

## BAB 4

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Pemilihan Metode DL

Memilih metode DL yang tepat perlu melewati beberapa tahap terlebih dahulu. Hal ini karena metode DL dipilih berdasarkan tugas yang nantinya harus dikerjakan. Sehingga, perlu mengetahui fokus permasalahan yang akan diselesaikan, kemudian merumuskan ide media pembelajaran apa yang cocok untuk digunakan. Setelah menentukan karakteristik media pembelajaran, baru lah tugas DL ditentukan. Hasil penentuan tugas DL inilah yang menjadi pertimbangan untuk memilih metode DL yang sesuai. Berikut adalah penjelasan dari setiap langkah untuk menentukan metode DL yang digunakan pada penelitian ini.

##### 4.1.1 Deskripsi Data

Penelitian dilakukan di Program Studi Pendidikan Teknologi dan Kejuruan Sekolah Pascasarjana UPI. Model komputasi ini dibangun sebagai jaringan penyusun media pembelajaran berbasis DL. Model komputasi yang dibuat terdiri dari beberapa bagian yakni, proses *input*, pembuatan jaringan CNN, dan *output* yang dihasilkan. *Input* data yang digunakan pada model komputasi ini adalah data berupa gambar 2D yang dibuat menggunakan *inventor*.

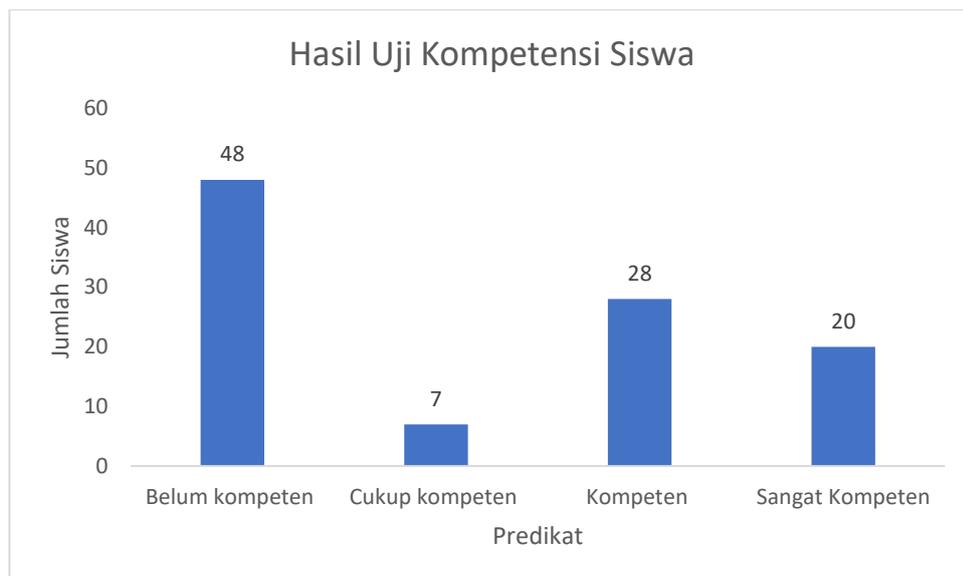
Gambar dibuat sebanyak 720 buah yang terdiri dari 420 gambar *extrude* dan 300 gambar *revolve* (Lampiran 4). Dimensi seluruh gambar dibuat sama yakni sebesar 100 X 100. Hasil dari model komputasi ini adalah membuat sistem yang mampu mengklasifikasikan perintah pembuatan gambar di *inventor*.

##### 4.1.2 Studi Pendahuluan dan Perumusan Masalah

Mengumpulkan data awal untuk mengungkapkan permasalahan yang terjadi di lapangan dilakukan dengan studi pendahuluan. Berdasarkan permasalahan yang terkumpul, dipilih masalah yang menarik untuk diangkat dan dicari solusinya dalam sebuah penelitian. Berikut adalah hal-hal yang peneliti dapatkan dan angkat pada penelitian ini.

### 1) Studi Lapangan

Studi pendahuluan dilakukan dengan melakukan wawancara dan observasi lapangan menggunakan sampel sebanyak satu sekolah. Wawancara dilakukan kepada siswa dan juga guru mata pelajaran TMDC sedangkan observasi dilakukan oleh peneliti untuk melihat lingkungan dan fasilitas belajar siswa di sekolah. Selain itu, peneliti juga menyebarkan angket dalam bentuk google form untuk melihat motivasi siswa dalam mempelajari CAD.



Gambar 4. 1 Hasil Uji Kompetensi Siswa

Berdasarkan hasil studi lapangan didapatkan data berupa nilai uji kompetensi siswa dalam mata pelajaran TMDC. Hasil uji kompetensi menunjukkan sebanyak 48 (47%) siswa dinyatakan “BELUM KOMPETEN” (Gambar 4.1). Data selanjutnya adalah kondisi lingkungan dan fasilitas belajar siswa. Data menunjukkan bahwa lingkungan belajar siswa di sekolah baik dan sesuai untuk melakukan pembelajaran. Namun, fasilitas sekolah (Tabel 4.2) dan fasilitas yang dimiliki siswa (Gambar 4.2) kurang mumpuni untuk melaksanakan pembelajaran CAD dengan baik. Hal tersebut terlihat dari peralatan yang dibutuhkan untuk melaksanakan pembelajaran TMDC belum terpenuhi dengan baik seperti, tidak adanya *printer* yang mampu mencetak gambar ukuran A4, alat gambar di lab tidak lengkap, kondisi dan jumlah komputer yang tidak memadai, *software* CAD yang kurang *upgrade*, hingga tidak adanya

buku sumber yang sesuai untuk siswa belajar mandiri. Selain itu, banyak siswa yang tidak memiliki laptop sehingga, proses pembelajaran CAD hanya bisa dilakukan di sekolah saja.

Tabel 4. 1  
Kondisi Lingkungan Sekolah

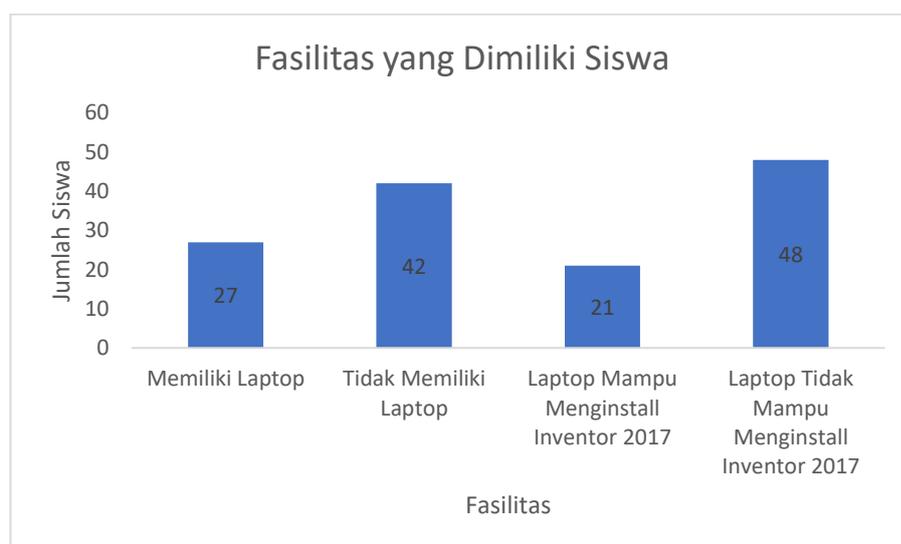
Aspek Penilaian	Indikator	Skor	
		Ya	Tidak
Lingkungan Fisik	Bangunan terawat dengan baik	√	
	Lingkungan sekolah bersih	√	
	Tanaman hijau ditanam/ditata di dalam lingkungan sekolah	√	
	Lingkungan sekolah jauh dari kebisingan		√
	Lokasi sekolah strategis	√	
Lingkungan Sosial	Metode mengajar yang digunakan oleh guru membuat siswa berpikir kritis		√
	Guru menggunakan metode ceramah dan demonstrasi	√	
	Metode mengajar yang digunakan guru membuat siswa aktif dan inisiatif untuk menggali materi lebih dalam lagi		√
	Dalam proses pembelajaran siswa dapat menyelesaikan gambarnya sendiri tanpa bertanya		√
	Lingkungan belajar siswa cukup kondusif	√	
	Guru dapat mengontrol dan membimbing siswa secara penuh satu per satu		√
	Dalam 8 jam pelajaran siswa dapat menyelesaikan gambarnya dengan baik, lengkap, dan sesuai dengan aturan gambar		√
	Guru membagi 2 sesi pembelajaran di Lab untuk satu kelas	√	
	Siswa lebih senang bertanya kepada temannya jika ada kesulitan	√	
	Siswa saling membantu untuk menyelesaikan gambarnya	√	
Siswa senang bertanya kepada guru jika mengalami kesulitan		√	

Aspek Penilaian	Indikator	Skor	
		Ya	Tidak
	Siswa takut/malu untuk bertanya kepada Guru	√	
Budaya Sekolah	Seluruh siswa hadir tepat waktu di Lab/sekolah		√
	Seluruh siswa memenuhi SOP ketika di Lab	√	

Tabel 4. 2  
Fasilitas Sekolah

Aspek Penilaian	Indikator	Skor	
		Ya	Tidak
Ruang Kelas	Ruang kelas terawat dengan baik	√	
	Ruang kelas bersih	√	
	Peralatan pembelajaran di ruang kelas lengkap		√
	Sekolah sigap memperbaiki kerusakan kelas	√	
Alat Bantu Mengajar	Papan tulis memiliki ukuran yang sesuai dan dalam kondisi yang baik	√	
	Proyektor dan layar proyektor dalam kondisi yang baik	√	
	Printer terawat dengan baik dan dapat digunakan	√	
	Printer yang digunakan siswa dapat mencetak kertas hingga ukuran A3		√
	Terdapat alat gambar yang lengkap		√
Laboratorium	Terdapat laboratorium komputer khusus untuk mata pelajaran TMDC	√	
	Jumlah komputer sesuai dengan banyaknya siswa		√
	Komputer memiliki spek dan kondisi yang baik untuk digunakan		√
	Semua mouse yang ada di komputer siswa dalam kondisi yang baik dan bisa untuk digunakan dalam proses pembelajaran		√
	Memiliki setidaknya dua software/aplikasi untuk pembelajaran CAD	√	
	Software CAD yang digunakan merupakan versi terbaru		√

Aspek Penilaian	Indikator	Skor	
		Ya	Tidak
	Software yang dipelajari merupakan software terbaru di bidang CAD (Katia, Creo, dsb)		√
	AC di lab dalam kondisi yang baik, terawat, dan dapat digunakan	√	
Perpustakaan	Perpustakaan terawat dengan baik	√	
	Buku tertata dengan baik dan penempatan sesuai dengan bidang keahlian	√	
	Terdapat meja dan kursi untuk siswa membaca	√	
	Meja dan kursi di perpustakaan dapat digunakan oleh siswa untuk belajar		√
	Terdapat komputer yang dapat digunakan oleh siswa		√
Buku Pelajaran	Tersedia buku pelajaran khusus TMDC yang dapat dipinjam dan digunakan siswa		√
	Tersedia modul pembelajaran untuk siswa		√
Alat Tulis	Siswa memiliki alat tulis yang lengkap		√
	Siswa memiliki alat gambar yang lengkap sesuai standar dan seragam dengan siswa lain		√



Gambar 4. 2 Fasilitas Perangkat Siswa

Berkaitan dengan fasilitas sekolah dan fasilitas yang dimiliki siswa, motivasi siswa dalam mempelajari CAD pun menjadi rendah. Pasalnya seperti yang ditunjukkan pada Tabel 4.3, rata-rata motivasi siswa hanya mencapai 73% saja. Bahkan pada beberapa poin seperti pada Tabel 4.4 menunjukkan bahwa 92% jawaban siswa mereka lebih senang jika Guru mendemonstrasikan terlebih dahulu pembuatan gambar yang akan mereka buat sebelum mereka mencobanya sendiri, 67% menunjukkan bahwa mereka mempelajari CAD hanya sebagai tuntutan dan memenuhi kewajiban mata pelajaran sekolah saja, dan hanya 58% siswa yang rutin mempelajari atau berlatih CAD setiap harinya. Poin-poin tersebut menjadi salah satu faktor internal yang menjadi penyebab kesulitan yang dialami siswa ketika mempelajari CAD.

Tabel 4. 3

## Rata-rata Tingkat Motivasi Siswa

No.	Jumlah Responden	Tingkat Motivasi
1.	124 Siswa	73%
<b>Rata-rata Akhir Penilaian</b>		73%

Tabel 4. 4

## Persentase Motivasi per Indikator

NO	Indikator	Total	Persentase (%)
1	Saya menyimak dengan baik setiap materi yang disampaikan oleh guru	526	85
2	Saya mencatat setiap materi yang disampaikan oleh guru	458	74
3	Saya senang bertanya mengenai materi yang belum saya pahami	472	76
4	Saya senang jika guru mendemonstrasikan terlebih dulu apa yang akan saya kerjakan	568	92
5	Saya senang mempelajari gambar dengan CAD dibanding gambar manual	515	83

NO	Indikator	Total	Persentase (%)
6	Saya senang mencari dan mempelajari tutorial secara mandiri mengenai pembuatan suatu part di CAD melalui youtube atau blog	465	75
7	Saya tertarik untuk mempelajari berbagai jenis aplikasi CAD	513	83
8	Saya belajar CAD hanya untuk memenuhi kewajiban dan tuntutan mapel dari sekolah	416	67
9	Saya memahami seluruh materi mengenai CAD	420	68
10	Saya suka membuat catatan poin penting untuk pembeajaran CAD	450	73
11	Saya membuat jadwal khusus untuk belajar CAD	395	64
12	Saya tetap belajar CAD meskipun sulit	507	82
13	Saya lebih termotivasi belajar CAD setelah mengalami kegagalan	490	79
14	Saya mempunyai target belajar setiap minggunya untuk mempelajari CAD	447	72
15	Saya menghindari kebiasaan buruk yang akan menyebabkan saya malas untuk belajar CAD	482	78
16	Saya belajar CAD minimal 3 – 4 kali seminggu	376	61
17	Saya mengikuti kursus CAD	273	44
18	Saya belajar CAD setiap hari meski sedikit	267	59
19	Saya belajar CAD hanya jika ada tugas saja	451	73
20	Saya mengulang kembali apa yang sudah dipelajari di sekolah	448	72
21	Saya berlatih membuat gambar dengan CAD setiap hari	359	58
22	Saya bertanya mengenai materi yang sulit dipahami kepada guru	439	71
23	Saya bertanya mengenai materi yang sulit dipahami kepada teman	509	82
24	Saya mempelajari CAD karena ingin menjadi seorang drafter	450	73
25	Saya bercita-cita untuk bekerja di Industri manufaktur	469	76
26	Saya ingin menjadi pengusaha di bidang manufaktur	472	76
27	Saya ingin menjadi pengusaha di luar bidang manufaktur	503	81

Berdasarkan hasil studi pendahuluan yang telah dilakukan, berikut parameter yang sesuai untuk penelitian ini:

a) Pengguna

Media CAD berbasis DL ini ditujukan untuk siswa SMK kelas 12 akhir yang mengambil mata pelajaran TMDC. Berdasarkan hasil wawancara diketahui bahwa siswa mengalami kesulitan dalam mempelajari CAD. Kesulitan yang dihadapi oleh siswa diantaranya kesulitan dalam mempelajari teknologi baru, kebingungan untuk memilih *command* 3D yang tepat, serta *error* yang sulit untuk dipahami.

Kemampuan dasar gambar siswa yang kurang juga menjadi pemicu kesulitan yang mereka alami dalam mempelajari CAD. Hal-hal tersebut dibuktikan dengan banyaknya siswa yang dinyatakan belum kompeten pada uji kompetensi yang dilakukannya. Maka dari itu, media akan berfokus dalam membantu siswa memilih *command* 3D yang tepat dengan tujuan untuk meningkatkan literasi teknologi siswa.

b) Materi Pembelajaran

Materi pembelajaran yang dipilih pada media ini adalah materi mengoperasikan fungsi perintah pada gambar 3D sederhana. Materi ini berisi tentang bagaimana siswa memilih perintah 3D pada *software* CAD untuk menggambar 3D sederhana. Pada penelitian ini perintah utama yang ditampilkan hanya "*extrude*" dan juga "*revolve*". Nantinya, media akan membantu siswa untuk memilih perintah *extrude* atau *revolve* untuk membuat gambar yang mereka inginkan.

2) Studi Literatur

Pada studi pendahuluan studi literatur digunakan untuk mencari karakteristik perangkat lunak (*software*) yang sesuai untuk membuat media. Selain itu, disini juga ditampilkan mengenai spek perangkat keras (*hardware*) yang peneliti gunakan untuk membuat media pembelajaran.

a) Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan adalah sebagai berikut:

(1) *Python 3.9.0*

*Python* merupakan Bahasa pemrograman yang membantu programmer untuk mengintegrasikan dan membangun sistem dengan lebih cepat. Versi *python* yang digunakan pada penelitian ini adalah *Python 3.9.0*. Penelitian ini menggunakan jaringan CNN untuk membuat sistemnya. Ini juga menggunakan beberapa *library* milik *Python* seperti *keras*, *tensorflow*, *scikit-learn*, *open cv*, *matplotlib*, *numpy*, *imutils*, *argparse*, *random*, *pickle*, dan *os*.

(2) *Inventor 2017*

*Inventor* merupakan suatu aplikasi keteknikan yang berfungsi untuk membuat gambar 2D/3D maupun simulasi. Pada penelitian ini *Inventor* digunakan untuk membuat data training berupa gambar 3D. Gambar training dibuat dalam dua jenis yakni gambar yang dibuat dengan menggunakan perintah *extrude* dan gambar yang dibuat dengan menggunakan perintah *revolve*. [Lampiran 4](#) merupakan hasil modifikasi bentuk dasar dari yang paling sederhana hingga yang kompleks.

(3) *OS: WindowsXp7 32-bit*

Sistem yang digunakan untuk merancang multimedia pembelajaran pada mata pelajaran TMDC.

b) Kebutuhan Perangkat Keras (*hardware*)

Perangkat keras yang digunakan untuk merancang sistem komputasi ini adalah sebagai berikut:

*Prosesor* : *Ryzen 5*

*RAM* : 16 GB

*Harddisk* : 512 GB

*System* : 64-bit *Operating System*

Layar : 14"

**4.1.3 Perumusan Ide Media Pembelajaran**

Pada pembuatan media pembelajaran ini, Peneliti berfokus pada kesulitan siswa dalam memilih perintah yang tepat untuk membuat gambar 3D. Sehingga,

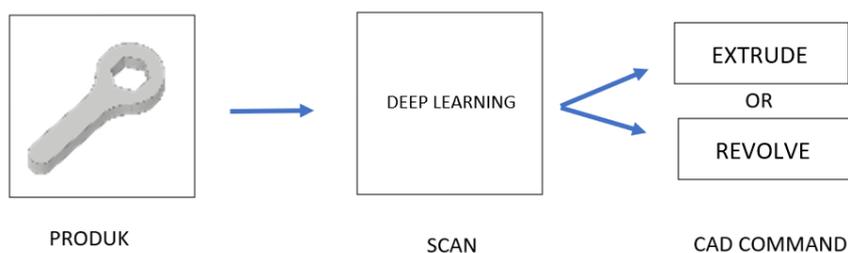
Nisa Aulia Saputra, 2023

**MODEL KOMPUTASI MEDIA PEMBELAJARAN COMPUTER-AIDED DESIGN BERBASIS DEEP LEARNING PADA MATERI MEMODIFIKASI GAMBAR 3D SEDERHANA**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

media akan dibuat mengikuti alur *reverse engineering* seperti yang terlihat pada [Gambar 4.3](#). Pemilihan *reverse engineering* sebagai tolak ukur ide pembelajaran dikarenakan sebagian besar gambar yang dibuat oleh siswa merupakan produk rill.

Pada pelaksanaannya, siswa akan diminta untuk mempersiapkan contoh gambar 3D yang akan dibuatnya, kemudian mereka mengupload gambar tersebut ke dalam sistem DL. Setelah siswa berhasil mengupload gambarnya pada sistem, mereka akan mendapatkan hasil berupa perintah (*extrude/revolve*) yang tepat untuk menggambar gambar tersebut. Selain itu, media pembelajaran yang dibuat diharapkan mampu meningkatkan literasi teknologi siswa. Maka dari itu, beberapa kompetensi yang harus dimiliki siswa terkait dengan peningkatan literasi teknologi dapat dilihat pada [Tabel 4.5](#).



Gambar 4. 3 *Reverse Engineering*

Tabel 4. 5  
Indikator Literasi Teknologi

No	Indikator Kemampuan Literasi Teknologi	Sub Indikator Kemampuan Literasi Teknologi
1	Konten	<ul style="list-style-type: none"> <li>- Kemampuan memahami piranti keras (hardware)</li> <li>- Kemampuan memahami piranti lunak (software)</li> </ul>
2	Proses	<ul style="list-style-type: none"> <li>- Kemampuan menggunakan piranti keras (hardware)</li> <li>- Kemampuan menggunakan piranti lunak (software)</li> </ul>
3	Konteks	Menggunakan konsep teknologi dalam pembelajaran
4	Sikap	<ul style="list-style-type: none"> <li>- Pemahaman yang baik dalam menanggapi informasi</li> <li>- Sikap terhadap perkembangan teknologi terhadap pembelajaran</li> </ul>

Sumber: (Suhendi et al., 2017)

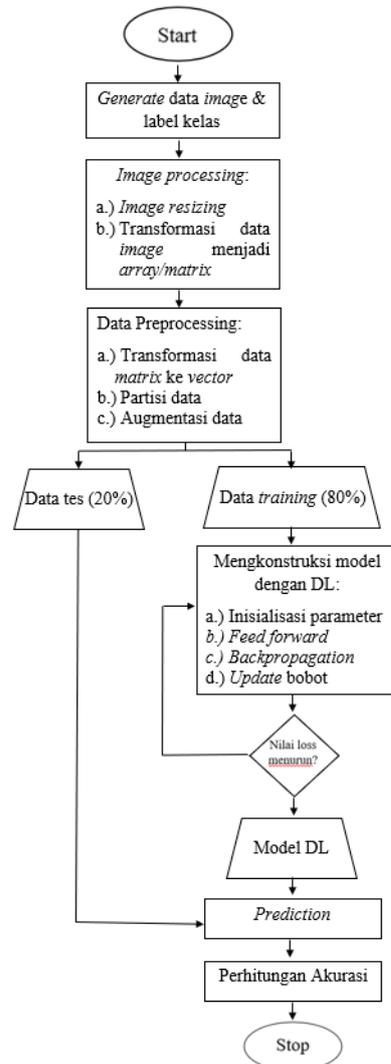
#### 4.1.4 Metode DL yang Dipilih untuk Membuat Model Komputasi

Metode DL yang digunakan pada pembuatan media ini adalah metode CNN. Berdasarkan perumusan masalah dan juga ide media pembelajaran, metode CNN dianggap cocok untuk menyelesaikan tugas yang mirip seperti alur *reverse engineering*. Metode CNN juga efektif dan mudah digunakan oleh seorang *programmer* pemula.

Hasil studi literatur pun menunjukkan bahwa metode ini sesuai untuk melakukan tugas *supervised learning* seperti pada kasus penelitian ini. Tugas *supervised learning* yang dilakukan berupa prediksi dua jenis kelas yakni, kelas *extrude* dan juga kelas *revolve*.

#### 4.2 Perancangan Model Komputasi

Perancangan model komputasi menggunakan DL memiliki beberapa tahap yakni tahap *preprocessing*, *training*, model, dan prediksi. [Gambar 4.4](#) menampilkan diagram alir yang menunjukkan tahapan tersebut.



Gambar 4. 4 Diagram Alir Perancangan Model Komputasi

Setiap tahap memiliki makna masing-masing dalam proses perancangan model. Perancangan model pada penelitian ini menggunakan metode CNN dengan arsitektur *smaller VGGNet*. Penjelasan lebih lengkap dari setiap tahap pada diagram alir ada pada setiap sub bab berikut.

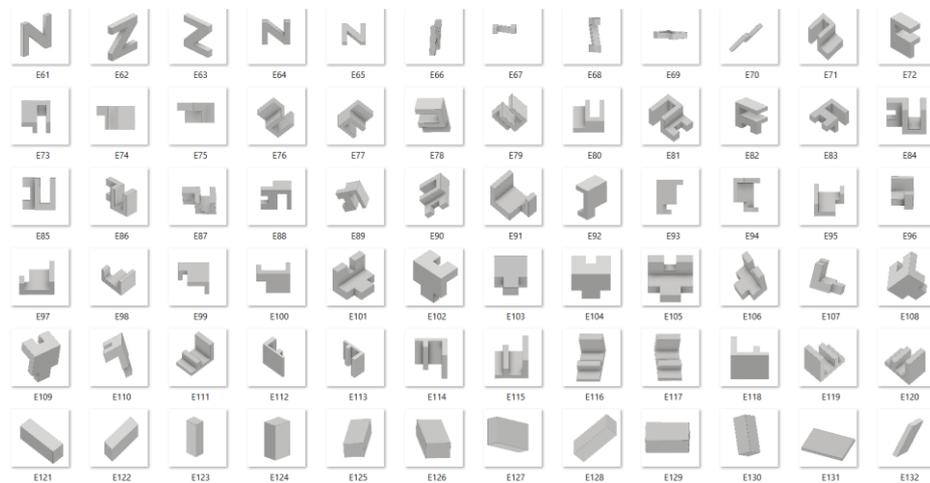
#### 4.2.1 *Generate Data Image dan Label Kelas*

Data pertama yang peneliti buat adalah data *extrude*. Data *extrude* merupakan data yang proses pemesinannya menggunakan proses *milling*. Data *extrude* terdiri dari 42 gambar dasar dengan ukuran beragam yang dimodifikasi masing-masing sebanyak 10 jenis sehingga, menghasilkan 420 data *extrude*.

Gambar 4.5 menunjukkan beberapa contoh data *extrude*. Data *extrude* diberi label E1, E2, dst.

Data kedua yang disiapkan adalah data *revolve*. Data *revolve* merupakan data yang proses pemesinannya menggunakan proses *turning*. Data ini terdiri dari 30 gambar dasar yang memiliki ukuran beragam dan dimodifikasi seperti proses modifikasi data *extrude*, sehingga pada kasus ini kami memiliki 300 data *revolve*.

Gambar 4.6 memperlihatkan sampel dari data *revolve*. Data *revolve* diberi label R1, R2, dst.



Gambar 4. 5 Dataset Extrude



Gambar 4. 6 Dataset Revolve

#### 4.2.2 Image Processing

*Image processing* dilakukan untuk mempersiapkan data *image* agar siap untuk diproses/dilatih. *Image processing* memiliki dua tahap yakni *image resizing*

dan transformasi data *image* menjadi *array*. Berikut adalah penjelasan mengenai tahapan *image processing*.

#### 4.2.2.1 *Image Resizing*

*Image dimension* dilakukan untuk membuat ukuran piksel pada data *training* maupun data *testing* seragam. Hal ini dilakukan agar proses *training* dapat dijalankan dengan baik. Selain itu, *resize image* juga berfungsi untuk menghindari terjadinya *miss* data pada saat *training* yang mampu menyebabkan nilai *loss* meningkat dan nilai akurasi menurun.

Pembuatan kode pada *image dimension* dilakukan dengan cara menunjukkan ukuran *pixel dataset* yang diinginkan, kemudian diikuti oleh jumlah warna dasar pada gambar yakni RGB (*red, green, blue*). Penulisan kode untuk RGB cukup ditulis dengan angka 3. Angka 3 menunjukkan jumlah warna dasar *image*. Setelah nilai dimensi ditentukan, inisialisasi variable data dan label juga dilakukan. Kode untuk *resize image* dan membuat variable adalah sebagai berikut.

```
IMAGE_DIMS = (96, 96, 3)
data = [ ]
labels = [ ]
```

#### 4.2.2.2 Transformasi Data *Image* Menjadi *Array*

Hal yang dilakukan setelah *resize image* adalah melakukan penyesuaian *dataset image* agar lebih mudah untuk dipelajari oleh DL. Pada kasus ini untuk mengatur ukuran gambar, mengubah data gambar menjadi *array* (lampiran 5), mengekstrak kelas, dan memperbaharui *list* label digunakan kode *loop*. Kode *loop* memudahkan peneliti agar tidak membuat perintah satu per satu sesuai banyaknya gambar seperti dapat dilihat di bawah ini.

```
# loop over the input images

for imagePath in imagePaths:
```

```

# load the image, pre-process it, and store it in the data list

image = cv2.imread(imagePath)

image = cv2.resize(image, (IMAGE_DIMS[1], IMAGE_DIMS[0]))

image = img_to_array(image)

data.append(image)

# extract the class label from the image path and update the
# labels list

label = imagePath.split(os.path.sep)[-2]

labels.append(label)

# scale the raw pixel intensities to the range [0, 1]

data = np.array(data, dtype="float") / 255.0

print (data)

labels = np.array(labels)

print("[INFO] data matrix: {:.2f}MB".format(
    data.nbytes / (1024 * 1000.0)))

```

### 4.2.3 Data Preprocessing

Data *preprocessing* merupakan tahapan untuk menyiapkan data *image* untuk proses *training* setelah *image processing* berhasil dilakukan. Beberapa hal yang dilakukan pada tahap ini adalah mentransformasikan data *array* menjadi data *vector* untuk dilakukan partisi data, dan juga melakukan augmentasi data. Langkah tersebut akan dijelaskan pada sub sub bab di bawah ini.

#### 4.2.3.1 Transformasi Data dan Mempartisi Data

Jenis kelas pada penelitian ini adalah *categorical*. Label pada data *categorical* dapat dibaca dengan baik dengan menggunakan *package*

*LabelBinarizer* pada *library scikit-learn*. Sehingga, kita dapat mengetahui jika pembelajaran dilakukan dengan baik maka label pada *output* akan memberikan jawaban kelas yang tepat untuk gambar yang diinputkan. Selanjutnya untuk meningkatkan akurasi *training* dilakukan partisi data. Partisi data merupakan pemisahan data *training* dan juga data *test* untuk keperluan iterasi DL. Partisi dilakukan dengan menggunakan perbandingan 80:20 dimana data *training* terdiri dari 80% *dataset* sedangkan 20% *dataset* berada pada data *test*. Berikut adalah kode yang digunakan untuk membaca label dan mempartisi data.

```
# binarize the labels

lb = LabelBinarizer()

labels = lb.fit_transform(labels)

# partition the data into training and testing splits using 80% of
# the data for training and the remaining 20% for testing
(trainX, testX, trainY, testY) = train_test_split(data,
                                                  labels, test_size=0.2, random_state=42)
```

#### 4.2.3.2 Augmentasi Data

Augmentasi data dilakukan untuk menghindari *overfitting* ketika proses *training*. *Overfitting* merupakan kegagalan dalam mempelajari data baru meskipun selama proses *training* sistem DL menunjukkan kinerja yang baik. Berikut adalah kode yang digunakan pada proses augmentasi.

```
# construct the image generator for data augmentation

aug = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
                          height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
                          horizontal_flip=True, fill_mode="nearest")
```

Augmentasi atau pengolahan data gambar yang diatur diantaranya *rotation range* yang berfungsi untuk merotasi gambar agar mendapatkan perspektif dari segala arah. Besar derajat rotasi yang dipilih adalah  $25^\circ$ . *Width\_shift\_range=0,1* digunakan untuk merubah posisi gambar searah dengan lebar gambar awal. Artinya di sini peneliti mengatur agar posisi maksimal gambar yang di rubah secara acak oleh sistem hanya sebesar 10% dari lebar awal gambar. *Height\_shift\_range=0,1* memiliki fungsi yang sama seperti *width\_shift\_range=0,1* namun acuan posisinya adalah tinggi awal gambar. *Shear\_rage=0,2* artinya sudut geser diatur sebesar  $0,2^\circ$  berlawanan arah jarum jam. *Zoom rage=0,2* menunjukkan angka intensitas pembesaran gambar sebesar 0,2. *Horizontal\_flip=True* menunjukkan nilai benar untuk gambar yang diputar horizontal. *Fill\_mode="nearest"* berfungsi untuk mengisi area yang kosong akibat pergeseran posisi sebesar 10% dengan piksel terdekat dan merentangkannya.

#### 4.2.4 Proses Pemodelan dan *Training*

Proses pemodelan dan *training* merupakan proses inti dari tugas prediksi DL. Model DL didapatkan dari proses *training* yang dilakukan. Proses *training* ini dilakukan dengan menggunakan metode CNN model *Smaller VGGNet*. Arsitekturnya memiliki 3 *layer convolutional*. Setiap *layernya* memiliki tugas untuk melakukan iterasi. Berikut akan dijelaskan proses *training* hingga didapatkan model DL.

##### 4.2.4.1 Inisialisasi Data dan Parameter

Proses inisialisasi parameter dilakukan untuk mendefinisikan dan menyesuaikan parameter awal yang akan digunakan. Pada bagian ini ditentukan parameter *epoch*, *learning rate*, dan *batch size*. *Epoch* menunjukkan jumlah iterasi yang kita inginkan ketika proses training. *Batch size* menunjukkan banyaknya contoh pelatihan yang digunakan dalam satu kali proses iterasi (Rochmawati et al., 2021).

Perhitungan koreksi bobot ketika proses *training* ditentukan oleh nilai *learning rate*. Semakin besar nilai *learning rate* maka kemungkinan

proses pembelajaran yang kurang optimal cukup besar namun, semakin kecil nilai *learning rate* proses pembelajaran akan semakin lama untuk dilakukan (Rochmawati et al., 2021).

Proses pemodelannya sendiri diawali dengan membuat dataset untuk *input* proses. Dataset berisi sejumlah gambar yang dibuat menggunakan CAD dan akan diklasifikasikan. Dataset gambar diklasifikasikan ke dalam dua kelas, maka dari itu perlu membuat label untuk menamai masing-masing kelas. Selanjutnya, membuat partisi data untuk membagi gambar menjadi gambar *training* dan gambar *testing*.

Setelah semua proses *preprocessing* selesai dilakukan, maka selanjutnya membuat model jaringan CNN 3 layer. Jaringan CNN yang telah dibuat akan otomatis melakukan *training* data untuk menghasilkan model yang mampu mengklasifikasikan gambar ke dalam dua kelas yakni *extrude* dan *revolve*. Kode yang digunakan untuk menginisialisasi model dan mempersiapkan jaringan pelatihan adalah sebagai berikut.

```
EPOCHS = 300
```

```
INIT_LR = 1e-3
```

```
BS = 2
```

```
# initialize the model
```

```
print("[INFO] compiling model...")
```

```
model = SmallerVGGNet.build(width=IMAGE_DIMS[1],
height=IMAGE_DIMS[0],
```

```
depth=IMAGE_DIMS[2], classes=len(lb.classes_))
```

```
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
```

```
model.compile(loss="categorical_crossentropy", optimizer=opt,
```

```
metrics=["accuracy"])
```

```

# train the network

print("[INFO] training network...")

H = model.fit_generator(
    aug.flow(trainX, trainY, batch_size=BS),
    validation_data=(testX, testY),
    steps_per_epoch=len(trainX) // BS,
    epochs=EPOCHS, verbose=1)

# save the model to disk

print("[INFO] serializing network...")

model.save(args["model"])

# save the label binarizer to disk

print("[INFO] serializing label binarizer...")

f = open(args["labelbin"], "wb")

f.write(pickle.dumps(lb))

f.close()

```

#### 4.2.4.2 *Feed Forward*

Model CNN yang digunakan pada kasus ini adalah *Smaller VGGNet*. *Smaller VGGNet* dipilih karena memiliki ukuran *kernel* yang kecil yakni 3 x 3 di seluruh arsitekturnya. Masalah klasifikasi yang terjadi di luar area jaringan awalnya dilatih dapat digeneralisasi oleh *kernel Smaller VGGNet* ini. Metode *build* pada *Smaller VGGNet* memerlukan beberapa parameter seperti *width*, *height*, *depth*, dan *classes*. Parameter *width* berfungsi untuk menunjukkan lebar gambar dalam piksel, *height* menunjukkan tinggi gambar dalam piksel, *depth* menunjukkan jumlah *channels* untuk gambar, dan *classes* menunjukkan jumlah kelas yang harus

diprediksi oleh model. *Layer* yang digunakan untuk jaringan CNN ini sebanyak tiga lapis *convolutional layer* dan dua lapis *fully connected layer*. Kode dapat ditulis seperti di bawah ini.

```
class SmallerVGGNet:

    @staticmethod

    def build(width, height, depth, classes):

        # initialize the model along with the input shape to be

        # "channels last" and the channels dimension itself

        model = Sequential()

        inputShape = (height, width, depth)

        chanDim = -1

        # if we are using "channels first", update the input shape

        # and channels dimension

        if K.image_data_format() == "channels_first":

            inputShape = (depth, height, width)

            chanDim = 1
```

Setelah parameter ditetapkan, proses training pada CNN dimulai dengan memasuki layer pertama. Data yang diinputkan adalah data gambar *extrude/revolve* dengan ukuran 96 x 96 piksel. Proses pertama adalah *convolutional* yang berfungsi untuk mengekstrak fitur data gambar. Ukuran piksel gambar adalah 128 dengan ukuran *kernel* 3 x 3. *Padding "same"* digunakan pada *training* ini untuk mencegah hilangnya informasi piksel yang berada di area rawan (Gülgün & Erol, 2020). Fungsi aktivasi yang digunakan adalah fungsi ReLU. Jika tahap *convolutional* selesai dilakukan maka tahap selanjutnya adalah *pooling* dan *max pooling*. Ini dilakukan untuk

memperkecil ukuran gambar dengan maksud memudahkan dan membuat proses iterasi menjadi lebih detail. Ukuran *pooling* yang digunakan adalah 3 x 3. Fungsi *dropout* digunakan untuk mencegah *overfitting*. Kode untuk layer *convolutional* pertama dapat dituliskan sebagai berikut.

```
# CONV => RELU => POOL
    model.add(Conv2D(128, (3, 3), padding="same",
                    input_shape=inputShape))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3, 3)))
    model.add(Dropout(0.25))
```

Lapisan *convolutional* kedua dan ketiga memiliki cara kerja yang sama dengan lapisan *convolutional* pertama. Setiap lapisan hanya dibedakan dengan ukuran piksel gambar. Lapisan *convolutional* kedua dan ketiga masing-masing piksel gambar berukuran 64 dan 32. Kode untuk *convolutional layer* kedua dan ketiga dapat ditulis sebagai berikut.

```
# (CONV => RELU) * 2 => POOL
    model.add(Conv2D(64, (3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(64, (3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
```

```
# (CONV => RELU) * 2 => POOL
    model.add(Conv2D(32, (3, 3), padding="same"))
    model.add(Activation("relu"))
```

```

model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(32, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

```

Proses selanjutnya adalah *flattening*. *Flattening* sendiri merupakan proses konversi data *array* 2D hasil dari proses *pooling* menjadi data 1D *single vector*. *Layer* terakhir pada CNN yang berfungsi melakukan proses klasifikasi adalah *fully connected layer*. Penambahan *fully connected layer* ini menggunakan Dense 1024. Selanjutnya adalah tahap pengklasifikasian oleh fungsi aktivasi "*softmax*". Ia akan melakukan pengklasifikasian sesuai dengan jumlah kelas yang kita buat. Berikut adalah kode yang digunakan untuk proses *flattening* dan *fully connected layer*. Seluruh alur training pada CNN dapat dilihat pada [Gambar 4.7](#).

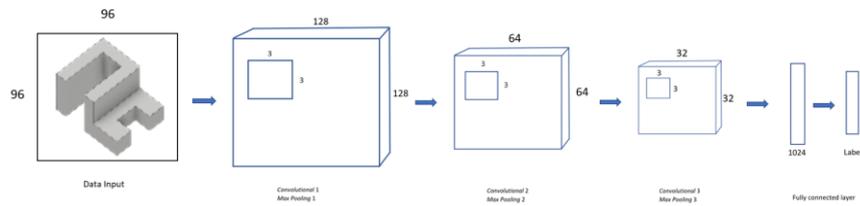
```

# first (and only) set of FC => RELU layers
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# softmax classifier
model.add(Dense(classes))
model.add(Activation("softmax"))

# return the constructed network architecture
return model

```



Gambar 4. 7 *Convolutional Neural Network*

#### 4.2.4.3 Skenario Eksperimen/ *Backpropagation*

Skenario eksperimen dilakukan untuk mendapatkan hasil *loss* dan akurasi yang sesuai. Skenario ini berdasarkan *trial and error* ketika menyusun kode. Peneliti melibatkan variable *epoch* dengan pasrtisi data 80:20, 60:40, dan 50:50 untuk dilakukan *trial and error*. Eksperimen untuk *epoch* ditampilkan pada [Table 4.6](#). Berdasarkan eksperimen yang dilakukan, peneliti menetapkan nilai epoch yang digunakan untuk training hingga menghasilkan model adalah 300 *epoch*, 150 *epoch*, dan 50 *epoch*. Nilai *epoch* maksimal yang diambil hanya 300 *epoch* karena menyesuaikan dengan kapasitas komputer yang digunakan untuk melakukan *training*. Tingginya *epoch* bisa membuat nilai akurasi menjadi lebih tinggi karena jaringan membuat jumlah variabel menjadi sekecil mungkin untuk dapat dioptimalkan dalam setiap proses trainingnya. Namun, kekurangannya adalah semakin banyak iterasi yang dilakukan maka membutuhkan waktu yang semakin lama pula untuk melakukan training (Neb et al., 2020).

Tabel 4. 6

Eksperimen Nilai Akurasi

No. Percobaan	Partisi data	Learning rate	Epoch	Akurasi?	Loss?	Waktu training?
1	80:20	0,001 (1e-3)	300	0,8650	0,2940	55:33,17
2				0,8750	0,3128	55:30,27
3	60:40	0,001 (1e-3)	150	0,7930	0,4400	27:47,22
4				0,7339	0,5038	27:30,02
5	50:50	0,001 (1e-3)	50	0,6350	0,6525	08:19,53
6				0,6646	0,6251	08:45,84
7	80:20	0,001 (1e-3)	150	0,7262	0,5184	35:49,83
8				0,7519	0,4830	38:56,35
9	60:40	0,001 (1e-3)	50	0,6209	0,7196	09:11,97
10				0,6348	0,6300	11:17,09
11	50:50	0,001 (1e-3)	300	0,8122	0,4048	52:33,89
12				0,8122	0,3903	51:11,29

#### 4.2.5 Model DL dan Prediksi dalam Bentuk *Web*

Model komputasi DL ini ditujukan sebagai alat bantu pembelajaran. Kita tahu bahwa pengguna model ini adalah siswa SMK jurusan Teknik gambar mesin, dimana fokus mereka bukan sebagai *programmer* tapi sebagai pengguna. Maka dari itu, kemudahan penggunaan model juga perlu diperhatikan. Kemudahan penggunaan model ini akan dibuat dalam bentuk *web*. Berikut merupakan kode yang digunakan untuk membuat model dalam bentuk *web*. Hasil dari *running* kode ini akan menghasilkan alamat *web* seperti yang terlihat pada [Gambar 4.8](#). Alamat *web* yang tertera selanjutnya di *copykan* pada *search engine* untuk dapat menampilkan *web* “Media Pembelajaran CAD Berbasis DL”. Tampilan Media Pembelajaran CAD Berbasis *Deep Learning* dapat dilihat pada [Gambar 4.9](#).

```
def detect():
```

```
    if request.method == 'POST':
```

```
        MODEL_PATH = "modeling.model"
```

```
        PICKLE_PATH = "lb.pickle"
```

```
        filestr = request.files['image'].read()
```

```
        npimg = np.fromstring(filestr, np.uint8)
```

```
        image = cv2.imdecode(npimg, cv2.IMREAD_COLOR)
```

```
        # pre-process the image for classification
```

```

image = cv2.resize(image, (96, 96))

image = image.astype("float") / 255.0

image = img_to_array(image)

image = np.expand_dims(image, axis=0)

# load the trained CNN and the label binarizer

print("[INFO] loading network...")

model = load_model(MODEL_PATH)

lb = pickle.loads(open(PICKLE_PATH, "rb").read())

# classify the input image

print("[INFO] classifying image...")

proba = model.predict(image)[0]

idx = np.argmax(proba)

label = lb.classes_[idx]

return jsonify(success=1, label=label, percent=(proba[idx] * 100))

return jsonify(error=1, message='Unsupported HTTP method')

@app.route('/')

def index():

    return render_template('index.html')

if __name__ == '__main__':

    app.run(debug=True)

```

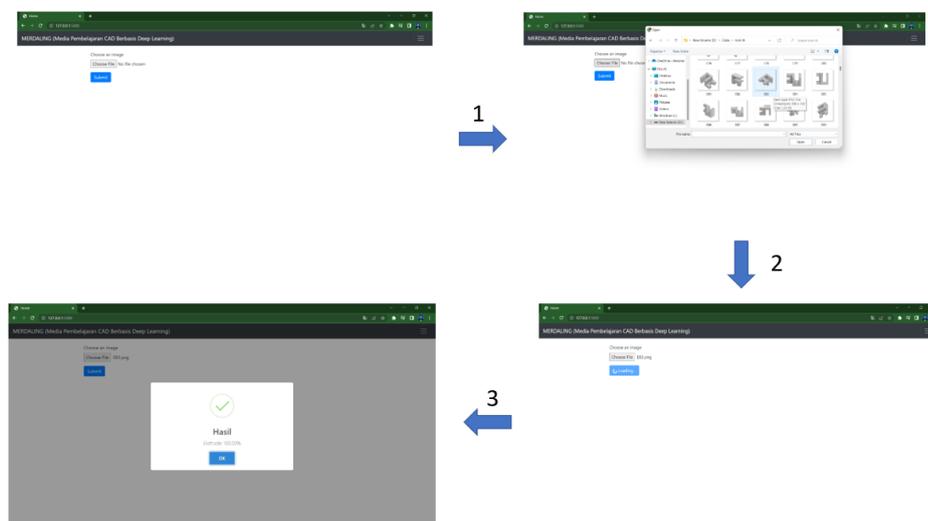
```

Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

D:\Thesis Nisa Aulia>app.py
2023-01-14 19:17:24.757284: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library
py 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-01-14 19:17:24.757516: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
2023-01-14 19:17:29.861748: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-01-14 19:17:29.861925: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.
* Debugger is active!
* Debugger PIN: 143-288-287
127.0.0.1 - - [14/Jan/2023 19:17:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2023 19:17:42] "GET /static/js/app.js HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2023 19:17:42] "GET /static/css/style.css HTTP/1.1" 200 -

```

Gambar 4. 8 Running Kode Aplikasi



Gambar 4. 9 Tampilan Media Pembelajaran CAD Berbasis Deep Learning

#### 4.2.6 Perhitungan Akurasi

Proses training yang telah dilakukan perlu dilihat akurasi dan *error*nya. Hal ini untuk mengetahui apakah iterasi dilakukan dengan baik oleh DL atau tidak. Hasil akurasi dan *loss* diplot dalam bentuk angka dan juga grafik menggunakan kode berikut.

```

# plot the training loss and accuracy

plt.style.use("ggplot")

plt.figure()

N = EPOCHS

```

Nisa Aulia Saputra, 2023

**MODEL KOMPUTASI MEDIA PEMBELAJARAN COMPUTER-AIDED DESIGN BERBASIS DEEP LEARNING PADA MATERI MEMODIFIKASI GAMBAR 3D SEDERHANA**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

```

plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")

plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")

plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")

plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")

plt.title("Training Loss and Accuracy")

plt.xlabel("Epoch #")

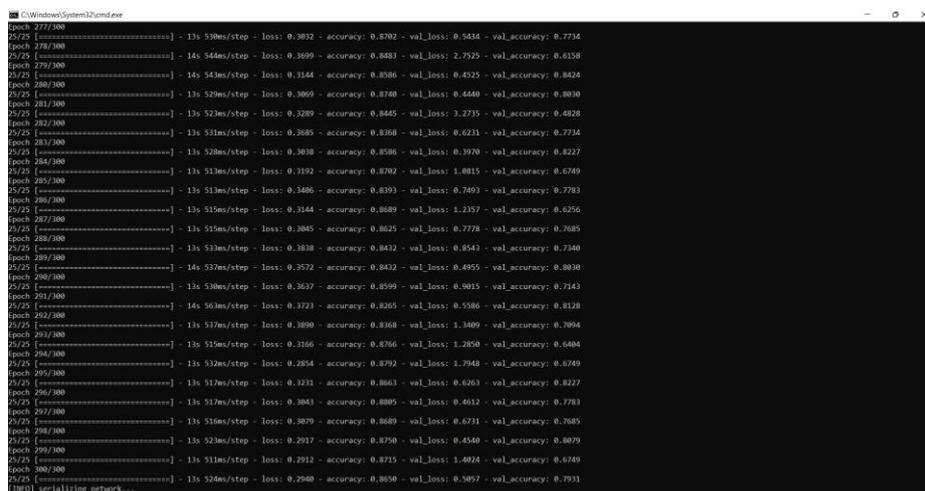
plt.ylabel("Loss/Accuracy")

plt.legend(loc="upper left")

plt.savefig(args["plot"])

```

Hasil *running* dari kode di atas dapat dilihat pada [Gambar 4.10](#). Proses iterasi dilakukan sebanyak 300 kali atau *epoch* 300. Setiap *epoch* memiliki hasil akurasi dan *loss* yang berbeda-beda. Berdasarkan hasil percobaan diketahui bahwa semakin banyak sistem melakukan iterasi, nilai akurasi semakin meningkat dan nilai *loss* semakin menurun. Artinya proses iterasi berhasil dilakukan dengan baik.



```

C:\Windows\System32\cmd.exe
epoch 277/300 ----- 13s 530ms/step - loss: 0.3832 - accuracy: 0.8702 - val_loss: 0.5434 - val_accuracy: 0.7734
epoch 278/300 ----- 14s 544ms/step - loss: 0.3899 - accuracy: 0.8683 - val_loss: 0.7525 - val_accuracy: 0.6158
epoch 279/300 ----- 14s 543ms/step - loss: 0.3144 - accuracy: 0.8586 - val_loss: 0.4525 - val_accuracy: 0.8424
epoch 280/300 ----- 13s 529ms/step - loss: 0.3869 - accuracy: 0.8748 - val_loss: 0.4440 - val_accuracy: 0.8030
epoch 281/300 ----- 13s 523ms/step - loss: 0.3289 - accuracy: 0.8445 - val_loss: 0.7235 - val_accuracy: 0.4828
epoch 282/300 ----- 13s 531ms/step - loss: 0.3685 - accuracy: 0.8368 - val_loss: 0.6231 - val_accuracy: 0.7734
epoch 283/300 ----- 13s 528ms/step - loss: 0.3838 - accuracy: 0.8586 - val_loss: 0.3970 - val_accuracy: 0.8227
epoch 284/300 ----- 13s 513ms/step - loss: 0.3192 - accuracy: 0.8702 - val_loss: 0.4015 - val_accuracy: 0.6749
epoch 285/300 ----- 13s 513ms/step - loss: 0.3486 - accuracy: 0.8393 - val_loss: 0.7493 - val_accuracy: 0.7783
epoch 286/300 ----- 13s 515ms/step - loss: 0.3144 - accuracy: 0.8689 - val_loss: 0.1257 - val_accuracy: 0.6256
epoch 287/300 ----- 13s 515ms/step - loss: 0.3845 - accuracy: 0.8625 - val_loss: 0.7778 - val_accuracy: 0.7685
epoch 288/300 ----- 13s 513ms/step - loss: 0.3838 - accuracy: 0.8432 - val_loss: 0.4543 - val_accuracy: 0.7340
epoch 289/300 ----- 14s 517ms/step - loss: 0.3572 - accuracy: 0.8432 - val_loss: 0.4955 - val_accuracy: 0.8930
epoch 290/300 ----- 13s 518ms/step - loss: 0.3637 - accuracy: 0.8599 - val_loss: 0.9015 - val_accuracy: 0.7143
epoch 291/300 ----- 14s 563ms/step - loss: 0.3123 - accuracy: 0.8265 - val_loss: 0.5586 - val_accuracy: 0.8128
epoch 292/300 ----- 13s 517ms/step - loss: 0.3890 - accuracy: 0.8368 - val_loss: 0.1409 - val_accuracy: 0.7994
epoch 293/300 ----- 13s 515ms/step - loss: 0.3166 - accuracy: 0.8706 - val_loss: 0.1850 - val_accuracy: 0.6484
epoch 294/300 ----- 13s 520ms/step - loss: 0.2854 - accuracy: 0.8792 - val_loss: 0.7948 - val_accuracy: 0.6749
epoch 295/300 ----- 13s 517ms/step - loss: 0.3231 - accuracy: 0.8663 - val_loss: 0.6263 - val_accuracy: 0.8227
epoch 296/300 ----- 13s 517ms/step - loss: 0.3843 - accuracy: 0.8685 - val_loss: 0.4612 - val_accuracy: 0.7783
epoch 297/300 ----- 13s 516ms/step - loss: 0.3879 - accuracy: 0.8689 - val_loss: 0.6731 - val_accuracy: 0.7685
epoch 298/300 ----- 13s 523ms/step - loss: 0.2917 - accuracy: 0.8798 - val_loss: 0.4540 - val_accuracy: 0.8079
epoch 299/300 ----- 13s 511ms/step - loss: 0.2912 - accuracy: 0.8715 - val_loss: 0.4024 - val_accuracy: 0.6749
epoch 300/300 ----- 13s 524ms/step - loss: 0.2940 - accuracy: 0.8650 - val_loss: 0.5057 - val_accuracy: 0.7911
[INFO] serializing network...

```

Gambar 4. 10 Hasil Akurasi

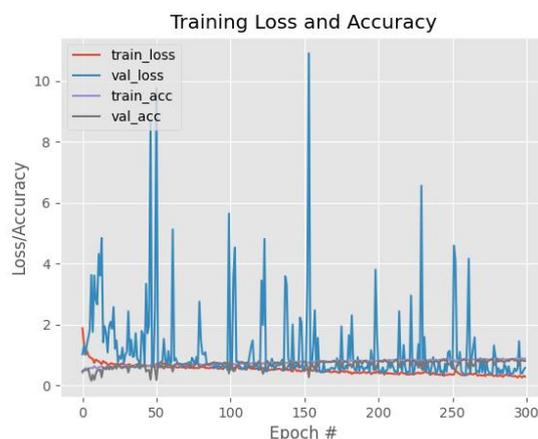
[Gambar 4.11](#) menunjukkan grafik dari proses iterasi yang dilakukan. Kita berfokus pada *training loss* dan juga *training accuracy*. Terlihat bahwa kenaikan dan penurunan tidak begitu ekstrim. Namun, hasil menunjukkan akurasi yang

Nisa Aulia Saputra, 2023

**MODEL KOMPUTASI MEDIA PEMBELAJARAN COMPUTER-AIDED DESIGN BERBASIS DEEP LEARNING PADA MATERI MEMODIFIKASI GAMBAR 3D SEDERHANA**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

didapat dari *training* yang telah dilakukan mendekati 1 dan *loss* yang dihasilkan mendekati 0. Ini membuktikan bahwa tingkat kesalahan dalam pemberian label kelas pada data *input* kecil dan tingkat kebenaran dalam pemberian label kelas cukup besar.



Gambar 4. 11 Grafik *Training Loss* dan *Training Accuracy*

### 4.3 Penggunaan Model Komputasi dalam Pembelajaran

Model komputasi ini dirancang sebagai sistem utama dari media pembelajaran. Model ini diharapkan dapat membuat siswa memiliki pemikiran analisis dalam meningkatkan literasi teknologinya. Model komputasi dirancang untuk membuat siswa berpikir analisis terkait pembelajaran CAD *inventor* materi pengoprasian fungsi perintah pada gambar 3D sederhana. Materi tersebut merupakan materi dasar *inventor* 3D yang proses penggambarannya harus disesuaikan dengan proses pemesinannya.

Model komputasi ini bekerja dengan cara memberikan prediksi perintah pada *inventor* yang sesuai untuk penggambaran suatu produk, dalam kasus ini perintah yang diprediksi adalah *extrude* dan *revolve*. Pelaksanannya dilakukan dengan cara meminta siswa untuk menyiapkan gambar produk yang akan dibuatnya. Kemudian siswa harus mengupload gambar produk tersebut di media pembelajaran CAD berbasis DL. Nantinya media akan memberikan hasil prediksi apakah produk tersebut dibuat dengan menggunakan perintah *extrude* atau *revolve*.

Setelah hasil prediksi muncul proses stimulus pun dilakukan. Proses stimulus dilakukan dengan bantuan Guru. Guru nantinya akan meminta siswa untuk

menjelaskan alasan media pembelajaran CAD berbasis DL memprediksi penggunaan *extrude/revolve* pada gambar yang mereka *upload*. Siswa harus mampu menjelaskan alasannya sesuai dengan proses pemesinan untuk membuat produk tersebut. Penjelasan siswa terkait perintah *inventor* yang media pembelajaran CAD prediksi, akan memengaruhi ketepatan pembubuhan aturan gambar pada gambar mereka. Desain pembelajaran lebih lengkap dapat dilihat pada [lampiran 5](#).

#### 4.4 Posisi Media Pembelajaran CAD berbasis DL

Media pembelajaran CAD yang dihasilkan harus mampu membantu siswa dalam memelajari CAD dengan baik. Siswa harus mampu menguasai penggambaran CAD pada materi memodifikasi gambar 3D sederhana. Tentu saja jika melihat kerucut pengalaman Edgar Dale, setidaknya media harus mampu membangun proses pembelajaran yang interaktif. Hal ini karena jika pembelajaran sudah dilakukan secara aktif, maka siswa mampu mengingat 70% materi pembelajaran yang telah disampaikan.

Melihat keandalan model komputasi untuk media pembelajaran CAD berbasis *deep learning* ini, kita mengetahui bahwa model komputasi yang dihasilkan mampu menjadi media yang dapat membangun proses diskusi di dalam kelas. Pada scenario pembelajaran ([Lampiran 5](#)), dengan menggunakan metode pembelajaran *discovery learning*, media mampu menjadi awal untuk proses diskusi siswa. Pasalnya media akan menampilkan sebuah prediksi yang belum tentu disetujui oleh semua siswa. Perbedaan pendapat dan pemahaman siswa ini yang nantinya akan membuat siswa berdiskusi dan juga mampu untuk menganalisis jawaban dan pendapat mereka masing-masing. Sehingga, jika bertumpu pada [Gambar 2.13](#) mengenai kerucut pengalaman Edgar Dale, maka model komputasi untuk media pembelajaran ini sudah berada pada posisi pembelajaran aktif.

#### 4.5 Temuan dan Pembahasan

Tujuan dari penelitian ini adalah untuk menghasilkan model komputasi media pembelajaran CAD berbasis DL untuk materi memodifikasi gambar 3D sederhana.

Nisa Aulia Saputra, 2023

MODEL KOMPUTASI MEDIA PEMBELAJARAN COMPUTER-AIDED DESIGN BERBASIS DEEP LEARNING PADA MATERI MEMODIFIKASI GAMBAR 3D SEDERHANA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Model komputasi dibuat dengan menggunakan bahasa pemrograman python dan algoritma DL yakni CNN. Proses awal pembuatan model komputasi ini adalah studi pendahuluan. Studi pendahuluan yang terdiri dari observasi, wawancara, dan studi literatur dilakukan untuk merumuskan masalah, membuat batasan penelitian, hingga perancangan ide model komputasi yang sesuai dengan kebutuhan siswa.

Studi pendahuluan diawali dengan proses observasi dan wawancara di salah satu sekolah kejuruan di kota Bandung. Sekolah kejuruan ini dipilih untuk dijadikan sampel penelitian karena sekolah ini merupakan sekolah pusat keunggulan yang berada di Ibu kota provinsi. Sehingga, bisa mewakili dan menjadi tolak ukur untuk melihat kondisi yang terjadi di SMK. Observasi dilakukan untuk melihat kondisi lingkungan sekolah, fasilitas sekolah, dan fasilitas yang dimiliki siswa. Sedangkan wawancara dilakukan untuk melihat nilai uji kompetensi siswa dan juga mengetahui kesulitan yang dialami siswa saat belajar CAD. Kuesioner juga disebar untuk melihat tingkat motivasi siswa ketika belajar CAD.

Berdasarkan hasil observasi diketahui bahwa lingkungan sekolah sudah sesuai dan baik untuk dilakukan pembelajaran. Hal ini ditandai dengan kondisi lingkungan fisik yang terawat dengan baik dan juga sesuai untuk melakukan proses pembelajaran. Selain itu, aspek lingkungan sosial dan budaya sekolah juga saling berinteraksi dengan baik dan menunjukkan budaya sekolah yang sehat.

Selanjutnya, berdasarkan obesrvasi yang dilakukan untuk melihat fasilitas sekolah dan fasilitas yang dimiliki siswa, nampaknya belum layak dan belum terpenuhi dengan baik. Terlihat dari peralatan mengajar yang belum lengkap, media pembelajaran yang tidak sesuai, spek komputer yang tidak sesuai, *software* CAD yang tidak diperbaharui, buku sumber kurang sesuai, dan juga alat tulis yang dimiliki siswa tidak lengkap. Siswa juga tidak bisa melakukan pembelajaran mandiri dan berkelanjutan di rumah karena tidak memiliki laptop yang menjadi komponen utama dalam pembelajaran CAD. Hasil menunjukkan bahwa hanya 20% siswa yang memiliki laptop dengan spek yang layak untuk menginstall *software* CAD. Sedangkan, 80% siswa lainnya didominasi oleh siswa yang tidak memiliki laptop dan juga siswa yang memiliki laptop namun spek laptop tidak sesuai dengan kebutuhan.

Selain penilaian lingkungan dan fasilitas, motivasi siswa juga dinilai. Hal ini dilakukan untuk melihat apakah ada pengaruh internal dari diri siswa yang memengaruhi proses pembelajaran CAD menjadi lebih sulit. Diketahui bahwa rata-rata motivasi siswa dalam mempelajari CAD sebesar 73%. Beberapa poin bahkan menunjukkan bahwa siswa belajar CAD hanya sebagai formalitas dan tuntutan sekolah saja. Siswa juga enggan untuk mencari tahu atau mencoba memikirkan proses pembuatan gambar menggunakan CAD sendiri. Secara statistik, 92% siswa lebih senang jika Guru mendemonstrasikan terlebih dahulu pembuatan gambar yang akan mereka buat sebelum mereka mencobanya sendiri dan 67% siswa mempelajari CAD hanya sebagai tuntutan dan memenuhi kewajiban mata pelajaran sekolah saja

Proses selanjutnya adalah wawancara yang dilakukan kepada Guru dan juga siswa. Wawancara dilakukan untuk memperkuat dan menambah data untuk merumuskan masalah penelitian. Wawancara yang dilakukan kepada Guru berfungsi untuk melihat nilai uji kompetensi siswa terkait penggambaran dengan menggunakan CAD. Hasilnya menunjukkan bahwa 47% siswa dinyatakan belum kompeten, 7% siswa dinyatakan cukup kompeten, 28% siswa dinyatakan kompeten, dan 20% siswa dinyatakan sangat kompeten. Perbandingan ini tentu sangat jauh, melihat hampir setengah jumlah total siswa yang dinyatakan belum kompeten.

Wawancara selanjutnya dilakukan kepada siswa untuk mengetahui penyebab sulitnya mencapai kompetensi CAD yang sesuai. Rata-rata siswa mengatakan bahwa CAD sulit untuk dipelajari karena itu merupakan teknologi baru bagi mereka. Beberapa diantaranya juga kesulitan secara teknis untuk menggunakan template, perintah, dan menyimpan gambar. Kesulitan-kesulitan yang dialami siswa tersebut ternyata sesuai dengan hasil penelitian yang dilakukan oleh Khoo et al. (2014), Ia menyebutkan bahwa siswa mengalami kesulitan mempelajari CAD ketika hendak membuka file yang tidak sesuai format, mengakses *software*, lambat dalam mempelajari *software* karena dirasa teknologi baru, sering terjadi *error* atau *crash* ketika menggunakan *software*, kesulitan dalam menggunakan fitur-fitur tertentu, dan kesulitan dalam memahami konsep

penggunaan fitur atau *command*.

Hasil studi pendahuluan menunjukkan bahwa siswa mengalami kesulitan CAD karena itu merupakan teknologi baru bagi mereka. Disamping itu fasilitas dan motivasi belajar yang mereka miliki tidak cukup untuk membantu mereka menguasai CAD. Faktor-faktor tersebut cukup memengaruhi proses pembelajaran CAD. Penelitian sebelumnya juga mengatakan bahwa proses belajar siswa dipengaruhi oleh lingkungan sosial keluarga, lingkungan sosial sekolah, sosial masyarakat, lingkungan alamiah, serta faktor instrumental (gedung sekolah, alat-alat belajar, fasilitas belajar, kurikulum, peraturan sekolah, buku panduan, serta silabus) (Bharuddin & Wahyuni, 2010).

Berdasarkan faktor-faktor yang ditemukan di lapangan mengenai faktor-faktor yang memengaruhi kesulitan belajar siswa mempelajari CAD, dipilihlah pembuatan media pembelajaran yang mampu menyelesaikan permasalahan siswa. Dipilihnya media pembelajaran dari berbagai macam faktor yang memengaruhi karena media pembelajaran sifatnya substansial dan juga fleksibel. Artinya media pembelajaran merupakan salah satu inti dari proses pembelajaran yang mampu diubah, dikembangkan, ataupun dimodifikasi dengan mudah tanpa harus merubah komponen substansi yang lain seperti guru, siswa, kurikulum, silabus, dll. Maka dari itu, proses perancangan model komputasi untuk media pembelajaran CAD berbasis DL dilakukan.

Perancangan model komputasi ini telah melalui beberapa eksperimen trial and error untuk mendapatkan nilai akurasi yang tinggi dan juga hasil prediksi yang tepat. Beberapa batasan diberlakukan dan disesuaikan dengan peralatan yang tersedia, seperti partisi data, banyaknya iterasi, dan spek laptop. Laptop yang digunakan sebagai alat untuk perancangan model komputasi ini memiliki spek seperti berikut: *prosesor Ryzen 5, RAM 16 GB, Harddisk 512 GB, System 64-bit operating system, dan layer 14"*. Ditetapkan beberapa parameter yang sesuai dengan kondisi laptop seperti, partisi data yang digunakan adalah 80:20, 60:40, dan 50:50; jumlah iterasi yang digunakan sebanyak 300 *epoch*, 150 *epoch*, dan 50 *epoch*; dan nilai learning rate sebesar  $1e-3$ . Ketiga parameter tersebut ditetapkan untuk meningkatkan nilai akurasi sistem. Parameter partisi data dan juga epoch

dirubah untuk melihat peningkatan akurasi yang lebih signifikan. Hal ini didukung oleh penelitian sebelumnya yang menyebutkan bahwa nilai akurasi akan meningkat lebih signifikan ketika partisi data dan jumlah iterasi ditingkatkan (Ng et al., 2020) (Afaq & Rao, 2020).

Skenario eksperimen yang dilakukan adalah dengan mencoba masing-masing partisi data dengan nilai *epoch* yang berbeda-beda seperti yang dapat dilihat pada Tabel 4.6. Hasil menunjukkan bahwa semakin banyak jumlah iterasi maka semakin tinggi pula nilai akurasi. Tidak hanya itu, semakin banyak data training yang digunakan, maka akan semakin tinggi pula nilai akurasi. Sehingga, hasil eksperimen ini sesuai dengan penelitian yang dilakukan oleh Ng et al. (2020) dan Afaq & Rao (2020) mengenai jumlah data training dan iterasi yang tinggi membuat nilai akurasi juga semakin meningkat. Percobaan setiap iterasi dan partisi data dilakukan sebanyak dua kali dengan menggunakan angka yang sama. Hal ini untuk melihat apakah ada perubahan nilai akurasi yang signifikan atau tidak. Setelah dilakukan eksperimen, ternyata ada peningkatan nilai akurasi rata-rata sebesar 0,02.

Terkait dengan nilai akurasi dan juga proses training yang dilakukan, perbandingan dengan beberapa penelitian lain juga dilakukan, guna untuk melihat seberapa unggul model komputasi media pembelajaran CAD berbasis DL ini. Serta, melihat *state of the art* penelitian DL untuk prediksi/klasifikasi. Penelitian yang dilakukan oleh Neb et al. (2020) mengenai pengenalan standar dan fitur pada 3D CAD model. Ia melakukan pengenalan fitur untuk mengklasifikasikan ke dalam 10 kelas. Hasil evaluasi dalam klasifikasi ini menunjukkan nilai akurasi sebesar 98,5% dengan 546.000 *epoch* menggunakan jaringan CNN. Akurasi yang didapatkan sangat tinggi karena Ia melakukan enam kali percobaan dengan jumlah *epoch* yang berbeda-beda dan waktu *training* yang lama.

Penelitian selanjutnya dilakukan oleh Ilahiyah & Nilogiri (2018) yang melakukan identifikasi tumbuhan berdasarkan gambar daun menggunakan CNN. Hasil penelitian menunjukkan akurasi sebesar 90% menggunakan metode *fold cross validation*. Ia mendapatkan hasil akurasi 90% setelah melakukan tujuh kali percobaan dengan data training sebanyak 2000 gambar.

Penelitian yang baru-baru ini dilakukan oleh Qotrunnada & Utomo (2022) adalah tentang klasifikasi wajah bermasker dengan metode CNN. Hasil penelitiannya menunjukkan bahwa Ia dapat mencapai akurasi sebesar 98,20% setelah mengatur layer CNN sebanyak 5 layer. Sebelumnya ia hanya menggunakan 3 layer namun validasinya masih di kisaran 70,95%.

Berdasarkan beberapa penelitian yang sudah dilakukan, hasil dari jaringan yang dirancang oleh peneliti pada pembuatan model komputasi untuk media pembelajaran CAD berbasis DL ini memiliki beberapa keunggulan. Pertama, arsitektur CNN yang digunakan adalah *Smaller VGGNet* yang memungkinkan semua piksel dapat diambil untuk melakukan training. Ini membuktikan hanya dengan menggunakan 3 *layer* CNN akurasi yang didapatkan mencapai 87%. Kedua, akurasi yang didapat tersebut dapat dikatakan cukup tinggi. Pasalnya *epoch* yang digunakan pada penelitian ini hanya 300 *epoch*. Meskipun *epoch* yang digunakan tidak sebanyak penelitian-penelitian lain, namun sistem sudah mampu mencapai tingkat akurasi di atas 80%. Ketiga, waktu pelatihan yang peneliti gunakan tidak memakan waktu yang banyak.

Akhirnya, setelah mencapai tingkat akurasi yang tinggi, model komputasi dirubah ke dalam bentuk web untuk memudahkan pengguna. Model komputasi media pembelajaran CAD berbasis DL ini diterapkan pada pembelajaran untuk melatih berpikir analisis terkait dengan pemilihan perintah pada CAD untuk menggambar suatu produk. Harapannya ketika siswa sudah mampu menganalisis perintah yang tepat untuk digunakan, kemampuan literasi teknologi siswa juga akan meningkat secara beriringan. Model komputasi media pembelajaran CAD berbasis DL ini juga diharapkan mampu membuat proses pembelajaran menjadi lebih aktif.