

Pengenalan Wajah dengan Algoritma Local Binary Pattern Histogram Menggunakan Python

¹Shedriko, ²Muhammad Firdaus

^{1,2}Universitas Indraprasta PGRI Jakarta, Indonesia

shedriko@gmail.com, dasurichi@gmail.com

*Corresponding Author

Diajukan : 12/04/2022

Diterima : 13/04/2022

Dipublikasi : 14/04/2022

ABSTRAK

Penelitian ini bertujuan untuk pengenalan wajah dengan *algoritma local binary pattern histogram* menggunakan *python*. Penelitian ini menggunakan metode *Applied* atau Aplikatif untuk penyelesaian tulisan secara keseluruhan. Metode ini bertujuan untuk mencari solusi langsung dalam menyelesaikan masalah industri, dalam hal ini berupa penggunaan salah satu bahasa pemrograman yaitu Python. Data yang digunakan merupakan data gambar yang disimpan dalam suatu folder yang telah dibuat lebih dulu untuk kemudian dilakukan proses perekaman data dengan menggunakan bahasa python. Hasil penelitian berupa Dalam penelitian ini nilai kecocokan yang muncul adalah 59%, namun di kesempatan yang lain nilai yang muncul dapat mencapai hingga sekitar 80%. Perubahan akurasi dapat terjadi ketika objek bergerak dalam pengukuran pengenalan wajah. Kualitas kamera memiliki peran yang sangat penting yang menjadi salah satu faktor bagi keakuratan data. Keterdapatannya data *training* yang gelap seperti misalnya di awal perekaman data, dapat menjadi faktor ketidakakuratan dalam pengujian dari objek input yang misalnya diambil dalam kondisi yang juga cenderung gelap. Karena dalam bentuk histogramnya, gambar gelap akan membuat grafik mengumpul ke sebelah kiri, sehingga bila dipertemukan dengan data pengujian yang memberikan gambar histogram yang cenderung sama, maka objek akan dikatakan

Kata Kunci: Algoritma Local Binary, Pattern, Histogram, Python.

I. PENDAHULUAN

Dewasa ini, teknologi komunikasi dan informasi telah berkembang dengan sangat pesat di semua sektor. Demikian pula dengan penyebarannya yang relatif cepat dan merambah semua daerah di dunia. Teknologi ini sangat membantu tugas dan kegiatan manusia. Keberadaannya dapat meningkatkan produktifitas baik pada bidang yang telah ada sebelumnya maupun yang belum pernah ada, secara lebih efisien dan efektif. Teknologi ini juga telah menciptakan banyak lapangan kerja baru disamping juga memiliki potensi untuk menghilangkan beberapa pekerjaan lama karena sifat otomatisasinya.

Penelitian mendalam berkaitan teknologi komunikasi dan informasi juga telah dilakukan oleh banyak pihak di seluruh dunia. Dari teknologi ini telah lahir berbagai bidang ilmu pengetahuan baru, seperti artificial intelligence, machine learning, robotic, bioteknologi dan banyak lagi. Berbagai produk teknologi baru yang terkaitpun terus bermunculan, seperti teknologi pengenalan identitas, kontrol, berbagai otomatisasi, penentuan posisi dan arah dengan bantuan satelit, pengolahan citra, keamanan, robot untuk tujuan tertentu, pemrograman, spesial efek di dunia perfilman, komunikasi digital melalui berbagai *tools*, konversi terjemahan dari suara ke teks atau sebaliknya, akurasi penargetan baik diam maupun bergerak dan sebagainya.

Kelahiran salah satu produk yang sangat penting adalah teknologi pengenalan identitas, seperti identitas biometrik. Dengan pengenalan identitas, maka pemeriksaan hak akses ke tempat tertentu ataupun penggunaan suatu tools maupun sebagai tanda kehadiran akan diberlakukan. Pada teknologi termutakhir, proses pengenalan ini telah dilakukan secara digital melalui berbagai media. Salah satunya adalah pengenalan digital dengan komputasi kamera. Penelitian ini akan membahas teknologi pengenalan wajah secara sederhana menggunakan perangkat yang cukup sederhana berupa komputer beserta kameranya yang masih bertipe VGA *Web Camera* dengan menggunakan data yang sederhana pula dan pengolahannya yang menggunakan bahasa pemrograman python.

II. STUDI LITERATUR

Sistem pengenalan wajah umumnya dapat diklasifikasikan ke dalam 2 tipe model operasi, yaitu: (i) verifikasi wajah dan (ii) identifikasi wajah (Mohamed, 2013). Verifikasi wajah melibatkan pengecekan *one to one* sebagai konfirmasi yang tersimpan di sistem. Bila terkonfirmasi maka sistem akan menerima sedangkan bila tidak maka akan ditolak. Contoh dari sistem ini adalah *log-in* pada smarphone atau komputer, kontrol pagar otomatis dan e-passport. Sedangkan sistem identifikasi wajah melibatkan pengecekan *one to many*. Gambar wajah yang diambil akan dicocokkan dengan banyak gambar lainnya untuk menentukan identitas orang tersebut. Contoh dari sistem ini adalah penentuan informasi seseorang pada database kepolisian, interaksi manusia dengan komputer pada video game, sistem presensi di sekolah dan video pengamatan terhadap seseorang (Parmar and Mehta, 2014). Penelitian ini bisa dikategorikan sebagai identifikasi wajah, namun menggunakan data sederhana dari seseorang yang direkam wajahnya selama 30 detik dengan berbagai pose dengan bahasa python.

LBP secara umum dapat didefinisikan sebagai perbandingan nilai biner pixel pada pusat citra dengan 8 buah pixel di sekelilingnya (Rangkuti, 2020). Metodologi ini berkaitan erat dengan salah satu cabang keilmuan teknologi informasi, yaitu pengolahan citra/gambar. Berikut adalah hal-hal yang terkait sesuai dengan cabang keilmuan tersebut dalam penelitian ini (Kosasih and C. Daomara, 2021)..

III. METODE

Penelitian ini menggunakan metode *Applied* atau Aplikatif untuk penyelesaian tulisan secara keseluruhan (Kothari, 2014). Metode ini bertujuan untuk mencari solusi langsung dalam menyelesaikan masalah industri, dalam hal ini berupa penggunaan salah satu bahasa pemrograman yaitu Python. Sedangkan metodologi yang digunakan adalah *Local Binary Pattern* (LBP) yang dikombinasikan dengan *Histogram of Gradients* (HOG), yang selanjutnya akan kita sebut dengan *Local Binary Pattern Histogram* (LBPH), sebagai algoritma untuk penyelesaian masalah (Anantharam, 2016). Data yang digunakan merupakan data gambar yang disimpan dalam suatu folder yang telah dibuat lebih dulu untuk kemudian dilakukan proses perekaman data dengan menggunakan bahasa python.

Sistem pengenalan wajah umumnya dapat diklasifikasikan ke dalam 2 tipe model operasi, yaitu: (i) verifikasi wajah dan (ii) identifikasi wajah (Mohamed, 2013). Verifikasi wajah melibatkan pengecekan *one to one* sebagai konfirmasi yang tersimpan di sistem. Bila terkonfirmasi maka sistem akan menerima sedangkan bila tidak maka akan ditolak. Contoh dari sistem ini adalah *log-in* pada smarphone atau komputer, kontrol pagar otomatis dan e-passport. Sedangkan sistem identifikasi wajah melibatkan pengecekan *one to many*. Gambar wajah yang diambil akan dicocokkan dengan banyak gambar lainnya untuk menentukan identitas orang tersebut. Contoh dari sistem ini adalah penentuan informasi seseorang pada

database kepolisian, interaksi manusia dengan komputer pada video game, sistem presensi di sekolah dan video pengamatan terhadap seseorang (Parmar and Mehta, 2014) . Penelitian ini bisa dikategorikan sebagai identifikasi wajah, namun menggunakan data sederhana dari seseorang yang direkam wajahnya selama 30 detik dengan berbagai pose dengan bahasa python.

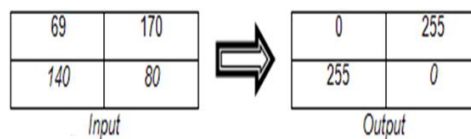
LBP secara umum dapat didefinisikan sebagai perbandingan nilai biner pixel pada pusat citra dengan 8 buah pixel di sekelilingnya (Rangkuti, 2020). Metodologi ini berkaitan erat dengan salah satu cabang keilmuan teknologi informasi, yaitu pengolahan citra/gambar. Berikut adalah hal-hal yang terkait sesuai dengan cabang keilmuan tersebut dalam penelitian ini (Kosasih and C. Daomara, 2021).

2.1 Thresholding

Merupakan pemetaan citra warna menjadi citra biner atau citra 2 warna (Willow, 2017). Pada thresholding dikenal dengan adanya batas ambang. Misalkan terdapat suatu fungsi dibawah:

$$f_o(x,y) = \begin{cases} 0, & f_o(x,y) < 128 \\ 255, & f_o(x,y) \geq 128 \end{cases} \tag{1}$$

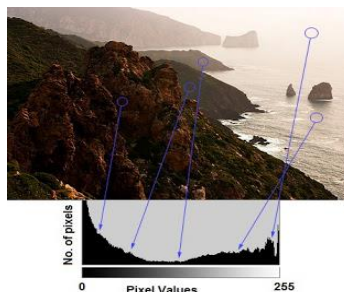
maka, bila terdapat citra input dengan digit tertentu seperti di bawah, citra outputnya adalah:



Gambar 1. Contoh perubahan digit citra *input* menjadi citra *output* pada *thresholding*

2.2 Histogram

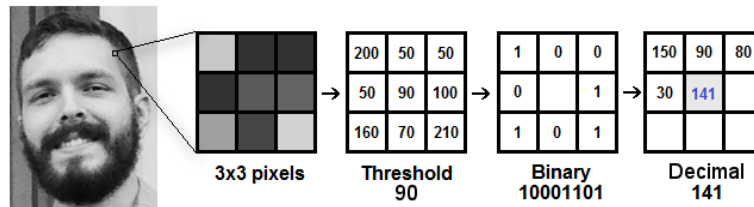
Merupakan gambaran atau plot dari keseluruhan ide dari distribusi intensitas pada suatu citra (Willow, 2017). Histogram memiliki rentang nilai dari 0 hingga 255, yaitu nilai maksimum dari suatu citra yang memiliki kedalaman 8 pixel (2⁸). Nilai 0 atau mendekati 0 akan memberikan warna yang gelap atau cenderung gelap, sedangkan nilai 255 atau mendekati 255 akan memberikan warna putih atau cenderung terang. Tampilannya dapat dilihat pada contoh berikut:



Gambar 2. Contoh histogram dari suatu citra pantai berbukit terjal dengan variasi warna terang gelap

2.3 LBP (Local Binary Pattern)

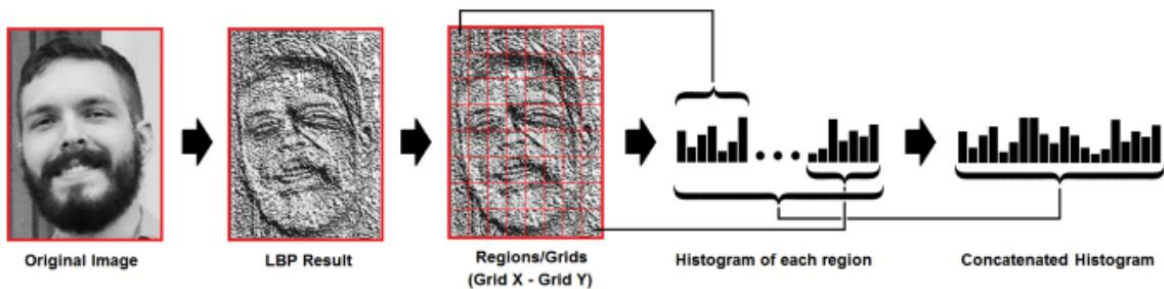
Merupakan metodologi operator tekstur yang simple dan sangat efisien dengan *thresholding* pixel-pixel yang bertetangga menjadi bilangan biner (Salton, 2017). Bilangan biner tersebut kemudian digabungkan berdasarkan baris atau putaran searah jarum jam untuk menjadi bilangan biner baru, lalu didesimalkan sehingga menjadi nilai pixel baru yang lebih baik. Urut-uratanya dapat kita lihat pada ilustrasi di bawah sebagai berikut:



Gambar 3. Ilustrasi LBP

2.4 LBPH (Local Binary Pattern Histogram)

Merupakan kelanjutan dari metodologi LBP, dimana pada gambar baru yang terbentuk dibagi menjadi multiple grid yang terdiri dari grid X dan Y, menjadi histogramnya. Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 4. Ilustrasi LBPH

Kemudian dilakukan komparasi histogram data *train* dan data *test* dengan mencari jarak (perbedaan) yang terdekat antara keduanya. Rumus umum yang digunakan adalah *euclidean distance*, dengan rumus:

$$D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2} \tag{2}$$

dimana D merupakan jarak antara histogram 1 dengan 2 yang diambil jarak terdekatnya.

Bahasa pemrograman yang digunakan pada penelitian ini adalah Python, yaitu bahasa pemrograman umum yang diciptakan oleh Guido Van Rossum, seorang berkebangsaan Belanda yang mendapat gelar master di jurusan matematika dan ilmu komputer dari Universitas Amsterdam. Python dapat begitu populer dalam waktu singkat dikarenakan *code* programnya yang simple dan mudah dimengerti (Willow, 2017). Bahasa tersebut memudahkan programmer dalam mengekspresikan idenya dalam baris *code* yang lebih sedikit namun tetap mudah dibaca. Sedangkan kamera yang digunakan dalam penelitian adalah kamera laptop VGA Web Camera dengan resolusi array 640 x 480 pixel. Meskipun saat ini

telah muncul jenis kamera laptop yang baru, namun kamera ini masih sangat berguna untuk beberapa *tools* tertentu.

IV. HASIL DAN PEMBAHASAN

Penelitian dilakukan dengan mengambil hanya beberapa data *sample*, untuk kemudian digali dengan lebih detil. Uraian akan menampilkan beberapa ekspresi berbeda hingga 30 jenis ekspresi dari suatu *sample* yang diambil dalam kurun waktu 30 detik melalui perekaman kamera laptop. Otomatisasi tersebut merupakan penjabaran dari Marcelo Rovai (Rovai, 2018) dalam artikelnya mengenai *Face Recognition* menggunakan bahasa python. Tahapan dimulai dari *input* data, *training* data dan *testing* data.

Tahap awal merupakan *input* data yang diambil menggunakan kamera laptop dengan durasi 30 detik. Selama durasi tersebut objek data akan mengambil beberapa ekspresi wajah untuk perekaman, yang contohnya dapat dilihat pada gambar berikut.



Gambar 5. Beberapa contoh ekspresi wajah dari data *input* dalam masing-masing detik

Import library yang digunakan pada program perekaman ini adalah cv2. Kemudian perekaman selama 30 detik dapat dilihat pada *sintax* berikut.

```
-----
count = 0
while(True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1
        cv2.imwrite("dataset/User." + str(face_id) + '!' + str(count) + ".jpg", gray[y:y+h,x:x+w])
        cv2.imshow('image', img)
    k = cv2.waitKey(100) & 0xff
    if k == 27:
        break
    elif count >= 30:
        break
-----
```

Data perekaman yang diambil akan disimpan ke dalam suatu *folder*, dalam hal ini *folder* tersebut diberi nama sebagai *dataset*.

Tahapan selanjutnya adalah *men-training* data. Program training data menggunakan beberapa library, yaitu cv2, numpy, PIL dan os untuk fungsi interaksi program dengan sistem operasi. Pada tahapan ini dilakukan pengambilan gambar dari *folder dataset* (Parkhi, A. Vedaldi, and A. Zisserman, 2015) dan proses *labeling* dengan menggunakan syntax sebagai berikut.

```
-----  
def getImagesAndLabels(path):  
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]  
    faceSamples=[]  
    ids = []  
    for imagePath in imagePaths:  
        PIL_img = Image.open(imagePath).convert('L')  
        img_numpy = np.array(PIL_img,'uint8')  
        id = int(os.path.split(imagePath)[-1].split(".")[1])  
        faces = detector.detectMultiScale(img_numpy)  
        for (x,y,w,h) in faces:  
            faceSamples.append(img_numpy[y:y+h,x:x+w])  
            ids.append(id)  
    return faceSamples,ids  
-----
```

Kemudian dilakukan training data dengan menggunakan syntax sebagai berikut.

```
-----  
faces,ids = getImagesAndLabels(path)  
recognizer.train(faces, np.array(ids))  
-----
```

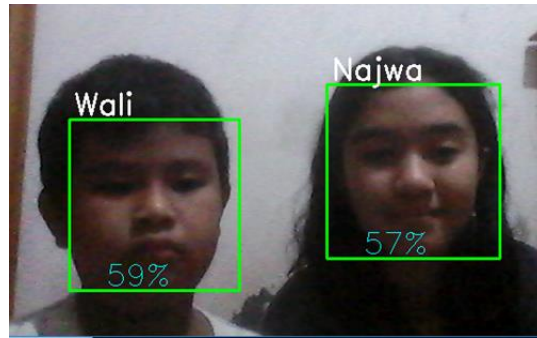
Selanjutnya adalah melakukan penyimpanan ke dalam *folder* dan file yang berekstensi *.yml* dan diberi nama *trainer*.

```
-----  
recognizer.write('trainer/trainer.yml')  
-----
```

Namun sebelum semua *syntax* tersebut digunakan, program kita arahkan untuk menggunakan konsep LBPH dengan syntax sebagai berikut.

```
-----  
recognizer = cv2.face.LBPHFaceRecognizer_create()  
-----
```

Setelah proses *training* selesai dilakukan, maka akan masuk ke tahap berikutnya yaitu pengujian. Pada tahapan ini, akan dideteksi siapa sebenarnya objek yang ada di depan kamera, sekaligus memunculkan namanya serta berapa prosentase kemiripannya dengan data yang ada di *dataset*. Contoh hasil analisisnya dapat dilihat pada gambar berikut.



Gambar 6. Contoh hasil *face recognition* dengan nama dan prosentase kemiripan

Pergerakan dari objek akan mengakibatkan perubahan prosentase kemiripan dari objek baik meningkat *maupun* menurun. Pada program pengujian ini dilakukan penginputan nama data yang telah direkam pada *dataset* sesuai dengan nama masing-masing objek yang telah direkam secara array. Library yang digunakan pada program pengujian ini adalah cv2. Diawali dengan penggunaan *sintax* penggunaan dari konsep LBPH dengan penulisan *sintax* seperti sebelumnya, lalu mulai dilakukan inisiasi pen-*settingan* pengambilan gambar melalui kamera dengan men-*setting* lebar dan tinggi *capture* menggunakan *sintax* sebagai berikut.

```
-----
cam = cv2.VideoCapture(0)
cam.set(3, 640)
cam.set(4, 480)
-----
```

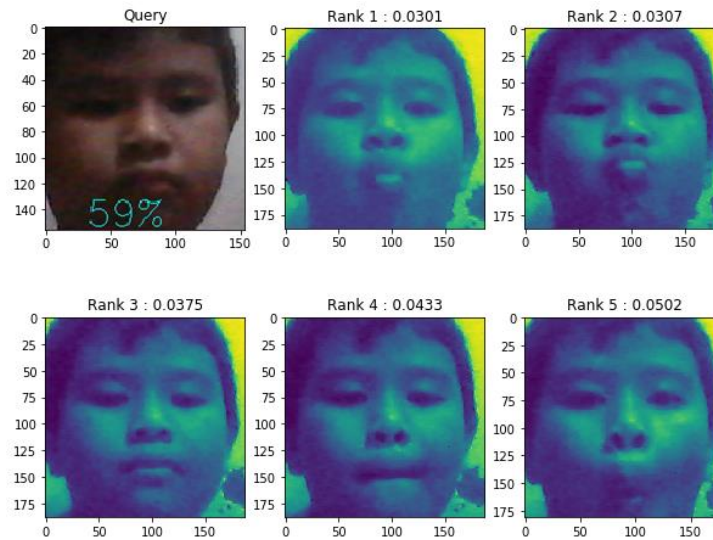
Pengambilan mulai dilakukan yang disertai dengan pengubahan gambar ke bentuk grey, dilanjutkan dengan pengubahan ke dalam bentuk LBPHnya dan pengecekan kecocokan dalam prosentase dengan menggunakan *sintax* berikut.

```
-----
while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
    )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = " {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = " {0}%".format(round(100 - confidence))
-----
```

```

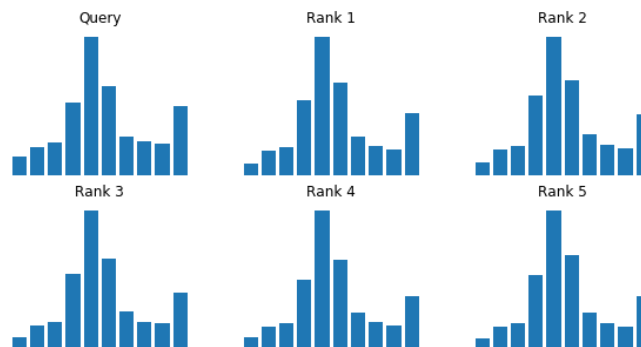
cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)
cv2.imshow('camera',img)
k = cv2.waitKey(10) & 0xff
if k == 27:
    break
    
```

Bila dilakukan penguraian dari *step* di atas dengan mengambil beberapa *dataset* secara acak dengan perbedaan yang cukup ekstrem untuk melihat jarak terdekat dari perhitungan euclidean-nya, dengan menggunakan *syntax* terpisah untuk object data Wali maka kita akan memperoleh gambaran dengan pola tertentu dan jelas. Tampilan dibuat berwarna namun dengan tetap menggunakan konsep perhitungan LBPH pada programnya, sehingga diperoleh contoh perbandingan sebagai berikut.



Gambar 7. Perbandingan acak beberapa *dataset* dari objek Wali dengan data input pengujian yang disertai dengan *ranking* dan jarak euclidean-nya

Dengan mengambil data input yang diberi nama “Query”, maka dapat terlihat bahwa gambar kedualah (sampingnya) yang memiliki jarak euclidean terkecil. Dengan demikian dapat kita lihat pola histogram dari gambar-gambar yang dinilai tersebut sebagai berikut.



Gambar 8. Perbandingan histogram dari *dataset* Wali dengan data input

V. KESIMPULAN

Python sebagai bahasa pemrograman berbasis objek sangat handal dan *powerfull* dalam melakukan modifikasi dan penilaian dengan berbagai peruntukkan terhadap suatu gambar. Ditambah belakangan ini keberadaan *library-library* pada python sudah semakin lengkap. Pengkodean dalam bahasa ini relatif sederhana dan mudah dipahami oleh para programmer. Dengan python penelitian dapat dilakukan dengan baik. Beberapa yang dapat disimpulkan pada penelitian ini adalah sebagai berikut Akurasi pengenalan wajah menggunakan metodologi LBPH memberikan perolehan hasil yang baik. Dalam penelitian ini nilai kecocokan yang muncul adalah 59%, namun di kesempatan yang lain nilai yang muncul dapat mencapai hingga sekitar 80%. Perubahan akurasi dapat terjadi ketika objek bergerak dalam pengukuran pengenalan wajah. Kualitas kamera memiliki peran yang sangat penting yang menjadi salah satu faktor bagi keakuratan data. Keterbatasan data *training* yang gelap seperti misalnya di awal perekaman data, dapat menjadi faktor ketidak akuratan dalam pengujian dari objek input yang misalnya diambil dalam kondisi yang juga cenderung gelap. Karena dalam bentuk histogramnya, gambar gelap akan membuat grafik mengumpul ke sebelah kiri, sehingga bila dipertemukan dengan data pengujian yang memberikan gambar histogram yang cenderung sama, maka objek akan dikatakan mirip dengan data yang ada di *dataset*. Tingkat kecerahan atau terang gelapnya gambar secara otomatis juga akan mempengaruhi akurasi. Hal ini agak bertentangan dengan beberapa referensi yang mengatakan bahwa tingkat kecerahan tidak memberikan pengaruh kepada akurasi penilaian. Tapi kenyataannya sangat berkaitan erat selama penilaian dilakukan berdasarkan bentuk dari histogramnya. Karena pada prinsipnya histogram sendiri merupakan representasi dari kecerahan gambar serta distribusi dari warna putih hitam dari gambar. Banyak hal yang dapat dikembangkan dari metodologi LBPH ini, juga keterkaitan *face recognition* dengan metodologi-metodologi lainnya seperti *machine learning*.

VI. ACKNOWLEDGEMENT

Penulis ucapkan terima kasih yang sedalam-dalamnya kepada Allah swt dan rosulNya, yang telah memberikan banyak karunia, seperti akal, inspirasi, ide, semangat dan kemudahan-kemudahan serta rezeki dalam penyelesaian penelitian ini.

VII. REFERENSI

- Kothari, (2014). *Research Methodology, Methods & Techniques*, Second Rev. New Age International (P) Ltd., 2004.
- Anantharam, (2016). "Local Binary Patterns". [Online]. Available: <https://medium.com/@rajatanantharam/local-binary-patterns-8807ecf7f87c>. [Accessed: 11-Jan-2022].
- Mohamed, (2013) "Face Recognition Using Statistical Adapted Local Binary Patterns," University of Louisville.
- Parmar and Mehta, (2013). "Face Recognition Methods & Application." *International Journal Computer Technology & Applications*, pp. 84–86.
- Rangkuti, (2020). "Proses Ekstraksi Ciri Citra Dengan Algoritma Local Binary Pattern Dalam Mendukung Klasifikasi Citra Bertekstur". [Online]. Available: <https://binus.ac.id/bandung/2020/12/proses-ekstraksi-ciri-citra-dengan-algoritma-local-binary-pattern-dalam-mendukung-klasifikasi-citra-bertekstur/>. [Accessed: 12-Jan-2022].
- Kosasih and C. Daomara, (2021). "Pengenalan Wajah Dengan Menggunakan Metode Local Binary Pattern Histograms (LBPH)." *Jurnal Media Informatika Budidarma*, pp. 1258–1264.
- Willow, (2017). "OpenCV-Python Tutorials." Release Beta.
- Salton, (2017). "Face Recognition: An End-To-End Project". [Online]. Available: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>. [Accessed: 12-Jan-2022].

- Rovai, (2018). "Real-Time Face Recognition," 2018. [Online]. Available: <https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826>. [Accessed: 10-Jan-2022].
- Parkhi, A. Vedaldi, and A. Zisserman, (2015). "Deep Face Recognition." Visual Geometry Group, Department of Engineering Science, University of Oxford.