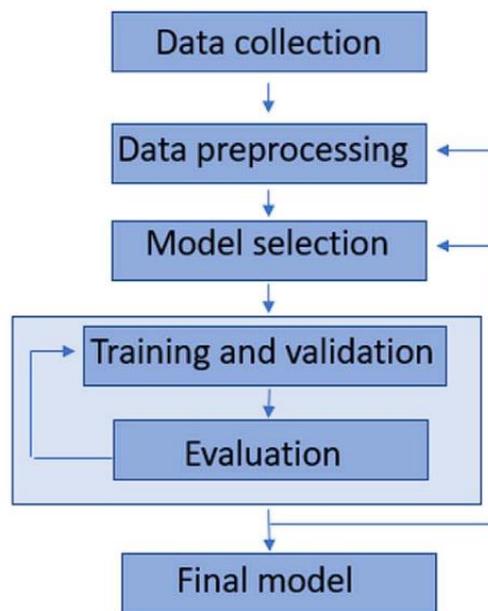


BAB III METODE PENELITIAN

3.1 Metode penelitian

Pada penelitian ini menggunakan metode Machine Learning dengan kategori *Supervised Learning*. Alur rancangan machine learning bisa dilihat pada gambar 3.1



Gambar 3. 1 Workflow Machine Learning

3.1.1 Data Collection atau Pengumpulan data

Penelitian ini menggunakan data yang sudah ada sebelumnya yang didapat dari situs Kaagle.com. Data yang didapatkan memiliki label kematangan jeruk yang mendukung untuk penelitian ini. Kematangan jeruk yang tercakup pada penelitian ini yaitu Matang, Mentah, dan Busuk. Dataset yang diambil berasal dari Fresh and Stale Images of Fruits and Vegetables (<https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables>) , dan Orange disease dataset (

<https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset>).

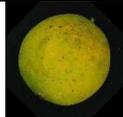
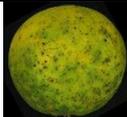
Dalam Dataset Fresh and Stale Images of Fruits and Vegetables terdapat 6 citra buah yaitu appel, banana, bitter gourd, capsium, orange, dan tomato. Dataset tersebut sudah terbagi menjadi dua kondisi buah yaitu segar dan busuk, jumlah citra dari dataset keseluruhan yaitu 14.700 citra. Karena dalam penelitian ini hanya menggunakan jeruk sebagai objeknya maka hanya data buah jeruk yang diambil, dan jumlah citra buah jeruk dari dataset Fresh and Stale Images of Fruits and Vegetables yaitu sebanyak 3.061 citra yang terbagi dari 1.466 buah segar dan 1.595 buah busuk.

Tabel 3. 1 Contoh Dataset Fresh and Stale Images of Fruits and Vegetables

| Kematangan | Contoh Citra | | | |
|------------|---|--|---|---|
| Matang |  |  |  |  |
| Busuk |  |  |  |  |

Orange disease dataset merupakan dataset yang berisikan beberapa jenis penyakit buah jeruk, jadi terbagi menjadi empat kondisi yaitu blackspot, canker, fresh dan greening. Pada dataset ini sudah terdapat label test dan train. Jumlah citra dari dataset ini yaitu 1.789 citra, dan pada label test terdapat sebanyak 624 citra dan pada label train terdapat 1.165 citra. Karena pada penelitian ini hanya memakai kondisi jeruk yang matang, mentah, dan busuk, maka dalam dataset ini hanya diambil kondisi fresh dan greening, dimana greening sendiri sebagai sample untuk citra mentah. Jadi diambil citra buah jeruk fresh sebanyak 552 citra dan greening sebanyak 545 citra.

Tabel 3. 2 Contoh Dataset Orange disease

| Kematangan | Contoh Citra | | | |
|------------|---|---|---|---|
| Matang |  |  |  |  |
| Mentah |  |  |  |  |

3.1.2 Preprocessing Data

Pada preprocessing data ini dataset yang sudah di dapatkan tadi akan dilakukan resize dan pembagian data

a. Resize Data

Karena terdapat perbedaan data pada setiap dataset yang sudah diperoleh maka akan dilakukannya resize data atau mengubah data, tujuan dari pengubahan data sendiri supaya nantinya pada saat proses reduksi data maupun klasifikasi tidak terjadi error karena perbedaan ukuran. Ukuran pixel yang akan ditetapkan nantinya pada penelitian ini yaitu 100x100 piksel, serta akan dilakukan pengubahan format data dari jpg ke png pada dataset Orange disease pada data buah jeruk mentah, dilakukannya perubahan format data ke png karena keseluruhan data yang ada pada dataset Fresh and Stale Images of Fruits and Vegetables maupun dataset Orange disease menggunakan format png kecuali data buah jeruk mentah.

b. Pembagian data

Tujuan dari pembagian data adalah untuk menyamakan jumlah dataset yang ada sesuai dengan kondisi kematangan dari dua dataset awal. Setelah dilakukannya penggabungan serta

penyamaan jumlah dataset selanjutnya membagi dataset tersebut ke beberapa model.

Pada penelitian ini dataset dibagi menjadi dua model data dengan masing – masing pembagiannya yaitu 80% data training dan 20% data testing. Dimana data training untuk melatih model, dan data testing digunakan untuk menguji data yang sudah dilatih. Jumlah data awal bisa dilihat pada tabel 3.3

Tabel 3. 3 Total dataset awal yang diperoleh

| Dataset | Kematangan | | | Total |
|---|------------|--------|-------|-------|
| | Mentah | Matang | Busuk | |
| Fresh and Stale Images of Fruits and Vegetables | - | 1466 | 1595 | 3061 |
| Orange disease | 545 | 552 | - | 1097 |
| Total | 545 | 2018 | 1595 | 4158 |

Dan pada tabel 3.4 dataset setiap kematangan di buat sama yaitu 500

Tabel 3. 4 Dataset setelah dilakukan perubahan

| Dataset | Kematangan | | | Total |
|---|------------|--------|-------|-------|
| | Mentah | Matang | Busuk | |
| Fresh and Stale Images of Fruits and Vegetables | - | 400 | 500 | 900 |

| | | | | |
|----------------|-----|-----|-----|------|
| Orange disease | 500 | 100 | - | 600 |
| Total | 500 | 500 | 500 | 1500 |

Tabel 3. 5 Pembagian model dataset

| Model Data | Dataset |
|---------------|---------|
| Data Training | 1200 |
| Data Testing | 300 |
| Total | 1500 |

c. Konversi warna RGB ke Grayscale

Grayscale adalah proses pengolahan citra untuk mengubah citra berwarna yang mempunyai nilai RGB diubah menjadi citra grayscale (keabuan). Format citra grayscale disebut skala keabuan karena pada umumnya warna yang dipakai adalah warna hitam sebagai warna minimal dan warna putih sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu (Muwardi & Fadlil, 2018).



Gambar 3. 2 Contoh konversi rgb ke grayscale

3.1.3 Model Selection

Pada Penelitian ini nantinya akan menggunakan dimension reduction dan klasifikasi model. Pemilihan Dimension reduction atau mereduksi dimensi sendiri bertujuan untuk mengurangi dimensi dataset yang ada dan nantinya akan digunakan pada saat klasifikasi.

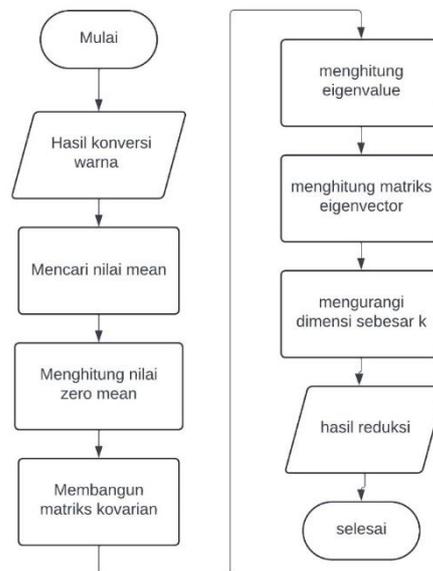
Algoritma yang dipakai untuk mereduksi data pada penelitian ini yaitu algoritma PCA.

Tujuan klasifikasi pada penelitian ini sendiri yaitu untuk menentukan tingkat kematangan buah jeruk berdasarkan warna kulit buah. Algoritma yang dipakai dalam proses klasifikasi sendiri meliputi dua algoritma yaitu CNN dan SVM.

3.1.4 Training and Validation

a. Reduksi Data menggunakan PCA

Metode PCA akan membentuk sekumpulan dimensi baru yang kemudian di ranking berdasarkan varian datanya. PCA akan menghasilkan Principal component yang didapat dari dekomposisi eigen value dan eigen vektor dari matriks kovariansi.



Gambar 3. 3 Flowchart alur proses PCA

Sebagai contoh disini saya menggunakan matriks 3 x 3

$$X = \begin{matrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \\ x_{13} & x_{23} & x_{33} \end{matrix}$$

$$X = \begin{matrix} 2 & 4 & 1 \\ 1 & 3 & 2 \\ 0 & 1 & 3 \end{matrix}$$

Langkah yang pertama dilakukan ialah menentukan nilai *mean* (rata – rata) dari matriks diatas dan nilai matriks yang sudah dinormalisasi

$$\text{Mean } x_n = \frac{x_{n1} + x_{n2} + x_{n3}}{\text{jumlah kolom } x_n}$$

$$\text{Mean } x_1 = \frac{2 + 1 + 0}{3} = \frac{3}{3} = 1$$

$$\text{Mean } x_2 = \frac{4 + 3 + 1}{3} = \frac{8}{3} = 2,67$$

$$\text{Mean } x_3 = \frac{1 + 2 + 3}{3} = \frac{6}{3} = 2$$

Selanjutnya menentukan matriks yang sudah dinormalisasi

$$X_{\text{normalized}} = \begin{matrix} x_{11} - \text{mean}x_1 & x_{21} - \text{mean}x_2 & x_{31} - \text{mean}x_3 \\ x_{12} - \text{mean}x_1 & x_{22} - \text{mean}x_2 & x_{32} - \text{mean}x_3 \\ x_{13} - \text{mean}x_1 & x_{23} - \text{mean}x_2 & x_{33} - \text{mean}x_3 \end{matrix}$$

$$X_{\text{normalized}} = \begin{matrix} 2 - 1 & 4 - 2,67 & 1 - 2 & 1 & 1,33 & -1 \\ 1 - 1 & 3 - 2,67 & 2 - 2 & = 0 & 0,33 & 0 \\ 0 - 1 & 1 - 2,67 & 3 - 2 & -1 & -1,67 & 1 \end{matrix}$$

Selanjutnya menentukan matriks kovarian

$$\text{Cov}(X) = \frac{1}{n - 1} X^T \cdot X$$

n = jumlah data baris dalam matriks

X = matriks yang sudah dinormalisasi

X^T = transpose matriks normalisasi

$$\text{Cov}(X) = \frac{1}{3 - 1} \begin{bmatrix} 1 & 0 & -1 \\ 1,33 & 0,33 & -1,67 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1,33 & -1 \\ 0 & 0,33 & 0 \\ -1 & -1,67 & 1 \end{bmatrix}$$

$$\text{Cov}(X) = \frac{1}{2} \begin{bmatrix} 2 & 3 & -2 \\ 3 & 5,34 & -4 \\ -2 & -4 & 4 \end{bmatrix}$$

$$\text{Cov}(X) = \begin{bmatrix} 1 & 1,5 & -1 \\ 1,5 & 2,67 & -2 \\ -1 & -2 & 2 \end{bmatrix}$$

Selanjutnya menentukan nilai eigen dan vektor eigen

1. Menentukan nilai eigen

Rumus yang dipakai yaitu

$$\det(\text{Cov}(X) - \lambda I) = 0$$

Dimana

λ = nilai eigen

I = matriks identitas

$$\text{Cov}(X) - \lambda I = \begin{bmatrix} 1 - \lambda & 1,5 & -1 \\ 1,5 & 2,67 - \lambda & -2 \\ -1 & -2 & 2 - \lambda \end{bmatrix}$$

Selanjutnya menghitung determinan matriks

$$\det(\text{Cov}(X) - \lambda I) = \begin{vmatrix} 1 - \lambda & 1,5 & -1 \\ 1,5 & 2,67 - \lambda & -2 \\ -1 & -2 & 2 - \lambda \end{vmatrix}$$

untuk langkah berikutnya yaitu menghitung determinan dengan ekspansi baris pertama

$$\det = (1 - \lambda) \times \begin{vmatrix} 2,67 - \lambda & -2 \\ -2 & 2 - \lambda \end{vmatrix} - 1,5 \times \begin{vmatrix} 1,5 & -2 \\ -1 & 2 - \lambda \end{vmatrix} - (-1) \times \begin{vmatrix} 1,5 & 2,67 - \lambda \\ -1 & -2 \end{vmatrix}$$

Perhitungan sub determinan pertama

$$\begin{aligned} \begin{vmatrix} 2,67 - \lambda & -2 \\ -2 & 2 - \lambda \end{vmatrix} &= (2,67 - \lambda)(2 - \lambda) - (-2)(-2) \\ &= (2,67 - \lambda)(2 - \lambda) - 4 \\ &= 5,34 - 2,67\lambda - 2\lambda + \lambda^2 - 4 \\ &= \lambda^2 - 4,67\lambda + 1,34 \end{aligned}$$

Perhitungan submatriks kedua

$$\begin{aligned} \begin{vmatrix} 1,5 & -2 \\ -1 & 2 - \lambda \end{vmatrix} &= (1,5)(2 - \lambda) - (-2)(-1) \\ &= 3 - 1,5\lambda - 2 = -1,5\lambda + 1 \end{aligned}$$

Perhitungan submatriks ketiga

$$\begin{aligned} \begin{vmatrix} 1,5 & 2,67 - \lambda \\ -1 & -2 \end{vmatrix} &= (1,5)(-2) - (2,67 - \lambda)(-1) \\ &= -3 - 2,67 + \lambda = \lambda - 5,67 \end{aligned}$$

Setelah semua submatriks diselesaikan selanjutnya substitusi kembali ke determinan

$$\det = (1 - \lambda)(\lambda^2 - 4,67\lambda + 1,34) - 1,5(-1,5\lambda + 1) + (\lambda - 5,67)$$

Bagian pertama

$$\begin{aligned} (1 - \lambda)(\lambda^2 - 4,67\lambda + 1,34) &= \lambda^2 - 4,67\lambda + 1,34 - \lambda^3 + 4,67\lambda^2 - 1,34\lambda \\ &= -\lambda^3 + 5,67\lambda^2 - 6,01\lambda + 1,34 \end{aligned}$$

Bagian kedua

$$-1,5(-1,5 + 1) = 2,25\lambda - 1,5$$

Bagian ketiga

$$\lambda - 5,67$$

Setelah semua bagian telah diselesaikan maka selanjutnya dilakukan penggabungan

$$\det = -\lambda^3 + 5,67\lambda^2 - 6,01\lambda + 1,34 + 2,25\lambda - 1,5 + \lambda - 5,67$$

$$\det = -\lambda^3 + 5,67\lambda^2 - 2,76\lambda - 5,83$$

Maka didapatkan persamaan berikut

$$-\lambda^3 + 5,67\lambda^2 - 2,76\lambda - 5,83 = 0$$

Karena tidak memungkinkan untuk menyelesaikan persamaan ini secara manual maka dilakukan menggunakan alat bantu dan didapatkan hasil nilai eigen seperti berikut

$$\lambda_1 = 4,73 \quad \lambda_2 = 0,77 \quad \lambda_3 = 0,17$$

2. Menentukan vektor eigen

Untuk menentukan vektor eigen menggunakan rumus berikut

$$(\text{Cov}(x) - \lambda I)v = 0$$

Kemudian masukkan nilai eigen yang sudah didapatkan tadi dan didapatkan hasil vektor eigen

$$v_1 = \begin{bmatrix} 0,49 \\ 0,73 \\ -0,47 \end{bmatrix}, v_2 = \begin{bmatrix} -0,67 \\ 0,01 \\ 0,74 \end{bmatrix}, v_3 = \begin{bmatrix} 0,56 \\ -0,67 \\ -0,49 \end{bmatrix}$$

3. Proyeksi data ke komponen utama

Setelah vektor eigen didapatkan maka selanjutnya dimasukkan ke komponen utama dan berikut prosesnya

Matriks vektor eigen yang didapatkan

$$V = \begin{bmatrix} 0,49 & -0,67 & 0,56 \\ 0,73 & 0,01 & -0,67 \\ -0,47 & 0,74 & -0,49 \end{bmatrix}$$

Untuk memasukan hasil yang sudah didapatkan ke komponen utama maka menggunakan rumus berikut

$$X_{pca} = X_{normalisasi} \times V$$

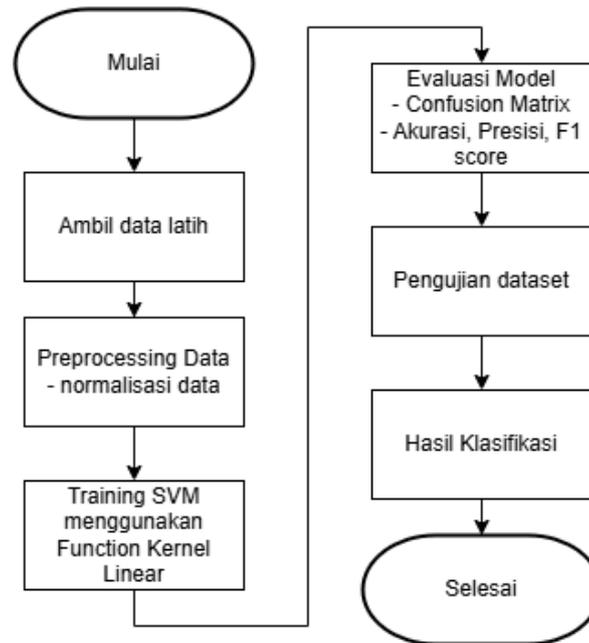
$$X_{pca} = \begin{bmatrix} 1 & 1,33 & -1 \\ 0 & 0,33 & 0 \\ -1 & -1,67 & 1 \end{bmatrix} \times \begin{bmatrix} 0,49 & -0,67 & 0,56 \\ 0,73 & 0,01 & -0,67 \\ -0,47 & 0,74 & -0,49 \end{bmatrix}$$

Maka didapatkan hasil akhir

$$X_{pca} = \begin{bmatrix} 1,66 & -0,29 & -0,16 \\ 0,24 & 0,23 & -0,06 \\ -1,90 & 0,06 & 0,22 \end{bmatrix}$$

b. Klasifikasi Pada Algoritma SVM

Setelah dilakukannya pengelompokan data serta reduksi data maka selanjutnya akan dilakukan klasifikasi pada algoritma SVM, untuk alur program bisa dilihat pada gambar 3.4.



Gambar 3. 4 Flowchart Alur Algoritma SVM

Untuk perhitungan manual berikut penjabarannya

Disini misalkan menggunakan data matriks dari hasil PCA sebelumnya yaitu

$$X_{pca} = \begin{bmatrix} 1,66 & -0,29 & -0,16 \\ 0,24 & 0,23 & -0,06 \\ -1,90 & 0,06 & 0,22 \end{bmatrix}$$

Karena pada data sebelumnya tidak ada label (y) yaitu data target untuk x, maka disini untuk y dimisalkan sebagai

$$y = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Pada algoritma SVM ini menggunakan rumus

$$f(x) = w^T x + b$$

Dimana

w = vektor yang menentukan orientasi hyperlane

b = bias (konstanta) yang menentukan posisi hyperlane

x = vektor data input

dan harus memenuhi syarat $y_i(w^T x_i + b) \geq 1$, dimana margin tidak boleh kurang dari 1

1. Normalisasi hyperlane

Untuk normalisasi hyperlane disini menggunakan rumus

$$y_i(w^T x_i + b)$$

Dengan w dan b sendiri disini misal $w = (1, 1, 1)$ dan $b = 0$

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1 * 1,66) + (1 * -0,29) + (1 * -0,16) + 0)$$

$$f(x_1) = 1,66 - 0,29 - 0,16 = 1,21$$

Margin :

$$y_1 * f(x_1) = 1 * 1,21 = 1,21$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_2 + b$$

$$f(x_2) = (1 * 0,24) + (1 * 0,23) + (1 * -0,06) + 0)$$

$$f(x_2) = 0,24 + 0,23 - 0,06 = 0,41$$

Margin

$$y_2 * f(x_2) = 1 * 0,41 = 0,41$$

Margin data ke-2 tidak memenuhi syarat karena kurang dari 1

Data 3

$$(x_3 = [-1,90, 0,06, 0,22], y_3 = -1)$$

$$f(x_3) = w^T x_3 + b$$

$$f(x_3) = (1 * -1,90) + (1 * 0,06) + (1 * 0,22) + 0)$$

$$f(x_3) = -1,90 + 0,06 + 0,22 = -1,62$$

Margin

$$y_3 * f(x_3) = -1 * -1,62 = 1,62$$

Margin ke-3 memenuhi syarat dimana hasil margin lebih dari 1

Karena data ke-2 tidak memenuhi syarat maka akan dilakukan perbaikan pada bobot (w) dan bias (b) dengan menggunakan rumus

$$w = w + n * (y_i * x_i)$$

$$b = b + n * y_i$$

Dimana

b = bias

w = bobot

n = learning rate

x_i = data input i

y_i = label kelas dari data input

Pembaruan pada bobot (w)

Pada pembaruan pertama ini untuk w sendiri dimisalkan menggunakan nilai (1,1,1)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,1,1] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,1,1] + [0,024, 0,023, -0,006]$$

$$w = [1,24 ,1,023 ,0,994]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0 + 0,1 * 1 = 0,1$$

Setelah didapatkan untuk pembaruan w dan b maka selanjutnya dihitung kembali seperti pada awal

Dengan w dan b sendiri disini misal w = (1,24 , 1,023, 0,994) dan b = 0,1

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,24 * 1,66) + (1,023 * -0,29) + (0,994 * -0,16) + 0,1)$$

$$f(x_1) = 1,7 - 0,297 - 0,159 + 0,1 = 1,344$$

Margin :

$$y_1 * f(x_1) = 1 * 1,344 = 1,344$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_1 + b$$

$$f(x_2) = (1,24 * 0,24) + (1,023 * 0,23) + (0,994 * -0,06) + 0,1)$$

$$f(x_2) = 0,246 + 0,235 - 0,060 + 0,1 = 0,521$$

Margin :

$$y_2 * f(x_2) = 1 * 0,521 = 0,521$$

Untuk data ke-2 masih belum memenuhi syarat

$$y_i(w^T x_i + b) \geq 1$$

Karena data ke-2 belum memenuhi syarat maka akan dilakukan pembaruan lagi pada w dan b kembali sampai data ke-2 memenuhi syarat

Pembaruan pada bobot (w) ke-2

Pada pembaruan kali ini untuk w menggunakan nilai sebelumnya yaitu (1,24 ,1,023 ,0,994)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,24 ,1,023 ,0,994] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,24 ,1,023 ,0,994] + [0,024, 0,023, -0,006]$$

$$w = [1,048, 1,046, 0,988]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0,1 + 0,1 * 1 = 0,2$$

Setelah didapatkan untuk pembaruan w dan b yang ke-2 maka selanjutnya dihitung kembali seperti pada awal

Dengan w dan b sendiri disini w = (1.048,1.046,0.988) dan b = 0,2

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,048 * 1,66) + (1,046 * -0,29) + (0,988 * -0,16) + 0,2)$$

$$f(x_1) = 1,740 - 0,303 - 0,158 + 0,2 = 1,479$$

Margin :

$$y_1 * f(x_1) = 1 * 1,479 = 1,479$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_1 + b$$

$$f(x_2) = (1,048 * 0,24) + (1,046 * 0,23) + (0,988 * -0,06) + 0,2)$$

$$f(x_2) = 0,251 + 0,240 - 0,059 + 0,2 = 0,632$$

Margin :

$$y_2 * f(x_2) = 1 * 0,632 = 0,632$$

Data ke-2 masih belum memenuhi syarat $y_i(w^T x_i + b) \geq 1$

Maka akan dilakukan pembaruan untuk w dan b kembali sampai data ke-2 tersebut memenuhi syarat

Pembaruan pada bobot (w) ke-3

Pada pembaruan kali ini untuk w menggunakan nilai sebelumnya yaitu (1,048,1,046,0,988)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,048,1,046,0,988] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,048,1,046,0,988] + [0,024, 0,023, -0,006]$$

$$w = [1,072, 1,069, 0,982]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0,2 + 0,1 * 1 = 0,3$$

Hitung kembali Dengan w dan b sendiri disini $w = (1,072, 1,069, 0,982)$ dan $b = 0,3$

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,072 * 1,66) + (1,069 * -0,29) + (0,982 * -0,16) + 0,3)$$

$$f(x_1) = 1,779 - 0,310 - 0,157 + 0,3 = 1,612$$

Margin :

$$y_1 * f(x_1) = 1 * 1,6129 = 1,612$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_1 + b$$

$$f(x_2) = (1,072 * 0,24) + (1,069 * 0,23) + (0,982 * -0,06) + 0,3)$$

$$f(x_2) = 0,257 + 0,246 - 0,059 + 0,3 = 0,744$$

Margin :

$$y_2 * f(x_2) = 1 * 0,744 = 0,744$$

Data ke-2 masih belum memenuhi syarat $y_i(w^T x_i + b) \geq 1$

Dilakukan pembaruan kembali untuk w dan b

Pembaruan pada bobot (w) ke-3

Pada pembaruan kali ini untuk w menggunakan nilai sebelumnya yaitu (1,072, 1,069, 0,982)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,072, 1,069, 0,982] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,072, 1,069, 0,982] + [0,024, 0,023, -0,006]$$

$$w = [1,096, 1,092, 0,976]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0,3 + 0,1 * 1 = 0,4$$

Selanjutnya dihitung kembali

Dengan w dan b sendiri disini $w = (1,096, 1,092, 0,976)$ dan $b = 0,4$

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,096 * 1,66) + (1,092 * -0,29) + (0,976 * -0,16) + 0,4)$$

$$f(x_1) = 1,819 - 0,317 - 0,156 + 0,4 = 1,746$$

Margin :

$$y_1 * f(x_1) = 1 * 1,746 = 1,746$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_2 + b$$

$$f(x_2) = (1,096 * 0,24) + (1,092 * 0,23) + (0,976 * -0,06) + 0,4$$

$$f(x_2) = 0,263 + 0,251 - 0,059 + 0,4 = 0,855$$

Margin :

$$y_2 * f(x_2) = 1 * 0,855 = 0,855$$

Data ke-2 masih belum memenuhi syarat $y_i(w^T x_i + b) \geq 1$

Dilakukan pembaruan kembali untuk w dan b

Pembaruan pada bobot (w) ke-4

Pada pembaruan kali ini untuk w menggunakan nilai sebelumnya yaitu (1,096,1,092,0,976)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,096, 1,092, 0,976] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,096, 1,092, 0,976] + [0,024, 0,023, -0,006]$$

$$w = [1,120, 1,115, 0,970]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0,4 + 0,1 * 1 = 0,5$$

Selanjutnya dihitung kembali

Dengan w dan b sendiri disini $w = (1,120,1,115,0,970)$ dan $b = 0,5$

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,120 * 1,66) + (1,115 * -0,29) + (0,970 * -0,16) + 0,5$$

$$f(x_1) = 1,859 - 0,324 - 0,155 + 0,5 = 1,880$$

Margin :

$$y_1 * f(x_1) = 1 * 1,880 = 1,880$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_2 + b$$

$$f(x_2) = (1,120 * 0,24) + (1,115 * 0,23) + (0,970 * -0,06) + 0,5)$$

$$f(x_2) = 0,269 + 0,257 - 0,058 + 0,5 = 0,968$$

Margin :

$$y_2 * f(x_2) = 1 * 0,968 = 0,968$$

Data ke-2 masih belum memenuhi syarat $y_i(w^T x_i + b) \geq 1$

Dilakukan pembaruan kembali untuk w dan b

Pembaruan pada bobot (w) ke-5

Pada pembaruan kali ini untuk w menggunakan nilai sebelumnya yaitu (1,120, 1,115, 0,970)

$$w = w + n * (y_2 * x_2)$$

$$w = [1,120, 1,115, 0,970] + 0,1 * (1 * [0,24, 0,23, -0,06])$$

$$w = [1,120, 1,115, 0,970] + [0,024, 0,023, -0,006]$$

$$w = [1,144, 1,138, 0,964]$$

Pembaruan b

$$b = b + n * y_2$$

$$b = 0,5 + 0,1 * 1 = 0,6$$

Selanjutnya dihitung kembali

Dengan w dan b sendiri disini $w = (1,144, 1,138, 0,964)$ dan $b = 0,6$

Data 1

$$(x_1 = [1,66, -0,29, -0,16], y_1 = 1)$$

$$f(x_1) = w^T x_1 + b$$

$$f(x_1) = (1,144 * 1,66) + (1,138 * -0,29) + (0,964 * -0,16) + 0,6)$$

$$f(x_1) = 1,899 - 0,330 - 0,154 + 0,6 = 2,015$$

Margin :

$$y_1 * f(x_1) = 1 * 2,015 = 2,015$$

Memenuhi syarat karena $y_i(w^T x_i + b) \geq 1$

Data 2

$$(x_2 = [0,24, 0,23, -0,06], y_2 = 1)$$

$$f(x_2) = w^T x_2 + b$$

$$f(x_2) = (1,144 * 0,24) + (1,138 * 0,23) + (0,964 * -0,06) + 0,6$$

$$f(x_2) = 0,274 + 0,262 - 0,058 + 0,6 = 1,078$$

Margin :

$$y_2 * f(x_2) = 1 * 1,078 = 1,078$$

Data ke-2 memenuhi syarat $y_i(w^T x_i + b) \geq 1$

Data 3

$$(x_3 = [-1,90, 0,06, 0,22], y_3 = -1)$$

$$f(x_3) = w^T x_3 + b$$

$$f(x_3) = (1,144 * -1,90) + (1,138 * 0,06) + (0,9964 * 0,22) + 0,6$$

$$f(x_3) = -2,174 + 0,068 + 0,221 + 0,6 = -1,294$$

Margin

$$y_3 * f(x_3) = -1 * -1,294 = 1,294$$

Data ke-3 memenuhi syarat $y_i(w^T x_i + b) \geq 1$

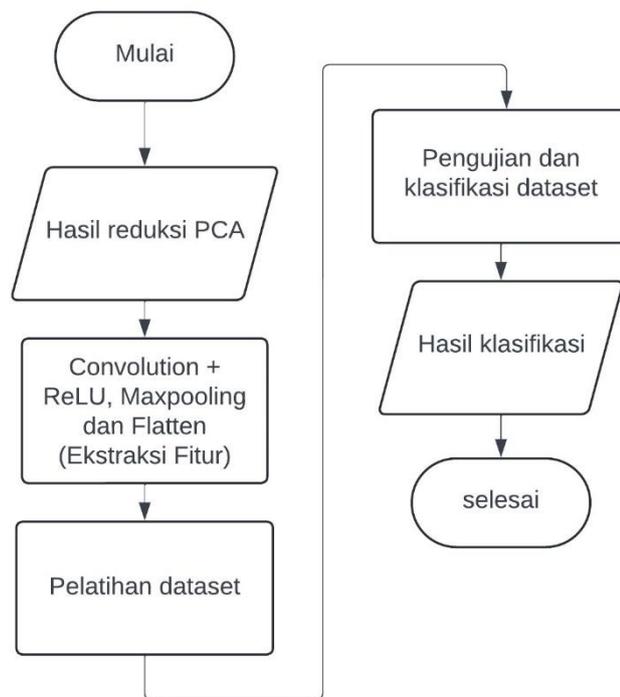
Maka output akhir yang didapat adalah

$$w = [1,144, 1,138, 0,964]$$

$$b = 0,6$$

c. Klasifikasi pada algoritma CNN

Setelah didapatkan hasil reduksi data dari algoritma PCA selanjutnya dilakukan klasifikasi CNN, alur pengujian dataset menggunakan algoritma CNN bisa dilihat pada gambar 3.4 berikut:



Gambar 3. 5 Flowchart proses klasifikasi pada algoritma CNN

Untuk perhitungan secara manualnya sebagai berikut

Untuk matriks yang dipakai yaitu hasil dari reduksi PCA sebelumnya yaitu

$$X_{pca} = \begin{bmatrix} 1,66 & -0,29 & -0,16 \\ 0,24 & 0,23 & -0,06 \\ -1,90 & 0,06 & 0,22 \end{bmatrix}$$

1. Proses Convolution

Karena ukuran kernel yang dihasilkan yaitu 3x3 maka untuk filter kernel yang digunakan yaitu ukuran kernel = ukuran input – ukuran kernel + 1, ukuran kernel = 3 – 2 + 1 = 2. Maka kernel filter yang digunakan yaitu 2x2 dengan misal isi dari kernel fitur sebagai berikut

$$K = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Setelah kernel filter dilakukan maka proses selanjutnya yaitu menggeser kernel k mengelilingi matriks x yang dimana elemen yang berada pada posisi yang sama dengan kernel fitur akan dikalikan

Untuk area pertama yaitu

$$\begin{bmatrix} 1,66 & -0,29 \\ 0,24 & 0,23 \end{bmatrix}$$

$$\begin{aligned} C11 &= (1 \times 1,66) + (0 \times -0,29) + (0 \times 0,24) + (-1 \times 0,23) \\ &= 1,66 - 0,23 = 1,43 \end{aligned}$$

Area kedua

$$\begin{bmatrix} -0,29 & -0,16 \\ 0,23 & -0,06 \end{bmatrix}$$

$$\begin{aligned} C12 &= (1 \times -0,29) + (0 \times -0,16) + (0 \times 0,23) + (-1 \times -0,06) \\ &= -0,29 + 0,06 = -0,23 \end{aligned}$$

Area ketiga

$$\begin{bmatrix} 0,24 & 0,23 \\ -1,90 & 0,06 \end{bmatrix}$$

$$\begin{aligned} C21 &= (1 \times 0,24) + (0 \times 0,23) + (0 \times -1,90) + (-1 \times 0,06) \\ &= 0,24 - 0,06 = 0,18 \end{aligned}$$

Area keempat

$$\begin{bmatrix} 0,23 & -0,06 \\ 0,06 & 0,22 \end{bmatrix}$$

$$\begin{aligned} C22 &= (1 \times 0,23) + (0 \times -0,06) + (0 \times 0,06) + (-1 \times 0,22) \\ &= 0,23 - 0,22 = 0,01 \end{aligned}$$

Maka hasil dari proses convolution adalah

$$C = \begin{bmatrix} 1,43 & -0,23 \\ 0,18 & 0,01 \end{bmatrix}$$

2. Relu layer

Relu layer sendiri berfungsi untuk mengubah nilai negatif menjadi 0 pada sebuah matriks.

$$C = \begin{bmatrix} 1,43 & -0,23 \\ 0,18 & 0,01 \end{bmatrix}$$

Jika pada proses convolution sebelumnya terdapat angka minus maka akan digantikan 0 dan berikut hasilnya

$$C = \begin{bmatrix} 1,43 & 0 \\ 0,18 & 0,01 \end{bmatrix}$$

3. Pooling layer

Proses yang selanjutnya dilalui yaitu pooling layer dimana pooling layer berfungsi untuk mereduksi dimensi matriks dengan cara mengambil nilai maksimum dalam suatu matriks.

$$\begin{bmatrix} 1,43 & 0 \\ 0,18 & 0,01 \end{bmatrix}$$

Dengan mengambil hasil convolution tadi maka hasil yang didapatkan yaitu

$$P = [1,43]$$

4. Fully connected

Dari hasil pooling layer sebelumnya yaitu [1,43] akan menjadikan inputan untuk proses selanjutnya yaitu fully connected. Dengan menggunakan rumus berikut

$$y = W + b$$

Dimana

Y = fully connected

W = bobot nilai

b = bias

misalkan $W = 0,5$ dan $b = 0,2$

maka $y = (0,5 * 1,43) + 0,2 = 0,715 + 0,2 = 0,915$

5. Softmax

Langkah berikutnya yaitu softmax dimana proses ini digunakan untuk menghitung probabilitas. Karena pada proses sebelumnya. Misalkan sudah didapatkan kelas dengan skor akhir

$$z1 = 0,915 \text{ dan } z2 = 0,5$$

Maka selanjutnya menghitung probabilitas dengan menggunakan rumus berikut

$$P(\text{kelas 1}) = \frac{e^{z_1}}{e^{z_1} + e^{z_2}}$$

$$P(\text{kelas 2}) = \frac{e^{z_2}}{e^{z_1} + e^{z_2}}$$

Dimana

P = probabilitas prediksi pada suatu kelas

e = nilai eksponensial (2,718)

Jika dimasukkan kenilai yang sudah ada menjadi

$$P(\text{Kelas1}) = \frac{2,718^{0,915}}{2,718^{0,915} + 2,718^{0,5}}$$

$$P(\text{Kelas2}) = \frac{2,718^{0,5}}{2,718^{0,915} + 2,718^{0,5}}$$

$$P(\text{Kelas1}) = \frac{2,5}{2,5 + 1,65} = 0,60$$

$$P(\text{Kelas2}) = \frac{1,65}{2,5 + 1,65} = 0,40$$

3.1.5 Evaluation

Setelah proses proses klasifikasi selesai nantinya akan dilakukan pengujian model dari data yang dihasilkan sebelumnya, pengujian ini nantinya menggunakan Confusion matrix akurasi sebagai acuan perbandingan. Dimana didalamnya terdapat nilai True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Dimana TP merupakan data positif yang diprediksi benar, TN merupakan data negatif yang diprediksi benar, FP merupakan kondisi dimana data negatif atau data yang salah diprediksi positif atau benar, dan FN merupakan kondisi dimana data tersebut benar namun diprediksi salah (Nugroho, 2019). Selain Confusion matrix ada juga bahan evaluasi lain seperti Akurasi, Presisi, Recall, dan F1-Score untuk rumusnya sebagai berikut:

a. Akurasi

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

b. Presisi

$$Precision = \frac{TP}{TP + FP}$$

c. Recall

$$Recall = \frac{TP}{TP + FN}$$

d. F1-Score

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$