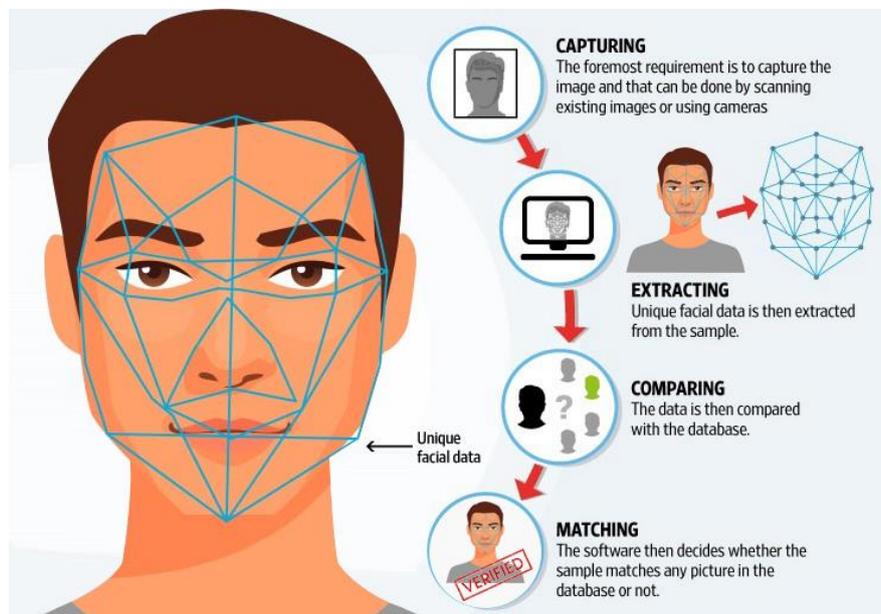


**Tugas Besar 2 IF2123 Aljabar Linier dan Geometri**  
**Penerapan Metrik Berbasis Vektor di dalam Sistem Pengenalan Wajah**  
**(Face Recognition)**  
**Semester 1 Tahun 2019/2020**

---

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 basisdata. Pengenalan wajah telah digunakan sebagai sebuah sistem biometrik. Alur proses sebuah sistem pengenalan wajah diperlihatkan pada Gambar 1.



**Gambar 1.** Alur proses di dalam 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 basisdata menggunakan metrik *similarity*. Metrik *similarity* itu mengukur seberapa dekat atau mirip antara dua buah vektor.



**Gambar 2.** Koleksi wajah di dalam basisdata

(Sumber: <https://www.dreamstime.com/stock-photo-people-faces-collection-set-smiling-face-group-image80688279>)

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

1. **Jarak Euclidean** (*Euclidean distance*). Jarak antara vektor  $\mathbf{v}$  dan  $\mathbf{w}$  diukur dengan rumus

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

Nilai  $d$  yang kecil menunjukkan kedekatan. Jika  $\mathbf{v}$  dihitung jaraknya masing-masing dengan vektor  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m$ , yaitu  $d(\mathbf{v}, \mathbf{w}_i)$  untuk  $i = 1, 2, \dots, m$ , maka nilai  $d$  yang paling minimum menunjukkan jarak dua vektor yang paling mirip.

2. *Cosine similarity*. Metrik *cosine similarity* dihitung dari perkalian titik (*dot product*) antara dua buah vektor. Perkalian titik antara  $\mathbf{v}$  dan  $\mathbf{w}$  dihitung dengan rumus

$$\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\| \|\mathbf{w}\| \cos \theta \quad (2)$$

Sudut antara  $\mathbf{v}$  dan  $\mathbf{w}$  adalah

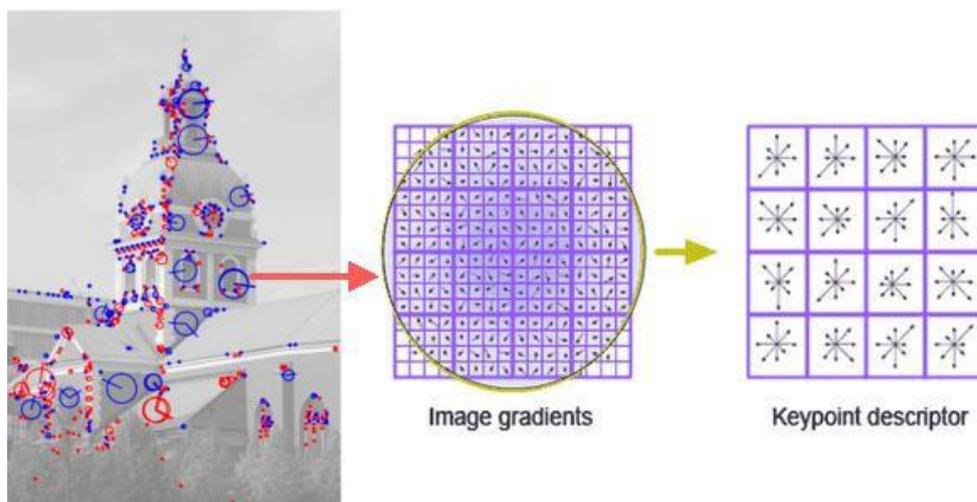
$$\cos \theta = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|} \quad (3)$$

Persamaan (3) digunakan untuk mengukur *similarity* antara dua buah vektor. Dua buah vektor  $\mathbf{v}$  dan  $\mathbf{w}$  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, anda diminta 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 basisdata diasumsikan sudah tersedia. Fitur-fitur dari wajah diekstraksi, begitu pula fitur-fitur wajah di dalam basisdata. Pencocokan kemiripan dilakukan antara vektor fitur citra wajah input dengan vektor-vektor fitur wajah di dalam basisdata. Hasil pencocokan di-*ranking* 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*.

Ada banyak algoritma untuk ekstraksi fitur, yang paling populer adalah [SURF](#), [ORB](#), [SIFT](#), [BRIEF](#). Kebanyakan dari algoritma ini didasarkan pada [image gradient](#). Kita akan menggunakan descriptor KAZE karena sudah tersedia di dalam pustaka OpenCV (lihat Gambar 3).



**Gambar 3.** Ekstraksi fitur dari sebuah citra

(Sumber: <https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774> )

Kode program untuk ekstraksi fitur dapat anda baca pada tulisan ini: *Feature extraction and similar image search with OpenCV for newbies*, pada laman: <https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774>

Hasil ekstraksi fitur adalah vektor berukuran 2048 elemen.

## **SPEKIFIKASI TUGAS**

Buatlah program pengenala wajah dalam Bahasa Python dengan spesifikasi sebagai berikut:

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

## **PROSEDUR Pengerjaan**

1. Tugas dikerjakan secara berkelompok yang terdiri dari 3 orang. Anggota kelompok tidak boleh sama dengan anggota kelompok sebelumnya.
2. Tugas ini dikumpulkan hari Jumat 8 November 2019 paling lambat pukul 9.00 pagi di atas loker Lab IRK (*hard copy* laporan saja, sedangkan *softcopy* program akan dimasukkan secara daring ke sebuah *GoogleDrive*). Silakan isi presensi pengumpulan dan tanggal untuk demo program di depan asisten. Penilaian tugas adalah melalui demo di depan asisten.

## **LAPORAN**

1. *Cover*: *Cover* laporan ada foto anggota kelompok (foto bertiga, bebas gaya). Foto ini menggantikan logo “gajah” ganesha.  
Bab 1: Deskripsi masalah (dapat meng-*copy paste* file tugas ini).  
  
Bab 2: Teori singkat mengenai vektor, norma Euclidean, perkalian titik, pengenalan wajah, dan materi pendukung seperti *OpenCV* dan pustaka yang relevan.

Bab 3: Implementasi program dalam Python.

Bab 4: Eksperimen. Bab ini berisi hasil eksekusi program terhadap contoh-contoh kasus yang diberikan berikut analisis hasil eksekusi tersebut

Bab 5: Kesimpulan, saran, dan refleksi (hasil yang dicapai, saran pengembangan, dan refleksi anda terhadap tugas ini).

Tuliskan juga referensi (buku, web), yang dipakai/diacu di dalam Daftar Referensi.

**Keterangan laporan dan program:**

- a) Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
- b) Laporan tidak perlu memakai *cover* mika dan dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.
- c) Laporan boleh menggunakan kertas rius, boleh bolak-balik, boleh dalam satu halaman kertas terdapat dua halaman tulisan asalkan masih terbaca.
- d) Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).
- e) *Listing* program tidak perlu disertakan pada laporan.
- e) Program disimpan di dalam *folder* Algeo-xxxxx. Lima digit terakhir adalah NIM anggota terkecil. Didalam *folder* tersebut terdapat tiga folder bin, src dan doc yang masing-masing berisi :
  - Folder *bin* berisi *java byte code* (.class)
  - Folder *src* berisi *source code* dari program java
  - Folder *test* berisi data uji.
  - Folder *doc* berisi dokumentasi program dan *readme*

**PENGUMPULAN TUGAS**

1. Yang diserahkan saat pengumpulan tugas adalah:
  - a) Laporan (*hard copy*)
  - b) Kode program + soft copy laporan dikirim ke alamat *Google Drive* yang nanti akan diumumkan oleh asisten.
2. Laporan akan dikembalikan setelah dinilai.

**PENILAIAN**

Komposisi penilaian umum adalah sebagai berikut :

1. Program: 80 %
2. Laporan : 20 %