

PEMBELAJARAN MESIN

TUGAS AKHIR

CORP RECOMMENDATION USING CLASIFICATION



Disusun oleh:

Atmayanti (20142720016)

Ahmad Rafif Alauddin (2041720230)

Rosi Latansa Salsabela (2041720026)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota
Malang, Jawa Timur 65141

DAFTAR ISI

DAFTAR ISI.....	2
ABSTRAK	3
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
BAB II METODOLOGI	6
2.1 <i>Decision Tree</i>	6
2.2 <i>Naïve Bayes</i>	8
2.3 <i>Support Vector Machine</i>	9
2.4 <i>Random Forest</i>	12
BAB III HASIL DAN PEMBAHASAN	13
3.1 Pengumpulan dan Perencanaan Dataset	13
3.2 Perhitungan Manual	15
3.3 Pengimplementasian Pada Aplikasi	28
BAB IV KESIMPULAN.....	36
DAFTAR PUSTAKA.....	37

ABSTRAK

Salah tanam merupakan salah satu faktor yang menyebabkan gagal panen sehingga petani mengalami kerugian. Salah tanam terjadi karena petani tidak memperhatikan lingkungan dimana tanaman tersebut ditanam, contohnya seperti padi yang tidak akan tumbuh maksimal ketika ditanam pada musim kemarau yang memiliki curah hujan yang rendah, sedangkan padi merupakan tanaman yang tumbuh membutuhkan banyak air. Selain padi, tanaman lain juga memiliki kebutuhan khusus untuk dapat tumbuh dengan maksimal. penelitian ini dilakukan untuk membuat aplikasi berbasis machine learning yang menggunakan teknik klasifikasi untuk dapat menentukan atau memberi rekomendasi tanaman yang cocok untuk ditanam pada lingkungan tertentu dengan memperhatikan faktor tingkat curah hujan, tingkat kelembapan, tingkat ph tanah, suhu, serta komposisi unsur makro pada tanah, yakni N, P, dan K. Pada penelitian ini akan membandingkan empat metode klasifikasi, yakni Decission Tree, Random Forest, Suport Vector Machine, serta Gaussain Naive Bayes untuk dapat menentukan metode paling efektif. Setelah melakukan analisis dan melakukan perbandingan terhadap nilai akurasi yang dihasilkan dari hasil melakukan training pada tiap-tiap model, dapat disimpulkan bahwa metode Random Forest dan Gaussain Naive Bayes merupakan metode yang paling cocok untuk diterapkan pada aplikasi karena sama-sama memiliki nilai akurasi tertinggi.

Kata Kunci : *Machine Learning, Klasifikasi, Pertanian.*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara agraris yang memiliki lahan pertanian yang luas, sumber daya alam beraneka ragam dan berlimpah sehingga tidak heran sebagian besar penduduknya berprofesi sebagai petani. Sebagai negara agraris dengan sebagian penduduk berprofesi sebagai petani, nyatanya banyak petani di Indonesia masih belum sejahtera (Anon 2022). Adapun berbagai alasan mengapa petani di Indonesia belum sejahtera, salah satu penyebabnya adalah terjadinya gagal panen. Gagal panen merupakan kondisi yang terjadi dimana petani menerima kerugian akibat tanaman yang mereka tanam tidak dapat dipanen. Apabila gagal panen ini berulang terjadi maka hal inilah yang dapat menyebabkan petani tidak bisa sejahtera, untuk itu petani perlu mengetahui penyebab dari gagal panen agar hal tersebut tidak terjadi.

Dwikorita Karnawati, Kepala BMKG, pernah mengatakan bahwa salah satu penyebab terjadinya gagal panen adalah petani yang tidak memperhatikan cuaca dalam menentukan tanaman yang akan ditanam atau bisa dikatakan salah tanam (Media 2020). Padi yang seharusnya ditanam di musim penghujan akan menjadi tidak maksimal ketika ditanam pada musim kemarau, karena padi merupakan tanaman yang membutuhkan banyak air.

Selain padi yang butuh banyak air agar dapat tumbuh, terdapat tanaman-tanaman lain yang memiliki kondisi yang khusus agar dapat tumbuh dengan maksimal. Maka dari itu petani perlu memahami kondisi lingkungan terlebih dahulu sebelum menentukan tanaman apa yang ingin ditanam. Adapun beberapa faktor yang menjadi penentu tanaman apa yang cocok untuk di tanam, seperti tingkat curah hujan, suhu, kelembaban, pH tanah, serta tingkat unsur makro tanah, yaitu N, P, dan K. Untuk dapat menentukan tanaman apa yang cocok ditanam berdasarkan kriteria tersebut dengan mudah, maka diperlukanlah sebuah aplikasi yang dapat melakukan prediksi berdasarkan beberapa faktor diatas sehingga akan mengeluarkan output berupa rekomendasi nama tanaman yang cocok untuk ditanam.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka perumusan masalah yang muncul dipaparkan sebagai berikut :

1.2.1 Bagaimana aplikasi yang dikembangkan dapat menentukan rekomendasi tanaman yang cocok pada suatu lingkungan dengan kondisi tertentu?

1.2.2 Bagaimana cara menentukan metode yang paling cocok diterapkan untuk menyelesaikan permasalahan pada penelitian ini?

1.3 Tujuan

Tujuan dari penelitian ini disajikan dalam poin-poin berikut :

1.3.1 menentukan rekomendasi tanaman yang cocok untuk ditanam pada suatu lingkungan dengan kondisi tertentu menggunakan klasifikasi.

1.3.2 menentukan metode klasifikasi yang paling efektif digunakan untuk memecahkan permasalahan pada penelitian ini.

1.4 Manfaat Penelitian

1.4.1 Manfaat penelitian bagi peneliti

Penelitian ini bermanfaat untuk mengimplementasikan dan mengembangkan ilmu pengetahuan peneliti sesuai bidang yang diampuh dengan cara membuat aplikasi yang dapat menentukan rekomendasi tanaman yang cocok untuk ditanam pada lingkungan dengan kondisi tertentu menggunakan dengan memanfaatkan teknologi machine learning dengan menerapkan teknik klasifikasi. Selain itu dapat menambah wawasan peneliti mengenai bidang machine learning dengan mengetahui mekanisme penerapan berbagai metode dalam teknik pengklasifikasian dalam menentukan rekomendasi tanaman.

1.4.2 Manfaat penelitian bagi masyarakat

Penelitian ini bermanfaat bagi masyarakat, khususnya para petani untuk dapat mencegah terjadinya kerugian akibat gagal panen yang disebabkan karena salah menentukan tanaman yang akan ditanam berdasarkan kondisi lingkungan tanamannya tersebut ditanam.

BAB II METODOLOGI

2.1 *Decision Tree*

Decision Tree adalah sebuah diagram alir yang berbentuk seperti struktur pohon yang mana *setiap internal node* menyatakan pengujian terhadap suatu atribut, *setiap cabang* menyatakan *output* dari pengujian tersebut dan *leaf node* menyatakan kelas-kelas atau distribusi kelas. *Node* yang paling atas disebut sebagai *root node* atau *node* akar. Sebuah *root node* akan memiliki beberapa *edge* keluar tetapi tidak memiliki *edge* masuk, *internal node* akan memiliki satu *edge* masuk dan beberapa *edge* keluar, sedangkan *leaf node* hanya akan memiliki satu *edge* masuk tanpa memiliki *edge* keluar. *Decision Tree* digunakan untuk mengklasifikasikan suatu sampel data yang belum diketahui kelasnya ke dalam kelas-kelas yang sudah ada (Qadrini, et al., 2021).

Ada banyak algoritma pada klasifikasi *Decision Tree* ini. Suatu algoritma biasanya dikembangkan untuk meningkatkan kinerja algoritma yang sudah ada. Penentuan algoritma yang terbaik dalam *Decision Tree* tentunya tidak bisa ditentukan secara mutlak tetapi sangat tergantung dengan karakteristik *training set*-nya. Beberapa algoritma *Decision Tree* yang cukup populer antara lain : ID3, C4.5, dan CART (Qadrini, et al., 2021).

Pada *decision tree* pembuatan *tree* merupakan Langkah awal yang harus dilaksanakan. Dalam pembuatan *tree* diperlukan perhitungan menggunakan *Information Gain* ataupun *Gini Index* untuk menentukan *root node* / *node* selanjutnya. Berikut merupakan tahapan pembuatan *tree* secara *top-down*:

- 2.1.1 Pada iterasi pertama dilakukan pemilihan *root node* didasarkan pada *Gini Index* terendah atau *Information Gain* tertinggi dari keseluruhan fitur (*variable x*) pada data *training*.
- 2.1.2 Kemudian memisahkan himpunan *S* untuk menghasilkan subset data.
- 2.1.3 Selanjut proses akan diulang untuk mencari *decision node* yang akan menjadi cabang selanjutnya. Penentuan *decision node* juga menggunakan *Gini Index* terendah atau *Information Gain* tertinggi. Akan tetapi yang membedakan adalah fitur yang digunakan adalah fitur yang tidak pernah menggunakan sebelum nya (subset data).
- 2.1.4 Algoritma terus menerus berulang pada setiap subset data hingga menemukan *leaf* yang merupakan label kelas (Polinema, 2021).

Entropy digunakan untuk menghitung ketidakaturan dalam artian *entropy* akan menghitung homogenitas suatu fitur atribut (*A*) dari sebuah fitur yang ada di *sample* data (*S*). *Entropy* digunakan untuk mengukur jumlah ketidakpastian dalam *variable*. Berikut merupakan formula yang digunakan untuk mengukur *entropy* :

$$H(X) = - \sum_{i=1}^n P(X_i) \log_b P(x_i)$$

Dimana *n* merupakan berapa banyak *outcome* bisa berarti berapa banyak kelas *P (xi)*, merupakan probabilitas dari *outcome i*. *b*, dapat bernilai 2, e, atau 10. Karena *log* dari

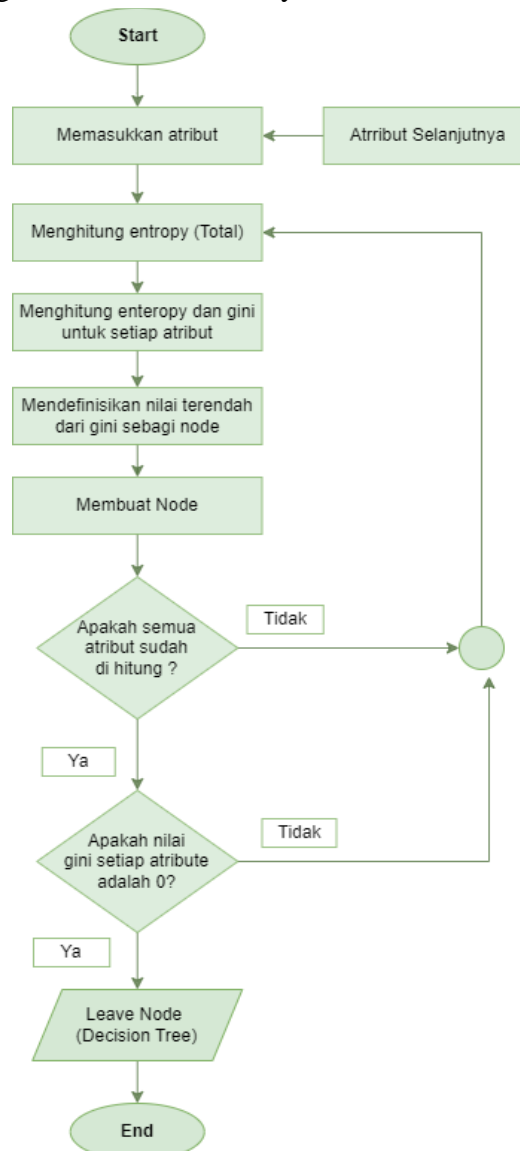
angka kurang dari 1 akan negatif, seluruh jumlah dinegasikan untuk menghasilkan nilai positif. Penjumlahan dari total *entropy* bisa lebih dari 1 (Polinema, 2021).

Gini Index Heuristik umum untuk mempelajari *decision tree* dan Mengukur proporsi kelas dalam satu set. *Gini index* dihitung dengan mengurangi jumlah probabilitas kuadrat setiap kelas. Formula *gini index* :

$$Gini(t) = 1 - \sum_{i=1}^j P(i|t)^2$$

Dimana j adalah jumlah kelas. t adalah bagian dari *instance* untuk *node*. $P(i|t)$ adalah probabilitas memilih elemen kelas i dari subset *node*. Pada *Gini index* penentuan *node* berdasarkan nilai *gini index* paling kecil, CART memilih *node* tersebut sebagai *node* keputusan, dan *node* tersebut merupakan fitur tidak pernah digunakan pada *node* sebelumnya. Proses berulang ini berhenti sampai pohon keputusan mendapatkan *leaf node* (Polinema, 2021).

Berikut flowchart alur algoritma dari Naïve Bayes :



2.2 Naïve Bayes

Naïve Bayes merupakan salah satu metode yang dapat digunakan untuk mengklasifikasikan data. *Bayesian classification* merupakan pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu *class* (Borman & Wati, 2020).

Teorema ini dikemukakan oleh ilmuwan Inggris, Thomas Bayes, dengan memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya (Gandhi, et al., 2020). Teorema Bayes dikombinasikan dengan “*Naïve*” yang berarti setiap atribut/*variable* bersifat bebas (*independent*) (Putri & Surahman, 2020).

Naïve Bayes dapat dilatih dengan efisien dalam pembelajaran terawasi (*supervised learning*). Keuntungan dari pengklasifikasi adalah bahwa hanya membutuhkan sejumlah kecil data pelatihan untuk memperkirakan para-meter (sarana dan varians dari *variable*) yang diperlukan untuk klasifikasi. Karena *variable* independen diasumsikan, hanya variasi dari *variable* untuk masing-masing kelas harus ditentukan, bukan seluruh matriks kovarians. Dalam prosesnya, *Naïve Bayes* mengasumsikan bahwa ada atau tidaknya suatu fitur pada suatu kelas tidak berhubungan dengan ada atau tidaknya fitur lain dikelas yang sama. Pada saat klasifikasi, pendekatan bayes akan menghasilkan label kategori yang paling tinggi probabilitasnya (Eko, 2012).

Persamaan teorema Bayes adalah :

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Keterangan :

$P(C_i|X)$ = Probabilitas hipotesis C_i jika diberikan fakta atau *record* X (*Posterior probability*)

$P(X|C_i)$ = mencari nilai parameter yang memberi kemungkinan yang paling besar (*likelihood*)

$P(C_i)$ = *Prior probability* dari X (*Prior probability*)

$P(X)$ = Jumlah *probability tuple* yg muncul.

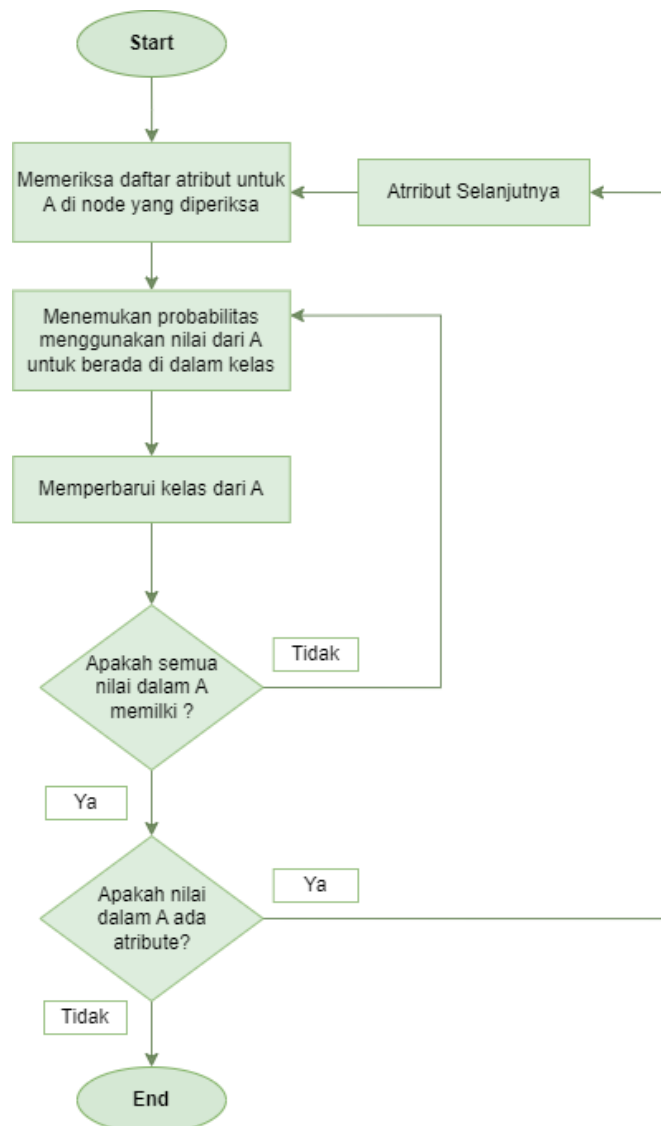
Pada perkembangannya, $P(X)$ dapat dihilangkan karena nilai tetap, sehingga saat dibandingkan dengan tiap kategori, nilai ini dapat dihapuskan karena asumsi atribut tidak selalu terkait (*conditionally indenpenden*) maka:

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i) \end{aligned}$$

Bila $P(X)$ dapat diketahui melalui perhitungan diatas, maka klas (label) dari data sampel X adalah klas (label) yang memiliki $P(X|C_i) \times P(C_i)$ maksimum.

$$\text{Posterior} = \frac{\text{likelihood} \times \text{prior probability}}{\text{Evidence}}$$

Berikut flowchart alur algoritma dari Naïve Bayes :



2.3 Support Vector Machine

Support Vector Machine adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) yang berdimensi tinggi dan mengimplementasikan *learning* bias yang berasal dari teori pembelajaran statistik yang dilatih dengan algoritma pembelajaran (Borman & Wati, 2020). Teori yang mendasari SVM telah berkembang sejak tahun 1960-an, tetapi baru diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992. Secara sederhana konsep SVM adalah usaha mencari *hyperplane* “terbaik” yang berperan penting sebagai garis batas dua buah class (Jumeilah, 2017).

SVM mencari *hyperplane* ini berdasarkan *support vectors* dan *margin*. *Support vectors* adalah seluruh vektor data yang berjarak paling mendekati *hyperplane*, sedangkan *margin* menyatakan lebar dari *separating hyperplane*. *Linearly separable* data merupakan data yang dapat dipisahkan secara linier.

Misalkan $\{x_1, \dots, x_n\}$ adalah *dataset* dan $y_i \in \{+1, -1\}$ adalah label kelas dari data x_i , label +1 menandakan bahwa data tersebut diklasifikasikan sebagai kelas +1 dan label -1 menandakan sebaliknya. Tujuan dari SVM adalah menghasilkan sebuah model klasifikasi berupa fungsi $sign(x)$, $f(x) = y$, agar dapat mengklasifikasikan data pada proses *testing*.

Secara umum, cara kerja dari SVM adalah menemukan jarak terjauh dari *hyperplane* dengan kedua kelas. Proses penentuan jarak terjauh dilakukan berulang kali hingga menemukan *hyperplane* terbaik. Untuk itulah diperlukan optimasi pada SVM untuk menemukan jarak maksimum *hyperplane* dengan kedua kelas tersebut.

Dalam pembangunan SVM, terdapat dua bentuk optimasi yang digunakan untuk menemukan *hyperplane*. Bentuk optimasi pertama yaitu Primal Form SVM dan yang kedua adalah Dual Form SVM. Dual Form SVM, dibangun dengan menggunakan pendekatan *Lagrange*. Bentuk Dual Form merupakan perubahan dari bentuk Primal Form yang dimodifikasi menggunakan *Lagrange* sehingga pencarian *hyperplane* dapat dilakukan. Persamaan *Lagrange* dibangun dengan menggunakan konstanta α yang untuk selanjutnya konstanta tersebut yang digunakan sebagai penanda *support vectors*. Persoalan ini akan lebih mudah diselesaikan jika ditransformasi ke dalam persamaan Lagrange yang menggunakan *lagrange multiplier*. Dengan demikian permasalahan optimasi konstrain dapat diubah menjadi:

$$\max^{L_D} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j$$

Nilai α seperti yang telah dijelaskan sebelumnya, merupakan penanda apakah data tersebut merupakan *support vectors* atau tidak yang ditandai dengan nilai $\alpha \geq 0$ merupakan *support vectors*, sedangkan sisanya memiliki nilai $\alpha_i = 0$. Dengan demikian, dapat diperoleh nilai α_i yang nantinya digunakan untuk menemukan w menggunakan Persamaan berikut:

$$\frac{\partial}{\partial w} L_p(w, b, \alpha) = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i$$

Keterangan:

w : Normal bidang

b : Posisi bidang relatif terhadap pusat koordinat

Formula pencarian bidang yang pembatas terbaik pada persamaan diatas adalah *quadratic programming*, sehingga nilai maksimum global dari α_i selalu dapat ditemukan. Sedangkan untuk mencari nilai bias (b) digunakan persamaan berikut:

$$b = \frac{1}{\#SV} \sum_{x_i \in SV} \left(\frac{1}{y_i} \right) - \sum_{x_j \in SV} \alpha_j y_j k(x_j x_i)$$

Keterangan:

#SV : Jumlah *Support Vector*

Setelah solusi permasalahan *quadratic programming* (nilai α_i) dan nilai bias (b) ditemukan. Maka kelas dari data pengujian x dapat ditentukan berdasarkan nilai dari fungsi keputusan:

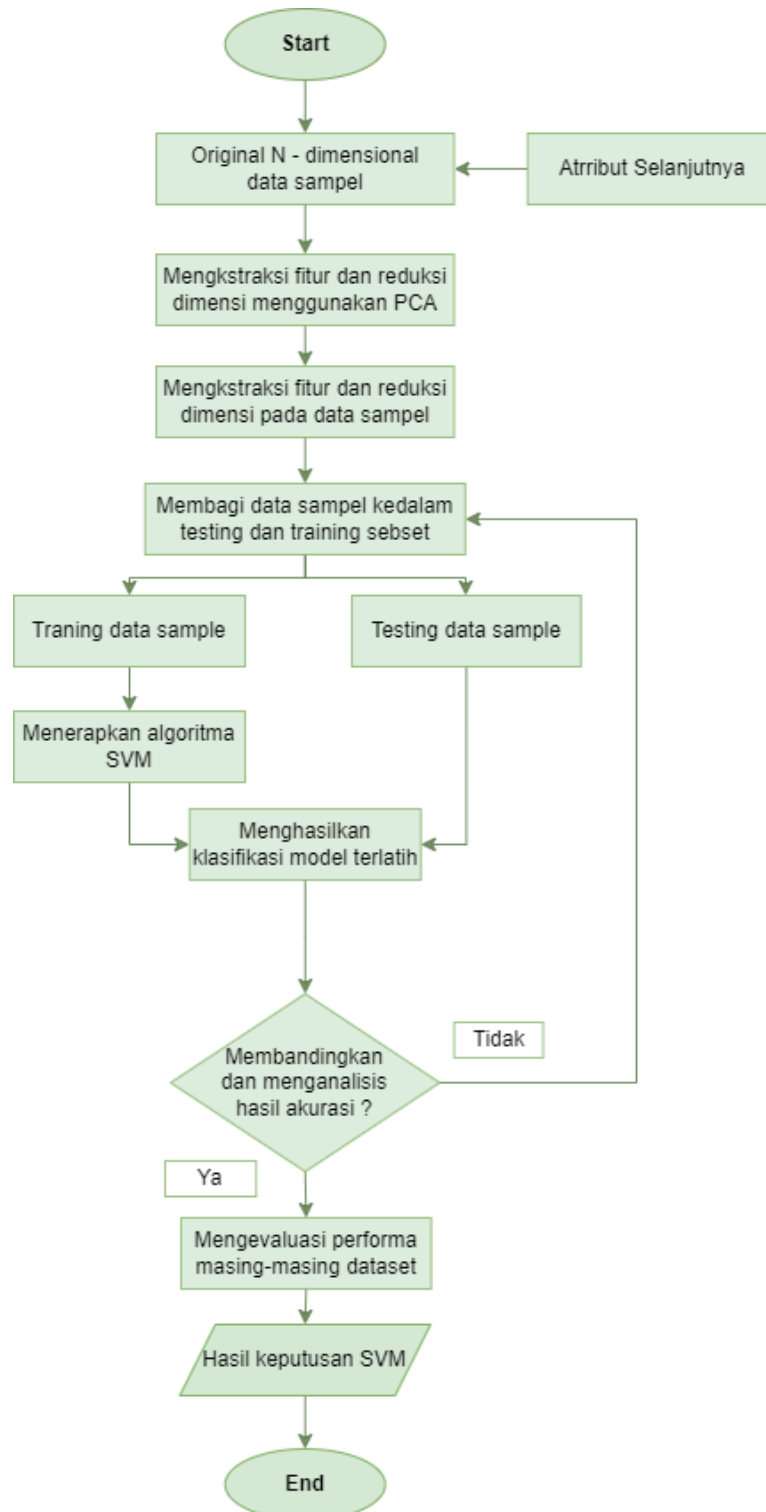
$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i x_i x_d + b$$

Keterangan:

ns : Jumlah *Support Vector*

xd : Data yang akan diklasifikasikan

Berikut flowchart alur algoritma dari *Support Vector Machine*:



2.4 Random Forest

Metode *Random Forest* (RF) merupakan metode yang dapat meningkatkan hasil akurasi, karena dalam membangkitkan simpul anak untuk setiap node dilakukan secara acak. Metode ini digunakan untuk membangun pohon keputusan yang terdiri dari *root node*, *internal node*, dan *leaf node* dengan mengambil atribut dan data secara acak sesuai ketentuan yang diberlakukan. *Root node* merupakan simpul yang terletak paling atas, atau biasa disebut sebagai akar dari pohon keputusan. *Internal node* adalah simpul percabangan, dimana *node* ini mempunyai *output* minimal dua dan hanya ada satu input. Sedangkan *leaf node* atau *terminal node* merupakan simpul terakhir yang hanya memiliki satu *input* dan tidak mempunyai *output*. Pohon keputusan dimulai dengan cara menghitung nilai *entropy* sebagai penentu tingkat ketidakmurnian atribut dan nilai *information gain*. Untuk menghitung nilai *entropy* digunakan rumus seperti pada persamaan 1, sedangkan nilai *information gain* menggunakan persamaan 2 (Schouten, et al., 2016).

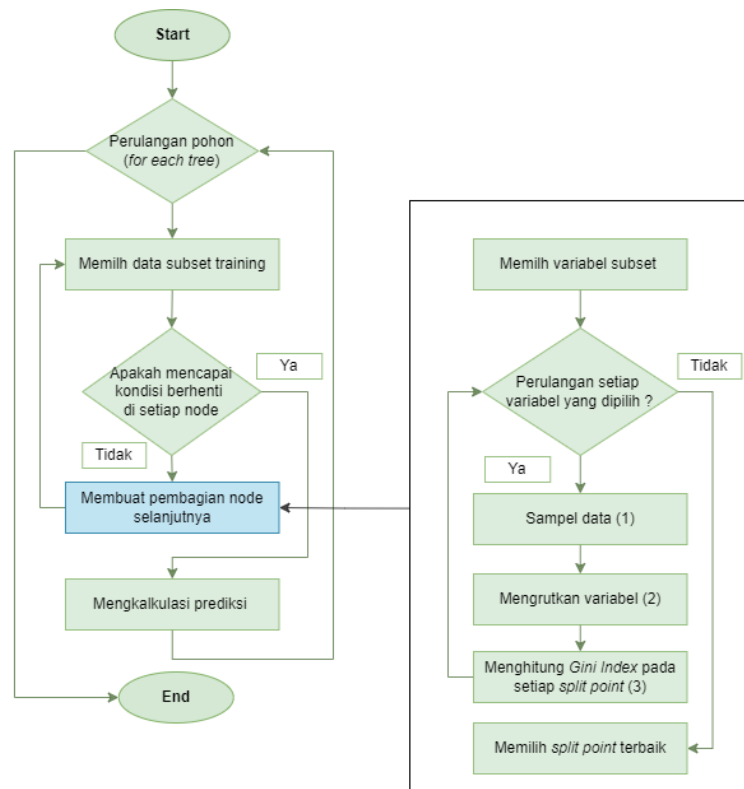
$$Entropy(Y) = \sum_i p(c|Y) \log_2 p(c|Y) \quad (1)$$

Dimana Y adalah himpunan kasus dan $p(c|Y)$ merupakan proporsi nilai Y terhadap kelas c.

$$Information\ Gain(Y, a) = Entropy(Y) - \sum_{v \in Values(a)} \frac{|Y_v|}{|Y_a|} Entropy(Y_v) \quad (2)$$

Dimana *Values(a)* merupakan semua nilai yang mungkin dalam himpunan kasus a. Y_v adalah subkelas dari Y dengan kelas v yang berhubungan dengan kelas a. Y_a adalah semua nilai yang sesuai dengan a.

Berikut flowchart alur algoritma dari *Random Forest*:



BAB III HASIL DAN PEMBAHASAN

3.1 Pengumpulan dan Perencanaan Dataset

Pemngimplementasian pada proses pengumpulan dataset berdasarkan latar belakang yang telah disebutkan, di lakukan perencanaan dan analisis kebutuhan dataset untuk bisa dijadikan sample dalam pengujian aplikasi *machine learning* dengan menerapkan metode klasifikasi yang dibuat. Proses pengumpulan dan analisa data dilakukan dengan mencari kriteria data yang sesuai permasalahan melalui internet, sehingga didapat dataset yang paling mewakili bersumber dari kaggle.com.

Dataset yang digunakan merupakan data pertanian presisi (*Precision agirculture*) yang menggunakan data karakteristik tanah, jenis tanah, data hasil panen, kondisi cuaca dan menyarankan petani dengan tanaman paling optimal untuk ditanam di lahan mereka untuk hasil dan keuntungan maksimal. Teknik ini dapat mengurangi kegagalan panen dan akan membantu petani untuk mengambil keputusan yang tepat tentang strategi pertanian mereka. Tujuan dari proyek ini ialah untuk merekomendasikan tanaman yang optimal dibudidayakan oleh petani berdasarkan beberapa parameter dan membantu mereka membuat keputusan sebelum budidaya.

Data yang digunakan dalam proyek ini dibuat dengan menambah dan menggabungkan berbagai kumpulan data India yang tersedia untuk umum seperti cuaca, tanah, dll. Anda dapat mengakses kumpulan data tersebut di sini. Data ini relatif sederhana dengan fitur yang sangat sedikit tetapi berguna tidak seperti fitur rumit yang memengaruhi hasil panen. Didalamnya memuat data nilai Nitrogen, Fosfor, Kalium dan pH tanah. Selain itu jua mengandung kelembaban, suhu dan curah hujan yang dibutuhkan untuk tanaman tertentu.

Berikut detail atribut beserta keterangannya :

No	Attribute Dataset	Keterangan
1.	N	Rasio dari nitrogen yang terkandung di dalam tanah
2.	P	Rasio dari fosfor yang terkandung di dalam tanah
3.	K	Rasio dari kalium yang terkandung di dalam tanah
4.	temperature	Suhu dalam derajat celcius
5.	humidity	Kelembapan reatif dalam %

6.	Ph	Nilai PH tanah
7.	rainfall	Curah hujan dalam mm

Adapun mengenai data yang dijadikan label sebagai berikut:

No	Attribute Label	Keterangan
1.	Rice	Tanaman Padi
2.	Maize	Tanaman Jagung
3.	Chickpea	Tanaman Buncis
4.	Kidneybeans	Tanaman Kacang Merah
5.	Pigeonpeas	Tanaman Kacang Gude
6.	Mothbeans	Tanaman Kacang Ngngat / Kacang Matki
7.	Mungbean	Tanaman Kacang Hijau
8.	Blackgram	Tanaman Lentil hitam
9.	Lentil	Tanaman Lentil
10.	Pomegranate	Tanaman Delima
11.	Banana	Tanaman Pisang
12.	Mango	Tanaman Buah Mangga
13.	Grapes	Tanaman Buah Anggur
14.	Watermelon	Tanaman Buah Semangka
15.	Muskmelon	Tanaman Buah Melon
16.	Apple	Tanaman Buah Apel
17.	Orange	Tanaman Buah Jeruk
18.	Papaya	Tanaman Buah Pepaya
19.	Coconut	Tanaman Kelapa
20.	Cotton	Tanaman Kapas
21.	Jute	Tanaman Rami
22.	Coffee	Tanaman Kopi

Berdasarkan atribut data juga label, disajikan sejumlah 2201 data nilai dari masing-masing atribut dan juga label yang sesuai sebagai berikut :

N	P	K	Temperature	humidity	ph	rainfall
90	42	43	20,87974371	82,00274423	6,502985292	202,9355362
85	58	41	21,77046169	80,31964408	7,038096361	226,6555374
60	55	44	23,00445915	82,32076290	7,840207144	263,9642476
74	35	40	26,49109635	80,15836264	6,980400905	242,8640342
78	42	42	20,13017482	81,60487287	7,628472891	262,7173405
69	37	42	23,05804872	83,37011772	7,073453503	251,0549998
Dst..

Dari data yang tersaji diatas, akan dilakukan pengolahan melalui metode klasifikasi untuk memperoleh tanaman yang tepat dalam lingkungan tertentu.

3.2 Perhitungan Manual

Link spreadsheet perhitungan manual :

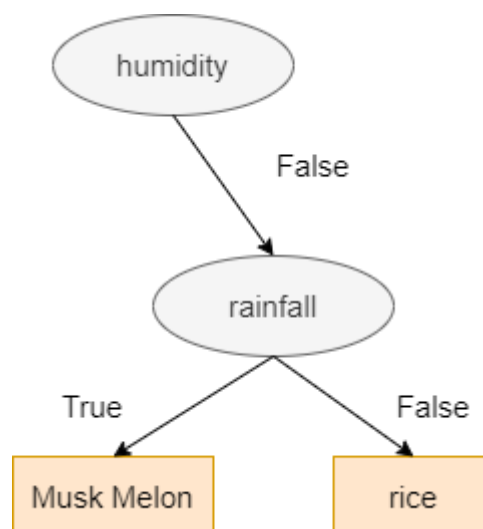
<https://1drv.ms/x/s!At7iaBCoi5dZgXexEzzDwoig-NW0?e=vVNGTx>

Perhitngan manual digunakan untuk mengetahui label yang tepat berdasarkan masing-masing record di tiap attribute secara manual menggunakan rumus. Perhitungan tersebut diterapkan dalam metodologi klasifikasi dengan menggunakan metode *Decision Tree*, *Support Vector Machine*, *Naïve Bayes Classifier*, dan *Random Forest*. Adapun Perhitungan manual dari masing-masing metode tersebut secara detail, sebagai berikut :

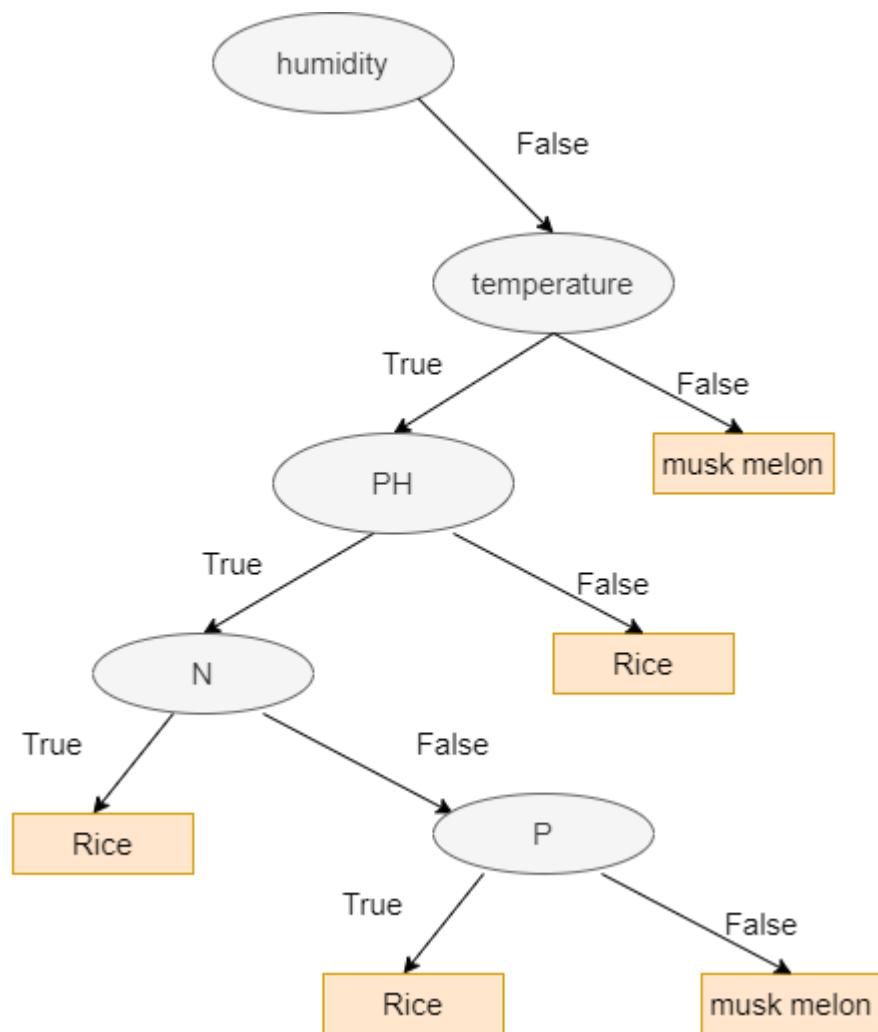
3.3.1 Decission Tree

Decisson Tree 1

Decisson Tree 2



Decission Tree 3



3.3.2 Support Vector Machine

Berikut merupakan dataset yang digunakan untuk membuat model SVM, dimana data yang berwarna biru merupakan data training, sedangkan untuk data yang hijau merupakan data testing.

Raw Dataset									
N	P	K	temperature	humidity	ph	rainfall	label	encode label	
90	42	43	20,87974371	82,00274423	6,502985292	202,9355	rice	1	
85	58	41	21,77046169	80,31964408	7,038096361	226,6555	rice	1	
60	55	44	23,00445915	82,3207629	7,840207144	263,9642	rice	1	
74	35	40	26,49109635	80,15836264	6,980400905	242,864	rice	1	
61	38	20	18,47891261	62,69503871	5,970458434	65,43835	maize	-1	
71	54	16	22,61359953	63,69070564	5,749914421	87,75954	maize	-1	
61	44	17	26,10018422	71,57476937	6,931756558	102,2662	maize	-1	
80	43	16	23,55882094	71,59351368	6,657964753	66,71995	maize	-1	
73	58	21	19,97215954	57,68272924	6,596060648	60,65171	maize	-1	
78	42	42	20,13017482	81,60487287	7,628472891	262,7173	rice	1	

a Training model

Dari tabel diatas terdapat 7 feature sebagai variabel x, yakni kolom N, P, K, temperature, humidity, ph, dan rainfall, sehingga nanti terbentuklah rumus persamaan sebagai berikut

- $-(w1 \cdot X1) - (w2 \cdot X2) - (w3 \cdot X3) - (w4 \cdot X4) - (w5 \cdot X5) - (w6 \cdot X6) - (w7 \cdot X7) - b = -1$, untuk data maize, serta
- $(w1 \cdot X1) + (w2 \cdot X2) + (w3 \cdot X3) + (w4 \cdot X4) + (w5 \cdot X5) + (w6 \cdot X6) + (w7 \cdot X7) + b = 1$, untuk data rice

Kemudian menerapkan persamaan tersebut dengan mensubtitusi nilai-nilai x sesuai yang ada pada dataset maka akan menghasilkan persamaan-persaman berikut

- $90w1 + 42w2 + 43w3 + 20,8797w4 + 82.00274423w5 + 6.5029 w6 + 202.935 w7 + b = 1$
- $85w1 + 58w2 + 41w3 + 21.7704 w4 + 80.3196 w5 + 7.0380 w6 + 226.6555 w7 + b = 1$
- $60w1 + 55w2 + 44w3 + 23.0044 w4 + 82.3207 w5 + 7.8402 w6 + 263.9642 w7 + b = 1$
- $74w1 + 35w2 + 40w3 + 26.4910 w4 + 80.1583 w5 + 6.9804 w6 + 242.8640 w7 + b = 1$
- $61w1 - 38w2 - 20w3 - 18,4789 w4 - 62,6950 w5 - 5,9704 w6 - 65,4383 w7 - b = -1$
- $71w1 - 54w2 - 16w3 - 22,6135 w4 - 63,6907 w5 - 5,7499 w6 - 87,7595 w7 - b = -1$
- $61w1 - 44w2 - 17w3 - 26,1001w4 - 71,5747 w5 - 6,9317 w6 - 102,2662 w7 - b = -1$
- $80w1 - 43w2 - 16w3 - 23,5588 w4 - 71,5935 w5 - 6,6579 w6 - 66,7199 w7 - b = -1$

Langkah berikutnya adalah melakukan elmininasi subtitusi menggunakan metode gauss-jordan terhadap persamaan diatas agar mendapatkan nilai $w1, w2, w3, w4, w5, w6, w7$, dan b (bias) sehingga dapat membentuk sebuah model. Setelah melakukan eliminasi dan subtitusi maka hasil yang didapatkan adalah sebagai berikut,

w1	=	-0.000992667492848
w2	=	0.009710799249905
w3	=	0.072344513896742
w4	=	0.093965349206197
w5	=	-0.033268263348611
w6	=	0.40572325565583
w7	=	-0.057730094286981
b	=	0.057730094286981

Rumus model yang didapatkan :

$$-0,00099 \cdot X1 + 0,0097 \cdot X2 + 0,07234 \cdot X3 + 0,09396 \cdot X4 - 0,03226 \cdot X5 + 0,40572 \cdot W6 - 0,57730 \cdot W7 + 0,0577$$

b Melakukan testing model

Setelah mendapatkan model diatas, langkah selanjutnya adalah melakukan testing terhadap data test dengan cara memasukkan nilai x1 sampai x7 dalam rumus diatas.

- Adapun test untuk data t1

73	58	21	19,97	57,68	6,6	60,65	maize	-1	t1
----	----	----	-------	-------	-----	-------	-------	----	----

$$-0,00099 \cdot 73 + 0,0097 \cdot 58 + 0,07234 \cdot 21 + 0,09396 \cdot 19,97 - 0,03226 \cdot 57,68 + 0,40572 \cdot 6,6 - 0,57730 \cdot 60,65 + 0,0577 = 1,2439$$

Berdasarkan hasil perhitungan diatas menghasilkan angka lebih dari sama dengan 0, sedangkan pada data testing menunjukkan hasilnya – 1 dimana kurang dari sama dengan 0, yang artinya hasil yang didapatkan tidak valid.

- Adapun test untuk data t2

78	42	42	20,13	81,6	7,63	262,72	rice	1	t2
----	----	----	-------	------	------	--------	------	---	----

$$-0,00099 \cdot 78 + 0,0097 \cdot 42 + 0,07234 \cdot 42 + 0,09396 \cdot 20,13 - 0,03226 \cdot 81,6 + 0,40572 \cdot 7,63 - 0,57730 \cdot 262,72 + 0,0577 = - 9,8512$$

Berdasarkan hasil perhitungan diatas menghasilkan angka kurang dari sama dengan 0, sedangkan pada data testing menunjukkan hasilnya 1

dimana lebih dari sama dengan 0, yang artinya hasil yang didapatkan tidak valid.

- Berdasarkan testing yang telah dilakukan dapat disimpulkan penyelesaian permasalahan klasifikasi pada kasus penentuan rekomendasi tanaman menggunakan metode suport vector machine memiliki akurasi yang rendah.

3.3.3 Naïve Bayes Classifier

Tabel 2. 1 Contoh soal

Contoh soal
N = 91
P = 42
K = 44
<i>Temperature</i> = 21
<i>Humidity</i> = 80
ph = 5
<i>Rainfall</i> = 206
Ditanyakan : label ??

Tabel 2. 2 Label

Label	
<i>rice</i>	100
<i>maize</i>	100
<i>chickpea</i>	100
<i>kidneybeans</i>	100
<i>pigeonpeas</i>	100
<i>mothbeans</i>	100
<i>mungbean</i>	100
<i>blackgram</i>	100
<i>lentil</i>	100
<i>pomegranate</i>	100
<i>banana</i>	100
<i>mango</i>	100
<i>grapes</i>	100
<i>watermelon</i>	100
<i>muskmelon</i>	100
<i>apple</i>	100
<i>orange</i>	100
<i>papaya</i>	100
<i>coconut</i>	100

<i>cotton</i>	100
<i>jute</i>	100
<i>coffee</i>	100

Tabel 2. 3 Peluang kemuculan setiap label

P(C_i X)	
P(<i>rice</i>)	0,045454545
P(<i>maize</i>)	0,045454545
P(<i>chickpea</i>)	0,045454545
P(<i>kidneybeans</i>)	0,045454545
P(<i>pigeonpeas</i>)	0,045454545
P(<i>mothbeans</i>)	0,045454545
P(<i>mungbean</i>)	0,045454545
P(<i>blackgram</i>)	0,045454545
P(<i>lentil</i>)	0,045454545
P(<i>pomegranate</i>)	0,045454545
P(<i>banana</i>)	0,045454545
P(<i>mango</i>)	0,045454545
P(<i>grapes</i>)	0,045454545
P(<i>watermelon</i>)	0,045454545
P(<i>muskmelon</i>)	0,045454545
P(<i>apple</i>)	0,045454545
P(<i>orange</i>)	0,045454545
P(<i>papaya</i>)	0,045454545
P(<i>coconut</i>)	0,045454545
P(<i>cotton</i>)	0,045454545
P(<i>jute</i>)	0,045454545
P(<i>coffee</i>)	0,045454545

Tabel 2. 4 Peluang kemunculan $N = 91$ pada setiap label

P(N=91 Ci)	
P(N=91 rice)	0,06
P(N=91 maize)	0,01
P(N=91 chickpea)	0
P(N=91 kidneybeans)	0
P(N=91 pigeonpeas)	0
P(N=91 mothbeans)	0
P(N=91 mungbean)	0
P(N=91 blackgram)	0
P(N=91 lentil)	0
P(N=91 pomegranate)	0
P(N=91 banana)	0,03
P(N=91 mango)	0
P(N=91 grapes)	0
P(N=91 watermelon)	0,05
P(N=91 muskmelon)	0,01
P(N=91 apple)	0
P(N=91 orange)	0
P(N=91 papaya)	0
P(N=91 coconut)	0
P(N=91 cotton)	0
P(N=91 jute)	0,02
P(N=91 coffee)	0,02

Tabel 2. 5 Peluang kemunculan $P = 42$ pada setiap label

P(P=42 Ci)	
P(P=42 rice)	0,04
P(P=42 maize)	0,01
P(P=42 chickpea)	0
P(P=42 kidneybeans)	0
P(P=42 pigeonpeas)	0
P(P=42 mothbeans)	0,02
P(P=42 mungbean)	0,01
P(P=42 blackgram)	0
P(P=42 lentil)	0
P(P=42 pomegranate)	0
P(P=42 banana)	0
P(P=42 mango)	0
P(P=42 grapes)	0
P(P=42 watermelon)	0
P(P=42 muskmelon)	0
P(P=42 apple)	0

$P(P=42 orange)$	0
$P(P=42 papaya)$	0
$P(P=42 coconut)$	0
$P(P=42 cotton)$	0,02
$P(P=42 jute)$	0,02
$P(P=42 coffee)$	0

Tabel 2. 6 Peluang kemunculan $K = 44$ pada setiap label

$P(K=44 Ci)$	
$P(K=44 rice)$	0,11
$P(K=44 maize)$	0
$P(K=44 chickpea)$	0
$P(K=44 kidneybeans)$	0
$P(K=44 pigeonpeas)$	0
$P(K=44 mothbeans)$	0
$P(K=44 mungbean)$	0
$P(K=44 blackgram)$	0
$P(K=44 lentil)$	0
$P(K=44 pomegranate)$	0,11
$P(K=44 banana)$	0
$P(K=44 mango)$	0
$P(K=44 grapes)$	0
$P(K=44 watermelon)$	0
$P(K=44 muskmelon)$	0
$P(K=44 apple)$	0
$P(K=44 orange)$	0
$P(K=44 papaya)$	0
$P(K=44 coconut)$	0
$P(K=44 cotton)$	0
$P(K=44 jute)$	0,06
$P(K=44 coffee)$	0

Tabel 2. 7 Peluang kemunculan $temperature = 21$ pada setiap label

$P(temperature=21 Ci)$	
$P(temperature=21 rice)$	0,11
$P(temperature=21 maize)$	0,05
$P(temperature=21 chickpea)$	0,12
$P(temperature=21 kidneybeans)$	0,12
$P(temperature=21 pigeonpeas)$	0,05
$P(temperature=21 mothbeans)$	0
$P(temperature=21 mungbean)$	0
$P(temperature=21 blackgram)$	0
$P(temperature=21 lentil)$	0,08

P(temperature=21 pomegranate)	0,08
P(temperature=21 banana)	0
P(temperature=21 mango)	0
P(temperature=21 grapes)	0,03
P(temperature=21 watermelon)	0
P(temperature=21 muskmelon)	0
P(temperature=21 apple)	0,15
P(temperature=21 orange)	0,06
P(temperature=21 papaya)	0
P(temperature=21 coconut)	0
P(temperature=21 cotton)	0
P(temperature=21 jute)	0
P(temperature=21 coffee)	0

Tabel 2. 8 Peluang kemunculan *humidity* = 80 pada setiap label

P(humidity=80 Ci)	
P(humidity=80 rice)	0,15
P(humidity=80 maize)	0
P(humidity=80 chickpea)	0
P(humidity=80 kidneybeans)	0
P(humidity=80 pigeonpeas)	0
P(humidity=80 mothbeans)	0
P(humidity=80 mungbean)	0,07
P(humidity=80 blackgram)	0
P(humidity=80 lentil)	0
P(humidity=80 pomegranate)	0
P(humidity=80 banana)	0,08
P(humidity=80 mango)	0
P(humidity=80 grapes)	0,13
P(humidity=80 watermelon)	0,06
P(humidity=80 muskmelon)	0
P(humidity=80 apple)	0
P(humidity=80 orange)	0
P(humidity=80 papaya)	0
P(humidity=80 coconut)	0
P(humidity=80 cotton)	0,06
P(humidity=80 jute)	0,04
P(humidity=80 coffee)	0

Tabel 2. 9 Peluang kemunculan $ph = 5$ pada setiap label

P(ph=5 Ci)	
P(ph=5 rice)	0,12
P(ph=5 maize)	0
P(ph=5 chickpea)	0
P(ph=5 kidneybeans)	0
P(ph=5 pigeonpeas)	0,37
P(ph=5 mothbeans)	0,15
P(ph=5 mungbean)	0
P(ph=5 blackgram)	0
P(ph=5 lentil)	0
P(ph=5 pomegranate)	0
P(ph=5 banana)	0
P(ph=5 mango)	0,4
P(ph=5 grapes)	0
P(ph=5 watermelon)	0
P(ph=5 muskmelon)	0
P(ph=5 apple)	0
P(ph=5 orange)	0
P(ph=5 papaya)	0
P(ph=5 coconut)	0
P(ph=5 cotton)	0
P(ph=5 jute)	0
P(ph=5 coffee)	0

Tabel 2. 10 Peluang kemunculan $rainfall = 206$ pada setiap label

P(rainfall=206 Ci)	
P(rainfall=206 rice)	0,01
P(rainfall=206 maize)	0
P(rainfall=206 chickpea)	0
P(rainfall=206 kidneybeans)	0
P(rainfall=206 pigeonpeas)	0
P(rainfall=206 mothbeans)	0
P(rainfall=206 mungbean)	0
P(rainfall=206 blackgram)	0
P(rainfall=206 lentil)	0
P(rainfall=206 pomegranate)	0
P(rainfall=206 banana)	0
P(rainfall=206 mango)	0
P(rainfall=206 grapes)	0
P(rainfall=206 watermelon)	0
P(rainfall=206 muskmelon)	0
P(rainfall=206 apple)	0

$P(\text{rainfall}=206 \text{orange})$	0
$P(\text{rainfall}=206 \text{papaya})$	0,01
$P(\text{rainfall}=206 \text{coconut})$	0,02
$P(\text{rainfall}=206 \text{cotton})$	0
$P(\text{rainfall}=206 \text{jute})$	0
$P(\text{rainfall}=206 \text{coffee})$	0

Tabel 2. 11 Peluang total dari setiap kejadian terhadap masing-masing label

P(X Ci)	
P(X rice)	5,2E-09
$P(X \text{maize})$	0
$P(X \text{chickpea})$	0
$P(X \text{kidneybeans})$	0
$P(X \text{pigeonpeas})$	0
$P(X \text{mothbeans})$	0
$P(X \text{mungbean})$	0
$P(X \text{blackgram})$	0
$P(X \text{lentil})$	0
$P(X \text{pomegranate})$	0
$P(X \text{banana})$	0
$P(X \text{mango})$	0
$P(X \text{grapes})$	0
$P(X \text{watermelon})$	0
$P(X \text{muskmelon})$	0
$P(X \text{apple})$	0
$P(X \text{orange})$	0
$P(X \text{papaya})$	0
$P(X \text{coconut})$	0
$P(X \text{cotton})$	0
$P(X \text{jute})$	0
$P(X \text{coffee})$	0

Tabel 2. 12 Label dengan peluang tertinggi

P(X Ci) * P(Ci)			
P(X rice)	*	P(rice)	2,376E-10
$P(X \text{maize})$	*	$P(\text{maize})$	0
$P(X \text{chickpea})$	*	$P(\text{chickpea})$	0
$P(X \text{kidneybeans})$	*	$P(\text{kidneybeans})$	0
$P(X \text{pigeonpeas})$	*	$P(\text{pigeonpeas})$	0
$P(X \text{mothbeans})$	*	$P(\text{mothbeans})$	0
$P(X \text{mungbean})$	*	$P(\text{mungbean})$	0
$P(X \text{blackgram})$	*	$P(\text{blackgram})$	0
$P(X \text{lentil})$	*	$P(\text{lentil})$	0

$P(X pomegranate)$	*	$P(pomegranate)$	0
$P(X banana)$	*	$P(banana)$	0
$P(X mango)$	*	$P(mango)$	0
$P(X grapes)$	*	$P(grapes)$	0
$P(X watermelon)$	*	$P(watermelon)$	0
$P(X muskmelon)$	*	$P(muskmelon)$	0
$P(X apple)$	*	$P(apple)$	0
$P(X orange)$	*	$P(orange)$	0
$P(X papaya)$	*	$P(papaya)$	0
$P(X coconut)$	*	$P(coconut)$	0
$P(X cotton)$	*	$P(cotton)$	0
$P(X jute)$	*	$P(jute)$	0
$P(X coffee)$	*	$P(coffee)$	0

Jadi berdasarkan soal diatas, didapatkan hasil bahwa klasifikasi yang cocok dengan setiap fitur yang diberikan adalah label “**Rice**” atau padi.

3.3.4 Random Forest

Tabel 3. 1 Data yang akan diolah

N	P	K	temperatur	humidity	ph	rainfall	label
90	42	43	21	82	7	203	<i>rice</i>
85	58	41	22	80	7	227	<i>rice</i>
60	55	44	23	82	8	264	<i>rice</i>
117	25	53	29	92	6	25	<i>muskmelon</i>
85	21	52	30	90	6	24	<i>muskmelon</i>
104	25	55	30	90	6	23	<i>muskmelon</i>
102	24	54	28	91	7	23	<i>muskmelon</i>

Tabel 3. 2 Data sample

Name	N	P	K	temperature	humidity	ph	rainfall	label
S1	90	42	43	21	82	7	203	<i>rice</i>
S2	60	55	44	23	82	8	264	<i>rice</i>
S3	90	42	43	21	82	7	203	<i>rice</i>
S4	117	25	53	29	92	6	25	<i>muskmelon</i>
S5	104	25	55	30	90	6	23	<i>muskmelon</i>
S6	102	24	54	28	91	7	23	<i>muskmelon</i>

Tabel 3. 3 Data out of bag

Name	N	P	K	temperature	humidity	ph	rainfall	label
O1	85	58	41	22	80	7	227	<i>rice</i>
O2	85	21	52	30	90	6	24	<i>muskmelon</i>

Tabel 3. 4 Data *training*

<i>Name</i>	N	P	K	<i>temperature</i>	<i>humidity</i>	<i>ph</i>	<i>rainfall</i>	label
TR1	90	42	43	21	82	7	203	<i>rice</i>
TR2	60	55	44	23	82	8	264	<i>rice</i>
TR3	90	42	43	21	82	7	203	<i>rice</i>
TR4	117	25	53	29	92	6	25	<i>muskmelon</i>
TR5	104	25	55	30	90	6	23	<i>muskmelon</i>
TR6	102	24	54	28	91	7	23	<i>muskmelon</i>

Tabel 3. 5 Data *testing*

<i>Name</i>	N	P	K	<i>temperature</i>	<i>humidity</i>	<i>ph</i>	<i>rainfall</i>	label
TS1	85	58	41	22	80	7	227	<i>rice</i>
TS2	85	21	52	30	90	6	24	<i>muskmelon</i>

Nama	Label	<i>Decission Tree 1</i>		<i>Decission Tree 2</i>		<i>Decission Tree 3</i>	
		<i>Rice</i>	<i>Muskmelon</i>	<i>Rice</i>	<i>Muskmelon</i>	<i>Rice</i>	<i>Muskmelon</i>
S1	<i>Rice</i>	v		v		v	
S2	<i>Rice</i>	v		v		v	
S3	<i>Rice</i>	v		v		v	
S4	<i>Muskmelon</i>		v		v		v
S5	<i>Muskmelon</i>		v		v		v
S6	<i>Muskmelon</i>		v		v		v
O1	<i>Rice</i>	v		v		v	
O2	<i>Muskmelon</i>		v		v		v
TR1	<i>Rice</i>	v		v		v	
TR2	<i>Rice</i>	v		v		v	
TR3	<i>Rice</i>	v		v		v	
TR4	<i>Muskmelon</i>		v		v		v
TR5	<i>Muskmelon</i>		v		v		v
TR6	<i>Muskmelon</i>		v		v		v
TS1	<i>Rice</i>	v		v		v	
TS2	<i>Muskmelon</i>		v		v		v

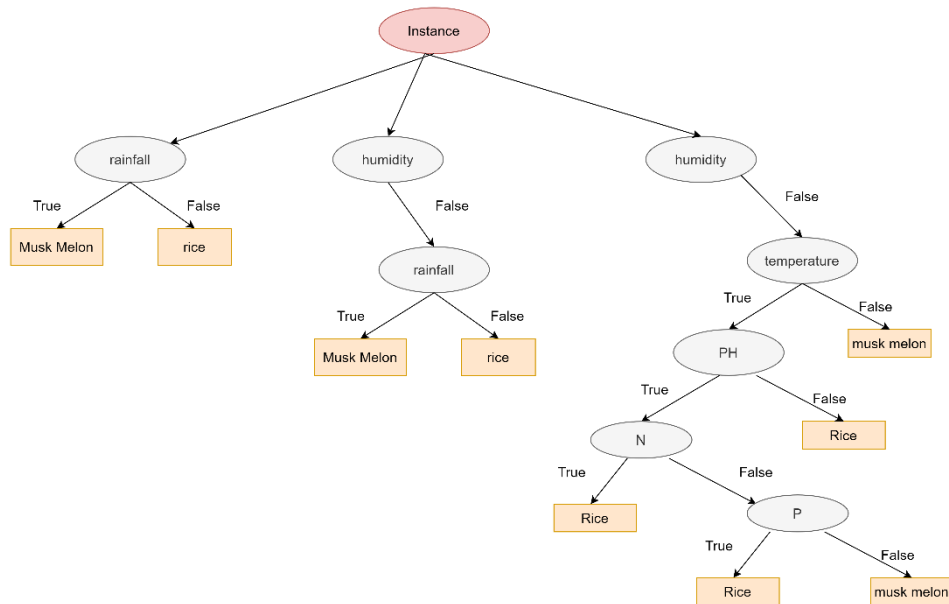
Perhitungan akurasi

Total tes dilakukan = 48

Hasil sesuai harapan = 48

Persentase akurasi = $48 / 48 * 100\% = 100\%$

Diagram random forest



3.3 Pengimplementasian Pada Aplikasi

Berdasarkan metodologi dan perhitungan manual yang dilakukan di dapat hasil penerapan pada aplikasi menggunakan library pada phyton sebagai berikut :

a. Mengimport library

```

# import library yang dibutuhkan
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
  
```

b. Menyiapkan dataset

```

# import dataset

df = pd.read_csv('/content/drive/MyDrive/KULIAH/- SEMESTER 5 -/Crop_recommendation.csv')
df.head()
  
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

c. Menampilakn informasi pada dataset

```
[ ] # detail dataset  
df.shape
```

```
(2200, 8)
```

```
# jenis label pada dataset  
df['label'].unique()
```

```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',  
      'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',  
      'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',  
      'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],  
      dtype=object)
```

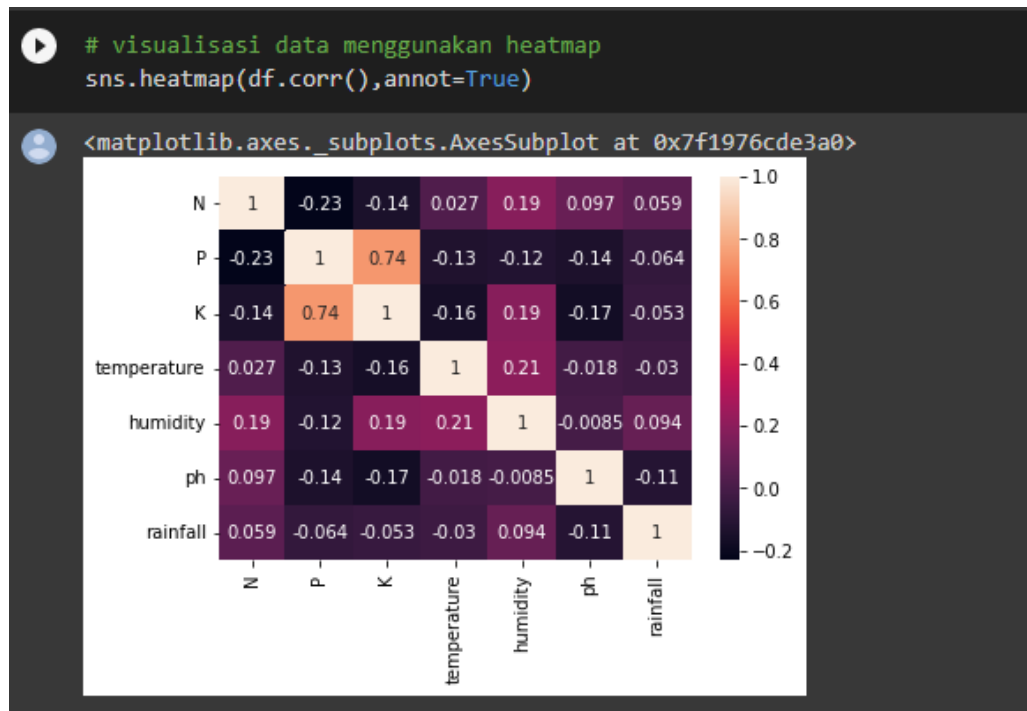
```
# tipe data tiap kolom dataset  
df.dtypes
```

```
N          int64  
P          int64  
K          int64  
temperature float64  
humidity    float64  
ph          float64  
rainfall    float64  
label       object  
dtype: object
```

```
# jumlah data tiap label  
df['label'].value_counts()
```

```
rice          100  
maize         100  
jute          100  
cotton        100  
coconut       100  
papaya        100  
orange        100  
apple         100  
muskmelon     100  
watermelon    100  
grapes        100  
mango         100  
banana        100  
pomegranate   100  
lentil        100  
blackgram     100  
mungbean      100  
mothbeans     100  
pigeonpeas    100  
kidneybeans   100  
chickpea      100  
coffee       100  
Name: label, dtype: int64
```

- d. Visualisasi dataset kedalam heatmap diagram



- e. Memisahkan dan split dataset

```
[ ] # pisahkan variabel x (features) dan variabel y (target & label)

features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
labels = df['label']

[ ] # buat array untuk menyimpan akurasi dan nama setiap model
acc = []
model = []

[ ] # Splitting data train dan test dengan perbandingan 8 : 2

Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_
```

- f. Membuat Model Descision Tree Entropy

```
[ ] # inisialisasi model
DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

# fitting data train
DecisionTree.fit(Xtrain,Ytrain)

# testing
predicted_values = DecisionTree.predict(Xtest)

# hitung akurasi model
x = metrics.accuracy_score(Ytest, predicted_values)

# simpan akurasi model dan nama model
acc.append(x)
model.append('Decision Tree')

print("Akurasi Menggunakan Decission Tree Entropy : ", x*100)
```

Akurasi Menggunakan Decission Tree Entropy : 90.0

```
# melakukan data validation menggunakan cross validation :

score = cross_val_score(DecisionTree, features, target,cv=5)
score
```

array([0.93636364, 0.90909091, 0.91818182, 0.87045455, 0.93636364])

g. Gaussian Naïve Bayes

```
#inisialisasi model
NaiveBayes = GaussianNB()

# fitting data train
NaiveBayes.fit(Xtrain,Ytrain)

# testing dengan data test
predicted_values = NaiveBayes.predict(Xtest)

# hitung akurasi model
x = metrics.accuracy_score(Ytest, predicted_values)

# simpan akurasi model dan nama model
acc.append(x)
model.append('Naive Bayes')
print("Akurasi Menggunakan Naive Bayes Gaussian: ", x*100)
```

Akurasi Menggunakan Naive Bayes Gaussian: 99.0909090909091

```
# melakukan data validation menggunakan cross validation :

score = cross_val_score(NaiveBayes,features,target,cv=5)
score
```

array([0.99772727, 0.99545455, 0.99545455, 0.99545455, 0.99090909])

h. Support Vector Machine (SVM)

```
# inisialisasi model
SVM = SVC(gamma='auto')

# fitting data train
SVM.fit(Xtrain,Ytrain)

# testing menggunakan data test
predicted_values = SVM.predict(Xtest)

# hitung akurasi model
x = metrics.accuracy_score(Ytest, predicted_values)

# simpan akurasi model dan nama model
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x*100)
```

SVM's Accuracy is: 10.6818181818182


```
# melakukan data validation menggunakan cross validation :

score = cross_val_score(SVM,features,target,cv=5)
score

array([0.27727273, 0.28863636, 0.29090909, 0.275      , 0.26818182])
```

i. Random Forest

```
# inisialisasi model
RF = RandomForestClassifier(n_estimators=20, random_state=0)

# fitting data train
RF.fit(Xtrain,Ytrain)

# testing menggunakan data test
predicted_values = RF.predict(Xtest)

# hitung akurasi model
x = metrics.accuracy_score(Ytest, predicted_values)

# simpan akurasi model dan nama model
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x*100)

RF's Accuracy is: 99.0909090909091
```

```
# melakukan data validation menggunakan cross validation :

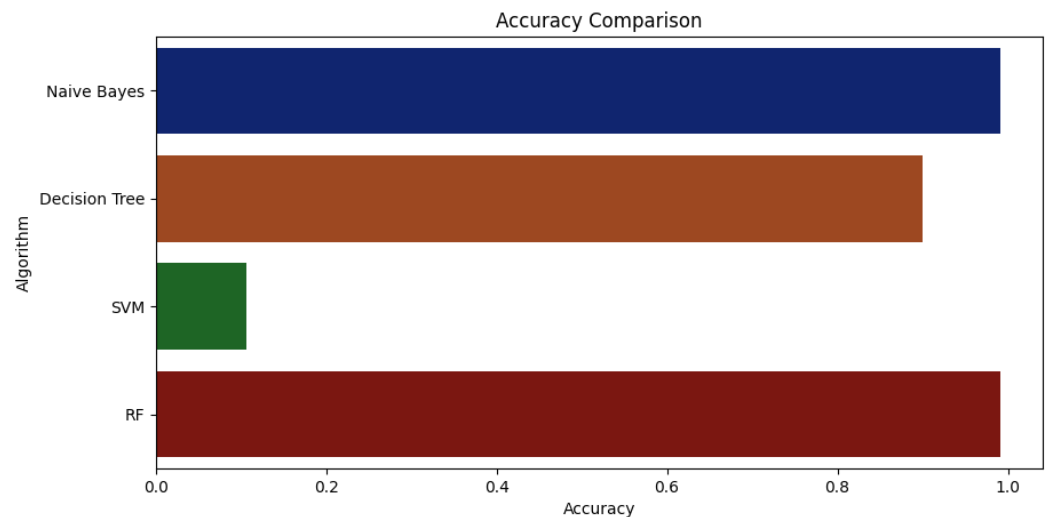
score = cross_val_score(RF,features,target,cv=5)
score

array([0.99772727, 0.99545455, 0.99772727, 0.99318182, 0.98863636])
```

j. Evaluasi hasil model dengan akurasi tertinggi

```
[ ] # visualisasi

plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
```



Dari visualisasi diatas dapat disimpulkan model yang menghasilkan akurasi paling tinggi, atau model yang cocok digunakan untuk melakukan klasifikasi berdasarkan feature yang ada pada dataset adalah model *Random Forest* dan *Naive Bayes*, karena keduanya memiliki nilai akurasi sama dan paling tinggi dari model lainnya. Selanjutnya menyimpan model *Naive Bayes* yang sudah terlatih sebagai berikut :

```
[ ] import pickle

# membuat nama file
NB_pkl_filename = 'NBClassifier.pkl'

# buka file dan simpan sebagai ekstensi 'pkl'
NB_Model_pkl = open(NB_pkl_filename, 'wb')
pickle.dump(NaiveBayes, NB_Model_pkl)

# tutup instans pickle
NB_Model_pkl.close()
```

```
# inisialisasi array untuk menyimpan nilai inputan
list = []

# input data
ask = 'Masukkan jumlah Senyawa N : '
val = float(input(ask))
list.append(val)

ask = 'Masukkan jumlah Senyawa P : '
val = float(input(ask))
list.append(val)

ask = 'Masukkan jumlah Senyawa K : '
val = float(input(ask))
list.append(val)

ask = 'Masukkan Temperature suhu : '
val = float(input(ask))
list.append(val)
```

```

ask = 'Masukkan tingkat kelembaban : '
val = float(input(ask))
list.append(val)

ask = 'Masukkan tingkat PH tanah : '
val = float(input(ask))
list.append(val)

ask = 'Masukkan tingkat Curah Hujan : '
val = float(input(ask))
list.append(val)

# masukkan pada model yang telah dilatih
data = np.array([list])
prediction = NaiveBayes.predict(data)

print("\n\n\n Jadi berdasarkan inputan Anda, tanaman yang cocok ditanam pada lingkungan

```

Output

```

Masukkan jumlah Senyawa N : 88
Masukkan jumlah Senyawa P : 53
Masukkan jumlah Senyawa K : 48
Masukkan Temperature suhu : 23
Masukkan tingkat kelembaban : 82
Masukkan tingkat PH tanah : 6.5
Masukkan tingkat Curah Hujan : 230

Jadi berdasarkan inputan Anda, tanaman yang cocok ditanam pada lingkungan tersebut adalah : rice

```

BAB IV

KESIMPULAN

Berdasarkan hasil pengolahan dan analisis dataset *crop recommendation* menggunakan metodologi klasifikasi dengan menerapkan metode *Decision Tree*, *Support Vector Machine*, *Naïve Bayes Classifier*, dan *Random Forest* didapat rentang akurasi antara 10% - 99% dengan hasil nilai akurasi tertinggi menggunakan metode model *Random Forest* dan *Naive Bayes* berjumlah 99% mendekati sempurna. Hal ini menunjukkan bahwa kedua model tersebut cocok diterapkan berdasarkan dataset yang telah dibuat.

Hasil nilai akhir akurasi tersebut bisa dijadikan acuan akurat dalam menentukan tanaman yang sesuai berdasarkan unsur hara dan keadaan lingkungan. Sehingga dengan demikian Tujuan dari proyek ini ialah untuk merekomendasikan tanaman yang optimal dibudidayakan oleh petani berdasarkan beberapa parameter dan membantu mereka membuat keputusan sebelum budidaya dapat terpenuhi dan bisa membantu menyelesaikan masalah yang dialami para petani terkait gagal panen.

DAFTAR PUSTAKA

- Borman, R. I. & Wati, M., 2020. Penerapan Data Maining Dalam Klasifikasi Data Anggota Kopdit Sejahtera Bandarlampung Dengan Algoritma Naïve Bayes. *Jurnal Ilmiah Fakultas Ilmu Komputer*, 9(1), pp. 25-34.
- Eko, P., 2012. *Data Mining : Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: CV Andi Offset.
- Gandhi, B. S., Megawaty, D. A. & Alita, D., 2020. Aplikasi Monitoring dan Penentuan Peringkat Kelas Menggunakan Naive Bayes Classifier. *Informatika dan Rekayasa Perangkat Lunak (Jatika)*, 2(1), pp. 54-63.
- Jumeilah, F. S., 2017. Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian. *Jurnal Resti*, 1(1), pp. 19-25.
- Polinema, T. A. M. L., 2021. Modul : Machine Learning. : *Machine Learning* . s.l.:s.n.
- Putri, S. E. Y. & Surahman, A., 2020. Penerapan Model Naive Bayes Untuk Memprediksi Potensi Pendaftaran Siswa Di Smk Taman Siswa Teluk Betung Berbasis Web. *Jurnal Informatika dan Rekayasa Perangkat Lunak (Jatika)*, 1(1), pp. 93-99.
- Qadrini, L., Seppewali, A. & Aina, A., 2021. DECISION TREEDAN ADABOOSTPADA KLASIFIKASI PENERIMA PROGRAM BANTUAN SOSIAL. *JIP : Jurnal Informasi Pendidikan*, 2(7), pp. 1959-1966.
- Schouten, K., Francisar, F. & Dekker, R., 2016. An Information gain-Driven Feature Study for Aspect-Based Sentiment Analysis. *Natural Language Processing and Information Systems*, pp. 48-59
- Anon. 2022. "Indonesia Negara Agraris dan Maritim, tapi Banyak Petani dan Nelayan Belum Sejahtera." Sekretariat Kabinet Republik Indonesia. Diambil 21 Desember 2022 (<https://setkab.go.id/indonesia-negara-agraris-dan-maritim-tapi-banyak-petani-dan-nelayan-belum-sejahtera/>).
- Media, Kompas Cyber. 2020. "Antisipasi Dampak Perubahan Iklim, BMKG Ajak Petani Rakayasa Komoditas Halaman all." KOMPAS.com. Diambil 21 Desember 2022 (<https://www.kompas.com/sains/read/2020/08/05/133700123/antisipasi-dampak-perubahan-iklim-bmkg-ajak-petani-rakayasa-komoditas>).