

SKRIPSI
PERANCANGAN SISTEM PENDETEKSI KERUSAKAN PADA
PCB MENGGUNAKAN SSD MOBILENET V2

Oleh
Arif Priyam Budi
065119106



PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024

SKRIPSI
PERANCANGAN SISTEM PENDETEKSI KERUSAKAN PADA
PCB MENGGUNAKAN SSD MOBILENET V2

Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

Oleh
Arif Priyam Budi
065119106



PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024

HALAMAN PENGESAHAN

JUDUL : Perancangan Sistem Pendeteksi Kerusakan pada *PCB*
Menggunakan *SSD MobileNet V2*
NAMA : Arif Priyam Budi
NPM : 065119106

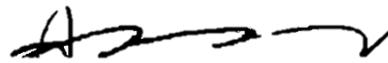
Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA – UNPAK



(Teguh Pujanegara S.Kom. , M.Si)

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA - UNPAK



(Dr. Andi Chairunnas S.Kom., M.Pd)

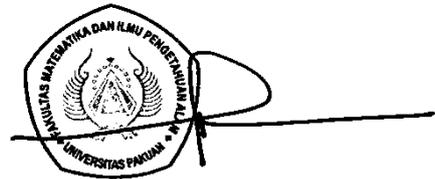
Mengetahui,

Ketua
Program Studi Ilmu Komputer
FMIPA - UNPAK



(Arie Qur'ania, M.Kom)

Dekan
FMIPA - UNPAK



(Asep Denih, S.Kom., M.Sc., Ph.D)

PERNYATAAN KEASLIAN KARYA TULIS

Dengan ini saya menyatakan bahwa :

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian – bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar – benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai peraturan yang berlaku.

Bogor, 24 Juli 2024



Arif Priyam Budi
065119106

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertandatangan di bawah ini :

Nama : Arif Priyam Budi
NPM : 065119106
Judul Skripsi : Perancangan Sistem Pendeteksi Kerusakan pada *PCB*
Menggunakan *SSD MobileNet V2*

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, 24 Juli 2024



Arif Priyam Budi
065119106

RIWAYAT HIDUP



Arif Priyam Budi lahir Bogor pada tanggal 18 Maret 2001 dari pasangan Bapak Wawan Suryana dan Ibu Nur Faiza sebagai anak kedua dari tiga bersaudara. Penulis memulai pendidikan di Sekolah Dasar yang bertempat di SDN Citeureup 04, kemudian tahun 2014 masuk SMP Negeri 1 Citeureup. Penulis adalah Alumni dari SMKN 1 Gunung Putri. Pada tahun 2019 penulis melanjutkan pendidikan ke Universitas Pakuan, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Pada bulan Juli 2024 penulis telah menyelesaikan penelitian dengan judul “Perancangan Sistem Pendeteksi Kerusakan pada *PCB* Menggunakan *SSD MobileNet V2*”.

RINGKASAN

Arif Priyam Budi 2024. Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2 **Teguh Pujanegara S.Kom. , M.Si dan Dr. Andi Chairunnas S.Kom., M.Pd.**

Dalam era digital yang terus berkembang, teknologi semakin menjadi bagian integral dalam berbagai aspek kehidupan manusia. Salah satu teknologi yang saat ini sedang mengalami perkembangan pesat adalah kecerdasan buatan atau *Artificial Intelligence*. AI telah memasuki berbagai bidang, mulai dari pengenalan wajah hingga otomatisasi proses industri. Salah satu aplikasi yang menarik dari AI adalah dalam bidang *computer vision* yang merupakan sub-bidang dari kecerdasan buatan dalam pengolahan citra, bertujuan untuk memungkinkan mesin untuk menganalisis dan menginterpretasi data visual, mirip dengan cara manusia memahami dunia visual. Sehingga AI dalam bidang tersebut memiliki potensi besar untuk meningkatkan efisiensi dan akurasi dalam proses identifikasi masalah pengolahan citra (Yadav, 2023). Inspeksi visual pada *Printed Circuit Board (PCB)* merupakan langkah kritis dalam proses produksi elektronik untuk memastikan kualitas dan kinerja perangkat. Meskipun telah ada kemajuan dalam teknologi, inspeksi manual masih rentan terhadap kesalahan dan keterbatasan manusia. (Prastio, 2023). Beberapa penelitian tentang inspeksi PCB yang telah dilakukan salah satunya oleh Rangga Ade Juliano pada tahun 2022 dengan judul “Pembuatan Alat Inspeksi Visual Jalur PCB menggunakan Pengolahan Citra” pada penelitian tersebut menunjukkan keberhasilan menggunakan arsitektur YOLO (*You Only Look Once*) CNN dalam mendeteksi jalur yang terputus pada papan PCB. Namun pada penelitian tersebut arsitektur hanya mendeteksi jalur terputus saja, belum menggunakan klasifikasi lainnya seperti komponen terbakar. Berdasarkan beberapa permasalahan diatas diperlukan metode pendeteksi kerusakan PCB yang efisien agar mempermudah pekerjaan manusia. Terdapat berbagai macam arsitektur neural network yang tersedia di internet yang biasa disebut *pre-trained model*. Arsitektur tersebut dapat dilatih ulang dengan data baru untuk melakukan tugas tugas yang berbeda dengan metode *transfer learning*. Salah satunya arsitektur yang cukup populer adalah *Single Shot Multibox Detector (SSD) MobileNet V2* (Nufus, 2021). Oleh karena itu dibuat penelitian dengan judul “Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2”. Penelitian terkait *Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2* menunjukkan hasil yang positif. Dalam penelitian ini, model SSD MobileNet V2 berhasil melakukan transfer learning dengan data baru yaitu kerusakan PCB. Teknik transfer learning dalam pelatihan model mempermudah peneliti dalam menghasilkan model yang dapat menyelesaikan tugas yang spesifik. Library Tensorflow sangat membantu dalam proses transfer learning, dimana peneliti hanya perlu mengatur parameter dengan tepat sehingga model dapat sesuai dengan harapan. Pembuatan model AI tidak lepas dari hardware yang digunakan, mengingat proses training merupakan proses yang cukup kompleks dan membutuhkan komputasi yang tinggi. Salah satu hardware yang penting dalam proses training adalah GPU (*Graphical Processing Unit*). Dalam penelitian ini, digunakan GPU Nvidia Geforce RTX3050 sebagai alat komputasi selama training dan menghabiskan waktu sekitar 2 jam dengan nilai step sebanyak 20.000. Selama

proses training didapatkan nilai metrik yaitu **Classification Loss 0.02352**, **Localization Loss 0.006221**, **Regulation Loss 0.06596** dan **Total Loss 0.0957**. Hasil dari setiap metrik menunjukkan tren penurunan yang signifikan, yang menandakan bahwa model belajar dengan baik dan mengurangi kesalahan prediksi secara efektif. Selain itu, pada tahap uji validasi, inspeksi visual menunjukkan rata-rata score deteksi sebesar **76.63%**. Inspeksi kelistrikan dilakukan untuk memastikan kebenaran deteksi, dimana dilakukan 15 kali pengujian menggunakan multimeter. Hasil dari pengujian tersebut menunjukkan bahwa 12 dari 15 deteksi atau **80%** menggunakan SSD MobileNet V2 menunjukkan hasil yang sesuai dengan inspeksi menggunakan multimeter. Perancangan *Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2* ini masih memiliki banyak kekurangan. Seperti kamera mikroskopis memiliki tingkat resolusi dan pencahayaan yang perlu diperbaiki. Pada penelitian berikutnya disarankan, diperlukan upaya untuk meningkatkan resolusi dan kualitas kamera mikroskopis. Selain itu pada penelitian selanjutnya disarankan untuk meningkatkan jumlah klasifikasi kerusakan dan meningkatkan jumlah data sehingga penggunaan aplikasi dapat lebih luas lagi dan akurasi model dapat meningkat.

KATA PENGANTAR

Puji Syukur kehadiran Allah SWT, karena rahmat dan hidayah-Nya penulis dapat menyelesaikan karya tulis skripsi yang berjudul **“Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2”** sebagai salah satu syarat dalam mendapatkan gelar sarjana.

Dalam penulisan tugas akhir ini, penulis mengucapkan banyak terimakasih yang sebesar besarnya kepada :

1. Dr. Andi Chairunnas S.Kom., M.Pd selaku pembimbing utama yang telah berkenan meluangkan waktunya, memberikan petunjuk penulisan laporan ini serta memberikan dorongan moral dan motivasi kepada penulis selama ini.
2. Teguh Pujanegara S.Kom., M.Si selaku pembimbing pendamping yang telah meluangkan waktunya dan memberikan arahan.
3. Arie Qur'ania, M.Kom. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Pakuan.
4. Orang Tua yang telah memberikan perhatian dan senantiasa memberikan dorongan moral, materil dan motivasi serta doanya kepada penulis.
5. Sahabat, teman seperjuangan yang telah membantu menyelesaikan penulisan laporan tugas akhir ini.
6. Semua pihak yang telah membantu penulis dalam melaksanakan penyusunan laporan ini.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna. Untuk itu penulis mengharapkan kritik dan saran yang sifatnya membangun demi perbaikan ke arah kesempurnaan. Walaupun demikian, penulis berharap karya tulis ini dapat bermanfaat bagi penulis pada khususnya dan para pembaca pada umumnya.

Penulis,

Arif Priyam Budi
065119106

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	i
PERNYATAAN KEASLIAN KARYA TULIS	ii
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	iii
RIWAYAT HIDUP	iv
RINGKASAN	v
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Ruang Lingkup Penelitian	2
1.4 Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	3
2.1 Tinjauan Pustaka.....	3
2.1.1 Printed Circuit Board (PCB)	3
2.1.2 <i>SSD MobileNet V2</i>	3
2.1.3 Transfer Learning	4
2.1.4 Kamera Mikroskopis	5
2.2 Penelitian Terdahulu	5
2.3 Tabel Perbandingan	6
BAB III METODE PENELITIAN	7
3.1 Metode Penelitian	7
3.2 Analisis	7
3.3 Desain Model & Dataset.....	8
3.4 Training Model	8
3.5 Pengujian Model.....	9
3.6 Pemeliharaan.....	9
BAB IV RANCANGAN DAN IMPLEMENTASI	10
4.1 Analisis Kebutuhan Sistem.....	10
4.1.1 Identifikasi Kebutuhan Sistem	10

4.1.2.	Studi Literatur.....	10
4.1.3.	Identifikasi Data	10
4.1.4.	Langkah Penginstalan Framework dan Software	10
4.2.	Desain Software	12
4.2.1.	Arsitektur Model	12
4.2.2.	Desain Dataset	12
4.2.3.	Desain Flowchart.....	14
4.3.	Implementasi.....	15
4.3.1.	Persiapan Data	15
4.3.2.	Pelatihan Model.....	16
4.4.	Testing	17
BAB V HASIL DAN PEMBAHASAN		18
5.1	Hasil Penelitian	18
5.2	Test Fungsional Keseluruhan Sistem (<i>Overall Testing</i>).....	18
5.2.1	Pengujian Struktural	18
5.2.2	Pengujian Fungsional	19
5.2.3	Uji Validasi.....	20
6.1	Kesimpulan	32
6.2	Saran	32
DAFTAR PUSTAKA		33
LAMPIRAN.....		36

DAFTAR GAMBAR

Gambar 1. Arsitektur SSD MobileNet V2	3
Gambar 2. Kamera Mikroskopis	5
Gambar 3. Tahapan SDLC	7
Gambar 4. Menginstall Bahasa Perograman Python.....	10
Gambar 5. Menginstall Library Tensorflow.....	11
Gambar 6. Menginstall Object Detection API	11
Gambar 7. Menginstall CUDA dan CuDNN.....	11
Gambar 8. Menginstall Library LabelImg.....	12
Gambar 9. Menginstall Library Pendukung	12
Gambar 10. Pengumpulan Data Jalur Terputus.....	13
Gambar 11. Pengumpulan Data Komponen Terbakar	13
Gambar 12. Pengumpulan Data Solder Defect	14
Gambar 13. Proses Anotasi Data.....	14
Gambar 14. Desain Flowchart.....	15
Gambar 15. Persiapan Data	16
Gambar 16. Pelatihan Model.....	17
Gambar 17. Testing	17
Gambar 18. Tampilan Alat.....	18
Gambar 19. Pengujian Model AI	18
Gambar 20. Pengujian Kode Program.....	19
Gambar 21. Pengujian Kamera Mikroskop	19
Gambar 22. Pengujian Keseluruhan Sistem	20
Gambar 23. Classification Loss.....	20
Gambar 24. Localization Loss.....	21
Gambar 25. Regularization Loss	21
Gambar 26. Total Loss	22
Gambar 27. Inspeksi Kelistrikan Solder Defect.....	27
Gambar 28. Inspeksi Kelistrikan Jalur Terputus	28
Gambar 29. Inspeksi Kelistrikan Komponen Terbakar.....	28
Gambar 30. Cara Kerja Sistem.....	30
Gambar 31. Pengaturan Score Threshold 0.....	31
Gambar 32. Pengaturan Score Threshold 50.....	31

DAFTAR TABEL

Tabel 1. Penelitian Terdahulu.....	5
Tabel 2. Perbandingan Penelitian	6
Tabel 3. Sampel PCB	13
Tabel 4. Parameter Training	16
Tabel 5. Argument Training	17
Tabel 6. Inspeksi Visual Jalur Terputus	22
Tabel 7. Inspeksi Visual Solder Defect	24
Tabel 8. Inspeksi Visual Komponen Terbakar	24
Tabel 9. Perbandingan Inspeksi Visual dan Model SSD MobileNet V2	26
Tabel 10. Perbandingan Inspeksi Kelistrikan dan Model SSD MobileNet V2	29

DAFTAR LAMPIRAN

Lampiran 1 Sample PCB Rusak.....	36
Lampiran 2 Inspeksi Visual.....	37
Lampiran 3 Data Train	37
Lampiran 4 Data Test	37
Lampiran 5 File Konfigurasi Training	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang terus berkembang, teknologi semakin menjadi bagian integral dalam berbagai aspek kehidupan manusia. Salah satu teknologi yang saat ini sedang mengalami perkembangan pesat adalah kecerdasan buatan atau *Artificial Intelligence*. AI telah memasuki berbagai bidang, mulai dari pengenalan wajah hingga otomatisasi proses industri. Salah satu aplikasi yang menarik dari AI adalah dalam bidang *computer vision* yang merupakan sub-bidang dari kecerdasan buatan dalam pengolahan citra, bertujuan untuk memungkinkan mesin untuk menganalisis dan menginterpretasi data visual, mirip dengan cara manusia memahami dunia visual. Sehingga AI dalam bidang tersebut memiliki potensi besar untuk meningkatkan efisiensi dan akurasi dalam proses identifikasi masalah pengolahan citra (Yadav, 2023).

Inspeksi visual pada *Printed Circuit Board (PCB)* merupakan langkah kritis dalam proses produksi elektronik untuk memastikan kualitas dan kinerja perangkat. Meskipun telah ada kemajuan dalam teknologi, inspeksi manual masih rentan terhadap kesalahan dan keterbatasan manusia. Masalah utama meliputi kesulitan dalam mendeteksi kerusakan kecil, biaya produksi yang tinggi, dan waktu yang diperlukan untuk inspeksi. Kompleksitas desain PCB dan variasi dalam kerusakan menambah kompleksitas proses. Mengatasi tantangan ini memerlukan inovasi dalam metodologi inspeksi, termasuk pengembangan teknik otomatisasi yang lebih canggih untuk mendeteksi dan mengidentifikasi kerusakan dengan akurat dan efisien (Prastio, 2023).

Beberapa penelitian tentang inspeksi PCB yang telah dilakukan salah satunya oleh Rangga Ade Juliano pada tahun 2022 dengan judul “Pembuatan Alat Inspeksi Visual Jalur PCB menggunakan Pengolahan Citra” pada penelitian tersebut menunjukkan keberhasilan menggunakan arsitektur YOLO (*You Only Look Once*) CNN dalam mendeteksi jalur yang terputus pada papan PCB. Namun pada penelitian tersebut arsitektur hanya mendeteksi jalur terputus saja, belum menggunakan klasifikasi lainnya seperti komponen terbakar.

Berdasarkan beberapa permasalahan diatas diperlukan metode pendeteksi kerusakan PCB yang efisien agar mempermudah pekerjaan manusia. Terdapat berbagai macam arsitektur neural network yang tersedia di internet yang biasa disebut *pre-trained model*. Arsitektur tersebut dapat dilatih ulang dengan data baru untuk melakukan tugas tugas yang berbeda dengan metode *transfer learning*. Salah satunya arsitektur yang cukup populer adalah *Single Shot Multibox Detector (SSD)* MobileNet V2 (Nufus, 2021). Mohit Phadtare pada tahun 2021 pada penelitiannya yang berjudul “*Comparison between YOLO and SSD Mobile Net for Object Detection in a Surveillance Drone*” menunjukkan bahwa arsitektur SSD MobileNet V2 memiliki kinerja yang lebih baik jika dibandingkan dengan arsitektur YOLO untuk mendeteksi objek yang kecil, sehingga penggunaan arsitektur SSD MobileNet V2 dirasa cocok untuk mendeteksi kerusakan pada PCB. Oleh karena itu dibuat penelitian dengan judul “**Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2**”.

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang *Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2*

1.3 Ruang Lingkup Penelitian

Ruang lingkup penelitian ini meliputi :

1. Model AI menggunakan arsitektur SSD MobileNet V2 yang dapat mendeteksi kerusakan PCB
2. Klasifikasi kerusakan PCB yaitu meliputi Jalur terputus, komponen terbakar dan defect pada solder.
3. Pengumpulan data berasal dari sampel PCB yang rusak yang dapat ditemui pada perangkat handphone dan gadget.
4. Gambar ditangkap menggunakan kamera mikroskop dengan resolusi 320x320 pixel sebanyak 30 foto pada masing masing kelas.
5. Jenis PCB yang digunakan merupakan jenis SMD (*Surface Mounted Device*) yang dapat ditemui pada perangkat Handphone, dan Gadget

1.4 Manfaat Penelitian

Manfaat penelitian adalah sebagai berikut:

1. Mempermudah pekerjaan manusia dalam mendeteksi kerusakan PCB
2. Mengurangi potensi kesalahan dalam mendeteksi kerusakan PCB.

BAB II TINJAUAN PUSTAKA

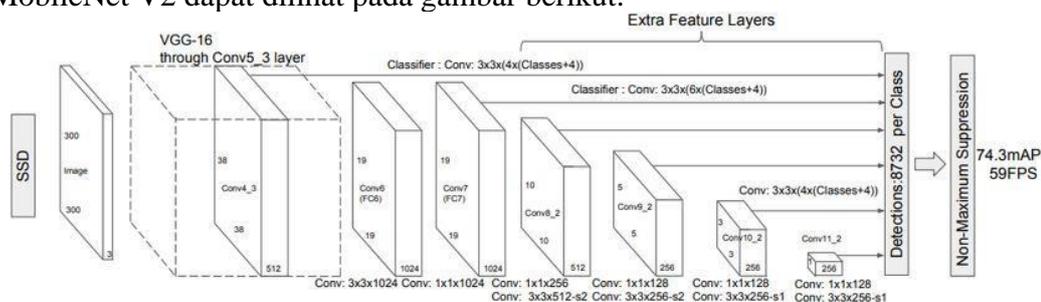
2.1 Tinjauan Pustaka

2.1.1 Printed Circuit Board (PCB)

Printed Circuit Board (PCB) adalah komponen kritis dalam berbagai perangkat elektronik modern. PCB memiliki struktur yang terdiri dari substrat yang terbuat dari bahan isolasi seperti fiberglass, yang dilapisi dengan lapisan tembaga membentuk jalur konduktif. Jalur konduktif ini menghubungkan komponen seperti resistor, kapasitor, dan transistor untuk membentuk sirkuit elektronik yang kompleks. Dengan menyediakan jalur konduktif yang teratur dan terpadu, PCB memungkinkan aliran listrik yang efisien dan merupakan inti dari perangkat elektronik seperti ponsel, komputer, dan perangkat lainnya. Keandalan dan kualitas PCB sangat penting untuk memastikan kinerja yang optimal dari perangkat elektronik tersebut. Meskipun demikian, produksi PCB tidak terlepas dari risiko terjadinya cacat atau kerusakan. Oleh karena itu, inspeksi visual PCB menjadi langkah penting dalam proses produksi. Inspeksi visual dilakukan untuk memastikan tidak adanya kecacatan pada jalur konduktif, sambungan yang rusak, atau komponen yang tidak terpasang dengan baik. Dengan inspeksi visual yang cermat, dapat diidentifikasi dan diatasi masalah potensial sebelum PCB dipasang ke dalam perangkat elektronik (Juliano, 2022).

2.1.2 SSD MobileNet V2

SSD MobileNet V2 adalah salah satu arsitektur dalam bidang *computer vision* yang dikembangkan oleh Google. Arsitektur ini menggabungkan dua konsep utama yaitu SSD (*Single Shot MultiBox Detector*) untuk deteksi objek dan MobileNet V2 untuk ekstraksi fitur. Keunggulan utama SSD MobileNet V2 terletak pada keseimbangan antara kecepatan dan akurasi deteksi objek. Dengan memanfaatkan arsitektur MobileNet V2 yang ringan, SSD MobileNet V2 mampu menghasilkan deteksi objek yang akurat dalam waktu yang relatif singkat. Hal ini membuatnya cocok untuk aplikasi deteksi objek real-time pada perangkat dengan sumber daya terbatas, seperti perangkat *mobile* atau sistem embedded. Selain itu, SSD MobileNet V2 juga dapat diimplementasikan dengan mudah dan efisien dalam berbagai proyek komputer vision, baik untuk tujuan akademis maupun industri. Dengan kombinasi kecepatan, akurasi, dan kemudahan implementasi, SSD MobileNet V2 telah menjadi pilihan yang populer dalam pengembangan sistem deteksi objek dan pengolahan citra modern (Yadav, 2023). Adapun arsitektur SSD MobileNet V2 dapat dilihat pada gambar berikut.



Gambar 1. Arsitektur SSD MobileNet V2

Selain itu cara kerja model SSD MobileNet V2 dapat terbagi menjadi beberapa tahap

1. **Input Gambar**
Gambar input dalam format RGB di-resample ke ukuran yang sesuai untuk model yaitu 320x320 pixel.
2. **Ekstraksi Fitur**
Ekstraksi fitur adalah proses mengidentifikasi dan memperoleh informasi signifikan dari gambar atau data. Fitur-fitur ini adalah atribut atau karakteristik dari gambar yang membantu dalam mendeskripsikan dan memahami konten gambar tersebut. Dalam konteks visi komputer, fitur mencakup pola, tekstur, warna dan garis tepi.
3. **Deteksi Objek**
Peta fitur dari MobileNet dikirim ke lapisan deteksi SSD. Lapisan ini memiliki dua komponen utama yaitu Bounding Box Predictions yang Menghasilkan prediksi untuk lokasi bounding box dalam peta fitur dan Class Predictions: Menghasilkan prediksi untuk kelas objek dalam setiap bounding box.
4. **Prediksi**
Setiap prediksi bounding box disertai dengan skor untuk setiap kelas, yang menunjukkan seberapa yakin model bahwa bounding box tersebut mengandung objek dari kelas tertentu.
5. **Non-Maximum Suppression (NMS)**
Untuk mengurangi deteksi duplikat dan memilih bounding box terbaik, algoritma Non-Maximum Suppression (NMS) diterapkan. NMS menghapus bounding boxes yang memiliki overlap tinggi (melebihi threshold) dan hanya menyisakan bounding box dengan skor tertinggi.
6. **Output**
Setelah NMS diterapkan, bounding boxes yang tersisa, bersama dengan kelas dan skor akhir mereka, dianggap sebagai hasil akhir deteksi objek.

2.1.3 Transfer Learning

Transfer learning adalah metode dalam pembelajaran mesin di mana pengetahuan yang telah dipelajari oleh model dari satu tugas atau domain digunakan untuk meningkatkan kinerja model dalam tugas atau domain yang berbeda. Prinsip dasar dari transfer learning adalah menggunakan representasi fitur yang telah dipelajari oleh model dalam tugas sebelumnya untuk mempercepat dan meningkatkan pembelajaran dalam tugas baru. Dalam transfer learning, model yang telah dilatih sebelumnya disebut sebagai model pre-trained. Model pre-trained ini dapat digunakan secara langsung atau disesuaikan (fine-tuned) dengan dataset yang spesifik untuk tugas yang diinginkan. Proses fine-tuning melibatkan menyesuaikan parameter model dengan dataset baru sehingga model dapat belajar representasi fitur yang lebih relevan untuk tugas yang baru. Transfer learning sangat berguna ketika dataset untuk tugas baru terbatas atau ketika ingin menghindari proses pelatihan model dari awal yang membutuhkan sumber daya komputasi yang besar. Dengan memanfaatkan pengetahuan yang telah ada dari tugas sebelumnya, transfer learning dapat menghasilkan model yang lebih efisien dan akurat dalam menyelesaikan tugas yang baru. Oleh karena itu, transfer learning telah menjadi salah satu teknik yang populer dan efektif dalam bidang pembelajaran mesin dan kecerdasan buatan (Hosna, 2022).

2.1.4 Kamera Mikroskopis

Kamera mikroskopis adalah perangkat kamera yang dirancang khusus untuk merekam gambar dari objek yang diperbesa. Kamera ini menggunakan lensa khusus untuk menangkap detail-detail kecil dari objek yang diamati. Kamera mikroskopis sering digunakan dalam berbagai aplikasi, termasuk dalam ilmu biologi, kedokteran forensik, dan ilmu material. Dengan bantuan kamera mikroskopis, pengguna dapat merekam gambar-gambar yang berukuran kecil, menyimpannya dalam format digital, dan membagikannya untuk tujuan analisis, dokumentasi, atau presentasi. Hal ini memungkinkan pengamatan objek mikroskopis dengan lebih nyaman dan memfasilitasi pertukaran informasi antara ilmuwan dan peneliti. (Ariska, 2019)



Gambar 2. Kamera Mikroskopis

2.2 Penelitian Terdahulu

Penelitian terdahulu tentang inspeksi PCB dan SSD MobileNet yang pernah dilakukan dapat dilihat pada tabel berikut:

Tabel 1. Penelitian Terdahulu

No		
1	Nama	Rangga Ade Juliano (2022)
	Judul	Pembuatan Alat Inspeksi Visual Jalur PCB menggunakan Pengolahan Citra
	Kelebihan	Menggunakan arsitektur YOLO CNN dalam mendeteksi kerusakan jalur PCB
	Kekurangan	Inspeksi terbatas hanya jalur saja belum ada inspeksi dengan komponen.
2	Nama	Rizki Putra Prastio (2023)
	Judul	<i>Selection of autofocus algorithms for printed circuit board automated optical inspection system</i>
	Kelebihan	Menggunakan algoritma <i>automated optical inspection (AOI)</i> untuk mengatur autofocus kamera mikroskopis yang diterapkan dalam inspeksi PCB
	Kekurangan	Tidak dapat mendeteksi kerusakan pada PCB
3	Nama	Ivan Besando Pakpahan (2021)

Judul	Pendeteksian Lubang Pada Jalanan Menggunakan Metode SSD-MobileNet
Kelebihan	Menggunakan model AI SSD MobileNet untuk mendeteksi kerusakan jalan nilai akurasi sebesar 76%
Kekurangan	Input menggunakan kamera konvensional, belum menerapkan kamera mikroskop untuk objek yang lebih kecil

2.3 Tabel Perbandingan

Perbandingan penelitian sebagai berikut:

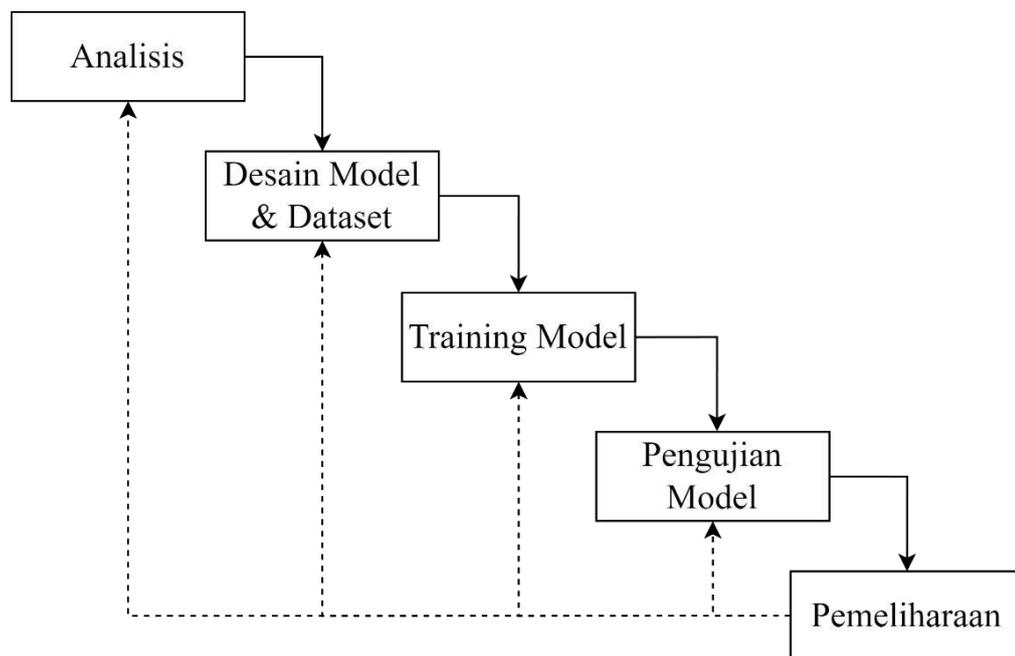
Tabel 2. Perbandingan Penelitian

No	Penelitian dan Tahun	Input		Metode			Jumlah Klasifikasi		Implementasi	
		Kamera Mikroskopis	Kamera Konvensional	<i>automated optical inspection (AOI)</i>	SSD MobileNet	YOLO CNN	1	3	Papan PCB	Jalanan
1	Rangga Ade Juliano (2022)	✓				✓	✓		✓	
2	Rizki Putra Prastio (2023)	✓		✓					✓	
3	Ivan Besando Pakpahan (2021)		✓		✓		✓			✓
4	Arif Priyam Budi (2024)	✓			✓			✓	✓	

BAB III METODE PENELITIAN

3.1 Metode Penelitian

Metode yang digunakan dalam penelitian “Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2” ini menggunakan metode penelitian SDLC (*System Development Life Cycle*). Model SDLC yang dipakai dalam penelitian ini adalah model Waterfall. Waterfall Model atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Menurut Bassil (2012) disebut waterfall karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan. Berikut merupakan tahap dalam penelitian ini dapat dilihat pada berikut :



Gambar 3. Tahapan SDLC

3.2 Analisis

Proses analisis membantu dalam memastikan bahwa solusi yang dikembangkan akan memenuhi kebutuhan bisnis dan pengguna akhir, serta membantu dalam mengidentifikasi potensi masalah atau risiko yang mungkin timbul selama pengembangan sistem. Beberapa kegiatan dalam proses analisis mencakup:

- a. Identifikasi Kebutuhan Sistem
Dilakukan analisis kebutuhan untuk mendeteksi kerusakan pada PCB menggunakan kecerdasan buatan. Ini mencakup pemahaman menyeluruh tentang apa yang diinginkan oleh pengguna akhir dari sistem tersebut, serta aspek-aspek fungsional dan non-fungsional yang harus dipenuhi oleh solusi yang dikembangkan.
- b. Studi Literatur
Meninjau penelitian terdahulu yang terkait dengan deteksi kerusakan pada PCB dan penerapan kecerdasan buatan. Studi literatur membantu dalam memahami perkembangan terbaru dalam bidang ini, teknik-teknik yang

telah terbukti berhasil, serta tantangan dan solusi yang telah diidentifikasi oleh penelitian sebelumnya.

c. Identifikasi Data:

Tentukan jenis data yang diperlukan untuk pelatihan dan pengujian model, seperti dataset gambar PCB yang berisi gambar kerusakan dan non-kerusakan. Proses identifikasi data melibatkan pemahaman tentang sumber data yang tersedia, kualitas data yang dibutuhkan, serta proses pengumpulan dan pengolahan data yang diperlukan untuk memastikan ketersediaan dataset yang tepat untuk melatih dan menguji model deteksi kerusakan pada PCB.

3.3 Desain Model & Dataset

Proses desain membantu dalam menghasilkan panduan yang jelas untuk implementasi sistem, memastikan bahwa solusi yang dikembangkan sesuai dengan kebutuhan dan tujuan bisnis, serta memfasilitasi komunikasi dan kolaborasi antara anggota tim pengembangan. Beberapa kegiatan dalam proses desain meliputi:

b. Desain Arsitektur Model:

Proses ini melibatkan menentukan arsitektur model AI. Berdasarkan pertimbangan penelitian terdahulu maka penelitian ini menggunakan arsitektur SSD MobileNet V2

c. Desain Dataset:

1. Pengumpulan Data: Identifikasi sumber-sumber data yang relevan untuk melatih dan menguji model deteksi kerusakan pada PCB. Data ini dapat berupa gambar PCB yang berisi kerusakan dan non-kerusakan. Pengumpulan data harus memperhatikan keberagaman dalam jenis dan tingkat kerusakan PCB untuk memastikan keakuratan dan keandalan model.
2. Anotasi Data: Jika diperlukan, lakukan proses anotasi data untuk menandai lokasi dan jenis kerusakan pada gambar PCB. Anotasi ini penting untuk melatih model dalam mengidentifikasi dan membedakan antara area yang rusak dan tidak rusak pada PCB.
3. Pembersihan dan Pemrosesan Data: Lakukan pembersihan data untuk menghilangkan noise dan artefak yang tidak diinginkan. Selain itu, lakukan pemrosesan data seperti resizing, normalisasi, dan augmentasi untuk memperbaiki kualitas dan konsistensi data serta meningkatkan keberagaman dataset.
4. Pembagian Dataset: Bagi dataset menjadi subset pelatihan, validasi, dan pengujian sesuai dengan rasio yang ditentukan. Pembagian dataset yang baik membantu dalam menguji dan memvalidasi kinerja model dengan benar serta mencegah overfitting atau underfitting.

d. Desain Flowchart:

Merencanakan proses kerja pengembangan model AI dari pengumpulan data hingga implementasi.

3.4 Training Model

Proses training adalah tahapan dimana solusi atau sistem yang telah dirancang dikodekan dan diintegrasikan menjadi sebuah produk yang dapat berjalan. Aktivitas utama dalam proses ini meliputi persiapan data, pengembangan

kode atau perangkat lunak sesuai dengan spesifikasi desain, serta integrasi berbagai komponen sistem. Proses ini juga melibatkan pengujian unit untuk memastikan setiap bagian sistem berfungsi dengan baik sebelum integrasi keseluruhan. Implementasi juga mencakup penerapan solusi atau sistem ke dalam lingkungan produksi yang sesungguhnya serta pelatihan pengguna yang mungkin diperlukan. Keseluruhan, proses implementasi merupakan langkah kritis dalam mengubah desain menjadi produk yang dapat digunakan secara efektif. Pada penelitian ini terdapat beberapa proses implementasi yaitu:

- a. Persiapan data:
Mempersiapkan data pelatihan dan pengujian sesuai dengan kebutuhan model.
- b. Implementasi model:
Implementasikan model SSD MobileNet V2 dengan pendekatan transfer learning menggunakan framework Tensorflow.

3.5 Pengujian Model

Proses pengujian adalah tahap di mana solusi atau sistem yang telah diimplementasikan diuji untuk memastikan bahwa itu berfungsi sebagaimana diharapkan dan memenuhi kebutuhan bisnis serta persyaratan pengguna. Aktivitas dalam proses ini meliputi penyusunan rencana pengujian, pembuatan skenario pengujian, eksekusi pengujian, serta analisis dan pelaporan hasil pengujian. Pengujian dapat mencakup berbagai jenis, termasuk pengujian fungsional, pengujian non-fungsional, pengujian keamanan, dan pengujian kinerja. Tujuan utama dari proses ini adalah untuk mengidentifikasi bug, cacat, atau masalah kinerja yang mungkin ada dalam solusi yang dikembangkan dan memastikan bahwa sistem siap untuk digunakan oleh pengguna akhir.

- a. Uji Fungsional: Melakukan pengujian untuk mengevaluasi fungsional model dalam mendeteksi kerusakan pada PCB.
- b. Uji Validasi: Memverifikasi hasil pengujian untuk memastikan bahwa model memberikan hasil yang akurat dan dapat diandalkan.

3.6 Pemeliharaan

Proses pemeliharaan adalah tahapan dimana solusi atau sistem yang telah diimplementasikan dan diuji dipelihara dan diperbaiki seiring waktu untuk memastikan kinerjanya yang optimal dan sesuai dengan kebutuhan yang terus berubah. Aktivitas dalam proses ini meliputi pemantauan kinerja sistem, perbaikan bug dan cacat yang ditemukan setelah implementasi, peningkatan fungsionalitas berdasarkan umpan balik pengguna, serta penyesuaian dengan perubahan lingkungan atau persyaratan bisnis. Proses pemeliharaan juga mencakup pembaruan perangkat lunak, penanganan keamanan, dan manajemen versi untuk memastikan sistem tetap relevan dan aman

BAB IV RANCANGAN DAN IMPLEMENTASI

4.1. Analisis Kebutuhan Sistem

Proses analisis membantu memastikan bahwa solusi yang dikembangkan akan memenuhi kebutuhan bisnis dan pengguna akhir, serta membantu mengidentifikasi potensi masalah atau risiko yang mungkin timbul selama pengembangan sistem.

4.1.1. Identifikasi Kebutuhan Sistem

Dalam melakukan penelitian dibutuhkan analisa kebutuhan sistem diantaranya :

1. Pemilihan Model Neural Network

Model neural network pada penelitian ini menggunakan pre-trained model dengan jenis model object detection, yang memungkinkan input gambar pada model dan model akan memberikan output berupa klasifikasi dan bounding box.

2. Pemilihan Input Gambar

Input gambar yang digunakan yaitu kamera mikroskopis, kamera tersebut dirasa cocok untuk kebutuhan system dalam menangkap gambar PCB berukuran kecil

4.1.2. Studi Literatur

Pada tahap ini dilakukan peninjauan penelitian yang sudah dilakukan sebelumnya dari berbagai jurnal terkait dengan penelitian ini

4.1.3. Identifikasi Data

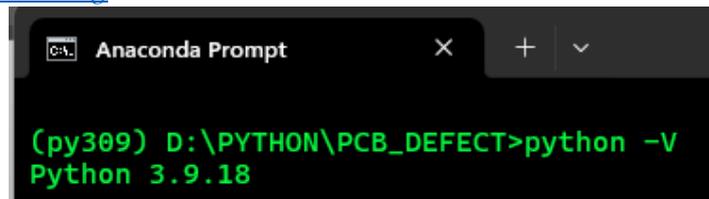
Tahap identifikasi data dilakukan pengumpulan data yang relevan untuk penelitian yaitu data gambar PCB yang rusak. Pengumpulan data dilakukan sebanyak 30 data untuk setiap kelas. Klasifikasi data diantaranya jalur terputus, komponen terbakar dan defect pada solder.

4.1.4. Langkah Penginstalan Framework dan Software

Pada tahap ini dilakukan penginstalan frameworks dan software pendukung yang dibutuhkan dalam proses transfer learning.

1. Menginstall Bahasa Pemrograman Python

Bahasa pemrograman yang digunakan pada penelitian ini adalah Python dengan versi 3.9.0 LTS yang dapat diunduh pada situs resmi <https://python.org>



```

C:\> Anaconda Prompt
(py309) D:\PYTHON\PCB_DEFECT>python -V
Python 3.9.18
```

Gambar 4. Menginstall Bahasa Perograman Python

2. Menginstall Library Tensorflow

Tensorflow merupakan salah satu library python untuk memudahkan proses tranfer learning. Pada penelitian ini digunakan Tensorflow versi 2.7.0 dengan menjalankan perintah `pip install tensorflow==2.7.0` pada command prompt windows.

```
(py309) D:\PYTHON\PCB_DEFECT>pip show tensorflow
Name: tensorflow
Version: 2.7.0
Summary: TensorFlow is an open source machine learning framework for everyone.
Home-page: https://www.tensorflow.org/
Author: Google Inc.
Author-email: packages@tensorflow.org
License: Apache 2.0
Location: c:\programdata\anaconda3\envs\py309\lib\site-packages
Requires: absl-py, astunparse, flatbuffers, gast, google-pasta, grpcio, h5py, keras, keras-preprocessing, libclang, numpy, opt-einsum, protobuf, six, tensorboard, tensorflow-estimator, tensorflow-io-gcs-filesystem, termcolor, typing-extensions, wheel, wrapt
Required-by: deepface, retina-face, tensorflow-text, tf-models-official
```

Gambar 5. Menginstall Library Tensorflow

3. Menginstall Library Object Detection API

Object detection API tersedia pada repositori Github Tensorflow, Library tersebut menyediakan berbagai script salah satunya script training yang digunakan pada penelitian ini yaitu `model_main_tf2.py` dapat diunduh di http://github.com/tensorflow/models/tree/master/research/object_detection

```
(py309) D:\PYTHON\PCB_DEFECT>pip show object-detection
Name: object-detection
Version: 0.1
Summary: Tensorflow Object Detection Library
Home-page:
Author:
Author-email:
License:
Location: c:\programdata\anaconda3\envs\py309\lib\site-packages
Requires: apache-beam, avro-python3, contextlib2, Cython, keras, lvis, lxml, matplotlib, pandas, pillow, pycocotools, pyparsing, sacrebleu, scipy, six, tensorflow-io, tf-models-official, tf-slim
Required-by:
```

Gambar 6. Menginstall Object Detection API

4. Menginstall Library NVIDIA CUDA dan CuDNN

Untuk mempercepat proses training digunakan hardware GPU Nvidia Geforce RTX3050. Library CUDA da CuDNN dapat diunduh pada situs resmi Nvidia. <https://developer.nvidia.com/cudnn>

```
(py309) D:\PYTHON\PCB_DEFECT>nvidia-smi
Mon Jul 8 21:45:57 2024

+-----+-----+-----+
| NVIDIA-SMI 551.86                Driver Version: 551.86          CUDA Version: 12.4          |
+-----+-----+-----+
| GPU  Name                   TCC/WDDM | Bus-Id      Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf              Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+
|   MIG M. |
+-----+-----+-----+
|  0  NVIDIA GeForce RTX 3050 ... WDDM | 00000000:01:00.0 Off |              |
| N/A |
| N/A  48C    P3              10W / 30W |  0MiB / 4096MiB |      0%   Default |
| N/A |
+-----+-----+-----+
```

Gambar 7. Menginstall CUDA dan CuDNN

6. Mengunduh Model Dasar SSD MobileNet V2

Model dasar SSD MobileNet dibutuhkan untuk proses transfer learning, model tersebut dapat diunduh pada Github resmi tensorflow yang bernama TensorFlow 2 Detection Model Zoo https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

7. Menginstall Library LabelImg

Library labelimg digunakan untuk memberikan label pada gambar PCB, library tersebut dapat diinstall menggunakan perintah `pip install labelimg` pada command prompt windows

```
(py309) D:\PYTHON\PCB_DEFECT>pip show labelimg
Name: labelimg
Version: 1.8.6
Summary: LabelImg is a graphical image annotation tool and label object bounding
boxes in images
Home-page: https://github.com/tzutalin/LabelImg
Author: TzuTa Lin
Author-email: tzu.ta.lin@gmail.com
License: MIT license
Location: c:\programdata\anaconda3\envs\py309\lib\site-packages
Requires: lxml, pyqt5
Required-by:
```

Gambar 8. Menginstall Library LabelImg

8. Menginstall Library Pendukung

Beberapa library pendukung lainnya digunakan pada penelitian ini diantaranya OpenCV, Tensorboard, Google Protobuf dan PySimpleGUI. Beberapa library tersebut dapat diinstall dengan perintah `pip install opencv-python tensorboard google-protobuf pysimplegui`

```
(py309) D:\PYTHON\PCB_DEFECT>pip show opencv-python
Name: opencv-python
Version: 4.9.0.80
Summary: Wrapper package for OpenCV python bindings.
Home-page: https://github.com/opencv/opencv-python
Author:
Author-email:
License: Apache 2.0
Location: c:\programdata\anaconda3\envs\py309\lib\site-packages
Requires: numpy, numpy, numpy
Required-by: deepface, lvis, mtcnn, retina-face
```

Gambar 9. Menginstall Library Pendukung

4.2. Desain Software

Proses desain membantu dalam menghasilkan panduan yang jelas untuk implementasi sistem, memastikan bahwa solusi yang dikembangkan sesuai dengan kebutuhan dan tujuan.

4.2.1. Arsitektur Model

Arsitektur model pada penelitian ini menggunakan *pre-trained* SSD MobileNet V2, model tersebut sudah dilatih dengan data gambar umum sehingga pada penelitian ini tidak perlu merangkai ulang arsitektur neural network. SSD MobileNet V2 bersifat *open-source* sehingga dapat digunakan secara bebas oleh peneliti.

4.2.2. Desain Dataset

Pada tahap ini dilakukan pengumpulan, anotasi, pembersihan dan pembagian dataset. Dataset berupa foto PCB yang rusak sebanyak 30 foto setiap klasifikasi.

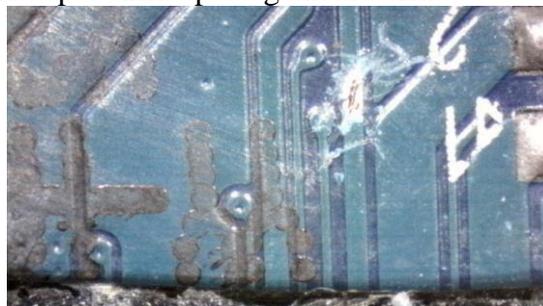
4.2.2.1. Pengumpulan Data

Dilakukan pengumpulan data pada setiap kelas diantaranya jalur terputus, komponen terbakar dan defect pada solder. Pengambilan sampel kerusakan PCB didapatkan dari peralatan elektronik yang mengalami kerusakan. Beberapa sampel PCB dapat dilihat pada gambar dibawah:

Tabel 3. Sampel PCB

No	Gambar PCB	Keterangan
1		Mainboard Xiaomi Mi Mix. Kerusakan mati total terdapat komponen power atau buck converter yang terbakar
2		PCB Model Bolt Kerusakan disebabkan terjatuh dan terdapat beberapa goresan pada jalur PCB
3		Mainboard Hanphone Umi Rome Kerusakan terdapat solder defect pada beberapa komponen resistor dan kapasitor, defect tersebut diakibatkan human error saat dilakukan solder uap.

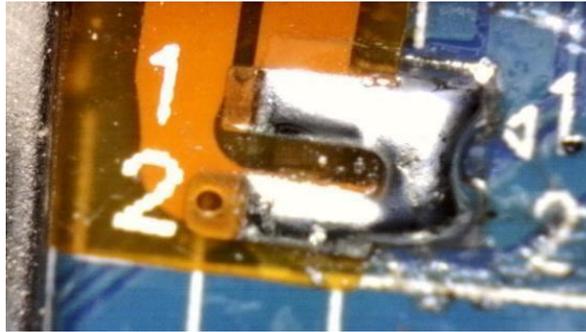
Sampel PCB lebih lengkap dapat dilihat pada lampiran, PCB tersebut dilakukan pemotretan pada beberapa bagian yang mengalami kerusakan, pengambilan gambar dapat dilihat pada gambar berikut:



Gambar 10. Pengumpulan Data Jalur Terputus



Gambar 11. Pengumpulan Data Komponen Terbakar



Gambar 12. Pengumpulan Data Solder Defect

Pada Gambar diatas dapat dilihat proses pengumpulan data berupa gambar PCB yang rusak, Terdapat 3 jenis kerusakan pada PCB yaitu jalur terputus, komponen terbakar dan solder defect. Dataset yang lebih lengkap dapat dilihat pada Lampiran 1 dan Lampiran 2

4.2.2.2. Anotasi Data

Anotasi data dilakukan menggunakan library LabelImg pada program python, anotasi berfungsi untuk menandai suatu objek pada gambar landscape dan menentukan kelasnya



Gambar 13. Proses Anotasi Data

4.2.2.3. Pembersihan dan Pemrosesan Data

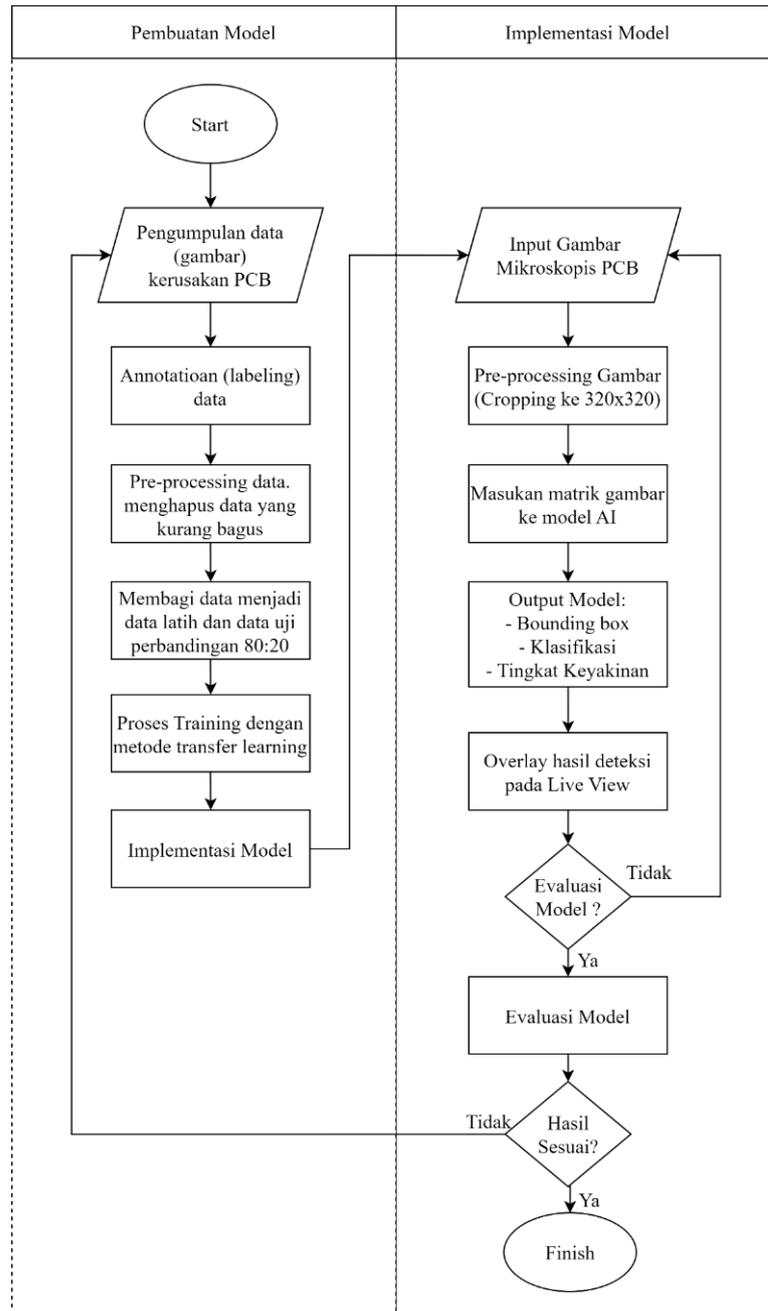
Pada tahap ini dilakukan pembersihan data, menghapus gambar yang dirasa kurang bagus seperti buram atau tidak relevan dengan penelitian. Pada tahap ini juga dilakukan resize pada gambar menjadi 320x320px hal ini dilakukan agar gambar dapat diproses oleh model mengingat input model SSD MobileNet V2 adalah 320x320px.

4.2.2.4. Pembagian Dataset

Dataset gambar yang telah di anotasi dan dibersihkan kemudian dibagi kedalam dua folder train dan test. Pada penelitian ini menggunakan rasio 80% sebagai data train dan 20% sebagai data test.

4.2.3. Desain Flowchart

Merencanakan proses kerja pengembangan model AI dari pengumpulan data hingga implementasi. Desain flowchart dapat dilihat pada gambar berikut:



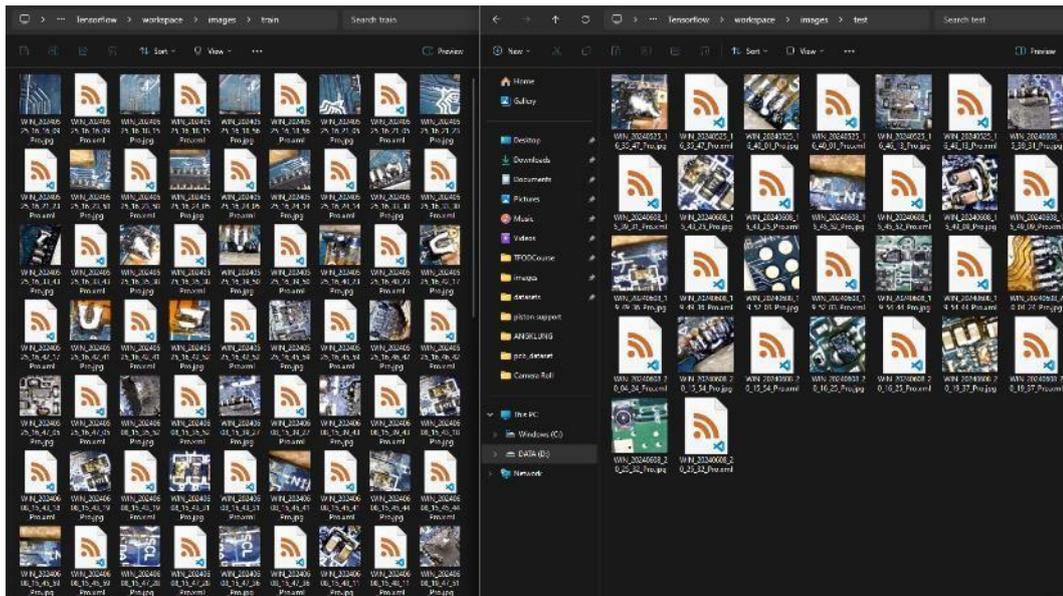
Gambar 14. Desain Flowchart

4.3. Implementasi

Proses implementasi adalah tahapan dimana solusi atau sistem yang telah dirancang dikodekan dan diintegrasikan menjadi sebuah produk yang dapat berjalan

4.3.1. Persiapan Data

Mempersiapkan data dengan membagi kedalam folder train dan folder test dengan perbandingan 20:80



Gambar 15. Persiapan Data

4.3.2. Pelatihan Model

Sebelum melakukan pelatihan model diperlukan persiapan seperti mengatur path folder dan mengatur parameter parameter training. Untuk mempermudah proses training library Tensorflow menyediakan file konfigurasi yang akan diatur selama training file tersebut bernama *pipeline.config* yang dapat dilihat pada lampiran 4. Adapun parameter yang disesuaikan pada file konfigurasi dapat dilihat pada tabel berikut:

Tabel 4. Parameter Training

No	Nama Parameter	Nilai	Ket
1	<i>num_classes</i>	3	Banyaknya kelas
2	<i>height</i>	320	Tinggi gambar
3	<i>width</i>	320	Lebar gambar
4	<i>batch_size</i>	8	Banyaknya gambar yang diproses dalam 1 waktu
5	<i>learning_rate_base</i>	0.08	Tingkat pembelajaran
6	<i>warmup_learning_rate</i>	0.026666	Tingkat pembelajaran pada 100 step pertama
7	<i>fine_tune_checkpoint</i>	Tensorflow\\workspace\\pre-trained-models\\ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8\\checkpoint\\ckpt-0	Path menunjukan model dasar
8	<i>train_input_reader</i>	Tensorflow\\workspace\\annotations\\train.record	Path menunjukan folder train
9	<i>eval_input_reader</i>	Tensorflow\\workspace\\annotations\\test.record	Path menunjukan folder test
10	<i>label_map_path</i>	Tensorflow\\workspace\\annotations\\label_map.pbtxt	Path menunjukan file list label

Setelah parameter diatur dilanjutkan proses training dengan pada command prompt windows dengan perintah sebagai berikut:

```
python model_main_tf2.py model_dir=[output] pipeline_config_path=[file_konfigurasi] num_train_steps=[banyaknya_step]
```

Perintah diatas menunjukkan bahwa proses training tersebut menjalankan script python model_main_tf2.py Adapun argument yang ditambahkan pada perintah tersebut dapat dilihat pada tabel dibawah:

Tabel 5. Argument Training

No	Nama Argument	Nilai	Ket
1	<i>model_dir</i>	Tensorflow\workspace\model s\v6	Folder output hasil model
2	<i>pipeline_config_path</i>	Tensorflow\workspace\model s\v6\pipeline.config	File konfigurasi pipeline.config
3	<i>num_train_step</i>	20000	Banyaknya step training

Dengan menjalankan perintah tersebut maka proses transfer learning dapat berjalan

```
2020-06-08 21:10:10.688375: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 1635 MB memory: -> device: 0, name: NVIDIA GeForce RTX 3050 Ti Laptop GPU, pci bus id: 0000:01:00:00, compute capability: 8.6
INFO:tensorflow:Using MirroredStrategy with devices ('/job:localhost/replica:0/task:0/device:GPU:0',)
I0608 21:10:10.695946: 7988 mirrored_strategy.py:376] Using MirroredStrategy with devices ('/job:localhost/replica:0/task:0/device:GPU:0',)
INFO:tensorflow:Maybe overwriting train_steps: 20000
I0608 21:10:10.706321: 7988 config_util.py:552] Maybe overwriting train_steps: 20000
INFO:tensorflow:Maybe overwriting use_bfloat16: False
I0608 21:10:10.708138: 7988 config_util.py:552] Maybe overwriting use_bfloat16: False
WARNING:tensorflow:From C:\ProgramData\anaconda3\envs\py309\lib\site-packages\object_detection\model_lib_v2.py:563: StrategyBase.experimental_distribute_datasets_from_function (from tensorflow.python.distribute_lib) is deprecated and will be removed in a future version.
Instructions for updating:
rename to distribute_datasets_from_function
I0608 21:10:10.737034: 7988 deprecation.py:341] From C:\ProgramData\anaconda3\envs\py309\lib\site-packages\object_detection\model_lib_v2.py:563: StrategyBase.experimental_distribute_datasets_from_function (from tensorflow.python.distribute_lib) is deprecated and will be removed in a future version.
Instructions for updating:
rename to distribute_datasets_from_function
INFO:tensorflow:Reading unweighted datasets: ['Tensorflow\workspace\annotations\train_record']
I0608 21:10:10.747242: 7988 dataset_builder.py:162] Reading unweighted datasets: ['Tensorflow\workspace\annotations\train_record']
INFO:tensorflow:Reading record datasets for input file: ['Tensorflow\workspace\annotations\train_record']
I0608 21:10:10.747242: 7988 dataset_builder.py:79] Reading record datasets for input file: ['Tensorflow\workspace\annotations\train_record']
INFO:tensorflow:Number of filenames to read: 1
I0608 21:10:10.747242: 7988 dataset_builder.py:80] Number of filenames to read: 1
```

Gambar 16. Pelatihan Model

4.4. Testing

Pada tahap ini dilakukan pengujian pada model apakah model dapat mendeteksi kerusakan PCB berdasarkan dataset yang telah di training sebelumnya. Testing menggunakan kode sederhana untuk menangkap gambar dari kamera mikroskop lalu gambar tersebut di proses menggunakan model yang telah di training.

```
cap = cv2.VideoCapture(1)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while cap.isOpened():
    ret, frame = cap.read()
    image_np = np.array(frame)
    # cv2.imshow('obj', frame)

    input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
    detections = detect_fn(input_tensor)

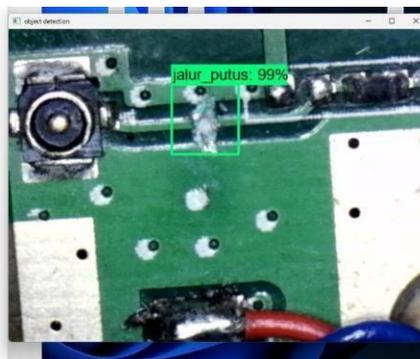
    num_detections = int(detections.pop('num_detections'))
    detections = {key: value[0, num_detections: num_detections+1] for key, value in detections.items()}
    detections['num_detections'] = num_detections

    # detection_classes should be ints.
    detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

    label_id_offset = 1
    image_np_with_detections = image_np.copy()

    viz_utils.visualize_boxes_and_labels_on_image_array(
        image_np_with_detections,
        detections['detection_boxes'],
        detections['detection_classes']+label_id_offset,
        detections['detection_scores'],
        category_index,
        max_to_draw=10,
        min_score_thresh=.8,
        agnostic_mode=False)

    cv2.imshow('object detection', cv2.resize(image_np_with_detections, (0, 0),
```



Gambar 17. Testing

BAB V HASIL DAN PEMBAHASAN

5.1 Hasil Penelitian

Pada tahap sebelumnya telah dijelaskan proses perancangan dan implementasi penelitian yang telah dilakukan. Hasil penelitian ini penulis menyelesaikan beberapa hal yang menjadi acuan referensi untuk mendapatkan hasil yang maksimal dengan desain model seefisien mungkin. Berikut gambar alat yang telah dibuat :



Gambar 18. Tampilan Alat

5.2 Test Fungsional Keseluruhan Sistem (*Overall Testing*)

Tahapan ini dilakukan pengujian seluruh fungsi sistem, dari pengujian kamera mikroskopis, model SSD MobileNet hingga kode program. Uji keseluruhan meliputi uji fungsional, dan uji validasi.

5.2.1 Pengujian Struktural

Pada tahapan ini dilakukan pengujian yang bertujuan untuk mengetahui apakah setiap unit kode program dan model sudah sesuai.

5.2.1.1 Pengujian Model AI

Dilakukan pengujian model AI untuk mengetahui apakah checkpoint model yang telah di training dapat digunakan.

8. Load Train Model From Checkpoint

```
In [14]: import os
import tensorflow as tf
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.builders import model_builder
from object_detection.utils import config_util

In [15]: # Load pipeline config and build a detection model
configs = config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])
detection_model = model_builder.build(model_config=configs['model'], is_training=False)

# Restore checkpoint
ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'], 'ckpt-21')).expect_partial()

print(ckpt)

@tf.function
def detect_fn(image):
    image, shapes = detection_model.preprocess(image)
    prediction_dict = detection_model.predict(image, shapes)
    detections = detection_model.postprocess(prediction_dict, shapes)
    return detections

<tensorflow.python.training.tracking.util.Checkpoint object at 0x0000011f19a10910>
```

Gambar 19. Pengujian Model AI

Pada gambar diatas dapat dilihat bahwa checkpoint model AI berhasil dibaca oleh library Tensorflow menandakan model AI siap digunakan.

5.2.1.2 Pengujian Kode Program

Pengujian dilakukan untuk mengetahui apakah kode program dapat berjalan dengan baik.

10. Real Time Detections

```
In [*]: cap = cv2.VideoCapture(1)

width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while cap.isOpened():
    ret, frame = cap.read()
    image_np = np.array(frame)

    input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
    detections = detect_fn(input_tensor)

    num_detections = int(detections.pop('num_detections'))
    detections = {key: value[0, :num_detections].numpy()
                  for key, value in detections.items()}
    detections['num_detections'] = num_detections

    # detection_classes should be ints.
    detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

    label_id_offset = 1
    image_np_with_detections = image_np.copy()

    viz_utils.visualize_boxes_and_labels_on_image_array(
        image_np_with_detections,
        detections['detection_boxes'],
        detections['detection_classes']+label_id_offset,
        detections['detection_scores'],
        category_index,
        use_normalized_coordinates=True,
        max_boxes_to_draw=5,
        min_score_thresh=.5,
        agnostic_mode=False)

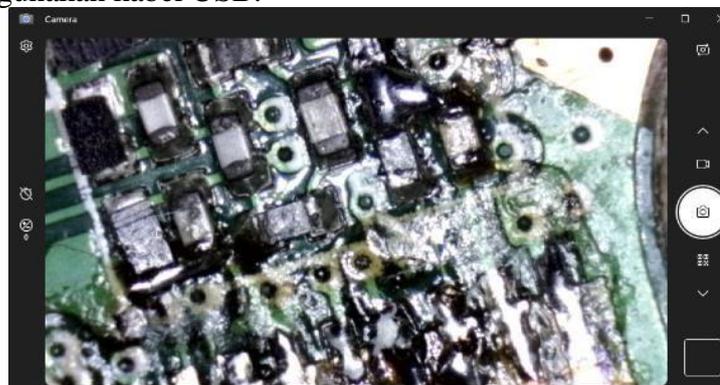
    cv2.imshow('object detection', cv2.resize(image_np_with_detections, (800, 600)))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        cap.release()
        cv2.destroyAllWindows()
        break
```

Gambar 20. Pengujian Kode Program

5.2.2 Pengujian Fungsional

Dilakukan pengujian fungsional untuk mengetahui apakah komponen dapat bekerja dengan baik. Beberapa komponen yang diuji yaitu kamera mikroskopis. Pengujian dilakukan dengan menggunakan menghubungkan kamera mikroskop ke laptop menggunakan kabel USB.



Gambar 21. Pengujian Kamera Mikroskop

Pada pengujian yang dilakukan pada kamera mikroskop dapat dilihat bahwa kamera dapat terhubung dan mengirimkan gambar dengan baik

5.2.2.1 Pengujian Keseluruhan Sistem

Setelah beberapa rangkaian pengujian yang telah dilakukan pada setiap komponen yang ada, maka tahap selanjutnya akan dilakukan pengujian keseluruhan pada sistem yang dibuat. Pengujian dilakukan dengan mengintegrasikan semua komponen seperti kamera mikroskopis, model AI yang telah di training dan kode program pendukung.



Gambar 22. Pengujian Keseluruhan Sistem

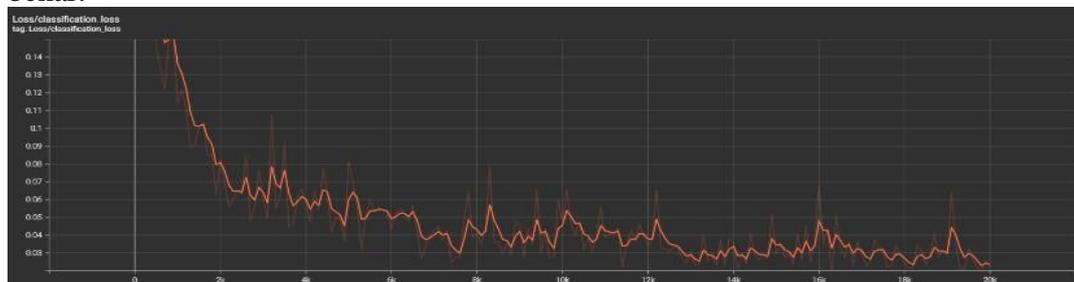
Pada gambar diatas dapat dilihat bahwa system dapat berjalan dengan baik dan sudah dapat mendeteksi kerusakan untuk pertama kali.

5.2.3 Uji Validasi

Tahap ini bertujuan untuk menguji kinerja model AI SSD MobileNet V2, apakah model dapat dilatih dengan baik dan dapat mendeteksi kerusakan PCB. Pengujian dilakukan menggunakan library Tensorboard untuk mengetahui kinerja model selama dilatih, dan pengujian juga dilakukan dengan cara mendeteksi langsung kerusakan PCB menggunakan kamera mikroskopis.

5.2.3.1 Uji Coba Validasi Classification Loss

Uji coba validasi classification loss bertujuan untuk menilai kemampuan model dalam menghasilkan prediksi yang akurat. Proses ini melibatkan evaluasi model menggunakan dataset validasi yang berbeda dari dataset pelatihan. Selama uji coba, metrik seperti loss dan akurasi dihitung untuk menentukan kinerja model. Classification loss menunjukkan seberapa baik model memprediksi kelas yang benar.



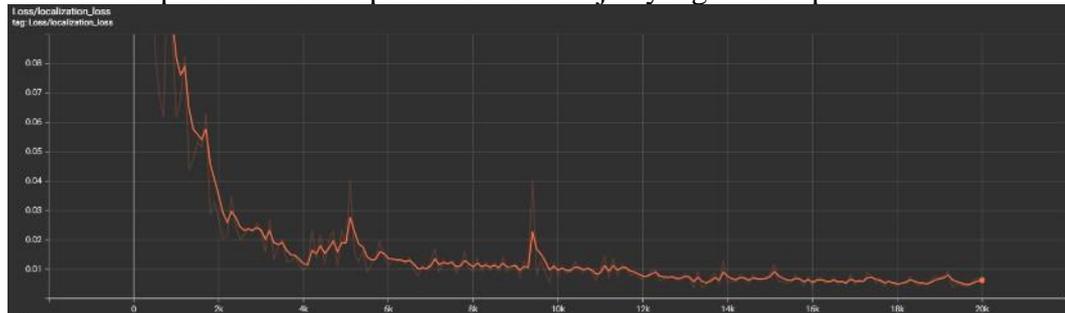
Gambar 23. Classification Loss

Grafik menunjukkan penurunan loss secara signifikan dari awal hingga iterasi ke-20k. Penurunan yang stabil dan konsisten pada grafik menunjukkan bahwa

model belajar dengan baik, mengurangi kesalahan prediksi seiring waktu dengan nilai akhir **0.02352** . Hal ini menunjukkan bahwa model memiliki performa yang baik dalam mengklasifikasikan data sesuai dengan label yang diharapkan.

5.2.3.2 Uji Coba Validasi Localization Loss

Uji coba validasi localization loss bertujuan untuk mengevaluasi kemampuan model dalam menentukan lokasi objek secara akurat. Dalam proses ini, model diuji menggunakan dataset validasi yang mengandung anotasi lokasi objek. Localization loss mengukur seberapa dekat prediksi model dengan lokasi objek yang sebenarnya. Evaluasi ini penting untuk mendeteksi kesalahan dalam prediksi posisi dan ukuran bounding box. Hasilnya membantu dalam mengidentifikasi area perbaikan pada model serta melakukan penyesuaian hyperparameter yang diperlukan, sehingga model mampu memberikan prediksi lokasi objek yang lebih tepat.

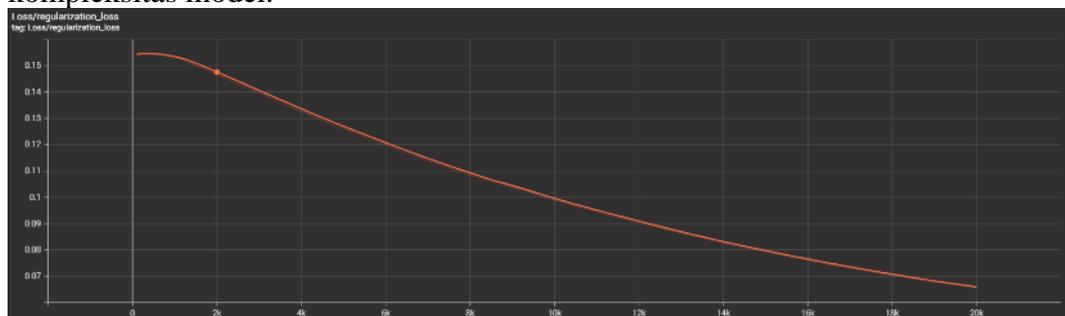


Gambar 24. Localization Loss

Grafik menunjukkan penurunan localization loss secara signifikan dari awal hingga iterasi ke-20k. Pada awal pelatihan, loss mengalami fluktuasi yang cukup tinggi, namun seiring bertambahnya iterasi, loss cenderung stabil dan menurun secara konsisten dengan nilai akhir **0.006221**. Penurunan ini menunjukkan bahwa model semakin baik dalam memprediksi posisi objek dengan tepat. Meskipun terdapat beberapa lonjakan kecil, tren keseluruhan tetap menurun, mengindikasikan peningkatan kinerja model dalam hal lokalisasi.

5.2.3.3 Uji Coba Validasi Regularization Loss

Uji coba ini bertujuan untuk mengevaluasi efektivitas model dalam mengurangi regulasi kesalahan selama proses pelatihan. Regularization loss sering digunakan untuk mencegah overfitting dengan menambahkan penalti pada kompleksitas model.



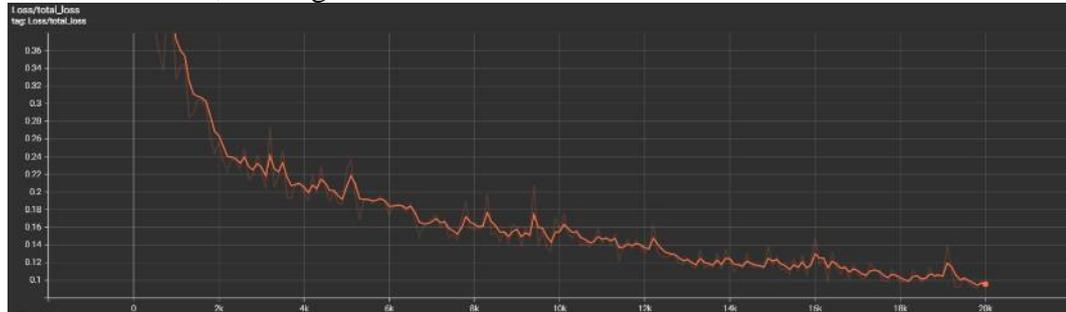
Gambar 25. Regularization Loss

Grafik menunjukkan penurunan regularization loss secara signifikan dari awal hingga iterasi ke-20k. Pada awal pelatihan, terdapat fluktuasi yang cukup tinggi, tetapi seiring dengan bertambahnya iterasi, loss cenderung menurun dan

stabil dengan nilai akhir **0.06596**. Penurunan yang konsisten ini menunjukkan bahwa model berhasil mengurangi kompleksitas dan kesalahan regulasi secara efektif.

5.2.3.4 Uji Coba Validasi Total Loss

Uji coba ini bertujuan untuk mengevaluasi performa keseluruhan model dengan melihat total loss, yang merupakan kombinasi dari classification loss, localization loss, dan regulation loss.



Gambar 26. Total Loss

Grafik menunjukkan penurunan total loss yang signifikan dari awal hingga iterasi ke-20k. Pada awal pelatihan, total loss mengalami fluktuasi yang cukup tinggi, namun seiring bertambahnya iterasi, total loss cenderung menurun dan stabil secara konsisten dengan nilai akhir **0.0957**. Penurunan yang stabil ini mengindikasikan bahwa model secara keseluruhan semakin baik dalam melakukan klasifikasi, lokalisasi, dan regulasi. Meskipun terdapat beberapa lonjakan kecil, tren keseluruhan menunjukkan penurunan yang signifikan, menandakan bahwa model belajar dengan baik dan mengurangi kesalahan prediksi secara efektif.

5.2.3.5 Uji Coba Validasi Inspeksi Visual

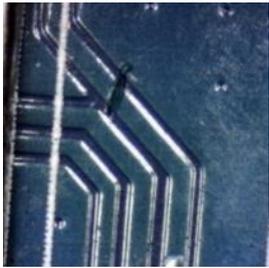
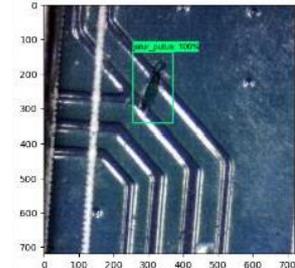
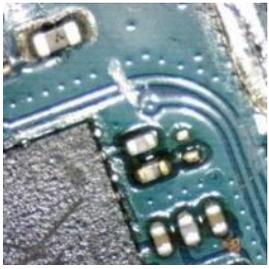
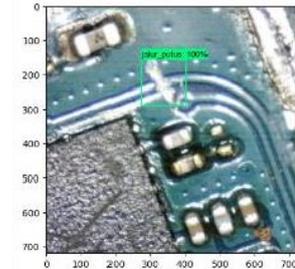
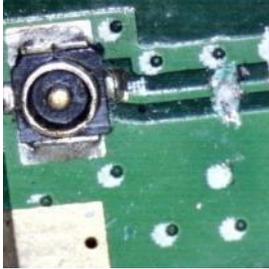
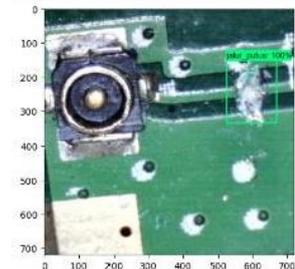
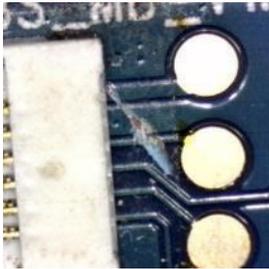
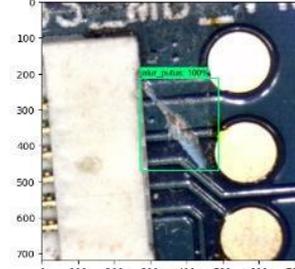
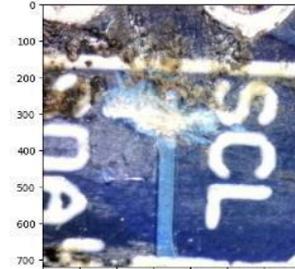
Pengujian dilakukan dengan cara memeriksa kondisi PCB secara visual untuk mengidentifikasi kerusakan seperti jalur terputus, solder defect, dan komponen yang terbakar. Selanjutnya, hasil inspeksi visual dibandingkan dengan sistem deteksi menggunakan model AI. Tujuannya adalah untuk mengevaluasi kinerja AI dalam mendeteksi kerusakan PCB dibandingkan dengan metode inspeksi visual konvensional.

1. Deteksi Jalur Terputus oleh SSDMobileNet V2

Jalur terputus pada PCB dapat diakibatkan oleh benturan fisik atau goresan selama pemakaian, serta cacat produksi yang mungkin terjadi selama proses pembuatan komponen elektronik. Contoh kasus jalur terputus dapat dilihat pada tabel berikut

Tabel 6. Inspeksi Visual Jalur Terputus

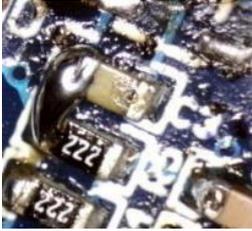
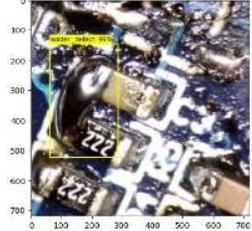
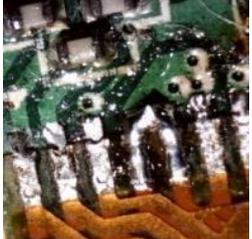
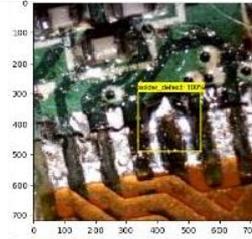
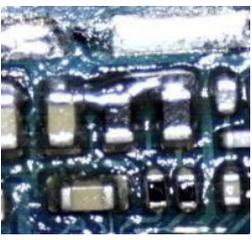
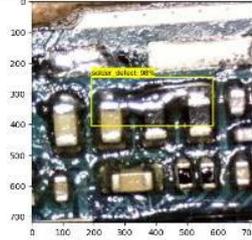
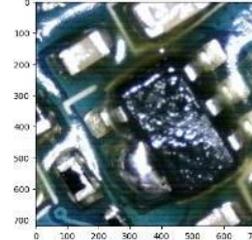
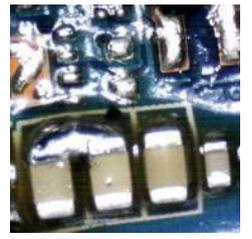
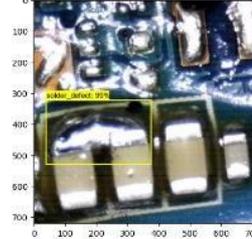
No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
----	-----------------	-----------	---------------	-----

1		Jalur terputus karena cacat produksi (PCB MPU6050)		Terdeteksi
2		Terdapat goresan pada jalur untuk IC Mediatek (PCB Handphone)		Terdeteksi
3		Terdapat jalur terputus di dekat konektor antenna (PCB Handphone)		Terdeteksi
4		Terdapat goresan pada jalur untuk konektor tombol (PCB Handphone)		Terdeteksi
...
10		Terdapat cacat pada jalur SCL (PCB MPU6050)		Tidak Terdeteksi

2. Deteksi Solder Defect oleh SSDMobileNet V2

Uji coba validasi deteksi kerusakan PCB solder defect bertujuan untuk mengevaluasi kemampuan model dalam mengenali cacat solder pada PCB. Cacat solder pada PCB dapat ditemui di beberapa komponen yang disebabkan komponen terlalu dengan dengan komponen lainnya dan timah solder terlalu banyak sehingga timah solder menggumpal dan mengakibatkan *shortcircuit* Adapun beberapa contoh kasus solder defect dapat dilihat pada tabel berikut

Tabel 7. Inspeksi Visual Solder Defect

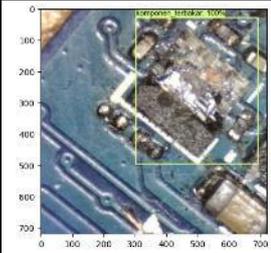
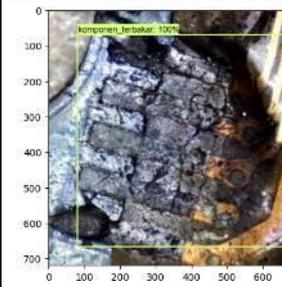
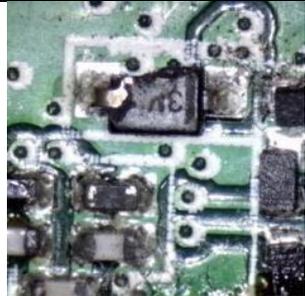
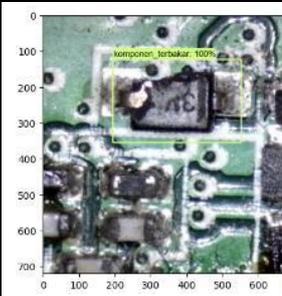
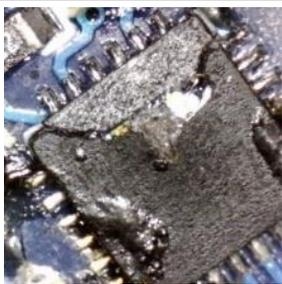
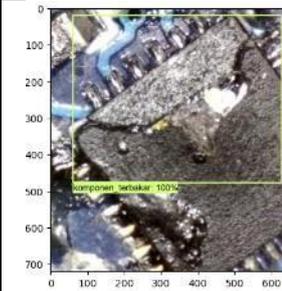
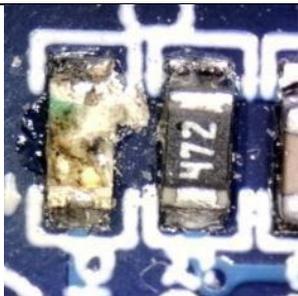
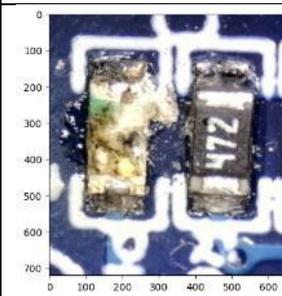
No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
1		Kapasitor dan resistor terdapat timah yang menggumpal		Terdeteksi
2		Flexible konektor terdapat timah yang menggumpal		Terdeteksi
3		Terdapat tiga buah komponen yang menyatu yaitu satu buah kapasitor dan dua buah resistor		Terdeteksi
4		Terdapat cacat solder pada komponen buck converter 3.3volt		Tidak Terdeteksi
...
10		Terdapat cacat solder pada komponen kapasitor		Terdeteksi

3. Deteksi Komponen Terbakar oleh SSDMobileNet V2

Uji coba validasi deteksi kerusakan PCB komponen terbakar bertujuan untuk mengevaluasi kemampuan model dalam mengenali tanda-tanda kerusakan akibat komponen yang terbakar pada komponen PCB. Contoh kasus komponen terbakar dapat dilihat pada tabel berikut

Tabel 8. Inspeksi Visual Komponen Terbakar

No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
----	-----------------	-----------	---------------	-----

1		Komponen terbakar IC power PM8937 (PCB Handphone)		Terdeteksi
2		Flexible konektor terbakar		Terdeteksi
3		Terdapat komponen dioda 3M terbakar		Terdeteksi
4		IC MPU 6050 terbakar		Terdeteksi
...
10		LED SMD terbakar		Tidak Terdeteksi

Pada pengujian masing masing kelas diatas dapat dilihat bahwa pengujian dilakukan 10 kali pada setiap kelas dengan total 30 kali pengujian. Pengujian selengkapnya dapat dilihat pada lampiran. Sebagian besar kerusakan dapat dideteksi dengan baik oleh model AI, untuk memperjelas pengujian dapat dilihat pada tabel berikut

Tabel 9. Perbandingan Inspeksi Visual dan Model SSD MobileNet V2

No	Kelas	Inspeksi Visual	Hasil Deteksi Menggunakan SSD MobileNet V2	Score (%)
1	Jalur Terputus	Jalur terputus karena cacat produksi (PCB MPU6050)	Terdeteksi dengan kelas Jalur Terputus	100
2	Jalur Terputus	Terdapat goresan pada jalur untuk IC Mediatek (PCB Handphone)	Terdeteksi dengan kelas Jalur Terputus	100
3	Jalur Terputus	Terdapat jalur terputus di dekat konektor antenna (PCB Handphone)	Terdeteksi dengan kelas Jalur Terputus	100
4	Jalur Terputus	Terdapat goresan pada jalur untuk konektor tombol (PCB Handphone)	Terdeteksi dengan kelas Jalur Terputus	100
5	Jalur Terputus	Terdapat cacat pada jalur <i>Serial Clock Line</i> (PCB MPU6050)	Tidak Terdeteksi	0
6	Jalur Terputus	Terdapat cacat pada jalur	Terdeteksi dengan kelas Jalur Terputus	100
7	Jalur Terputus	Terdapat cacat pada jalur	Terdeteksi dengan kelas Jalur Terputus	70
8	Jalur Terputus	Terdapat cacat pada jalur	Terdeteksi dengan kelas Jalur Terputus	67
9	Jalur Terputus	Terdapat cacat pada jalur	Tidak Terdeteksi	0
10	Jalur Terputus	Terdapat cacat pada jalur konektor	Terdeteksi dengan kelas Jalur Terputus	100
11	Solder Defect	Kapasitor dan resistor terdapat timah yang menggumpal	Terdeteksi dengan kelas Solder Defect	99
12	Solder Defect	Flexible konektor terdapat timah yang menggumpal	Terdeteksi dengan kelas Solder Defect	100
13	Solder Defect	Terdapat komponen yang menyatu yaitu kapasitor dan dua buah resistor	Terdeteksi dengan kelas Solder Defect	98
14	Solder Defect	Terdapat cacat solder pada komponen <i>buck converter</i> 3.3volt	Tidak Terdeteksi	0
15	Solder Defect	Terdapat cacat solder pada komponen kapasitor	Terdeteksi dengan kelas Solder Defect	98
16	Solder Defect	Terdapat cacat solder pada konektor	Terdeteksi dengan kelas Solder Defect	100
17	Solder Defect	Terdapat cacat solder pada konektor	Terdeteksi dengan kelas Solder Defect	100
18	Solder Defect	Terdapat cacat solder pada konektor LED	Terdeteksi dengan kelas Solder Defect	100

19	Solder Defect	Terdapat cacat solder pada IC Power	Terdeteksi dengan kelas Solder Defect	70
20	Solder Defect	Terdapat cacat solder pada resistor dan kapasitor	Tidak Terdeteksi	0
21	Komponen Terbakar	Komponen terbakar IC power PM8937 (PCB Handphone)	Terdeteksi dengan kelas Komponen Terbakar	100
22	Komponen Terbakar	Flexible konektor terbakar	Terdeteksi dengan kelas Komponen Terbakar	100
23	Komponen Terbakar	Terdapat komponen dioda 3M terbakar	Terdeteksi dengan kelas Komponen Terbakar	100
24	Komponen Terbakar	IC MPU6050 terbakar	Terdeteksi dengan kelas Komponen Terbakar	100
25	Komponen Terbakar	LED SMD terbakar	Tidak Terdeteksi	0
26	Komponen Terbakar	IC Power Terbakar	Terdeteksi dengan kelas Komponen Terbakar	100
27	Komponen Terbakar	LED SMD terbakar	Terdeteksi dengan kelas Komponen Terbakar	98
28	Komponen Terbakar	IC Terbakar	Terdeteksi dengan kelas Komponen Terbakar	99
29	Komponen Terbakar	Dioda terbakar	Tidak Terdeteksi	0
30	Komponen Terbakar	IC Regulator terbakar	Terdeteksi dengan kelas Komponen Terbakar	100
Rata Rata Hasil Deteksi				76.63

Pada tabel diatas dapat dilihat rata rata *score* hasil deteksi pada semua kelas yaitu **76.63%**.

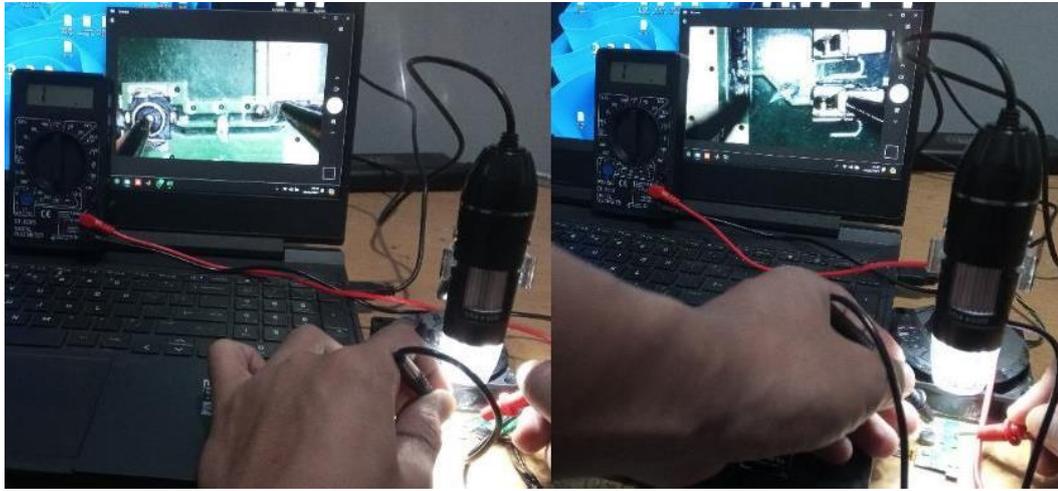
5.2.3.6 Uji Coba Validasi Inspeksi Pengukuran Kelistrikan

Inspeksi pengukuran kelistrikan dilakukan untuk bertujuan untuk memastikan apakah kerusakan pada PCB mengakibatkan tidak berfungsinya PCB tersebut. Pengujian dilakukan dengan cara membandingkan hasil deteksi menggunakan model AI SSD MobileNet V2 dengan inspeksi mendalam menggunakan multimeter untuk mengukur berbagai macam parameter kelistrikan seperti arus, voltase dan resistansi. Inspeksi kelistrikan dapat dilihat pada gambar dibawah



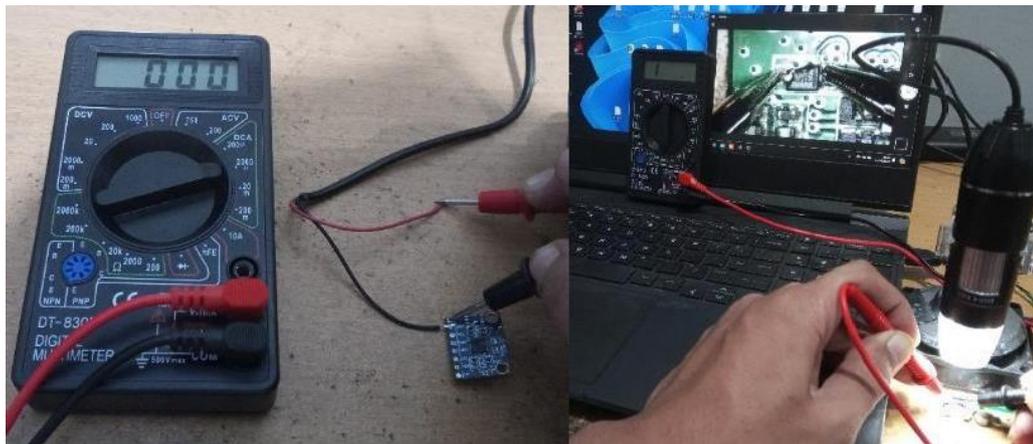
Gambar 27. Inspeksi Kelistrikan Solder Defect

Inspeksi kelistrikan pada solder defect dilakukan dengan mengecek resistansi komponen yang terdeteksi solder defect. Jika resistansi antar komponen mendekati 0 maka diasumsikan bahwa komponen tersebut mengalami *short circuit*.



Gambar 28. Inspeksi Kelistrikan Jalur Terputus

Inspeksi kelistrikan pada jalur terputus dilakukan dengan mengecek resistansi jalur PCB yang terdeteksi. Jika multimeter menunjukkan angka 1 maka jalur tersebut terputus.



Gambar 29. Inspeksi Kelistrikan Komponen Terbakar

Inspeksi kelistrikan pada komponen terbakar dilakukan dengan cara mengukur resistansi pada komponen dioda atau LED, komponen tersebut dapat mengalirkan arus satu arah jika tidak ada kerusakan. Selain itu untuk komponen IC dilakukan dengan cara mengukur konsumsi arus listrik menggunakan ampere meter, jika ampere meter mendekati 0 maka IC tersebut tidak berfungsi. Percobaan diatas dapat dilihat pada tabel berikut.

Tabel 10. Perbandingan Inspeksi Kelistrikan dan Model SSD MobileNet V2

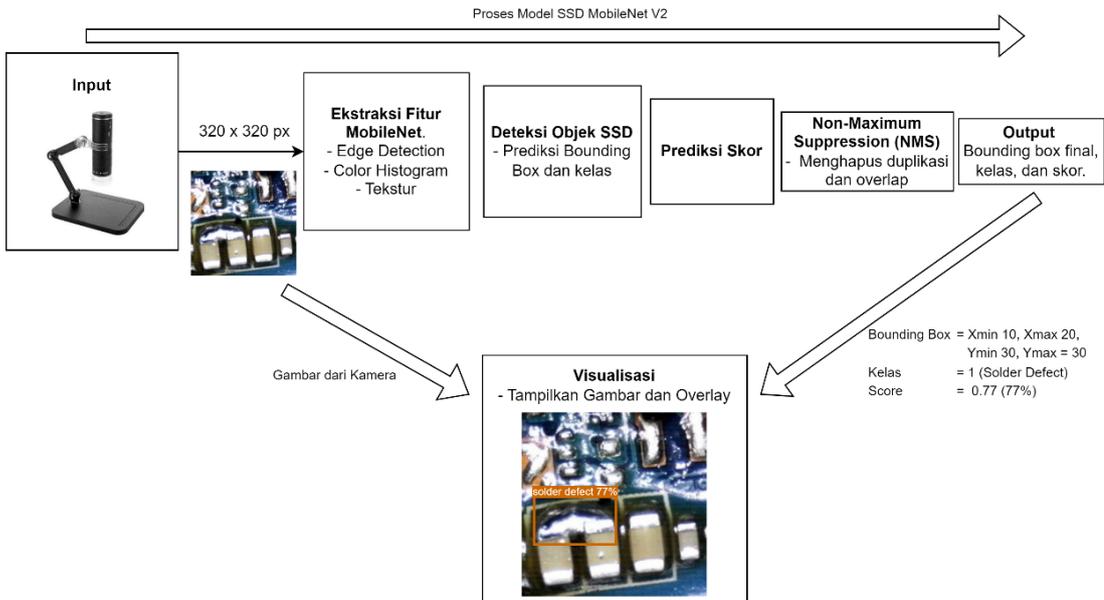
No	Komponen	Inspeksi Kelistrikan	Kesimpulan Inspeksi Kelistrikan	Hasil Deteksi Menggunakan SSD MobileNet V2	Ket
1	Jalur terputus karena cacat produksi (PCB MPU6050)	Multimeter menunjukkan angka 1	jalur putus	Terdeteksi dengan kelas Jalur Terputus	Sesuai
2	Terdapat goresan pada jalur untuk IC Mediatek (PCB Handphone)	Multimeter menunjukkan angka 1	jalur putus	Terdeteksi dengan kelas Jalur Terputus	Sesuai
3	Terdapat jalur terputus di dekat konektor antenna (PCB Handphone)	Multimeter menunjukkan angka 1	jalur putus	Terdeteksi dengan kelas Jalur Terputus	Sesuai
4	Terdapat goresan pada jalur untuk konektor tombol (PCB Handphone)	Multimeter menunjukkan angka 1	jalur putus	Terdeteksi dengan kelas Jalur Terputus	Sesuai
5	Terdapat cacat pada jalur SCL (PCB MPU6050)	Multimeter menunjukkan angka 1	jalur putus	Tidak Terdeteksi	Tidak Sesuai
6	Kapasitor dan resistor terdapat timah yang menggumpal	Multimeter menunjukkan resistansi 0,5	short circuit	Terdeteksi dengan kelas Solder Defect	Sesuai
7	Flexible konektor terdapat timah yang menggumpal	Multimeter menunjukkan resistansi 0,8	short circuit	Terdeteksi dengan kelas Solder Defect	Sesuai
8	Terdapat tiga buah komponen yang menyatu yaitu satu buah kapasitor dan dua buah resistor	Multimeter menunjukkan resistansi 0,7	short circuit	Terdeteksi dengan kelas Solder Defect	Sesuai
9	Terdapat cacat solder pada komponen buck converter 3.3volt	Multimeter menunjukkan resistansi 0,4	short circuit	Tidak Terdeteksi	Tidak Sesuai
10	Terdapat cacat solder pada komponen kapasitor	Multimeter menunjukkan resistansi 0,5	short circuit	Terdeteksi dengan kelas Solder Defect	Sesuai
11	Komponen terbakar IC power PM8937 (PCB Handphone)	Multimeter menunjukkan arus 0,1 mA	komponen terbakar	Terdeteksi dengan kelas Komponen Terbakar	Sesuai
12	Flexible konektor terbakar	Multimeter menunjukkan angka 1	komponen terbakar	Terdeteksi dengan kelas Komponen Terbakar	Sesuai

13	Terdapat komponen dioda 3M terbakar	Multimeter menunjukkan angka 1	komponen terbakar	Terdeteksi dengan kelas Komponen Terbakar	Sesuai
14	IC MPU 6050 terbakar	Multimeter menunjukkan arus 0 mA	komponen terbakar	Terdeteksi dengan kelas Komponen Terbakar	Sesuai
15	LED SMD terbakar	Multimeter menunjukkan angka 1	komponen terbakar	Tidak Terdeteksi	Tidak Sesuai

Pada tabel diatas dapat dilihat bahwa dari 15 inspeksi kelistrikan yang dilakukan terdapat 3 kerusakan yang tidak terdeteksi oleh sistem, dengan kata lain sistem dapat mendeteksi **80%** dari total kerusakan sebenarnya.

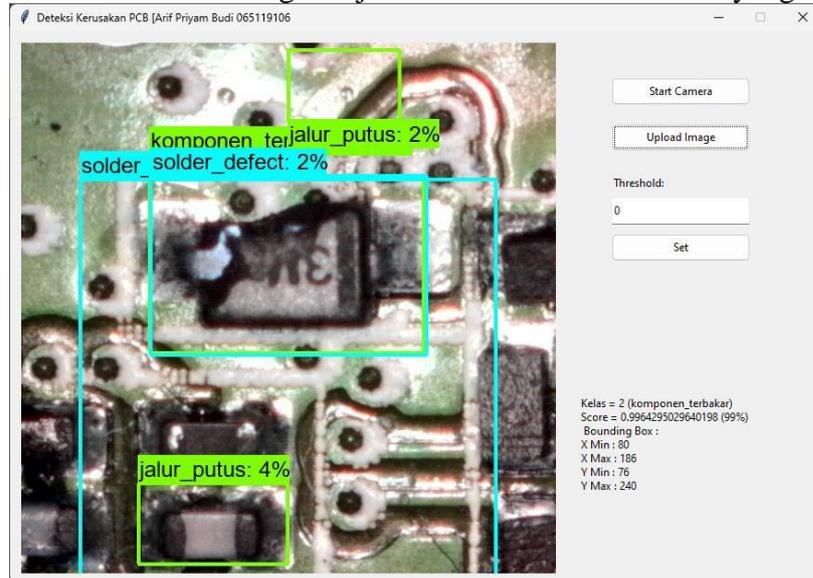
5.2.3.7 Uji Coba Cara Kerja SSD MobileNet V2

Cara kerja model SSD MobileNet V2 memiliki langkah utama yaitu Input Gambar, Ekstraksi Fitur, Deteksi Objek, Prediksi, NMS (*Non-Maximum Suppression*) dan Output. Pada penelitian ini digunakan input gambar dari kamera mikroskopis yang menangkap landscape PCB. Untuk memperjelas cara kerja sistem dapat dilihat pada gambar berikut.



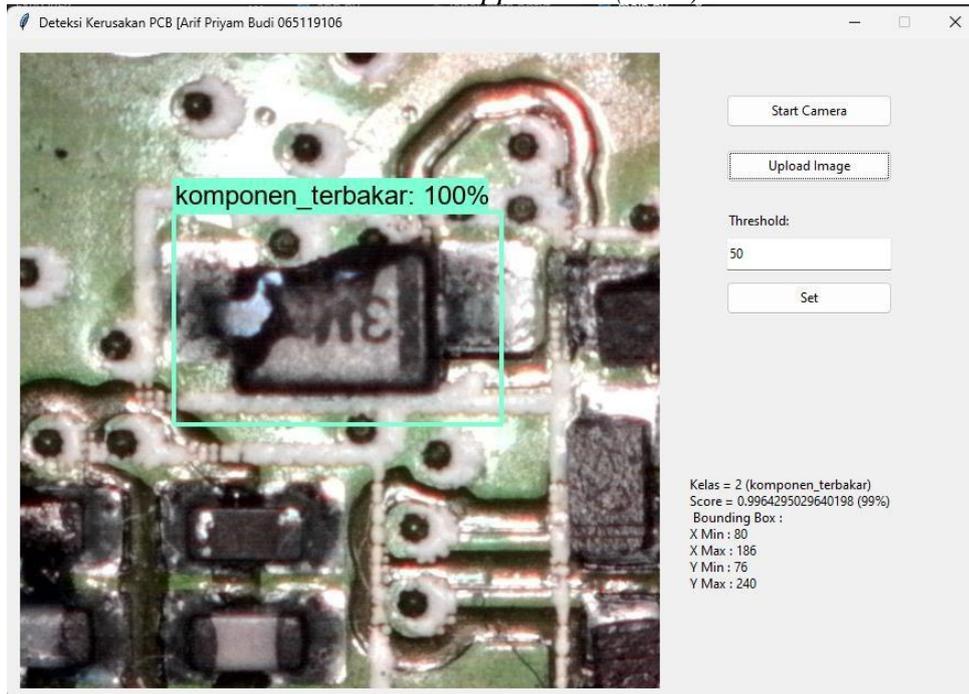
Pada gambar diatas dapat dilihat cara kerja system yang dimulai dengan tangkapan gambar PCB dari kamera mikroskop, gambar tersebut kemudian di resize menjadi 320x320 pixel agar dapat diproses oleh model. Model melakukan ekstraksi fitur diantaranya garis tepi, warna tekstur dan pola. Langkah selanjutnya deteksi bounding box dan kelas. Dilanjutkan dengan prediksi score dan Non-Maximum Suppression NMS untuk menghapus duplikasi bounding box. Lalu bagian output dari model merupakan nilai array bounding box [xmin, xmax, ymin, ymax] nilai kelas 0 sampai 2, dan nilai score 0 sampai 1. Pada system yang telah

dibuat pengguna memungkinkan mengatur *threshold score* deteksi, system akan melakukan visualisasi bounding box jika melebihi batas threshold yang ditentukan.



Gambar 31. Pengaturan Score Threshold 0

Pada gambar diatas dapat dilihat jika score threshold diatur menjadi 0 menandakan bahwa semua hasil deteksi akan ditampilkan. Proses tersebut menonaktifkan modul *Non-Maximum Suppression (NMS)*.



Gambar 32. Pengaturan Score Threshold 50

Pada gambar diatas dapat dilihat jika score threshold diatur menjadi 50 maka hasil deteksi yang memiliki nilai diatas 50% akan ditampilkan. Disebelah kanan aplikasi juga terdapat output dari system yaitu kelas, score dan koordinat bounding box.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penelitian terkait *Perancangan Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2* menunjukkan hasil yang positif. Dalam penelitian ini, model SSD MobileNet V2 berhasil melakukan transfer learning dengan data baru yaitu kerusakan PCB. Teknik transfer learning dalam pelatihan model mempermudah peneliti dalam menghasilkan model yang dapat menyelesaikan tugas yang spesifik. Dengan teknik ini, peneliti tidak perlu membangun neural network dari awal, melainkan neural network model yang dibuat akan meniru neural network pada model dasar yaitu SSD MobileNet V2. Selain itu, library Tensorflow sangat membantu dalam proses transfer learning, dimana peneliti hanya perlu mengatur parameter dengan tepat sehingga model dapat sesuai dengan harapan.

Pembuatan model AI tidak lepas dari hardware yang digunakan, mengingat proses training merupakan proses yang cukup kompleks dan membutuhkan komputasi yang tinggi. Salah satu hardware yang penting dalam proses training adalah GPU (*Graphical Processing Unit*). Dalam penelitian ini, digunakan GPU Nvidia Geforce RTX3050 sebagai alat komputasi selama training dan menghabiskan waktu sekitar 2 jam dengan nilai step sebanyak 20.000.

Selama proses training didapatkan nilai metrik yaitu **Classification Loss 0.02352, Localization Loss 0.006221, Regulation Loss 0.06596** dan **Total Loss 0.0957**. Hasil dari setiap metrik menunjukkan tren penurunan yang signifikan, yang menandakan bahwa model belajar dengan baik dan mengurangi kesalahan prediksi secara efektif. Selain itu, pada tahap uji validasi, inspeksi visual menunjukkan rata-rata score deteksi sebesar **76.63%**. Inspeksi kelistrikan dilakukan untuk memastikan kebenaran deteksi, dimana dilakukan 15 kali pengujian menggunakan multimeter. Hasil dari pengujian tersebut menunjukkan bahwa 12 dari 15 deteksi atau **80%** menggunakan SSD MobileNet V2 menunjukkan hasil yang sesuai dengan inspeksi menggunakan multimeter.

6.2 Saran

Perancangan *Sistem Pendeteksi Kerusakan pada PCB Menggunakan SSD MobileNet V2* ini masih memiliki banyak kekurangan. Seperti kamera mikroskopis memiliki tingkat resolusi dan pencahayaan yang perlu diperbaiki. Pada penelitian berikutnya disarankan, diperlukan upaya untuk meningkatkan resolusi dan kualitas kamera mikroskopis. Selain itu pada penelitian selanjutnya disarankan untuk meningkatkan jumlah klasifikasi kerusakan dan meningkatkan jumlah data sehingga penggunaan aplikasi dapat lebih luas lagi dan akurasi model dapat meningkat.

DAFTAR PUSTAKA

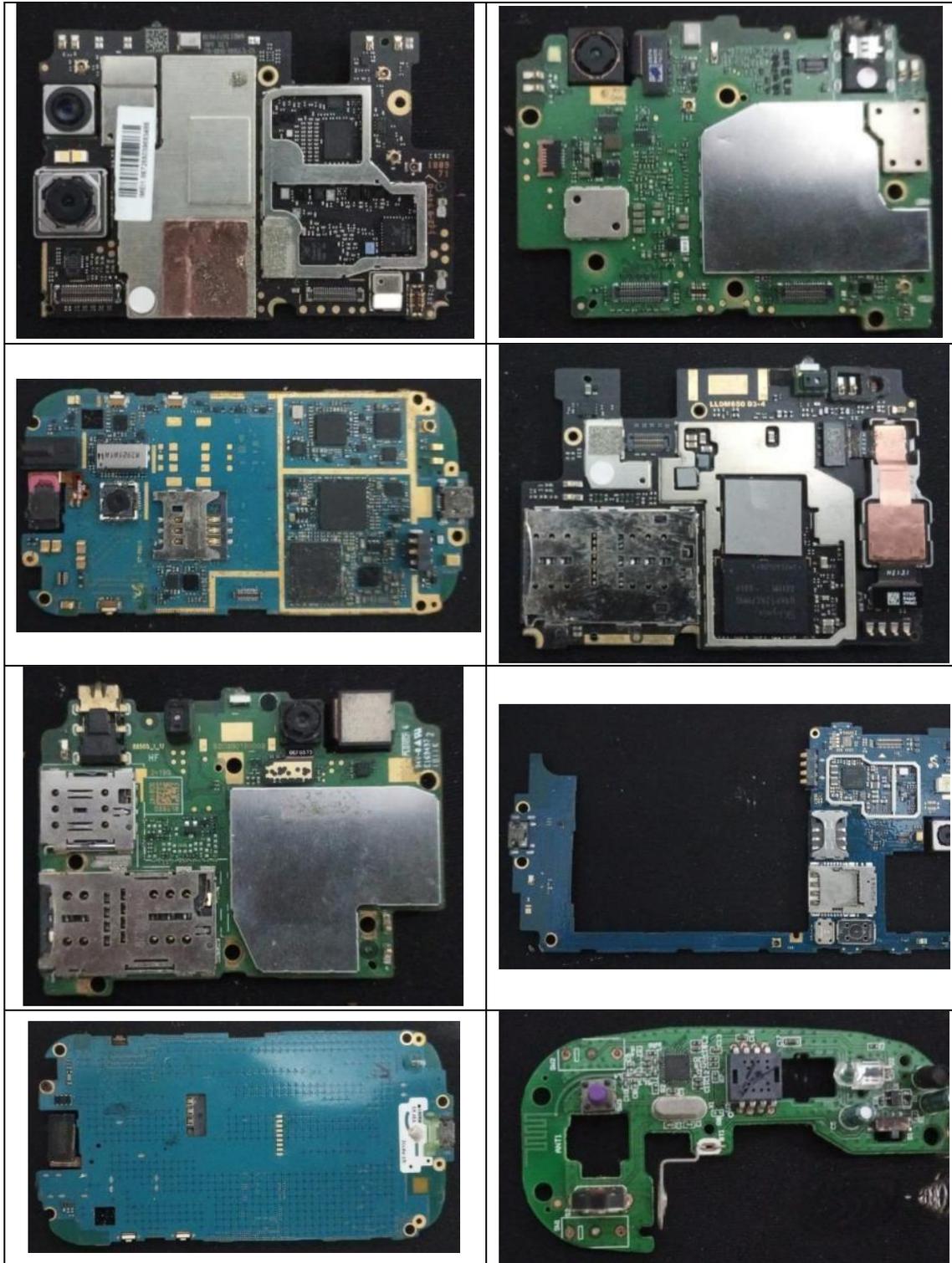
- Ardiansyah, Alief, dkk. 2019. "Inertial Navigation System Design Using Inertial Measurement Unit 10 DoF on Diver Propulsion Vehicle." IPTEK Journal of Proceedings Series. Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.
- Ariska, Melly Sakinah Alawiyah. 2019. "Mikroskop Digital Berbasis Kamera Smartphone." JIPFRI (Jurnal Inovasi Pendidikan Fisika dan Riset Ilmiah). FKIP Universitas Sriwijaya.
- Atina. 2019. "Aplikasi Matlab pada Teknologi Pencitraan Medis." JUPITER: Jurnal Penelitian Fisika dan Terapannya, 1(1). Universitas PGRI Palembang.
- Bustamin, Anugrayani dan A. Ais Prayogi. 2019. "Perbandingan Kinerja Filter Butterworth Berdasarkan Spesifikasi Frekuensi untuk Pengolahan Sinyal Suara." Techno.COM, Vol 18, No 4. Departemen Informatika, Fakultas Teknik, Universitas Hasanuddin, Makassar.
- Diastomo, Hanif, dkk. 2021. "Simulasi Gerak Sampah Laut Terapung Menggunakan Model Numerik di Selat Bali." Jurnal Teknik Lingkungan, Vol 27, No 1. Fakultas Ilmu dan Teknologi Kebumihan, Institut Teknologi Bandung.
- Fahmizal, dkk. 2022. "Implementasi Kinematika Trajectory Lingkaran pada Robot Roda Mecanum." Jurnal Listrik, Instrumentasi, dan Elektronika Terapan (JuLIET), Vol 3, No 1. Departemen Teknik Elektro dan Informatika, Universitas Gadjah Mada.
- Farouq, Ardiansyah Al, dkk. 2020. "Analisa Kesalahan Trajektori dan Algoritma Solusi Invers pada Robot Lengan Berdasarkan Jaringan QACA-RBF." COMPLETE Journal of Computer, Electronic, and Telecommunication. Fakultas Teknik Elektro, Institut Teknologi Telkom Surabaya.
- Hosna, Asmaul, dkk. 2022. "Transfer learning: a friendly introduction." Journal of Big Data. Department of Computer Science, Asian University for Women, Chittogram, Bangladesh.
- Mardiyah, Nur Alif, dkk. 2019. "Deteksi Dan Prediksi Trajektori Objek Bergerak Dengan Omni-Vision Menggunakan Particle Swarm Optimization-Neural Network (PSO-NN) Dan Interpolasi Polynomial." Multitek Indonesia: Jurnal Ilmiah, Vol 13, No 1. Teknik Elektro, Universitas Muhammadiyah Malang.
- Mudarris, Satria Gunawan Zain. 2020. "Implementasi Sensor Inertial Measurement Unit (IMU) untuk Monitoring Perilaku Roket." AVITEC, Vol 2, No 1. Universitas Negeri Makassar, Makassar.
- Nufus, N, dkk. 2021. "Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi." Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia. Akademi Angkatan Udara, Yogyakarta.

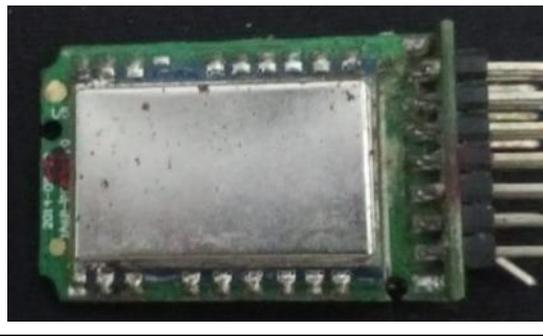
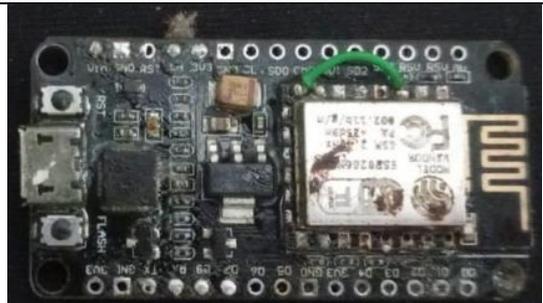
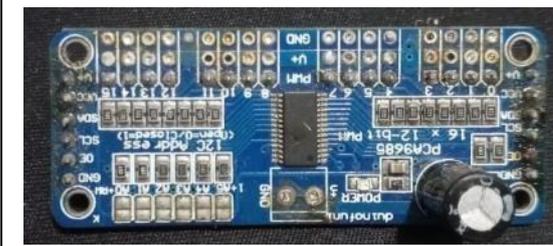
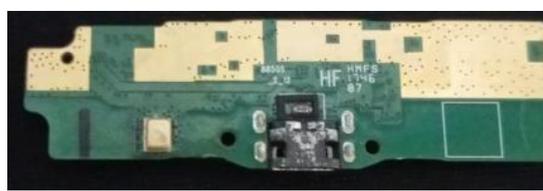
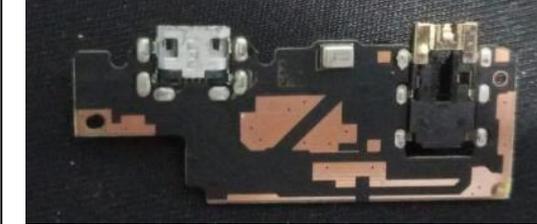
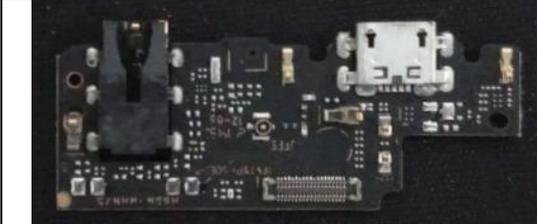
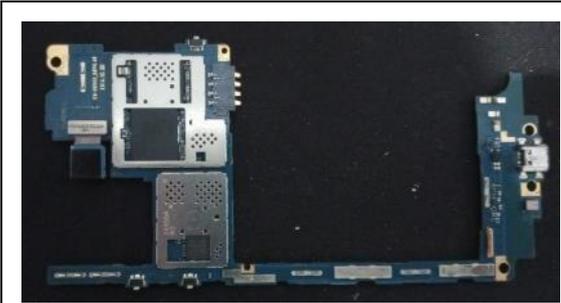
- Nursyeha, Muhammad Agung, dkk. 2023. "Implementasi Sensor Inertial Measurement Unit untuk Sistem Odometri Kendaraan Otonom." *SPECTA Journal of Technology*, Vol 7, No 2. Institut Teknologi Kalimantan, Balikpapan, Indonesia.
- Phadtare, Mohit, dkk. 2021. "Comparison between YOLO and SSD Mobile Net for Object Detection in a Surveillance Drone." *International Journal Of Scientific Research In Engineering And Management (Ijsrem)*. School of Mechanical Engineering, Dr. Vishwanath Karad World Peace University, Pune, India.
- Prastio, Rizki Putra. 2023. "Selection of autofocus algorithms for printed circuit board automated optical inspection system." *Indonesian Journal of Electrical Engineering and Computer Science*. Universitas Airlangga, Surabaya, Indonesia.
- Rahmadany, Fadila, dkk. 2019. "Implementasi Dan Analisis Kalman Filter Pada Sensor Inertial Measurement Unit (IMU)." *Tel-U Collection*. Fakultas Teknik Elektro, Universitas Telkom.
- Rembet, Michael, dkk. 2019. "Simulasi Respon Frekuensi Filter Butterworth Dan Chebyshev Tipe II Akibat Eksitasi Sinyal Acak." *Jurnal Tekno Mesin*, Vol 5, No 2. Jurusan Teknik Mesin, Universitas Sam Ratulangi, Manado.
- Setyawan, Novendra, dkk. 2022. "Navigasi Robot Sepak Bola Beroda Menggunakan Particle Filter Localization." *CYCLOTRON: Jurnal Teknik Elektro*, Vol 5, No 1. Fakultas Teknik, Universitas Muhammadiyah Malang.
- Sriyanto, Sesar Prabu Dwi. 2018. "Pengaruh Penggunaan Filter Butterworth Pada Penentuan Waktu Tiba Gelombang P." *Jurnal Sains dan Teknologi*, Vol 7, No 2. Stasiun Geofisika Winangun, Manado, Sulawesi Utara, Indonesia.
- Yadav, Darshan, dkk. 2023. "Real-Time Object Detection Using SSD Mobile Net Model of Machine Learning." *International Journal of Engineering and Computer Science (ijecs)*. Raj Kumar Goel Institute of Technology, Ghaziabad, UP, India
- Atina. 2019. *Aplikasi Matlab pada Teknologi Pencitraan Medis JUPITER: Jurnal Penelitian Fisika dan Terapannya*, 1 (1). Universitas PGRI Palembang
- Ardiansyah, Alief, dkk. 2019. *Inertial Navigation System Design Using Inertial Measurement Unit 10 DoF on Diver Propulsion Vehicle*. *IPTEK Journal of Proceedings Series*. Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.
- Ariska, Melly. Sakinah Alawiyah. 2019. *Mikroskop Digital Berbasis Kamera Smartphone*. *JIPFRI (Jurnal Inovasi Pendidikan Fisika dan Riset Ilmiah)*. FKIP Universitas Sriwijaya
- Hosna, Asmaul, dkk. 2022. *Transfer learning: a friendly introduction*. *Journal of Big Data*. Department of Computer Science, Asian University for Women. Chittogram, Bangladesh.

- Nufus, N, dkk. 2021. Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi. Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia. Akademi Angkatan Udara. Yogyakarta.
- Prastio, Rizki Putra, Rizki Putra. 2023. Selection of autofocus algorithms for printed circuit board automated optical inspection system. Indonesian Journal of Electrical Engineering and Computer Science. Universitas Airlangga. Surabaya. Indonesia

LAMPIRAN

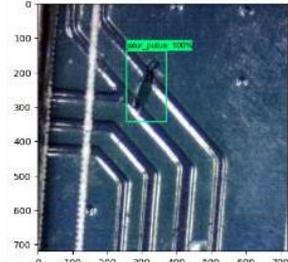
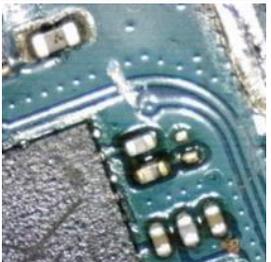
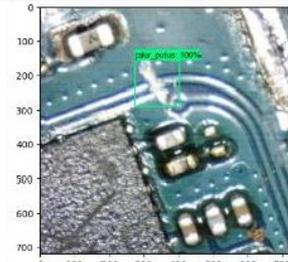
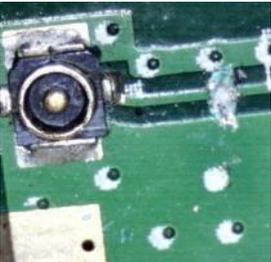
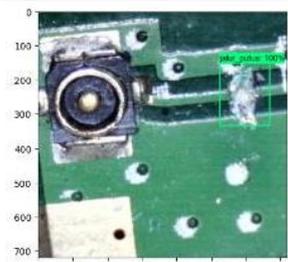
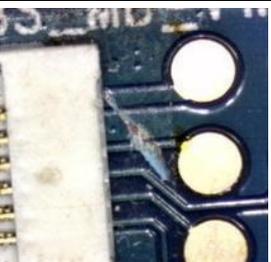
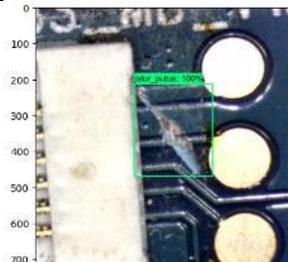
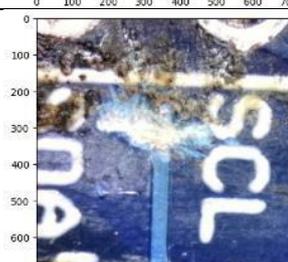
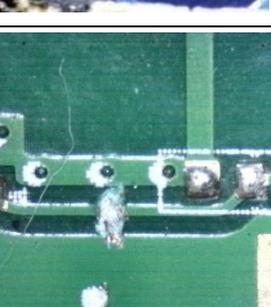
Lampiran 1 Sample PCB Rusak

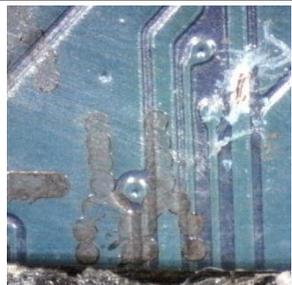
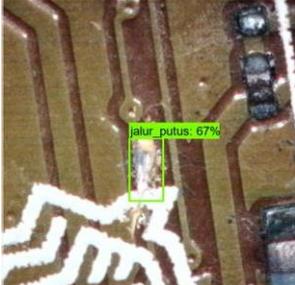
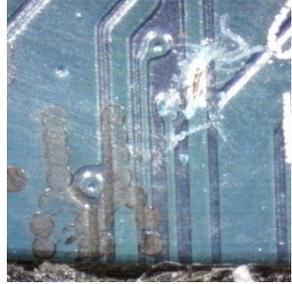
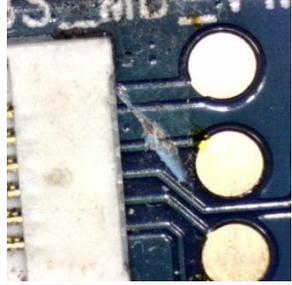
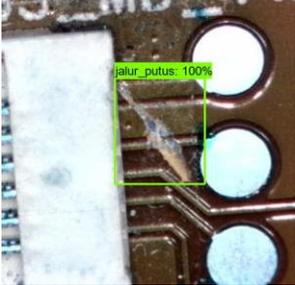


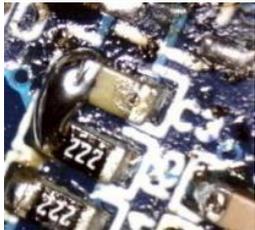
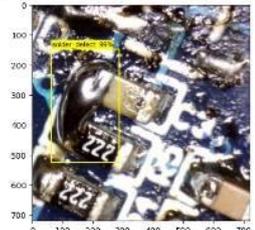


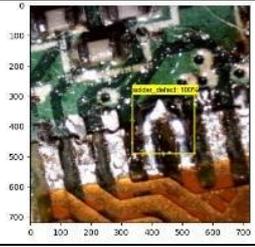
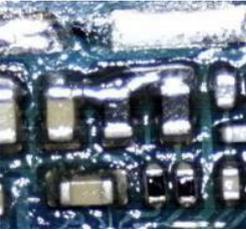
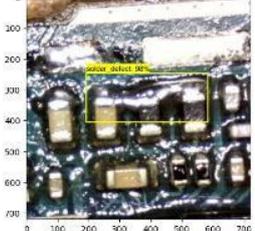
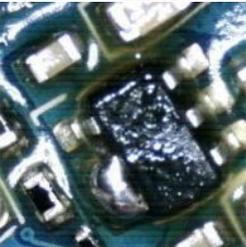
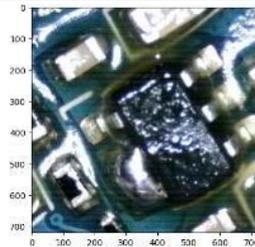
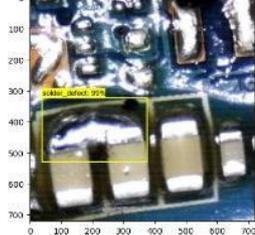
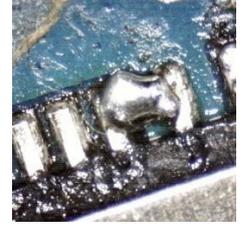
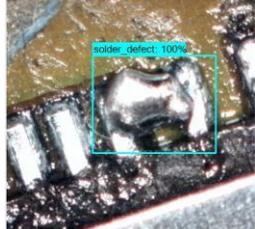
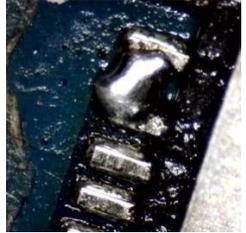
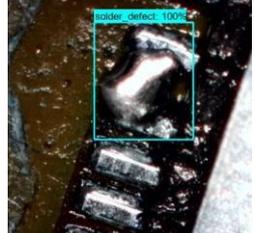


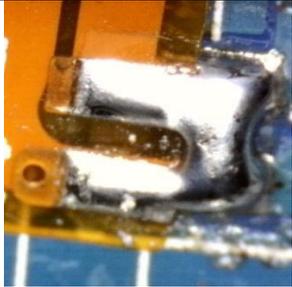
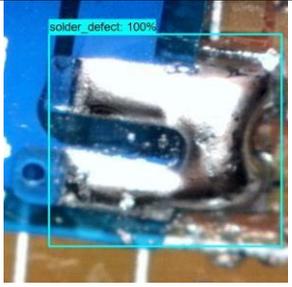
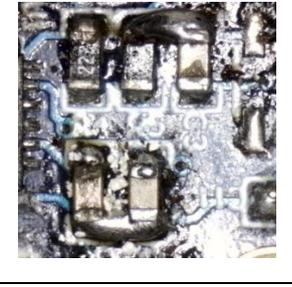
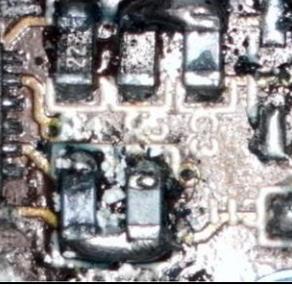
Lampiran 2 Inspeksi Visual

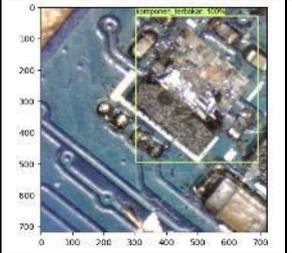
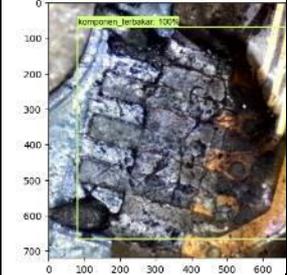
No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
1		Jalur terputus karena cacat produksi (PCB MPU6050)		Terdeteksi
2		Terdapat goresan pada jalur untuk IC Mediatek (PCB Handphone)		Terdeteksi
3		Terdapat jalur terputus di dekat konektor antenna (PCB Handphone)		Terdeteksi
4		Terdapat goresan pada jalur untuk konektor tombol (PCB Handphone)		Terdeteksi
5		Terdapat cacat pada jalur SCL (PCB MPU6050)		Tidak Terdeteksi
6		Terdapat cacat pada jalur		Terdeteksi

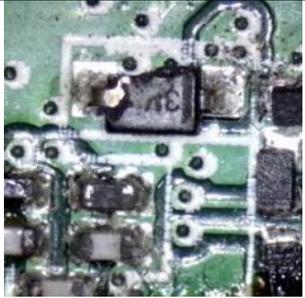
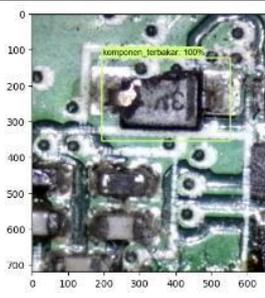
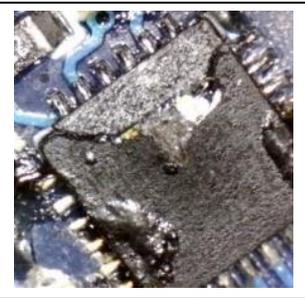
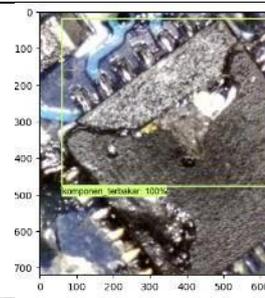
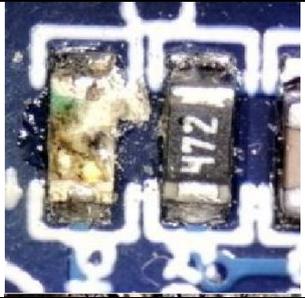
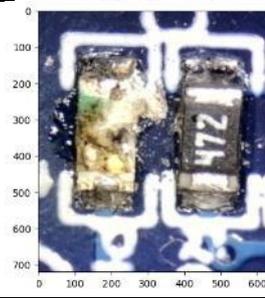
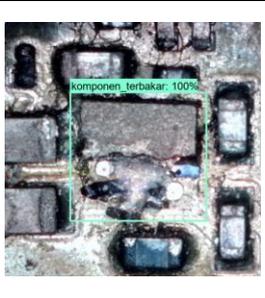
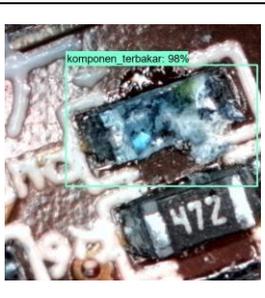
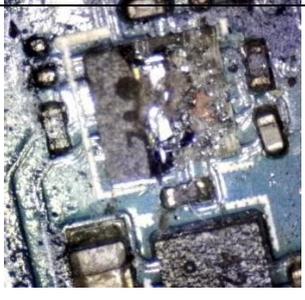
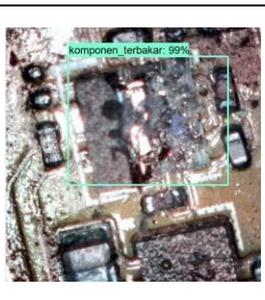
7		Terdapat cacat pada jalur		Terdeteksi
8		Terdapat cacat pada jalur		Terdeteksi
9		Terdapat cacat pada jalur		Tidak Terdeteksi
10		Terdapat cacat pada jalur konektor		Terdeteksi

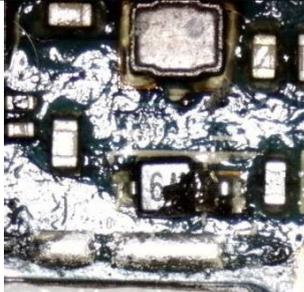
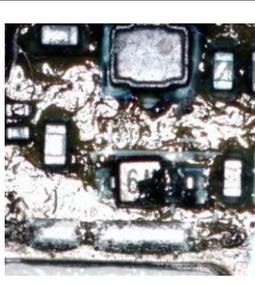
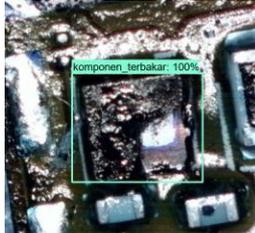
No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
1		Kapasitor dan resistor terdapat timah yang menggumpal		Terdeteksi

2		Flexible konektor terdapat timah yang menggumpal		Terdeteksi
3		Terdapat tiga buah komponen yang menyatu yaitu satu buah kapasitor dan dua buah resistor		Terdeteksi
4		Terdapat cacat solder pada komponen <i>buck converter</i> 3.3volt		Tidak Terdeteksi
5		Terdapat cacat solder pada komponen kapasitor		Terdeteksi
6		Terdapat cacat solder pada konektor		Terdeteksi
7		Terdapat cacat solder pada konektor		Terdeteksi

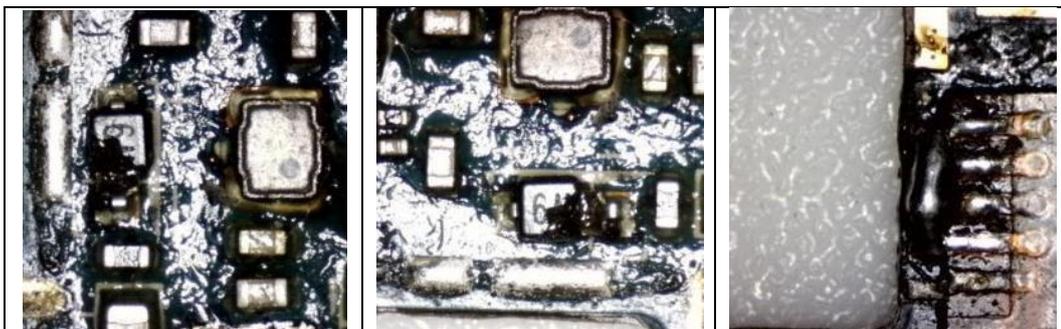
8		Terdapat cacat solder pada konektor LED		Terdeteksi
9		Terdapat cacat solder pada IC Power		Terdeteksi
10		Terdapat cacat solder pada resistor dan kapasitor		Tidak Terdeteksi

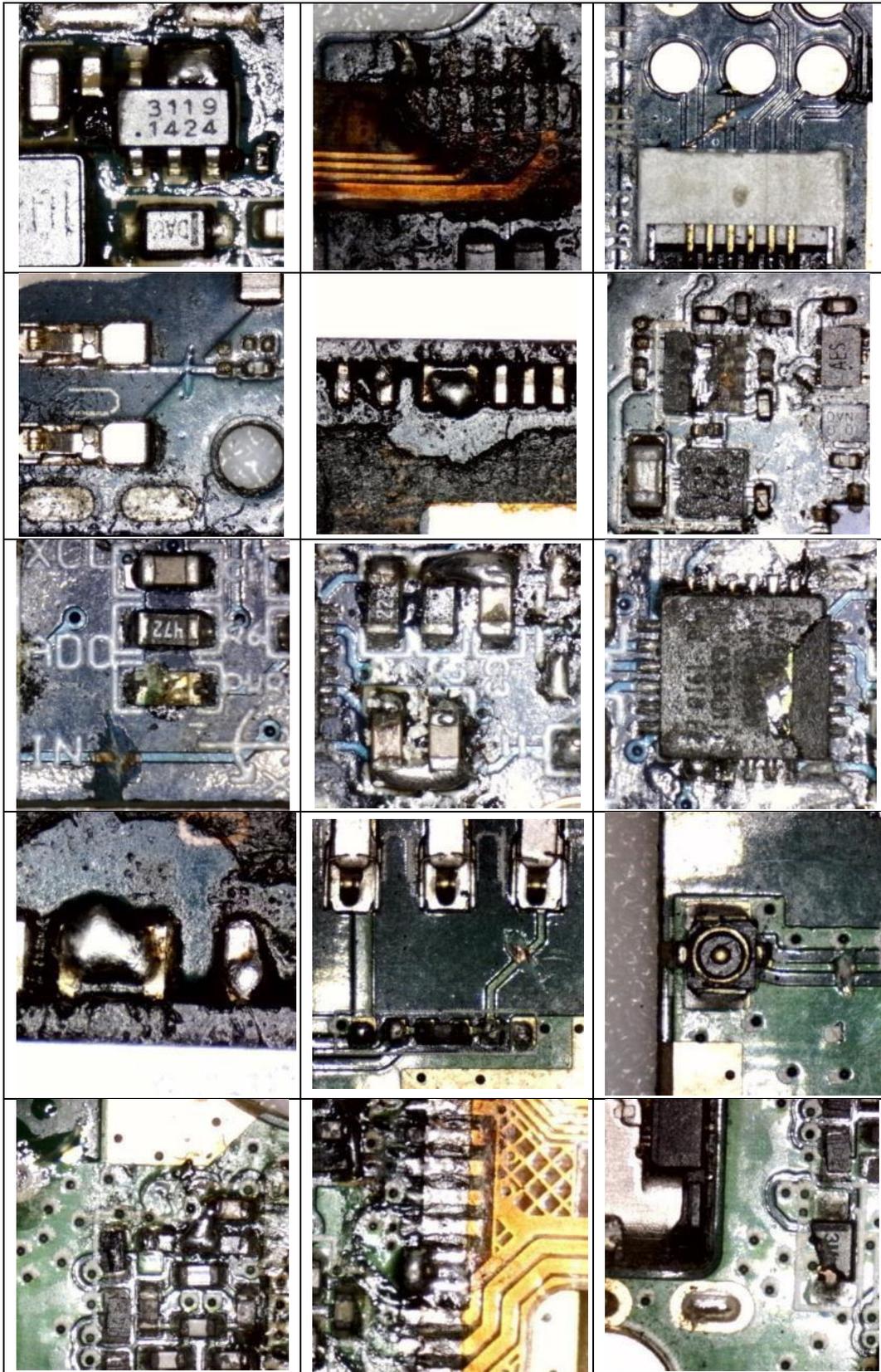
No	Inspeksi Visual	Kerusakan	Hasil Deteksi	Ket
1		Komponen terbakar IC power PM8937 (PCB Handphone)		Terdeteksi
2		Flexible konektor terbakar		Terdeteksi

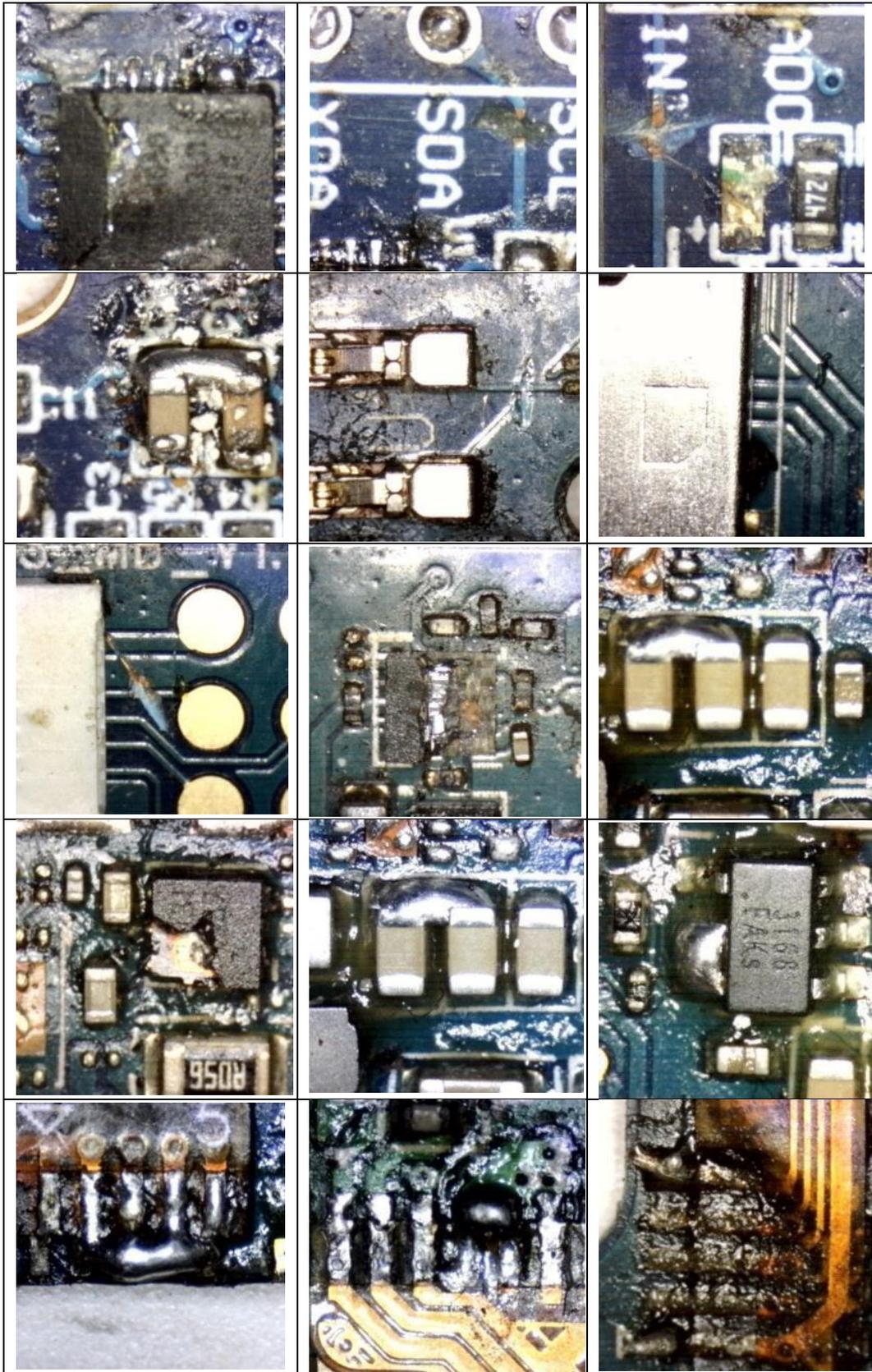
3		<p>Terdapat komponen dioda 3M terbakar</p>		<p>Terdeteksi</p>
4		<p>IC MPU 6050 terbakar</p>		<p>Terdeteksi</p>
5		<p>LED SMD terbakar</p>		<p>Tidak Terdeteksi</p>
6		<p>IC Power Terbakar</p>		<p>Terdeteksi</p>
7		<p>LED SMD terbakar</p>		<p>Terdeteksi</p>
8		<p>IC Terbakar</p>		<p>Terdeteksi</p>

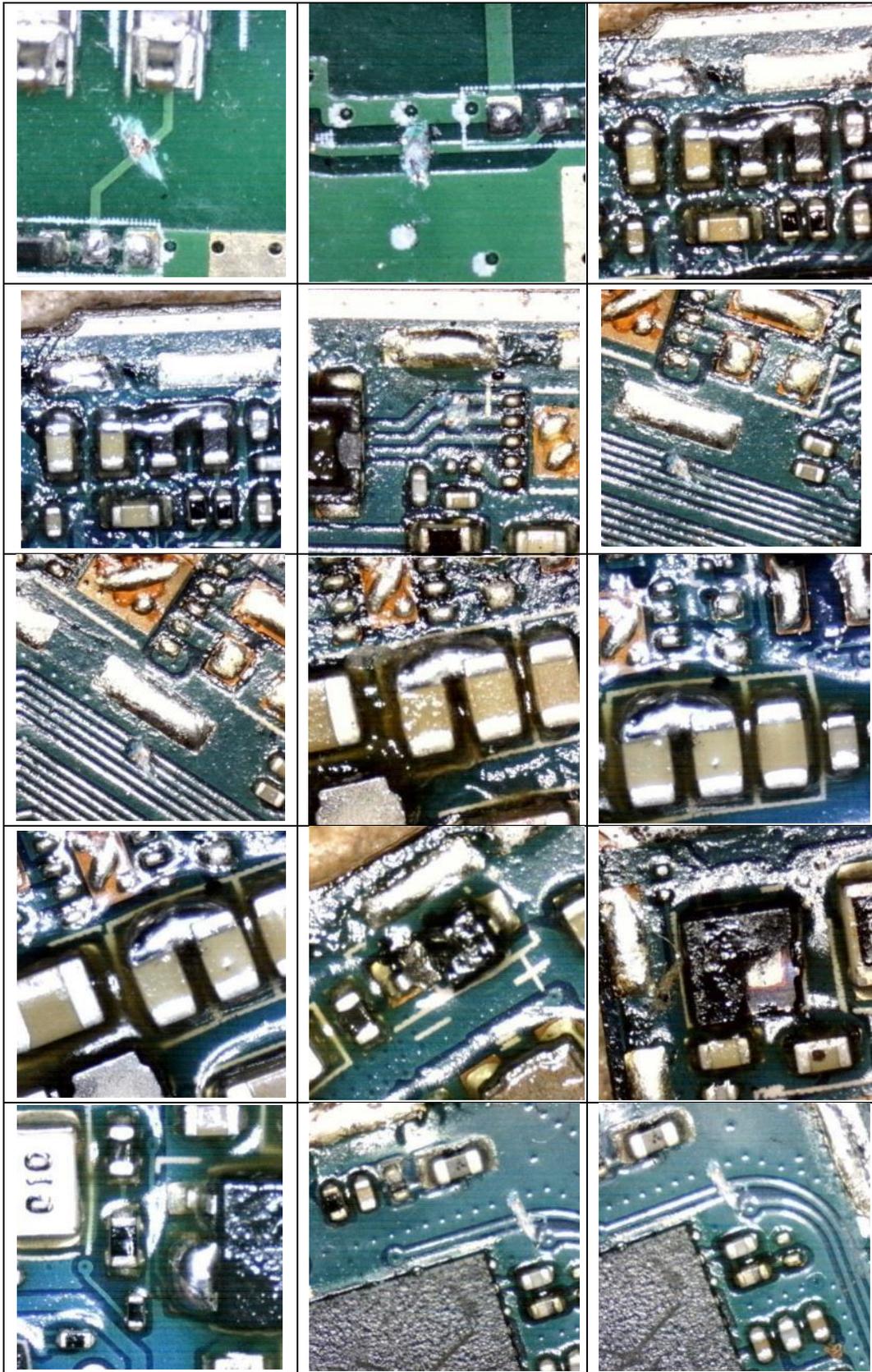
9		<p>Dioda terbakar</p>		<p>Tidak Terdeteksi</p>
10		<p>IC Regulator</p>		<p>Terdeteksi</p>

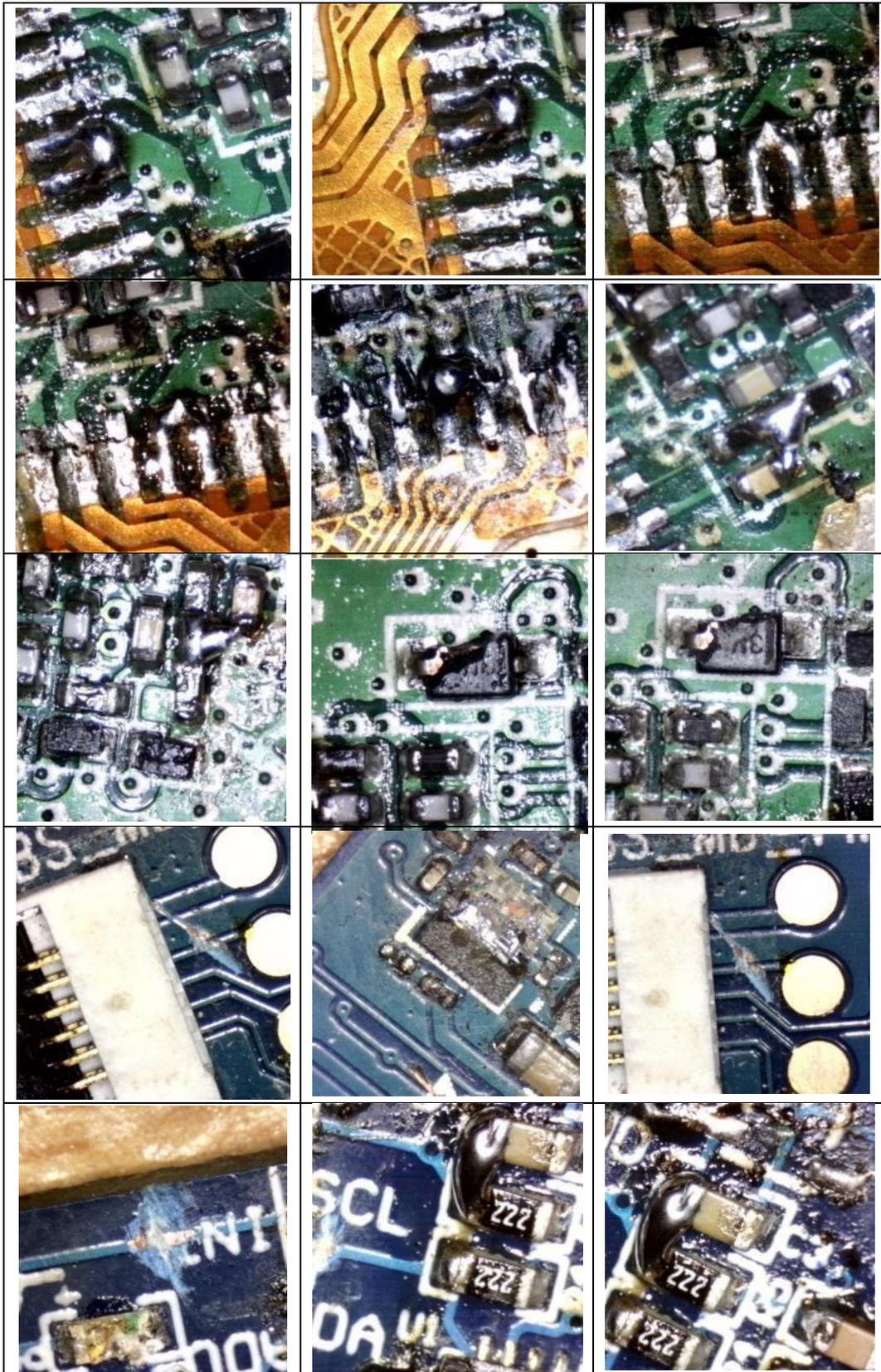
Lampiran 3 Data Train





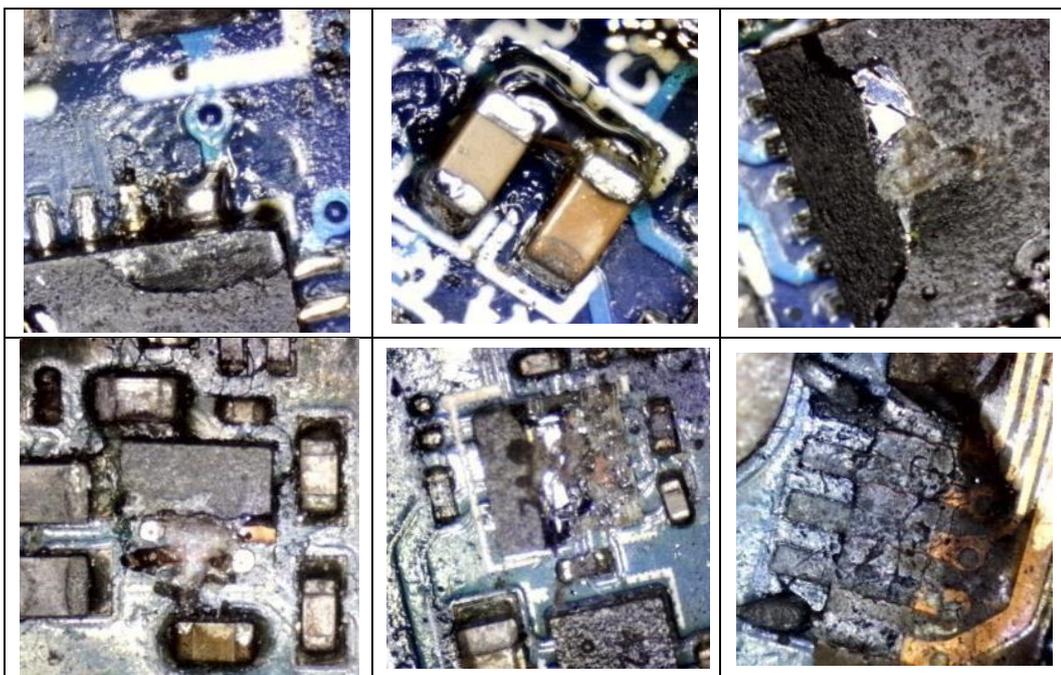


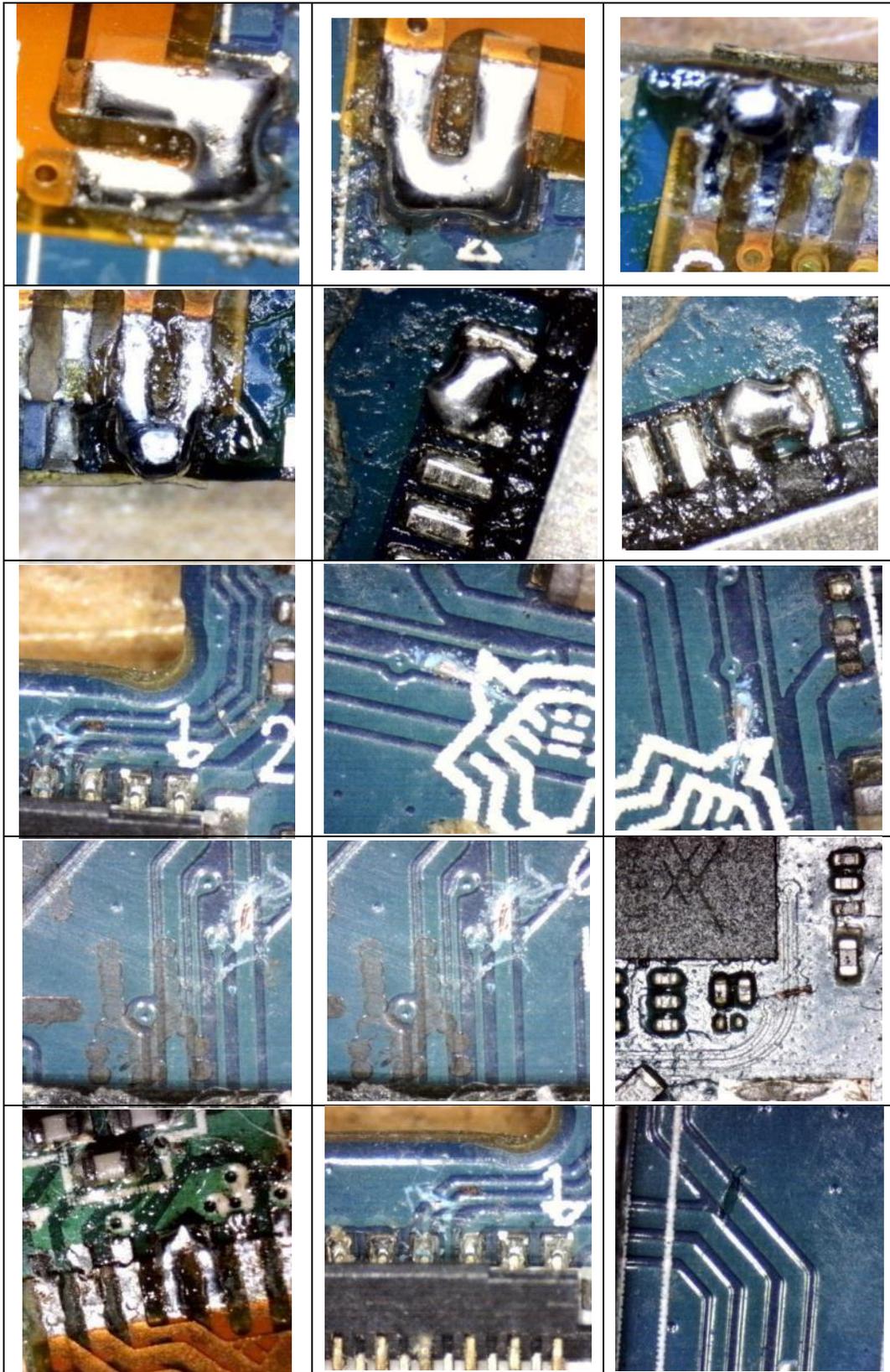






Lampiran 4 Data Test





Lampiran 5 File Konfigurasi Training

```
model {
  ssd {
    num_classes: 3
    image_resizer {
      fixed_shape_resizer {
        height: 320
        width: 320
      }
    }
    feature_extractor {
      type: "ssd_mobilenet_v2_fpn_keras"
      depth_multiplier: 1.0
      min_depth: 16
      conv_hyperparams {
        regularizer {
          l2_regularizer {
            weight: 4e-05
          }
        }
        initializer {
          random_normal_initializer {
            mean: 0.0
            stddev: 0.01
          }
        }
      }
      activation: RELU_6
      batch_norm {
        decay: 0.997
        scale: true
        epsilon: 0.001
      }
    }
    use_depthwise: true
    override_base_feature_extractor_hyperparams: true
    fpn {
      min_level: 3
      max_level: 7
      additional_layer_depth: 128
    }
  }
  box_coder {
    faster_rcnn_box_coder {
      y_scale: 10.0
      x_scale: 10.0
      height_scale: 5.0
      width_scale: 5.0
    }
  }
  matcher {
    argmax_matcher {
      matched_threshold: 0.5
      unmatched_threshold: 0.5
      ignore_thresholds: false
      negatives_lower_than_unmatched: true
      force_match_for_each_row: true
    }
  }
}
```

```

        use_matmul_gather: true
    }
}
similarity_calculator {
  iou_similarity {
  }
}
box_predictor {
  weight_shared_convolutional_box_predictor {
    conv_hyperparams {
      regularizer {
        l2_regularizer {
          weight: 4e-05
        }
      }
      initializer {
        random_normal_initializer {
          mean: 0.0
          stddev: 0.01
        }
      }
      activation: RELU_6
      batch_norm {
        decay: 0.997
        scale: true
        epsilon: 0.001
      }
    }
    depth: 128
    num_layers_before_predictor: 4
    kernel_size: 3
    class_prediction_bias_init: -4.6
    share_prediction_tower: true
    use_depthwise: true
  }
}
anchor_generator {
  multiscale_anchor_generator {
    min_level: 3
    max_level: 7
    anchor_scale: 4.0
    aspect_ratios: 1.0
    aspect_ratios: 2.0
    aspect_ratios: 0.5
    scales_per_octave: 2
  }
}
post_processing {
  batch_non_max_suppression {
    score_threshold: 1e-08
    iou_threshold: 0.6
    max_detections_per_class: 100
    max_total_detections: 100
    use_static_shapes: false
  }
  score_converter: SIGMOID
}
}

```

```

normalize_loss_by_num_matches: true
loss {
  localization_loss {
    weighted_smooth_l1 {
    }
  }
  classification_loss {
    weighted_sigmoid_focal {
      gamma: 2.0
      alpha: 0.25
    }
  }
  classification_weight: 1.0
  localization_weight: 1.0
}
encode_background_as_zeros: true
normalize_loc_loss_by_codesize: true
inplace_batchnorm_update: true
freeze_batchnorm: false
}
}
train_config {
  batch_size: 8
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    random_crop_image {
      min_object_covered: 0.0
      min_aspect_ratio: 0.75
      max_aspect_ratio: 3.0
      min_area: 0.75
      max_area: 1.0
      overlap_thresh: 0.0
    }
  }
}
sync_replicas: true
optimizer {
  momentum_optimizer {
    learning_rate {
      cosine_decay_learning_rate {
        learning_rate_base: 0.08
        total_steps: 50000
        warmup_learning_rate: 0.026666
        warmup_steps: 1000
      }
    }
    momentum_optimizer_value: 0.9
  }
  use_moving_average: false
}
fine_tune_checkpoint: "Tensorflow\\workspace\\pre-trained-models\\ssd_
mobilenet_v2_fpnlite_320x320_coco17_tpu-8\\checkpoint\\ckpt-0"
num_steps: 50000
startup_delay_steps: 0.0
replicas_to_aggregate: 8

```

```
max_number_of_boxes: 100
unpad_groundtruth_tensors: false
fine_tune_checkpoint_type: "detection"
fine_tune_checkpoint_version: V2
}
train_input_reader {
  label_map_path: "Tensorflow\\workspace\\annotations\\label_map.pbtxt"
  tf_record_input_reader {
    input_path: "Tensorflow\\workspace\\annotations\\train.record"
  }
}
eval_config {
  metrics_set: "coco_detection_metrics"
  use_moving_averages: false
}
eval_input_reader {
  label_map_path: "Tensorflow\\workspace\\annotations\\label_map.pbtxt"
  shuffle: false
  num_epochs: 1
  tf_record_input_reader {
    input_path: "Tensorflow\\workspace\\annotations\\test.record"
  }
}
}
```