

Sistem Deteksi Penyakit Covid-19 Berdasarkan Gejala Awal Menggunakan Algoritma Naïve Bayes Berbasis Android

Dede Kurniadi^{1*}, Asri Mulyani², Diar Nur Rizky³

^{1,2,3} Institut Teknologi Garut, Garut, Indonesia

Email: ^{1*} dede.kurniadi@itg.ac.id, ² asrimulyani@itg.ac.id, ³ 1806050@itg.ac.id

(Naskah masuk: 28 Jul 2023, direvisi: 20 Okt 2023, diterima: 24 Okt 2023)

Abstrak

Data penelitian terakhir menunjukkan bahwa masyarakat merasa takut untuk melakukan pemeriksaan ke instansi kesehatan akibat kurangnya pengetahuan Covid-19, sehingga menyebabkan ketidakpedulian dalam aktivitas sehari-hari terhadap dampak dari situasi penyakit Covid-19. Oleh karena itu dibutuhkan sebuah aplikasi sistem deteksi gejala awal penyakit Covid-19 berbasis *mobile*. Tujuan dari penelitian ini membuat aplikasi sistem deteksi penyakit Covid-19 dengan menerapkan metode pengklasifikasian Naïve Bayes sehingga mempermudah pengguna dalam melakukan tes mandiri gejala awal Covid-19. Metode perancangan yang digunakan adalah *Extreme Programming* (XP) yang terdiri dari *planning, analysis, design, implementation, dan maintenance*. Data yang digunakan terdiri dari 2 *dataset* yaitu *dataset* untuk pengklasifikasian penyakit Covid-19 dengan jumlah data sebanyak 44.453 dan *dataset* untuk pengklasifikasian varian Covid-19 berjumlah 128.769. Penelitian ini melakukan 2 kali pemodelan menggunakan *Split Data* dengan perbandingan 5:5 untuk klasifikasi penyakit Covid-19 dan perbandingan 3:7 untuk klasifikasi varian Covid-19. Hasil penelitian ini berhasil membangun Sistem deteksi penyakit Covid-19 berdasarkan gejala awal menggunakan algoritma *Naïve Bayes* berbasis android dan telah mampu memprediksi penyakit Covid-19 ke dalam 4 *class* dengan nilai *F1-Score* yaitu *Allergy* 0,98, *Cold* 0,61, *Covid* 0,56, dan *Flu* 0,95, serta gejala yang paling berpengaruh pada *class Allergy* yaitu CS13 (*Loss of taste*) dengan nilai 0,50, *class Cold* yaitu CS3 (*Tiredness*) dengan nilai 0,52, *class Covid* yaitu CS12 (*Difficulty breathing*) dengan nilai 0,51, dan *class Flu* yaitu CS19 (*Sneezing*) dengan nilai 0,53, sistem yang dibangun juga mampu memprediksi varian Covid-19 ke dalam 3 *class* dengan nilai *F1-Score* yaitu *alpha* 0,85, *delta* 0,78, dan *omicron* 0,93, serta gejala yang paling berpengaruh pada *class Alpha* yaitu VS3 (*Loss of appetite*) dengan nilai 0,74, *class Delta* yaitu VS12 (*Cough*) dengan nilai 0,87, dan *class Omicron* yaitu VS10 (*Sore throat*) dengan nilai 0,67, juga aplikasi berhasil dan dapat dirancang dengan pendekatan *Extreme Programming* (XP).

Kata Kunci: Sistem Deteksi, Covid-19, Naïve Bayes, *Extreme Programming*, Python.

Covid-19 Disease Detection System Based On Early Symptoms Using Android Android-Based Naïve Bayes Algorithm

Abstract

Some of the latest research data shows that people feel afraid to carry out examinations at health agencies due to a lack of knowledge of Covid-19, causing ignorance in daily activities about the impact of the Covid-19 disease. Therefore, we need an early symptom detection system application for mobile-based Covid-19 disease. This study aims to make a Covid-19 disease detection system application by implementing a classification using the Naïve Bayes method to make it easier for users to carry out self-tests for early symptoms of Covid-19. The research method used is Extreme Programming (XP), which consists of planning, analysis, design, implementation, and maintenance. The data consists of 2 datasets, namely the dataset for classifying Covid-19 disease with a total of 44,453 data and the dataset for classifying the variant Covid-19, totaling 128,769. This study conducted two modelings to produce a classification model for Covid-19 disease and Covid-19 variants using Split Data with a 5:5 ratio for the classification of Covid-19 disease and a 3:7 ratio for the classification of the Covid-19 variant. The results of this study succeeded in building a Covid-19 disease detection system based on early symptoms using the Android-based Naïve Bayes algorithm and have been able to predict Covid-19 disease into 4 classes with F1-Score values, namely Allergy 0.98, Cold 0.61, Covid 0.56, and Flu 0.95, as well as the most influential symptom in the Allergy class, namely CS13 (Loss of taste) with a value of 0.50, class Cold, namely CS3 (Tiredness) with a value of 0.52, class Covid, namely CS12 (Difficulty breathing) with a

value of 0.51, and class Flu, namely CS19 (Sneezing) with a value of 0.54, the system built is also able to predict the variant Covid-19 into 3 classes with F1-Score values, namely alpha 0.85, delta 0.78, and omicron 0.93, as well as the most influential symptoms in the Alpha class, namely VS3 (Loss of appetite) with a value of 0.75, Delta class namely VS12 (Cough) with a value of 0.88, and the Omicron class, namely VS10 (Sore throat) with a value of 0.68, also the application is successful and can be designed with the Extreme Programming (XP) approach.

Keywords: Detection System, Covid-19, *Naïve Bayes*, Extreme Programming, Python.

I. PENDAHULUAN

Dengan pertumbuhan dan perkembangan penyakit Covid-19 belakangan ini banyak sekali masyarakat yang merasa takut dengan hal tersebut, dan merasa takut untuk melakukan pemeriksaan ke instansi atau ahli terkait. Misalnya pada penelitian yang dilakukan oleh Amu, dkk menunjukkan bahwa di Kabupaten Gorontalo 52,64% masyarakat kabupaten tersebut mengalami kecemasan pada saat melakukan tes *swab* dan 47,56 % menolak untuk melakukan tes *swab* [1], sehingga dapat menyebabkan keacuhan dalam aktivitas sehari-hari terhadap dampak dari situasi penyakit Covid-19. Oleh sebab itu, dibutuhkan aplikasi *mobile* yang bisa diakses di mana saja, karena perangkat *mobile* ini tidak terbatas oleh area dan dapat diakses kapan saja [2], serta aplikasi yang dibangun harus mampu mendeteksi penyakit Covid-19, dengan adanya sistem *machine learning* yang diterapkan dalam perangkat *mobile*, hal ini bisa menjadi solusi sebagai penanggulangan rasa takut masyarakat dalam melakukan tes mandiri untuk mendeteksi gejala awal dari penyakit covid. Algoritma yang umum digunakan dalam pembuatan aplikasi *machine learning* mengenai Covid-19 ini di antaranya, *Logistic Regression*, *Artificial Neural Network* (ANN), *Convolutional Neural Network* (CNN), *Linear Regression*, *K-Means*, *K-Nearest Neighbors* (K-NN), *Naïve Bayes* [3]

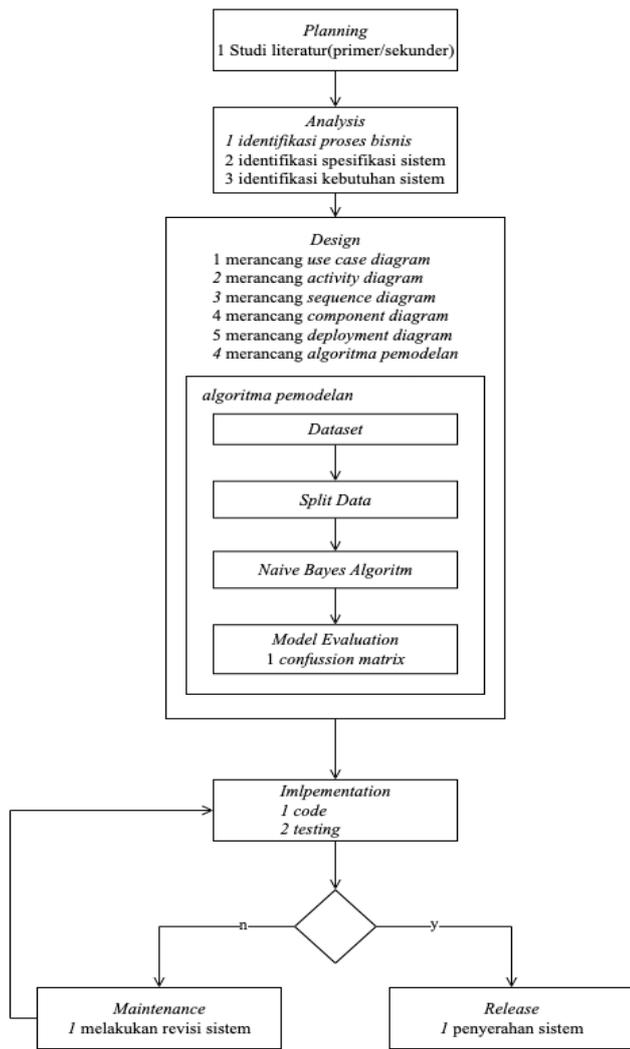
Adapun beberapa penelitian yang telah dilakukan peneliti sebelumnya diantaranya, penelitian yang dilakukan oleh Al-Zubidi, dkk [4] yang membahas tentang penerapan dua algoritma sistem pakar yaitu *Back Propagation* (BP) dan *Fuzzy C-Mean* (FCM) untuk mencari tahu jenis virus serta membedakan antara *influenza* dengan Covid-19 dengan hasil keakuratan sistem 89%. Penelitian berikutnya dilakukan oleh Fahidra dan Al-Amin [5] yang membahas mengenai penerapan algoritma *Certainty Factor* (CF) pada sistem pakar dalam menentukan derajat keyakinan terinfeksi Covid-19 dalam 3 kategori yaitu positif, negatif, dan terindikasi positif. Selanjutnya penelitian yang dilakukan oleh Al-Hakim, dkk [6] yang membahas tentang gejala Covid-19 yang ambigu dan tidak spesifik karena banyak penyakit pernafasan yang memiliki gejala yang sama, hasil dari penelitian ini memperoleh diagnosa terpapar Covid-19 dari pengguna dengan status ODP 99,96%, PDP 99,99%, positif 99,99%, dan negatif 99,76%. Sedangkan Penelitian yang dilakukan oleh Etikasari, dkk [7] membahas mengenai Komunikasi Informasi dan Edukasi Masyarakat (KIE) tentang Covid-19 supaya dapat menjadi sistem informasi dini Covid-19 yang dapat diakses oleh masyarakat dengan metode pengembangan *Design Sprint*, hasil dari penelitiannya menunjukkan aplikasi dapat diselesaikan dengan cepat menggunakan metode *Design Sprint*

yang menerapkan *rule base*, serta sistem juga mampu mengimplementasikan *rule based* dengan memberikan solusi setiap hasil deteksi. Pada penelitian yang dilakukan oleh Bhatia, dkk [8] mengenai desain mekanisme untuk memprediksi penyakit Covid-19, menyimpulkan bahwa teknik *data mining* dengan pendekatan *Naïve Bayes Classifier* dapat diterapkan ke dalam permasalahan prediksi penyakit Covid-19. Penelitian berikutnya dilakukan oleh Akbar, dkk [9] menganalisis masyarakat sejahtera dalam konten media sosial dalam menganggapi bagaimana pemerintah menyikapi pandemi Covid-19, hasilnya menunjukkan bahwa masyarakat sudah mendorong dan mendukung pemerintah untuk mengatasi pandemi yang sedang terjadi, namun pemerintah belum menggunakan media sosial sebagai media untuk berkomunikasi dengan publik. Berikutnya penelitian yang dilakukan oleh Mulyani, dkk [10], membahas mengenai minimnya informasi yang diterima oleh masyarakat mengenai sebaran wilayah yang terpapar virus Covid-19, sehingga tingkat kesadaran sebagian besar masyarakat akan Covid-19 masih terbilang rendah yang kemudian berimbas pada peningkatan angka penularan yang menjadi tinggi, hasil dari penelitian ini yaitu perancangan sistem informasi geografis pemetaan sebaran kasus Covid-19 yang mampu menyajikan informasi tentang peta sebaran kasus Covid-19.

Berdasarkan penelitian rujukan yang telah diuraikan maka penulis akan melakukan penelitian mengenai Sistem Deteksi Penyakit Covid-19 Berdasarkan Gejala Awal Menggunakan Algoritma *Naïve Bayes* Berbasis Android. Adapun alasan penerapan algoritma *Naïve Bayes* untuk klasifikasinya karena berdasarkan penelitian sebelumnya menunjukkan bahwa algoritma ini tergolong cocok dan cukup banyak digunakan dalam bidang kesehatan.

II. METODE PENELITIAN

Kerangka pemikiran dalam penelitian ini merupakan tahapan aktivitas dalam proses perancangan sebuah sistem yang disajikan dalam bentuk diagram, supaya dapat dengan mudah dibaca serta memahami arah secara garis besar alur logika berjalannya sebuah penelitian. Tahapan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Penelitian

Gambar 1 menunjukkan kerangka penelitian yang dilakukan dan merujuk pada tahapan metode *Extreme Programming (XP)* yaitu dimulai dari tahapan *planning, analysis, design, dan implementation and maintenance* [11]. Pada tahapan *design* dilakukan pemodelan menggunakan metode *Naïve Bayes* untuk menghasilkan prediksi penyakit dan *model performance* terbaik yang akan diimplementasikan ke dalam aplikasi.

1. Naïve Bayes

Metode *Naïve Bayes* merupakan metode yang menggunakan metode perhitungan probabilitas dan statistik. Keuntungan klasifikasi *Naïve Bayes* adalah metode ini hanya membutuhkan jumlah data pelatihan yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Langkah-langkah perhitungan dengan metode *Naïve Bayes* sebagai berikut : a) Menentukan kategori (penyakit) yang muncul berdasarkan data latih; b) Menghitung nilai probabilitas penyakit dan gejala; c) Menghitung nilai *bayes* berdasarkan probabilitas penyakit dan gejala yang

timbul; d) Menentukan persentase nilai prediksi kategori. Dalam proses rumus probabilitas *Naïve Bayes* sebagai berikut:

$$P(C|X) = \frac{P(C|H)P(C)}{P(C)} \tag{1}$$

Keterangan dari rumus tersebut adalah a) C yaitu data *class* yang belum diketahui; b) H yaitu hipotesis data suatu *class* yang spesifik; c) P(H|C) yaitu probabilitas hipotesis H berdasarkan kondisi X (*posterior* probabilitas); d) P(C) yaitu probabilitas hipotesis H (*prior* probabilitas); e) P(C|H) yaitu probabilitas X berdasarkan kondisi H; f) P(C) yaitu probabilitas dari X [12].

2. Confussion Matrix

Confusion matrix adalah tabel yang digunakan untuk menentukan kinerja algoritma klasifikasi. *Confusion matrix* memvisualisasikan dan meringkas kinerja algoritma klasifikasi [13]. Sebuah *confusion matrix* ditunjukkan pada Tabel 1.

Tabel 1. *Confusion Matrix*

	Predicted Value	
Actual value	Negative	Positive
Negative	TP	FN
Positive	FP	TN

Confusion matrix terdiri dari empat karakteristik dasar (angka) yang digunakan untuk menentukan metrik pengukuran pengklasifikasi. Keempat nomor ini adalah:

1. TP (*True Positive*): TP mewakili jumlah data yang telah diklasifikasikan dengan benar untuk memiliki nilai negatif.
2. TN (*True Negative*): TN mewakili jumlah data yang diklasifikasikan dengan benar bernilai positif.
3. FP (*False Positive*): FP mewakili jumlah data yang salah klasifikasi dengan nilai tersebut tetapi sebenarnya mereka memiliki nilai positif. FP juga dikenal sebagai kesalahan Tipe I.
4. FN (*False Negative*): FN mewakili jumlah data yang salah diklasifikasikan memiliki nilai positif tetapi sebenarnya data tersebut memiliki nilai negatif. FN juga dikenal sebagai kesalahan Tipe II.

Kinerja metrik suatu algoritma adalah *accuracy, precision, recall, dan FI-Score*, yang dihitung berdasarkan TP, TN, FP, dan FN yang disebutkan di atas.

Akurasi algoritma direpresentasikan sebagai rasio data yang diklasifikasikan dengan benar (TP+TN) dengan jumlah total data (TP+TN+FP+FN).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{2}$$

Ketepatan suatu algoritma direpresentasikan sebagai rasio data dengan nilai yang diklasifikasikan dengan benar negatif (TP) dengan total data yang diprediksi memiliki nilai positif (TP+FP).

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

Recall didefinisikan sebagai rasio benar negatif (TP) dibagi dengan jumlah data yang benar-benar memiliki nilai negatif.

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

Recall adalah berapa banyak data yang telah diklasifikasikan memiliki nilai tersebut. *F1-Score* juga dikenal sebagai *F Measure*. *F1-Score* menyatakan keseimbangan antara *precision* dan *recall*.

$$F1-Score = \frac{TP(2 * (Precision * Recall))}{(Precision + Recall)} \tag{5}$$

3. Sumber Data

Sumber data yang digunakan untuk kebutuhan pengklasifikasian penyakit alergi, penyakit demam, penyakit Covid-19, dan penyakit flu dalam penelitian ini bersumber dari Kaggle, berikut adalah link untuk mendapatkan *dataset*-nya https://www.kaggle.com/datasets/walterconway/covid-flu-cold-symptoms?select=large_data.csv. Adapun data *head* dan *tail*-nya dapat dilihat pada Tabel 2.

Tabel 2. *Dataset* Gejala Penyakit

No.	Cough	Muscle aches	...	Sneezing	Pink eye	Type
1	0	0	...	0	1	Allergy
2	0	0	...	1	1	Allergy
3	0	1	...	0	1	Allergy
4	0	0	...	1	1	Allergy
5	0	0	...	1	1	Allergy
...
44449	1	0	...	1	0	Flu
44450	1	1	...	1	0	Flu
44451	0	0	...	1	0	Flu
44452	0	0	...	0	0	Flu
44453	1	1	...	0	0	Flu

Berikut adalah rincian data yang diperoleh beserta atribut yang digunakan meliputi:

- 1) Data gejala yaitu: *Cough, Muscle aches, Tiredness, Sore throat, Runny nose, Stuffy nose, Fever, Nausea, Vomiting, Diarrhea, Shortness of breath, Difficulty breathing, Loss of appetite, Loss of smell, Itchy nose, Itchy eyes, Itchy mouth, Itchy inner ear, Sneezing, dan Pink eye*;
- 2) Data penyakit yang dialami yaitu: *Type*;
- 3) Menggunakan 21 atribut dengan jumlah data sebanyak 44.453 yang terdiri dari:
 - a. Berpenyakit alergi: 16.381
 - b. Berpenyakit demam: 1.024

- c. Berpenyakit Covid-19: 2.048
- d. Berpenyakit flu: 25.000

Sedangkan untuk sumber data yang digunakan untuk kebutuhan pengklasifikasian varian Covid-19 dalam penelitian ini bersumber dari data *generate* berdasarkan persentase gejala yang berpengaruh, berikut adalah tautan untuk mendapatkan *dataset* tersebut <https://drive.google.com/file/d/1NYTINmf5LTM9WXhX0yxVwF8bJS482uV/view?usp=sharing>. Adapun data *head* dan *tail*-nya dapat dilihat pada Tabel 3

Tabel 3. *Dataset* Gejala Varian Covid-19

No.	Fatigue	Headache	Loss of appetite	...	Cough	Fever	Other	Type
1	1	1	1	...	0	1	1	omicron
2	0	0	0	...	0	1	0	omicron
3	0	1	0	...	1	1	1	omicron
4	1	0	0	...	0	1	1	omicron
5	0	0	0	...	1	0	0	omicron
...
128765	0	1	1	...	1	1	0	omicron
128766	0	1	0	...	0	0	1	alpha
128767	1	0	0	...	1	0	1	alpha
128768	0	0	1	...	0	0	1	alpha
128769	1	0	0	...	0	0	1	omicron

- Berikut adalah rincian data beserta atribut yang diperoleh:
- 1) Data gejala yaitu: *Fatigue, Headache, Loss of appetite, Irritability or confusion, General weakness, Myalgia or bodypain, Diarrhae, Nausea or vomiting, Shortness of breath, Sore throat, Chills, Cough, History of fever, Fever,* dan *Other*;
 - 2) Data penyakit yang dialami yaitu: *Type*;
 - 3) Menggunakan 16 atribut dengan jumlah data sebanyak 128.769 yang terdiri dari:
 - a. Berpenyakit *alpha*: 34.375
 - b. Berpenyakit *delta*: 13.949
 - c. Berpenyakit *omicron*: 80.445

II. HASIL DAN PEMBAHASAN

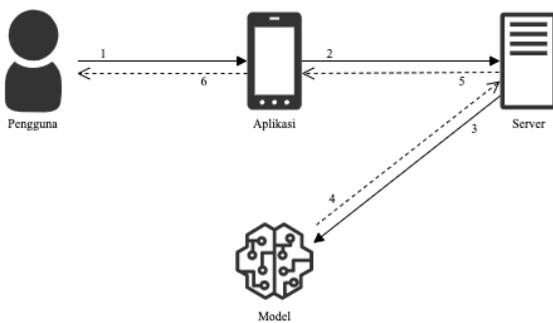
A. Hasil Penelitian

1. Planning

Pada tahapan pertama ini diawali dengan pengumpulan data yang berasal dari studi literatur primer ataupun sekunder kemudian menghasilkan rumusan masalah dan tujuan penelitian yaitu Mengklasifikasikan penyakit *Coronavirus Disease 2019 (Covid-19)*, Mengklasifikasikan varian *Coronavirus Disease 2019 (Covid-19)* berdasarkan gejala awal dengan metode *Naïve Bayes* dan membangun aplikasi Android yang bisa mendeteksi penyakit *Coronavirus Disease 2019 (Covid-19)* berdasarkan gejala awal.

2. Analysis

Setelah melakukan tahap *planning* maka dibuatlah solusi penyelesaian dengan sistem yang akan dibuat. Sistem akan menggunakan metodologi perancangan XP, sedangkan untuk metode pengklasifikasiannya menggunakan algoritma *Naïve Bayes* yang akan diimplementasikan pada sistem berbasis android.



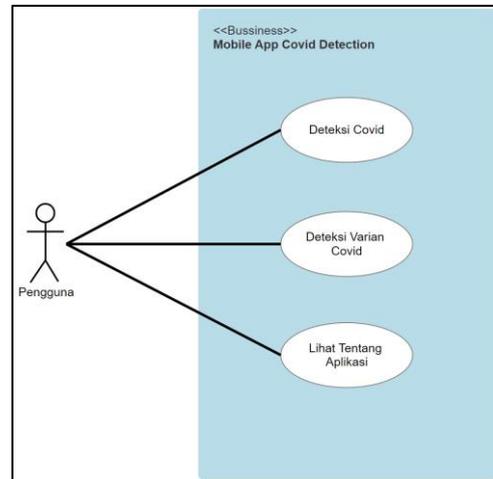
Gambar 2. Alur Sistem Aplikasi

Berikut penjelasan tentang Gambar 2 1) Pengguna mengisi gejala-gejala yang telah ditentukan aplikasi 2) Setelah pengguna mengisi gejala-gejala yang dialami, maka aplikasi akan mengirim data gejala pengguna ke server 3) Server mengirimkan data gejala untuk diproses dalam tahap pemodelan 4) Setelah pemodelan mengeluarkan hasilnya kemudian hasil dikirim kembali ke server 5) Server mengirimkan hasil analisis pemodelan ke aplikasi 6) Pengguna mendapatkan hasil persentase penyakit berdasarkan gejala-gejala yang dialaminya.

3. Design

Setelah menentukan proses bisnis, kebutuhan, dan spesifikasi sistem, langkah selanjutnya adalah membuat dokumentasi untuk *programmer* seperti membuat *use case diagram, activity diagram, sequence diagram, class diagram, component diagram,* dan desain algoritma *Naïve Bayes*.

a. Use Case Diagram

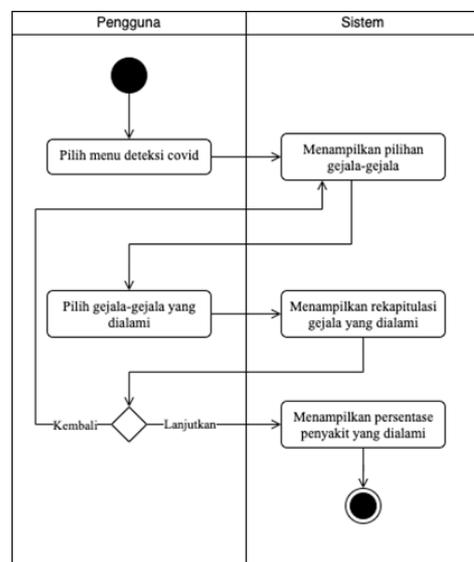


Gambar 3. Use Case Diagram

Gambar 3 menggambarkan beberapa aktivitas yang dapat dilakukan pada aplikasi yang dibuat yaitu pengguna dapat melakukan pendeteksian Covid-19 dengan *input* gejala yang dialami, pendeteksian varian Covid-19 dengan *input* gejala yang dialami, serta melihat info tentang aplikasi.

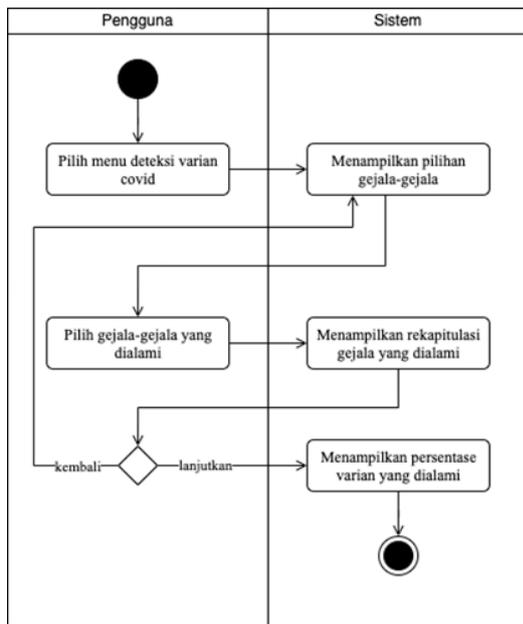
b. Activity Diagram

Activity diagram merupakan sebuah pemodelan aktivitas dari *use case* yang telah dibuat. Terdapat dua *activity diagram* pada penelitian ini yaitu *activity diagram* deteksi Covid-19 dan *activity diagram* deteksi varian Covid-19.



Gambar 4. Activity Diagram Deteksi Covid-19

Gambar 4 menunjukkan perancangan *activity diagram* deteksi Covid-19 di mana proses ini dimulai dengan prakondisi pengguna sudah ada pada halaman utama aplikasi, pengguna akan mulai melakukan aktivitas dengan menekan tombol menu deteksi Covid-19. Selanjutnya sistem akan memunculkan gejala-gejala, kemudian pengguna mengkonfirmasi gejala mana saja yang kiranya pengguna alami pada saat itu. Lalu sistem akan memunculkan *modal* rekapitulasi gejala yang dialami oleh penggunadan menekan tombol lanjutkan sebagai tanda bahwa pengguna mengkonfirmasi gejala-gejala tersebut. Terakhir sistem akan menampilkan persentase penyakit yang pengguna alami.



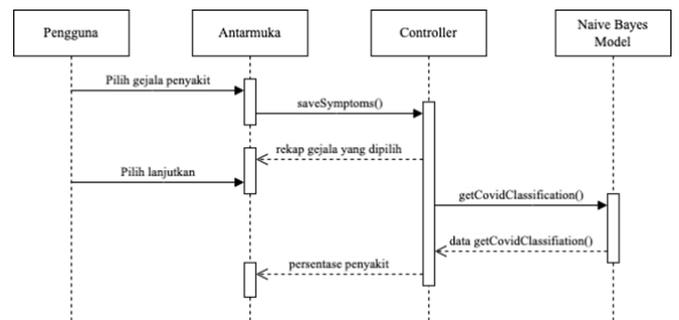
Gambar 5. *Activity Diagram* Deteksi Varian Covid-19

Gambar 5 menunjukkan perancangan *activity diagram* Deteksi Varian Covid-19 di mana proses ini akan dimulai dengan prakondisi pengguna sudah ada pada halaman utama aplikasi, kemudian pengguna akan mulai melakukan aktivitas dengan menekan tombol menu deteksi varian Covid-19, setelah itu sistem akan memunculkan gejala-gejala kemudian pengguna akan mengkonfirmasi gejala mana saja yang kiranya pengguna alami pada saat itu, setelah itu sistem akan memunculkan *modal* rekapitulasi gejala yang dialami oleh pengguna, kemudian pengguna akan menekan lanjutkan sebagai tanda bahwa pengguna mengkonfirmasi gejala-gejala tersebut, setelah itu sistem akan menampilkan persentase varian Covid-19 yang pengguna alami.

c. *Sequence Diagram*

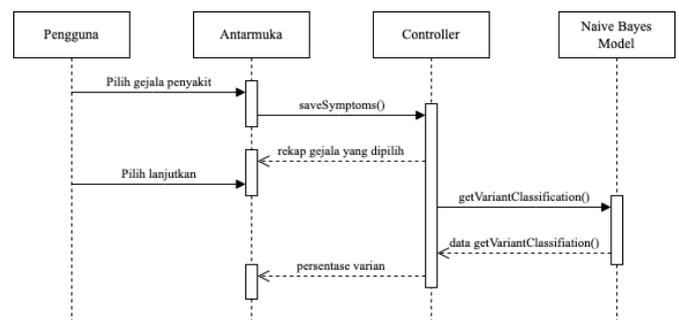
Sequence Diagram menggambarkan kegiatan aktor dengan sistem serta menjelaskan cara kerja sistem itu sendiri. Terdapat dua *sequence diagram* yang telah dirancang seperti pada Gambar 6 dan Gambar 7. Pada Gambar 6 menunjukkan *sequence diagram* deteksi Covid-19 saat pertama melakukan aktivitas pendeteksian penyakit Covid-19 pengguna akan memilih satu-persatu gejala penyakit yang pengguna alami, secara bersamaan sistem akan menyimpan gejala yang

pengguna konfirmasi pada *temporary data* yang telah disiapkan dalam *controller* dengan *method saveSymptoms()* untuk kemudian nanti ditampilkan pada *modal* rekapitulasi gejala yang pengguna alami, setelah sistem menampilkan *modal* rekapitulasi gejala, selanjutnya pengguna akan memilih lanjutkan untuk menuju ke tahap pengklasifikasian penyakit Covid-19, secara bersamaan sistem dengan *controller* yang telah disediakan akan mengirimkan gejala-gejala yang pengguna alami dengan *method asynchronous* yaitu *getCovidClassification()*, yang akan mengembalikan data berupa hasil pengklasifikasian, setelah proses *asynchronous* tersebut berhasil mengembalikan data, kemudian data tersebut akan ditampilkan pada pengguna lewat *modal* yang baru.



Gambar 6. *Sequence Diagram* Deteksi Covid-19

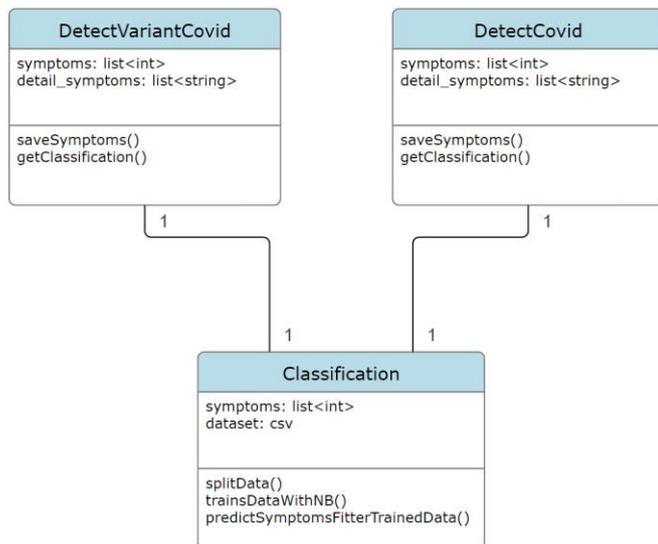
Gambar 7 menunjukkan *sequence diagram* deteksi varian Covid-19 dimana saat pertama melakukan aktivitas pendeteksian varian Covid-19 pengguna akan memilih satu-persatu gejala penyakit yang pengguna alami, secara bersamaan sistem akan menyimpan gejala yang pengguna konfirmasi pada *temporary data* yang telah disiapkan dalam *controller* dengan *method saveSymptoms()*, untuk kemudian nanti ditampilkan pada *modal* rekapitulasi gejala yang pengguna alami, setelah sistem menampilkan *modal* rekapitulasi gejala, selanjutnya pengguna akan memilih lanjutkan untuk menuju ke tahap pengklasifikasian varian Covid-19, secara bersamaan sistem dengan *controller* yang telah disediakan akan mengirimkan gejala-gejala yang pengguna alami dengan *method asynchronous* yaitu *getVariantClassification()*, yang kemudian *method* tersebut akan mengembalikan data berupa hasil pengklasifikasian, setelah proses *asynchronous* tersebut berhasil mengembalikan data, kemudian data tersebut akan ditampilkan pada pengguna lewat *modal* yang baru.



Gambar 7. *Sequence Diagram* Deteksi Varian Covid-19

d. Class Diagram

Class diagram sering disebut juga dengan desain pemrograman karena merepresentasikan struktur dari program itu sendiri. Gambar 8 menunjukkan visualisasi umum tentang class yang ada pada sistem yang dibangun, di mana pada sistem ini terdapat 3 class utama yaitu DetectVariantCovid, class ini memuat informasi tentang komponen yang ada di dalamnya diantaranya ada symptoms dengan tipe data list integer, detail_symptoms dengan tipe data list string, adapun fungsi didalamnya diantaranya adalah saveSymptoms() untuk menyimpan gejala yang dikonfirmasi oleh pengguna, dan getClassification() untuk mendapatkan hasil pemodelan.

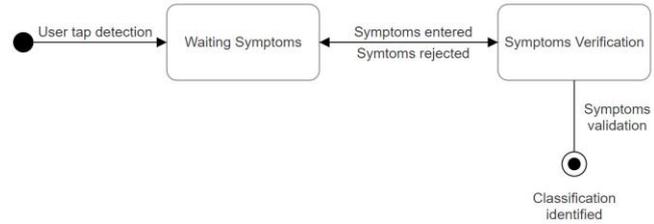


Gambar 8. Class Diagram

Selanjutnya DetectCovid, di mana class ini memuat informasi tentang komponen yang ada di dalamnya diantaranya ada symptoms dengan tipe data list integer, detail symptoms dengan tipe data list string. Adapun fungsi didalamnya diantaranya adalah saveSymptoms() untuk menyimpan gejala yang dikonfirmasi oleh pengguna, dan getClassification() untuk mendapatkan hasil pemodelan. Terakhir class Classification, pada class ini memuat komponen data symptoms dengan tipe data list integer, kemudian ada dataset yaitu dengan tipe .csv, adapun fungsi pada class ini adalah, splitData(), yaitu untuk memisahkan dataset menjadi data training dan data testing, trainDataWithNB(), yaitu untuk melatih data training dengan algoritma Naïve Bayes, dan yang terakhir predictSymptomsFittedTrainedData(), dimana fungsi ini adalah untuk memprediksi gejala yang telah dimasukkan oleh pengguna berdasarkan data yang telah dilatih

e. State Diagram

State diagram digunakan untuk merepresentasikan kondisi sistem atau bagian dari sistem pada waktu yang terbatas, dan mewakili perilaku kondisi dengan transisi keadaan terbatas pula.

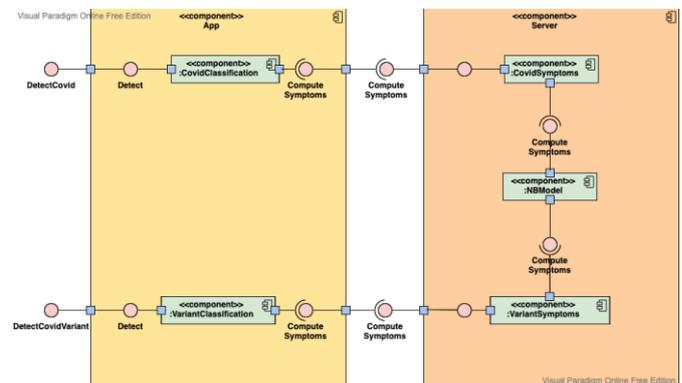


Gambar 9. State Diagram

Gambar 9 menunjukkan perubahan-perubahan state yang dialami oleh sistem ketika pengguna melakukan pendeteksian penyakit Covid-19 atau varian Covid-19, dimana untuk state awal dari pendeteksian ini diawali dengan sistem yang menunggu input gejala dari pengguna setelah menerima input sistem melakukan verifikasi gejala, kemudian mendapatkan persentase pengklasifikasian penyakit atau varian.

f. Component Diagram

Component diagram menggambarkan struktur serta keterkaitan di antara komponen software, termasuk dependency. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain

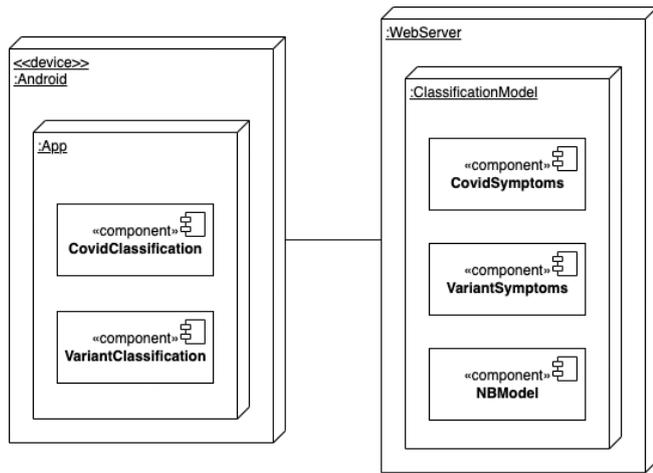


Gambar 10. Component Diagram

Gambar 10 menunjukkan komponen-komponen yang ada pada sistem yang akan dibuat, dimana terdapat 2 sistem yang digunakan yaitu App dan juga Server, proses pendeteksian penyakit Covid-19 terdapat pada sistem App setelah itu modul tersebut memerlukan gejala-gejala penyakit serta proses pemodelan yang dimiliki oleh sistem yang ada pada server, begitu pun dengan pendeteksian varian Covid-19

g. Deployment Diagram

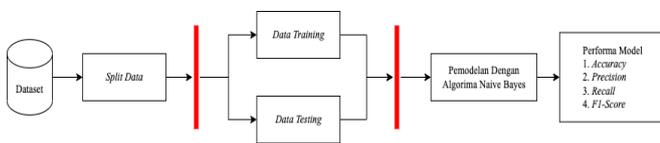
Deployment diagram merepresentasikan bagaimana sistem di-deploy dalam sebuah infrastruktur sistem, di mana setiap komponen akan digambarkan sesuai letak ia disimpan



Gambar 11. *Deployment Diagram*

Gambar 11. menunjukkan bahwa komponen pengklasifikasian terdapat pada aplikasi android, sedangkan untuk pemodelannya ada pada *webservice*.

h. Tahapan Pemodelan



Gambar 12. Tahapan Pemodelan

Gambar 12. menunjukkan proses pemodelan dari penelitian ini dengan dua pengklasifikasian, yang pertama adalah pengklasifikasian Covid-19 dan yang ke-dua adalah pengklasifikasian varian Covid-19. Adapun prosesnya cukup sama namun yang menjadi perbedaan ada dalam tahap pembagian datanya, di mana untuk pengklasifikasian penyakit Covid-19 menggunakan rasio 5:5 dari total 44.453 data, sedangkan dalam pengklasifikasian varian Covid-19 menggunakan rasio 3:7 dari total 128.769 data. Setelah melakukan pemisahan data kemudian dilakukan pemodelan menggunakan algoritma *Naïve Bayes*, berikutnya adalah melakukan evaluasi model menggunakan *confussion matrix* yaitu *accuracy, precisioun, recall, dab f1-score*, adapun *tools* yang dipakai dalam pemodelan ini diantaranya: *Python, Scikit-Learn, Jupyter, Numpy, Pandas, Matplotlib*, dan *Flask* berikut adalah penjelasan lebih lanjut mengenai Gambar 11.

i. *Dataset*

Untuk memudahkan tahap pemodelan maka *dataset* yang telah didapatkan akan diidentifikasi yaitu dengan cara mengidentifikasi jenis *dataset* dan menentukan atribut yang digunakan

Tabel 4. Atribut *Dataset* Pengklasifikasian Covid-19

Atribut	Tipe Data	Indikator
<i>Cough</i>	<i>Binominal</i>	0 dan 1
<i>Muscle aches</i>	<i>Binominal</i>	0 dan 1
<i>Tiredness</i>	<i>Binominal</i>	0 dan 1

<i>Sore throat</i>	<i>Binominal</i>	0 dan 1
<i>Runny nose</i>	<i>Binominal</i>	0 dan 1
<i>Stuffy nose</i>	<i>Binominal</i>	0 dan 1
<i>Fever</i>	<i>Binominal</i>	0 dan 1
<i>Nausea</i>	<i>Binominal</i>	0 dan 1
<i>Vomiting</i>	<i>Binominal</i>	0 dan 1
<i>Diarrhea</i>	<i>Binominal</i>	0 dan 1
<i>Shortness of breath</i>	<i>Binominal</i>	0 dan 1
<i>Difficulty breathing</i>	<i>Binominal</i>	0 dan 1
<i>Loss of taste</i>	<i>Binominal</i>	0 dan 1
<i>Loss of smell</i>	<i>Binominal</i>	0 dan 1
<i>Itchy nose</i>	<i>Binominal</i>	0 dan 1
<i>Itchy eyes</i>	<i>Binominal</i>	0 dan 1
<i>Itchy mouth</i>	<i>Binominal</i>	0 dan 1
<i>Itchy inner ear</i>	<i>Binominal</i>	0 dan 1
<i>Sneezing</i>	<i>Binominal</i>	0 dan 1
<i>Pink eye</i>	<i>Binominal</i>	0 dan 1
<i>Type</i>	<i>Polynomial</i>	<i>Allergy, Cold, Covid, dan Flu</i>

Berdasarkan pada Tabel 4 terdapat 20 atribut dengan tipe data *binominal* pada atribut *Cough, Muscle aches, Tiredness, Sore throat, Runny nose, Stuffy nose, Fever, Nausea, Vomiting, Diarrhea, Shortness of breath, Difficulty breathing, Loss of taste, Loss of smell, Itchy nose, Itchy eyes, Itchy mouth, Itchy inner ear, Sneezing, dan Pink eye*. Terdapat 1 data *polynomial* pada atribut *Type*.

Berdasarkan pada Tabel 5. terdapat 15 atribut dengan tipe data *binominal* pada atribut *Fatigue, Headache, Loss of appetite, Irritability or confusion, General weakness, Myalgia or body pain, Diarrhea, Nausea or vomiting, Shortness of breath, Sore throat, Chills, Cough, History of fever, Fever, dan Other*. Terdapat 1 data *polynomial* pada atribut *Type*.

Tabel 5. Atribut *Dataset* Pengklasifikasian Varian Covid-19

Atribut	Tipe Data	Indikator
<i>Fatigue</i>	<i>Binominal</i>	0 dan 1
<i>Headache</i>	<i>Binominal</i>	0 dan 1
<i>Loss of appetite</i>	<i>Binominal</i>	0 dan 1
<i>Irritability or confusion</i>	<i>Binominal</i>	0 dan 1
<i>General weakness</i>	<i>Binominal</i>	0 dan 1
<i>Myalgia or body pain</i>	<i>Binominal</i>	0 dan 1
<i>Diarrhea</i>	<i>Binominal</i>	0 dan 1
<i>Nausea or vomiting</i>	<i>Binominal</i>	0 dan 1
<i>Shortness of breath</i>	<i>Binominal</i>	0 dan 1
<i>Sore throat</i>	<i>Binominal</i>	0 dan 1
<i>Chills</i>	<i>Binominal</i>	0 dan 1
<i>Cough</i>	<i>Binominal</i>	0 dan 1
<i>History of fever</i>	<i>Binominal</i>	0 dan 1
<i>Fever</i>	<i>Binominal</i>	0 dan 1
<i>Other</i>	<i>Binominal</i>	0 dan 1
<i>Type</i>	<i>Polynomial</i>	<i>Alpha, delta, dan omicron</i>

ii. *Split Data*

Split Data adalah teknik dalam membagi *dataset* menjadi data latih dan data uji berdasarkan rasio. Adapun perbandingan rasio yang digunakan pada pengklasifikasian penyakit alergi, penyakit demam, penyakit Covid-19, dan penyakit flu adalah dari rasio 1:9 sampai dengan rasio 9:1, hal ini dilakukan untuk mendapatkan hasil terbaik dari beberapa rasio *Split Data*, begitu pun sama halnya dengan perbandingan rasio yang digunakan pada pengklasifikasian varian Covid-19.

iii. Pemodelan dengan Algoritma *Naïve Bayes*

Algoritma pelatihan model menggunakan metode *Naïve Bayes* pada pengklasifikasian covid-19 dan varian covid-19 digunakan untuk mencari *model performance* terbaik. Adapun untuk *pseudo code* sebagai berikut:

In:
Dataset

Out:
Hasil Prediksi

Step:

1. Membaca *Dataset*;
2. Membagi *Dataset*;
3. Membaca *Data Training*;
4. Menghitung *mean* dan standar deviasi dari variabel gejala di setiap *class*;

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

5. **Foreach** x (gejala) **in class**
Hitung probabilitas gejala di setiap *class* (nilai *bayes*) yaitu dengan cara menghitung jumlah data yang sesuai dengan *class*, kemudian dibagi dengan jumlah data pada *class* tersebut;

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

6. Menghitung nilai probabilitas keseluruhan masing-masing *class*
 $P(C_i) = P(x_i|C_i) \cdot P(x_{i+1}|C_i) \dots P(x_n|C_i)$
7. Dapatkan nilai tertinggi dari setiap nilai probabilitas keseluruhan

Adapun untuk hasil pengujian berdasarkan rasio data dalam menemukan *model performance* terbaik seperti pada Tabel 6.

Tabel 6. Perbandingan *Split Data* Penyakit Covid-19

Rasio	Akurasi	Allergy			Cold			Covid			Flu		
		P	R	F	P	R	F	P	R	F	P	R	F
1:9	0,93	0,97	0,99	0,98	0,57	0,52	0,54	0,50	0,57	0,54	0,96	0,94	0,95
2:8	0,93	0,99	0,97	0,98	0,50	0,91	0,65	0,50	0,62	0,55	0,97	0,93	0,95
3:7	0,93	0,99	0,98	0,98	0,51	0,83	0,63	0,49	0,61	0,54	0,97	0,93	0,95
4:6	0,93	0,98	0,98	0,98	0,54	0,73	0,62	0,49	0,65	0,56	0,97	0,93	0,95
5:5	0,93	0,98	0,98	0,98	0,53	0,71	0,61	0,50	0,62	0,56	0,97	0,93	0,95
6:4	0,93	0,97	0,98	0,98	0,55	0,67	0,60	0,49	0,65	0,56	0,97	0,93	0,95
7:3	0,93	0,98	0,98	0,98	0,53	0,73	0,61	0,48	0,60	0,53	0,97	0,94	0,95
8:2	0,93	0,98	0,98	0,98	0,55	0,66	0,60	0,48	0,61	0,54	0,97	0,94	0,95
9:1	0,93	0,97	0,99	0,98	0,54	0,50	0,52	0,46	0,64	0,54	0,97	0,93	0,95

Keterangan: P = *Precision*, R = *Recall*, F = *F1-Score*

Berdasarkan Tabel 6. dari beberapa rasio yang digunakan pada penelitian ini nilai akurasi dari seluruh rasio sama yaitu 0,93, nilai presisi *Allergy* ada pada rasio 2:8 dan 3:7, nilai *recall Allergy* tertinggi ada pada rasio 1:9 dan 9:1, nilai *F1-Score Allergy* memiliki nilai yang sama pada setiap rasio, serta nilai presisi *Cold* ada pada rasio 1:9, nilai *recall Cold* tertinggi ada pada rasio 2:8, nilai *F1-Score Cold* tertinggi ada pada rasio

2:8, sedangkan nilai presisi Covid tertinggi ada pada rasio 1:9, 2:8, dan 5:5, nilai *recall Covid* tertinggi ada pada rasio 4:6 dan 6:4, nilai *F1-Score Covid* tertinggi ada pada rasio 4:6 dan 5:5, dan terakhir nilai presisi *Flu* tertinggi ada pada rasio selain 1:9 sampai 9:1, nilai *recall Flu* tertinggi ada pada rasio 1:9, 7:3, dan 8:2, nilai *F1-Score Flu* memiliki nilai yang sama pada semua rasio, oleh karena itu dapat disimpulkan bahwa rasio 5:5 adalah yang terbaik untuk mengklasifikasikan penyakit Covid-19 karena memiliki nilai presisi dan *F1-Score* yang tinggi.

Tabel 7. Perbandingan *Split Data* Varian Covid-19

Rasio	Akurasi	Alpha			Delta			Omicron		
		P	R	F	P	R	F	P	R	F
1:9	0,89	0,97	0,74	0,84	0,99	0,64	0,78	0,86	1,00	0,92
2:8	0,90	0,97	0,75	0,85	0,99	0,65	0,78	0,86	1,00	0,93

Rasio	Akurasi	Alpha			Delta			Omicron		
		P	R	F	P	R	F	P	R	F
3:7	0,90	0,97	0,76	0,85	0,99	0,64	0,78	0,86	1,00	0,93
4:6	0,90	0,97	0,75	0,85	0,99	0,64	0,78	0,86	1,00	0,93
5:5	0,89	0,97	0,75	0,85	0,99	0,63	0,77	0,86	1,00	0,93
6:4	0,89	0,97	0,76	0,85	1,00	0,62	0,76	0,86	1,00	0,93
7:3	0,89	0,97	0,76	0,85	0,99	0,62	0,76	0,86	1,00	0,93
8:2	0,89	0,97	0,76	0,85	0,99	0,62	0,76	0,86	1,00	0,93
9:1	0,89	0,97	0,75	0,84	0,99	0,62	0,76	0,86	1,00	0,92

Keterangan: P = *Precision*, R = *Recall*, F = *F1-Score*

Berdasarkan Tabel 7. dari beberapa rasio yang digunakan pada penelitian ini nilai akurasi dari tertinggi ada pada rasio 2:8, 3:7, dan 4:6, nilai presisi *Alpha* memiliki nilai yang sama pada semua rasio, nilai *recall Alpha* tertinggi ada pada rasio 3:7, 6:4, 7:3, dan 8:2, nilai *F1-Score Alpha* tertinggi ada pada semua rasio kecuali 1:9 dan 9:1, serta nilai presisi *Delta* ada pada rasio 6:4, nilai *recall Delta* tertinggi ada pada rasio 2:8, nilai *F1-Score Delta* tertinggi ada pada rasio 1:9 sampai 4:6, sedangkan nilai presisi *Omicron* memiliki nilai yang sama pada semua rasio, nilai *recall Omicron* memiliki nilai yang sama pada semua rasio, nilai *F1-Score Omicron* tertinggi ada pada rasio 2:8 sampai 8:2, oleh karena itu dapat disimpulkan bahwa rasio 3:7 adalah yang terbaik untuk mengklasifikasikan varian Covid-19 karena memiliki nilai akurasi yang tinggi.

4. *Implementation and Testing*

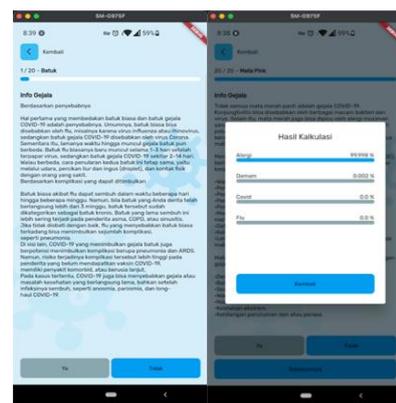
Tahapan implementasi adalah tahapan di mana *programmer* akan menerapkan hasil desain ke dalam bentuk struktur kode, dalam metode XP ini *programmer* akan melakukan *software* testing bersamaan dalam tahapan implementasi. Berikut adalah hasil implementasi pemodelan ke dalam aplikasi:



Gambar 13. Tampilan *Home Screen*

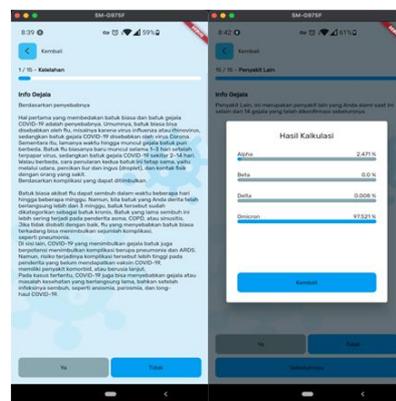
Gambar 13. merupakan tampilan halaman utama aplikasi hasil dari perancangan yang telah dibuat, pada halaman utama ini pengguna akan suguhkan beberapa menu yang bisa diakses

diantaranya adalah: Cek Gejala Covid, Cek Gejala Varian Covid, dan Tentang Aplikasi.



Gambar 14. Tampilan Deteksi Covid-19

Gambar 14. merupakan hasil tampilan dari perancangan yang telah dibuat, pada halaman ini pengguna akan disuguhkan tampilan gejala-gejala penyakit beserta dengan infonya yang disuguhkan dengan *mode stepper* dari 1 sampai 20 sesuai dengan gejala penyakit Covid-19, setelah pengguna mengkonfirmasi setiap gejala yang dialami, maka akan disuguhkan rekapitulasi gejala yang pengguna konfirmasi, setelah itu hasil dari prediksi akan ditampilkan



Gambar 15. Tampilan Deteksi Varian Covid-19

Gambar 15. merupakan hasil tampilan dari perancangan yang telah dibuat, pada halaman ini pengguna akan

disuguhkan tampilan gejala-gejala varian Covid-19 beserta dengan infonya, yang disuguhkan dengan *mode stepper* dari 1 sampai 14 sesuai dengan gejala varian Covid-19, setelah pengguna mengkonfirmasi setiap gejala yang dialami, maka akan disuguhkan rekapitulasi gejala yang pengguna konfirmasi, setelah itu hasil dari prediksi akan ditampilkan

Pada tahapan ini juga dilakukan pengujian *blackbox testing* yang diuji berdasarkan aktivitas yang ada dalam aplikasi, tahapan ini berfungsi untuk menguji apakah aplikasi berjalan seperti seharusnya dengan tanpa adanya kesalahan. Berikut adalah hasil pengujian *blackbox testing* aplikasi sistem deteksi seperti pada Tabel 8:

Tabel 8. *Blackbox Testing*

Aktivitas	Kelas Uji	Hasil yang diharapkan	Kesimpulan
Deteksi Covid-19	Menampilkan Gejala-gejala	Menampilkan gejala-gejala Covid-19	Sesuai
	Menampilkan hasil prediksi	Menampilkan hasil prediksi penyakit yang dialami berdasarkan gejala yang dipilih	Sesuai
Deteksi Varian Covid-19	Menampilkan Gejala-gejala	Menampilkan gejala-gejala varian Covid-19	Sesuai
	Menampilkan hasil prediksi	Menampilkan hasil prediksi varian Covid-19 yang dialami berdasarkan gejala yang dipilih	Sesuai
Lihat Tentang Aplikasi	Menampilkan informasi aplikasi	Menampilkan informasi tambahan mengenai aplikasi	Sesuai

Setelah melakukan *blackbox testing* selanjutnya melakukan pengujian aplikasi kepada beberapa pengguna yang pernah mengalami penyakit covid dan pengguna yang belum

pernah mengalami Covid-19, berikut adalah pengujian yang dilakukan dapat dilihat pada Tabel 9 dan 10.

Tabel 9. Deteksi Covid

User	Pernah Covid	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	Allergy (%)	Cold (%)	Covid (%)	Flu (%)	
User1	Ya	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
User2	Ya	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
User3	Ya	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
User4	Tidak	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
User5	Tidak	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Keterangan: S1 = *Cough*, S2 = *Muscleache*, ..., S20 = *Pink Eye*

Tabel 10. Deteksi Varian Covid

User	Pernah Covid	SV1	SV2	SV3	SV4	SV5	SV6	SV7	SV8	SV9	SV10	SV11	SV12	SV13	SV14	SV15	Alpha (%)	Beta (%)	Delta (%)	Omi cron (%)
User1	Ya	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	63,54	0	0,07	36,39
User2	Ya	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	74,6	0	2,1	23,3
User3	Ya	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	58,38	0	4,01	37,61

Keterangan: SV1 = *Fatigue*, SV2 = *Headache*, ..., SV15 = *Other*

B. Pembahasan Hasil

Hasil penelitian yang telah dilakukan telah berhasil menjawab 3 rumusan permasalahan yaitu 1) Pengklasifikasian penyakit Covid-19, penyakit demam, penyakit alergi dan flu berdasarkan gejala awal, dapat dilakukan dengan metode pemodelan *Naïve Bayes* dengan tahapannya yaitu penentuan *dataset*, *Split Data*, dan evaluasi model. Hasil *performance* terbaik dari pengklasifikasian penyakit Covid-19 didapatkan dari rasio 5:5 yaitu akurasi sebesar 0,93 dan nilai *f1-score*

sebesar 0,95; 2) Pengklasifikasian varian Covid-19 berdasarkan gejala awal dapat dilakukan dengan metode pemodelan *Naïve Bayes* dengan tahapannya yaitu penentuan *dataset*, *Split Data*, dan evaluasi model. Hasil *performance* terbaik dari pengklasifikasian varian Covid-19 didapatkan dari rasio 3:7 yaitu akurasi sebesar 0,90 dan nilai *F1-score* sebesar 0,93; 3) Penelitian ini berhasil membangun aplikasi Android dengan pemrograman *Dart* dan *framework Flutter*, serta menggunakan pengembangan *Extreme Programming* untuk

mendeteksi penyakit Covid-19 menggunakan metode *Naïve Bayes*.

Keselarasan penelitian disimpulkan bahwa hasil penelitian yang dilakukan yaitu dapat memberikan pengetahuan bahwa dalam bidang kedokteran dapat dilakukan penerapan teknologi *machine learning* yang dapat melakukan pendeteksian atau *screening* gejala awal Covid-19 bahkan untuk variannya. Penerapan metode *Naïve Bayes* pada penelitian ini berhasil menghasilkan *performance* dengan akurasi sangat baik sebesar 90% yang dihasilkan oleh Pengklasifikasian penyakit Covid-19 dan akurasi sebesar 93% untuk Pengklasifikasian varian Covid-19 berdasarkan gejala awal. Hasil pemodelan tersebut lebih besar dari dibandingkan penelitian oleh Al-Zubidi, dkk [4] yang membuat sistem deteksi Covid-19 berbasis *mobile* menggunakan 2 metode Teknik pengklasifikasian yaitu FCM dan BP dengan akurasi 89%. Sehingga dari hal tersebut menunjukkan bahwa penerapan metode *Naïve Bayes* tepat digunakan dalam pendeteksian atau *screening* gejala awal Covid-19.

III. KESIMPULAN

Berdasarkan hasil dan pembahasan mengenai sistem deteksi penyakit covid-19 menggunakan algoritma *Naïve Bayes* berbasis android maka diperoleh kesimpulan yaitu 1) Sistem deteksi penyakit covid-19 berdasarkan gejala awal menggunakan algoritma *Naïve Bayes* berbasis android yang dibangun telah mampu memprediksi penyakit Covid-19 ke dalam 4 *class* dengan nilai *F1-Score* yaitu *Allergy* 0,98, *Cold* 0,61, *Covid* 0,56, dan *Flu* 0,95, serta gejala yang paling berpengaruh pada *class Allergy* yaitu CS13 (*Loss of taste*) dengan nilai 0,50, *class Cold* yaitu CS3 (*Tiredness*) dengan nilai 0,52, *class Covid* yaitu CS12 (*Difficulty breathing*) dengan nilai 0,51, dan *class Flu* yaitu CS19 (*Sneezing*) dengan nilai 0,54; 2) sistem yang dibangun mampu memprediksi varian Covid-19 ke dalam 3 *class* dengan nilai *F1-Score* yaitu *alpha* 0,85, *delta* 0,78, dan *omicron* 0,93, serta gejala yang paling berpengaruh pada *class Alpha* yaitu VS3 (*Loss of appetite*) dengan nilai 0,75, *class Delta* yaitu VS12 (*Cough*) dengan nilai 0,88, dan *class Omicron* yaitu VS10 (*Sore throat*) dengan nilai 0,68; 3) Aplikasi Android yang berhasil mendeteksi penyakit *Coronavirus Disease 2019* (Covid-19) berdasarkan gejala awal menggunakan metode klasifikasi *Naïve Bayes* dapat dirancang dengan pendekatan *Extreme Programming* (XP).

Saran untuk pengembangan penelitian selanjutnya adalah melakukan penyeimbangan data pada *dataset* yang digunakan untuk pengklasifikasian penyakit Covid-19 dengan menggunakan teknik SMOTE, serta menggunakan *dataset real* dalam pengklasifikasian varian Covid-19.

REFERENSI

- [1] H. Jusuf, I. V. AMU, and V. AMU, "Pengaruh Kecemasan Terhadap Pemeriksaan Swab Pada Suspek Covid-19 Di Kota Gorontalo," 2021.
- [2] E. Turban, D. King, J. K. Lee, T.-P. Liang, and D. C. Turban, *Electronic Commerce - A Managerial and Social Perspective*. 2015.
- [3] A. S. Kwekha-Rashid, H. N. Abduljabbar, and B. Alhayani, "Coronavirus disease (COVID-19) cases analysis using machine-learning applications," *Appl. Nanosci.*, no. 0123456789, 2021, doi: 10.1007/s13204-021-01868-7.
- [4] A. F. Al-zubidi, N. F. AL-Bakri, R. K. Hasoun, S. H. Hashim, and H. T. S. Alrikabi, "Mobile Application to Detect Covid-19 Pandemic by Using Classification Techniques: Proposed System," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 16, pp. 34–51, 2021, doi: 10.3991/ijim.v15i16.24195.
- [5] A. R. Fahindra and I. H. Al Amin, "Sistem Pakar Deteksi Awal Covid-19 Menggunakan Metode Certainty Factor," *J. Tekno Kompak*, vol. 15, no. 1, p. 92, 2021, doi: 10.33365/jtk.v15i1.914.
- [6] R. R. Al Hakim, E. Rusdi, and M. A. Setiawan, "Android Based Expert System Application for Diagnose COVID-19 Disease: Cases Study of Banyumas Regency," *J. Intell. Comput. Heal. Informatics*, vol. 1, no. 2, p. 26, 2020, doi: 10.26714/jichi.v1i2.5958.
- [7] B. Etikasari, T. D. Puspitasari, A. A. Kurniasari, and L. Perdanasari, "Sistem Informasi Deteksi Dini Covid-19," *J. Tek. Elektro dan Komput.*, vol. 9, no. 2, pp. 101–108, 2020.
- [8] S. Bhatia and J. Malhotra, "Naïve bayes classifier for predicting the novel coronavirus," *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV 2021*, no. Icicv, pp. 880–883, 2021, doi: 10.1109/ICICV50876.2021.9388410.
- [9] G. G. Akbar, D. Kurniadi, and N. Nurliawati, "Content Analysis of Social Media: Public and Government Response to COVID-19 Pandemic in Indonesia," *J. Ilmu Sos. dan Ilmu Polit.*, vol. 25, no. 1, pp. 16–31, 2021, doi: 10.22146/jsp.56488.
- [10] A. Mulyani, D. Kurniadi, and N. Fauziah, "Perancangan Sistem Informasi Geografis Pemetaan Sebaran Kasus Covid-19 di Kabupaten Garut," *J. Algoritma*, vol. 18, no. 1, pp. 119–130, 2021, doi: 10.33364/algoritma/v.18-1.938.
- [11] J. Hoffer, *Modern Systems Analysis and Design*, 6/e. 2012.
- [12] F. Fadhilah, "Penerapan Metode Naive Bayes Pada Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Kulit Pada Kucing," *J. Infomedia Tek. Inform. Multimed. Jar.*, 2020.
- [13] A. A. Singh, Krishna Kant; Elhoseny, Mohamed; Singh, Akansha; Elngar, *Machine Learning and the Internet of Medical Things in Healthcare*. 2021.