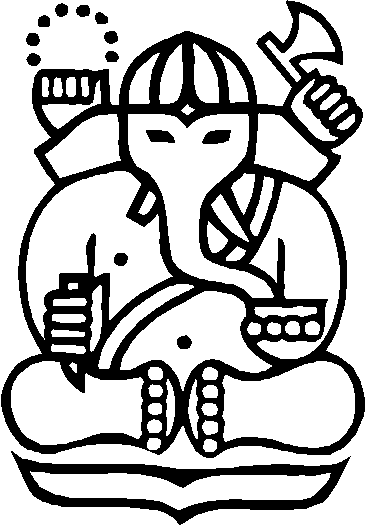
**Tugas UAS**

**IF4073 Interpretasi dan Pengolahan Citra**

**Aplikasi INSTACIT**



Disusun Oleh

Stephen 13512025

Jeffrey Lingga Binangkit 13512059

Ahmad Zaky 13512076

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2015/2016

1. **Pendahuluan**

Aplikasi yang kami buat bernama ***Instacit*** yang memungkinkan pengguna untuk mengunggah foto. Setelah foto diunggah, akan dikenali siapa saja yang ada di foto tersebut berdasarkan data yang didapat dari anggota kelas Pengolahan Citra. Pengguna dapat pula menerapkan *outline effect* untuk foto ini berdasarkan deteksi *edge.*

Untuk mengenali foto, aplikasi ini menerapkan konsep-konsep sebagai berikut:

1. Mendeteksi lokasi yang merupakan wajah dari manusia. Pendeteksian dilakukan dengan berdasarkan.
   1. Warna kulit yang ada dalam palet.
   2. Proporsi bentuk kepala dan mencocokkan lokasi-lokasi mana yang harus berwarna gelap (mata, mulut).
2. Gaussian blur, untuk menghaluskan citra wajah.
3. K-Means, untuk melakukan pengelompokan sebuah citra wajah menjadi elemen-elemen : mata kanan, mata kiri, hidung, mulut.
4. Algoritma Brensenham, untuk menghubungkan titik-titk dalam data.
5. Pencocokan grid untuk masing-masing elemen wajah.
6. **Algoritma**

Untuk memperoleh hasil rekognisi, aplikasi ini memproses gambar dengan mengikuti urutan sebagai berikut.

1. **Memasukkan gambar**

Gambar diinput melalui Kamera atau melalui Galeri.

1. **Mendeteksi lokasi wajah**

Program telah memiliki palet warna yang berisi *range* warna kulit anggota kelas.

* Menelusuri gambar
* Bila ditemukan sebuah pixel warna yang merupakan anggota palet
  + Melakukan penelusuran ke pixel-pixel lain di sekitar pixel tersebut yang warnanya juga merupakan anggota dari palet warna. Penelusuran dilakukan secara *flooding* (mirip *floodfill*).
  + Telah ditemukan kotak terduga wajah, disimpan *boundary-*nya (xmin, xmax, ymin, ymax).
* Setelah ditemukan lokasi-lokasi yang diduga wajah, untuk memastikan bagian gambar tersebut merupakan wajah, maka dilakukan pengecekan sebagai berikut
  + Pengecekan proporsi : wajah memiliki proporsi tinggi:lebar sebesar 1:1, sampai 1:1,5. Bila proporsinya tidak cocok, maka dikeluarkan dari kandidat wajah.
  + Pengecekan mata : pada sekitar 1/3 atas wajah, seharusnya ada dua bagian yang lebih gelap yaitu mata. Bila tidak ada, maka kandidat ini dikeluarkan.
  + Pengecekan mulut : pada sekitar ¼ bawah wajah, seharusnya ada sebuah bagian yang berbeda warnanya yaitu mulut. Bila tidak ada, maka dikeluarkan dari kandidat wajah.
* Telah ditemukan kandidat-kandidat wajah.

1. **Memperhalus citra wajah menggunakan Gaussian Blur**

Setelah didapatkan wajah-wajah yang akan dideteksi, maka dilakukan penghalusan wajah dengan menggunakan algoritma Gaussian Blur.

Algoritma Gaussian Blur adalah sebagai berikut :

* Untuk setiap pixel dalam gambar di lokasi (x,y), terapkan rumus distribusi Gauss sebagai berikut

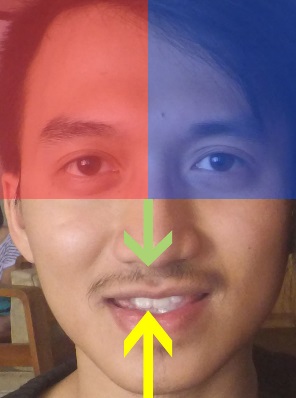
G(x,y)={\frac {1}{2\pi \sigma ^{2}}}e^{-{\frac {x^{2}+y^{2}}{2\sigma ^{2}}}}

* + X dan Y di sini adalah koordinat di sumbu horizontal dan vertikal di bidang kartesian. σ adalah standar deviasi dari distribusi gauss.
  + Terbentuk sebuah convolution matrix. Setiap pixel diberikan nilai baru yang merupakan rata-rata bobot pixel tetangganya, dimana titik original memiliki bobot terbesar, dan tetangga-tetangganya semakin jauh dari titik original akan semakin kecil bobotnya. Convolution matrix yang perlu dihitung adalah berukuran 6σ x 6σ.
* Telah terbentuk citra-citra wajah yang telah lebih halus.

1. **Melakukan clustering dengan K-Means**

Untuk setiap wajah yang didapatkan dalam tahap sebelumnya, dilakukan pengelompokan untuk mata, hidung, dan mulut. Pengelompokan ini dilakukan dengan algoritma clustering K-Means yang dimodifikasi, yang dapat dilihat sebagai berikut.

* Ubah gambar menjadi grayscale, tentukan threshold 182, kemudian ubah gambar menjadi biner sehingga kulit biasa menjadi putih, dan elemen-elemen mata, hidung, dan mulut menjadi hitam.
* Tentukan 4 buah *centroid*. Dalam algoritma K-Means biasa, *centroid* ditentukan secara acak. Namun, dalam menentukan *centroid* untuk kelompok mata kanan, kiri, hidung, dan mulut, digunakan cara heuristik.
  + Untuk mencari *centroid* mata kanan, hanya perlu menelusuri bagian merah saja, dari kanan ke kiri (sudut pandang kita) sampai ditemukan titik hitam.
  + Untuk mencari *centroid* mata kiri, hanya perlu menelusuri bagian biru saja, dari kiri ke kanan (sudut pandang kita) sampai ditemukan titik hitam.
  + Untuk mencari *centroid* hidung, dimulai dari titik tengah gambar, kemudian ditelusuri pixel-per-pixel ke arah bawah sampai ditemukan titik hitam. Dapat dilihat dengan panah warna hijau.
  + Untuk mencari *centroid* mulut, dimulai dari titik tengah-bawah gambar, kemudian ditelusuri pixel-per-pixel ke arah atas sampai ditemukan titik warna hitam. Dapat dilihat dengan panah warna cokelat.
* Lakukan langkah-langkah di bawah ini, sampai tidak ada anggota *cluster* yang berubah lagi:
  + Untuk setiap titik berwarna hitam, cari *centroid* yang terdekat darinya.
    - Masukkan titik tersebut ke *cluster* milik *centroid* tersebut.
  + Untuk setiap *cluster* mata kanan, kiri, hidung, dan mulut
    - Cari rata-rata nilai x dari seluruh titik yang ada di *cluster* tersebut.
    - Cari rata-rata nilai y dari seluruh titik yang ada di *cluster* tersebut.
    - Update koordinat *centroid* menjadi nilai (x, y) hasil rata-rata tadi.
* Setelah tidak ada perubahan, maka *cluster-cluster* mata, hidung, dan mulut sudah terbentuk dan stabil.



Gambar 1. Area yang Digunakan Dalam Penentuan Centroid Seed dengan Menggunakan Cara Heuristik

1. **Melakukan pengenalan per cluster menggunakan Grid**
2. **Menampilkan ke pengguna**
3. **Melakukan *outline effect* (opsional)**