



TESIS - TM185400

**DETEKSI CACAT PADA PERMUKAAN HASIL
PENGELASAN *FRICTION STIR WELDING* (FSW)
BERBASIS IMAGE PROCESSING DENGAN JARINGAN
FULLY CONVOLUTION NETWORK (FCN) UNTUK
MATERIAL AA6061-T651**

**ABRI ANDRY SARESA MARBUN
6007201016**

Dosen Pembimbing
Arif Wahjudi, S.T., M.T., Ph.D.

Departemen Teknik Mesin
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
2022

Halaman ini sengaja dikosongkan



TESIS - TM185400

**DETECTION OF DEFECTS ON THE SURFACE
RESULTING FROM FRICTION STIR WELDING (FSW)
BASED ON IMAGE PROCESSING WITH THE FULLY
CONVOLUTION NETWORK (FCN) METHOD FOR
AA6061-T651 MATERIAL**

**ABRI ANDRY SARESA MARBUN
6007201016**

**Dosen Pembimbing
Arif Wahjudi, S.T., M.T., Ph.D.**

**Departemen Teknik Mesin
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
2022**

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

ABRI ANDRY SARESA MARBUN

NRP: 6007201016

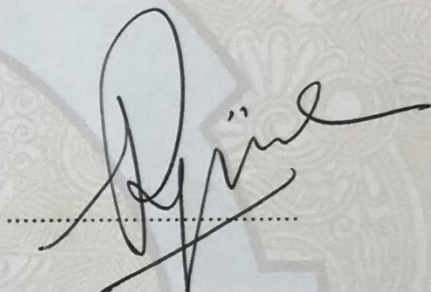
Tanggal Ujian: 11 Agustus 2022

Periode Wisuda: Maret 2023

Disetujui oleh:

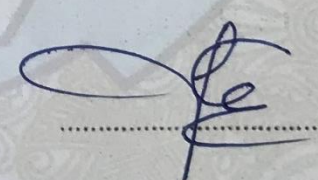
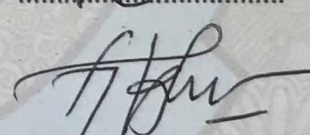
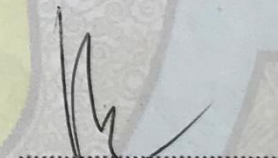
Pembimbing:

1. Arif Wahjudi, S.T., M.T., Ph.D
NIP: 197303222001121001



Penguji:

1. Prof. Dr. Ing. Ir. I Made Londen Batan, M.Eng
NIP: 195811061986011001
2. Dr. Ir. Agus Sigit Pramono, DEA
NIP: 196508101991021001
3. M. Khoirul Effendi, S.T., M.Sc. Eng., Ph.D
NIP: 198204142010121001



Kepala Departemen Teknik Mesin
Fakultas Teknologi Industri dan Rekayasa Sistem



Dr. Ir. Atok Setiawan, M.Eng.Sc.
NIP: 196604021989031002

Halaman ini sengaja dikosongkan

**DETEKSI CACAT PADA PERMUKAAN HASIL PENGELASAN
FRICITION STIR WELDING (FSW) BERBASIS *IMAGE PROCESSING*
DENGAN JARINGAN *FULLY CONVOLUTION NETWORK (FCN)* UNTUK
MATERIAL AA6061-T651**

Nama Mahasiswa : Abri Andry Saresa Marbun
NRP : 6007201016
Pembimbing : Arif Wahjudi, S.T., M.T., Ph.D.

ABSTRAK

Monitoring hasil pengelasan perlu dilakukan agar proses pengelasan yang dilakukan telah sesuai dengan ketentuan dan standart yang digunakan selain itu untuk memberikan informasi kepada operator apabila terjadi hambatan dan penyimpangan, serta memberi masukan dalam melakukan evaluasi. Pengujian hasil pengelasan dapat dilakukan dengan metode *non desctructive test* (NDT) seperti *visual inspection* yaitu pemeriksaan hasil pengelasan dengan cara pengamatan langsung pada permukaan hasil pengelasan melalui visual manusia. Dengan pengamatan langsung dapat diketahui cacat yang terjadi pada permukaan logam hasil pengelasan. Selain pengamatan langsung inspeksi juga dapat dilakukan dengan mesin berbasis komputer AVIS (*automated visual inspection systems*).

Penelitian ini mengembangkan metode pengolahan citra untuk memonitoring hasil pengelasan *Friction Stir Welding (FSW)* pada material AA6061-T651 melalui *visual inspection* yang dilakukan yaitu mendeteksi cacat *flash* pada permukaan hasil pengelasan FSW. Deteksi dilakukan dengan menggunakan metode *deep learning* yaitu *Fully Convolutional Network (FCN)*. Jaringan FCN yang dirancang menggunakan 5 *convolutional layer*, 5 *ReLU layer*, 5 *batchnormalization layer*, 5 *maxpool layer* dan 1 *fully connected layer*. Sebagai citra pembelajaran, citra validasi dan pengujian digunakan citra hasil pengelasan yang telah dikategorikan menjadi kelas normal dan *defect* dengan jumlah seluruh citra sebanyak 566 citra. Dari hasil pembelajaran dilakukan pengujian jaringan

dengan menggunakan citra yang ada untuk dapat mengetahui performa dari jaringan yang telah dirancang.

Berdasarkan hasil pengujian *visual inspection* oleh jaringan FCN, jaringan mampu mengklasifikasikan citra hasil pengelasan yang digunakan sebagai *testing* menjadi citra hasil pengelasan dengan kelompok baik dan kelompok cacat (*defect*). Kinerja jaringan FCN yang dirancang mampu mencapai akurasi deteksi cacat *flash* sebesar 97.6% dan menghasilkan performa jaringan yaitu *precision* 96,8% dan *sensitivity* 96,8%. Dari hasil-hasil tersebut diketahui keberhasilan jaringan FCN dalam melakukan *visual inspection* sangat baik dan dapat digunakan dalam pengamatan hasil pengelasan FSW. Dengan jaringan FCN yang dirancang dapat digunakan dalam penelitian selanjutnya untuk memonitoring proses pengelasan FSW, sehingga memungkinkan untuk mendeteksi cacat *flash* pada proses pengelasan FSW sedini mungkin.

Kata kunci: FSW, *deep learning*, *image processing*, FCN dan akurasi.

**DETECTION OF DEFECTS ON THE SURFACE RESULTING FROM
FRICTION STIR WELDING (FSW) BASED ON IMAGE PROCESSING
WITH THE FULLY CONVOLUTION NETWORK (FCN) METHOD FOR
AA6061-T651 MATERIAL**

Nama Mahasiswa : Abri Andry Saresa Marbun
NRP : 6007201016
Pembimbing : Arif Wahjudi, S.T., M.T., Ph.D.

ABSTRACT

Monitoring welding results needs to be carried out so that the welding process carried out is in accordance with the provisions and standards used in addition to providing information to the operator in the event of obstacles and deviations, as well as providing input in conducting evaluations. Testing welding results can be done with non-destructive test (NDT) methods such as visual inspection, namely the examination of welding results by direct observation on the surface of welding results through human visuals. By direct observation, it can be known the defects that occur on the surface of the metal from welding. In addition to direct observation, inspections can also be carried out with computer-based machines AVIS (automated visual inspection systems).

This study developed an image processing method to monitor the results of Friction Stir Welding (FSW) welding on material AA6061-T651 through visual inspection, which was carried out, namely detecting flash defects on the surface of the FSW welding results. Detection is carried out using a deep learning method, namely the Fully Convolutional Network (FCN). The FCN network is designed using 5 convolutional layers, 5 ReLU layers, 5 batchnormalization layers, 5 maxpool layers and 1 fully connected layer. As a learning image, validation and testing images are used welding images that have been categorized into normal and defect classes with a total of 566 images. From the learning results, network testing is carried out using existing images to be able to determine the performance of the network that has been designed.

Based on the results of visual inspection testing by the FCN network, the network is able to classify the welding result image used as testing into an image of welding results with good groups and defect groups. The performance of the designed FCN network is able to achieve flash flaw detection accuracy of 97.6% and produce network performance, namely 96.8% precision and 96.8% sensitivity. From these results, it is known that the success of the FCN network in conducting visual inspection is very good and can be used in observing FSW welding results. With the designed FCN network can be used in subsequent research to monitor the FSW welding process, making it possible to detect flash defects in the FSW welding process as early as possible.

Keywords: *FSW, deep learning, image processing, FCN and accuracy.*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada kepada Tuhan dan Yesus Kristus atas segala kasih dan karunia-Nya, sehingga tesis berjudul “**DETEKSI CACAT PADA PERMUKAAN HASIL PENGELASAN *FRICTION STIR WELDING* (FSW) BERBASIS IMAGE PROCESSING DENGAN JARINGAN *FULLY CONVOLUTION NETWORK* (FCN) UNTUK MATERIAL AA6061-T651**” ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Pengerjaan tesis ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan di Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Mesin. terselesaikannya buku tesis ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Arif Wahjudi, S.T., M.T., Ph.D. selaku dosen pembimbing penulis yang telah meluangkan waktu berharganya demi memberikan semangat, ide, bimbingan, dan evaluasi yang sangat besar dampaknya dalam penyelesaian Tesis ini.
2. Bapak Prof. Dr. Ing. Ir. I Made Londen Batan, M.Eng, bapak Dr. Ir. Agus Sigit Pramono, DEA, dan bapak M. Khoirul Effendi, S.T., M.Sc. Eng., Ph.D selaku dosen penguji yang telah memberikan saran, masukan, dan arahan untuk sempurnanya Tesis ini.
3. Bapak Mulyadi, S.T., M.T., dan ibu Eka Marliana, S.T., M.T., selaku mahasiswa doktoral yang telah memberikan waktunya untuk membimbing dan mengarahkan penulis dalam menyelesaikan Tesis ini.
4. Seluruh Dosen, Staf dan Karyawan Departemen Teknik Mesin FTIRS ITS
5. Teman-teman di Laboratorium Rekayasa Produksi dan P3.
6. Teman-teman pascasarjana Departemen Teknik Mesin FTIRS ITS Angkatan 2020.
7. Serta semua pihak yang telah banyak membantu dalam menyelesaikan Tesis ini yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari dalam penulisan tesis ini masih banyak kekurangan, oleh sebab itu penulis sangat mengharapkan kritik dan saran yang membangun untuk

lebih menyempurnakan penulisan tesis ini dan dapat menjadi lebih baik dimasa yang akan datang serta dapat digunakan dalam hal mendukung kemajuan pada dunia teknologi. Semoga bermanfaat bagi kita semua.

Surabaya, Agustus 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 Pengertian Pengelasan	7
2.2 Jenis-Jenis Pengelasan.....	7
2.3 Friction stir welding (FSW)	8
2.4 Prinsip Kerja FSW.....	9
2.5 Parameter FSW.....	12
2.6 Cacat Pengelasan FSW.....	14
2.7 Kecerdasan Buatan (AI)	16
2.8 Machine Learning.....	17
2.9 <i>Deep Learning</i>	18
2.10 Fully Convolutional Network	20
2.10.1 Convolutional Layer	22
2.10.2 Pooling Layer.....	23

2.10.3	Fully Connected Layer	24
2.10.4	SoftmaxLayer	24
2.10.5	BatchNormalization Layer	25
2.10.6	Classification layer	25
2.10.7	Stride dan Padding.....	26
2.11	Neural Network (NN).....	27
2.11.1	Fungsi Aktivasi.....	30
2.11.2	Evaluasi Kinerja	33
2.12	Penelitian Sebelumnya	34
BAB 3 METODOLOGI		39
3.1	Diagram Alir Penelitian	39
3.2	Parameter pengelasan FSW	41
3.3	Pembuatan dan Penyusunan <i>Dataset</i> Citra Permukaan Las FSW	41
3.4	Perancangan jaringan FCN	45
3.5	Pelatihan Algoritma FCN	46
3.6	Tahapan Image Processing	48
3.7	Implementasi.....	48
3.7.1	Perangkat lunak	48
3.7.2	Perangkat keras.....	49
3.7.3	Pra-pemrosesan data set.....	49
3.7.4	Arsitektur FCN	49
3.8	Evaluasi dan Analisis Hasil VT oleh CNN.....	53
3.9	Pengambilan Kesimpulan dan Saran	54
BAB 4 HASIL DAN PEMBAHASAN		55
4.1	Hasil Pengolahan dan Klasifikasi Citra	55
4.2	Hasil Pelatihan Model.....	56
4.4	Hasil Perhitungan Performa Jaringan FCN	58
4.3	Perhitungan Nilai Learning Rate	62
4.5	Hasil Pengujian Citra	63
4.6	Pembahasan	65
BAB 5 KESIMPULAN DAN SARAN		69

5.1	Kesimpulan.....	69
5.2	Saran.....	69
DAFTAR PUSTAKA		71
LAMPIRAN.....		75
	Lampiran 1 Citra Hasil Pengelasan.....	75
	Lampiran 1A Citra Hasil Pengelasan <i>defect</i>	76
	Lampiran 1B Citra Hasil Pengelasan Normal	77
	Lampiran 2 A Standart Kualitas Pengelasan FSW	78
	Lampiran 2 B Kriteria Inspeksi Visual berdasarkan AWS	79
	Lampiran 3 Listing Program	80
	Lampiran 3A Listing pengujian dengan data diluar training dan validasi	81
BIOGRAFI PENULIS		83

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Klasifikasi Pengelasan[8].....	8
Gambar 2. 2(a) skema pengelasan FSW (b) hasil las FSW (c) geometri <i>tool</i> [9]....	9
Gambar 2. 3 Prinsip FSW[9].....	10
Gambar 2. 4 Rangkaian Gerakan pada FSW[10].....	11
Gambar 2. 5 Contoh desain <i>Probe</i> pada Tool FSW[11]	14
Gambar 2. 6 Defect pada pengelasan FSW[13]	15
Gambar 2. 7 Hubungan <i>Artificial Intelligence, Deep Learning</i> dengan <i>Machine Learning</i> [14]	18
Gambar 2. 8 Arsitektur FCN VGG-16[14]	21
Gambar 2. 9 <i>Training</i> dan <i>Testing</i> proses pada FCN[15]	22
Gambar 2. 10 <i>Convolutional layer example</i> [16].....	23
Gambar 2. 11 Maxpooling Layer[15]	23
Gambar 2. 12 <i>Fully Connected Layer</i> [17]	24
Gambar 2. 13 Contoh Klasifikasi pada spam email[19].....	26
Gambar 2. 14 Syaraf Secara Biologi[20]	28
Gambar 2. 15 Struktur MLP[21].....	29
Gambar 2. 16 Fungsi Aktivasi Linear[22]	31
Gambar 2. 17 Fungsi Aktivasi <i>Sigmoid</i> [22].....	31
Gambar 2. 18 Fungsi Aktivasi <i>Tanh</i> [22].....	32
Gambar 2. 19 Fungsi aktivasi <i>ReLU</i> [22]	32
Gambar 2. 20 Jaringan CNN AlexNet[28].....	37
Gambar 2. 21 Jaringan VGG-16[28].....	37
Gambar 3. 1 Diagram alir Penelitian.....	40
Gambar 3. 2 Dimensi objek pengelasan.....	41
Gambar 3. 3 Pengambilan gambar hasil pengelasan FSW	42
Gambar 3. 4 Hasil pengelasan FSW	43
Gambar 3. 5 hasil resizing (a) normal (b) cacat atau defect.....	43
Gambar 3. 6 Alur proses training[21]	45
Gambar 3. 7 Arsitektur Jaringan FCN	46
Gambar 3. 8 Fungsi penyeragaman dimensi citra	49

Gambar 3. 9 Implementasi membuka dan membagi data set	49
Gambar 3. 10 Implementasi perancangan jaringan FCN	50
Gambar 3. 11 Implementasi pelatihan jaringan FCN	51
Gambar 3. 12 Implementasi pengujian jaringan FCN	52
Gambar 3. 13 Implementasi pengujian jaringan FCN untuk citra <i>testing</i>	52
Gambar 3. 14 Implementasi <i>load</i>	53
Gambar 4. 1 Hasil pembuatan <i>data set</i> (a)citra hasil pengelasan, (b) citra defect, (c) citra normal.....	55
Gambar 4. 2 Hasil training FCN.....	57
Gambar 4. 3 <i>Confusion Matrix</i>	59
Gambar 4. 4 Hasil Pengujian validasi Citra	63
Gambar 4. 5 Hasil Pengujian Citra <i>Defect</i>	64
Gambar 4. 6 Hasil Pengujian Citra Normal.....	64
Gambar 4. 7 Pelatihan FCN pada learning rate 0,1	65
Gambar 4. 8 Pelatihan FCN pada learning rate 0,01	66
Gambar 4. 9 Pelatihan FCN pada learning rate 0,001	67

DAFTAR TABEL

Tabel 2. 1 Perbedaan <i>machine learning</i> dan <i>deep learning</i>	19
Tabel 2. 2 <i>Confussion Matrix</i>	34
Tabel 2. 3 Perbandingan Penelitian.....	38
Tabel 3. 1 Komposisi Kimia Pelat A6061-T651.....	41
Tabel 3. 2 Parameter Pelatihan.....	46
Tabel 3. 3 <i>Confussion Matrix</i>	47
Tabel 4. 1 perbandingan <i>learning rate</i> yang digunakan	62

Halaman ini sengaja dikosongkan

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Friction stir welding (FSW) merupakan proses penyambungan logam tanpa *filler* dan tanpa meleleh. Proses penyambungan logam terjadi pada fasa padat karena berlangsung pada temperatur dibawah titik lebur dari material yang akan disambung. Proses pengelasan dilakukan dengan memasukkan *pin tool* ke material yang akan disambung sampai *shoulder* menyentuh permukaan atas material menggunakan panas yang disebabkan oleh gesekan antara *tool* dan benda kerja. Karena panas tersebut material di sekitar sambungan menjadi panas dan lunak, kemudian *tool* akan digerakkan berputar mengaduk bagian tersebut sehingga terbentuklah sambungan las. Untuk mendapatkan hasil pengelasan yang baik selain pemilihan parameter yang tepat seperti kecepatan putaran pahat (*rpm*), *feedrate*, kemiringan *pin* dan bentuk *pin* juga memberikan pengaruh terhadap hasil pengelasan metode *Friction stir welding* (FSW)[1]. Agar dapat memastikan hasil pengelasan yang dilakukan telah sesuai dengan yang diharapkan dan agar mengetahui karakteristik dari hasil pengelasan yang telah dilakukan serta mendapatkan data yang paling sesuai untuk parameter yang digunakan, maka inspeksi terhadap logam hasil pengelasan yang telah dikerjakan perlu dilakukan.

Metode inspeksi hasil pengelasan secara umum dilakukan menggunakan metode *Destructive Testing* (DT) dan *Non-Destructive Test* (NDT). *Destructive Testing* (DT) adalah suatu cara pengujian hasil pengelasan dengan cara merusak hasil pengelasan yang telah dikerjakan. Tujuannya adalah untuk mengetahui kekuatan las terhadap suatu jenis pembebanan. Pengujian DT antara lain adalah pengujian tarik, pengujian beban kejut (*impact*), pengujian kekerasan, pengujian macro (*structure test*)[2]. NDT adalah aktivitas tes atau inspeksi terhadap suatu benda untuk mengetahui adanya cacat, retak, *discontinuity*, dan lain tanpa merusak benda yang dilakukan pengetesan atau inspeksi. Pada dasarnya, tes ini dilakukan untuk menjamin bahwa material yang kita gunakan masih aman dan belum melewati *damage tolerance*. Metode NDT dibagi menjadi beberapa kategori yaitu *Visual Inspection* (VT), *Liquid Penetrant Testing* (LT), *Magnetic Particle Testing* (MT), *Radiographic Testing* (RT), *Ultrasonic Testing* (UT), *Eddy Current Testing*

(ET), *Acoustic Emission Testing* (AET), *Phased Array Ultrasonics* (PA), *Time of Flight Diffraction* (TOFD)[2]. Salah satu pengujian saat ini sedang dikembangkan yaitu *visual inspection* dengan menggunakan metode *computer vision* berbasis pengolahan citra (*image processing*).

Secara tradisional, memonitoring hasil pengelasan melalui inspeksi visual dan *quality control* dapat dilakukan oleh manusia secara manual. Pekerjaan ini lebih lambat jika dibandingkan dengan menggunakan mesin yang berbasis komputer. Dengan mesin berbasis komputer inspeksi visual dan *quality control* dapat dilakukan secara terus menerus dengan kualitas yang sama. Salah satu mesin berbasis komputer adalah AVIS (*automated visual inspection system*) dimana biaya pemeliharaan lebih murah dan akurasi tinggi. Walaupun demikian beberapa industri pengelasan masih belum menggunakannya hal tersebut dikarenakan inspeksi visual secara otomatis terlambat diperkenalkan. Faktor yang menghambat penggunaan sistem inspeksi otomatis adalah sistem yang tidak sederhana, kurangnya fleksibilitas dasar perancangan, kurangnya metoda kajian kinerja dan kurangnya studi kasus yang menghambat pengembangan inspeksi visual otomatis[3]. AVIS memanfaatkan teknik pengolahan citra dan kecerdasan buatan yang dapat menerjemahkan informasi yang diperoleh menjadi informasi lain yang berguna dalam pengambilan keputusan[4]. Perkembangan riset untuk deteksi *defect* pada permukaan hasil pengelasan berbasis pengolahan citra digital bertujuan membantu mengenali adanya *defect* pada permukaan hasil pengelasan lebih cepat.

Image processing khususnya berbasis *Artificial Neural Network* menjadi trend dalam beberapa tahun terakhir karena semakin berkembangnya ketertarikan yang tinggi pada bidang *deep learning*. *Image processing* menjadi sorotan karena kemampuannya dalam mengenali dan mengklasifikasikan suatu objek. Penelitian yang dilakukan beberapa tahun belakangan menunjukkan metode *deep learning* untuk *image processing* dalam mengklasifikasi maupun mendeteksi cacat pada hasil pengelasan FSW menunjukkan hasil akurasi yang tinggi seperti pada penelitian R.Ranjan, *et al.*(2016) dengan metode *image pyramid* and *image reconstruction algorithms* untuk mengidentifikasi cacat yang terjadi pada permukaan las FSW seperti *voids*, *grooves*, *crack*, *key-hole* dan *flash*. Penelitian ini mampu mengklasifikasikan cacat yang terjadi pada permukaan las dengan

membandingkan dimensi dari gumpalan yang telah didefinisikan. , S. Sudhagar *et al* (2019) menggunakan *maximally stable extremal region algorithm* (MSER) untuk identifikasi citra permukaan las dan metode *support vector machine* (SVM) untuk mengklasifikasikan citra pengelasan FSW. Pada penelitian ini program mampu mencapai akurasi hingga sebesar 95,8%. Dari penelitian tersebut dapat diketahui pengujian secara otomatis menggunakan metode image processing sangat baik digunakan. Namun dalam penelitian tersebut masih membutuhkan proses *feature extraction* yang terpisah dengan metode klasifikasinya. Pengembangan secara *deep learning* dibutuhkan untuk dapat menghemat daya yang dibutuhkan dan mempercepat proses komputasi dibandingkan dengan *machine learning*. Pada penelitian selanjutnya FCN adalah arsitektur yang biasanya digunakan untuk *semantic segmentation* menggunakan *locally connected layers*, seperti *convolution*, *pooling*, dan *upsampling*. FCN tidak berisi dense layer seperti pada CNN pada umumnya melainkan berisi konvolusi 1x1 yang menjalankan tugas *fully connected layers*. Menghindari penggunaan *dense layer* berarti lebih sedikit parameter dimana membuat jaringan lebih cepat untuk di-train. FCN dapat berfungsi untuk ukuran gambar bervariasi karena semua koneksi bersifat local[5]. Istilah *fully convolutional network* dikenalkan oleh Long *et al.* (2015) yang fokus pada segmentasi semantik. Pengertian segmentasi semantik adalah memahami citra dalam tingkat pixel dimana setiap *pixel* mempunyai label yang berkorespondensi dalam objek atau wilayah yang bersangkutan. Tujuan dari implementasi FCN pada *paper*-nya adalah membuat arsitektur yang mampu menerima input dengan ukuran yang bervariasi serta output yang mempunyai ukuran yang sama dengan input namun dengan *learning* dan inferensi yang efektif. Long *et al.* mengubah tiga arsitektur CNN juara ISLVRc yaitu : AlexNet, GoogLeNet dan VGGnet yang kemudian diadaptasi menjadi FCN. Perubahan dilakukan dengan membuang *classifier layer* pada jaringan dan mengubah *fully connected layer* menjadi *convolutional layer*[6].

Xuefeng Jiang (2019) dalam penelitiannya membandingkan 3 buah arsitektur *deep learning* yaitu CapsNet, CNN dan FCN dalam mengklasifikasikan citra. Xuefeng Jiang (2019) dalam penelitiannya menggunakan data set yaitu MNIST dan CIFAR-10. Dari penelitian yang dilakukan didapatkan hasil perbandingan yaitu dari kedua model *dataset* yang diuji jaringan FCN menghasilkan akurasi yang lebih tinggi yaitu 0,519 dibandingkan dengan CNN

yaitu sebesar 0,45. Selain memiliki akurasi yang lebih tinggi, parameter yang dihasilkan FCN juga lebih sedikit yaitu sebanyak 77850 dibandingkan dengan jaringan CNN sebanyak 4420170 hal ini berpengaruh terhadap kecepatan jaringan dalam mengklasifikasikan citra.

Berdasarkan uraian yang telah disebutkan, penelitian ini akan membangun sebuah sistem pendeteksian visual cacat *flash* dari hasil proses FSW dengan metode FCN yang digunakan untuk memonitoring proses pengelasan FSW. Cacat *flash* ini terjadi jika lebar las melebihi sambungan las dan pada akhirnya tidak menyatu dengan logam induk. Peneliti lebih menekankan pada proses pemeriksaan yang disertai penerapan standar inspeksi bahan khususnya metode VT serta mengamati bagaimana FCN dapat mendefinisikan cacat *flash* sebagaimana dimaksud, yang belum termuat dalam pembahasan beberapa penelitian yang telah dibahas di atas. FCN akan berperan sebagai pengganti *human vision* dalam metode VT, dari sini akan dilakukan analisa seberapa tinggi akurasi FCN serta seberapa baik performa FCN dalam mendeteksi cacat *flash*. Proses VT dengan FCN akan disebut berhasil mendeteksi cacat *flash* pada hasil proses FSW apabila memiliki persentase akurasi dan performa yang mendekati 100%. Proses VT dengan CNN diharapkan dapat mendeteksi cacat *flash* dengan baik, sehingga cacat tersebut dapat dideteksi sedini mungkin serta dapat diminimalisir dalam hasil proses FSW.

1.2 Rumusan Masalah

Rumusan masalah yang dibahas pada penelitian ini adalah sebagai berikut:

1. Bagaimana mendeteksi cacat pada permukaan las menggunakan jaringan FCN berdasarkan citra hasil pengelasan?
2. Bagaimana kinerja jaringan FCN untuk mendeteksi cacat permukaan las menggunakan FCN berdasarkan citra hasil pengelasan?

1.3 Tujuan Penelitian

Tujuan penelitian ini berdasarkan rumusan masalah yang telah ditetapkan adalah sebagai berikut:

1. Mendapatkan hasil deteksi cacat pada permukaan las menggunakan jaringan FCN berdasarkan citra hasil pengelasan.

2. Mendapatkan analisa kinerja jaringan FCN untuk mendeteksi cacat permukaan las menggunakan FCN berdasarkan citra hasil pengelasan.

1.4 Batasan Masalah

Agar penelitian menjadi terarah dan dapat mencapai sesuai dengan tujuan yang telah ditetapkan maka batasan masalah yang diberikan pada penelitian ini adalah sebagai berikut:

1. Klasifikasi yang digunakan yaitu setiap ada cacat yang dideteksi maka respon dari jaringan saraf tiruan yaitu hanya menghasilkan *output* berupa *label* pada gambar yang diproses yaitu “normal” atau “*defect*”.
2. Klasifikasi jenis-jenis cacat pengelasan yang diambil tidak diklasifikasikan pada respon.
3. Data gambar yang digunakan diambil setelah pengelasan selesai dilakukan.
4. Pengambilan data gambar dilakukan pada satu kondisi dengan pengaturan yang dijelaskan pada BAB 3.
5. Jaringan untuk klasifikasi citra yang digunakan adalah *fully convolutional network* (FCN)
6. Penentuan cacat *flash* dan tidak (normal) pada gambar hasil proses FSW berdasarkan aturan atau standar VT *ISO 25239-5* dan *American Welding Society* (AWS) D17.3.
7. Jaringan syaraf tiruan yang digunakan yaitu berbasis *deep learning*.
8. Proses yang dilakukan dibatasi hanya untuk deteksi dini cacat *flash* yang terjadi pada permukaan hasil pengelasan FSW.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Menambah pengetahuan dalam perancangan jaringan syaraf tiruan khususnya FCN untuk mengklasifikasi hasil pengelasan FSW.
2. Sebagai bahan referensi dalam penelitian selanjutnya untuk mengembangkan pengetahuan tentang pengolahan citra dan jaringan syaraf tiruan.
3. Sebagai referensi untuk penggunaan jaringan syaraf tiruan pada proses monitoring dan klasifikasi hasil pengelasan dengan metode FSW.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

2.1 Pengertian Pengelasan

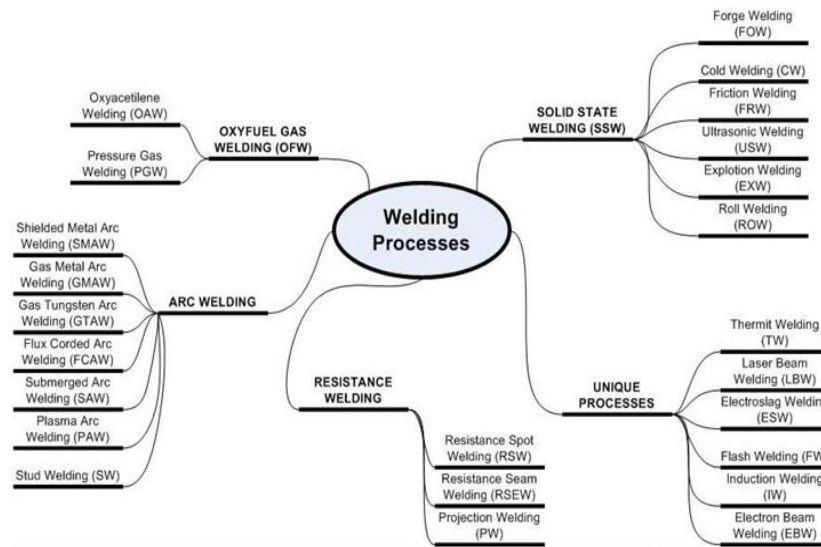
Sambungan las (*weld*) didefinisikan oleh *American Welding Society* (AWS) sebagai “penyatuan setempat dari material logam atau material bukan logam yang dihasilkan melalui pemanasan material untuk mencapai temperatur las, baik dengan atau tanpa memberikan tekanan dan dengan atau tanpa menggunakan material pengisi (*filler*)[7].

2.2 Jenis-Jenis Pengelasan

Sampai pada saat ini sudah terdapat banyak jenis-jenis pengelasan, dan diklasifikasikan dalam 2 golongan yaitu klasifikasi berdasarkan cara kerja dan klasifikasi berdasarkan energi yang digunakan. Berdasarkan klasifikasi tersebut pengelasan dapat dibagi dalam tiga kelas utama yaitu pengelasan cair, pengelasan tekan dan pematrian.

1. Pengelasan cair adalah cara pengelasan dimana sambungan dipanaskan sampai mencair dengan sumber panas dari busur listrik atau semburan api gas yang terbakar.
2. Pengelasan tekan adalah cara pengelasan dimana sambungan dipanaskan dan kemudian ditekan hingga menjadi satu.
3. Pematrian adalah cara pengelasan dimana sambungan diikat dan disatukan dengan menggunakan paduan logam yang mempunyai titik cair rendah. Dalam cara ini logam induk tidak turut mencair.

Perincian lebih lanjut dari klasifikasi ini dapat dilihat pada bagan di bawah ini. Pada bagan tersebut menerangkan beberapa cara dan jenis jenis pengelasan yang banyak dilakukan. Bagan klasifikasi pengelasan dapat dilihat pada gambar 2.1



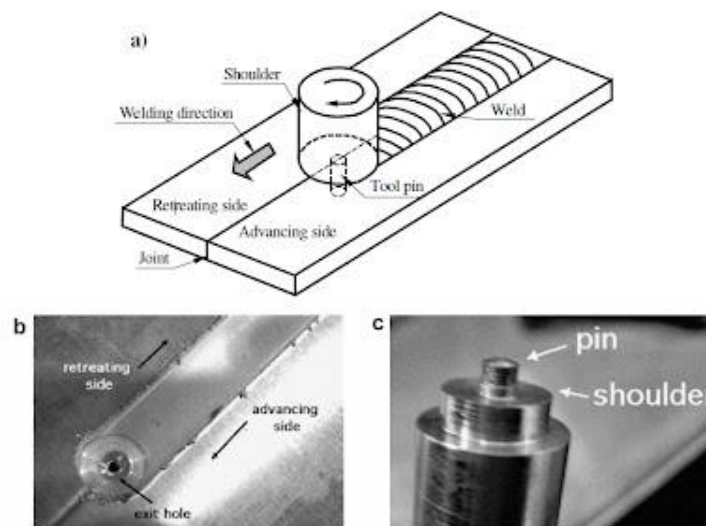
Gambar 2. 1 Klasifikasi Pengelasan[8]

2.3 Friction stir welding (FSW)

FSW (*friction stir welding*) adalah sebuah metode pengelasan yang termasuk pengelasan gesek, yang pada prosesnya tidak memerlukan bahan penambah atau pengisi. Panas yang digunakan untuk mencairkan logam kerja dihasilkan dari gesekan antara benda yang berputar (*pin*) dengan benda yang diam (benda kerja). *Pin* berputar dengan kecepatan konstan disentuhkan ke material kerja yang telah dicekam. Gesekan antara kedua benda tersebut menimbulkan panas sampai $\pm 80\%$ dari titik cair material kerja dan selanjutnya *pin* ditekan dan ditarik searah daerah yang akan dilas. Putaran dari *pin* bisa searah jarum jam atau berlawanan dengan arah jarum jam. *Pin* yang digunakan pada pengelasan *Friction stir welding* harus mempunyai titik cair dan kekerasan yang lebih tinggi dibandingkan dengan material kerja, sehingga hasil pengelasan menjadi baik. Pengelasan dengan menggunakan metode FSW biasa digunakan untuk menyambungkan material yang sama (*similar metal*) ataupun material yang tidak sama (*dissimilar metal*) seperti baja dengan baja tahan karat, aluminium dengan kuningan dan memungkinkan untuk mengelas kombinasi material lain yang tidak dapat dilas dengan menggunakan metode pengelasan yang biasa. Parameter pengelasan yang dilakukan harus disesuaikan sedemikian rupa, sehingga pengurangan volume dari pin ketika terjadi gesekan dengan material kerja bisa

diperkecil. Hal ini bertujuan untuk menjaga masukan panas (*heat input*) yang konstan sepanjang pengelasan.

Skema pengelasan *Friction stir welding* yang ditunjukkan pada gambar 2.2 dengan gesekan dua benda yang terus-menerus akan menghasilkan panas, ini menjadi suatu prinsip dasar terjadinya suatu proses pengelasan gesek. Pada proses *Friction stir welding*, sebuah *tool* yang berputar ditekan pada material yang akan disatukan. Gesekan *tool* yang berbentuk silindris (*cylindrical-shoulder*) yang dilengkapi *pin/probe* dengan material, mengakibatkan pemanasan setempat yang mampu melunakkan bagian tersebut. *Tool* yang bergerak pada kecepatan tetap (parameter 1) dan bergerak melintang (parameter 2) pada jalur pengelasan (*joint line*) dari material yang akan disatukan.

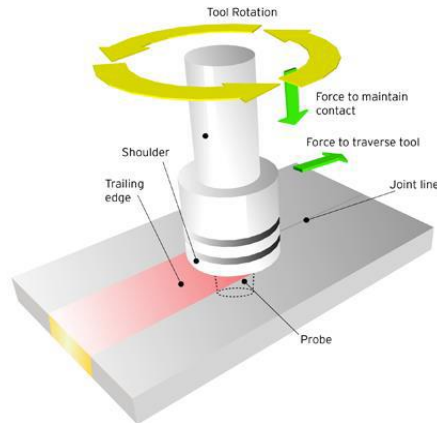


Gambar 2. 2(a) skema pengelasan FSW (b) hasil las FSW (c) geometri *tool*[9]

2.4 Prinsip Kerja FSW

Pada pengelasan FSW, *tool/probe* berputar dan bergerak dengan kecepatan konstan sepanjang jalur sambungan antara dua material yang akan dilas. Benda kerja harus dicekam dengan kuat pada ragum atau *fixture* untuk mempertahankan posisinya akibat gaya yang terjadi pada proses pengelasan. Panjang dari *probe* harus lebih pendek dari tebal benda kerja dan *shoulder* dari *tool* harus bersentuhan dengan permukaan benda kerja. Gesekan panas pada FSW dihasilkan oleh gesekan antara *probe* dan *shoulder* dari *welding tool* dengan material benda kerja. Panas lokal yang terjadi mengakibatkan adanya pelunakan logam induk pada bagian

adukan *tool*. Kombinasi putaran dan translasi *tool* FSW memungkinkan material bergerak dari sisi depan *pin* hingga sisi belakang *pin*[9].

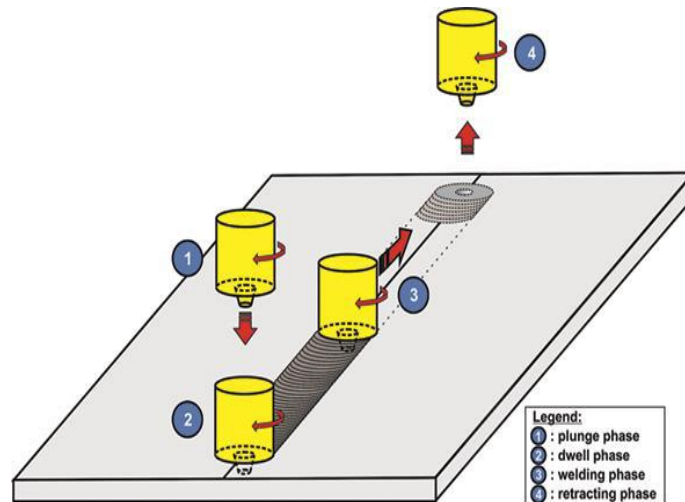


Gambar 2. 3 Prinsip FSW[9]

Ketika *pin welding tool* bergerak sepanjang jalur pengelasan, permukaan depan *pin* akan memberikan gaya dorong plastis terhadap material. Material yang akan dilas harus dicekam dengan baik dan ditempatkan di *backing* plat sehingga beban yang diberikan pada *tool* dan diteruskan ke benda kerja tidak menyebabkan bagian bawah plat yang dilas terdeformasi. Panas yang terjadi membuat material yang ada di sekitar *pin* melunak dan akibat adanya gerak rotasi dan translasi dari *tool* material di depan *pin* bergerak ke belakang *pin* dan ini akan terjadi terus menerus selama gerak translasi berlangsung dan menghasilkan sambungan yang diinginkan[9].

Untuk menggabungkan dua plat dengan menggunakan proses FSW, urutan gerakan yang akan dilakukan terbagi menjadi beberapa fase yang sesuai. Urutan ini biasanya dibagi menjadi empat fase yang berbeda. Setiap fase memainkan peran spesifik dalam proses pengelasan. Fase ini diilustrasikan pada gambar 2.4 dan diidentifikasi sebagai berikut:

1. *Plunge phase,*
2. *Dwell or stabilization phase,*
3. *Welding or advancing phase,*
4. *Tool removal or retraction phase.*



Gambar 2. 4 Rangkaian Gerakan pada FSW[10]

Selama *plunge phase*, bahannya relatif dingin, hanya pin yang kontak dengan benda kerja. Gaya aksial (juga disebut gaya tempa) dan torsi yang diterapkan pada alat itu tinggi, dan dalam kebanyakan kasus, mencapai nilai tertinggi mereka. Pada akhir *plunge phase*, *pin* telah sepenuhnya menembus (*penetrated*) benda kerja dan *shoulder* bersentuhan dengan permukaan. Kecepatan putaran alat dan fase awal seringkali sama.

Dwell phase dimulai saat kedalaman terjun yang diinginkan telah tercapai. Gaya aksial gambar 2.4 bagian 2 dipertahankan pada alat selama *stabilization phase* ini. Efek gabungan dari kecepatan relatif antara probe berputar dan material dengan gaya aksial yang diterapkan menghasilkan panas karena gesekan pada antarmuka *tool material*. Alat ini disimpan dengan waktu yang cukup lama untuk mencapai suhu yang dibutuhkan pengelasan. Setelah *dwell phase*, alat ini mulai bergerak maju dan mempercepat kecepatan translasi yang ditentukan sepanjang jalur pengelasan. Akselerasinya bisa cepat jika *dwell phase* cukup panjang dan suhunya tinggi di zona pengelasan. Namun, percepatan yang terlalu cepat dapat menghasilkan tekanan mekanis yang tinggi untuk peralatan pengelasan, dimana dapat mengurangi masa pakai alat pengelasan[10].

Proses FSW memiliki keterbatasan tertentu. Untuk membawa bahan ke dalam keadaan *plastic (plastic state)*, torsi dan kekuatan yang dibutuhkan bisa sangat tinggi. Untuk alasan ini, mesin las harus kuat, biasanya mengarah ke

peralatan yang relatif mahal. Agar memiliki lasan berkualitas tinggi, penting juga untuk memastikan penjepitan dan penyangga yang tepat. Keterbatasan lebih lanjut dari proses FSW sebagian besar terkait dengan faktor geometris. Selama pengelasan, alat bahu (*shoulder*) harus memiliki tekanan konstan dan seragam pada material[10].

2.5 Parameter FSW

Pada standar ASM (2007), *Friction stir welding* dalam prosesnya memiliki beberapa parameter yang akan menentukan seberapa baik kualitas lasan yang dihasilkan. Parameter tersebut adalah[11].

1. Kecepatan Rotasi (rpm) dan Kecepatan Las (*feeding*)

Pada proses pengelasan dengan metode *Friction stir welding*, dibutuhkan sebuah *tool* yang bergerak secara rotasi dan translasi sekaligus sebagai fungsi untuk menghasilkan panas dan pengadukan material. Pemilihan kedua parameter kecepatan ini harus dilakukan dengan tepat supaya proses pengelasan dapat berjalan dengan baik dan efisien. Untuk menghasilkan panas yang optimal, dapat dilakukan dengan dua cara, yaitu dengan memperbesar kecepatan putaran dan menurunkan kecepatan lasnya. Atau bisa juga dengan mengkombinasikan dua jenis kecepatan tersebut untuk mendapat panas yang diharapkan. Sehingga, material yang dilas dirasa cukup panas dan *viscous* sesuai yang dibutuhkan dalam pengelasan ini dan menimalisir beban pada *tool*[11].

2. *Plunge Depth*, Tekanan dan Kemiringan *Tool*

Plunge depth didefinisikan sebagai kedalaman titik terendah *probe* di bawah permukaan material yang dilas dan telah diketahui sebagai parameter kritis yang menjamin kualitas lasan. Hal ini perlu diatur dengan baik untuk menjamin tekanan ke bawah tercapai dan memastikan *tool* penuh menembus lasan. *Plunge depth* yang dangkal dapat mengakibatkan cacat dalam lasan, sebaliknya jika berlebihan bisa mengakibatkan kerusakan *tool* karena berinteraksi dengan alasnya. Tekanan pada *tool* diharapkan untuk menjaga material lunak tidak keluar jalur *joint line* dan memberi efek tempa (*forging*). Material panas ditekan dari atas oleh *shoulder* dan ditahan dari bawah oleh alas. Proses ini bertujuan

untuk memadatkan material sehingga penguatan sambungan terjadi akibat efek tempa tersebut. Selain itu, tekanan yang diberikan juga menghasilkan panas tambahan karena adanya penambahan permukaan yang bergesekan dengan material. Kemiringan *tool* adalah besarnya sudut yang dibuat antara sumbu *tool* dengan permukaan benda kerja, hal ini harus sangat diperhatikan karena akan sangat mempengaruhi hasil dari pengelasan *Friction stir welding*. Kemiringan yang dibuat berkisar $0-4^{\circ}$ di mana bagian belakang *shoulder* lebih rendah dibandingkan dengan bagian depannya[11].

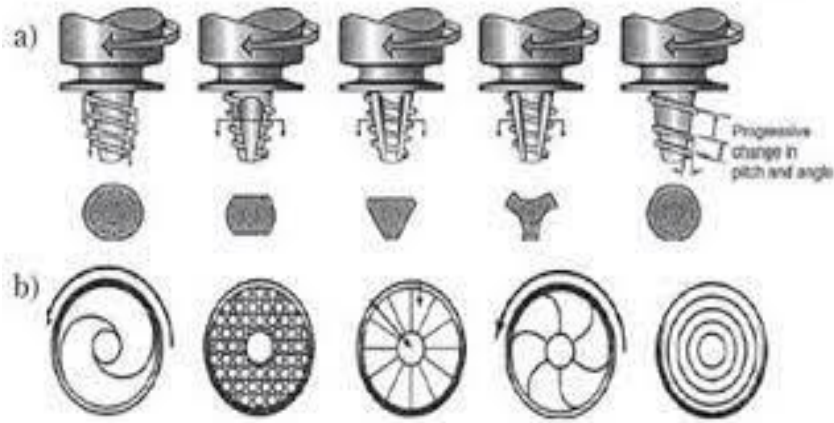
3. Desain *Tool*

Pada saat melakukan perancangan *tool*, hal pertama yang harus diperhatikan adalah pemilihan material dengan karakteristik yang harus dipenuhi. Karakteristik tersebut ialah :

- a. Memiliki kekuatan yang baik di temperatur tinggi.
- b. Stabilitas material baik di temperatur tinggi.
- c. Tahan gesek dan aus.
- d. Material tidak dapat bereaksi dengan benda kerja yang akan dilas.
- e. Ketangguhan yang baik.
- f. *Thermal Expansion* yang rendah.
- g. *Machinability* yang baik.
- h. Mikrostruktur yang homogen.

Terdapat dua bagian utama dari sebuah *tool* yang digunakan pada proses *Friction stir welding*, yaitu *shoulder* dan *probe/pin*. *Shoulder* adalah bagian yang membangun panas dengan gesekan yang dilakukan terhadap benda kerja dan mampu menjadi bagian penahan material panas yang ada di bawahnya. Serta memberikan gaya *vertical* ke arah benda kerja yang menjaga kondisi *contact tool* dengan benda kerja. *pin/nib/probe* adalah bagian yang melakukan penetrasi ke dalam benda kerja di mana bagian *tool* inilah yang mengaduk material atau mengalirkan material yang sudah melunak akibat panas yang dihasilkan *shoulder* sehingga dapat menciptakan suatu sambungan antara dua material. Dan untuk ukuran desain lain seperti ukuran *tool*, diameter *shoulder*, diameter *probe*, panjang *probe* dan panjang *tool* secara keseluruhan diatur sesuai dengan proses pengelasan yang akan dilakukan dengan melibatkan

parameter ketebalan benda kerja, kemiringan *tool* pada saat proses, kekuatan sambungan nantinya yang diharapkan serta *clearance* antar benda kerja yang dilas[11].



Gambar 2. 5 Contoh desain *Probe* pada Tool FSW[11]

2.6 Cacat Pengelasan FSW

Pada proses *Friction stir welding* terdapat ketidak sempurnaan dikarenakan parameter pengelasan yang digunakan tidak sesuai, sehingga hasil yang diinginkan kurang maksimal. Berikut beberapa kemungkinan *defect* yang dapat terjadi:

1. *Channel Defect*

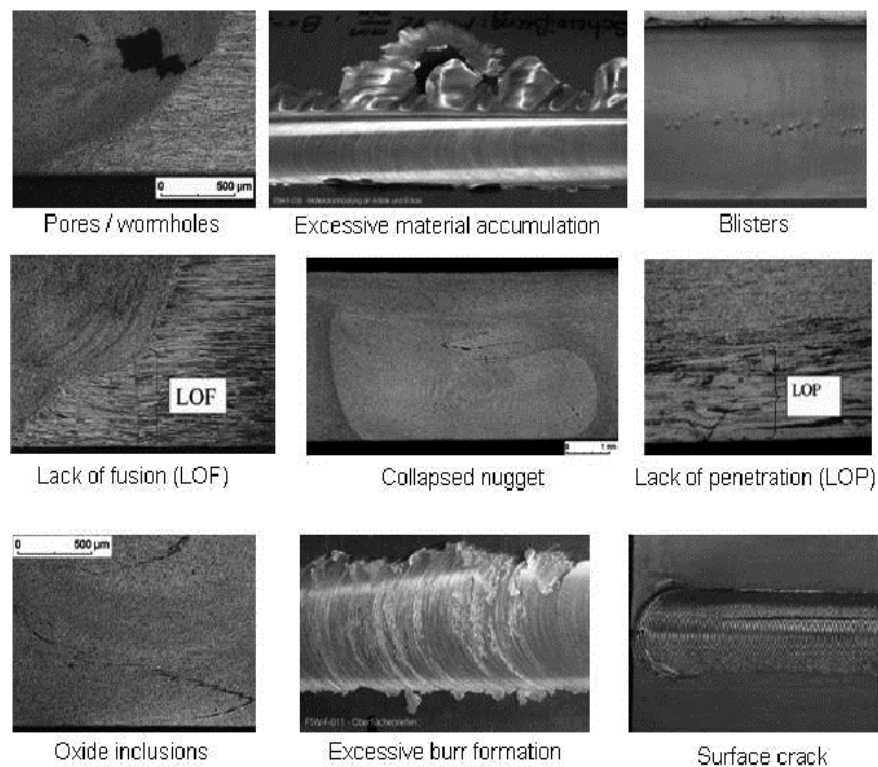
Terjadinya *channel defect* akan sulit dikendalikan apabila material yang akan disambung memiliki kuat panas yang tinggi serta konduktivitas termal yang rendah. Pada *Friction stir welding* umumnya jika terjadi kecacatan ini maka akan ditemukan pada daerah sekitar *advancing side*. *Channel defect/wormhole/tunnel defect* adalah salah satu jenis ketidaksempurnaan yang paling sering terjadi pada pengelasan dengan metode *Friction stir welding*. Cacat ini merupakan ketidaksempurnaan *volumetric* di sepanjang sambungan las yang terjadi akibat temperatur yang dihasilkan terlalu tinggi hingga mendekati titik lebur. Dengan adanya cacat ini, tentu sangat berpengaruh pada kekuatan tarik dan elongasi yang menurun cukup besar. Selain *temperature*, *channel defect* juga dapat terjadi akibat sudut kemiringan yang terlalu kecil dan kurangnya tekanan tempa dari *tool*[12].

2. *Voids*

Void dapat terbuka hingga ke permukaan meskipun sangat jarang terjadi, fenomena ini disebut dengan *surface breaking*. Fenomena ini juga dapat terjadi akibat plat yang akan disambung bergerak relatif satu sama lain atau ketika *tool* yang digunakan terangkat. Seperti halnya *channel defect*, *voids* bersifat volumetrik yang merupakan ketidaksempurnaan berupa timbulnya rongga pada sambungan las. Namun *voids* tidak terjadi sepanjang *joint line*, tetapi *intermittent* yaitu hanya pada bagian tertentu saja cacat ini terjadi[12].

3. Lack of Penetration

Penyebab terjadinya kecacatan ini adalah kurang memadainya penetrasi *pin* pada material, hal ini sebagai implikasi dari kurang panjangnya *tool* ataupun akibat terlalu besarnya sudut kemiringan. Selain itu, *lack of penetration* dapat terjadi karena *missalignment* benda kerja. Seperti halnya cacat yang lain, *lack of penetration* sering terjadi pada konfigurasi pengelasan *butt joint* dan terjadi pada bagian *root* dari sambungan.



Gambar 2. 6 Defect pada pengelasan FSW[13]

2.7 Kecerdasan Buatan (AI)

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil keputusan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik. Demikian juga agar mesin bisa cerdas (bertindak seperti dan sebaik manusia) maka harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan:

1. Basis Pengetahuan (knowledge base), bersifat fakta-fakta, teori, pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor Inferensi (inference engine), kemampuan menarik kesimpulan berdasarkan pengetahuan dan pengalaman.

Pada 1950-an, Alan Turing, seorang pionir Artificial Intelligence dan ahli matematika Inggris melakukan percobaan. Turing (turing test) yaitu sebuah komputer melalui terminalnya ditempatkan pada jarak jauh. Di ujung yang satu ada terminal dengan software *Artificial Intelligence* dan di ujung lain ada sebuah terminal dengan seorang operator. Operator itu tidak mengetahui kalau di ujung terminal lain dipasang software *Artificial Intelligence*. Mereka berkomunikasi di mana terminal di ujung memberikan respon terhadap serangkaian pertanyaan yang diajukan oleh operator. Sang operator itu mengira bahwa ia sedang berkomunikasi dengan operator lainnya yang berada pada terminal lain. Turing beranggapan bahwa jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia)[14].

2.8 Machine Learning

Machine learning atau pembelajaran mesin adalah bidang ilmu yang merupakan salah satu fokus dari bidang kecerdasan buatan. Bidang ini berbeda dengan bidang di dalam kecerdasan buatan lainnya karena tidak memerlukan programming secara eksplisit dalam pengembangannya, namun menggunakan sample data atau dataset sebagai acuan untuk membentuka keputusan.

1. Definisi *Machine Learning*

Machine Learning adalah seperangkat metode yang bisa mendeteksi pola dari data secara otomatis dan menggunakan pola yang didapat untuk memprediksi data di masa yang akan datang atau melakukan bentuk pengambilan keputusan lain dalam ketidakpastian (Bishop, 2006). Berdasarkan metodenya, bentuk dari *Machine learning* itu sendiri dapat dibagi menjadi tiga tipe berdasarkan *feedback*-nya, yaitu *Supervised Learning*, *Unsupervised Learning* dan *Reinforcement Learning*. Penelitian ini menerapkan *supervised learning* untuk mengklasifikasikan citra hasil pengelasan terdapat *flash* dan tidak.

2. *Supervised Learning*

Supervised learning adalah tipe pembelajaran mesin di mana *goal* dari sebuah agen kecerdasan buatan memperkirakan *output* berdasarkan contoh pasangan *input-ouput*. Pasangan *input-ouput* tersebut disebut training data atau *training set*. Pada *training set* data input berupa vektor dengan nilai-nilai yang merepresentasikan fitur dalam bentuk angka. Bentuk dari *output* adalah *response variable* dengan tipe data kategorikal untuk masalah klasifikasi atau nominal untuk masalah regresi[6].

3. Klasifikasi

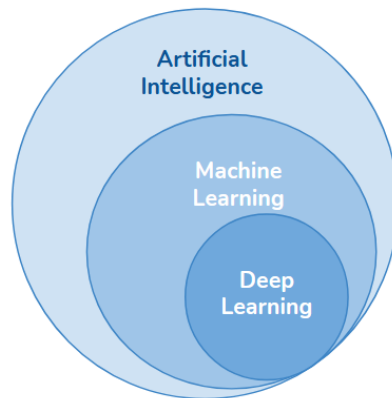
Klasifikasi adalah kasus dimana *output* yang diinginkan dari *supervised learning* berupa data kategori yang diskrit. Salah satu contoh dari masalah klasifikasi adalah pengenalan digit tulisan tangan. Dalam masalah tersebut, computer harus mengenali digit dari 0-9 dari berbagai bentuk tulisan tangan. Contoh lainnya adalah implementasi spam filter untuk memisahkan antara email asli dengan spam[6].

4. Regresi

Regresi adalah tipe masalah dimana output yang diinginkan berupa data nominal atau kontinyu. Contoh regresi seperti memprediksi harga saham di masa yang akan datang jika diberikan harga saham terkini atau parameter lainnya.

2.9 Deep Learning

Gambar 2.7 memperlihatkan bahwa *deep Learning* adalah sub-bidang yang spesifik dari *Machine Learning* (pembelajaran mesin) dan merupakan cara baru untuk mempelajari representasi dari data yang menekankan pada pembelajaran *layer* (lapisan) yang berturut-turut dan semakin bermakna.



Gambar 2. 7 Hubungan *Artificial Intelligence*, *Deep Learning* dengan *Machine Learning*[14]

Kata 'deep' pada *deep Learning* tidak mengacu pada pemahaman yang lebih dalam untuk pendekatan, sebaliknya ini merupakan gagasan dari layer yang berurutan. Banyak lapisan yang digunakan pada model data disebut kedalaman ('depth') dari model. *Deep learning* moderen sering melibatkan puluhan atau bahkan ratusan *layers* yang berurutan, dan semuanya dipelajari secara otomatis dari data pelatihan. Sementara itu, pendekatan lain untuk machine learning cenderung berfokus pada pembelajaran hanya satu atau dua lapis layer. Karena hal tersebut, mereka kadang-kadang disebut pembelajaran yang dangkal.

Meskipun *deep learning* adalah sub-bidang dari *machine learning* yang cukup lama, (mulai terkenal di awal tahun 2010-an). Dalam beberapa tahun kemudian, *deep learning* telah mencapai tidak lebih dari sebuah revolusi di

lapangan, dengan hasil yang luar biasa pada masalah persepsi seperti melihat dan mendengar, masalah yang melibatkan keterampilan yang tampak alami dan intuitif bagi manusia tetapi telah lama sulit dipahami oleh mesin. Beberapa terobosan yang telah dicapai deep learning adalah peningkatan konversi *text-to-speech*, asisten digital seperti Google Now dan Amazon Alexa dan kemampuan untuk menjawab pertanyaan-pertanyaan bahasa alami. Beberapa metode dari *deep learning* adalah *Long Short-Term Memory* (LSTM) dan *Convolutional Neural Network* (CNN)[14].

Perbedaan utama antara *deep learning* dan *machine learning* disajikan pada tabel 2.1

Tabel 2. 1 Perbedaan *machine learning* dan *deep learning*

No	Perbedaan	<i>Machine Learning</i>	<i>Deep Learning</i>
1	Jumlah Data	Mampu mengolah data dalam jumlah yang kecil	Tidak mampu mengolah data dalam jumlah kecil secara maksimal
2	<i>Hardware</i>	Dapat menggunakan perangkat dengan komputasi yang biasa saja	Membutuhkan perangkat dengan komputasi dengan kemampuan memproses data dalam jumlah tinggi
3	<i>Feature engineering</i>	mengidentifikasi fitur yang diterapkan dan membuat kode secara manual sesuai domain dan tipe data	Algoritmanya berusaha mempelajari fitur tingkat tinggi dari data
4	Pendekatan Penyelesaian Masalah	Penyelesaian <i>case</i> dilakukan secara terpisah	Menggunakan jaringan syaraf tiruan yang mampu menyelesaikan <i>case</i> dari awal sampai selesai
5	Waktu Perhitungan	Waktu yang dibutuhkan singkat berkisar beberapa menit saja	Waktu yang dibutuhkan lama
6	Kemudahan Interpretasi	Beberapa algoritma mudah diinterpretasi	Algoritmanya sulit untuk dipahami

2.10 Fully Convolutional Network

FCN dapat melakukan pelatihan *end-to-end* dan piksel demi piksel, yang memungkinkan segmentasi gambar tanpa *preprocessing* apapun. Hal ini didasarkan terutama pada beberapa jenis struktur, seperti VGGNet, AlexNet, GoogLeNet, dan sebagainya. Pengoperasian lapisan konvolusi yang sesuai dengan jaringan CNN di atas adalah sebagai berikut:

$$x_j^l = f\left(\sum_{i \in M_j} k_{ij}^l * k_i^{l-1} + b_j^l\right) \quad (2.1)$$

Dengan:

l = jumlah lapisan,

k = kernel konvolusi,

M_j = menunjukkan pilihan peta fitur *input*, dan setiap peta output memiliki *offset* b .

Fitur keluaran lapisan keluaran jaringan CNN, diikuti oleh pengklasifikasi *softmax* untuk membedakan karakteristik objek yang berbeda. Pengklasifikasi dinyatakan sebagai:

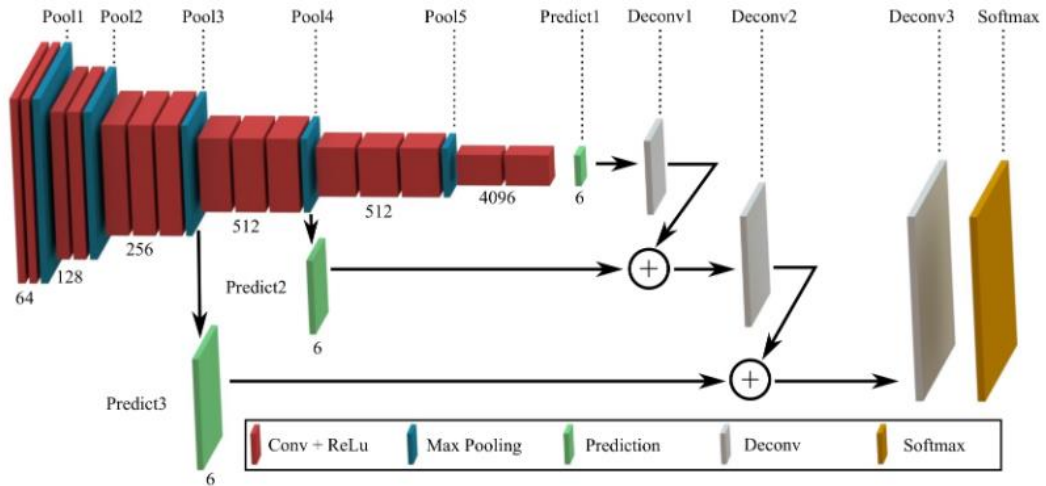
$$L_i = -\log\left(\frac{e^{f_{yi}}}{\sum e^{f_j}}\right) \quad (2.2)$$

Dengan:

f = mewakili skor

f_j = mewakili elemen ke- j dari vektor skor f

Dalam kasus di mana L_i mewakili kerugian rata-rata dari semua sampel pelatihan ketika mempertimbangkan istilah *reguler*, dan istilah *reguler* digunakan untuk menghindari *over-fitting*. Selain itu, gambar 2.8 menunjukkan struktur jaringan *Fully convolutional Network* (FCN) yang digunakan untuk segmentasi semantik.

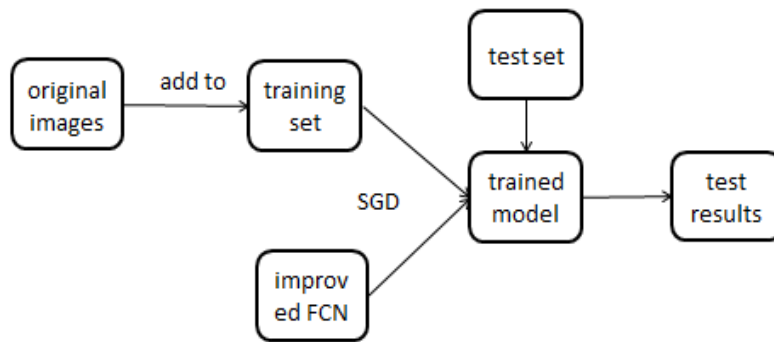


Gambar 2. 8 Arsitektur FCN VGG-16[14]

Lima *convolutional layer* di depan FCN sama dengan CNN, tetapi tiga *layer* terakhir digantikan oleh *convolutional layer*. Proses spesifik dari FCN adalah proses *up-sampling*. Kita dapat mengasumsikan bahwa ukuran citra adalah $W_1 * W_1$ dalam jaringan konvolusi lapisan tertentu, ukuran citra setelah *up-sampling* adalah $W_2 * W_2$, ukuran kernel konvolusi adalah K , panjang langkah adalah S , dan komplement tepinya adalah P . Persamaan untuk operasi dekonvolusi adalah:

$$W_2 = (W_1 - 1) * S + K - 2P \quad (2.3)$$

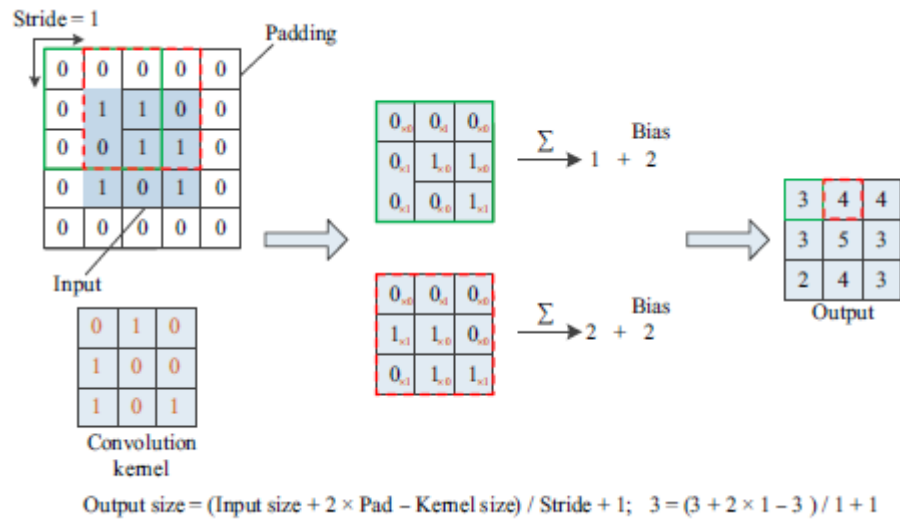
Dimana *up-sampling* dapat dipahami lebih lanjut, yaitu, gambar dengan ukuran $512 * 512$ pertama-tama direduksi menjadi $W_1 * W_1$, dan kemudian dilakukan operasi untuk mendapatkan $W_2 * W_2$, yang sesuai dengan operasi dekonvolusi. Berbeda dari *output* satu dimensi setelah sepenuhnya terhubung oleh lapisan *output* dari lapisan konvolusi adalah matriks dua dimensi memberikan kemungkinan prediksi tingkat piksel. FCN menggunakan *ground truth masks* dari gambar yang diberi label secara manual untuk memantau informasi guna melatih sebuah jaringan *end-to-end*, sehingga memungkinkan jaringan untuk membuat prediksi tingkat piksel dan akhirnya menghasilkan gambar label. Tahapan pelatihan pada FCN seperti pada gambar 2.9.



Gambar 2. 9 *Training* dan *Testing* proses pada FCN[15]

2.10.1 Convolutional Layer

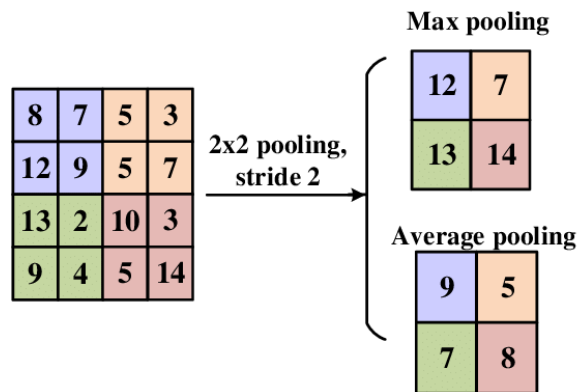
Di CNN maupun FCN, sebuah lapisan konvolusi melakukan operasi konvolusi menggunakan satu set *kernels* (filter) dengan bobot yang dapat dipelajari seperti yang ditunjukkan pada gambar 2.10. Kedalaman *convolutional kernels* selalu sama dengan kedalaman lapisan konvolusi input, dan tinggi dan lebar yang identik yang umumnya berukuran lebih kecil dari input. Konvolusi diimplementasikan antara input dan kernel di mana setiap kernels meluncur dimasukan dengan ukuran langkah tertentu. Ukuran langkah geser ditentukan sebagai langkah yang biasanya identik dalam tinggi dan lebar arah. Nilai konvolusi yang dihitung dari setiap kernels ditambahkan bersama-sama dengan bias untuk menghasilkan hasil dari masing-masing inti. Hasil ini digabungkan untuk dapat menghasilkan spasial keluaran dari lapisan konvolusi yang ada. Untuk mempertahankan ukuran keluaran, zero-padding (Pad) selalu diadopsi untuk input. Hasil ukuran lapisan convolutional tergantung pada ukuran input, Pad, ukuran kernels, dan langkah, yang dapat dihitung menggunakan persamaan[16] jaringan konvolusi yang disajikan pada gambar di bawah ini.



Gambar 2. 10 Convolutional layer example[16]

2.10.2 Pooling Layer

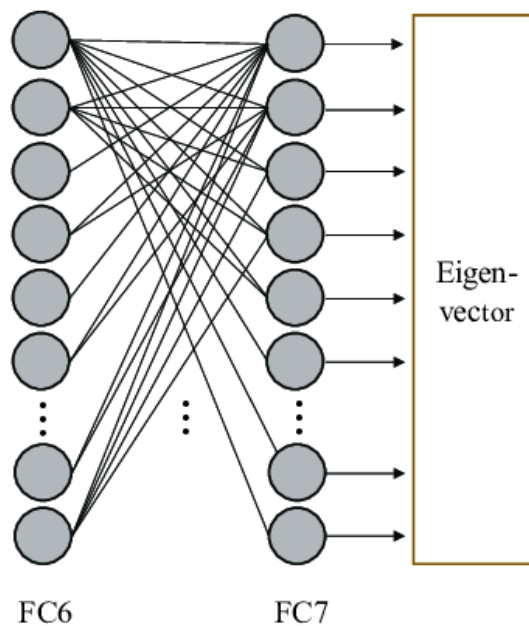
Pooling layer adalah lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari feature map (downsampling) sehingga mempercepat komputasi karena parameter yang diperbarui semakin sedikit. Selain itu, berguna untuk mengekstraksi fitur dominan sehingga proses pelatihan model lebih efektif. Ada dua jenis pooling layer, yaitu max pooling dan average pooling[15]. *Max pooling* mengembalikan nilai maksimum dari bagian gambar yang dicakup oleh kernel sedangkan *average pooling* mengembalikan nilai rata – rata dari bagian gambar yang dicakup oleh kernel.



Gambar 2. 11 Maxpooling Layer[15]

2.10.3 Fully Connected Layer

Fully connected layer adalah lapisan yang digunakan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Untuk mendapatkan hasil keluaran dari *layer* ini tidak dibutuhkan operasi konvolusi, tetapi menggunakan komputasi perkalian matriks yang diikuti dengan bias *offset*. Dengan penggunaan operasi tersebut, setiap neuron memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya, sehingga *layer* ini disebut sebagai *fully connected layer*[16].



Gambar 2. 12 *Fully Connected Layer* [17]

2.10.4 SoftmaxLayer

Softmax Classifier adalah sebuah fungsi aktivasi yang di gunakan untuk permasalahan klasifikasi, biasanya fungsi aktivasi ini digunakan pada output layer. Pada dasarnya fungsi ini adalah probabilitas eksponensial yang dinormalisasi dari pengamatan kelas yang di wakili sebagai aktivasi neuron. Fungsi eksponensial akan meningkatkan probabilitas nilai maksimum lapisan sebelumnya dibandingkan dengan nilai lainnya. *Softmax function* adalah perhitungan kemungkinan dari masing-masing kelas target atas semua kelas target yang memungkinkan dan membantu untuk menentukan kelas target untuk input yang diberikan. *Softmax* mempunyai keuntungan, seperti nilai rentang probabilitas yang dihasilkan dari 0

hingga 1 dan jumlah semua kemungkinan sama dengan satu. Ketika *softmax* digunakan untuk model klasifikasi multi, maka akan mengembalikan peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas lebih tinggi dari kelas yang lain. Perhitungan softmax menggunakan eksponensial dari nilai masukan yang diberikan dan jumlah dari nilai eksponensial dari semua nilai dalam masukan[18]

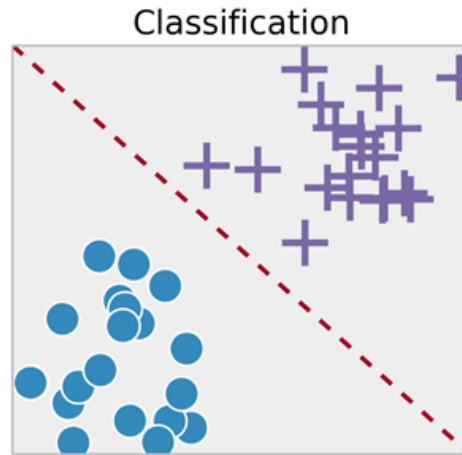
2.10.5 Batch Normalization Layer

Pertama kali diperkenalkan oleh Loffe dan Szegedy dalam makalah 2015 mereka, normalisasi *Batch*: mempercepat pelatihan jaringan dalam dengan mengurangi *internal covariate shift*, lapisan normalisasi *batch* (atau singkatnya BN), seperti namanya digunakan untuk menormalkan aktivasi sebuah volume input sebelum meneruskannya ke lapisan berikutnya dalam jaringan. Normalisasi *batch* telah terbukti sangat efektif dalam mengurangi jumlah epoch yang dibutuhkan untuk melatih jaringan saraf. Normalisasi *batch* juga memiliki manfaat tambahan untuk membantu pelatihan dalam menstabilkan serta memungkinkan variasi tingkat pembelajaran dan kekuatan regularisasi yang lebih besar. Menggunakan normalisasi *batch* tidak mengurangi kebutuhan untuk menyempurnakan parameter, tetapi itu akan membuat proses pembelajaran lebih mudah dengan membuat tingkat belajar dan regularisasi menjadi lebih stabil dan lebih mudah. Kelemahan terbesar dari normalisasi *batch* adalah program dapat benar-benar memperlambat waktu yang diperlukan untuk melatih jaringan (meskipun akan membutuhkan lebih sedikit waktu untuk mendapatkan akurasi yang lebih baik) sebesar 2 sampai 3 kali karena perhitungan *statistic per-batch* dan normalisasi. Disarankan menggunakan normalisasi *batch* di hampir setiap situasi karena itu membuat perbedaan yang signifikan[19].

2.10.6 Classification layer

Classification dapat didefinisikan sebagai proses memprediksi kelas atau kategori dari nilai yang diamati atau titik data yang diberikan. *Output* yang dikategorikan dapat memiliki bentuk seperti “Black” atau “White” atau “spam” atau “no spam”. Secara matematis, *classification* adalah tugas mendekati fungsi

pemetaan (f) dari variabel *input* (X) ke variabel *output* (Y). Ini pada dasarnya milik pembelajaran mesin yang diawasi di mana target juga disediakan bersama dengan set data input.



Gambar 2. 13 Contoh Klassifikasi pada spam email[19]

Contoh masalah *classification* dapat berupa deteksi *spam* di email. Hanya ada dua kategori *output*, “spam” dan “no spam”; maka ini adalah *classification* tipe biner. Untuk mengimplementasikan *classification* ini, pertama-tama kita perlu melatih *classifier*. Untuk contoh ini, email “spam” dan “no spam” akan digunakan sebagai data pelatihan. Setelah berhasil melatih *classifier*, ini dapat digunakan untuk mendeteksi email yang tidak dikenal.

2.10.7 Stride dan Padding

1. Stride

Stride adalah parameter yang menentukan berapa jumlah pergeseran *filter*. Jika nilai *stride* adalah 1, maka *conv.filter* akan bergeser sebanyak 1 pixels secara *horizontal* lalu *vertical*. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah input, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Dan perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil kita tidak selalu akan mendapatkan performa yang bagus.

2. Padding

Padding atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *conv. layer* (Feature Map).

Tujuan dari penggunaan padding adalah :

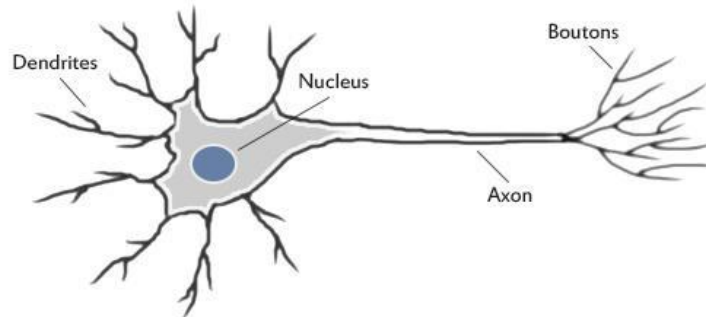
- a. Dimensi *output* dari *conv. layer* selalu lebih kecil dari inputnya (kecuali penggunaan 1x1 filter dengan stride 1). Output ini akan digunakan kembali sebagai input dari *conv. layer* selanjutnya, sehingga makin banyak informasi yang terbuang. Dengan menggunakan *padding*, kita dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan *conv. layer* yang lebih dalam/*deep* sehingga lebih banyak *features* yang berhasil di-*extract*.
- b. Meningkatkan performa dari model karena *conv. filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut.

2.11 Neural Network (NN)

Neural network merupakan bagian dari sistem kecerdasan buatan yang digunakan untuk memproses informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya[20]. Pada gambar 2.14 menunjukkan salah satu contoh syaraf secara biologis dimana setiap sel syaraf (*Neuron*) akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh *dendrit*.

Selain menerima informasi, *dendrit* juga menyertai *Axon* sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi *neuron* lain dimana antar *dendrit* kedua sel tersebut dipertemukan dengan sinapsisnya. Informasi yang dikirimkan antar *neuron* ini berupa rangsangan yang dilewatkan melalui *dendrit*. Informasi yang datang dan diterima oleh *dendrit* akan dijumlahkan dan dikirim melalui *axon* lain. Informasi ini akan diterima oleh *neuron*

lain jika memenuhi batasan tertentu dikenal dengan nilai ambang (*threshold*) yang dikatakan teraktivasi.



Gambar 2. 14 Syaraf Secara Biologi[20]

Pada umumnya jaringan syaraf tiruan memiliki tiga lapisan, yaitu *input layer*, *hidden layer*, dan *output layer*. Berikut penjelasan mengenai *layer* pada *Neural Network*.

1 *Input Layer*

Input layer berisi *neuron-neuron* yang masing-masing menyimpan sebuah nilai masukan yang tidak berubah pada fase latih dan hanya bisa berubah jika diberikan nilai masukan baru. *Neuron* pada lapisan ini tergantung pada banyaknya input dari suatu pola.

2 *Hidden Layer*

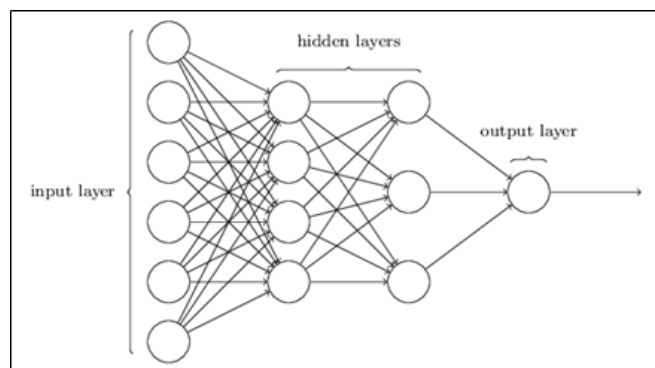
Lapisan ini tidak pernah muncul sehingga dinamakan *hidden layer*. Akan tetapi semua proses pada fase pelatihan dan fase pengenalan dijalankan di lapisan ini. Jumlah lapisan ini tergantung dari arsitektur yang akan dirancang, tetapi pada umumnya terdiri dari satu lapisan *hidden layer*.

3 *Output Layer*

Output layer berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada lapisan *hidden layer* berdasarkan *input* yang diterima.

Neural network ditentukan oleh tiga hal, yaitu pola hubungan antar-*Neuron* yang disebut jaringan. Metode untuk menentukan bobot penghubung yang disebut metode *training / learning / algoritma*, dan Fungsi *aktivasi* atau fungsi transfer. Salah satu arsitektur NN yang sangat populer adalah *multi layer feedforward networks*. Secara umum, jaringan seperti ini terdiri dari sejumlah unit *Neuron* sebagai lapisan *input*, satu atau lebih lapisan simpul-simpul *Neuron* komputasi *hidden* (lapisan tersembunyi), dan sebuah lapisan simpul-simpul *Neuron* komputasi output. Sinyal input dipropagasikan ke arah depan (arah lapisan *output*), lapisan demi lapisan. Jenis jaringan ini adalah hasil generalisasi dari arsitektur *perceptron* satu lapisan, jadi biasa disebut sebagai *Multi Layer Perceptron (MLP)*. Propagasi balik (ke arah lapisan input) terjadi setelah jaringan menghasilkan *output* yang mengandung *error*. Pada fase ini seluruh bobot *synaptic* (yang tidak memiliki aktivasi nol) dalam jaringan akan disesuaikan untuk mengkoreksi/memperkecil *error* yang terjadi (*error correction rule*). Untuk pelatihan jaringan, pasangan fase propagasi ke depan dan balik dilakukan secara berulang untuk satu set data latihan, kemudian diulangi untuk sejumlah *epoch* (satu sesi lewatan untuk seluruh data latihan dalam sebuah proses pelatihan jaringan) sampai *error* yang terjadi mencapai batas kecil toleransi tertentu atau nol.

MLP terdiri dari beberapa unit pemroses (*Neuron*) seperti pada gambar 2.10, yang terhubung dan mempunyai beberapa masukan serta memiliki satu atau beberapa *output*. *Perceptron* digunakan untuk menghitung jumlah nilai perkalian penimbang dan masukan dari parameter yang kemudian dibandingkan dengan nilai *threshold*, jika keluran lebih besar dari *threshold* maka keluaran adalah satu, sebaliknya adalah nol[21].



Gambar 2. 15 Struktur MLP[21]

Pernyataan ini merupakan hasil proses pelatihan yang dalam bentuk bahasanya adalah pernyataan iya atau tidak. Secara matematis dapat ditulis dengan persamaan berikut:

Jumlah perkalian penimbang dan parameter masukan adalah:

$$I = w_{ji}x_i \quad (2.4)$$

Dari persamaan 2.4, x_i adalah sinyal masukan sedangkan w_i adalah penimbang. Bila $I > T$ maka keluaran $O = 1$ sebaliknya $O = 0$, dengan T adalah *threshold*. Pelatihan pada perceptron dilakukan dengan merubah nilai penimbangnya sehingga sesuai dengan kebutuhan, dilakukan dengan membandingkan keluaran dari jaringan syaraf tiruan dengan targetnya, proses tersebut dapat dituliskan pada persamaan 2.5:

$$w_{baru_{ji}} = w_{lama_{ji}} + \alpha(t_j - O_j)X_i \quad (2.5)$$

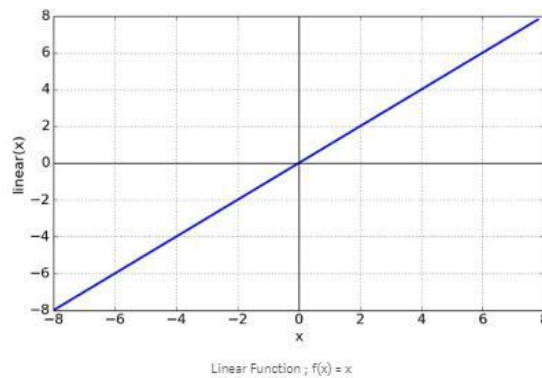
t_j adalah target dan α adalah bilangan konstan bernilai antara 0,1 sampai 0,9 yang disebut dengan laju pelatihan (*learning rate*). Proses diatas dijalankan pada setiap Neuron yang ada pada setiap layer sampai penimbang tersebut sesuai dengan yang diinginkan. Nilai awal penimbang adalah bilangan kecil yang diambil secara acak.

2.11.1 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas *internal* (*summation function*) yang mungkin berbentuk *linear* ataupun *non-linear*. Fungsi ini bertujuan untuk menentukan apakah *neuron* diaktifkan atau tidak. Menurut Samuel Sena dalam artikelnya yang diunggah dalam Website Medium[22]. Ada beberapa fungsi aktivasi yang sering digunakan dalam *Neural Network*, yaitu sebagai berikut:

1. Fungsi Aktivasi *Linear*

Fungsi Aktivasi *linear* merupakan fungsi yang memiliki nilai *output* yang sama dengan nilai *input*-nya. Hal ini berkaitan dengan jika sebuah *neuron* menggunakan *linear activation*, maka keluaran dari *neuron* tersebut adalah *weighted sum* dari *input* + *bias*. Grafik fungsi *linear* ditunjukkan oleh gambar 2.16.

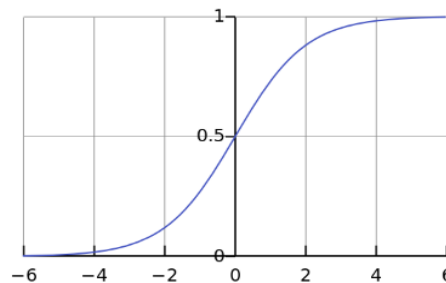


Gambar 2. 16 Fungsi Aktivasi Linear[22]

2. Fungsi aktivasi *sigmoid*

Fungsi aktivasi *sigmoid* merupakan fungsi *nonlinear*. Masukan untuk fungsi aktivasi ini berupa bilangan *real* dan *output* dari fungsi tersebut memiliki *range* antara 0 sampai 1. Berikut ini grafik fungsi aktivasi *sigmoid* pada gambar 2.17:

$$A = \frac{1}{1+e^{-x}}$$



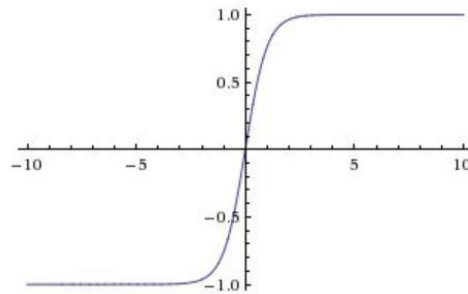
Gambar 2. 17 Fungsi Aktivasi *Sigmoid*[22]

Jika input dari suatu *node* pada *neural network* bernilai negatif maka keluaran yang didapatkan adalah nol, sedangkan jika masukannya bernilai positif maka keluaran nilainya adalah satu. Fungsi ini memiliki kekurangan yaitu *sigmoid* dapat mematikan *gradient*, ketika aktivasi dari *neuron* mengeluarkan nilai yang berada pada *range* nol atau satu, dimana *gradient* di wilayah ini hampir bernilai nol. Kemudian *output* dari *sigmoid* tidak *zero-centered*.

3. Fungsi Aktivasi *Tanh*

Fungsi aktivasi *Tanh* merupakan fungsi *nonlinear*. Masukan untuk fungsi aktivasi ini berupa bilangan real dan *output* dari fungsi tersebut memiliki range antara -1 sampai 1. Berikut ini grafik fungsi aktivasi *tanh*:

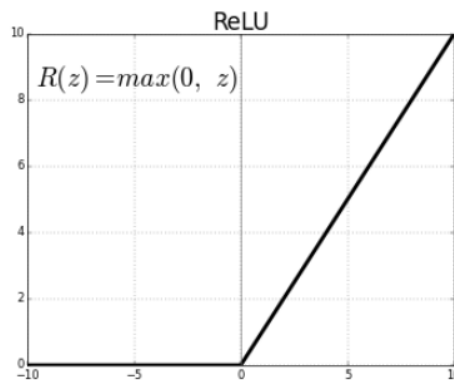
$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.6)$$



Gambar 2. 18 Fungsi Aktivasi Tanh[22]

4. Fungsi Aktivasi ReLU

Pada dasarnya fungsi ReLU (*Rectified Linear Unit*) melakukan “*threshol*” dari nol hingga *infinity*. Fungsi ini menjadi salah satu fungsi yang populer saat ini. Berikut ini grafik fungsi aktivasi ReLU.



Gambar 2. 19 Fungsi aktivasi ReLU[22]

Pada fungsi ini masukan dari *neuron-neuron* berupa bilangan negatif, maka fungsi ini akan menerjemahkan nilai tersebut kedalam nilai 0, dan jika masukan bernilai positif maka *output* dari neuron adalah nilai aktivasi itu sendiri. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan

dengan fungsi *sigmoid* dan *tanh*. Namun aktivasi ini juga memiliki kelemahan yaitu aktivasi ini bisa menjadi rapuh pada proses *training* dan bisa membuat unit tersebut mati.

2.11.2 Evaluasi Kinerja

Pengukuran kinerja yang dibuat menggunakan basis konsep dari *confusion matrix* seperti pada Tabel 2.3 dimana kelas (+) merepresentasikan kelas *actual data*. Sedangkan kelas (-) merupakan kelas *predicted data*. Evaluasi kinerja model terhadap dua kelas yang berbeda diberikan dalam bentuk Akurasi, *Recall*, *Precision*, *F-Measure* untuk masing-masing kelas. Formulasi masing-masing ukuran kinerja dapat dilihat dalam bentuk persamaan 2.7 persamaan 2.8. Akurasi adalah ketepatan pengukuran tingkat kedekatan terhadap nilai yang sebenarnya. Sebuah sistem pengukuran dapat akurat dan tepat, atau akurat tetapi tidak tepat, atau tepat tetapi tidak akurat atau tidak tepat dan tidak akurat.

Precision merupakan perhitungan ketepatan klasifikasi pada jumlah data berlabel positif atau data kelas minoritas yang memang benar secara aktual merupakan kelas positif. Sedangkan, *recall* (disebut juga *sensitivity*) adalah perhitungan ketepatan klasifikasi pada jumlah data positif yang teridentifikasi benar sebagai kelas positif. *F-measure* sendiri adalah *relative importance* antar *precision* dan *recall*.

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.8)$$

$$F - measure = \frac{(1 + \beta^2) \times Recall \times Precision}{\beta^2 \times Recall \times Precision} \quad (2.9)$$

Pengukuran kinerja pada tesis ini menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP)

merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negatif berikut[23].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

$$Specificity = \frac{TN}{TN + FP} \quad (2.11)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.12)$$

$$F1\ Score = \frac{2(Recall * Precision)}{Recall + precision} \quad (2.13)$$

Tabel 2. 2 *Confussion Matrix*

	Prediksi (+)	Prediksi (-)
Aktual (+)	TP	FN
Aktual (-)	FP	TN

2.12 Penelitian Sebelumnya

Berdasarkan penelitian yang akan dilakukan, beberapa penelitian sebelumnya menjadi sangat penting digunakan oleh penulis sebagai acuan dalam melakukan penelitian. Sehingga penulis dapat memperkaya referensi yang digunakan dalam mengkaji penelitian yang akan dikerjakan. Selain itu dengan membaca penelitian sebelumnya diharapkan dapat menghindari adanya suatu duplikasi dalam penelitian yang akan dikerjakan. Penelitian sebelumnya diangkat dari naskah publikasi-publikasi yang telah dikerjakan sebelumnya.

Banyak pengembangan sistem yang meneladani *Computer Vision* seperti *face detection*, *image recognition* maupun pengenalan pola tertentu. Pengembangan sistem ini menjadi sebuah fungsionalitas yang dapat mempermudah pekerjaan diberbagai bidang. Pengembangan dari *deep learning* ini sangat tepat dan efektif untuk digunakan dalam menyelesaikan permasalahan tersebut. Hal ini tidak lepas dengan adanya riset atau penelitian di bidang tersebut. Penelitian terdahulu mengenai metode *image processing* dalam mendeteksi cacat pada permukaan hasil pengelasan sudah banyak dilakukan oleh para *reaseacher* pada berbagai macam

object maupun identifikasi seperti mendeteksi cacat pada permukaan logam hasil pengelasan FSW maupun pengelasan lainnya. Adapun penelitian yang dilakukan oleh R. Ranjan *et al.* (2016) teknik pengolahan citra digunakan untuk mengidentifikasi cacat yang terjadi di permukaan sambungan FSW. Cacat seperti *voids*, *grooves*, *crack*, *key-hole*, dan *flash* diidentifikasi melalui algoritma *image pyramid* dan *image reconstruction* untuk menunjukkan cacat yang didefinisikan sebagai gumpalan. Masing-masing cacat yang terjadi diklasifikasikan dengan menganalisis ukuran dan orientasi gumpalan putih pada gambar berdasarkan fitur unik yang dimilikinya[24]. Dari R. Ranjan *et al* (2016) diperoleh kesimpulan bahwa pengujian hasil pengelasan dengan menggunakan metode *machine learning* dan *image processing* dapat digunakan untuk mendeteksi dan mengklasifikasikan cacat yang terdapat pada permukaan hasil pengelasan FSW.

S. Sudhagar *et al.* (2019) dengan metode *image processing* untuk mengklasifikasikan hasil pengelasan FSW. Citra yang diolah dihasilkan dengan melakukan pengelasan menggunakan parameter yang berbeda-beda. Algoritma *maximally stable extremal region* digunakan dalam mendeteksi permukaan pengelasan yang diprediksi memiliki cacat dan *Support Vector Machines (SVM)* digunakan dalam mengklasifikasikan citra pengelasan yang telah dideteksi sebelumnya. Untuk meningkatkan akurasi algoritma SVM dilatih dengan memberikan variasi pada fungsi kernel yang digunakan yaitu *linear*, *quadratic*, *polynomial* dan *radial basis*. Hasil SVM menunjukkan 95.8% akurasi pada *Linear Kernel Function* dihasilkan untuk mengidentifikasi pengelasan yang baik dan pengelasan yang tidak baik[25]. Dari penelitian R. Ranjan *et al.* (2016) dan S. Sudhagar *et al.* (2019) yang menggunakan metode *machine learning* peneliti menggunakan metode *deep learning* dalam proses klasifikasi yang dilakukan hal ini bertujuan untuk menghemat daya dan menghasilkan program yang lebih kompleks sehingga *feature extraction* dapat dilakukan oleh jaringan *deep learning* yang dirancang.

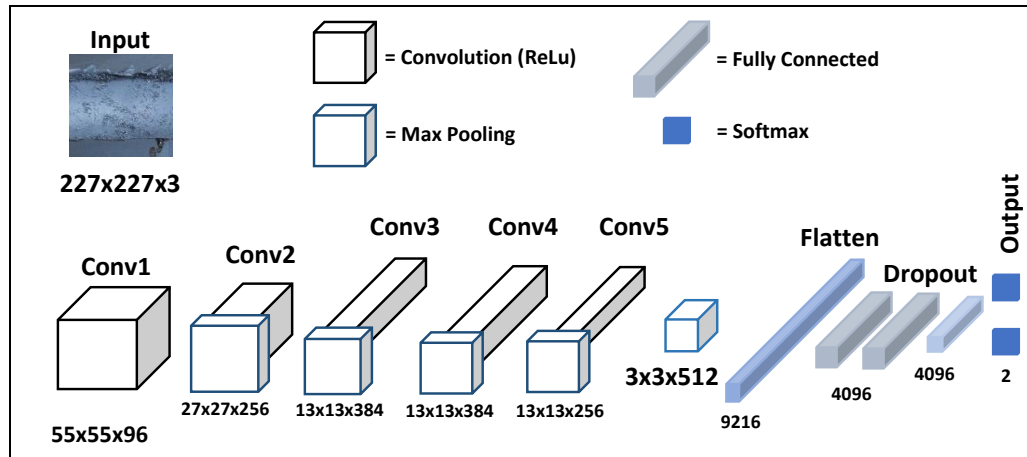
Hartl *et al.*(2020), permasalahan yang dibahas pada penelitian ini adalah membandingkan 3 buah arsitektur jaringan *deep learning* yang digunakan untuk mendeteksi cacat permukaan pada FSW yaitu FCNN, RNN dan CNN. Metode yang digunakan adalah dengan pemantauan secara tidak langsung dari kualitas

permukaan. Berbagai sensor digunakan untuk akuisisi *inline* dari akselerasi, gaya, torsi spindel dan temperatur. Tiga arsitektur dicoba dalam memprediksi kualitas permukaan las akan baik atau cacat. Hasil terbaik yang didapat saat mengevaluasi torsi spindel oleh CNN dengan akurasi prediksi 87,4%. Dari penelitian Hart *et al* (2020), peneliti hanya menggunakan modifikasi jaringan CNN menjadi FCN karena dari perbandingan dengan metode deep learning lainnya CNN memiliki nilai akurasi yang paling tinggi[26].

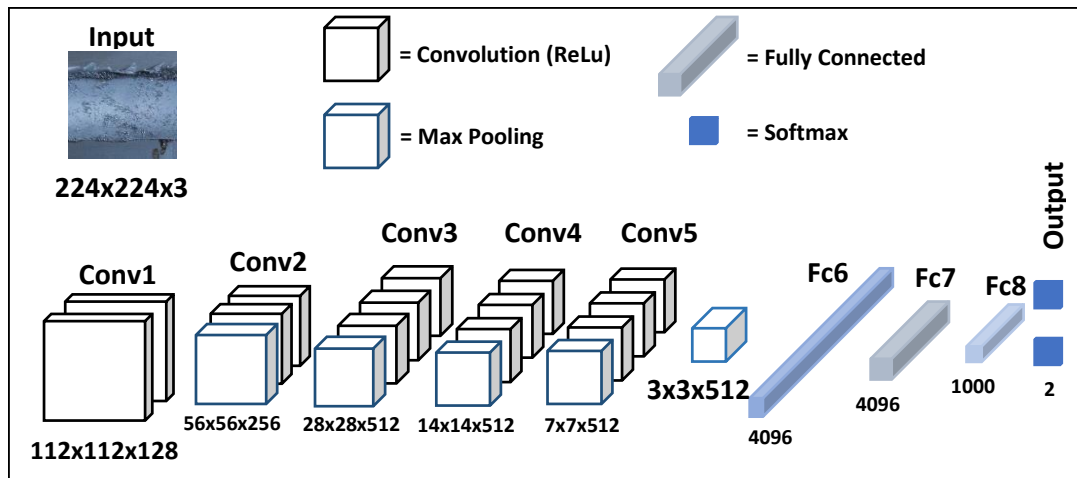
S Wang *et al.* (2021) jaringan *fully convolutional network* (FCN) diusulkan untuk mengatasi masalah hilangnya informasi lokal dan pengurangan kapasitas perbaikan parsial untuk peningkatan akurasi deteksi. Penggunaan jaringan berbasis FCN digunakan untuk menentukan lokasi dan ukuran setiap retakan yang terdapat pada balok baja. Penelitian yang dilakukan menggunakan 6 jenis jaringan FCN yaitu: FCN-8s-Conv4, FCN-8s-Conv4-Conv5, FCN-8s, FCN-8s+2conv, FCN-8s+2conv+Conv6 dan FCN-8s+2conv+Conv+Dec4. Hasil penelitian tersebut menyatakan bahwa jaringan FCN-8s-Conv4-Conv5 memiliki nilai presisi dan *recall* yang lebih tinggi dibandingkan jaringan FCN lainnya. Berbeda dengan penelitian yang dilakukan S Wang *et al.* (2021) dalam penelitian ini metode FCN yang digunakan hanya untuk klasifikasi dan bukan untuk segmentasi citra sehingga modifikasi pada jaringan FCN akan dilakukan yaitu dengan mengubah *pixel classification* menjadi *layer classification* hal ini disesuaikan dengan kebutuhan dari penelitian ini. Selain itu perbedaan juga dapat dilihat pada benda kerja yang diuji dimana dalam penelitian S Wang *et al.* (2021) menggunakan struktur *beam* sedangkan dalam penelitian ini menggunakan citra hasil pengelasan FSW[27].

Ulya (2022) dalam penelitiannya menggunakan jaringan CNN dalam mendeteksi cacat *flash* pada permukaan hasil pengelasan FSW. Penelitian yang dilakukan dengan menggunakan dua jenis jaringan CNN yaitu model AlexNet dan model VGG-16. Menggunakan citra FSW sebanyak 620 citra sebagai *data set* yang telah dikelompokkan kedalam kelas defect dan normal. Jaringan CNN yang dirancang mampu mengklasifikasikan citra pengelasan FSW hingga akurasi 91,03% pada model AlexNet dan 77,35% dengan model VGG-16[28]. Dari penelitian yang telah dilakukan dibandingkan dengan CNN yang dirancang sebelumnya, jaringan FCN memiliki jaringan yang lebih sederhana, karena hanya

menggunakan jaringan *convolution* dan hampir tidak memiliki jaringan *dense* seperti halnya . Oleh sebab itu parameternya menjadi lebih sedikit dan proses perhitungan menjadi lebih singkat[28]. Perbandingan jaringan disajikan pada gambar 2.20



Gambar 2. 20 Jaringan CNN AlexNet[28]



Gambar 2. 21 Jaringan VGG-16[28]

Secara singkat seluruh penelitian yang digunakan sebagai acuan dalam penelitian ini disajikan pada tabel 2.3

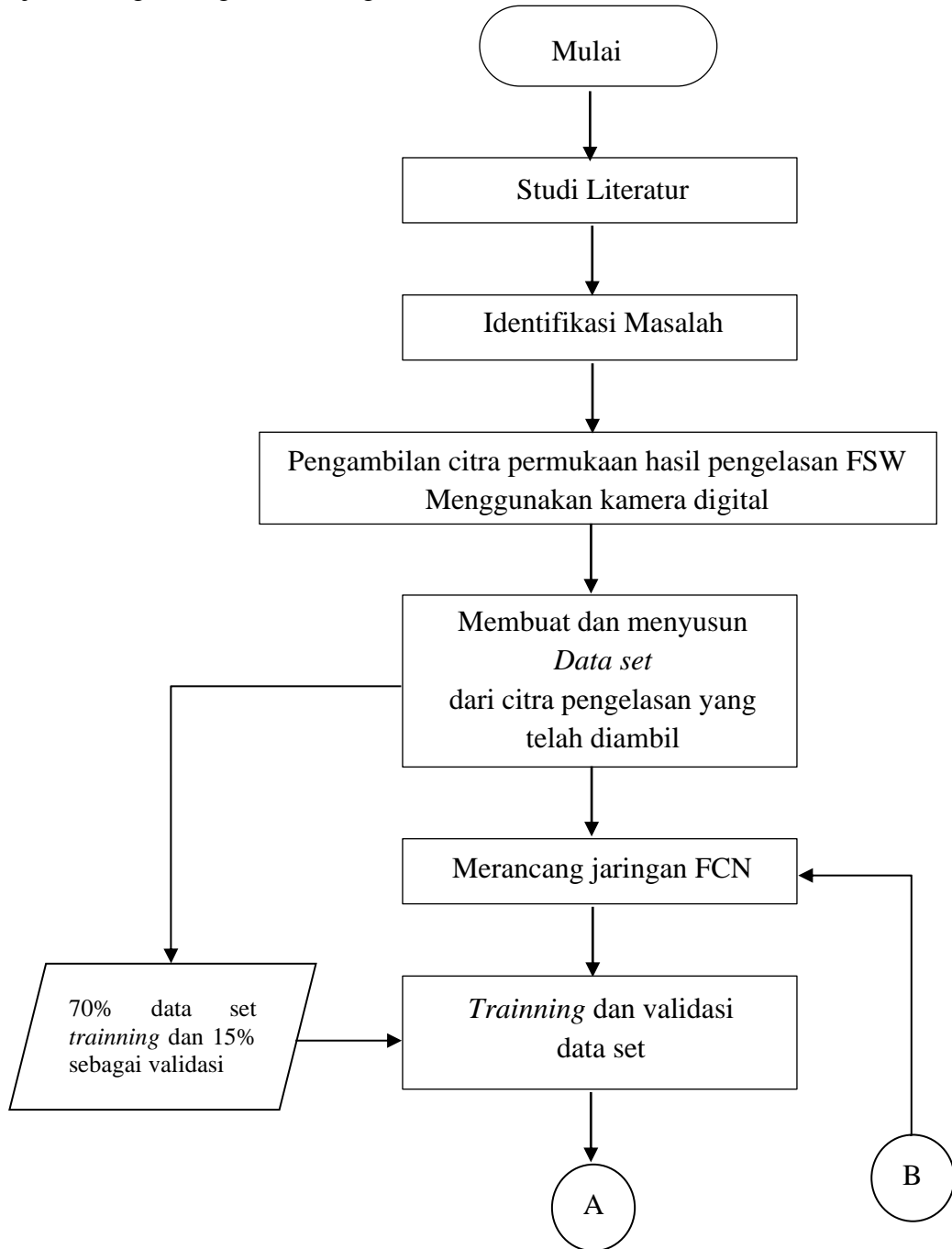
Tabel 2. 3 Perbandingan Penelitian

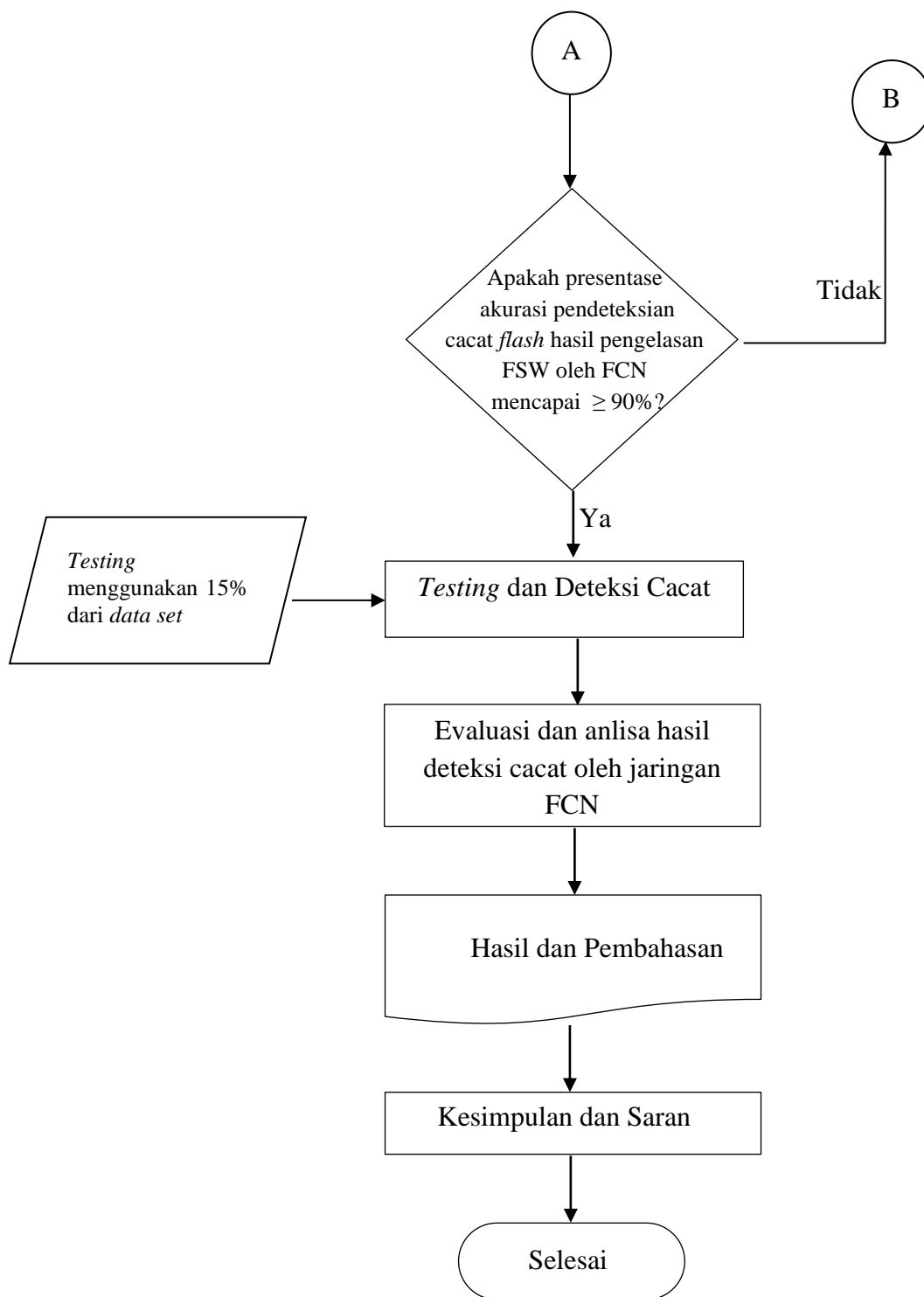
No	Peneliti dan Tahun	Proses Pengelasan	Metode	Persamaan	Perbedaan
1	R. Ranjan <i>et al.</i> (2016)	FSW	<i>image pyramid</i> dan <i>image recrontuiction</i>	Menggunakan objek yang sama yaitu citra hasil pengelasan FSW	<ul style="list-style-type: none"> • Mengganti jaringan <i>image pyramid</i> dan <i>image recrontuiction</i> menjadi FCN • Citra diklasifikasi tanpa berdasarakan lalyer bukan <i>pixel</i>
2	S. Sudhagar <i>et al</i> (2019)	FSW	MSER dan SVM	Menggunakan objek yang sama yaitu citra hasil pengelasan FSW	Menggati jaringan MSER dan SVM menjadi FCN.
3	Hartl <i>et al</i> (2020)	FSW	FCNN, CNN, RNN	<ul style="list-style-type: none"> • Menggunakan objek yang sama yaitu citra hasil pengelasan FSW • Menggunakan jaringan FCN untuk klasifikasi 	Jaringan yang digunakan hanya FCN sedangkan jaringan CNN dan RNN tidak digunakan.
4	S Wang <i>et al</i> (2022)	<i>Beam</i>	FCN	Menggunakan Jaringan yang sama yaitu FCN	Menggunakan objek yang berbeda yaitu beam dengan citra hasil pengelasan FSW.
5	Ulya (2022)	FSW	CNN AlexNet dan VGG-16	<ul style="list-style-type: none"> • Objek yang sama yaitu citra hasil pengelasan FSW • Menggunakan klasifikasi yang sama yaitu <i>classification layer</i> 	Mengganti jarigan CNN menjadi FCN

BAB 3 METODOLOGI

3.1 Diagram Alir Penelitian

Tahapan-tahapan yang dilakukan pada penelitian ini secara singkat disajikan dengan diagram alir di gambar 3.1

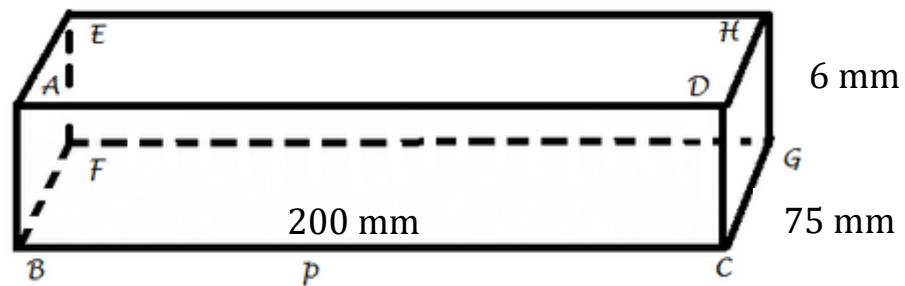




Gambar 3. 1 Diagram alir Penelitian

3.2 Parameter pengelasan FSW

Dalam penelitian ini proses pengelasan *Friction Stir Welding* dimana hal tersebut sudah dijelaskan pada BAB 2 sebelumnya. Parameter penelitian yang didapatkan mengacu pada penelitian sebelumnya milik Pontjono, N. A. pada tahun 2021. Dari penelitian tersebut pengelasan dilakukan pada pelat paduan Aluminium 6061-T651 dengan ketebalan 6 mm, panjang 75 mm dan lebar 100 mm yang ditunjukkan pada Gambar 3.2, serta memiliki komposisi kimia (*chemical properties*) seperti pada Tabel 3.1 berikut.



Gambar 3. 2 Dimensi objek pengelasan

Tabel 3. 1 Komposisi Kimia Pelat A6061-T651

Cu (%)	Mg (%)	Mn (%)	Others (%)	Al (%)
0.15-0.4	0.8-1.2	Max 0.15	Max 0.15	95.8-98.6

Parameter pengelasan FSW yang digunakan adalah sebagai berikut:

1. Jenis *tool* adalah *hexagonal* (setelah proses optimasi).
2. Kecepatan rotasi *tool* (507, 765, 1208, 1907) rpm.
3. *Tilt angle* (2, 2.5, 3, 3.5)°.
4. Sudut bahu cekung (5.5, 4, 2.5, 2)°.
5. Kecepatan pengelasan (24, 43, 65, 90) mm/min.

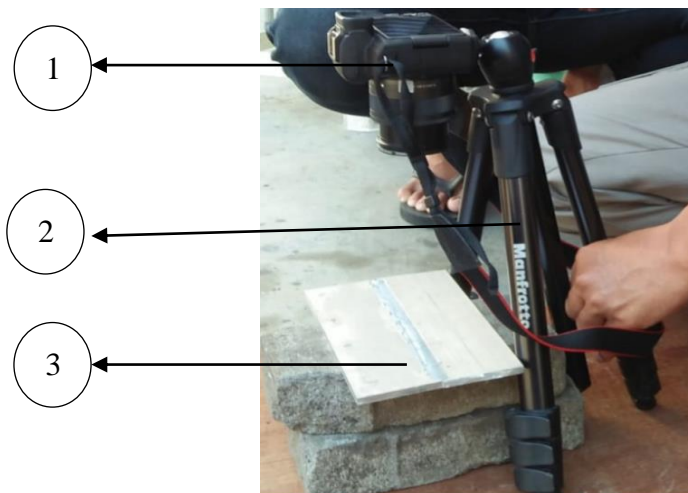
3.3 Pembuatan dan Penyusunan *Dataset Citra Permukaan Las FSW*

Untuk membangun dataset termasuk gambar permukaan yang dianggap normal dan *defect* pada hasil pengelasan FSW 16 gambar dengan resolusi 5472 × 3648

piksel diambil menggunakan kamera digital dalam kondisi pencahayaan dan jarak pengambilan yang telah diatur sama. Untuk mengumpulkan gambar dengan kerusakan kecil, seperti garis-garis kasar (*surface galling*) pada permukaan hasil pengelasan dan terjadiya *overlap*. Semua gambar diambil dengan pengaturan yang telah dibuat. Kebenaran dasar dari gambar yang dikumpulkan secara manual diberi label pada tingkat piksel dengan membedakannya dengan mata manusia gambar dikelompokkan ke dalam sebuah folder berdasarkan kualitas yang diklasifikasikan yaitu normal dan *defect*. Parameter dalam pengambilan citra pengelasan FSW adalah sebagai berikut:

1. Kamera Canon EOS M50.
2. Kamera dipasang pada dudukan.
3. Jarak kamera ke spesimen hasil proses FSW 18 cm.
4. Tidak dilakukan *zoom* dan noise.
5. Menggunakan *frame rate* 24 fps.
6. Pengambilan gambar dilakukan di luar ruangan pada siang hari
7. Kondisi lingkungan dan pencahayaan sama.

Proses pengambilan gambar hasil pengelasan dan pengaturan parameternya seperti yang disajikan pada gambar 3.3



Gambar 3. 3 Pengambilan gambar hasil pengelasan FSW

Keterangan:

1. Kamera Digital

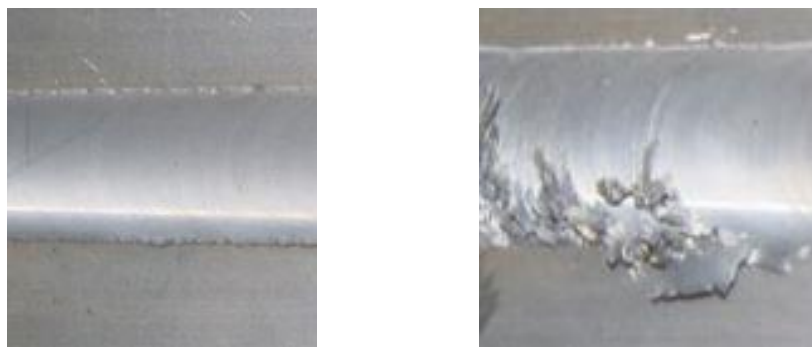
2. Dudukan kamera (Tripod)
3. Aluminium hasil pengelasan

Hasil pengambilan gambar yang digunakan sebagai dataset dapat dilihat pada gambar 3.4



Gambar 3. 4 Hasil pengelasan FSW

Untuk jaringan saraf yang dalam kinerjanya dapat ditingkatkan dengan peningkatan jumlah data pelatihan. Karena itu, untuk memanfaatkan gambar yang dikumpulkan dan mengurangi kemungkinan *overfitting*, maupun *dropout* data augmentasi diimplementasikan pada database kecil. Setelah augmentasi data, jumlah total gambar dalam database digandakan menjadi Gambar untuk kualitas yang akan diklasifikasikan dengan ukuran 224×224 *pixel*. Dari parameter pengelasan yang dilakukan diperoleh 87 citra pengelasan FSW, citra tersebut diolah dengan melakukan pemotongan *cropping* yang dilakukan dengan perangkat lunak *snipping tools* setelah itu dilakukan *resizing* citra pada perangkat matlab. Citra yang telah selesai selanjutnya dikelompokkan secara manual menjadi dua bagian yaitu menjadi citra pengelasan sebagai data set yang terdiri 361 citra *defect* dan 205 citra normal. Hasil *resizing* dapat dilihat pada gambar 3.5



(a)

(b)

Gambar 3. 5 hasil resizing (a) normal (b) cacat atau defect

Untuk menilai kemampuan generalisasi FCN, 87 citra gambar akan diolah dan dibagi menjadi dua dataset kelas yang terdiri dari dua kelas yaitu *defect* sebanyak dan normal. Pada masing-masing dataset terdapat 80 sampel citra. Pada penelitian ini data *training* sebanyak 70% dan data *testing* 15% pada setiap kelasnya sedangkan data *validasi* sebanyak 15%.

Untuk mendapatkan hasil klasifikasi mendekati hasil klasifikasi langsung secara visual oleh manusia atau hasil klasifikasi yang tinggi, maka dibutuhkan data set yang sangat banyak untuk digunakan sebagai data set *input* pada program yang dirancang. Proses *training* yang digunakan pada penelitian ini adalah *feedforward* dan *backpropagation*.

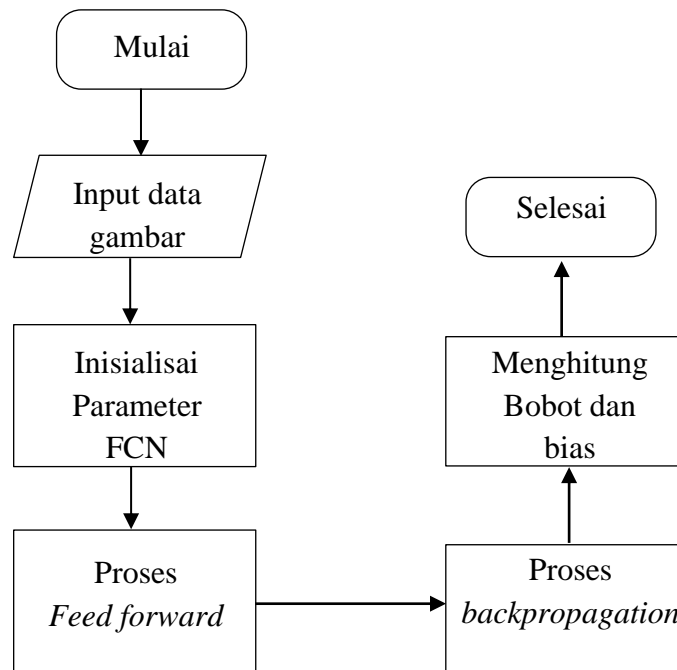
Dari gambar yang telah ada kemudian dilakukan proses *training* data seperti pada gambar 3.6. Proses pelatihan terdiri dari 3 tahapan utama yaitu sebagai berikut:

1. Proses *feedforward*

Proses ini akan menghasilkan beberapa lapisan untuk mengklasifikasi data citra yang mana menggunakan bobot dan bias yang telah diperbarui dari proses *backpropagation*. Tahap ini juga akan digunakan kembali saat proses pengujian.

2. Proses *backpropagation*

Proses ini merupakan tahapan pencarian kesalahan dari lapisan output sampai ke lapisan pertama. Agar dapat memberikan tanda bahwa data yang sebelumnya telah diproses sehingga mendapatkan nilai bobot dan bias. Kemudian dilakukan *training* kembali sehingga didapat nilai bobot dan bias yang baru.



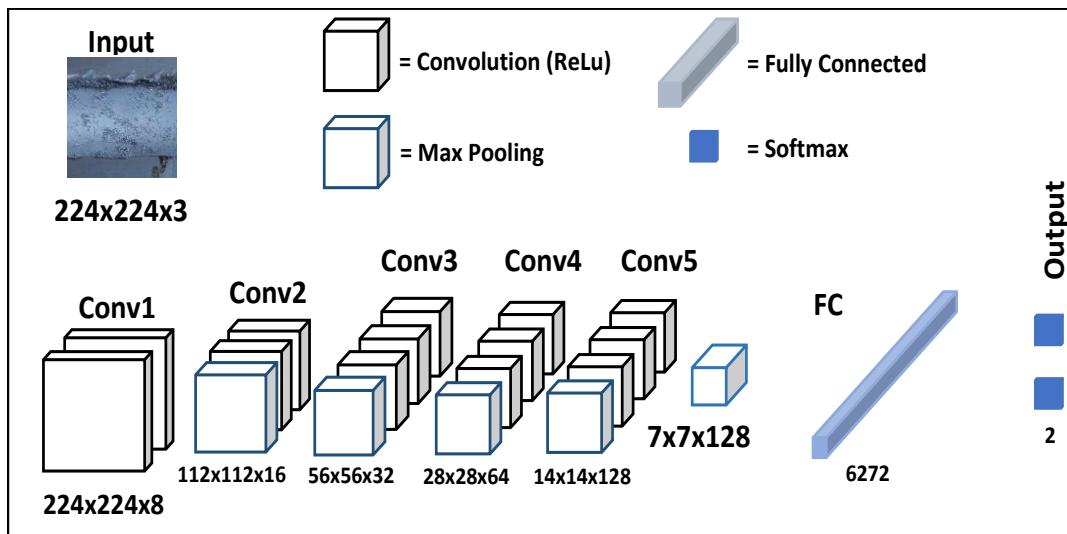
Gambar 3. 6 Alur proses training[21]

3.4 Perancangan jaringan FCN

Untuk mengidentifikasi dan menentukan lokasi beberapa *defect* pada hasil pengelasan yang telah dilakukan, metode FCN dilakukan pemerosesan citra hasil pengelasan kemudian citra hasil pengelasan diekstraksi dari *background* gambar sesuai dengan hasil prediksi yang dilakukan oleh FCN. Secara keseluruhan arsitektur skema metode deteksi berbasis FCN untuk beberapa kerusakan digunakan untuk memprediksi dan mengklasifikasikan citra menjadi kelas yang telah ditentukan kerusakan pada gambar, dan kerusakan diekstraksi sesuai dengan hasil prediksi FCN. Dalam metode akan menggunakan arsitektur FCN dibangun dengan menggunakan 5 *convolutional layer*, 5 *ReLU layer*, 5 *batchnormalization layer*, 5 *maxpool layer* dan 1 *fully connected layer*.

Pengujian merupakan proses klasifikasi dari data yang di-*inputkan* ke dalam program dengan menggunakan nilai bobot dan bias yang telah didapat dari proses *training*. Proses pengujian dilakukan dengan pengambilan data secara acak dari data gambar *input*. Proses pengujian ini sama dengan proses *training* namun proses dilakukan dalam sekali pelatihan sehingga data yang didapat berupa nilai akurasi data dari klasifikasi yang dilakukan.

Pada pengujian data lapisan sudah *fully connected* dengan label yang telah disediakan. Hasil dari *fully connected* tersebut diperoleh nilai *error* pada masing-masing vektor output, pada penelitian ini digunakan hanya 2 vektor output yaitu normal dan *defect*. Untuk klasifikasi jenis cacat yang terjadi tidak dilakukan dalam pengambilan cacat hanya melihat apabila permukaan yang tidak halus maka dikategorikan menjadi *defect*. Arsitektur jaringan yang dirancang disajikan pada gambar 3.7



Gambar 3. 7 Arsitektur Jaringan FCN

3.5 Pelatihan Algoritma FCN

Pelatihan jaringan pada penelitian ini dilakukan dengan menggunakan perangkat lunak MATLAB 2020b. Dengan memperhatikan ketersediaan GPU dan RAM serta spesifikasi dari perangkat keras yang digunakan. Spesifikasi yang lebih tinggi memungkinkan model untuk memuat sampel lebih banyak untuk setiap *batch* dan memangkas waktu pelatihan secara signifikan. Berikut parameter tetap untuk pelatihan yang digunakan disajikan pada tabel 3.2.

Tabel 3. 2 Parameter Pelatihan

<i>Optimizer</i>	<i>Stochastic gradient descent with momentum (Sgdm)</i>
<i>Learning Rate</i>	0,001
<i>Validation Frequency</i>	5
<i>Epoch</i>	50
<i>Batch Size</i>	<i>Default</i>

Semakin besar nilai *learning rate* (α) akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika *learning rate* digunakan terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya jika *learning rate* diset terlalu kecil, maka algoritma akan konvergen dalam waktu yang sangat lama[29]. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Jika harga α semakin besar, maka iterasi yang dipakai semakin sedikit. Hal ini menyebabkan pola yang sudah benar menjadi rusak sehingga pemahaman menjadi lambat. Nilai *learning rate* tidak dapat ditentukan secara pasti sehingga perlu dilakukan *trial and error* untuk mendapatkan nilai *learning rate* yang dapat menghasilkan iterasi tercepat dalam mencapai konvergen[29].

Karena data set hanya terdiri dari dua kelas maka satu kelas dianggap sebagai positif dan kelas lainnya negatif maka matriks evaluasi untuk penelitian ini akan menggunakan *Confusion matrix*. *Confusion matrix* digunakan karena penelitian ini adalah klasifikasi biner (klasifikasi citra hasil pengelasan normal dan *defect*) serta dapat menunjukkan performa klasifikasi model dari nilai akurasi, *recall* dan *precision confusion matrix* seperti pada Tabel 3.1. *Confusion matrix* mempunyai empat jenis kelas berdasarkan hasil prediksi dan observasi. Empat hal tersebut adalah sebagai berikut:

1. *True Positive (TP)*

Kelas dengan hasil prediksi terdapat *defect* dan observasi benar *defect*

2. *True Negative (TN)*

Kelas dengan hasil prediksi normal dan observasi normal.

3. *False Positive (FP)*

Kelas dengan hasil prediksi terdapat *defect* namun observasi normal

4. *False Negative (FN)*

Kelas dengan hasil prediksi normal namun observasi terdapat *defect*. Akurasi model ditentukan dengan persamaan 2.6 – 2.11

Tabel 3. 3 *Confussion Matrix*

Jumlah data (N)	Kelas Prediksi (<i>Predicted Class</i>)	
	Ya (+)	Tidak (-)

Kelas Aktual	Ya (+)	TP	FP
(<i>Actual Class</i>)	Tidak (-)	FN	TN

3.6 Tahapan Image Processing

Tahapan pelatihan citra yang dilakukan pada penelitian ini adalah sebagai berikut:

1. *Image Pre-processing*

Tujuan dari tahapan ini adalah untuk meningkatkan *features* pada data gambar dengan cara menekan distorsi yang tidak diinginkan dan peningkatan beberapa *features* dari gambar yang dianggap penting dengan begitu pemodelan pada *computer vision* dapat menghasilkan data yang ditingkatkan untuk dikerjakan.

2. *Detection of defect*

Deteksi mengacu pada lokalisasi pada objek yang berarti segmentasi gambar dan mengidentifikasi posisi cacat yang dideteksi.

3. *Feature extraction and training*

Ini merupakan langkah yang penting dimana metode statistik atau *deep learning* digunakan untuk mengidentifikasi pola gambar yang paling menarik, fitur unik untuk kelas tertentu dan nantinya dapat digunakan oleh program untuk membedakan antara kelas yang berbeda. Proses diamna model atau program akan mempelajari fitur-fitur yang ada dari kumpulan data yang disebut proses *training*.

4. *Classification of the object*

Langkah ini mengkategorikan objek yang telah dideteksi ke dalam kelas yang telah ditentukan sebelumnya dengan menggunakan teknik klasifikasi yang sesuai dengan membandingkan pola gambar dengan pola target.

3.7 Implementasi

3.7.1 Perangkat lunak

Penelitian ini menggunakan perangkat lunak matlab 2020b dengan aplikasi deep learning toolbox dan image processing and computer vision.

3.7.2 Perangkat keras

Spesifikasi perangkat keras yang digunakan dalam menjalankan program FCN memiliki spesifikasi sebagai berikut:

CPU : Intel(R) Celeron(R) CPU N2840 @ 2.16GHz 2.16
System : 64-bit operating system, x64- based processor
RAM : 2 GB

3.7.3 Pra-pemrosesan data set

Data yang dihasilkan dari proses pemotongan menggunakan perangkat lunak *snippingtool* tidak seragam, maka citra harus diseragamkan menjadi ukuran 224*224. Proses penyeragaman data set dapat dilakukan dengan program matlab menggunakan fungsi batch *image processor* listing yang digunakan adalah sebagai berikut:

```
function results = myimfcn(im)
results = imresize(im,[224 224]);
```

Gambar 3. 8 Fungsi penyeragaman dimensi citra

3.7.4 Arsitektur FCN

Proses pembuatan jaringan diawali dengan pembuatan listing pada program matlab tahapan awal yang dilakukan adalah membuka data set yang telah disiapkan sebelumnya sesuai dengan nama dan lokasi folder data set dibuat pada perangkat keras. Listing yang digunakan adalah sebagai berikut:

```
DatasetPath= './DATASET RESIZE';
Images = imageDatastore(DatasetPath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');
numTrainFiles=0.70;
numValidationFiles=0.15;
numTestImages=0.15;
[TrainImages, ValidationImages,
TestImages]=splitEachLabel(Images,numTrainFiles,numValidationFiles
,numTestImages, 'randomize');
```

Gambar 3. 9 Implementasi membuka dan membagi data set

Dari program yang ada diatas bahwa lokasi folder *data set* yang telah disiapkan brada pada folder yang sama dengan dimana listing program disimpan. Kemudian data dibagi menjadi 2 kelompok oleh program yaitu data citra yang digunakan sebagai citra pelatihan dan data yang digunakan sebagai citra pengujian. Pada *listing* yang dibuat sebanyak 70% dari data citra akan digunakan sebagai pelatihan sedangkan 15% akan digunakan sebagai data uji dan 15% sebagai data validasi oleh program data yang digunakan diambil secara acak dengan jumlah pembagian yang telah ditentukan.

Proses selanjutnya membuat arsitektur jaringan FCN yang akan digunakan pada program akan ditentukan ukuran *input* citra yang digunakan, jumlah lapisan *convolutional* yang akan digunakan dan fungsi aktivasi yang digunakan pada setiap layer *convolutional*. Implementasinya adalah sebagai berikut:

```

layers=[
    imageInputLayer([224 224 3], 'Name', 'Input')

    convolution2dLayer(3,8, 'Padding', 'same', 'Name', 'Conv_1')
    batchNormalizationLayer('Name', 'BN_1')
    reluLayer('Name', 'Relu_1')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'MaxPool_1')

    convolution2dLayer(3,16, 'Padding', 'same', 'Name', 'Conv_2')
    batchNormalizationLayer('Name', 'BN_2')
    reluLayer('Name', 'Relu_2')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'MaxPool_2')

    convolution2dLayer(3,32, 'Padding', 'same', 'Name', 'Conv_3')
    batchNormalizationLayer('Name', 'BN_3')
    reluLayer('Name', 'Relu_3')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'MaxPool_3')

    convolution2dLayer(3,64, 'Padding', 'same', 'Name', 'Conv_4')
    batchNormalizationLayer('Name', 'BN_4')
    reluLayer('Name', 'Relu_4')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'MaxPool_4')

    convolution2dLayer(3,128, 'Padding', 'same', 'Name', 'Conv_5')
    batchNormalizationLayer('Name', 'BN_5')
    reluLayer('Name', 'Relu_5')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'MaxPool_5')

    fullyConnectedLayer(2, 'Name', 'FC')
    softmaxLayer('Name', 'SoftMax');
    classificationLayer('Name', 'Output Classification');
];

```

Gambar 3. 10 Implementasi jaringan FCN

Dari *listing* program diatas dapat diketahui dimensi dari gambar yang digunakan yaitu 224*224 pixel dengan citra warna RGB [224 224 3]. Selain itu jaringan yang digunakan juga terdiri dari 5 lapisan jaringan *convolutional* dengan ukuran filter yaitu *matrix* 3*3 dan jumlah filter yang terus meningkat dengan ukuran 8, 16, 32, 64, 128. Untuk mempercepat pelatihan dan mengurangi sensitivitas terhadap inialisasi jaringan pada jaringan convolusi maka diberikan lapisan *batchNormalizationLayer* diantara lapisan convolusi dan fungsi aktivasi yang digunakan. Pada program digunakan fungsi aktivasi ReLU. Untuk menghasilkan klasifikasi sesuai kelas yang telah ditentukan maka pada jaringan akhir diberikan *fullyConnectedLayer* dengan ukuran 2 karena program nantinya hanya menghasilkan 2 kelas yaitu normal dan *defect*. Output yang digunakan yaitu klasifikasi layer dimana program akan melakukan klasifikasi pada tingkat layer citra yang dimasukkan.

Selanjutnya dilakukan proses pelatihan jaringan yang telah dibuat sebelumnya. Pada pelatihan akan diberikan nilai-nilai untuk program menjalankan perhitungan seperti jumlah *epoch*, perhitungan validasi dan *learning rate* yang digunakan. Hal tersebut akan berpengaruh pada tingkat akurasi dan waktu yang dibutuhkan untuk pelatihan. Implementasinya adalah sebagai berikut:

```
options = trainingOptions('sgdm',...
    'InitialLearnRate',0.001,...
    'MaxEpochs',50,...
    'Shuffle',...
    'every-epoch',...
    'ValidationData',TestImages,...
    'ValidationFrequency',5,...
    'Verbose',false,...
    'Plots',...
    'training-progress');

net = trainNetwork(TrainImages, layers, options);
```

Gambar 3. 11 Implementasi pelatihan jaringan FCN

Dari listing pada gambar 3. Diketahui program akan dilakukan perhitungan dengan menggunakan *solver stochastic gradient descent with momentum* (SGDM) *optimizer* dengan *learning rate* sebesar 0,01. Melakukan perhitungan hingga *epoch* maksimal sebanyak 50 selain itu proses validasi pada data dilakukan setiap 5 kali

proses perhitungan yang dilakukan oleh jaringan. Selanjutnya grafik perhitungan akan ditampilkan oleh program dengan listing net.

Untuk mengetahui apakah program telah sesuai dengan yang diharapkan, maka dilakukan pengujian pada program dengan data set yang telah dibuat yaitu data diluar proses *training* pada program implementasinya adalah sebagai berikut:

```
YPred = classify(net,TestImages);
YValidation = TestImages.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)

%Plotting confusion matrix;
plotconfusion(YValidation,YPred)
```

Gambar 3. 12 Implementasi pengujian jaringan FCN

Dari listing diatas akan didapatkan grafik nilai akurasi dan loss yang dihasilkan dari perhitungan 50 epoch yang telah dikerjakan oleh jaringan FCN. Selain itu pada akhir perhitungan akan dihasilkan *matrix confusion* untuk mengetahui nilai akurasi dan *recall* dari seluruh proses pengujian pada data uji yang digunakan yaitu 25% dari total data yang digunakan.

Kemudian dari jaringan yang telah dirancang akan dimunculkan hasil klasifikasi pada citra yang digunakan sebagai data testing. Data testing diambil dari 25% data secara *random* yang digunakan sebagai *data set* citra. Implementasinya adalah sebagai berikut:

```
% Testing The Image
for i = 1:4
    subplot(2,2,i)
    I = readimage(Images, index(i));
    label = classify(net,I);
    imshow(I)
    title(string(label)+ ", " +
num2str(100*max(Probability(index(i), :)),4) + "%");
end
```

Gambar 3. 13 Implementasi pengujian jaringan FCN untuk citra *testing*

Untuk dapat menggunakan program tanpa pelatihan kembali sehingga dapat menguji dengan cepat maka listing program yang telah dibuat disimpan kedalam sebuah file dengan format matlab data menggunakan perintah *save*. Perintah *save*

yang digunakan akan menyimpan seluruh variabel yang terdapat pada *workspace* dari program FCN baik citra pelatihan yang digunakan. Saat ingin digunakan kembali untuk pengujian citra pengelasan maka digunakan perintah *load*. Perintah *load* akan memuat kembali variabel yang tersimpan melalui perintah *save* sebelumnya. Ketika perintah *load* dieksekusi, semua variabel dalam file akan di-*import* ke *workspace* matlab. Implementasi listing pengujiannya sebagai berikut:

```
% Predicting the type of the cell
clc;
close all;
clear all;

% Selecting and importing image from file;
[filename,pathname]=uigetfile('*.jpg','Select an image');
filewithpath=strcat(pathname,filename);
I=imread(filewithpath);

load net1
% Binarizing;
level = 0.75;
BW = imbinarize(I,level);

% Visualization and prediction;
figure
imshow(I)
label=classify(net,I);
title(['hasil pengelasan ini dikategorikan ' char(label)])
```

Gambar 3. 14 Implementasi *load*

3.8 Evaluasi dan Analisis Hasil VT oleh CNN

Evaluasi merupakan penilaian terhadap performa dari CNN dalam melakukan monitoring untuk mendeteksi cacat *flash* yang terjadi pada hasil proses FSW. Perhitungan nilai performa tersebut menggunakan metode *confusion matrix* dan didapatkan nilai dari parameter kinerja seperti *precision*, *recall*, *F-measure*, serta akurasi. Arsitektur FCN dapat dikatakan memiliki performa yang bagus apabila hasil perhitungan parameter tersebut bernilai mendekati 1 (100%). Setelah dilakukan evaluasi kemudian dilakukan analisis dari keseluruhan hasil yang telah didapatkan mulai dari hasil *training*, validasi, *testing*, serta performa dari FCN yang telah dibangun. Analisis bertujuan untuk melihat bagaimana proses FCN dapat melakukan pendeteksian cacat *flash* pada hasil proses FSW serta melihat apakah

ada kendala ataupun keuntungan selama proses tersebut berlangsung, sehingga dari hal-hal tersebut dapat diambil kesimpulan untuk menentukan langkah selanjutnya.

3.9 Pengambilan Kesimpulan dan Saran

Proses ini merupakan tahap akhir dari penelitian ini. Pada proses ini diambil kesimpulan untuk merangkum seluruh hasil yang telah diperoleh pada penelitian ini sehingga dapat diputuskan langkah untuk kedepannya bagi penelitian ini. Sedangkan saran bertujuan untuk memberikan masukan bagi kelanjutan penelitian ini. Apabila tahap ini terlalui maka bisa disimpulkan penelitian ini sudah selesai.

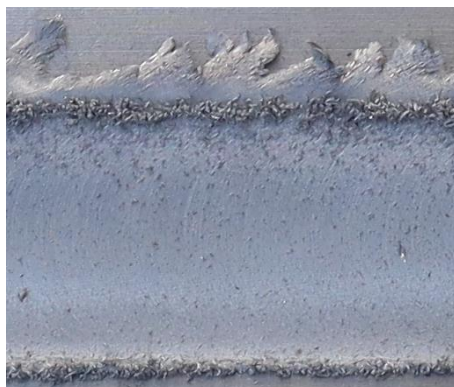
BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Pengolahan dan Klasifikasi Citra

Proses diawali dengan pembuatan *data set* yang dihasilkan dari pemotongan gambar sebagai *region of interest* (ROI) dengan dimensi yang dapat memperlihatkan bagian dari hasil pengelasan yang mengalami *overlap*. *Data set* yang diperoleh dengan melakukan pemotongan pada foto hasil pengelasan dengan menggunakan perangkat lunak *snipping tools*. Data dibagi menjadi dua bagian yaitu data *defect* yaitu citra yang mengalami *overlap* dan citra normal yang ketika diamati tidak terdapat *overlap* pada citra. Untuk memperoleh citra dengan dimensi yang seragam, sehingga dapat dilakukan pengolahan citra untuk mendeteksi hasil pengelasan *defect* atau normal, maka dilakukan proses *resizing* dengan menggunakan perangkat lunak Matlab 2020. Hasil *resizing* menjadi citra dengan dimensi 224×224 pixel. Hasil pengolahan citra sebagai *data set* yang digunakan dapat dilihat pada gambar 4.1



(a)



(b)



(c)

Gambar 4. 1 Hasil pembuatan *data set*

(a)citra hasil pengelasan, (b) citra defect, (c) citra normal

Dari pembuatan *data set* yang dilakukan dengan membagi 87 citra hasil pengelasan menjadi 361 citra *defect* dan 205 citra normal seperti pada gambar 4.1 (b) dan (c).

4.2 Hasil Pelatihan Model

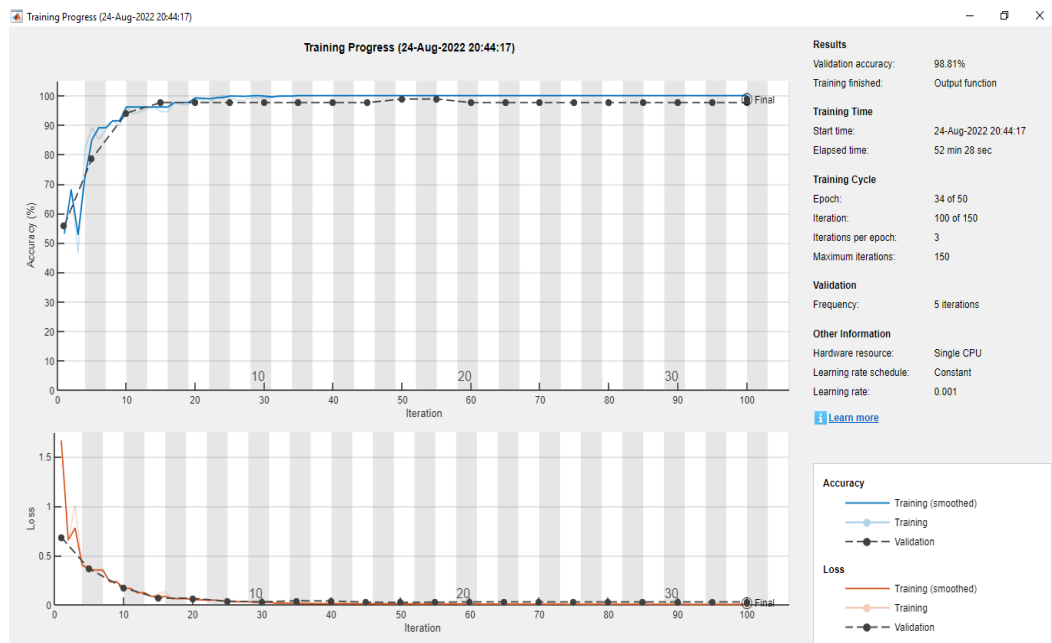
Hasil pelatihan metode FCN yang telah dirancang dan dilatih menghasilkan grafik pelatihan yang disajikan pada gambar 4.2 grafik biru menunjukkan *training accuracy* yang dan grafik merah menunjukkan nilai *Mean Square Error* (MSE) yang terjadi selama proses pelatihan. Dari grafik yang diperoleh dapat diketahui memiliki nilai yang cukup baik dengan nilai secara umum seperti pada matriks *accuracy* dan *F1-score* cukup tinggi. Seperti pada gambar 4.3 pada beberapa sampel mampu memprediksi *defect* yang terjadi pada permukaan las FSW. Dari perhitungan nilai akurasi yang terus naik hal ini dikarenakan program yang terus mempelajari citra yang di-*inputkan* ke program FCN melalui convolusi yang dilakukan. Convolusi akan mengubah citra menjadi matriks angka sehingga menghasilkan *classification map* yang hasilnya terus dibandingkan dengan data yang lain yang dianggap memiliki kecocokan dengan *classification map* yang dihasilkan melalui proses *training*. Dari grafik yang disajikan perhitungan terus naik hingga perhitungan ke 20 hal ini menunjukkan banyaknya perbedaan data yang harus pelajari sehingga saat dibandingkan dengan *classification map* hasilnya belum memiliki kemiripan sehingga *accuracy* kecil.

Sedangkan pada grafik *Mean Square Error* (MSE) atau rata-rata kesalahan yang terjadi selama proses pelatihan dapat diamati *loss function* yang terus menurun. Pengamatan pada *loss function* digunakan untuk mengevaluasi seberapa baik program FCN dalam dalam memodelkan data dan menghasilkan keluaran yang tepat. Evaluasi yang dilakukan yaitu dengan persamaan matematika tertentu. Jika prediksi menghasilkan banyak kesalahan, maka *loss function* akan menghasilkan nilai yang besar. Sedangkan pada program FCN yang dihasilkan grafik *loss function* terus menurun hal tersebut menunjukkan model telah menghasilkan klasifikasi yang baik.

Pada penelitian ini digunakan batasan untuk pelatihan yaitu dibatasi pada jumlah *epoch*. Penelitian ini menggunakan jumlah epoch sebanyak yang dibatasi

yaitu sebanyak 50 kali dan setelah *epoch* selesai, maka proses pelatihan akan selesai. jumlah 50 *epoch* dipilih karena jaringan telah terus mengalami peningkatan performa yaitu pada grafik akurasi selain itu juga tidak mengalami *overfitting* maupun *underfitting* seperti yang disajikan pada grafik *loss*. Untuk mengetahui apakah *epoch* yang telah digunakan telah sesuai, maka dapat dilakukan dengan menggunakan *early stop monitoring* yaitu kriteria pemberhentian proses *training* ketika pada proses pelatihan tidak mengalami kenaikan ataupun penurunan yang signifikan pada grafik yang dihasilkan. Hal tersebut tentu akan mempermudah dan lebih menghemat waktu yang dibutuhkan untuk proses pelatihan. Dalam penelitian ini digunakan *early stop monitoring* sehingga *epoch* yang dihitung sebanyak 34 dari 50. Namun hal ini masih perlu dikembangkan kembali dengan melakukan pengamatan pada *loss* hal ini dengan tujuan agar proses perhitungan dihentikan apabila *loss* naik.

Dari grafik yang disajikan pada gambar 4.3 setelah perhitungan ke 20 kemudian tidak ada peningkatan atau penurunan pada *accuracy* dan *loss* yang signifikan pada perhitungan selanjutnya seperti pada grafik yang ditampilkan gambar 4.2. Evaluasi tingkat layer memperlihatkan kelas setiap layer dalam 1 citra berukuran 224×224 (50.176 pixel). Analisa *training progress*



Gambar 4. 2 Hasil training FCN

Dari grafik yang disajikan pada gambar 4.2 diketahui kinerja yang dilakukan oleh jaringan sangat baik, hal tersebut dengan dapat diamati dengan grafik yang terus menuju normal dan tidak mengalami *overfitting* maupun *underfitting*. Grafik yang baik ini dikarenakan proses konvolusi yang terjadi pada jaringan tidak mengalami penghilangan melalui *dense layer* yang hampir sama sekali tidak ada pada jaringan FCN. Dibandingkan dengan penelitian Ulya (2022)[28] dengan menggunakan data set yang dihasilkan pada kondisi yang sama namun menggunakan jaringan CNN diperoleh bahwa jaringan FCN yang dirancang menghasilkan kurasi yang lebih tinggi apabila dibandingkan dengan AlexNet, namun waktu yang dibutuhkan lebih lama dibandingkan dengan waktu pelatihan CNN AlexNet. Sedangkan perbandingan hasil dengan VGG-16, jaringan FCN yang dirancang memiliki hasil akurasi lebih tinggi dan waktu yang dibutuhkan lebih cepat. Hal ini dikarenakan pada CNN menggunakan *dense layer* sehingga menghasilkan parameter yang lebih banyak apabila dibandingkan dengan FCN. parameter yang lebih banyak akan membuat program membutuhkan waktu yang lebih lama dalam proses pelatihan seperti pada jaringan VGG-16 oleh Ulya (2020). Sedangkan jaringan AlexNet menggunakan 3 *fully connected layer* dengan jumlah masing-masing yang telah ditentukan dibandingkan dengan FCN yang hanya terdiri dari 1 *fully connected layer* akurasi yang dihasilkan lebih tinggi. Selain itu pada jaringan FCN menggunakan *batchnormalize* setelah *convolutional layer* hal ini dengan tujuan untuk untuk menghasilkan jaringan pelatihan menjadi lebih stabil dan menghasilkan akurasi yang lebih tinggi, pada penelitian Asmida Ismail (2019) dengan menggunakan *batchnormalize* nilai akurasi naik sebanyak 2% dibandingkan tanpa menggunakan *batchnormalize* hal ini seperti pada CNN AlexNet dengan FCN yang dirancang.

4.4 Hasil Perhitungan Performa Jaringan FCN

Pada *confusion matrix* pada gambar 4.3 dapat diamati bahwa dari data set yang digunakan untuk proses testing yaitu sebanyak 15% dari seluruh data yaitu sebanyak 54 citra kelas *defect* dan 30 citra normal dapat diketahui program mampu mendeteksi citra menjadi kelas *true positive* (TP) yaitu citra dengan kelas digunakan sebagai *defect* pada kelas sebenarnya dan terdeteksi defect pada hasil

prediksi oleh program FCN. Sedangkan sebanyak 1 citra terdeteksi sebagai kelas *false positive* (FP) yaitu citra normal pada kelas sebenarnya namun 52 terdeteksi sebagai kelas *defect* pada hasil prediksi program FCN. Selanjutnya program dapat mendeteksi 1 data citra sebagai kelas *false negative* (FN) yaitu citra pengelasan *defect* pada kelas sebenarnya, namun terdeteksi sebagai kelas normal pada program FCN. Sebanyak 30 citra dideteksi pada kelas *true negative* (TN) yaitu citra pada kelas sebenarnya normal dan hasil deteksi normal.

		Confusion Matrix		
		DEFECT	NORMAL	
Output Class	DEFECT	52 61.9%	1 1.2%	98.1% 1.9%
	NORMAL	1 1.2%	30 35.7%	96.8% 3.2%
		98.1% 1.9%	96.8% 3.2%	97.6% 2.4%
		DEFECT	NORMAL	
		Target Class		

Gambar 4. 3 *Confusion Matrix*

Dari *Confusion matrix* pada gambar 4.3 diketahui kelas terdapat kelas yang saat di dalam pengujian dengan *data testing* masuk ke kelas *False Positif* (FP) yaitu kelas hasil prediksi masuk ke dalam kriteria *defect* namun dalam kenyataan ataupun evaluasi gambar *input* tersebut masuk dalam kelas normal atau tidak terjadi *flash*. Hal tersebut dapat terjadi karena kesalahan pada jaringan saat membandingkan hasil *classification map* dari data test yang telah diperoleh dengan *classification map* data yang dilatih. Sama seperti kesalahan type I pada statistik yaitu pada saat jaringan telah menentukan terlebih dahulu bahwa citra yang diuji adalah citra dengan kelas normal menjadi hipotesa 0 (H_0) namun saat melakukan pengujian

yaitu membandingkan *classification map* citra *test* dengan *classification map* citra pelatihan terjadi kesalahan pembacaan pada *classification layer*, sehingga citra yang seharusnya masuk ke dalam kelas normal sesuai hipotesa awal oleh program salah diklasifikasikan membuat H_0 ditolak sehingga citra uji masuk menjadi kelas *defect* atau terdapat *flash*.

Selain itu terdapat 1 data sebagai kelas *False Negative* (FN) yaitu kelas dengan hasil klasifikasi masuk ke dalam kelas normal namun pada observasi citra masuk ke dalam kelas *defect* atau terdapat *flash*. Hal ini dapat terjadi karena program gagal menolak H_0 yang bernilai salah. Dari *classification map* yang dibandingkan pada *classification layer* program salah mendeteksi citra yang diuji dianggap sudah benar memiliki kelas normal melalui proses klasifikasi di *classification layer*.

Dengan menggunakan persamaan 2.10 untuk mengetahui nilai performa dari jaringan FCN maka dapat dilakukan perhitungan dengan data testing pada gambar 4.3 sebagai berikut:

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

$$Accuracy = \frac{\text{jumlah deteksi benar (defect + normal)}}{\text{jumlah seluruh data}}$$

$$Accuracy = \frac{52 + 30}{52 + 30 + 1 + 1}$$

$$Accuracy = 0,976 \text{ atau } 97,6 \%$$

Dari perhitungan *Accuracy* yang telah dilakukan didapatkan bahwa model FCN yang dibuat mampu untuk melakukan klasifikasi dengan benar dengan akurat yaitu 97,6%. Karena *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya).

Perhitungan *Precision* (*Positive Predictive Value*) yaitu tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model FCN. Sehingga *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai *precision* dapat diperoleh dengan persamaan (2.7).

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{\text{jumlah citra diprediksi benar defect}}{\text{jumlah citra diprediksi defect}}$$

$$Precision = \frac{52}{52 + 1}$$

$$Precision = 0,968 \text{ atau } 96,8\%$$

Dengan menghitung *precision* dapat diketahui berapa persen *defect* yang diprediksi benar oleh program dari keseluruhan prediksi citra *defect*.

Perhitungan *Recall* atau *Sensitivity* (True Positive Rate)

Recall menggambarkan keberhasilan model FCN dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan persamaan (2.8).

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{\text{jumlah citra dideteksi benar defect}}{\text{jumlah citra defect}}$$

$$Recall = \frac{52}{52 + 1}$$

$$Recall = 0,968 \text{ atau } 96,8\%$$

Dengan menghitung *recall* dapat diketahui berapa persen *defect* yang diprediksi benar oleh program dari jumlah keseluruhan citra *defect* yang digunakan untuk perhitungan.

Nilai dari klasifikasi yang dilakukan oleh program FCN yang telah dibuat hasilnya bukannya nilai atau kelompok yang pasti benar, Tetapi klasifikasi yang dikeluarkan adalah yang paling mendekati nilai sebenarnya. Jika hasil klasifikasi oleh program mendekati kelas yang sesungguhnya, maka dapat dikatakan hasil klasifikasi mempunyai nilai akurasi yang tinggi. Klasifikasi dilakukan secara berulang-ulang untuk memperoleh klasifikasi dengan kelas yang sebenarnya.

Perbedaan antar hasil pengulangan klasifikasi dinamakan presisi. Jika selisih nilai yang didapatkan kecil, maka dapat dikatakan program memiliki nilai presisi yang tinggi.

4.3 Perhitungan Nilai Learning Rate

Untuk mendapat nilai *learning rate* yang paling sesuai, sehingga dapat diketahui akurasi yang paling tinggi dari program yang dirancang. Selain itu penggunaan *learning rate* yang paling sesuai untuk dapat mengetahui waktu yang paling efisien yang paling dibutuhkan oleh program dapat melakukan perhitungan. Dengan jumlah epoch yang sama dan 3 buah *learning rate* yang digunakan adalah 0,1, 0,01 dan 0,001. Hasil yang didapat seperti disajikan pada tabel 4.1.

Tabel 4. 1 perbandingan *learning rate* yang digunakan

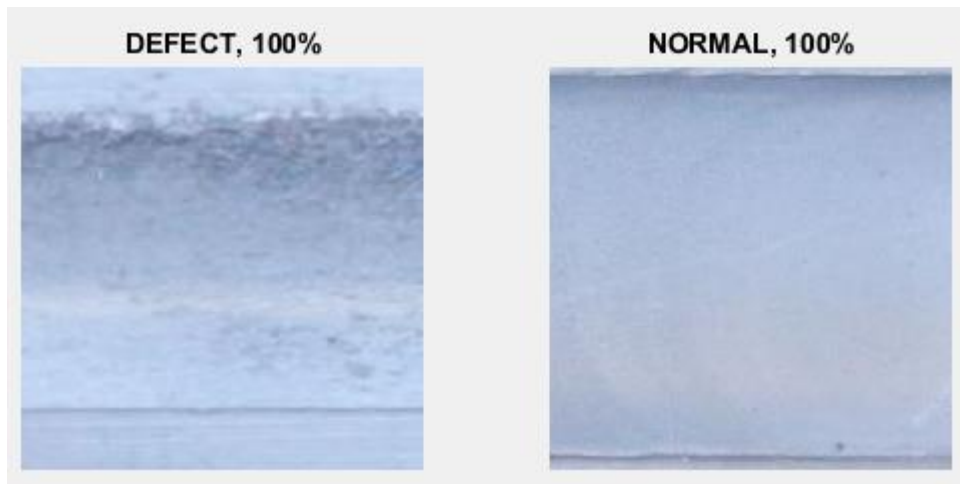
<i>Learning rate</i>	<i>Epoch</i>	<i>Vallidation Accuracy</i>	<i>Time</i>
0,1	22	67,66%	38 min 41 sec
0,01	24	92,86%	38 min 37 sec
0,001	35	98,81%	52 min 28 sec

Dari hasil perhitungan yang telah dikerjakan oleh program FCN pada tabel 4.1 dapat diketahui dengan menggunakan *learning rate* yang berbeda menghasilkan perbedaan pada jumlah *epoch* yang dibutuhkan, nilai *validation accuracy* dan waktu yang dibutuhkan untuk mencapai akurasi tertinggi pada program FCN. Dari tabel yang disajikan dapat diketahi *learning rate* berpengaruh terhadap jumlah *epoch* dan performa yang dikerjakan oleh program FCN hal ini dikarenakan *learning rate* yang semakin kecil akan membuat proses perhitungan yang dilakukan oleh program menjadi semakin teliti. Nilai *learning rate* yang besar akan menyebabkan proses konvergensi dari program menjadi lebih cepat, namun program menjadi tidak stabil hal ini ditunjukkan pada penggunaan *learning rate* 0,1 proses perhitungan cepat namun akurasi yang dihasilkan kecil, sedangkan sebaliknya pada *learning rate* 0,001 dibutuhkan waktu yang lama dan jumlah *epoch* yang lebih banyak dalam proses perhitungan, namun akurasi yang dihasilkan lebih tinggi. Dari hasil perhitungan akurasi dan waktu yang dibutuhkan maka pada

program ini digunakan 0,001 karena menghasilkan tingkat akurasi tertinggi dibandingkan dengan *learning rate* lainnya yang digunakan pada perhitungan.

4.5 Hasil Pengujian Citra

Pengujian (testing) selanjutnya dilakukan oleh program FCN dengan menggunakan dari *data set* yang telah dibagi sebelumnya. Pada penelitian digunakan sebanyak 15% sebagai validasi dan 15% sebagai citra *testing*. Citra pengujian diambil secara *random* dan dilakukan klasifikasi oleh program hasil klasifikasi dari data yang digunakan disajikan pada gambar 4.4.



Gambar 4. 4 Hasil Pengujian validasi Citra

Selain pengujian dengan data validasi dan *testing*, pengujian juga dapat dilakukan dengan menggunakan citra yang telah disiapkan yaitu citra diluar data *training* yang digunakan. Pengujian dilakukan dengan memilih citra dari data set yang telah disiapkan sebelumnya. Pada penelitian ini disiapkan data citra *defect* dan normal yang telah ditentukan kelasnya sebelumnya secara manual dengan penilaian visual. Hasil pengujian beberapa sampel yang digunakan pada penelitian ini seperti disajikan pada gambar 4.5



Gambar 4. 5 Hasil Pengujian Citra *Defect*

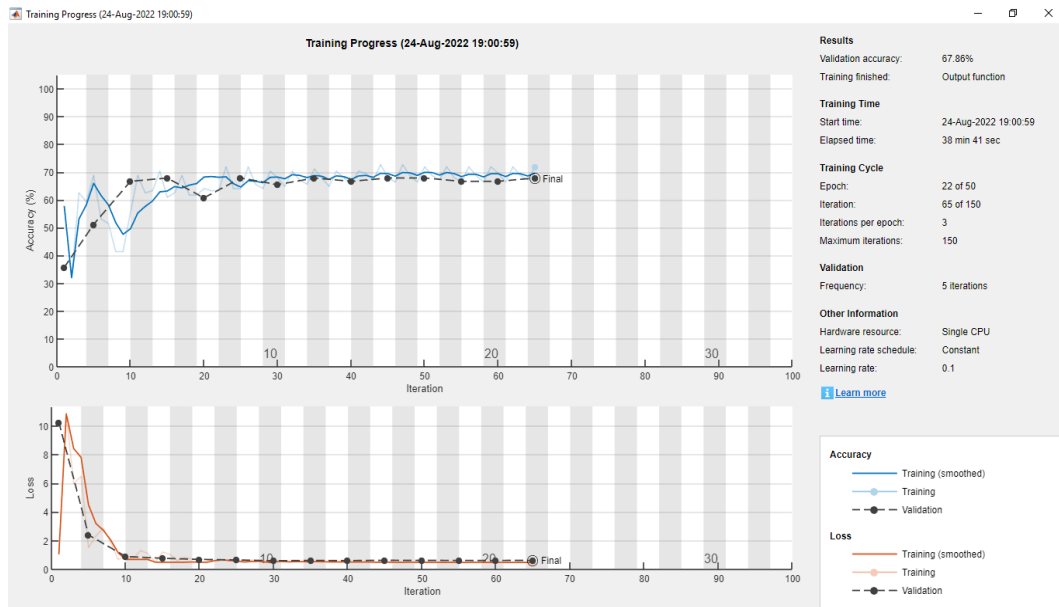
Hasil pengujian citra normal dengan data set yang digunakan diluar data training disajikan pada gambar 4.6.



Gambar 4. 6 Hasil Pengujian Citra Normal

4.6 Pembahasan

Penelitian ini membahas tentang jaringan FCN yang telah dirancang dan uji untuk mengklasifikasikan citra pengelasan yang telah dikelompokkan sebagai citra *defect* dan citra normal. Pengujian dilakukan untuk menghasilkan jaringan FCN yang paling baik dalam menghasilkan akurasi maupun efisien digunakan untuk *visual inspection* (VT). Untuk mendapat jaringan pelatihan yang paling sesuai maka dilakukan beberapa variasi pada *learning rate* yang digunakan yaitu 0,1, 0,01, dan 0,001 hal ini disesuaikan dengan kemampuan dari komputer yang digunakan dalam menjalankan perangkat lunak seperti ketersediaan RAM dan GPU. Penggunaan *learning rate* yang terlalu rendah akan mengakibatkan proses training menjadi lama sedangkan apabila terlalu besar maka akan menghasilkan akurasi yang rendah. Hasil pengujian seperti disajikan pada gambar 4.7

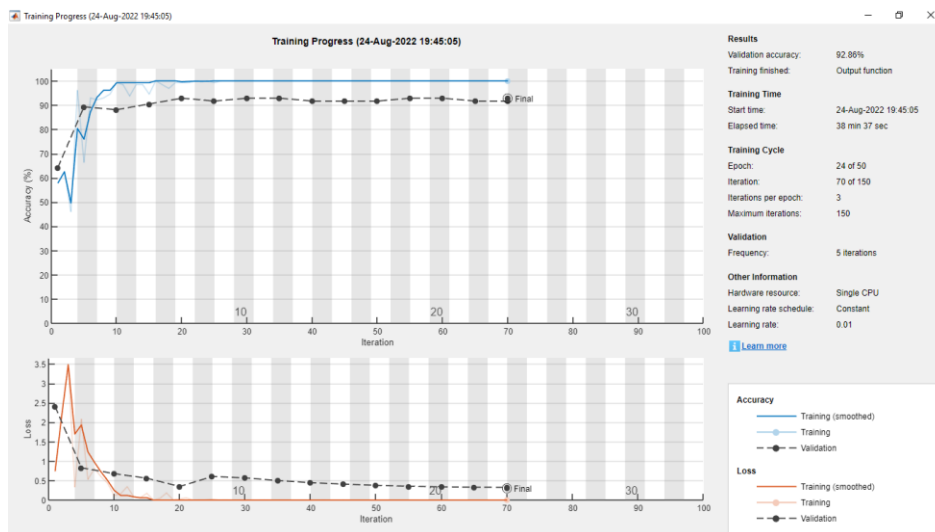


Gambar 4. 7 Pelatihan FCN pada learning rate 0,1

Pada grafik diatas dapat diamati perhitungan akurasi dimulai dari perhitungan pertama dengan akurasi 65% pada validasi awal kemudian grafik terus meningkat hingga perhitungan ke 16 kemudian grafik menjadi normal atau tidak mengalami kenaikan atau penurunan perhitungan akurasi yang signifikan hingga perhitungan seluruhnya sebanyak 22 *epoch* selesai dikerjakan oleh program. Dari perhitungan yang dilakukan program mampu mencapai akurasi hingga 67,6%. Namun dalam pelatihan dibutuhkan waktu mencapai 38 menit 41 detik untuk

menyelesaikan seluruh 22 dari 50 *epoch* yang telah ditentukan. Sedangkan pada pelatihan *loss function* grafik *loss* paling tinggi pada perhitungan 2 mencapai 11% seperti disajikan pada gambar 4.6 Setelah perhitungan ke 10 hasil grafik *loss* menjadi stabil. Pada grafik yang disajikan dapat dilihat pada kurva *loss* terjadi fenomena *unrepresentative train dataset* yaitu pada awal perhitungan yang dikerjakan jaringan. Hal tersebut dikarenakan *dataset* yang digunakan untuk pelatihan tidak memberikan informasi yang cukup untuk dibandingkan dengan data yang digunakan sebagai validasi[30]. Keadaan ini dapat diidentifikasi dengan kurva pada garis *training* dan validasi menunjukkan kesenjangan yang besar diantara kedua garis kurva tersebut hal ini dapat dilihat pada grafik gambar 4.6 yaitu antara nilai validasi pada *loss* kedua dan *training* memiliki jarak yang cukup jauh.

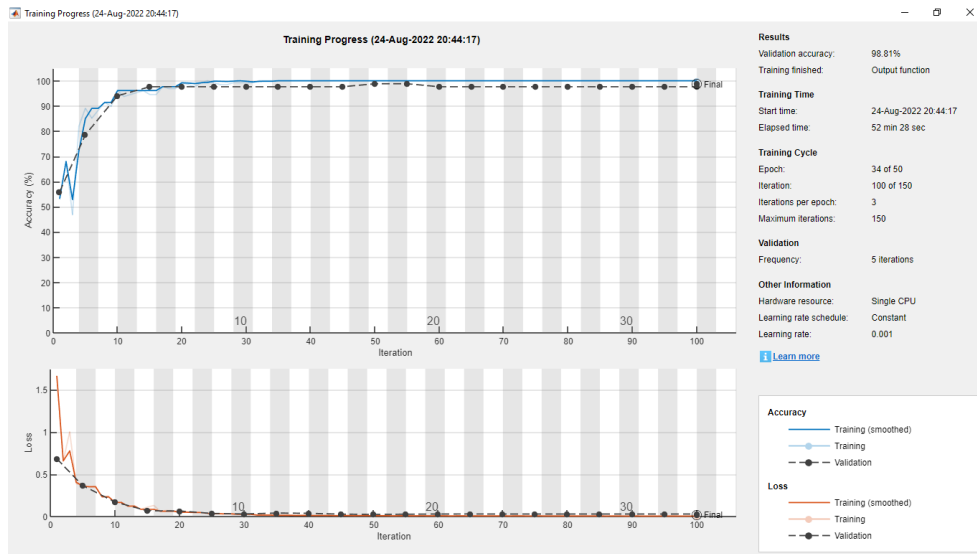
Pada pelatihan *learning rate* 0,01 oleh program seperti disajikan pada gambar 4.8



Gambar 4. 8 Pelatihan FCN pada learning rate 0,01

Dengan pelatihan pada *learning rate* 0,01 diperoleh tingka akurasi yang lebih tinggi dan waktu pengerjaan yang lebih singkat yaitu 38 menit 37 detik. Dari grafik perhitungan yang dihasilkan program dapat mencapai akurasi tertinggi pada perhitungan ke 20 setelah itu hasil perhitungan akurasi dan validasi menjadi stabil. Pada grafik *loss* dihasilkan yang tertinggi pada perhitungan ke 2 yaitu 3,5% dan terus menurun hingga perhitungan ke 15 dan grafik menjadi stabil hingga perhitungan selesai.

Perhitungan pada *learning rate* 0,001 disajikan pada gambar 4.9



Gambar 4. 9 Pelatihan FCN pada *learning rate* 0,001

Dengan *learning rate* 0,001 program FCN yang dirancang mampu mencapai akurasi hingga 98% dan hasil ini merupakan hasil terbaik dibandingkan dengan penggunaan *learning rate* lainnya. Dari grafik yang dihasilkan dapat diamati program mampu mencapai akurasi tertinggi pada perhitungan ke 15 akurasi mencapai nilai tertinggi hingga 100%, setelah itu grafik menjadi stabil dan tidak mengalami kenaikan ataupun penurunan yang signifikan pada *training* dan validasi. Namun pada pelatihan membutuhkan waktu yang lebih lama dibandingkan dengan *learning rate* lainnya yaitu 52 menit 28 detik hal tersebut dikarenakan program akan mempelajari citra yang diinputkan dalam bagian yang lebih kecil, sehingga dibutuhkan waktu yang lama dalam proses pelatihannya. Dari ketiga grafik yang dihasilkan dapat diketahui grafik dengan *learning rate* 0,001 menghasilkan grafik dengan *good fit* hal ini dapat dilihat dengan ciri yang dihasilkan oleh kurva pada gambar 4.9 yaitu plot pada pelatihan *loss* yang terus menurun dan menuju titik stabil. Selain itu plot *loss* validasi juga menuju titik yang stabil dan memiliki *gap* yang sangat kecil dengan plot *loss training*[30]. Dari ketiga pelatihan pada *learning rate* yang berbeda disarankan untuk menggunakan 0,001 karena menghasilkan akurasi yang paling tinggi.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari perancangan jaringan FCN yang telah dikerjakan pada penelitian ini maka didapatkan kesimpulan sebagai berikut:

1. Program FCN yang dirancang melakukan klasifikasi dengan mengubah citra hasil pengelasan menjadi sebuah *classification map* untuk citra dengan kelas *defect* dan normal menggunakan *convolutional layer* yang ada pada jaringan FCN. *Classification map testing* yang dihasilkan akan dibandingkan dengan *Classification map training* yang nilainya paling mendekati klasifikasi yang telah ditentukan maka program akan menghasilkan keluaran berupa klasifikasi kelas dari citra yang di-*inputkan*.
2. Dengan melakukan pengujian pada jaringan FCN yang telah dirancang menggunakan 15% data *testing* dan perhitungan pada *confusion matrix* diperoleh hasil kinerja dari jaringan yaitu akurasi 97,6% nilai presisi 96,8% dan nilai *recall* sebesar 96,8%.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya agar dapat dilakukan analisa dan pengujian kembali pada jaringan FCN yang telah dirancang tentang jumlah *epoch* yang paling sesuai digunakan untuk proses pelatihan jaringan.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Freeman, R., (2003), Friction Stir Welding (FSW), TWI Bulletin, September-October 2003, The Welding Institute (TWI) Inggris.
- [2] Primo, J. (2012). *Welding Inspection Qualifications & Testing Procedures. PDHonline Course M, 415*
- [3] Witama, M. N., & Nazori, A. Z. (2016). *Teknik Pengolahan Citra Dan Adaptive Neuro Fuzzy Inference System (Anfis) Untuk Mendeteksi Kecacatan Pada Accu Kering Di Pt. Nipress Tbk. Telematika MKOM, 7(1), 7-16*
- [4] Cucu, T. R. (2020). Implementasi Adaptive Neural Fuzzy Inference System (ANFIS) Untuk Menguji Kualitas Printed Circuit Board (PCB). *Techno Xplore: Jurnal Ilmu Komputer dan Teknologi Informasi, 5(2), 79-90.*
- [5] Farih, M., Suciati, N., & Navastara, D. A. (2021). Deteksi Kerumunan Menggunakan Metode Fully Convolutional Network pada Kamera Drone. *Jurnal Teknik ITS, 10(2), A339-A344.*
- [6] Wicaksono, M. I. (2020) *Pendeteksian Api Berbasis Pengolahan Citra Digital Menggunakan Metode Convolutional Neural Network*, Skripsi Universitas Gadjah Mada, Yogyakarta.
- [7] Wiryosumarto, H., & Okumura, T. (2000). *Teknologi pengelasan logam, PT. Pradnya Paramita, Jakarta.*
- [8] Ambriz, R. R., Mayagoitia, V., & Ahmad, Z. (2011). Welding of aluminum alloys. *Recent Trends in Processing and Degradation of Aluminium Alloys. Croatia: InTech, 63-86.*
- [9] Fraser, K., St-Georges, L., & Kiss, L. I. (2016). A mesh-free solid-mechanics approach for simulating the friction stir-welding process. *Joining technologies, 27-52.*
- [10] American Society for Metals Handbook Commite, (1990). *Properties and Selection: Nonferrous Alloys and Special-purpose Materials, Volume 09. ASM International. The Materials Information Company.*
- [11] Zettler, R., Vugrin, T., & Schmücker, M. (2010). Effects and defects of friction stir welds. In *Friction Stir Welding* (pp. 245-276). Woodhead Publishing.





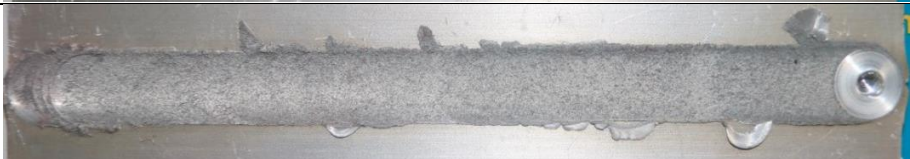

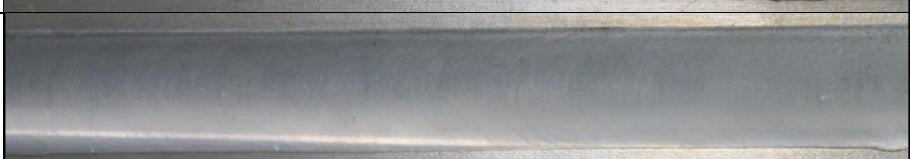


- [12] Handika, Radians Tri. (2016). Pengaruh profil pin dan temperatur preheating terhadap sifat mekanik dan struktur mikro. Fakultas Teknik, Universitas Surakarta
- [13] Zettler, R., Vugrin, T., & Schmücker, M. (2010). Effects and defects of friction stir welds. In *Friction Stir Welding* (pp. 245-276). Woodhead Publishing.
- [14] Tai, L., Ye, H., Ye, Q., & Liu, M. (2017, July). PCA-aided fully convolutional networks for semantic segmentation of multi-channel fMRI. In *2017 18th International Conference on Advanced Robotics (ICAR)* (pp. 124-130). IEEE.
- [15] Yingge, H., Ali, I., & Lee, K. Y. (2020, February). Deep neural networks on chip-A survey. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 589-592). IEEE
- [16] Suyanto. (2018). Machine Learning Tingkat Dasar dan Lanjut. Bandung: Informatika Bandung
- [17] Li, J., Qiu, T., Wen, C., Xie, K., & Wen, F. Q. (2018). Robust face recognition using the deep C2D-CNN model based on decision-level fusion. *Sensors*, 18(7), 2080
- [18] Zahia, S., Sierra-Sosa, D., Garcia-Zapirain, B., & Elmaghraby, A. (2018). Tissue classification and segmentation of pressure injuries using convolutional neural networks. *Computer methods and programs in biomedicine*, 159, 51-58.
- [19] Putra, I. W. S. E. (2016). *Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101* (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [20] Wang, S., Pan, Y., Chen, M., Zhang, Y., & Wu, X. (2020). FCN-SFW: steel structure crack segmentation using a fully convolutional network and structured forests. *IEEE Access*, 8, 214358-214373
- [21] Khumaidi, Agus (2017) *Klasifikasi Image Sequence Hasil Pengelasan Menggunakan Metode Convolution Neural Network (Cnn) Untuk Non Destructive Test*. Master thesis, Institut Teknologi Sepuluh Nopember.
- [22] Triano, N. (2018). Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek. *Universitas Islam Indonesia, Yogyakarta*.

- [23] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.
- [24] Ranjan, R., Khan, A. R., Parikh, C., Jain, R., Mahto, R. P., Pal, S., ... & Chakravarty, D. (2016). *Classification and identification of surface defects in friction stir welding: An image processing approach*. *Journal of Manufacturing Processes*, 22, 237-253
- [25] Sudhagar, S., Sakthivel, M., & Ganeshkumar, P. (2019). *Monitoring of friction stir welding based on vision system coupled with Machine learning algorithm*. *Measurement*, 144, 135-143.
- [26] Hartl, R., Praehofer, B., & Zaeh, M. F. (2020). *Prediction of the surface quality of friction stir welds by the analysis of process data using Artificial Neural Networks*. *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, 234(5), 732-751.
- [27] Zhang, J., Li, Y., Niu, S., Cao, Z., & Wang, X. (2019). *Improved fully convolutional network for digital image region forgery detection*. *CMC-Computers Materials & Continua*, 60(1), 287-303.
- [28] Ulya Ganeswara Alamy, (2022). *Sistem Deteksi Cacat Flash Untuk Kontrol Proses Friction Stir Welding (FSW) Pada Material Aa6061-T651 Dengan Metode Convolutional Neural Network (CNN)*. Master Tesis, Institut Teknologi Sepuluh Nopember.
- [29] Zhang, D., Luo, T., Wang, D., & Liu, R. (2015). Learning from lda using deep neural networks. *Proceedings of International Conference on Computer Processing of Oriental Languages National CCF Conference on Natural Language Processing and Chinese Computing Dongxu Zhang at University of Massachusetts Amherst*, pp. 657-664.
- [30] Anzanello, M. J., & Fogliatto, F. S. (2011). Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, 41(5), 573-583.

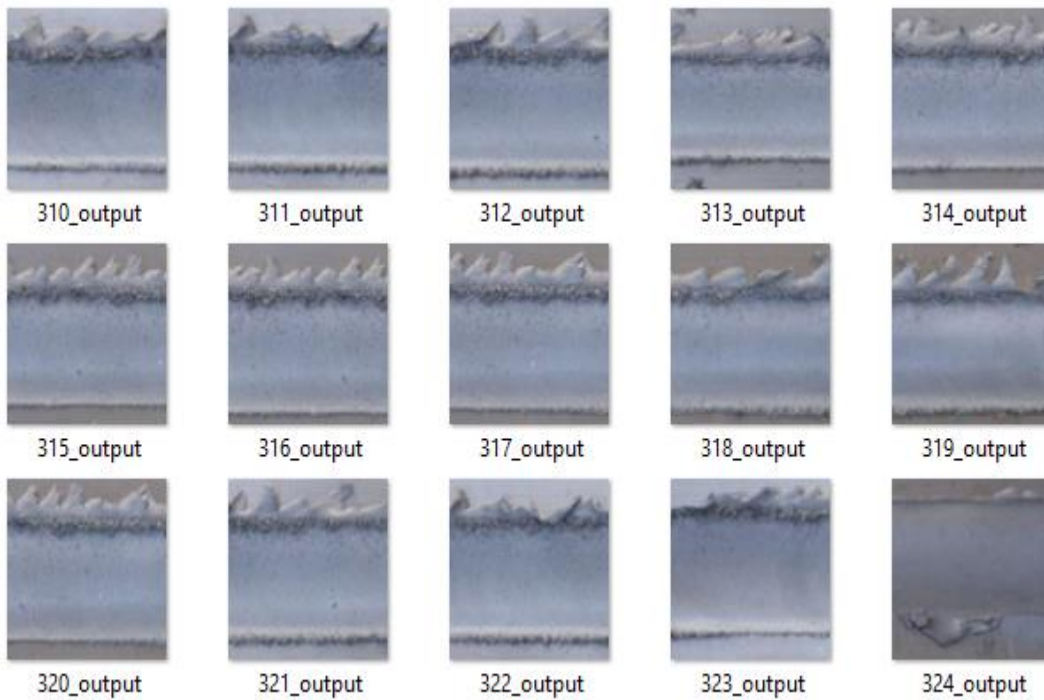
Halaman ini sengaja dikosongkan

LAMPIRAN

Lampiran 1 Citra Hasil Pengelasan

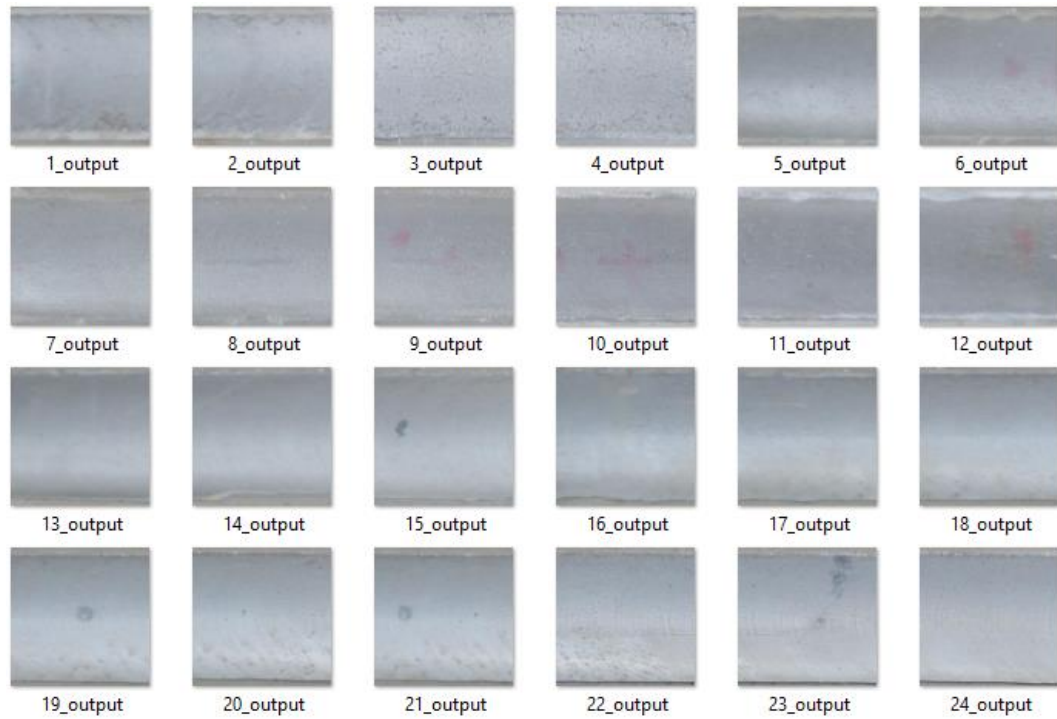
Image No	Image
1	 A horizontal metal rod with a weld bead. The surface shows some irregularities and small pits.
2	 A close-up view of the weld bead, showing a relatively smooth surface.
3	 A close-up view of the weld bead, showing significant surface irregularities and a rough texture.
4	 A horizontal metal rod with a weld bead, showing surface irregularities and a rough texture.
5	 A horizontal metal rod with a weld bead, showing surface irregularities and a rough texture.
6	 A close-up view of the weld bead, showing significant surface irregularities and a rough texture.
7	 A horizontal metal rod with a weld bead, showing a smooth surface.
8	 A horizontal metal rod with a weld bead, showing a smooth surface.
9	 A horizontal metal rod with a weld bead, showing surface irregularities and a rough texture.

Lampiran 1A Citra Hasil Pengelasan *defect*



Dari gambar-gambar yang disajikan diatas akan dibandingkan dengan kriteria pengelasan menurut standart yang ada pada lampiran 2 mengenai *flash* yang terjadi setelah proses pengelasan. Gambar yang ada disajikan diatas menunjukkan citra dengan memiliki *flash* sehingga masuk ke dalam citra dengan kelas defect untuk proses *training*.

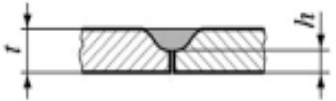


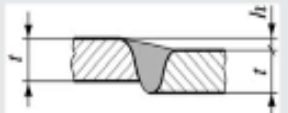
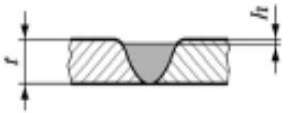
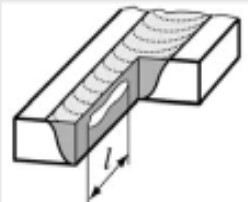
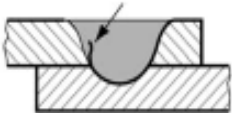
Lampiran 1B Citra Hasil Pengelasan Normal



Dari gambar-gambar yang disajikan digunakan sebagai citra hasil pengelasan normal karena tidak terdapat *flash* pada permukaan hasil pengelasan. Dari hasil tersebut citra akan digunakan sebagai kelas normal pada proses pelatihan untuk dapat mengklasifikasikan citra *test* berdasarkan kualitasnya.

Lampiran 2 Standart Kualitas Pengelasan FSW

Tabel Lampiran 1 Imperfections, testing and examination, acceptance levels according to ISO 25239-5

No.	Name of imperfection	Sketch/Remarks	Acceptance level	Reference to standard 6520-1
1	2	3	4	5
1	Incomplete penetration		Not permitted	421
2	Excess penetration		≤ 3 mm	504
3	Toe flash (flash)		Acceptable size of the imperfection established in accordance with separate regulations or designer's requirements	-
4	Linear misalignment		$h \leq 0,2t$ max. 2 mm	507
5	Underfill		$h \leq 0,1t$ max. 0.5 mm	-
6	Irregular width	Excessive variation in width of the weld	Acceptable size of the imperfection established in accordance with separate regulations or designer's requirements	513
7	Irregular surface	Excessive weld surface roughness	Acceptable size of the imperfection established in accordance with separate regulations or designer's requirements	514
8	Elongated cavity		$l \leq 0,05t$ max. 0,5 mm	2105
9	Hooking		Acceptable size of the imperfection established in accordance with separate regulations or designer's requirements	-

Remarks:

t - nominal thickness of the parent material,

h - height of an imperfection,

l - length of an elongated cavity in the longitudinal direction of the weld

Lampiran 2 B Kriteria Inspeksi Visual berdasarkan AWS

Tabel Lampiran 2 Acceptance criteria according to AWS D17.3

Imperfection	Class A	Class B	Class C
Cracks	Not permitted	Not permitted	Not permitted
Incomplete root penetration (only if complete penetration is required)	Not permitted	Not permitted	Not permitted
Inclusions* (only of presented in a figure)			
a) Single dimension (maximum)	0.33 T or 1.5 mm, whichever dimension is smaller	0.50 T or 2.3 mm, whichever dimension is smaller	Not applicable
b) Distance (minimum)	4 times the dimension of the greater distance from the adjacent imperfection	2 times the dimension of the greater distance from the adjacent imperfection	Not applicable
c) Total length for each 76 mm of a weld [maximum]	1.33 T or 6.1 mm, whichever dimension is smaller	1.33 T or 6.1 mm, whichever dimension is smaller	Not applicable
Internal discontinuities or discontinuities coming up to the surface	Not permitted	Not permitted	Imperfections coming up to the surface are not permitted
Linear misalignment (maximum) One-sided welding	1.05 times material thickness	1.075 times material thickness	Not applicable
Toe flash (flash) Overlapping metal	Overlapping material in the longitudinal direction of the weld should be removed after visual testing but before other non-destructive tests. The method of removing a flash should not reduce the properties of a weld or those of a base metal. Post-welding surface treatment should be carried out in such a manner that the weld and the thickness of a base metal remain within dimensional tolerance in conformity with documentation.		
Angular deformation of joints (maximum) One-sided welding	3°	3°	Not applicable
Underfill (maximum) (applied only if the face of a weld is not subjected to post-weld mechanical treatment)			
a) for the whole length of a weld (maximum depth)	0.05 T	0.075 T	0.10 T
b) single imperfection	0.07 T or 0.76 mm, whichever dimension is smaller	0.10 T or 0.76 mm, whichever dimension is smaller	0.125 T or 0.76 mm, whichever dimension is smaller
c) Total length for each 76 mm of a weld [maximum]	5,1 mm	15 mm	25 mm
Hooking	Imperfection in a welded joint, lacking arrangements related to the boundary values of imperfections		

remarks: T – sheet thickness

Lampiran 3 Listing Program

```
% Image Classification Using FCN

% Importing dataset to MATLAB;
DatasetPath= './DATASET RESIZE';

Images = imageDatastore(DatasetPath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');

% Test-train splitting;
numTrainFiles=0.75;
[TrainImages,
TestImages]=splitEachLabel(Images,numTrainFiles,'randomize');

% Defining layers;
layers=[
    imageInputLayer([224 224 3],'Name','Input')

    convolution2dLayer(3,8,'Padding','same','Name','Conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','Relu_1')
    maxPooling2dLayer(2,'Stride',2,'Name','MaxPool_1')

    convolution2dLayer(3,16,'Padding','same','Name','Conv_2')
    batchNormalizationLayer('Name','BN_2')
    reluLayer('Name','Relu_2')
    maxPooling2dLayer(2,'Stride',2,'Name','MaxPool_2')

    convolution2dLayer(3,32,'Padding','same','Name','Conv_3')
    batchNormalizationLayer('Name','BN_3')
    reluLayer('Name','Relu_3')
    maxPooling2dLayer(2,'Stride',2,'Name','MaxPool_3')

    convolution2dLayer(3,64,'Padding','same','Name','Conv_4')
    batchNormalizationLayer('Name','BN_4')
    reluLayer('Name','Relu_4')
    maxPooling2dLayer(2,'Stride',2,'Name','MaxPool_4')

    convolution2dLayer(3,128,'Padding','same','Name','Conv_5')
    batchNormalizationLayer('Name','BN_5')
    reluLayer('Name','Relu_5')
    maxPooling2dLayer(2,'Stride',2,'Name','MaxPool_5')

    fullyConnectedLayer(2,'Name','FC')
    softmaxLayer('Name','SoftMax');
    classificationLayer('Name','Output Classification');
];

% Defining FCN's options;
options = trainingOptions('sgdm',...
    'InitialLearnRate',0.001,...
    'MaxEpochs',50,...
    'Shuffle',...
    'every-epoch',...
    'ValidationData',TestImages,...
    'ValidationFrequency',5,...
```

```

    'Verbose',false,...
    'Plots',...
    'training-progress');

% Creating FCN;
net = trainNetwork(TrainImages, layers, options);

% Accuracy;
YPred = classify(net, TestImages);
YValidation = TestImages.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)

%Plotting confusion matrix;
plotconfusion(YValidation, YPred)

% Testing The Image
for i = 1:4
    subplot(2,2,i)
    I = readimage(Images, index(i));
    label = classify(net, I);
    imshow(I)
    title(string(label) + ", " +
num2str(100*max(Probability(index(i), :)), 4) + "%");
end

```

Lampiran 3A Listing pengujian dengan data diluar training dan validasi

```

% Predicting the type of the cell

% Selecting and importing image from file;
[filename, pathname]=uigetfile('*.jpg', 'Select an image');
filewithpath=strcat(pathname, filename);
I=imread(filewithpath);

% Binarizing;
level = 0.75;
BW = imbinarize(I, level);

% Visualization and prediction;
figure
imshow(I)
label=classify(net, I);
title(['hasil pengelasan ini dikategorikan ' char(label)])

```

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Abri Andry Saresa Marbun, lahir di Tebing Tinggi, 5 Oktober 1997. Penulis telah menempuh pendidikan di SD N 167699 Kota Tebing Tinggi, SMP Negeri 1 Kota Tebing Tinggi, dan SMA Negeri 3 Kota Tebing Tinggi.

Setelah menyelesaikan pendidikan dasar serta menengahnya, penulis melanjutkan kuliah dan memperoleh gelar Sarjana (S1) di Program Studi Teknik Mesin Universitas Bengkulu.

Pada Tahun 2020 penulis melanjutkan pendidikan Magister (S2) di Departemen Teknik Mesin Fakultas Teknologi Industri dan Rekayasa Sistem (FTIRS) Institut Teknologi Sepuluh Nopember (ITS). Penulis mengambil bidang Rekayasa Produksi. Penulis dapat dihubungi via surel dengan alamat abri.andry.pa123@gmail.com

Halaman ini sengaja dikosongkan