



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

**PENGENALAN AKTIVITAS OLAHRAGA MANUSIA  
PADA CITRA FOTO MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK**

Bagus Prabandaru  
NRP 2210106070

Dosen Pembimbing  
Reza Fuad Rachmadi, ST., MT.  
Christyowidiasmoro, ST., MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 141599**

**HUMAN SPORTS ACTIVITIES RECOGNITION  
IN THE PHOTO IMAGE USING  
CONVOLUTIONAL NEURAL NETWORK**

Bagus Prabandaru  
NRP 2210106070

Supervisor  
Reza Fuad Rachmadi, ST., MT.  
Christyowidiasmoro, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015

**PENGENALAN AKTIVITAS OLAHRAGA MANUSIA  
PADA CITRA FOTO MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK**

**TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Teknik Komputer dan Telematika**

**Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing I**

**Dosen Pembimbing II**



**Reza Fuad Rachmadi, ST., MT.**  
**NIP. 19850403 201212 1 001**

**Christyowidiasmoro, ST., MT.**  
**NIP. 19830127 200912 1 004**

**SURABAYA**  
**JULI, 2015**

# **Pengenalan Aktivitas Olahraga Manusia Pada Citra Foto Menggunakan *Convolutional Neural Network***

Bagus Prabandaru – 2210106070

Reza Fuad Rachmadi, ST., MT. – 19850403 201212 1 001

Christyowidiasmoro, ST., MT. – 19830127 200912 1 004

## **ABSTRAK**

Seorang manusia dapat mengklasifikasikan belasan atau puluhan citra foto aktivitas olahraga manusia dengan akurat dan cepat, karena mata manusia yang diproses oleh otak melalui jaringan saraf memiliki kemampuan yang sangat baik dalam mengenali suatu objek. Tetapi jika berjumlah ratusan atau lebih dari ribuan, maka seorang manusia tidak akan bisa lagi mengklasifikasikannya dengan akurat dan cepat, karena sifat alami manusia yang mudah mengalami kelelahan dan kebosanan. Sehingga dibangunlah sistem *deep learning* dengan metode *convolutional neural network* yang terinspirasi dari kinerja mata dan saraf otak manusia dalam mengenali suatu objek khususnya citra foto. Sistem ini diimplementasikan pada GPU NVIDIA Tesla s2050 dengan melakukan pengujian sebanyak lima kali dan masukan yang digunakan adalah dataset UIUC *sport event image* 8 kelas yang berbeda pada setiap pengujian.

Hasil pengujian dapat mengklasifikasikan citra foto aktivitas olahraga 8 kelas dengan hasil akurasi rata-rata sebesar 71.48% pada pengujian pertama. Pengujian kedua mendapatkan sebesar 68.68%. Pengujian ketiga mendapatkan sebesar 80.37%. Pengujian keempat mendapatkan sebesar 76.17%. Pengujian kelima mendapatkan sebesar 76.68%. Berdasarkan kelima pengujian tersebut disimpulkan bahwa sistem yang dibangun dapat mengenali dan melakukan klasifikasi citra foto aktivitas olahraga 8 kelas dengan hasil kinerja rata-rata 74.67% dan lebih baik daripada Li-Jia Li dan Li Fei-Fei dengan hasil kinerja rata-rata 73.38%.

***Kata Kunci:*** citra foto, convolutional neural network, deep learning, olahraga, pengenalan aktivitas.

# ***Human Sports Activities Recognition in The Photo Image Using Convolutional Neural Network***

Bagus Prabandaru – 2210106070

Reza Fuad Rachmadi, ST., MT. – 19850403 201212 1 001

Christyowidiasmoro, ST., MT. – 19830127 200912 1 004

## ***ABSTRACT***

*A human can classify dozens of photo images of human sports activities accurately and quickly, because the human eye is processed by the brain through the neural network has an excellent ability to recognize an object. But if the number is the hundreds or thousands, then a human will no longer be able to classify accurately and quickly, because human nature is prone to fatigue and boredom. So we will build deep learning system with convolutional neural network method that inspired by the performance of the human eye and brain nerve to recognize an object particularly photograph images. This system is implemented on the GPU NVIDIA Tesla S2050 by testing five times and the input dataset used is UIUC sport event image 8 classes with different images in each test.*

*The test results can classify the photo image of sports activities 8 classes with an average accuracy results is 71.48% in the first test. The second test gain of 68.68%. The third test gain of 80.37%. The fourth test gain of 76.17%. The fifth test gain of 76.68%. Based on the five tests concluded that the system can recognize and classify the photo image of sports activities 8 classes with the average performance results is 74.67% and better than Li-Jia Li and Li Fei-Fei with the average performance results is 73.38%.*

***Keywords:*** *photo image, convolutional neural network, deep learning, sport, activity recognition.*

## KATA PENGANTAR

Puji dan syukur alhamdulillah kehadiran Allah SWT, karena atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “Pengenalan Aktivitas Olahraga Manusia Pada Citra Foto Menggunakan *Convolutional Neural Network*”.

Tugas akhir ini disusun dan diajukan sebagai persyaratan dalam menyelesaikan pendidikan lintas jalur Strata I di Bidang Studi Teknik Komputer dan Telematika, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Pada kesempatan ini penulis menyampaikan terima kasih kepada :

1. Allah SWT berkat rahmat dan hidayah yang telah dikaruniakan kepada penulis sehingga dapat menyelesaikan tugas akhir ini.
2. Nabi Muhammad SAW, sebagai panutan umat muslim di seluruh alam semesta.
3. Kedua orang tua serta keempat adik-adik penulis. Terimakasih atas doa dan dukungan yang telah diberikan kepada penulis sehingga dapat menyelesaikan studi.
4. Bapak Dr. Ir. Tri Arief Sardjono, ST., MT., selaku ketua Jurusan Teknik Elektro ITS.
5. Bapak Reza Fuad Rachmadi, ST., MT. dan bapak Christyowidiasmoro, ST., MT., selaku dosen pembimbing. Terimakasih banyak atas kesabaran dan keikhlasan dalam membimbing penulis.
6. Seluruh Dosen Pengajar, terima kasih banyak atas semua ilmu yang diberikan dan semoga bermanfaat.
7. Teman-teman seperjuangan Jurusan Teknik Elektro khususnya Popy Cahya Harmianto.

Penulis menyadari bahwa masih banyak kekurangan dalam tugas akhir ini. Sehingga kritik dan saran yang membangun dari berbagai pihak sangat penulis harapkan demi kesempurnaan tugas akhir ini.

Surabaya, Juli 2015

Penulis

# DAFTAR ISI

<b>Halaman Judul</b> .....	<b>i</b>
<b>Pernyataan Keaslian Tugas Akhir</b> .....	<b>iii</b>
<b>Halaman Pengesahan</b> .....	<b>v</b>
<b>Abstrak</b> .....	<b>vii</b>
<b>Abstract</b> .....	<b>ix</b>
<b>Kata Pengantar</b> .....	<b>xi</b>
<b>Daftar Isi</b> .....	<b>xiii</b>
<b>Daftar Gambar</b> .....	<b>xv</b>
<b>Daftar Tabel</b> .....	<b>xvii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	4
1.5 Metodologi Penelitian .....	4
1.6 Sistematika Penulisan.....	5
1.7 Relevansi.....	6
<b>BAB II DASAR TEORI</b> .....	<b>7</b>
2.1 Citra Foto .....	7
2.2 <i>Machine Learning</i> .....	9
2.3 <i>Deep Learning – Convolutional Neural Network (CNN)</i> ..	9
2.4 <i>Backpropagation</i> .....	11
2.5 <i>Stochastic Gradient Descent</i> .....	12
2.6 Konvolusi .....	17
2.7 <i>Rectifier Linear Units (ReLU)</i> .....	18
2.8 <i>Pooling</i> .....	18
<b>BAB III PERANCANGAN SISTEM</b> .....	<b>21</b>
3.1 Data Masukan Citra Foto untuk <i>Training</i> dan <i>Testing</i> .....	22
3.2 Proses <i>Convolutional Neural Network (CNN)</i> .....	23
3.2.1 <i>Convolutional Layer</i> .....	23
3.2.2 <i>Rectified linear units (ReLU)</i> .....	24
3.2.3 <i>Pooling Layer</i> .....	24
3.3 Keluaran Hasil Klasifikasi 8 Kelas dengan Nilai Akurasi .	24

3.4	Arsitektur Sistem <i>Convolutional Neural Network</i> (CNN) .	26
<b>BAB IV</b>	<b>PENGUJIAN DAN ANALISA .....</b>	<b>37</b>
4.1	Pengujian Pertama.....	42
4.2	Pengujian Kedua .....	50
4.3	Pengujian Ketiga .....	58
4.4	Pengujian Keempat .....	66
4.5	Pengujian Kelima.....	76
4.6	Seluruh Pengujian .....	84
<b>BAB V</b>	<b>PENUTUP.....</b>	<b>87</b>
5.1	Kesimpulan .....	87
5.2	Saran .....	88
<b>DAFTAR PUSTAKA .....</b>		<b>89</b>
<b>LAMPIRAN.....</b>		<b>91</b>

## DAFTAR TABEL

<b>Tabel 4.1</b> Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan <i>image testing</i> .....	49
<b>Tabel 4.2</b> Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan <i>image testing</i> .....	56
<b>Tabel 4.3</b> Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan <i>image testing</i> .....	64
<b>Tabel 4.4</b> Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan <i>image testing</i> .....	72
<b>Tabel 4.5</b> Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan <i>image testing</i> .....	82
<b>Tabel 4.6</b> Hasil akurasi rata-rata seluruh kelas pada pengujian pertama hingga kelima dengan masukan <i>image testing</i> dalam bentuk prosentase.....	85

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Citra <i>grayscale</i> (abu-abu) (kiri) dan citra berwarna (kanan)[1] .....	8
<b>Gambar 2.2</b> Kolom dan baris membentuk matriks citra foto digital .	8
<b>Gambar 2.3</b> LeNet model <i>convolutional neural network</i> (CNN)[5]..	10
<b>Gambar 2.4</b> Ilustrasi konvolusi pada matriks citra foto .....	13
<b>Gambar 2.5</b> Hasil <i>rectified linear units</i> (ReLUs) pada dataset CIFAR-10[15] .....	18
<b>Gambar 2.6</b> Ilustrasi <i>pooling layers</i> yang dapat berupa <i>average pooling</i> atau <i>max pooling</i> .....	19
<b>Gambar 3.1</b> Blok diagram sistem <i>deep learning</i> metode <i>convolutional neural network</i> (CNN).....	21
<b>Gambar 3.2</b> Dataset UIUC <i>sport event image</i> 8 kelas[1].....	22
<b>Gambar 3.3</b> Desain arsitektur sistem <i>convolutional neural network</i> (CNN).....	25
<b>Gambar 3.4</b> Ilustrasi kernel konvolusi pada <i>convolutional layer</i> pertama .....	27
<b>Gambar 3.5</b> Ilustrasi <i>max pooling</i> pada sebuah <i>image</i> .....	28
<b>Gambar 3.6</b> Ilustrasi proses konvolusi pada <i>convolutional layer</i> .....	29
<b>Gambar 3.7</b> Ilustrasi pad ( <i>zero padding</i> ) pada sebuah <i>image</i> .....	31
<b>Gambar 3.8</b> Ilustrasi proses <i>dropout</i> pada <i>dropout layer</i> [18].....	34
<b>Gambar 4.1</b> Visualisasi <i>feature map</i> dari layer terakhir dengan hasil yang benar .....	37
<b>Gambar 4.2</b> Visualisasi <i>feature map</i> dari layer terakhir dengan hasil yang salah .....	38
<b>Gambar 4.3</b> Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian pertama iterasi ke-60000.....	44
<b>Gambar 4.4</b> Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan <i>image training</i> ..	48
<b>Gambar 4.5</b> Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan <i>image testing</i> ....	50
<b>Gambar 4.6</b> Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian kedua iterasi ke-200000 .....	52
<b>Gambar 4.7</b> Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan <i>image training</i> .....	56

<b>Gambar 4.8</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan <i>image testing</i> .....	58
<b>Gambar 4.9</b>	Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian ketiga iterasi ke-90000.....	60
<b>Gambar 4.10</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan <i>image training</i> .....	63
<b>Gambar 4.11</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan <i>image testing</i> .....	65
<b>Gambar 4.12</b>	Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian keempat iterasi ke-180000.....	67
<b>Gambar 4.13</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan <i>image training</i> .	71
<b>Gambar 4.14</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan <i>image testing</i> ...	73
<b>Gambar 4.15</b>	Log hasil <i>training</i> sistem <i>convolutional neural network</i> (CNN) pada pengujian keempat .....	76
<b>Gambar 4.16</b>	Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian kelima iterasi ke-30000	78
<b>Gambar 4.17</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan <i>image training</i> ....	82
<b>Gambar 4.18</b>	Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan <i>image testing</i> .....	83
<b>Gambar 4.19</b>	Hasil klasifikasi 8 kelas dengan akurasi yang dilakukan pengujiannya oleh Li-Jia Li dan Li Fei-Fei[1] .....	86

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Aktivitas adalah suatu hal yang pasti dilakukan oleh seluruh makhluk hidup khususnya manusia. Salah satu aktivitas yang dilakukan oleh manusia adalah olahraga. Aktivitas olahraga yang dilakukan dapat berupa hal sederhana atau kompleks, dapat pula secara sendiri atau berkelompok. Berlari, berlayar, memanjat tebing, bermain bulutangkis, bermain basket, bermain sepakbola, dan pencak silat adalah beberapa contoh aktivitas olahraga yang dapat dilakukan oleh manusia. Setiap aktivitas olahraga tersebut dapat pula direkam melalui berbagai macam media khususnya media citra foto. Citra foto yang dihasilkan oleh para fotografer tersebut seringkali digunakan sebagai album foto atau dimuat dalam surat kabar dan majalah secara *offline* maupun *online*.

Apabila seorang fotografer menghasilkan citra foto olahraga dari sebuah keluarga yang bertujuan akan dimasukkan ke dalam sebuah album foto keluarga, maka citra foto yang dihasilkan tersebut dapat dengan cepat dan akurat dikenalnya serta dapat langsung diklasifikasikan antara aktivitas olahraga satu dengan yang lainnya. Karena mata manusia yang diproses oleh otak melalui jaringan saraf memiliki kemampuan yang sangat baik dalam mengenali suatu objek sehingga dapat mengenalinya dengan cepat dan akurat. Citra foto keluarga yang dihasilkan oleh fotografer tadi secara umum memiliki jumlah yang sedikit dengan kisaran belasan atau puluhan, sehingga seorang manusia masih mampu melakukan klasifikasi dengan cepat dan akurat. Akan tetapi manusia memiliki kelemahan utama yaitu kebosanan dan kelelahan, dimana kedua hal tersebut sangat berpengaruh terhadap kecepatan dan tingkat akurasi yang dihasilkan.

Apabila pengklasifikasian citra foto aktivitas olahraga manusia yang dilakukan memiliki jumlah yang besar seperti dapat dilihat dalam acara olahraga Asian Games tahun 2014, maka kelemahan utama tadi akan muncul. Dalam acara tersebut terdapat berbagai macam cabang olahraga yang ditampilkan yaitu pencak silat, bulutangkis, berkuda, basket, tenis meja, voli, kayak, gulat dan masih banyak lagi dengan total seluruhnya ada 40 cabang olahraga

yang ditampilkan[19]. Jika satu cabang olahraga menghasilkan ratusan citra foto, maka untuk seluruh cabang yang ditampilkan pada acara Asian Games 2014 tersebut bisa berjumlah ribuan atau bahkan lebih dari itu. Apabila seorang manusia melakukan pengklasifikasian terhadap citra foto tersebut dengan harapan tetap mendapatkan hasil yang akurasi yang tinggi, maka dapat dipastikan akan membutuhkan waktu yang lama karena seorang manusia membutuhkan istirahat dan rekreasi. Manusia memang bukanlah mesin atau robot yang bisa bekerja secara terus menerus tanpa henti.

Kelemahan utama yang terdapat pada manusia tersebut adalah hal yang alamiah karena dalam satu hari secara umum manusia membutuhkan istirahat berupa tidur sekitar 8 jam, makan – minum, mandi, berolahraga, bersenda gurau dan aktivitas yang lain. Karena hal itu salah seorang pemikir utama sosialisme utopis kelahiran Inggris tahun 1771 yang bernama Robert Marcus Owen mencetuskan ide dan melakukan kampanye yang meminta semua pekerja untuk tidak bekerja lebih dari 8 jam per hari. Slogannya yang terkenal adalah "8 jam kerja, 8 jam rekreasi, 8 jam istirahat", tetapi perintis tersebut tidak bisa merasakan hasil yang maksimal dari idenya tersebut sampai ajal merenggutnya di tahun 1858. Selanjutnya pada tahun 1914, sebuah perusahaan yang akhirnya akan dikenal sebagai pelopor dari sistem kerja 8 jam per hari yaitu perusahaan Ford Motor. Perusahaan Ford Motor yang dimiliki oleh Henry Ford menerapkan ide dari Robert Marcus Owen dan menaikkan gaji karyawannya menjadi dua kali lipat. Hal itu mengejutkan para pelaku industri yang lain. Tetapi perusahaan Ford Motor dapat membuktikan bahwa hal itu sangat efektif. Hanya dalam waktu dua tahun, perusahaan itu dapat meningkatkan keuntungannya hingga dua kali lipat dibandingkan dengan tahun-tahun sebelumnya. Sejak saat itu perusahaan-perusahaan lain mengikuti jejak dari Ford Motor dan akhirnya hingga sampai sekarang ide dari Robert Marcus Owen masih diterapkan[20].

Berdasarkan kelemahan utama dari seorang manusia tersebut, sehingga dibutuhkan sistem untuk mengenali citra foto berbagai jenis aktivitas olahraga manusia yang berjumlah ratusan atau lebih dari ribuan. Proses dari sistem yang dilakukan untuk mengenali berbagai jenis aktivitas olahraga manusia pada citra foto tersebut juga harus dapat mengikuti cara kerja mata dan otak manusia yang dapat mengenali sebuah citra foto serta

memprosesnya dengan tingkat akurasi yang tinggi. Serta dapat menutupi kelemahan utama manusia dalam hal kebosanan dan kelelahan, jadi memiliki ketahanan yang melebihi manusia sehingga dapat bekerja dengan waktu yang jauh lebih panjang dan hanya membutuhkan waktu istirahat yang sangat sedikit.

Sistem yang sesuai dengan kemampuan manusia tersebut adalah *deep learning*. Karena *deep learning* merupakan bagian baru dari *machine learning* yang memiliki hubungan lebih dekat dengan tujuan utamanya yaitu *artificial intelligence*. Sedangkan metode dari *deep learning* yang digunakan adalah *convolutional neural network* (CNN). Karena metode *convolutional neural network* (CNN) terinspirasi dari kinerja mata dan saraf otak manusia dalam mengenali suatu objek khususnya citra foto.

## 1.2 Permasalahan

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya. Permasalahan yang akan dibahas adalah sistem manual yang dilakukan oleh seorang manusia dalam mengklasifikasikan citra foto aktivitas olahraga memiliki kelemahan utama jika citra foto tersebut berjumlah ratusan atau lebih dari ribuan. Kelemahan utama tersebut adalah karena kebosanan dan kelelahan yang sangat mudah dialami oleh manusia sehingga proses pengklasifikasian yang dilakukan oleh seorang manusia tanpa henti akan menghasilkan akurasi yang rendah.

## 1.3 Batasan Masalah

Batasan masalah dalam pengerjaan laporan tugas akhir ini adalah sebagai berikut:

1. Data masukan yang digunakan berupa citra foto digital dengan format jpeg.
2. Citra foto yang digunakan adalah *sport event dataset* 8 kelas dari <http://vision.stanford.edu> (UIUC *Sport*).
3. Metode *deep learning* yang digunakan adalah *convolutional neural network* (CNN).
4. Parameter yang digunakan pada sistem *convolutional neural network* (CNN) didapatkan dari <http://caffe.berkeleyvision.org>.
5. Implementasi sistem menggunakan GPU NVIDIA Tesla s2050.

## 1.4 Tujuan

Tujuan dari laporan tugas akhir ini adalah membangun sistem menggunakan *deep learning* dengan metode *convolutional neural network* (CNN) untuk mendapatkan hasil klasifikasi berupa kelas yang berjumlah 8 jenis aktivitas olahraga manusia pada citra foto dan tingkat akurasi pengklasifikasian dari masing-masing kelas tersebut serta mendapatkan hasil akurasi rata-rata lebih baik daripada sebelumnya yang dilakukan oleh Li-Jia Li dan Li Fei-Fei.

## 1.5 Metodologi Penelitian

Langkah-langkah yang ditempuh dalam menyelesaikan laporan tugas akhir ini adalah sebagai berikut:

1. Pengumpulan literatur  
Pengumpulan pustaka untuk dipelajari tentang dasar-dasar *deep learning* dan metode *convolutional neural network* (CNN).
2. Pengumpulan *dataset* aktivitas manusia  
Mengumpulkan *dataset* aktivitas olahraga manusia berupa citra foto, digunakan sebagai masukan dari sistem yang akan dibuat.
3. Penentuan ukuran citra foto dan jumlah *convolutional layer* serta *pooling layer*  
Menentukan ukuran resolusi citra foto yang akan dijadikan masukan dalam satuan *pixel* dan jumlah *convolutional layer* serta *pooling layer* yang digunakan dalam proses pengenalan citra foto.
4. Penentuan model neuron dalam implementasi *convolutional neural network* (CNN)  
Menentukan model neuron yang digunakan dalam implementasi *convolutional neural network* (CNN). Model neuron untuk aktivasi yang dimaksudkan seperti *traditional neuron models nonlinearity*, *rectified linear units* (ReLU), atau yang lainnya.
5. Melakukan *training* menggunakan GPU  
Mengimplementasikan sistem yang dibuat dengan terlebih dahulu melakukan *training* menggunakan GPU NVIDIA Tesla s2050.

6. Melakukan analisa dan pengujian  
Melakukan analisa dengan membandingkan hasil pengujian yang telah mendapatkan pengklasifikasian 8 kelas dengan tingkat akurasi pada masing-masing kelas tersebut.
7. Penulisan Laporan  
Penulisan laporan tugas akhir dilakukan sejak awal pengumpulan literatur hingga sistem yang dibuat mendapatkan hasil pengujian dan dianalisa. Selanjutnya laporan tersebut disusun menjadi buku tugas akhir.

## 1.6 Sistematika Penulisan

Sistematika penulisan dalam laporan tugas akhir ini terbagi atas lima bab, dimana setiap bab saling berhubungan satu sama lain sesuai dengan urutan permasalahan yang akan dibahas. Selain itu juga disertai lampiran sebagai bahan pendukung. Masing-masing bab membahas hal-hal sebagai berikut:

Bab I : Pendahuluan

Pendahuluan mendeskripsikan tentang latar belakang tugas akhir, permasalahan, batasan masalah, tujuan, metodologi penelitian, sistematika penulisan, dan relevansi atau manfaat.

Bab II : Dasar Teori

Dasar teori berisi tentang teori-teori dasar sebagai penunjang terhadap penelitian yang dilakukan. Dalam hal ini berupa konsep dasar citra foto, konsep *deep learning*, metode *convolutional neural network* (CNN), konsep dasar dari konvolusi, *backpropagation* atau *gradien* dan *rectified linear units* (ReLU) serta *pooling*.

Bab III : Perancangan Sistem

Perancangan sistem menguraikan tentang pembuatan sistem dalam bentuk blok diagram yang akan disertai dengan keterangannya.

Bab IV : Pengujian dan Analisa

Pengujian dan analisa menjelaskan tentang hasil pengujian dari sistem yang telah dibuat serta analisa dari hasil pengujian dari sistem tersebut.

Bab V : Penutup

Penutup terdiri dari kesimpulan dan saran. Kesimpulan adalah hal-hal mengenai seluruh penelitian tugas akhir yang telah dilakukan. Sedangkan saran adalah hal-hal yang dapat digunakan untuk pengembangan sistem ini lebih lanjut.

## **1.7 Relevansi**

Hasil dari penelitian tugas akhir ini diharapkan, sistem yang dibangun dapat membantu pengklasifikasian dari pengenalan citra foto aktivitas olahraga manusia dengan tetap memperhatikan tingkat akurasi yang diperoleh dari sistem tersebut. Serta sebagai alternatif dari sistem pengklasifikasian citra foto yang menggunakan metode lain.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Citra Foto**

Citra foto merupakan tampilan dua dimensi yang menggambarkan suatu visualisasi objek dan dihasilkan dengan menggunakan sensor kamera. Dimana objek yang diambil dapat berupa pemandangan, benda, manusia atau yang lainnya. Citra foto dapat diwujudkan dalam bentuk tercetak ataupun digital. Citra digital merupakan matriks dimana indeks baris – kolomnya menyatakan suatu titik pada citra tersebut yang disebut pixel.

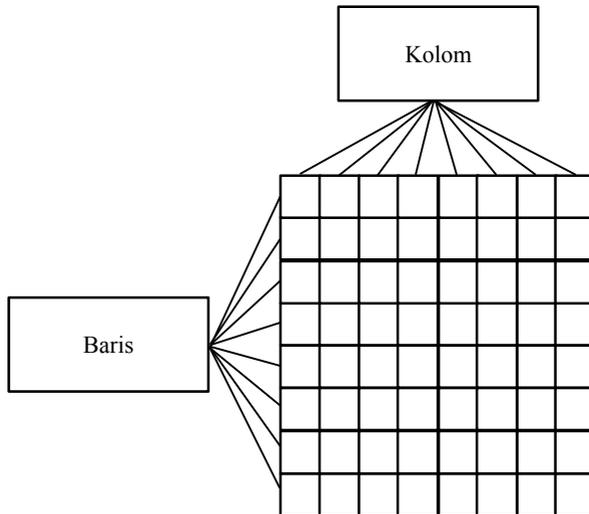
Citra foto digital yang berupa citra grayscale (abu-abu) adalah citra satu kanal  $f(x,y)$  yang merupakan fungsi tingkat keabuan dari hitam ke putih. maka nilai pixel-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Pada umumnya citra grayscale (abu-abu) memiliki kedalaman pixel 8 bit, tetapi ada juga citra grayscale (abu-abu) yang kedalaman pixel-nya 16 bit untuk penggunaan yang memerlukan ketelitian tinggi.

Citra foto digital yang berupa citra berwarna adalah citra beberapa kanal yang menyatakan komponen-komponen warna penyusunnya. Dimana setiap pixel pada citra berwarna merupakan kombinasi dari tiga intensitas warna dasar; derajat keabuan merah  $f_{merah}(x,y)$ , derajat keabuan hijau  $f_{hijau}(x,y)$ , dan derajat keabuan biru  $f_{biru}(x,y)$  yang merupakan fungsi tingkat keabuan dari hitam ke putih. Pada umumnya citra berwarna memiliki kedalaman pixel 24 bit, tetapi ada juga citra berwarna yang kedalaman pixel -nya 32 bit. Sehingga persepsi visual citra berwarna lebih kaya dibandingkan dengan citra grayscale (abu-abu) karena citra berwarna dapat menampilkan objek seperti warna aslinya.

Gambar 2.1 pada sebelah kiri adalah contoh citra foto digital yang berupa citra grayscale (abu-abu) dan sebelah kanan adalah contoh citra foto digital yang berupa citra berwarna. Sedangkan gambar 2.2 menjelaskan tentang matriks citra foto digital yang terdiri baris dan kolom, dimana indeks baris – kolomnya merupakan pixel dengan  $f(x,y)$ .



**Gambar 2.1** Citra grayscale (abu-abu) (kiri) dan citra berwarna (kanan)[1]



**Gambar 2.2** Kolom dan baris membentuk matriks citra foto digital

## 2.2 Machine Learning

*Machine learning* merupakan bagian dari *computer vision* (CS) dan *artificial intelligence* (AI) yang berhubungan dengan konstruksi atau pembangunan serta studi sistem yang dapat belajar dari data. Pada tahun 1959, Arthur Samuel mendefinisikan bahwa *machine learning* adalah disiplin ilmu yang memberikan kemampuan pada suatu sistem untuk belajar tanpa diprogram secara eksplisit[7]. Kemampuan belajar yang menjadi dominan, ditentukan oleh kemampuan perangkat lunak atau algoritmanya. Implementasi kemampuan belajar dapat dicapai dengan berbagai teknik, ada yang menggunakan *rule*, ada yang menggunakan statistika, ada yang menggunakan pendekatan fisiologi yaitu sistem saraf manusia yang disebut dengan *artificial neural network* (ANN) atau jaringan saraf tiruan (JST). *Machine learning* dapat berfungsi untuk beradaptasi dengan suatu keadaan yang baru, serta untuk mendeteksi dan memperkirakan suatu pola.

## 2.3 Deep Learning – Convolutional Neural Network (CNN)

*Deep learning* adalah sekumpulan algoritma dalam *machine learning* yang mencoba untuk memodelkan abstraksi tingkat tinggi dengan menggunakan arsitektur model yang terdiri dari beberapa transformasi *non-linear*[8].

Salah satu arsitektur atau metode yang umum digunakan untuk *deep learning* adalah *convolutional neural network* (CNN).

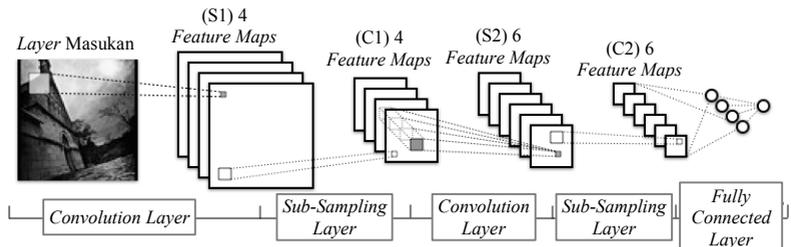
*Convolutional neural network* (CNN) adalah jenis *artificial neural network* (ANN) *feed-forward* dimana masing-masing neuron ditata sedemikian rupa untuk menanggapi daerah *overlapping* di bidang visual. CNN terinspirasi oleh proses biologi dan variasi *perceptrons multilayer* yang dirancang untuk menggunakan *preprocessing* dengan jumlah yang minimal. Proses biologi yang dimaksud adalah dari karya awal Hubel dan Wiesel pada visual korteks kucing, dimana terdapat susunan sel-sel yang kompleks dalam visual korteks. Sel-sel ini sensitif terhadap sub-daerah kecil dari suatu ruang masukan, disebut *receptive field*, dan tertata sedemikian rupa untuk menutupi seluruh bidang visual[5].

Selain itu, telah diidentifikasi terdapat dua jenis sel dasar: sel sederhana (S) dan sel kompleks (C). Sel sederhana (S) menanggapi secara maksimal untuk pola stimulus tepi tertentu dalam *receptive field* mereka. Sel kompleks (C) memiliki *receptive*

*field* lebih besar dan secara lokal invarian untuk posisi yang tepat dari stimulus[5].

Ketika digunakan untuk pengenalan citra foto, *convolutional neural network* (CNN) terdiri dari beberapa *layer* neuron kecil yang melihat bagian-bagian kecil dari citra foto masukan, disebut *receptive field*. Hasil ini kemudian ditata sehingga mereka tumpang tindih untuk mendapatkan representasi yang lebih baik dari citra foto asli dan ini diulang untuk setiap *layer* tersebut. Karena itu, mereka mampu mentolerir terjemahan dari citra foto masukan. *Convolutional neural network* (CNN) bisa termasuk *layer pooling* lokal atau global, yang menggabungkan keluaran dari kluster neuron. Mereka juga terdiri dari berbagai kombinasi *layer convolutional* dan *layer fully connected*, dengan *pointwise nonlinearity* diterapkan pada akhir atau setelah setiap *layer*. Salah satu keunggulan utama dari *convolutional neural network* (CNN) adalah penggunaan *weight* bersama dalam *layer convolutional*, yang berarti bahwa filter yang sama (*weights bank*) digunakan untuk setiap *pixel* dalam *layer*, sehingga memiliki efek positif dalam mengurangi ukuran memori yang diperlukan dan meningkatkan kinerja[6]. Informasi lebih lengkap tentang CNN yang digunakan dapat dilihat pada referensi[2], [3], [4].

Gambar 2.3 merupakan salah satu contoh sistem model dari *convolutional neural network* (CNN). *Layer* masukan berupa citra foto yang dikonvolusikan pada *convolution layer* lalu diproses dalam *sub-sampling layer* untuk mendapatkan 4 *feature maps* dan 6 *feature maps*. Hasil yang telah didapatkan tadi lalu masuk ke *fully connected layer* hingga menghasilkan keluaran yang diharapkan.



**Gambar 2.3** LeNet model *convolutional neural network* (CNN)[5]

## 2.4 Backpropagation

Salah satu teknik dari jaringan saraf tiruan atau *artificial neural network* (ANN) yang dapat diaplikasikan dengan baik dalam bidang peramalan atau *forecasting* adalah *backpropagation*. *Backpropagation* melatih jaringan atau *network* untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama *training*[9].

*Training* yang dilakukan dengan *backpropagation* sama halnya seperti *training* pada *neural network* (NN) yang lain. Pada jaringan *feedforward* (umpan maju), *training* dilakukan dalam rangka perhitungan bobot sehingga pada akhir *training* akan diperoleh bobot-bobot yang baik. Jadi selama proses *training*, bobot-bobot diatur secara iterasi untuk meminimalkan *error* (kesalahan) yang terjadi. Kesalahan dihitung berdasarkan rata-rata kuadrat kesalahan atau *mean square error* (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan kerja fungsi aktivasi[10].

Sebagian besar *training* untuk jaringan *feedforward* (umpan maju) menggunakan gradien dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimalkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut *backpropagation*. Dan pada dasarnya, algoritma *training* standar *backpropagation* akan menggerakkan bobot dengan arah gradien negatif. Prinsip dasar dari algoritma *backpropagation* adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat[11].

Fase *training* dengan *backpropagation* berjumlah 3 buah[9]:

1. Fase 1; propagasi maju.

Dalam propagasi maju, setiap sinyal masukan dipropagasi (dihitung maju) ke *hidden layer* hingga *layer* keluaran dengan menggunakan fungsi aktivasi yang ditentukan.

2. Fase 2; propagasi mundur.

Kesalahan (selisih antara keluaran jaringan dengan target yang diinginkan) yang terjadi dipropagasi mundur mulai dari garis yang berhubungan langsung dengan unit-unit di *layer* keluaran.

3. Fase 3; perubahan bobot.

Pada fase ini dilakukan modifikasi bobot untuk menurunkan

kesalahan yang terjadi. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

## 2.5 Stochastic Gradient Descent

*Gradient descent* (GD) adalah algoritma optimasi orde pertama yang menggunakan seluruh data *training* untuk mengubah parameternya pada iterasi tertentu, sedangkan *stochastic gradient descent* (SGD) merupakan metode optimasi *gradient descent* (GD) yang menggunakan sebagian data *training* (berdasarkan *batch size* yang digunakan) untuk masalah optimasi *unconstrained* dengan meminimalkan *objective function*[12].

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \dots\dots\dots(1)$$

$$W_{t+1} = W_t + V_{t+1} \dots\dots\dots(2)$$

*Stochastic gradient descent* (SGD) mengubah nilai bobot **W** dengan kombinasi linier dari gradien negatif  $\nabla L(\mathbf{W})$  dan perubahan nilai bobot sebelumnya **V<sub>t</sub>**. *Learning rate*  $\alpha$  adalah nilai bobot gradien negatif dan *momentum*  $\mu$  adalah perubahan nilai bobot sebelumnya.

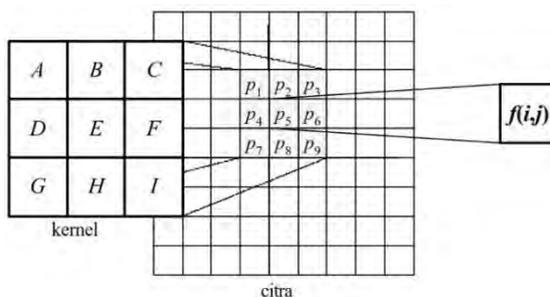
Persamaan 1 dan 2 adalah formula untuk menghitung perubahan nilai **V<sub>t+1</sub>** dan perubahan nilai bobot **W<sub>t+1</sub>** pada iterasi **t+1**, dengan nilai bobot sebelumnya **V<sub>t</sub>** dan nilai bobot sekarang **W<sub>t</sub>**. Pembelajaran “*hyperparameter*” ( $\alpha$  dan  $\mu$ ) memerlukan konfigurasi untuk hasil yang terbaik[12].

## 2.6 Konvolusi

Konvolusi pada pengolahan citra foto merupakan salah satu metode untuk mendapatkan ekstraksi fitur dari sebuah *image*[13]. Proses konvolusi adalah mengalikan sebuah citra foto dengan sebuah *convolution filter*, *convolution mask*, *convolution kernel*, atau disebut juga dengan *template*. Dalam ranah diskrit *kernel* konvolusi dinyatakan dalam bentuk matriks (umumnya 3 x 3, namun ada juga yang berukuran 2 x 2 atau 2 x 1 atau 1 x 2). Ukuran matriks ini biasanya lebih kecil dari ukuran citra foto. Setiap elemen matriks disebut koefisien konvolusi. Operasi konvolusi dilakukan dengan menggeser *kernel* konvolusi *pixel* per *pixel*. Dan hasil dari operasi konvolusi tersebut disimpan di dalam matriks yang baru.

Gambar 2.4 merupakan ilustrasi konvolusi pada matriks citra foto. *Kernel* konvolusi berukuran  $3 \times 3$  dan citra foto berukuran  $9 \times 9$ . Lalu dilakukan konvolusi menggunakan *kernel* konvolusi yang bernilai A, B, C, D, E, F, G, H, dan I pada citra foto dengan area *pixel*  $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8,$  dan  $p_9$ . Sehingga menghasilkan matriks citra foto yang baru yaitu  $f(i,j)$ . Formula yang digunakan untuk mendapatkan  $f(i,j)$  adalah:

$$f(i,j) = A p_1 + B p_2 + C p_3 + D p_4 + E p_5 + F p_6 + G p_7 + H p_8 + I p_9$$



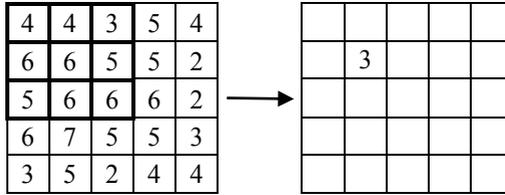
**Gambar 2.4** Ilustrasi konvolusi pada matriks citra foto

Contoh lebih detail, misal sebuah citra  $f(x, y)$  yang berukuran  $5 \times 5$  *pixel* dan sebuah *kernel* atau *mask* yang berukuran  $3 \times 3$  *pixel* masing-masing adalah sebagai berikut:

$$f(x, y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix} \quad g(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Operasi konvolusi antara citra  $f(x, y)$  dengan *kernel*  $g(x, y)$ ,  $f(x, y) * g(x, y)$  dapat diilustrasikan seperti berikut:

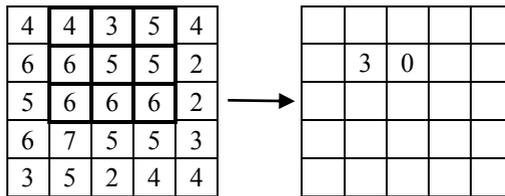
1. Tempatkan *kernel* pada sudut kiri atas citra, kemudian hitung nilai *pixel*-nya.



Hasil konvolusi = 3. Nilai ini dihitung dengan cara berikut:

$$(0 \times 4) + (-1 \times 4) + (0 \times 3) + (-1 \times 6) + (4 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 6) + (0 \times 6) = 3$$

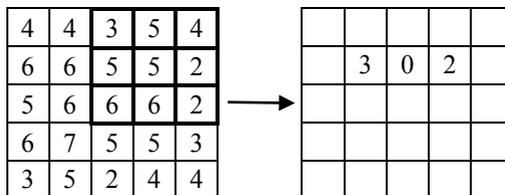
2. Geser *kernel* satu *pixel* ke kanan, kemudian hitung nilai *pixel*-nya seperti sebelumnya.



Hasil konvolusi = 0. Nilai ini dihitung dengan cara berikut:

$$(0 \times 4) + (-1 \times 3) + (0 \times 5) + (-1 \times 6) + (4 \times 5) + (-1 \times 5) + (0 \times 6) + (-1 \times 6) + (0 \times 6) = 0$$

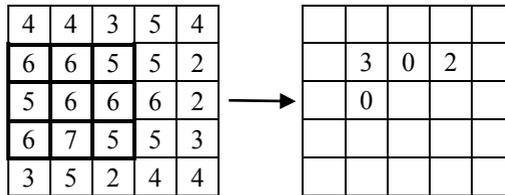
3. Geser *kernel* satu *pixel* lagi ke kanan, kemudian hitung nilai *pixel*-nya seperti sebelumnya.



Hasil konvolusi = 2. Nilai ini dihitung dengan cara berikut:

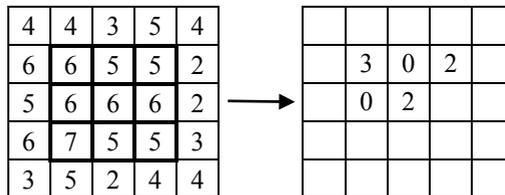
$$(0 \times 3) + (-1 \times 5) + (0 \times 4) + (-1 \times 5) + (4 \times 5) + (-1 \times 2) + (0 \times 6) + (-1 \times 6) + (0 \times 2) = 2$$

4. Selanjutnya, geser *kernel* satu *pixel* ke bawah, lalu mulai lagi melakukan konvolusi dari sisi kiri citra. Setiap kali menyelesaikan konvolusi, geser *kernel* satu *pixel* ke kanan.



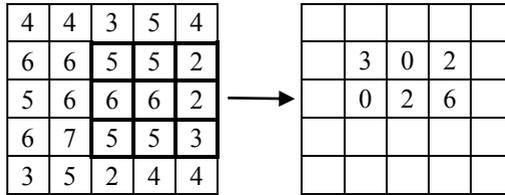
Hasil konvolusi = 0. Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 6) + (0 \times 5) + (-1 \times 5) + (4 \times 6) + (-1 \times 6) + (0 \times 6) + (-1 \times 7) + (0 \times 5) = 0$$



Hasil konvolusi = 2. Nilai ini dihitung dengan cara berikut:

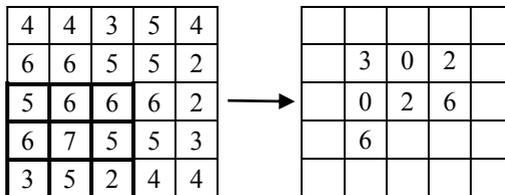
$$(0 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 6) + (4 \times 6) + (-1 \times 6) + (0 \times 7) + (-1 \times 5) + (0 \times 5) = 2$$



Hasil konvolusi = 6. Nilai ini dihitung dengan cara berikut:

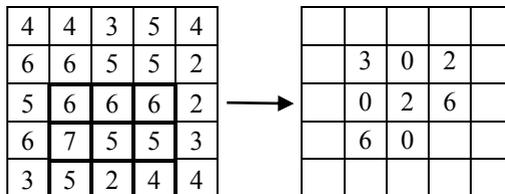
$$(0 \times 5) + (-1 \times 5) + (0 \times 2) + (-1 \times 6) + (4 \times 6) + (-1 \times 2) + (0 \times 5) + (-1 \times 5) + (0 \times 3) = 6$$

5. Geser *kernel* satu *pixel* ke bawah lagi, lalu mulai melakukan konvolusi dari sisi kiri citra. Setiap kali menyelesaikan konvolusi, geser *kernel* satu *pixel* ke kanan.



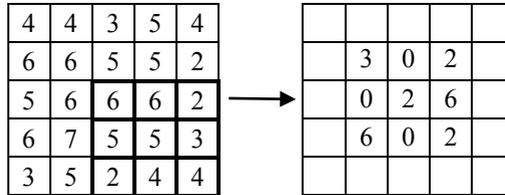
Hasil konvolusi = 6. Nilai ini dihitung dengan cara berikut:

$$(0 \times 5) + (-1 \times 6) + (0 \times 6) + (-1 \times 6) + (4 \times 7) + (-1 \times 5) + (0 \times 3) + (-1 \times 5) + (0 \times 2) = 6$$



Hasil konvolusi = 0. Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 6) + (0 \times 6) + (-1 \times 7) + (4 \times 5) + (-1 \times 5) + (0 \times 5) + (-1 \times 2) + (0 \times 4) = 0$$



Hasil konvolusi = 2. Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 6) + (0 \times 2) + (-1 \times 5) + (4 \times 5) + (-1 \times 3) + (0 \times 2) + (-1 \times 4) + (0 \times 4) = 2$$

Hasil dari konvolusi jika mendapatkan nilai negatif maka akan diubah menjadi nol (0). Tetapi bila hasil dari konvolusi mendapat nilai yang lebih besar dari nilai keabuan maksimum, maka nilai tersebut diubah menjadi nilai keabuan maksimum.

## 2.7 Rectifier Linear Units (ReLU)

Cara yang umum untuk memodelkan fungsi keluaran neuron dengan  $f$  sebagai fungsi dari masukan  $x$ .

$$f(x) = \tanh(x)$$

atau

$$f(x) = (1 + e^{-x})^{-1}$$

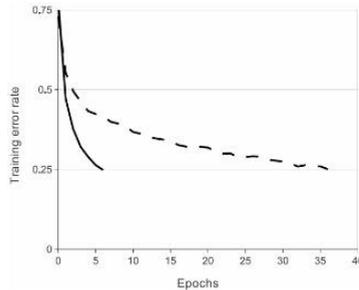
Apabila dilihat *gradient descent* dari waktu *training*, fungsi *saturating nonlinearities* di atas jauh lebih lambat daripada fungsi *non-saturating nonlinearity*

$$f(x) = \max(0; x)$$

Berdasarkan prosiding dari Vinod Nair dan Geoffrey E. Hinton[14], fungsi keluaran neuron *non-saturating nonlinearity* tersebut sebagai *rectified linear units* (ReLU). *Convolutional neural networks* (CNN) dengan *Rectified Linear Units* (ReLU)

dapat melakukan proses *training* beberapa kali lebih cepat daripada fungsi persamaan dengan menggunakan tanh *units*.

Gambar 2.5 adalah hasil demonstrasi dari penjelasan di atas, yang memperlihatkan angka iterasi yang dibutuhkan untuk mencapai *training error* sebesar 25% pada dataset CIFAR-10 dengan menggunakan empat *layer convolutional neural network* (CNN)[15].



**Gambar 2.5** Hasil *rectified linear units* (ReLU) pada dataset CIFAR-10[15]

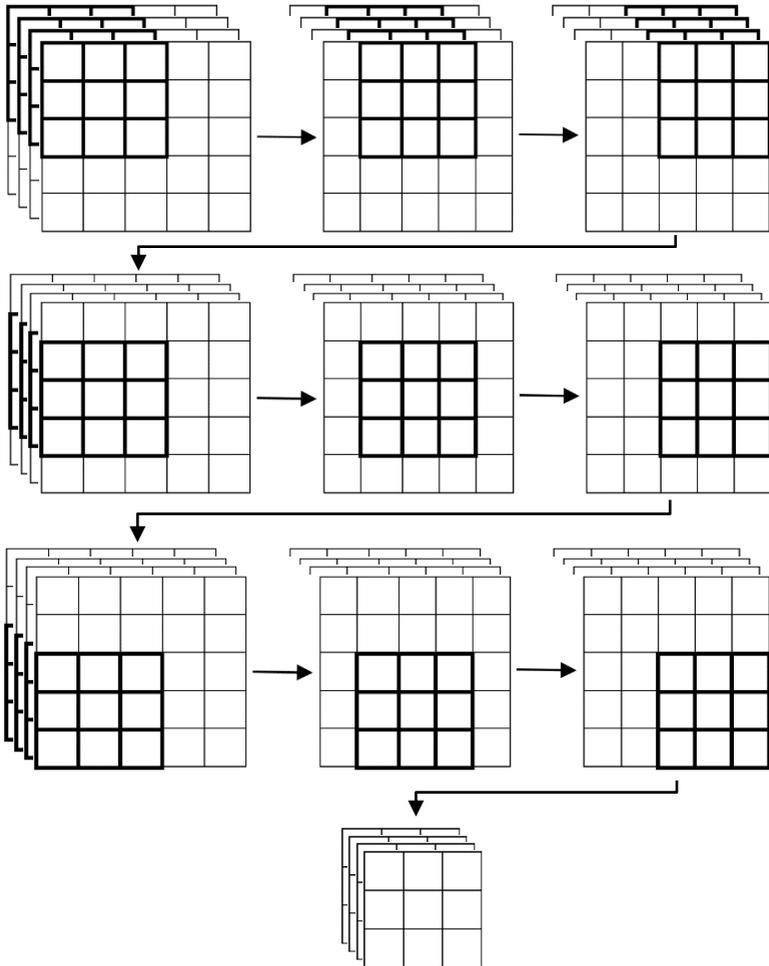
## 2.8 Pooling

Tujuan dari *pooling layers* adalah mendapatkan invarian spasial dengan mengurangi ukuran dari *feature map*. Setiap *pooled feature map* berhubungan dengan satu *feature map* dari *layer* sebelumnya. Nilainya adalah kombinasi masukan dari *kernel* yang lebih kecil  $n \times n$ . Proses *pooling* ini bisa menggunakan nilai pergeseran (*stride*) yang sesuai dengan *kernel*-nya atau lebih kecil dari nilai *kernel*-nya sehingga dapat terjadi *overlapping*.

Proses *pooling* yang digunakan ada dua jenis yaitu *average pooling* dan *max pooling*. Proses *average pooling* adalah mengambil nilai rata-rata dari data masukan, sedangkan proses *max pooling* adalah mengambil nilai maksimum dari data masukan dalam area tetangganya. Dua jenis proses *pooling* tersebut memiliki persamaan yaitu menghasilkan *feature map* yang memiliki ukuran lebih kecil dan jumlah *feature map* yang sama dengan data masukan[16].

Gambar 2.6 merupakan ilustrasi dari proses *pooling* dengan *kernel*  $3 \times 3$  *pixel* dan *stride* 1 *pixel* yang diaplikasikan pada data

masukannya dengan ukuran  $5 \times 5$  *pixel* dan mempunyai *feature map* sebanyak 4 buah. Sehingga akan menghasilkan data dengan ukuran lebih kecil yaitu  $3 \times 3$  *pixel* dan jumlah *feature map* yang tetap yaitu 4 buah.

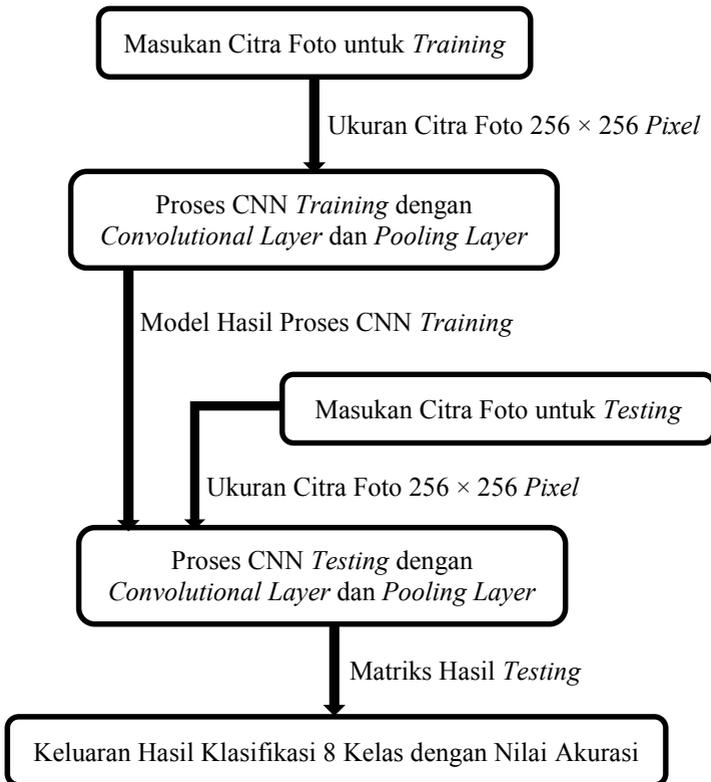


**Gambar 2.6** Ilustrasi *pooling layers* yang dapat berupa *average pooling* atau *max pooling*

*Halaman ini sengaja dikosongkan*

### BAB III PERANCANGAN SISTEM

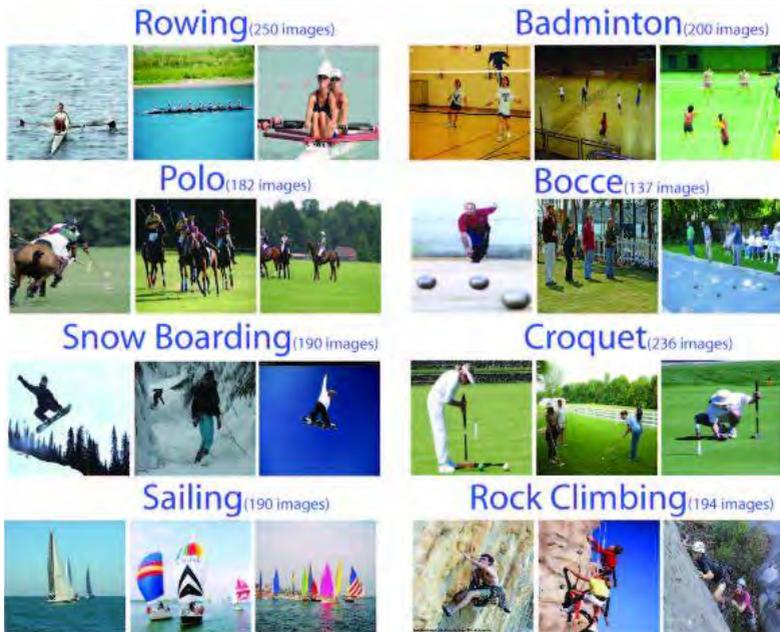
Pada bab III akan dibahas tentang perancangan sistem *deep learning* metode *convolutional neural network* (CNN). Blok diagram sistem dari *deep learning* metode *convolutional neural network* (CNN) dapat dilihat pada gambar 3.1 dan akan dijelaskan di subbab selanjutnya.



**Gambar 3.1** Blok diagram sistem *deep learning* metode *convolutional neural network* (CNN)

### 3.1 Data Masukan Citra Foto untuk *Training* dan *Testing*

Data masukan citra foto menggunakan dataset *UIUC sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas *image* yaitu *badminton*, *bocce*, *croquet*, *polo*, *rockclimbing*, *rowing*, *sailing*, *snowboarding*. Gambar 3.2 adalah dataset *UIUC sport event image* 8 kelas dengan jumlah *image* pada masing-masing kelasnya.



**Gambar 3.2** Dataset *UIUC sport event image* 8 kelas[1]

1. *Badminton* = 200 *image* Kelas 0
2. *Bocce* = 137 *image* Kelas 1
3. *Croquet* = 236 *image* Kelas 2
4. *Polo* = 182 *image* Kelas 3
5. *Rockclimbing* = 194 *image* Kelas 4
6. *Rowing* = 250 *image* Kelas 5
7. *Sailing* = 190 *image* Kelas 6
8. *Snowboarding* = 190 *image* Kelas 7

Dataset UIUC *sport event image* tersebut memiliki jumlah total 1579 *image* dengan format jpeg. *Image* pada dataset tersebut merupakan gabungan dari tingkat kesulitan dan jarak yaitu 3 jenis berdasarkan tingkat kesulitannya (*easy – medium – hard*) dan 3 jenis berdasarkan jaraknya (*close – mid – far*). Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas  $\times$  70 *image* dan untuk *testing* adalah 8 kelas  $\times$  60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-masing kelas tidak akan sama. Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga sebelum *image-image* tersebut dijadikan masukan maka ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256$  *pixel*.

### **3.2 Proses Convolutional Neural Network (CNN)**

Proses *convolutional neural network* (CNN) terdiri dari *convolutional layer* dan *pooling layer*, dimana pada setiap bagian *convolutional layer* akan dilakukan aktivasi menggunakan *rectified linear units* (ReLU) sebelum masuk ke layer selanjutnya. Total iterasi yang dilakukan untuk mendapatkan model hasil *convolutional neural network* (CNN) *training* adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

#### **3.2.1 Convolutional Layer**

*Image* yang masuk ke dalam *convolutional layer* akan dilakukan konvolusi dan sebelum masuk ke *convolutional layer* berikutnya atau *pooling layer* akan dilakukan aktivasi menggunakan *rectified linear units* (ReLU). Jumlah *convolutional layer* yang digunakan adalah 12 buah *layer*. Tiga *convolutional layer* pertama akan menghasilkan *feature maps* berjumlah 96. Tiga *convolutional layer* kedua akan menghasilkan *feature maps* berjumlah 256. Tiga *convolutional layer* ketiga akan menghasilkan *feature maps* berjumlah 384. Dua *convolutional layer* selanjutnya akan menghasilkan *feature maps* berjumlah 1024. Dan satu *convolutional layer* terakhir akan

menghasilkan *feature maps* berjumlah 8 yang sesuai dengan jumlah kelas dalam dataset UIUC *sport event image*.

### 3.2.2 *Rectified linear units (ReLU)*

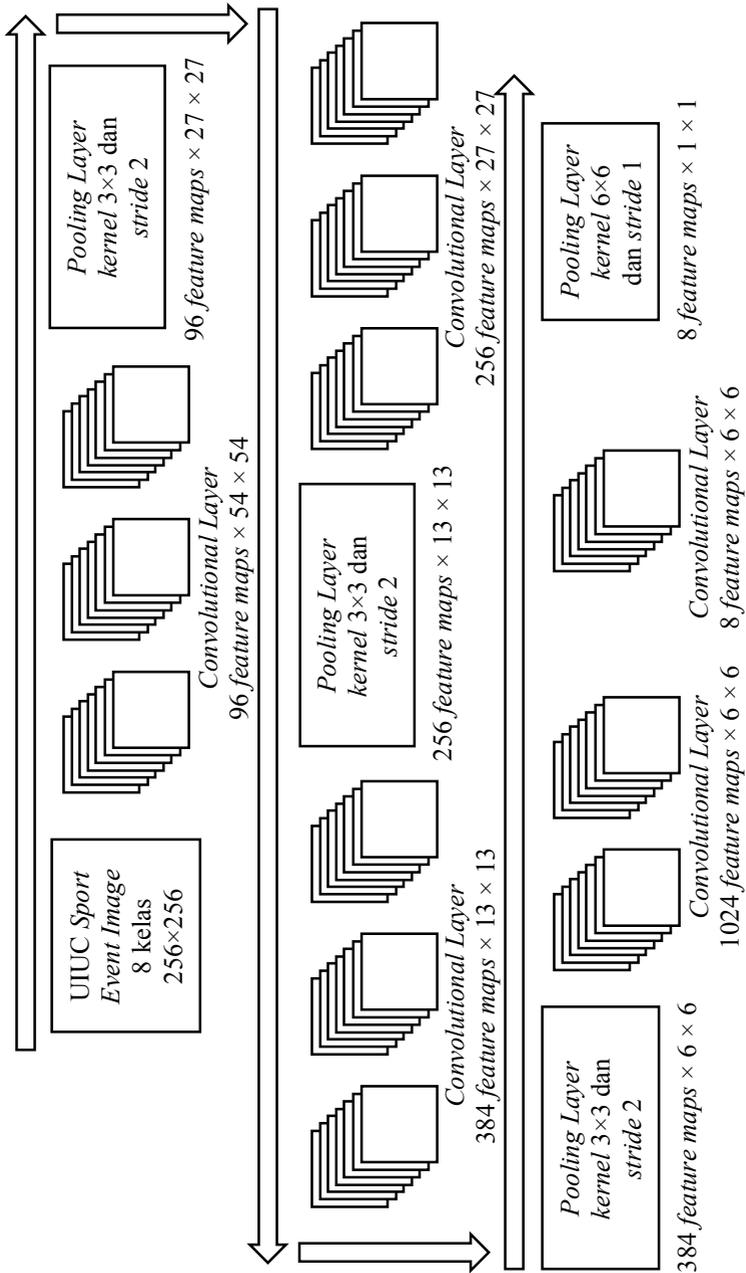
*Rectified linear units (ReLU)* dilakukan agar didapatkan hasil *feature maps* yang lebih optimal dan proses yang dilakukan dapat berjalan lebih cepat, sesuai dengan referensi[11]. *Rectified linear units (ReLU)* dilakukan setiap *convolutional layer* selesai melakukan proses konvolusi.

### 3.2.3 *Pooling Layer*

*Pooling layer* yang digunakan berjumlah 4 buah *layer*. Tiga buah *pooling layer* awal sama-sama memiliki *kernel size*  $3 \times 3$  dan *stride* 2 sedangkan sebuah *pooling layer* akhir memiliki *kernel size*  $6 \times 6$  dan *stride* 1. *Pooling layer* pertama terdapat pada setelah tiga *convolutional layer* pertama, *pooling layer* kedua terdapat pada setelah tiga *convolutional layer* kedua, *pooling layer* ketiga terdapat pada setelah tiga *convolutional layer* ketiga, dan *pooling layer* keempat terdapat pada setelah *convolutional layer* terakhir.

## 3.3 Keluaran Hasil Klasifikasi 8 Kelas dengan Nilai Akurasi

Model hasil *convolutional neural network (CNN) training* yang berjumlah 20 buah dari iterasi 10000, 20000, 30000, sampai iterasi 200000 dan citra foto 8 kelas  $\times$  60 *image* akan dijadikan masukan untuk proses *convolutional neural network (CNN) testing*. Setiap model hasil *convolutional neural network (CNN) training* akan dibandingkan dengan citra foto 8 kelas  $\times$  60 *image* untuk *testing* di dalam proses *convolutional neural network (CNN) testing*, proses tersebut diulang sampai model hasil *convolutional neural network (CNN) training* iterasi 200000. Setelah selesai maka akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversi menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase dimana hasil yang didapatkan tersebut adalah hasil keluaran akhir dari sistem *convolutional neural network (CNN)* yang telah dibangun.



**Gambar 3.3** Desain arsitektur sistem *convolutional neural network* (CNN)

### 3.4 Arsitektur Sistem *Convolutional Neural Network* (CNN)

Gambar 3.3 adalah desain arsitektur sistem *convolutional neural network* (CNN) yang dibangun. Masukan berupa UIUC *sport event image* dengan ukuran  $256 \times 256$  *pixel* dengan jumlah 8 kelas  $\times$  70 *image* untuk *training* dan 8 kelas  $\times$  60 *image* untuk *testing*, jadi total *image* untuk *training* berjumlah 560 *image* dan total *image* untuk *testing* berjumlah 480 *image*. Setelah masuk ke dalam sistem maka ukuran *image* tadi di-*crop* tepat pada bagian tengah dengan ukuran  $224 \times 224$  *pixel* dan diambil secara acak dengan jumlah 110 dari jumlah total *image* untuk sekali proses *training*. Sehingga masukan pada *convolutional layer* pertama adalah 110 *image* dengan parameter  $3 \times 224 \times 224$ .

Masukan tersebut selanjutnya dilakukan konvolusi pada *convolutional layer* pertama yang memiliki *kernel size*  $11 \times 11$  dan *stride* 4. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.01 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* pertama adalah 110 *image* dengan 96 *feature maps*  $\times$   $54 \times 54$  (30792960 *neuron*). Sebelum hasil konvolusi pada *convolutional layer* pertama masuk ke *convolutional layer* kedua, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 96 *feature maps*  $\times$   $54 \times 54$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Gambar 3.4 adalah ilustrasi dari *kernel* konvolusi dengan inisialisasi awal menggunakan *filler* Gaussian pada *convolutional layer* pertama. Huruf  $A_1, B_1, C_1, D_1, E_1, F_1, G_1, H_1, I_1$  dan angka 0 adalah representasi dari nilai yang ada di dalam *kernel* konvolusi pada *convolutional layer* pertama dengan *kernel size*  $11 \times 11$ , *stride* 4, dan *filler* Gaussian yang memiliki  $\sigma$  (*standard deviation*) 0.01. Nilai yang direpresentasikan oleh  $B_1, C_1, D_1, E_1, F_1, G_1, H_1,$  dan  $I_1$  memiliki angka sangat kecil hingga mendekati 0. Sedangkan nilai yang direpresentasikan huruf  $A_1$  memiliki angka yang mendekati 1.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	G <sub>1</sub>	H <sub>1</sub>	I <sub>1</sub>	H <sub>1</sub>	G <sub>1</sub>	0	0	0	0
0	0	G <sub>1</sub>	F <sub>1</sub>	E <sub>1</sub>	D <sub>1</sub>	E <sub>1</sub>	F <sub>1</sub>	G <sub>1</sub>	0	0	0
0	0	H <sub>1</sub>	E <sub>1</sub>	C <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>	E <sub>1</sub>	H <sub>1</sub>	0	0	0
0	0	I <sub>1</sub>	D <sub>1</sub>	B <sub>1</sub>	A <sub>1</sub>	B <sub>1</sub>	D <sub>1</sub>	I <sub>1</sub>	0	0	0
0	0	H <sub>1</sub>	E <sub>1</sub>	C <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>	E <sub>1</sub>	H <sub>1</sub>	0	0	0
0	0	G <sub>1</sub>	F <sub>1</sub>	E <sub>1</sub>	D <sub>1</sub>	E <sub>1</sub>	F <sub>1</sub>	G <sub>1</sub>	0	0	0
0	0	0	G <sub>1</sub>	H <sub>1</sub>	I <sub>1</sub>	H <sub>1</sub>	G <sub>1</sub>	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

**Gambar 3.4** Ilustrasi kernel konvolusi pada *convolutional layer* pertama

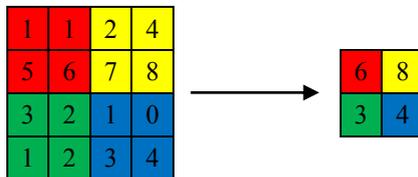
Pada *convolutional layer* kedua, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kedua adalah tetap yaitu 110 *image* dengan 96 *feature maps*  $\times 54 \times 54$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* kedua masuk ke *convolutional layer* ketiga, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 96 *feature maps*  $\times 54 \times 54$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *convolutional layer* ketiga, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kedua adalah tetap yaitu 110 *image* dengan 96

$feature\ maps \times 54 \times 54$ , dimana yang berubah adalah nilai di dalam  $pixel$ -nya. Sebelum hasil konvolusi pada  $convolutional\ layer$  ketiga masuk ke  $pooling\ layer$  pertama, dilakukan aktivasi menggunakan fungsi aktivasi berupa  $rectified\ linear\ units$  (ReLU) pada hasil tersebut.  $Rectified\ linear\ units$  (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan  $rectified\ linear\ units$  (ReLU) adalah tetap yaitu 110  $image$  dengan  $96\ feature\ maps \times 54 \times 54$ , dimana yang berubah adalah nilai di dalam  $pixel$ -nya.

Pada  $pooling\ layer$  pertama digunakan  $max\ pooling$  dengan  $kernel\ size\ 3 \times 3$  dan  $stride\ 2$ .  $Max\ pooling$  akan mengambil satu nilai maksimal dari 9  $pixel$  yang dijaring menggunakan  $kernel\ size\ 3 \times 3$ .  $Kernel$  mulai dari tepi paling kiri dan paling atas, lalu  $kernel$  akan bergerak 2  $pixel$  ke kanan sesuai dengan  $stride$ -nya. Setelah mencapai tepi paling kanan maka  $kernel$  akan mulai kembali dari tepi paling kiri tapi posisinya sekarang turun 2  $pixel$  ke bawah dibandingkan dengan posisi pertama yang juga sesuai dengan  $stride$ -nya. Proses ini akan terus berlangsung sampai  $kernel$  berada pada bagian tepi paling kanan dan paling bawah. Proses pada  $pooling\ layer$  pertama ini akan didapatkan 110  $image$  dengan  $96\ feature\ maps \times 27 \times 27$  (7698240  $neuron$ ), terlihat bahwa jumlah  $feature\ maps$  tetap sedangkan ukurannya yang berubah.

Gambar 3.5 adalah ilustrasi dari proses  $max\ pooling$  pada  $pooling\ layer$  dengan parameter  $kernel\ size\ 2 \times 2$  dan  $stride\ 2$  yang terjadi di sebuah  $image$  dengan ukuran  $4 \times 4\ pixel$  dan memiliki nilai  $pixel$  masing-masing (sebelah kiri). Warna merah, kuning, hijau, dan biru adalah representasi gerakan dari  $kernel$ . Lalu hasilnya adalah sebuah  $image$  baru dengan ukuran  $2 \times 2\ pixel$  (sebelah kanan).



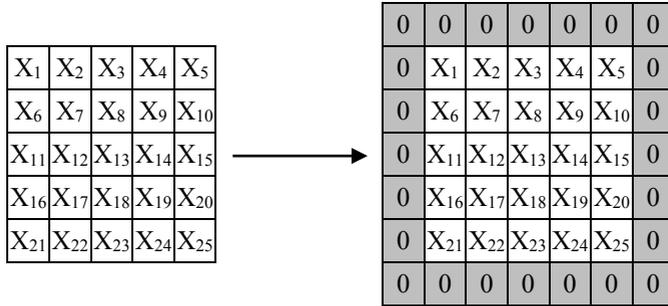
**Gambar 3.5** Ilustrasi  $max\ pooling$  pada sebuah  $image$



**Gambar 3.6** Ilustrasi proses konvolusi pada *convolutional layer*

Gambar 3.6 adalah ilustrasi dari proses konvolusi pada *convolutional layer* dengan parameter *kernel size*  $3 \times 3$  dan *stride* 3 dengan *filler* yang telah ditentukan serta bias *constant* bernilai 0 yang terjadi di dalam 5 buah *image* berwarna dengan ukuran  $6 \times 6$  *pixel* dan memiliki nilai *pixel* masing-masing. Hasil dari proses konvolusi ini akan menghasilkan 5 *image* dengan 10 *feature maps*  $\times 2 \times 2$  (sebelah kanan). Warna merah, kuning, hijau, dan biru adalah representasi gerakan dari *kernel* konvolusi. Sedangkan warna abu-abu adalah representasi dari *kernel* konvolusi dan warna putih adalah representasi dari bias. Pada sebelah kiri terdapat 5 buah *image* dengan ukuran  $6 \times 6$  *pixel* yang telah dibagi menjadi 3 *channel* (R-G-B) dimana paling atas merepresentasikan *red* (R), bagian tengah merepresentasikan *green* (G), dan paling bawah merepresentasikan *blue* (B). Masing-masing *channel* terlebih dahulu dilakukan konvolusi pada gerakan *kernel* pertama (warna merah) lalu hasil itu dijumlahkan antar *channel* dan dijumlahkan juga dengan bias sehingga mendapatkan sebuah nilai. Selanjutnya proses diulang menuju gerakan *kernel* kedua (warna kuning), lalu gerakan *kernel* ketiga (warna hijau), dan terakhir gerakan *kernel* keempat (warna biru). Hal tersebut dilakukan kembali secara berulang dan berlangsung sampai 10 kali sesuai dengan hasil yang ingin didapatkan yaitu 5 *image* dengan 10 *feature maps*  $\times 2 \times 2$  (sebelah kanan).

Pada *convolutional layer* keempat, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $5 \times 5$  dan *stride* 1 serta *pad* (*zero padding*) 2. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* keempat adalah 110 *image* dengan 256 *feature maps*  $\times 27 \times 27$  (20528640 *neuron*). Sebelum hasil konvolusi pada *convolutional layer* keempat masuk ke *convolutional layer* kelima, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 256 *feature maps*  $\times 27 \times 27$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.



**Gambar 3.7** Ilustrasi pad (*zero padding*) pada sebuah *image*

Gambar 3.7 adalah ilustrasi dari sebuah *image* dengan ukuran  $5 \times 5$  (sebelah kiri) dan diberikan penambahan pad (*zero padding*) 1 menjadi berukuran  $7 \times 7$  (sebelah kanan). Huruf X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub>, X<sub>4</sub>, X<sub>5</sub>, X<sub>6</sub>, X<sub>7</sub>, X<sub>8</sub>, X<sub>9</sub>, X<sub>10</sub>, X<sub>11</sub>, X<sub>12</sub>, X<sub>13</sub>, X<sub>14</sub>, X<sub>15</sub>, X<sub>16</sub>, X<sub>17</sub>, X<sub>18</sub>, X<sub>19</sub>, X<sub>20</sub>, X<sub>21</sub>, X<sub>22</sub>, X<sub>23</sub>, X<sub>24</sub>, X<sub>25</sub> adalah representasi dari nilai yang ada di dalam *pixel* sebuah *image* dan angka nol (0) adalah penambahan pad (*zero padding*) 1.

Pada *convolutional layer* kelima, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kelima adalah tetap yaitu 110 *image* dengan  $256 \text{ feature maps} \times 27 \times 27$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* kelima masuk ke *convolutional layer* keenam, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan  $256 \text{ feature maps} \times 27 \times 27$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *convolutional layer* keenam, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard*

*deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* keenam adalah tetap yaitu 110 *image* dengan 256 *feature maps*  $\times 27 \times 27$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* keenam masuk ke *pooling layer* kedua, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 256 *feature maps*  $\times 27 \times 27$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *pooling layer* kedua digunakan *max pooling* dengan *kernel size*  $3 \times 3$  dan *stride* 2. *Max pooling* akan mengambil satu nilai maksimal dari 9 *pixel* yang dijarang menggunakan *kernel size*  $3 \times 3$ . *Kernel* mulai dari tepi paling kiri dan paling atas, lalu *kernel* akan bergerak 2 *pixel* ke kanan sesuai dengan *stride*-nya. Setelah mencapai tepi paling kanan maka *kernel* akan mulai kembali dari tepi paling kiri tapi posisinya sekarang turun 2 *pixel* ke bawah dibandingkan dengan posisi pertama yang juga sesuai dengan *stride*-nya. Proses ini akan terus berlangsung sampai *kernel* berada pada bagian tepi paling kanan dan paling bawah. Proses pada *pooling layer* kedua ini akan didapatkan 110 *image* dengan 256 *feature maps*  $\times 13 \times 13$  (4759040 *neuron*), terlihat bahwa jumlah *feature maps* tetap sedangkan ukurannya yang berubah.

Pada *convolutional layer* ketujuh, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $3 \times 3$  dan *stride* 1 serta pad (*zero padding*) 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.01 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* keempat adalah 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$  (7138560 *neuron*). Sebelum hasil konvolusi pada *convolutional layer* ketujuh masuk ke *convolutional layer* kedelapan, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan

*rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

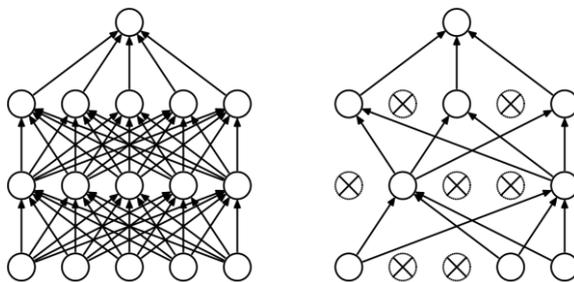
Pada *convolutional layer* kedelapan, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kedelapan adalah tetap yaitu 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* kedelapan masuk ke *convolutional layer* kesembilan, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *convolutional layer* kesembilan, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kesembilan adalah tetap yaitu 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* kesembilan masuk ke *pooling layer* ketiga, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 384 *feature maps*  $\times 13 \times 13$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *pooling layer* ketiga digunakan *max pooling* dengan *kernel size*  $3 \times 3$  dan *stride* 2. *Max pooling* akan mengambil satu nilai maksimal dari 9 *pixel* yang dijaring menggunakan *kernel size*  $3 \times 3$ . *Kernel* mulai dari tepi paling kiri dan paling atas, lalu *kernel*

akan bergerak 2 *pixel* ke kanan sesuai dengan *stride*-nya. Setelah mencapai tepi paling kanan maka *kernel* akan mulai kembali dari tepi paling kiri tapi posisinya sekarang turun 2 *pixel* ke bawah dibandingkan dengan posisi pertama yang juga sesuai dengan *stride*-nya. Proses ini akan terus berlangsung sampai *kernel* berada pada bagian tepi paling kanan dan paling bawah. Proses pada *pooling layer* ketiga ini akan didapatkan 110 *image* dengan 384 *feature maps*  $\times 6 \times 6$  (1520640 *neuron*), terlihat bahwa jumlah *feature maps* tetap sedangkan ukurannya yang berubah.

Hasil *pooling layer* ketiga masuk ke *dropout layer* dengan parameter *ratio* 0.5. Di dalam *dropout layer* tidak terjadi perubahan pada jumlah *feature maps*, ukuran, dan nilai masing-masing *pixel*. Fungsi utama dari *dropout layer* adalah mencegah terjadinya *overfitting*. Parameter yang digunakan adalah *ratio* 0.5, berarti hanya 50% dari total *neuron* (1520640 *neuron*) yang akan muncul dan dihubungkan ke *neuron-neuron* selanjutnya. Gambar 3.8 adalah ilustrasi proses *dropout* pada *dropout layer*. *Neuron-neuron* pada *standard neural network* saling berhubungan satu sama lain (sebelah kiri). Setelah diaplikasikan *dropout* menggunakan *dropout layer* pada *layer* pertama, kedua, dan ketiga maka hubungan antar *neuron* yang telah terjadi sebelumnya akan berubah, dimana hanya beberapa *neuron* yang akan muncul (*neuron* diberi tanda silang berarti tidak muncul) (sebelah kanan). Informasi lebih lengkap tentang *overfitting* dapat dilihat pada referensi[17] dan untuk pencegahannya dengan *dropout* pada referensi[18].



**Gambar 3.8** Ilustrasi proses *dropout* pada *dropout layer*[18]

Pada *convolutional layer* kesepuluh, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $3 \times 3$  dan *stride* 1

serta pad (*zero padding*) 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kesepuluh adalah 110 *image* dengan  $1024 \text{ feature maps} \times 6 \times 6$  (4055040 *neuron*). Sebelum hasil konvolusi pada *convolutional layer* kesepuluh masuk ke *convolutional layer* kesebelas, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan  $1024 \text{ feature maps} \times 6 \times 6$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *convolutional layer* kesebelas, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.05 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* kesebelas adalah tetap yaitu 110 *image* dengan  $1024 \text{ feature maps} \times 6 \times 6$ , dimana yang berubah adalah nilai di dalam *pixel*-nya. Sebelum hasil konvolusi pada *convolutional layer* kesebelas masuk ke *convolutional layer* keduabelas, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan  $1024 \text{ feature maps} \times 6 \times 6$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

Pada *convolutional layer* keduabelas, hasil tadi akan dilakukan konvolusi yang memiliki *kernel size*  $1 \times 1$  dan *stride* 1. Untuk inisialisasi awal menggunakan *filler* Gaussian dengan  $\sigma$  (*standard deviation*) 0.01 dan bias *constant* bernilai 0. Hasil konvolusi pada *convolutional layer* keduabelas adalah 110 *image* dengan  $8 \text{ feature maps} \times 6 \times 6$  (31680 *neuron*). Sebelum hasil konvolusi pada *convolutional layer* keduabelas masuk ke *pooling layer* keempat, dilakukan aktivasi menggunakan fungsi aktivasi berupa *rectified linear units* (ReLU) pada hasil tersebut. *Rectified*

*linear units* (ReLU) akan memberikan nilai sesuai dengan masukan jika lebih dari nol ( $> 0$ ) tapi jika nilai masukan kurang dari sama dengan nol ( $\leq 0$ ), maka nilai yang diberikan adalah nol (0). Hasil setelah diaktivasi menggunakan *rectified linear units* (ReLU) adalah tetap yaitu 110 *image* dengan 8 *feature maps*  $\times 6 \times 6$ , dimana yang berubah adalah nilai di dalam *pixel*-nya.

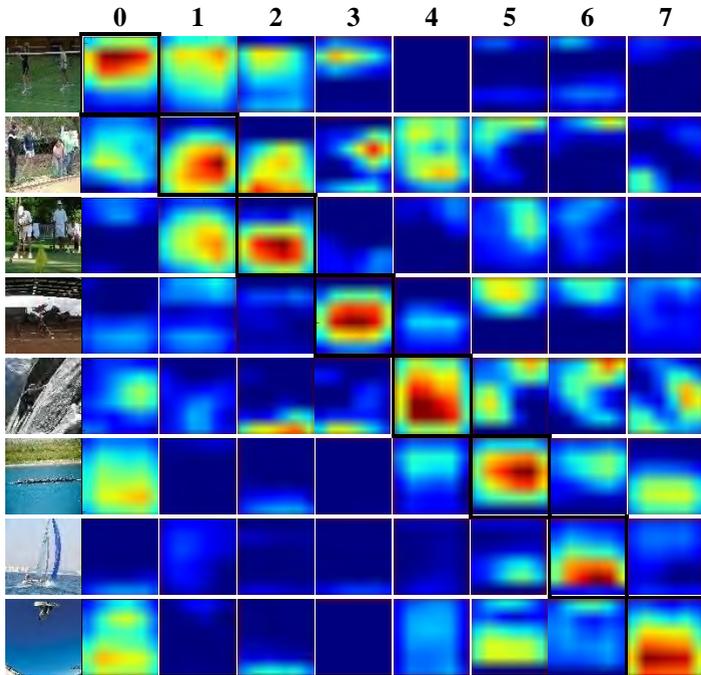
Pada *pooling layer* keempat digunakan *average pooling* dengan *kernel size*  $6 \times 6$  dan *stride* 1. *Average pooling* akan mengambil satu nilai rata-rata dari 12 *pixel* yang dijarah menggunakan *kernel size*  $6 \times 6$ . Hasil dari convolutional layer keduabelas adalah  $6 \times 6$  sehingga *kernel* dari pooling layer hanya melakukan sekali proses. Proses pada *pooling layer* keempat ini akan didapatkan 110 *image* dengan 8 *feature maps*  $\times 1 \times 1$  (880 *neuron*), terlihat bahwa jumlah *feature maps* tetap sedangkan ukurannya yang berubah.

Hasil tersebut masuk ke *layer* terakhir yaitu *softmax\_loss layer*. Pada bagian *softmax\_loss layer* akan dilakukan komputasi dengan *label* untuk mendapatkan *loss* dan 1 buah *neuron* yang berfungsi sebagai model. Hasil dari komputasi tadi akan dilakukan proses *backpropagation* melalui masing-masing *neuron* dan kembali ke *layer* pertama untuk meng-*update* parameter-parameternya. Lalu dilakukan kembali proses mulai dari awal tetapi dengan nilai parameter yang telah berubah. Proses tersebut akan dilakukan secara terus-menerus sampai iterasi 200000, dimana setiap iterasi 10000 model yang didapatkan akan disimpan.

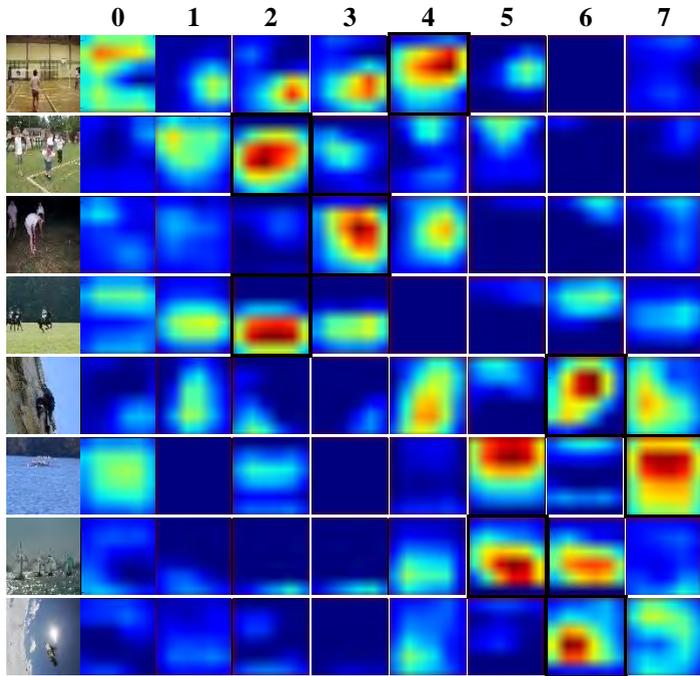
Pada proses *testing* sama dengan proses *training* yaitu tetap menggunakan arsitektur sistem *convolutional neural network* (CNN). Perbedaan terletak pada *dropout layer* dan *layer* terakhir. *Dropout layer* pada proses *testing*, parameter *ratio* tidak berfungsi untuk menghilangkan kehadiran sebagian *neuron* seperti pada proses *training* melainkan berfungsi untuk mengubah hubungan dengan *neuron* pada layer selanjutnya dengan cara bobot (*weight*) dikalikan dengan parameter *ratio* yang ada di dalam *dropout layer*. *Layer* terakhir pada proses testing tidak menggunakan *softmax\_loss layer* melainkan menggunakan *softmax layer*, sehingga hanya menghasilkan 1 *neuron*. Hasil tersebut akan dibandingkan dengan model hasil *training* untuk mendapatkan matriks berupa klasifikasi 8 kelas dengan nilai akurasi pada masing-masing kelas yang selanjutnya hasil tersebut dikonversikan menjadi bentuk prosentase.

## BAB IV PENGUJIAN DAN ANALISA

Bab IV adalah penjelasan tentang hasil pengujian dari sistem *deep learning* metode *convolutional neural networks* (CNN) yang telah dirancang sebelumnya dan selanjutnya dilakukan analisa dari hasil pengujian tersebut. Pengujian dilakukan sebanyak 5 kali lalu seluruh hasil dari kelima pengujian akan dijadikan satu untuk diambil nilai rata-ratanya yang selanjutnya akan dilakukan analisa dari masing-masing hasil pengujian tersebut. Analisa dilakukan pada setiap *image testing* seperti yang ditunjukkan pada gambar 4.1 dan gambar 4.2.



**Gambar 4.1** Visualisasi *feature map* dari *layer* terakhir dengan hasil yang benar



**Gambar 4.2** Visualisasi *feature map* dari *layer* terakhir dengan hasil yang salah

Gambar 4.1 dan 4.2 adalah visualisasi dari *feature map* pada *layer* terakhir yang menunjukkan kategori dari 8 kelas yaitu kelas 0 atau *badminton*, kelas 1 atau *bocce*, kelas 2 atau *croquet*, kelas 3 atau *polo*, kelas 4 atau *rockclimbing*, kelas 5 atau *rowing*, kelas 6 atau *sailing*, kelas 7 atau *snowboarding*. Gradasi warna dari biru tua – biru muda – hijau – kuning – orange sampai merah adalah merepresentasikan nilai kebenaran dari rendah sampai tinggi sedangkan yang diberikan tanda kotak hitam berarti masuk ke dalam kelas tersebut. Gambar 4.1 menampilkan hasil klasifikasi yang benar dan gambar 4.2 menampilkan hasil klasifikasi yang salah.

Pada gambar 4.1 di baris pertama dan kolom pertama, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 0 atau *badminton* dengan benar. Tetapi pada kolom kedua, ketiga, dan

keempat terdapat nilai walaupun rendah pada bagian tengah agak atas. Hal ini disebabkan garis net *badminton* yang berwarna terang dideteksi sebagai pinggiran arena *bocce* (kolom kedua) dan sebagai balok kayu yang digunakan pada *croquet* serta *polo* (kolom ketiga dan keempat).

Pada gambar 4.2 di baris pertama dan kolom pertama, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 0 atau *badminton* dengan benar. Pada kolom pertama terdapat nilai yang tersebar walaupun rendah pada bagian tengah tetapi nilainya masih tinggi pada kolom kelima, hal ini disebabkan raket dan kok *badminton* yang tidak terlihat dan posisi pemain yang terlihat adalah bagian punggung sehingga diklasifikasikan sebagai kelas 4 atau *rockclimbing*. Sedangkan kolom ketiga dan keempat terdapat nilai walaupun rendah pada bagian tengah kanan bawah, hal ini disebabkan struktur dan warna lantai arena *badminton* sehingga dideteksi seperti pada arena *croquet* atau *polo*.

Pada gambar 4.1 di baris kedua dan kolom kedua, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 1 atau *bocce* dengan benar. Tetapi pada kolom ketiga terdapat nilai walaupun rendah pada bagian tengah bawah, hal ini disebabkan pinggiran arena *bocce* dideteksi sebagai balok kayu yang digunakan dalam *croquet*. Pada kolom keempat terdapat nilai walaupun rendah pada bagian tengah agak kanan atas, hal ini disebabkan tiang vertikal yang berada di tepi jalan dideteksi sebagai balok kayu yang digunakan pada *polo*.

Pada gambar 4.2 di baris kedua dan kolom kedua, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 1 atau *bocce* dengan benar. Pada kolom kedua terdapat nilai yang tersebar walaupun rendah pada bagian tengah tetapi nilainya masih tinggi pada kolom ketiga, hal ini disebabkan terdapat balok kayu yang biasa digunakan pada *croquet* dan posisi para pemain yang terlihat lebih menyerupai gerakan sedang bermain *croquet* sehingga diklasifikasikan sebagai kelas 2 atau *croquet*.

Pada gambar 4.1 di baris ketiga dan kolom ketiga, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 2 atau *croquet* dengan benar. Tetapi pada kolom kedua terdapat nilai walaupun rendah pada bagian tengah agak kanan bawah, hal ini disebabkan bola dan orang yang berdiri di pinggiran arena *croquet* dideteksi sedang memainkan *bocce*.

Pada gambar 4.2 di baris ketiga dan kolom ketiga, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 2

atau *croquet* dengan benar. Pada kolom kelima terdapat nilai walaupun rendah pada bagian tengah kanan, hal ini disebabkan warna gelap yang menyerupai tebing pada *rockclimbing*. Pada kolom keempat posisi pemain yang terlihat lebih menyerupai gerakan sedang bermain *polo* yaitu membungkuk dengan kaki seperti menaiki kuda serta latar yang gelap dan kurang jelas sehingga diklasifikasikan sebagai kelas 3 atau *polo*.

Pada gambar 4.1 di baris keempat dan kolom keempat, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 3 atau *polo* dengan benar. Tetapi pada kolom keenam terdapat nilai walaupun rendah pada bagian tengah agak atas, hal ini disebabkan atap pada arena *polo* memiliki struktur bergaris-garis sehingga dideteksi sebagai dayung yang digunakan dalam *rowing*.

Pada gambar 4.2 di baris keempat dan kolom keempat, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 3 atau *polo* dengan benar. Pada kolom keempat terdapat nilai walaupun rendah pada bagian tengah agak bawah, hal ini disebabkan jarak pemain *polo* yang menaiki kuda cukup jauh dari kamera sehingga tetap terdeteksi walaupun memiliki nilai rendah. Pada kolom kedua terdapat nilai walaupun rendah pada bagian tengah bawah, hal ini disebabkan arena yang terlihat seperti pada arena *bocce*. Pada kolom ketiga terdapat nilai yang tinggi pada bagian tengah, hal ini disebabkan gerakan pemain dan tongkat yang dipakai terlihat dengan jelas seperti pada *croquet* sehingga diklasifikasikan sebagai kelas 2 atau *croquet*.

Pada gambar 4.1 di baris kelima dan kolom kelima, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 4 atau *rockclimbing* dengan benar. Tetapi pada kolom ketiga dan keempat terdapat nilai walaupun rendah pada bagian bawah, hal ini disebabkan struktur dan warna terang yang dideteksi seperti arena dalam *croquet* atau *polo*. Pada kolom keenam, ketujuh, dan kedelapan terdapat nilai yang tersebar walaupun rendah pada bagian kiri, atas, dan kanan, hal ini disebabkan posisi pemanjat, warna dan struktur tebing bagian atas kanan, latar pemandangan bagian kiri dideteksi seperti pada *rowing*, *sailing*, ataupun *snowboarding*.

Pada gambar 4.2 di baris kelima dan kolom kelima, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 4 atau *rockclimbing* dengan benar. Pada kolom kelima terdapat nilai yang tersebar walaupun rendah pada bagian tengah bawah, hal ini disebabkan struktur dan warna batu masih bisa dideteksi seperti tebing. Pada kolom

kedelapan terdapat nilai yang tersebar walaupun rendah pada bagian tengah kiri, hal ini disebabkan struktur dan warna batu bagian kiri dideteksi seperti bukit salju yang ada di *snowboarding*. Pada kolom ketujuh terdapat nilai yang tinggi pada bagian tengah agak atas, hal ini disebabkan posisi gerakan pemanjat dengan tali disampingnya seperti sedang memegang layar pada kapal layar dan latar langit yang mendukung posisi pemanjat tersebut sehingga diklasifikasikan sebagai kelas 6 atau *sailing*.

Pada gambar 4.1 di baris keenam dan kolom keenam, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 5 atau *rowing* dengan benar. Tetapi pada kolom pertama terdapat nilai yang tersebar walaupun rendah pada bagian tengah, hal ini disebabkan perahu yang memanjang horisontal dideteksi sebagai net yang digunakan dalam arena *badminton*. Pada kolom kedelapan terdapat nilai walaupun rendah pada bagian tengah agak bawah, hal ini disebabkan warna air sungai yang biru dideteksi sebagai latar langit yang banyak terdapat pada citra foto *snowboarding*.

Pada gambar 4.2 di baris keenam dan kolom keenam, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 5 atau *rowing* dengan benar. Pada kolom pertama terdapat nilai yang tersebar walaupun rendah pada bagian tengah, hal ini disebabkan perahu yang memanjang horisontal dideteksi sebagai net yang digunakan dalam arena *badminton*. Pada kolom keenam terdapat nilai yang cukup tinggi pada bagian tengah atas, hal ini disebabkan latar pepohonan berderet dideteksi sebagai latar pepohonan yang banyak terdapat pada bagian pinggir sungai pada olahraga *rowing*. Pada kolom kedelapan terdapat nilai yang lebih tinggi daripada kolom keenam dan tersebar pada seluruh bagian terutama pada bagian tengah agak atas, hal ini disebabkan dayung dan pemain yang tidak terlihat jelas serta perahu yang lebih mirip dengan bentuk papan yang digunakan pada *snowboarding* tetapi hal paling utama adalah warna biru air sungai yang lebih mendominasi dideteksi sebagai latar langit yang banyak terdapat pada citra foto *snowboarding* sehingga diklasifikasikan sebagai kelas 7 atau *snowboarding*.

Pada gambar 4.1 di baris ketujuh dan kolom ketujuh, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 6 atau *sailing* dengan benar. Dan kolom yang lain memiliki nilai sangat rendah, hal ini disebabkan citra foto *testing sailing* yang digunakan sangat jelas menampilkan orang yang sedang naik perahu layar di laut.

Pada gambar 4.2 di baris ketujuh dan kolom ketujuh, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 6 atau *sailing* dengan benar. Pada kolom ketujuh terdapat nilai yang tersebar pada bagian tengah, hal ini disebabkan layar pada kapal layar walaupun tidak terlihat jelas masih dapat dideteksi sebagai layar. Pada kolom keenam terdapat nilai yang tersebar pada bagian tengah agak bawah, hal ini disebabkan kapal layar terlihat berderet seperti membentuk perahu yang memanjang seperti pada *rowing* dan para pemain yang berjumlah banyak ikut mendukungnya sehingga diklasifikasikan sebagai kelas 5 atau *rowing*.

Pada gambar 4.1 di baris kedelapan dan kolom kedelapan, sistem yang dibangun dapat mengklasifikasikan *image testing* ke dalam kelas 7 atau *snowboarding* dengan benar. Tetapi pada kolom pertama dan keenam terdapat nilai yang tersebar walaupun rendah pada bagian tengah agak bawah, hal ini disebabkan penonton yang berderet dideteksi sebagai pemain *badminton* atau *rowing*.

Pada gambar 4.2 di baris kedelapan dan kolom kedelapan, sistem yang dibangun tidak dapat mengklasifikasikan *image testing* ke dalam kelas 7 atau *snowboarding* dengan benar. Meskipun pada kolom kedelapan terdapat nilai yang tersebar walaupun rendah pada bagian tengah agak atas, hal ini disebabkan kumpulan awan dideteksi seperti salju. Tetapi pada kolom ketujuh terdapat nilai yang cukup tinggi pada bagian tengah kiri, hal ini disebabkan citra foto close-up menampilkan bukit salju secara vertical dan terlalu mendominasi yang dideteksi sebagai layar pada kapal layar sehingga diklasifikasikan sebagai kelas 6 atau *sailing*.

#### 4.1 Pengujian Pertama

Data masukan citra foto menggunakan dataset UIUC *sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas image yaitu *badminton*, *bocce*, *croquet*, *polo*, *rockclimbing*, *rowing*, *sailing*, *snowboarding*.

Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas  $\times$  70 *image* dan untuk *testing* adalah 8 kelas  $\times$  60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-masing kelas serta masing-masing pengujian tidak akan sama.

Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256 \text{ pixel}$ .

Citra foto untuk *training* berjumlah 8 kelas  $\times$  70 *image* dengan ukuran  $256 \times 256 \text{ pixel}$  menjadi masukan proses *convolutional neural network* (CNN) *training* dengan *convolutional layer* dan *pooling layer*. Proses tersebut bertujuan untuk mendapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi yang dilakukan untuk mendapatkan hasil tersebut adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

Setelah didapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi 200000 maka dilakukan proses *convolutional neural network* (CNN) *testing* dengan menggunakan citra foto *testing*. Citra foto untuk *testing* berjumlah 8 kelas  $\times$  60 *image* dengan ukuran  $256 \times 256 \text{ pixel}$ . Citra foto untuk *testing* dan model hasil *convolutional neural network* (CNN) *training* tersebut menjadi masukan dari proses *convolutional neural network* (CNN) *testing*. Proses tersebut akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversikan menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase.

Gambar 4.3 adalah hasil matriks, total jumlah tes, dan akurasi rata-rata seluruh kelas (bagian atas) serta hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase (bagian bawah) pada pengujian pertama iterasi ke-60000. Pada pengujian pertama hasil dari iterasi ke-60000 merupakan yang terbaik dengan total jumlah tes 480 dan akurasi rata-rata 73.125%. Pada hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase, baris pertama dan kolom pertama merepresentasikan kelas 0 atau *badminton*. Baris kedua dan kolom kedua merepresentasikan kelas 1 atau *bocce*. Baris ketiga dan kolom ketiga merepresentasikan kelas 2 atau *croquet*. Baris keempat dan kolom keempat merepresentasikan kelas 3 atau *polo*. Baris kelima dan kolom kelima merepresentasikan kelas 4 atau *rockclimbing*. Baris keenam dan kolom keenam merepresentasikan kelas 5 atau *rowing*. Baris ketujuh dan kolom ketujuh merepresentasikan kelas 6 atau *sailing*.

Baris kedelapan dan kolom kedelapan merepresentasikan kelas 7 atau *snowboarding*.

38	5	5	2	8	1	0	1
6	28	11	2	8	1	1	3
0	12	44	3	1	0	0	0
1	12	3	42	0	1	0	1
2	1	0	3	50	2	1	1
4	4	0	2	4	41	4	1
2	0	0	0	2	1	54	1
0	1	0	0	1	1	3	54

Jumlah Tes = 480  
Akurasi = 73.125%

	ba	bo	cr	po	Ro	ro	sa	sn
badminton	63.3	8.3	8.3	3.3	13.3	1.7	0.0	1.7
bocce	10.0	46.7	18.3	3.3	13.3	1.7	1.7	5.0
croquet	0.0	20.0	73.3	5.0	1.7	0.0	0.0	0.0
polo	1.7	20.0	5.0	70.0	0.0	1.7	0.0	1.7
RockClimbing	3.3	1.7	0.0	5.0	83.3	3.3	1.7	1.7
rowing	6.7	6.7	0.0	3.3	6.7	68.3	6.7	1.7
sailing	3.3	0.0	0.0	0.0	3.3	1.7	90.0	1.7
snowboarding	0.0	1.7	0.0	0.0	1.7	1.7	5.0	90.0

**Gambar 4.3** Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian pertama iterasi ke-60000

Pada hasil matriks di baris pertama dan kolom pertama bernilai 38, berarti dari 60 *image testing* kelas 0 atau *badminton* hanya sebanyak 38 *image* yang benar-benar dikenali sebagai olahraga *badminton* atau sebesar 63.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 22 *image* tersebar di kelas 1 atau

*bocce*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 0 atau *badminton* tidak menampilkan raket *badminton* dan atau kok *badminton* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *badminton*, para pemain hanya menampilkan bagian punggung mereka, serta pencahayaan pada lapangan *badminton* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris kedua dan kolom kedua bernilai 28, berarti dari 60 *image testing* kelas 1 atau *bocce* hanya sebanyak 28 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 46.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 32 *image* tersebar di kelas 0 atau *badminton*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 1 atau *bocce* tidak menampilkan bola *bocce* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *bocce*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *bocce* yang kurang terang atau malah terlalu terang, serta terdapat balok kayu yang biasa digunakan pada olahraga *croquet* atau *polo*.

Pada hasil matriks di baris ketiga dan kolom ketiga bernilai 44, berarti dari 60 *image testing* kelas 2 atau *croquet* hanya sebanyak 44 *image* yang benar-benar dikenali sebagai olahraga *croquet* atau hanya sebesar 73.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 16 *image* tersebar di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 4 atau *rockclimbing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 2 atau *croquet* tidak menampilkan bola *croquet* dan atau tongkat *croquet* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *croquet*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *croquet* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris keempat dan kolom keempat

bernilai 42, berarti dari 60 *image testing* kelas 3 atau *polo* hanya sebanyak 42 *image* yang benar-benar dikenali sebagai olahraga *polo* atau hanya sebesar 70% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 18 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 3 atau *polo* tidak menampilkan bola *polo* dan atau tongkat *polo* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *polo* yang kurang terang atau malah terlalu terang, serta para pemain dan kuda mereka terlihat tidak jelas karena posisi yang terlalu jauh dari kamera.

Pada hasil matriks di baris kelima dan kolom kelima bernilai 50, berarti dari 60 *image testing* kelas 4 atau *rockclimbing* hanya sebanyak 50 *image* yang benar-benar dikenali sebagai olahraga *rockclimbing* atau hanya sebesar 83.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 10 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 5 atau *rowing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 4 atau *rockclimbing* tidak menampilkan gerakan memanjat dengan jelas, jumlah para pemanjat yang terlalu banyak, pencahayaan pada *image* yang kurang terang atau malah terlalu terang, tali yang digunakan memanjat tidak vertical melainkan horizontal, terdapat balok kayu yang biasa digunakan pada olahraga *polo*, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemanjatnya.

Pada hasil matriks di baris keenam dan kolom keenam bernilai 41, berarti dari 60 *image testing* kelas 5 atau *rowing* hanya sebanyak 41 *image* yang benar-benar dikenali sebagai olahraga *rowing* atau hanya sebesar 68.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 19 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing*

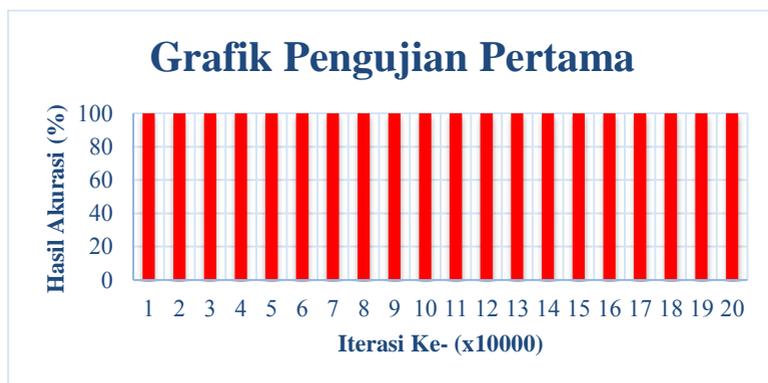
kelas 5 atau *rowing* tidak menampilkan perahu dengan jelas sehingga terlihat seperti garis pada net *badminton*, tidak menampilkan gerakan bermain *rowing* dengan jelas seperti terlalu membungkuk seperti pada olahraga *bocce* dan *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemain yang mendayung dengan perahunya.

Pada hasil matriks di baris ketujuh dan kolom ketujuh bernilai 54, berarti dari 60 *image testing* kelas 6 atau *sailing* hanya sebanyak 54 *image* yang benar-benar dikenali sebagai olahraga *sailing* atau hanya sebesar 90% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 6 *image* tersebar di kelas 0 atau *badminton*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 6 atau *sailing* menampilkan layar perahu-perahu yang berderet dengan jarak yang cukup jauh dari kamera sehingga terkesan seperti net pada arena *badminton*, layar yang gelap dan terlalu mendominasi *image* sehingga seperti bagian tebing pada *rockclimbing*, para pemain menampilkan gerakan yang lebih menyerupai *rowing*, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit biru dan layar perahu yang berwarna putih seperti area yang bersalju.

Pada hasil matriks di baris kedelapan dan kolom kedelapan bernilai 54, berarti dari 60 *image testing* kelas 7 atau *snowboarding* hanya sebanyak 54 *image* yang benar-benar dikenali sebagai olahraga *snowboarding* atau hanya sebesar 90% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 6 *image* tersebar di kelas 1 atau *bocce*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 7 atau *snowboarding* tidak menampilkan papan yang digunakan pada *snowboarding* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *snowboarding* dengan jelas, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, porsi yang ditampilkan langit biru terlalu mendominasi seperti warna air laut.

Gambar 4.4 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan *image training*. Dapat dilihat bahwa hasil yang ditampilkan oleh grafik dari iterasi 10000 sampai iterasi 200000 semuanya memiliki akurasi 100%. Hal ini terjadi karena *overfitting*. *Overfitting* adalah keadaan dimana sistem yang dibangun cenderung “menyimpan memori” data *training* daripada melakukan proses pembelajaran atau *training*. Dalam sistem *convolutional neural network* (CNN) yang dibangun sudah diterapkan salah satu cara untuk mencegah terjadinya keadaan *overfitting*, yaitu dengan menggunakan *dropout* pada *dropout layer*. Ternyata *dropout* yang telah diaplikasikan pada sistem *convolutional neural network* (CNN) melalui *dropout layer* tidak dapat mencegah terjadinya *overfitting*. Setelah ditelusuri, perihal utama yang menyebabkan terjadinya keadaan *overfitting* ini adalah data training yang terlalu sedikit.

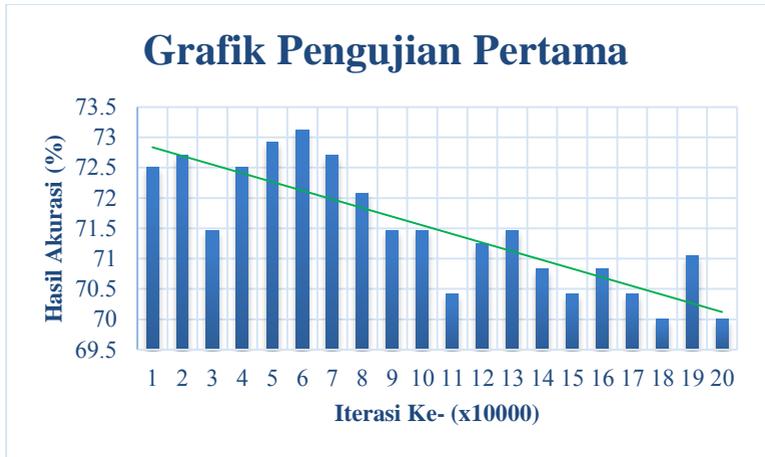
Tabel 4.1 dan gambar 4.5 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan *image testing*. Data akurasi rata-rata yang ada di dalam tabel 4.1 tersebut dibuat grafiknya pada gambar 4.5, dimana dapat dilihat grafik tersebut memiliki kecenderungan menurun yang seharusnya grafik memiliki kecenderungan naik. Hal ini merupakan salah satu akibat dari keadaan sistem *convolutional neural network* (CNN) yang *overfitting*, sehingga grafik yang berkecenderungan menurun memiliki kemungkinan besar untuk terjadi.



**Gambar 4.4** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan *image training*

**Tabel 4.1** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan *image testing*

<b>Iterasi Ke-</b>	<b>Hasil Akurasi (%)</b>
<b>10000</b>	<b>72.50000000</b>
<b>20000</b>	<b>72.70833333</b>
<b>30000</b>	<b>71.45833333</b>
<b>40000</b>	<b>72.50000000</b>
<b>50000</b>	<b>72.91666667</b>
<b>60000</b>	<b>73.12500000</b>
<b>70000</b>	<b>72.70833333</b>
<b>80000</b>	<b>72.08333333</b>
<b>90000</b>	<b>71.45833333</b>
<b>100000</b>	<b>71.45833333</b>
<b>110000</b>	<b>70.41666667</b>
<b>120000</b>	<b>71.25000000</b>
<b>130000</b>	<b>71.45833333</b>
<b>140000</b>	<b>70.83333333</b>
<b>150000</b>	<b>70.41666667</b>
<b>160000</b>	<b>70.83333333</b>
<b>170000</b>	<b>70.41666667</b>
<b>180000</b>	<b>70.00000000</b>
<b>190000</b>	<b>71.04166667</b>
<b>200000</b>	<b>70.00000000</b>



**Gambar 4.5** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama dengan masukan *image testing*

## 4.2 Pengujian Kedua

Data masukan citra foto menggunakan dataset UIUC *sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas *image* yaitu *badminton*, *bocce*, *croquet*, *polo*, *rockclimbing*, *rowing*, *sailing*, *snowboarding*.

Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas  $\times$  70 *image* dan untuk *testing* adalah 8 kelas  $\times$  60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-masing kelas serta masing-masing pengujian tidak akan sama. Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256$  *pixel*.

Citra foto untuk *training* berjumlah 8 kelas  $\times$  70 *image* dengan ukuran  $256 \times 256$  *pixel* menjadi masukan proses *convolutional neural network* (CNN) *training* dengan *convolutional layer* dan *pooling layer*. Proses tersebut bertujuan untuk mendapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi yang dilakukan untuk mendapatkan

hasil tersebut adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

Setelah didapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi 200000 maka dilakukan proses *convolutional neural network* (CNN) *testing* dengan menggunakan citra foto *testing*. Citra foto untuk *testing* berjumlah 8 kelas  $\times$  60 *image* dengan ukuran  $256 \times 256$  *pixel*. Citra foto untuk *testing* dan model hasil *convolutional neural network* (CNN) *training* tersebut menjadi masukan dari proses *convolutional neural network* (CNN) *testing*. Proses tersebut akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversikan menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase.

Gambar 4.6 adalah hasil matriks, total jumlah tes, dan akurasi rata-rata seluruh kelas (bagian atas) serta hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase (bagian bawah) pada pengujian kedua iterasi ke-200000. Pada pengujian kedua hasil dari iterasi ke-200000 merupakan yang terbaik dengan total jumlah tes 480 dan akurasi rata-rata 69.792%. Pada hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase, baris pertama dan kolom pertama merepresentasikan kelas 0 atau *badminton*. Baris kedua dan kolom kedua merepresentasikan kelas 1 atau *bocce*. Baris ketiga dan kolom ketiga merepresentasikan kelas 2 atau *croquet*. Baris keempat dan kolom keempat merepresentasikan kelas 3 atau *polo*. Baris kelima dan kolom kelima merepresentasikan kelas 4 atau *rockclimbing*. Baris keenam dan kolom keenam merepresentasikan kelas 5 atau *rowing*. Baris ketujuh dan kolom ketujuh merepresentasikan kelas 6 atau *sailing*. Baris kedelapan dan kolom kedelapan merepresentasikan kelas 7 atau *snowboarding*.

Pada hasil matriks di baris pertama dan kolom pertama bernilai 37, berarti dari 60 *image testing* kelas 0 atau *badminton* hanya sebanyak 37 *image* yang benar-benar dikenali sebagai olahraga *badminton* atau hanya sebesar 61.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 23 *image* tersebar di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 4 atau *rockclimbing*, di kelas

5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 0 atau *badminton* tidak menampilkan raket *badminton* dan atau kok *badminton* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *badminton*, para pemain hanya menampilkan bagian punggung mereka, serta pencahayaan pada lapangan *badminton* yang kurang terang atau malah terlalu terang.

$\begin{bmatrix} 37 & 9 & 5 & 0 & 5 & 3 & 0 & 1 \\ 7 & 36 & 5 & 2 & 7 & 1 & 0 & 2 \\ 0 & 20 & 38 & 1 & 0 & 1 & 0 & 0 \\ 3 & 20 & 5 & 31 & 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 2 & 50 & 3 & 1 & 0 \\ 11 & 2 & 0 & 1 & 4 & 37 & 3 & 2 \\ 0 & 0 & 1 & 1 & 2 & 2 & 51 & 3 \\ 0 & 2 & 0 & 1 & 0 & 0 & 2 & 55 \end{bmatrix}$	<p>Jumlah Tes = 480 Akurasi = 69.792%</p>
---	---

	ba	bo	cr	po	Ro	ro	sa	sn
badminton	61.7	15.0	8.3	0.0	8.3	5.0	0.0	1.7
bocce	11.7	60.0	8.3	3.3	11.7	1.7	0.0	3.3
croquet	0.0	33.3	63.3	1.7	0.0	1.7	0.0	0.0
polo	5.0	33.3	8.3	51.7	1.7	0.0	0.0	0.0
RockClimbing	5.0	1.7	0.0	3.3	83.3	5.0	1.7	0.0
rowing	18.3	3.3	0.0	1.7	6.7	61.7	5.0	3.3
sailing	0.0	0.0	1.7	1.7	3.3	3.3	85.0	5.0
snowboarding	0.0	3.3	0.0	1.7	0.0	0.0	3.3	91.7

**Gambar 4.6** Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian kedua iterasi ke-200000

Pada hasil matriks di baris kedua dan kolom kedua bernilai 36, berarti dari 60 *image testing* kelas 1 atau *bocce* hanya sebanyak 36 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 60% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 24 *image* tersebar di kelas 0 atau *badminton*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 1 atau *bocce* tidak menampilkan bola *bocce* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *bocce*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *bocce* yang kurang terang atau malah terlalu terang, serta terdapat balok kayu yang biasa digunakan pada olahraga *croquet* atau *polo*.

Pada hasil matriks di baris ketiga dan kolom ketiga bernilai 38, berarti dari 60 *image testing* kelas 2 atau *croquet* hanya sebanyak 38 *image* yang benar-benar dikenali sebagai olahraga *croquet* atau hanya sebesar 63.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 22 *image* tersebar di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 5 atau *rowing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 2 atau *croquet* tidak menampilkan bola *croquet* dan atau tongkat *croquet* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *croquet*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *croquet* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris keempat dan kolom keempat bernilai 31, berarti dari 60 *image testing* kelas 3 atau *polo* hanya sebanyak 31 *image* yang benar-benar dikenali sebagai olahraga *polo* atau hanya sebesar 70% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 29 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, dan di kelas 4 atau *rockclimbing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 3 atau *polo* tidak menampilkan bola *polo* dan atau tongkat *polo* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *polo*, para pemain

hanya menampilkan bagian punggung mereka, pencahayaan pada arena *polo* yang kurang terang atau malah terlalu terang, serta para pemain dan kuda mereka terlihat tidak jelas karena posisi yang terlalu jauh dari kamera.

Pada hasil matriks di baris kelima dan kolom kelima bernilai 50, berarti dari 60 *image testing* kelas 4 atau *rockclimbing* hanya sebanyak 50 *image* yang benar-benar dikenali sebagai olahraga *rockclimbing* atau hanya sebesar 83.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 10 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 5 atau *rowing*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 4 atau *rockclimbing* tidak menampilkan gerakan memanjat dengan jelas, jumlah para pemanjat yang terlalu banyak, pencahayaan pada *image* yang kurang terang atau malah terlalu terang, tali yang digunakan memanjat tidak vertical melainkan horizontal, terdapat balok kayu yang biasa digunakan pada olahraga *polo*, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemanjatnya.

Pada hasil matriks di baris keenam dan kolom keenam bernilai 37, berarti dari 60 *image testing* kelas 5 atau *rowing* hanya sebanyak 37 *image* yang benar-benar dikenali sebagai olahraga *rowing* atau hanya sebesar 61.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 23 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 5 atau *rowing* tidak menampilkan perahu dengan jelas sehingga terlihat seperti garis pada net *badminton*, tidak menampilkan gerakan bermain *rowing* dengan jelas seperti terlalu membungkuk seperti pada olahraga *bocce* dan *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemain yang mendayung dengan perahunya.

Pada hasil matriks di baris ketujuh dan kolom ketujuh bernilai 51, berarti dari 60 *image testing* kelas 6 atau *sailing* hanya sebanyak 51 *image* yang benar-benar dikenali sebagai olahraga

*sailing* atau hanya sebesar 85% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 9 *image* tersebar di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 6 atau *sailing* menampilkan layar perahu-perahu yang berderet dengan jarak yang cukup jauh dari kamera sehingga terkesan seperti net pada arena *badminton*, layar yang gelap dan terlalu mendominasi *image* sehingga seperti bagian tebing pada *rockclimbing*, para pemain menampilkan gerakan yang lebih menyerupai *rowing*, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit biru dan layar perahu yang berwarna putih seperti area yang bersalju.

Pada hasil matriks di baris kedelapan dan kolom kedelapan bernilai 55, berarti dari 60 *image testing* kelas 7 atau *snowboarding* hanya sebanyak 55 *image* yang benar-benar dikenali sebagai olahraga *snowboarding* atau hanya sebesar 91.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 5 *image* tersebar di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 7 atau *snowboarding* tidak menampilkan papan yang digunakan pada *snowboarding* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *snowboarding* dengan jelas, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, porsi yang ditampilkan langit biru terlalu mendominasi seperti warna air laut.

Gambar 4.7 adalah grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image training*. Dapat dilihat bahwa hasil yang ditampilkan oleh grafik dari iterasi 10000 sampai iterasi 200000 semuanya memiliki akurasi 100%. Hal ini terjadi karena *overfitting*. *Overfitting* adalah keadaan dimana sistem yang dibangun cenderung “menyimpan memori” data *training* daripada melakukan proses pembelajaran atau *training*. Dalam sistem *convolutional neural network* (CNN) yang dibangun sudah diterapkan salah satu cara untuk mencegah terjadinya keadaan *overfitting*, yaitu dengan menggunakan *dropout*

pada *dropout layer*. Ternyata *dropout* yang telah diaplikasikan pada sistem *convolutional neural network* (CNN) melalui *dropout layer* tidak dapat mencegah terjadinya *overfitting*. Setelah ditelusuri, perihal utama yang menyebabkan terjadinya keadaan *overfitting* ini adalah data training yang terlalu sedikit.



**Gambar 4.7** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image training*

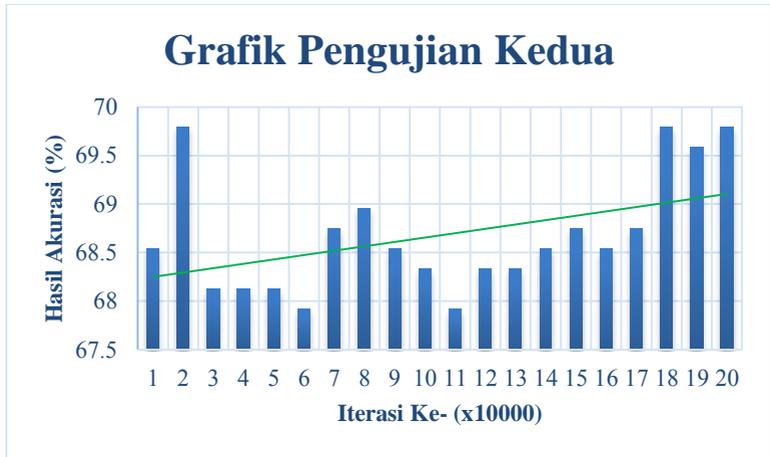
**Tabel 4.2** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image testing*

Iterasi Ke-	Hasil Akurasi (%)
10000	68.54166667
20000	69.79166667
30000	68.12500000
40000	68.12500000
50000	68.12500000
60000	67.91666667
70000	68.75000000

**Tabel 4.2** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image testing* (lanjutan)

Iterasi Ke-	Hasil Akurasi (%)
80000	68.95833333
90000	68.54166667
100000	68.33333333
110000	67.91666667
120000	68.33333333
130000	68.33333333
140000	68.54166667
150000	68.75000000
160000	68.54166667
170000	68.75000000
180000	69.79166667
190000	69.58333333
200000	69.79166667

Tabel 4.2 dan gambar 4.8 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image testing*. Dari data akurasi rata-rata yang ada didalam tabel 4.2 tersebut dibuat grafiknya pada gambar 4.8, dimana dapat dilihat grafik tersebut memiliki kecenderungan naik tetapi memiliki akurasi yang lebih rendah jika dibandingkan dengan hasil pengujian sebelumnya. Hal ini merupakan salah satu akibat dari keadaan sistem *convolutional neural network* (CNN) yang *overfitting*, sehingga walaupun grafik yang berkecenderungan naik tapi memiliki akurasi yang rendah.



**Gambar 4.8** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kedua dengan masukan *image testing*

### 4.3 Pengujian Ketiga

Data masukan citra foto menggunakan dataset UIUC *sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas image yaitu *badminton, bocce, croquet, polo, rockclimbing, rowing, sailing, snowboarding*.

Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas  $\times$  70 *image* dan untuk *testing* adalah 8 kelas  $\times$  60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-masing kelas serta masing-masing pengujian tidak akan sama. Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256$  *pixel*.

Citra foto untuk *training* berjumlah 8 kelas  $\times$  70 *image* dengan ukuran  $256 \times 256$  *pixel* menjadi masukan proses *convolutional neural network* (CNN) *training* dengan *convolutional layer* dan *pooling layer*. Proses tersebut bertujuan untuk mendapatkan model hasil *convolutional neural network* (CNN)

*training* dengan total iterasi yang dilakukan untuk mendapatkan hasil tersebut adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

Setelah didapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi 200000 maka dilakukan proses *convolutional neural network* (CNN) *testing* dengan menggunakan citra foto *testing*. Citra foto untuk *testing* berjumlah 8 kelas  $\times$  60 *image* dengan ukuran  $256 \times 256$  *pixel*. Citra foto untuk *testing* dan model hasil *convolutional neural network* (CNN) *training* tersebut menjadi masukan dari proses *convolutional neural network* (CNN) *testing*. Proses tersebut akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversikan menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase.

Gambar 4.9 adalah hasil matriks, total jumlah tes, dan akurasi rata-rata seluruh kelas (bagian atas) serta hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase (bagian bawah) pada pengujian ketiga iterasi ke-90000. Pada pengujian ketiga hasil dari iterasi ke-90000 merupakan yang terbaik dengan total jumlah tes 480 dan akurasi rata-rata 81.667%. Pada hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase, baris pertama dan kolom pertama merepresentasikan kelas 0 atau *badminton*. Baris kedua dan kolom kedua merepresentasikan kelas 1 atau *bocce*. Baris ketiga dan kolom ketiga merepresentasikan kelas 2 atau *croquet*. Baris keempat dan kolom keempat merepresentasikan kelas 3 atau *polo*. Baris kelima dan kolom kelima merepresentasikan kelas 4 atau *rockclimbing*. Baris keenam dan kolom keenam merepresentasikan kelas 5 atau *rowing*. Baris ketujuh dan kolom ketujuh merepresentasikan kelas 6 atau *sailing*. Baris kedelapan dan kolom kedelapan merepresentasikan kelas 7 atau *snowboarding*.

Pada hasil matriks di baris pertama dan kolom pertama bernilai 50, berarti dari 60 *image testing* kelas 0 atau *badminton* hanya sebanyak 50 *image* yang benar-benar dikenali sebagai olahraga *badminton* atau hanya sebesar 83.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 10 *image* tersebar di kelas 2 atau

*croquet*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 0 atau *badminton* tidak menampilkan raket *badminton* dan atau kok *badminton* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *badminton*, para pemain hanya menampilkan bagian punggung mereka, serta pencahayaan pada lapangan *badminton* yang kurang terang atau malah terlalu terang.

$$\begin{bmatrix} 50 & 0 & 4 & 0 & 2 & 1 & 1 & 2 \\ 5 & 35 & 10 & 3 & 2 & 0 & 0 & 5 \\ 2 & 6 & 47 & 3 & 1 & 0 & 0 & 1 \\ 0 & 4 & 3 & 49 & 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 50 & 5 & 0 & 4 \\ 2 & 3 & 1 & 1 & 2 & 49 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 55 & 3 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 57 \end{bmatrix} \quad \begin{array}{l} \text{Jumlah Tes} = 480 \\ \text{Akurasi} = 81.667\% \end{array}$$

	ba	bo	cr	po	Ro	ro	sa	sn
badminton	83.3	0.0	6.7	0.0	3.3	1.7	1.7	3.3
bocce	8.3	58.3	16.7	5.0	3.3	0.0	0.0	8.3
croquet	3.3	10.0	78.3	5.0	1.7	0.0	0.0	1.7
polo	0.0	6.7	5.0	81.7	5.0	0.0	1.7	0.0
RockClimbing	0.0	1.7	0.0	0.0	83.3	8.3	0.0	6.7
rowing	3.3	5.0	1.7	1.7	3.3	81.7	1.7	1.7
sailing	0.0	0.0	1.7	0.0	0.0	1.7	91.7	5.0
snowboarding	1.7	1.7	0.0	0.0	0.0	1.7	0.0	95.0

**Gambar 4.9** Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian ketiga iterasi ke-90000

Pada hasil matriks di baris kedua dan kolom kedua bernilai 35, berarti dari 60 *image testing* kelas 1 atau *bocce* hanya sebanyak 35 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 58.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 25 *image* tersebar di kelas 0 atau *badminton*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 1 atau *bocce* tidak menampilkan bola *bocce* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *bocce*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *bocce* yang kurang terang atau malah terlalu terang, serta terdapat balok kayu yang biasa digunakan pada olahraga *croquet* atau *polo*.

Pada hasil matriks di baris ketiga dan kolom ketiga bernilai 47, berarti dari 60 *image testing* kelas 2 atau *croquet* hanya sebanyak 47 *image* yang benar-benar dikenali sebagai olahraga *croquet* atau hanya sebesar 78.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 13 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 2 atau *croquet* tidak menampilkan bola *croquet* dan atau tongkat *croquet* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *croquet*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *croquet* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris keempat dan kolom keempat bernilai 49, berarti dari 60 *image testing* kelas 3 atau *polo* hanya sebanyak 49 *image* yang benar-benar dikenali sebagai olahraga *polo* atau hanya sebesar 81.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 11 *image* tersebar di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 4 atau *rockclimbing*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 3 atau *polo* tidak menampilkan bola *polo* dan atau tongkat *polo* dengan sempurna karena tertutupi oleh tubuh pemain, para

pemain tidak menampilkan gerakan bermain *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *polo* yang kurang terang atau malah terlalu terang, serta para pemain dan kuda mereka terlihat tidak jelas karena posisi yang terlalu jauh dari kamera.

Pada hasil matriks di baris kelima dan kolom kelima bernilai 50, berarti dari 60 *image testing* kelas 4 atau *rockclimbing* hanya sebanyak 50 *image* yang benar-benar dikenali sebagai olahraga *rockclimbing* atau hanya sebesar 83.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 10 *image* tersebar di kelas 1 atau *bocce*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 4 atau *rockclimbing* tidak menampilkan gerakan memanjat dengan jelas, jumlah para pemanjat yang terlalu banyak, pencahayaan pada *image* yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemanjatnya.

Pada hasil matriks di baris keenam dan kolom keenam bernilai 49, berarti dari 60 *image testing* kelas 5 atau *rowing* hanya sebanyak 49 *image* yang benar-benar dikenali sebagai olahraga *rowing* atau hanya sebesar 81.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 11 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 5 atau *rowing* tidak menampilkan perahu dengan jelas sehingga terlihat seperti garis pada net *badminton*, tidak menampilkan gerakan bermain *rowing* dengan jelas seperti terlalu membungkuk seperti pada olahraga *bocce* dan *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemain yang mendayung dengan perahunya.

Pada hasil matriks di baris ketujuh dan kolom ketujuh bernilai 55, berarti dari 60 *image testing* kelas 6 atau *sailing* hanya sebanyak 55 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 91.7% seperti yang tertera pada hasil

klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 5 *image* tersebar di kelas 2 atau *croquet*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 6 atau *sailing* para pemain menampilkan gerakan yang lebih menyerupai *rowing*, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit biru dan layar perahu yang berwarna putih seperti area yang bersalju.

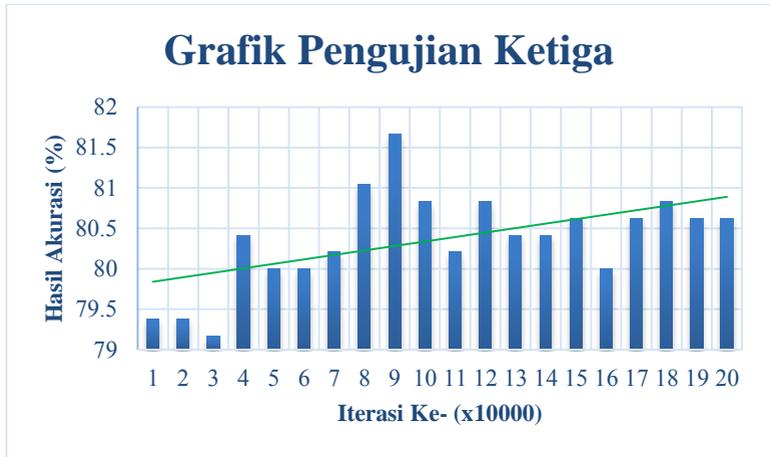
Pada hasil matriks di baris kedelapan dan kolom kedelapan bernilai 57, berarti dari 60 *image testing* kelas 7 atau *snowboarding* hanya sebanyak 57 *image* yang benar-benar dikenali sebagai olahraga *snowboarding* atau hanya sebesar 95% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 3 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, dan di kelas 5 atau *rowing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 7 atau *snowboarding* tidak menampilkan papan yang digunakan pada *snowboarding* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *snowboarding* dengan jelas, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, porsi yang ditampilkan langit biru terlalu mendominasi seperti warna air.



**Gambar 4.10** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan *image training*

**Tabel 4.3** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan *image testing*

<b>Iterasi Ke-</b>	<b>Hasil Akurasi (%)</b>
<b>10000</b>	<b>79.37500000</b>
<b>20000</b>	<b>79.37500000</b>
<b>30000</b>	<b>79.16666667</b>
<b>40000</b>	<b>80.41666667</b>
<b>50000</b>	<b>80.00000000</b>
<b>60000</b>	<b>80.00000000</b>
<b>70000</b>	<b>80.20833333</b>
<b>80000</b>	<b>81.04166667</b>
<b>90000</b>	<b>81.66666667</b>
<b>100000</b>	<b>80.83333333</b>
<b>110000</b>	<b>80.20833333</b>
<b>120000</b>	<b>80.83333333</b>
<b>130000</b>	<b>80.41666667</b>
<b>140000</b>	<b>80.41666667</b>
<b>150000</b>	<b>80.62500000</b>
<b>160000</b>	<b>80.00000000</b>
<b>170000</b>	<b>80.62500000</b>
<b>180000</b>	<b>80.83333333</b>
<b>190000</b>	<b>80.62500000</b>
<b>200000</b>	<b>80.62500000</b>



**Gambar 4.11** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan *image testing*

Gambar 4.10 adalah grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan *image training*. Dapat dilihat bahwa hasil yang ditampilkan oleh grafik dari iterasi 10000 sampai iterasi 200000 semuanya memiliki akurasi 100%. Hal ini terjadi karena *overfitting*. *Overfitting* adalah keadaan dimana sistem yang dibangun cenderung “menyimpan memori” data *training* daripada melakukan proses pembelajaran atau *training*. Dalam sistem *convolutional neural network* (CNN) yang dibangun sudah diterapkan salah satu cara untuk mencegah terjadinya keadaan *overfitting*, yaitu dengan menggunakan *dropout* pada *dropout layer*. Ternyata *dropout* yang telah diaplikasikan pada sistem *convolutional neural network* (CNN) melalui *dropout layer* tidak dapat mencegah terjadinya *overfitting*. Setelah ditelusuri, perhal utama yang menyebabkan terjadinya keadaan *overfitting* ini adalah data training yang terlalu sedikit.

Tabel 4.3 dan gambar 4.11 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian ketiga dengan masukan *image testing*. Dari data akurasi rata-rata yang ada didalam tabel 4.3 tersebut dibuat grafiknya pada gambar 4.11, dimana dapat dilihat grafik tersebut memiliki kecenderungan naik dan juga

memiliki akurasi yang lebih tinggi jika dibandingkan dengan hasil pengujian sebelumnya. Hal ini membuktikan bahwa pengujian ketiga tidak terpengaruh dengan keadaan sistem *convolutional neural network* (CNN) yang *overfitting*, sehingga grafik yang berkecenderungan naik dan juga memiliki akurasi yang tinggi.

#### 4.4 Pengujian Keempat

Data masukan citra foto menggunakan dataset UIUC *sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas image yaitu *badminton*, *bocce*, *croquet*, *polo*, *rockclimbing*, *rowing*, *sailing*, *snowboarding*.

Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas  $\times$  70 *image* dan untuk *testing* adalah 8 kelas  $\times$  60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-masing kelas serta masing-masing pengujian tidak akan sama. Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256$  *pixel*.

Citra foto untuk *training* berjumlah 8 kelas  $\times$  70 *image* dengan ukuran  $256 \times 256$  *pixel* menjadi masukan proses *convolutional neural network* (CNN) *training* dengan *convolutional layer* dan *pooling layer*. Proses tersebut bertujuan untuk mendapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi yang dilakukan untuk mendapatkan hasil tersebut adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

Setelah didapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi 200000 maka dilakukan proses *convolutional neural network* (CNN) *testing* dengan menggunakan citra foto *testing*. Citra foto untuk *testing* berjumlah 8 kelas  $\times$  60 *image* dengan ukuran  $256 \times 256$  *pixel*. Citra foto untuk *testing* dan model hasil *convolutional neural network* (CNN) *training* tersebut menjadi masukan dari proses *convolutional neural network* (CNN) *testing*. Proses tersebut akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata

dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversikan menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase.

53	0	3	1	0	1	1	1	1	Jumlah Tes = 480 Akurasi = 79.375%
4	39	11	1	3	0	0	0	2	
2	9	44	4	1	0	0	0	0	
2	3	1	52	0	0	2	0	0	
2	3	0	1	47	7	0	0	0	
6	2	0	3	2	40	2	5	5	
2	1	1	0	0	2	51	3	3	
2	0	0	0	1	1	1	55	5	

	ba	bo	cr	po	Ro	ro	sa	sn
badminton	88.3	0.0	5.0	1.7	0.0	1.7	1.7	1.7
bocce	6.7	65.0	18.3	1.7	5.0	0.0	0.0	3.3
croquet	3.3	15.0	73.3	6.7	1.7	0.0	0.0	0.0
polo	3.3	5.0	1.7	86.7	0.0	0.0	3.3	0.0
RockClimbing	3.3	5.0	0.0	1.7	78.3	11.7	0.0	0.0
rowing	10.0	3.3	0.0	5.0	3.3	66.7	3.3	8.3
sailing	3.3	1.7	1.7	0.0	0.0	3.3	85.0	5.0
snowboarding	3.3	0.0	0.0	0.0	1.7	1.7	1.7	91.7

**Gambar 4.12** Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian keempat iterasi ke-180000

Gambar 4.12 adalah hasil matriks, total jumlah tes, dan akurasi rata-rata seluruh kelas (bagian atas) serta hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase (bagian bawah) pada pengujian keempat iterasi ke-180000. Pada pengujian keempat hasil dari iterasi ke-180000 merupakan yang terbaik dengan total jumlah

tes 480 dan akurasi rata-rata 79.375%. Pada hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase, baris pertama dan kolom pertama merepresentasikan kelas 0 atau *badminton*. Baris kedua dan kolom kedua merepresentasikan kelas 1 atau *bocce*. Baris ketiga dan kolom ketiga merepresentasikan kelas 2 atau *croquet*. Baris keempat dan kolom keempat merepresentasikan kelas 3 atau *polo*. Baris kelima dan kolom kelima merepresentasikan kelas 4 atau *rockclimbing*. Baris keenam dan kolom keenam merepresentasikan kelas 5 atau *rowing*. Baris ketujuh dan kolom ketujuh merepresentasikan kelas 6 atau *sailing*. Baris kedelapan dan kolom kedelapan merepresentasikan kelas 7 atau *snowboarding*.

Pada hasil matriks di baris pertama dan kolom pertama bernilai 53, berarti dari 60 *image testing* kelas 0 atau *badminton* hanya sebanyak 53 *image* yang benar-benar dikenali sebagai olahraga *badminton* atau hanya sebesar 88.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 7 *image* tersebar di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 0 atau *badminton* tidak menampilkan raket *badminton* dan atau kok *badminton* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *badminton*, para pemain hanya menampilkan bagian punggung mereka, serta pencahayaan pada lapangan *badminton* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris kedua dan kolom kedua bernilai 39, berarti dari 60 *image testing* kelas 1 atau *bocce* hanya sebanyak 39 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 65% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 21 *image* tersebar di kelas 0 atau *badminton*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 1 atau *bocce* tidak menampilkan bola *bocce* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *bocce*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada

arena *bocce* yang kurang terang atau malah terlalu terang, serta terdapat balok kayu yang biasa digunakan pada olahraga *croquet* atau *polo*.

Pada hasil matriks di baris ketiga dan kolom ketiga bernilai 44, berarti dari 60 *image testing* kelas 2 atau *croquet* hanya sebanyak 44 *image* yang benar-benar dikenali sebagai olahraga *croquet* atau hanya sebesar 73.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 16 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 4 atau *rockclimbing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 2 atau *croquet* tidak menampilkan bola *croquet* dan atau tongkat *croquet* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *croquet*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *croquet* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris keempat dan kolom keempat bernilai 52, berarti dari 60 *image testing* kelas 3 atau *polo* hanya sebanyak 52 *image* yang benar-benar dikenali sebagai olahraga *polo* atau hanya sebesar 86.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 8 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 3 atau *polo* tidak menampilkan bola *polo* dan atau tongkat *polo* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *polo* yang kurang terang atau malah terlalu terang, serta para pemain dan kuda mereka terlihat tidak jelas karena posisi yang terlalu jauh dari kamera.

Pada hasil matriks di baris kelima dan kolom kelima bernilai 47, berarti dari 60 *image testing* kelas 4 atau *rockclimbing* hanya sebanyak 47 *image* yang benar-benar dikenali sebagai olahraga *rockclimbing* atau hanya sebesar 78.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 13 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 5 atau *rowing*.

Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 4 atau *rockclimbing* tidak menampilkan gerakan memanjat dengan jelas, jumlah para pemanjat yang terlalu banyak, tali yang digunakan memanjat tidak vertical melainkan horizontal, terdapat balok kayu yang biasa digunakan pada olahraga *polo*, pencahayaan pada *image* yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemanjatanya.

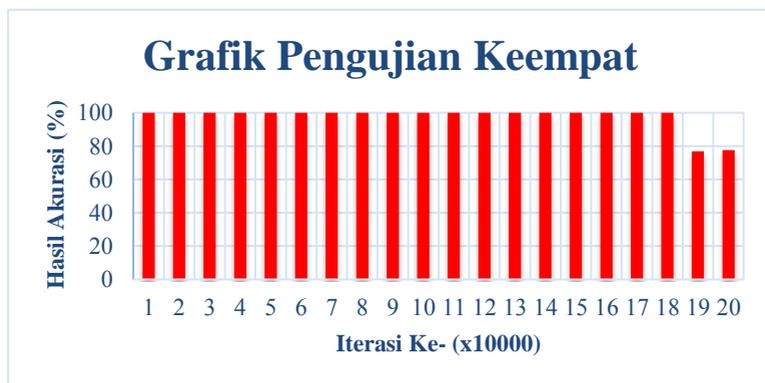
Pada hasil matriks di baris keenam dan kolom keenam bernilai 40, berarti dari 60 *image testing* kelas 5 atau *rowing* hanya sebanyak 40 *image* yang benar-benar dikenali sebagai olahraga *rowing* atau hanya sebesar 66.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 20 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 5 atau *rowing* tidak menampilkan perahu dengan jelas sehingga terlihat seperti garis pada net *badminton*, tidak menampilkan gerakan bermain *rowing* dengan jelas seperti terlalu membungkuk seperti pada olahraga *bocce* dan *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemain yang mendayung dengan perahunya.

Pada hasil matriks di baris ketujuh dan kolom ketujuh bernilai 51, berarti dari 60 *image testing* kelas 6 atau *sailing* hanya sebanyak 51 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 85% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 9 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 6 atau *sailing* menampilkan layar perahu-perahu yang berderet dengan jarak yang cukup jauh dari kamera sehingga terkesan seperti net pada arena *badminton*, para pemain menampilkan gerakan yang lebih menyerupai *rowing*, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit biru dan layar

perahu yang berwarna putih seperti area yang bersalju.

Pada hasil matriks di baris kedelapan dan kolom kedelapan bernilai 55, berarti dari 60 *image testing* kelas 7 atau *snowboarding* hanya sebanyak 55 *image* yang benar-benar dikenali sebagai olahraga *snowboarding* atau hanya sebesar 91.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 5 *image* tersebar di kelas 0 atau *badminton*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 6 atau *sailing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 7 atau *snowboarding* tidak menampilkan papan yang digunakan pada *snowboarding* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *snowboarding* dengan jelas, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, porsi yang ditampilkan langit biru terlalu mendominasi seperti warna air.

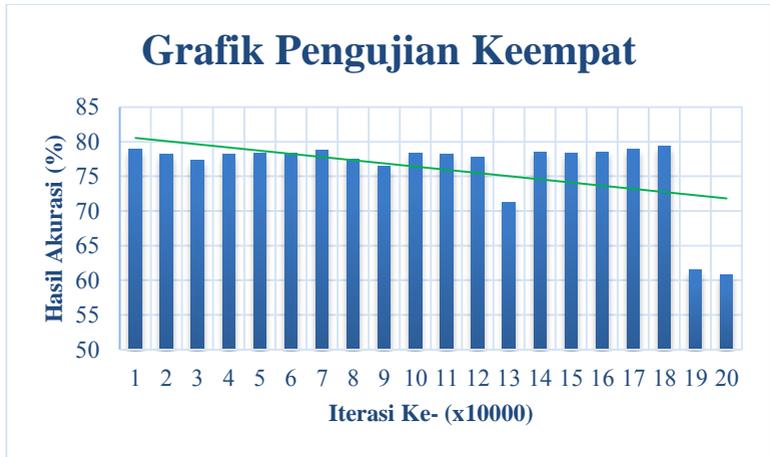
Gambar 4.13 adalah grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan *image training*. Dapat dilihat bahwa hasil yang ditampilkan oleh grafik dari iterasi 10000 sampai iterasi 180000 semuanya memiliki akurasi 100%. Hal ini terjadi karena *overfitting*. *Overfitting* adalah keadaan dimana sistem yang dibangun cenderung “menyimpan memori” data *training* daripada melakukan proses pembelajaran



**Gambar 4.13** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan *image training*

**Tabel 4.4** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan *image testing*

<b>Iterasi Ke-</b>	<b>Hasil Akurasi (%)</b>
<b>10000</b>	<b>78.95833333</b>
<b>20000</b>	<b>78.12500000</b>
<b>30000</b>	<b>77.29166667</b>
<b>40000</b>	<b>78.12500000</b>
<b>50000</b>	<b>78.33333333</b>
<b>60000</b>	<b>78.33333333</b>
<b>70000</b>	<b>78.75000000</b>
<b>80000</b>	<b>77.50000000</b>
<b>90000</b>	<b>76.45833333</b>
<b>100000</b>	<b>78.33333333</b>
<b>110000</b>	<b>78.12500000</b>
<b>120000</b>	<b>77.70833333</b>
<b>130000</b>	<b>71.25000000</b>
<b>140000</b>	<b>78.54166667</b>
<b>150000</b>	<b>78.33333333</b>
<b>160000</b>	<b>78.54166667</b>
<b>170000</b>	<b>78.95833333</b>
<b>180000</b>	<b>79.37500000</b>
<b>190000</b>	<b>61.45833333</b>
<b>200000</b>	<b>60.83333333</b>



**Gambar 4.14** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan *image testing*

atau *training*. Dalam sistem *convolutional neural network* (CNN) yang dibangun sudah diterapkan salah satu cara untuk mencegah terjadinya keadaan *overfitting*, yaitu dengan menggunakan *dropout* pada *dropout layer*. Ternyata *dropout* yang telah diaplikasikan pada sistem *convolutional neural network* (CNN) melalui *dropout layer* tidak dapat mencegah terjadinya *overfitting*. Setelah ditelusuri, perihal utama yang menyebabkan terjadinya keadaan *overfitting* ini adalah data training yang terlalu sedikit. Pada iterasi ke-190000 dan ke-200000 terjadi keanehan yaitu terjadi penurunan akurasi rata-rata yang tajam dari 100% menjadi 61.46% (iterasi ke-190000) lalu 60.83% (iterasi ke-200000). Keanehan tersebut ternyata disebabkan proses *training* pada iterasi ke-180760, iterasi ke-180780, dan iterasi ke-180800 menghasilkan loss yang semakin besar seperti yang ditunjukkan pada gambar 4.15.

Tabel 4.4 dan gambar 4.14 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian keempat dengan masukan *image testing*. Dari data akurasi rata-rata yang ada didalam tabel 4.4 tersebut dibuat grafiknya pada gambar 4.14, dimana dapat dilihat grafik tersebut pada iterasi ke-10000 sampai iterasi ke-180000 memiliki kecenderungan naik dan juga memiliki akurasi

yang lebih tinggi jika dibandingkan dengan hasil pengujian pertama dan pengujian kedua serta memiliki nilai akurasi yang mendekati pengujian ketiga. Hal ini membuktikan bahwa pengujian keempat seperti pada pengujian ketiga yaitu tidak terpengaruh dengan keadaan sistem *convolutional neural network* (CNN) yang *overfitting*, sehingga grafik yang berkecenderungan naik dan juga memiliki akurasi yang tinggi. Tetapi pada iterasi ke-190000 dan ke-200000 terjadi keanehan yaitu terjadi penurunan akurasi rata-rata yang tajam. Keanehan tersebut ternyata juga berhubungan dengan proses *training* pada iterasi ke-180760, iterasi ke-180780, dan iterasi ke-180800 menghasilkan *loss* yang semakin besar seperti yang ditunjukkan pada gambar 4.15.

Gambar 4.15 adalah sebagian log atau rekaman hasil *training* sistem *convolutional neural network* (CNN) pada pengujian keempat. Log tersebut menampilkan *learning rate* yang digunakan sebesar 0.001 dan *loss* yang didapatkan dari hasil *training* sistem *convolutional neural network* (CNN) pada pengujian keempat. Pada iterasi ke-180700 sampai dengan iterasi ke-180760, keadaan terlihat normal karena *loss* yang didapatkan sangat kecil. Nilai *loss* yang sangat kecil tersebut didapat setelah terjadi perubahan parameter pada sistem *convolutional neural network* (CNN) di setiap iterasi dengan hasil *loss* yang didapatkan di atas iterasi ke-100000 cenderung  $\frac{1}{1000}$  atau  $\frac{1}{10000}$ , seperti ditunjukkan pada iterasi ke-180700 sampai dengan iterasi ke-180760.

Pada iterasi ke-180760, iterasi ke-180760, dan iterasi ke-180780 terjadi keanehan dimana nilai *loss* yang didapatkan mengalami perubahan yang semakin besar. Perubahan terjadi dari nilai *loss* 0.000971931 (iterasi ke-180760) menjadi 0.218229 (iterasi ke-180780) lalu sampai puncaknya menjadi 2.0837 (iterasi ke-180800). Nilai *loss* pada iterasi ke-180800 itu memiliki nilai yang menyerupai dengan nilai *loss* pada iterasi ke-0 sampai dengan iterasi ke-40. Selanjutnya nilai *loss* pada iterasi ke-180800 tersebut berkurang sedikit demi sedikit sampai pada iterasi ke-184040. Pada iterasi ke-184040 sampai iterasi ke-200000 terjadi keanehan lagi dimana nilai *loss* tidak berkurang, melainkan terjadi keadaan osilasi dengan nilai *loss* berada antara 0.38 sampai 0.7.

```
Iteration 0, loss = 2.09256
Iteration 0, lr = 0.001
Iteration 20, loss = 2.07712
Iteration 20, lr = 0.001
Iteration 40, loss = 2.08171
Iteration 40, lr = 0.001
Iteration 60, loss = 2.01594
Iteration 60, lr = 0.001
Iteration 80, loss = 1.86104
Iteration 80, lr = 0.001
Iteration 100, loss = 1.82987
Iteration 100, lr = 0.001
.
.
.
Iteration 180700, loss = 0.001456
Iteration 180700, lr = 0.001
Iteration 180720, loss = 0.000206538
Iteration 180720, lr = 0.001
Iteration 180740, loss = 0.00517483
Iteration 180740, lr = 0.001
Iteration 180760, loss = 0.000971391
Iteration 180760, lr = 0.001
Iteration 180780, loss = 0.218229
Iteration 180780, lr = 0.001
Iteration 180800, loss = 2.0837
Iteration 180800, lr = 0.001
Iteration 180820, loss = 2.06907
Iteration 180820, lr = 0.001
Iteration 180840, loss = 2.06113
Iteration 180840, lr = 0.001
Iteration 180860, loss = 2.03102
Iteration 180860, lr = 0.001
Iteration 180880, loss = 1.93435
Iteration 180880, lr = 0.001
Iteration 180900, loss = 1.89322
Iteration 180900, lr = 0.001
.
.
.
Iteration 184040, loss = 0.597939
Iteration 184040, lr = 0.001
Iteration 184060, loss = 0.535556
Iteration 184060, lr = 0.001
```

```
Iteration 184080, loss = 0.500959
Iteration 184080, lr = 0.001
Iteration 184100, loss = 0.60655
Iteration 184100, lr = 0.001
.
.
.
Iteration 199820, loss = 0.397691
Iteration 199820, lr = 0.001
Iteration 199840, loss = 0.380022
Iteration 199840, lr = 0.001
Iteration 199860, loss = 0.474324
Iteration 199860, lr = 0.001
Iteration 199880, loss = 0.456757
Iteration 199880, lr = 0.001
Iteration 199900, loss = 0.492693
Iteration 199900, lr = 0.001
Iteration 199920, loss = 0.624064
Iteration 199920, lr = 0.001
Iteration 199940, loss = 0.625516
Iteration 199940, lr = 0.001
Iteration 199960, loss = 0.700718
Iteration 199960, lr = 0.001
Iteration 199980, loss = 0.567786
Iteration 199980, lr = 0.001
Iteration 200000, loss = 0.549485
Optimization Done.
```

**Gambar 4.15** Log hasil *training* sistem *convolutional neural network* (CNN) pada pengujian keempat

### 4.5 Pengujian Kelima

Data masukan citra foto menggunakan dataset UIUC *sport event image* yang berasal dari *website* <http://vision.stanford.edu/> berisi 8 kelas image yaitu *badminton, bocce, croquet, polo, rockclimbing, rowing, sailing, snowboarding*.

Citra foto yang digunakan dari dataset tersebut pada masing-masing kelas berjumlah 70 *image* untuk *training* dan 60 *image* untuk *testing*, sehingga total *image* untuk *training* adalah 8 kelas × 70 *image* dan untuk *testing* adalah 8 kelas × 60 *image*. Pemilihan *image* untuk *training* dan *testing* tersebut dilakukan secara acak dimana *image* untuk *training* dan *image* untuk *testing* pada masing-

masing kelas serta masing-masing pengujian tidak akan sama. Ukuran *image* dari masing-masing kelas tersebut memiliki berbagai variasi sehingga ukuran dari semua *image* tersebut harus disamakan menjadi  $256 \times 256$  *pixel*.

Citra foto untuk *training* berjumlah 8 kelas  $\times$  70 *image* dengan ukuran  $256 \times 256$  *pixel* menjadi masukan proses *convolutional neural network* (CNN) *training* dengan *convolutional layer* dan *pooling layer*. Proses tersebut bertujuan untuk mendapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi yang dilakukan untuk mendapatkan hasil tersebut adalah 200000 iterasi dimana setiap 10000 iterasi maka model hasil *convolutional neural network* (CNN) *training* tersebut akan disimpan.

Setelah didapatkan model hasil *convolutional neural network* (CNN) *training* dengan total iterasi 200000 maka dilakukan proses *convolutional neural network* (CNN) *testing* dengan menggunakan citra foto *testing*. Citra foto untuk *testing* berjumlah 8 kelas  $\times$  60 *image* dengan ukuran  $256 \times 256$  *pixel*. Citra foto untuk *testing* dan model hasil *convolutional neural network* (CNN) *training* tersebut menjadi masukan dari proses *convolutional neural network* (CNN) *testing*. Proses tersebut akan mendapatkan matriks hasil *testing*, total jumlah *image testing* yang dilakukan dan hasil akurasi rata-rata dari seluruh kelas. Selanjutnya matriks hasil *testing* tersebut akan dikonversikan menjadi klasifikasi 8 kelas dengan nilai akurasi pada setiap kelasnya dalam bentuk prosentase.

Gambar 4.16 adalah hasil matriks, total jumlah tes, dan akurasi rata-rata seluruh kelas (bagian atas) serta hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase (bagian bawah) pada pengujian kelima iterasi ke-30000. Pada pengujian kelima hasil dari iterasi ke-30000 merupakan yang terbaik dengan total jumlah tes 480 dan akurasi rata-rata 79.167%. Pada hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase, baris pertama dan kolom pertama merepresentasikan kelas 0 atau *badminton*. Baris kedua dan kolom kedua merepresentasikan kelas 1 atau *bocce*. Baris ketiga dan kolom ketiga merepresentasikan kelas 2 atau *croquet*. Baris keempat dan kolom keempat merepresentasikan kelas 3 atau *polo*. Baris kelima dan kolom kelima merepresentasikan kelas 4 atau *rockclimbing*. Baris keenam dan kolom keenam merepresentasikan kelas 5 atau *rowing*. Baris

ketujuh dan kolom ketujuh merepresentasikan kelas 6 atau *sailing*. Baris kedelapan dan kolom kedelapan merepresentasikan kelas 7 atau *snowboarding*.

47	2	3	3	3	0	0	2
4	33	6	6	6	1	0	4
3	4	51	1	1	0	0	0
1	4	1	47	6	1	0	0
0	0	0	1	52	4	0	3
2	3	2	1	4	44	3	1
2	0	0	0	1	1	54	2
2	2	0	0	2	2	0	52

Jumlah Tes = 480  
Akurasi = 79.167%

	ba	bo	cr	po	Ro	ro	sa	sn
badminton	78.3	3.3	5.0	5.0	5.0	0.0	0.0	3.3
bocce	6.7	55.0	10.0	10.0	10.0	1.7	0.0	6.7
croquet	5.0	6.7	85.0	1.7	1.7	0.0	0.0	0.0
polo	1.7	6.7	1.7	78.3	10.0	1.7	0.0	0.0
RockClimbing	0.0	0.0	0.0	1.7	86.7	6.7	0.0	5.0
rowing	3.3	5.0	3.3	1.7	6.7	73.3	5.0	1.7
sailing	3.3	0.0	0.0	0.0	1.7	1.7	90.0	3.3
snowboarding	3.3	3.3	0.0	0.0	3.3	3.3	0.0	86.7

**Gambar 4.16** Hasil matriks dan klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase pada pengujian kelima iterasi ke-30000

Pada hasil matriks di baris pertama dan kolom pertama bernilai 47, berarti dari 60 *image testing* kelas 0 atau *badminton* hanya sebanyak 47 *image* yang benar-benar dikenali sebagai olahraga *badminton* atau hanya sebesar 78.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk

prosentase. Sisanya berjumlah 13 *image* tersebar di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 0 atau *badminton* tidak menampilkan raket *badminton* dan atau kok *badminton* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *badminton*, para pemain hanya menampilkan bagian punggung mereka, serta pencahayaan pada lapangan *badminton* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris kedua dan kolom kedua bernilai 33, berarti dari 60 *image testing* kelas 1 atau *bocce* hanya sebanyak 33 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 55% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 27 *image* tersebar di kelas 0 atau *badminton*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 1 atau *bocce* tidak menampilkan bola *bocce* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *bocce*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *bocce* yang kurang terang atau malah terlalu terang, serta terdapat balok kayu yang biasa digunakan pada olahraga *croquet* atau *polo*.

Pada hasil matriks di baris ketiga dan kolom ketiga bernilai 51, berarti dari 60 *image testing* kelas 2 atau *croquet* hanya sebanyak 51 *image* yang benar-benar dikenali sebagai olahraga *croquet* atau hanya sebesar 85% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 9 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 3 atau *polo*, dan di kelas 4 atau *rockclimbing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 2 atau *croquet* tidak menampilkan bola *croquet* dan atau tongkat *croquet* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *croquet*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *croquet* yang kurang terang atau malah terlalu terang.

Pada hasil matriks di baris keempat dan kolom keempat bernilai 47, berarti dari 60 *image testing* kelas 3 atau *polo* hanya sebanyak 47 *image* yang benar-benar dikenali sebagai olahraga *polo* atau hanya sebesar 78.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 13 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 4 atau *rockclimbing*, dan di kelas 5 atau *rowing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 3 atau *polo* tidak menampilkan bola *polo* dan atau tongkat *polo* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan pada arena *polo* yang kurang terang atau malah terlalu terang, serta para pemain dan kuda mereka terlihat tidak jelas karena posisi yang terlalu jauh dari kamera.

Pada hasil matriks di baris kelima dan kolom kelima bernilai 52, berarti dari 60 *image testing* kelas 4 atau *rockclimbing* hanya sebanyak 52 *image* yang benar-benar dikenali sebagai olahraga *rockclimbing* atau hanya sebesar 86.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 8 *image* tersebar di kelas 3 atau *polo*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 4 atau *rockclimbing* tidak menampilkan gerakan memanjat dengan jelas, jumlah para pemanjat yang terlalu banyak, pencahayaan pada *image* yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemanjatnya.

Pada hasil matriks di baris keenam dan kolom keenam bernilai 44, berarti dari 60 *image testing* kelas 5 atau *rowing* hanya sebanyak 44 *image* yang benar-benar dikenali sebagai olahraga *rowing* atau hanya sebesar 73.3% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 16 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 2 atau *croquet*, di kelas 3 atau *polo*, di kelas 4 atau *rockclimbing*, di kelas 6 atau *sailing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 5 atau *rowing* tidak menampilkan perahu dengan jelas sehingga terlihat seperti garis pada net

*badminton*, tidak menampilkan gerakan bermain *rowing* dengan jelas seperti terlalu membungkuk seperti pada olahraga *bocce* dan *polo*, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit lebih banyak dari pada pemain yang mendayung dengan perahunya.

Pada hasil matriks di baris ketujuh dan kolom ketujuh bernilai 54, berarti dari 60 *image testing* kelas 6 atau *sailing* hanya sebanyak 54 *image* yang benar-benar dikenali sebagai olahraga *bocce* atau hanya sebesar 90% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 6 *image* tersebar di kelas 0 atau *badminton*, di kelas 4 atau *rockclimbing*, di kelas 5 atau *rowing*, dan di kelas 7 atau *snowboarding*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 6 atau *sailing* menampilkan layar perahu-perahu yang berderet dengan jarak yang cukup jauh dari kamera sehingga terkesan seperti net pada arena *badminton*, para pemain menampilkan gerakan yang lebih menyerupai *rowing*, pencahayaan yang kurang terang atau malah terlalu terang, serta porsi yang ditampilkan dari langit biru dan layar perahu yang berwarna putih seperti area yang bersalju.

Pada hasil matriks di baris kedelapan dan kolom kedelapan bernilai 52, berarti dari 60 *image testing* kelas 7 atau *snowboarding* hanya sebanyak 52 *image* yang benar-benar dikenali sebagai olahraga *snowboarding* atau hanya sebesar 86.7% seperti yang tertera pada hasil klasifikasi 8 kelas dengan akurasi dalam bentuk prosentase. Sisanya berjumlah 8 *image* tersebar di kelas 0 atau *badminton*, di kelas 1 atau *bocce*, di kelas 4 atau *rockclimbing*, dan di kelas 5 atau *rowing*. Kesalahan pengklasifikasian yang terjadi disebabkan beberapa *image testing* kelas 7 atau *snowboarding* tidak menampilkan papan yang digunakan pada *snowboarding* dengan sempurna karena tertutupi oleh tubuh pemain, para pemain tidak menampilkan gerakan bermain *snowboarding* dengan jelas, para pemain hanya menampilkan bagian punggung mereka, pencahayaan yang kurang terang atau malah terlalu terang, porsi yang ditampilkan langit biru terlalu mendominasi seperti warna air.



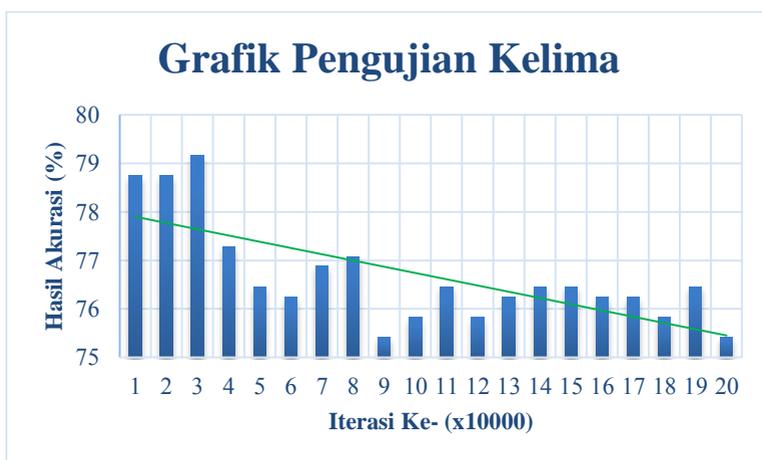
**Gambar 4.17** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image training*

**Tabel 4.5** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image testing*

Iterasi Ke-	Hasil Akurasi (%)
10000	78.75000000
20000	78.75000000
30000	79.16666667
40000	77.29166667
50000	76.45833333
60000	76.25000000
70000	76.87500000
80000	77.08333333
90000	75.41666667
100000	75.83333333
110000	76.45833333

**Tabel 4.5** Hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image testing* (lanjutan)

Iterasi Ke-	Hasil Akurasi (%)
120000	75.83333333
130000	76.25000000
140000	76.45833333
150000	76.45833333
160000	76.25000000
170000	76.25000000
180000	75.83333333
190000	76.45833333
200000	75.41666667



**Gambar 4.18** Grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image testing*

Gambar 4.17 adalah grafik hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image training*. Dapat dilihat bahwa hasil yang ditampilkan oleh grafik dari iterasi 10000 sampai iterasi 200000 semuanya memiliki akurasi 100%. Hal ini terjadi karena *overfitting*. *Overfitting* adalah keadaan dimana sistem yang dibangun cenderung “menyimpan memori” data *training* daripada melakukan proses pembelajaran atau *training*. Dalam sistem *convolutional neural network* (CNN) yang dibangun sudah diterapkan salah satu cara untuk mencegah terjadinya keadaan *overfitting*, yaitu dengan menggunakan *dropout* pada *dropout layer*. Ternyata *dropout* yang telah diaplikasikan pada sistem *convolutional neural network* (CNN) melalui *dropout layer* tidak dapat mencegah terjadinya *overfitting*. Setelah ditelusuri, perihal utama yang menyebabkan terjadinya keadaan *overfitting* ini adalah data training yang terlalu sedikit.

Tabel 4.5 dan gambar 4.18 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian kelima dengan masukan *image testing*. Dari data akurasi rata-rata yang ada didalam tabel 4.5 tersebut dibuat grafiknya pada gambar 4.18, dimana dapat dilihat bahwa bentuk grafik tersebut seperti pada grafik pengujian pertama yaitu memiliki kecenderungan menurun yang seharusnya grafik memiliki kecenderungan naik. Hal ini merupakan salah satu akibat dari keadaan sistem *convolutional neural network* (CNN) yang *overfitting*, sehingga grafik yang berkecenderungan menurun memiliki kemungkinan besar untuk terjadi.

#### 4.6 Seluruh Pengujian

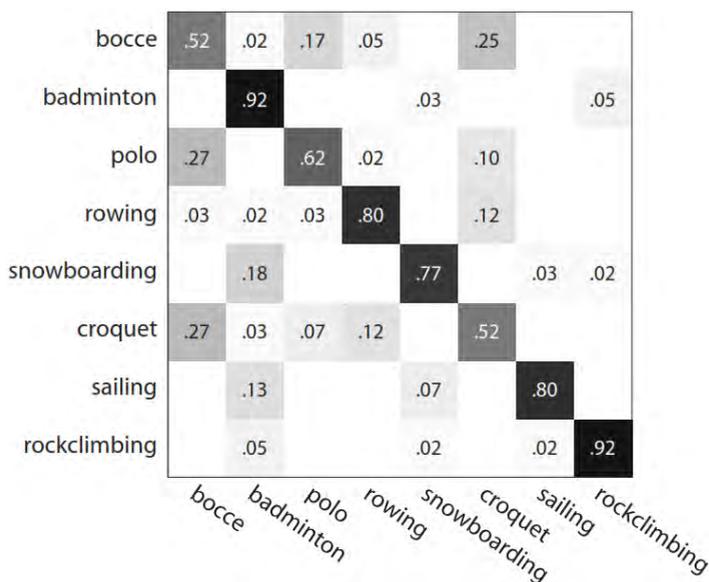
Tabel 4.6 adalah hasil akurasi rata-rata seluruh kelas setiap iterasi 10000 pada pengujian pertama, pengujian kedua, pengujian ketiga, pengujian keempat, dan pengujian kelima. Lalu masing-masing pengujian dihitung nilai rata-ratanya. Selanjutnya hasil akurasi rata-rata seluruh pengujian dijumlahkan dan dibagi 5 untuk mendapatkan hasil kinerja rata-rata dari sistem yang dibangun. Sistem *convolutional neural network* (CNN) yang dibangun dengan menggunakan masukan dataset UIUC *sport event image* 8 kelas memiliki hasil kinerja rata-rata 74.673%.

Gambar 4.19 adalah hasil pengujian yang dilakukan oleh Li-Jia Li dan Li Fei-Fei menggunakan dataset UIUC *sport event image* 8 kelas. Pada hasil klasifikasi 8 kelas dengan akurasi tersebut, baris

**Tabel 4.6** Hasil akurasi rata-rata seluruh kelas pada pengujian pertama hingga kelima dengan masukan *image testing* dalam bentuk prosentase

Iterasi Ke-	Akurasi Pengujian 1 (%)	Akurasi Pengujian 2 (%)	Akurasi Pengujian 3 (%)	Akurasi Pengujian 4 (%)	Akurasi Pengujian 5 (%)
10000	72.500000	68.541667	79.375000	78.958333	78.750000
20000	72.708333	69.791667	79.375000	78.125000	78.750000
30000	71.458333	68.125000	79.166667	77.291667	79.166667
40000	72.500000	68.125000	80.416667	78.125000	77.291667
50000	72.916667	68.125000	80.000000	78.333333	76.458333
60000	73.125000	67.916667	80.000000	78.333333	76.250000
70000	72.708333	68.750000	80.208333	78.750000	76.875000
80000	72.083333	68.958333	81.041667	77.500000	77.083333
90000	71.458333	68.541667	81.666667	76.458333	75.416667
100000	71.458333	68.333333	80.833333	78.333333	75.833333
110000	70.416667	67.916667	80.208333	78.125000	76.458333
120000	71.250000	68.333333	80.833333	77.708333	75.833333
130000	71.458333	68.333333	80.416667	71.250000	76.250000
140000	70.833333	68.541667	80.416667	78.541667	76.458333
150000	70.416667	68.750000	80.625000	78.333333	76.458333
160000	70.833333	68.541667	80.000000	78.541667	76.250000
170000	70.416667	68.750000	80.625000	78.958333	76.250000
180000	70.000000	69.791667	80.833333	79.375000	75.833333
190000	71.041667	69.583333	80.625000	61.458333	76.458333
200000	70.000000	69.791667	80.625000	60.833333	75.416667
<b>Total Akurasi Rata-rata Masing-masing Pengujian</b>	71.479167	68.677083	80.364583	76.166667	76.677083
<b>Total Akurasi Rata-rata Seluruh Pengujian</b>	74.67292				

pertama dan kolom pertama merepresentasikan nilai akurasi dari kelas 0 atau *bocce*. Baris kedua dan kolom kedua merepresentasikan nilai akurasi dari kelas 1 atau *badminton*. Baris ketiga dan kolom ketiga merepresentasikan nilai akurasi dari kelas 2 atau *polo*. Baris keempat dan kolom keempat merepresentasikan nilai akurasi dari kelas 3 atau *rowing*. Baris kelima dan kolom kelima merepresentasikan nilai akurasi dari kelas 4 atau *snowboarding*. Baris keenam dan kolom keenam merepresentasikan nilai akurasi dari kelas 5 atau *croquet*. Baris ketujuh dan kolom ketujuh merepresentasikan nilai akurasi dari kelas 6 atau *sailing*. Baris kedelapan dan kolom kedelapan merepresentasikan nilai akurasi dari kelas 7 atau *rockclimbing*. Selanjutnya seluruh akurasi pada masing-masing kelas dijumlahkan dan dibagi dengan 8 untuk mendapatkan hasil kinerja rata-rata dari pengujian yang telah dilakukan oleh Li-Jia Li dan Li Fei-Fei, sehingga hasil kinerja rata-rata yang didapatkan dari sistem yang dibangun oleh mereka adalah 73.375%.



**Gambar 4.19** Hasil klasifikasi 8 kelas dengan akurasi yang dilakukan pengujiannya oleh Li-Jia Li dan Li Fei-Fei[1]

## BAB V PENUTUP

### 5.1 Kesimpulan

Proses *training* menggunakan sistem *convolutional neural network* (CNN) dengan masukan  $70 \text{ image} \times 8$  kelas dan total nilai iterasi 200000 yang diimplementasikan pada GPU NVIDIA Tesla s2050 membutuhkan waktu rata-rata 144 jam atau 6 hari untuk menyelesaikannya hingga mendapatkan model *training* yang dibutuhkan. Sedangkan proses *testing* dengan masukan  $60 \text{ image} \times 8$  kelas dan model *training* dari sistem *convolutional neural network* (CNN) membutuhkan waktu rata-rata 5 menit.

Grafik yang didapatkan dari iterasi 10000 sampai 200000 seharusnya memiliki kecenderungan naik dengan akurasi yang tinggi seperti ditunjukkan pada pengujian ketiga dan keempat. Tetapi grafik hasil pengujian pertama dan pengujian kelima memiliki kecenderungan turun, serta grafik hasil pengujian kedua walaupun memiliki kecenderungan naik hanya memiliki akurasi rata-rata yang rendah yaitu 68.677%. Hal ini terjadi akibat dari keadaan *overfitting* pada sistem *convolutional neural network* (CNN) yang dibangun walaupun sudah diterapkan *dropout layer* untuk mencegahnya, seperti ditunjukkan pada grafik-grafik yang menggunakan data masukan berupa *image training*, dimana semuanya memiliki akurasi 100%.

Pada pengujian keempat, saat iterasi ke-180760, iterasi ke-180760, dan iterasi ke-180780 terjadi keanehan dimana nilai *loss* yang didapatkan mengalami perubahan yang semakin besar. Perubahan terjadi dari nilai *loss* 0.000971931 (iterasi ke-180760) menjadi 0.218229 (iterasi ke-180780) lalu sampai puncaknya menjadi 2.0837 (iterasi ke-180800). Nilai *loss* pada iterasi ke-180800 itu memiliki nilai yang menyerupai dengan nilai *loss* pada iterasi ke-0 sampai dengan iterasi ke-40. Selanjutnya nilai *loss* pada iterasi ke-180800 tersebut berkurang sedikit demi sedikit sampai pada iterasi ke-184040. Pada iterasi ke-184040 sampai iterasi ke-200000 terjadi keanehan lagi dimana nilai *loss* tidak berkurang, melainkan terjadi keadaan osilasi dengan nilai *loss* berada antara 0.38 sampai 0.7. Hal ini diduga disebabkan juga karena sistem yang dibangun memiliki keadaan *overfitting* serta *image training* yang

belum pernah muncul sebelum iterasi ke-180800, tiba-tiba muncul pada iterasi tersebut.

Berdasarkan pengujian pertama, kedua, ketiga, keempat, dan kelima dapat disimpulkan hasil pengenalan aktifitas olahraga manusia menggunakan sistem *convolutional neural network* (CNN) yang dibangun menghasilkan nilai akurasi rata-rata sebesar 74.673%. Hasil tersebut juga lebih baik daripada pengujian yang dilakukan oleh Li-Jia Li dan Li Fei-Fei dengan akurasi rata-rata yang didapatkan adalah 73.375%.

## 5.2 Saran

Nilai akurasi rata-rata sebesar 74.673% yang didapatkan dari sistem *convolutional neural network* (CNN) yang dibangun dengan menggunakan dataset UIUC *sport event image* 8 kelas (kelas 0 atau *badminton*, kelas 1 atau *bocce*, kelas 2 atau *croquet*, kelas 3 atau *polo*, kelas 4 atau *rockclimbing*, kelas 5 atau *rowing*, kelas 6 atau *sailing*, kelas 7 atau *snowboarding*) bisa ditingkatkan dengan melakukan perubahan terhadap parameter-parameternya. Keadaan *overfitting* yang terjadi dapat dilakukan pencegahan dengan cara yang lain yaitu memperbanyak *data training*, dimana pada setiap *image training* dapat dibagi menjadi 4 bagian.

Implementasi dari sistem *convolutional neural network* (CNN) dilakukan pada GPU NVIDIA Tesla s2050 yang memerlukan waktu rata-rata 144 jam atau 6 hari untuk mendapatkan model *training* yang dibutuhkan, sehingga perlu diimplementasikan pada GPU yang lebih baik agar waktu yang dibutuhkan untuk mendapatkan model *training* lebih cepat.

## DAFTAR PUSTAKA

- [1] Li-Jia Li and Li Fei-Fei, "What, Where and Who? Classifying Events by Scene and Object Recognition", University of Illinois at Urbana-Champaign and Princeton University, USA, 2007
- [2] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., "Caffe: Convolutional Architecture for Fast Feature Embedding", University of Berkeley, Berkeley, 2014
- [3] Lin, M., Chen, Q., YanScherer, S., "Network in Network", National University of Singapore, Singapore, Maret, 2014
- [4] Jaswal, D., Sowmya.V, K.P.Soman, "Image Classification Using Convolutional Neural Networks", In International Journal of Advancements in Research & Technology, Volume 3, Issue 6, June 2014
- [5] Lisa Lab, "Convolutional Neural Networks (LeNet)" <URL:<http://deeplearning.net/tutorial/lenet.html> >, November, 2014
- [6] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., Schmidhuber, J., "Flexible, High Performance Convolutional Neural Networks for Image Classification", In Intl. Joint Conference on Artificial Intelligence IJCAI, pages 1237–1242, 2011
- [7] Simon, Phil, "Too Big to Ignore: The Business Case for Big Data", Wiley, pg. 89, 18 Maret 2013
- [8] Deng, L. dan Yu, D., "Deep Learning: Methods and Applications", Foundations and Trends, Redmond, 2014
- [9] Siang, Jong Jek, "*Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*", Andi Offset, Yogyakarta, 2005
- [10] Kusumadewi, Sri, "*Membangun Jaringan Syaraf Tiruan dengan Matlab dan Excel Link*", Graha Ilmu, Yogyakarta, 2004
- [11] Purnomo, Mauridhi Heri dan Kurniawan, Agus, "Supervised Neural Networks dan Aplikasinya", Graha Ilmu, Yogyakarta, 2006
- [12] Bottou, L., "Stochastic Gradient Descent Tricks", Neural Networks: Tricks of the Trade: Springer, Microsoft, Washington, 2012
- [13] Amutiara, "Konvolusi dan Transformasi Fourier", Universitas Gunadarma, 2004

- [14] Nair, Vinod dan Hinton, Geoffrey E., “Rectified Linear Units Improve Restricted Boltzmann Machines”, In Proc. 27<sup>th</sup> International Conference on Machine Learning, 2010
- [15] Krizhevsky, A., Sutskever, I., Hinton, G.E., “ImageNet Classification with Deep Convolutional Neural Networks”, University of Toronto, In Advances in Neural Information Processing Systems (NIPS), 2012
- [16] Scherer, D., Muller, A., dan Behnke, S., “Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition”, In 20th International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece, September, 2010
- [17] Lawrence, S., Lee Giles, C., Chung Tsoi, A., “Lessons in Neural Network Training: Overfitting May be Harder than Expected”, University of Wollongong, Australia, In Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, Menlo Park, California, pp. 540–545, 1997
- [18] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, University of Toronto, In Journal of Machine Learning Research 15, 2014
- [19] “Asian Games Sports”  
<URL:<http://www.ocasia.org/game/GameSportsFed.aspx?dPbeBiz365g=>>, November, 2014
- [20] “Asal Mula 8 Jam Kerja Dan Mengapa Kita Harus Mempertimbangkannya Kembali”  
<URL:<http://www.tahupedia.com/content/show/415/Asal-Mula-8-Jam-Kerja-Dan-Mengapa-Kita-Harus-Mempertimbangkannya-Kembali>>, November, 2014

## BIODATA PENULIS



Penulis yang memiliki nama lengkap Bagus Prabandaru ini lahir di kota Kediri pada tanggal 21 Agustus 1989. Penulis bertempat tinggal di Jalan Melati 1 No. 15 Perumnas Ngronggo kota Kediri 64127. Pada tahun 1995 – 2001, penulis mengenyam bangku pendidikan di SDN Ngronggo VIII Kediri dan melanjutkan pendidikan di SMPN 1 Kediri pada tahun 2001 – 2004. Penulis meneruskan studi Sekolah Menengah Atas Negeri di SMAN 3 Kediri pada tahun 2004 – 2007, saat itu penulis aktif dalam berbagai kegiatan organisasi dan olahraga seperti pramuka, palang merah remaja, dan pencak silat. Pada tahun 2007 – 2010, penulis melanjutkan studi di Politeknik Negeri Malang (POLINEMA), saat itu penulis aktif dalam organisasi workshop elektro. Penulis sedang mengenyam pendidikan di Institut Teknologi Sepuluh Nopember Surabaya di Jurusan Teknik Elektro sejak tahun 2010 sampai saat ini. Penulis yang memiliki hobi bermain game dan menonton film ini juga memiliki keingintahuan yang besar dalam berbagai macam ilmu pengetahuan.