

**TUGAS AKHIR - EC184801**

**PENGEMBANGAN SIMULATOR MENGENAL UNTUK  
RISET DETEKSI PENGENAL MENGENAL**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Dosen Pembimbing  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*Halaman ini sengaja dikosongkan.*



**TUGAS AKHIR - EC184801**

**PENGEMBANGAN SIMULATOR MENGEMUDI UNTUK  
RISET DETEKSI PENGEMUDI MENGANTUK**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Dosen Pembimbing  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*Halaman ini sengaja dikosongkan.*



**FINAL PROJECT - EC184801**

## **DRIVING SIMULATOR DEVELOPMENT FOR DROWSY DRIVER DETECTION RESEARCH**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Advisors  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniarno, ST., MT.

Department of Computer Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2020

*Halaman ini sengaja dikosongkan.*

## **LEMBAR PENGESAHAN**

### **Pengembangan Simulator Mengemudi Untuk Riset Deteksi Pengemudi Mengantuk**

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Muhammad Nur Ady Maulana (NRP: 07211640000035)

Tanggal Ujian : 09 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh :

Ahmad Zaini, ST., MSc.

(Pembimbing I)

NIP: 1197504192002121003

Dr. Eko Mulyanto Yuniarno, ST., MT.

(Pembimbing II)

NIP: 196806011995121009

Diah Puspito Wulandari, ST., M.Sc.

(Penguji I)

NIP. 198012192005012001

Susi Juniastuti, S.T., M.Eng

(Penguji II)

NIP. 196506181999032001

Eko Premunanto, S.T., M.T.

(Penguji III)

NIP. 196612031994121001

Mengetahui  
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.

NIP. 197003131995121001

# ABSTRAK

Nama Mahasiswa : Charles Chang  
Judul Tugas Akhir : Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image  
Pembimbing : 1. Dr. Reza Fuad Rachmadi, S.T., M.T.  
2. Dr. I Ketut Eddy Purnama, S.T, M.T.

Sebagai alat pelengkap keamanan, sistem CCTV semakin banyak digunakan di setiap ruang publik untuk memantau dan menganalisa tindakan kriminal pada suatu lokasi. Akan tetapi, pencarian kriminal secara manual masih rentan akan kesalahan manusia. Salah satu solusi membuat pencarian kriminal lebih efektif dan efisien adalah dengan penggunaan Re-identifikasi.

Re-identifikasi merupakan sebuah teknik visi komputer dan deep learning dimana dilakukan pencarian ulang citra atau video milik sebuah identitas. Pada tugas akhir ini, akan dipelajari metode Re-identifikasi orang dengan citra multi-modal, adapun data Input yang akan digunakan berupa sketsa tubuh yang digambar oleh beberapa seniman berbeda.

Teknik yang dipelajari akan diimplementasikan menggunakan Lightweight Convolutional Neural Network dan pre-trained weights dari model yang digunakan untuk mengklasifikasi dataset CIFAR-10. Hasil yang diharapkan melalui Tugas Akhir ini adalah terciptanya sebuah model yang dapat melakukan Re-identifikasi orang riil dari input sketsa menggunakan Lightweight Convolutional Neural Network, sehingga pencarian kriminal di Indonesia dapat dilakukan dengan lebih efisien.

Kata Kunci : Re-identifikasi, Multi-Modal, Kriminal



# KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Re-identifikasi Orang menggunakan Lightweight Convolutional Neural Network pada Multi-Modal Image**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi *Telematika*, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Orang tua saya, atas semangat dan segala dukungan yang telah diberikan
3. Bapak Dr. Reza Fuad Rachmadi, S.T., M.T.
4. Bapak Dr. I Ketut Eddy Purnama, S.T, M.T.
5. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Serta teman - teman angkatan 2017 yang telah bersama - sama melalui kehidupan perkuliahan bersama penulis

Kesempurnaan hanya milik Tuhan Yang Maha Esa, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, April 2021

Penulis

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

Abstrak	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
NOMENKLATUR	xvii
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar belakang . . . . .	1
1.2 Permasalahan . . . . .	2
1.3 Tujuan . . . . .	3
1.4 Batasan masalah . . . . .	3
1.5 Sistematika Penulisan . . . . .	3
1.6 Relevansi . . . . .	4
<b>2 TINJAUAN PUSTAKA</b>	<b>7</b>
2.1 Machine Learning . . . . .	7
2.2 Convolutional Neural Network . . . . .	10
2.3 Pytorch . . . . .	10
2.4 <i>Triplet Loss</i> . . . . .	11
2.5 CycleGAN . . . . .	11
2.6 <i>Local Binary Pattern</i> . . . . .	12
2.7 <i>Fully-Connected Layer</i> . . . . .	13
2.8 <i>Residual Network</i> . . . . .	13
2.9 <i>Lightweight Residual Network</i> . . . . .	15
2.10 Metode Pengujian . . . . .	18
2.10.1 <i>Precision</i> . . . . .	18
2.10.2 <i>Recall</i> . . . . .	18
2.10.3 <i>mean Average Precision (mAP)</i> . . . . .	18
2.10.4 <i>Precision at n</i> . . . . .	19

<b>3</b>	<b>DESAIN DAN IMPLEMENTASI SISTEM</b>	<b>21</b>
3.1	Cakupan Tugas Akhir . . . . .	21
3.1.1	Penyesuaian Dataset PKU Sketch Re-ID . . .	21
3.1.2	Lightweight Convolutional Neural Network .	23
3.1.3	<i>Cross Domain Image-to-Image Translation</i> .	35
<b>4</b>	<b>PENGUJIAN DAN ANALISA</b>	<b>37</b>
4.1	Pengujian <i>User Interface</i> - Pemilihan Jumlah Lajur	37
4.2	Pengujian Pengambilan Data - Kecepatan . . . . .	37
4.3	Pengujian Pengambilan Data - Informasi Spasial . .	38
4.4	Pengujian Pengambilan Data - <i>Response Time</i> dan <i>Input</i> Pengemudi . . . . .	39
4.5	Pengujian Pengambilan Data - <i>Citra Webcam</i> . . . .	40
4.6	Pengujian Pengambilan Data - <i>Serial Data USB</i> . .	40
4.7	Pengujian Respon Sinyal dari <i>Steering Wheel Con-</i> <i>troller</i> terhadap simulator . . . . .	40
4.8	Pengujian <i>User Experience</i> / UX dari Pengguna . .	41
<b>5</b>	<b>PENUTUP</b>	<b>51</b>
5.1	Kesimpulan . . . . .	51
5.2	Saran . . . . .	51
	<b>DAFTAR PUSTAKA</b>	<b>53</b>
	<b>Biografi Penulis</b>	<b>57</b>

# **DAFTAR GAMBAR**

2.1	Cara Kerja Supervised Learning[1] . . . . .	7
2.2	Cara kerja Unsupervised Learning [1] . . . . .	9
2.3	Convolutional Neural Network . . . . .	10
2.4	Triplet Loss . . . . .	11
2.5	CycleGAN . . . . .	11
2.6	Local Binary Pattern . . . . .	12
2.7	Perbandingan layer ResNet, plain, dan VGG-19 [3] .	14
2.8	Skema struktur ResNet pada CIFAR10 . . . . .	15
2.9	Convolution Layer Residual Network . . . . .	16
2.10	Layer1 Residual Network . . . . .	16
2.11	Layer2 Residual Network . . . . .	17
2.12	Layer3 Residual Network . . . . .	17
2.13	Residual Network Formula . . . . .	17
3.1	Desain Sistem . . . . .	21
3.2	Beberapa contoh gambar pada Dataset PKU Sketch Re-ID . . . . .	22
3.3	Pembagian Dataset PKU Sketch Re-ID . . . . .	23
4.1	Korelasi <i>user interface</i> pemilihan lajur dengan <i>scene</i> yang dimuat oleh simulator . . . . .	43
4.2	<i>Frame webcam</i> yang tersimpan di dalam <i>harddrive</i> .	43
4.3	Diagram Sistem Komunikasi Arduino Dengan PC Si- mulator . . . . .	46
4.4	Contoh data serial yang diterima oleh arduino . . . .	46
4.5	<i>Input</i> dari pengemudi - <i>Horizontal Axis</i> adalah Data sudut <i>steering wheel</i> , <i>Vertical Axis</i> adalah data nilai tekanan pedal gas / rem . . . . .	46
4.6	Perbandingan sinyal dari <i>steering wheel controller</i> , se- belum dan sesudah grafik unity - <i>Horizontal Axis</i> / Sudut <i>steering wheel</i> . . . . .	46
4.7	Perbandingan sinyal dari <i>steering wheel controller</i> , se- belum dan sesudah grafik unity - <i>Vertical Axis</i> / Nilai tekanan pedal gas/rem . . . . .	46

4.8	Grafik analisa kuantitatif nilai error dari <i>steering wheel controller</i> . . . . .	46
-----	--	----

## DAFTAR TABEL

2.1	Parameter ResNet dibandingkan dengan Lightweight ResNet . . . . .	18
3.1	Jumlah parameter untuk beberapa model populer. .	23
3.2	Susunan Model ResNet20 . . . . .	24
3.3	Susunan Model ResNet56 . . . . .	25
3.4	Susunan Model ResNet110. . . . .	29
4.1	Skenario kuesioner untuk pengujian kepuasan pengguna . . . . .	42
4.2	Hasil pengujian kepuasan pengguna. . . . .	43
4.3	Tabel <i>Input</i> Pengemudi Selama Kurang Lebih 1 Detik	44
4.4	Data Informasi Kecepatan . . . . .	45
4.5	Data Informasi Spasial . . . . .	47
4.6	Tabel <i>Response Time</i> . . . . .	48
4.7	Data Deteksi <i>Colission</i> . . . . .	49

*Halaman ini sengaja dikosongkan*



# NOMENKLATUR

*fps* : *Frame Per Second* / Jumlah Citra Perdetik  
*unit* : unit pengukuran *Unity Game Engine*

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

### 1.1 Latar belakang

Saat ini teknologi telah berkembang pesat sehingga penggunaan kamera pengawasan atau *CCTV* sudah umum dipakai. Hasil rekaman dari kamera ini merupakan informasi visual yang sangat vital dan dapat berperan sebagai saksi terjadinya tindakan kriminal. Hasil rekaman ini mempunyai peran penting dalam memberikan bukti di investigasi kriminal dan perselisihan.

Pada tahun 2021 ini, Indonesia merupakan negara dengan jumlah penduduk terpadat ke-4 di dunia, dimana pada DKI Jakarta sendiri terdapat 10.4 juta penduduk, yaitu sekitar 333% lebih banyak dibanding Surabaya [6, 7]. Dari data yang diambil dari Badan Pusat Statistik, Polda Metro Jaya mencatat jumlah kejahatan terbanyak yaitu 31.934 kejadian, dan pada tahun 2020 ini terlihat trend adanya peningkatan tindak kriminal di seluruh Indonesia [8, 9, 10, 11]. Fakta fakta inilah yang mendorong riset riset mengenai pengurangan angka kriminalitas dengan berbagai macam cara, dibutuhkan adanya otomasi yang dapat mengurangi biaya dan beban kerja pihak kepolisian.

Re-identifikasi manusia merupakan sebuah teknik visi komputer dan deep learning dimana pada sebuah lingkungan yang terdapat beberapa kamera pengawas dilakukan adanya pencocokan citra seseorang, kepada citra yang ditangkap pada kamera lain. Masalah utama yang ditangani oleh Re-identifikasi manusia dapat disimpulkan sebagai cara untuk mencari representasi diskriminatif milik individu yang ingin dicari. Dengan membuat sistem re-identifikasi manusia, pemeriksaan hasil rekaman yang dilakukan oleh pihak kepolisian dapat dilakukan dengan jauh lebih cepat, dan dapat menekan biaya yang digunakan untuk membayar tenaga kerja. Tidak hanya untuk melakukan pencarian pelaku tindak kriminal, re-identifikasi dapat

digunakan juga untuk melihat apakah barang yang tertinggal pada suatu lokasi diambil individu yang sama, dan dapat membantu pihak sekuriti mencari orang yang hilang. Re-identifikasi manusia dapat mempermudah aktivitas - aktivitas yang sebelumnya dilakukan secara manual, maka dari itu pencarian individu sangat dibutuhkan dengan menggunakan teknologi *machine learning* untuk mempercepat proses yang sebelumnya memakan waktu yang sangat lama.

Namun untuk melakukan re-identifikasi manusia dibutuhkan citra dari pelaku, yang tidak selalu siap tersedia dimiliki oleh pihak kepolisian. Maka dari itu diperlukan adanya re-identifikasi yang menggunakan sketsa sebagai input dari model. Meskipun memiliki konsep yang mirip dengan *Face Recognition*, dikarenakan re-identifikasi sketsa menggunakan gambar *full-body* terdapat tambahan kompleksitas yang harus dihadapi oleh model yang dibuat. Selain itu masalah ini sangat menantang dikarenakan sketsa tidak memiliki warna sehingga informasi yang didapatkan oleh model lebih sedikit dibanding pada re-identifikasi manusia. Penelitian ini telah dilakukan sebelumnya oleh Lu Pang et al. dengan menggunakan model yang canggih dan *cross-domain adversarial learning*, dimana model yang dibuat dapat mencapai presisi Rank-1 34%. Namun penelitian lebih lanjut menggunakan model *lightweight* masih belum pernah dilakukan. Penggunaan model *lightweight* sangatlah berguna pada kasus *Edge Computing* dimana terdapat keterbatasan perangkat lunak.

Dengan adanya penelitian ini, diharapkan hasil re-identifikasi sketsa dapat diolah dan dikembangkan untuk membantu pencarian pelaku tindak kriminal oleh pihak kepolisian di Indonesia.

## 1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Indonesia merupakan negara dengan penduduk ke-empat terbanyak di dunia sehingga pencarian pelaku tindak kriminal diantara penduduk sipil sulit untuk dilakukan.
2. Data hasil rekaman kamera masih diperiksa secara manual oleh pihak kepolisian sehingga waktu dan ketepatan pencarian masih tidak optimal.

3. Dataset PKU Sketch Re-ID yang digunakan memiliki jumlah data yang relatif sedikit
4. Sketsa yang digunakan sebagai input tidak memiliki warna sehingga informasi yang didapat oleh model lebih sedikit.

### 1.3 Tujuan

Adapun tujuan dari penelitian Tugas Akhir ini adalah mengembangkan model re-identifikasi orang menggunakan *Lightweight Convolutional Neural Network* untuk mengidentifikasi ulang seseorang yang tertangkap pada beberapa *CCTV* berbeda dengan menggunakan sketsa dari beberapa seniman berbeda sebagai input.

### 1.4 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah:

1. Data *training* dan *testing* menggunakan data yang diambil dari PKU SketchRe-ID Dataset.
2. Jenis re-identifikasi orang yang akan dilakukan adalah *Closed Set* dan *Short Term*.
3. Training akan dilakukan menggunakan *Lightweight Convolutional Neural Network*

### 1.5 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini disusun dalam sistematika yang terstruktur, sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu sebagai berikut:

1. BAB I Pendahuluan  
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Dasar Teori  
Bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam pene-

litian, yaitu sistem simulator dan pengambilan data variabel - variabel uji.

3. BAB III Perancangan Sistem dan Impementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah pengolahan data hingga menghasilkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar penjelasan sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

## 1.6 Relevansi

1. Lightweight Residual Network for Person Re-Identification (Reza Fuad Rachmadi, Supeno Mardi Nugroho, I Ketut Eddy Purnama) [5]

*Lightweight Residual Network for Person Re-Identification* ini merupakan sebuah implementasi *lightweight CNN* untuk melakukan re-Identifikasi manusia, *lightweight CNN* yang digunakan berbasis *Residual Network* dengan menggunakan pre trained weights yang pernah digunakan untuk memecahkan masalah klasifikasi CIFAR-10. *Lightweight network* dibuat dengan membuat *ensemble* dari beberapa *Residual Network* lemah yang ada menghasilkan sebuah model yang lebih kuat. Berdasarkan hasil dari riset yang dilakukan, meskipun *lightweight CNN* tidak mendapatkan akurasi tercanggih dibandingkan model-model lainnya, banyak nya informasi yang

didapatkan oleh model ini sangat tinggi, dan dapat dikatakan model ini lebih efisien dari model-model lainnya.

2. Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. (Kaiyang Zhou, Tao Xiang). [12]

Torchreid merupakan sebuah *library deep learning* yang dibuat oleh Kaiyang Zhou untuk mempercepat implementasi dan percobaan re-identifikasi. Library ini secara umum dibuat dengan menggunakan bahasa Python dengan beberapa kode berbasis Cython untuk optimisasi. Pada library ini dataset sudah di *preprocess* dan di implementasi sesuai dengan protokol evaluasi masing masing dataset sehingga dapat dibandingkan dengan penelitian lain yang terkait.

3. Adaptive L2 Regularization in Person Re-Identification. (Xinyang Ni, Liang Fang, Heikki Huttunen) [13]

*Adaptive L2 Regularization in Person Re-Identification* merupakan sebuah penelitian untuk menambahkan regularisasi L2, dimana faktor regularisasi dapat berubah ubah secara adaptif pada *baseline* model. Dari hasil penelitian yang dilakukan regularisasi L2 secara adaptif dapat meningkatkan akurasi model sekitar 1 hingga 2% untuk *dataset* Market-1501, DukeMTMC dan MSMT17.

4. Cross-Domain Adversarial Feature Learning for Sketch Re-identification. (Lu Pang, Yaowei Wang, Yi-Zhe Song, Tiejun Huang, Yonghong Tian) [14]

*Cross-Domain Adversarial Feature Learning for Sketch Re-identification* merupakan penelitian yang pertama kali menggunakan gambar sketsa sebagai input dari model re-identifikasi manusia, namun dari model yang digunakan sendiri merupakan model yang telah di optimisasi untuk melakukan pengambilan informasi dari sketsa, seperti Triplet SN dan model GN Siamese yang merupakan gabungan dari dua cabang dari model GoogleNet yang dioptimisasi dengan menggunakan pair-wise verification loss.

Dapat dilihat dari penelitian-penelitian terkait diatas, selain pada penelitian *Cross-Domain Adversarial Feature Learning for Sketch Re-identification*, hal yang difokuskan adalah metode re-identifikasi pada gambar orang riil yang tertangkap pada CCTV menggunakan gambar lain dari individu tersebut. Sedangkan pada penelitian yang kami usulkan pada Tugas Akhir ini adalah untuk melakukan re-identifikasi pada gambar orang riil yang tertangkap pada CCTV menggunakan gambar sketsa *full body* sebagai *input*. Selain itu penelitian yang dilakukan menggunakan model lightweight classical seperti ResNet, yang bukan merupakan fokus dari penelitian terakhir.

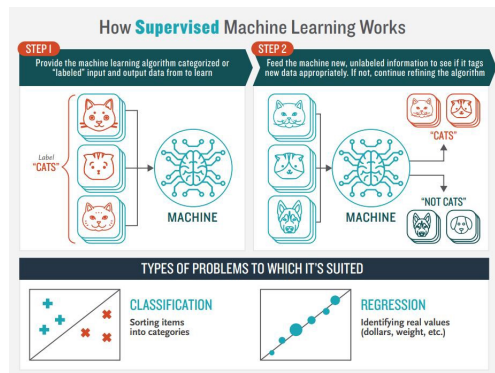


## BAB 2

# TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

### 2.1 Machine Learning



**Gambar 2.1:** Cara Kerja Supervised Learning[1]

*Machine learning* atau pembelajaran mesin adalah suatu cabang teknologi yang menerapkan penggunaan *artificial intelligence*. *Machine learning* pertama kali diperkenalkan oleh Thomas Bayes, Adrien-Marie Legendre, dan Andrey Markov pada sekitar tahun 1920[15]. Dengan menggunakan fundamental *Machine Learning* yang diciptakan oleh ilmuwan - ilmuwan tersebut, *Artificial Intelligence* kini dapat berkembang sampai dapat mengalahkan pemain catur profesional.

Dengan berkembangnya *machine learning*, tugas-tugas yang dilakukan oleh *machine learning* ini pun semakin beragam. Beberapa contoh dari tugas-tugas yang dapat dilakukan oleh *Machine Learning* adalah validasi data, menemukan pola-pola tertentu dari sumber data yang besar, mengklasifikasi grup dan objek berdasar-

kan kesamaan pola.

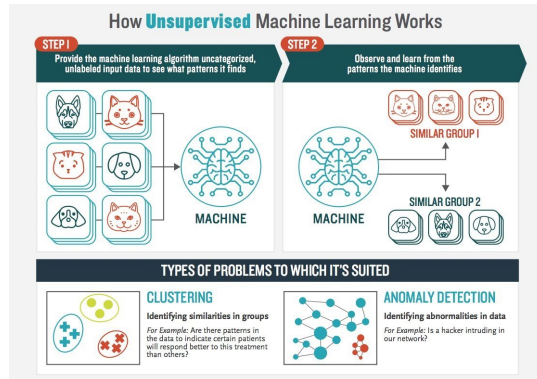
*Supervised learning* jika diartikan secara harfiah adalah pembelajaran yang ada supervisornya. Disini supervisi dilakukan oleh orang yang melakukan training kepada label di setiap datanya. Sebagai contoh dapat dilihat pada gambar 2.1.

Pada gambar diatas, masing-masing gambar kucing diberi label CATS dan yang bukan kucing (anjing,beruang,lain-lain) diberi label NOT CATS. Ketika gambar baru dimasukkan setiap label akan dicompare sampai selesai, dan yang memiliki persentase lebih banyak akan diambil sebagai prediksi akhir.

Pada pendekatan *supervised learning*, terdapat input dan output yang dapat dibuat menjadi hubungan matematis.*Supervised learning* cocok untuk digunakan untuk memprediksi dimana sudah ada contoh data yang lengkap, sehingga pola yang terbentuk adalah hasil pembelajaran dari data lengkap tersebut. Beberapa algoritma yang termasuk dalam *supervised learning* adalah sebagai berikut:

1. Regresi Linier Berganda
2. Analisis Deret Waktu
3. *Decision Tree* dan *Random Forest*
4. *Naive Bayes Classifier*
5. *Nearest Neighbor Classifier*
6. *Artificial Neural Network*

Jika dibandingkan dengan *supervised learning*, *unsupervised learning* tidak membutuhkan adanya label sebagai dasar prediksi melainkan menggunakan kesamaan atribut - atribut yang dimiliki oleh data tersebut. Jika atribut - atribut tersebut memiliki kesamaan maka data tersebut akan di *cluster* menjadi satu. Sebagai contoh dapat dilihat pada gambar di bawah :



**Gambar 2.2:** Cara kerja Unsupervised Learning [1]

Pada Gambar 2.2 dapat dilihat bahwa disediakan gambar-gambar yang tidak memiliki label ke algoritma *machine learning*. Setelah itu *artificial intelligence* akan memisahkan gambar mana yang memiliki kesamaan di dalam *cluster*. *Cluster* yang ada merupakan hasil akhir klasifikasi yang dilakukan.

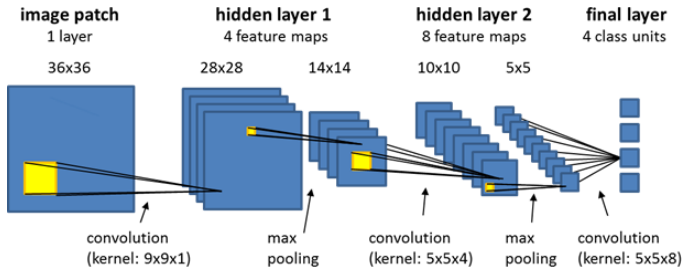
Namun *unsupervised learning* tidak memiliki hasil spesifik layaknya pada *supervised learning*. Hal ini dikarenakan tidak adanya label dasar (ground truth). Beberapa algoritma yang digunakan di *unsupervised learning* :

1. *Clustering*
2. *Anomaly Detection*
3. *Training Model*
4. *Association Discovery*

*Deep learning* (Pembelajaran Dalam) merupakan bagian yang dalam dari *machine learning* yang terdiri dari pemodelan fungsi yang ditata berlapis dan mendalam dengan menggunakan *Artificial Neural Network* (ANN). ANN merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Jenis pembelajaran dalam deep learning berupa *supervised*, *semi-supervised*, dan *unsupervised*. Deep learning

dapat diimplementasikan dalam pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya.

## 2.2 Convolutional Neural Network



**Gambar 2.3:** Convolutional Neural Network

Convolutional Neural Network merupakan salah satu algoritma *Deep Learning* yang umum digunakan untuk data berbentuk citra. Convolutional Neural Network memiliki kedalaman yang cukup tinggi sehingga termasuk dalam jenis *Deep Neural Network*. Pada umumnya CNN tidak jauh berbeda dengan neural network pada umumnya, CNN terdiri dari neuronyang memiliki *weight*, *bias*, dan *activation function*. Dimana *weight* dari CNN sendiri didapatkan dari persamaan sebagai berikut

$$neuroninput \times neuronoutput \times tinggi \times lebar$$

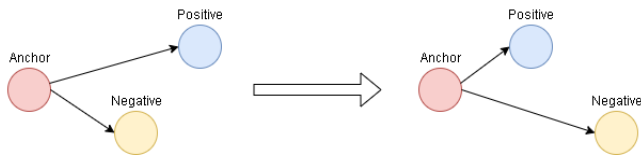
Pada Convolutional Neural Network, operasi yang paling utama merupakan Convolutional Layer, dimana terjadi operasi konvolusi pada *output* dari *layer* sebelumnya. Operasi konvolusi merupakan aplikasi kernel pada citra di semua offset sehingga citra secara keseluruhan diubah. Tujuan dari dilakukannya konvolusi itu sendiri adalah untuk melakukan ekstraksi dari fitur-fitur milik citra input.

## 2.3 Pytorch

*Pytorch* merupakan sebuah library *deep learning* pada *Python* untuk menggantikan *NumPy* dikarenakan membutuhkan kekuatan komputasi milik *Graphics Processing Unit* (GPU). Library *Pytorch*

menyediakan fleksibilitas dan kecepatan komputasi maksimal ketika melakukan penelitian *deep learning*. *Pytorch* menyediakan *pre-trained weights* dari model yang digunakan untuk mengklasifikasikan permasalahan lain sehingga dapat digunakan untuk mengurangi waktu dan kekuatan komputasi yang dibutuhkan. [16]

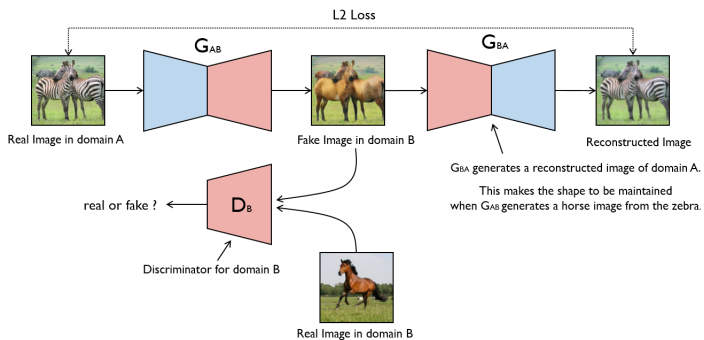
## 2.4 Triplet Loss



Gambar 2.4: Triplet Loss

*Triplet Loss* merupakan sebuah *Loss Function* yang umumnya digunakan dalam proses Re-Identifikasi. Pada fungsi ini dilakukan perbandingan jarak antara titik acuan terhadap titik positif yang merupakan gambar dalam kelas sama, dan perbandingan dengan titik negatif yang berasal dari kelas berbeda. Fungsi ini memastikan bahwa jarak ke titik positif akan lebih dekat dibandingkan dengan titik negatif.

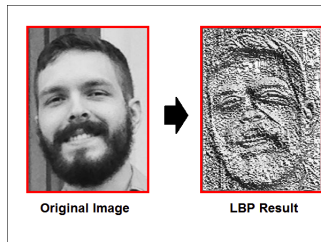
## 2.5 CycleGAN



Gambar 2.5: CycleGAN

CycleGAN merupakan sebuah teknik yang menggunakan *Deep Convolutional Neural Network* untuk melakukan sintesis sebuah gambar versi baru dengan modifikasi yang diinginkan, seperti mengubah gambar dari musim panas ke musim salju. Pada umumnya *training* model untuk melakukan hal tersebut membutuhkan data-set dengan contoh berpasangan(*paired example*) yang sangat besar. Cara seperti ini membutuhkan waktu yang sangat lama, dan pada beberapa kasus tertentu tidak dapat dilakukan. Namun dengan menggunakan CycleGAN, model dapat secara otomatis melakukan *training* untuk translasi *Image-to-image*. dilatih secara *unsupervised* dengan menggunakan kumpulan gambar dari sebuah domain X ke sebuah domain Y, tanpa harus memasangkan kedua gambar tersebut.

## 2.6 Local Binary Pattern



**Gambar 2.6:** Local Binary Pattern

*Local Binary Pattern* (LBP) merupakan salah satu deskriptor visual yang digunakan pada visi komputer. Pada umumnya LBP digunakan pada Face Recognition dikarenakan LBP merupakan deskriptor yang sangat kuat untuk melakukan klasifikasi tekstur. Selain itu telah ditemukan bahwa ketika LBP digabungkan dengan deskriptor *Histogram of Oriented Gradients* (HOG), performa yang didapatkan bertambah secara drastis pada beberapa dataset tertentu.

Cara kerja dari LBP sendiri adalah sebagai berikut:

1. Ubah citra menjadi bentuk *grayscale* / hitam putih.
2. Bagi citra menjadi beberapa bagian(*cell*)
3. Untuk setiap *pixel* yang terdapat pada sebuah *cell*, bandingkan dengan pixel milik 8 neighbor yang terdapat di sekelilingnya

4. Apabila nilai dari pixel yang di tengah lebih besar dari setiap pixel pada neighbornya maka pixel tersebut diberi nilai 0, selain itu pixel tersebut diberi nilai 1.
5. Hitung Histogram
6. Normalisasi Histogram.

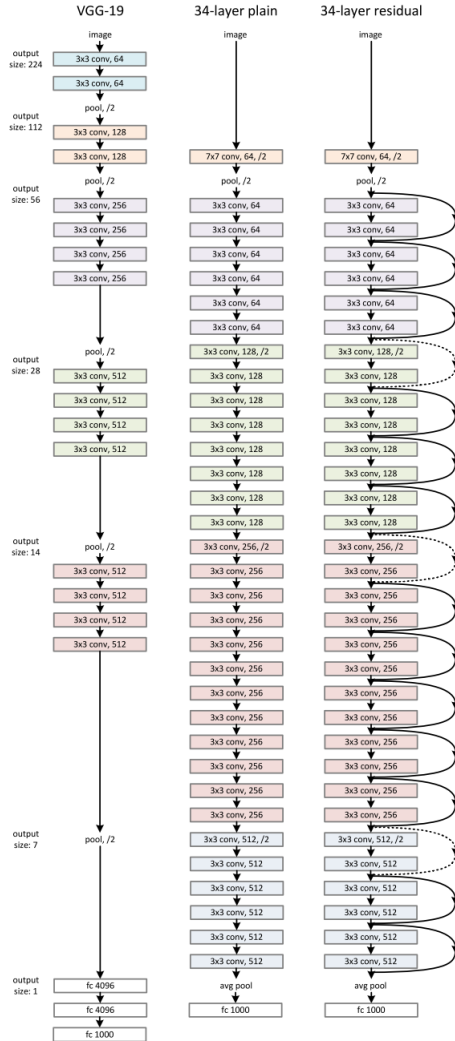
## 2.7 *Fully-Connected Layer*

*Fully Connected layer* merupakan *layer* yang bertujuan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linier, dikarenakan setiap neuron pada *convolution layer* ditransformasi terlebih dahulu menjadi satu dimensi. Namun hal tersebut dapat menyebabkan hilangnya data spasial, sehingga pada umumnya *fully connected layer* hanya diimplementasikan pada akhir jaringan. *Convolution layer* dengan ukuran kernel 1x1 dapat melakukan fungsi yang sama dengan *Fully Connected Layer*, namun dapat tetap mempertahankan data spasial.

## 2.8 *Residual Network*

*Residual Neural Network* atau *ResNet* merupakan sebuah *Artificial Neural Network* (ANN) yang dibuat berdasarkan bentuk korteks serebral milik manusia. *Residual Network* melakukan hal ini dengan memperkenalkan *skip connection* atau *shortcut*, dimana model dapat melompat dua atau tiga *layer* jika memang hal tersebut merupakan hasil terbaik. Sebelum adanya model *Residual Neural Network* penambahan *layer* pada suatu model hanya akan meningkatkan akurasi sampai suatu batas tertentu, sehingga penambahan *layer* setelah 20 hanya menambahkan kompleksitas model bukannya akurasi. Namun pada penelitian *Deep Residual Learning for Image Recognition* yang dibuat oleh Kaiming He pada tahun 2015 memproposisikan sebaliknya, apabila *layer* tambahan yang ada dapat mempelajari matriks identitas, maka minimal akurasi yang didapat pada *layer* akan sama dengan apabila tidak menambahkan *layer*. Untuk membuktikan hal tersebut dibuatlah sistem *skip connection* atau *shortcut* sehingga model dapat mempelajari matriks identitas dengan lebih mudah. Dari penelitian yang dilakukan pada dataset *ImageNet*, *Residual Network* dapat mengurangi *loss* yang didapat

ketika menambahkan lebih banyak *layer* pada *Artificial Neural Network* yang dibuat. [3]



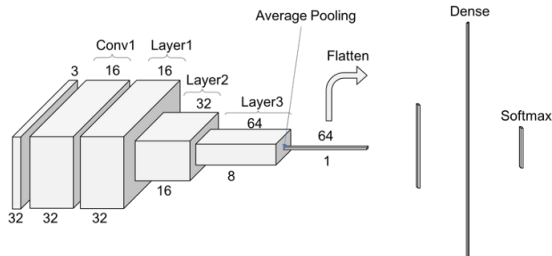
**Gambar 2.7:** Perbandingan layer ResNet, plain, dan VGG-19 [3]



## 2.9 Lightweight Residual Network

Lightweight Residual Network merupakan implementasi dari Residual Network pada dataset CIFAR-10, sebuah dataset yang terdiri dari 60000 gambar  $32 \times 32$ , dimana setiap kelas memiliki 6000 gambar. Residual Network yang digunakan untuk memecahkan masalah klasifikasi CIFAR-10 ini berbeda dengan Residual Network pada normalnya yang memiliki jumlah parameter jauh lebih besar, yaitu sekitar 23 juta parameter. Meskipun Lightweight Residual Network ini memiliki jumlah layer yang relatif dalam apabila dibandingkan dengan Residual Network yang *original*, parameter yang terdapat pada model ini jauh lebih sedikit, dengan yang paling banyak sebesar 3 juta parameter.

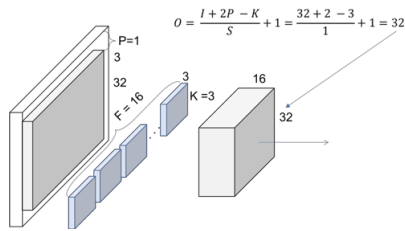
Pada gambar 2.8 dapat dilihat skema yang secara umum digunakan oleh Residual Network pada CIFAR10, dimana dapat dilihat pada skema bahwa tidak terdapat *pooling layer* setelah *convolution layer*.



**Gambar 2.8:** Skema struktur ResNet pada CIFAR10

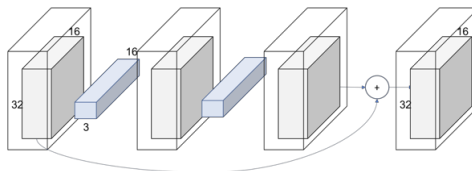
Pada *convolution layer* ini dapat dilihat bahwa layer pertama merupakan *convolution layer* dengan ukuran  $3 \times 3$  dan batch normalization. Ukuran *stride* dan *padding* yang diberikan adalah 1 untuk menyamakan bentuk output dengan input. Convolution Layer dari ResNet dapat dilihat dari gambar 2.9.

Kemudian setelah *Convolution Layer*, dibuat sebuah layer yang terdiri dari 6 konvolusi dengan ukuran  $3 \times 3$ , banyaknya kumpulan konvolusi yang digunakan akan menentukan ukuran ResNet yang dibuat. Penurunan dimensi dari data dilakukan pada tahap ini dengan melakukan penambahan *stride* menjadi dua untuk setiap konvolusi



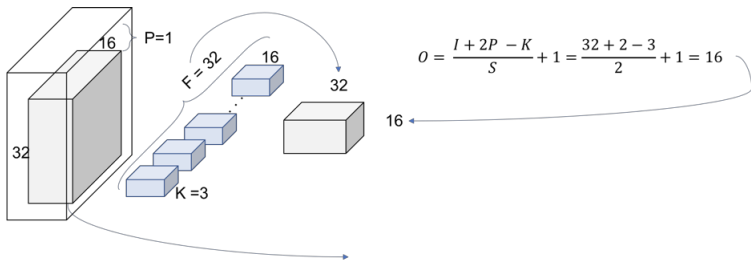
**Gambar 2.9:** Convolution Layer Residual Network

pertama di setiap layer berikut. Pada layer ini diberikan *bypass connection*, dimana apabila terjadi pengurangan performa dari model, dapat dilakukan *bypass connection* dengan cara menambah padding dimensi dengan *zeros* sehingga ukuran output sesuai dengan ukuran sebelumnya.



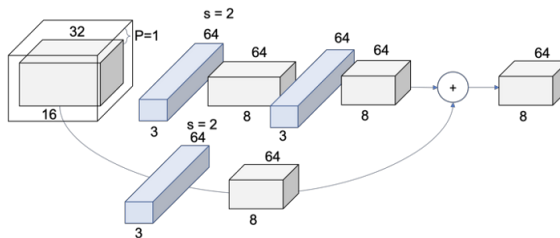
**Gambar 2.10:** Layer1 Residual Network

Layer kedua memiliki cara kerja yang sama dengan layer pertama, namun dikarenakan ukuran *stride* pada layer pertama dibuat menjadi dua, ukuran output adalah setengah dari input untuk layer kedua, sehingga diberikan *zero padding*. Selain itu pada *Bypass Connection* dibutuhkan adanya penyesuaian volume supaya sesuai dengan yang diinginkan.



**Gambar 2.11:** Layer2 Residual Network

Kemudian Layer 3 akan mengimplementasikan prinsip yang sama dengan Layer 2.



**Gambar 2.12:** Layer3 Residual Network

Sehingga formula akhir dari struktur Lightweight Residual Network dapat disimpulkan sebagai berikut 2.13

output map size	$32 \times 32$	$16 \times 16$	$8 \times 8$
# layers	$1+2n$	$2n$	$2n$
# filters	16	32	64

**Gambar 2.13:** Residual Network Formula

Berikut merupakan perbandingan parameter Lightweight Residual Network dengan Residual Network

Name	Parameters
ResNet20	0.27M
ResNet32	0.46M
ResNet44	0.66M
ResNet56	0.85M
ResNet110	1.7M
ResNet110	1.7M
ResNet50	23M

**Tabel 2.1:** Parameter ResNet dibandingkan dengan Lightweight ResNet

## 2.10 Metode Pengujian

### 2.10.1 *Precision*

*Precision* merupakan rasio dari prediksi jumlah total contoh positif yang benar diklasifikasikan dibagi dengan jumlah keseluruhan hasil yang diprediksi positif. *Precision* dapat melihat dari keseluruhan data, berapa persen yang diklasifikasikan secara benar.

$$Recall = \frac{TP}{(TP + FP)}$$

### 2.10.2 *Recall*

*Recall* merupakan rasio dari jumlah total positif yang benar diklasifikasikan, dibagi dengan jumlah total contoh yang benar positif. *Recall* yang tinggi menunjukkan bahwa kelas yang dikenali dengan benar banyak, atau False Negative yang didapatkan sedikit.

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

### 2.10.3 *mean Average Precision (mAP)*

*mean Average Precision (mAP)* merupakan sebuah metrik akurasi yang didapatkan dari rata rata *Average Precision*. Dimana *Average Precision* sendiri didapatkan dari perhitungan nilai presisi dan recall. *mean Average Precision* merupakan sebuah metrik evaluasi

yang sangat baik dikarenakan melibatkan presisi dan recall. Dengan menggunakan mean Average Precision, sebuah angka dapat diterima untuk mengukur kinerja dari pendeteksian sebuah objek.

$$AP = \Sigma(recall_{n+1}) - recall_n) \times precision_{interp} \times (recall_{n+1})$$

#### **2.10.4 Precision at n**

*Precision at n* merupakan sebuah metrik evaluasi yang didapatkan dari pengembalian informasi dalam bentuk daftar dengan ranking. Daftar ranking tersebut kemudian digunakan untuk melakukan penilaian dari akurasi model dengan cara melihat informasi ke-n paling atas yang dikembalikan. Rank-1 accuracy merupakan pengecekan berapa kali prediksi label sesuai dengan yang sebenarnya, sedangkan Rank-5 accuracy merupakan berapa kali prediksi label terdapat pada 5 prediksi paling atas.

*Halaman ini sengaja dikosongkan*

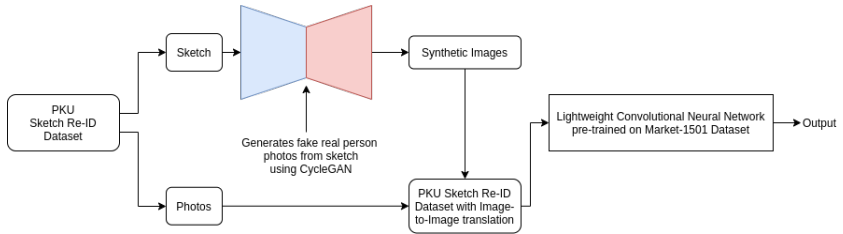
## BAB 3

# DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

### 3.1 Cakupan Tugas Akhir

Tugas akhir ini merupakan salah satu bentuk implementasi visi komputer untuk melakukan pencarian seorang individu menggunakan gambar multi-modal berupa sketsa dengan menggunakan Lightweight Convolutional Neural Network, berikut pada Gambar 3.1 adalah cakupan Tugas Akhir dari Desain Sistem.



**Gambar 3.1:** Desain Sistem

#### 3.1.1 Penyesuaian Dataset PKU Sketch Re-ID

Desain sistem secara umum pada gambar 3.1, yang mencakup beberapa hal, salah satunya ialah penyesuaian dataset. Dataset yang kami akan gunakan adalah dataset PKU Sketch Re-ID yang dibuat oleh Lu Pang [4]. Dataset ini terdiri dari 200 identitas yang unik, dimana setiap identitas telah ditangkap oleh dua kamera berbeda. Selain itu setiap individu memiliki satu gambar sketsa, sehingga seluruh dataset bertotal 600 gambar. Gambar 3.2

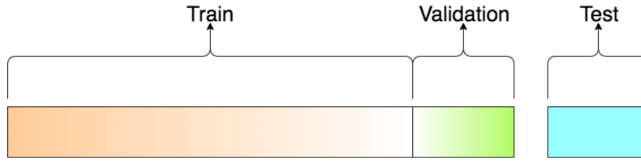
menunjukkan beberapa contoh dari dataset PKU Sketch Re-ID.



**Gambar 3.2:** Beberapa contoh gambar pada Dataset PKU Sketch Re-ID

Dataset PKU Sketch [4] Re-id kemudian akan melalui proses persiapan terlebih dahulu sehingga dapat dilakukan evaluasi performa dari Lightweight Convolutional Neural Network yang akan digunakan. Penyesuaian yang akan dilakukan pada dataset meliputi, penggabungan individu-individu sama yang tertangkap oleh kamera kamera berbeda, kemudian pemasangan individu tersebut ke gambar sketsa sehingga setiap individu unik akan menjadi kelas tersendiri. Kemudian dataset tersebut akan dibagi menjadi 150 identitas untuk training dan 50 identitas untuk testing, seperti yang ditunjukkan pada gambar 3.3. Setiap gambar telah di *crop* secara *manual* untuk memastikan setiap gambar dari dataset hanya berisi satu individu yang spesifik. Untuk gambar sketsa yang digunakan, terdapat 5 orang seniman berbeda yang memiliki 5 *art style* yang berbeda pula untuk menggambar masing masing identitas.





**Gambar 3.3:** Pembagian Dataset PKU Sketch Re-ID

### 3.1.2 Lightweight Convolutional Neural Network

Setelah penyesuaian dataset telah dilakukan, akan dilakukan training dengan menggunakan Lightweight Residual Network yang pernah digunakan untuk memecahkan masalah klasifikasi CIFAR-10. Namun pada model yang kami gunakan, *Fully-Connected Layer* terakhir dihilangkan, dan ditambahkan dua layer baru untuk menggantikan layer tersebut sehingga model dapat dipastikan akan mempelajari fitur-fitur yang terdapat pada dataset PKU Sketch Re-ID. Selain itu, berbeda dengan model ResNet aslinya, model yang kami gunakan menggunakan resolusi input sebesar  $32 \times 64 \times 3$ .

Nama	Parameter
ResNet56	0.85M
ResNet110	1.7M
GoogleNet	7M
DenseNet121	8.6M
ResNet50	23M

**Tabel 3.1:** Jumlah parameter untuk beberapa model populer.

Pada percobaan-percobaan yang kami lakukan, kami menggunakan tiga model Residual Network yang berbeda, yaitu ResNet20, ResNet56, dan ResNet110. Namun kami hanya melanjutkan menggunakan ResNet56 dan ResNet110 dikarenakan kedua model tersebut mendapatkan hasil yang jauh lebih baik dibandingkan dengan ResNet20. Meskipun layer dari kedua model yang kami gunakan relatif cukup dalam (56 dan 110 layer) apabila dibandingkan dengan Residual Network normalnya. Parameter yang terdapat pada model ini jauh lebih sedikit, dimana ResNet110 hanya memiliki

1.7 juta parameter, jauh dari jutaan parameter yang dimiliki model-model populer seperti DenseNet, GoogleNet, dan ResNet50. Pada tabel 3.2 dapat dilihat bentuk susunan layer dari ResNet20 sendiri. Untuk susunan model ResNet56 dapat dilihat pada 3.3 dan untuk ResNet110 dapat dilihat pada 3.4.

**Tabel 3.2:** Susunan Model ResNet20

Type	Channel	Size	Stride
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	32	3x3	2x2
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	32	3x3	1x1
BatchNorm2D	32	-	-
Convolutional2D	64	3x3	2x2
BatchNorm2D	64	-	-
Dilanjutkan ke halaman selanjutnya			

**Tabel 3.2 – Dilanjutkan dari halaman sebelumnya**

Type	Channel	Size	Stride
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
FullyConnected	-	-	-
BatchNorm1D	512	-	-
Dropout	-	-	-
FullyConnected	-	-	-

**Tabel 3.3:** Susunan Model ResNet56

Type	Channel	Size	Stride
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
BatchNorm2D	16	-	-
Convolutional2D	16	3x3	1x1
Dilanjutkan ke halaman selanjutnya			





**Tabel 3.3 – Dilanjutkan dari halaman sebelumnya**

Type	Channel	Size	Stride
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
FullyConnected	-	-	-
BatchNorm1D	512	-	-
Dropout	-	-	-
FullyConnected	-	-	-















**Tabel 3.4 – Dilanjutkan dari halaman sebelumnya**

Type	Channel	Size	Stride
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
Convolutional2D	64	3x3	1x1
BatchNorm2D	64	-	-
FullyConnected	-	-	-
BatchNorm1D	512	-	-
Dropout	-	-	-
FullyConnected	-	-	-

### ***3.1.3 Cross Domain Image-to-Image Translation***

*Halaman ini sengaja dikosongkan*