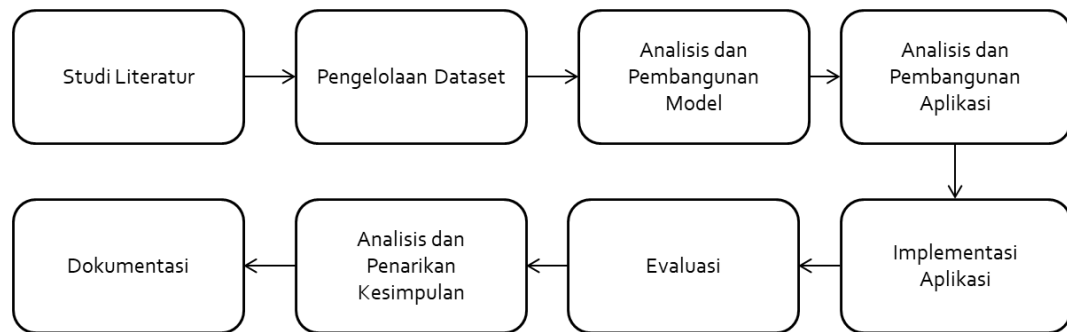


BAB III METODELOGI PENELITIAN

Bab ini menyajikan rencana dari penelitian yang akan dilakukan sesuai desain penelitian serta alat dan bahan komputasi yang dibutuhkan.

3.1. Desain Penelitian



Gambar 3.1 Desain Penelitian

Sesuai dengan desain penelitian pada Gambar 3.1 tahapan penelitian yang akan dilakukan setelah melakukan studi literatur akan dijelaskan sebagai berikut.

3.1.1. Studi Literatur

Setelah merumuskan rumusan masalah, dilakukan studi literatur untuk mendalami metode-metode yang digunakan untuk menyelesaikan rumusan masalah. Studi literatur menggunakan sumber pustaka yang berasal dari jurnal, paper, dan buku baik berasal dari Internet maupun bukan dari Internet seperti yang dituliskan pada Bab 2.

3.1.2. Pengelolaan Dataset

Pengelolaan dilakukan karena penelitian menggunakan dataset yang sudah tersedia. Pemilihan dari dataset mempertimbangkan akses terhadap label data usia di dalamnya. Umumnya, label usia didapatkan setelah mengakses informasi pribadi subjek berupa tanggal lahir yang sulit diakses atau harus melabeli gambar secara manual dari daftar label terpisah yang keduanya memakan waktu (Levi & Hassner, 2015). Dengan mempertimbangkan persyaratan tersebut, dataset dalam penelitian ini menggunakan All-Age-Faces Dataset oleh J. Cheng dkk (2019) seperti pada

Gambar 3.2. Informasi lebih lanjut mengenai dataset ini dapat diakses melalui alamat situs <https://github.com/JingchunCheng/All-Age-Faces-Dataset>. Kemudian, untuk mengunduhnya bisa menggunakan tautan baidu atau dropbox yang tersedia pada halaman tersebut secara bebas.

Dataset terdiri dari lima *folder* dimana pada *folder original images* berisi data gambar wajah seperti Gambar 3.2. Setiap gambar yang berada dalam dataset tersebut memiliki resolusi 350 x 325 pixel. Dataset ini awalnya digunakan untuk penelitian gender recognition karena menurut panduan yang disediakan, dataset memiliki indeks yang mewakili gender. Data dari indeks 00000 sampai 07380 berjenis kelamin wanita sedangkan dari indeks 07381 sampai 13321 berjenis kelamin laki-laki sehingga data gambar wajah wanita lebih banyak daripada laki-laki. Contohnya gambar wajah dengan nama file 00005A02 artinya gambar ini berada pada indeks nomer 5 berjenis kelamin perempuan berusia 2 tahun. Setiap gambar memiliki nama file berbeda yang disusun sesuai dengan indeks dan usia dengan format “indeksAusia” dengan rentang usia dari 2 – 80 tahun. Dengan demikian, informasi usia yang tersurat pada nama file akan digunakan sebagai label kelas.

Seluruh data dalam dataset ini dibagi menjadi kelas usia tanpa mempertimbangkan jenis kelamin dalam *folder* yang mewakili setiap kelas usia. Kemudian data tersebut dipindahkan ke dalam setiap *folder* kelas usia secara manual sesuai dengan nama file. Dengan demikian, setiap data kelas usia memiliki jumlah data seperti pada lampiran 1 kolom jumlah data asli.

Dataset yang sudah diberi label akan dilakukan *preprocessing image* berupa resize untuk memperkecil resolusi gambar sehingga mempercepat pemrosesan training pada machine learning. Penggunaan *preprocessing image* juga dilakukan untuk memenuhi persyaratan input dari arsitektur model machine learning yang dipilih. Tahap ini juga akan mengubah format data menjadi .png untuk mempermudah verifikasi dengan *backup* dari dataset apabila telah terjadi kesalahan atau adanya data yang rusak. Setelah gambar keluaran sudah memiliki format data berbeda

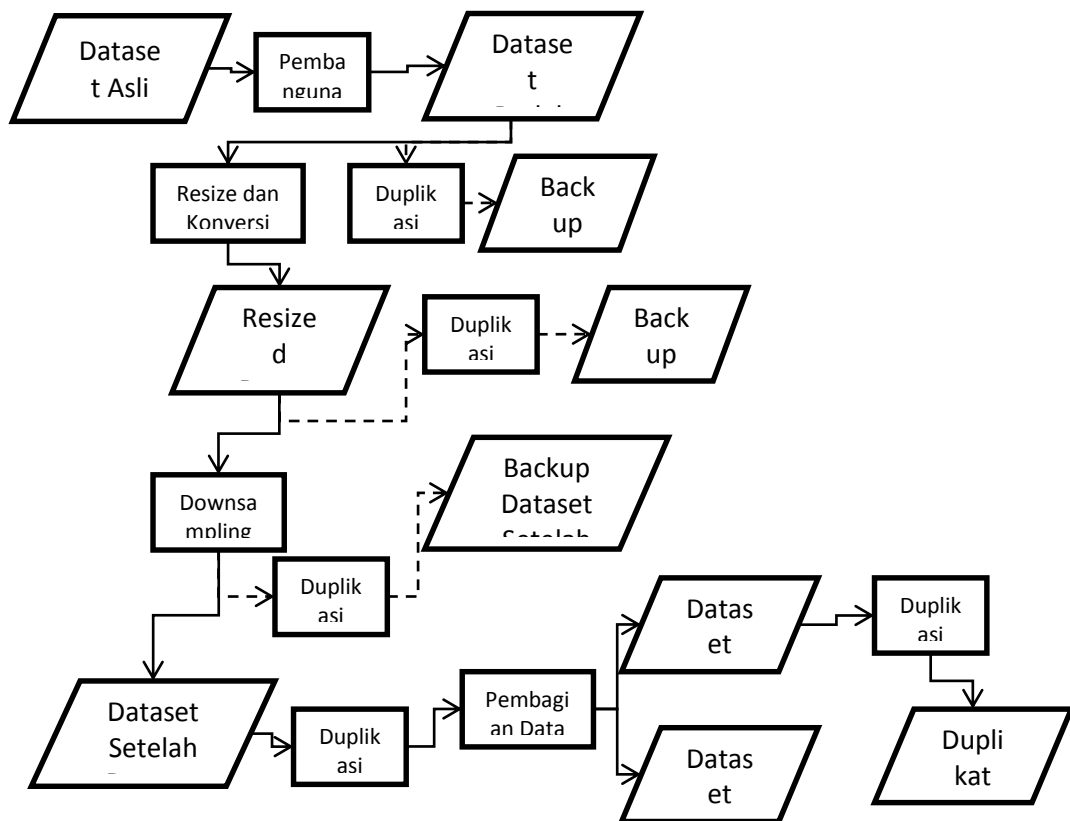
dengan data asli, pengelolaan dataset dapat dilanjutkan ke tahap *downsampling* atau pengurangan data.

Dikarenakan jumlah data dari jenis kelamin yang tidak seimbang, pembagian data untuk setiap label kelas akan tidak merata untuk setiap usia. Dengan demikian, diperlukan pengurangan data (*downsampling*) untuk menyeimbangkan jumlah isi data setiap kelas. Sebelum dilakukan pengurangan data, seluruh dataset diduplikat sebagai *backup* yang disimpan ke dalam *directory* yang berbeda. Duplikasi ini digunakan untuk memudahkan pengembalian data seperti semula apabila terjadi kesalahan. Setelah itu, pengurangan data bisa dimulai dengan menghapus data yang dilakukan secara manual dengan pemilihan data yang acak. Dengan demikian, jumlah data setiap kelas usia memiliki jumlah yang lebih seimbang dibandingkan data asli.

Dataset tersebut menjadi dataset *training* untuk membangun model generatif yang akan digunakan untuk data augmentation. Dataset akan diduplikat lagi ke *directory* berbeda yang membedakan antara dataset untuk pembangunan model generatif dan pembangunan model *age estimation*. Untuk dataset pembangunan model *age estimation* dilakukan pembagian data training dan data testing dengan komposisi persentase data 80:20. Data training kemudian dilakukan duplikasi untuk pembangunan dataset yang kemudian dinamai dataset B. Setelah tahap ini selesai, dataset sudah bisa digunakan dalam tahap pembangunan model generatif dan kemudian model *age estimation*.



Gambar 3.2 Sampel Wajah dari All-Age-Faces Dataset (J. Cheng dkk., 2019)



Gambar 3.3 Alur Pengelolaan Dataset

3.1.3. Analisis dan Pembangunan Model

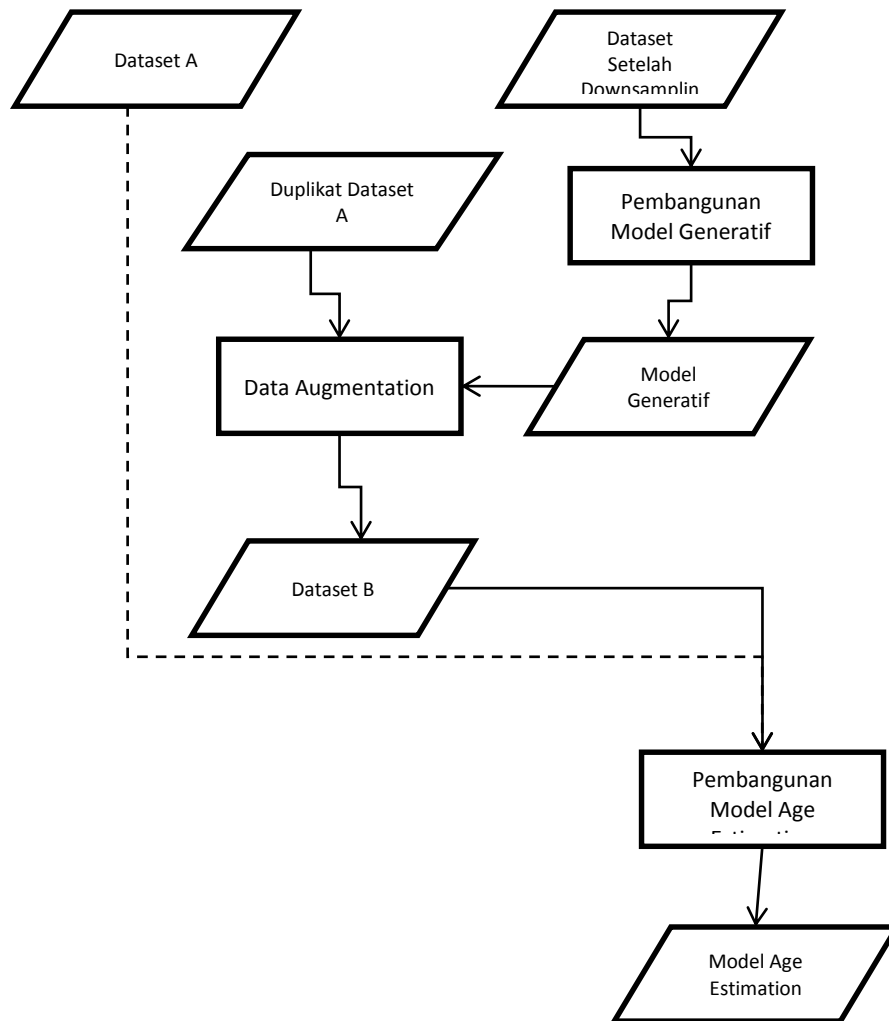
Berikut ini adalah analisis kebutuhan pembangunan model secara garis besar seperti pada Gambar 3.4. Pembangunan model diawali dari membangun model generatif yang akan digunakan untuk data augmentation menghasilkan dataset dengan data

Galih Abdul Muhyi, 2020

AGE ESTIMATION UNTUK INTELLIGENT ADVERTISING PADA POSTER DIGITAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

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

augmentation. Kemudian untuk membangun model *age estimation* dilakukan secara paralel menggunakan dataset awal tanpa *data augmentation* dan dataset dengan *data augmentation*. Pengujian model yang dibangun dilakukan saat tahap evaluasi.



Gambar 3.4 Alur Pembangunan Model

1. Pembangunan Model Generatif

Model generatif dibangun menggunakan metode GAN. Tujuannya untuk menghasilkan model generatif untuk tahap data augmentation dalam pembangunan model *age estimation*. Arsitektur GAN yang akan digunakan yaitu ACGAN (Odena dkk., 2017) dengan *hyperparameter* yang direkomendasikan seperti pada Tabel 3.1 dan *hyperparameter* hasil modifikasi berdasarkan Tabel 3.1 yang dapat dituliskan pada Tabel 3.2.

Tabel 3.1 Hyperparameter A Model ACGAN (Odena, 2017)

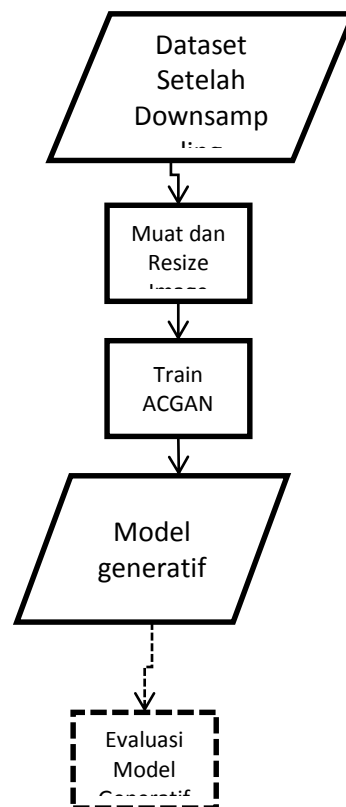
Operasi	Kernel	Stride	Peta Fitur	BN	Dropout	Nonlinearity
$G_x(z) - 79 \times 1 \times 1$ input						
Linear	-	-	768	Tidak	0.0	ReLU
Transposed Convolution	5 x 5	2 x 2	384	Ya	0.0	ReLU
Transposed Convolution	5 x 5	2 x 2	256	Ya	0.0	ReLU
Transposed Convolution	5 x 5	2 x 2	192	Ya	0.0	ReLU
Transposed Convolution	5 x 5	2 x 2	3	Tidak	0.0	Tanh
$D(x) - 128 \times 128 \times 3$ input						
Convolution	3 x 3	2 x 2	16	Tidak	0.5	Leaky ReLU
Convolution	3 x 3	1 x 1	32	Ya	0.5	Leaky ReLU
Convolution	3 x 3	2 x 2	64	Ya	0.5	Leaky ReLU
Convolution	3 x 3	1 x 1	128	Ya	0.5	Leaky ReLU
Convolution	3 x 3	2 x 2	256	Ya	0.5	Leaky ReLU
Convolution	3 x 3	1 x 1	512	Ya	0.5	Leaky ReLU
Linear	-	-	11	Tidak	0.0	Soft Sigmoid
Optimizer	ADAM ($\alpha = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999$)					
Batch Size	100					
Iterasi	5000					
Leaky ReLU slope	0.2					
Weight, bias initialization	Isotropic gaussian ($\mu = 0, \sigma = 0.02$)					

Tabel 3.2 Hyperparameter B Model ACGAN (Odena, 2017) Dengan Penyesuaian

Operasi	<i>Kernel</i>	<i>Stride</i>	Peta Fitur	BN	<i>Dro pout</i>	<i>Nonlinearity</i>
$G_x(z) - 79 \times 1 \times 1$ input						
<i>Linear</i>	-	-	768	Tidak	0.0	ReLU
<i>Transposed Convolution</i>	5 x 5	2 x 2	384	Ya	0.0	ReLU
<i>Transposed Convolution</i>	5 x 5	2 x 2	192	Ya	0.0	ReLU
<i>Transposed Convolution</i>	5 x 5	2 x 2	96	Ya	0.0	ReLU
<i>Transposed Convolution</i>	5 x 5	2 x 2	3	Tidak	0.0	Tanh
$D(x) - 128 \times 128 \times 3$ input						
<i>Convolution</i>	3 x 3	2 x 2	16	Tidak	0.5	Leaky ReLU
<i>Convolution</i>	3 x 3	1 x 1	32	Ya	0.5	Leaky ReLU
<i>Convolution</i>	3 x 3	2 x 2	64	Ya	0.5	Leaky ReLU
<i>Convolution</i>	3 x 3	1 x 1	128	Ya	0.5	Leaky ReLU
<i>Convolution</i>	3 x 3	2 x 2	256	Ya	0.5	Leaky ReLU
<i>Convolution</i>	3 x 3	1 x 1	512	Ya	0.5	Leaky ReLU
<i>Linear</i>	-	-	11	Tidak	0.0	Soft Sigmoid
<i>Optimizer</i>	ADAM ($\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)					
<i>Batch Size</i>	100					
<i>Iterasi</i>	5000					
<i>Leaky ReLU slope</i>	0.2					
<i>Weight, bias initialization</i>	Isotropic gaussian ($\mu = 0$, $\sigma = 0.02$)					

Tabel 3.2 merupakan hasil penyesuaian berupa perubahan pada *hyperparameter* model generator pada layer setelah *transpose convolution* pertama dengan jumlah peta fitur dari 256 menjadi 192. Kedua tabel memiliki jumlah input label sebesar 79 yang disesuaikan dengan jumlah seluruh kelas yang ada pada dataset. Hasil dari Tabel 3.1 dan Tabel 3.2 berupa model generatif yang selalu dihasilkan setiap epoch.

Gambar 3.4 menampilkan alur pembangunan dari model generatif menggunakan bahasa pemrograman Python melalui framework Tensorflow. Dataset pasca preprocessing image akan dimuat dan diresize untuk muat ke dalam jaringan discriminator ACGAN. Pada saat proses training dimulai, input pada generator akan membuat gambar baru dari noise dan label yang sesuai dengan jumlah kelas dari serangkaian layer *transposed convolution* yang berasal dari *noise* yang dihasilkan secara acak.



Gambar 3.5 Alur Kerja Program Pembangunan Model Generatif

Sedangkan *discriminator* membutuhkan *input* gambar untuk *training* dengan resolusi 128 x 128 dengan 3 channel RGB dimana setiap *layer* akan dilakukan konvolusi dengan stride 2 x 2 setiap layer ganjil dan jumlah peta fitur yang naik dua

kali lipat dari 16. Setiap layer setelah layer pertama akan dilakukan *batch normalization* (BN) dan kemudian dilakukan dropout pada semua layer untuk mengurangi *overfitting*. Baik *discriminator* maupun *generator* sama-sama menggunakan *optimizer* ADAM. *Noise* deviasi bernilai 0.2 dan α atau *learning rate* sebesar 0.0002 pada *optimizer* ADAM.

Sebelum memasuki model *discriminator*, gambar akan dimuat dan di-resize untuk memastikan resolusi gambar *data training* sesuai dengan syarat input dari konvolusi pertama. Setelah itu proses *training* dapat dimulai yang diawali dengan *generator* yang memproduksi gambar berlabel sedangkan *discriminator* memulai proses *training* menggunakan dataset. Kemudian output dari generator akan masuk ke *discriminator* untuk menilai keaslian dan mengklasifikasi gambar dari *generator*. Model dan sampel gambar yang dihasilkan akan disimpan setiap *epoch* tertentu untuk menghemat kapasitas isi *harddisk*.

Model *generatif* yang dibangun akan diuji secara kuantitatif menggunakan metode Inception Score dan Frechet Inception Distance dan secara kualitatif akan dinilai secara manual dengan pengamatan penulis yang akan dibahas lebih lanjut pada Subbab 3.1.6.

2. Pembangunan Model *Age estimation*

Model *age estimation* dibangun menggunakan bahasa pemrograman Python melalui *framework* Tensorflow dengan alur kerja seperti pada Gambar 3.6. Model dibangun menggunakan arsitektur ConvNet menggunakan *hyperparameter* dari *discriminator* yang terinspirasi dari Tabel 3.1 dengan penambahan pada *layer pooling* seperti yang dibahas pada Subbab 2.6. Tujuannya tetap sama untuk mengklasifikasi gambar dengan kelas usia pada dataset. Perbedaannya terletak pada kedalaman layer konvolusi dan penambahan layer *max pooling* di dalamnya seperti pada Tabel 3.3 dan Tabel 3.4. Dengan demikian eksperimen yang akan dilakukan menghasilkan dua model yang berbeda.

Data training yang digunakan menggunakan dua dataset, yaitu dataset A dan dataset B. Dataset A yaitu dataset tanpa penambahan data berasal dari *dataset training* dan dataset B yaitu duplikat dataset A dengan *data augmentation* yang berasal dari hasil sampel model generatif. Alur pembangunan dataset B dapat diilustrasikan pada Gambar 3.7 dimana sebelum digunakan dilakukan pembersihan gambar untuk menyamakan jumlah gambar setiap kelas dan menghapus sampel gambar dengan kualitas yang buruk.

Dataset A maupun dataset B akan dimuat lalu di-*resize* untuk memastikan setiap gambar memenuhi syarat yang ditentukan operasi konvolusi pada layer pertama. Setelah itu, tahap training bisa dimulai menggunakan parameter yang ditentukan sebelum program dieksekusi seperti jumlah epoch training dan batch data yang digunakan selama training. Semua dataset akan digunakan untuk pembangunan model dengan *hyperparameter A* dan *hyperparameter B*. Dengan demikian ada empat model yang dibangun untuk diuji menggunakan metode pada Subbab 3.1.6 untuk menentukan model terbaik yang akan digunakan pada tahap implementasi aplikasi yang akan dibahas dalam Subbab 3.1.5.

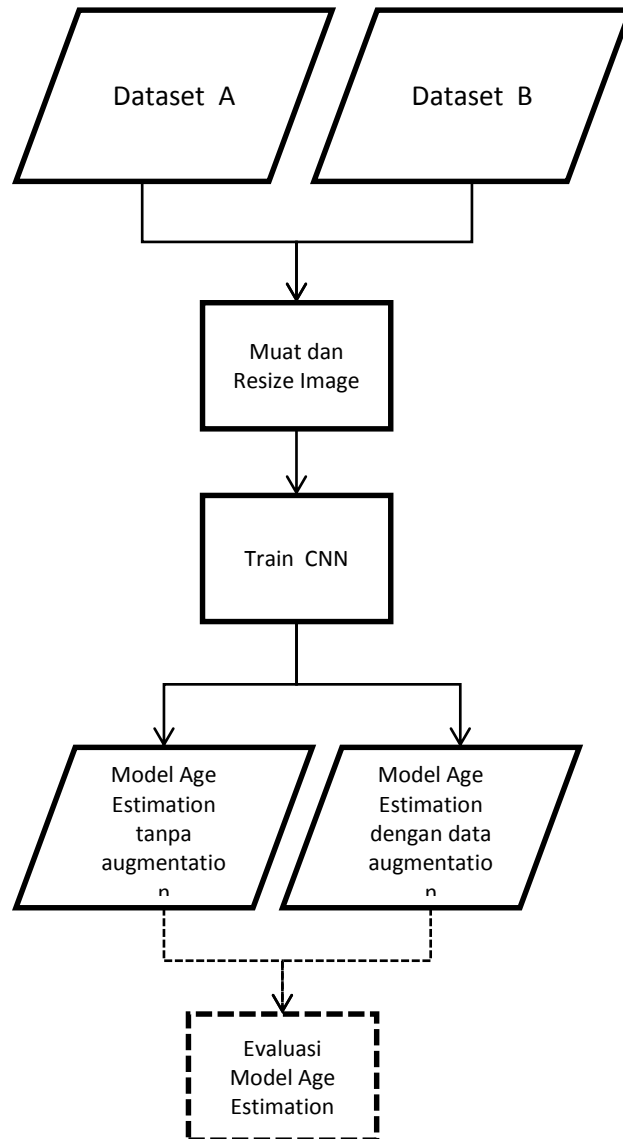
Tabel 3.3 Hyperparameter Arsitektur A Age Estimation

Operasi	<i>Kernel</i>	<i>Stride</i>	Peta Fitur	BN	<i>Dropout</i>	<i>Nonlinearity</i>	<i>Pooling</i>
<i>Convolution</i>	3 x 3	2 x 2	32	Tidak	0.25	Leaky ReLU	<i>Max</i>
<i>Convolution</i>	3 x 3	2 x 2	64	Ya	0.25	Leaky ReLU	-
<i>Convolution</i>	3 x 3	2 x 2	128	Ya	0.25	Leaky ReLU	-
<i>Convolution</i>	3 x 3	2 x 2	256	Ya	0.25	Leaky ReLU	-
<i>Linear</i>	-	-	81	Tidak	0.0	Softmax	-
<i>Optimizer</i>	ADAM ($\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)						
<i>Batch Size</i>	1000						
Iterasi	5000						
Leaky ReLU <i>slope</i>	0.2						

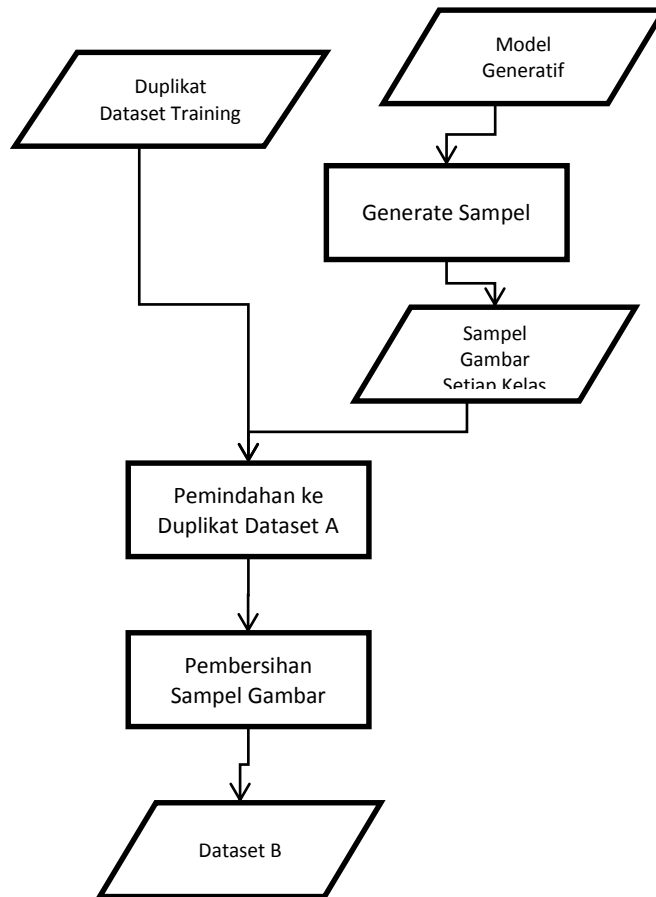
<i>Weight, bias initialization</i>	<i>Isotropic gaussian ($\mu = 0, \sigma = 0.02$)</i>
------------------------------------	---

Tabel 3.4 Hyperparameter Arsitektur B Age Estimation

Operasi	<i>Kernel</i>	<i>Stride</i>	Peta Fitur	BN	<i>Dropout</i>	<i>Nonlinearity</i>	<i>Pooling</i>
<i>Convolution</i>	3 x 3	2 x 2	32	Tidak	0.25	Leaky ReLU	<i>Max</i>
<i>Convolution</i>	3 x 3	2 x 2	64	Ya	0.25	Leaky ReLU	<i>Max</i>
<i>Convolution</i>	3 x 3	2 x 2	128	Ya	0.25	Leaky ReLU	-
<i>Convolution</i>	3 x 3	2 x 2	256	Ya	0.25	Leaky ReLU	-
<i>Convolution</i>	3 x 3	2 x 2	512	Ya	0.5	Leaky ReLU	-
<i>Linear</i>	-	-	81	Tidak	0.0	Softmax	-
<i>Optimizer</i>	ADAM ($\alpha = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999$)						
<i>Batch Size</i>	1000						
Iterasi	5000						
Leaky ReLU <i>slope</i>	0.2						
<i>Weight, bias initialization</i>	<i>Isotropic gaussian ($\mu = 0, \sigma = 0.02$)</i>						



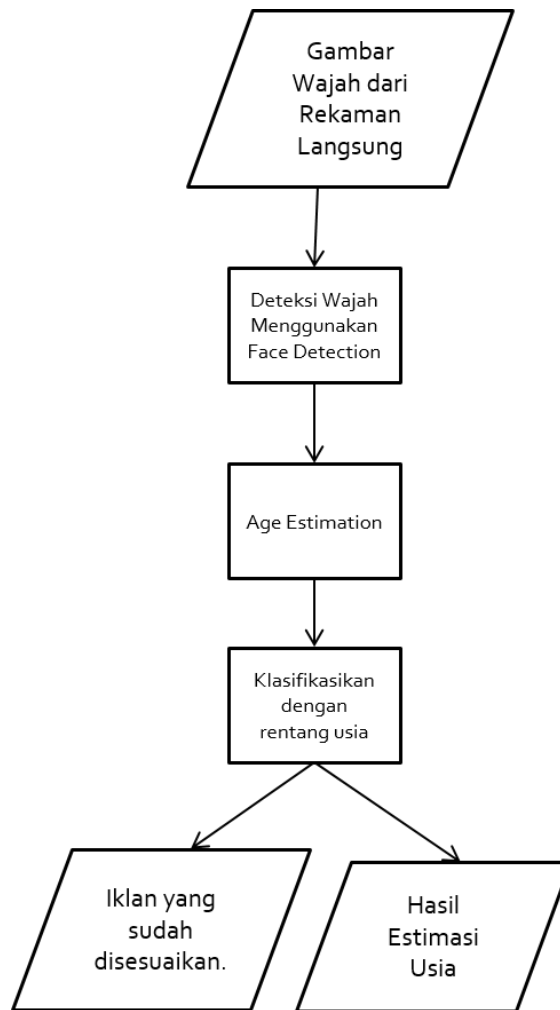
Gambar 3.6 Alur Pembangunan Model Age estimation



Gambar 3.7 Alur *Data Augmentation* Untuk Membangun Dataset B

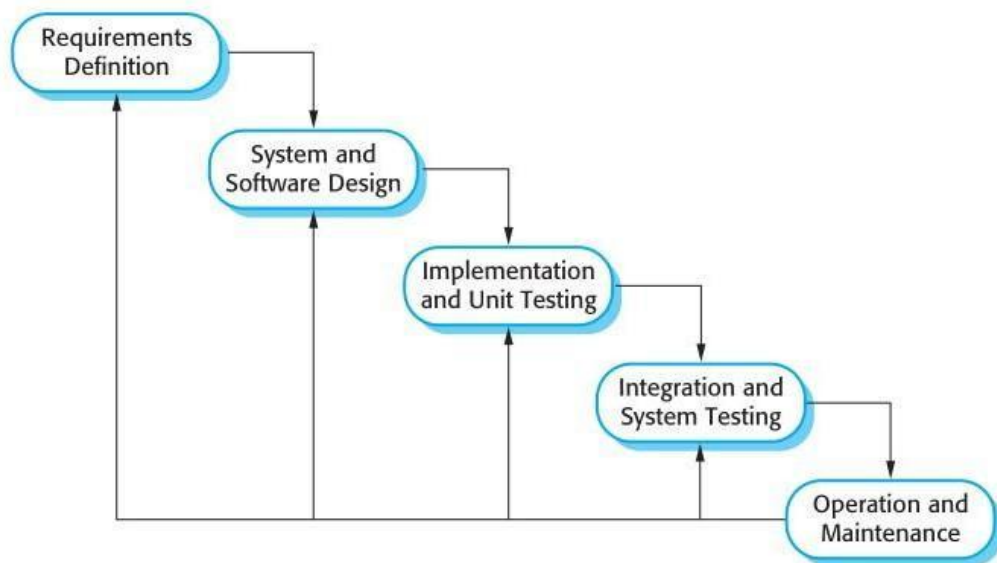
3.1.4. Analisis dan Pembangunan Aplikasi

Setelah model pendeteksi sudah dibangun, aplikasi sudah bisa dibangun untuk menampilkan atau mengganti iklan kepada *audience* apabila wajah dan usia yang berbeda terdeteksi alam jeda waktu tertentu. Alur kerja dari program dapat dilihat pada Gambar 3.8 yang dibangun menggunakan bahasa pemrograman python. Aplikasi membutuhkan input dari rekaman langsung berupa gambar di setiap frame yang tertangkap kamera. Gambar tersebut kemudian dilakukan *face detection* menggunakan *library* openCV untuk mengambil gambar wajah dari hasil rekaman. Setelah itu, gambar wajah dimasukkan ke dalam model untuk dilakukan *age estimation* dengan output berupa kelas usia untuk diklasifikasikan lagi sesuai rentang usia yang sudah ditentukan. Aplikasi ini menampilkan hasil *face detection* dengan hasil *age estimation* dari wajah yang terdeteksi dan tampilan iklan yang sudah disesuaikan.



Gambar 3.8 Alur Kerja Program *Intelligent Advertising*

Metode perangkat lunak yang akan digunakan dalam penelitian ini yaitu metode *waterfall*. Metode ini bekerja dengan merencanakan dan menjadwalkan semua kegiatan pengembangan yang terdiri dari penentuan persyaratan, pendesainan sistem dan perangkat lunak, penerapan dan unit testing, penggabungan dan pengujian sistem, serta *maintenance* dan operasi kemudian kembali ke awal seperti pada Gambar 3.9 apabila terjadi ketidaksesuaian seperti yang diharapkan.



Gambar 3.9 Alur Waterfall (Somerville, 2011)

Adapun penerapan metode waterfall terhadap desain penelitian pada Gambar 3.1 adalah sebagai berikut.

1. Penentuan persyaratan terjadi pada tahap studi pustaka, pengumpulan dan pengelolaan dataset, serta analisis dan pembangunan model baik itu untuk model generatif maupun model *age estimation*.
2. Pendesainan sistem dan perangkat lunak dilakukan pada analisis dan pembangunan aplikasi.
3. Tahap implementasi dan unit testing terjadi pada tahap evaluasi setelah model generative dan *age estimation* sudah dibangun, dan aplikasi akan diuji setelah evaluasi terhadap model *age estimation* selesai.
4. Tahap penggabungan dan pengujian sistem dilakukan pada tahap unit testing.
5. Tahap maintenance dan operasi dilakukan apabila hasil dari testing layak atau tidak. Jika tidak layak alur akan kembali lagi ke tahap penentuan persyaratan.

3.1.5. Implementasi Aplikasi

Implementasi aplikasi dilakukan dengan mengikuti cara kerja yang disebutkan pada Subbab 3.1.4. Aplikasi akan menampilkan atau mengganti iklan yang sesuai dari hasil *age estimation* dari wajah yang didapatkan dari hasil deteksi face detection. Aplikasi akan diimplementasikan terhadap seorang *audience* secara langsung

dengan jarak yang diwakili oleh resolusi gambar wajah yang sudah ditentukan.

3.1.6. Evaluasi

Evaluasi dilakukan setelah model generatif, model *age estimation* dan aplikasi dibuat. Tujuannya untuk mengetahui kualitas dan akurasi dari model yang dihasilkan serta keberhasilan dari aplikasi *intelligent advertising*.

1. Model Generatif

Sampai saat ini belum ada metode baku untuk mengevaluasi model generatif dengan baik (Borji, 2018), namun model generatif dapat dievaluasi secara kuantitatif dan kualitatif. Penilaian kuantitatif dilakukan menggunakan metode Inception Score dan FID dimana contohnya dapat dituliskan pada Tabel 3.5 sebagai perbandingan kualitas gambar yang dihasilkan (Salimans, 2016) untuk setiap model yang dihasilkan pada epoch tertentu setelah proses training selesai. Metode ini menggunakan arsitektur jaringan Inception v3 untuk menilai kualitas gambar yang dihasilkan model generatif (Salimans dkk, 2016). Aspek penilaian inception score meliputi kualitas gambar yang dihasilkan dibandingkan dengan gambar aslinya dan keberagaman gambar dari semua kelas memiliki nilai entropi yang tinggi (Salimans dkk, 2016). Pada pengujian ini, semakin dekat besar skor dengan jumlah kelas semakin baik generator. Sedangkan FID memiliki skala penilaian semakin kecil besar skor, semakin baik generator. FID dilakukan dengan membandingkan gambar dari dataset asli dengan gambar yang dihasilkan dari model generatif.

Penilaian kualitatif digunakan apabila skor FID dan inception score tidak sesuai dengan kualitas dari sampel gambar yang dihasilkan. Penilaian ini menggunakan mata manusia secara manual oleh penulis dengan membandingkan setiap sampel gambar dari setiap *epoch* dari model yang dibandingkan. Hasil dari penilaian ini menjadi pertimbangan pemilihan model untuk *data augmentation* apabila ada penurunan kualitas gambar saat mencapai akhir *epoch* meskipun memiliki skor FID dan inception score yang lebih baik, namun epoch sebelumnya memiliki kualitas sampel gambar yang lebih baik.

Tabel 3.5 Contoh Tabel Evaluasi Model Generatif ACGAN

Eksperimen	Epoch Model Generatif	Inception Score \pm Deviasi	Skor FID
1	Epoch ke-10	1,000 \pm 1,02e-02	100
1	Epoch ke-500	1,050 \pm 1,04e-02	500
2	Epoch ke-5000	2,000 \pm 2,04e-03	400

2. Model *Age estimation* dan Aplikasi Intelligent Advertising

Pengujian setiap model *age estimation* dilakukan setelah semua tahap training selesai. Pengujian diawali dengan mendeteksi wajah menggunakan *face detection* kemudian dilakukan proses estimasi terhadap data testing. Hasil estimasi setiap model akan ditampilkan dalam bentuk tabel seperti Tabel 3.6 yang menampilkan usia asli dan usia estimasi setiap model yang dibangun. Hasil training dari setiap model *age estimation* akan diukur menggunakan metrik akurasi sedangkan tahap data testing diukur menggunakan metrik MAE dan *cumulative score* (CS) menggunakan persamaan 2.1 dan 2.2 berdasarkan hasil Tabel 3.6 sehingga menampilkan hasil akhir seperti Tabel 3.7. Nilai SE pada Tabel 3.7 menggunakan nilai *standard error of estimation*.

Kemampuan model *age estimation* juga akan diuji sebagai bagian dari implementasi aplikasi untuk menguji kemampuan klasifikasi terhadap *audience* secara langsung. Dengan demikian, pengujian dari model *age estimation* bukan hanya menggunakan data testing, juga menggunakan *data-on-the-wild* berupa *audience* langsung.

Tabel 3.6 Contoh Hasil Estimasi Usia Menggunakan Gambar Wajah

Gambar	Usia Asli	Usia Estimasi

Wajah		Model 1	Model 2	Model 3	Model 4
a.jpg	10	10	29	15	9
b.jpg	25	78	27	20	80
c.jpg	50	15	30	56	45
d.jpg	80	12	65	80	80

Tabel 3.7 Contoh Hasil Pengukuran MAE dan *Cumulative Score* Model

Model	MAE	CS \pm SE
Model 1	25,5	45% \pm 4,5
Model 2	14,24	50% \pm 7,5
Model 3	17,45	75% \pm 15,5
Model 4	18,19	45% \pm 4,6

3.1.7. Analisis dan Penarikan Kesimpulan

Pada tahap ini hasil evaluasi akan dilakukan analisis hasil yang dapat ditampilkan dalam bentuk tabel agar terlihat perbandingan dari hasil estimasi usia seperti pada Tabel 3.7 yang berasal dari hasil pengukuran MAE dan akurasi CS dari setiap model *age estimation* terhadap gambar data testing sesuai Tabel 3.6. Selain itu, analisis juga dilakukan dengan hasil inception score dan FID dari model generatif dari kedua hyperparameter seperti Tabel 3.5 serta hasil implementasi model *age estimation* melalui intelligent advertising untuk dilakukan penarikan kesimpulan dari hasil terbaik setiap model yang dihasilkan.

3.1.8. Dokumentasi

Tahap terakhir dari penelitian ini adalah dokumentasi terhadap penelitian yang disusun dalam bentuk skripsi, dokumen teknis perangkat lunak, dan paper.

3.2. Alat dan Bahan Komputasi

Penelitian ini dilakukan dengan menggunakan beberapa perangkat atau alat penelitian baik perangkat keras maupun perangkat lunak. Berikut adalah spesifikasi perangkat yang digunakan dalam penelitian ini.

A. Spesifikasi Perangkat Keras

Galih Abdul Muhyi, 2020

AGE ESTIMATION UNTUK INTELLIGENT ADVERTISING PADA POSTER DIGITAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

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

Perangkat keras yang digunakan pada penelitian ini adalah komputer dengan spesifikasi sebagai berikut.

- Intel Core i7-6900 3,20 GHz
- Harddisk 2 TB
- RAM 64 GB
- NVIDIA GeForce GTX 1080 Ti

B. Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut.

- Sistem Operasi Windows 10 64 bit
- Python 3.6
- Python IDLE *shell* dan *editor*
- OpenCV
- *Framework* Tensorflow versi 2.0