# IF2123 ALJABAR LINIER DAN GEOMETRI

# LAPORAN TUGAS BESAR 2

**Penerapan Metrik Berbasiskan Vektor di dalam Sistem Pengenalan Wajah**

**(Face Recognition)**

**Diajukan untuk memenuhi tugas mata kuliah IF2123 Aljabar Linier dan Geometri**

**Semester I Tahun Akademik 2019-2020**

**Dosen : Nugraha Priya Utama ST, MA. Ph.D.**

Oleh


| | |
|---|---|
| Jovan Karuna Cahyadi | 13518024 |
| Jonathan Yudi Gunawan | 13518084 |
| William | 13518138 |




**PROGRAM STUDI TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**


**2019/2020**

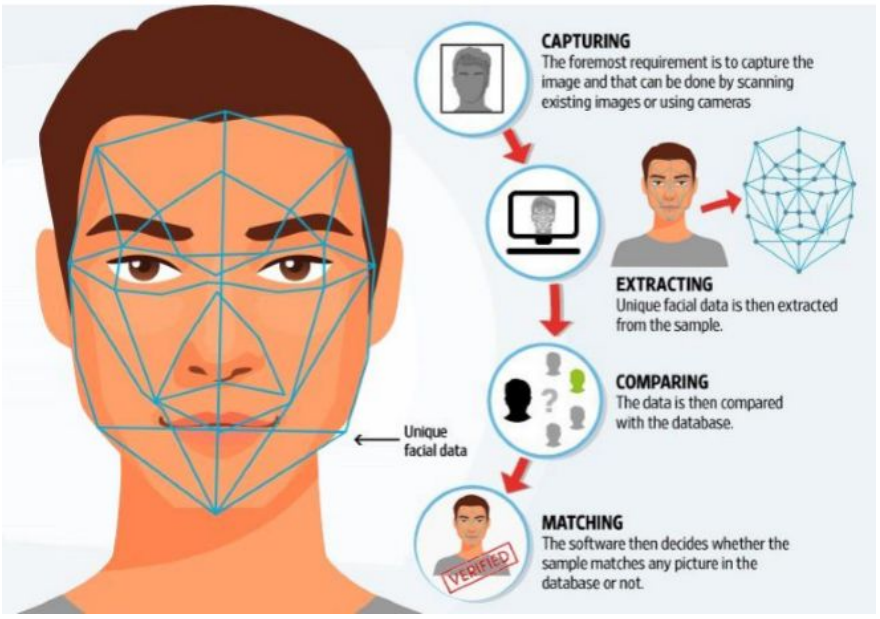
# DAFTAR ISI

# DAFTAR TABEL DAN GAMBAR

# BAB 1
# DESKRIPSI MASALAH

Kami ditugaskan untuk membuat sebuah program untuk pengenalan wajah yang ditulis dengan menggunakan bahasa Python, dengan ketentuan sebagai berikut:

Pengenalan wajah (*face recognition*) adalah teknologi untuk mengidentifikasi atau memverifikasi wajah seseorang melalui gambar digital. Caranya adalah dengan mencocokkan fitur-fitur yang diekstraksi dari wajah yang diidentifikasi dengan data wajah yang tersimpan di dalam basis data. Pengenalan wajah telah digunakan sebagai sebuah sistem biometrik. Alur proses sebuah sistem pengenalan wajah diperlihatkan pada Gambar 1.1.




**Gambar 1.1** Alur proses sistem pengenalan wajah
(Sumber: https://www.shadowsystem.com/page/20)

Fitur-fitur dari gambar wajah diekstraksi dengan sebuah prosedur komputasi menggunakan teknik-teknik di dalam pengolahan citra (*image processing*), namun di dalam tugas besar ini fungsi untuk ekstraksi fitur diasumsikan sudah tersedia. Sekumpulan fitur tersebut direpresentasikan sebagai vektor. Proses pencocokan antar vektor wajah yang ditanya dengan vektor-vektor wajah di dalam basis data menggunakan metriks similarity. Metrik similarity itu mengukur seberapa dekat atau mirip antara dua buah vektor.

Diberikan dua buah vektor, v = ($v_1$, $v_2$, …, $v_n$) dan w = ($w_1$, $w_2$, …, $w_n$). Ada dua metriks similarity yang umum digunakan di dalam pencocokan data, yaitu jarak Euclidean dan *cosine similarity*.

1. Jarak Euclidean (*Euclidean distance*). Jarak antara vektor v dan w diukur dengan

$$d = \sqrt{(v_1 - w_1)^2 + (v_2 - w_2)^2 + ... + (v_n - w_n)^2} \quad (1)$$

rumus

Nilai d yang kecil menunjukkan kedekatan. Jika v dihitung jaraknya masing-masing dengan vektor $\mathbf{w_1}$, $\mathbf{w_2}$, …, $\mathbf{w_m}$, yaitu $d(\mathbf{v}, \mathbf{w_i})$ untuk $i$ = 1, 2, …, $m$, maka nilai $d$ yang paling minimum menunjukkan jarak dua vektor yang paling mirip.

2. *Cosine similarity*. Metriks *cosine similarity* dihitung dari perkalian titik (*dot product*) antara dua buah vektor. Perkalian titik antara v dan w dihitung dengan rumus

$$\cos\theta = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|} \quad (3)$$

Persamaan (3) digunakan untuk mengukur similarity antara dua buah vektor. Dua buah vektor v dan w dikatakan sama atau berimpit jika sudut antara keduanya nol ( $\theta = 0$). Cosinus 0 adalah 1. Sifat ini dipakai di dalam proses pencocokan antara dua buah vektor. Nilai cosinus yang besar (maksimum 1) menunjukkan kemiripan. Jika nilai cosinus mendekati satu, maka dua vektor dikatakan hampir sama atau hampir mirip. Oleh karena itu , persamaan (3) dinamakan juga *cosine similarity*.

Di dalam tugas besar ini, kita membuat sistem pengenalan wajah dengan menggunakan metrik kemiripan berbasis vektor. Input untuk sistem adalah sebuah citra wajah yang ditanyakan, sedangkan koleksi citra wajah di dalam basis data diasumsikan sudah tersedia. Fitur-fitur dari wajah diekstraksi, begitu pula fitur-fitur wajah di dalam basis data. Pencocokan kemiripan dilakukan antara vektor fitur citra wajah input dengan vektor-vektor fitur wajah di dalam basis data. Hasil pencocokan di-rangking dari yang paling mirip hingga yang kurang mirip (gunakan nilai ambang T untuk menampilkan hasil pencocokan, misal T = 10, maka akan ditampilkan 10 wajah yang paling mirip hingga yang tidak terlalu mirip).

Program dibuat dengan Bahasa Python dengan memanfaatkan sejumlah library di OpenCV (*Computer Vision*). Fungsi untuk mengekstraksi fitur dari sebuah citra wajah tidak perlu anda buat lagi, tetapi menggunakan fungsi ekstraksi yang sudah tersedia di dalam library.

Kami membuat program pengenalan wajah dalam Bahasa Python dengan spesifikasi yang diberikan yaitu:

1. Program menerima input sebuah citra wajah
2. Basisdata wajah dapat diunduh secara mandiri melalui:
   https://www.kaggle.com/frules11/pins-face-recognition/version/1#
3. Dalam *basisdata* wajah tersebut, satu identitas memiliki banyak foto. Kalian wajib memisahkan dari *basisdata* tersebut menjadi: 80% foto dijadikan data referensi, dan 20% dijadikan data uji (untuk setiap identitas dalam *basisdata*).
4. Program melakukan pencocokan wajah dengan koleksi wajah di dalam *basisdata*. Metrik untuk mengukur kemiripan menggunakan jarak euclidean dan cosine similarity. Pengguna dapat memilih salah satu dari dua *metrik* tersebut.
5. Program me-rangking hasil pencocokan dan menampilkan T buah hasil pencocokan (T di-setting di dalam program). Luaran dari program adalah gambar-gambar citra wajah hasil pencocokan sebanyak T buah.
6. Program menghitung jarak euclidean dan cosine similarity yang ditulis sendiri kode programnya, tidak boleh menggunakan fungsi yang sudah tersedia di dalam library atau Bahasa Python.
7. Kode program di dalam tautan di atas boleh dijadikan panduan, namun tidak boleh di-copy paste atau sama.
8. Buatlah GUI yang menarik sebagai  antarmuka dengan pengguna program.

# BAB 2
# TEORI SINGKAT

Teori singkat mengenai vektor, norma Euclidean, perkalian titik, pengenalan wajah, dan materi pendukung seperti OpenCV dan pustaka yang relevan.

## 2.1 Vektor

Vektor adalah besaran yang memiliki besar dan arah. Besaran-besaran pada fisika banyak yang termasuk besaran vektor. Contohnya gaya, kecepatan, percepatan, perpindahan, momen gaya dan momentum. Pada besaran vektor memiliki penjumlahan yang berbeda dengan besaran skalar. Penjumlahan vektor disebut juga dengan resultan vektor. Resultan vektor sangat dipengaruhi oleh besar dan arah, sehingga perlu metode tertentu.

## 2.2 Norma Euklidian

Pada n- dimensi Euclidean ruang $R^n$ , gagasan intuitif panjang vektor $x = ( x_1 , x_2 , ..., x_n )$ ditangkap oleh rumus

$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_n^2}.$$

Ini adalah norma Euclidean, yang memberikan jarak biasa dari titik asal ke titik X , konsekuensi dari teorema Pythagoras. Norma Euclidean sejauh ini merupakan norma yang paling umum digunakan pada $R^n$ , tetapi ada norma lain pada ruang vektor ini seperti yang akan ditunjukkan di bawah ini. Namun, semua norma ini setara dalam arti bahwa mereka semua mendefinisikan topologi yang sama.

Pada ruang kompleks n- dimensi $C^n$ norma yang paling umum adalah

$$\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^* \, \boldsymbol{x}},$$

Dalam kedua kasus, norma dapat dinyatakan sebagai akar kuadrat dari produk dalam vektor dan itu sendiri:

$$\|\boldsymbol{z}\| := \sqrt{|z_1|^2 + \cdots + |z_n|^2} = \sqrt{z_1 \bar{z}_1 + \cdots + z_n \bar{z}_n}.$$

di mana x direpresentasikan sebagai vektor kolom ([ $x_1$ ; $x_2$ ; ...; $x_n$ ]), dan $x^*$ menunjukkan transpose konjugatnya .

Formula ini berlaku untuk ruang produk dalam apa pun, termasuk ruang Euclidean dan kompleks. Untuk ruang Euclidean, produk dalam setara dengan produk titik . Oleh karena itu, dalam kasus khusus ini formula dapat juga ditulis dengan notasi berikut:

## 2.3 Perkalian titik

Perkalian titik atau *dot product* dua buah vektor didefinisikan sebagai perkalian antara besar salah satu vektor (misalnya **A**) dengan komponen vektor kedua (**B**) pada arah vektor pertama (**A**).

$$\vec{A} \cdot \vec{B} = (a_x \hat{i} + a_y \hat{j} + a_z \hat{k}) \cdot (b_x \hat{i} + b_y \hat{j} + b_z \hat{k}) \quad = a_x b_x + a_y b_y + a_z b_z$$

Dari persamaan perkalian titik di atas maka dapat disimpulkan bahwa hasil perkalian titik dua buah vektor adalah skalar. Simbol dari perkalian titik adalah "." (baca: dot). Karena hasil perkalian titik adalah skalar maka perkalian titik atau *dot product* disebut juga dengan perkalian skalar atau *skalar product*.

## 2.4 Pengenalan Wajah

Pengenalan wajah (*face recognition*) adalah teknologi untuk mengidentifikasi atau memverifikasi wajah seseorang melalui gambar digital atau dari video. Ada berbagai cara untuk agar *face recognition* bekerja, tapi secara umum adalah dengan mencocokkan fitur-fitur yang diekstraksi dari wajah yang diidentifikasi dengan data wajah yang tersimpan di dalam basis data. Pengenalan wajah telah digunakan sebagai sebuah sistem biometrik. Fitur-fitur dari gambar wajah diekstraksi dengan sebuah prosedur komputasi menggunakan teknik-teknik di dalam pengolahan citra (*image processing*). Sekumpulan fitur tersebut direpresentasikan sebagai vektor. Proses pencocokan antar vektor wajah yang ditanya dengan vektor-vektor wajah di dalam basis data menggunakan metrik similarity. Metrik similarity itu mengukur seberapa dekat atau mirip antara dua buah vektor.

## 2.5 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini merupakan pustaka lintas platform. Program ini didedikasikan sebagian besar untuk pengolahan citra secara real-time. Jika pustaka ini menemukan pustaka Integrated Performance Primitives dari intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis.

## 2.6 *Graphical User Interface (GUI)*

*Graphical User Interface* adalah suatu sistem interaktif yang berisikan ikon serta objek objek grafis lainnya yang dapat membantu pengguna dalam menggunakan software - software pada komputer. *GUI* sendiri lebih sering digunakan dibanding *text-based* program yang dijalankan di *command line*. *GUI* memberikan ikon - ikon yang memudahkan pengguna dalam melakukan operasi - operasi dalam sistem komputer, seperti membuka, menghapus ataupun memindahkan file. Meskipun fokus utama *GUI* lebih kepada penggunaan *mouse*, sistem juga dapat dioperasikan dengan bantuan *keyboard*. Keunggulan dalam menggunakan *GUI* dibandingkan dengan menggunakan sistem lain seperti menggunakan command line, adalah memudahkan pengguna dalam mengakses fitur-fitur dalam suatu sistem. Dalam program tugas besar kali ini, penulis menggunakan *library* yang telah disediakan dari Python, yakni *GUI* wx. *GUI* wx yang digunakan memiliki fitur *drag and drop* yang memudahkan pengguna dalam mencari kemiripan gambar dengan gambar yang di drop di *GUI* program.

Keunggulan dari library wx adalah library ini bersifat *cross-platform* yaitu dapat dimigrasikan ke platform lain dengan mudah seperti C++ (dalam C++ bernama `wxWidgets`). `wxPython` juga dapat dijalankan pada platform lain tanpa perlu modifikasi seperti *UNIX-based* linux, macOS, dan max OS X.

## 2.7 KAZE *Features*

KAZE adalah sebuah algoritma dalam *computer vision* untuk mendeteksi fitur - fitur 2D dan mendeskripsikan fitur lokal dalam gambar secara ruang yang tidak linier. Berbeda dengan algoritma pendeteksi fitur citra lainnya seperti SIFT dan SURF yang mencari fitur memakai skala ruang *Gaussian*. Dengan menggunakan KAZE, suatu citra akan diubah menjadi bentuk vektor fitur lokal yang kemudian digunakan sebagai pendekatan dalam mendeteksi maupun mengenali objek yang dimaksud melalui metriks jarak *euclidian* dan jarak *cosine*. Kedekatan ini kemudian digunakan untuk mengenali citra dengan citra lain, yang dalam hal ini tugas kali ini berfungsi untuk mendeteksi bentuk muka pada citra wajah.

# BAB 3
# IMPLEMENTASI PROGRAM

## 3.1 Memisahkan Data *Test* dari Data *Train*

Sebelum kami melakukan ekstraksi fitur - fitur, kami melakukan pembagian data menjadi 2 (dua) bagian yaitu *training* data dan *testing* data. Data total dibagi dengan bobot 80 (delapan puluh) persen *train* data dan 20 (dua puluh) persen *test* data. Pembagian data ini dilakukan dengan menggunakan *script* split.py. Data untuk *training* diletakkan di *folder* yang dinamakan *base* sedangkan data untuk *test* diletakkan di *folder* yang dinamakan *test.* Khusus untuk data *training,* foto digabung dan dijadikan dalam 1(dalam) *folder*, yaitu *folder base* guna mempermudah proses ekstraksi *feature*.

Pemisahan data - data gambar dilakukan melalui beberapa tahap. Pertama, nama *folder* setiap gambar data yang merepresentasikan *folder* gambar orang yang berkesesuaian atau yang dirujuk disimpan pada suatu list. Kemudian, setelah semua nama *folder* dalam training data disimpan, langkah berikutnya adalah membentuk kedua *folder* baru, yaitu *base* dan *test.* Di langkah selanjutnya, kami memisahkan jumlah file setiap *folder* menjadi 80% di *base* dan 20% di *test.* Pembagian ini disesuaikan dengan spesifikasi yang ada di panduan / pedoman spesifikasi tugas besar Algeo II. Langkah terakhir yang kami ambil dalam pemisahan data gambar adalah dengan menggabungkan seluruh gambar pada setiap *folder* pada *base* menjadi 1(satu). *Folder* pada *test* tidak kami ubah demi kemudahan saat melakukan pengetesan pada program.

## 3.2 Mengekstraksi Fitur-Fitur dari Foto Training

Kami menggunakan metode KAZE Features untuk mengekstrak fitur-fitur pada gambar. Metode ini sudah tersedia pada library *opencv2* sehingga kami dapat langsung menggunakannya tanpa perlu mengimplementasikannya kembali. Kami juga mengubah ukuran *array* agar seragam untuk semua gambar dan tidak memakan banyak memori. Ukuran array dipilih 32*64 = 2048 karena nilai ini cukup optimal, tidak terlalu kecil sehingga menyebabkan performa buruk, namun juga tidak terlalu besar sehingga menyebabkan pencarian terlalu lama.

Kami juga memanfaatkan library os untuk memudahkan dalam mengekstrak fitur secara massal. Hasil ini kemudian disimpan ke dalam file bernama "*features.pck*" yang

berisikan *dictionary* dengan *key*-nya adalah nama file gambar (tanpa *full path* nya), dan *value*-nya berupa *vector-vector* yang merupakan *keypoint* yang berhasil diekstrak dengan metode di atas. Untuk menyimpan file ini digunakan *library pickle*, yaitu metode `pickle.dump`.

### 3.3 Mencocokkan Fitur-Fitur Sebuah Gambar dengan Basis Data

Setelah melakukan proses ekstraksi seperti yang telah dijelaskan di atas, kami melakukan pencocokan fitur - fitur terhadap gambar yang dimasukkan ke dalam program melalui *GUI*. Pencocokan dilakukan dengan menggunakan fungsi-fungsi untuk mencocokkan fitur-fitur sebuah gambar dengan *database* yang keduanya telah disimpan dalam tipe "*pickle*". Dalam file `matcher.py` ini kami meng-*import* file python bernama `vectorutils.py` yang berisi algoritma fungsi *cosine similarity* dan *euclidean distance* yang telah kami buat sebelumnya. Fungsi-fungsi tersebut digunakan di file *matcher.py* dengan nama `cosine_distance` untuk memanggil fungsi *cosine similarity* dan `euclidean_distance` untuk memanggil fungsi *euclidean distance*. Di file *matcher.py* ini fungsi-fungsi tersebut telah ditambahkan algoritma untuk me-*reshape* bentuk inputan dari fitur - fitur menjadi bentuk vektor yang merupakan *array* 1-dimensi. Hasil dari *cosine similarity* antara 0 sampai 1, jika mendekati 1 maka vektor gambar dan *database* mirip sedangkan untuk *euclidean distance* menghasilkan nilai dari 0 dan terus membesar, jika mendekati 0 maka vektor gambar dan *database* mirip. Namun, di fungsi `cosine_distance` hasil *cosine similarity* dikurangi 1, karena setelah melakukan fungsi `cosine_distance` ataupun euclidean_distance maka hasil dari pencocokkan fitur akan diurutkan setiap gambarnya terurut membesar sesuai hasilnya. Gambar tersebut akan diurutkan membesar sesuai hasil dari `cosine_distance` ataupun `euclidean_distance` dan akan ditampilkan di *GUI* sebanyak 20 foto yang hasilnya mendekati 0.

## 3.4 *Graphical User Interface (GUI)*

Program kami memanfaatkan library wx sebagai *GUI*, dengan pertimbangan adanya fitur *drag and dro*p pada wx. *GUI* berisi 2 "*panel*" atau "*activity*" yang berbeda, satu digunakan untuk menerima gambar yang akan dicocokkan, sedangkan yang satunya untuk menampilkan hasil pencocokan gambar tersebut dengan *database* yang telah diperoleh sebelumnya.

### 3.4.1 *Drag and Drop* Panel

Panel ini berisi judul dan instruksi, 1 slot untuk menerima foto, serta dua tombol di bawahnya. Penerimaan foto dilakukan dengan metode *drag and drop*. Dua tombol di bawahnya bertuliskan "*with Cosine similarity*" dan "*with Euclidean distance*" yang berfungsi untuk memilih metode yang akan digunakan dalam pencocokan fitur pada gambar pada panel selanjutnya.

Dalam pembuatan fitur *drag and drop* digunakan *library* `pubsub` untuk memberitahukan pada class `wx.FileDropTarget` bahwa ada foto yang di-*drop* ke dalam slot tersebut. Slot foto ini juga menggunakan library `wx.Image` untuk memuat foto dan mengubah ukurannya agar sesuai dengan ukuran slot yang diinginkan. Ketika tombol ditekan, maka program akan mencari gambar yang paling mirip dengan *database*. *Path* menuju gambar-gambar tersebut akan disimpan dalam sebuah *array*. Kemudian panel pertama akan disembunyikan dan akan diganti dengan panel kedua, yaitu panel yang akan menampilkan gambar yang mirip.

### 3.4.2 *Show Image* Panel

Panel ini berisi judul dan instruksi, N slot untuk menampilkan foto, serta satu tombol di bawahnya. *Path* foto yang telah disimpan sebelumnya, akan ditampilkan pada slot-slot ini. Panel ini menggunakan class `wx.lib.scrolledpanel.ScrolledPanel` agar dapat memuat foto lebih banyak dan lebih nyaman digunakan. Di paling bawah, ada sebuah tombol untuk kembali ke panel pertama untuk memilih gambar lain untuk dicocokkan dengan *database*. Ketika tombol tersebut ditekan, maka panel ini akan disembunyikan, lalu panel pertama akan ditampilkan.

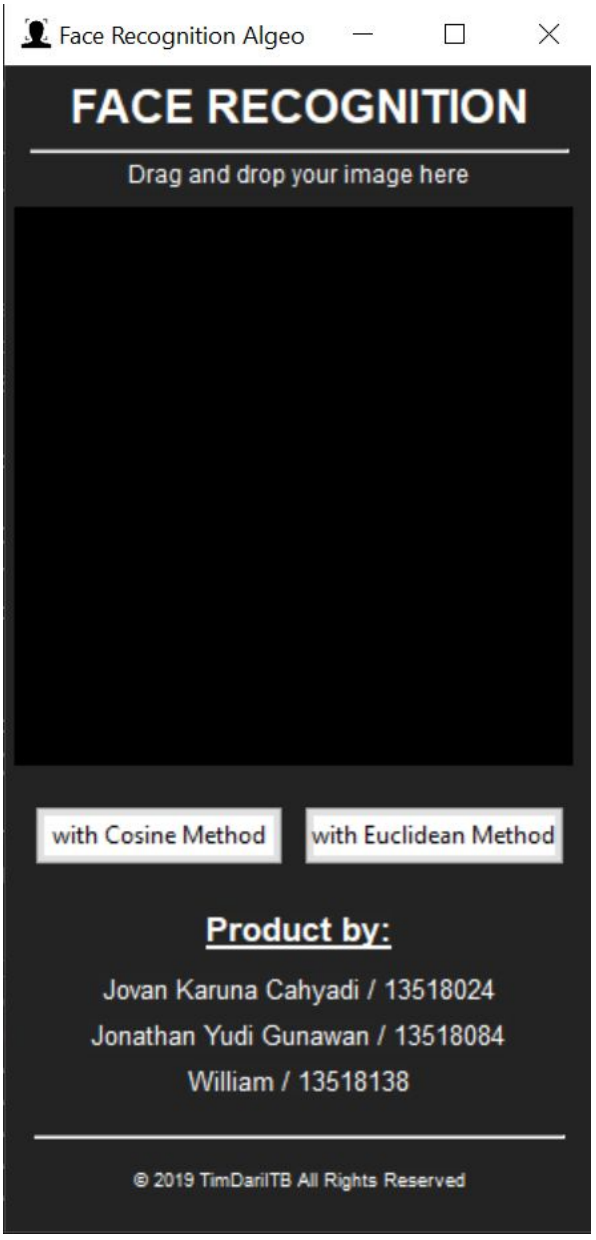
# BAB 4

# EKSPERIMEN

Untuk menguji kebenaran dari program kami, kami melakukan beberapa eksperimen. Eksperimen tersebut dapat dilihat sebagai berikut.

## 4.1 Tampilan Awal

### 4.1.1 Sebelum Menerima Foto



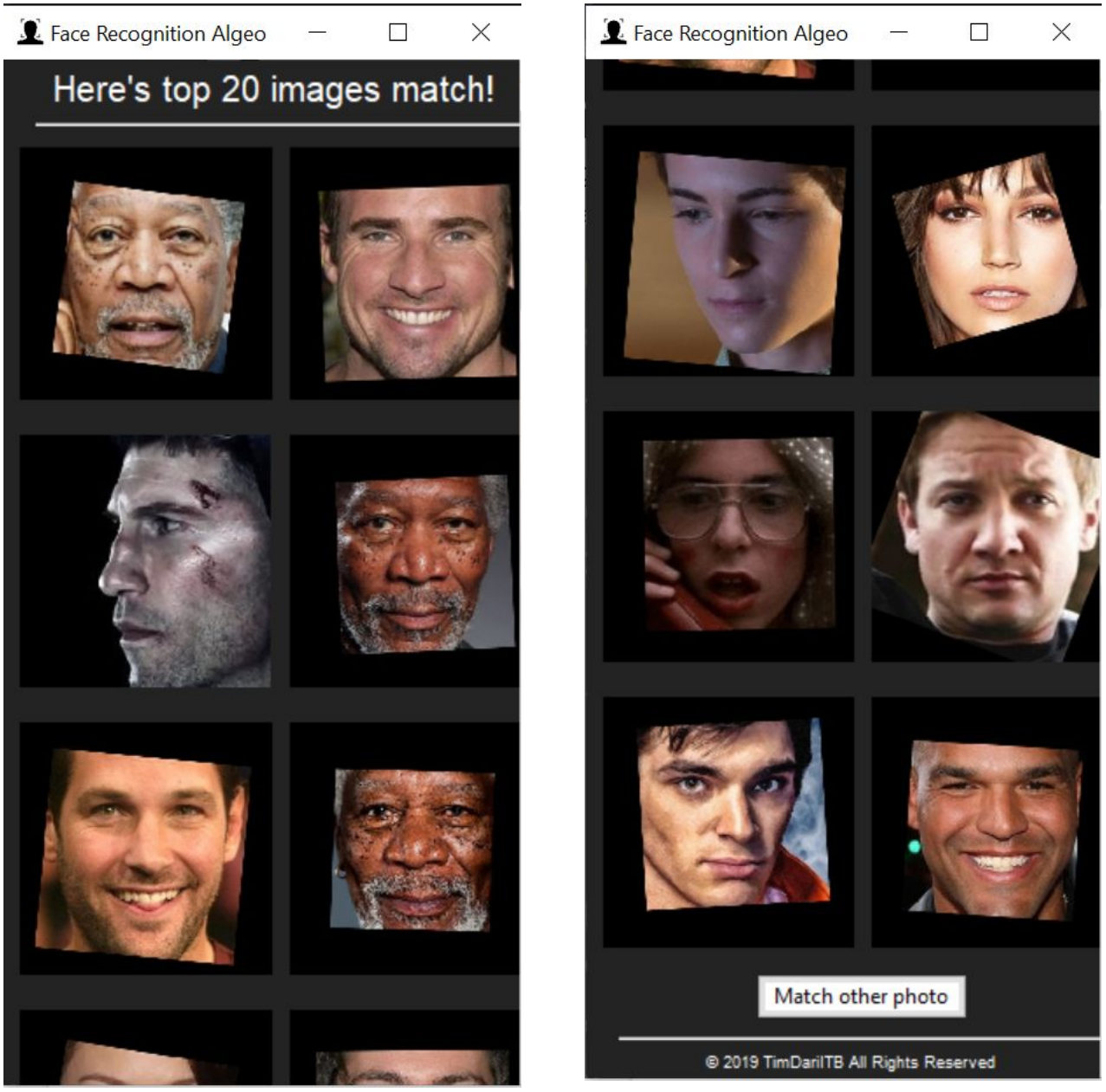

Gambar 4.1.1 Tangkapan Layar *GUI* sebelum Menerima Foto

### 4.1.2 Setelah Menerima Foto


Gambar 4.1.2 Tangkapan Layar *GUI* setelah Menerima Foto

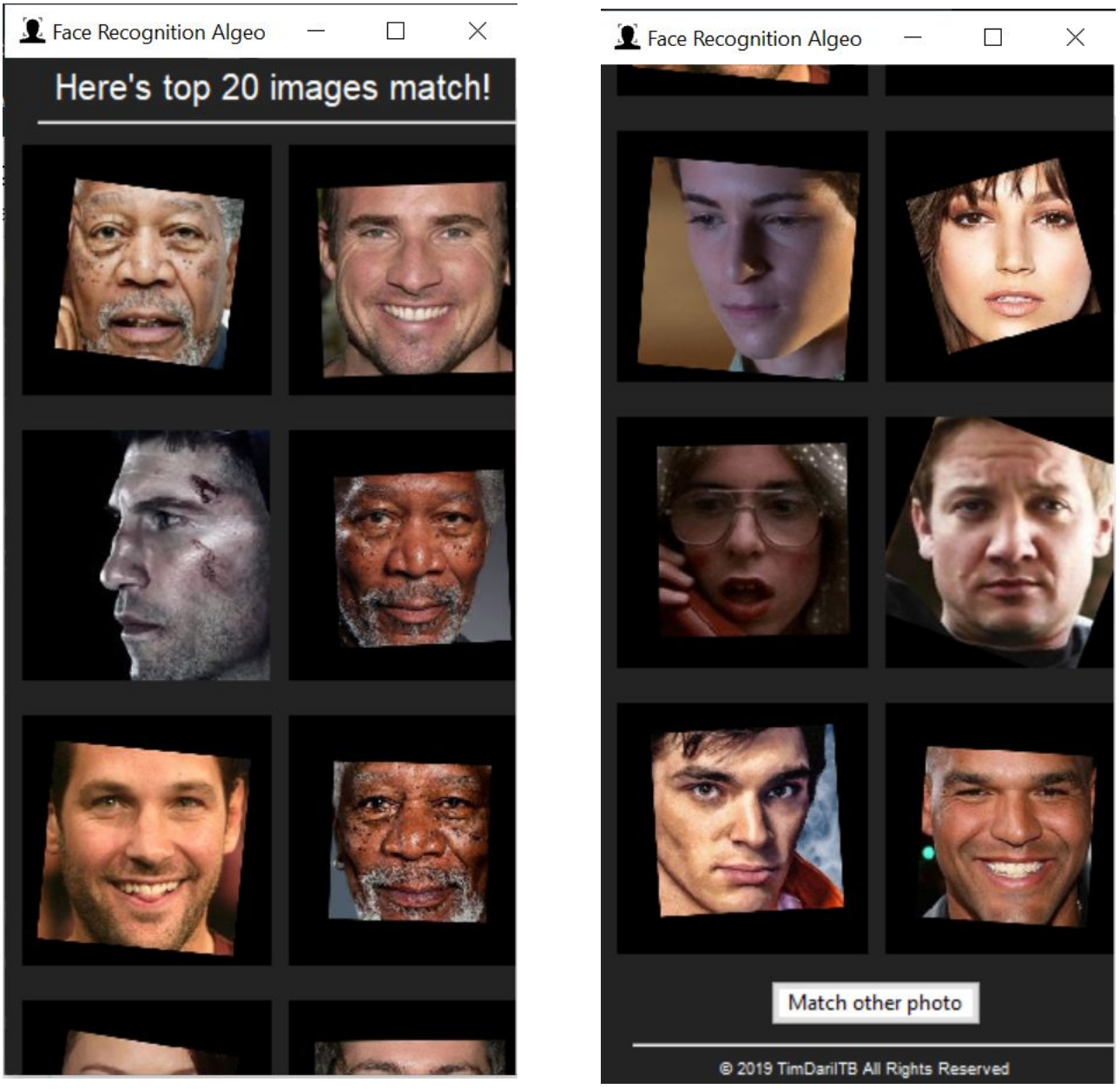
## 4.2 Tampilan Hasil Pencocokan Wajah

### 4.2.1 Menggunakan Metode *Cosine Similarity*



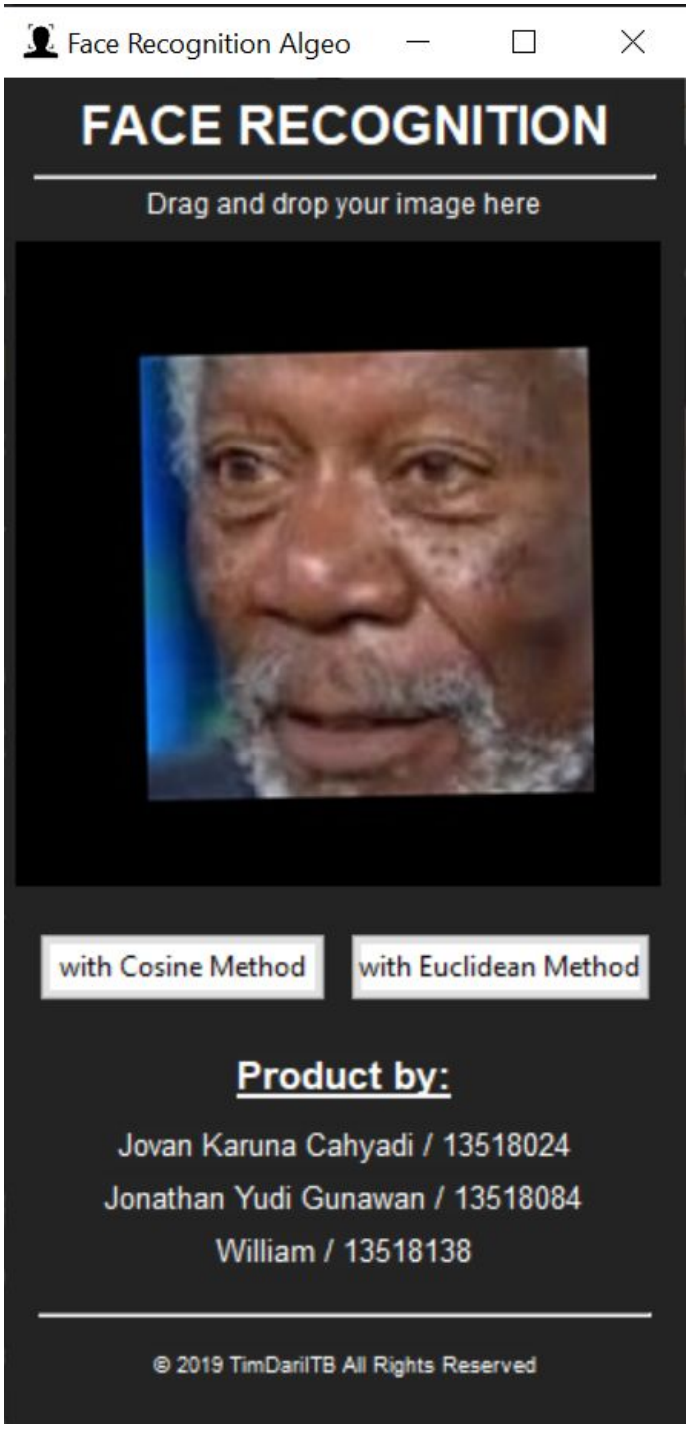

Gambar 4.2.1 Tangkapan Layar *GUI* Hasil Pencocokan dengan Metode *Cosine Similarity*

### 4.2.2 Menggunakan Metode *Euclidean Distance*




Gambar 4.2.2 Tangkapan Layar *GUI* Hasil Pencocokan dengan Metode *Eucidian Distance*

### 4.3 Tampilan Kembali ke Menu Utama




Gambar 4.3 Tangkapan Layar *GUI* Kembali ke Menu setelah Mencocokan Foto

# BAB 5
# KESIMPULAN, SARAN, DAN REFLEKSI

Dalam bab ini akan disajikan kesimpulan dari tugas besar yang telah dikerjakan oleh penulis. Setelah itu, penulis akan memberikan saran dan refleksi bagi para pembaca laporan.

## 5.1 Kesimpulan

Pada tugas besar kali ini, kami berhasil membuat sebuah "aplikasi" pencocokan gambar dengan *database* yang telah ditentukan sebelumnya. Fitur-fitur dalam aplikasi kami antara lain, *drag and drop* gambar, tampilan gambar hasil pencocokan yang rapih dan dapat di-*scroll* sehingga dapat memuat gambar cukup banyak. Tingkat akurasi program kami dapat dikatakan setara dengan *baseline* yang diberikan dalam spesifikasi karena kami menggunakan metode yang sama dengan spesifikasi yaitu, KAZE.

Hasil implementasi lengkap program dapat diakses melalui pranala https://github.com/JovanKaruna/Algeo2-Face.

## 5.2 Saran

Penulis menyarankan untuk ke depannya aplikasi dapat dikembangkan dengan cara, program dapat diimplementasikan dalam platform mobile seperti Android dan IOS. Setelah itu, program *face recognition* yang dibuat dapat dijalankan secara *real time* guna meningkatkan kegunaannya dalam dunia nyata.

Penulis juga menyarankan untuk kedepannya agar dapat meningkatkan tingkat akurasi program maka dapat menggunakan algoritma ekstraksi fitur dan pencocokkan fitur dengan metrik yang lebih akurat seperti menggunakan SIFT.

## 5.3 Refleksi

Setelah melakukan tugas besar ini, penulis merefleksikan bahwa tubes ini sangat penting dan bermanfaat untuk dipelajari, dikarenakan sekarang ini penggunaan *face recognition* sebagai fitur keamanan dan keperluan lainnya sudah mulai digunakan di Indonesia.

Kiranya tugas besar ini dapat membantu penulis berkembang menjadi lebih baik lagi. Dari eksperimen diatas juga dapat disimpulkan bahwa program yang penulis kerjakan dapat dijalankan dengan baik. Penulis juga dapat menyimpulkan bahwa dengan menggunakan bahasa Python dan OpenCV tugas besar ini lebih mudah dikerjakan.

# DAFTAR REFERENSI


Alcantaria, Pablo F. 2019. *KAZE Features*. http://www.robesafe.com/personal/pablo.alcantarilla/kaze.html diakses pada 5 November 2019.

Anonymous. 2019. *Facial recognition system*. https://en.wikipedia.org/wiki/Facial_recognition_ system di akses pada 28 Oktober 2019.

_____. 2019. *Norm (mathematics)*. https://en.wikipedia.org/wiki/Norm_(mathematics) di akses pada 28 Oktober 2019.

_____. 2013. *Pengertian Vektor*. https://www.temukanpengertian.com/2013/09/pengertian-vektor.html diakses pada 28 Oktober 2019.

_____. 2014. *OpenCV*. https://id.wikipedia.org/wiki/OpenCV diakses pada 28 Oktober 2019

Blog Mipa, Supervisor. 2017. *Rumus dan Sifat Perkalian Titik (Dot Product) 2 Vektor Beserta Contoh Soal dan Pembahasan.* https://www.fisikabc.com/2017/06/perkalian- titik-dua-vektor.html diakses pada 28 Oktober 2019.

Hope, Computer. 2019. *GUI*. https://www.computerhope.com/jargon/g/gui.htm diakses pada 31 Oktober 2019.