

BAB IV HASIL DAN PEMBAHASAN

4.1 Spesifikasi Perangkat keras

Dalam pelaksanaannya berikut adalah perangkat keras yang digunakan:

1. Processor : AMD Ryzen 7 4800h
2. RAM : 16 Gb
3. SSD : 512 Gb
4. Monitor : 15.6-inch, FHD (1920 x 1080) 16:9, anti-glare display, sRGB:62.5%, Adobe:47.1%, Refresh Rate:144Hz, Value IPS-level, Optimus
5. VGA : Nvidia GTX 1600 Ti 6Gb Vram, 1536 CUDA Core.
Card

4.2 Spesifikasi perangkat lunak

1. Sistem Operasi : Pop! OS 21.04 64 Bit. (Based on Ubuntu/Debian)
2. Bahasa Pemrograman : Python 3
3. Interface : Jupyter Notebook dengan Lokal Runtime
4. Browser : Linux Google Chrome 103.0.5060.134 (Official Build) (64-bit)

4.3 Data

Data yang dimaksud adalah berupa gambar wajah dari guru atau karyawan yang sudah purnabakti dari SD Immanuel Bandarlampung. Data berupa gambar wajah ini didapatkan dari berbagai sumber dan memiliki kualitas tidak baik. Proses pengumpulan data yang terjadi bergantung dengan kebutuhan akan pengolahan data. Berikut adalah contoh dari gambar seorang pensiunan guru yang didapat dari profil *Whatsapp* putrinya. Hal ini karena tidak ada foto formal yang dimiliki dan Guru yang bersangkutan sudah usia lanjut sehingga tidak dalam kapasitas mampu menggunakan ponsel pintar. Berikut merupakan data berupa gambar wajah yang akan diperbaiki yang dapat dilihat pada gambar 4.1



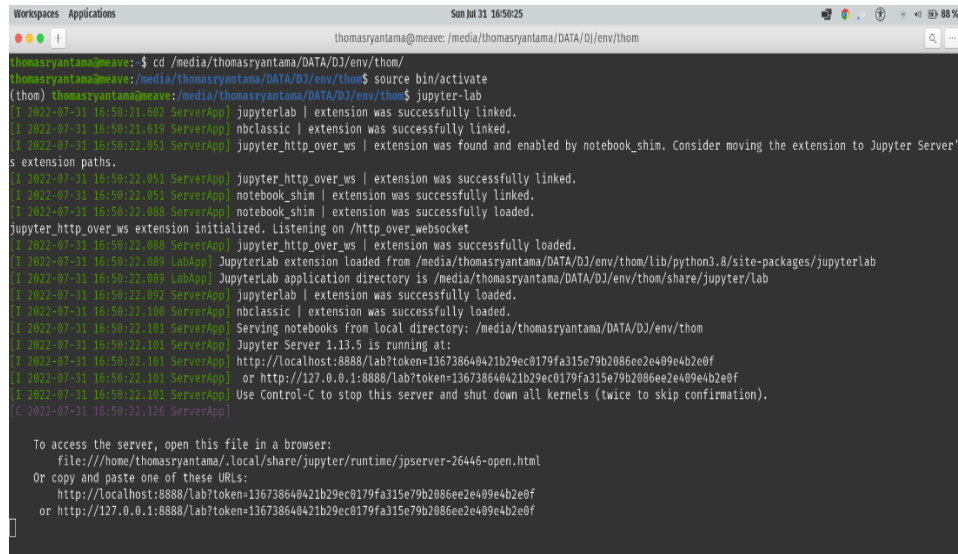
Gambar 4.1 Dataset

4.4 Implementasi

Pada tahap implementasi dan pengujian didapati hasil sebagai berikut:

4.4.1 Persiapan

Seluruh *Environment* dijalankan di atas Python3 yang diisolasi melalui python *enviroment*. Hal ini bertujuan agar sistem yang dijalankan juga termasuk dependensi yang akan dibutuhkan tidak mengganggu fungsi sistem operasi. Dengan melakukan isolasi maka aplikasi yang akan dijalankan berada dalam lingkungan yang privat dan terpisah dari sistem operasi dan python yang digunakan oleh sistem operasi. Jadi ketika terjadi kesalahan atau eror, sistem operasi masih aman dan tidak terganggu. Dan berikut adalah cuplikan layar python *environment* yang dijalankan yang dapat dilihat pada gambar 4.2 dibawah ini.



```

thomasryantama@meave: /media/thomasryantama/DATA/DJ/env/thom
thomasryantama@meave:~$ cd /media/thomasryantama/DATA/DJ/env/thom/
thomasryantama@meave:/media/thomasryantama/DATA/DJ/env/thom$ source bin/activate
(thom) thomasryantama@meave:/media/thomasryantama/DATA/DJ/env/thom$ jupyter-lab
[2022-07-31 16:58:21.602 ServerApp] jupyterlab | extension was successfully linked.
[2022-07-31 16:58:21.619 ServerApp] nbclassic | extension was successfully linked.
[2022-07-31 16:58:22.051 ServerApp] jupyter_http_over_ws | extension was found and enabled by notebook_shim. Consider moving the extension to Jupyter Server's extension paths.
[2022-07-31 16:58:22.051 ServerApp] jupyter_http_over_ws | extension was successfully linked.
[2022-07-31 16:58:22.051 ServerApp] notebook_shim | extension was successfully linked.
[2022-07-31 16:58:22.048 ServerApp] notebook_shim | extension was successfully loaded.
jupyter_http_over_ws extension initialized. Listening on /http_over_websocket
[2022-07-31 16:58:22.088 ServerApp] jupyter_http_over_ws | extension was successfully loaded.
[2022-07-31 16:58:22.089 LabApp] JupyterLab extension loaded from /media/thomasryantama/DATA/DJ/env/thom/lib/python3.8/site-packages/jupyterlab
[2022-07-31 16:58:22.089 LabApp] JupyterLab application directory is /media/thomasryantama/DATA/DJ/env/thom/share/jupyter/lab
[2022-07-31 16:58:22.092 ServerApp] jupyterlab | extension was successfully loaded.
[2022-07-31 16:58:22.100 ServerApp] nbclassic | extension was successfully loaded.
[2022-07-31 16:58:22.101 ServerApp] Serving notebooks from local directory: /media/thomasryantama/DATA/DJ/env/thom
[2022-07-31 16:58:22.101 ServerApp] Jupyter Server 1.13.5 is running at:
[2022-07-31 16:58:22.101 ServerApp] http://localhost:8888/lab?token=136738640421b29ec0179fa315e79b2086ee2e409e4b2e0f
[2022-07-31 16:58:22.101 ServerApp] or http://127.0.0.1:8888/lab?token=136738640421b29ec0179fa315e79b2086ee2e409e4b2e0f
[2022-07-31 16:58:22.101 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
C 2022-07-31 16:58:22.126 ServerApp]

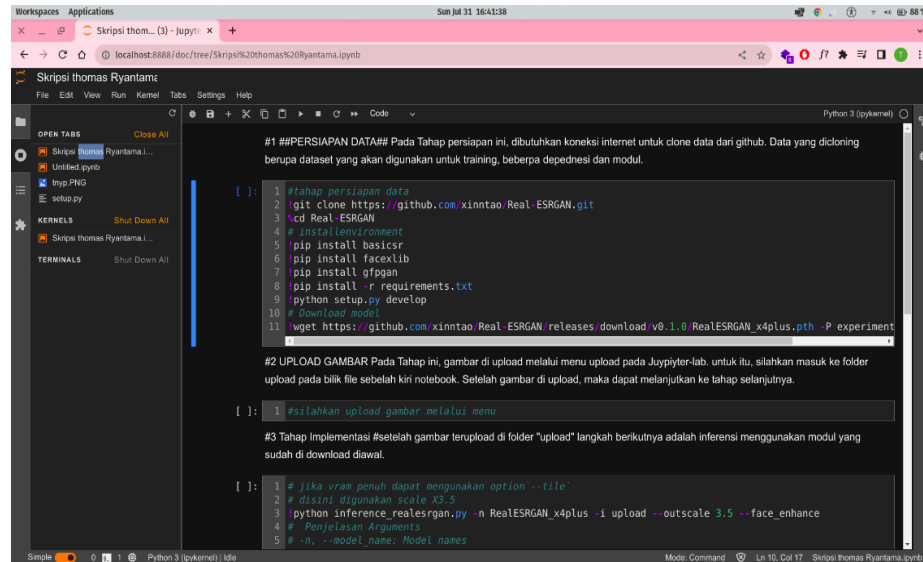
To access the server, open this file in a browser:
file:///home/thomasryantama/.local/share/jupyter/runtime/jpservers-26446-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=136738640421b29ec0179fa315e79b2086ee2e409e4b2e0f
or http://127.0.0.1:8888/lab?token=136738640421b29ec0179fa315e79b2086ee2e409e4b2e0f

```

Gambar 4.2 Running Jupyter Lab

4.4.2 Interface

Dengan interface menggunakan notebook. Implementasi ini memanfaatkan browser Google Chrome yang berjalan di atas sistem operasi Linux 64 bit seperti terdapat pada gambar 4.3 berikut ini.



```

#1 #PERSIAPAN DATA## Pada Tahap persiapan ini, dibutuhkan koneksi internet untuk clone data dari github. Data yang dicloning berupa dataset yang akan digunakan untuk training, beberapa dependensi dan modul.

[ ]: 1 #tahap persiapan data
      2 git clone https://github.com/xinntao/Real-ESRGAN.git
      3 cd Real-ESRGAN
      4 # Install environment
      5 pip install basicsr
      6 pip install facefix
      7 pip install gipgan
      8 pip install -r requirements.txt
      9 python setup.py develop
     10 # Download model
     11 wget https://github.com/xinntao/Real-ESRGAN/releases/download/v0.1.0/RealESRGAN_x4plus.pth -P experiment

#2 UPLOAD GAMBAR Pada Tahap ini, gambar di upload melalui menu upload pada Jupyterlab. untuk itu, silahkan masuk ke folder upload pada bilik file sebelah kiri notebook. Setelah gambar di upload, maka dapat melanjutkan ke tahap selanjutnya.

[ ]: 1 #silahkan upload gambar melalui menu

#3 Tahap Implementasi #setelah gambar terupload di folder "upload" langkah berikutnya adalah inferensi menggunakan modul yang sudah di download diawal.

[ ]: 1 # jika vram penuh dapat menggunakan option --tile
      2 # disini digunakan scale X3.5
      3 python inference_realesrgan.py -n RealESRGAN_x4plus -i upload --outscale 3.5 --face_enhance
      4 # Penjelasan Arguments
      5 # -n, --model_name: Model names

```

Gambar 4.3 Interface

4.4.3 Running Cells

Dalam perancangannya, proses ini dibagi menjadi empat bagian yang harus dimulai secara berurutan. Keempat proses itu adalah:

4.4.3.1 Persiapan

Pada tahap ini, dilakukan persiapan dari sistem yang akan dijalankan. Dibutuhkan koneksi internet untuk kloning data dari github. Data yang dikloning berupa dataset yang akan digunakan untuk training, beberapa dependensi dan modul. Beberapa dependensi yang dibutuhkan seperti : numpy, opencv-python, torch, torchvision

4.4.3.2 Upload Gambar

Upload gambar dilakukan menggunakan menu upload yang ada pada jupyter-lab. Melalui menu ini, user dapat mengupload file gambar yang ingin diperbaiki.

4.4.3.3 Inferensi atau Implementasi

Tahap inferensi adalah tahap pemrosesan gambar yang telah diupload. Dan hasilnya akan muncul pada folder result. Proses yang dilakukan disini menggunakan modul-modul yang sudah di download pada tahap persiapan. Inferensi yang dilakukan berkaitan dengan penghilangan atau pengurangan degradasi pada citra yang terjadi karena proses akuisisi citra. Degradasi yang dimaksud termasuk derau (yang merupakan eror dalam nilai piksel) atau efek optis.









4.4.3.4 Hasil

Tahap terakhir adalah hasil. Di tahap ini hasil pemrosesan dapat dilihat dan digunakan oleh user. User dapat langsung melihat hasil dari jupyter-lab atau dapat juga masuk ke folder result dan menggunakan gambar yang sudah diproses.











4.5 Hasil Penelitian

Setelah penerapan terhadap gambar melalui serangkaian proses di atas, maka didapatkan hasil perbandingan ukuran gambar sebelum dan sesudah dilakukan perbaikan yang dijelaskan dalam tabel 4.1 berikut ini:



Tabel 4.1 Perbandingan

| No | Original | | Sesudah Proses Perbaikan | |
|----|---|-----------|--|-------------|
| | Gambar | Ukuran | Gambar | Ukuran |
| 1 |  | 144x292px |  | 504x1022px |
| 2 |  | 149x156px |  | 521x546px |
| 3 |  | 320x372px |  | 1130x1302px |
| 4 |  | 355x543px |  | 1242x1900px |

Tabel 4.1 Perbandingan (Lanjutan)

| No | Original | | Sesudah Proses Perbaikan | |
|----|---|-----------|--|-------------|
| | Gambar | Ukuran | Gambar | Ukuran |
| 5 |  | 607x900px |  | 2124x3580px |
| 6 |  | 261x361px |  | 913x1123px |
| 7 |  | 384x474px |  | 1344x1659px |
| 8 |  | 700x700px |  | 3780x3780px |
| 9 |  | 681x880px |  | 3363x4900px |

Tabel 4.1 Perbandingan (Lanjutan)

| No | Original | | Sesudah Proses Perbaikan | |
|----|---|-----------|--|-------------|
| | Gambar | Ukuran | Gambar | Ukuran |
| 10 |  | 716x712px |  | 3192x3900px |

4.6 Pembahasan

Berdasarkan data hasil penelitian maka dapat diketahui bahwa terjadi perubahan dari gambar yang sudah mengalami proses perbaikan. Peningkatan itu berupa ukuran kepadatan pixel yang meningkat dan yang secara langsung berpengaruh dengan kejelasan gambar yang diproses.

Sebagai contoh seperti yang ditunjukkan pada gambar nomor 2 dari tabel 4.1 di atas dengan resolusi orisinal 149x156px setelah melalui proses perbaikan gambar meningkat menjadi 521x546px.

Kendala yang dihadapi selama proses yang dilakukan adalah proses *cleaning* data masih dilakukan manual dengan *cropping* gambar secara manual menggunakan aplikasi pengolahan grafis terpisah seperti Ms Paint. Hal ini dapat menghambat proses karena memerlukan waktu yang lebih lama dan membutuhkan ketelitian. Selain itu ketika didapati bahwa gambar yang akan diproses memiliki resolusi yang lebih besar dari batasan, didapati bahwa memori VRAM penuh dan proses berhenti tanpa hasil. Hal ini dapat diantisipasi dengan mengompres gambar secara manual atau memotong gambar hingga ukurannya sesuai. Namun jika tidak memungkinkan untuk melakukan kompresi atau pemotongan gambar langkah yang bisa dilakukan adalah melakukan proses di komputer dengan spesifikasi lebih tinggi.