

BAB III

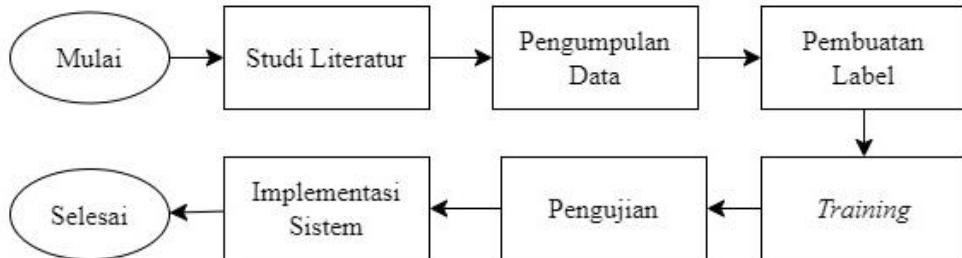
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini berupa citra dan data yang digunakan pada penelitian ini adalah foto wajah Satgas Penanggulangan Bencana di BPBD Karawang, banyaknya sampel yang diambil yaitu sebanyak 5 orang. Setiap Satgas Penanggulangan Bencana akan dilakukan pengambilan gambar sebanyak 100 gambar.

3.2 Prosedur Penelitian

Tahapan prosedur penelitian dari Tugas Akhir ini yang akan dilakukan dapat digambarkan sebagai berikut.



Gambar 3.1 Diagram Alur Prosedur Penelitian

3.3.1 Studi Literatur

Studi literatur merupakan tahapan penting dalam prosedur penelitian ini, yang melibatkan eksplorasi mendalam untuk memperoleh landasan teori yang solid serta referensi yang relevan. Proses pencarian meliputi identifikasi konsep-konsep kunci, pengertian mendasar, serta teori-teori terkait yang akan menjadi dasar bagi tinjauan pustaka. Referensi dari berbagai macam sumber ini akan membantu menggambarkan latar belakang dan konteks penelitian dengan lebih komprehensif. Analisis ini penting untuk mengidentifikasi kekosongan pengetahuan yang masih ada atau peluang-peluang penelitian lebih lanjut.

Tidak hanya itu, studi literatur juga akan mengungkapkan berbagai metode yang telah diterapkan dalam penelitian sebelumnya. Dengan memahami metode-metode tersebut, peneliti dapat merencanakan

pendekatan yang tepat untuk mengumpulkan dan menganalisis data dalam penelitian Tugas Akhir ini.

3.3.2 Pengumpulan Data

Pengumpulan data pada penelitian ini, *dataset* yang digunakan terdiri dari 100 citra wajah dari setiap anggota satuan tugas penanggulangan bencana. Pengumpulan data dilakukan dengan cara melakukan pemotretan manual menggunakan kamera smartphone. Setiap anggota akan memiliki folder sendiri yang berisi citra-citra wajah mereka. Struktur folder ini akan memiliki dua subfolder, yaitu train dan val. Dalam subfolder train, akan disimpan sekitar 70% dari total citra yang diambil, sedangkan subfolder val akan berisi sekitar 30 sisanya. Data latih akan digunakan oleh model untuk belajar dan mengidentifikasi pola-pola wajah yang berbeda, sementara data validasi akan digunakan untuk mengukur kinerja model pada data yang belum pernah dilihat sebelumnya.

3.3.3 Pembuatan Label

Labeling *dataset* atau dapat disebut pelabelan adalah proses pemberian informasi pada objek. Proses pelabelan ini dilakukan pada setiap gambar yang telah dikumpulkan. Seluruh gambar yang telah terkumpul sebelumnya diberikan label sebagai identitas agar dapat memuat nama gambar. Proses pelabelan ini dilakukan dengan cara membuat nama kelas dan *bounding box* pada setiap gambar. Pada proses pelabelan ini menggunakan layanan dari situs makesense.ai agar mempercepat proses pelabelan dan menjadikan *dataset* yang siap digunakan di YOLOv5. Setelah melalui proses akuisisi dan dikumpulkan kedalam folder yang dibuat agar dapat menjadi *dataset* yang siap digunakan untuk deteksi objek.

3.3.4 Training

Data yang sudah diberi label pada tahap sebelumnya akan dilatih sehingga membentuk sebuah pola yang hasilnya berbentuk bobot baru. Terdapat 5 varian model pada YOLOv5, pada penelitian ini model yang akan digunakan merupakan varian model YOLOv5s. Model YOLOv5s memiliki parameter yang ringan sehingga deteksi dan proses komputasi dapat dilakukan lebih cepat dengan akurasi mAP (mean-average *Precision*) tetap

baik. Jenis dan perbandingan varian model YOLOv5 dapat dilihat pada gambar 3.2.

|  |  |  |  |  |
|---|---|---|---|---|
| Nano YOLOv5n | Small YOLOv5s | Medium YOLOv5m | Large YOLOv5l | XLarge YOLOv5x |
| 4 MB _{FP16} 6.3 ms _{V100} 28.4 mAP _{coco} | 14 MB _{FP16} 6.4 ms _{V100} 37.2 mAP _{coco} | 41 MB _{FP16} 8.2 ms _{V100} 45.2 mAP _{coco} | 89 MB _{FP16} 10.1 ms _{V100} 48.8 mAP _{coco} | 166 MB _{FP16} 12.1 ms _{V100} 50.7 mAP _{coco} |

Gambar 3.2 Varian Model YOLOv5

```
# YOLOv5 🚀 by Ultralytics, AGPL-3.0 license

# Parameters
nc: 16 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple
anchors:
    - [10,13, 16,30, 33,23] # P3/8
    - [30,61, 62,45, 59,119] # P4/16
    - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 v6.0 backbone
backbone:
    # [from, number, module, args]
    [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
     [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
     [-1, 3, C3, [128]],
     [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
     [-1, 6, C3, [256]],
     [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
     [-1, 9, C3, [512]],
     [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
     [-1, 3, C3, [1024]],
     [-1, 1, SPPF, [1024, 5]], # 9
    ]

# YOLOv5 v6.0 head
head:
    [[-1, 1, Conv, [512, 1, 1]],
     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
     [[-1, 6], 1, Concat, [1]], # cat backbone P4
     [-1, 3, C3, [512, False]], # 13

     [-1, 1, Conv, [256, 1, 1]],
     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
     [[-1, 4], 1, Concat, [1]], # cat backbone P3
     [-1, 3, C3, [256, False]], # 17 (P3/8-small)

     [-1, 1, Conv, [256, 3, 2]],
     [[-1, 14], 1, Concat, [1]], # cat head P4
     [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

     [-1, 1, Conv, [512, 3, 2]],
     [[-1, 10], 1, Concat, [1]], # cat head P5
     [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

     [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5
    ]]
```

Gambar 3.3 Arsitektur YOLOv5s

Dalam arsitektur YOLOv5s seperti pada Gambar 3.3, nilai "depth_multiple" dan "width_multiple" mempengaruhi kedalaman dan lebar arsitektur. Semakin kecil nilai "depth_multiple", maka arsitektur akan semakin dangkal. Semakin kecil nilai "width_multiple", maka jumlah filter pada setiap layer akan semakin sedikit. Sementara *nc* adalah total jumlah dari kelas. Proses komputasi menggunakan YOLOv5s ini lebih cepat dengan akurasi mAP (mean-average *Precision*) tetap baik.

Pada bagian *anchors*, YOLOv5 ditetapkan untuk setiap fitur map dengan tiga ukuran (80×80 , 40×40 , 20×20), yaitu 10×13 , 16×30 , 33×23 , 30×61 , 62×45 , 59×119 ; 116×90 , 156×198 , 373×326 masing-masing, dan tiga fitur map adalah input ke lapisan deteksi multi-skala, masing-masing digunakan untuk mendekripsi objek kecil, sedang, dan besar.

Ada 4 elemen pada arsitektur YOLOv5s yaitu "from" merujuk pada layer sebelumnya yang digunakan sebagai input untuk layer saat ini, "number" merujuk pada jumlah filter yang digunakan dalam layer saat ini, "module" merujuk pada tipe layer yang digunakan dalam arsitektur YOLOv5, "args" merujuk pada argumen yang diperlukan oleh layer saat ini.

Pada Gambar 3.6 bagian *backbone*, terdapat beberapa elemen arsitektur, dimana sebagai contoh `[-1, 1, Conv, [128, 3, 2]], # 1-P2/4` adalah layer konvolusi dengan 128 filter, kernel size 3x3 dan stride 2. "-1" menunjukkan bahwa layer ini menggunakan output dari layer sebelumnya sebagai input. `[-1, 1, Conv, [64, 6, 2, 2]]` adalah layer konvolusi dengan 64 filter dan kernel size 6x6 dan stride 2. Layer ini digunakan untuk mengekstrak fitur dari gambar. `[-1, 6, C3, [256]]` adalah layer konvolusi dengan 256 filter dan kernel size 3x3. `[-1, 1, SPPF, [1024, 5]], # 9` adalah layer SPPF (Spatial Pyramid Pooling) dengan output 1024 dan pyramid level 5. Layer ini digunakan untuk menghasilkan fitur yang lebih baik dari gambar dengan ukuran yang berbeda-beda.

Bagian *head* pada Gambar 3.3 memiliki beberapa elemen arsitektur *head*, seperti `[-1, 1, Conv, [512, 1, 1]]` yaitu layer konvolusi dengan 512 filter dan kernel size 1x1. `[-1, 1, nn.Upsample, [None, 2, 'nearest']]` yaitu layer *upsampling* dengan faktor 2 dengan metode interpolasi untuk memperbaiki [[-

[1, 6], 1, Concat, [1]] yaitu layer *concatenation* dengan output dari layer sebelumnya. [-1, 3, C3, [512, False]] yaitu layer konvolusi dengan 512 filter dan kernel size 3x3. [[17, 20, 23], 1, Detect, [nc, anchors]] yaitu layer deteksi objek dengan menggunakan output dari beberapa layer sebelumnya sebagai input.

Pada penelitian ini, pelatihan model YOLOv5 dilakukan dengan dataset 500 data, ukuran batch 16, ukuran gambar 640, dan model YOLOv5 atau –weights menggunakan YOLOv5s. Semua hasil pelatihan akan disimpan ke dalam folder run/train/ dengan direktori run yang bertambah, yaitu run/train/exp2, run/train/exp3 dan lainnya. Proses *training* dilakukan dengan 100 epoch. Hasil *training* yang muncul akan berupa hasil training terbaik (best.pt), hasil *training* terakhir (last.pt) dan metrics/Precision, metrics/Recall, metrics/mAP0.5, val/box_loss, val/cls_loss serta hasil train.

```
keys = ('metrics/precision', 'metrics/recall',
        'metrics/mAP_0.5',
        'metrics/mAP_0.5:0.95',
        'val/box_loss', 'val/obj_loss', 'val/cls_loss')
print_mutation(keys, results, hyp.copy(),
```

Gambar 3.4 Kode Output Train

3.3.5 Pengujian

Pada tahapan ini, dilakukan pengujian model yang dihasilkan dari proses *training*, akan diuji untuk menentukan apakah sistem dapat bekerja dengan baik sesuai dengan yang diharapkan. Tahapan pengujian ini mirip dengan tahapan prediksi dan proses pengenalan. Pada penelitian ini, tahap pengujian terdiri dari pengujian terhadap satu orang yang ada di dalam *dataset*, pengujian terhadap satu orang yang tidak terdapat dalam *dataset*, Pengujian gabungan lebih dari satu orang yang ada di dalam *dataset* dan tidak ada di dalam *dataset*, dan sinkronisasi *dataset* satuan tugas penanggulangan bencana sistem presensi pada BPBD Karawang. Dalam proses pengujian dilakukan dengan menggunakan *confusion matrix* dengan rumus sebagai berikut.

$$Akurasi : \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

Keterangan:

TP = *True Positif*

FP = *False Positif*

TN = *True Negatif*

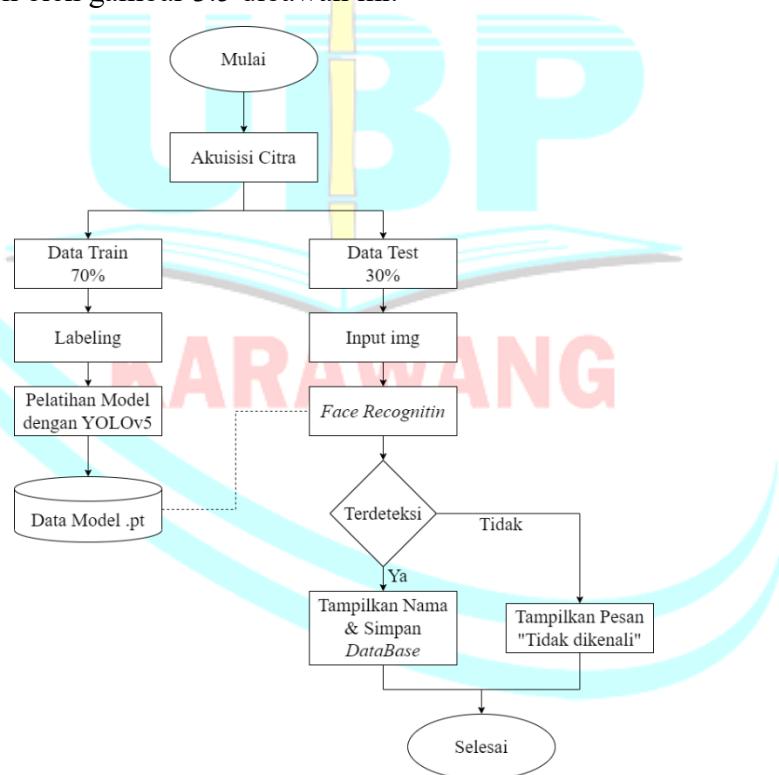
FN = *False Negatif*

3.3.6 Implementasi Sistem

Pada tahap ini, penulis melaksanakan pembuatan perancangan-perancangan berdasarkan proses usulan yang telah diidentifikasi sebelumnya. Berikut adalah beberapa perancangan yang telah dirancang dalam rangka mengembangkan solusi yang sesuai:

3.3.6.1 Alur Sistem

Berikut rancangan sistem yang digunakan dalam penelitian ini ditunjukkan oleh gambar 3.5 dibawah ini.



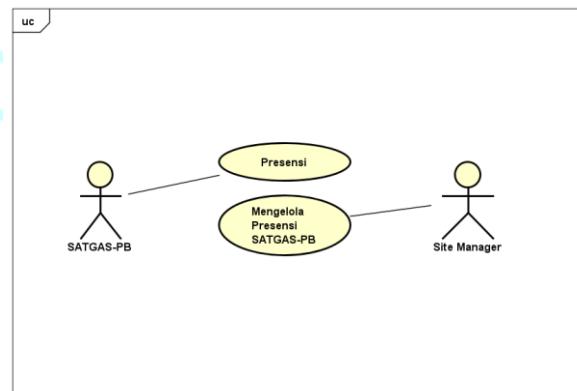
Gambar 3.5 Alur Sistem

Pada gambar 3.5 diatas menjelaskan alur sistem dalam mengenali wajah (*face recognition*) yakni dimulai dengan *input* wajah melalui kamera. Kemudian *dataset* dibagi menjadi dua dengan komposisi 30% data uji dan 70% data latih. Sistem akan

membandingkan dengan referensi wajah yang ada pada data latih. Setelah identifikasi selesai, maka sistem akan memberikan hasil identifikasi wajah pada *website*, seperti nama individu dan menyimpan ke *database*.

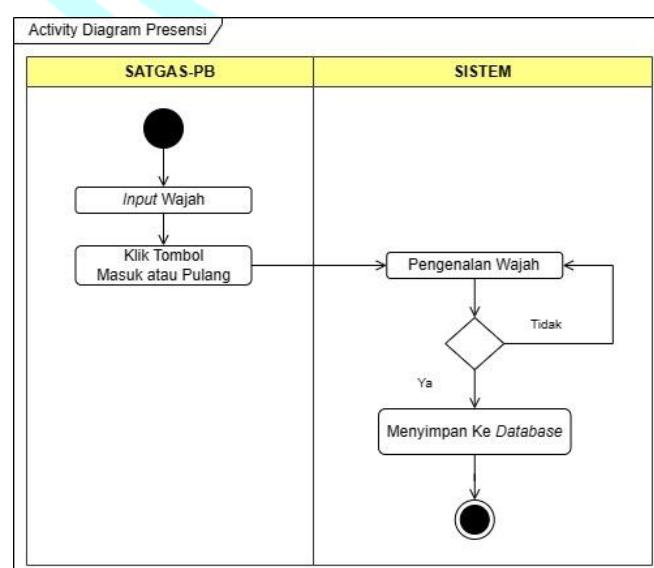
3.3.6.2 Diagram

Berikut adalah diagram yang digunakan dalam penelitian.



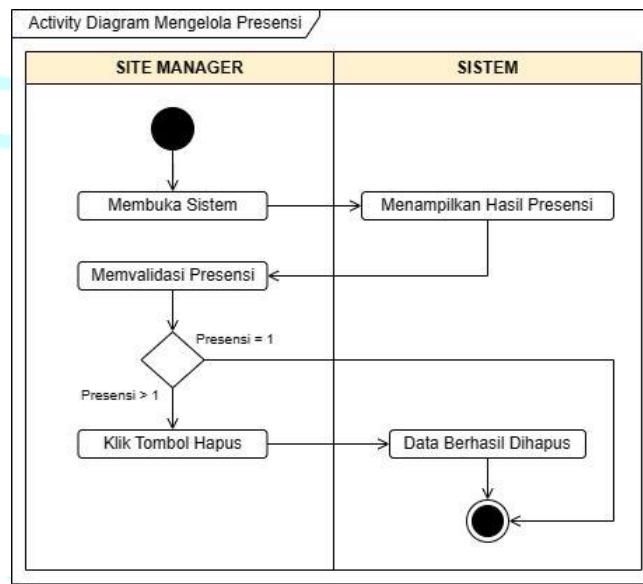
Gambar 3.6 UseCase Diagram

Pada Gambar 3.6 diatas terdapat dua user yaitu satuan tugas penanggulangan bencana dan site manager. satuan tugas penanggulangan bencana dapat melakukan presensi, selain itu site manager dapat mengelola presensi satuan tugas penanggulangan bencana.



Gambar 3.7 Activity Diagram Presensi

Pada Gambar 3.7 satuan tugas penanggulangan bencana melakukan presensi dengan menginputkan wajah, kemudian memilih tombol masuk atau tombol pulang, lalu sistem akan mengidentifikasi wajah. Jika wajah terdeteksi maka sistem akan menampilkan nama dan menyimpan ke dalam *database*, jika wajah tidak terdeteksi maka sistem akan memunculkan pesan tidak dikenali.



Gambar 3.8 Activity Diagram Mengelola Presensi

Pada Gambar 3.8 site manager memulai sistem sehingga hasil presensi semua satuan tugas penanggulangan bencana ditampilkan oleh sistem. Lalu site manager dapat memvalidasi dan menghapus data jika presensi lebih dari satu.

3.3.6.3 Desain Tampilan

Pada Gambar 3.9 dibawah terdapat tampilan presensi masuk untuk satuan tugas penanggulangan bencana, pada halaman tersebut terdapat tombol hapus yang berfungsi menghapus apabila presensi tercatat ganda. Dengan adanya halaman ini, satuan tugas penanggulangan bencana dapat dengan mudah melakukan presensi masuk.

| No | Nama | Tanggal | Waktu | Foto | Status | Action |
|----|------|------------|----------|------|--------|------------------------|
| 1 | Nama | 11-01-2023 | 07:55:10 | | Masuk | <button>Hapus</button> |

Gambar 3.9 Tampilan Halaman Presensi Masuk

Pada Gambar 3.10 terdapat menu halaman presensi pulang, yang dirancang sebagai halaman presensi saat jam pulang bagi satuan tugas penanggulangan bencana. Pada halaman ini juga terdapat tombol hapus yang dapat digunakan ketika presensi pulang tercatat ganda.

| No | Nama | Tanggal | Waktu | Foto | Status | Action |
|----|------|------------|----------|------|--------|------------------------|
| 1 | Nama | 11-01-2023 | 15:35:10 | | PULANG | <button>Hapus</button> |

Gambar 3.10 Tampilan Halaman Presensi Pulang

Pada Gambar 3.11 terdapat halaman daftar presensi yang terdapat keterangan waktu masuk dan waktu pulang bagi satuan tugas penanggulangan bencana, terdapat sebuah tombol cetak pdf, yang memiliki fungsi untuk site manager mencetak daftar presensi.

| No | Nama | Tanggal | Jam Masuk | Foto | Jam Pulang | Foto |
|----|------|------------|-----------|------|------------|------|
| 1 | Nama | 11-01-2023 | 15:35:10 | | PULANG | |

Gambar 3.11 Halaman Daftar Presensi satuan tugas penanggulangan bencana

3.3.7 Pengujian Sistem

Pada tahap ini pengujian sistem dilakukan dengan data sebanyak 5 satuan tugas penanggulangan bencana dengan menjalankan sistem kehadiran satuan tugas penanggulangan bencana yang sudah ter-*include* model YOLOv5. Kemudian proses perhitungan akurasi sistem dilakukan dengan menggunakan rumus sebagai berikut.

$$\text{Tingkat akurasi} = \frac{\text{jumlah citra wajah yang sesuai}}{\text{jumlah uji coba citra wajah}} \times 100\%$$