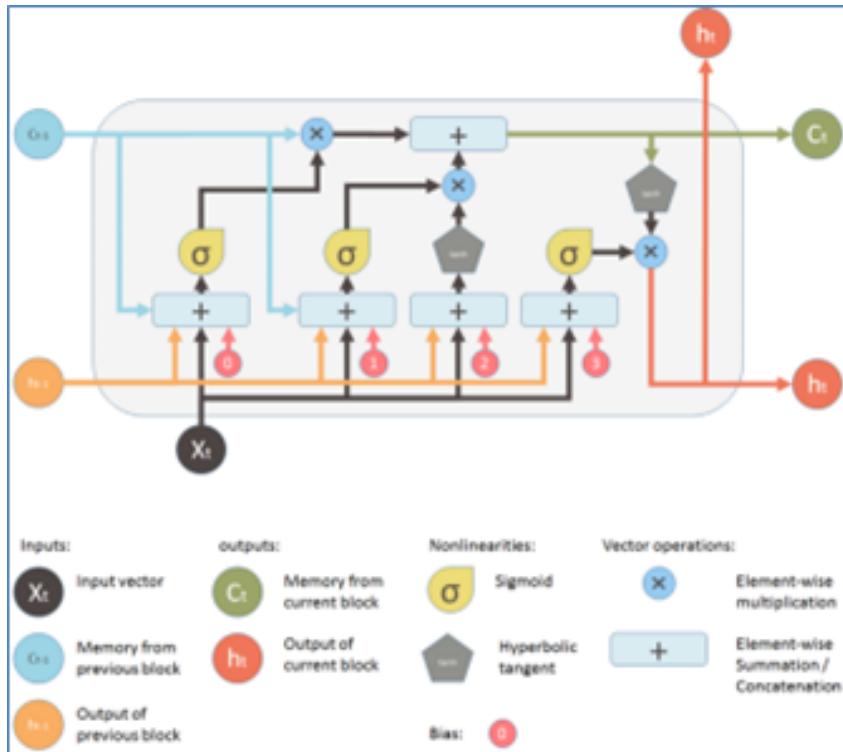
Diagram network RNN seperti pada Gambar 2.14 berikut.



Gambar 2.14: Diagram *Recurrent Neural Network* (RNN)
(Sumber: K. Dass, 2020)

2.6.5.2 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) merupakan building unit untuk *layer-layer* pada recurrent neural network (RNN). LSTM mula-mula diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. LSTM juga banyak diimplementasikan pada bidang NLP. Boleh dibilang LSTM merupakan pengembangan dari RNN. Secara teoritis, jaringan saraf yang terhubung secara naif, yang disebut jaringan saraf berulang, dapat bekerja. Namun dalam praktiknya, mengalami dua masalah: gradien menghilang dan gradien meledak, yang membuatnya tidak dapat digunakan (R. Primartha, 2018). Kemudian, LSTM ditemukan untuk mengatasi masalah ini dengan secara eksplisit memasukkan unit memori, yang disebut sel ke dalam jaringan. Ini adalah diagram blok bangunan LSTM pada Gambar 2.15.



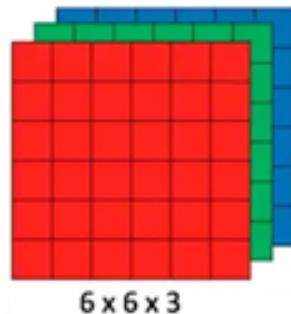
Gambar 2.15: Diagram *Long Short Term Memory* (LSTM)
(Sumber: S. Yan, 2016)

2.6.5.3 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN atau ConvNet) merupakan salah satu model deep learning yang banyak digunakan untuk keperluan analisis citra/visual. CNN adalah salah satu kategori utama untuk melakukan pengenalan dan klasifikasi gambar, deteksi objek, pengenalan wajah, dan sebagainya merupakan beberapa area dimana CNN banyak digunakan (R. Primartha, 2018).

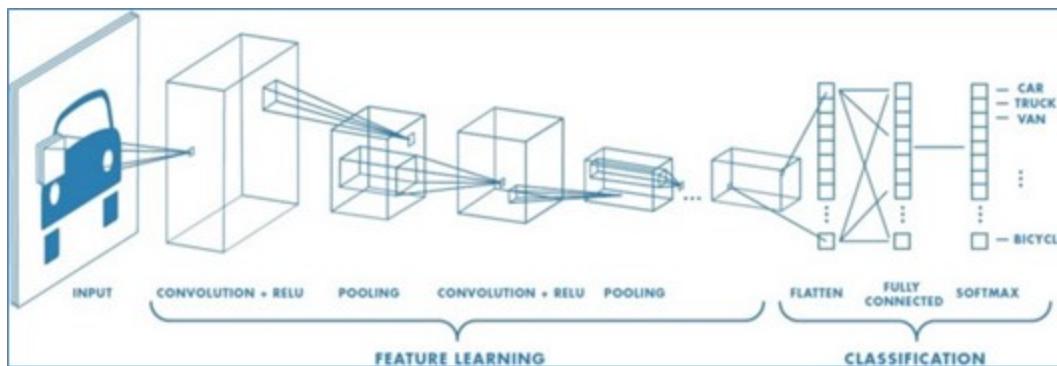
Klasifikasi gambar CNN mengambil input gambar, memproses, dan mengklasifikasikannya dalam kategori tertentu, misalnya kucing, harimau, singa. Komputer melihat gambar input sebagai susunan piksel dan itu tergantung pada resolusi gambar. Berdasarkan resolusi gambar, akan terlihat $h \times w \times d$ (h = Tinggi, w = Lebar, d = Dimensi). Misalnya, gambar array matriks RGB $6 \times 6 \times 3$ (3 mengacu pada nilai RGB) dan gambar array matriks $4 \times 4 \times 1$

dari gambar skala abu-abu, seperti pada Gambar 2.16.



Gambar 2.16: Array dari Matriks RGB
(Sumber: R. Prabhu, 2018)

Secara teknis, model *deep learning* CNN untuk latih dan uji, setiap gambar input akan melewati serangkaian lapisan konvolusi dengan filter (Kernel), Pooling, *fully connected layers* (FC) dan menerapkan fungsi softmax untuk mengklasifikasikan objek dengan nilai probabilistic antara 0 dan 1. Gambar 2.17. adalah alur dari CNN untuk memproses gambar input dan mengklasifikasikan objek berdasarkan nilai.



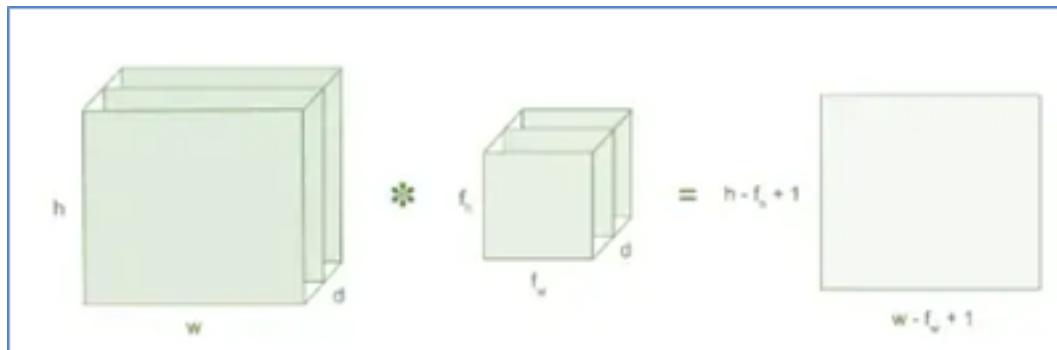
Gambar 2.17: Neural Network dengan banyak Convoluti Layer
(Sumber: R. Prabhu, 2018)

1. *Convolution Layer*

Konvolusi adalah lapisan pertama untuk mengekstraksi fitur dari gambar masukan. Konvolusi mempertahankan hubungan antara piksel dengan mempelajari fitur gambar menggunakan kotak kecil data masukan. Ini

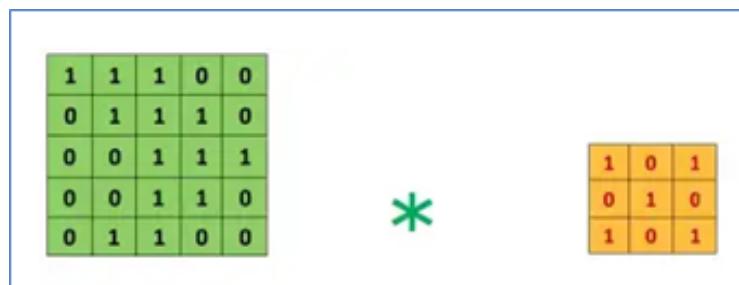
adalah operasi matematika yang mengambil dua input, seperti matriks gambar dan filter atau kernel.

- Sebuah gambar matriks (volume) dari dimensi $(h \times w \times d)$
- Sebuah filter ($f_h \times f_w \times d$)
- Output volume dimensi $(h - f_h + 1) \times (w - f_w + 1) \times 1$



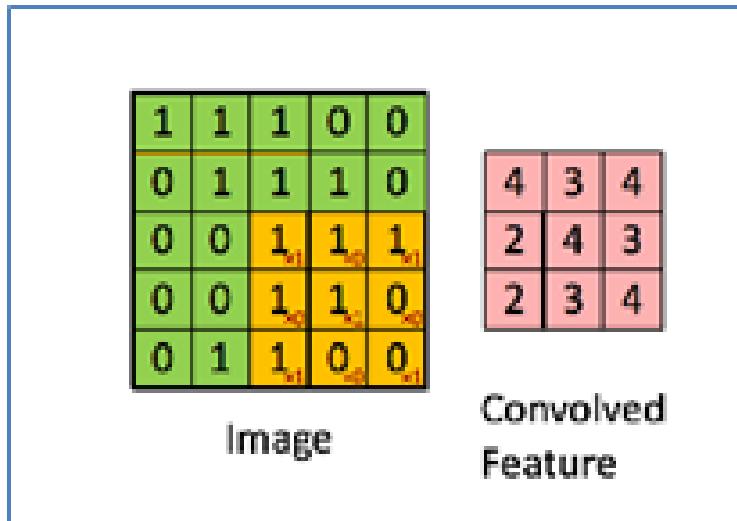
Gambar 2.18: Gambar Matriks Multiplies Kernel atau Filter Matriks
(Sumber: R. Prabhu, 2018)

Pertimbangkan gambar 5×5 yang nilai piksel gambarnya adalah 0, 1 dan matriks filter 3×3 , seperti pada Gambar 2.19.



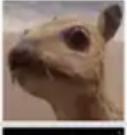
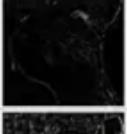
Gambar 2.19: Gambar matriks 5×5 dikalikan dengan Filter matiks 3×3

Kemudian, konvolusi matriks gambar 5×5 dikalikan dengan filter matriks 3×3 yang disebut "Feature Map" sebagai output yang ditunjukkan pada Gambar 2.20.



Gambar 2.20: Output Matriks 3 x 3

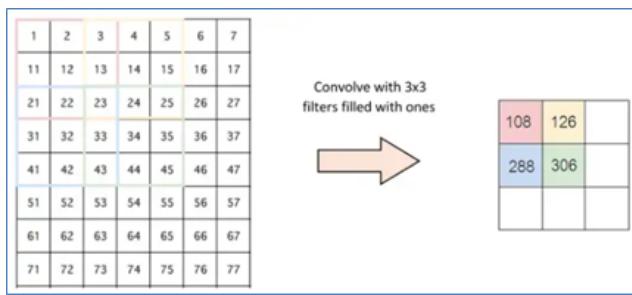
Konvolusi gambar dengan filter berbeda dapat melakukan operasi seperti deteksi tepi, mengaburkan, dan memeprtajam dengan menerapkan filter. Berikut ini contoh yang menunjukkan berbagai gambar konvolusi setelah menerapkan berbagai jenis filter (Kernel) pada Gambar 2.21.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Gambar 2.21: Beberapa Filter Umum
(Sumber: R. Prabhu, 2018)

2. Strides

Stride adalah jumlah piksel yang bergeser di atas matriks input. Saat langkahnya 1, maka memindahkan filter ke 1 piksel sekaligus. Saat langkahnya 2, maka memindahkan filter ke 2 piksel sekaligus dan seterusnya. Gambar 2.22 menunjukkan konvolusi akan bekerja dengan Langkah 2.



Gambar 2.22: Stride 2 Piksel
(Sumber: R. Prabhu, 2018)

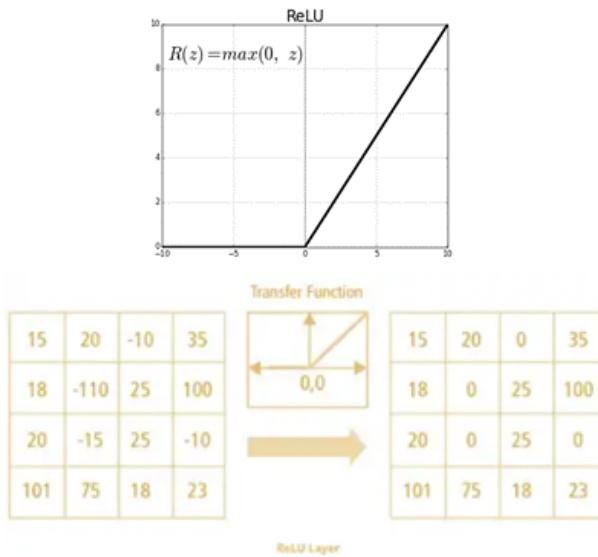
3. Padding

Pada saat penggunaan filter, terkadang filter tidak pas dengan gambar masukan. Maka, terdapat dua pilihan:

- Memadatkan gambar dengan angka nol (*zero padding*) agar pas;
- Menghilangkan bagian gambar yang tidak sesuai dengan filter. Ini disebut dengan *valid padding* yang hanya menyimpan bagian gambar yang valid.

4. Non Linearity (ReLU)

ReLU adalah singkatan dari *Rectified Linear Unit* untuk operasi non-linear. Outputnya adalah $f(x) = \max(0, x)$. ReLU penting karena tujuan ReLU adalah untuk mengenalkan non-linearitas di ConvNet, karena data dunia nyata ingin ConvNet pelajari adalah nilai linier non-negatif. Fungsi ini hanya mengembalikan nilai 0 jika nilai tersebut bernilai negatif, selain itu mengembalikan nilai yang sama dengan yang diberikan, tidak lain adalah menghilangkan keluaran negative dan mempertahankan nilai antara 0 hingga + tak terhingga, seperti pada Gambar 2.23.

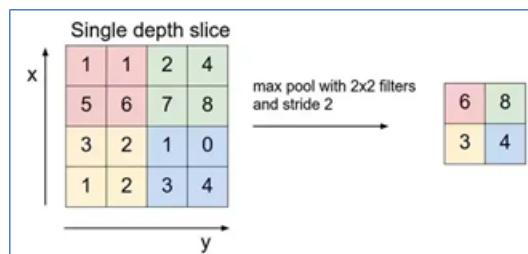


Gambar 2.23: Operasi ReLU
(Sumber: P. Ratan, 2021)

5. Pooling Layer

Bagian layer pooling akan mengurangi jumlah parameter ketika gambar terlalu besar. Penyatuan spasial juga disebut subsampling atau down-sampling yang mengurangi dimensi setiap peta tetapi tetap mempertahankan informasi penting. Penyatuan spasial dapat dari berbagai jenis, diantaranya Max Pooling, Average Pooling, dan Sum Pooling.

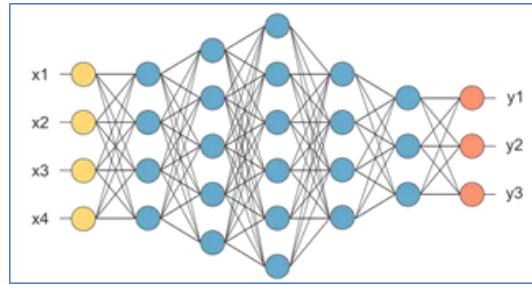
Max Pooling mengambil elemen tersbesar dari peta fitur yang diperbaiki. Mengambil elemen terbesar juga bisa mengambil pooling rata-rata. Jumlah semua elemen dalam peta fitur disebut sebagai kumpulan jumlah.



Gambar 2.24: Max Pooling

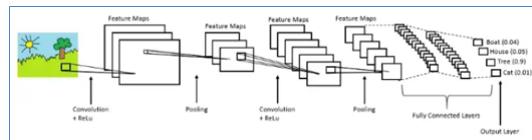
6. Fully Connected Layer

Lapisan yang disebut *Fully Connected Layer*, diratakan matriks menjadi vector dan memasukkannya ke dalam *fully connected layer*, seperti jaringan saraf (*neural network*), seperti Gambar 2.25.



Gambar 2.25: Setelah Pooling Layer Diratakan sebagai FC Layer

Pada Gambar 2.25, matriks peta fitur akan diubah menjadi vector (x_1 , x_2 , x_3 , ...). Dengan lapisan yang terhubung sepenuhnya, digabungkan fitur ini bersama untuk membuat model. Setelah itu, akhirnya memiliki fungsi aktivasi seperti softmax atau sigmoid untuk mengklasifikasikan keluaran sebagai objek, misalnya rumah, pohon, kucing, mobil, truk, dan sebagainya, seperti pada Gambar 2.26.

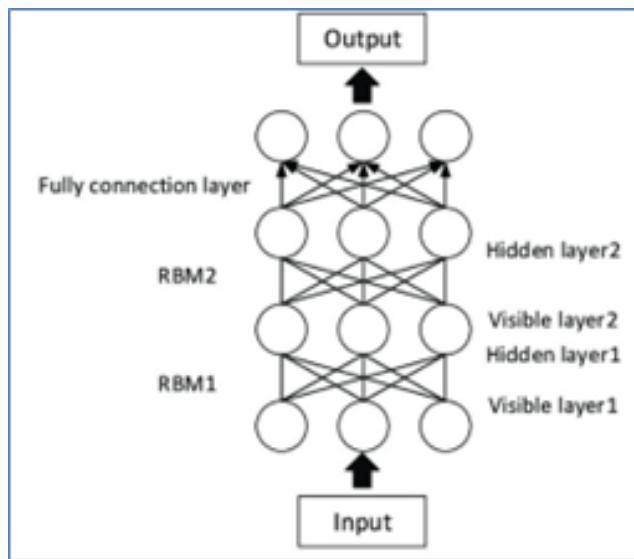


Gambar 2.26: Arsitektur CNN Lengkap

2.6.5.4 Deep Believe Networks (DBN)

Deep Belief Networks (DBN) merupakan model *deep learning* yang memanfaatkan tumpukan/*stack Restricted Boltzmann Machines* (RBM) atau kadangkala *Autoencoders*. *Autoencoders* adalah model *neural networks* yang memiliki input dan output yang sama. *Autoencoder* mempelajari data input dan

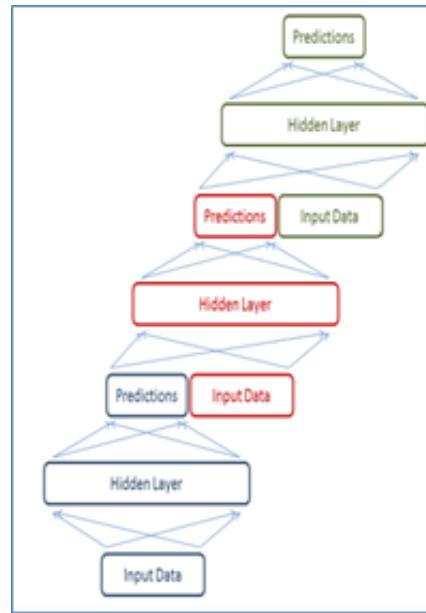
berusaha untuk melakukan rekonstruksi terhadap data input tersebut (R. Primartha, 2018). Skema diagram DBN seperti pada Gambar 2.27.



Gambar 2.27: Skema Diagram DBN
(Sumber: H. Liu & B. Lang, 2019)

2.6.5.5 Deep Stacking Networks (DSN)

Salah satu masalah pada *deep learning* adalah proses learning sangat sulit dilakukan dan memerlukan komputasi yang cukup kompleks. Pada tahun 2011 Deng Yu mengusulkan model *Deep Convex Networks* (DCN) atau *Deep Stacking Network* (DSN), yang sedikit berbeda dibandingkan model *deep learning* lain (R. Primartha, 2018). Secara umum model DSN terdiri atas ub-nets berukuran kecil dengan hanya sebuah hidden layer, seperti pada Gambar 2.28.



Gambar 2.28: Skema Diagram DSN
(Sumber: L. Deng et al, 2012)

Model *deep learning* yang popular lainnya adalah Region Based CNN, Google Net, Generative Adversarial Network (GAN), dan You Only Look Once (YOLO).

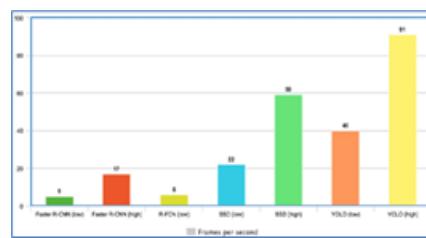
2.7 You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah algoritma deteksi objek real-time yang diperkenalkan pada tahun 2015 oleh Joseph Redmon, Santosh Divvala, Ross Girshick dan Ali Farhadi dalam paper dengan judul "You Only Look Once: Unified, Real-Time Object Detection" (J. Redmon et al., 2015). Penulis membingkai masalah deteksi objek sebagai masalah regresi klasifikasi dengan memisahkan kotak pembatas (*bounding box*) secara spasial dan menghubungkan probabilitas ke masing-masing gambar yang terdeteksi menggunakan *convolutional neural network* (CNN). YOLO adalah *Convolutional Neural Network* (CNN) untuk melakukan deteksi objek secara real-time (V. Meel, 2022) (G. Boesch, 2022).

Beberapa alasan mengapa YOLO baik digunakan untuk deteksi objek real-time, diantaranya:

1. Kecepatan (*speed*)

YOLO sangat cepat karena tidak berurusan dengan jalur pipa (*pipelines*) yang rumit. YOLO dapat memproses gambar pada 45 frames per second (FPS). Selain itu, YOLO mencapai rata-rata presisi atau mean average precision (mAP) lebih dari dua kali dibandingkan dengan sistem real-time lainnya, yang menjadikannya kandidat yang bagus untuk pemrosesan real-time. Dari grafik pada Gambar 2.29 diamati bahwa YOLO jauh melampaui pendeksi objek lainnya dengan 91 FPS.



Gambar 2.29: Kecepatan YOLO dibandingkan dengan detector Objek Lainnya

(Sumber: S. A. S. Hernandez et al, 2020)

2. Akurasi deteksi tinggi (*high detection accuracy*)

YOLO jauh melampaui model *state-of-the-art* dalam akurasi dengan sedikit kesalahan latar belakang.

3. Generalisasi yang bagus (*good generalization*)

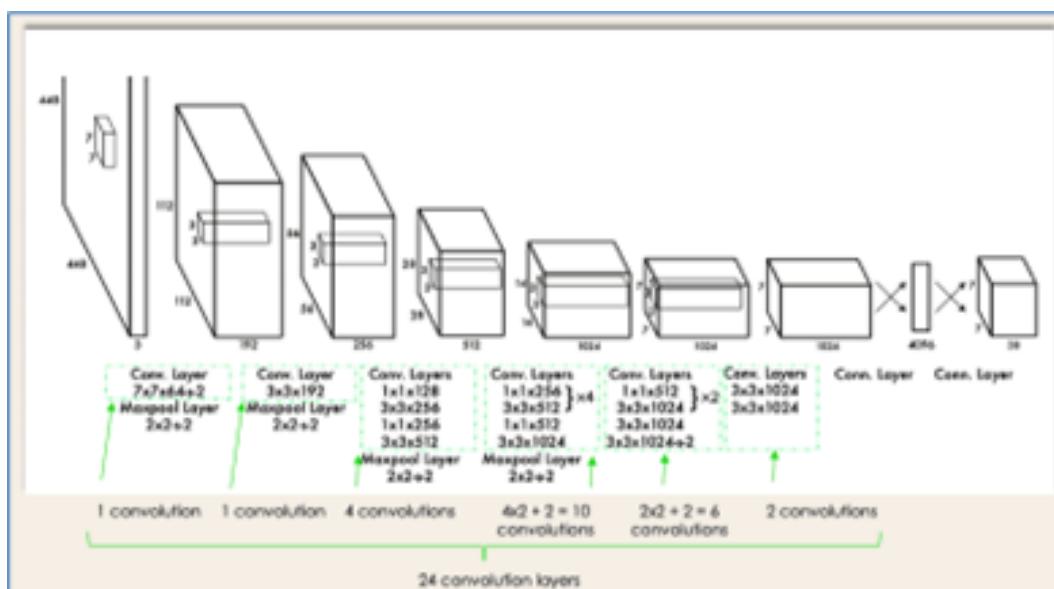
YOLO mendorong sedikit lebih jauh dengan memberikan generalisasi yang lebih baik untuk domain baru, yang menjadikannya bagus untuk aplikasi yang mengandalkan deteksi objek yang cepat dan kuat.

4. Sumber terbuka (*open-source*)

Membuat YOLO open-source membuat komunitas terus meningkatkan model. Inilah salah satu alasan mengapa YOLO telah melakukan begitu banyak perbaikan dalam waktu yang begitu terbatas.

2.7.1 Arsitektur YOLO

Arsitektur YOLO memiliki keseluruhan 24 lapisan konvolusional, empat lapisan penyatuan maksimum, dan dua lapisan yang terhubung sepenuhnya, arsitektur YOLO secara umum pada Gambar 2.30.



Gambar 2.30: Arsitektur YOLO dari *Original Paper*
(J. Redmon et al., 2015)

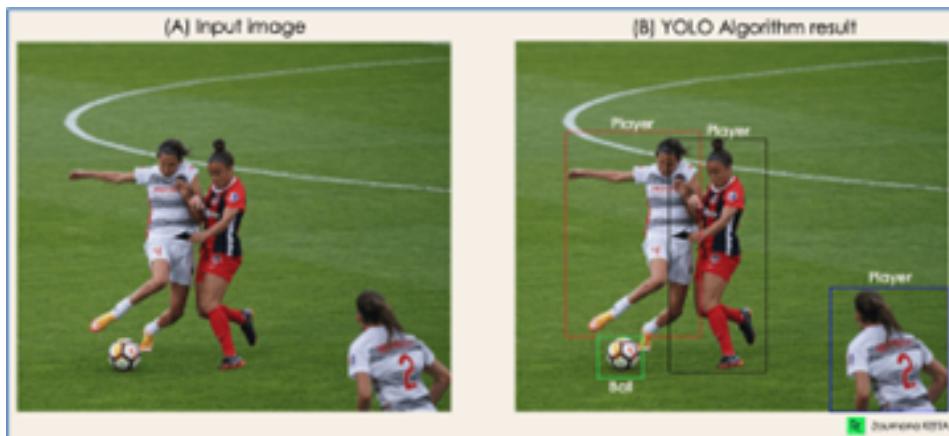
Arsitektur YOLO bekerja sebagai berikut:

- Mengubah ukuran gambar input menjadi 448x448 sebelum melalui *convolutional network*.
- Konvolusi 1x1 pertama kali diterapkan untuk mengurangi jumlah saluran, yang kemudian diikuti oleh konvolusi 3x3 untuk menghasilkan output kuboid.

- Fungsi aktivasi ReLU, kecuali lapisan terakhir, yang menggunakan fungsi aktivasi linier.
- Beberapa Teknik tambahan, seperti normalisasi batch dan dropout, masing-masing mengatur model dan mencegah overfitting.

2.7.2 Cara kerja Deteksi Objek YOLO

Berikut ini adalah proses bagaimana YOLO melakukan deteksi objek untuk mendapatkan gambar (b) dari gambar (a) pada Gambar 2.31.

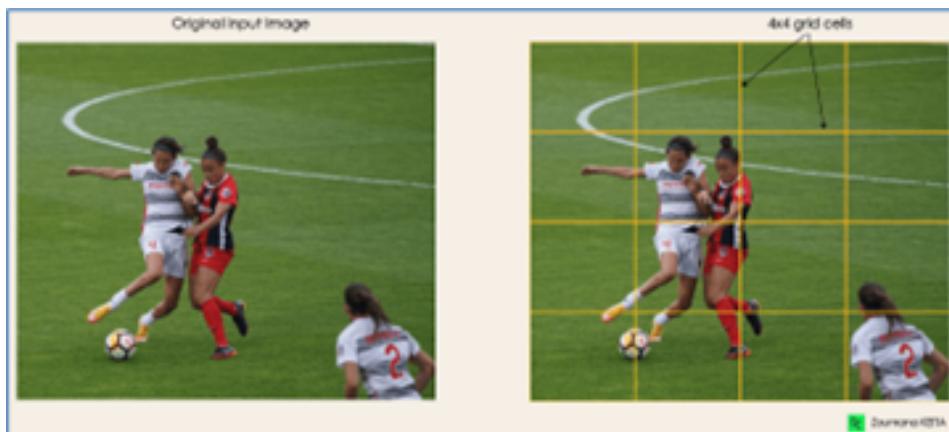


Gambar 2.31: (A) Input Image dan (B) Hasil Algoritma YOLO
(Sumber: Z. Kelta, 2022)

Algoritma YOLO bekerja berdasarkan empat pendekatan, sebagai berikut:

- (a) *Residual Blocks*(Blok Sisa)

Langkah pertama dimulai dengan membagi gambar asli (A) menjadi sel grid ($N \times N$) dengan bentuk yang sama, di mana N dalam hal ini adalah 4×4 grid sel pada Gambar 2.32. Setiap sel dalam grid bertanggung jawab untuk melokalkan dan memprediksi kelas objek yang dicakupnya, bersama dengan nilai probabilitas/kepercayaan.



Gambar 2.32: Residual Blocks
(Sumber: Z. Kelta, 2022)

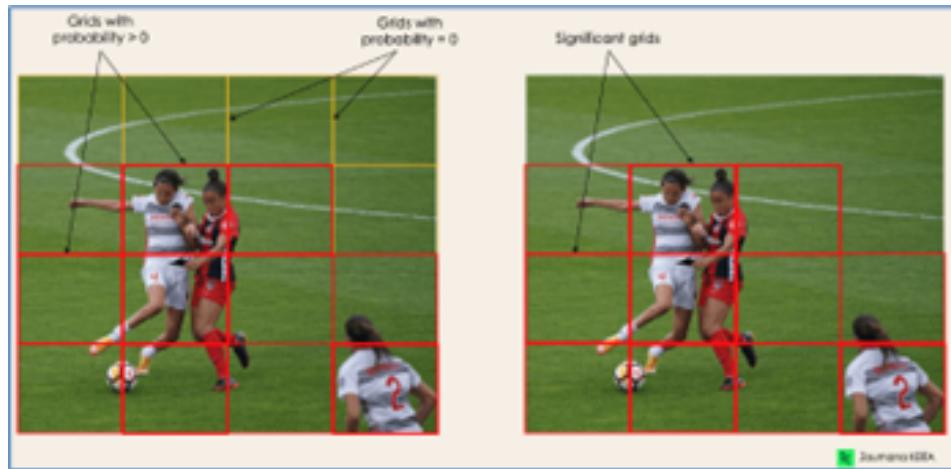
(b) *Bounding Box Regression* (Regresi Kotak Pembatas)

Langkah selanjutnya adalah menentukan kotak pembatas (*bounding box*) yang sesuai dengan persegi Panjang yang menyoroti semua objek dalam gambar. Dapat memiliki kotak pembatas sebanyak objek di dalam gambar yang diberikan. YOLO menentukan atribut kotak pembatas ini menggunakan modul regresi tunggal dalam format berikut, dimana Y adalah representasi vector terakhir untuk setiap kotak pembatas.

$$Y = [pc, bx, by, bh, bw, c1, c2]$$

Ini sangat penting selama fase pelatihan model.

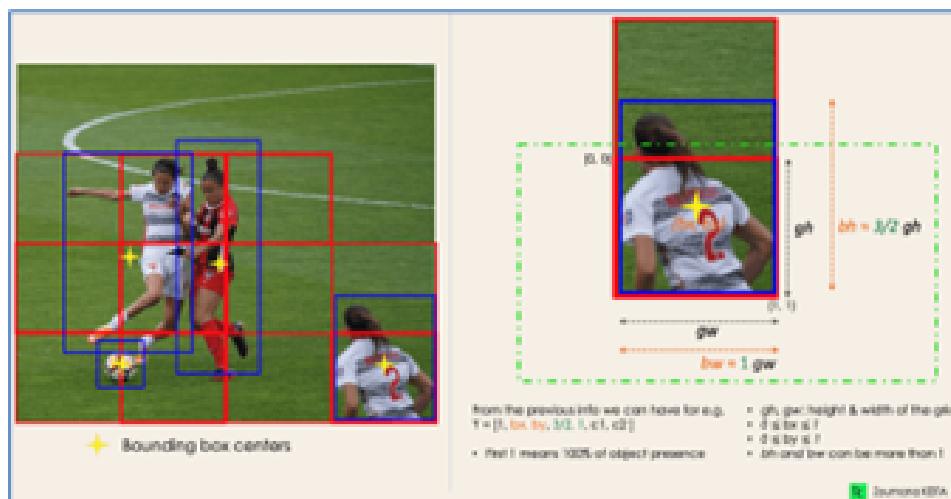
- pc sesuai dengan skor probabilitas dari grid yang berisi objek. Misalnya, semua grid yang berwarna merah akan memiliki skor probabilitas lebih tinggi dari nol. Gambar 2.33 adalah versi yang disederhanakan karena probabilitas setiap sel kuning adalah nol (tidak signifikan)



Gambar 2.33: Grid dengan Probabilitas
(Sumber: Z. Kelta, 2022)

- bx dan by adalah koordinat x dan y dari pusat kotak pembatas (*center of bounding box*) sehubungan dengan grid sel pembungkus.
- bh dan bw sesuai dengan tinggi dan lebar kotak pembatas sehubungan dengan sel grid pembungkus
- $c1$ dan $c2$ sesuai dengan dua kelas Player dan Ball, dapat memiliki kelas sebanyak yang dibutuhkan oleh pengguna.

Untuk dapat memahami dan terlihat, seperti pada Gambar 2.34 berikut.



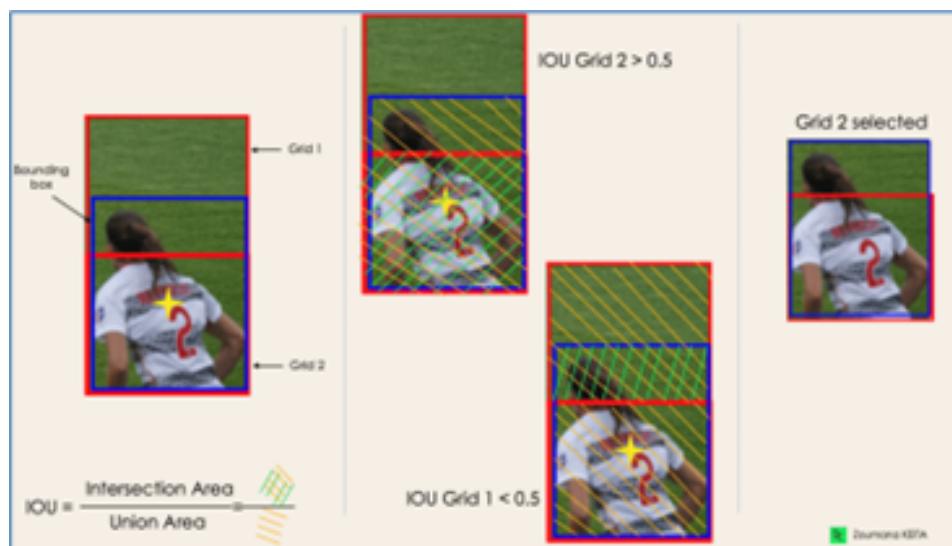
Gambar 2.34: Cara Bounding Box
(Sumber: Z. Kelta, 2022)

(c) *Intersection Over Unions* (IOU)

Sebagian besar waktu, satu objek dalam gambar dapat memiliki beberapa kandidat kotak petak untuk prediksi, meskipun tidak semuanya relevan. Tujuan dari IOU (nilai antara 0 dan 1) adalah untuk membuang kotak kisi tersebut agar hanya menyimpan yang relevan. Inilah logika dari IOU:

- Pengguna menentukan ambang pemilihan IOU-nya, misalnya, 0,5.
- Kemudian YOLO menghitung IOU dari setiap sel grid yang merupakan area persimpangan dibagi dengan Union Area.
- Terakhir, ia mengabaikan prediksi sel kisi yang memiliki $\text{IOU} \leq$ ambang batas dan mempertimbangkannya dengan $\text{IOU} > \text{ambang}$ batas.

Pada Gambar 2.35 adalah ilustrasi penerapan proses pemilihan grid pada objek kiri bawah. Dapat diamati bahwa objek awalnya memiliki dua kandidat kisi, kemudian hanya "Kisi 2" yang dipilih di bagian akhir.



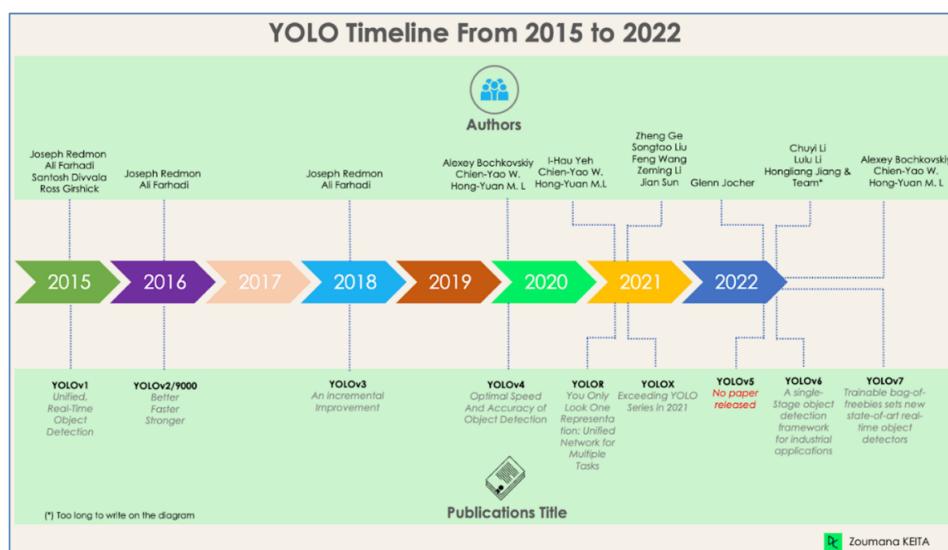
Gambar 2.35: IOU
(Sumber: Z. Kelta, 2022)

(d) Non-Maximum Supression (NMS)

Menetapkan ambang batas untuk IOU tidak selalu cukup karena sebuah objek dapat memiliki beberapa kotak dengan IOU di luar ambang batas, dan meninggalkan semua kotak tersebut mungkin termasuk kebisingan (*noise*). Di sinilah, dapat menggunakan NMS untuk menyimpan hanya kotak dengan skor probabilitas deteksi tertinggi.

2.7.3 Perkembangan YOLO

Sejak rilis pertama YOLO pada tahun 2015, YOLO telah banyak berkembang dengan versi berbeda, seperti pada Gambar 2.36.



Gambar 2.36: Perkembangan YOLO
(Sumber: Z. Kelta, 2022)

1. YOLO atau YOLOv1

Versi pertama YOLO ini adalah pengubah permainan untuk deteksi objek, karena kemampuannya mengenali objek dengan cepat dan efisien. Namun, seperti banyak solusi lainnya, versi pertama YOLO memiliki keterbatasannya sendiri:

- Kesulitan untuk mendeteksi gambar yang lebih kecil dalam sekelompok gambar, seperti sekelompok orang di stadion. Ini karena setiap kisi dalam arsitektur YOLO dirancang untuk deteksi objek tunggal.
- Kemudian, YOLO tidak berhasil mendeteksi bentuk baru atau tidak biasa.
- Terakhir, fungsi kerugian yang digunakan untuk memperkirakan kinerja pendekripsi memperlakukan kesalahan yang sama untuk kotak pembatas kecil dan besar, yang sebenarnya membuat pelokalan yang salah.

2. YOLOv2 atau YOLO9000

YOLOv2 dibuat pada tahun 2016 dengan ide membuat model YOLO lebih baik, lebih cepat, dan lebih kuat.

Peningkatan termasuk tetapi tidak terbatas pada penggunaan Darknet-19 sebagai arsitektur baru, normalisasi batch, resolusi input yang lebih tinggi, lapisan konvolusi dengan anchors, pengelompokan dimensi, dan (5) fitur-fitur halus.

- *Batch Normalization*

Menambahkan lapisan normalisasi batch meningkatkan kinerja sebesar 2% mAP. Normalisasi batch ini menyertakan efek regularisasi, mencegah overfitting.

- *Higher input resolution*

YOLOv2 secara langsung menggunakan input 448x448 beresolusi lebih tinggi daripada 224x224, yang membuat model menyesuaikan filternya untuk bekerja lebih baik pada gambar beresolusi lebih tinggi. Pendekatan ini meningkatkan akurasi sebesar 4% mAP, setelah dilatih selama 10 epochs pada data ImageNet.

3. YOLOv3 - Peningkatan Bertahap

Perubahan tersebut terutama mencakup arsitektur jaringan baru: Darknet-53. Ini adalah jaringan saraf 106, dengan jaringan upsampling dan blok residual. Jauh lebih besar, lebih cepat, dan lebih akurat dibandingkan dengan Darknet-19, yang merupakan tulang punggung YOLOv2. Arsitektur baru ini telah bermanfaat di banyak tingkatan:

- Prediksi *Bounding Box* Lebih Baik

Model regresi logistic digunakan oleh YOLOv3 untuk memprediksi skor objektivitas untuk setiap kotak pembatas (bounding box).

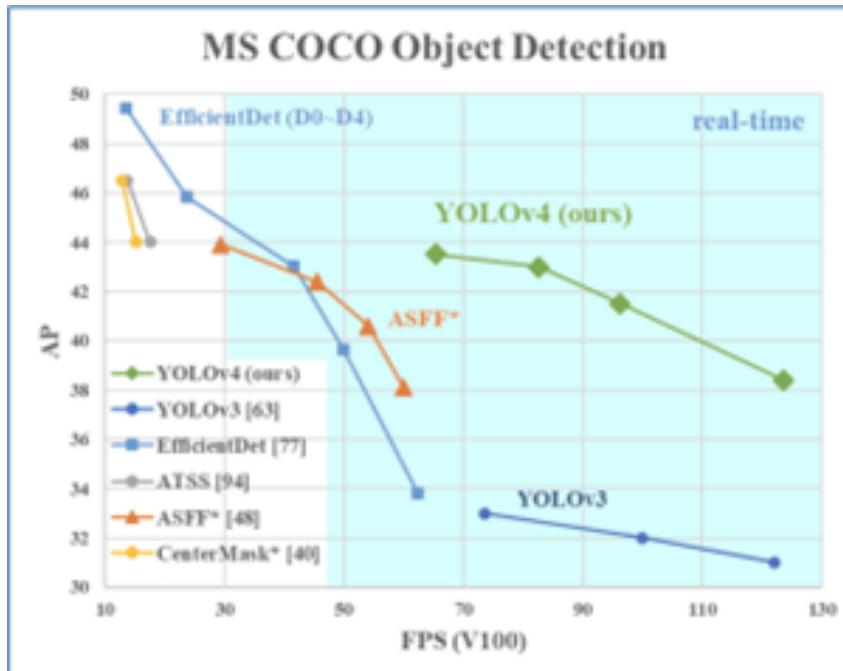
- Prediksi Kelas yang Lebih Akurat

Menggantikan penggunaan softmax seperti yang dilakukan di YOLOv2, pengklasifikasi logistik independen telah diperkenalkan untuk memprediksi kelas kotak pembatas secara akurat. Ini bahkan berguna saat menghadapi domain yang lebih kompleks dengan label yang tumpang tindih (Misalnya, → Pemain Sepak Bola). Menggunakan softmax akan membatasi setiap kotak hanya memiliki satu kelas, yang tidak selalu benar.

4. YOLOv4 - *Optimal Speed and Accuracy of Object Detection*

Versi YOLO ini memiliki Kecepatan dan Akurasi Deteksi Objek Optimal dibandingkan dengan semua versi sebelumnya dan detektor objek canggih lainnya.

Gambar 2.37 menunjukkan YOLOv4 mengungguli YOLOv3 dan FPS dalam kecepatan masing-masing sebesar 10% dan 12%.



Gambar 2.37: Komparasi YOLOv4 dengan YOLOv3 dan *state-of-the-art*

Deteksi Objek Lain

(Sumber: Z. Kelta, 2022)

YOLOv4 dirancang khusus untuk sistem produksi dan dioptimalkan untuk komputasi paralel.

Backbone arsitektur YOLOv4 adalah CSPDarknet53, jaringan yang berisi 29 lapisan konvolusi dengan filter 3 x 3 dan sekitar 27,6 juta parameter. Arsitektur ini, dibandingkan dengan YOLOv3, menambahkan informasi berikut untuk deteksi objek yang lebih baik:

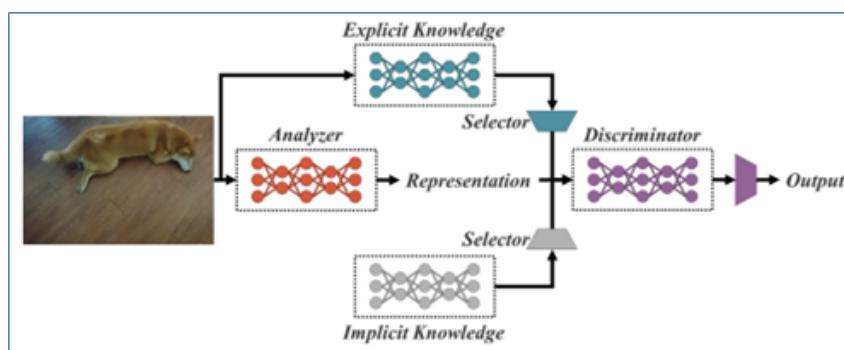
- *Spatial Pyramid Pooling* (SPP) secara signifikan meningkatkan bidang reseptif, memisahkan fitur konteks yang paling relevan, dan tidak memengaruhi kecepatan jaringan.
- Menggantikan *Feature Pyramid Network* (FPN) yang digunakan di YOLOv3, YOLOv4 menggunakan PANet untuk agregasi parameter dari tingkat deteksi yang berbeda.
- Augmentasi data menggunakan teknik mosaik yang menggabungkan

empat gambar pelatihan selain pendekatan pelatihan permusuhan diri.

- Menggunakan pemilihan hyper-parameter yang optimal menggunakan algoritma genetika.

5. YOLOR - You Only Look One Representation

Sebagai *Unified Network for Multiple Tasks*, YOLOR didasarkan pada jaringan terpadu yang merupakan kombinasi dari pendekatan pengetahuan eksplisit dan implisit.



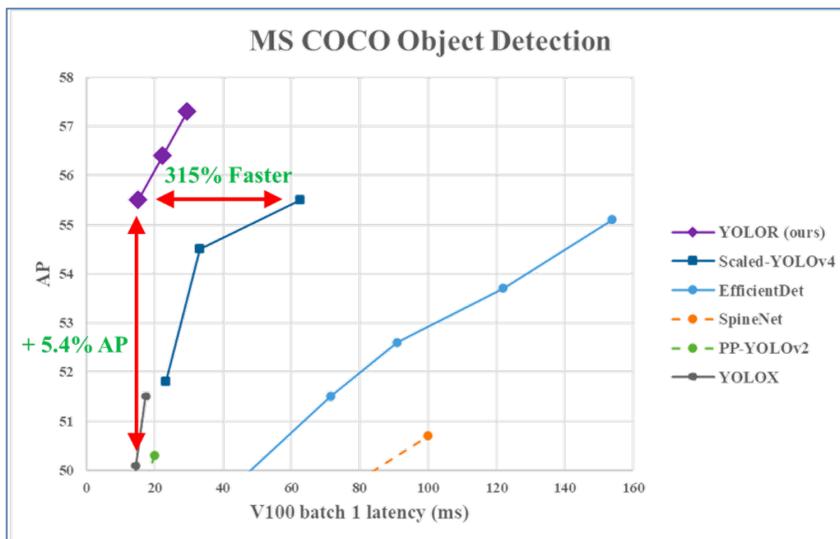
Gambar 2.38: Unified Network Architecture
(Sumber: C. Y. Wang et al, 2021)

Pengetahuan eksplisit adalah pembelajaran normal atau conscious learning. Pembelajaran implisit di sisi lain dilakukan secara tidak sadar (dari pengalaman).

Menggabungkan kedua teknik ini, YOLOR mampu menciptakan arsitektur yang lebih kuat berdasarkan tiga proses: (1) penyelarasan fitur, (2) penyelarasan prediksi untuk deteksi objek, dan (3) representasi kanonis untuk pembelajaran multi-tugas.

Salah satu yang mengalami peningkatan adalah penajaran prediksi. Pendekatan ini memperkenalkan representasi implisit ke dalam peta fitur dari setiap jaringan piramida fitur (FPN), yang meningkatkan presisi sekitar 0,5%.

Dari grafik berikut, dapat diamati bahwa YOLOR mencapai kecepatan inferensi data MS COCO yang canggih dibandingkan dengan model lain seperti pada Gambar 2.39.



Gambar 2.39: Performance YOLOR vs YOLOv4 dan Model Lainnya

(Sumber: C. Y. Wang et al, 2021)

6. YOLOX - Exciding *YOLO Series in 2021*

Ini menggunakan baseline yang merupakan versi modifikasi dari YOLOv3, dengan Darknet-53 sebagai *backbone*.

Diterbitkan dalam makalah *Exceeding YOLO Series in 2021*, YOLOX menghadirkan empat karakteristik utama berikut untuk membuat model yang lebih baik dibandingkan dengan versi yang lebih lama.

- Kepala terpisah yang efisien: Coupled head yang digunakan pada versi YOLO sebelumnya terbukti mengurangi performa model. YOLOX menggunakan decoupled sebagai gantinya, yang memungkinkan pemisahan tugas klasifikasi dan lokalisasi, sehingga meningkatkan kinerja model.
- Augmentasi data yang kuat: Integrasi Mosaic dan MixUp ke dalam pendekatan augmentasi data sangat meningkatkan kinerja YOLOX.

- *Anchor free system*: Algoritma *anchor-based* melakukan pengelompokan di bawah tenda, yang meningkatkan waktu inferensi. Menghapus mekanisme jangkar di YOLOX mengurangi jumlah prediksi per gambar, dan meningkatkan waktu inferensi secara signifikan.
- SimOTA untuk penetapan label: Pergantian penggunaan pendekatan interseksi penyatuan (IoU), penulis memperkenalkan SimOTA, strategi penetapan label yang lebih kuat yang mencapai hasil canggih dengan tidak hanya mengurangi waktu pelatihan tetapi juga menghindari masalah hiperparameter tambahan. Selain itu, ini meningkatkan peta deteksi sebesar 3%.

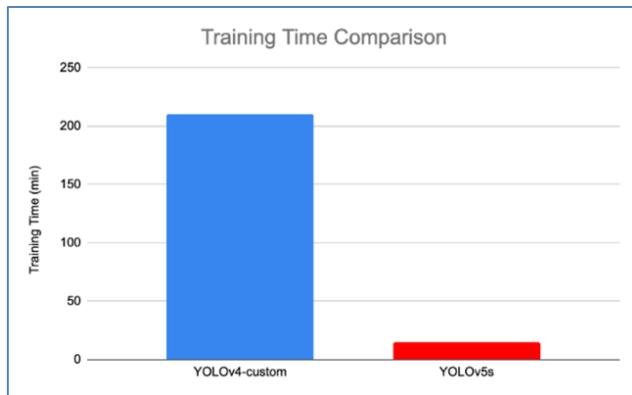
7. YOLOv5

YOLOv5, dibandingkan dengan versi lain, tidak memiliki makalah penelitian yang diterbitkan, dan ini adalah versi YOLO pertama yang diimplementasikan di Pytorch, bukan di Darknet.

Dirilis oleh Glenn Jocher pada Juni 2020, YOLOv5, mirip dengan YOLOv4, menggunakan CSPDarknet53 sebagai tulang punggung arsitekturnya. Rilis ini mencakup lima ukuran model yang berbeda: YOLOv5s (terkecil), YOLOv5m, YOLOv5l, dan YOLOv5x (terbesar).

Salah satu peningkatan besar dalam arsitektur YOLOv5 adalah integrasi lapisan Fokus, yang diwakili oleh satu lapisan, yang dibuat dengan mengganti tiga lapisan pertama YOLOv3. Integrasi ini mengurangi jumlah lapisan, dan jumlah parameter dan juga meningkatkan kecepatan maju dan mundur tanpa dampak besar pada peta.

Ilustrasi Gambar 2.40 membandingkan waktu pelatihan antara YOLOv4 dan YOLOv5.



Gambar 2.40: Perbandingan Waktu Pelatihan antara YOLOv4 dan YOLOv5
(Sumber: J. Nelson, 2020)

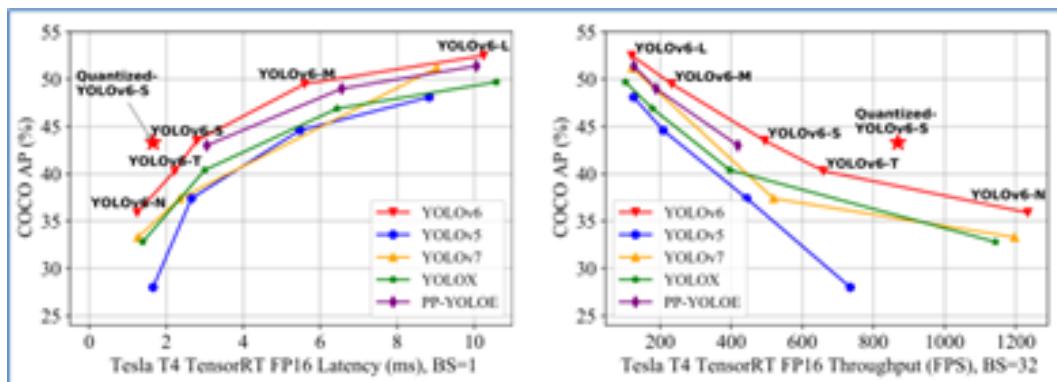
8. YOLOv6

Didedikasikan untuk aplikasi industri dengan desain efisien yang ramah perangkat keras dan kinerja tinggi, kerangka kerja YOLOv6 (MT-YOLOv6) dirilis oleh Meituan, sebuah perusahaan e-commerce Tiongkok.

Ditulis dalam Pytorch, versi baru ini bukan bagian dari YOLO resmi tetapi tetap diberi nama YOLOv6 karena tulang punggungnya terinspirasi oleh arsitektur YOLO satu tahap yang asli.

YOLOv6 memperkenalkan tiga peningkatan signifikan pada YOLOv5 sebelumnya: desain tulang punggung dan leher yang ramah perangkat keras, kepala terpisah yang efisien, dan strategi pelatihan yang lebih efektif.

YOLOv6 memberikan hasil yang luar biasa dibandingkan dengan versi YOLO sebelumnya dalam hal akurasi dan kecepatan pada dataset CO-CO seperti yang diilustrasikan pada Gambar 2.41



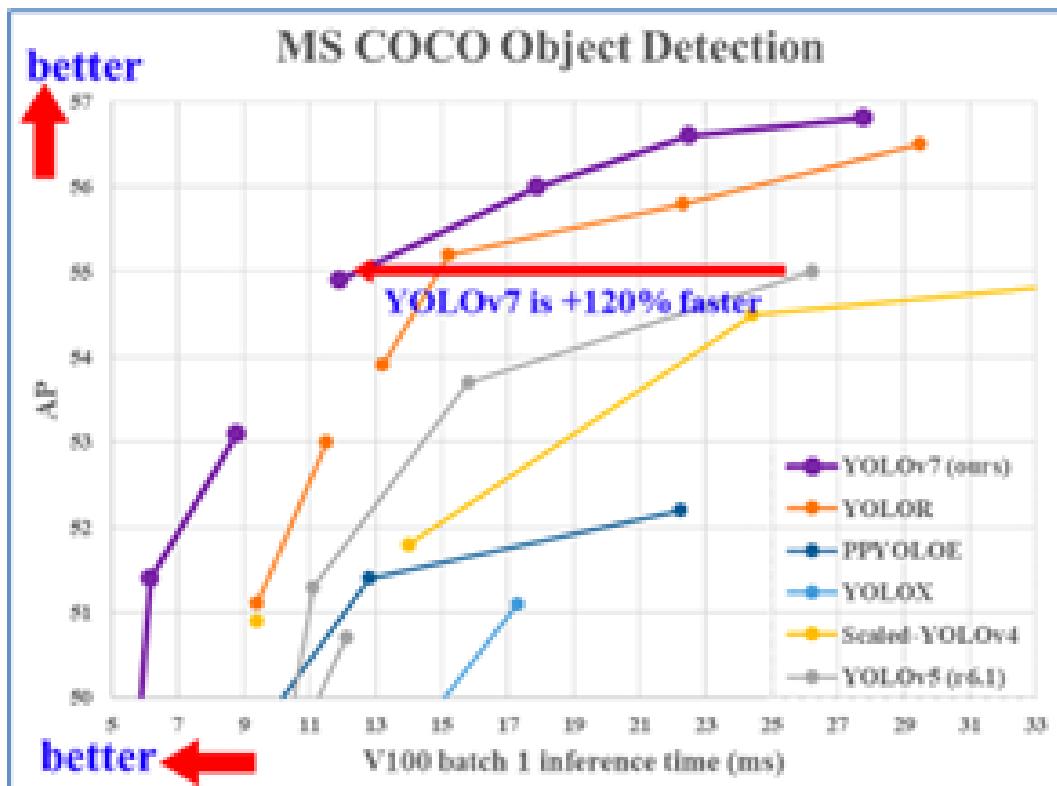
Gambar 2.41: Comparison of state-of-the-art efficient object detectors. All models are tested with TensorRT 7 except that the quantized model is with TensorRT 8

(Sumber: C. Li et al, 2022)

- YOLOv6-N mencapai 35,9% AP pada dataset COCO dengan throughput 1234 (throughput) FPS pada GPU NVIDIA Tesla T4.
- YOLOv6-S mencapai AP 43,3% yang canggih pada 869 FPS.
- YOLOv6-M dan YOLOv6-L juga mencapai kinerja akurasi yang lebih baik masing-masing sebesar 49,5% dan 52,3% dengan kecepatan inferensi yang sama.

9. YOLOv7

YOLOv7 dirilis pada Juli 2022 di Paper *Trained bag-of-freebies set state-of-the-art real-time object detector*. Versi ini membuat langkah signifikan di bidang deteksi objek, dan melampaui semua model sebelumnya dalam hal akurasi dan kecepatan.



Gambar 2.42: Perbandingan YOLOv7 inference time dengan real-time object detector lainnya

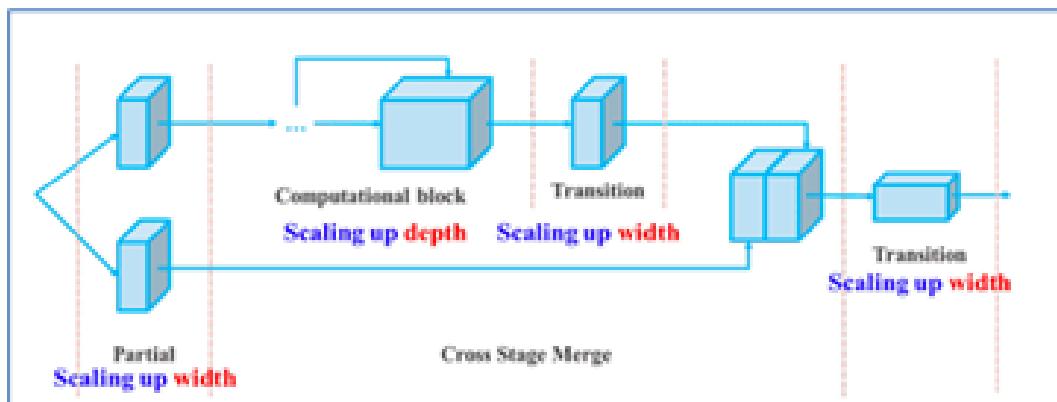
(Sumber: C. Y. Wang et al, 2022)

YOLOv7 telah membuat perubahan besar dalam (1) arsitekturnya dan (2) pada tingkat bag-of-freebies yang dapat dilatih:

- Level Arsitektur

YOLOv7 mereformasi arsitekturnya dengan mengintegrasikan Extended Efficient Layer Aggregation Network (E-ELAN) yang memungkinkan model mempelajari fitur yang lebih beragam untuk pembelajaran yang lebih baik.

Selain itu, YOLOv7 menskalakan arsitekturnya dengan menggabungkan arsitektur model asalnya seperti YOLOv4, Scaled YOLOv4, dan YOLO-R. Hal ini memungkinkan model untuk memenuhi kebutuhan kecepatan inferensi yang berbeda.



Gambar 2.43: *Compound scaling up depth and width for concatenation-based model*

(Sumber: C. Y. Wang et al, 2022)

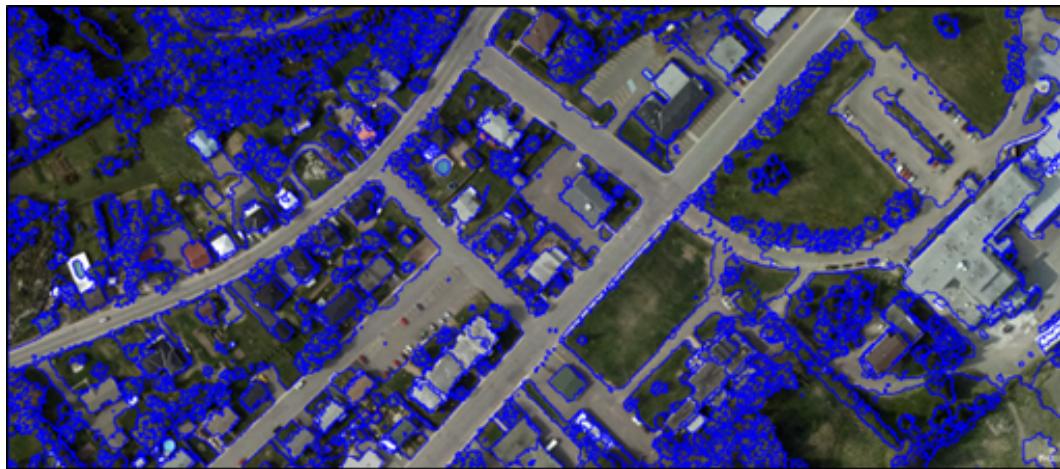
- Trainable bag-of-freebies

Istilah bag-of-freebies mengacu pada peningkatan akurasi model tanpa meningkatkan biaya pelatihan, dan inilah alasan mengapa YOLOv7 tidak hanya meningkatkan kecepatan inferensi tetapi juga akurasi deteksi.

2.8 Object Based Image Analysis (OBIA)

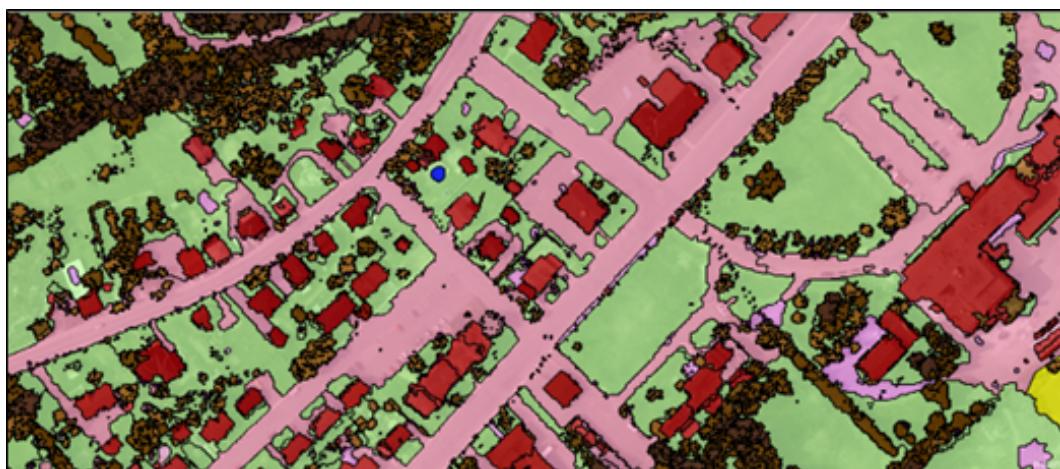
OBIA merupakan teknik klasifikasi yang tidak hanya memandang rona dan tekstur piksel namun berdasarkan dari kesatuan objek, atau dapat dikatakan OBIA adalah pendekatan yang proses klasifikasinya tidak hanya mempertimbangkan aspek spectral namun aspek spasial objek. Data citra penginderaan jauh yang digunakan untuk klasifikasi ini biasanya menggunakan data citra penginderaan jauh resolusi tinggi seperti Quickbird, Ikonos, World View, dll. Klasifikasi ini hamper mirip dengan klasifikasi unsupervised, akan tetapi basis dari klasifikasi OBIA yaitu dengan segmentasi (A. Bakar, 2014).

OBIA mengelompokkan gambar yang mengelompokkan piksel kecil menjadi objek vektor. Alih-alih berbasis per-pixel, segmentasi secara otomatis mendigitalkan gambar untuk pengguna (GIS Geography, 2020).



Gambar 2.44: OBIA Segmentasi Proses Pengelompokkan Pixel yang hampir sama ke dalam Objek
(Sumber: GISGeography, 2020)

Apa yang dilakukan segmentasi adalah meniru apa yang dilakukan mata pengguna. Tetapi dengan objek tersegmentasi ini, pengguna menggunakan properti spektral, geometris, dan spasialnya untuk mengklasifikasikan ke dalam tutupan lahan.



Gambar 2.45: OBIA Klasifikasi menggunakan shape, size, dan spectral properties objek untuk klasifikasi setiap objek

Dua prinsip dasar OBIA adalah klasifikasi dan segmentasi. Segmentasi adalah memecah gambar menjadi objek yang mewakili fitur berbasis darat, sedangkan Klasifikasi adalah mengklasifikasi objek-objek tersebut menggu-

nakan bentuk, ukuran, sifat spasial dan spektralnya. Analis sering menggunakan statistik ini untuk mengklasifikasikan tutupan lahan menggunakan OBIA, yaitu *trees* (pohon) memiliki ketinggian yang bervariasi (standar deviasi *normalized Digital Surface Model* (nDSM)) yang merupakan model elevasi yang menangkap fitur alami dan fitur buatan, seperti gedung, pohon, kabel listrik, dan objek lainnya. Di samping itu, memiliki reflektansi inframerah-dekat yang tinggi (*normalized difference vegetation index* (NDVI) tinggi), sedangkan *grass* memiliki pendek (nDSM rendah), datar (deviasi standar nDSM rendah) dan memiliki reflektansi inframerah-dekat sedang (NDVI sedang). NDVI digunakan untuk mengukur indeks yang menggambarkan tingkat kepadatan, kehijauan suatu tanaman dan kondisi dari suatu vegetasi.

2.9 Sentinel-2

Data yang digunakan dalam penelitian ini diambil dari gambar citra satelit Sentinel-2. Sentinel-2 diluncurkan sebagai bagian dari program Copernicus Komisi Eropa pada tanggal 23 Juni 2015 yang dirancang khusus untuk memberikan banyak data dan citra. Satelit dilengkapi dengan sensor multispectral opto-elektronik untuk survei dengan resolusi Sentinel-2 yaitu 10 hingga 60m di zona spektral tampak, VNIR, SWIR, termasuk 13 saluran spectral yang memastikan penangkapan perbedaan dalam keadaan vegetasi, termasuk perubahan temporal, dna juga meminimalkan dampkan kualitas fotografi atmosfer. Sentinel-2 memiliki dua satelit dalam misi memungkinkan survei berulang setiap 5 hari di ekuator dan 2-3 hari di garis lintang tengah (EOS, 2020).

Misi Sentinel-2 terdiri dari dua satelit yang dikembangkan untuk mendukung vegetasi, tutupan lahan, dan pemantauan lingkungan. Satelit Sentinel-2A diluncurkan oleh European Space Agency (ESA) pada 23 Juni 2015, dan beroperasi di orbit sinkron matahari dengan siklus berulang 10 hari. Setelit

kedua yaitu Sentinel-2B diluncurkan pada 7 Maret 2017 dan beroperasi dengan akuisisi data yang tersedia di EarthExplorer (USGS EROS, 2020).

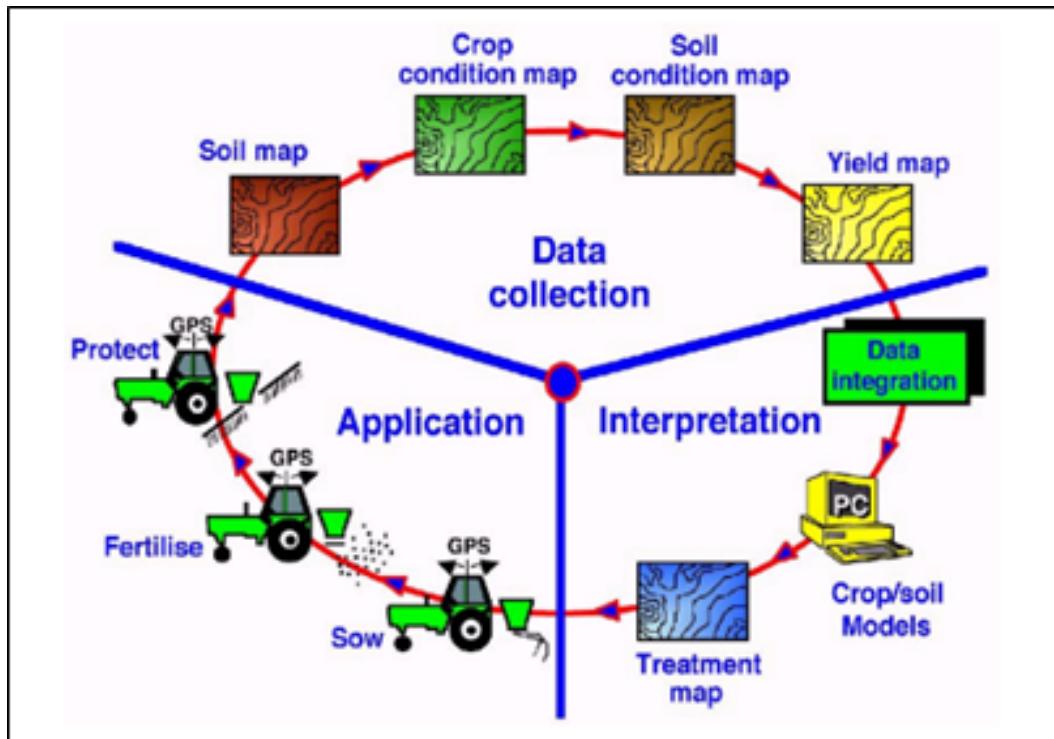
Terdapat Perbandingan antara Sentinel-2 dengan UAV Drone, seperti pada Tabel 2.2.

2.10 Pertanian Presisi

Pertanian presisi atau precision agriculture merupakan suatu sistem pertanian yang mengintegrasikan penggunaan teknologi dalam mengumpulkan informasi, sehingga dapat melakukan proses pertanian secara presisi atau dengan input, tempat, dan waktu yang tepat. Istilah presisi berarti akurat, terakar, dan terukur. Pertanian presisi disebut juga menggunakan input pertanian yang tepat dengan teknik, jumlah, tempat, dan waktu yang tepat untuk menghasilkan produksi panen secara maksimal. Input pertanian meliputi seperti pemupukan, herbisida, insektisida, benih, dan lainnya. Meskipun untuk melakukan pertanian secara akurat ini membutuhkan banyak informasi, cenderung kompleks untuk kebanyakan petani, dan membutuhkan kerjasama dari berbagai multidisiplin ilmu, namun sistem pertanian presisi mampu meningkatkan laba, mengurangi limbah, mengurangi biaya produksi, dan menjaga kualitas lingkungan (J. Taylor et al., 2016).

Pertanian presisi (PA) merupakan ilmu untuk meningkatkan hasil panen dan membantu keputusan manajemen menggunakan sensor dan alat analisis teknologi tinggi. Pertanian presisi adalah konsep yang digunakan untuk meningkatkan produksi, mengurangi waktu kerja, dan memastikan pengelolaan seperti pupuk, irigasi, dan lainnya berjalan efektif. Perkembangan teknologi, khususnya ketersediaan citra satelit beresolusi tinggi, perkembangan teknologi kendaraan udara tak berawak (UAV), menunjukkan bahwa adopsi sumber data penginderaan jauh dalam pertanian presisi mengalami peningkatan (P. Singh et al., 2020).

Siklus pertanian presisi terdiri dari tahapan seperti Gambar 2.46.



Gambar 2.46: Siklus Pertanian Presisi
(Sumber: A. Comparetti, 2011)

- Pengukuran parameter tanah dan tanaman dengan bantuan pemetaan spasial dan pemantauan kondisi cuaca setempat (pengumpulan data);
- pemetaan parameter tanah dan tanaman dalam lapangan (pengumpulan data);
- pemetaan dan integrasi data dengan bantuan aplikasi (interpretasi);
- pemantauan kesuburan dan penaburan (aplikasi);

Dalam menerapkan pertanian presisi diperlukan instrumen berikut (A. Comparetti, 2011):

- Sistem penentuan posisi satelit, untuk penginderaan posisi di mana setiap parameter lapangan yang diukur harus direferensikan secara geografis dan, kemudian, juga posisi di mana input (seperti, tanaman, lahan) yang

terdeteksi oleh mesin dapat diterapkan kebutuhannya di setiap area lahan (zona pengelolaan).

- Sensor, untuk mengukur parameter tanah dan tanaman di lapangan.
- Perangkat (devices), untuk menyiapkan dan mengontrol aplikasi input tanaman tingkat variabel spasial.
- Perangkat lunak, untuk membuat peta parameter tanah dan tanaman dalam lahan dan aplikasi input tanaman dengan data dari citra pengim-deraan, dan juga untuk menginterpretasikan data terukur;

2.11 Kelapa Sawit

Kelapa sawit adalah tanaman sejenis palma berakar serabut atau monokotil.

Bagian tanaman yang bernilai ekonomis adalah buah. Masing-masing dari seribu atau lebih buah dalam tandan buah segar itu terdiri dari inti sawit (kernel) yang dikelilingi oleh daging buah (mesocarp). Di pabrik kelapa sawit, sangat sedikit buah yang terbuang sia-sia karena pabrik tersebut mengubah setiap buah kelapa sawit menjadi minyak kelapa sawit dan minyak inti sawit), bahkan limbah dari setiap buah sawitpun di daur ulang di perkebunan sebagai pupuk atau diolah sebagai bahan bakar biomass. Sangat sedikit sekali buah kelapa sawit terbuang karena baik inti kelapa sawit dan dagingnya sama-sama digunakan untuk menghasilkan minyak.



Gambar 2.47: Buah Kelapa Sawit(Inti dan Daging Sawit)

Satu tandan tanaman dewasa beratnya mencapai 20 - 35 kg, bahkan ada yang mencapai di atas 40 kg, tergantung pada perawatan. Tandan tersusun dari 200 - 600 buah masing-masing sekitar 20-35 gram. Buah diambil minyaknya dengan hasil sabut (daging buah/mesocarp) menghasilkan minyak kasar (CPO) 20-26%, inti sawit sebanyak 6% yang menghasilkan minyak inti (PKO) 3-4%. Kadar % dihitung dari berat tandan buah segar (Smart Agribusiness and Food, 2017).

(a) Usia Tanam

Umur atau usia ekonomis tanaman kelapa sawit yang dibudidayakan bisa mencapai usia hingga 25 tahun. Pada usia tanam sudah tinggi, sehingga sulit dipanen, tandanya sudah jarang sehingga secara perhitungan tidak ekonomis lagi. Pengelompokkan berdasarkan masa berbuah (PTPN, 2018), seperti Tabel 2.2 berikut ini:

Tabel 2.2: Pengelompokkan Berdasarkan Masa Berbuah

Kelompok	Masa Berbuah (Tahun)
Tanaman Belum Menghasilkan (TBM)	0-3
Tanaman Menghasilkan	>3

(b) Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman

Produktivitas tanaman kelapa sawit (tandan buah segar (TBS)) menurut umur tanaman dalam Kondisi Kebun Percobaan Balit Marihat berdasarkan PTPN. VII tahun 1993 dalam *lecture note* "Budi Daya Kelapa Sawit" (S. Yahya & Suwarto, 2021), seperti Tabel 2.4. berikut ini:

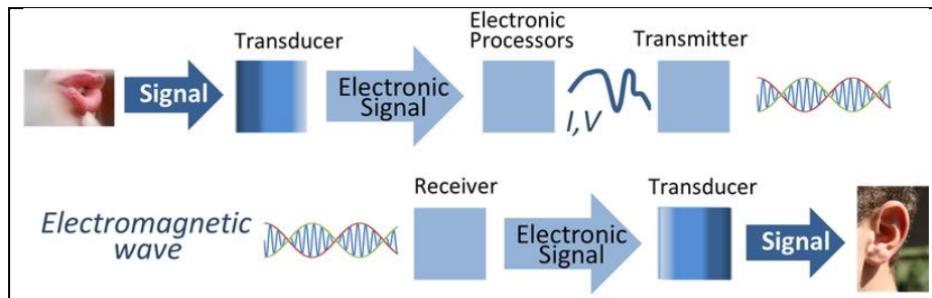
Tabel 2.3: Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman dalam Kondisi Kebun Percobaan Balit Marihat

Umur Di Lapangan (Tahun)	Produksi Produksi TBS (ton/ha/Thn)
4	8

2.12 Teknologi Informasi

Informasi adalah proses transmisi dan transfer pengetahuan: bentuk, data, dan konsep, studi dengan tujuan membuatnya dapat diakses oleh orang lain, lembaga atau masyarakat. Kualitas proses ini akan menentukan diterima atau tidaknya perubahan perilaku dan sikap individu tersebut. Teknologi informasi berupa perangkat apapun yang memiliki kapasitas untuk mengolah data dan atau informasi, baik secara sistematis maupun dinamis. Teknologi informasi juga dapat diterapkan pada produk maupun dalam suatu proses (Victoria 2020).

Telekomunikasi adalah sistem komunikasi jarak jauh dengan teknologi, terutama melalui sinyal listrik atau gelombang elektromagnetik. sistem komunikasi juga merupakan bagian dari perangkat keras (seperti TV, ponsel) atau algoritma yang membaca informasi data masukan, memprosesnya dan mengirimkan data keluaran melalui saluran tertentu. gambar 2.48 merupakan gambaran umum dasar telekomunikasi.



Gambar 2.48: dasar telekomunikasi (El-Saba 2018)

Digital Television (DTV) merupakan transmisi sinyal audiovisual televisi menggunakan pengkodean digital, berbeda dengan teknologi televisi analog sebelumnya yang menggunakan sinyal analog. Perkembangan ini dianggap sebagai kemajuan inovatif dan merupakan evolusi signifikan pertama dalam teknologi televisi sejak televisi berwarna pada 1950-an. DTV dapat berupa *High Definition Television* (HDTV) atau transmisi simultan dari berbagai program *Standar Definition Television* (SDTV), yang merupakan kualitas gambar yang lebih rendah daripada HDTV tetapi jauh lebih baik daripada televisi analog (Kruger & Guerrero 2002).

2.13 Teknik Televisi Digital

Layanan DTV mempunyai tiga komponen utama yang harus ada agar konsumen dapat menikmati pengalaman menonton televisi "definisi tinggi" yang sepenuhnya terwujud.

1. pemrograman digital harus tersedia. Pemrograman digital adalah konten yang diproduksi dengan kamera digital dan peralatan produksi digital lainnya. Peralatan tersebut berbeda dari apa yang saat ini digunakan untuk menghasilkan pemrograman analog konvensional.
2. pemrograman digital harus dikirimkan kepada konsumen melalui sinyal digital. Sinyal digital dapat disiarkan melalui gelombang udara (membu-

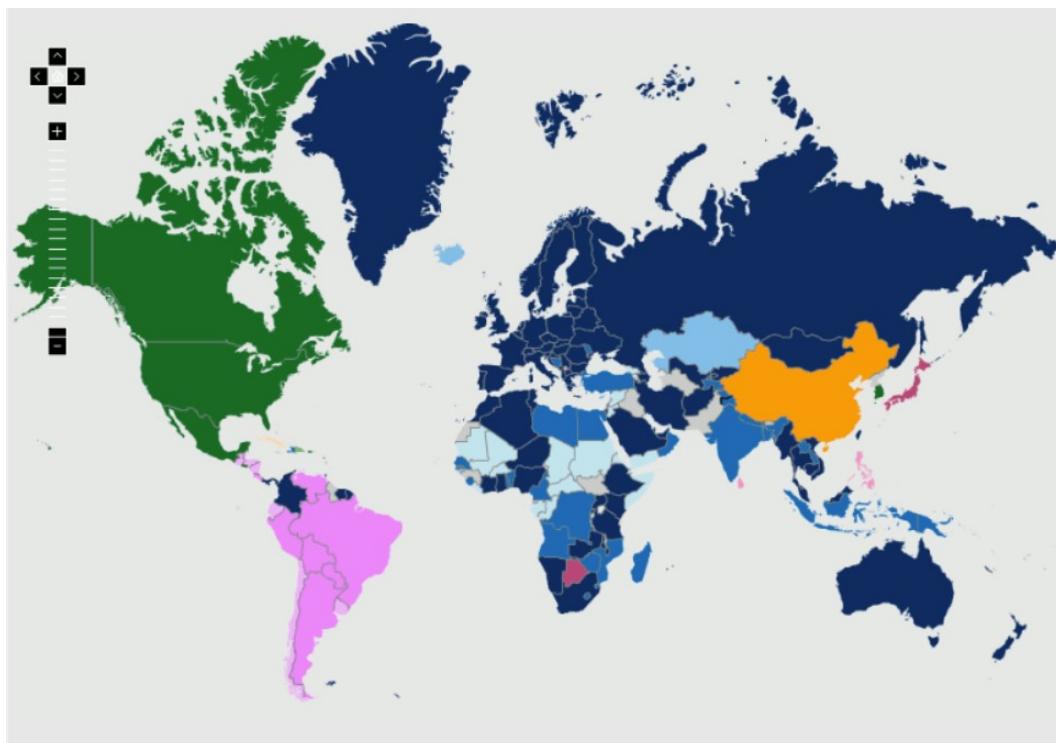
tuhkan menara transmisi baru atau antena DTV di menara yang ada). Sinyal digital ditransmisikan oleh teknologi televisi kabel atau satelit, atau disampaikan oleh sumber yang direkam sebelumnya seperti *disk video digital* (DVD)

3. konsumen harus memiliki produk televisi digital yang mampu menerima sinyal siaran digital, konsumen dapat membeli monitor digital disertai dengan *set-top converter box*. televisi digital juga dapat terintegrasi dengan kemampuan pengaturan digital yang sudah dibuat oleh produsen (Kruger & Guerrero 2002)

Standar penyiaran televisi digital yang berbeda telah diadopsi di berbagai belahan dunia diantaranya *Digital Video Broadcasting* (DVB) menggunakan modulasi *Orthogonal frequency-division multiplexing* (OFDM) dan mendukung transmisi hierarkis. Standar ini telah diadopsi di Eropa, Afrika, Asia, dan Australia, dengan total sekitar 60 negara. *Advanced Television Systems Committee* (ATSC) menggunakan modulasi *8-level vestigial sideband modulation* (8VSB) untuk penyiaran terestrial. Standar ini telah diadopsi oleh 6 negara: Amerika Serikat, Kanada, Meksiko, Korea Selatan, Republik Dominika dan Honduras (DTVstatus 2017). *Integrated Services Digital Broadcasting* (ISDB) adalah sistem yang dirancang untuk menyediakan penerimaan yang baik untuk penerima tetap dan juga penerima portabel atau seluler. ISDB mendukung transmisi hierarkis hingga tiga lapisan dan menggunakan video MPEG-2 dan *Advanced Audio Coding*. Standar ini telah diadopsi di Jepang dan Filipina. ISDB-T International adalah adaptasi dari standar ini menggunakan H.264 / MPEG-4 AVC, yang telah diadopsi di sebagian besar negara-negara Amerika Selatan dan Afrika berbahasa Portugis (yen Ong et al. 2010).

Digital Terrestrial Multimedia Broadcasting (DTMB) mengadopsi teknologi OFDM *Time-Domain Synchronous* (TDS) dengan kerangka sinyal pseudo-acak untuk berfungsi sebagai *Gate Interval* (GI) dari blok OFDM dan sim-

bol pelatihan. Standar DTMB telah diadopsi di Republik Rakyat Tiongkok, termasuk Hong Kong dan Makau (yen Ong et al. 2010). *Digital Multimedia Broadcasting* (DMB) adalah teknologi transmisi radio digital yang dikembangkan di Korea Selatan sebagai bagian dari proyek TI nasional untuk mengirim multimedia seperti TV, radio dan datacasting ke perangkat seluler seperti ponsel, laptop dan sistem navigasi GPS (?). Gambar 2.49 dan tabel 2.4 merupakan penyebaran sistem DVB pada seluruh dunia (DTVstatus 2017).



Gambar 2.49: Sistem penyiaran digital untuk televisi terestrial (DTVstatus 2017)

Tabel 2.4: Status DTV membandingkan sistem penyiaran digital di seluruh dunia ATSC, DTMB, DVB-T/DVB-T2, dan ISDB-T (DTVstatus 2017)

System	Explanation
DVB-T/DVB-T2	Broadcasting via DVB-T/DVB-T2 is actively in use.
DVB-T/DVB-T2 adopted	Countries which have adopted the DVB-T/DVB-T2 system.
DVB-T/DVB-T2 trial broadcasts	Those countries undertake trials with DVB-T/DVB-T2.
RRC06	The according countries participate in the Regional Radiocommunications Conference 2006 of the ITU (International Telecommunication Union). It can be assumed that all countries taking part will ultimately use the DVB-T/DVB-T2 system when they move from analog to digital.
ATSC	Broadcasting via the ATSC system is actively in use.
ATSC adopted	Countries which have adopted the ATSC system.
ATSC trial broadcasts	Those countries undertake trials with ATSC.
ISDB-T	Broadcasting via ISDB-T is actively in use.
ISDB-T adopted	Countries which have adopted the ISDB-T system.
ISDB-T trial broadcasts	Those countries undertake trials with ISDB-T.
SBTVD-T	Broadcasting via SBTVD-T is actively in use.
SBTVD-T adopted	Countries which have adopted the SBTVD-T system.
DTMB	Broadcasting via DTMB is actively in use.
DTMB adopted	Countries which have adopted the DTMB system.
DTMB trial broadcasts	Those countries undertake trials with DTMB.
Commercial DVB-T services	No formal adoption of a DTT standard or undecided countries

2.14 Digital Video Broadcasting Terestrial 2 (DVB-T2)

Proyek *Digital Video Broadcasting* (DVB), merupakan suatu badan yang bertanggung jawab untuk membuat spesifikasi DVB dan secara resmi diresmikan

pada bulan September 1993. Proyek ini didahului oleh Kelompok Peluncuran Eropa untuk Penyiaran Video Digital. Proyek ini terdiri dari kelompok sukarela yang terdiri dari lebih dari 210 organisasi yang telah bergabung untuk memungkinkan pengembangan standar untuk DVB di semua bagian dunia, serta pengenalan awal layanan DVB (Reimers 1998). Sistem DVB mendistribusikan data menggunakan berbagai pendekatan yaitu:

1. Transmisi Satellite (S):
 - (a) DVB-S
 - (b) DVB-S2
2. Transmisi Kabel (C):
 - (a) DVB-C
 - (b) DVB-C2
3. Transmisi televisi terestrial (T):
 - (a) DVB-T
 - (b) DVB-T2 untuk televisi terestrial digital

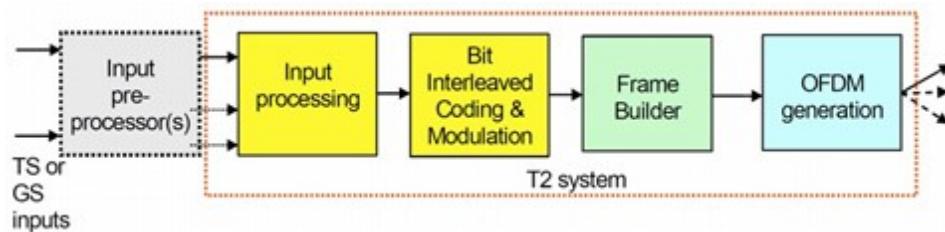
Tabel 2.5: Fitur DVB-S2, DVB-C2 dan DVB-T2 (SANDY 4.)

Category	DVB-T2	DVB-C2	DVB-S2
Input Interface	Multiple Transport Stream and Generic Stream Encapsulation (GSE)	Multiple Transport Stream and Generic Stream Encapsulation (GSE)	Multiple Transport Stream and Generic Stream Encapsulation (GSE)
Code rate	1/2, 3/5, 2/3, 3/4, 4/5, 5/6	1/2, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10	1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10
Interleaving	Bit, Cell, Time, Frequency	Bit, Time, Frequency	Bit
Modulation Scheme	QPSK, 16QAM, 64QAM, 256QAM	16QAM to 4096QAM	QPSK to 8-PSK, 16-APSK, 32-APSK
Modulation	OFDM	OFDM	Singe carrier
Guard Interval	1/4, 19/256, 1/8, 9/128, 1/16, 1/32, 1/128	1/64, 1/128	-
FFT Size	1K, 2K, 4K, 8K, 16K, 32K	4K	-
Pilots	Scaterred, edge, continual, P2, frame-closing	Scaterred, edge, continual, P2, preamble	36 pilot symbols

Tabel 2.5 merupakan fitur yang ada pada DVB-S2, DVB-C2 dan DVB-T2. DVB-T2 dipilih pada penelitian ini karena merupakan salah satu standar teknis terbaru yang dikembangkan oleh Proyek DVB untuk DTT (*Digital Terrestrial Television*). DVB-T2 juga dikenal sebagai siaran digital melalui sistem terestrial sejak tahun 2006 dan merupakan perpanjangan dari sistem DVB-T sebagai program generasi kedua untuk meningkatkan efisiensi sistem total (Ya'acob et al. 2019). DVB-T2 adalah standardisasi generasi kedua dari DVB-T. DVB-T2 menyediakan enam ukuran *fast Fourier transform* (FFT) hingga 32K FFT, tujuh *guard interval* (GI) yang beragam, dan empat skema modulasi berbasis modulasi OFDM hingga 256-*Quadrature Amplitude Modulation* (QAM). DVB-T2 disasarkan pada kode LDPC (Low-Density Parity-

Check) dan BCH (Bose-Chaudhuri-Hocquenghem) yang digabungkan dengan berbagai tingkat kode (Lee & Kook 2019).

Tabel 2.6 menunjukkan parameter transmisi yang dapat digunakan pada DVB-T dan DVB-T2. Perbedaan diantaranya ditandai dengan huruf cetak tebal. Parameter yang membedakan salah satunya *Input Interface* pada DVB-T2 sudah mendukung GSE. GSE adalah protokol lapisan tautan data (*data link layer*) yang ditentukan oleh DVB. GSE menyediakan sarana untuk membawa protokol berorientasi paket seperti IP di atas lapisan fisik (*physical layer*) unidirectional dan memiliki dukungan untuk enkapsulasi multi-protokol (IPv4, IPv6, MPEG, ATM, Ethernet, VLAN 802.1pQ). *Bitrate* yang digunakan pada DVB-T2 lebih besar dari DVB-T yaitu hingga 32kbit/s per simbol menjadi sistem yang cocok untuk membawa sinyal HDTV pada saluran TV terestrisial (DVBProject 2013). Alur proses transmitter DVB-T2 ditunjukkan pada gambar 2.50.



Gambar 2.50: Sistem pemancar DVB-T2 (Hou 2010)

Tabel 2.6: Spesifikasi DVB-T dan DVB-T2 (Ya'acob et al. 2019)

Category	DVB-T	DVB-T2
Input Interface	Single Transport Stream (TS)	Multiple Transport Stream and Generic Stream Encapsulation (GSE)
FEC Rate	Convolutional Coding + Reed-Solomon, 1/2, 2/3, 3/4, 7/8	LDPC + BCH , 1/2, 3/5 , 2/3, 4/5 , 5/6
Modulation Scheme	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM, 256QAM
Guard Interval	1/4, 1/8, 1/16, 1/32	1/4, 19/256 , 1/8, 9/128 , 1/16, 1/32, 1/128
IFFT Point	2048, 8192	2048, 4096, 8192, 16384 , 32768

2.15 Jenis Kompresi Transmisi Video Digital

Kompresi transmisi video digital adalah proses mengurangi ukuran data video sebelum dikirimkan melalui jaringan. Ini dilakukan untuk memastikan bahwa video dapat diterima dengan cepat dan dengan kualitas yang baik, meskipun dengan kapasitas bandwidth yang terbatas. Kompresi video menggunakan algoritma kompresi untuk menghapus informasi redundant dalam video dan mengurangi ukuran data. Setelah video diterima, algoritma dekompresi digunakan untuk memulihkan video ke bentuk aslinya. Ada berbagai jenis kompresi video digital yang digunakan, seperti MPEG-2, MPEG-4 AVC/H.264, VC-1, VP9, dan HEVC. (Salomon & Motta 2010).

2.15.1 Video H.222 MPEG-TS

Moving Pictures Experts Group Transport Stream (MPEG-TS) dipilih untuk pengkodean sumber audio dan video untuk pembuatan aliran dasar program. *Transport Stream* MPEG-2, juga disebut MPEG-2 atau MPEG-2 TS atau hanya TS, adalah format khusus untuk mentransmisikan video MPEG (MPEG-1, MPEG-2, atau MPEG-4) dalam sebuah kontainer. TS menetapkan

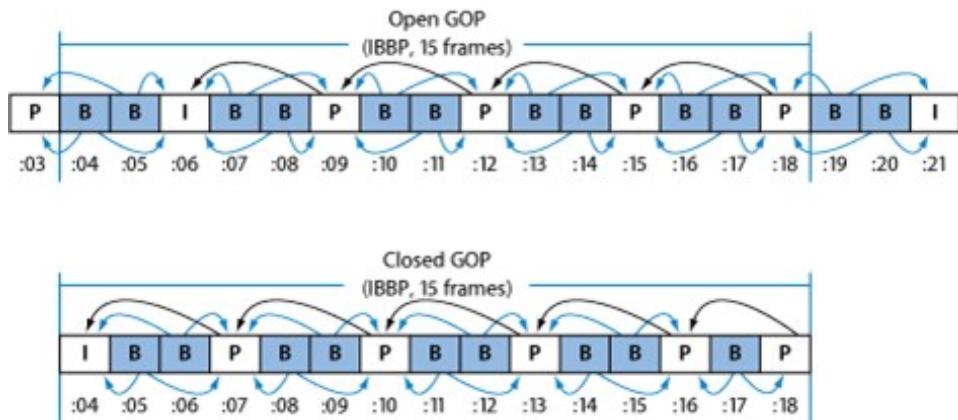
format wadah yang mengenkapsulasi aliran elemental paket (PES), dengan fitur koreksi kesalahan dan pola sinkronisasi untuk menjaga integritas transmisi ketika saluran komunikasi yang membawa aliran mengalami degradasi (ISO & IEC 2022).

MPEG-TS (MPEG Transport Stream) adalah standar transmisi untuk mengirimkan video digital dan audio dalam siaran televisi digital. Dalam siaran televisi digital, data video dan audio dikompresi menggunakan standar kompresi seperti MPEG-2 atau H.264, kemudian diteruskan dalam bentuk paket-paket yang disebut Transport Stream (TS). TS ini mengandung informasi tentang data video dan audio, serta informasi tambahan seperti pengaturan siaran dan informasi program. TS ini kemudian diteruskan melalui jaringan transmisi seperti saluran satelit atau jaringan kabel. Setelah sampai pada penerima, TS didekode dan diterjemahkan menjadi video dan audio yang dapat dilihat dan didengar (Hoelzer 2005, ISO & IEC 2022).

MPEG-TS menggunakan tiga jenis frame (I, P, dan B) untuk mewakili video. Pengaturan GOP menentukan pola tiga jenis frame yang akan digunakan. Ketiga tipe gambar ini antara lain Intra (*I-frame*), juga dikenal sebagai *frame* kunci. Setiap GOP berisi satu *I-frame*. *I-frame* adalah satu-satunya jenis *frame* MPEG-TS yang dapat didekompresi sepenuhnya tanpa referensi ke *frame* yang mendahului atau mengikutinya. Ini juga merupakan data yang paling berat, membutuhkan ruang disk paling banyak.

Berikutnya *Predicted frame (P-frame)*, dienkodekan dari gambar diprediksi berdasarkan pada *frame* I- atau P yang terdekat dan sebelumnya. *P-frame* biasanya membutuhkan ruang disk jauh lebih sedikit daripada *I-frame* karena mereka merujuk pada I- atau P-*frame* sebelumnya dalam GOP. Terakhir ada *Bi-directional (B-frame)*, dikodekan dari interpolasi *frame* referensi yang berhasil dan sebelumnya, baik dari *I-frame* atau *P-frame*. *B-frame* adalah jenis *frame* yang paling efisien penyimpanan, membutuhkan ruang disk paling sedikit.

it. Penggunaan B dan *P-frame* memungkinkan MPEG-TS untuk menghapus redundansi temporal, berkontribusi pada kemampuannya untuk mengompres video secara efisien (Apple-Inc. 2012).



Gambar 2.51: H.264/AVC Struktur GOP (Apple-Inc. 2012)

2.15.2 Video H.264 MPEG-4

MPEG-4 adalah metode mendefinisikan kompresi data digital audio dan visual (AV). Ini diperkenalkan pada akhir 1998 dan menetapkan standar untuk sekelompok format pengkodean audio dan video dan teknologi terkait yang disetujui oleh Kelompok Ahli Gambar Bergerak ISO / IEC (MPEG) (ISO / IEC JTC1 / SC29 / WG11) di bawah standar formal ISO / IEC 14496-Pengkodean objek audio-visual. Penggunaan MPEG-4 termasuk kompresi data AV untuk web (*streaming media*) dan distribusi CD, suara (telepon, *videophone*) dan aplikasi siaran televisi. Standar MPEG-4 dikembangkan oleh kelompok yang dipimpin oleh Touradj Ebrahimi (kemudian presiden JPEG) dan Fernando Pereira (Ebrahimi & Pereira 2002).

Dibandingkan dengan MPEG-2, AVC MPEG-4 yang jauh lebih baik (H.264) video codec memungkinkan kecepatan data diturunkan 30 hingga 50%. Ini berarti bahwa sinyal SDTV sekarang dapat dikompresi hingga kira-kira 1,5 - 3 Mbit/s dibandingkan dengan laju data 2-7 Mbit/s, laju data asli yang tidak terkompresi adalah 270 Mbit/s. Menggunakan MPEG-4, sinyal HDTV

dapat menyusut menjadi sekitar 10 Mbit / dtk dari aslinya 1,5 Gbit/ dtk. MPEG-2 membutuhkan sekitar 20 Mbit/s untuk ini (Fischer 2010).

2.16 *Signal to Noise Ratio (SNR)*

SNR didefinisikan dengan baik dan dipahami dalam teknik listrik dan komunikasi. SNR sering digunakan dalam pekerjaan yang berkaitan dengan proses pengukuran dan desain instrumen, tetapi relatif sedikit digunakan dalam berorientasi aplikasi(SANDY 5). SNR dalam bentuknya yang paling sederhana didefinisikan sebagai rasio kekuatan sinyal terhadap kekuatan *noise*. Dalam praktiknya, *noise* dikenali pada echogram sebagai latar belakang umum acak. *Noise* tersebut dapat dihilangkan dengan memilih *threshold* yang akan memberikan sinyal bebas interferensi tetapi tidak secara signifikan mengurangi sinyal yang diinginkan dalam rentang kedalaman sinyal yang diinginkan (Welvaert & Rosseel 2013). Nilai SNR yang lebih tinggi menunjukkan kualitas sinyal yang lebih baik dan SNR yang rendah menunjukkan ada banyaknya *noise* pada sinyal tersebut (Altunian 2021).

SNR biasanya dinyatakan dalam Desibel (dB), terutama dalam aplikasi audio dan suara karena rentang dinamis pendengaran manusia yang sangat besar (Kessel 2018). Desibel (dB) sering digunakan untuk menyatakan rasio tidak berunit. dB bukanlah "satuan" dalam arti meter, newton, detik. Desibel dapat dianggap seperti persen, lusin atau bagian per juta sehingga dB adalah bilangan tak berdimensi (Centauri 2013). Desibel menjadi cara untuk mengekspresikan nilai pada skala logaritmik. Komponen audio, sebagai contoh mempunyai SNR senilai 100 dB, itu berarti level sinyal audio 100 dB lebih tinggi daripada *noise*. Spesifikasi SNR 100 dB jauh lebih baik daripada yang 70 dB atau kurang.

Sebagai ilustrasi, katakanlah Anda sedang berbicara dengan seorang teman di dapur yang kebetulan juga memiliki lemari es yang sangat keras.

Katakan juga bahwa kulkas menghasilkan 50 dB dengungan, anggap ini kebisingan karena membuat isinya tetap dingin. Jika teman yang Anda ajak bicara berbicara dengan suara 30 dB, anggap suara tersebut adalah sinyal. Anda tidak akan bisa mendengar sepatah kata pun karena suara kulkas menyalahkan ucapan teman Anda. Anda mungkin meminta teman Anda untuk berbicara lebih keras, tetapi bahkan pada 60 dB, Anda mungkin masih perlu meminta mereka untuk mengulanginya. Berbicara pada 90 dB mungkin tampak lebih seperti pertandingan berteriak, tetapi setidaknya kata-kata akan didengar dan dipahami (Altunian 2021). Formula 2.1 merupakan rumus untuk mencari nilai SNR (Kieser et al. 2005)

$$SNR_{db} = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \quad (2.1)$$

Dimana:

P_{signal} : nilai sinyal asli dalam bentuk watt

P_{noise} : nilai sinyal *noise* dalam bentuk watt

2.17 *Signal Strength*

Kekuatan sinyal atau *signal strength* berarti ukuran seberapa kuat sinyal transmisi yang diterima, diukur atau diprediks, pada titik referensi yang merupakan jarak dari antena pemancar (Hendrickson 2022). *signal strength* umumnya berkurang dengan bertambahnya jarak (Matthews et al. 2018). Sinyal TV adalah tegangan, dB mengacu pada skala mikro-volt yang digunakan untuk implementasi sistem TV menjadi proses penambahan dan pengurangan yang sederhana. *signal strength* didasarkan pada skala mikro-volt dB. Skala dimulai pada -30dBm akan menjadi mikro-Watt, -60dBm menjadi 1 nano-Watt, 0dBm yang akan menjadi 1 miliWatt, 30dBm akan menjadi 1 Watt, . *signal strength*

mengalami peningkatan sebanyak 30 dB per kelipatan 1000. Peningkatan tersebut menunjukkan bahwa ketika berhadapan dengan pengukuran sinyal yang lemah atau terbatas, peningkatan kecil dalam dB sering kali dapat membuat perbedaan besar (Frost 2014). Tingkat kekuatan sinyal minimum yang disarankan adalah sebagai berikut (Hendrickson 2022):

- TV Digital Terrestrial -45dB (Idealnya tidak kurang dari -50dB) <- Cari referensinya lagi
- TV Analog -60dB
- Sinyal TV Satelit -47dB (Idealnya tidak kurang tidak -52dB)

Kualitas atau sinyal yang *robust* atau kokoh diukur dalam hal kekuatan sinyal asli yang dikurangi *noise* yang dapat berasal dari berbagai sumber. *Noise* listrik sebenarnya akan ada di dalam sinyal, jadi inilah mengapa men-gandalkan kekuatan sinyal saja tidak selalu merupakan faktor penentu kualitas sinyal yang handal. Meskipun demikian, semakin banyak sinyal yang dapat diperoleh dari pemancar TV, semakin besar perlindungan yang Anda miliki terhadap gangguan dan gangguan listrik (Matthews et al. 2018). Formula 2.2 merupakan rumus untuk mencari nilai *signal strength*

$$L_{dBm} = 10 \log_{10} \left(\frac{P}{0,001W} \right) \quad (2.2)$$

Dimana:

L_{dBm} : nilai *signal strength* dalam satuan dBm

P : nilai sinyal yang diterima oleh antena *receiver* dalam satuan mW

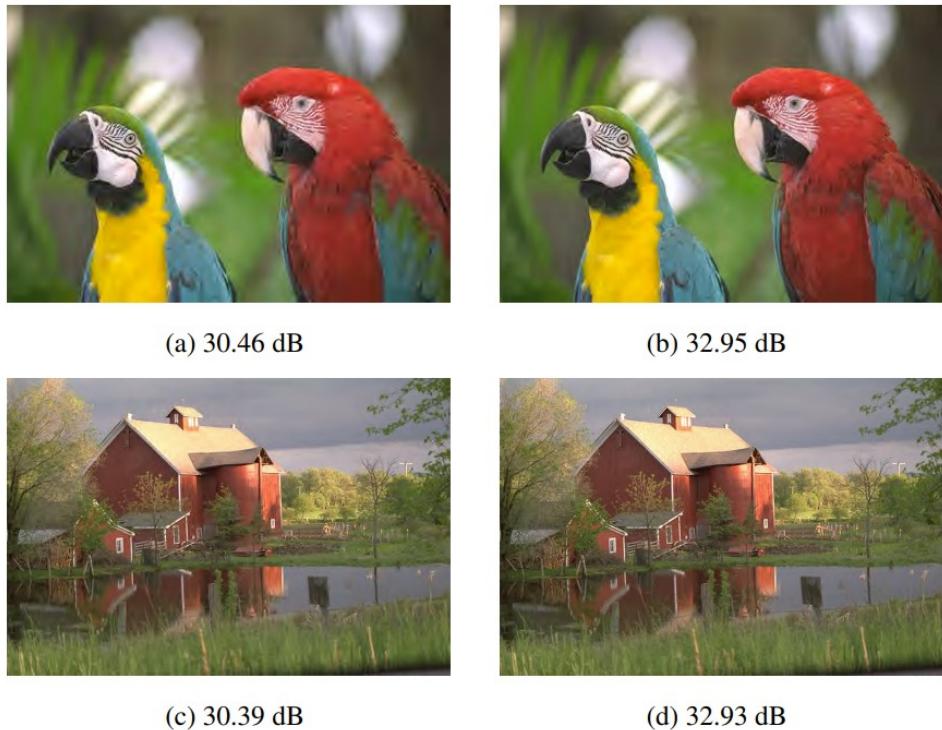
2.18 Metrik Kualitas Gambar

Penilaian kualitas citra dapat diartikan sebagai menilai atau mengukur kualitas suatu citra yang sesuai atau mengacu pada citra aslinya. Gambar dalam kompresi, jika diambil terdapat distorsi yang besar maka tidak akan cocok dengan gambar asli yang disimpan dalam dataset sehingga menemukan kualitas gambar di area tersebut sangat diperlukan. *image quality assessment algorithms* (IQA) merupakan salah satu proses penilaian kualitas gambar secara objektif, algoritma tersebut akan memprediksi kualitas gambar secara objektif. Metode objektif yang dipakai adalah metode klasifikasi tanpa referensi (Kumar & Prajapati 2015).

Metode objektif bukan satu-satunya perspektif dimana suatu gambar hanya dinilai dari sudut pandang komputer. Suatu gambar dapat dinilai dengan mata manusia sehingga mempertimbangkan figur tambahan jasa atau Pengamat dapat memberikan wawasan yang jauh lebih informatif. DAta keluaran pada gambar, secara umum dinyatakan dari metrik objektif untuk stimulus visual yang diberikan dinyatakan sebagai nilai tunggal pada suatu skala berkelanjutan. Data keluaran dengan metode objektif tersebut menunjukkan bahwa ketika skor kualitas yang diprediksi pada gambar dengan nilai yang diperoleh, keunggulan kualitas tiap gambar selalu terbentuk, tidak peduli seberapa kecil perbedaannya. Perbedaan skor kualitas yang bukan bernilai nol antara dua gambar serupa dapat menyebabkan keambiguan ketika perbedaan kualitas tidak terlihat oleh Pengamat. Kepekaan visual manusia terbatas dalam arti bahwa sejumlah kecil perbedaan nilai piksel terkadang tidak dapat dibedakan secara visual tergantung pada beberapa faktor seperti pencahayaan keseluruhan dan nilai piksel (Cheon et al. 2021). Metode penilaian subjektif biasanya digunakan untuk menghitung kualitas gambar. Penilaian subjektif ini digunakan oleh Pengamat untuk menilai kualitas gambar. Gambar

diberikan kepada Pengamat. Pengamat diberikan persyaratan waktu, Pengamat tersebut memberikan skor atau nilai pada gambar. Hasil subjektif dapat memberikan hasil yang akurat (Kumar & Prajapati 2015).

Gambar 2.52 menunjukkan contoh gambar yang menunjukkan adanya ambiguitas (Cheon & Lee 2016). Dua gambar referensi (burung beo dan rumah) diambil dari *database* Penilaian Kualitas Gambar LIVE (SANDY 18), kompresi JPEG2000 dilakukan untuk memberi *noise* pada gambar dengan bitrate yang berbeda. Ketika Gambar 2.52a dan 2.52b dibandingkan secara visual, perbedaan kualitasnya dapat dengan mudah dibedakan. Penilaian kualitas subjektif dilakukan percobaan, dimana sebagian besar Pengamat (14 dari 15) memilih Gambar 2.52b sebagai yang memiliki kualitas lebih baik. Metode objektif dengan PSNR juga menilai Gambar 2.52b memiliki kualitas yang lebih baik (dengan perbedaan 2,49 dB). Perbedaan antara Gambar 2.52c dan 2.52d hampir tidak terlihat, hampir setengah dari Pengamat (6 dari 15) memilih Gambar 2.52c. Kualitas yang diukur dengan *peak signal-to-noise ratio* (PSNR) masih menentukan bahwa Gambar 2.52d lebih baik. Gambar 2.52d menunjukkan perbedaan sebesar 2,54 dB, yang bahkan lebih besar dari perbedaan antara Gambar 2.52a dan 2.52b. Hasil yang tidak konsisten antara pengukuran kualitas subjektif dan objektif tidak dapat dilakukan untuk sistem multimedia yang mengoptimalkan kualitas. Sistem yang mengandalkan PSNR mungkin mencoba memberikan Gambar 2.52d daripada Gambar 2.52c untuk meningkatkan QoE dengan bit yang meningkat (20 hingga 35 kbyte), yang sebenarnya tidak begitu layak bagi Pengamat.



Gambar 2.52: Contoh gambar dari Basis Data Penilaian Kualitas Gambar LIVE, yang menunjukkan ambiguitas metrik kualitas objektif (dalam hal ini, PSNR)(Sheikh et al. 2006)

2.18.1 Penilaian Kualitas Gambar Referensi Penuh

Penilaian secara objektif dapat diklasifikasikan menjadi dua jenis. *Full-reference image quality assessment (FR-IQA)* atau Penilaian kualitas referensi penuh yaitu membandingkan gambar referensi atau asli dengan gambar uji. *No-reference image quality assessment (NR-IQA)* atau tidak ada penilaian kualitas referensi yang mengacu pada pemeriksaan kualitas oleh algoritma dimana hanya gambar uji (yaitu, gambar terdistorsi) yang dapat diakses (Varga 2021, Dihin et al. 2020).

IQA yang sangat menjanjikan tetapi relatif kurang dipelajari digunakan sebagai tujuan untuk desain dan optimalisasi algoritma pemrosesan gambar baru. Parameter metode pemrosesan gambar biasanya disesuaikan untuk meminimalkan *mean squared error* (MSE). MSE merupakan metode yang

paling sederhana dari semua metode pembanding kualitas gambar, meskipun telah banyak dikritik karena korelasinya yang buruk dengan persepsi manusia tentang kualitas gambar. Upaya awal optimasi persepsi menggunakan indeks *structural similarity* (SSIM) sebagai pengganti MSE untuk mencapai keuntungan persepsi dalam aplikasi restorasi gambar, streaming video nirkabel, pengkodean video dan sintesis gambar, meskipun ini belum diuji terhadap penilaian manusia (Ding et al. 2021).

2.18.2 Penilaian Kualitas Gambar Tanpa Referensi

Penilaian kualitas gambar tanpa referensi (No-Reference Image Quality Assessment, NR-IQA) adalah metode penilaian kualitas gambar yang tidak memerlukan gambar referensi untuk melakukan penilaian. penilaian kualitas gambar atau video dalam kategori ini dilakukan secara buta berdasarkan fitur yang diekstraksi dari konten multimedia yang dinilai karena tidak ada referensi yang tersedia. Evaluasi kualitas gambar dan video berbasis NR adalah tugas yang menantang karena fitur yang diekstraksi dapat memberikan informasi yang sangat terbatas.(Dost et al. 2022).

Metode NR-IQA dapat digunakan untuk mengevaluasi kualitas gambar yang telah mengalami distorsi seperti kompresi, blur, noise, blocking artifact, dan kerusakan temporal. Dalam metode NR-IQA, tidak ada gambar referensi yang digunakan sebagai pembanding, sehingga penilaian kualitas gambar dilakukan secara independen dan tidak bergantung pada gambar referensi yang digunakan.

2.18.2.1 Metrik NR-Bloking

Blocking muncul di semua teknik kompresi berbasis blok dan disebabkan oleh kuantisasi kasar komponen frekuensi. Artefak ini dapat diamati sebagai diskontinuitas permukaan atau tepi pada batas blok. Masalah dengan skema berba-

sis blok seperti JPEG adalah gambar dibagi menjadi sub-blok dengan ukuran masing-masing piksel berukuran 8x8. Transformasi dan proses kuantisasi kemudian diterapkan pada sub-blok secara individual dan independen. Korelasi antara sub-blok yang berdekatan secara spasial tidak diperhitungkan selama proses pengkodean, oleh karena itu transisi halus antara batas tepi setiap sub-blok berkurang. Selama proses decoding, batas tepi tidak dapat dipulihkan sepenuhnya seperti yang terlihat pada gambar asli. Batas blok sekarang terlihat. *blockiness* atau artefak pemblokiran dapat dengan mudah diamati pada gambar seperti yang ditunjukkan pada Gambar 2.53, bahkan lebih terlihat ketika bit rate atau jumlah bit untuk mewakili gambar dikurangi (Kusuma 2005).



Gambar 2.53: Contoh gambar asli Lena dan versi terdistorsinya dengan artefak pemblokiran (Kusuma 2005)

2.18.2.2 Metrik NR-Blur

Blur diamati sebagai kehalusan pada tepi atau kurangnya detail seperti yang ditunjukkan pada Gambar 4.13. Hal ini disebabkan oleh hilangnya komponen frekuensi tinggi jika dibandingkan dengan gambar aslinya. Secara matematis, gambar *blur* dapat dimodelkan sebagai berikut:

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (2.3)$$

dimana $g(x, y)$, $f(x, y)$, dan $h(x, y)$ masing-masing mewakili gambar buram, gambar asli, dan *point spread function* (PSF) atau fungsi *blur*. Fungsi $n(x, y)$ menunjukkan *noise* tambahan dari akuisisi citra jika ada. Simbol $*$ adalah operator konvolusi. Artefak *blur* sebagian besar terjadi pada gambar terkompresi berbasis wavelet, seperti JPEG2000. Hal ini disebabkan oleh dekomposisi multi-resolusi dari transformasi wavelet. Jika sebuah gambar sangat terkompresi, maka hanya koefisien frekuensi rendah yang dipertahankan dalam gambar terkompresi. Akibatnya, gambar kehilangan detail halus yang terkait dengan komponen frekuensi tinggi. Informasi bentuk pada dasarnya dipertahankan sementara informasi tekstur sangat diperhalus. Skema kompresi berbasis DCT seperti JPEG juga menunjukkan *blur*, meskipun itu bukan artefak utama (Kusuma 2005).

2.18.2.3 Metrik NR-Temporal

Frozen frame sebagai *frame* video yang identik dengan yang sebelumnya (*frame repeat*) dan mendefinisikan peristiwa pembekuan sebagai satu set *frozen frame* berturut-turut. Setiap peristiwa pembekuan dicirikan oleh durasi pembekuan tergantung pada jumlah *frame* yang dibekukan secara berurutan dalam peristiwa itu. Total durasi pembekuan video kemudian diwakili oleh akumulasi semua durasi semua peristiwa pembekuan. Dalam contoh ilustrasi Gambar 2.54, video referensi memiliki kecepatan *frame* 25 fps. Oleh karena itu, setiap *frame* unik memiliki durasi 40 ms. Video yang terdegradasi berisi tiga peristiwa pembekuan, dengan total sepuluh *frozen frame*. Setiap *frame* yang dibekukan memiliki durasi 40 ms dan ketiga peristiwa pembekuan memiliki durasi masing-masing 80, 120 dan 200 ms. Total durasi pembekuan dalam video adalah 400 ms.

Ref video:	A	...	D	E	F	G	I	J	...	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Deg video:	A	...	D	E	E	E	I	J	...	L	L	L	L	P	Q	Q	Q	Q	Q	Q	W	X	Y	Z

Gambar 2.54: Contoh video yang terganggu oleh pembekuan frame temporal. Setiap huruf yang berbeda mewakili frame yang unik. (Huynh-Thu & Gha 2009)

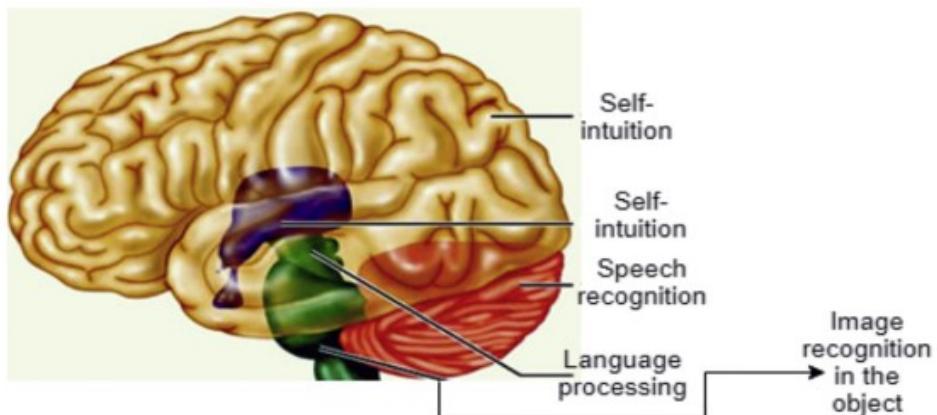
Pendekatan tanpa referensi, identifikasi *frozen frame* harus dilakukan hanya dengan menggunakan urutan yang diproses karena akses ke referensi tidak dimungkinkan. Kasus pendekatan tanpa referensi, tidak mungkin untuk membedakan konten diam (yaitu konten yang sengaja tidak bergerak) dari *frozen frame* (yaitu konten yang tidak bergerak sebagai akibat dari gangguan video). Pendekatan umum untuk mendeteksi *frozen frame* adalah dengan menghitung *mean-squared error* (MSE) antara *frame* saat ini dan sebelumnya dan mempertimbangkan *frame* saat ini untuk dibekukan jika MSE sama dengan nol (Huynh-Thu & Gha 2009).

2.18.3 Artificial Neural Network (ANN)

ANN diterapkan dalam prediksi berbagai proses. JST telah berhasil diterapkan di berbagai bidang matematika, teknik, kedokteran, ekonomi, neurologi. ANN dapat didefinisikan sebagai jaringan yang kompleks, yang terdiri dari unit pemrosesan dasar yang saling berhubungan yang disebut neuron. ANN dapat ditentukan oleh tiga faktor yaitu; Struktur, Algoritma pembelajaran, dan fungsi aktivasi (SANDY 26). Aplikasi ANN dapat dievaluasi sehubungan dengan faktor analisis data seperti akurasi, kecepatan pemrosesan, latensi, kinerja, toleransi kesalahan, volume, skalabilitas, dan konvergensi. Potensi besar ANN adalah pemrosesan berkecepatan tinggi yang disediakan dalam implementasi paralel besar-besaran dan ini telah meningkatkan kebutuhan untuk penelitian dalam domain ini. ANN dapat dikembangkan dan digunakan untuk pengenalan gambar, pemrosesan bahasa alami dan sebagainya.

Saat ini, ANN banyak digunakan untuk pendekatan fungsi universal dalam paradigma numerik karena sifat yang sangat baik dari belajar mandiri, adaptif, toleransi kesalahan, nonlinier, dan kemajuan input ke pemetaan output (SANDY 24).

Gambar 2.55 adalah demonstrasi koneksi di dalam otak yang bekerja seperti jaringan saraf yang melakukan fungsi penalaran kecerdasan. Brain-storming untuk memahami suatu skenario (seperti platform pencarian web internet), mengenali ucapan (misalnya dari orang yang dikenal dan orang yang tidak dikenal) seperti otak manusia, mengenali gambar (dari suatu objek) seperti otak, dapat memproses bahasa (menerjemahkan bahasa) seperti yang dilakukan otak manusia dan dapat melakukan hal-hal lain seperti makan, mengendarai sepeda (intuisi diri). ANN melihat penggunaan besar-besaran dalam domain tertentu, seperti diagnosis hepatitis, pengenalan suara, pemulihan data dalam telekomunikasi dari perangkat lunak yang rusak, interpretasi pesan multi-bahasa, pengenalan objek tiga dimensi, analisis tekstur, pengenalan wajah, deteksi ranjau bawah laut, dan pengenalan kata tulisan tangan. ANN dapat belajar dengan contoh seperti orang. Dalam beberapa kasus, ANN dapat dirancang untuk aplikasi tertentu seperti klasifikasi data atau pengenalan pola melalui proses pembelajaran. Pembelajaran di otak manusia memerlukan penyesuaian hubungan sinaptik antara dan antar neuron, demikian juga pembelajaran di ANN. Secara umum, ANN berfungsi seperti tiruan dari otak manusia (SANDY 24).



Gambar 2.55: Struktur otak manusia dengan kemampuan operasional (SANDY 24)

Data masukan pada ANN akan diberikan pada setiap masukan bobot, yang bisa berupa angka positif atau negatif. Sebuah input dengan bobot positif yang besar atau bobot negatif yang besar, akan memiliki pengaruh yang kuat terhadap output neuron. Sebelum dimulai, ANN harus menetapkan setiap bobot ke nomor acak, kemudian memulai proses pelatihan.

1. Ambil input dari contoh set pelatihan, sesuaikan dengan bobot, dan berikan melalui formula khusus untuk menghitung output neuron.
2. Hitung error, yang merupakan selisih antara output neuron dan output yang diinginkan dalam contoh set pelatihan.
3. Bergantung pada arah kesalahan, sesuaikan bobotnya sedikit.
4. Ulangi proses ini 10.000 kali.

Akhirnya bobot neuron akan mencapai optimal untuk set pelatihan. Jika pengguna membiarkan neuron berpikir tentang situasi baru dengan mengikuti pola yang sama, itu akan membuat prediksi yang baik (SANDY 25). Gambar 2.57 merupakan gambaran *training* pada ANN. formula 2.4 merupakan rumus dari jumlah bobot dari data masukan neuron.

$$\sum w_i \cdot x_i = w_{i \cdot 1} + w_2 \cdot x_2 + \dots + w_n \cdot x_n \quad (2.4)$$

Dimana:

$\sum w_i \cdot x_i$: data keluaran

w_i : bobot

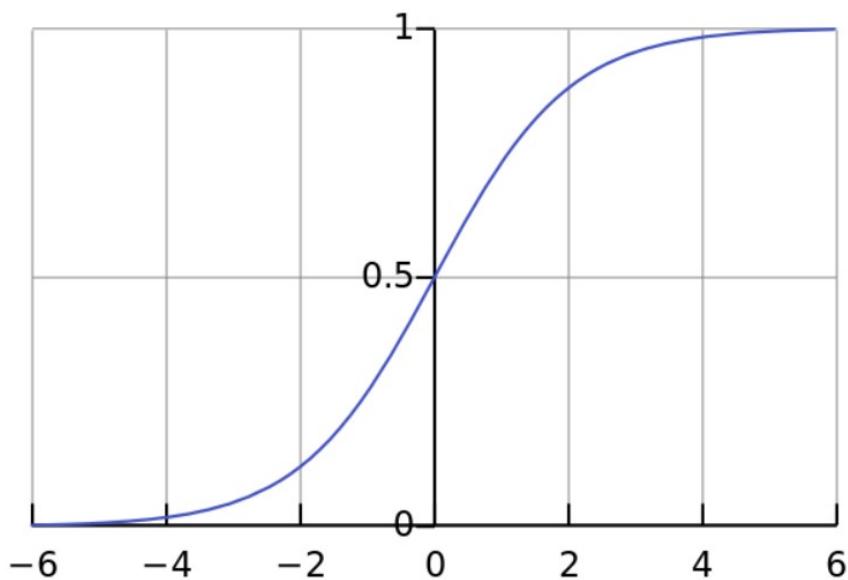
x_i : data masukan

Hasil dari data keluaran tersebut dinormalisasi sehingga hasilnya antara 0 dan 1. Normalisasi tersebut menggunakan fungsi yang sesuai secara matematis yang disebut fungsi Sigmoid.

$$\frac{1}{1 + e^{-x}} \quad (2.5)$$

Fungsi Sigmoid jika diplot pada grafik, fungsi Sigmoid menggambarkan kurva berbentuk S yang ditunjukkan pada gambar 2.56. Jadi dengan men-substitusi formula 2.4 ke formula 2.5. Rumus akhir untuk keluaran neuron ditunjukkan pada formula 2.6, dimana y adalah data keluaran neuron

$$y = DataKeluaranNeuron = \frac{1}{1 + e^{-(\sum w_i \cdot x_i)}} \quad (2.6)$$



Gambar 2.56: Fungsi Sigmoid

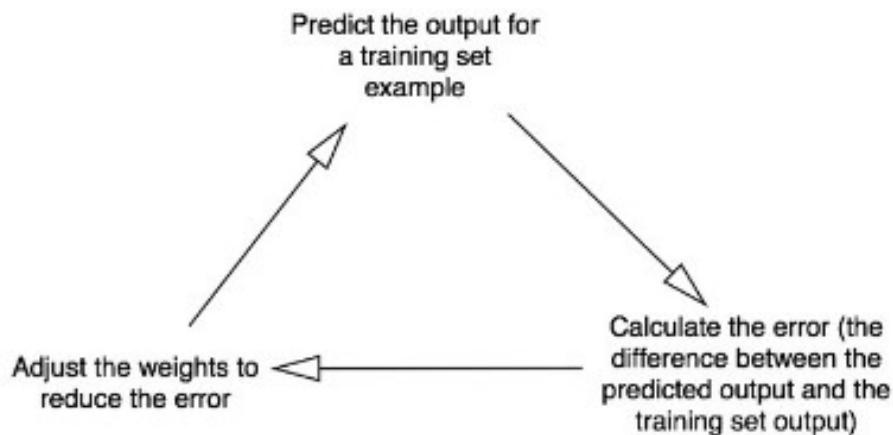
Gambar 2.57 terdapat penyesuaian bobot. Penyesuaian bobot tersebut dapat menggunakan rumus *Error Weighted Derivative* yang ditunjukkan pada formula 2.7.

$$\text{PenyesuaianBobot} = \text{Error.Input.SigmoidCurveGradient}(y) \quad (2.7)$$

Dimana:

error : data keluaran - data keluaran asli

input : data terukur



Gambar 2.57: Langkah training ANN (SANDY 25)

Tujuan dari formula 2.7 adalah membuat penyesuaian proporsional dengan ukuran kesalahan. Kalikan dengan data masukannya, yaitu 0 atau 1. Jika data masukannya 0, bobotnya tidak disesuaikan. Kalikan dengan gradien kurva Sigmoid (Gambar 2.56) untuk langkah selanjutnya. Berikut adalah langkah-langkah dari perhitungan formula 2.7

1. Menggunakan kurva Sigmoid untuk menghitung data keluaran dari neuron
2. Jika data keluarannya adalah angka positif atau negatif yang besar, itu menandakan neuron cukup percaya diri dengan satu atau lain cara.
3. Dari gambar 2.56, kita dapat melihat bahwa pada bilangan besar, kurva Sigmoid memiliki gradien yang dangkal.
4. Jika neuron yakin bahwa bobot yang ada benar, neuron tidak ingin terlalu banyak menyesuaikannya. Mengalikan dengan gradien kurva Sigmoid dapat mencapai bobot yang benar.

Gradien kurva Sigmoid, dapat ditemukan dengan mengambil turunan:

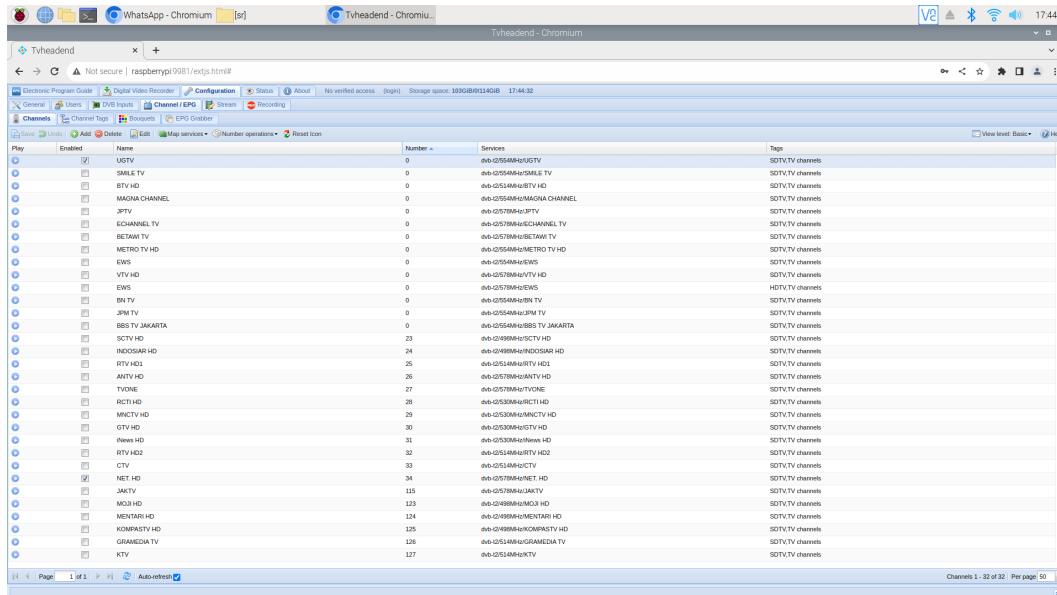
$$\text{SigmoidCurveGradient}(y) = y.(1 - y) \quad (2.8)$$

Jadi dengan memasukkan forluma 2.8 ke formula 2.7, rumus akhir untuk menyesuaikan bobot adalah:

$$\text{PenyesuaianBobot} = \text{Error.input}.y.(1 - y) \quad (2.9)$$

2.19 Perangkat Lunak *tvheadend*

Tvheadend adalah aplikasi server TV streaming yang open-source dan gratis. Aplikasi ini memungkinkan pengguna untuk menonton TV langsung dari komputer atau perangkat lain melalui jaringan. Tvheadend dapat digunakan dengan berbagai jenis perangkat lunak client, termasuk Kodi, VLC, dan banyak lagi. Tvheadend mendukung berbagai jenis tunner TV dan format pemrosesan video. Aplikasi ini juga menyediakan berbagai fitur, seperti merekam dan menunda TV langsung, dan memungkinkan pengguna untuk mengatur saluran dan membuat daftar putar. Tvheadend tersedia untuk berbagai platform, termasuk Linux, macOS, dan Windows (Foundation 2015).



Gambar 2.58: Antarmuka website perangkat lunak *tvheadend*

TVHeadend digunakan untuk menangkap, mengakses, dan mengelola sinyal TV dan radio yang disiarkan melalui antena, satelit, dan kabel. Fungsi utama TVHeadend adalah sebagai server TV yang memungkinkan pengguna untuk menonton dan merekam program TV dan radio melalui jaringan lokal atau internet. TVHeadend juga menyediakan fitur EPG (Electronic Program Guide) untuk memberikan informasi tentang jadwal program dan fitur PVR (Personal Video Recorder) untuk merekam dan menyimpan program TV seperti yang ditunjukkan pada gambar 2.58. Selain itu, TVHeadend dapat diintegrasikan dengan beberapa perangkat lunak media center seperti Kodi dan Plex untuk menyediakan pengalaman media center yang lebih lengkap (Emmet 2022).

2.20 *Subjective Assessment*

Metodologi untuk penilaian subyektif kualitas gambar televisi ITU-R adalah seperangkat prosedur dan pedoman yang digunakan untuk melakukan penilaian kualitas gambar televisi dengan melibatkan orang-orang sebagai pe-

nilai. Pedoman ini dikembangkan oleh International Telecommunication Union-Radiocommunication Sector (ITU-R) dan digunakan secara luas oleh industri penyiaran dan penyedia layanan televisi untuk mengevaluasi kualitas gambar televisi mereka. Penilaian subyektif kualitas gambar televisi dapat memberikan umpan balik yang berharga tentang bagaimana kualitas gambar tersebut diterima oleh penonton dan membantu penyedia layanan meningkatkan kualitas gambar mereka (Rodriguez et al. 2014).

Pada penelitian ini digunakan standar pengukuran subyektif ITU-R BT.500-14 yang merupakan seperangkat pedoman dan prosedur untuk penilaian subyektif kualitas gambar televisi. Pedoman tersebut menyediakan metode standar untuk mengevaluasi kualitas gambar televisi secara konsisten dan dapat diulang. Standar tersebut banyak digunakan dalam industri penyiaran dan telekomunikasi untuk memastikan bahwa gambar televisi memenuhi tingkat kualitas tertentu. Pada standar tersebut juga mendefinisikan beberapa metode penilaian subyektif yang berbeda. Metode-metode ini dirancang untuk mengevaluasi berbagai aspek kualitas gambar, seperti resolusi, akurasi warna, dan visibilitas artefak (ITU-R BT500 2019). Beberapa metode yang terdapat pada standar tersebut diantaranya:

1. Double-stimulus impairment scale (DSIS).

Metode ini melibatkan dua gambar atau video yang ditampilkan secara bergantian di layar, yaitu gambar atau video referensi yang dianggap berkualitas tinggi dan gambar atau video uji yang mengalami gangguan atau distorsi tertentu. Panelis diminta untuk memberikan skor kualitas pada gambar atau video uji berdasarkan seberapa besar gangguan atau distorsi yang mereka rasakan dibandingkan dengan gambar atau video referensi. Metode DSIS memiliki keuntungan dalam mengurangi efek bias atau kecenderungan panelis dalam memberikan skor kualitas karena panelis harus membandingkan gambar atau video uji dengan gambar

atau video referensi yang dianggap berkualitas tinggi. Namun, metode ini dapat memakan waktu yang lebih lama karena panelis harus melihat dua gambar atau video secara bergantian dan memberikan skor kualitas untuk setiap gambar atau video uji.

2. Single Stimulus (SS)

Metode Single-stimulus (SS) adalah metode subjektif dalam penilaian kualitas citra atau video di mana penonton diberi satu citra atau video pada setiap waktu dan diminta untuk memberikan nilai kualitas. Penonton memberikan nilai kualitas pada skala tertentu, seperti skala MOS (Mean Opinion Score) yang berkisar dari 1 hingga 5 atau 1 hingga 10, di mana nilai yang lebih tinggi menunjukkan kualitas yang lebih baik. Metode SS terdiri dari beberapa varian, seperti Single Stimulus Continuous Quality Evaluation (SSCQE) dan Absolute Category Rating (ACR). Metode SS sering digunakan untuk penilaian kualitas video pada televisi dan perangkat seluler.

3. Single Stimulus Continuous Quality Evaluation (SSCQE)

Metode penilaian subyektif dalam penilaian kualitas gambar atau video. Dalam metode ini, penilai diberikan tampilan gambar atau video tunggal untuk dinilai secara terus menerus tanpa adanya perbandingan dengan tampilan gambar atau video lainnya. Penilai diminta memberikan penilaian kontinu pada kualitas gambar atau video yang mereka lihat menggunakan skala penilaian dari sangat buruk hingga sangat baik. Metode ini dapat digunakan untuk mengukur kualitas gambar atau video pada aspek-aspek tertentu seperti kecerahan, kontras, ketajaman, atau aspek lainnya.

4. Simultaneous Double Stimulus for cCntinuous Evaluation (SDSC)

Metode subyektif untuk mengevaluasi kualitas gambar televisi. Metode ini melibatkan pemutaran dua sumber video secara bersamaan, yaitu sumber video referensi dan sumber video yang diuji. Penonton kemudian diminta untuk menilai kualitas video yang diuji dengan cara memberikan nilai pada skala kualitas yang berkelanjutan pada layar monitor, sementara video referensi tetap diputar. Metode SDSC memungkinkan penonton untuk membandingkan kualitas gambar dari dua sumber video secara langsung, sehingga dapat memberikan penilaian yang lebih akurat dan objektif terhadap kualitas gambar televisi.

Standar ITU-R BT.500 juga memberikan panduan tentang bagaimana memilih dan melatih panel penonton untuk penilaian kualitas subyektif. Ini menjelaskan kualifikasi yang diperlukan untuk panelis, jumlah panelis yang dibutuhkan, dan proses pelatihan untuk memastikan bahwa panelis konsisten dan dapat diandalkan dalam penilaian mereka. Secara keseluruhan, standar ITU-R BT.500 menyediakan kerangka kerja yang komprehensif untuk penilaian subyektif kualitas gambar televisi, yang penting untuk memastikan bahwa penonton menerima pengalaman menonton berkualitas tinggi (Martínez-Rach et al. 2014).

2.20.1 Raspberry Pi + TV Hat

2.20.2 DVB-T2 TV Tunner

2.20.3 *Confusion Matrix*

Confusion matrix adalah tabel evaluasi performa model yang digunakan untuk menunjukkan jumlah prediksi benar dan salah yang dibuat oleh model pada data uji. Tabel ini digunakan untuk menghitung akurasi, presisi, dan recall dengan membandingkan hasil prediksi model dengan nilai sebenarnya.

narnya dari data uji. Confusion matrix digunakan pada tugas klasifikasi dengan dua kelas target dan terdiri dari empat kemungkinan hasil: true positive, false positive, true negative, dan false negative.

Multiple class confusion matrix adalah bentuk tabel evaluasi performa model yang digunakan untuk mengevaluasi performa model pada data uji dengan lebih dari dua kelas target. Tabel ini terdiri dari baris dan kolom, di mana setiap baris dan kolom mewakili kelas target dan prediksi yang berbeda. Diagonal matriks ini menunjukkan jumlah prediksi benar, sementara sel di luar diagonal menunjukkan jumlah prediksi salah. Multiple class confusion matrix digunakan untuk menghitung akurasi, presisi, dan recall pada setiap kelas target, serta metrik evaluasi lainnya seperti F1 score. Metrik evaluasi ini berguna dalam mengevaluasi performa model pada tugas klasifikasi dengan lebih dari dua kelas target.

Akurasi dan presisi adalah dua metrik yang digunakan untuk mengukur kinerja model dalam klasifikasi, termasuk dalam konteks matriks kebingungan multi-kelas (MCCM). Berikut adalah perbedaan antara keduanya:

Akurasi: Akurasi adalah ukuran kinerja yang menggambarkan seberapa baik model klasifikasi memprediksi kelas yang benar. Dalam konteks MCCM, akurasi dihitung dengan mengambil jumlah prediksi yang benar (nilai pada diagonal utama matriks) dibagi dengan jumlah total prediksi (semua elemen dalam matriks). Akurasi mencakup semua kelas dan memberikan gambaran umum tentang seberapa baik model bekerja.

Presisi: Presisi adalah ukuran kinerja yang menggambarkan seberapa baik model klasifikasi mengidentifikasi hasil positif yang benar dari keseluruhan hasil positif yang diprediksi. Dalam konteks MCCM, presisi dihitung untuk setiap kelas secara terpisah menggunakan rumus: Presisi = TP / (TP + FP), di mana TP (True Positives) dan FP (False Positives) dihitung dari matriks kebingungan. Kemudian, presisi untuk setiap kelas digabungkan meng-

gunakan metode seperti rata-rata tertimbang berdasarkan jumlah sampel per kelas, rata-rata sederhana (macro-average), atau micro-average.

Perbedaan utama antara akurasi dan presisi adalah fokus pada keseluruhan kinerja model (akurasi) vs. kinerja model dalam mengidentifikasi hasil positif yang benar (presisi). Akurasi memberikan gambaran umum tentang seberapa baik model memprediksi kelas yang benar, sedangkan presisi menyoroti kinerja model dalam mengidentifikasi hasil positif yang benar untuk setiap kelas. Presisi sangat berguna dalam kasus di mana false positives memiliki konsekuensi yang lebih tinggi, seperti dalam diagnosis medis atau deteksi spam.

BAB III

METODE PENELITIAN

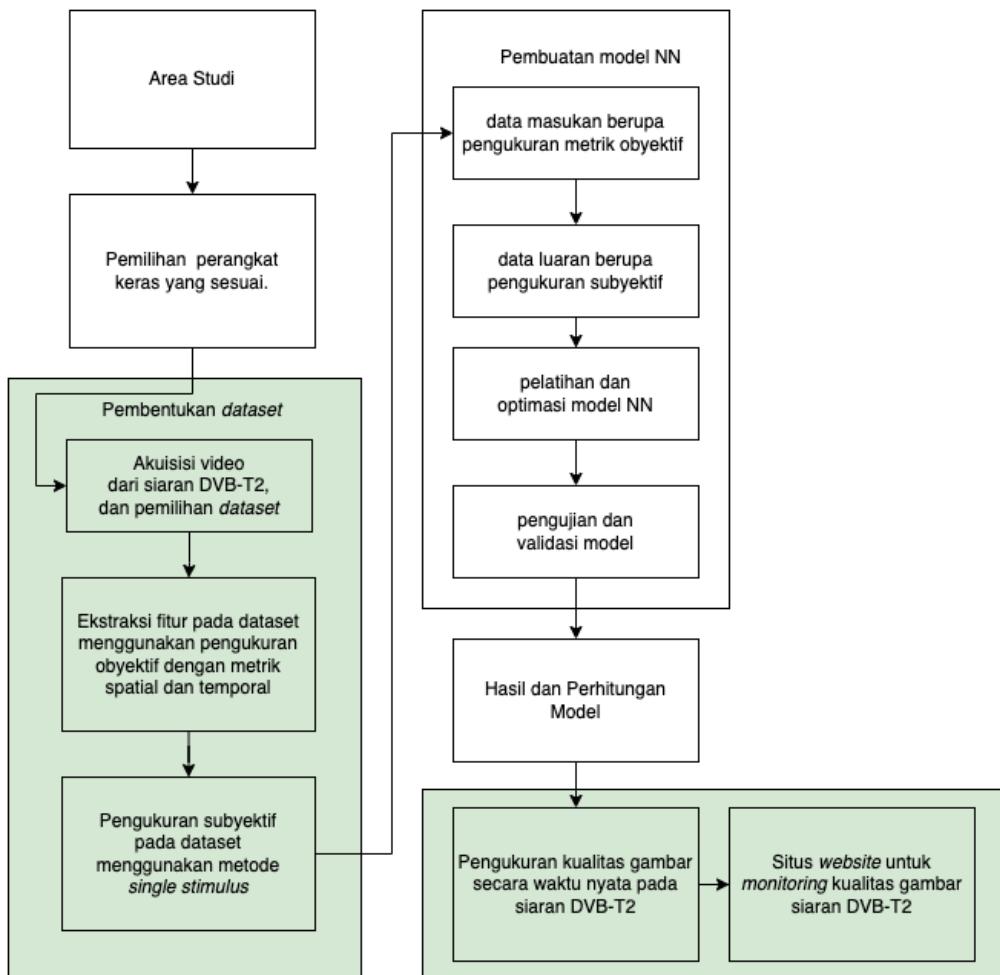
3.1 Tahapan Penelitian

Penelitian ini bertujuan untuk merancang dan membuat suatu sistem yang dapat memantau kualitas video dari siaran televisi digital DVB-T2. Rencana penelitian dilakukan dengan menggabungkan dan mengembangkan dari algoritma yang sudah dilakukan oleh peneliti terdahulu. Seperti yang ditunjukkan pada gambar 3.1, penelitian ini diawali dari area studi yang merupakan gambar dari siaran televisi digital DVB-T2 yang ada di Indonesia, khususnya wilayah Jabodetabek. Gambar diperoleh langsung dari siaran DVB-T2 melalui perangkat komputer yang terhubung dengan *tv tunner*.

Pembentukan dataset diawali dari proses pengambilan gambar/video dengan kualitas FHD dari perangkat komputer. Tahap berikutnya adalah mempelajari parameter apa saja yang dapat mempengaruhi kualitas gambar dari siaran DVB-T2 dan mencari metrik pengukuran apa saja yang dapat digunakan. Pemilihan metrik diutamakan pada metrik pengukuran kualitas gambar tanpa referensi (*No-Reference, NR*) yang didapat dari peneliti terdahulu. Metrik tersebut selanjutkan diprogram kembali dengan menggunakan bahasa pemrograman *python* untuk mempermudah dalam pengaplikasiannya. Hasil dari pengukuran menggunakan metrik obyektif ini nantinya digunakan sebagai parameter masukan dari model *neural network (NN)*.

Tahapan pembentukan dataset berikutnya adalah dengan menguji sampel kepada beberapa responden untuk memperoleh nilai yang digunakan sebagai parameter pengukuran subjektif. Proses pengumpulan data subjektif

dilakukan dengan merancang suatu aplikasi web yang mengikuti standar ITU-R BT.500-14 tahun 2019 mengenai metodologi untuk pengukuran subjektif pada kualitas gambar televisi. Hasil dari pengukuran subyektif nantinya akan digunakan sebagai parameter luaran dari model NN yang dibuat di tahapan selanjutnya.



Gambar 3.1: Tahapan penelitian yang dilakukan

Tahap berikutnya adalah dengan membuat model NN dari data pengukuran metrik objektif sebagai parameter masukan, dan data pengukuran subjektif sebagai parameter luaran. Model NN yang dibuat dilakukan dengan melakukan pengujian dari beberapa NN yang ada hingga memperoleh hasil yang sesuai dengan hasil pengukuran subjektif. Proses pembuatan model di-

lakukan dengan menggunakan Super Komputer DGX A100 milik Universitas Gunadarma. Selanjutnya dilakukan pengukuran akurasi dan korelasi untuk menvalidasi antara hasil dari model dibandingkan dengan pengukuran subjektif. Tahapan terakhir adalah mengimplementasikan hasil pengukuran dan model NN ke dalam perangkat *edge computing*. Perangkat yang digunakan adalah Raspberry Pi 4 dengan tambahan modul TV *Hat*. Pengukuran kualitas video dapat dilakukan secara waktunya pada siaran DVB-T2. Hasil dari pengukuran selanjutnya ditampilkan pada *website* pemantauan.

3.2 Area Studi

Penelitian ini dilakukan di Indonesia lebih tepatnya di Wilayah Kota Depok, Jawa Barat, berbatasan dengan Kota Madya Jakarta Selatan. Proses pengambilan data video dan pengukuran kualitas gambarnya yang diperoleh dari siaran DVB-T2 dilakukan di lokasi Kampus F8, Universitas Gunadarma. Video diambil dengan menggunakan perangkat komputer yang terhubung secara langsung ke siaran DVB-T2 menggunakan *tv tunner* dan juga antena.

3.2.1 Obyek Penelitian

Obyek penelitian ini adalah *frame* video definisi tinggi penuh (*Full High Definition, FHD*) yang diperoleh dari siaran televisi DVB-T2. FHD memiliki resolusi gambar 1080p atau 1920×1080 piksel. Pada penelitian ini durasi pengambilan sampel video sampel pada siaran DVB-T2 hanya 10 detik untuk setiap samplenya, mengikuti standar pengukuran kualitas gambar ITU-R BT.500. Video diambil dari beberapa layanan televisi lokal yang ada di Indonesia, khususnya di wilayah studi. Beberapa layanan tv digital yang didapat seperti Metro TV, Net TV, TVRI World, TVRI, Sport, dan MyTV. Layanan yang dipilih merupakan layanan yang sudah mendukung resolusi video FHD.

Pada gambar 3.2 merupakan hasil gambar FHD yang diperoleh dari siaran DVB-T2 menggunakan perangkat komputer.



Gambar 3.2: Contoh Gambar FHD dari layanan TVRI World

Proses pengambilan video membutuhkan beberapa perangkat keras dan perangkat lunak. Perangkat keras yang digunakan adalah perangkat komputer, raspberry pi, *tv tunner*, antena dan monitor. Perangkat lunak yang digunakan adalah *tvheadend*, *visual studio code*, *jupyterlab*, dan bahasa pemrograman *python* beserta librari pendukungnya.

3.2.2 Pemilihan Perangkat Keras

Penelitian ini memiliki beberapa tahapan seperti yang dijelaskan pada gambar 3.1, sehingga memerlukan perangkat yang mendukung dimulai dari proses pengambilan data, pengukuran obyektif dan subyektif, hingga memperoleh model, dan hasil. Perangkat keras yang digunakan juga tidak hanya satu jenis, dikarena setiap proses pada tahapan penelitian, menggunakan spesifikasi perangkat keras yang berbeda. Proses pengambilan data dan pengujian hasil dilakukan pada perangkat *edge computing* yang terhubung langsung dengan

siaran DVB-T2. Sedangkan pada proses pengukuran dan pembuatan model untuk jaringan saraf tiruan (NN) menggunakan perangkat komputer atau cloud computing. Perangkat keras yang digunakan diantaranya:

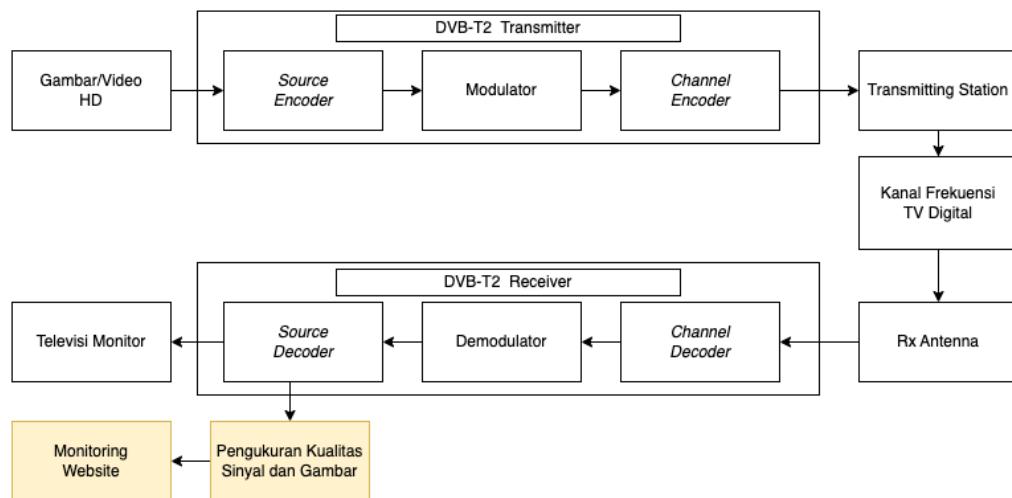
- Perangkat *Edge Computing* untuk pengambilan data dan juga pengukuran secara waktu nyata menggunakan Raspberry Pi 4 model B dengan prosesor Cortex-A72 ARM v8 64-bit up to 1.5 Mhz, VideoCore VI support OpenGL 3.0, dan RAM LPDDR4 8GB.
- Perangkat *tv-tunner* untuk *edge computing* yang digunakan adalah raspberry pi TV HAT (*Hardware Attach on Top*) dengan Tuner Sony CXD2880, mendukung DVB-T2 dengan frekuensi pembawa VHF dan UHV, mendukung format video MPEG-2 dan H.264, yang umum digunakan dalam siaran televisi digital.
- Perangkat pengukuran video subyektif yang digunakan adalah Macbook 2019, prosesor intel hexa-core i7 Gen-9, GPU AMD Radeon Pro 555X, RAM DDR4 16GB, layar 15.4 inci resolusi 2880×1800 .
- Perangkat untuk pengukuran dan membuat model *neural network* menggunakan *Super Computer* NVIDIA DGX A100 dengan virtual machine yang digunakan RAM 20GB dan GPU 40GB

Pemilihan perangkat komputer yang digunakan sebagai *edge computing* berdasarkan pada kemampuan untuk dapat dapat terhubung langsung dengan modul TV Hat yang bisa menerima siaran televisi digital DVB-T2. Sedangkan perangkat yang digunakan dalam membuat model NN berdasarkan pada kecepatan memproses data dan juga merupakan salah satu fasilitas dari Universitas Gunadarma. Kedua perangkat tersebut menggunakan versi perangkat lunak yang sama untuk memudahkan proses duplikasi program.

Tabel 3.1: Versi bahasa pemrograman python dan librariinya

Nama Library	python	opencv	matplotlib	numpy	tensorflow
Versi	3.8.5	4.6.0	3.2.2	1.21.6	2.11.0

Fokus dari penelitian terdapat pada sisi penerima siaran televisi digital. Siaran DVB-T2 yang diterima dari antenna menuju *receiver* dan mengalami beberapa tahapan *decoder* untuk memperoleh video dan audio. Frame gambar dari video hasil akhir proses tersebut yang akan diukur dan didapat nilai kualitasnya berdasarkan beberapa parameter.

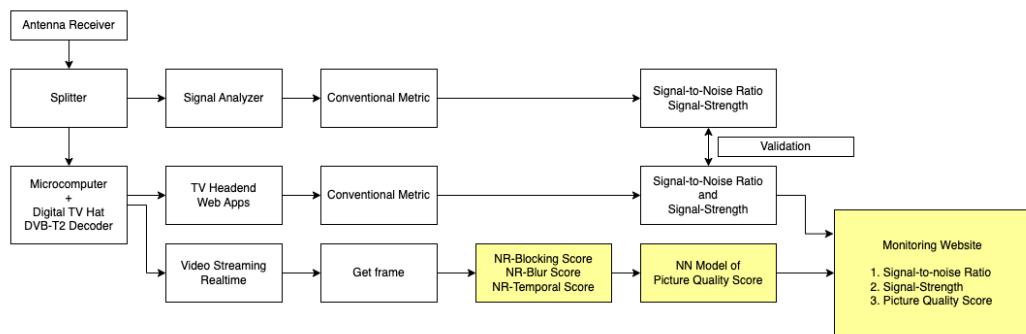


Gambar 3.3: Fokus Penelitian

Luaran akhir dari penelitian ini adalah mendapat model yang sesuai untuk dapat melakukan pengukuran kualitas gambar pada siaran DVB-T2. Pengukuran juga dilakukan terhadap parameter sinyal dari siaran yang sedang dilakukan pengukuran gambarnya. Hasil dari pengukuran akan secara waktu nyata ditampilkan pada *website* dan tersimpan secara berkala.

3.3 Pembentukan *Dataset*

Dalam mencapai tujuan dari penelitian maka dibuat sebuah rancangan dari sistem secara keseluruhan. Sistem terdiri dari beberapa perangkat keras dan lunak yang memiliki tugas dan fungsi masing-masing yang saling terhubung. Keterbaruan dari penelitian ini juga adalah sistem secara keseluruhan yang ditunjukkan pada gambar 3.4 dan NN untuk dalam proses pengukuran kualitas gambar secara waktunya.



Gambar 3.4: Rancangan Sistem Pengukuran Kualitas Gambar Siaran DVB-T2

Sistem diawali dari pengambilan sinyal siaran tv digital menggunakan Antenna yang kemudian di split ke dua alat yaitu yang pertama adalah mikro komputer *Raspberry Pi* dengan *tv tunner*, kedua *signal analyzer*. Aplikasi yang digunakan pada raspberry pi adalah *tvheadend* yang berfungsi menampilkan siaran tv digital pada perangkat *raspberry pi* dan menampilkan status pengukuran sinyalnya. Spectrum analyzer digunakan hanya untuk melakukan validasi pada parameter sinyal siaran tv digital. Selanjutnya dengan menggunakan aplikasi *ffmpeg* dan pemrograman python setiap video yang sedang ditayangkan dapat diambil frame gambarnya. Frame yang diambil diukur menggunakan obyektif metriks untuk memperoleh parameter input dari kualitas gambar frame tersebut. Hasil pengukuran berupa angka yang nantinya akan dimasukkan ke dalam model NN untuk memperoleh nilai kualitas yang

sudah disesuaikan dengan penilaian subyektif manusia. Hasil akhir pengukuran sinyal dan juga kualitas gambar ditampilkan dalam sebuah *website*.

3.3.1 Akuisisi *Dataset* Video

Video yang dijadikan dataset adalah video yang diperoleh dari siaran DVB-T2 secara langsung dengan resolusi HD dan FHD, yaitu resolusi 1280x720 piksel (720p) dan 1920x1080 piksel (1080p). Proses pengambilan video dilakukan dengan menggunakan perangkat keras *Raspberry Pi 4* dengan modul *TV HAT*. Kemudian perangkat tersebut harus terhubung dengan antena televisi agar dapat menerima siaran DVB-T2. Antena yang digunakan pada penelitian ini adalah PX HDA-5000.

Perangkat lunak yang digunakan dalam proses pengambilan data video adalah *tvheadend*. Aplikasi tersebut dapat menampilkan siaran televisi digital dari tv *tunner* ke perangkat komputer. Aplikasi *tvheadend* kemudian diatur untuk melakukan perekaman video HD dari siaran televisi digital. Beberapa layanan televisi yang sudah HD diantaranya TVRI HD, TVRI World, Net TV HD dan Metro TV HD. Dalam pengambilan dataset, diambil sejumlah 100 video dari masing-masing layanan tersebut dengan durasi per video adalah 10 detik.

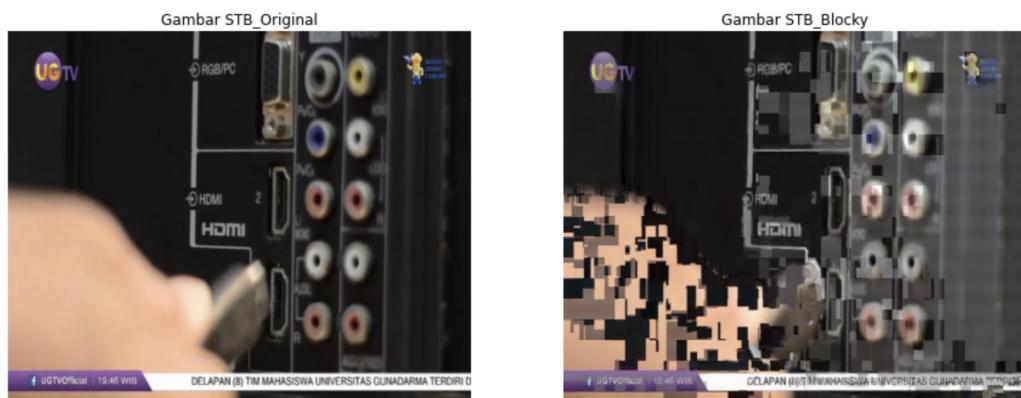
3.3.2 Metrik Obyektif Pengukuran Kualitas Gambar

Metrik obyektif merupakan suatu ukuran atau parameter yang dapat diukur secara kuantitatif. Pada penelitian digunakan beberapa metrik obyektif untuk melakukan pengukuran pada kualitas video. Metrik obyektif yang digunakan diantarauntuk pengukuran spatial pada gambar seperti blocking dan blur, ada juga metrik yang digunakan untuk melakukan pengukuran temporal terhadap frame yang terdapat pada video. Metrik obyektif juga digunakan untuk melakukan pemilihan dataset dan sebagai parameter masukan yang

nantinya digunakan dalam NN.

3.3.2.1 Metrik Bloking

Blocking pada gambar adalah efek visual yang terjadi ketika akibat kompresi data yang dilakukan pada gambar atau video dengan mengurangi jumlah bit yang digunakan untuk merepresentasikan piksel di dalamnya. Saat terjadi kerusakan pada bit data tersebut dan gambar didekompresi kembali, maka akan muncul kerusakan seperti munculnya kotak-kotak besar yang terlihat di dalam gambar, yang disebut sebagai *bloking*. Gambar dengan kondisi *blocking* dapat dilihat pada gambar 3.5.



Gambar 3.5: Perbandingan gambar normal dan *blocking*

Pada penelitian ini digunakan salah satu algoritma pengukuran tingkat *blocking* pada gambar yang dikemukakan oleh *Zhou Wang*. Teknik pengkodean gambar yang membagi gambar menjadi sub-blok 8x8 piksel. Proses transformasi dan kuantisasi kemudian diterapkan pada masing-masing sub-blok secara individual dan independen. Teknik pengkodean gambar ini umumnya digunakan dalam teknologi kompresi gambar, seperti pada standar kompresi JPEG, dan dapat menyebabkan munculnya artefak blok atau blocking artifact pada gambar. Sehingga pengukuran *bloking* pada gambar dilakukan setiap 8 piksel secara horizontal dan vertikal.

$$\begin{aligned} \text{image_resolution} &= x(m, n) \\ \text{dimana, } m &\in [1, M], \quad n \in [1, N] \end{aligned} \tag{3.1}$$

Gambar diubah ke dalam skala keabuan (grayscale) untuk dapat memperoleh satu saluran warna dengan jumlah m baris dan n kolom. Kemudian dilakukan pengukuran secara terpisah antara baris dan kolom. Ide dasar dari algoritma ini adalah untuk mendeteksi sinyal blok dan memperkirakan dayanya dengan menggunakan asumsi bahwa gambar yang berblok adalah gambar yang tidak berblok yang terganggu oleh sinyal blok ideal. Algoritma ini dimulai dengan menghitung selisih antara piksel-piksel sepanjang baris untuk mendapatkan selisih horizontal d_h dan sepanjang kolom untuk mendapatkan selisih vertikal d_v .

$$d_h(m, n) = x(m, n + 1) - x(m, n), \quad n \in [1, N - 1] \tag{3.2}$$

$$d_v(m, n) = x(m + 1, n) - x(m, n), \quad n \in [1, M - 1] \tag{3.3}$$

Setelah selisih-selisih ini dihitung, gambar selisih dapat diperoleh. Selanjutnya, keberblokan dihitung dengan mengambil rata-rata selisih di sepanjang batas blok. Untuk gambar berukuran $M \times N$, pengukuran horizontal didefinisikan sebagai persamaan berikut:

$$B_h = \frac{1}{M(\lfloor N/8 \rfloor - 1)} \sum_{i=1}^M \sum_{j=1}^{\lfloor N/8 \rfloor - 1} |d_h(i, 8j)| \tag{3.4}$$

Selanjutkan dilakukan pengukuran tingkat aktifitas sinyal. Aktivitas sinyal gambar mengacu pada ukuran intensitas sinyal atau informasi yang

terkandung dalam gambar. Semakin tinggi aktivitas sinyal, semakin banyak detail yang terdapat dalam gambar dan semakin sedikit pengaburan atau keberblokan yang terlihat. Pengukuran aktivitas sinyal dapat membantu dalam mengidentifikasi pengaburan atau keberblokan pada gambar. Pada metode ini dilakukan pengukuran aktifitas sinyal gambar meliputi rata-rata selisih absolut (A) antar sampel gambar di dalam blok dan tingkat *zero-crossing* (Z) (Wang et al. 2002).

$$A_h = \frac{1}{7} \left[\frac{8}{M(N-1)} \sum_{i=1}^M \sum_{j=1}^{N-1} |d_h(i,j)| - B_h \right] \quad (3.5)$$

Pengukuran aktifitas gambar kedua adalah menghitung rata-rata dari *zero-crossing*. Zero crossing adalah fenomena di mana sinyal atau gelombang melintasi sumbu nol. Hal ini dapat dihitung dengan menghitung jumlah kali di mana sinyal melintasi sumbu nol dalam interval waktu tertentu. Semakin banyak tingkat nol-crossing yang terjadi, semakin tinggi aktivitas sinyal dan semakin banyak detail yang terdapat dalam gambar atau sinyal tersebut.

$$z_h(m, n) = \begin{cases} 1 & \text{horizontal } ZC \text{ pada } d_h(m, n) \\ 0 & \text{selain itu} \end{cases} \quad (3.6)$$

$$Z_h = \frac{1}{M(N-2)} \sum_{i=1}^M \sum_{j=1}^{N-2} d_h(i,j) \quad (3.7)$$

Metode yang sama dilakukan pada pengukuran secara vertikal pada B_v , A_v , dan Z_v . Kemudian, rata-rata blockiness, B, rata-rata perbedaan absolut, A, dan rata-rata tingkat *zero crossing*, Z, diperoleh dengan menggunakan pengukuran vertikal dan horizontal yang dijumlahkan dan dibagi dua.

$$B = \frac{B_h + B_v}{2}, A = \frac{A_h + A_v}{2}, B = \frac{Z_h + Z_v}{2} \quad (3.8)$$

Ukuran blockiness akhir, S, kemudian dapat diperoleh menggunakan rumus:

$$S = \alpha + \beta B^{\gamma_1} A^{\gamma_2} B^{\gamma_3} \quad (3.9)$$

Dimana $\alpha, \beta, \gamma_1, \gamma_2$, dan γ_3 , mewakili parameter prediksi yang diperoleh melalui fitting kurva data eksperimen subjektif yang dilakukan oleh Wang. Kemudian berdasarkan algoritma diatas maka dibuatlah program ke dalam bahasa pemrograman python.

3.3.2.2 Metrik Blur

Blur pada gambar adalah kondisi di mana gambar menjadi kurang tajam dan terlihat kabur seperti yang ditunjukkan pada gambar 3.6. Hal ini disebabkan oleh hilangnya komponen frekuensi tinggi pada gambar, yang dapat terjadi karena berbagai faktor seperti pergerakan kamera saat pengambilan gambar atau ketidakfokusan saat pemotretan. Gambar bisa menjadi blur juga karena transmisi, yaitu ketika sinyal gambar dikirimkan melalui jaringan atau media transmisi, seperti kabel atau gelombang radio. Selama transmisi, sinyal gambar dapat terdistorsi atau terganggu oleh kebisingan dan interferensi. Hal ini dapat menyebabkan kehilangan atau penyimpangan data gambar, terutama pada detail halus dan frekuensi tinggi, yang akhirnya menyebabkan gambar tampak blur atau kabur. Kualitas transmisi bergantung pada faktor-faktor seperti kecepatan transfer data, kualitas sinyal, dan jarak antara pemancar dan penerima.



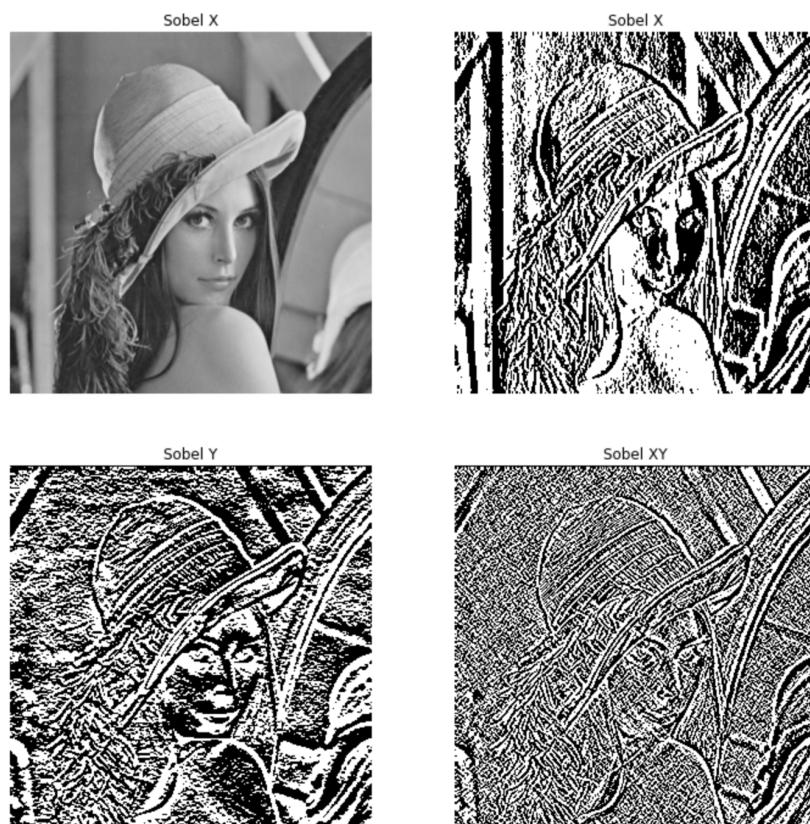
Gambar 3.6: Perbandingan Gambar Blur dan tidak, dari kiri ke kanan: Tumbling Es, Lena, Street View (Xu et al. 2021)

Gejala blur pada gambar dapat diidentifikasi dengan melihat kehalusan tepi atau kekurangan detail halus pada gambar. Terdapat beberapa teknik yang dapat digunakan untuk mengukur tingkat blur pada gambar, salah satunya adalah dengan menggunakan deteksi tepi. Metode ini didasarkan pada observasi bahwa gambar yang kabur memiliki tepian yang kurang tajam atau tidak jelas, karena kehilangan komponen frekuensi tinggi. Oleh karena itu, dengan mendeteksi tepi pada gambar, kita dapat mengukur tingkat kejelasan tepi yang terdapat pada gambar dan menggunakan nilai ini untuk mengindikasikan tingkat blur pada gambar tersebut (Ferzli & Karam 2009).

Deteksi tepi dapat dilakukan dengan menggunakan algoritma Sobel. Dalam metode ini, gambar diubah menjadi citra grayscale dan dilakukan operasi perataan pada gambar dengan kernel tertentu untuk menghilangkan noise.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & +2 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.10)$$

Selanjutnya, operasi deteksi tepi dilakukan dengan menghitung turunan parsial gambar pada arah vertikal dan horizontal. Hasil dari kedua arah ini kemudian digabungkan dengan menghitung magnitude gradien pada setiap titik pada gambar seperti yang terlihat pada gambar 3.7. Hasil akhir yang dihasilkan adalah citra biner dengan nilai 0 dan 1, yang menunjukkan lokasi tepi pada gambar.

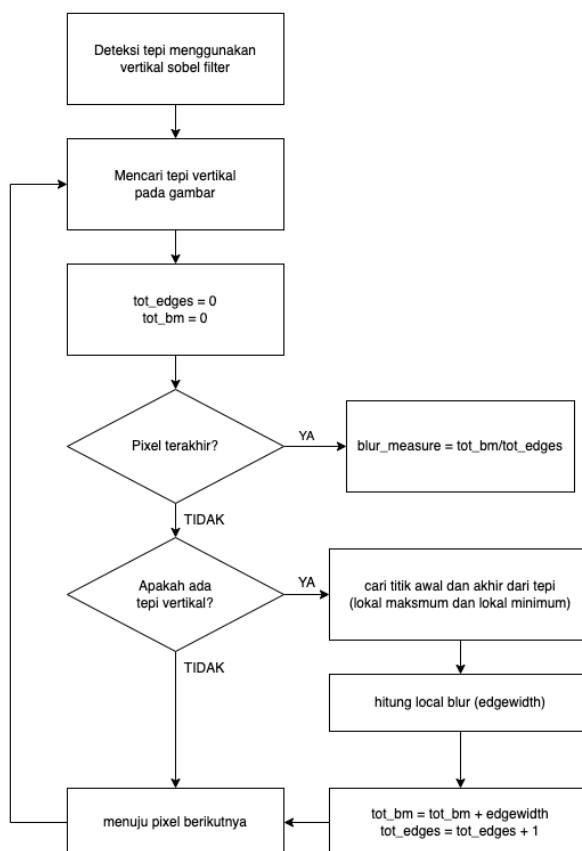


Gambar 3.7: Hasil Transformasi ke bentuk deteksi tepi menggunakan sobal pada gambar Lenna

Dalam deteksi tepi, local minimum dan local maximum mengacu pada titik-titik di mana perubahan intensitas mencapai minimum atau maksimum dalam sekelompok piksel pada gambar. Dalam konteks algoritma Marziliano, local minimum dan local maximum digunakan untuk menentukan spread dari tepi vertikal pada gambar yang kabur. Local minimum dan local maximum pada tepi vertikal dihitung pada setiap baris gambar, dan jarak antara ke-

dua titik tersebut memberikan informasi tentang tingkat blur pada gambar (Marziliano et al. 2002).

Detailnya jika tepi ditemukan, maka lokal ekstremum (maksimum lokal dan minimum lokal) dicari ke arah kiri dan kanan dari piksel tepi. Untuk mencari jarak atau panjang tepi, selisih antara maksimum lokal dan minimum lokal dihitung. Hasilnya diidentifikasi sebagai ukuran blur lokal untuk lokasi tepi saat ini. Akhirnya, ukuran blur global dihitung dengan menghitung rata-rata *blur* lokal di seluruh lokasi tepi. Algoritma dari teknik pengukuran blur ini ditunjukkan pada Gambar ??.



Gambar 3.8: Diagram alur algoritma pengukuran blur pada gambar (Marziliano et al. 2002)

Metrik pengukuran *blur* tersebut digunakan pada penelitian ini sebagai salah satu parameter masukan untuk pembuatan model NN. Metrik tersebut

ditulis kembali menggunakan bahas apemprograman python untuk mempermudah proses pengukuran yang dilakukan pada hardware yang digunakan.

3.3.2.3 Metrik Temporal

Metrik temporal merujuk pada metode pengukuran kualitas video yang fokus pada perubahan dari waktu ke waktu. Metrik ini digunakan untuk mengukur bagaimana perubahan frame dari video yang dihasilkan mempengaruhi kualitas keseluruhan dari video tersebut. Beberapa faktor yang dapat mempengaruhi metrik temporal meliputi jumlah frame yang hilang atau rusak, kualitas gambar pada setiap frame, dan kehalusan perpindahan antar frame. Metrik temporal dapat digunakan untuk mengukur kualitas video pada berbagai jenis aplikasi, termasuk video streaming dan komunikasi video.

Pendekatan umum untuk mendeteksi frame yang beku adalah dengan menghitung mean-squared error (MSE) antara frame saat ini dan frame sebelumnya, dan mempertimbangkan frame saat ini sebagai frame yang beku jika $MSE=0$. Pertama, video input dikonversi ke ruang warna YUV. Frame yang potensial beku kemudian diidentifikasi berdasarkan MSE antara frame saat ini dan frame sebelumnya:

$$YM_1(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (Y(x, y, i) - Y(x, y, i-1))^2 \quad (3.11)$$

$$UM_1(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (U(x, y, i) - U(x, y, i-1))^2 \quad (3.12)$$

$$VM_1(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (V(x, y, i) - V(x, y, i-1))^2 \quad (3.13)$$

Dimana:

Y : saluran warna *luminance*

U : sarulan warna *chrominance* hijau ke merah

V : sarulan warna *chrominance* biru ke kuning

M : variabel pengukuran antar frame

W : lebar resolusi gambar

H : panjang resolusi gambar

x : piksel baris pada gambar

y : piksel kolom pada gambar

i : jumlah frame dalam video ($i = 2, \dots, n$)

Konten dengan gerakan sangat rendah dapat menghasilkan MSE yang sangat kecil antara *frame* yang berurutan meskipun *frame* ini tidak beku (*freeze*). Untuk membatasi jumlah positif palsu, mekanisme berikut diterapkan: jika sebuah *frame* berpotensi dibekukan maka *frame* tersebut juga diperiksa terhadap *frame* pertama dari peristiwa pembekuan (kecuali jika *frame* saat ini juga merupakan *frame* pertama dari peristiwa pembekuan). $YM_2(i)$, $UM_2(i)$ dan $VM_2(i)$ menunjuk MSE (masing-masing dalam bidang warna Y, U dan V) antara *frame* saat ini dan *frame* pertama dari peristiwa pembekuan(Huynh-Thu & Gha 2009).

$$YM_2(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (Y(x, y, i) - Y(x, y, i - k))^2 \quad (3.14)$$

$$UM_2(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (U(x, y, i) - U(x, y, i - k))^2 \quad (3.15)$$

$$VM_2(i) = \frac{1}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (V(x, y, i) - V(x, y, i - k))^2 \quad (3.16)$$

di mana $(i - k)$ adalah indeks temporal dari *frame* pertama dari peristiwa pembekuan di mana *frame* i berada. Sebuah *frame* i akan ditandai sebagai beku jika MSE antara *frame* sekarang dan *frame* sebelumnya tetapi juga antara *frame* ini dan *frame* pertama dari *event freeze* berada dibawah *threshold*

T yang sangat kecil dimana *threshold* T ditetapkan ke 1 (Huynh-Thu & Gha 2009).

$$FreezeFlag = \begin{cases} 1 & if(YM, UM, VM)_{1,2}(1) < T \\ 0 & lainnya \end{cases} \quad (3.17)$$

Selanjutnya, nilai durasi freeze dijumlahkan untuk memperolah nilai seberapa besar freeze yang terjadi dalam interveal waktu tertentu. Metrik untuk mendekripsi kebekuan pada video ini juga dituliskan dalam bahasa pemrograman phthon sebagai parameter masukan.

3.3.3 Pengukuran Subyektif Kualitas Gambar

Subyektif assessment adalah metode pengukuran yang melibatkan partisipasi manusia untuk memberikan penilaian kualitatif terhadap suatu produk atau layanan berdasarkan pengalaman mereka secara langsung. Dalam konteks pengolahan citra atau video, subyektif assessment dapat digunakan untuk mengukur kualitas visual dari hasil pengolahan tersebut berdasarkan persepsi pengguna. Pada penelitian ini digunakan standar pengukuran subyektif berdasarkan rekomendasi ITU-R BT.500-14 yang diterbitkan Oktober 2019. Standar tersebut menyediakan pedoman dan prosedur untuk penilaian subyektif kualitas gambar televisi (ITU-R BT500 2019).

3.3.3.1 Single Stimulus Standar ITU-R 500-14

Single stimulus methods (SS) dipilih sebagai salah satu metode subjective assessment pada ITU-R BT.500-14 karena lebih mudah dilakukan dan lebih efisien dalam hal waktu. Metode ini hanya memerlukan satu gambar atau video per *sample* sebagai stimulus, sehingga tidak memerlukan waktu yang lama dalam proses evaluasi. Selain itu, metode ini juga tidak memerlukan gambar referensi atau acuan, sehingga lebih fleksibel dan dapat digunakan

dalam berbagai kondisi yang berbeda. Meskipun demikian, single stimulus methods juga memiliki kekurangan yaitu rentan terhadap bias pengamat, sehingga perlu dilakukan pengujian ulang dan dilakukan dengan panelis yang berbeda untuk menghasilkan hasil yang lebih akurat dan konsisten.

Metode Single Stimulus dapat digunakan untuk mengevaluasi kualitas video dan gambar. Namun, karena video adalah sekuens gambar yang saling terkait, maka pada pengujian video, stimulus yang diberikan pada setiap pengujian harus sesuai dengan konteks dari adegan yang dipresentasikan. Hal ini bisa dilakukan dengan memilih frame yang mewakili adegan atau dengan memilih klip video pendek yang mewakili adegan tersebut (Seshadrinathan et al. 2010).

Terdapat beberapa persyaratan pengukuran Single Stimulus (SS) pada ITU-R BT.500-14 yang terdiri dari beberapa hal, antara lain:

1. Pemilihan materi uji yang representatif dan bervariasi, serta disajikan dalam kondisi yang sama untuk semua panelis.
2. Jumlah panelis minimal 15 orang, yang diwajibkan untuk memenuhi syarat tertentu, seperti usia, pengalaman dalam menonton televisi, kesehatan mata, dan tidak memiliki kecacatan warna.
3. Prosedur pelatihan yang ketat untuk panelis dalam memberikan penilaian yang konsisten dan obyektif.
4. Penggunaan skala pengukuran yang sesuai, yaitu skala likert yang terdiri dari 5 hingga 7 pilihan untuk menilai kualitas gambar.
5. Pelaksanaan pengujian dan pengolahan data dengan standar yang ketat dan terdokumentasi dengan baik.

Selain dari faktor pemilihan panelis dan materi uji, standar ITU-R BT.500-14 memberikan panduan yang mencakup beberapa faktor, seperti ca-

haya lingkungan, warna dinding, dan pencahayaan ruangan yang diperlukan untuk memastikan konsistensi dan validitas penilaian subjektif. Selain itu, panduan ini juga menyarankan penggunaan layar televisi yang sesuai dengan kondisi pengujian dan memberikan instruksi tentang posisi penonton dan jarak pandang yang optimal. Tujuannya adalah untuk meminimalkan faktor-faktor yang dapat memengaruhi hasil penilaian subjektif dan memastikan bahwa penilaian dilakukan dalam kondisi yang optimal untuk memberikan hasil yang akurat dan dapat diandalkan. Panduan ini sangat penting untuk memastikan bahwa hasil penilaian subjektif kualitas gambar televisi dapat dipertanggung-jawabkan dan diandalkan untuk digunakan dalam pengembangan produk dan teknologi baru dalam industri televisi.

Pada pengujinya dipilih standar *home environment* dengan alasan konteks penggunaan yang lebih realistik dibandingkan dengan di lab, biaya pengujian yang lebih renda, dan waktu yang lebih fleksibel disesuaikan dengan panelis. Namun tetap terdapat beberapa syarat untuk pengujian SS dengan *home environment* yang ditunjukkan pada tabel??.

Tabel 3.2: Standar pengujian SS dengan *home-environment*

Penerangan sekitar pada layar (cahaya insiden dari lingkungan yang jatuh pada layar harus diukur secara tegak lurus ke layar)	200 lux
Pencahayaan puncak dari layar (peak luminance)	70-500 lux
Rasio pencahayaan antara kondisi layar tidak aktif dan kondisi layar puncak	≤ 0.02
Resolusi gambar/video	1920×1080
Aspect rasio	16:9
Sudut lihat optimal dari layar	31°
Jarak lihat optimal dari layar (H = lebar gambar di layar)	3.2 H
Durasi keseluruhan pengujian	≤ 30 menit

Pada umumnya durasi klip yang digunakan berkisar antara 5 hingga 10 detik. Durasi ini dianggap cukup untuk menampilkan adegan yang mewakili isi dari keseluruhan konten. Selain itu, durasi klip yang terlalu lama dapat membuat panelis cepat lelah dan mempengaruhi konsistensi hasil penilaian. Oleh karena itu, durasi klip perlu dipertimbangkan secara cermat untuk memastikan keakuratan dan konsistensi dari hasil penilaian (ITU-R BT500 2019).

3.3.3.2 Tahap 1 - Pemilihan materi dan luaran

Sebelum memulai pengujian subyektif dipilih terlebih dahulu materi/video yang akan dinilai. Video dipilih dari dataset yang sudah diambil dari siaran DVB-T2. Jumlah materi/video klip yang diujikan sebanyak 105 video

FHD dengan durasi per video 10 detik. Perangkat yang digunakan dalam proses pengujian adalah Macbook Pro 2019 dengan ukuran layar 16 inci. Penilaian pada video uji menggunakan skala kualitas gambar ITU-R yang ditunjukkan pada tabel 3.3.

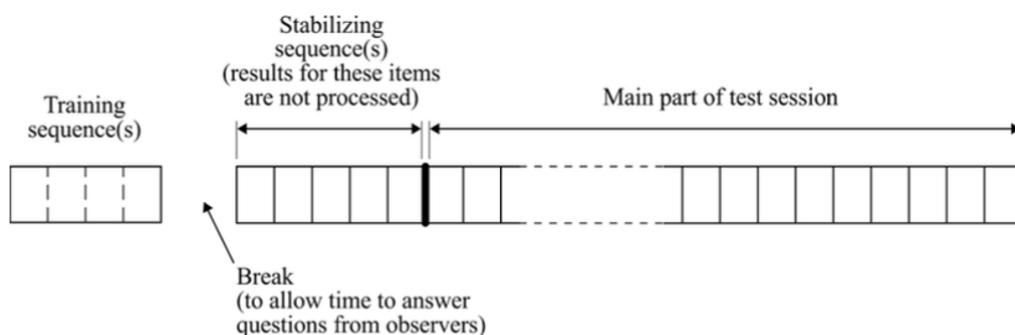
Tabel 3.3: Skala ITU-R Quality dan Impairment

Skala lima kelas	
Quality	Impairment
5 Excellent	5 Imperceptible
4 Good	4 Perceptible, but not annoying
3 Fair	3 Slightly annoying
2 Poor	2 Annoying
1 Bad	1 Very annoying

Luaran dari subyektif assessment pada gambar siaran televisi berupa skala pengukuran. Terdapat dua jenis skala pengukuran, yaitu skala kualitas dan skala *impairment*. Skala kualitas digunakan untuk mengukur sejauh mana penonton merasakan kualitas gambar atau video yang ditampilkan. Skala ini biasanya berbentuk angka atau kata-kata yang merepresentasikan level kualitas. Responden diminta untuk memilih nilai yang paling sesuai dengan pengalaman menonton mereka. Sedangkan skala impairment digunakan untuk mengukur sejauh mana gangguan pada gambar atau video memengaruhi kualitas tampilan secara keseluruhan. Responden diminta untuk memilih nilai yang paling sesuai dengan tingkat gangguan pada gambar atau video. Dalam beberapa kasus, skala kualitas dan skala *impairment* digunakan bersamaan untuk memberikan gambaran yang lebih lengkap tentang kualitas gambar atau video yang ditampilkan (Mittal et al. 2012).

3.3.3.3 Tahap 2 - Sesi tes

Terdapat dua bagian pada tes seperti yang ditunjukkan pada gambar 3.9, yang pertama adalah urutan latihan atau *training sequence*, dan selanjutnya sesi tes (test session). *Training sequence* atau urutan latihan berupa serangkaian langkah atau instruksi yang diberikan kepada panelis dalam rangka melatih mereka untuk melakukan penilaian subyektif dengan konsisten dan akurat. *Training sequence* biasanya mencakup langkah-langkah seperti familiarisasi dengan peralatan pengujian, instruksi tentang melaksanakan tugas penilaian, dan latihan tentang mengenali dan mengevaluasi aspek-aspek kualitas gambar yang berbeda. Langkah ini penting untuk memastikan bahwa panelis memahami dan menggunakan skala penilaian dengan konsisten dan bahwa hasil penilaian subyektif akurat dan dapat diandalkan.

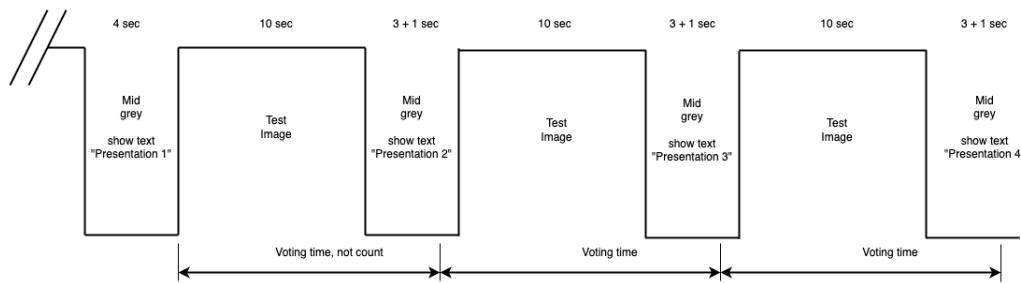


Gambar 3.9: Struktur presentasi dari subyektif tes

Pada awal sesi pertama, sekitar lima "presentasi *dummy*" harus diperkenalkan untuk menstabilkan pendapat para pengamat. Data yang dihasilkan dari presentasi ini tidak boleh dipertimbangkan dalam hasil pengujian. Selanjutnya secara langsung tanpa sepengetahuan pengamat dilakukan bagian utama dari sesi tes. Setiap presentasi diawali dan diakhiri dengan *midgrey* yang diantara berupa materi uji. *Midgrey* merupakan citra dengan nilai keabuan antara putih dan hitam dengan nilai RGB (128,128,128) atau HEX #808080. Mid-grey biasanya digunakan sebagai referensi dalam pengujian

subyektif untuk mengevaluasi kualitas kontras, ketajaman, dan keseimbangan warna.

Durasi yang umumnya digunakan adalah sekitar 10 detik untuk materi uji dan 1 hingga 2 detik untuk *midgrey*. Durasi ini dapat disesuaikan tergantung pada kebutuhan pengujian dan karakteristik materi uji yang digunakan. Waktu antara *midgrey* dan materi uji pada penelitian adalah 1, 10, dan 3 detik seperti yang ditunjukkan pada gambar 3.10.



Gambar 3.10: Interval antara *midgrey* dan materi uji

Berdasarkan interval tersebut maka dapat dihitung jumlah waktu yang dibutuhkan untuk menyelesaikan rangkaian pengukuran subyektif. Jumlah materi yang digunakan sesi tes termasuk *dummy* sebanyak $100 + 5$ video klip. Waktu yang dibutuhkan dalam 1 video adalah 14 detik termasuk dengan *midgrey*. Sehingga waktu yang dibutuhkan sebanyak $105 \times 14 = 1470$ detik atau sekitar 24.5 menit. Jika ditambah dengan *training sequence* yang membutuhkan waktu 3-5 menit, maka total waktu yang dibutuhkan untuk melakukan rangkaian secara keseluruhan masih ≤ 30 menit.

3.3.3.4 Tahap 3 - Pengukuran hasil tes

Pengukuran hasil pengukuran subyektif *single stimulus* melibatkan sejumlah panelis yang menilai kualitas gambar dari materi uji pada layar monitor. Skor subyektif diberikan dan dianalisis untuk menentukan rata-rata skor, standar deviasi, dan interval kepercayaan. Materi uji terdiri dari beberapa

klip video yang dipilih secara acak dari berbagai jenis siaran televisi, yang ditampilkan dalam urutan acak. Panelis diberikan klip latihan sebelumnya untuk membiasakan diri dengan tampilan antarmuka dan jenis gambar yang akan dinilai. Skor diberikan pada skala kualitas ITU-R dengan 1 menunjukkan kualitas gambar yang sangat buruk dan 5 menunjukkan kualitas gambar yang sangat baik. Hasil pengukuran dapat digunakan untuk mengevaluasi kualitas gambar pada siaran televisi dengan menghitung skor rata-rata, standar deviasi, dan interval kepercayaan (ITU-R P.910 2022).

Pengukuran hasil yang pertama adalah mengukur rata-rata dari setiap materi yang dipresentasikan ke panelis, dengan persamaan berikut:

$$\bar{u}_{ijkr} = \frac{1}{N} \sum_{i=1}^N u_{ijkr} \quad (3.18)$$

dimana:

\bar{u}_{ijkr} : nilai dari panelis i untuk kondisi tes j, urutan ke k, pengulangan r
 N : jumlah panelis

Demikian pula, skor rata-rata keseluruhan, u_j dan u_k , dapat dihitung untuk setiap kondisi uji dan setiap urutan/gambar uji. Selanjutnya saat menyajikan hasil pengujian, semua skor rata-rata harus memiliki interval kepercayaan yang terkait yang berasal dari standar deviasi dan ukuran masing-masing sampel. Berdasarkan ITU-R BT.500-14 dianjurkan untuk menggunakan interval kepercayaan 95% yang dinyatakan sebagai berikut:

$$[\bar{u}_{jkr} - \delta_{jkr}, \bar{u}_{jkr} + \delta_{jkr}] \quad (3.19)$$

dimana:

$$\delta_{jkr} = 1.96 \frac{S_{jkr}}{N} \quad (3.20)$$

Standar deviasi untuk setiap presentasi S_{jkr} dapat dinyatakan dalam persamaan:

$$S_{jkr} = \sqrt{\sum_{i=1}^N \frac{(\bar{u}_{jkr} - \bar{u}_{ijk})^2}{(N - 1)}} \quad (3.21)$$

Dengan probabilitas 95%, nilai mutlak dari selisih antara skor rata-rata eksperimental dan skor rata-rata sebenarnya (untuk jumlah pengamat yang sangat tinggi) lebih kecil dari interval kepercayaan 95%, asalkan distribusi skor individual memenuhi persyaratan tertentu. Demikian pula, deviasi standar S_j dapat dihitung untuk setiap kondisi uji. Namun, perlu dicatat bahwa deviasi standar ini, dalam kasus penggunaan jumlah urutan/gambar uji yang sedikit, akan lebih dipengaruhi oleh perbedaan antara urutan/gambar uji yang digunakan daripada variasi antara penilai yang berpartisipasi dalam penilaian.

3.3.3.5 Desain Aplikasi Subyektif *Assessment*

Subyektif *assessment* yang dilakukan untuk pengukuran kualitas gambar menggunakan perangkat lunak berbasis *local website*. Perangkat lunak dirancang untuk memudahkan proses pengambilan data dan juga memudahkan panelis dalam melakukan penilaian. Pada awal tampilan terdapat pilihan antara *training sequence* dan sesi tes. Pada tampilan utama saat tes terdapat jendela pemutaran materi video dan juga tab penilaian seperti yang ditunjukkan pada gambar 3.11 rancangan aplikasi subyektif *assessment*. Rangkaian materi video dan *mid-grey* akan ditampilkan secara berkala sesuai dengan skema yang sudah direncanakan.



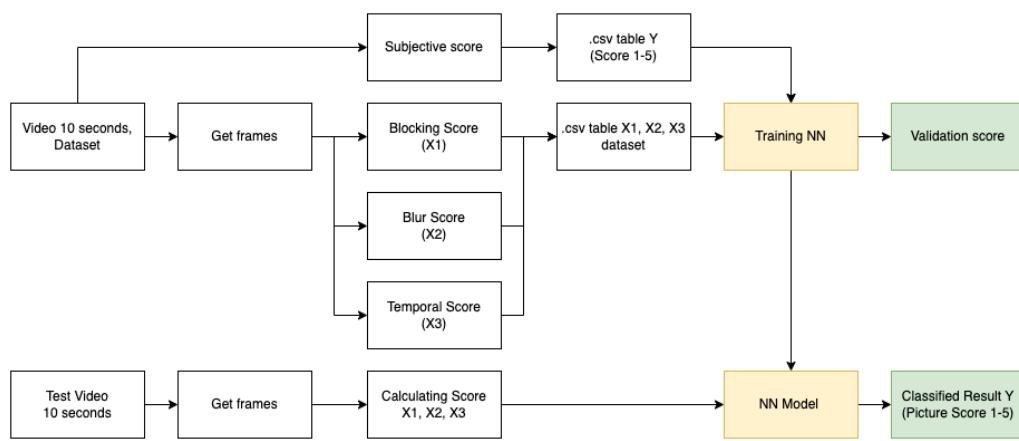
Gambar 3.11: Rancangan tampilan aplikasi web subyektif *assessment*

Aplikasi ditujukan untuk mempermudah proses pengujian pada subjektif *assessment*. Beberapa fitur yang dibutuhkan seperti melakukan input materi video yang akan diujikan, melakukan pengambilan data dan perekaman jawaban dari panelis, dan menarik keseluruhan data uji hasil evaluasi serta dapat di export ke dalam format .csv. Setiap panelis juga memiliki ID masing-masing sehingga data bisa dikelompokkan berdasarkan panelis atau berdasarkan materi yang diujikan.

3.4 Pembuatan Model NN

Jaringan saraf tiruan (*Neural network, NN*) digunakan dalam subjektif *assessment* untuk memprediksi kualitas gambar atau video berdasarkan fitur-fitur tertentu. Proses pembuatan model NN dimulai dengan pemilihan fitur-fitur yang relevan dengan kualitas gambar atau video yang ingin diprediksi. Fitur-fitur tersebut dapat berupa metrik kualitas gambar seperti metrik *blocking*, metrik *blur*, dan metrik *temporal*. Setelah pemilihan fitur, langkah selanjutnya adalah pemilihan arsitektur dan parameter untuk mod-

el NN. Pada penelitian ini dipilih *Recurrent Neural Network (RNN)* untuk memodelkan hubungan antara penilaian kualitas gambar/video oleh manusia dengan fitur-fitur gambar/video yang dijadikan input pada model seperti yang ditunjukkan pada gambar 3.12. Diharapkan penilaian kualitas gambar/video yang dihasilkan oleh model dengan RNN lebih akurat karena model dapat mengambil informasi sebelumnya pada urutan input yang diberikan.

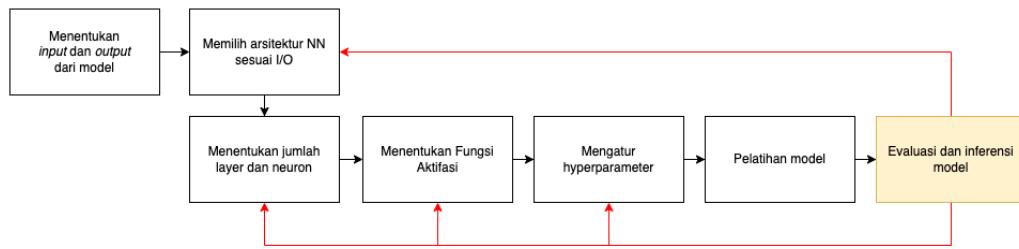


Gambar 3.12: Diagram parameter I/O untuk membangun model NN

Setelah arsitektur dan parameter ditentukan, langkah selanjutnya adalah pelatihan model NN menggunakan data latih yang telah dilabeli dengan nilai kualitas yang sesuai. Data latih harus mencakup variasi yang cukup dari kualitas gambar atau video yang ingin diprediksi. Setelah model dilatih, kinerjanya dievaluasi menggunakan data uji yang belum pernah dilihat oleh model (Shahreza et al. 2019). Berikut proses dan tahapan untuk membangun arsitektur NN yang akan dilakukan pada penelitian ini meliputi:

1. Menentukan masukan (input) dan keluaran (output) dari model: Langkah pertama adalah menentukan jenis data masukan yang akan digunakan dan keluaran yang diharapkan dari model.
2. Memilih arsitektur NN: Berdasarkan jenis masukan dan keluaran yang telah ditentukan, kemudian dipilih arsitektur NN yang tepat.

3. Menentukan jumlah lapisan dan neuron: Setelah memilih arsitektur NN yang tepat, selanjutnya ditentukan jumlah lapisan dan neuron pada setiap lapisan. Jumlah lapisan dan neuron ini dapat disesuaikan dengan kompleksitas data dan sumber daya komputasi yang tersedia.
4. Menentukan fungsi aktivasi: Fungsi aktivasi digunakan untuk mengaktifkan neuron pada setiap lapisan. Beberapa fungsi aktivasi yang umum digunakan adalah sigmoid, relu, dan tanh.
5. Mengatur hyperparameter: Hyperparameter adalah parameter yang diatur sebelum memulai pelatihan model. Beberapa hyperparameter yang perlu diatur adalah learning rate, batch size, dan jumlah epoch.
6. Pelatihan model: Setelah menentukan arsitektur NN dan hyperparameter, selanjutnya model dilatih dengan menggunakan data latih. Proses pelatihan ini dilakukan dengan mengoptimalkan fungsi loss yang dihasilkan.
7. Evaluasi model: Setelah selesai dilatih, model dievaluasi dengan menggunakan data uji untuk mengevaluasi performa dan akurasi model. Apabila hasil evaluasi masih belum memuaskan, dapat dilakukan iterasi kembali pada tahap pelatihan model atau penyesuaian parameter lainnya.
8. Prediksi atau inferensi: Setelah model dievaluasi dan dianggap cukup baik, model dapat digunakan untuk memprediksi nilai output dari data yang baru atau melakukan inferensi.
9. Penggunaan dan pemeliharaan model: Setelah berhasil dibuat, model dapat digunakan untuk memproses data yang baru. Namun, model juga perlu dipelihara secara berkala agar tetap optimal dalam memproses data.



Gambar 3.13: Tahapan pembuatan arsitektur dan model NN

Seluruh tahapan dilakukan dalam penelitian ini untuk memperoleh model RNN yang tepat seperti pada gambar 3.13. Seperti memilih jenis RNN yang sesuai dengan jenis data dan tugas yang akan dilakukan. Beberapa jenis RNN yang umum digunakan adalah LSTM, GRU, dan SimpleRNN. Kemudian penentuan hyperparameter seperti jumlah neuron, learning rate, jumlah epochs, dan batch size sangat penting untuk membangun model yang optimal. Evaluasi model secara sistematis setelah model dibangun, model harus dievaluasi secara sistematis menggunakan metrik yang tepat seperti akurasi, presisi, dan recall. Hal ini dapat membantu memastikan bahwa model dapat digunakan dengan efektif untuk tugas yang diberikan.

3.5 Hasil dan Perhitungan Model

Setelah model NN terbentuk, yang perlu dilakukan adalah melakukan pengukuran model tersebut. Akurasi, presisi, *recall*, *F1-score*, *confusion matrix*, dan *loss function* merupakan parameter yang perlu diukur pada model NN. Akurasi dihitung dengan membandingkan nilai prediksi dengan nilai sebenarnya pada data uji. Presisi dihitung dengan membagi jumlah prediksi positif benar dengan jumlah keseluruhan prediksi positif. Recall dihitung dengan membagi jumlah prediksi positif benar dengan jumlah keseluruhan data positif pada data uji. *F1-score* merupakan ukuran gabungan antara presisi dan *recall* dan dihitung dengan menghitung rata-rata harmonis antara presisi

dan recall.

3.5.1 *Confusion* matriks

Confusion matrix digunakan untuk mengevaluasi performa model NN dengan memperlihatkan jumlah prediksi benar dan salah yang dibuat oleh model. Confusion matrix digunakan untuk menghitung akurasi, presisi, dan *recall*. *Loss function* merupakan fungsi yang digunakan untuk mengukur seberapa besar kesalahan prediksi model pada setiap iterasi selama proses pelatihan. Loss function digunakan untuk mengoptimalkan bobot pada model NN. Semua parameter ini penting untuk mengukur dan mengevaluasi performa model NN dalam memprediksi hasil pengukuran subyektif pada sistem monitoring kualitas siaran DVB-T2.

Pada penelitian ini terdapat 5 klasifikasi pada hasil NN, yaitu nilai kualitas 1 sampai 5 berdasarkan ITU-R. Sehingga digunakan *multiple class confusion matrix* (MCCM) untuk mengevaluasi performa dari model NN. Sama seperti confusion matrix sederhana terdapat juga parameter TP, TN, FP, dan FN yang merupakan empat nilai dalam confusion matrix yang digunakan untuk mengevaluasi performa model pada tugas klasifikasi.

- TP (true positive) adalah jumlah kasus positif yang terdeteksi dengan benar oleh model.
- TN (true negative) adalah jumlah kasus negatif yang terdeteksi dengan benar oleh model.
- FP (false positive) adalah jumlah kasus negatif yang salah terdeteksi sebagai positif oleh model.
- FN (false negative) adalah jumlah kasus positif yang salah terdeteksi sebagai negatif oleh model.

		Predicted Class			
		C ₁	C ₂	...	C _N
Actual Class	C ₁	C _{1,1}	FP	...	C _{1,N}
	C ₂	FN	TP	...	FN

	C _N	C _{N,1}	FP	...	C _{N,N}

Gambar 3.14: *Multiple Class Confusion Matrix* (Markouliakis et al. 2021)

Dalam konteks MCCM, TP, TN, FP, dan FN dihitung berdasarkan kelas target dan prediksi yang berbeda pada data uji yang memiliki lebih dari dua kelas. Diagonal matriks MCCM menunjukkan jumlah prediksi benar untuk setiap kelas, sementara sel di luar diagonal menunjukkan jumlah prediksi salah untuk setiap kelas seperti yang ditunjukkan pada gambar 3.14. Oleh karena itu, TP, TN, FP, dan FN pada MCCM terdapat pada setiap sel matriks dan digunakan untuk menghitung akurasi, presisi, dan recall pada setiap kelas. Akurasi digunakan untuk mengukur seberapa sering model NN dapat memprediksi dengan benar kelas target pada seluruh data uji (Powers 2020). Menghitung akurasi pada MCCM dapat dilakukan dengan menjumlahkan seluruh TP dibagi dengan seluruh data pada matriks, atau dapat dituliskan dalam rumus:

$$Akurasi = \frac{\sum_{i=1}^k TP_i}{\sum_{i=1}^k (TP_i + FP_i)} \quad (3.22)$$

dimana:

TP_i : jumlah prediksi model yang benar dari kelas i

FP_i : jumlah prediksi model yang salah pada kelas i .

k : jumlah kelas pada model i .

Presisi digunakan mengukur seberapa sering model NN dapat memprediksi dengan benar kelas positif dari semua prediksi positif yang dibuat. Presisi dapat dianggap sebagai tingkat ketepatan dari prediksi positif model NN. Rumus untuk menghitung presisi adalah sebagai berikut:

$$\text{Presisi}_i = \frac{TP_i}{(TP_i + FP_i)} \quad (3.23)$$

Kemudian untuk menghitung *recall* atau True Positive Rate (TPR) atau sensitivitas, adalah metrik kinerja yang digunakan untuk mengukur seberapa baik model klasifikasi mengidentifikasi hasil positif yang benar dari keseluruhan hasil positif yang sebenarnya. Recall dihitung dengan menggunakan rumus berikut:

$$\text{Recall}_i = \frac{TP_i}{(TP_i + FN_i)} \quad (3.24)$$

Parameter berikutnya adalah F1-score, merupakan metrik kinerja yang digunakan untuk mengukur kinerja model klasifikasi, terutama dalam situasi di mana distribusi kelas tidak seimbang. F1-score merupakan rata-rata harmonik dari presisi dan recall, yang memperhitungkan keduanya dalam satu metrik. Rumus F1-score:

$$F1 - score_i = 2 \frac{\text{Presisi}_i * \text{Recall}_i}{(\text{Presisi}_i + \text{Recall}_i)} \quad (3.25)$$

Dalam konteks klasifikasi multi-kelas, presisi, *recall*, dan *f1-score* dihitung untuk setiap kelas secara terpisah dan kemudian digabungkan menggunakan metode seperti rata-rata sederhana (*macro-average*), rata-rata mikro (*micro-average*), atau rata-rata tertimbang (*weighted average*), tergantung pada konteks masalah dan jumlah sampel per kelas.

Rumus rata-rata sederhana (*macro-average*):

$$Average_{macro} = \frac{\sum_{i=1}^k Metrik_i}{k} \quad (3.26)$$

Rumus rata-rata mikro *micro-average*:

$$Average_{micro} = \frac{\sum_{i=1}^k TP_i}{(\sum_{i=1}^k TP_i + \sum_{i=1}^k FP_i)} \quad (3.27)$$

Rumus rata-rata tertimbang *weighted-average*:

$$Average_{weighted} = \frac{\sum_{i=1}^k (Metrik_i \times Bobot_i)}{(\sum_{i=1}^k Bobot_i)} \quad (3.28)$$

Memilih rata-rata yang tepat untuk klasifikasi multi-kelas bergantung pada konteks masalah dan tujuan evaluasi. Macro-average memberikan bobot yang sama pada setiap kelas, cocok untuk menekankan kinerja seimbang di semua kelas, terutama ketika ada ketidakseimbangan kelas yang signifikan. Micro-average memberikan bobot yang sama pada setiap sampel, lebih cocok jika Anda lebih peduli tentang kinerja model secara keseluruhan daripada kinerja kelas individu. Weighted average memberikan bobot yang proporsional terhadap jumlah sampel per kelas, lebih memperhatikan kinerja kelas yang lebih besar dalam evaluasi.

3.5.2 Pengukuran Korelasi

Korelasi antara pengukuran subyektif dan obyektif dapat digunakan untuk mengevaluasi sejauh mana metode pengukuran obyektif dapat memprediksi pengukuran subyektif yang dilakukan oleh manusia. Dengan mengetahui korelasi antara hasil pengukuran subyektif dan objektif, kita dapat mengevaluasi seberapa baik pengukuran objektif dapat merepresentasikan penilaian kualitas gambar atau video oleh manusia. Untuk mengukur korelasi antara pengukuran subyektif dan obyektif, seringkali digunakan koefisien korelasi Pearson. Koefisien korelasi Pearson mengukur hubungan linier antara dua variabel, dalam

hal ini pengukuran subyektif dan obyektif. Koefisien ini memiliki nilai antara -1 hingga 1, di mana 1 menunjukkan hubungan positif sempurna, -1 menunjukkan hubungan negatif sempurna, dan 0 menunjukkan tidak ada hubungan linier antara kedua variabel.

Selain koefisien korelasi Pearson, terdapat juga metode pengukuran korelasi lainnya yang dapat digunakan, seperti koefisien korelasi Spearman dan koefisien korelasi Kendall. Korelasi Spearman juga merupakan salah satu jenis korelasi yang digunakan untuk mengukur hubungan antara dua variabel. Korelasi Spearman menghitung koefisien korelasi rho (ρ) antara peringkat dua variabel yang tidak berdistribusi normal atau tidak memiliki asumsi normal (M.H. Pinson 2004). Korelasi Spearman dapat digunakan untuk mengukur korelasi antara nilai subyektif dan obyektif dalam penilaian kualitas video atau gambar.

Berikut rumus yang digunakan untuk menghitung nilai koefisien korelasi Pearson:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.29)$$

dimana:

- r_{xy} : koefisien korelasi Pearson i
- x_i : nilai dari variabel x dalam sample i .
- \bar{x} : rata-rata variabel x.
- y_i : nilai dari variabel y dalam sample i .
- \bar{y} : rata-rata variabel y.
- n : jumlah sample.

Sedangkan untuk menghitung nilai koefisien korelasi Spearman digunakan rumus sebagai berikut:

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (3.30)$$

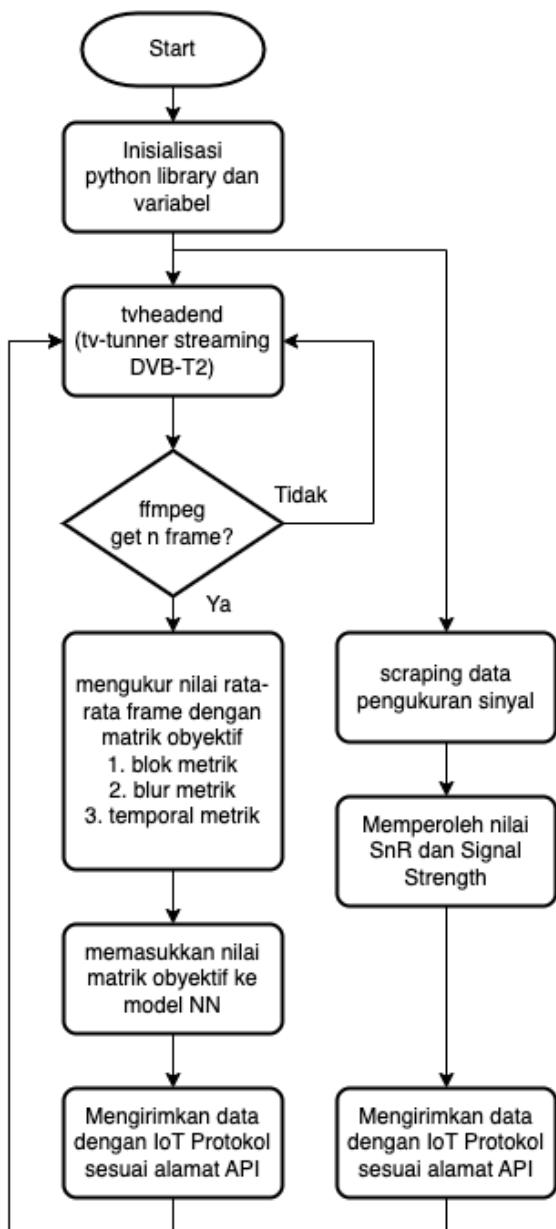
dimana:

- ρ : koefisien korelasi peringkat Spearman i
 d_i : perbedaan antara peringkat variabel X dan Y pada observasi ke- i .
 n : jumlah sample.

Korelasi Pearson dan Spearman merupakan metode yang berbeda untuk mengukur hubungan antara dua variabel. Korelasi Pearson mengukur hubungan linier antara variabel dengan menggunakan nilai aktual, dan mengasumsikan distribusi normal serta hubungan linier. Sementara itu, korelasi Spearman mengukur hubungan monoton dengan menggunakan peringkat variabel, dan tidak mengharuskan distribusi normal atau hubungan linier, sehingga lebih cocok untuk data ordinal atau data yang tidak berdistribusi normal (Akoglu 2018).

3.6 Sistem Pengukuran Waktu Nyata

Sistem pengukuran waktu nyata pada kualitas gambar siaran DVB-T2 dilakukan setelah didapat model yang dapat melakukan pengukuran dengan tingkat akurasi yang cukup tinggi dibandingkan dengan penilaian subyektif manusia. Prosedur yang dilakukan dalam melakukan pengukuran waktu nyata seperti yang digambarkan pada gambar 3.15. Hardware yang digunakan adalah raspberry pi 4 model B dengan menggunakan TV HAT dan juga antenna. Raspberry pi sudah terisntall raspbian OS 64-bit, dan sudah diinstall aplikasi *tvheadend*, *ffmpeg*, dan juga *python* dengan *library*-nya.



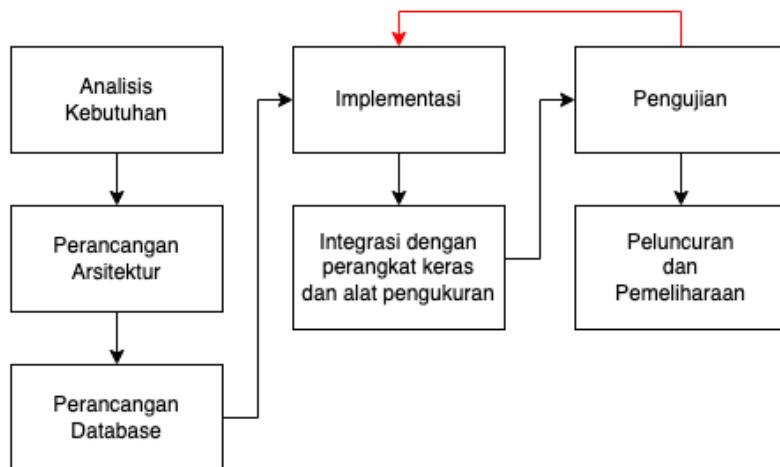
Gambar 3.15: Diagram alur program python pada pengukuran kualitas gambar siaran DVB-T2 waktu nyata

Pertama gambar ditangkap dari web aplikasi `tvheadend` yang diperoleh langsung pada siaran DVB-T2. Gambar ditangkap menggunakan aplikasi `ffmpeg` yang sudah diprogram menggunakan python untuk melakukan pengambilan data gambar dengan interval tertentu secara berkala. Gambar yang diamambil merupakan urutan frame video sebanyak 5 frame dengan interval 0.5 detik.

Frame gambar tersebut kemudian diukur satu persatu menggunakan metrik obyektif blur, blok, dan temporal. Hasil pengukuran obyektif dirata-ratakan kemudian digunakan sebagai parameter masukan pada model NN. Nilai yang diperoleh dari model NN selanjutnya akan dikirim ke IoT dashboard menggunakan protokol IoT seperti *thingspeak* milik *Matlab*.

3.6.1 Tahapan pembuatan *website* pemantauan

Web aplikasi monitoring kualitas penyiaran DVBT-2 merupakan sebuah aplikasi web yang dapat digunakan untuk memonitor kualitas penyiaran televisi secara waktu nyata *real-time*. Aplikasi ini dapat menampilkan informasi tentang kualitas siaran seperti kualitas video dan kualitas sinyal. Aplikasi ini dilengkapi dengan fitur-fitur seperti pemantauan kualitas siaran secara real-time, pemantauan kualitas siaran secara historis, analisis data kualitas siaran, serta notifikasi jika terjadi gangguan atau perbaikan dalam kualitas siaran. Web aplikasi ini dapat berguna bagi stasiun televisi atau radio untuk memastikan bahwa kualitas siaran yang mereka kirimkan kepada pemirsa adalah yang terbaik, serta memastikan bahwa siaran tidak mengalami gangguan atau masalah teknis. Tahapan dari pembuatan web ditunjukkan pada gambar



Gambar 3.16: Tahapan pembuatan *website monitoring real-time* kualitas gambar DVB-T2

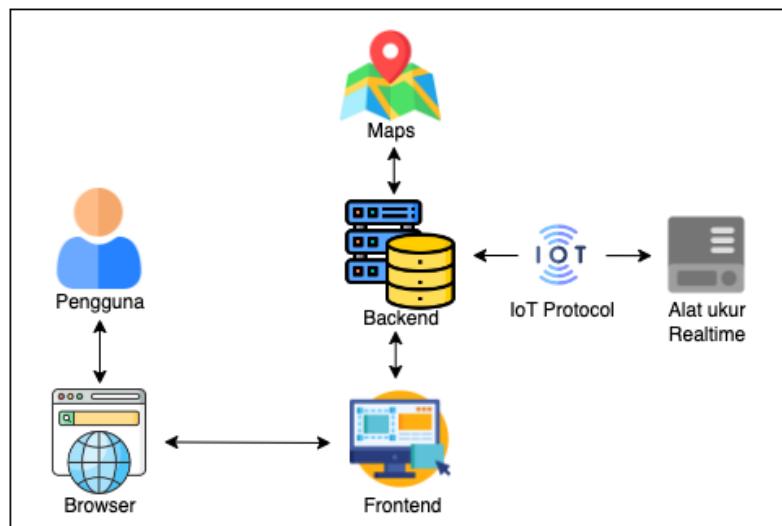
Perancangan sistem web aplikasi monitoring televisi digital melibatkan beberapa tahapan yang harus dilakukan secara terstruktur. Tahapan pertama adalah melakukan analisis kebutuhan untuk menentukan fitur dan fungsi yang diperlukan dalam aplikasi. Analisis kebutuhan dilakukan dengan mengumpulkan informasi dari stakeholder dan melakukan studi literatur. Selanjutnya, dilakukan perancangan arsitektur sistem, yang mencakup infrastruktur server, database, dan interaksi antarmuka pengguna dengan sistem. Arsitektur ini harus dirancang dengan mempertimbangkan aspek keamanan, skalabilitas, dan kinerja sistem.

Setelah perancangan arsitektur, dilakukan perancangan database untuk menyimpan data yang dibutuhkan oleh aplikasi, seperti data pengguna, data channel televisi, dan data kualitas sinyal. Tahapan selanjutnya adalah implementasi, yang mencakup pembuatan kode program, integrasi komponen sistem, pengujian, dan debugging. Penting juga untuk melakukan integrasi dengan perangkat perekam dan alat ukur kualitas sinyal agar aplikasi dapat merekam dan memonitor kualitas siaran secara real-time. Tahap pengujian dilakukan untuk memastikan aplikasi berfungsi sesuai dengan yang diharapkan

dan memenuhi standar kualitas yang ditetapkan. Setelah aplikasi diuji dan dirasa siap untuk digunakan, dilakukan tahap peluncuran dengan menginstal aplikasi di server dan memberikan akses kepada pengguna. Perlu diingat bahwa setelah aplikasi diluncurkan, perlu dilakukan pemeliharaan sistem secara berkala untuk menjaga kinerja dan keamanan sistem.

3.6.2 Arsitektur *Website Monitoring*

Penerapan hasil pengukuran kualitas gambar pada siaran DVB-T2 ditampilkan pada halaman situs online untuk melakukan pemantauan secara waktu nyata. Pada penelitian ini rancangan arsitektur *website* ditunjukkan pada gambar 3.17. Aplikasi pemantauan dapat diakses melalui web browser. Aplikasi ini ditujukan untuk mempermudah pengguna untuk melihat hasil pengukuran yang sedang berlangsung secara realtime.



Gambar 3.17: Diagram arsitektur *website* pemantauan kualitas gambar siaran DVB-T2

Laman yang diakses atau diminta, memiliki tampilan antarmuka atau yang dikenal dengan user interface. Front end pada penerapan sistem ini menggunakan *native javascript*. Frontend Native JavaScript merujuk pada pengem-

bangun antarmuka pengguna (user interface) berbasis web yang dibangun menggunakan bahasa pemrograman JavaScript murni tanpa mengandalkan framework atau library tertentu. Frontend Native JavaScript memungkinkan pengembang untuk lebih fleksibel dalam menyesuaikan dan mengubah kode mereka karena mereka memiliki kontrol penuh atas kode tersebut.

Server yang digunakan pada penelitian ini adalah virtual machine yang terdapat pada server universitas gunadarma. Sistem operasi yang digunakan Ubuntu 22.1 dan *apache* web server. Penyimpanan basis data menggunakan MySQL. Tampilan Maps yang ditampilkan pada front end merupakan hasil dari permintaan yang terintegrasi terhubung dari Mapbox API. Peta digunakan hanya untuk mengukur jarak antara alat pengukur dengan stasiun pemancar. Kemudian digunakan juga protokol IoT yang berfungsi untuk mengirimkan data pengukuran kualitas sinyal dan gambar ke IoT dashboard yang kemudian API dari IoT tersebut diambil dari web server. Pengguna hanya dapat memilih lokasi pemantauan yang sebelumnya sudah dibuat oleh admin. Lokasi pemantauan merupakan titik dimana perangkat melakukan pengukuran. Selanjutnya data hasil pengukuran ditampilkan secara realtime pada antarmuka web.

BAB IV

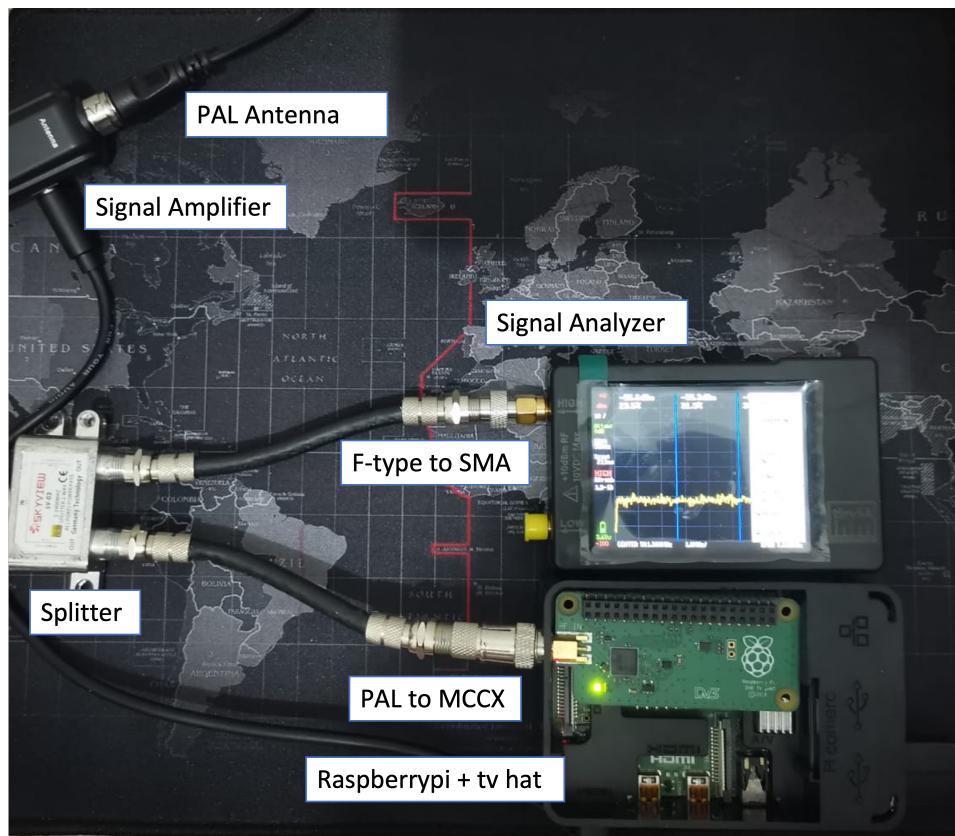
HASIL DAN PEMBAHASAN

4.1 Material yang digunakan

Material dalam penelitian ini berupa video yang diambil dari siaran DVB-T2. Material yang digunakan akan diujikan secara obyektif menggunakan beberapa metrik spatial dan metrik temporal, serta diujikan secara subjektif ke beberapa panelis. Pemilihan material menjadi aspek penting dalam proses evaluasi kualitas video. Material yang dipilih harus mencerminkan berbagai jenis konten dan situasi yang mungkin dihadapi oleh penonton dalam kehidupan nyata. Memilih material untuk tes subjektif melibatkan beberapa kriteria penting, seperti variasi genre, kualitas sumber, durasi klip, tingkat detail, gerakan dan kecepatan, warna dan kontras, serta artefak khas.

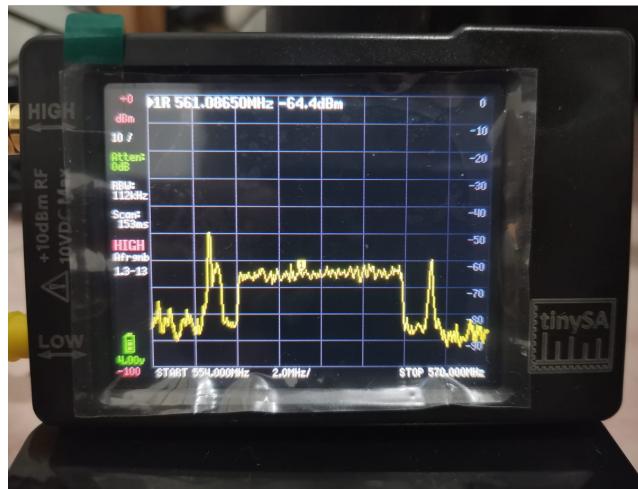
4.1.1 Proses akuisisi material video

Perangkat keras yang digunakan untuk melakukan akuisisi pada video secara langsung dari siaran DVB-T2 adalah raspberry pi dengan modul tv hat, antena dan periferal untuk komputer yang ditunjukkan pada gambar 4.1. Sebelum melakukan pengambilan data dilakukan beberapa setup untuk mendapatkan sampel yang sesuai. Pertama adalah memastikan seluruh perangkat terpasang dan terinstall dengan baik dan benar. Perangkat *tv hat* harus *attach* di atas raspberry pi. Perangkat *tv hat* memiliki *jack input* dari antena berbentuk *micro miniature coaxial (MMCX) to PAL-female*. Konektor PAL female kemudian terhubung ke penguat sinyal eksternal dengan daya 5watt untuk selanjutnya terhubung ke antena tv.



Gambar 4.1: Perangkat keras yang digunakan untuk akuisisi video siaran DVB-T2

Langkah kedua, memastikan kekuatan sinyal siaran DVB-T2 yang diterima oleh perangkat ada di antara -60 dB sampai -75 dB. Kekuatan sinyal dapat diperkuat menggunakan alat penguat sinyal dan mengubah posisi antenna. Kekuatan sinyal dapat diukur menggunakan *tv hat* dengan aplikasi *tvheadend* atau bisa juga menggunakan perangkat signal analyzer seperti yang ditunjukkan pada gambar 4.2. Ketiga membuat rencana perekaman video menggunakan fitur *record* pada aplikasi *tvheadend*. Perekaman dilakukan dengan beberapa skema untuk memperoleh video dengan kualitas yang baik dan juga video dengan kualitas yang buruk.



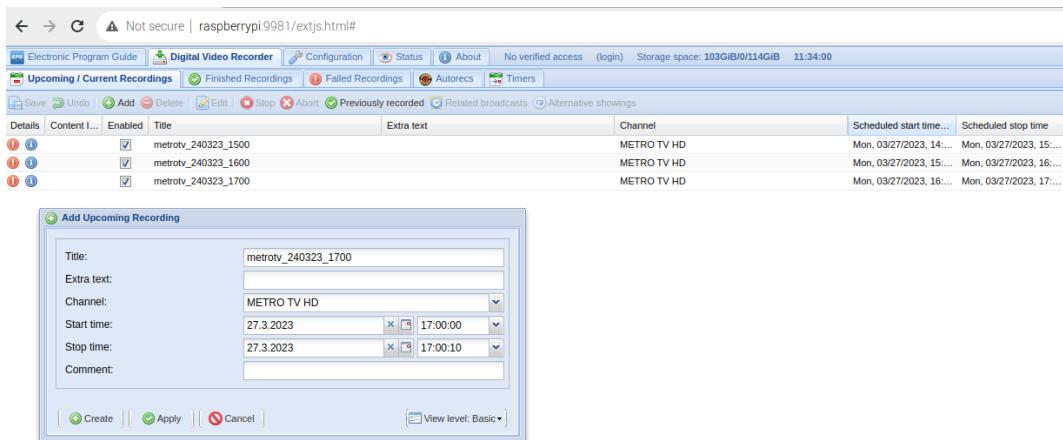
Gambar 4.2: Pengukuran kekuatan sinyal menggunakan *signal analyzer*, *TinySA*

Antena diletakkan diluar bangunan (*outdoor*) dengan ketinggian 2 meter diatas bangunan. Antena yang digunakan adalah antena televisi dengan tipe PX HDA-5000 seperti yang ditunjukkan pada gambar 4.3. Kemudian dilakukan beberapa skema dengan mengubah posisi antena ke empat arah yang berbeda selama pengambilan data video. Posisi antena awal 0° posisi menghadap ke stasiun pemancar, serta diputar 90° , 180° , dan 270° . Hal ini dilakukan agar mendapatkan gambar dengan kualitas yang bervariasi untuk melengkapi *dataset* yang dibutuhkan.



Gambar 4.3: Penempatan antena outdoor PX HDA-5000

Setelah seluruh perangkat sudah siap, maka pengambil data video dapat dilakukan menggunakan aplikasi *tvheadend*. Sebelum menjalankan *tvheadend* pastikan juga sudah melakukan instalasi pada perangkat keras, serta instalasi aplikasi *tvheadend* yang disertai proses pencarian layanan siaran DVB-T2. Selanjutnya dilakukan beberapa konfigurasi pada *tvheadend* dengan malakukan proses perekaman dengan durasi 10 detik per video, dengan interval waktu tertentu. Perekaman dapat menggunakan fitur "*digital video recorder*", kemudian klik "*upcoming/current recording*", lalu tekan tombol "+" untuk menambahkan jadwal perekaman video seperti yang ditunjukkan pada gambar 4.4. Jendela *pop-up* akan muncul untuk mengisikan judul/nama rekaman, channel/layanan yang dipilih, serta waktu melakukan perekaman. Setelah itu tekan "*apply*". Hasil dari rekaman dapat di download pada tab "*finished recordings*"



Gambar 4.4: Fitur melakukan perekaman video siaran DVB-T2 menggunakan *tvheadend*

Selain menggunakan fitur yang terdapat pada *tvheadend* pengambilan video juga dapat dilakukan menggunakan *ffmpeg* dan *python*, namun tetap membutuhkan alamat lokal layanan dari *tvheadend*. Alamat layanan dapat diperoleh dari channel layanan yang terdapat dari *tvheadend* dan dapat di download dengan format .m3u (*Moving Picture Experts Group Audio Layer*

3 Uniform Resource Locator). Digunakan beberapa *library* untuk dapat menjalankan ffmpeg dalam melakukan perekaman dengan interval waktu tertentu. Library yang digunakan adalah *time*, *schedule*, dan *subprocess*. Contoh program yang digunakan adalah sebagai berikut:

```
import schedule
import time
import subprocess

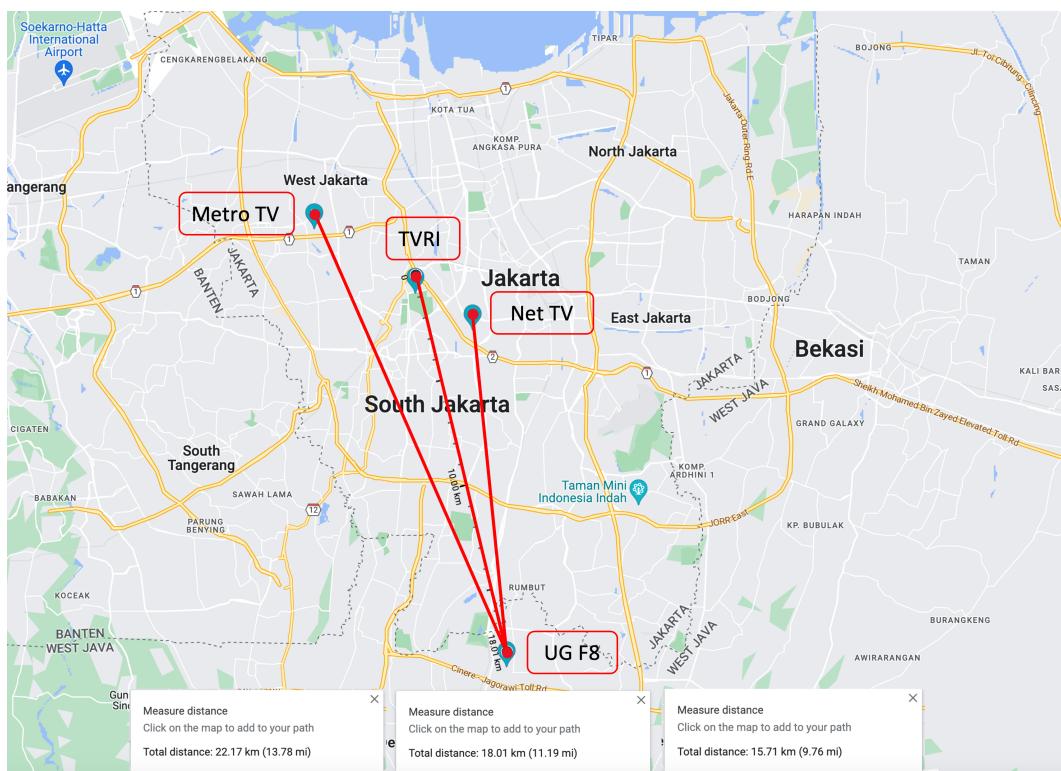
def my_function():
    command = 'perintah ffmpeg record + timestamp'
    subprocess.call(shlex.split(command))

schedule.every(1).hours.do(my_function)
while True:
    schedule.run_pending()
    time.sleep(1)
```

Perekaman materi uji dilakukan dengan durasi 10 detik per video tayangan. Materi video direkam dari lima layanan televisi yang diperoleh di wilayah kota Depok, Jawa Barat. Layanan televisi yang digunakan sebagai *dataset* adalah TVRI World, TVRI Sport, Metro TV HD, Net TV, dan My TV dengan tiga frekuensi pembawa berbeda . Sudut antena diatur sehingga mendapat kekuatan sinyal yang bervariasi agar mendapatkan hasil data yang dapat menggambarkan kondisi siaran real. Terdapat tiga frekuensi pembawa yang berbeda dari kelima layanan tv tersebut yang ditunjukkan pada tabel 4.1. Lokasi dari ketiga stasiun pemancarnya juga dicatat sebagai penghitungan variabel jarak antara pengirim dan penerima siaran. Jarak line of sight (LOS) antara ketiga stasiun pemancara terhadap perangkat yang digunakan pada penelitian relatif, dimana lokasi pengambilan video dari siaran dilakukan dari kampus F8 Universitas Gunadarma.

Tabel 4.1: Pengambilan dataset dari layanan siaran DVB-T2 pada area studi

Frekuensi Pembawa	Nama Layanan	Lokasi Pemancar	Latitude, Langitude	Kekuatan sinyal di 0°, 90°, 180°, 270° dari pemancar
626 MHz	TVRI World	Tanah Abang, Jakarta Pusat	-6.21254, 106.80071	-64 dBm, -74 dBm -81 dBm, -75 dBm
	TVRI Sport	Jakarta Barat	106.75871	-80 dBm, -75 dBm
562 MHz	Metro TV	Kebon Jeruk, Jakarta Barat	-6.18698, 106.75871	-64 dBm, -75 dBm -80 dBm, -75 dBm
	My TV	Jakarta Barat	106.71896	-81 dBm, -74 dBm
650 MHz	Net TV	Kembangan, Jakarta Barat	-6.22192, 106.71896	-65 dBm, -75 dBm -81 dBm, -74 dBm
	F8 Universitas Gunadarma	Cimanggis, Kota Depok	-6.36390, 106.84019	Penerima siaran

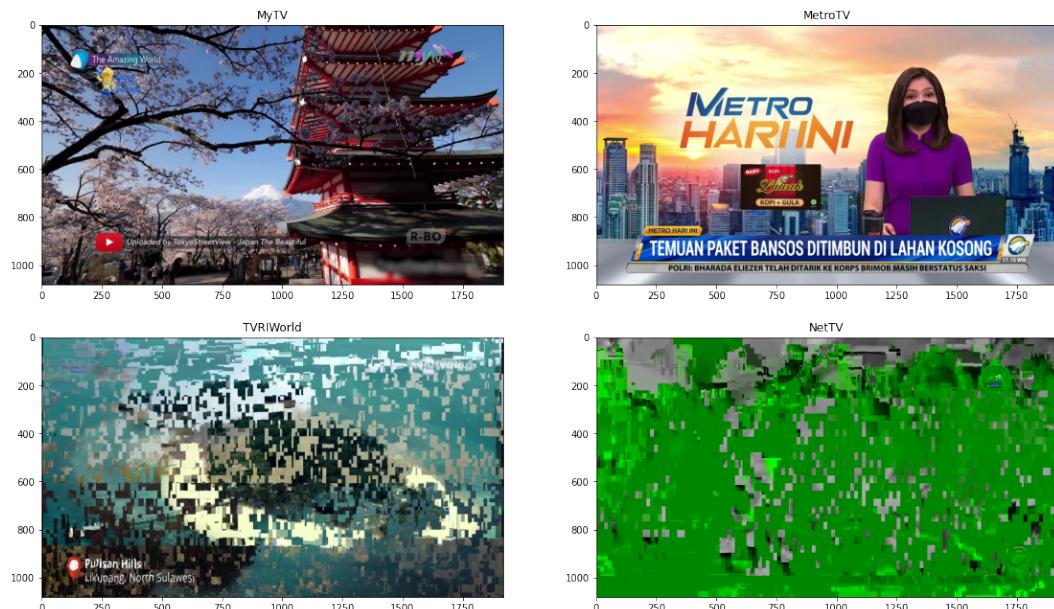


Gambar 4.5: Jarak stasiun pemancar terhadap lokasi pengambilan data siaran DVB-T2

Stasiun pemancar terjauh adalah metro tv dengan jarak 22 km. Stasiun siaran TVRI yang berlokasi di kec. Tanah Abang memiliki jarak 18 km, sedangkan yang terdekat adalah stasiun net tv dengan jarak 16 km. Semua pengukuran jarak tersebut diukur secara LOS menggunakan google maps seperti yang ditunjukkan pada gambar 4.5.

4.1.2 Koleksi data

Pengumpulan data merupakan proses terstruktur untuk menghimpun informasi dari berbagai sumber guna mendapatkan wawasan, menganalisis situasi, atau membantu pengambilan keputusan. Melibatkan tahapan perencanaan, identifikasi sumber, pemilihan metode pengumpulan data yang tepat, dan pengolahan data untuk analisis lebih lanjut. Tahapan ini merupakan komponen penting dalam penelitian, uji coba, evaluasi, dan beragam disiplin ilmu lainnya untuk mendukung hipotesis, menguji teori, atau memvalidasi hasil (Yin n.d.).



Gambar 4.6: Citra/frame video dengan resolusi FHD 1920×1080 piksel hasil rekaman dari siaran DVB-T2

Data yang sudah terkumpul dari hasil akuisisi berupa video dengan durasi 10 detik. Ukuran dari video sekitar 2 MB sampai 10 MB. Gambar 4.6 merupakan contoh frame yang didapat dari hasil pengambilan video menggunakan prosedur sebelumnya. Diperoleh data dari masing-masing siaran sejumlah 100 video, sehingga terdapat 500 video. Video yang akan dijadikan dataset untuk pengujian obyektif dan subyektif dipilih secara manual dari video yang ada. Pemilihan manual video untuk dataset tersebut berdasarkan kualitas dari video, dari jumlah artefak bloking, bluring dan juga pembekuan frame. Dipilih sejumlah 105 video yang dijadikan *dataset*. Jumlah dataset awal yang digunakan tidak terlalu besar karena dataset tersebut juga digunakan untuk pengukuran subyektif yang dinilai oleh panelis.

4.1.3 Perangkat pengolahan data

Setelah dipilih sebanyak 105 video untuk *dataset*, maka video akan diukur menggunakan beberapa metrik obyektif spatial dan juga temporal. Pengukuran dilakukan menggunakan dua perangkat komputer untuk pembanding yaitu vm milik *google-colab* dan juga vm *dgx-a100* milik UG. Perangkat tersebut juga digunakan untuk pembuatan model NN dari hasil pengukuran obyektif dan subyektif dari dataset. Spesifikasi umum perangkat yang digunakan untuk pengolahan data ditunjukkan pada tabel 4.2

Tabel 4.2: Perangkat pengolahan data yang digunakan

Mesin GPU NVIDIA	Keterangan
Google Colab	TU104-895-16GB (Tesla 4)
Universitas Gunadarma	A-100-SXM4-40GB (DGX A00)

4.2 Pengujian metrik obyektif

Proses pengukuran kualitas video secara obyektif, digunakan tiga metrik.

Algoritma metrik obyektif spatial yang digunakan terdiri metrik *blocking* dan metrik *blur*, untuk temporal menggunakan metrik *temporal freeze*. Metrik tersebut diubah ke dalam bahasa pemrograman *python* yang kemudian dilakukan pengujian ke beberapa citra/video. Metrik tersebut juga akan digunakan sebagai parameter masukan pada jaringan saraf buatan (NN).

4.2.1 Pengujian metrik *blocking*

Tahap awal mengubah algoritma metrik *blocking* ke dalam bahasa pemrograman *python*. Kemudian dilakukan pengujian metrik tersebut untuk beberapa citra, dimana semakin besar kerusakannya maka nilainya akan semakin kecil. Interval pengukuran dari metrik ini yaitu 1 – 100 dengan skor 1 citra dengan tingkat noise bloking yang tinggi, dan skor 100 citra tanpa bloking. Pseudocode 4.1 menuliskan secara umum perhitungan yang dilakukan algoritma blok milik Zhou Wang.

Algoritma 4.1 Metrik pengukuran blok

INPUT: citra

OUTPUT: *blok_score*

```

1: function BLOK_METRIC(image_path)
2:   img  $\leftarrow$  load_image(image_path)                                 $\triangleright$  Load citra
3:   dh  $\leftarrow$  img[:, 1 :] – img[:, : – 1]                          $\triangleright$  Selisih horizontal
4:   dv  $\leftarrow$  img[1 :, :] – img[: – 1, :]                          $\triangleright$  Selisih vertikal
5:   Bh, Bv  $\leftarrow$  Hitung nilai Bh, Bv                       $\triangleright$  Mean bloking 8x8
6:   Ah, Av  $\leftarrow$  Hitung nilai Ah, Av                       $\triangleright$  Mean dari mean blok - Bh
7:   sigh, sigv  $\leftarrow$  Hitung nilai sigh, sigv           $\triangleright$  Zero crossing
8:   Zh, Zv  $\leftarrow$  Hitung nilai Zh, Zv                       $\triangleright$  Mean zero crossing
9:   B, A, Z  $\leftarrow$  Hitung nilai B, A, Z                   $\triangleright$  Mean variabel v dan h
10:  score  $\leftarrow$  Hitung nilai score                                $\triangleright$  Skor akhir
11:  return score
12: END function

```

Pada gambar 4.7 dicontohkan nilai luaran yang dihasilkan dari pengukuran metrik *bloking*. Citra yang dicontohkan merupakan hasil cuplikan frame dari video yang ada di dataset.

Blok score : 85



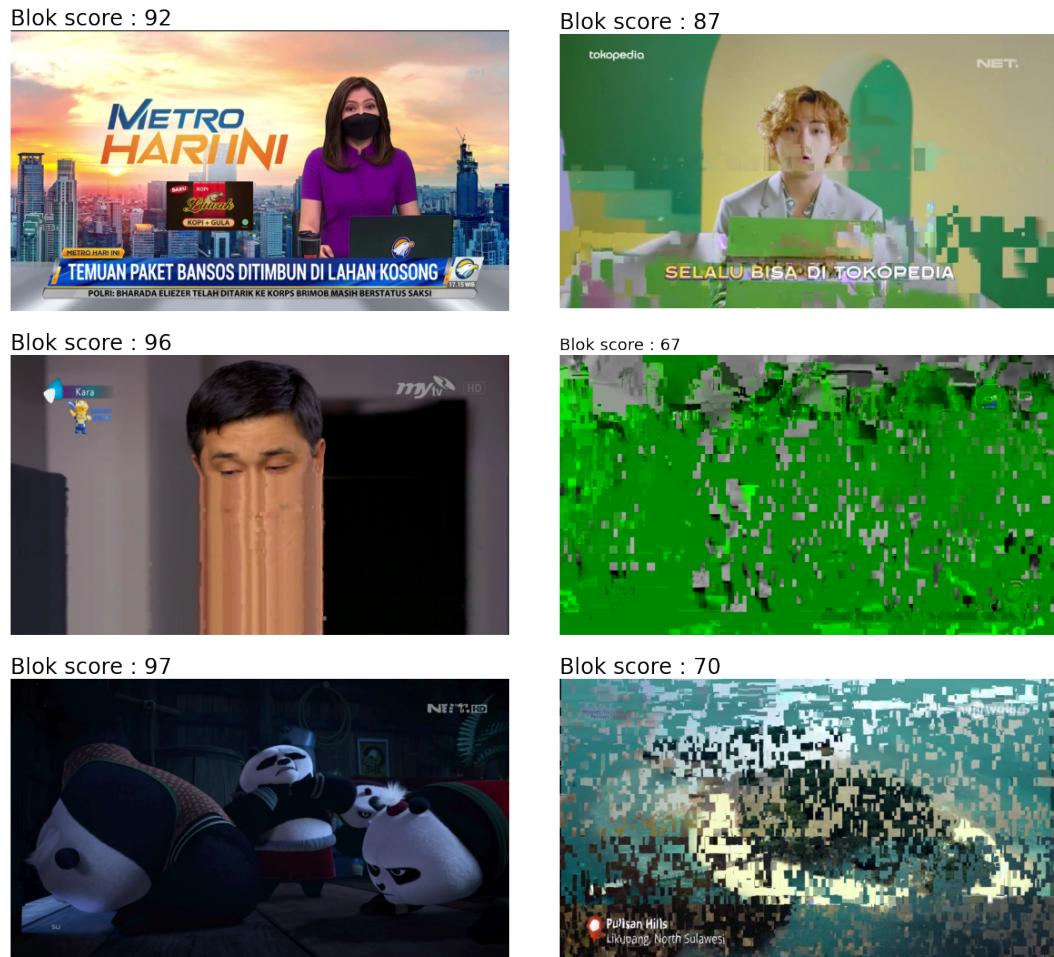
Blok score : 75



Gambar 4.7: Pengukuran metrik obyektif blok pada citra normal dan *blocking artifact*

Pada gambar 4.7 menunjukkan pengukuran blok pada citra frame siaran DVB-T2 yang ada di *dataset*. Citra bagian atas merupakan citra asli dari siaran, sedangkan citra pada bagian bawah merupakan citra yang sudah diberikan gangguan. Gangguan yang diberikan berupa noise akibat kerusakan data yang terencode dalam proses transmisi. Ditunjukkan citra dengan gangguan blok memiliki skor lebih rendah (skor = 75) dibanding citra asli yang tidak blur

(skor = 85) meskipun tidak terlalu signifikan.



Gambar 4.8: Contoh pengukuran metrik obyektif blok pada beberapa citra siaran televisi DVB-T2

Pengukuran metrik blok dilakukan secara langsung terhadap citra dari siaran televisi DVB-T2 memiliki beberapa kendala. Seperti yang terlihat pada gambar 4.8, metrik blok tidak signifikan dalam penurunan skor antara gambar dengan kualitas yang baik dan gambar dengan *blocking artifact*. Metrik *bloking* ini juga hanya dapat mendeteksi adanya bloking secara lokal di dalam citra, dan tidak dapat mendeteksi bloking secara menyeluruh di seluruh citra. Hal ini bisa menjadi kekurangan dalam kasus-kasus tertentu di mana bloking yang terjadi lebih kompleks atau terdistribusi secara luas di seluruh citra. Metrik ini juga hanya berfokus pada jenis distorsi *bloking* dan tidak mempertimbangkan

distorsi lainnya, seperti distorsi *blur*, *noise*, dan *masking*, sehingga perlu juga dilakukan pengukuran dengan metrik lain.

4.2.2 Pengujian metrik *blur*

Metrik blur pada dasarnya mengukur tingkat kekaburan atau ketidakjelasan pada gambar dengan membandingkan nilai piksel pada tepi gambar dengan nilai piksel sekitarnya. Pengukuran dilakukan dengan cara menghitung perbedaan nilai antara setiap piksel pada tepi gambar dengan nilai piksel terdekat di sekitarnya. Metrik blur menggunakan deteksi tepi pada gambar dengan menggunakan filter Sobel dan menghitung nilai lokal maksimum dan lokal minimum pada tiap tepi gambar. Dalam proses pengukuran, semakin tinggi nilai perbedaan pada tepi gambar dengan nilai piksel di sekitarnya, maka semakin tinggi pula tingkat kekaburan pada citra.

Selanjutnya berdasarkan beberapa percobaan pengukuran pada gambar, hasil dari metrik blur dinormalisasi pada rentang skala 1 hingga 100. Nilai skor 100 menandakan bahwa gambar tidak mengalami kekaburan sedangkan nilai skor 1 menunjukkan bahwa gambar memiliki tingkat kekaburan yang sangat tinggi . Pengukuran yang dilakukan oleh metrik blur dapat membantu dalam mengukur tingkat kekaburan pada gambar yang mungkin sulit diukur secara subjektif oleh manusia, sehingga dapat digunakan sebagai alat bantu dalam memperbaiki atau meningkatkan kualitas gambar. Pada penelitian ini metrik blur dibuat ke bahasa pemrograman *python* untuk melakukan proses pengukuran.

Citra yang sudah disiapkan untuk dialukan pengujian di unggah ke dalam aplikasi *Jupyter Notebook*. Program juga sudah dituliskan di dalam aplikasi yang sama. Pengukuran pada gambar dilakukan satu persatu dengan urutan proses seperti yang ditunjukkan pada pseudocode algoritma 4.3.

Algoritma 4.2 Metrik pengukuran blur

INPUT: citra**OUTPUT:** blur_score

```

1: function BLUR_METRIC(image_path)
2:   img  $\leftarrow$  load_image(image_path)                                 $\triangleright$  Load citra
3:   grayImg  $\leftarrow$  convert_to_grayscale(img)                       $\triangleright$  Skala abu
4:   edgeImg  $\leftarrow$  apply_sobel_filter(img)                          $\triangleright$  Sobel filter
5:   maxVals  $\leftarrow$  dilate_edges(edgeImg)                         $\triangleright$  Lokal maksimum
6:   minVals  $\leftarrow$  erode_edges(edgeImg)                          $\triangleright$  Lokal minimum
7:   diffImg  $\leftarrow$  diff(maxVals, minVals)                      $\triangleright$  Hitung lebar tepi
8:   blurMetric  $\leftarrow$  sum_(diffImg)                            $\triangleright$  Jumlah lebar tepi
9:   totalEdges  $\leftarrow$  count_edges(edgeImg)                       $\triangleright$  Jumlah seluruh tepi
10:  score  $\leftarrow$  score(blurMetric, totalEdges)                   $\triangleright$  Menghitung skor
11:  score  $\leftarrow$  limit_round_score(score)                          $\triangleright$  Limit dan bulatkan skor
12:  return score
13: END function

```

Blur score : 82

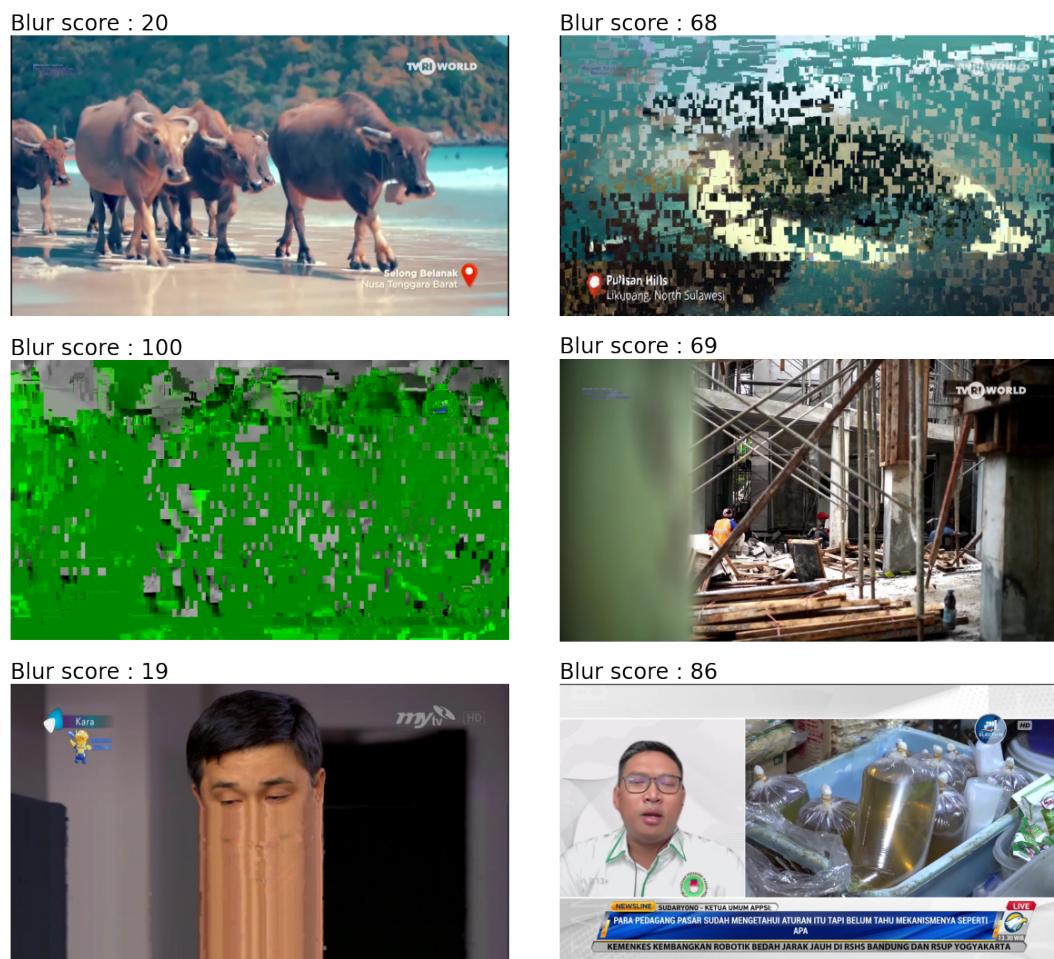


Blur score : 18



Gambar 4.9: Pengukuran metrik obyektif blur pada citra normal dan blur

Pada gambar 4.9 menunjukkan pengukuran blur pada citra frame siaran DVB-T2 yang ada di *dataset*. Citra bagian atas merupakan citra asli dari siaran, sedangkan citra pada bagian bawah merupakan citra yang sudah diberikan gangguan *gaussian noise blur* menggunakan library *opencv*. Pada gambar tersebut ditunjukkan citra dengan gangguan blur memiliki skor lebih rendah (skor = 18) dibanding citra asli yang tidak blur (skor = 82).



Gambar 4.10: Contoh pengukuran metrik obyektif blur pada beberapa citra siaran televisi DVB-T2

Pengukuran metrik *blur* dilakukan secara langsung terhadap citra dari siaran televisi DVB-T2 memiliki beberapa kendala seperti yang terlihat pada gambar 4.10. Metrik blur memiliki ketergantungan pada jenis citra dan kondisi pencahayaan dapat mempengaruhi hasil deteksi tepi dan oleh karena itu da-

pat mempengaruhi metrik blur. Gambar yang mengalami *bloking* atau *noise* dapat memberikan skor blur yang tinggi karena terdapat banyak tepi baru dari noise *tersebut*, sementara gambar yang gelap dapat memberikan skor blur yang rendah karena tepi yang terlihat samar. Selain itu, pada gambar dengan efek bokeh atau gambar yang dengan sengaja di-*blur*, skor *blur* yang dihasilkan mungkin tidak mencerminkan kualitas *blur* secara objektif.

Algoritma 4.3 Metrik pengukuran blur

INPUT: citra

OUTPUT: blur_score

```

1: function BLUR_METRIC(image_path)
2:   img  $\leftarrow$  load_image(image_path)                                 $\triangleright$  Load citra
3:   grayImg  $\leftarrow$  convert_to_grayscale(img)                       $\triangleright$  Skala abu
4:   edgeImg  $\leftarrow$  apply_sobel_filter(img)                         $\triangleright$  Sobel filter
5:   maxVals  $\leftarrow$  dilate_edges(edgeImg)                       $\triangleright$  Lokal maksimum
6:   minVals  $\leftarrow$  erode_edges(edgeImg)                         $\triangleright$  Lokal minimum
7:   diffImg  $\leftarrow$  diff(maxVals, minVals)                       $\triangleright$  Hitung lebar tepi
8:   blurMetric  $\leftarrow$  sum_(diffImg)                                 $\triangleright$  Jumlah lebar tepi
9:   totalEdges  $\leftarrow$  count_edges(edgeImg)                       $\triangleright$  Jumlah seluruh tepi
10:  score  $\leftarrow$  score(blurMetric, totalEdges)                   $\triangleright$  Menghitung skor
11:  score  $\leftarrow$  limit_round_score(score)                          $\triangleright$  Limit dan bulatkan skor
12:  return score
13: END function
```

Metrik *blur* juga dapat bergantung pada parameter yang digunakan untuk proses deteksi tepi dan dapat menghasilkan banyak *noise* atau tepi palsu. Oleh karena itu, perlu dilakukan analisis yang lebih komprehensif dan pemilihan metode deteksi tepi yang tepat untuk memperoleh metrik blur yang akurat dan konsisten pada berbagai jenis gambar. Dalam praktiknya, sebaiknya metrik blur digunakan bersama dengan metode pengukuran kualitas gambar lainnya untuk memperoleh hasil yang lebih akurat dan konsisten. Dengan mempertimbangkan faktor-faktor yang dapat mempengaruhi hasil deteksi tepi dan metrik blur, dapat ditemukan teknik-teknik yang lebih canggih dan adaptif untuk pengukuran tingkat blur pada gambar.

4.2.3 Pengujian metrik *temporal*

Pada metrik temporal pengukuran dilakukan terhadap beberapa frame dalam selang waktu tertentu. Metrik ini mengukur tingkat pembekuan (*freeze*) yang terjadi pada frame citra dengan cara menghitung rasio antara jumlah frame membeku terhadap seluruh frame yang diambil. Dalam implementasinya, frame yang dianggap membeku biasanya didefinisikan dengan perubahan piksel yang sangat kecil atau bahkan tidak ada perubahan sama sekali dalam interval waktu tertentu. Hal ini dapat dicapai dengan membandingkan nilai piksel dari dua frame berurutan dan menghitung nilai MSE atau nilai perubahan piksel yang ada. Jika nilai perubahan piksel kurang dari atau sama dengan threshold tertentu, maka frame tersebut dianggap membeku.



Gambar 4.11: Pengukuran metrik temporal *freeze* dengan MSE skor

Parameter utama yang digunakan pada pengukuran metrik temporal adalah membandingkan dua frame citra saat ini (i) dan sebelumnya (i-1) menggunakan pengukuran *mean square error* (MSE). Nilai maksimal dari MSE bergantung pada ukuran gambar yang digunakan. Semakin besar ukuran gam-

bar, semakin besar nilai maksimal dari MSE. Pada penelitian ini gambar yang digunakan menggunakan ukuran 1920×1080 yang artinya nilai maksimal dari MSE nya bisa mencapai $1920 \times 1080 \times 3 = 6220800$ dengan menghitung masing-masing kanal RGB. Sedangkan nilai terkecilnya adalah 0 dengan kondisi kedua citra yang diukur sama persis. Ditunjukkan gambar 4.11 terdapat empat citra berbeda A, B, C, D. Nilai MSE citra A dan B , sedangkan nilai MSE antara citra A dan C adalah besar. Digunakan treshold untuk menetukan apakah gambar mengalami pembekuan dengan nilai MSE kurang dari sama dengan 1, berdasarkan referensi (Huynh-Thu & Gha 2009) dan juga pengujian metrik pada beberapa frame.

Algoritma 4.4 Metrik pengukuran blur

INPUT: video

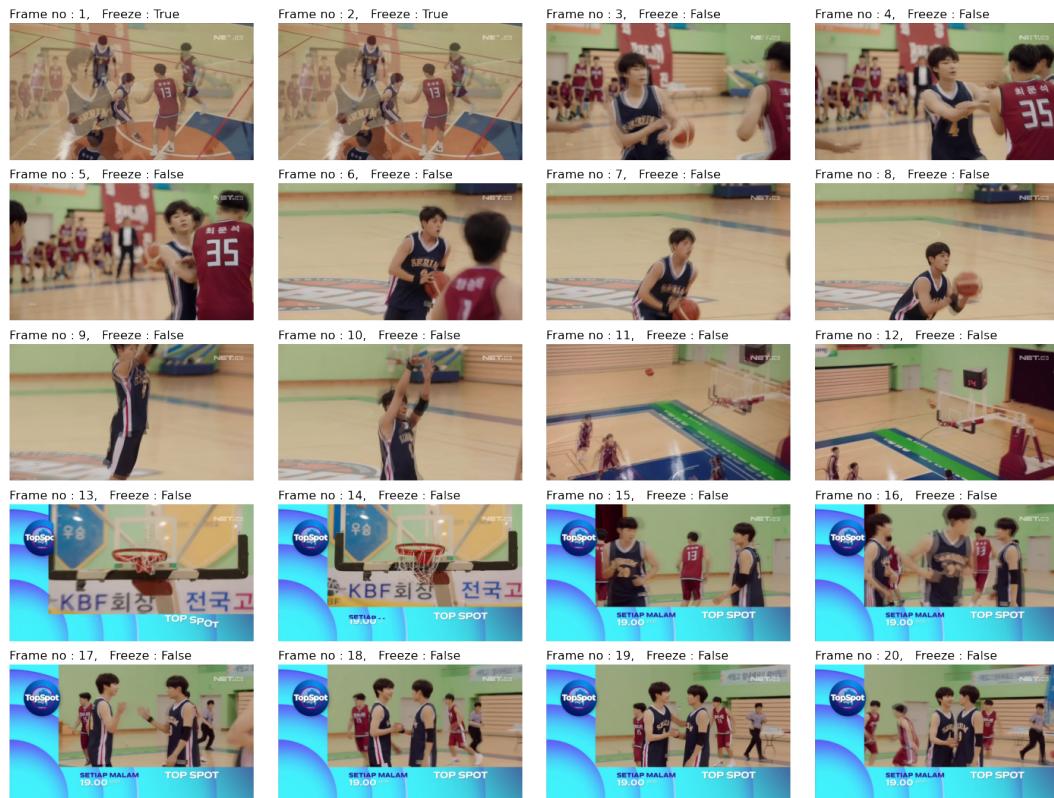
OUTPUT: temporal_score

```

1: function TEMPORAL_METRIC( $image_{path_i}, image_{path_{i-1}}$ )
2:    $list\_temporal \leftarrow []$ 
3:   for  $i = 1$  to  $20$  do
4:      $image_i \leftarrow$  Baca gambar ke- $i$  dari  $image\_path_i$ 
5:      $image_{i-1} \leftarrow$  Baca gambar ke- $(i-1)$  dari  $image\_path_{i-1}$ 
6:      $mse \leftarrow$  Hitung nilai MSE antara  $image_i$  dan  $image_{i-1}$ 
7:     IF  $mse \leq 1$  THEN
8:       Tambahkan nilai 1 ke  $list\_temporal$ 
9:     ELSE
10:      Tambahkan nilai 0 ke  $list\_temporal$ 
11:    END IF
12:   END for
13:    $score \leftarrow sum(list\_temporal)/len(list\_temporal)$ 
14:    $score \leftarrow round(1 - score) * 100$ 
15:   return  $score$ 
16: END function
```

Algoritma metrik temporal pembekuan frame ini selanjutnya diterapkan pada video. Dari video dengan durasi 10 detik, diambil 20 frame citra, selanjutnya dilakukan pengukuran terhadap masing-masing citra dibandingkan secara MSE terhadap citra pada frame sebelumnya. Hasilnya akan dicatat kemudian dihitung rata-rata e yang mengalami pembekuan terhadap jumlah

seluruh fram yang diambil. Program dituliskan dalam bahasa python seperti yang ditunjukkan pada algoritma pseudocode 4.4. Pada algoritma metrik tersebut, semakin besar skornya maka menggambarkan kualitasnya video yang semakin baik dengan interval skor 0-100.



Gambar 4.12: Contoh pengukuran metrik obyektif temporal pada video *dataset* no.84 siaran televisi DVB-T, temporal skor : 90

Pada gambar 4.12 didapat kemiripan 2 frame dari 20 frame sehingga metrik temporal memberikan skor 90. Sedangkan pada gambar 4.13 didapat kemiripan frame sebanyak 7 frame, sehingga metrik temporal memberikan skor 65. Semakin banyak frame yang mirip/sama dalam satu video dengan MSE dibawah *threshold* maka akan semakin kecil skor dari metrik temporalnya.

Tabel 4.3: Waktu yang dibutuhkan untuk memproses satu video dalam satuan detik

No.	Nama dataset	A Nvidia T4 (Google-colab)	B Nvidia A100 (UG)	Perbandingan A/B
1	001.mp4	3.8352	2.0829	1.8413
2	007.mp4	3.3550	1.7997	1.8642
3	015.mp4	3.9755	2.1446	1.8537
4	074.mp4	4.8772	2.2676	2.1508
5	084.mp4	4.4204	1.9829	2.2293
6	097.mp4	4.5780	1.9934	2.2966
7	099.mp4	4.6574	2.0846	2.2342
Rata-rata				2.0671

Percobaan dilakukan dengan menggunakan dua mesin GPU untuk mendapat perbedaan waktu proses seperti yang ditunjukkan pada tabel 4.3 Pengujian menggunakan beberapa video untuk menghitung perbandingan skor metrik temporal dengan pengambilan citra sebanyak 20 frame. Dihitung perbandingan antara GPU A yaitu Google colab Nvidia T4 terhadap GPU B milik Universitas Gunadarma NVIDIA A100. Diperoleh waktu proses yang dibutuhkan GPU A 2.0671 kali lebih cepat dari GPU B.



Gambar 4.13: Contoh pengukuran metrik obyektif temporal pada video *dataset* no.15 siaran televisi DVB-T, temporal skor : 65

Salah satu kekurangan dari metrik temporal freeze adalah kurang mampu menggambarkan kualitas video secara keseluruhan. Metrik ini hanya mampu mendeteksi apakah terjadi freeze pada suatu waktu tertentu dalam video, namun tidak dapat memberikan informasi tentang kualitas video secara keseluruhan atau apakah terdapat jenis distorsi lain seperti blur atau bloking. Selain itu, metrik ini juga dapat memberikan hasil yang tidak akurat jika terdapat frame duplikat, ketidakstabilan dalam pengambilan gambar, atau frame berupa berita yang hanya menampilkan *slide statis*. Oleh karena itu, metrik ini sebaiknya digunakan bersamaan dengan metrik lainnya untuk mendapatkan hasil yang lebih akurat dan lengkap tentang kualitas video.

4.2.4 Pengukuran obyektif pada keseluruhan *dataset*

Pengukuran obyektif dilakukan pada seluruh dataset yang pada tahapan selanjutnya akan digunakan untuk melakukan pengukuran subyektif. Ketiga metrik obyektif yaitu blok, blur, dan temporal akan diterapkan pada seluruh video untuk memperoleh nilai obyektif dari masing-masing video yang akan digunakan sebagai nilai masukan pada jaringan saraf tiruan NN. Pengukuran dilakukan menggunakan dua super komputer berbeda untuk mendapatkan waktu proses dan kesamaan nilai ukur dari setiap metrik. Pada tabel 4.5 ditunjukkan sebagian pengukuran yang dilakukan pada dataset. Selanjutnya hasil pengukuran disimpan ke dalam file.csv untuk dilakukan proses selanjutnya ke pengukuran subyektif dan pembuatan model NN.

Tabel 4.4: Pengukuran metrik obyektif terhadap dataset

Nama video dataset	skor metrik blok	skor metrik bluk	skor metrik temporal
000.mp4	87	39	79
001.mp4	87	79	80
002.mp4	73	20	90
003.mp4	77	49	90
004.mp4	84	80	85
005.mp4	74	50	90
...
100.mp4	74	50	90
101.mp4	80	74	85
102.mp4	94	38	100
103.mp4	78	52	85
104.mp4	89	60	75

4.3 Hasil pengambilan data pengukuran subyektif

Pengukuran subjektif kualitas video merupakan langkah penting dalam menilai sejauh mana video memenuhi harapan penonton dan memberikan pengalaman visual yang baik. Proses ini melibatkan penilaian kualitas video berdasarkan persepsi manusia, yang mencakup berbagai faktor seperti kejelasan, kehalusan gerakan, warna, kontras, dan artefak kompresi. Evaluasi kualitas video ini sangat berguna untuk memastikan bahwa video sesuai dengan standar kualitas yang diharapkan oleh penonton dan industri.

Selain itu, pengukuran subjektif juga berperan dalam validasi metrik obyektif yang dikembangkan untuk mengukur kualitas video secara otomatis. Penilaian subjektif digunakan sebagai *ground truth* untuk menguji dan men-goptimalkan metrik obyektif, sehingga mereka dapat lebih akurat dan efektif dalam menggantikan penilaian manusia. Memvalidasi metrik obyektif melalui pengukuran subjektif memungkinkan peneliti dan pengembang untuk menciptakan algoritma penilaian kualitas video yang lebih efisien dan handal.

4.3.1 Material pengukuran subyektif

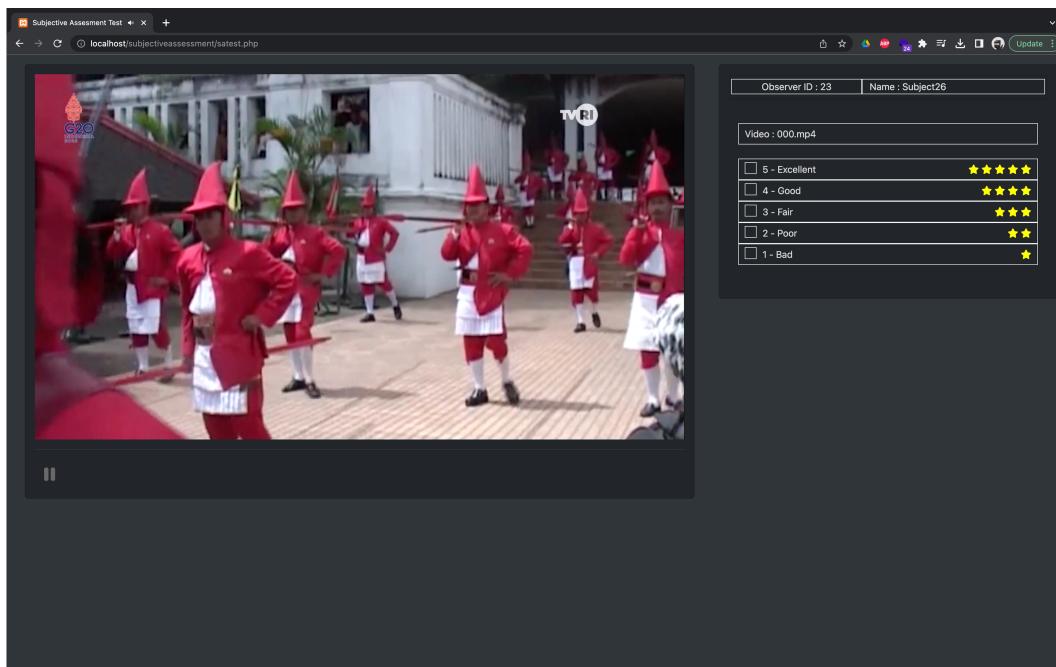
Pada penelitian ini material yang akan digunakan untuk pengujian adalah dataset berupa video dari siaran DVB-T2 sebanyak 105 video. Pada tahap sebelumnya sudah dilakukan pengukuran menggunakan metrik obyektif spatial dan temporal pada video tersebut. Terdapat tiga metrik obyektif yang sudah diterapkan dan mendapatkan skornya masing-masing untuk seluruh video. Di pengujian kali ini akan dilakukan pengukuran secara subyektif pada video tersebut. Tujuan dari pengukuran subyektif pada kualitas video adalah untuk mendapatkan penilaian kualitas video dari sudut pandang manusia, dengan melibatkan partisipasi pengamat manusia untuk memberikan penilaian secara langsung terhadap video yang ditampilkan.

4.3.2 Perangkat yang digunakan

Perangkat yang digunakan untuk melakukan pengukuran subyektif berupa perangkat keras laptop Macbook Pro 2019 hexa-core i7 RAM DDR4 16GB dan perangkat lunak yang dibuat secara mandiri. Perangkat lunak berupa aplikasi web yang menggunakan *native javascript*, css dan html yang terinstall secara lokal pada laptop. Seluruh materi video yang akan diujian dimasukan ke dalam aplikasi. Selanjutnya pengujian akan dilakukan menggunakan laptop tersebut satu persatu oleh panelis/partisipan.

4.3.3 Proses pengukuran subyektif

Pengukuran subyektif atau (*subjective assessment*) yang dilakukan pada penelitian ini menggunakan standar pengukuran ITU-R BT.500-14 tahun 2019 mengenai standarisasi prosedur untuk melakukan subyektif tes pada citra televisi. Pengukuran subyektif melibatkan 25 yang terdiri dari 21 laki-laki dan 4 perempuan. Kriteria dari panelis yaitu rentang usia 20 - 40 tahun dikarenakan berdasarkan data dari organisasi survei Nielsen diperoleh jumlah penonton televisi terbanyak di Indonesia yaitu 41% berada pada rentang umur tersebut (Annur 2023). Kriteria lainnya yaitu pernah menonton televisi digital dan tidak memiliki gangguan mata. Selanjutnya masing-masing panelis akan memberikan penilaian terhadap keseluruhan dataaset sebanyak 105 video menggunakan aplikasi yang sudah dibuat.



Gambar 4.14: Tampilan aplikasi pengukuran subyektif kualitas citra video siaran DVB-T2

Metode yang digunakan untuk pelakuan pengukuran subyektif pada penelitian ini adalah *single stimulus* (SS). Panelis akan duduk di depan laptop dengan kondisi *home-environment*. Kemudian panelis menjalankan aplikasi tes, melakukan latihan dan pengarahan, selanjutkan masuk kebagian sesi utama tes. Pada tes ini setiap video dengan durasi 10 detik akan diputar dengan jeda 4 detik antar videonya. Sehingga untuk menyelesaikan 105 video, panelis membutuhkan waktu sekitar 24.5 menit. Panelis harus memberikan penilaian pada setiap video dengan skor 1 sampai 5. Skor tersebut menggambarkan kualitas video dengan skor 1 artinya sangat buruk dan skor 5 berarti sangat baik dari sisi kualitas citranya. Tampilan dari sesi utama tes ditunjukkan pada gambar 4.14.

The screenshot shows a web-based administrative dashboard titled "Dashboard". On the left, there is a sidebar with navigation links: "Dashboard" (which is active and highlighted in blue), "Setting & Input", "User", and "Pluge test". The main content area is titled "Dashboard" and contains a table with the following data:

Video	Total Data	Average Rating (0 - 5)	Percentage Rating (0% - 100%)	Info
000.mp4	20	3.90	78.00 %	<button>Detail</button>
001.mp4	20	4.60	92.00 %	<button>Detail</button>
002.mp4	20	2.35	47.00 %	<button>Detail</button>
003.mp4	20	1.60	32.00 %	<button>Detail</button>
004.mp4	20	1.85	37.00 %	<button>Detail</button>
005.mp4	20	2.10	42.00 %	<button>Detail</button>
006.mp4	20	1.50	30.00 %	<button>Detail</button>
007.mp4	20	1.00	20.00 %	<button>Detail</button>
008.mp4	20	1.70	34.00 %	<button>Detail</button>
009.mp4	20	4.10	82.00 %	<button>Detail</button>
010.mp4	20	4.30	86.00 %	<button>Detail</button>
011.mp4	20	3.30	66.00 %	<button>Detail</button>
012.mp4	20	1.60	32.00 %	<button>Detail</button>
013.mp4	20	4.35	87.00 %	<button>Detail</button>
014.mp4	20	3.95	79.00 %	<button>Detail</button>

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

Gambar 4.15: Tampilan dashboard aplikasi pengukuran subyektif dari sisi administrator

Setelah panelis melakukan tes, maka data dapat diakses oleh administrator web pada bagian dashboard. Ditampilkan jumlah peserta yang sudah memerlukan penilaian dan skor rata-rata per video seperti yang ditunjukkan pada gambar 4.15. Data juga bisa di download dengan format .xls untuk selanjutnya diolah.

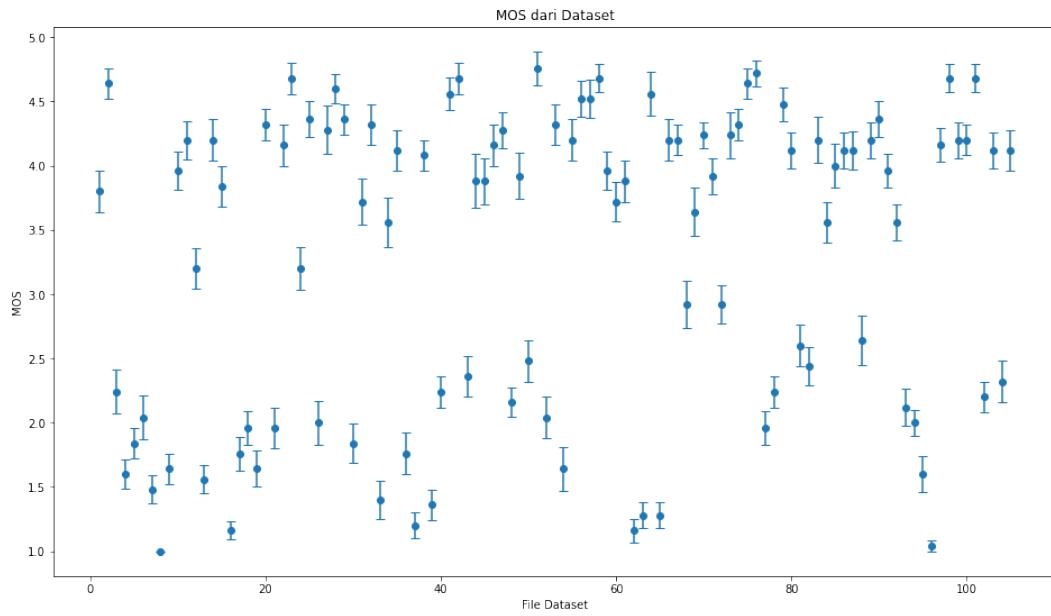
4.3.4 Hasil pengukuran subyektif

Setelah data penilaian subjektif terkumpul dari panelis, beberapa langkah pengolahan data perlu dilakukan untuk menghitung hasil pengukuran subjektif dan menganalisis hasil tersebut. Seperti yang ditunjukkan pada tabel ?? langkah yang perlu diambil meliputi menghitung nilai rata-rata (Mean Opinion Score/MOS) untuk setiap klip video, menghitung deviasi standar dan interval kepercayaan 95%.

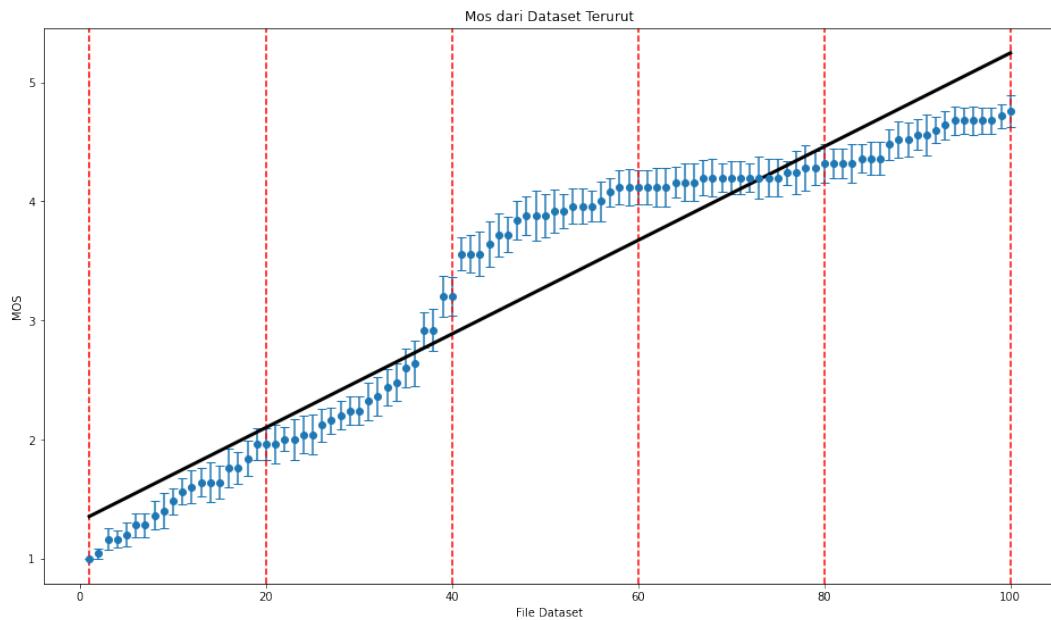
Tabel 4.5: Hasil pengukuran subyektif terhadap dataset

Nama	MOS	Stdev	Conf_Int	Batas_bawah	Batas_atas
000.mp4	3.80	0.82	0.16	3.64	3.96
001.mp4	4.64	0.64	0.12	4.52	4.76
002.mp4	2.24	0.88	0.17	2.07	2.41
003.mp4	1.60	0.58	0.11	1.49	1.71
004.mp4	1.84	0.62	0.12	1.72	1.96
005.mp4	2.04	0.89	0.17	1.87	2.21
006.mp4	1.48	0.59	0.11	1.37	1.59
007.mp4	1.00	0.00	0.00	1.00	1.00
008.mp4	1.64	0.64	0.12	1.52	1.76
009.mp4	3.96	0.79	0.15	3.81	4.11
010.mp4	4.20	0.76	0.15	4.05	4.35
...
091.mp4	3.56	0.71	0.14	3.42	3.70
092.mp4	2.12	0.73	0.14	1.98	2.26
093.mp4	2.00	0.50	0.10	1.90	2.10
094.mp4	1.60	0.71	0.14	1.46	1.74
095.mp4	1.04	0.20	0.04	1.00	1.08
096.mp4	4.16	0.69	0.13	4.03	4.29
097.mp4	4.68	0.56	0.11	4.57	4.79
098.mp4	4.20	0.71	0.14	4.06	4.34
099.mp4	4.20	0.65	0.12	4.08	4.32
100.mp4	4.68	0.56	0.11	4.57	4.79
101.mp4	2.20	0.65	0.12	2.08	2.32
102.mp4	4.12	0.73	0.14	3.98	4.26
103.mp4	2.32	0.85	0.16	2.16	2.48
104.mp4	4.12	0.83	0.16	3.96	4.28

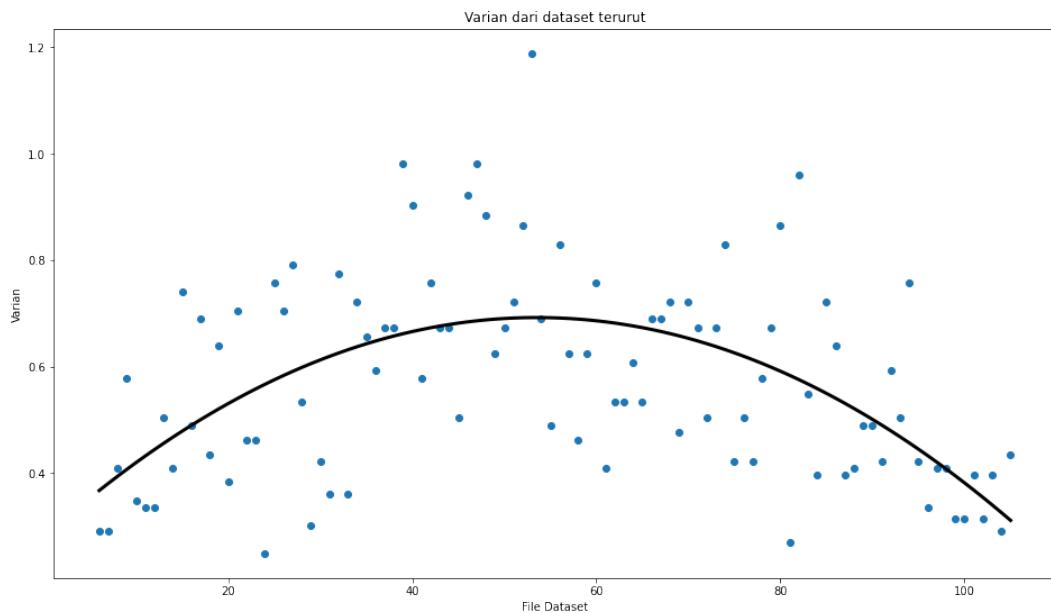
Pengukuran dilakukan pada masing-masing video, MOS menunjukkan nilai rata-rata dari seluruh panelis yang melakukan tes. Semakin besar nilai dari MOS menunjukkan semakin bagus kualitas video tersebut. Standar deviasi dari data (Stdev) menunjukkan variasi panelis dalam memberikan nilai, semakin besar stdev-nya, maka semakin bervariasi panelis dalam memberikan penilaian. Dilakukan juga pengukuran interval kepercayaan 95% untuk memberikan nilai batas bawah dan batas atas rata-rata dari populasi sebenarnya.



Gambar 4.16: Sebaran MOS pengukuran tes subyektif dari dataset video



Gambar 4.17: Pembagian kelas MOS pengukuran subyektif dataset



Gambar 4.18: Varian pengukuran subyektif dataset dengan skor terurut

Berdasarkan hasil *scatter plot* dari gambar 4.16 yang menunjukkan sebaran data dari pengukuran subyektif, diperoleh lebih banyak skor MOS pengukuran terdapat pada rentang 3.5 sampai 4.5. Kemudian data diurutkan dari skor MOS terkecil hingga terbesar seperti yang ditunjukkan pada gambar 4.17. Jika data dipisahkan menjadi 5 kelas berbeda sesuai dengan penilaian pada standar ITU-R BT.500-14 maka diperoleh batasan seperti pada tabel 4.6. Batasan tersebut yang digunakan sebagai acuan untuk menentukan batasan skor pada pengukuran secara waktu nyata. Selanjutnya dilakukan pengukuran terhadap variansi dari masing-masing data. Seperti yang ditunjukkan pada gambar 4.18 nilai variansi tertinggi berada pada data urutan ke 40-60. Hal tersebut berarti banyak varian panelis pada rentang tersebut yaitu pada kelas klasifikasi 3.

Tabel 4.6: Batasan masing-masing kelas berdasarkan pengukuran subyektif

Kelas	Jumlah data	Batas bawah	Batas atas	Interval batas
1	20	1.00	1.96	0.96
2	20	1.96	3.20	1.06
3	20	3.20	4.12	0.92
4	20	4.12	4.32	0.20
5	20	4.32	5.00	0.68

Berdasarkan hasil dari pengukuran subyektif kemudian dibagi menjadi 5 kelas dengan jumlah data yang sama. Dikarenakan distribusi yang tidak merata maka interval antar kelas berbeda-beda. Interval tertinggi berada pada kelas 2, dan interval terendah berada pada kelas 4. Selanjutnya data hasil pengukuran subyektif ini akan digunakan sebagai parameter luaran pada NN.

4.4 Pengujian dan hasil pada NN

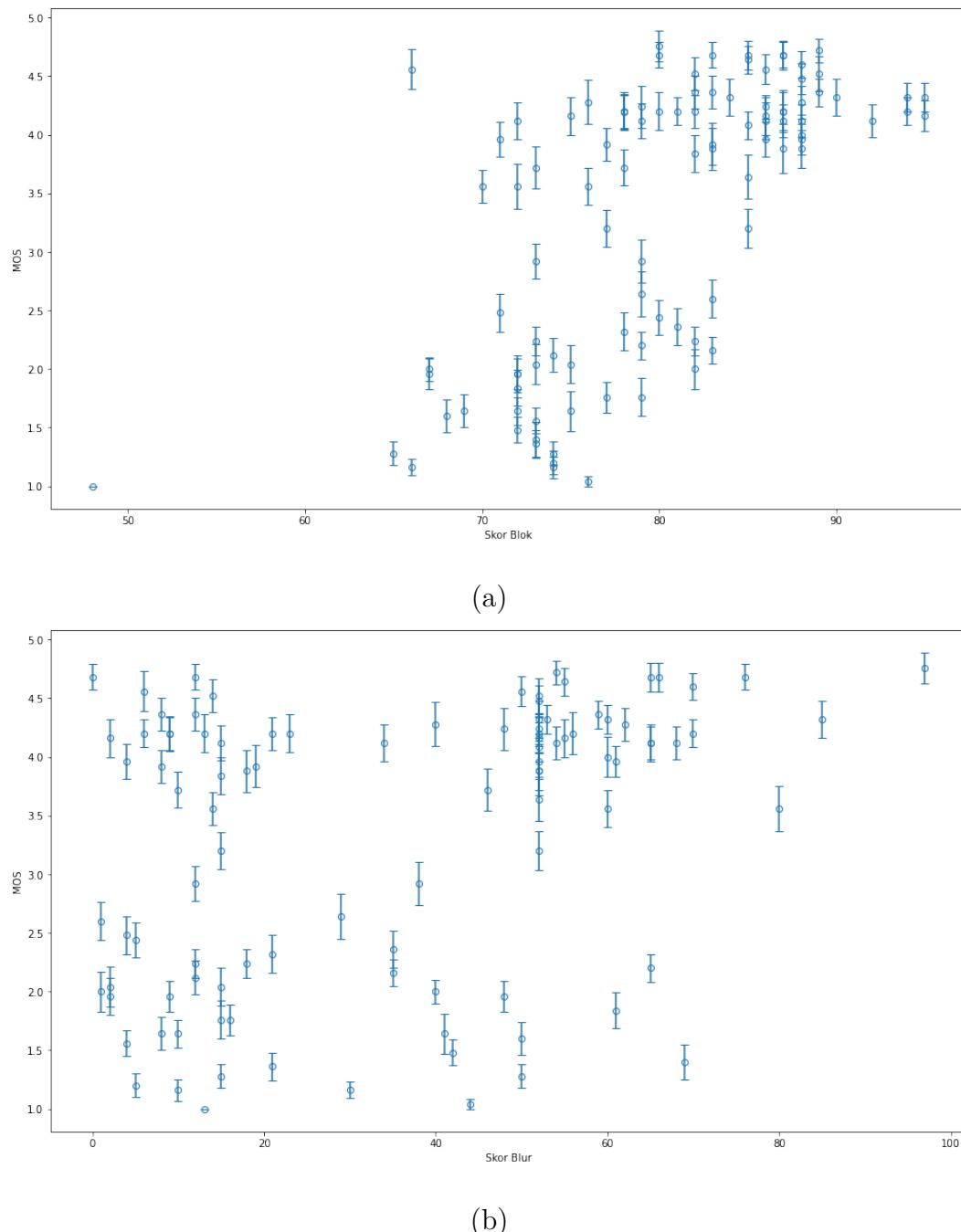
Pengujian yang dilakukan dalam membentuk model NN adalah akurasi dan korelasi dari model tersebut. Luaran model NN yang dibentuk yaitu untuk menentukan kualitas suatu video pada siaran DVB-T2. Terdapat 100 dataset yang sudah di label menggunakan hasil pengukuran subyktif. Parameter tersebut berupa luaran dari NN yang akan dibentuk. Terdapat juga tiga parameter masukan pada NN yaitu metrik blok, metrik blur dan metrik temporal freeze. Masukan dan luaran pada NN tersebut buat skor dari metrik dengan obyek video yang sudah diukur, bukan videonya secara langsung. Pengujian dilakukan menggunakan beberapa jenis NN, seperti artificial Neural Netrwok (ANN), atau recurrent neural network (RNN).

4.4.1 Data masukan dan luaran model

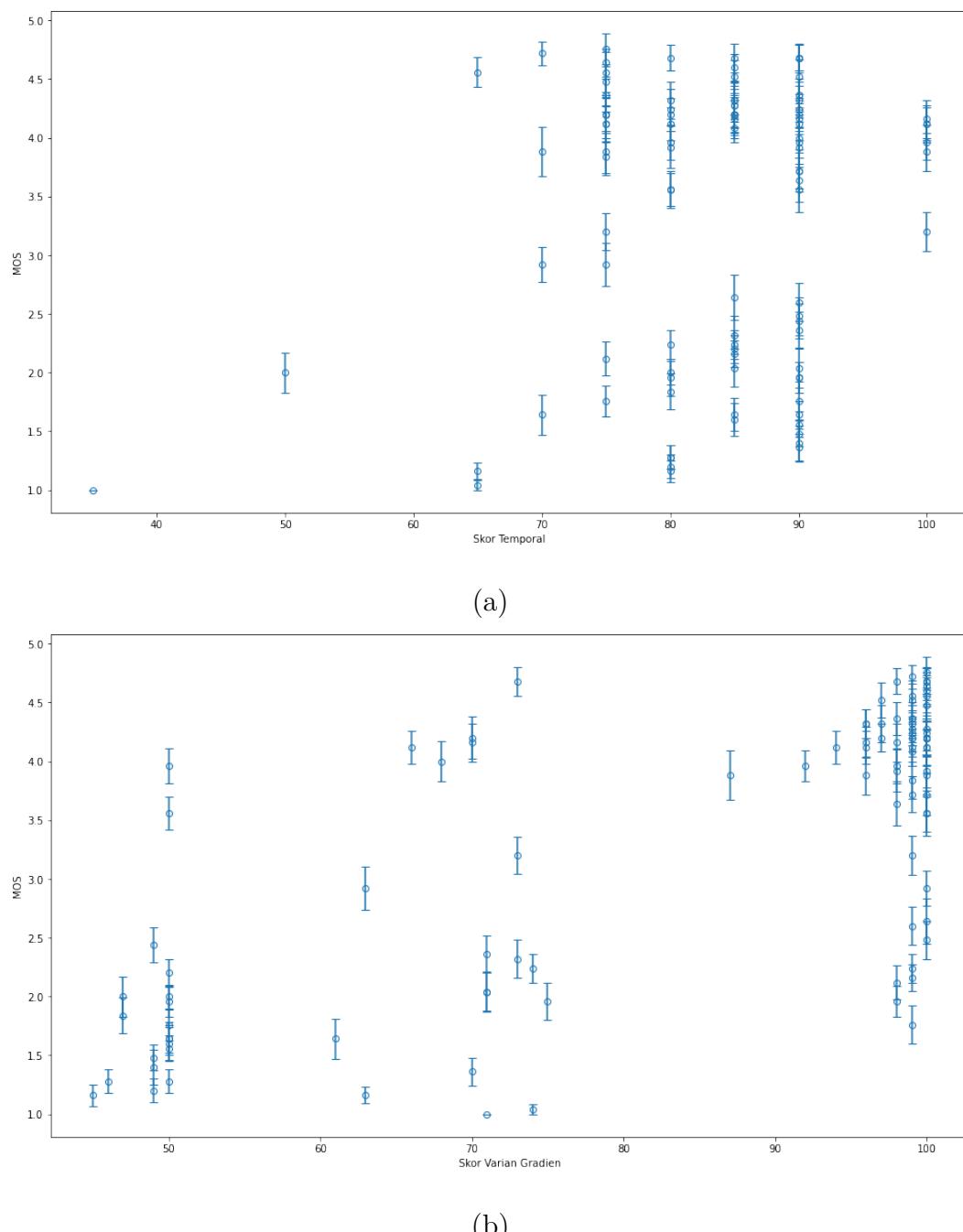
Terdapat 100 dataset berupa skor dari pengukuran menggunakan metrik obyektif pada video siaran DVB-T2. Luaran yang ingin dicapai adalah kemiripan dengan MOS hasil pengukuran subyektif yang sudah dilakukan oleh panelis pada 100 video. Pada tabel 4.7 ditunjukkan sebagian hasil pengukuran metrik obyektif dan juga skor MOS pada pengukuran subyektif.

Tabel 4.7: Hasil pengukuran subyektif terhadap dataset

nama	m_blok	m_blur	m_grad	std_grad	move_fr	m_temp	mos	class
000.mp4	87	52	4.84	4.76	95	75	3.8	3
001.mp4	88	55	0.29	0.25	100	75	4.64	5
002.mp4	73	1	-0.26	0.12	50	90	2.24	2
003.mp4	79	8	3.85	8.5	71	90	1.6	1
004.mp4	80	10	0.33	4.94	75	85	1.84	1
005.mp4	73	2	3.67	8.12	71	90	2.04	2
006.mp4	72	42	-0.51	0.59	49	90	1.48	1
007.mp4	48	13	3.9	9.97	71	35	1	1
008.mp4	72	10	-0.35	0.63	50	90	1.64	1
009.mp4	86	52	1.63	0.78	98	80	3.96	3
010.mp4	78	9	0	0.31	100	85	4.2	4
...
091.mp4	70	14	-0.25	0.51	50	80	3.56	3
092.mp4	74	12	2.21	2.6	98	75	2.12	2
093.mp4	67	40	-0.28	0.59	50	80	2	2
094.mp4	68	50	-0.23	0.38	50	85	1.6	1
095.mp4	76	44	1.3	2.92	74	65	1.04	1
096.mp4	95	52	4	1.57	96	90	4.16	4
097.mp4	80	0	0.29	0.46	100	90	4.68	5
098.mp4	82	9	0.1	0.26	100	75	4.2	4
099.mp4	94	70	2.68	1.82	97	90	4.2	4
100.mp4	87	76	2.12	2.95	98	90	4.68	5
101.mp4	79	65	-0.47	0.58	50	85	2.2	2
102.mp4	92	52	9.33	16.3	66	100	4.12	4
103.mp4	78	21	1.86	7.08	73	85	2.32	2
104.mp4	86	52	1.36	2.3	99	75	4.12	4



Gambar 4.19: Diagram scatter pengukuran metrik spatial terhadap MOS pada pengukuran subyektif (a) metrik blok, (b) metrik blur.



Gambar 4.20: Diagram scatter pengukuran metrik temporal terhadap MOS pada pengukuran subyektif (a) metrik freeze, (b) metrik variansi gradien .

4.4.2 Pengujian arsitektur NN

Pengujian untuk mencari arsitektur NN yang sesuai dengan menggunakan tiga metode. Metode pertama adalah dengan menggunakan regresi multivariable, kedua menggunakan ANN tanpa klasifikasi kelas, yang ketiga dengan menggunakan ANN dan RNN dengan klasifikasi. Ketiga metode tersebut dilakukan untuk mencari tingkat akurasi dan korelasi paling tinggi, serta waktu paling cepat dalam menentukan nilai dari data baru.

4.4.2.1 Hasil regresi multivariabel

Regresi multivariabel digunakan untuk mencari bobot dari setiap variabel masukan yang digunakan untuk menentukan nilai dari luaran. Akurasi dan korelasi dari prediksi regresi multivariabel dihitung terhadap hasil luaran yang diperoleh dari skor MOS pada pengukuran subyektif. Pengujian dilakukan menggunakan dataset dan memecahnya menjadi 4:1 untuk data latih dan data tes. Pada tabel 4.8 ditunjukkan hasil pengujian yang dilakukan untuk mendapatkan persamaan dari model regresi multivariabel.

Tabel 4.8: Hasil pengujian terhadap percobaan regresi multivariabel

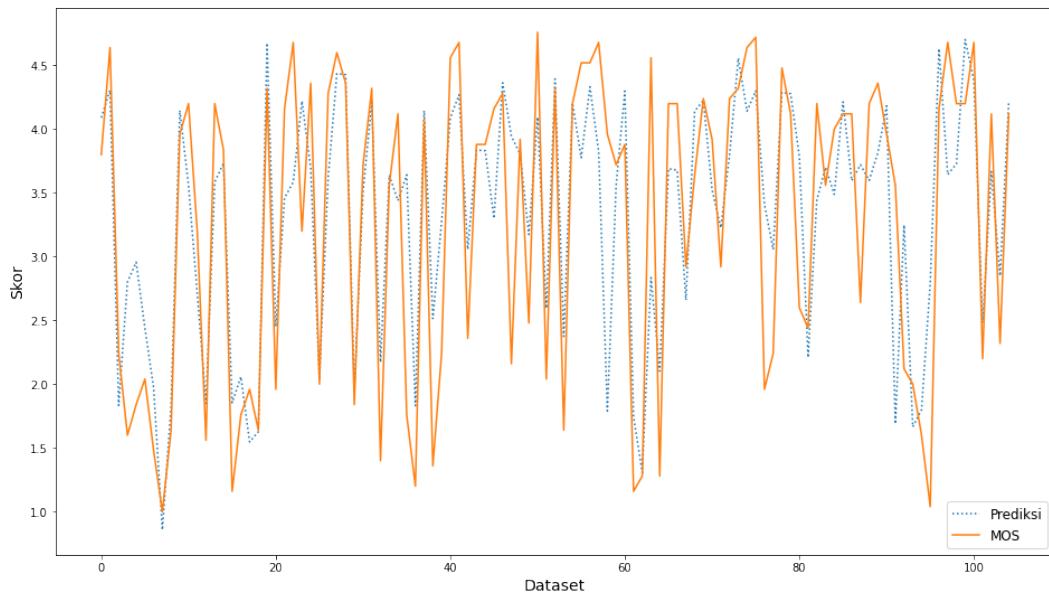
Random State	Data Tes					Data Total	
	R2	MSE	MAE	PLCC	SLCC	PLCC	SLCC
1	0.7983	0.3366	0.4682	0.9192	0.7922	0.8884	0.8194
2	0.8009	0.2686	0.4073	0.9164	0.8416	0.8478	0.8504
3	0.7696	0.4259	0.4816	0.9163	0.7923	0.8331	0.8241
4	0.7851	0.2876	0.4577	0.8971	0.7491	0.8167	0.7471
5	0.8352	0.2209	0.3930	0.9191	0.8152	0.8033	0.7588

Pada pembuatan model menggunakan regresi multivariabel dilakukan lima kali percobaan dengan mengacak antara data latih dan data tes. Berdasarkan tabel 4.8 diperoleh pada percobaan kedua memiliki nilai akurasi R2

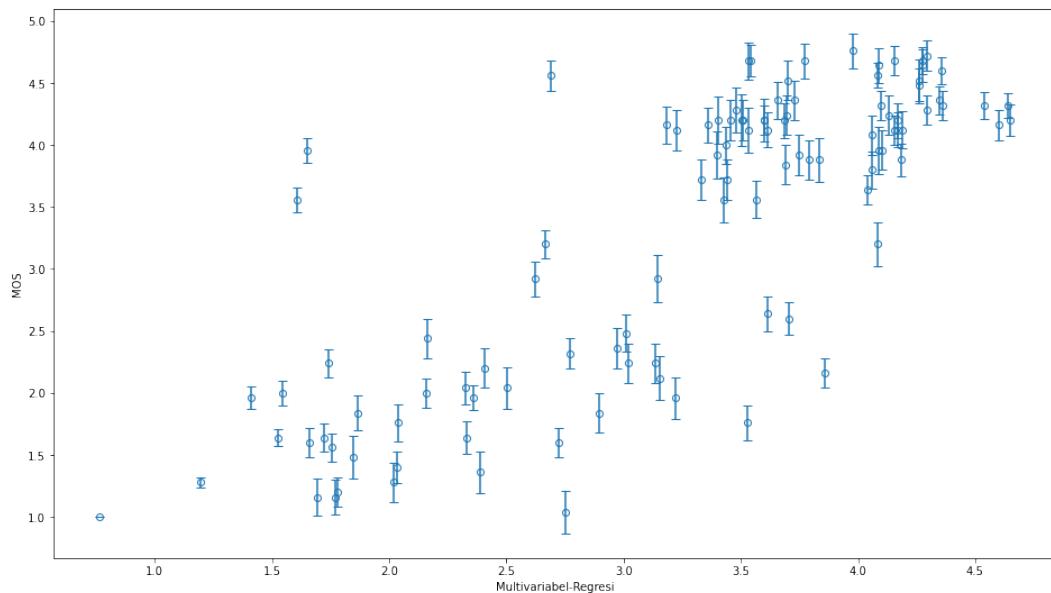
tertinggi dan nilai error terendah. Pengukuran terhadap metrik korelasinya pada percobaan kedua baik menggunakan metrik korelasi spearman ataupun korelasi pearson pada data tes dan total juga lebih tinggi dibandingkan percobaan lainnya. Sehingga dipilih model regresi multivabel yang akan digunakan adalah dari hasil percobaan kedua, dengan nilai bobot yaitu:

$$y = w_1 \cdot X_1 + w_2 \cdot X_2 + w_3 \cdot X_3 + w_4 \cdot X_4 - c \quad (4.1)$$

$$\begin{aligned} y = & 0.05474 \cdot X_1 + 0.00540 \cdot X_2 \\ & + 0.00498 \cdot X_3 + 0.02867 \cdot X_4 - 4.05 \end{aligned} \quad (4.2)$$



Gambar 4.21: Diagram hasil prediksi terhadap MOS



Gambar 4.22: Diagram Scater hasil prediksi terhadap MOS

4.4.2.2 Hasil ANN tanpa klasifikasi

4.4.2.3 Hasil ANN dengan klasifikasi

4.5 Aplikasi pemantauan kualitas gambar siaran DVB-T2

Hasil pengukuran secara keseluruhan sistem ditampilkan pada sebuah aplikasi pemantauan kualitas gambar siaran DVB-T2. Aplikasi berupa situs jejaring atau *website* yang menampilkan hasil pengukuran secara waktu nyata terhadap kualitas gambar siaran dan juga kualitas sinyal yang diterima di lokasi pengukuran. Hasil pengukuran kualitas gambar berupa skor 1 sampai 5 yang ditampilkan dalam grafik

4.5.1 Arsitektur aplikasi

4.5.2 Pengujian dan implementasi aplikasi

BAB V

PENUTUP

5.1 Kesimpulan

5.2 Saran

DAFTAR PUSTAKA

- Abid Haleem, Mohd Javaid, M. A. Q. R. S. (2022), ‘Understanding the role of digital technologies in education: A review’, **3**, 275–285.
- Akoglu, H. (2018), ‘User's guide to correlation coefficients’, *Turkish Journal of Emergency Medicine* **18**(3), 91–93.
- Altunian, G. (2021), ‘What Is Signal-to-Noise Ratio and Why Does It Matter?’, URL: <https://www.lifewire.com/signal-to-noise-ratio-3134701>. (Diakses tanggal 07 Juli 2022).
- Annur, C. M. (2023), ‘Demografi Pengguna Televisi di Indonesia Berdasarkan Usia (2022)’, URL: <https://databoks.katadata.co.id/datapublish/2022/12/09/survei-nielsen-indonesia-majoritas-pengguna-television-di-ri-berusia-50-tahun-ke-atas>. (diakses tanggal 20 Januari 2023).
- Apple-Inc. (2012), ‘Compressor 4 - User Manual, MPEG-2 Reference Information’, URL: <https://help.apple.com/compressor/mac/4.0/en/compressor/usermanual/#chapter=18%26section=5>. (Diakses tanggal 25 April 2021).
- Baek, S. & Kim, B. J. (2008), Digital Multimedia Broadcasting (DMB) in Korea, in ‘Multimedia Technologies’, IGI Global, pp. 1377–1390.
- Centauri, A. (2013), ‘Why Decibels are used to measure Signal to Noise Ratio?’, URL: <https://electronics.stackexchange.com/q/82823>. (Diakses tanggal 21 Juli 2022).

- Cheon, M. & Lee, J.-S. (2016), ‘On ambiguity of objective image quality assessment’, *Electronics Letters* **52**(1), 34–35.
- Cheon, M., Vigier, T., Krasula, L., Lee, J., Callet, P. L. & Lee, J.-S. (2021), ‘Ambiguity of objective image quality metrics: A new methodology for performance evaluation’, *Signal Processing: Image Communication* **93**, 116150.
- Chochliouros, I. (2009), ‘Digital Video Broadcasting (DVB) Evolution, Encyclopedia of Multimedia Technology and Networking, Second Edition’, *IGI Global Science and Technology*.
- Dai, L., Wang, Z. & Yang, Z. (2012), ‘Next-generation digital television terrestrial broadcasting systems: Key technologies and research trends’, *IEEE Communications Magazine* **50**(6), 150–158.
- Dihin, R. A., Hamza, N. R. & Toman, Z. H. (2020), ‘Full-Reference Facial Image Quality Assessment and Identification by Two Proposed Measures’, *Journal of Southwest Jiaotong University* **55**(2), 1.
- Ding, K., Ma, K., Wang, S. & Simoncelli, E. P. (2021), ‘Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems’, *International Journal of Computer Vision* **129**(4), 1258–1281.
- Dost, S., Saud, F., Shabbir, M., Khan, M. G., Shahid, M. & Lovstrom, B. (2022), ‘Reduced reference image and video quality assessments: review of methods’, *EURASIP Journal on Image and Video Processing* **2022**(1).
- DTVstatus (2017), ‘DTV Status Compares The Worldwide Digital Broadcasting Systems ATSC, DTMB, DVB-T/DVB-T2, and ISDB-T’, URL: <http://en.dtvstatus.net/>. (Diakses tanggal 27 Juni 2021).
- DVBProject (2013), Digital Video Broadcasting (DVB) Bluebook A116-2, in ‘Digital Video Broadcasting’, Digital Video Broadcasting.

- Ebrahimi, T. & Pereira, F. (2002), *The MPEG-4 Book*, illustrated edition edn, Prentice Hall PTR.
- El-Saba, M. (2018), *Telecommunication Systems and Data Networks*, third edn.
- Emmet (2022), ‘Stream TV from your Raspberry Pi with Tvheadend’, URL: <https://pimylifeup.com/raspberry-pi-tvheadend/>. (Diakses tanggal April 2022).
- Ferzli, R. & Karam, L. (2009), ‘A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)’, *IEEE Transactions on Image Processing* **18**(4), 717–728.
- Fischer, W. (2010), *Digital Video and Audio Broadcasting Technology*, third edn, Springer Berlin Heidelberg.
- Foundation, T. (2015), ‘Tvheadend 4.0 User Guide’, URL: https://docs.tvheadend.org/about_project/. (Diakses tanggal 5 Oktober 2021).
- Frost, P. (2014), ‘How is signal strength calculated?’, Electrical Engineering Stack Exchange. URL: <https://electronics.stackexchange.com/q/98691> (version: 2014-02-04). (Diakses tanggal 30 Oktober 2022).
- Gultom, A. D. (2018), ‘Digitalisasi Penyiaran Televisi di Indonesia’, *Buletin Pos dan Telekomunikasi* **16**(2), 91.
- Havinal, R. (2020), ‘The role and potential of information technology in agricultural development’.
- Hendrickson, J. (2022), ‘How to Check Your Wi-Fi Signal Strength’, URL: <https://www.howtogeek.com/426642/how-to-check-your-wi-fi-signal-strength/>. (Diakses tanggal 30 Oktober 2022).

- Hoelzer, S. (2005), ‘MPEG-2 overview and MATLAB codec project’, URL: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/Lecture_Examples/Compression/mpeg2.pdf. (Diakses tanggal 25 April 2021).
- Hou, H.-A. (2010), ‘DVB-T2 technology overview - Industrial Technology Research Institute’, URL: https://www.ncc.gov.tw/chinese/files/10051/1661_2.pdf. (Diakses tanggal 2 Oktober 2021).
- Huynh-Thu, Q. & Gha, M. (2009), ‘No-reference temporal quality metric for video impaired by frame freezing artefacts’.
- ISO & IEC (2022), ‘13818-1:2022 Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems’, URL: <https://www.iso.org/obp/ui/#0023iso:std:83239:en>. (Diakses tanggal 25 Juli 2022).
- ITU-R BT500 (2019), *BT500-14: Methodology for the Subjective Assessment of the Quality of Television Pictures*, International Telecommunication Union - Radiocommunication, International Telecommunication Union - Radiocommunication.
- ITU-R P.910 (2022), ‘International telecommunication union - telecommunication standarization’.
- Jason Dedrick, Kenneth L. Kraemer, E. S. (2014), ‘Information technology and productivity in developed and developing countries’, pp. 97–122.
- Kennedy, G. & Davis, B. (2009), *Electronic Communication System, Fourth Edision*, 4 edn, Tata McGraw Hill Education Private Limited.
- Kessel, R. (2018), ‘Why isn’t signal-to-noise ratio expressed in a percentage instead of in dB?’.

- Kieser, R., Reynisson, P. & Mulligan, T. J. (2005), ‘Definition of signal-to-noise ratio and its critical role in split-beam measurements’, *ICES Journal of Marine Science* **62**(1), 123–130.
- Kominfo, H. (2022), ‘Update Perkembangan Analog Switch Off (ASO)’, URL: https://www.kominfo.go.id/content/detail/43905/siaran-pers-no-346hmkominfo082022-tentang-update-perkembangan-analog-switch-off-aso/0/siaran_pers. (Diakses tanggal 2 Oktober 2022).
- Kruger, L. G. & Guerrero, P. F. (2002), *Digital Television: An Overview*, Novinka Books.
- Kumar, S. & Prajapati, P. (2015), ‘A Review Paper on Image Quality Assessment Metrics’, *International Journal of Advance Research in Computer Science and Management* **2**, 129–132.
- Kusuma, T. M. (2005), A Perceptual-based Objective Quality Metric for Wireless Imaging, phdthesis, Curtin University of Technology.
- Lee, Y.-S. & Kook, J. (2019), ‘Integrated DVB-X2 Receiver Architecture with Common Acceleration Engine’, *Applied Sciences* **9**(19), 3983.
- Lundstrom, L.-I. (2006), *Understanding Digital Television*, Focal Press.
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A. & Doulamis, N. (2021), ‘Multiclass confusion matrix reduction method and its application on net promoter score classification problem’, *Technologies* **9**(4), 81.
- Martínez-Rach, M. O., Piñol, P., López, O. M., Malumbres, M. P., Oliver, J. & Calafate, C. T. (2014), ‘On the Performance of Video Quality Assessment Metrics under Different Compression and Packet Loss Scenarios’, *The Scientific World Journal* **2014**, 1–18.

- Marziliano, P., Dufaux, F., Winkler, S. & Ebrahimi, T. (2002), A no-reference perceptual blur metric, *in* ‘Proceedings. International Conference on Image Processing’, IEEE.
- Matthews, V. O., Idiaké, S. U., Noma-Osaghae, E. & Nwukor, F. (2018), ‘Comparative Analysis of the Effect of Distance on Signal Strength using Selected Communication Media Comparative Analysis of the Effect of Distance on Signal Strength using Selected Communication Media’, *International Journal of Advances in Scientific Research and Engineering* **4**(8), 82–91.
- M.H. Pinson, S. W. (2004), ‘A New Standardized Method for Objectively Measuring Video Quality’, *IEEE Transactions on Broadcasting* **50**(3), 312–322.
- Mittal, A., Moorthy, A. K. & Bovik, A. C. (2012), ‘No-reference image quality assessment in the spatial domain’, *IEEE Transactions on Image Processing* **21**(12), 4695–4708.
- Mustafa, W., Yazid, H., Jaafar, M., Zainal, M., Abdul Nasir, A. S. & Mazlan, N. (2017), ‘A Review of Image Quality Assessment (IQA): SNR, GCF, AD, NAE, PSNR, ME’, *Journal of Advanced Research in Computing and Applications* **7**, 1–7.
- Nguyen, H. H. & Shwedyk, E. (2009), *A First Course in Digital Communications*, Cambridge University Press.
- Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L. & Moschetti, F. (2003), ‘A no-reference quality metric for measuring image blur’.
- Pauliks, R., Tretjaks, K., Belahs, K. & Pauliks, R. (2013), A survey on some measurement methods for subjective video quality assessment, *in* ‘2013 World Congress on Computer and Information Technology (WCC-IT)’, IEEE.

- Powers, D. M. W. (2020), ‘Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation’.
- Reimers, U. (1998), ‘Digital video broadcasting’, *IEEE Communications Magazine* **36**(6), 104–110.
- Reimers, U. (2006), ‘DVB The Family of International Standards for Digital Video Broadcasting’, *Proceedings of the IEEE*. **94**(1), 173–182.
- Robin, M. & Poulin, M. (2000), *Digital Televisin Fundamentals 2nd Edition*, McGraw-Hill Professional.
- Rodriguez, D. Z., Rosa, R., AbrahÃ£o, J., Bressan, G. & Costa Alfaia, E. (2014), ‘Video Quality Assessment in Video Streaming Services Considering User Preference for Video Content’, *IEEE Transactions on Consumer Electronics* **60**, 436–444.
- Salomon, D. & Motta, G. (2010), *Handbook of Data Compression*, fifth edn, Springer London.
- Seshadrinathan, K., Soundararajan, R., Bovik, A. C. & Cormack, L. K. (2010), ‘Study of Subjective and Objective Quality Assessment of Video’, *IEEE Transactions on Image Processing* **19**(6), 1427–1441.
- Shahreza, H. O., Amini, A. & Behroozi, H. (2019), ‘No-reference video quality assessment using recurrent neural networks’.
- Sheikh, H., Sabir, M. & Bovik, A. (2006), ‘A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms’, *IEEE Transactions on Image Processing* **15**(11), 3440–3451.
- Suryanto (2021), ‘Indonesia mulai masuki era TV Digital’, URL: https://www.kominfo.go.id/content/detail/3400/indonesia-mulai-masuki-era-tv-digital/0/sorotan_media. (Diakses tanggal 26 April 2021).

- Tom (2022), ‘What Is The Difference Between TV Signal Strength and Quality?’, URL: <https://www.smartaerials.co.uk/blog/what-is-the-difference-between-tv-signal-strength-quality>. (Diakses tanggal 30 Juli 2022).
- Varga, D. (2021), ‘No-Reference Image Quality Assessment with Global Statistical Features’, *Journal of Imaging* **7**(2), 29.
- Victoria, A. (2020), ‘Information Technology’.
- Wang, Z. (2003), ‘Objective Video Quality Assessment, The Handbook of Video Databases: Design and Applications’, *B.Furht and O. Marqure, ed., CRC Press.* pp. 1041–1078.
- Wang, Z., Bovic, A. & Hamid, S. (2002), ‘No-reference perceptual quality assessment of JPEG compressed images’, *IEEE International Conference on Image Processing* .
- Welvaert, M. & Rosseel, Y. (2013), ‘On the Definition of Signal-To-Noise Ratio and Contrast-To-Noise Ratio for fMRI Data’, *PLoS ONE* **8**(11), e77089.
- Xu, J., Tao, C., Mao, X., Lu, X., Bao, J., Drobe, B. & Chen, H. (2021), ‘Blur Detection Sensitivity Increases in Children Using Orthokeratology’, *Frontiers in Neuroscience* **15**.
- Ya’acob, N., Tajudin, N., Zamiri, M. M. . M., Yusof, A. L., Sarnin, S. S. & Idris, A. (2019), ‘Measurement of digital video broadcasting-second generation terrestrial (DVB-T2) signal in Kuala Lumpur, Malaysia’, *Indonesian Journal of Electrical Engineering and Computer Science* **13**(3), 1286.
- yen Ong, C., Song, J., Pan, C. & Li, Y. (2010), ‘Technology and standards of digital television terrestrial multimedia broadcasting, Topics in Wireless Communications’, *IEEE Communications Magazine* **48**(5), 119–127.

Yin, R. K. (n.d.), *Case Study Research and Applications : Design and Methods*,
SAGE Publications, Inc.

LAMPIRAN

1. Program Memperoleh Kualitas Sinyal dari TVheadend

Program ditulis dalam bahasa python sebagai berikut.

Program here ...

2. Program Pengukuran Metrik Objektif NR-IQA Blockiness

Program ditulis dalam bahasa python sebagai berikut.

Program here ...

3. Tabel hasil pengukuran obyektif dan subyektif pada dataset

Tabel 5.1: A sample long table.

nama	m_blok	m_blur	m_grad	std_grad	move_fr	m_temp	mos	class
000.mp4	87	52	4.84	4.76	95	75	3.8	3
001.mp4	88	55	0.29	0.25	100	75	4.64	5
002.mp4	73	1	-0.26	0.12	50	90	2.24	2
003.mp4	79	8	3.85	8.5	71	90	1.6	1
004.mp4	80	10	0.33	4.94	75	85	1.84	1
005.mp4	73	2	3.67	8.12	71	90	2.04	2
006.mp4	72	42	-0.51	0.59	49	90	1.48	1
007.mp4	48	13	3.9	9.97	71	35	1	1
008.mp4	72	10	-0.35	0.63	50	90	1.64	1
009.mp4	86	52	1.63	0.78	98	80	3.96	3
010.mp4	78	9	0	0.31	100	85	4.2	4
011.mp4	77	15	1.63	6.58	73	75	3.2	3
012.mp4	73	4	-0.47	1.67	50	90	1.56	1
013.mp4	78	23	0.31	0.25	100	75	4.2	4
014.mp4	82	15	0.84	0.68	99	75	3.84	3
015.mp4	66	30	11.8	17.7	63	65	1.16	1

016.mp4	77	16	-0.09	0.25	50	75	1.76	1
017.mp4	67	9	-0.23	0.29	50	90	1.96	2
018.mp4	69	8	-0.44	0.56	50	85	1.64	1
019.mp4	95	52	3.91	1.32	96	90	4.32	5
020.mp4	72	2	0.38	1.98	75	80	1.96	2
021.mp4	86	52	5.45	7.42	70	100	4.16	4
022.mp4	87	52	2.06	3.89	73	85	4.68	5
023.mp4	85	52	0.83	0.98	99	100	3.2	3
024.mp4	82	8	1.33	0.15	99	75	4.36	5
025.mp4	82	1	27.66	30.23	47	50	2	2
026.mp4	76	40	0.07	0.1	100	85	4.28	4
027.mp4	88	70	0.22	0.08	100	85	4.6	5
028.mp4	89	52	0.85	0.46	99	90	4.36	5
029.mp4	72	61	-3.1	3.11	47	80	1.84	1
030.mp4	73	46	0.06	0.16	100	90	3.72	3
031.mp4	84	85	2.53	3.23	97	85	4.32	5
032.mp4	73	69	-0.9	0.27	49	90	1.4	1
033.mp4	72	80	0.21	0.1	100	90	3.56	3
034.mp4	72	34	0.33	0.24	100	100	4.12	4
035.mp4	79	15	1.29	2.43	99	90	1.76	1
036.mp4	74	5	-0.75	0.61	49	80	1.2	1
037.mp4	85	52	1.25	0.19	99	85	4.08	3
038.mp4	73	21	4.74	6.49	70	90	1.36	1
039.mp4	73	12	0.68	1.01	99	85	2.24	2
040.mp4	86	50	0.86	0.18	99	65	4.56	5
041.mp4	85	52	0.14	0.12	100	90	4.68	5
042.mp4	81	35	3.68	9.04	71	90	2.36	2
043.mp4	87	52	12.89	1.4	87	70	3.88	3
044.mp4	83	18	0.42	0.37	100	75	3.88	3
045.mp4	75	2	1.67	1.68	98	85	4.16	4
046.mp4	88	62	0.61	0.29	99	85	4.28	4
047.mp4	83	35	0.96	1.2	99	85	2.16	2
048.mp4	83	19	2.03	1.8	98	80	3.92	3
049.mp4	71	4	0.14	0.44	100	90	2.48	2
050.mp4	80	97	0.06	0.36	100	75	4.76	5
051.mp4	75	15	4.21	10.84	71	85	2.04	2
052.mp4	90	52	0.81	0.7	99	80	4.32	5
053.mp4	75	41	14.21	21.08	61	70	1.64	1
054.mp4	87	52	1.15	1.15	99	75	4.2	4
055.mp4	82	14	0.73	0.44	99	85	4.52	5
056.mp4	89	52	2.63	1.55	97	90	4.52	5
057.mp4	83	12	0.26	0.68	100	80	4.68	5
058.mp4	71	4	-0.19	0.22	50	100	3.96	3

059.mp4	78	10	1.16	0.17	99	90	3.72	3
060.mp4	88	52	3.63	4.03	96	100	3.88	3
061.mp4	74	10	-4.59	4.65	45	80	1.16	1
062.mp4	65	15	-4.4	3.07	46	80	1.28	1
063.mp4	66	6	0.01	0.03	100	75	4.56	5
064.mp4	74	50	-0.49	0.82	50	80	1.28	1
065.mp4	80	13	0.11	0.38	100	85	4.2	4
066.mp4	81	6	0.57	0.12	99	85	4.2	4
067.mp4	79	38	12.31	30.32	63	75	2.92	2
068.mp4	85	52	1.66	0.08	98	90	3.64	3
069.mp4	86	52	1.21	1.16	99	90	4.24	4
070.mp4	77	8	0.19	0.18	100	90	3.92	3
071.mp4	73	12	0.37	0.86	100	70	2.92	2
072.mp4	79	48	0.22	0.28	100	80	4.24	4
073.mp4	94	52	4.01	4.5	96	85	4.32	5
074.mp4	85	52	0.5	0.5	100	75	4.64	5
075.mp4	89	52	1.03	0.16	99	70	4.72	5
076.mp4	72	48	1.79	2.97	98	90	1.96	2
077.mp4	82	18	0.53	2.85	74	80	2.24	2
078.mp4	88	52	0.25	0.27	100	75	4.48	5
079.mp4	88	68	5.7	1.91	94	90	4.12	4
080.mp4	83	1	1.42	2.47	99	90	2.6	2
081.mp4	80	5	-0.88	2.69	49	90	2.44	2
082.mp4	87	52	4.95	6.8	70	85	4.2	4
083.mp4	76	60	0.38	0.26	100	80	3.56	3
084.mp4	88	52	6.71	9.62	68	90	4	3
085.mp4	87	52	3.86	1.54	96	80	4.12	4
086.mp4	79	15	0.31	0.17	100	75	4.12	4
087.mp4	79	29	0.07	1.13	100	85	2.64	2
088.mp4	78	21	0.1	0.52	100	80	4.2	4
089.mp4	83	12	1.75	1.51	98	90	4.36	5
090.mp4	88	61	8.22	5.61	92	90	3.96	3
091.mp4	70	14	-0.25	0.51	50	80	3.56	3
092.mp4	74	12	2.21	2.6	98	75	2.12	2
093.mp4	67	40	-0.28	0.59	50	80	2	2
094.mp4	68	50	-0.23	0.38	50	85	1.6	1
095.mp4	76	44	1.3	2.92	74	65	1.04	1
096.mp4	95	52	4	1.57	96	90	4.16	4
097.mp4	80	0	0.29	0.46	100	90	4.68	5
098.mp4	82	9	0.1	0.26	100	75	4.2	4
099.mp4	94	70	2.68	1.82	97	90	4.2	4
100.mp4	87	76	2.12	2.95	98	90	4.68	5
101.mp4	79	65	-0.47	0.58	50	85	2.2	2

102.mp4	92	52	9.33	16.3	66	100	4.12	4
103.mp4	78	21	1.86	7.08	73	85	2.32	2
104.mp4	86	52	1.36	2.3	99	75	4.12	4

Program here ...



RIWAYAT HIDUP

IDENTITAS DIRI

Nama : Sandy Suryo Prayogo
Tempat/Tanggal Lahir : Surabaya, 14 April 1993
Jenis Kelamin : Laki-laki
Pekerjaan : Dosen
NIP / NIDN : 171011 / 0314049301
Jabatan Fungsional : Lektor
Unit Kerja : Laboratorium Pengembangan Teknik Elektro
Institusi : Universitas Gunadarma
Telp : 085782382323
Email Staff : sandy_sr@staff.gunadarma.ac.id
Email Pribadi : sunariveru@gmail.com

PENDIDIKAN FORMAL

Tahun	Pendidikan	Institusi
1999	TK	TK Aisyiah 37 Jakarta
2005	SD	SD N Menteng 02 Jakarta
2008	SMP	SMP Negeri 216 Jakarta
2011	SMA	SMA Negeri 28 Jakarta
2015	S1 Teknik Elektro	Universitas Gunadarma
2016	S2 Teknik Elektro	Universitas Gunadarma
—	S3 Teknologi Informasi	Universitas Gunadarma

PENGALAMAN DILUAR PENDIDIKAN FORMAL

Tahun	Jabatan
2016	<i>Linneaus Palme Scholarship Swedia - Student Exchange</i>
2017 -	Dosen Tetap Universitas Gunadarma
2017 -	Reviewer Nasional Program Kreatifitas Mahasiswa - Kemdikbud
2017 -	Pilot Drone Licences Asosiasi Pilot Drone Indonesia - Kemenhub

PUBLIKASI ILMIAH

Tahun	Penulis-Judul	Keterangan
2020	Sandy Suryo Prayogo, Tubagus Maulana Kusuma - Analisis Sensitivitas Video MPEG-4 berdasarkan Struktur Frame Pada Transmisi DVB-T	Jurnal Ilmiah Informatika Komputer 25 (2), 86-97 , Universitas Gunadarma
2023	-	-
2023	-	-