



**TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI PADANG**

# **Feature Extraction (Template Matching)**

**AINIL MARDIAH, M.Cs.**  
**Program Studi Teknologi Rekayasa Perangkat Lunak**  
**Jurusan Teknologi Informasi**  
**Politeknik Negeri Padang**  
**Email: ainilmardiah@pnp.ac.id**

# Apakah Fitur Ekstraksi

---

- Perubahan data input menjadi sekelompok fitur
- Fitur adalah sifat khusus dari pola masukan yang membantu dalam membedakan antara kategori pola masukan
- Ini adalah proses yang digunakan untuk menghasilkan karakteristik tertentu yang mendasar dan membedakan dari sesuatu



# Fitur Citra

---

- Fitur Dasar Citra:
  - Warna/Intensitas
  - Tepi/Batasan
  - Sudut
  - Garis/Line
  - Tekstur
- Kita hanya akan belajar fitur berdasarkan intensitas:
  - Pencocokan fitur warna (Pencocokan Template)
  - Fitur Haar-like



# Ekstraksi Fitur Warna

- Kita secara langsung menggunakan nilai intensitas citra sebagai fitur yang disimpan dalam data array 1-D
- Jika kita memiliki dua citra:
  - Ubah ukuran dua citra menjadi ukuran yang sama
  - Ekstraksi fitur warna
  - Hitung jarak antara dua fitur
- Ini sederhana, tetapi terkadang tidak kuat terhadap rotasi



# Distance Metrics

- Euclidean Distance

$$\begin{aligned}d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\&= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.\end{aligned}$$

- Manhattan/City Block Distance

$$d(p, q) = \sum_i^n |p_i - q_i|$$



# Mengakses Nilai Piksel di Citra

```
>>> import cv2
>>> import numpy as np

>>> img = cv2.imread('messi5.jpg')

>>> px = img[100,100]
>>> print px
[157 166 200]

# accessing only blue pixel
>>> blue = img[100,100,0]
>>> print blue
157
```



# Perbandingan Dua Citra



# Pendekatan Citra dengan Intensitas

```
import cv2
import math

w=200
h=150

image1=cv2.imread("horse01.jpg")
gray1=cv2.cvtColor(image1,cv2.COLOR_BGR2GRAY)
resize1 = cv2.resize(gray1, (w,h))

image2=cv2.imread("face.jpg")
gray2=cv2.cvtColor(image2,cv2.COLOR_BGR2GRAY)
resize2 = cv2.resize(gray2, (w,h))

dist=0;
for i in range(1,w):
    for j in range(1,h):
        dist = dist + ((int(resize1[j,i])- int(resize2[j,i]))**2)

dist = math.sqrt(dist)/(w*h)
print(dist)
```





# Template Matching



# Template Matching in Python

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('messi.jpg',0)

template = cv2.imread('messi_face.jpg',0)
w, h = template.shape[::-1]

# Apply template Matching
res = cv2.matchTemplate(img,template,cv2.TM_CCOEFF)
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)

# Draw Bounding Box
top_left = max_loc
bottom_right = (top_left[0] + w, top_left[1] + h)
image=cv2.rectangle(img,top_left, bottom_right, 255, 2)

#Showing
cv2.imshow('img',image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



# Bisakah Template Matching Digunakan di Video?

- Deteksi wajah menggunakan template matching:
  - Ambil citra menggunakan webcam
  - Potong di sekitar wajah untuk digunakan sebagai template
  - Gunakan template matching
  - Coba template yang berbeda :
    - Contohnya menggunakan topi atau kaca mata atau lainnya, lihat apa yang terjadi?
    - Gunakan wajah template wajah temanmu



# Bagaimana Jika Menggunakan Banyak Wajah dalam Citra?

---



# Deteksi Banyak Wajah





# Deteksi Banyak Wajah

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('facemulti.jpg',0)

template = cv2.imread('template_multi.jpg',0)
w, h = template.shape[::-1]

# Apply template Matching
res = cv2.matchTemplate(img,template,cv2.TM_CCOEFF_NORMED)
threshold = 0.4
loc = np.where( res >= threshold)
for pt in zip(*loc[::-1]):
    img=cv2.rectangle(img, pt, (pt[0] + w, pt[1] + h), (0,0,255), 2)

#Showing
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



# Deteksi Banyak Wajah

