



# Image Processing and Unsupervised Learning

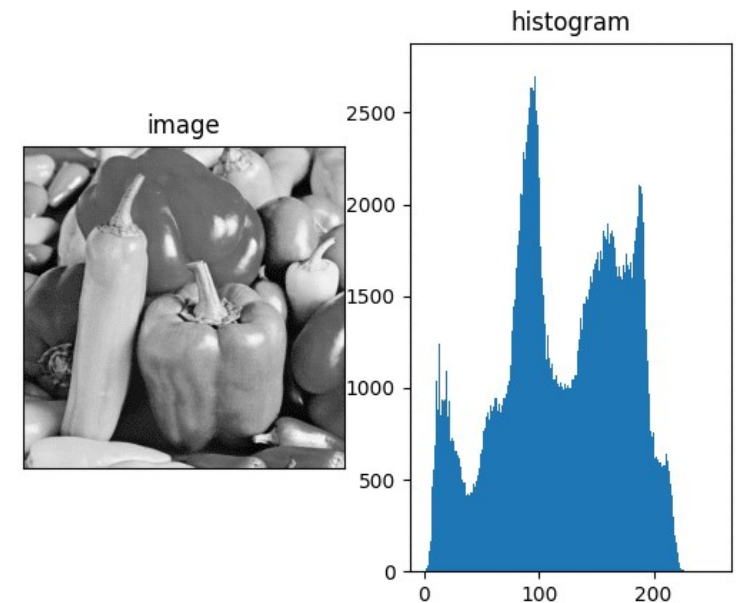
*Pertemuan 7*  
*MK Data Mining II*

**M. N. Fakhruzzaman, S.Kom., M.Sc.**  
**Ratih Ardiati Ningrung, S.Si., M.S., M.Stat.**  
**Malikhah, S.Kom., M.Kom.**

Program Studi S1 Teknologi Sains Data  
Fakultas Teknologi Maju dan Multidisiplin  
Universitas Airlangga Indonesia

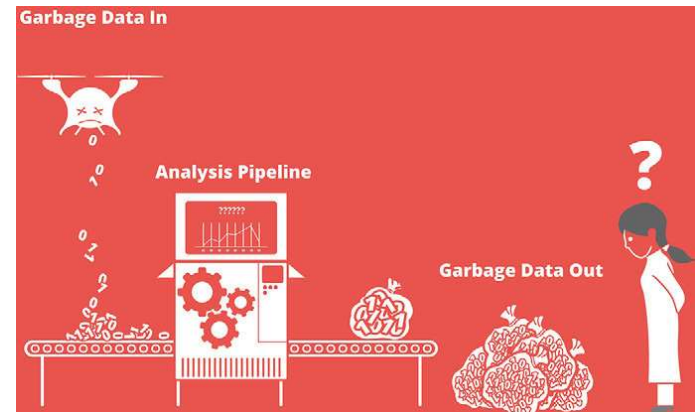
# Feature Extraction

- Fitur adalah bagian atau pola objek dalam citra untuk membantu mengidentifikasi citra.
- 2 macam fitur dalam citra:
  - Fitur alami: fitur yang merupakan bagian dari gambar, misal: kecerahan dan tepi dari objek
  - Fitur buatan: fitur yang diperoleh dengan operasi tertentu pada gambar, misal: histogram dari graylevel.
- Feature Extraction adalah pengambilan fitur atau ciri dari suatu citra, dimana nilai yang didapatkan kemudian bisa dianalisis pada proses selanjutnya.



# Feature Extraction

- Fitur digunakan untuk proses selanjutnya (pengenalan pola, klasifikasi, dll)
- Jika fitur bagus maka hasil akan bagus, dan sebaliknya
- Contoh kasus: Fitur apa yang dapat membedakan wajah satu dengan wajah yang lain?
  1. Warna kulit
  2. Tekstur wajah
  3. Jarak antara mata
  4. Jarak mata dengan hidung
  5. Jarak antara hidung dan mulut
  6. dsb



# Feature Vector

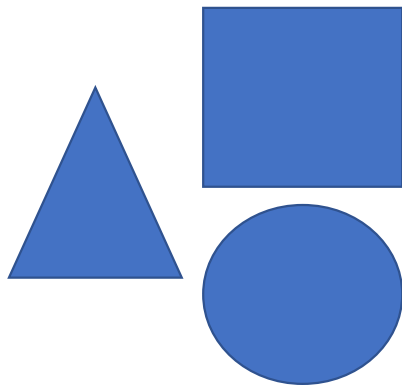
- vektor fitur adalah representasi ringkas dari suatu gambar (atau objek di dalam gambar)
- Vektor fitur direpresentasikan sebagai array  $n \times 1$  yang mengkodekan  $n$  fitur dari suatu gambar
- vektor fitur numerik  $x$  dapat direpresentasikan sebagai berikut:

$$x = (x_1, x_2, \dots, x_n)^T$$

dimana :  $n$  = jumlah fitur dan  $T$  = operasi transpose

# Type of Feature Extraction

Bentuk



Warna



Tekstur



# Color Feature Extraction

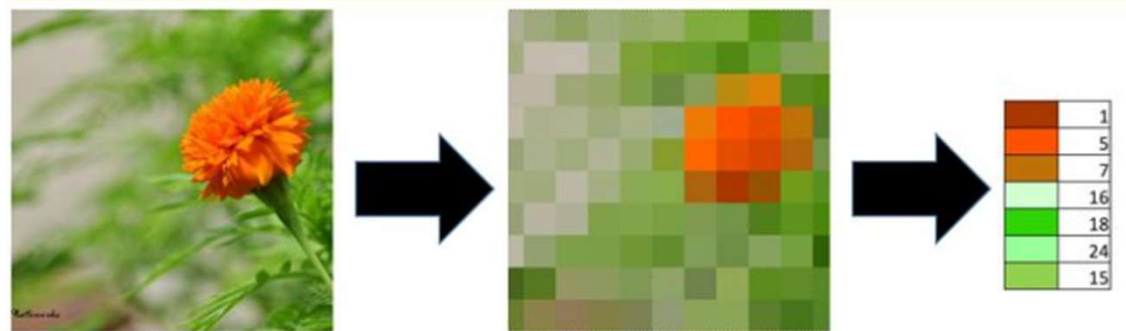
# Color Feature Extraction

- Warna bisa membedakan objek pada citra
- Beberapa objek dapat dibedakan berdasarkan warna: bunga, landscape



# Color Features in Image

- Ciri warna pada citra direpresentasikan dengan distribusi warna dan warna dominan pada citra
- Distribusi warna adalah berapa banyak suatu warna muncul, disebut juga histogram warna
- Warna dominan bisa dihasilkan dengan menggunakan segmentasi atau melakukan clustering warna



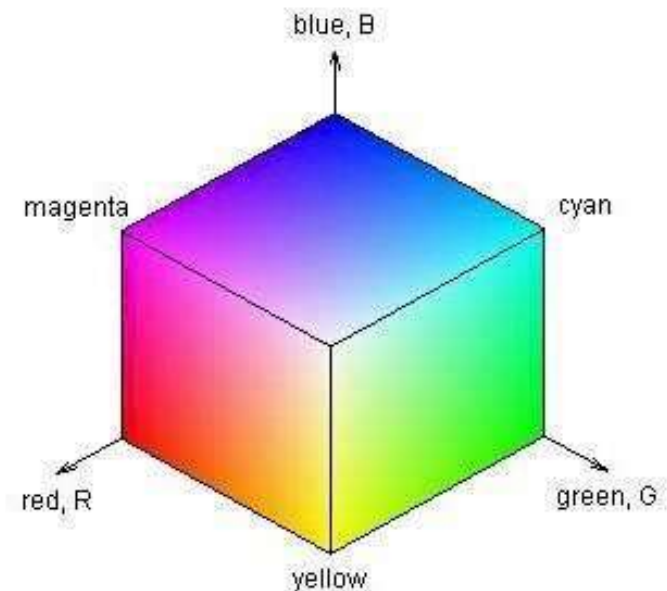


# Color

- Manusia melihat warna karena cahaya dipantulkan oleh objek. Spektrum cahaya kromatis 400 – 700 nm.
- Karakteristik persepsi mata manusia dalam membedakan warna satu dengan yang lain:
  - Hue: warna yang dikenal manusia, seperti merah, hijau, biru. Hue mencerminkan warna yang ditangkap oleh mata manusia yang menanggapi berbagai nilai panjang gelombang cahaya.
  - Saturation: tingkat kemurnian warna. Semakin besar nilai saturation maka semakin murni warna yang dihasilkan.
  - Brightness / lightness: kecerahan atau intensitas pantulan objek yang diterima mata.
- Ruang Warna (color space): spesifikasi system koordinat gabungan dan suatu sub ruang dari komponen setiap warna

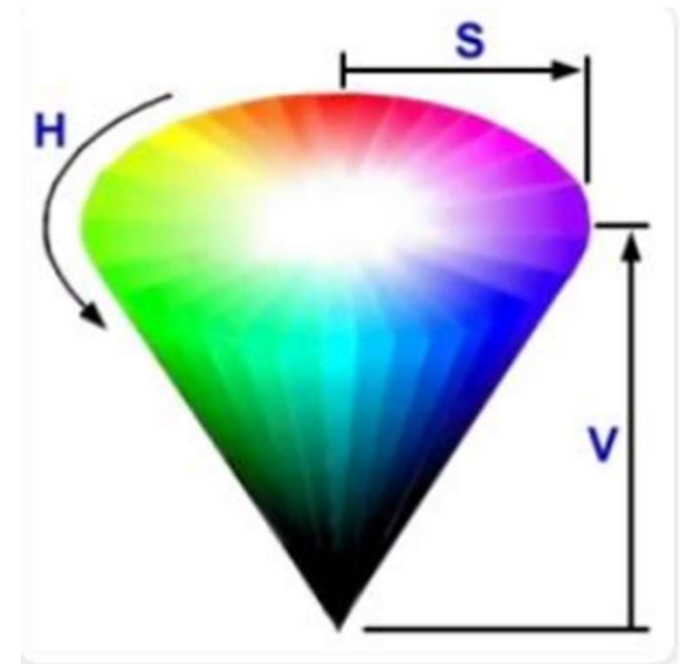
# Color space - RGB

- Dimodelkan dalam RGB Cube. RGB cube digunakan untuk mendefinisikan sebuah warna pada sebuah citra dengan model yang mudah dijelaskan.
- Ruang warna RGB diterapkan pada monitor CRT dan system grafika komputer.
- RGB Cube menunjukkan bagaimana warna dihasilkan oleh komponen/warna primer R, G, dan B yang didasarkan pada hasil akuisisi oleh sensor elektronik
- Sebuah warna  $W$  dinyatakan sebagai  $W(r, g, b)$
- Pada citra 24 bit, nilai dari R, G, dan B memiliki range 0 – 255 (tiap warna memiliki kedalaman 8 bit)



# Color space - HSV

- Ruang warna HSV merepresentasikan warna seperti yang dilihat oleh mata manusia
- HSV mengacu pada Hue, Saturation, dan Value.
  - Hue: warna sebenarnya, seperti merah, violet, kuning, dsb
  - Saturation: kemurnian atau kekuatan warna (amount of gray), semakin kecil nilai semakin mendekati gray.
  - Value: kecerahan warna, nilai 0 berarti berwarna hitam, dan semakin besar nilai maka semakin cerah.
- Untuk mendapatkan ketiga nilai tersebut, perlu dilakukan konversi ruang warna citra yang semula RGB (Red, Green, Blue) menjadi HSV (Hue, Saturation, Value)



# RGB to HSV

$$r = \frac{R}{255}$$

$$g = \frac{G}{255}$$

$$b = \frac{B}{255}$$

$$Cmax = \max(r, g, b)$$

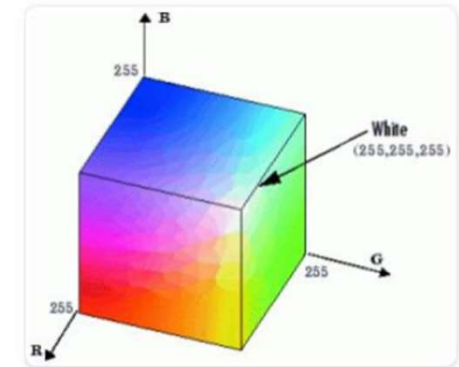
$$Cmin = \min(r, g, b)$$

$$\Delta = Cmax - Cmin$$

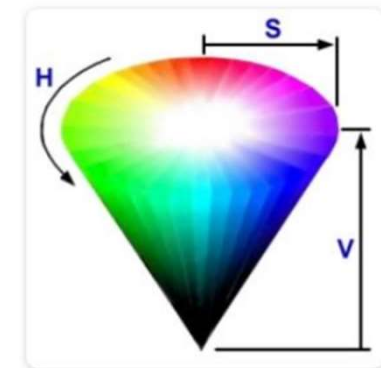
$$V = Cmax$$

$$S = \begin{cases} 0, & \text{jika } Cmax = 0 \\ \frac{\Delta}{Cmax}, & \text{jika } Cmax \neq 0 \end{cases}$$

$$H = \begin{cases} 0^\circ, & \text{jika } \Delta = 0 \\ 60^\circ \times \left( \frac{g - b}{\Delta} \bmod 6 \right), & \text{jika } Cmax = r \\ 60^\circ \times \left( \frac{b - r}{\Delta} + 2 \right), & \text{jika } Cmax = g \\ 60^\circ \times \left( \frac{r - g}{\Delta} + 4 \right), & \text{jika } Cmax = b \end{cases}$$



rgb colorspace



hsv colorspace

# Grayscale



- Citra grayscale hanya memiliki satu nilai warna (monokrom)
- Citra grayscale tidak memberikan informasi apapun tentang warna.
- Tiap pixel menentukan skala keabuan yang berbeda.
- Citra grayscale normal berisi 8 bit.
- Konversi RGB ke Grayscale:
  - Rata-rata:

$$grayscale = \frac{R + G + B}{3} = \frac{R}{3} + \frac{G}{3} + \frac{B}{3}$$

- Bobot:

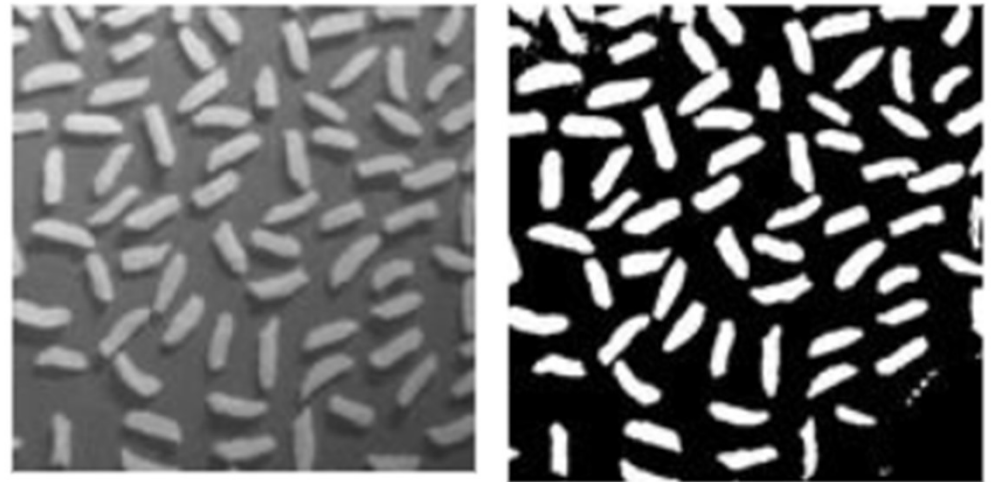
$$Grayscale = 0.299R + 0.587G + 0.114B$$

# Binary

- Citra biner hanya memiliki 2 nilai: Hitam dan Putih atau 0 dan 1
- Citra 1 bit
- Dihasilkan dengan melakukan operasi *threshold* ( $T$ ).

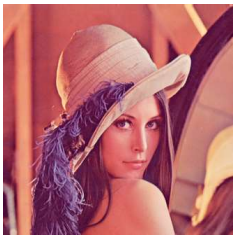
$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases}$$

- 2 cara menentukan threshold:
  - Local threshold
  - Global threshold



# Grayscale vs Color (RGB) Feature Vector

- Kompleksitas komputasi dari citra grayscale dan biner lebih kecil dari citra berwarna.



Ukuran citra berwarna: (330,330,3)

Ukuran fitur = **326.700**



Ukuran citra grayscale: (330,330)

Ukuran fitur = **108.900**

# Color Statistic

- Fitur warna dapat diperoleh melalui perhitungan statistik (Martinez & Martinez, 2002) pada komponen R, G, dan B.

Dimana:

$X$  = tinggi citra,

$Y$  = lebar citra,

$P_{xy}$  = nilai warna pada baris ke- $x$  kolom ke- $y$

$$\text{rata-rata } (\mu) = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y P_{xy}$$

$$\text{standar deviasi } (\sigma) = \sqrt{\frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y (P_{xy} - \mu)^2}$$

$$\text{skewness } (\theta) = \frac{\sum_{x=1}^X \sum_{y=1}^Y (P_{xy} - \mu)^3}{XY \sigma^3}$$

$$\text{kurtosis} = \frac{\sum_{x=1}^X \sum_{y=1}^Y (P_{xy} - \mu)^4}{XY \sigma^4}$$



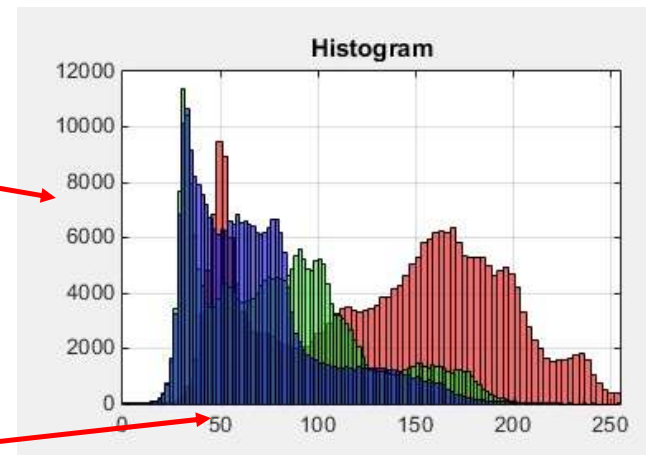
# Histogram Warna

- Histogram Warna  $H(w)$  adalah menyatakan frekuensi munculnya warna  $w$
- Warna  $w$  merupakan kombinasi elemen dasar

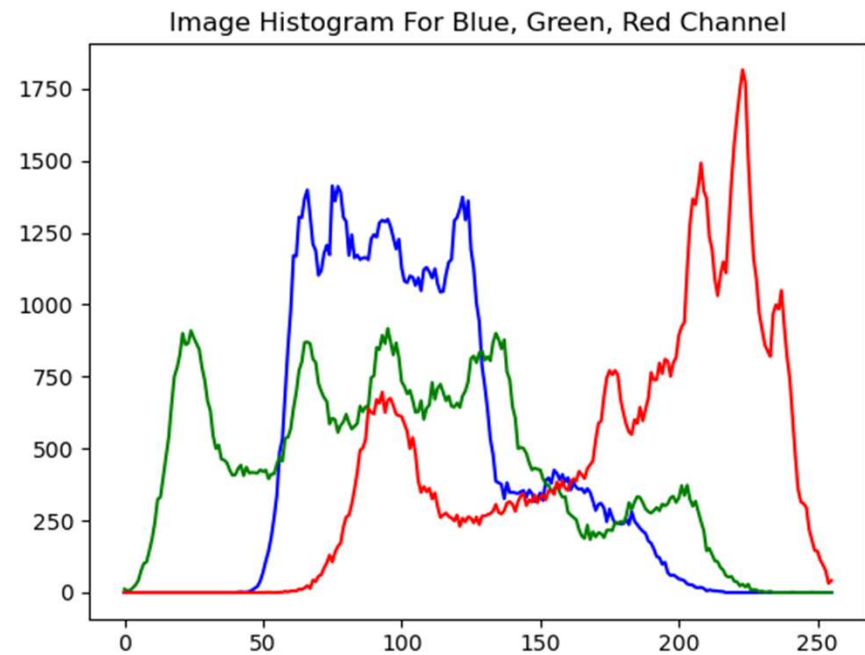


Frekuensi

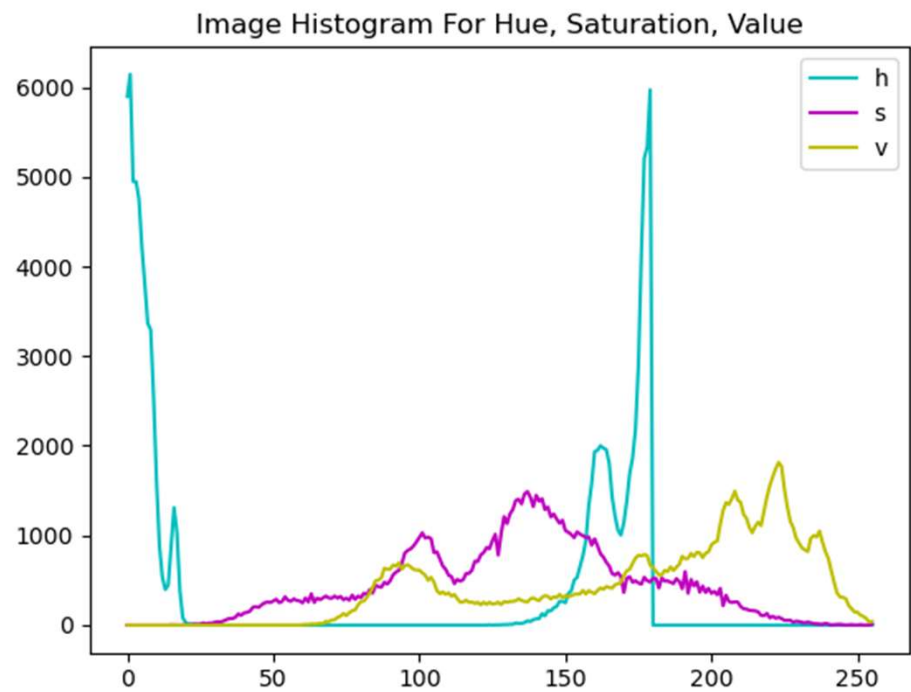
Nilai warna



# Histogram RGB

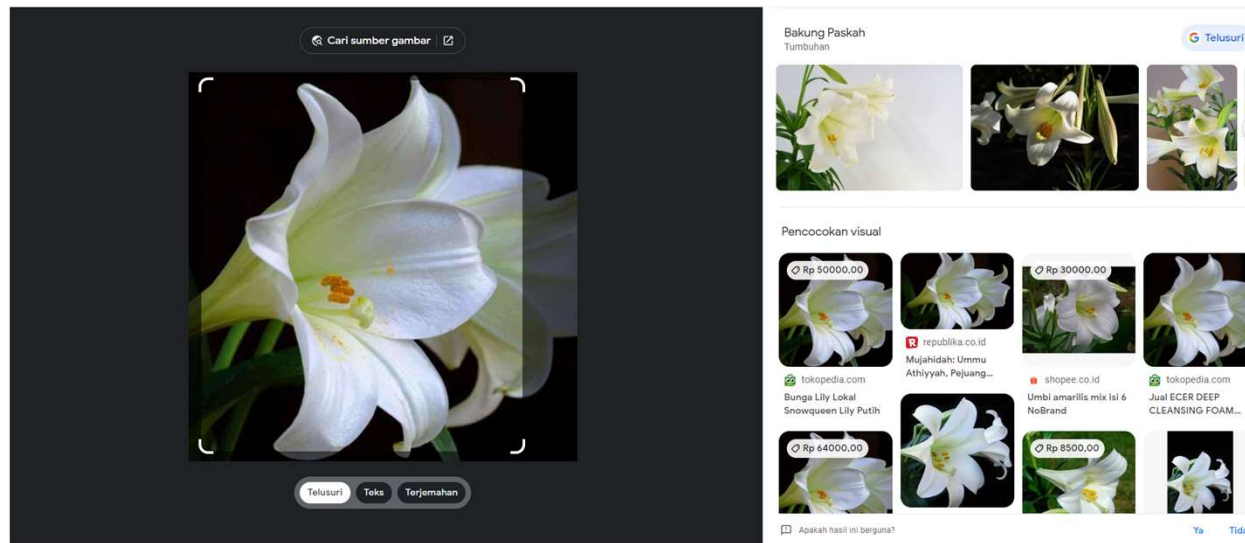


# Histogram HSV



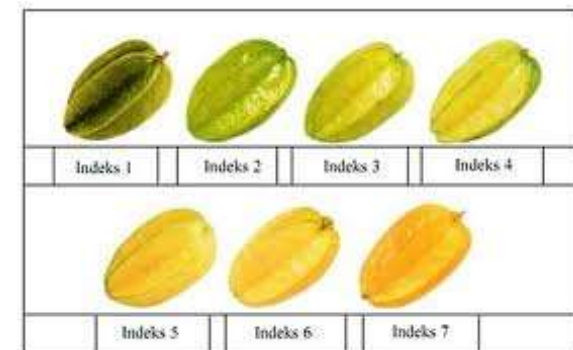
# Application (1)

- Image Google Search menggunakan konsep pencarian berdasarkan kemiripan gambar dan warna
- Salah satu fitur yang digunakan adalah warna



# Application (2)

- Mendeteksi kematangan buah menggunakan warna dominan sebagai fitur





# Application (3)

- Deteksi object berdasarkan warna

Original Image



Mask of red Color



# Feature Transformation

# Feature Transformation

- Proses merubah fitur ke representasi lain sehingga lebih mudah dipahami oleh mesin
- Alasan melakukan transformasi fitur:
  - Mendapatkan fitur penting
  - Mereduksi dimensi fitur
  - Memudahkan untuk visualisasi data





# Feature Transformation

- Beberapa cara transformasi fitur:
  - Transformasi wavelet
  - Discrete Wavelet Transform (DWT)
  - **Principal Component Analysis (PCA)**

# Principal Component Analysis

Country	Salesperson	Order date	order id	units	order amounts
USA	tom	1/2/2021	313	10	140000
UK	brown	1/5/2021	324	2	28000
USA	kevin	1/2/2021	213	3	42000
USA	Callum	1/4/2021	324	5	70000
UK	Scott	4/2/2021	452	12	168000
UK	Ryan	3/2/2021	524	3	42000
USA	Fullham	1/2/2021	123	6	84000
USA	tom	1/2/2021	455	2	28000
UK	brown	1/2/2021	244	3	42000
UK	scott	1/2/2021	424	4	56000

inkonsistensi

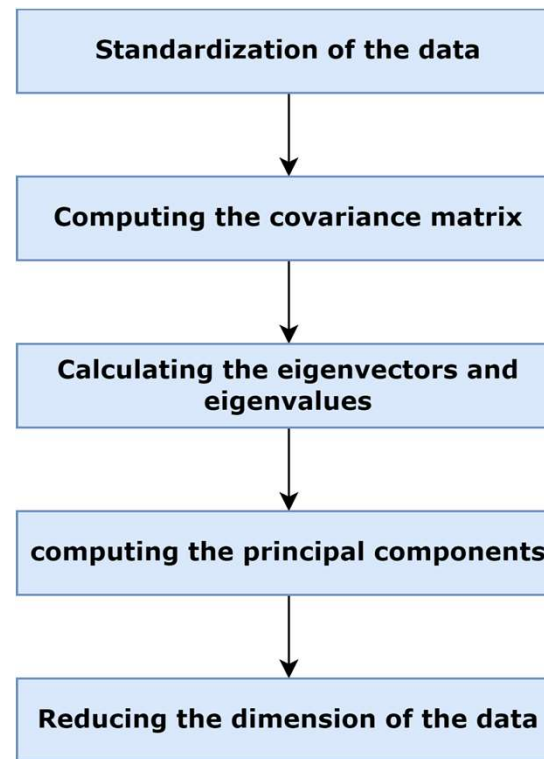
data redundan

fitur berkorelasi tinggi

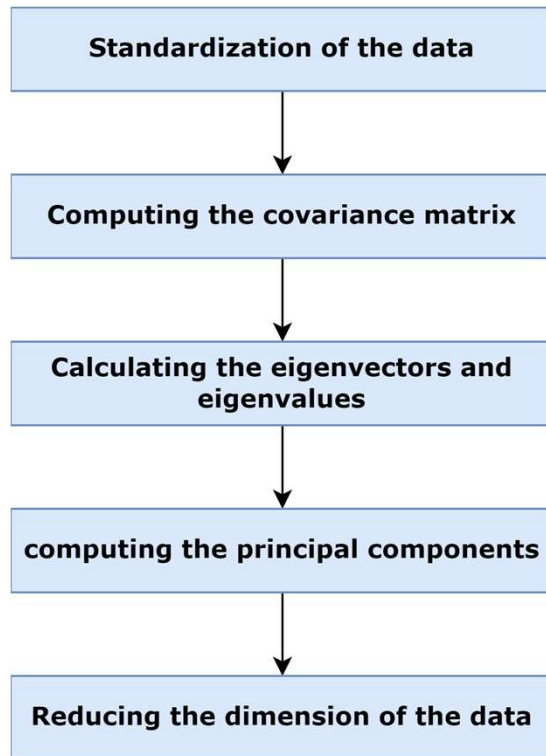
order id	komponen 1	komponen 2
313	370	5180000
324	74	1036000
213	111	1554000
324	185	2590000
452	444	6216000
524	111	1554000
123	222	3108000
455	74	1036000
244	111	1554000
424	148	2072000

**PCA** merupakan salah satu teknik transformasi data yang digunakan untuk mengidentifikasi korelasi dan pola dalam dataset sehingga dapat ditransformasikan menjadi dataset baru yang memiliki dimensi lebih rendah tanpa kehilangan informasi penting.

# Step by Step PCA



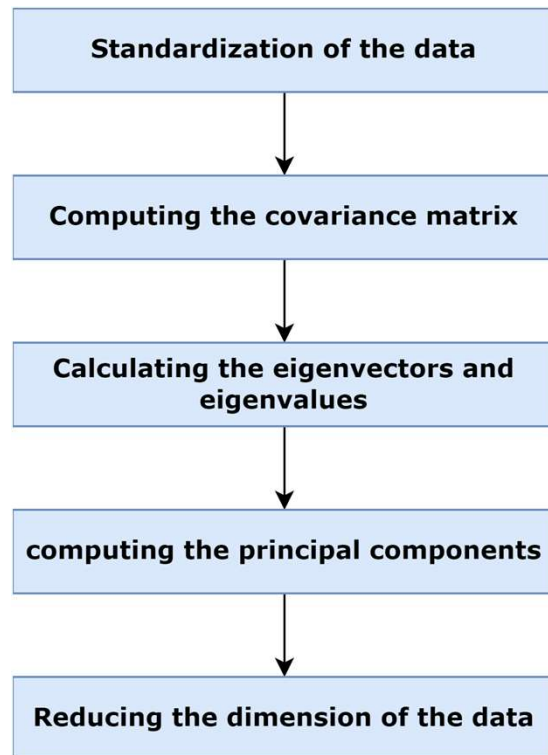
# (1) Standardization



$$\text{standardization} = \frac{\text{variable value} - \text{mean}}{\text{standard deviation}}$$

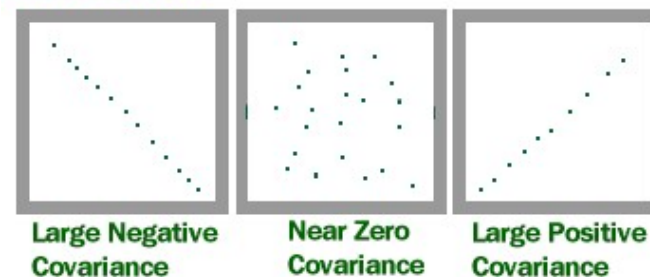
- scaling data sehingga semua variable berada pada range tertentu

## (2) Compute the covariance matrix

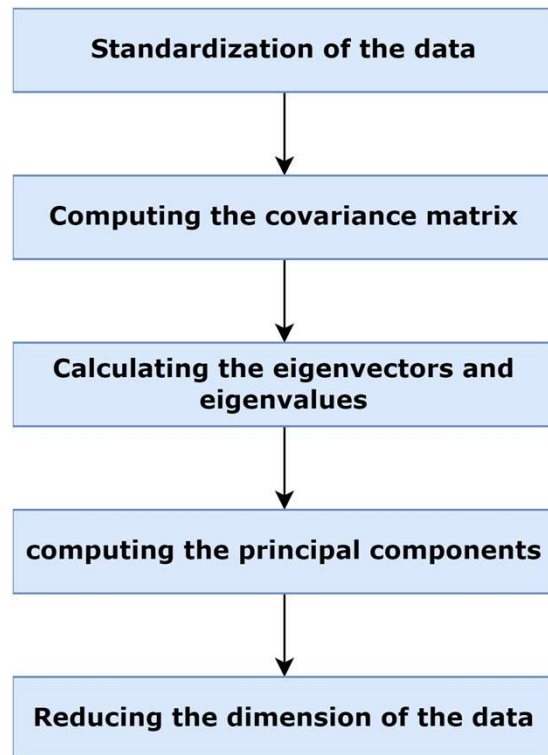


- Melihat hubungan antara 2 variabel
- Nilai covariance menyatakan:
  - Dependensi antar dua variable
- Kovarians negative: ada hubungan terbalik antara variable/ketika salah satu meningkat, yang lain cenderung menurun (inversely corelated).
- kovarians nol: tidak ada hubungan antara nilai-nilai dari dua variabel.
- Kovarians positive: hubungannya linier, kalau satu naik yang lain ikut naik (corelated)

### COVARIANCE



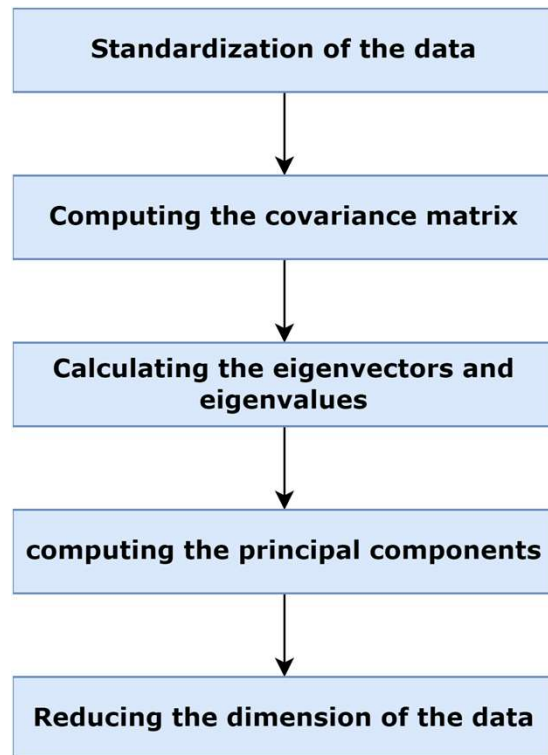
## (2) Compute the covariance matrix



- Matrix covariance merupakan matrix simetris dengan ukuran  $p \times p$ , dimana  $p$  merupakan jumlah dimensi
- Misal, dataset dengan 3 variable  $x, y, z$  akan memiliki matriks covariance sbb:

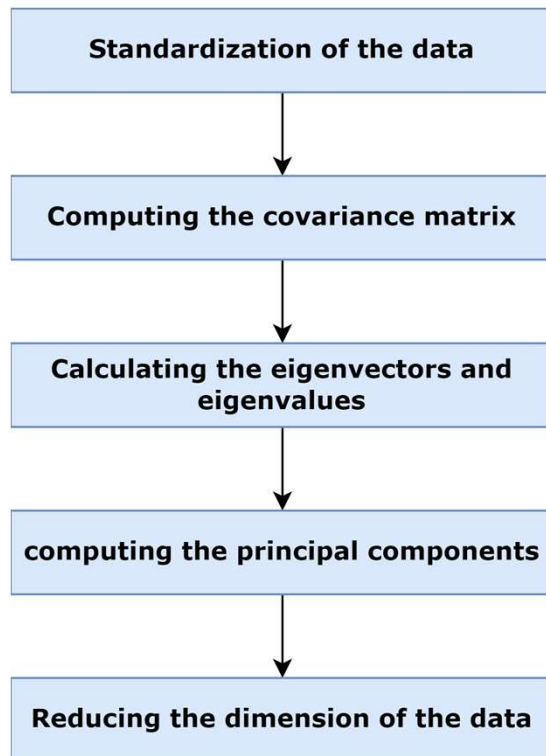
$$\begin{bmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{bmatrix}$$

### (3) Calculating the eigenvectors and eigenvalues

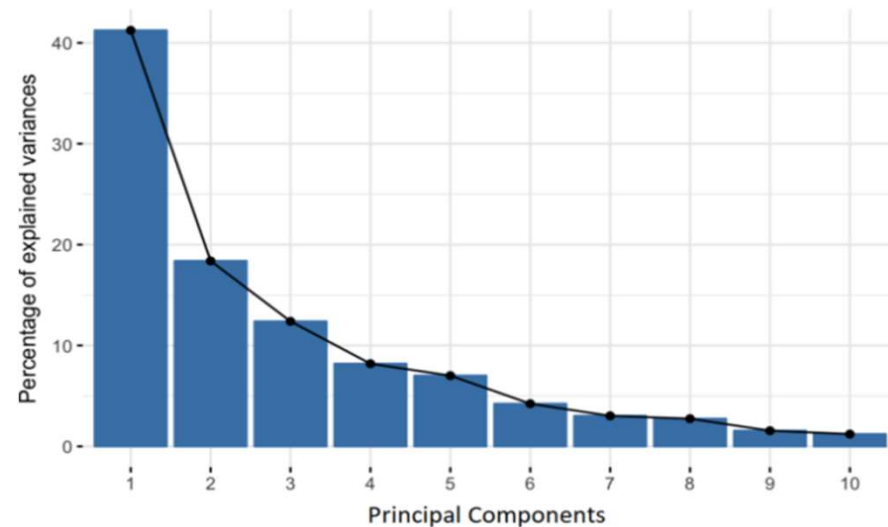


- Eigenvectors adalah vektor-vektor itu ketika transformasi linier dilakukan pada mereka, arahnya tidak berubah
- Eigenvalues hanya menurunkan skalar dari masing-masing eigenvectors
- Eigenvectors dan Eigenvalues digunakan untuk menentukan Principal Component (komponen utama) dari dataset

## (4) Computing princinal components



- Principal Component adalah himpunan variable/fitur baru yang diperoleh dari himpunan variable/fitur awal.
- Principal Component memampatkan dan memiliki sebagian besar informasi berguna yang tersebar di antara variable/fitur awal
  - PC1 → menangkap variasi paling banyak, komponen paling utama
  - PC2 → menangkap varisi paling banyak kedua, dst





# PCA for image processing

What if we have 32 instances of images?



Citra 321 x 216

Dimension =  $321 \times 261 =$   
**83781**



# PCA for image processing



**Original images  
(dimension = 83781)**

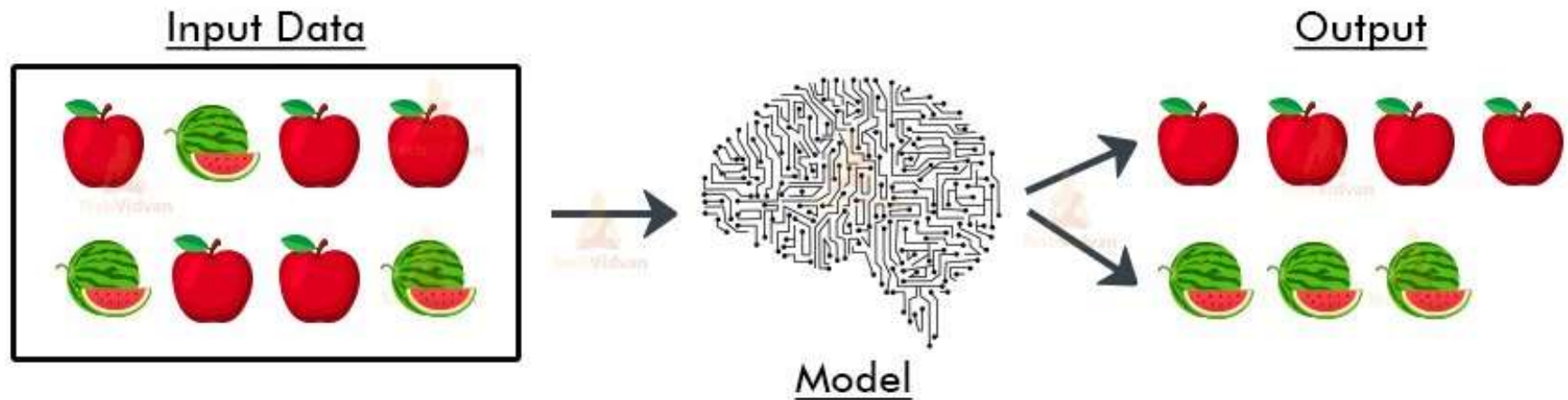


**4 components**

- PCA juga dapat digunakan untuk melakukan kompresi pada citra asli, misalkan dari 83781 fitur, kita hanya mengambil 4 komponen utama saja.
- Kualitas citra dengan 4 komponen utama tidak sama dengan citra yang asli namun masih memiliki karakteristik citra aslinya

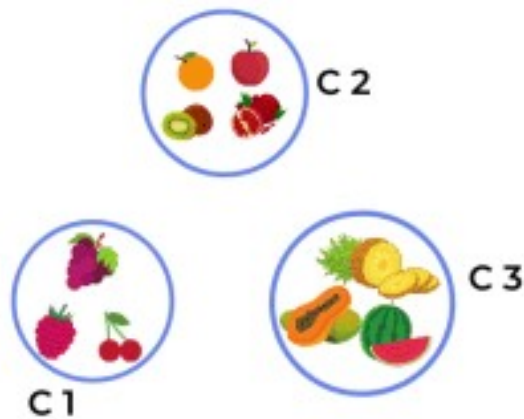
# Unsupervised Learning

# Unsupervised Learning

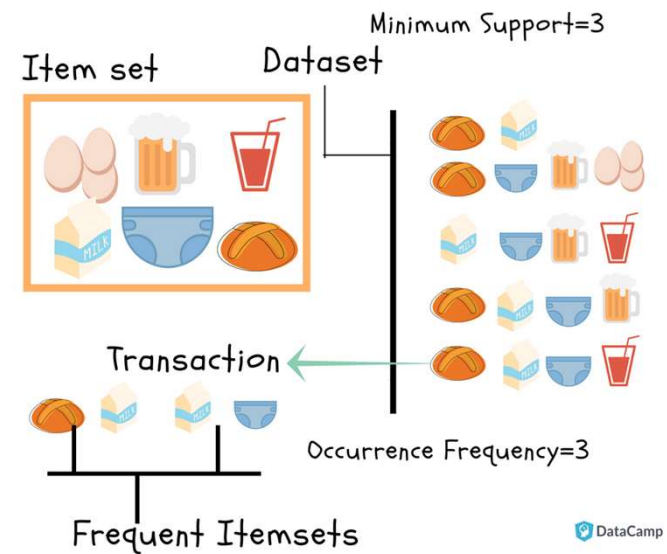


**Unsupervised Learning** mengidentifikasi pola dalam kumpulan data yang berisi titik data yang tidak **ditandai** (kelas/label)

# Type of Unsupervised Learning



## Clustering

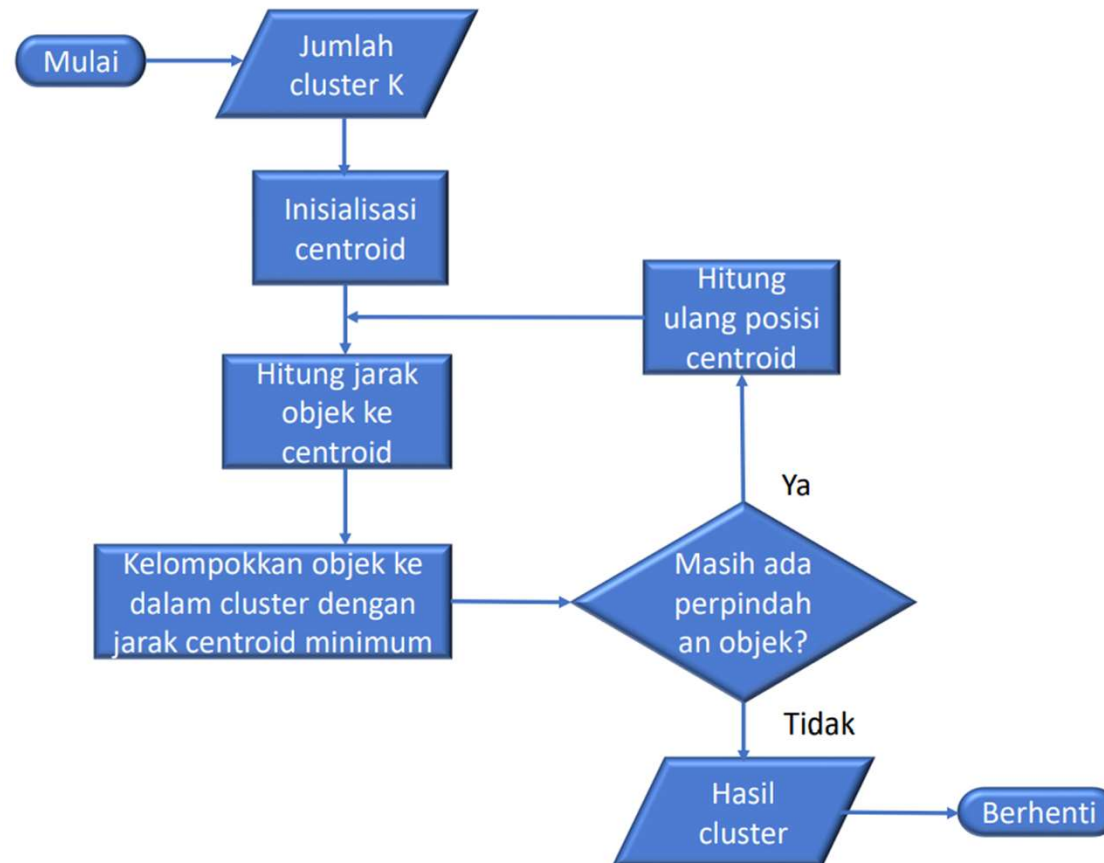


## Association

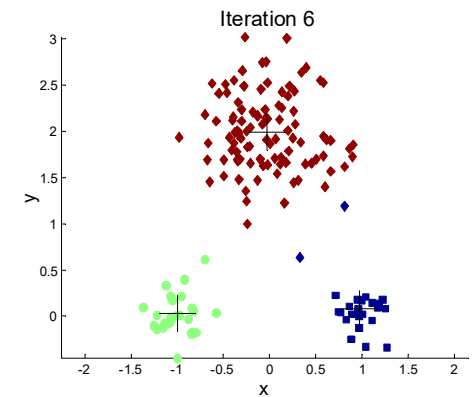
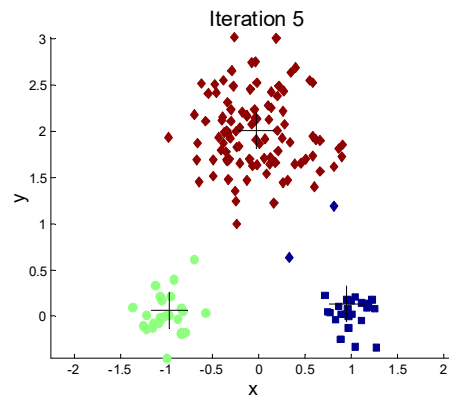
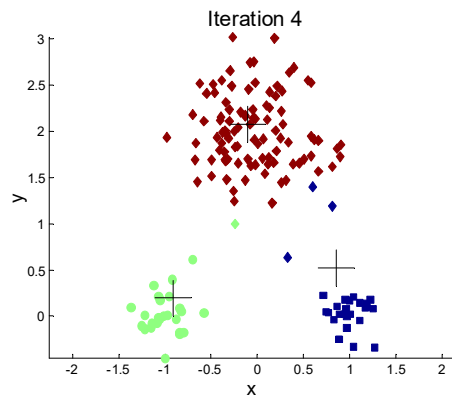
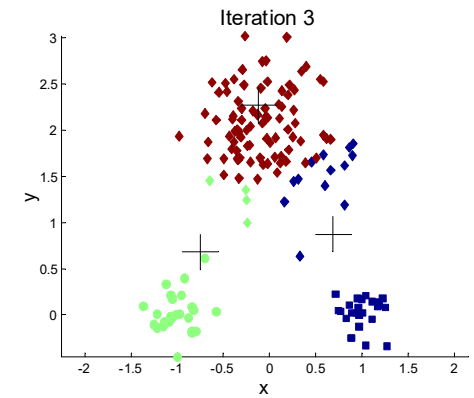
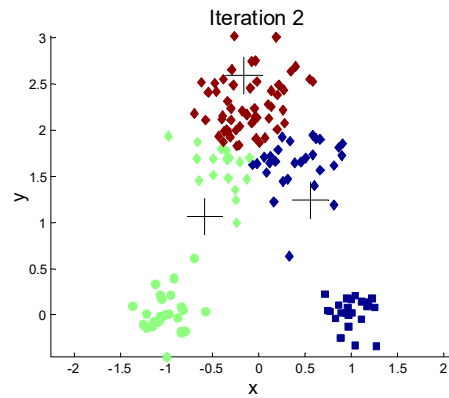
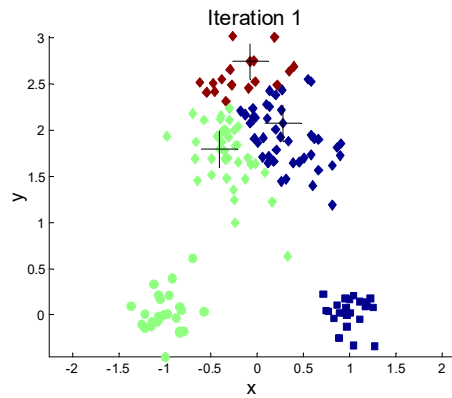
# Unsupervised Learning

- Clustering
  - **K-means**
  - Hierarchical Clustering
  - DBSCAN
  - Fuzzy C-Means
  - Local Outlier Factor
  - dll

# K-Means



# K-Means

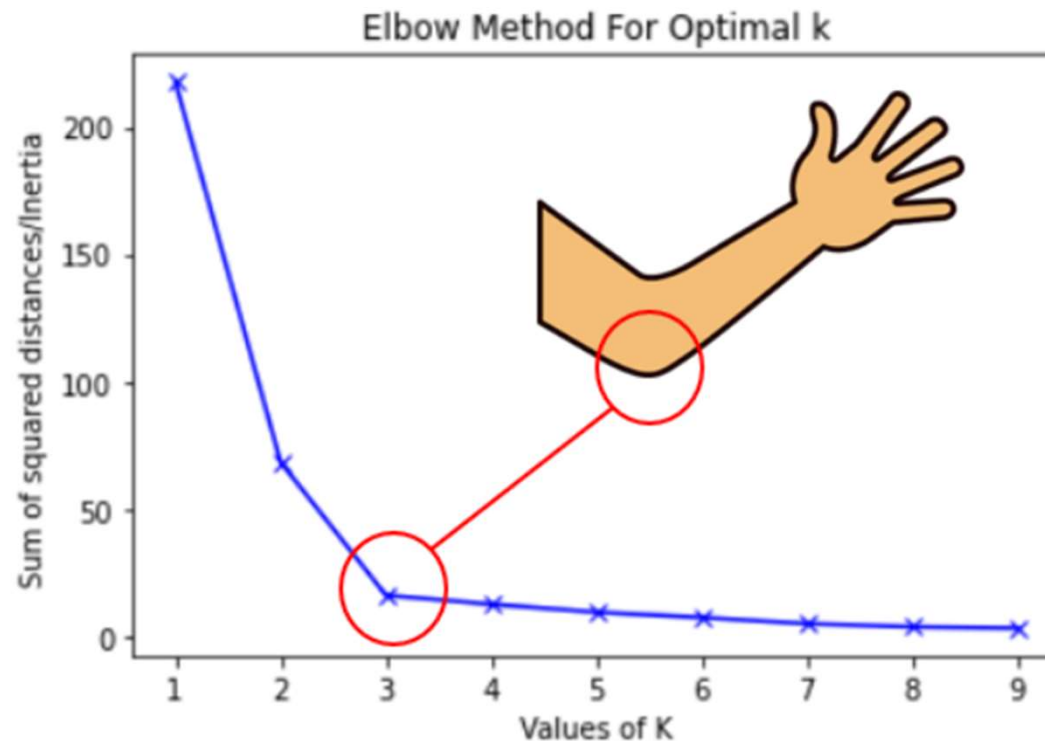




# K-Means

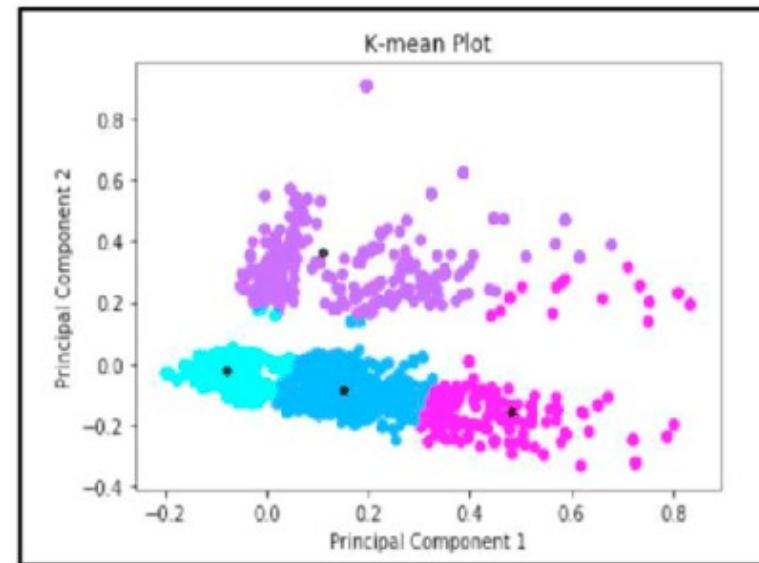
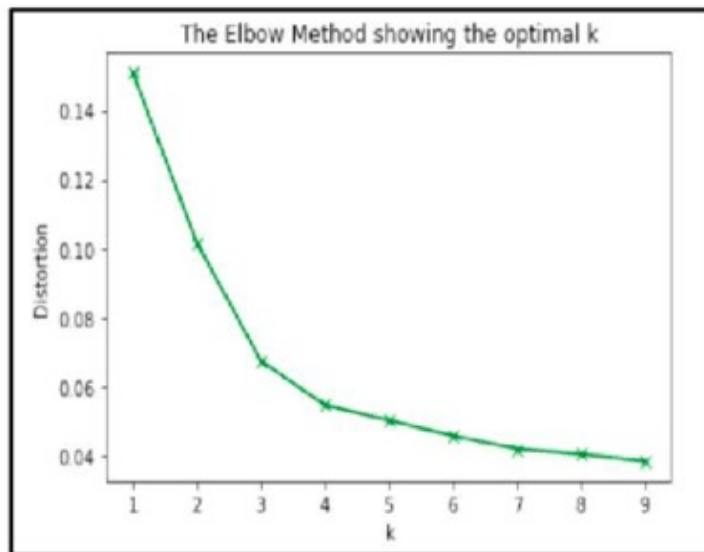
- Merupakan algoritma clustering yang cukup efisien dan cepat dengan  **$O(tkn)$** .
  - **$n$** : jumlah data
  - **$k$** : jumlah cluster
  - **$t$** : jumlah iterasi
- Jumlah cluster,  **$k$** , harus ditentukan di awal.
- Hasil clustering sensitive terhadap ***initial condition***. Pemilihan  **$k$**  dan posisi awal centroid awal yang berbeda bisa menghasilkan cluster yang berbeda.
- Algoritma bisa terjebak pada ***local optimum***.

# Pemilihan jumlah cluster pada K-Means

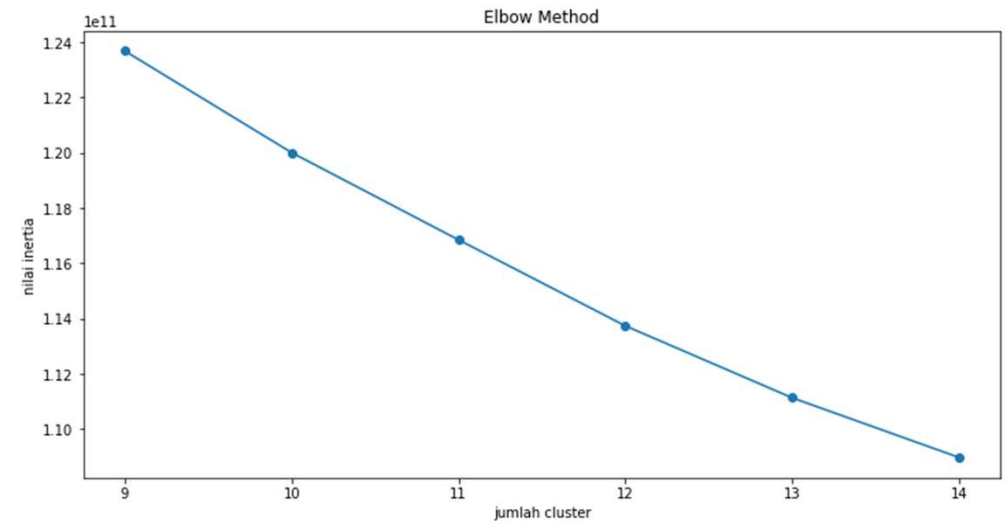
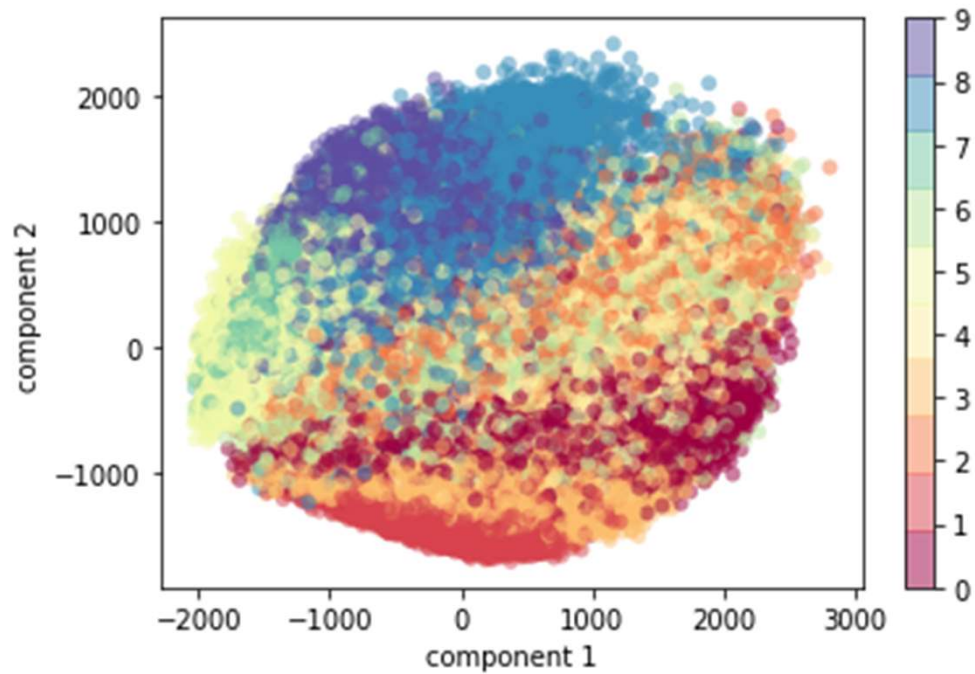


Line plot between K and inertia

# Elbow Method (Works Well)

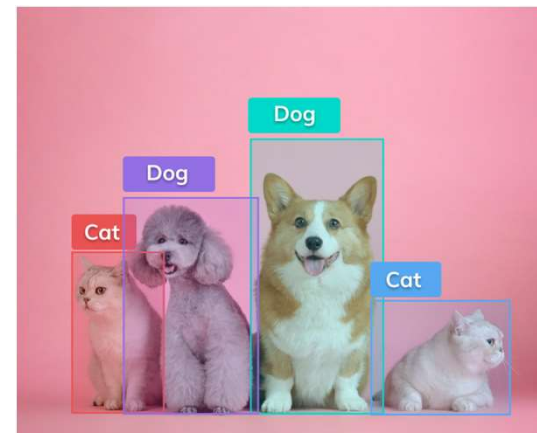
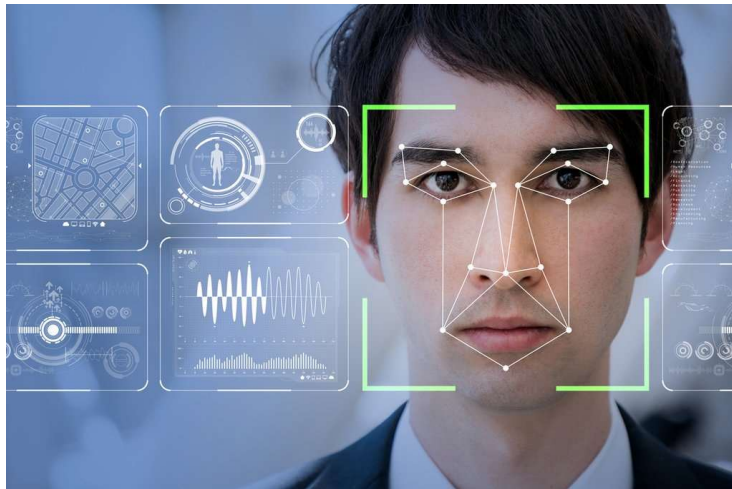


# Elbow Method (Doesn't work well)



# App of Unsupervised Learning in image processing

- Pengenalan wajah
- Deteksi objek
- Image segmentation



# Terima Kasih