TESIS

PENERAPAN METODE CORRELATED NAÏVE BAYES CLASSIFIER MENGGUNAKAN SELEKSI FITUR INFORMATION GAIN UNTUK KLASIFIKASI PENYAKIT JANTUNG

Disusun oleh:

Nama : **Hani Setiani**
NIM : **20.51.1288**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022

TESIS

PENERAPAN METODE CORRELATED NAÏVE BAYES CLASSIFIER MENGGUNAKAN SELEKSI FITUR INFORMATION GAIN UNTUK KLASIFIKASI PENYAKIT JANTUNG

APPLICATION OF CORRELATED NAÏVE BAYES CLASSIFIER METHOD USING INFORMATION GAIN FEATURE SELECTION FOR CLASSIFICATION OF HEART DISEASE

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : **Hani Setiani**
NIM : **20.51.1288**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022

HALAMAN PENGESAHAN

PENERAPAN METODE CORRELATED NAÏVE BAYES CLASSIFIER MENGGUNAKAN SELEKSI FITUR INFORMATION GAIN UNTUK KLASIFIKASI PENYAKIT JANTUNG

APPLICATION OF CORRELATED NAÏVE BAYES CLASSIFIER METHOD USING INFORMATION GAIN FEATURE SELECTION FOR CLASSIFICATION OF HEART DISEASE

Dipersiapkan dan Disusun oleh

Hani Setiani

20.51.1288

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 7 Desember 2022

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 7 Desember 2022
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

PENERAPAN METODE CORRELATED NAÏVE BAYES CLASSIFIER MENGGUNAKAN SELEKSI FITUR INFORMATION GAIN UNTUK KLASIFIKASI PENYAKIT JANTUNG

APPLICATION OF CORRELATED NAÏVE BAYES CLASSIFIER METHOD USING INFORMATION GAIN FEATURE SELECTION FOR CLASSIFICATION OF HEART DISEASE

Dipersiapkan dan Disusun oleh

Hani Setiani

20.51.1288

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 7 Desember 2022

Pembimbing Utama

Dr. Andi Sunyoto, M.Kom.
NIK. 190302052

Anggota Tim Penguji

Alva Hendi Muhammad, S.T., M.Eng., Ph.D.
NIK. 190302493

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Drs. Asro Nasiri, M.Kom.
NIK. 190302152

Dr. Andi Sunyoto, M.Kom.
NIK. 190302052

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 7 Desember 2022
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Hani Setiani
NIM : 20.51.1288
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung

Dosen Pembimbing Utama : Dr. Andi Sunyoto, M.Kom.
Dosen Pembimbing Pendamping : Drs. Asro Nasiri, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 7 Desember 2022
Yang Menyatakan,

Hani Setiani

HALAMAN PERSEMBAHAN

Tesis ini Saya persembahkan untuk Ke dua Orang Tua dan Adik-Adik tercinta Terima kasih atas segala do'a, semangat, dukungan, perhatian, motivasi serta kasih sayang yang diberikan selama ini. Semoga kelak tesis ini dapat menjadi support system bagi adik-adik ku untuk terus semangat dalam meraih pendidikan setinggi-tingginya.

HALAMAN MOTTO

"Perjuangan yang memiliki proses lika liku memang terasa berat tapi percayalah ketika kamu berhasil mencapainya kamu akan merasa bangga dengan dirimu"

"Ketika hari ini kamu sedang menghadapi hal yang sulit, percayalah hari ini akan berlalu seperti kesulitan yang kamu hadapi jadi jangan menyerah"

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT karena atas segala karunia dan ridho-Nya, sehingga tesis yang berjudul "Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung" dapat diselesaikan dengan baik.

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister Komputer pada program studi Magister Teknik Informatika Universitas Amikom Yogyakarta.

Penyelesaian tesis yang sangat berharga ini tidak lepas dari segala bantuan, bimbingan, dorongan dan doa dari berbagai pihak. Pada kesempatan ini, penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Kedua Orang Tua dan adik tercinta yang senantiasa memberikan do'a, semangat, dan dukungan kepada penulis agar senantiasa semangat dalam menuntut ilmu.
2. Bapak Dr. Andi Sunyoto, M.Kom. selaku pembimbing utama yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
3. Bapak Drs. Asro Nasiri, M.Kom. selaku pembimbing pendamping yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
4. Dosen Penguji yang telah memberikan saran yang baik demi kemajuan tesis ini.

5. Direktur Program Pascasarjana, jajarannya, staf dan rekan-rekan Magister Teknik Informatika Universitas Amikom Yogyakarta.

Semoga laporan ini dapat bermanfaat bagi penulis dan juga untuk para pembaca laporan. Penulis berharap adanya kritik dan saran guna memperbaiki dan pengembangan dari laporan ini kedepannya. Kritik dan saran tersebut dapat dikirim ke email penulis yaitu Hani.1288@students.amikom.ac.id. Akhir kata penulis ucapkan terima kasih dan selamat membaca.

Yogyakarta, 7 Desember 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Penyakit jantung atau penyakit kardiovaskular merupakan salah satu penyakit tidak menular (PTM) paling mematikan. Pada tahun 2008, sekitar 17,3 juta kematian akibat penyakit kardiovaskular diperkirakan akan terus meningkat mencapai 23,3 juta kematian pada tahun 2030. Pentingnya keputusan klinis dalam catatan pasien yang terkomputerisasi dapat mengurangi kesalahan medis selama melakukan diagnosis. Teknik data mining memiliki potensi untuk menciptakan lingkungan yang kaya pengetahuan, sehingga dapat membantu meningkatkan kualitas pengambilan keputusan klinis. Tujuan dari penelitian ini adalah melakukan teknik klasifikasi data mining menggunakan metode *Correlated Naïve Bayes Classifier* dengan menerapkan seleksi fitur *Information Gain* untuk klasifikasi penyakit jantung. Metode *Correlated Naïve Bayes Classifier* ini dipilih karena berpotensi memiliki nilai akurasi tinggi dengan cara menghitung nilai korelasi *value attribut* terhadap *class*, sehingga yang menjadi dasar ketepatan klasifikasi tidak hanya *probability* tetapi juga seberapa besar hubungan (korelasi) *attribute* dengan *class*. Sedangkan seleksi fitur *information gain* dipilih untuk mengurangi fitur yang tidak relevan. Pengujian yang dilakukan menggunakan empat tahap yaitu *Naïve Bayes*, *Naïve Bayes* dengan seleksi fitur *Information Gain*, *Correlated Naïve Bayes Classifier*, dan *Correlated Naïve Bayes Classifier* dengan seleksi fitur *Information Gain*. Berdasarkan beberapa hasil pengujian menggunakan 6 atribut yaitu cp, thal, ca, exang, slope dan num yang telah dilakukan akurasi terbaik terdapat pada metode *Correlated Naïve Bayes Classifier* dengan seleksi fitur Information Gain sebesar 91,20%.

Kata kunci: Correlated Naïve Bayes, Seleksi fitur, Information Gain, dan Penyakit jantung.

ABSTRACT

Heart disease or cardiovascular disease is one of the deadliest non-communicable diseases (NCDs). In 2008, about 17.3 million deaths caused by cardiovascular disease are expected to continue to increase reaching 23.3 million deaths by 2030. The importance of clinical decisions in computerized patient records can reduce medical errors during diagnosis. Data mining techniques have the potential to create a knowledge-rich environment, which can help improve the quality of clinical decision-making. The purpose of this study is to perform a data mining classification technique using the Correlated Naïve Bayes Classifier method by applying the Information Gain feature selection for the classification of heart disease. The Correlated Naïve Bayes Classifier method was chosen because it has the potential to have a high accuracy value by calculating the correlation value of the attribute value to the class, so that the basis for classification accuracy is not an only probability but also how big the relationship (correlation) of the attribute with the class is. Meanwhile, information gain feature selection is chosen to reduce irrelevant features. The test is carried out using four stages: Naive Bayes, Naive Bayes with Information Gain feature selection, Correlated Naive Bayes Classifier, and Correlated Naive Bayes Classifier with Information Gain feature selection. Based on several test results using 6 attributes, namely cp, thal, ca, exang, slope and num that have been carried out, the best accuracy is found in the Correlated Naïve Bayes method with Information Gain feature selection of 91.20%.

Keyword: Correlated Naïve Bayes Classifier, Feature selection, Information Gain, and Heart disease.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Penyakit jantung atau penyakit kardiovaskular merupakan salah satu penyakit tidak menular (PTM) paling mematikan, karena merupakan pembunuh nomor satu di dunia. Faktor utama penyebab terjadinya penyakit jantung adalah penggunaan tembakau, kurang olahraga, diet yang tidak sehat, konsumsi alkohol, meningkatnya usia, tekanan darah tinggi, kolesterol tinggi dan terdapat lemak yang berlebihan pada tubuh (obesitas) (Nawawi, H. M. et al., 2019).

Pada tahun 2008, sekitar 17,3 juta kematian disebabkan oleh penyakit kardiovaskular. Lebih dari 3 juta kematian ini terjadi sebelum usia 60 tahun. Kematian "awal" akibat penyakit jantung berkisar dari 4% di negara-negara berpenghasilan tinggi hingga 42% di negara-negara berpenghasilan rendah. Komplikasi hipertensi menyebabkan sekitar 9,4 kematian di seluruh dunia setiap tahun. Hipertensi bertanggung jawab atas setidaknya 45% kematian akibat penyakit jantung dan 51% kematian akibat stroke. Kematian akibat penyakit kardiovaskular, khususnya penyakit jantung koroner dan stroke, diperkirakan akan terus meningkat, mencapai 23,3 juta kematian pada tahun 2030 (Pusdatin kementrian Kesehatan RI, 2014).

Diagnosa penyakit jantung perlu diprediksi karena keputusan klinis sering didasarkan pada intuisi dan pengalaman dokter, daripada penggunaan data pengetahuan yang tersembunyi dalam database. Praktik ini mengarah pada

prasangka yang tidak diinginkan, kesalahan, dan mempengaruhi kualitas perawatan pasien. Dengan mengintegrasikan dukungan keputusan klinis ke dalam catatan pasien yang terkomputerisasi dapat mengurangi kesalahan medis, meningkatkan keselamatan pasien, dan mengurangi variasi praktik yang tidak diinginkan (Sciences, H., 2016).

Data mining berpotensi untuk menghasilkan lingkungan kaya akan pengetahuan yang dapat membantu meningkatkan kualitas keputusan klinis secara signifikan (Palaniappan, S. and Awang, R., 2008). Data mining merupakan serangkaian proses yang mengumpulkan pengetahuan atau pola dari kumpulan data. Data mining dapat memecahkan masalah dengan menganalisis data yang sudah ada di database. Output dari data mining ini dapat digunakan untuk memperbaiki pengambilan keputusan (Hasran, 2020).

Terdapat beberapa penelitian sebelumnya yang serupa mengenai klasifikasi data mining pada penyakit jantung diantaranya, Penelitian yang dilakukan oleh (Lestari, M., 2014) menggunakan metode K – Nearest Neighbor untuk deteksi penyakit jantung dengan dataset yang berisi 13 atribut. Pengujian yang dilakukan mendapatkan akurasi sebesar 70%. Penelitian ini bertujuan untuk mengetahui algoritma mana yang lebih akurat dan efisien, perlu melakukan deteksi menggunakan metode lain sehingga dapat ditentukan algoritma yang tepat untuk deteksi penyakit jantung.

Penelitian yang dilakukan oleh (Bianto, M. A. et al., 2020) pengujian menggunakan metode naïve bayes mendapatkan nilai akurasi sebesar 90,16%, dengan rata rata nilai presisi 87,44% dan rata rata nilai recall 87,95% menggunakan

data sebanyak 303 serta memiliki 2 kelas dan 14 atribut, sehingga masih perlu untuk meninjau berbagai teknik kombinasi pemilihan fitur lain untuk dapat meningkatkan nilai performa algoritma dalam melakukan klasifikasi.

Berdasarkan permasalahan dan beberapa penelitian diatas mengenai klasifikasi penyakit jantung, maka peneliti akan melakukan pengujian menggunakan metode *Correlated Naïve Bayes Classifier* dengan menerapkan seleksi fitur *Information Gain*. Metode *Correlated Naïve Bayes Classifier* ini dipilih karena berpotensi memiliki nilai akurasi tinggi dengan cara menghitung nilai korelasi *value attribut* terhadap *class*, sehingga yang menjadi dasar ketepatan klasifikasi tidak hanya *probability* tetapi juga seberapa besar hubungan (korelasi) *attribute* dengan *class (Muktamar, B. A. et al., 2015)*. Sedangkan seleksi fitur *information gain* dipilih untuk mengurangi fitur yang tidak relevan terhadap dataset (Hasran, 2020). Hasil dari pengujian ini diharapkan mampu meningkatkan nilai akurasi yang dimiliki oleh *Correlated Naïve Bayes Classifier* pada klasifikasi penyakit jantung.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, masalah yang diangkat pada penelitian ini adalah:

a. Berapa tingkat akurasi yang dimiliki *Naïve Bayes* sebelum dan sesudah diterapkan seleksi fitur *Information Gain*?
b. Berapa tingkat akurasi yang dimiliki *Correlated Naïve Bayes Classifier* sebelum dan sesudah diterapkan seleksi fitur *Information Gain*?

c. Pengujian menggunakan metode manakah yang memiliki akurasi terbaik pada klasifikasi penyakit jantung?

1.3. Batasan Masalah

Bagian ini membahas tentang batasan yang digunakan untuk penelitian agar terfokus pada aspek yang diangkat, adapun batasanya sebagai berikut:

a. Metode yang digunakan untuk klasifikasi adalah Naïve Bayes dan *Correlated Naïve Bayes Classifier*
b. Metode yang digunakan untuk seleksi fitur adalah *Information Gain*
c. Dataset yang digunakan berasal dari UCI Machine Learning Repository dengan format file Comma Sparated Value (CSV) yang berisikan 304 record dan memiliki 14 atribut.
d. Memiliki 1 atribut yang berisikan 2 kelas yaitu: ketidakhadiran (*absence*) penyakit jantung dan kehadiran (*presence*) penyakit jantung
e. Perhitungan akurasi yang dilakukan menggunakan *Confusion Matrix*.

1.4. Tujuan Penelitian

Berdasarkan latar belakang yang diangkat, maka dibuat tujuan yang akan diselesaikan pada penelitian ini, antara lain:

a. Mendapatkan tingkat akurasi pada metode *Naïve Bayes* sebelum dan sesudah diterapkan seleksi fitur *Information Gain*
b. Mendapatkan tingkat akurasi pada metode *Correlated Naïve Bayes Classifier* sebelum dan sesudah diterapkan seleksi fitur *Information Gain*

c. Mendapatkan metode yang memiliki akurasi terbaik pada pengujian klasifikasi penyakit jantung
d. Sebagai syarat kelulusan Magister Teknik Informatika Universitas Amikom Yogyakarta

1.5. Manfaat Penelitian

Manfaat dari penelitian ini yang ingin dicapai, diantaranya:

a. Penelitian ini bermanfaat untuk menambah wawasan penulis tentang bagaimana penerapan metode *Correlated Naïve Bayes Classifier* dengan seleksi fitur *Information Gain* untuk klasifikasi penyakit jantung.
b. Hasil penelitian ini hendaknya dapat dijadikan acuan untuk penelitian selanjutnya terkait penerapan metode *Correlated Naïve Bayes Classifier* dalam kaitannya dengan seleksi fitur Information Gain pada sistem pendukung keputusan.

1.6. Hipotesis

Berdasarkan penelitian sebelumnya (Marzuki, J. I. et al., 2018), telah menerapkan *Correlated Naïve Bayes Classifier* diagnosa penyakit diabetes, penelitian tersebut menyimpulkan bahwa penerapan metode *Correlated Naïve Bayes Classifier* memiliki nilai akurasi lebih baik dibanding hanya menggunakan algoritma naïve bayes. Sedangkan pada penelitian ini akan menggunakan metode *Correlated Naïve Bayes Classifier* dengan penerapan Seleksi Fitur *Information Gain* untuk klasifikasi penyakit jantung. Diharapkan dengan menerapkan *Correlated Naïve Bayes Classifier* dengan penerapan Seleksi Fitur *Information Gain*, dapat disimpulkan metode mana yang terbaik, dan memiliki nilai akurasi yang didapatkan lebih baik dari penelitian sebelumnya (Bianto, M. A. et al., 2020) yaitu diatas 90%.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh (Saputra, D. et al., 2021) menggunakan beberapa metode untuk mengetahui akurasi yang dimiliki naïve bayes dengan penerapan PSO kemudian membandingkan metode tersebut untuk mengetahui metode manakah yang memiliki akurasi tertinggi untuk klasifikasi penyakit jantung. Metode yang digunakan seperti C4.5, Naïve Bayes, dan Support Vector Machine. Hasilnya didapatkan nilai akurasi algoritma C4.5 sebesar 74,12% dan nilai akurasi algoritma Naïve Bayes sebesar 85,26% dan terakhir algoritma Support Vector Machine sebesar 85,26%.

Penelitian yang dilakukan oleh (Mubaroq, T. F. et al., 2019) melakukan peningkatan akurasi naïve bayes dengan menggunakan Discretization dan Information Gain menggunakan dataset penyakit jantung yang diperoleh dari UCI Machine Learning terdiri dari 270 intance dan 14 atribut. Untuk mengetahui akurasi yang dimiliki, perhitungan dilakukan menggunakan k-fold cross validation dengan nilai k = 10. Dan memperoleh hasil akurasi sebesar 85,5556%. Hal ini terdapat peningkatan ketika menambahkan Discretization dan Information Gain pada Naïve Bayes sebesar 0,3704%.

Penelitian yang dilakukan oleh (Annisa, R., 2019) menggunakan beberapa metode klasifikasi untuk menentukan akurasi dan membandingkan kelima metode tersebut. Setiap metode diuji dengan uji parametrik menggunakan t-test dan dataset

yang digunakan berupa penyakit jantung yang diderita oleh laki laki, pada pengujian yang dilakukan algoritma Decision Tree mendapatkan hasil akurasi senilai 76.08%, Naïve Bayes senilai 78,95%, k-Nearest Neighbour senilai 60.77%, Random Forest 80.38% dan Decison Stum 78,95%.

Penelitian yang dilakukan oleh (Dulhare, U. N., 2018) Meningkatkan akurasi Naïve Bayes menggunakan dua seleksi fitur, kemudian hasilnya akan dibandingkan untuk menentukan penerapan pemilih fitur mana yang tepat digunakan untuk klasifikasi, pengujian ini menggunakan dataset penyakit jantung yang berisi 270 memiliki 14 atribut dan 1 label kelas. pada penelitian ini pengujian dilakukan sebanyak tiga kali. pengujian pertama pada metode Naïve Bayes mendapatkan hasil 79,12%, pengujian kedua pada NB + PSO mendapatkan hasil 87,91%, pengujian ketiga pada NB + GA mendapatkan hasil 86,29%.

Penelitian yang dilakukan oleh (Reddy, K. V. V. et al., 2021) menggunakan beberapa algoritma machine learning dengan penerapan Principal Component Analysis (PCA) untuk mengetahui akurasi yang dimiliki metode mana yang tinggi. Dataset yang digunakan didapatkan dari kanggle yaitu penyakit jantung UCI Cleveland berisi 14 atribut. penggujian menggunakan 10 cross validation tanpa pca dan menggunkan pca pada setiap metode mendapatkan akurasi sebesar Descision Tree (DT) 78.2%, Discriminant Analysis (DA) 83.5%, Logistic Regression (LR) 83.5%, Naïve Bayes (NB) Gaussian 82.2%, Support Vector Machines (SVM) 83.8%, K-Nearest Neighbors (KNN) Consine 83.8%.

Penelitian yang dilakukan oleh (Marzuki, J. I. et al., 2018) yaitu komparasi dua metode Correlated-Naive Bayes Classifier dan Naive Bayes Classifier untuk

mendapatkan hasil yang terbaik yang digunakan untuk penyakit diabetes. pngujian yang dilakukan menggunakan dataset Pima Indian Diabetes diproleh dari UCI Repository, dataset ini berisikan 768 data, 9 atribut dan 2 kelas. Hasil pengujian menggunakan 10-Fold Cross Validation memperoleh akurasi pada metode Correlated Naive Bayes Classifier (CNBC) yaitu sebesar 67,15% dan metode Naive Bayes Classifier (NBC) sebesar 64,33%.

Penelitian yang dilakukan oleh (Hairani, H. and Innuddin, M., 2020) bertujan untuk memperoleh akurasi yang optimal untuk klasifikasi data kesehatan menggunakan metode dua metode yaitu metode Correlated Naive Bayes dan seleksi fitur berbasis Wrapper. Pengujian ini menggunakan dua dataset yaitu dataset Pima Indan Diabetes dan dataset Thyroid ini diperoleh dari UCI Machine Learning Repository. Memiliki tahapan pre-procesing seperti transformasi, scaling, dan seleksi fitur berbasis Wrapper. Pengujian akurasi yang dilakukan menggunakan 10-fold cross validation mendapatkan Hasil dataset Pima Indan Diabetes akurasinya sebesar 71,4% dan akurasi dataset Thyroid sebesar 79,38%.

2.2. Keaslian Penelitian

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	A Comparative Analysis of C4.5 Classification Algorithm, Naïve Bayes, and Support Vector Machine Based on Particle Swarm Optimization (PSO) for Heart Disease Prediction	Dedi Saputra, Windi Irmayani, Deasy Purwaningtyas, Juniato Sidauruk, Burcu Gurbuz International Journal of Advances in Data and Information Systems, dan 2021	Melakukan perbandingan dengan menggunakan algoritma klasifikasi yaitu C4.5, Naïve Bayes, dan Support Vector Machine dengan menerapkan PSO untuk prediksi penyakit jantung	Didapakan hasil akurasi setelah diterapkan PSO algoritma C4.5 sebesar 74,12%, algoritma Naïve Bayes sebesar 85,26%, dan terakhir algoritma Support Vector Machine sebesar 85,26%.	Saran untuk penelitian berikutnya adalah Dilakukan optimasi menggunakan fitur seleksi lainnya.	Penelitian Dedi Saputra, dkk melakukan penelitian membadingkan beberapa metode klasifikasi dan melakukan optimasi menggunakan PSO untuk meningkatkan akurasi dengan cara pembobotan atribut Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung.

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Application of Discretization and Information Gain on Naïve Bayes to Diagnose Heart Disease	Taufik Fajar Mubaroq, Endang Sugiharti, Isa Akhlis, Journal of Advances in Information Systems and Technology, dan 2019	Melakukan penerapan Discretization dan Information Gain ke algoritma naive bayes untuk menentukan akurasi sebelum dan sesudah menerapkannya untuk mendiagnosis penyakit jantung	Mendapatkan hasil akurasi algoritma naive bayes sebesar 85,1852% Sebelum diterapkan Discretization dan Information Gain, sedangkan Setelah dilakukan penerapan menjadi 85,5556%, terjadi peningkatan sebesar 0,3704%.	Saran untuk penelitian berikutnya adalah menggunkanan dataset yang lebih banyak apakah menghasilkan nilai akurasi yang lebih tinggi atau tidak dan dapat dijadikan perbandingan	Penelitian Taufik Fajar Mubaroq, dkk melakukan penelitian yaitu menerapkan Discretization dan Information Gain pada naïve bayes untuk diagnosa penyakit jantung dengan cara menghilangkan atribut yang memiliki nilai Information Gain yang paling kecil, Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Analisis Komparasi Algoritma Klasifikasi Data Mining untuk Prediksi Penderita Penyakit Jantung	Riski Annisa, Jurnal Teknik Informatika Kaputama (JTIK), dan 2019	Melakukan perbandingan dengan menggunakan beberapa algoritma klasifikasi yaitu Decision Tree, Naïve Bayes, k-Nearest Neighbour, Random Forest, dan Decison Stump di uji parametrik dengan t-test agar mendapatkan hasil perbandingan metode terbaik pada pria dengan penyakit jantung.	Mendapatkan nilai akurasi sebesar tertinggi sebesar 80.38% Pada Random Forest sedangkan yang lain C4.5 sebesar 76.08%, Decision Stump sebesar 78.95%, Naïve Bayes sebesar 8.95%, kemudian k-NN merupakan algoritma yang kurang baik dengan nilai 60.77%.	Melengkapi penelitian dengan mencantumkan bebepa dataset yang digunakan untuk menentukan akurasi pada metode yang digunakan	Penelitian Riski Annisa melakukan komparasi algoritma klasifikasi yang digunakan yaitu Decision Tree, Naïve Bayes, k-Nearest Neighbour, Random Forest, dan Decison Stump untuk menetukan akurasinya pada evaluasinya menggunakan 10-*fold cross validation*, Sedangkan penelitian ini pengujian akurasi menggunakan confusion matrix.

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Prediction System for Heart Disease using Naive Bayes and Particle Swarm Optimization.	Uma N. Dulhare, Biomedical Research, dan 2018.	Melakukan perbandingan pada metode naïve bayes dengan menggunakan kedua fitur seleksi untuk mengetahui akurasi yang tertinggi.	Akurasi algoritma Naïve bayes sebelum dan sesudah dilakukan penerapan GA dan Particle Swarm Optimization (PSO). pada penerapan GA mendapatkan akurasi 86.29% sebelum 86.29%. penerapan PSO mendapatkan akurasi 87.91% sebelum 86.29%	Saran untuk penelitian berikutnya adalah menggunakan metode klasifikasi lain dengan penerapan metode optimasi yang sama sehingga diharapkan memperoleh algoritma terbaik untuk klasifikasi	Penelitian Uma N. Dulhare melakukan perbandingan pada penerapan seleksi fitur terhadap naïve bayes menggunakan PSO dan GA untuk menentukan akurasi terbaik, Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Heart Disease Risk Prediction using Machine Learning with Principal Component Analysis	Karna Vishnu Vardhana Reddy, Irraivan Elamvazuthi, Azrina Abd Aziz, Sivajothi Paramasivam, Hui Na Chua, International Conference on Intelligent and Advanced Systems (ICIAS), dan 2021	Melatih, memvalidasi dan menganalisis pengklasifikasi machine learning dengan penerapan Principal Component Analysis (PCA)	Hasil dari penelitian ini mendapatkan akurasi 83.8% dengan menggunakan SVM linear dan KNN Consine tanpa PCA, dengan PCA model LR mendapatkan 85.8%	Saran untuk penelitian berikutnya adalah menggunakan algoritma pemilihan fitur untuk meningkatkan kinerja classifier.	Penelitian Karna Vishnu Vardhana Reddy, dkk melakukan penenelitian menggunakan beberapa algoritma Machine Learning dengan penerapan PCA untuk meningkatkan akurasi dengan cara seleksi fitur sehingga menghasilkan fitur yang tidak saling berkorelasi. Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung.

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Komparasi Akurasi Metode Correlated Naive Bayes Classifier dan Naive Bayes Classifier untuk Diagnosis Penyakit Diabetes	Hairani, Gibran Satya Nugraha, Mokhammad Nurkholis Abdillah, Muhammad Innuddin, InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan), dan 2018	Melakukan komparasi beberapa metode klasifikasi data mining yaitu metode Correlated-Naive Bayes Classifier dan Naive Bayes Classifier untuk mendapatkan akurasi terbaik sehingga dapat digunakan untuk diagnosis penyakit diabetes secara efektif	Hasil dari penelitin ini mendapatkan akurasi metode Correlated Naive Bayes Classifier (CNBC) sebesar 67,15%, sedangkan metode Naive Bayes Classifier (NBC) sebesar 64,33%.	Saran untuk penelitian berikutnya yaitu menambah algoritma optimasi lainnya sehingga mendapatkan akurasi yang lebih baik.	Penelitian Hairani, dkk melakukan penenelitian komparasi beberapa algoritma Metode Correlated Naive Bayes Classifier dan Naive Bayes Classifier untuk menentukan akurasi yang terbaik pada klasifikasi penyakit diabetes. untuk menghitung akurasi menggunakan 10-Fold Cross Validation, Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	Kombinasi Metode Correlated Naive Bayes dan Metode Seleksi Fitur Wrapper untuk Klasifikasi Data Kesehatan	Hairani, Muhammad Innuddin, Jurnal Teknik Elektro, dan 2020	Melakukan Kombinasi algoritma Correlated Naive Bayes dan seleksi fitur berbasis Wrapper untuk klasifikasi data kesehatan untuk mendapatkan akurasi optimal	Hasil dari penelitian mendapatkan untuk dataset Pima Indan Diabetes akurasinya sebesar 71,4% dan akurasi dataset Thyroid sebesar 79,38%.	Saran untuk penelitian berikutnya yaitu menggunakan algoritma klasifikasi lainnya diharpkan mendapatkan akurasi terbaik untuk klasifikasi data diabetes dan thyroid	Penelitian Hairani dkk, melakukan penenelitian kombinasi algoritma Correlated Naive Bayes dan seleksi fitur berbasis Wrapper untuk klasifikasi data kesehatan. untuk menghitung akurasi menggunakan 10-Fold Cross Validation, Sedangkan penelitian ini menggunakan seleksi fitur *information gain untuk meningkatkan akurasi yang dimiliki* metode *Correlated Naive Bayes Classifier* dengan cara menentukan atribut yang relevan pada dataset yang digunakan yaitu penyakit jantung. Untuk perhitungan akurasi menggunakan confusion matrix

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Penerapan Metode Correlated Naïve Bayes Classifier Menggunakan Seleksi Fitur Information Gain Untuk Klasifikasi Penyakit Jantung (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8	Perancangan Sistem Klasifikasi Penyakit Jantung mengunakan Naïve Bayes	Mufti Ari Bianto, Kusrini, Sudarmawan, Citec Journal, dan 2019	Mengetahui akurasi penyakit jantung menggunakan Naïve bayes dengan pengujian 5-fold cross validation	Didapatkan hasil akurasi dengan rata-rata akurasi senilai 90,61%, rata-rata hasil nilai presisi senilai 87,44% dan rata-rata nilai recall senilai 87,95% dengan konfigurasi data yang terdapat pada UCI Machine Learning yang berisi 2 kelas klasifikasi dan 15 atribut dengan jumlah 303 data.	Saran untuk penelitin dapat dilakukan penambahan algoritma optimasi lain untuk menambah tingkat akurasi, presisi, dan recall.	Penelitian MuftI Ari Bianto, dkk untuk pengujian akurasi menggunakan 5-fold cross validation, Sedangkan penelitian ini pengujian akurasi menggunakan confusion matrix

Berdasarkan Tabel 2.1 matriks literatur review dan posisi penelitian. Penelitian yang telah dilakukan mengenai berapa akurasi dan cara meningkatkan akurasi *Naïve Bayes Classifier* pada penerapan dataset penyakit jantung, sudah dilakukan oleh beberapa penelitian sebelumnya namun terdapat perbedaan disetiap penelitian seperti penelitian (Saputra, D. et al., 2021) dan (Dulhare, U. N., 2018) menggunakan metode optimasi yaitu PSO, penelitian (Mubaroq, T. F. et al., 2019) menggunakan Discretization dan Information Gain, penelitian (Reddy, K. V. V. et al., 2021) menggunakan Principal Componen Analysis (PCA). Adapun perbedaan dalam perhitungan akurasi seperti penelitian (Bianto, M. A. et al., 2020) menggunakan 5-fold cross Validation dan penelitian (Annisa, R., 2019) dan (Reddy, K. V. V. et al., 2021) menggunakan 10-*fold cross validation*. Kemudian terdapat beberapa penelitian Penelitian yang telah dilakukan mengenai berapa akurasi dan cara meningkatkan akurasi *Correlated Naïve Bayes Classifier* pada penerapan dataset penyakit seperti penelitian (Marzuki, J. I. et al., 2018) adapun perbedaan perhitungan akurasi menggunakan 10-*fold cross validation*. Penelitian (Hairani, H. and Innuddin, M., 2020) adapun perbedaan pada penerapan Seleksi Fitur Wrapper.

Pada penelitian ini penulis akan melengkapi kekurangan dari penelitian sebelumnya yaitu melakukan teknik kombinasi pemilihan fitur lain untuk dapat meningkatkan nilai performa algoritma dalam melakukan klasifikasi. Dengan menggunakan metode Correlated Naïve Bayes Classifier dengan menerapkan seleksi fitur Information Gain agar menjadi metode yang lebih baik untuk

klasifikasi penyakit jantung dan perhitungan akurasi menggunakan Confusion Matrix.

2.3. Landasan Teori

2.3.1 Data Mining

Data mining adalah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada dalam database yang besar. Dalam jurnal ilmiah, data mining disebut juga dengan KDD (*Knowledge Discovery in Databases*).

Data mining didefinisikan sebagai seperangkat teknik yang digunakan dengan otomatis untuk mengeksplorasi secara menyeluruh dan mengungkap hubungan kompleks dalam kumpulan data yang sangat besar. Kumpulan data yang dimaksud di sini adalah kumpulan data tabular yang banyak digunakan dalam teknologi manajemen basis data relasional. Namun, teknik data mining juga dapat diterapkan pada representasi data lainnya seperti spatial, berbasis teks, dan multimedia (citra) (Siregar, A. M. and Puspabhuana, A., 2017).

Adapun langkah-langkah KDD (*Knowledge Discovery in Databases*) (Marzuki, J. I. et al., 2018), sebagai berikut:

1. *Data Cleaning*

   Merupakan proses untuk menghapus duplikasi data, memeriksa data yang tidak konsisten, dan terakhir memperbaiki kesalah pada data seperti kesalahan dalam penulisan maupun data yang telah hilang.

2. *Data Integration*

Merupakan proses penambahan data yang sudah ada dengan informasi yang relevan atau dapat juga merupakan penggabungan data dari berbagai database kedalam satu database baru yang dibutuhkan untuk KDD.

3. *Data Selection*

   Merupakan proses pemilihan data yang relevan dan dapat dilakukan analisis dari data operasional.

4. *Data Transformation*

   Merupakan proses transformasi data kedalam bentuk format tertentu sehingga dapat digunakan untuk proses data mining.

5. *Pattern Evaluation*

   Merupakan proses untuk identifikasi pola yang benar-benar menarik dari hasil data mining. Pada proses ini hasil dari teknik data mining berupa pola pola yang khas maupun model prediksi yang dievaluasi untuk menilai apakah suatu hipotesa yang ada sudah tercapai atau belum.

6. *Knowledge Presentation*

   Merupakan proses untuk menampilkan pola informasi yang dihasilkan dari proses data mining, visualisasi yang dihasilkan memudahkan untuk memahami hasil dari data mining.

2.3.2 Klasifikasi

Klasifikasi adalah ekstraksi data yang menentukan item mana dalam koleksi milik kelas tertentu. Klasifikasi dimulai dengan pengumpulan data yang berisi kelas-kelas yang diketahui (Werdiningsih, I. et al., 2020).

Kumpulan data dilakukan dengan mengelompokan terlebih dahulu model data atau yang biasa dikenal sebagai data latih. Setelah model terbentuk dari prosel pelatihan, selanjutnya data akan memasuki tahap pengujian dalam proses pengelompokan yang biasa disebut sebagai proses uji (Wanto, A., 2020).

Proses kalsifikasi data memiliki dua tahap, yaitu:

1. *Learning*

   Learning merupakan proses training data yang kemudian data tersebut dianalisa dengan menggunakan algoritma klasifikasi.

2. *Classification*

   Classification merupakan proses pengujian data yang dipakai untuk mengetahui kecepatan dari *classification rules*. Akurasi yang keberadaanya dapat diterima, rule dapat diimplementaikan pada suatu klasifikasi dari tuple data baru. Klasifikasi sendiri hanya bisa digunakan unuk suatu data training yang kuat, dimana kelas positif telah mewakili minoritas tanpa harus kehilangan atribut pada umumnya.

2.3.3 Pre-processing Data

Prep-rocessing data memiliki banyak cara antara lain adalah *replace missing value* dan normalisasi. *Replace missing value* dapat dilakukan dengan cara mencari nilai tengah pada dataset tersebut kemudian mengganti data kosong dengan nilai tengah yang diperoleh menggunakan persamaan (2.1):

Rumus mencari nilai tengah dari data ganjil

$$\text{Me} = \text{X}\frac{n+1}{2} \qquad (2.1)$$

Adapun Rumus mencari nilai tengah dari data genap pada persamaan (2.2):

$$\text{Me} = \frac{\left(\left(\text{x}\frac{n}{2}\right) + \left(\text{x}\frac{n}{2} + 1\right)\right)}{2} \quad (2.2)$$

Keterangan:

Me = median (nilai tengah)

X = variabel data

n = index data

2.3.4 Naïve Bayes

Metode *Naive Bayes* adalah klasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas milik suatu class. *Naive Bayes* didasarkan pada toerama bayes, memiliki kemampuan klasifikasi yang mirip dengan algoritma *Decision Tree* dan *Neural Network*. *Naive Bayes* sudah terbukti memiliki nilai akurasi dan kecepatan yang tinggi, ketika diaplikasikan pada sebuah database yang didalamnya terdapat jumlah data yang besar (Kusrini and Luthfi, E. T., 2009). *Naïve Bayes* dapat dihitung menggunakan persamaan (2.3):

$$\text{P(H|X)} = \frac{\text{P(X|H)}.\text{P(H)}}{\text{P(X)}} \quad (2.3)$$

Keterangan:

X = data dengan class yang belum diketahui

H = hipotesis data X merupakan suatu *class* spesifik

P(H|X) = probabilitas hipotesis H berdasarkan kondisi X (*posteriori probability*)

P(H) = probabilitas hipotesis H (*prior probability*)

P(X|H) = probabilitas X berdasar kondisi pada hipotesis H

P(X) = probabilitas dari X

Untuk menjelaskan teorema *Naive Bayes*, perlu diperhatikan bahwa proses klasifikasi memerlukan beberapa petunjuk untuk menentukan kelas mana yang cocok dengan sampel yang dianalisis.

2.3.5 Correlated Naïve Bayes Classifier

Correlated Naive Bayes Classifier merupakan suatu metode hasil dari pengembangan *Naïve Bayes*. Pada *Correlated Naïve Bayes Classifier* memiliki parameter tambahan yaitu nilai korelasi antar fitur X atribut terhadap kelasnya Y dan bilangan laplacian. Adapun perhitungan korelasi (*R-square*) ini dilakukan untuk mengetahui hubungan antar fitur X dengan kelasnya Y pada metode *Correlated Naïve Bayes Classifier*. Kemudian bilangan laplacian ini digunakan untuk menghindari zero probability(Hairani, H. and Innuddin, M., 2020). *Correlated Naïve Bayes Classifier* dihitung menggunakan persamaan (2.4):

$$P(X|Y) = \frac{P(Y)\pi_{i=1}^{q} P(X_i|Y)^{\tau}.R(X_i|Y)}{P(X)} \quad (2.4)$$

Keterangan:

X = fitur dengan kelas yang belum diketahui.

Y = hipotesis fitur X merupakan suatu kelas spesifik.

P(X|Y) = probabilitas hipotesis Y berdasarkan dengan kondisi Y.

P(Y) = probabilitas awal hipotesis Y (prior probability).

$\pi_{i=1}^{q} P(X_i|Y)$ = probabilitas setiap atribut dari fitur X berdasarkan dengan kondisi hipetesis Y.

$R(X_i|Y)$ = r-Square setiap atribut dari fitur X berdasarkan kondisi hipotesis Y.

τ = bilangan laplacian.

P(X) = Probabilitas dari X.

Adapun rumus untuk menghitung nilai korelasi pada persamaan (2.5) dan (2.6):

$$r = \frac{n.(\Sigma XY) - (\Sigma X).(\Sigma Y)}{\sqrt{(n.\Sigma X^2 - (\Sigma X)^2)}\ \sqrt{(n.\Sigma Y^2 - (Y)^2)}} \quad (2.5)$$

$$R\ Square = r^2 \quad (2.6)$$

R = r-square fitur antar kelasnya

r = nilai korelasi fitur antar kelasnya

n = jumlah dataset

ΣXY = total perkalian fitur X dengan kelasnya Y

ΣX = total fitur X

ΣY = total fitur Y

ΣX^2 = total fitur X yang dikuadratkan.

$(\Sigma X)^2$ = kuadrat total fitur X

ΣY^2 = total fitur Y yang dikuadratkan.

$(\Sigma Y)^2$ = kuadrat total fitur Y

2.3.6 *Information Gain*

Information Gain adalah metode pemilihan fitur yang bekerja dengan cara melakukan perangkingan secara sederhana. Information gain mendeteksi fitur yang

paling relevan berdasarkan kelas tertentu dengan cara menghitung nilai entropy. Entropy adalah ukuran dari ketidakpastian kelas menggunakan probabilitas dari suatu atribut tertentu (Yessy Nabella, F. et al., 2019). Tahapan dalam proses perhitungan *Information Gain*, sebagi berikut:

Rumus menentukan nilai entropy seluruh kelas dihitung menggunakan persamaan (2.7):

$$Entropy(S) = \sum_{i=1}^{n} -pi * log_2 pi \quad (2.7)$$

Rumus menentukan nilai information gain seluruh kelas dapat dihitung menggunakan persamaaan (2.8):

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} - \frac{|S_v|}{|S|} * Entropy(Sv) \quad (2.8)$$

Keterangan:

A = fitur

v = nilai yang mungkin untuk fitur A

pi = jumlah sampel setiap kelas i

Values(A) = nilai nilai untuk fitur A

| Sv | = jumlah sampel v

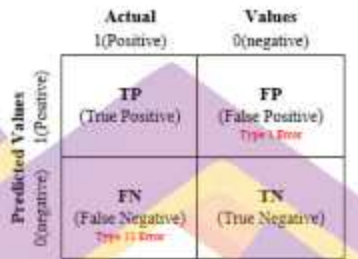
| S | = jumlah semua data

Entropy (Sv) = Entropy untuk data yang memiliki nilai v

2.3.7 ***Confusion Matrix***

Confusion Matrix yang dapat dikenal dengan *error matrix* merupakan salah satu teknik yang dapat digunakan untuk mengukur performa suatu model klasifikasi

dalam data mining. *Confusion Matrix* biasanya digambarkan sebagai tabel matriks yang menggambarkan kinerja model klasifikasi pada kumpulan data uji dari nilai sebenarnya yang diketahui (Mustika et al., 2021).



Gambar 2.1 Tabel Confusion Matrix

Gambar 2.1 terdapat istilah yang merepresentasikan hasil proses dari klasifikasi pada *Confusion Matrix* yaitu: *True Positive* (TP), *True Negative* (TN), *False Negative* (FN) dan *False Positive* (FP).

1. *Accuracy* adalah nilai yang menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar.

   Nilai akurasi dapat diperoleh dengan menggunakan persamaan (2.9):

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

2. *Precision* adalah nilai yang menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model.

   Nilai presisi dapat diperoleh dengan menggunakan persamaan (2.10):

$$Presisi = \frac{TP}{TP + FP} \quad (2.10)$$

3. *Recall* adalah nilai yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi

Nilai recall dapat diperoleh dengan menggunakan persamaan (2.11):

$$Recall = \frac{TP}{TP + FN} \tag{2.11}$$

4. *F1-Score*

F1–score merupakan mean dari precision dan recall. *F1-score* memiliki nilai terbaik 1.0 dan nilai terburuknya 0. Nilai *F1-score* dapat di peroleh dengan menggunakan persamaan (2.12):

$$F1 - Score = \frac{2 . (Recall \ . \ Precission)}{(Recall + Precission)} \tag{2.12}$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini memiliki jenis penelitian eksperimental yang mempelajari kausalitas, apakah menggunakan seleksi fitur information gain dapat meningkatkan akurasi pada Correlated Naïve Bayes Classifier, dan apakah hasil dari menggunakan Correlated Naïve Bayes Classifier dengan seleksi fitur information gain memiliki performa yang lebih baik dari penelitian sebelumnya tentang klasifikasi penyakit jantung. Sifat penelitian ini untuk menyelidiki kausalitas dan memperoleh hasil dalam bentuk informasi atau pengetahuan menggunakan seleksi fitur information gain dalam pengklasifikasi Correlated Naive Bayes Classifier. Pendekatan penelitian ini adalah kuantitatif, karena memperbaiki nilai akurasi dari penelitian sebelumnya sehingga mendapatkan akurasi yang lebih baik. Dalam penelitian ini melakukan penerapan seleksi fitur *Information Gain* pada *Corellated Naïve Bayes Classifier* untuk mendapatkan sebuah kesimpulan berupa akurasi yang optimal.

3.2. Metode Pengumpulan Data

Data koleksi penyakit jantung pada penelitian ini menggunakan dataset yang bersifat publik karena disediakan dan siapapun dapat mengunduh data

tersebut. Dataset yang didapatkan dari UCI Machine Learning Repository dengan nama file processed.cleveland.data sudah dimanfaatkan oleh peneliti sebelumnya seperti (Bianto, M. A. et al., 2020) yang digunakan untuk mengetahui tingkat akurasi pada algoritma *Naïve Bayes Classifier*.

Dataset processed.cleveland.data berisi 304 *record* memiliki 14 atribut seperti : age, sex, cp, trestbps, chol, fbs, respect,thalach, exang, oldpeak, slope, ca, thal, dan num. Deskripsi atribut processed.cleveland.data dapat dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi aribut processed.cleveland.data

No	Atribut	Tipe Data	Keterangan	Nilai
1	Age	Numerik	Umur	29 – 77
2	Sex	Kategori	Jenis Kelamin	Female, Male
3	Cp	Kategori	Tipe nyeri pada dada	Typical angina, Atypical angina, Non-anginal pain, Asymptomatic
4	Trestbps	Numerik	Tekanan darah istirahat (saat jantung istirahat) (dalam mm Hg saat masuk RSUD)	94 – 200
5	Chol	Numerik	Serum kolestrol (jumlah kolestrol dalam darah) dalam mg/dl	126 – 564
6	Fbs	Kategori	(Gula darah puasa/ sebelum makan>120m/dl)	False, True
7	Restecg	Kategori	Hasil elektrokardiografi (alat pemeriksaan jantung)	Normal, Having ST-T wave abnormaity, Having LV hypertrophy
8	Thalach	Numerik	Denyut jantung maksimum tercapai	71 – 202
9	Exang	Kategori	Olahraga yang diinduksi angina	No, Yes
10	Olpeak	Numerik	ST depresi yang disebabkan oleh olahraga relatif terhadap istirahat	0 - 6.2

Tabel 3.1 Deskripsi aribut processed.cleveland.data (Lanjutan)

No	Atribut	Tipe Data	Keterangan	Nilai
11	Slope	Kategori	Kemiringan segmen ST latihan puncak	Upsloping, Flat, Downslowping
12	Ca	Kategori	Jumlah pembuluh darah dijelaskan dengan fluoroskopi	0 -3 by fluoroscopy
13	Thal	Kategori	Detak jantung	Normal, Fixed defect, Reversable defect
14	Num	Kategori	Atribut yang Diprediksi	Absence, Presence

3.3. Metode Analisis Data

Dalam menganalisa data pada penelitian ini menggunakan metode *Correlated Naïve Bayes Classifier* dengan penerapan seleksi fitur *Information Gain*. Tahapannya yaitu pertama menyiapkan dataset yang akan digunakan, dataset didapatkan pada link https://archive.ics.uci.edu/ml/datasets/heart+disease dengan cara unduh *file* nya. Kedua melakukan *pre-processing* pada dataset yang digunakan. Ketiga melakukan seleksi fitur menggunakan *Information Gain*. Keempat melakukan *data sampling* yaitu membagi dataset menjadi 2 bagian data training dan data testing dengan presentasi 70:30. Terakhir, menguji sampel data untuk menentukan metode mana yang memiliki skor akurasi tertinggi pada dataset penyakit jantung.

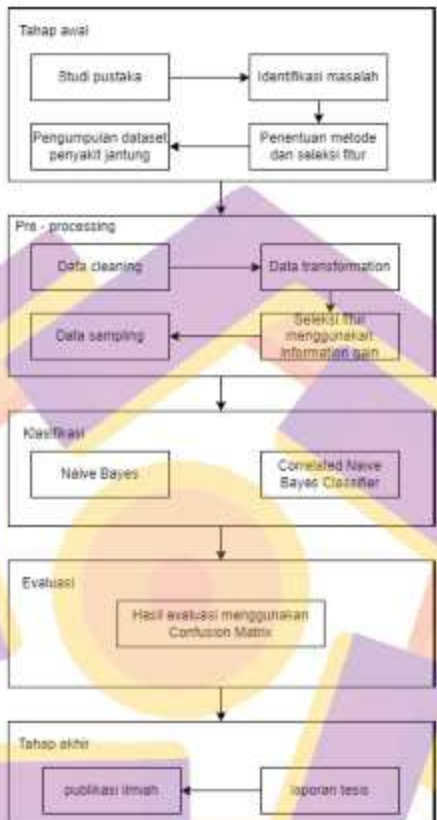
Adapun seperangkat komputer dan software yang digunakan untuk penelitian ini adalah:

1. Processor intel i5

2. RAM 4 GB
3. Harddisk 1 TB
4. Monitor, Mouse, dan Keyboard
5. Jupyter Notebook
6. Bahasa pemograman Phyton

3.4. Alur Penelitian

Alur penelitian yang dilakukan pada penelitian ini memiliki beberapa tahapan yaitu: Studi literatur, pre-processing, klasifikasi, evaluasi dan hasil. Pada penelitian ini alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Berikut adalah penjelasan tahapan pada Gambar 3.1:

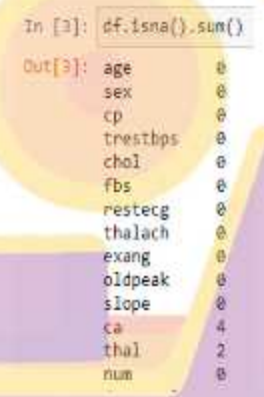
1. Tahap awal

Pada tahap awal penelitian terdapat beberapa proses seperti melakukan studi pustaka yaitu pengumpulan data tertentu dari jurnal dan buku ilmiah, selanjutnya melakukan identifikasi masalah yang masih terdapat dalam studi pustaka, kemudian

menentukan metode dan memilih fitur yang akan digunakan untuk memecahkan masalah, dan terakhir pengumpulan dataset yang akan digunakan untuk pengujian.

2. Pre processing

Dataset yang di *download* dari web UCI Machine Learning Repository tidak langsung digunakan untuk pengujian, masih perlu dilakukan *Pre-processing* yaitu mengubah data mentah menjadi data yang siap digunakan untuk *object* penelitian dan dapat diketahui hasil klasifikasi terbaik terhadap dataset penyakit jantung. Untuk mengetahui apakah dataset masih mengandung nilai yang hilang atau kosong (*missing value*), dapat menggunakan *script python* pada Gambar 3.2.



Gambar 3.2 *Script python* mengetahui nilai nol

Gambar 3.2 berisikan *script python* yang dijalankan menghasilkan *missing value* data pada atribut ca dan thal, oleh karena itu tahap pertama yang dilakukan dalam

Pre-processing adalah *Data Cleaning*. *Data Cleaning* merupakan proses perbaikan data yang masih terdapat nilai yang hilang atau kosong (*missing value*).

1) Cara mengatasi atribut ca dan thal yang terdapat *missing value* yaitu sama-sama mencari nilai yang sering muncul karena tipe data yang dimiliki berupa kategorial. Pada atribut ca terdapat 4 *missing value* sedangkan thal terdapat 2 *missing value*. Berikut adalah Contoh *script python* untuk memperbaiki data *missing value* pada Thal dapat dilihat pada Gambar 3.3.

```
In [6]: thal = df['thal'].mode()[0]
        df['thal'] = df['thal'].fillna(thal)
        df['thal'].isna().sum()

Out[6]: 0
```

Gambar 3.3 *Script python* memperbaiki data *missing value* pada Thal

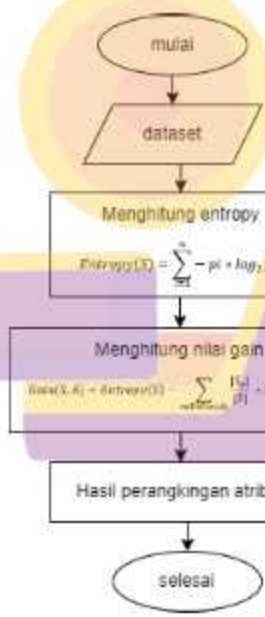
Gambar 3.3 berisikan *script python* yang dijalankan menghasilkan nilai 0 yang berarti sudah tidak ada data yang kosong. Tahapan *pre-processing* kedua adalah *Data Transformation* tahap ini mengubah variabel atau atribut yang menggandung nilai continous sehingga lebih optimal dalam proses *Naïve Bayes* (Arifin, T. and Syalwah, S., 2020).

Tabel 3.2 Transformation variabel menggunakan Discretization

No	Atribut	Nilai
1	Age	1 = < 45 (dewasa) 2 = 45 – 65 (lansia) 3 = > 65 (manula)
2	Sex	0 = female, 1 = male
3	Cp	1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 = asymptomatic
4	Trestbps	1 = < 100 (rendah) 2 = 100 – 140 (normal) 3 = > 140 (tinggi)
5	Chol	1 = < 200 normal 2 = 200 – 240 (sedang) 3 = >240 (tinggi)
6	Fbs	0 = false, 1 = true
7	Restecg	0 = normal 1 = having ST-T wave abnormaity 2 = having LV hypertrophy
8	Thalach	1 = 60-100 (normal) 2 = > 100 (tinggi)
9	Exang	0 = no, 1 = yes
10	Oldpeak	1 = < 1.5 (rendah) 2 = 1.6 – 2.55 (beresiko) 3 = > 2.55 (buruk)
11	Slope	1 = upsloping 2 = flat 3 = downslowping
12	Ca	0 -3 by fluoroscopy
13	Thal	3 = normal 6 = fixed defect 7 = reversable defect
14	Num	0 = < 50% diameter narrowing, absence 1 = > 50% diameter narrowing, presence

Proses transformasi pada Tabel 3.2 menggunakan discretization yang mengubah nilai baku atribut numerik menjadi label interval. Proses ini memiliki tujuan untuk mengubah semua variable continuous atau baku menjadi variabel dengan nilai kategorial atau interval.

Tahapan *Pre-processing* ketiga adalah proses seleksi fitur *Information Gain* ini dilakukan untuk membantu mengurangi *noise* yang diakibatkan oleh fitur yang tidak relevan dengan cara menghitung nilai *entropy (Syafitri Hidayatul AA, Yuita Arum S. A. A., 2018)*. Adapun algoritma dari seleksi fitur information gain dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart seleksi fitur information gain

Gambar 3.4 merupakan proses alur data pada *Information Gain* yang digunakan untuk mendapatkan nilai gain pada atribut. Memiliki beberapa tahap penyelesaian yaitu:

1. Memasukkan dataset
2. Perhitungan untuk menetukan nilai entropy
3. Perhitungan untuk menentukan nilai gain
4. Melakukan perangkingan atribut

Dan Tahapan *Pre-processing* keempat *Data Sampling* membagi dataset menjadi dua bagian yaitu data training sebesar 70% dan data testing sebesar 30% untuk dilakukan pegujian. Data training ini digunakan untuk pengembangan model sedangkan data testing digunakan untuk pengujian model, pembagian data dapat dilihat pada Tabel 3. 3.

Tabel 3. 3 Pembagian dataset penyakit jantung

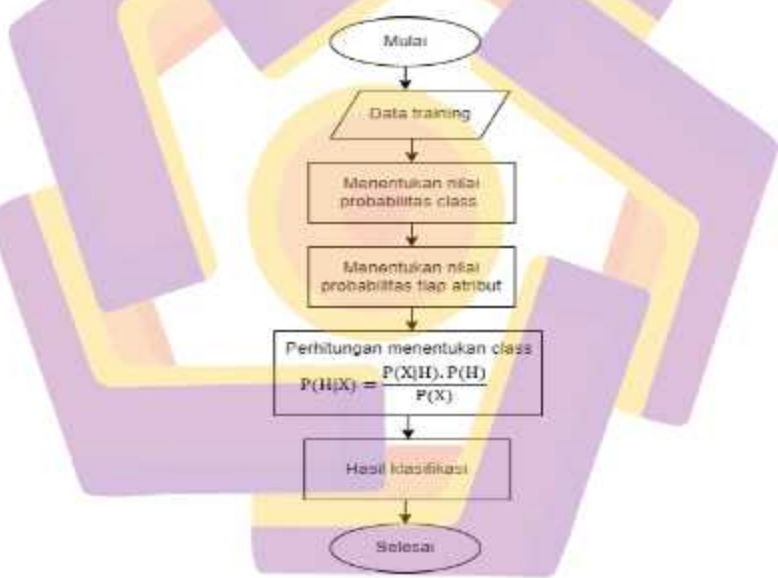
No	Class	Jumlah Record	Data Training (70%)	Data Testing (30%)
1	Absence	164	109	55
2	Presence	139	103	36
Jumlah		303	212	91

Tabel 3.3 berisikan class Absence sebanyak 164 record dan Presence sebanyak 139 dengan jumlah total record 303. Kemudian untuk pengujian dibagi menjadi dua yaitu training sebanyak 212 record dan testing sebanyak 91 record yang akan

digunakan untuk mengetahui nilai peforma pada metode Correlated Naïve Bayes Classifier menggunakan seleksi fitur Information gain.

3. Klasifikasi

Pada tahap ini melakukan klasifikasi menggunakan metode naïve bayes dan correlated naïve bayes classifier dengan menerapkan fitur seleksi information gain untuk klasifikasi penyakit jantung. Adapun *flowchart Naïve Bayes* dilihat pada Gambar 3.5 dan *Correlated Naïve Bayes Classifier* dilihat pada Gambar 3.6.



Gambar 3.5 *Flowchart Naïve Bayes*

Gambar 3.5 menjelaskan proses alur data *Naïve Bayes* yang akan digunakan untuk klasifikasi penyakit jantung. Memiliki beberapa tahap penyelesaian yaitu:

1. Memasukkan data training

   Contoh:

Tabel 3.4 Data training

Cp	Exang	Slope	Ca	Thal	Num
Typical angina	No	Downsloping	1	Fixed defect	Absence
Atypical angina	No	Upsloping	1	Normal	Absence
Asymptomatic	Yes	Upsloping	1	Normal	Absence
Asymptomatic	No	Flat	1	Fixed defect	Absence
Atypical angina	No	Upsloping	1	Reversable defect	Absence
Asymptomatic	Yes	Flat	1	Reversable defect	Presence
Typical angina	No	Upsloping	3	Normal	Absence
Non-anginal pain	No	Upsloping	1	Normal	Presence

2. Perhitungan untuk menetukan nilai probabilitas kelas atribut

   Contoh:

$$P(C1) = P(\text{Num "Absence"}) = \frac{6}{8} = 0{,}75$$

$$P(C2) = P(\text{Num "Presence"}) = \frac{2}{8} = 0{,}25$$

3. Perhitungan untuk menentukan nilai atribut terhadap kelasnya

   Contoh:

$$P(C1) = P(\text{Cp"typical angina"}) = \frac{2}{6} = 0{,}33$$

$$P(C1) = P(\text{Cp " atypical angina"}) = \frac{2}{6} = 0{,}33$$

$$P(C1) = P(\text{Cp " non} - \text{anginal pain"}) = \frac{0}{6} = 0$$

$$P(C1) = P(\text{Cp " asymptomatic"}) = \frac{2}{6} = 0{,}33$$

$$P(C2) = P(Cp \text{ " typical angina"}) = \frac{0}{2} = 0$$

$$P(C2) = P(Cp \text{ " atypical angina"}) = \frac{0}{2} = 0$$

$$P(C2) = P(Cp \text{ " non} - \text{anginal pain"}) = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Cp \text{ " asymptomatic"}) = \frac{1}{2} = 0{,}5$$

$$P(C1) = P(Exang \text{" no"}) = \frac{5}{6} = 0{,}83$$

$$P(C1) = P(Exang \text{" yes"}) = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Exang \text{" no"}) = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Exang \text{" yes"}) = \frac{1}{2} = 0{,}5$$

$$P(C1) = P(Slope \text{ " upsloping"}) = \frac{4}{6} = 0{,}66$$

$$P(C1) = P(Slope \text{ " flat"}) = \frac{1}{6} = 0{,}16$$

$$P(C1) = P(Slope \text{ " downsloping"}) = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Slope \text{ " upsloping"}) = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Slope \text{ " flat"}) = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Slope \text{ " downsloping"}) = \frac{0}{2} = 0$$

$$P(C1) = P(Ca\ "1") = \frac{5}{6} = 0{,}83$$

$$P(C1) = P(Ca\ "3") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Ca\ "1") = \frac{2}{2} = 1$$

$$P(C2) = P(Ca\ "3") = \frac{0}{2} = 0$$

$$P(C1) = P(Thal\ "normal") = \frac{3}{6} = 0{,}5$$

$$P(C1) = P(Thal\ "fixed\ defect") = \frac{2}{6} = 0{,}33$$

$$P(C1) = P(Thal\ "reversable\ defect") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Thal\ "normal") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Thal\ "fixed\ defect") = \frac{0}{2} = 0$$

$$P(C2) = P(Thal\ "reversable\ defect") = \frac{1}{2} = 0{,}5$$

4. Perhitungan data untuk menentukan class

Contoh:

Tabel 3.5 Data testing

Cp	Exang	Slope	Ca	Thal	Num
Asymtomatic	No	Upsloping	1	Normal	Absence

$$P(C1) = (P(Cp\ "asymtomatic") * P(Exang\ "no") * P(Slope\ "upsloping") * P(Ca\ "1") * P(Thal\ "normal") * P(Num\ "absence")$$

$$= 0{,}33 * 0{,}83 * 0{,}66 * 0.83 * 0{,}5 * 0{,}75$$

$$= 0{,}056$$

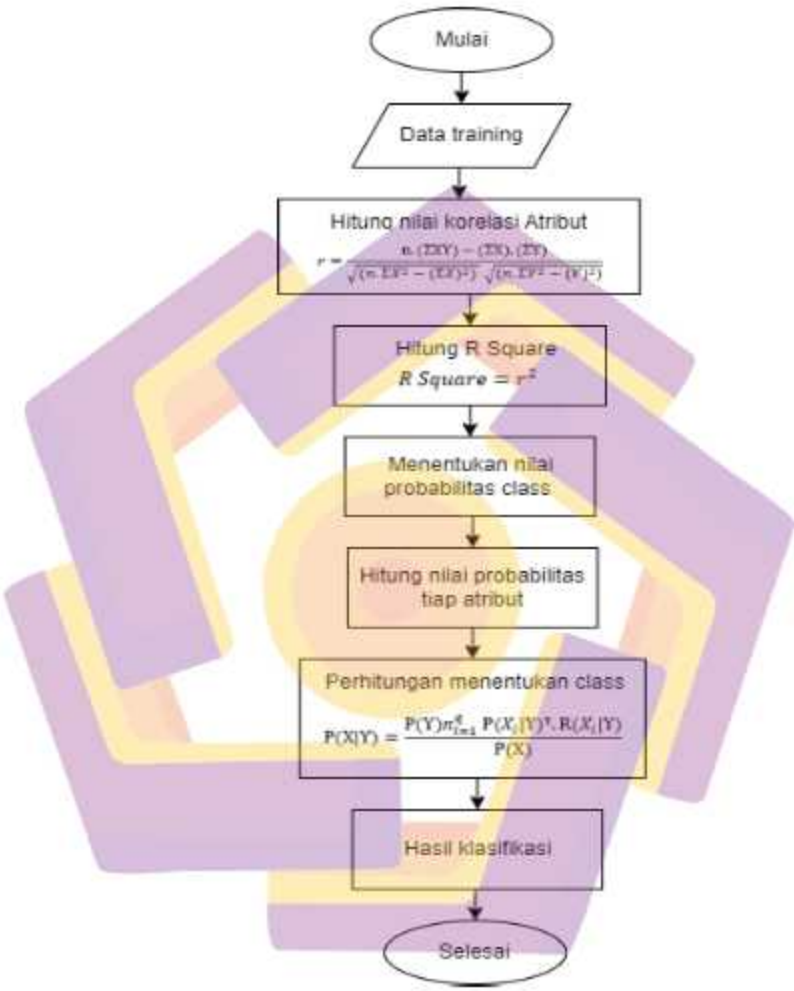
$$P(C2) = (P(Cp\ "asymtomatic") * P(Exang\ "no") * P(Slope\ "upsloping") * P(Ca\ "1") * P(Thal\ "normal") * P(Num\ "presence")$$

$$= 0{,}5 * 0{,}5 * 0{,}5 * 1 * 0{,}5 * 0{,}25$$

$$= 0{,}0156$$

5. Hasil klasifikasi naïve bayes

Dari data yang dicari adalah p(x) atau data ke 8 hasil klasifikasi yang didapatkan yaitu Absence.



Gambar 3.6 *Flowchart Corellated Naïve Bayes Classifier*

Pada Gambar 3.6 menjelaskan proses alur data *Corellated Naïve Bayes Classifier* yang akan digunakan untuk klasifikasi penyakit jantung. Memiliki beberapa tahap penyelesaian yaitu:

1. Memasukkan data training

Tabel 3.6 Data training

Cp	Exang	Slope	Ca	Thal	Num
1	1	3	1	6	1
2	1	1	1	3	1
4	2	1	1	3	1
4	1	2	1	6	1
2	1	1	1	7	1
4	2	2	1	7	2
1	1	1	3	3	1
3	1	1	1	3	2

2. Perhitungan korelasi tiap atribut (x) dengan class (y)

Tabel 3.7 Perhitungan korelasi

Cp (X1)	Exang (X2)	Slope (X3)	Ca (X4)	Thal (X5)	Num (Y)	Num (Y)	X_1^2	X_2^2	X_3^2	X_4^2	X_5^2	Y^2	$\Sigma X_1.Y$	$\Sigma X_2.Y$	$\Sigma X_3.Y$	$\Sigma X_4.Y$	$\Sigma X_5.Y$
1	1	3	1	6	Absence	1	1	1	9	1	36	1	1	1	3	1	6
2	1	1	1	3	Absence	1	4	1	1	1	9	1	2	1	1	1	3
4	2	1	1	3	Absence	1	16	4	1	1	9	1	4	2	1	1	3
4	1	2	1	6	Absence	1	16	1	4	1	36	1	4	1	2	1	6
2	1	1	1	7	Absence	1	4	1	1	1	49	1	2	1	1	1	7
4	2	2	1	7	Presence	2	16	4	4	1	49	4	8	4	4	2	14
1	1	1	3	3	Absence	1	1	1	1	9	9	1	1	1	1	3	3
3	1	1	1	3	Presence	2	9	1	1	1	9	4	6	2	2	2	6
ΣX_1	ΣX_2	ΣX_3	ΣX_4	ΣX_5		ΣY	ΣX_1^2	ΣX_2^2	ΣX_3^2	ΣX_4^2	ΣX_5^2	ΣY^2	$\Sigma X_1.Y$	$\Sigma X_2.Y$	$\Sigma X_3.Y$	$\Sigma X_4.Y$	$\Sigma X_5.Y$
21	10	12	10	38		10	67	14	22	16	206	14	28	13	15	12	48

3. Perhitungan untuk mendapatkan nilai R square

Contoh:

a. Perhitungan korelasi atribut Cp (X1) dengan kelasnya (Y)

$$r = \frac{8.(28) - (21).(10)}{\sqrt{(8.67) - (21)^2}\ \sqrt{(8.14) - (10)^2}}$$

$$r = \frac{224 - 210}{\sqrt{(536 - 441)}\ \sqrt{(112 - 100)}}$$

$$r = \frac{14}{\sqrt{95}\ \sqrt{12}} = \frac{14}{9{,}75 * 3{,}46} = \frac{14}{33{,}735} = 0.415$$

Dari nilai r Atribut Cp(X1) diatas dijadikan nilai korelasi R-Square

$$r\ Cp(X1) = 0{,}415^2 = 0.172$$

b. Perhitungan korelasi atribut Exang(X2) dengan kelasnya (Y)

$$r = \frac{8.(13) - (10).(10)}{\sqrt{(8.14) - (10)^2}\ \sqrt{(8{,}14) - (10)^2}}$$

$$r = \frac{104 - 100}{\sqrt{(112 - 100)}\ \sqrt{(112 - 100)}}$$

$$r = \frac{4}{\sqrt{12}\ \sqrt{12}} = \frac{4}{3{,}46 * 3{,}46} = \frac{4}{11{,}97} = 0{,}334$$

Dari nilai r Atribut Exang(X2) diatas dijadikan nilai korelasi R-Square

$$r\ \text{Exang(X2)} = 0{,}334^2 = 0{,}111$$

c. Perhitungan korelasi atribut Slope(X3) dengan kelasnya (Y)

$$r = \frac{8.(15) - (12).(10)}{\sqrt{(8.22) - (12)^2}\ \sqrt{(8.14) - (10)^2}}$$

$$r = \frac{120 - 120}{\sqrt{(176 - 144)}\ \sqrt{(112 - 100)}}$$

$$r = \frac{0}{\sqrt{32}\ \sqrt{12}} = \frac{0}{5.66 * 3{,}46} = \frac{0}{19{,}58} = 0$$

Dari nilai r Atribut Slope(X3) diatas dijadikan nilai korelasi R-Square

$$r\ \text{Slope(X3)} = 0^2 = 0$$

d. Perhitungan korelasi atribut Ca(X4) dengan kelasnya (Y)

$$r = \frac{8.(12) - (10).(10)}{\sqrt{(8.16) - (10)^2}\ \sqrt{(8.14) - (10)^2}}$$

$$r = \frac{96 - 100}{\sqrt{(128 - 100)}\ \sqrt{(112 - 100)}}$$

$$r = \frac{-4}{\sqrt{28}\sqrt{12}} = \frac{-4}{5{,}29 * 3{,}46} = \frac{-4}{18.30} = -0{,}219$$

Dari nilai r Atribut Ca(X4) diatas dijadikan nilai korelasi R-Square

$$r\ Ca(X4) = -0{,}219^2 = 0{,}048$$

e. Perhitungan korelasi atribut Thal(X5) dengan kelasnya (Y)

$$r = \frac{8.(48) - (38).(10)}{\sqrt{(8.206) - (38)^2}\ \sqrt{(8.14) - (10)^2}}$$

$$r = \frac{384 - 380}{\sqrt{(1648 - 1444)}\ \sqrt{(112 - 100)}}$$

$$r = \frac{4}{\sqrt{204}\sqrt{12}} = \frac{4}{14{,}28 * 3{,}46} = \frac{4}{49{,}41} = 0{,}081$$

Dari nilai r Atribut Thal(X5) diatas dijadikan nilai korelasi R-Square

$$r\ Thal(X5) = 0{,}081^2 = 0{,}007$$

4. Perhitungan untuk menentukan nilai atribut terhadap kelasnya

$$\mathrm{P(C1)} = \mathrm{P(Num\ "1")} = \frac{6}{8} = 0{,}75$$

$$\mathrm{P(C2)} = \mathrm{P(Num\ "2")} = \frac{2}{8} = 0{,}25$$

$$\mathrm{P(C1)} = \mathrm{P(Cp"1")} = \frac{2}{6} = 0{,}33$$

$$\mathrm{P(C1)} = \mathrm{P(Cp\ "\ 2")} = \frac{2}{6} = 0{,}33$$

$$P(C1) = P(Cp\ "3") = \frac{0}{6} = 0$$

$$P(C1) = P(Cp\ "4") = \frac{2}{6} = 0{,}33$$

$$P(C2) = P(Cp\ "1") = \frac{0}{2} = 0$$

$$P(C2) = P(Cp\ "2") = \frac{0}{2} = 0$$

$$P(C2) = P(Cp\ "3") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Cp\ "4") = \frac{1}{2} = 0{,}5$$

$$P(C1) = P(Exang"\ 1") = \frac{5}{6} = 0{,}83$$

$$P(C1) = P(Exang"\ 2") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Exang"\ 1") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Exang"\ 2") = \frac{1}{2} = 0{,}5$$

$$P(C1) = P(Slope\ "1") = \frac{4}{6} = 0{,}66$$

$$P(C1) = P(Slope\ "2") = \frac{1}{6} = 0{,}16$$

$$P(C1) = P(Slope\ "3") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(Slope\ "1") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(Slope\ "2") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(\text{Slope } "3") = \frac{0}{2} = 0$$

$$P(C1) = P(\text{Ca } "1") = \frac{5}{6} = 0{,}83$$

$$P(C1) = P(\text{Ca } "3") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(\text{Ca } "1") = \frac{2}{2} = 1$$

$$P(C2) = P(\text{Ca } "3") = \frac{0}{2} = 0$$

$$P(C1) = P(\text{Thal } "3") = \frac{3}{6} = 0{,}5$$

$$P(C1) = P(\text{Thal } "6") = \frac{2}{6} = 0{,}33$$

$$P(C1) = P(\text{Thal } "7") = \frac{1}{6} = 0{,}16$$

$$P(C2) = P(\text{Thal } "3") = \frac{1}{2} = 0{,}5$$

$$P(C2) = P(\text{Thal } "6") = \frac{0}{2} = 0$$

$$P(C2) = P(\text{Thal } "7") = \frac{1}{2} = 0{,}5$$

5. Perhitungan data untuk menentukan class

Tabel 3.8 Data Testing

Cp	Exang	Slope	Ca	Thal	Num
4	1	1	1	3	1

$$P(C1) = (P(Cp\ "4") * R(Cp\ "4")) + (P(Exang\ "1") * R(Exang\ "1")) + P(Slope\ "1") * R(Slope"1") + (P(Ca\ "1") * R(Ca"1")) + (P(Thal\ "3") * R(Thal\ "3")) * P(Num\ "1")$$

$$= (0{,}33 * 0.172) + (0{,}83 * 0{,}111) + (0{,}66 * 0) + (0.83 * 0{,}048) + (0{,}5 * 0{,}007) * 0{,}75$$

$$= 0{,}192458$$

$$P(C2) = (P(Cp\ "4") * R(Cp\ "4")) + (P(Exang\ "1") * R(Exang\ "1")) + P(Slope\ "1") * R(Slope"1") + (P(Ca\ "1") * R(Ca"1")) + (P(Thal\ "3") * R(Thal\ "3")) * P(Num\ "2")$$

$$= (0{,}5 * 0.172) + (0{,}5 * 0{,}111) + (0{,}5 * 0) + (1 * 0{,}048) + (0{,}5 * 0{,}007) * 0{,}25$$

$$= 0{,}190375$$

6. Hasil klasifikasi *Correlated Naïve Bayes*

   Dari data yang dicari adalah p(x) atau data ke 8 hasil klasifikasi yang didapatkan yaitu Absence

4. Evaluasi

Pada tahap ini melakukan evaluasi menggunakan Confusion Matrix bertujuan untuk menentukan nilai akurasi pada masing-masing model klasifikasi yang telah digunakan untuk pengujian.

5. Tahap akhir

Tahap akhir ini penelitian ini berupa pembuatan laporan tesis dan publikasi ilmiah yang berisikan penelitian menggunakan metode *Correlated Naïve Bayes Classifier* dan *Naïve Bayes* dengan menerapkan seleksi fitur *Information Gain* untuk mengetahui kinerja kedua metode tersebut mana yang paling baik dalam melakukan klasifikasi penyakit jantung.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

Penerapan metode *Correlated Naïve Bayes Classifier* dan seleksi fitur *Information Gain* bertujuan untuk mengetahui atribut yang berpengaruh dan meningkatkan akurasi terhadap dataset penyakit jantung yang didapat dari UCI Machine Learning Repository. Nilai akurasi akan meningkat atau tidak dapat dilihat dari hasil pengujian. Pengujian akan dilakukan dengan empat tahap, yaitu:

- Tahap pertama adalah pengujian menggunakan metode *Naïve Bayes*.
- Tahap kedua adalah pengujian menggunakan metode *Naïve Bayes* dan seleksi fitur *Information Gain*.
- Tahap ketiga adalah pengujian menggunakan metode *Correlated Naïve Bayes Classifier*.
- Tahap keempat adalah pengujian menggunakan metode *Correlated Naïve Bayes Classifier* dan seleksi fitur *Information Gain*.

4.1.1 Hasil pengujian metode *Naïve Bayes*

Tahap pengujian yang pertama menggunakan algoritma naïve bayes terdapat beberapa proses yaitu: memanggil data training, menentukan nilai probabilitas kelas, menentukan nilai probabilitas setiap atribut, perhitungan untuk menentukan class, memanggil data testing dan hasil pengujian data testing.

1. Memanggil data training

Proses ini merupakan langkah awal dalam melakukan pengujian yaitu memanggil file *Comma Separated Values* (CSV) yang dijadikan sebagai data training kemudian akan dilatih menggunakan algoritma *Naïve Bayes* dalam mencari model yang sesuai. Adapun *script python* memanggil file csv (Data Training) dapat dilihat pada Listing 4.1 :

```
1. Df_training=pd.read_csv('E:\\dataset\pengujian\df_training7
   0.csv')
2. df_training.head(10)
```

Listing 4.1 *Script phyton* memanggil file csv (Data Training)

Listing 4.1 berisikan perintah *script python* untuk memanggil data training, dengan cara menuliskan dimana letak file tersimpan.

2. Menentukan nilai probabilitas kelas

Proses ini melakukan perhitungan atribut *class* yang bertujuan untuk menghasilkan nilai probabilitas pada *class absence* dan *presence*. Adapun *script python* untuk menentukan nilai probabilitas pada atribut *class* dapat dilihat pada Listing 4.2 :

```
1. def round_value(value):
2. return round(value, 9)
3. prior_probability = dict()
4. count_labels=df_training[df_training.columns[1]].value_count
   s()
5. for label, value in count_labels.iteritems():
6. prob = round_value(value/ N)
7. prior_probability.update({label:prob})
8. print(prior_probability)
```

Listing 4.2 *Script phyton* perhitungan nilai probabilitas pada atribut kelas

Listing 4.2 berisikan perintah *script python* untuk menghitung nilai probabilitas *class*. Dengan cara mengetahui jumlah nilai masing-masing *class* kemudian dibagi

dengan jumlah total keseluruhan data. Sehingga dihasilkan *absence* memiliki nilai probabilitas: 0.514150943 dan *presence* memiliki nilai probabilitas: 0.485849057.

3. Menentukan nilai probabilitas setiap atribut

Proses ini melakukan perhitungan pada setiap atribut yang bertujuan wuntuk menghasilkan nilai probabilitas yaitu: age, sex, cp, trestbps, chol, fbs respect, thalach, exang, oldpeak, slope, ca, thal. Adapun *script phyton* untuk menentukan nilai probabilitas setiap atribut dapat dilihat pada Listing 4.3:

```
1. def get_independent_prob(x_data, y_value):
2.     x = next(iter(x_data))
3.     x_value = x_data.get(x)
4.     value   =   df_training[(df_training[x]   ==   x_value)   &
   (df_training['num'] == y_value)].count()[0]
5.     count_label     =     df_training[(df_training['num']     ==
   y_value)].count()[0]
6. return (value) / count_label
```

Listing 4.3 *Script phyton* perhitungan nilai probabilitas pada setiap Atribut

Listing 4.3 berisikan perintah *script python* untuk menghitung probabilitas pada atribut, dengan cara mengetahui jumlah nilai masing-masing atribut kemudian dibagi dengan total nilai class yang terdapat pada atribut num.

4. Perhitungan untuk menentukan *class*

Proses ini melakukan perhitungan untuk menentukan *class* pada data testing apakah *class* tersebut sudah sesuai dengan hasil *predict* yang didapatkan. Adapun *script python* perhitungan untuk menentukan *class* dapat dilihat pada Listing 4.4:

```
1. Def round_value(value):
2. return round(value, 9)
3. Def  get_attribute_probability(prior_probability,  data,  y,
   exclude=[]):
4.     sum_probability = None
5.     for x in data.iteritems():
6.         if x[0] in exclude:
```

```
7.              continue
8.          if x[0] == 'num':
9.              continue
10.           data = {
11.               x[0]:x[1]
12.           }
13.         if sum_probability == None: # init
14.   sum_probability=round_value(get_independent_prob(data,y))
15.         else:
16.   sum_probability=sum_probability*round_value(get_independe
   nt_prob(data, y))
17.   return sum_probability * prior_probability.get(y)
```

Listing 4.4 *Script python* perhitungan untuk menentukan class

Listing 4.4 berisikan *script python* perhitungan rumus metode *Naïve bayes* untuk menentukan record pada data testing apakah sudah sesuai dengan *class* prediksinya. Hasil *predict* didapatkan dengan dengan cara dilakukan perkalian masing masing hasil dari probabilitas atribut kemudian dikalikan lagi dengan probabilitas cclassnya.

5. Memanggil data testing

Proses ini membaca *file Comma Separated Values* (CSV) yang dijadikan sebagai data testing digunakan untuk menguji dan mengetahui peforma yang dimiliki naïve bayes pada dataset penyakit jantung. Adapun *script python* memanggil *file csv* (Data Testing) dapat dilihat pada Listing 4.5:

```
1. df_testing=pd.read_csv('E:\\dataset\pengujian\df_testing30.c
   sv')
2. df_testing.head()
```

Listing 4.5 *Script phyton* membaca *file csv* (Data Testing)

Listing 4.5 berisi perintah *script python* untuk memanggil data testing, dengan cara menuliskan dimana letak file tersimpan di dalam perangkat komputer.

6. Hasil pengujian data Testing

Proses ini menentukan *predict class* pada data testing dengan menggunakan model pembelajaran data training yaitu metode *Naïve Bayes*. Adapun *script python* pengujian data testing dapat dilihat pada Listing 4.6:

```
expected = []
predicts = []

absence_prob = []
presence_prob = []

df_testing_ga = df_testing.copy()

for index,data in df_testing_ga.iterrows():
  absence_probl=get_attribute_probability(prior_probability, data, 'absence')
  presence_probl=get_attribute_probability(prior_probability, data, 'presence')
  predict = 'absence' if absence_probl > presence_probl else 'presence'

    expected.append(data['num'])
    predicts.append(predict)

    absence_prob.append('{0:.9f}'.format(absence_probl))
    presence_prob.append('{0:.9f}'.format(presence_probl))

  df_testing_ga['predict'] = predicts
  df_testing_ga['num']=df_testing_ga['num'].replace({'absence':0, 'presence':1})
  df_testing_ga['predict']=df_testing_ga['predict'].replace({'absence':0, 'presence':1})

  df_testing_ga['absence_prob'] = absence_prob
  df_testing_ga['presence_prob'] = presence_prob
  df_testing_ga.head(30)
```

Listing 4.6 *Script python* untuk pengujian data testing

Listing 4.6 berisikan perintah *script python* menentukan *predict class* pada data testing, untuk memudahkan dalam *predict* pemahamanan maka label *absence* diganti dengan 0 dan label *presence* diganti dengan 1.

Hasil dari akurasi pengujian menggunakan metode *Naïve Bayes* dengan perbandingan data 70:30 yang dievaluasi menggunakan *Confusion Matrix*, dapat dilihat pada Tabel 4.1

Tabel 4.1 Hasil *Confusion Matrix* menggunakan *Naïve Bayes*

Accuracy: 84.61%	True Absence	True Presence	Class Precision
Pred. Absence	48	7	87.27%
Pred.Presence	7	29	80.55%
Class recall	87.27%	80.55%	

Berdasarkan Tabel 4.1 dapat ditentukan hasil akurasi pada klasifikasi penyakit jantung, sebagai berikut:

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

$$= \frac{29 + 48}{29 + 48 + 7 + 7}$$

$$= 84.61\%$$

4.1.2 Hasil Pengujian Naïve Bayes + Information Gain

Pada pengujian yang kedua menggunakan algoritma *Naïve Bayes* dengan seleksi fitur *Information Gain*, terdapat beberapa proses pada seleksi fitur yaitu: Menghitung nilai entropy total, Menghitung nilai *Information Gain*, Perangkingan nilai *Gain* aribut dan Hasil pengujian pada data testing.

1. Menghitung nilai entropy total

Proses ini menentukan nilai entropy total pada semua data yang ada pada masing-masing atribut, dengan cara menjumlah total probabilitas setiap kelas kemudian dilakukan perhitungan menggunakan rumus *entropy* untuk mendapatkan total nilai *entropy* kelas. Adapun *script python* menghitung nilai *entropy* dapat dilihat pada Listing 4.7:

```
1. from math import log2
2. def get_entropy(x, total):
3.     if x == 0:
4.         return 0
5.  entropy = (-x/total * log2(x/total))
6.  return entropy
```

Listing 4.7 *Script python* untuk menghitung nilai *entropy*

Listing 4.7 berisikan perintah *script python* untuk menghitung nilai entropy pada atribut penyakit jantung. Nilai entropy didapatkan dengan cara menjumlahkan sampel atribut pada atribut x kemudian dikalikan total keseluruhan atribut class, selanjutnya dikalikan kembali dengan log 2, terakhir dikurangin dengan jumlah sampel berikutnya yang dimiliki atribut x dan dikalikan total keseluruhan atribut class.

2. Menghitung nilai *Information Gain*

Proses ini melakukan perhitungan untuk mendapatkan nilai *Information Gain* pada masing-masing atribut yaitu age, sex, cp, trestbps, chol, fbs, respect, thalach, exang, oldpeak, slope, ca, thal. Dengan cara nilai *entropy* pada total data masing-masing atribut nilai dikurangkan dengan nilai *entropy* pada probabilitas setiap atribut dibagi dengan total keseluruhan *class* dikalikan dengan masing-masing nilai *entropy* pada setiap probabilitas atribut. Adapun *script python* menghitung nilai *Information Gain* dapat dilihat pada Listing 4.8

```
gain_entropy = dict()
for column in df_training:
    if column == 'num':
        continue

    # loop every attribute
    attribute_total_gain = 0
    for x in df_training[column].unique():
        data = {
            column:x
        }
        attribute_count_absence = get_count_row_with_condition_by_label(data, 'absence')
        attribute_count_presence = get_count_row_with_condition_by_label(data, 'presence')
        attribute_total_value = attribute_count_absence + attribute_count_presence

        attribute_entropy_absence = get_entropy(attribute_count_absence, attribute_total_value)
        attribute_entropy_presence = get_entropy(attribute_count_presence, attribute_total_value)

        attribute_total_entropy = attribute_entropy_absence + attribute_entropy_presence

        gain = attribute_total_value/N * attribute_total_entropy

        attribute_total_gain += gain

    gain = entropy_total - attribute_total_gain
    gain_entropy.update({column: round_value(gain)})

print(gain_entropy)
```

Listing 4.9 *Script python* menghitung nilai *Information Gain*

Listing 4.9 berisikan perintah *script python* yang berfungsi melakukan perhitungan pada atribut untuk mendapatkan nilai information gain. Hasil yang didapatkan sebagai berikut: 'age' sebesar 0.027404471, 'sex' sebesar 0.045646548, 'cp' sebesar 0.227411819, 'trestbps' sebesar 0.015545619, 'chol' sebesar 0.005719994, 'fbs' sebesar 0.001855936, 'restecg' sebesar 0.035249044, 'thalach' sebesar 0.041698997, 'exang' sebesar 0.12879388, 'oldpeak' sebesar 0.09704417, 'slope' sebesar 0.113525647, 'ca' sebesar 0.174215221, 'thal' sebesar 0.204942151.

3. Perangkingan nilai *Gain* aribut

Proses ini melakukan perangkingan pada setiap atribut dengan cara menentukan nilai *gain* yang tertinggi ke terendah, perangkingan ini digunakan untuk memudahkan mengetahui atribut yang memiliki nilai gain terendah akan digunakan untuk menseleksi fitur, nilai gain pada atribut yang terkecil akan diseleksi ataupun dihilangkan. Bertujuan untuk mengetahui atribut relasi yang kuat terhadap *class* yang diuji. Adapun *script python* untuk melakukan perangkingan atribut dapat dilihat pada Listing 4.10:

```
1. df_entropy = pd.DataFrame(gain_entropy.items(),
   columns=['attribyte','gain'])
2. df_entropy =
   df_entropy.sort_values(by=['gain'],ascending=False)
3. df_entropy
```

Listing 4.10 *Script python* perangkingan nilai *Gain* pada atribut

Listing 4.10 berisikan perintah *script python* untuk melakukan perangkingan sesuai dengan nilai gain dari terbesar hingga terkecil. Adapun urutan perangkingan atribut sebgai berikut: cp, thal, ca, exang, slope, oldpeak, sex, thalach, restecg, age, trestecg, age, trestbps, chol, dan fbs.

4. Hasil pengujian data testing

Proses ini menentukan *predict class* pada data testing dengan menggunakan model pembelajaran data training menggunakan metode *Naïve Bayes* setelah diseleksi menggunakan information gain. Adapun *script python* pengujian data testing dapat dilihat pada Listing 4.11:

```
1. exclude = []
2. expected = []
3. predicts = []
4.
5. absence_prob = []
```

```
6. presence_prob = []
7.
8. df_testing_ga = df_testing.copy()
9. df_testing_ga = df_testing_ga.drop(columns=exclude)
10.
11.   for index,data in df_testing_ga.iterrows():
12.   absence_prob1                                              =
   get_attribute_probability(prior_probability,              data,
   "absence",exclude=exclude)
13.       presence_prob1                                         =
   get_attribute_probability(prior_probability,              data,
   "presence",exclude=exclude)
14.       predict = "absence" if absence_prob1 > presence_prob1
   else "presence"
15.
16.   expected.append(data['num'])
17.   predicts.append(predict)
18.
19.   absence_prob.append('{0:.9f}'.format(absence_prob1))
20.   presence_prob.append('{0:.9f}'.format(presence_prob1))
21.
22.   df_testing_ga['predict'] = predicts
23.   df_testing_ga['num']   =   df_testing_ga['num'].replace({0:
   "absence", 1: "presence"})
24.   df_testing_ga['predict']                                   =
   df_testing_ga['predict'].replace({0:       "absence",      1:
   "presence"})
25.
26.   df_testing_ga['absence_prob'] = absence_prob
27.   df_testing_ga['presence_prob'] = presence_prob
28.
29.   df_testing_ga.head(30)
```

Listing 4.11 *Script python* pengujian data testing menggunakan menggunakan *Naïve Bayes + Information Gain*

Listing 4.11 berisikan perintah *script python* menentukan *predict class* pada data testing, untuk memudahkan pemahamanan maka *label absence* diganti dengan 0 dan *label presence* diganti dengan 1.

Hasil pengujian akurasi menggunakan metode *Naive Bayes* dengan seleksi fitur *Information Gain*, untuk perbandingan data 70:30 dievaluasi menggunakan *Confusion Matrix*. Adapun hasil pengujian *Confusion Matrix* menggunakan *Naïve Bayes* menggunakan *Information Gain* dapat dilihat pada Tabel 4.2:

Tabel 4.2 Hasil *Confusion Matrix* menggunakan *Naïve Bayes* menggunakan *Information Gain*

Accuracy: 89.01%	True Absence	True Presence	Class Precision
Pred. Absence	51	4	92.72%
Pred.Presence	6	30	83.33%
Class recall	89.47%	88.23%	

Berdasarkan Tabel 4.2 dapat ditentukan hasil akurasi pada klasifikasi penyakit jantung, sebagai berikut:

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

$$= \frac{30 + 51}{30 + 51 + 4 + 6}$$

$$= 89.01\%$$

Pengujian Seleksi Fitur Information Gain

Pengujian menggunakan metode *Naive Bayes* dengan seleksi fitur information dilakukan sebanyak 11 kali untuk mengetahui atribut mana yang memiliki relasi yang kuat terhadap dataset penyakit jantung. Adapun hasil dari seleksi fitur dapat dilihat pada tabel Tabel 4.3:

Tabel 4.3 Hasil pengujian menggunakan seleksi fitur pada *Naive Bayes*

	Jumlah Atribut yang digunakan										
	14	4	5	6	7	8	9	10	11	12	13
Akurasi	85	87	80	89	88	87	88	86	86	88	85
Presisi	81	83	78	88	88	85	86	81	81	86	81
Recall	81	83	69	83	81	81	83	83	83	83	81
F1-score	81	83	74	86	84	83	85	82	82	85	81

Tabel 4.3 terdapat nilai akurasi terbaik adalah 89% saat diuji dengan 6 atribut yaitu yaitu cp, thal, ca, exang, slope, dan num. Pengujian lain menggunakan atribut sebanyak 10 dan 11 medapatkan nilai akurasi yang sama sebesar 86%. Sedangkan penggunaan 5 atribut untuk pengujian mendapatkan akurasi terendah yaitu 80%.

4.1.3 Hasil Pengujian Correlated Naïve Bayes Classifier

Pada pengujian yang ketiga menggunakan algoritma *Correlated Naïve Bayes Classifier*, terdapat beberapa proses yaitu: Memanggil data training, menghitung korelasi atribut, menghitung nilai *R-square*, perhitungan untuk menentukan class, memanggil data testing dan hasil pengujian data testing.

1. Memanggil data training

Proses ini merupakan langkah awal dalam melakukan pengujian yaitu program akan membaca *file Comma Separated Values* (CSV) yang dijadikan sebagai data training kemudian akan dilatih menggunakan algoritma *Correlated Naïve Bayes* dalam mencari model yang sesuai. Adapun *script python* memanggil *file csv* (Data Training) dapat dilihat pada Listing 4.12:

```
1. df =pd.read_csv('E:\\dataset\pengujian\df_training70.csv')
2. df.head(10)
```

Listing 4.12 *Script python* memanggil *file csv* (Data Training)

Listing 4.12 berisi perintah *script python* untuk memanggil data training, dengan cara menuliskan dimana letak file tersimpan di dalam perangkat komputer.

2. Menghitung korelasi Atribut

Proses ini menghitung korelasi antar atribut untuk mendapatkan nilai korelasi yang dimiliki atribut. Nilai korelasi ini digunakan untuk perhitungan selanjutnya yaitu mencari nilai *R-Square* atribut. Proses untuk mencari nilai korelasi yang pertama semua nilai atribut dipangkat 2. Adapun *script python* perhitungan dipangkat 2 dapat dilihat pada Listing 4.13:

```
1. Df_pow_2 = df.pow(2)
2. df_pow_2.head(10)
```

Listing 4.13 *Script python* perpangkatan pada nilai Atribut

Listing 4.13 berisi perintah *script python* untuk melakukan perpangkatan pada nilai atribut. Proses yang kedua adalah dilakukan perkalian data antara semua atribut dengan *class*. Adapun *script python* perkalian atribut dengan *class* dapat dilihat pada Listing 4.14:

```
1. df_x_y = df_x.multiply(df[df.columns[-1]], axis=0)
2. df_x_y.head(20)
```

Listing 4.14 *Script python* perkalian atribut dengan *class*

3. Menghitung nilai *R-square*

Proses ini menghitung nilai korelasi atribut dengan *class*. Perhitungan dilakukan untuk mendapatkan nilai *R-square* pada masing-maing atribut yang digunakan. Adapun *script python* untuk mendapatkan nilai *r-square* setiap atibut dapat dilihat pada Listing 4.15:

```
1. r_square = dict()
2. sum_y = df[df.columns[-1]].sum()
3. sum_y_2 = df_pow_2[df_pow_2.columns[-1]].sum()
4.
5. Sfor column in df.columns[:-1]:
6.      sum_x = df[column].sum()
7.      sum_x_2 = df_pow_2[column].sum()
8.      sum_x_y = df_x_y[column].sum()
```

```
9.
10.    r = get_r_square(N, sum_x, sum_y, sum_x_2, sum_y_2,
    sum_x_y)
11.    r_square.update({column:r})
12.
13.    print(r_square)
```

Listing 4.15 *Script python* untuk mendapatkan nilai *r-square* setiap atribut

Listing 4.15 berisikan perintah *script python* melakukan perhitungan untuk mendaptkan nilai *r square*. Hasil yang didapatkan sebagai berikut: 'age' sebesar 0.029827583, 'sex' sebesar 0.061820232, 'cp' sebesar 0.184217543, 'trestbps' sebesar 0.017945829, 'chol' sebesar 0.004116464, 'fbs' sebesar 0.00257262, 'restecg' sebesar 0.04166421, 'thalach' sebesar 0.049993546, 'exang' sebesar 0.172032343, 'oldpeak' sebesar 0.120856052, 'slope' sebesar 0.119712574, 'ca' sebesar 0.212055118, 'thal' sebesar 0.269028475.

4. Perhitungan untuk menentukan *class*

Proses ini melakukan perhitungan untuk menentukan *class* pada data testing apakah *class* tersebut sudah sesuai dengan hasil klasifikasi yang didapatkan sebelumnya. Hasil *predict* didapatkan dengan cara melakukan perkalian pada semua nilai probabilitas atribut dan nilai masing-masing *R-Square* atribut, hasil dari perkalian yang memiliki nilai tertinggi akan menentukan *class* sebagai hasil *predict*. Adapun *script python* perhitunganya untuk menentukan *class* dapat dilihat pada Listing 4.16:

```
1. def get_attribute_probability(prior_probability, data,
   last_attribute, y, exclude=[]):
2.     sum_probability = 0
3.     data_last_attribute = dict ()
4.
5.     for x in data. iteritems():
6.         if x[0] in exclude:
7.             continue
8.
```

```
9.          if x[0] == 'num':
10.             continue
11.
12.         if x[0] == last_attribute:
13.             data_last_attribute = {
14.                 x[0]:x[1]
15.             }
16.             continue
17.
18.         data = {
19.             x[0]:x[1]
20.         }
21.         sum_probability +=
    round_value(get_independent_prob(data, y) *
    r_square.get(x[0]), 13)
22.
23.     last_probability =
    round_value(get_independent_prob(data_last_attribute, y) *
    r_square.get(last_attribute), 13)
24.     return round_value(sum_probability +
    (last_probability * prior_probability.get(y)), 13)
```

Listing 4.16 *Script phyton* Perhitungan untuk menentukan *class*

Listing 4.16 berisikan *script python* perhitungan rumus metode *Correlated Naïve bayes* untuk menentukan record pada data testing apakah sudah sesuai dengan class prediksinya.

5. Memanggil data testing

Proses ini membaca *file Comma Separated Values* (CSV) yang dijadikan sebagai data testing digunakan untuk menguji dan mengetahui peforma yang dimiliki *Correlated Naïve Bayes* pada dataset penyakit jantung. Adapun *script python* memanggil file csv (Data Testing) dapat dilihat pada Listing 4.17:

```
1. df_testing =
   pd.read_csv('E:\\dataset\pengujian\df_testing30.csv')
2. df_testing.head()
```

Listing 4.17 *Script phyton* memanggil *file csv* (Data Testing)

Listing 4.17 berisi perintah *script python* untuk memanggil data testing, dengan cara menuliskan dimana letak file tersimpan di dalam perangkat komputer.

6. Hasil pengujian data Testing

Proses ini menentukan *predict class* pada data testing dengan menggunakan model pembelajaran data training yaitu *Correlated Naïve Bayes*. Adapun *script python* pengujian data testing dapat dilihat pada Listing 4.18:

```
1. expected = []
2. predicts = []
3.
4. absence_prob = []
5. presence_prob = []
6.
7. df_testing_ga = df_testing.copy()
8.
9. for index,data in df_testing_ga.iterrows():
10.       absence_prob1 =
   get_attribute_probability(prior_probability, data, 'thal',
   1)
11.       presence_prob1 =
   get_attribute_probability(prior_probability, data, 'thal',
   2)
12.
13.   predict = 1 if absence_prob1 > presence_prob1 else 2
14.       expected.append(int(data['num']))
15.       predicts.append(predict)
16.
17.       absence_prob.append(absence_prob1)
18.       presence_prob.append(presence_prob1)
19.
20.   df_testing_ga['predict'] = predicts
21.   df_testing_ga['num'] =
   df_testing_ga['num'].replace({1:'absence', 2:'presence'})
22.   df_testing_ga['predict'] =
   df_testing_ga['predict'].replace({1:'absence',2:'presence'}
   )
23.
24.   df_testing_ga['absence_prob'] = absence_prob
25.   df_testing_ga['presence_prob'] = presence_prob
26.
27.   df_testing_ga.head(20)
```

Listing 4.18 *Script python* pengujian data testing menggunakan *Correlated Naïve Bayes*

Listing 4.18 berisikan *script python* menentukan *predict class* pada data testing, untuk memudahkan pemahamanan maka label 0 diganti dengan *absence* dan label 1 diganti dengan *presence*.

Hasil dari akurasi pengujian menggunakan metode *Correlated Naïve Bayes* dengan perbandingan data 70:30 yang dievaluasi menggunakan *Confusion Matrix*, dapat dilihat pada Tabel 4.4:

Tabel 4.4 Hasil *Confusion Matrix* menggunakan *Correlated Naïve Bayes*

Accuracy: 87.91%	True Absence	True Presence	Class Precision
Pred. Absence	53	2	96.36%
Pred.Presence	9	27	75%
Class recall	85.48%	93.10%	

Berdasarkan Tabel 4.4 dapat ditentukan hasil akurasi pada klasifikasi penyakit jantung, sebagai berikut:

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

$$= \frac{27 + 53}{27 + 53 + 2 + 9}$$

$$= 87.91\%$$

4.1.4 Hasil Pengujian Correlated Naïve Bayes Classifier + Information Gain

Pada pengujian yang kedua menggunakan algoritma *Correlated Naïve Bayes Classifier* dengan seleksi fitur *Information Gain*, terdapat beberapa proses pada seleksi fitur yaitu: Menghitung nilai entropy total, Menghitung nilai *Information Gain* dan Perangkingan nilai Gain atribut.

1. Menghitung nilai *entropy* total

Proses ini menentukan nilai *entropy* total pada semua data yang ada pada masing-masing atribut, dengan cara menjumlah total probabilitas setiap kelas kemudian dilakukan perhitungan menggunakan rumus *entropy* untuk mendapatkan total nilai *entropy* kelas. Adapun *script python* untuk mendaptkan nilai *entropy* total pada setiap atribut dapat dilihat pada Listing 4.19 :

```
1. from math import log2
2. def get_entropy(x, total):
3.     if x == 0:
4.         return 0
5.  entropy = (-x/total * log2(x/total))
6. return entropy
```

Listing 4.19 *Script python* untuk menghitung nilai *entropy*

Listing 4.19 berisikan perintah *script python* untuk menghitung nilai entropy pada atribut penyakit jantung. Nilai entropy didapatkan dengan cara menjumlahkan sampel atribut pada atribut x kemudian dikalikan total keseluruhan atribut class, selanjutnya dikalikan kembali dengan log 2, terakhir dikurangin dengan jumlah sampel berikutnya yang dimiliki atribut x dan dikalikan total keseluruhan atribut class.

2. Menghitung nilai *Information Gain*

Proses ini melakukan perhitungan untuk mendapatkan nilai *Information Gain* pada masing-masing atribut yaitu age, sex, cp, trestbps, chol, fbs, respect, thalach, exang, oldpeak, slope, ca, thal. Dengan cara nilai *entropy* pada total data masing-masing atribut nilai dikurangkan dengan nilai *entropy* pada probabilitas setiap atribut dibagi dengan total keseluruhan *class* dikalikan dengan masing-masing nilai *entropy* pada setiap probabilitas atribut. Adapun *script python* untuk mendapatkan nilai *Information Gain* dapat dilihat pada Listing 4.20:

```
gain_entropy = dict()
for column in df_training:
    if column == 'num':
        continue

    # loop every attribute
    attribute_total_gain = 0
    for x in df_training[column].unique():
        data = {
                    column:x
                }
                attribute_count_absence                    =
  get_count_row_with_condition_by_label(data, 1)
                attribute_count_presence                   =
  get_count_row_with_condition_by_label(data, 2)
                attribute_total_value                      =
  attribute_count_absence + attribute_count_presence

                attribute_entropy_absence                  =
  get_entropy(attribute_count_absence,
  attribute_total_value)
                attribute_entropy_presence                 =
  get_entropy(attribute_count_presence,
  attribute_total_value)

                attribute_total_entropy                    =
  attribute_entropy_absence + attribute_entropy_presence

                gain     =     attribute_total_value/N     *
  attribute_total_entropy

                attribute_total_gain += gain

        gain = entropy_total -  attribute_total_gain
        gain_entropy.update({column: round_value(gain)})

    print(gain_entropy)
```

Listing 4.20 *Script python* Menghitung nilai *Information Gain*

Listing 4.20 berisikan perintah *script python* yang berfungsi melakukan perhitungan pada atribut untuk mendapatkan nilai information gain. Hasil yang didapatkan sebagai berikut: 'age' sebesar 0.027404471, 'sex' sebesar 0.045646548, 'cp' sebesar 0.227411819, 'trestbps' sebesar 0.015545619, 'chol' sebesar 0.005719994, 'fbs' sebesar 0.001855936, 'restecg' sebesar 0.035249044, 'thalach'

sebesar 0.041698997, 'exang' sebesar 0.12879388, 'oldpeak' sebesar 0.09704417, 'slope' sebesar 0.113525647, 'ca' sebesar 0.174215221, 'thal' sebesar 0.204942151.

3. Perangkingan nilai Gain aribut

Proses ini melakukan perangkingan pada setiap atribut dengan cara menentukan nilai gain yang tertinggi ke terendah, perangkingan ini digunakan untuk memudahkan mengetahui atribut yang memiliki nilai gain terendah akan digunakan untuk menseleksi fitur, nilai gain pada atribut yang terkecil akan diseleksi ataupun dihilangkan. Bertujuan untuk mengetahui atribut relasi yang kuat terhadap *class* yang diuji. Adapun *script python* untuk melakukan perengkingan atribut dapat dilihat Listing 4.21:

```
1. df_entropy = pd.DataFrame(gain_entropy.items(),
   columns=['attribyte','gain'])
2. df_entropy =
   df_entropy.sort_values(by=['gain'],ascending=False)
3. df_entropy
```

Listing 4.21 *Script python* untuk melakukan perengkingan atribut

Listing 4.21 berisikan perintah *script python* untuk melakukan perangkingan sesuai dengan nilai gain dari terbesar hingga terkecil. Adapun urutan perangkingan atribut sebgai berikut: cp, thal, ca, exang, slope, oldpeak, sex, thalach, restecg, age, trestecg, age, trestbps, chol, dan fbs.

3. Hasil pengujian data Testing

Proses ini menentukan *predict class* pada data testing dengan menggunakan model pembelajaran data training menggunakan metode *Correlated Naïve Bayes* setelah diseleksi menggunakan *Information Gain*. Adapun *script python* pengujian data testing dapat dilihat pada Listing 4.22:

```
exclude = ['fbs','chol','trestbps','age','restecg']
expected = []
predicts = []

absence_prob = []
presence_prob = []

df_testing_ga = df_testing.copy()
df_testing_ga = df_testing_ga.drop(columns=exclude)

  for index,data in df_testing_ga.iterrows():
      absence_prob1 =
get_attribute_probability(prior_probability, data,'thal',
1,exclude=exclude)
      presence_prob1 =
get_attribute_probability(prior_probability, data,'thal',
2,exclude=exclude)

  predict = 1 if absence_prob1 > presence_prob1 else 2
    expected.append(data['num'])
    predicts.append(predict)

    absence_prob.append('{0:.13f}'.format(absence_prob1))
    presence_prob.append('{0:.13f}'.format(presence_prob1))

  df_testing_ga['predict'] = predicts
  df_testing_ga['num'] = df_testing_ga['num'].replace({1:
"absence", 2: "presence"})
  df_testing_ga['predict'] =
df_testing_ga['predict'].replace({1: "absence", 2:
"presence"})

  df_testing_ga['absence_prob'] = absence_prob
  df_testing_ga['presence_prob'] = presence_prob
  df_testing_ga.head(30)
```

Listing 4.22 *Script python* pengujian data testing menggunakan menggunakan *Correlated Naive Bayes + Information Gain*

Listing 4.22 berisikan perintah *script python* menentukan *predict class* pada data testing, untuk memudahkan pemahaman maka *label absence* diganti dengan 0 dan *label presence* diganti dengan 1.

Hasil pengujian akurasi menggunakan metode *Correlated Naive Bayes* dengan seleksi fitur *Information Gain*, untuk perbandingan data 70:30 yang dievaluasi menggunakan *Confusion Matrix*. Dapat dilihat pada Tabel 4.5:

Tabel 4.5 Hasil *Confusion Matrix* menggunakan *Correlated Naïve Bayes* menggunakan *Information Gain*

Accuracy: 91.20%	True Absence	True Presence	Class Precision
Pred. Absence	54	1	98.82%
Pred.Presence	7	29	80.55%
Class recall	88.52%	96.66%	

Berdasarkan Tabel 4.5 dapat ditentukan hasil akurasi pada klasifikasi penyakit jantung, sebagai berikut:

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

$$= \frac{29 + 54}{29 + 54 + 1 + 7}$$

$$= 91.20\%$$

Seleksi Fitur Information Gain

Pengujian menggunakan metode *Correlated Naive Bayes Classifier* dengan seleksi fitur information dilakukan sebanyak 11 kali untuk mengetahui atribut mana yang memiliki relasi yang kuat terhadap dataset penyakit jantung. Adapun hasil dari seleksi fitur dapat dilihat pada tabel Tabel 4.6:

Tabel 4.6 Hasil pengujian menggunakan seleksi fitur pada *Correlated Naive Bayes Classifier*

	Jumlah Atribut yang digunakan										
	14	4	5	6	7	8	9	10	11	12	13
Akurasi	88	87	84	91	87	88	88	88	88	88	88
Presisi	93	88	86	97	93	93	93	93	93	93	93
Recall	75	78	69	81	72	75	75	75	75	75	75
F1-Score	83	82	77	88	81	83	83	83	83	83	83

Pada Tabel 4.6 terdapat nilai akurasi terbaik adalah 91% saat diuji dengan 6 atribut yaitu yaitu cp, thal, ca, exang, slope, dan num. Pengujian lain menggunakan atribut sebanyak 8 sampai 13 tidak terjadi peningkatan nilai akurasi yang diperoleh sebesar 88%. Sedangkan penggunaan 5 atribut untuk pengujian mendapatkan akurasi terendah yaitu 84%.

4.2. Pembahasan

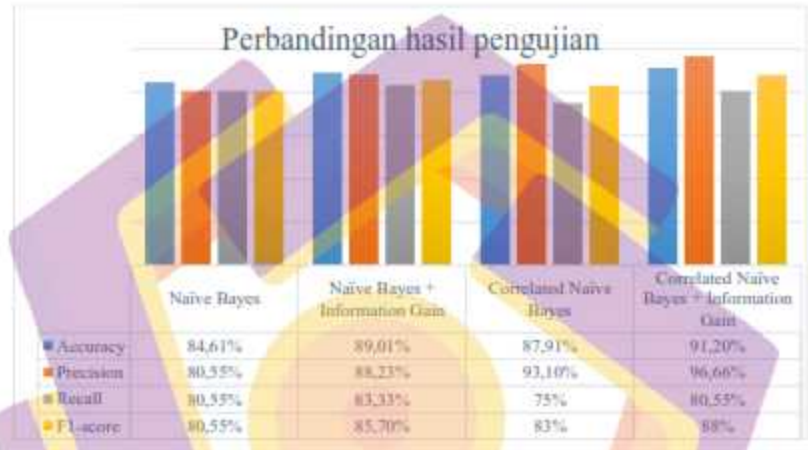
Berdasarkan hasil dari penelitian diatas, dalam tahap ini akan dilakukan evaluasi untuk mengetahui peningkatan nilai akurasi dari pengujian klasifikasi menggunakan beberapa metode yaitu: *Naive Bayes, Naive Bayes* dengan seleksi fitur *Information Gain, Corellated Naïve Bayes,* dan *Corellated Naïve Bayes* dengan seleksi fitur *Information Gain*. Untuk mempermudah pemahaman hasil pengujian berupa akurasi, presisi, recall dan f1-score akan dibuat kedalam bentuk tabel dan grafik.

4.2.1 Perbandingan hasil pengujian

Tabel 4.7 Hasil Penelitian

No	Metode	Accuracy	Precision	Recall	F1-score
1	Naïve Bayes	84.61%	80.55%	80.55%	80.55%
2	Naïve Bayes + Information Gain	89.01%	88.23%	83.33%	85.70%
3	Correlated Naïve Bayes	87.91%	93.10%	75%	83%
4	Correlated Naïve Bayes + Information Gain	91.20%	96.66%	80.55%	88%

Untuk memudahkan pemahaman mengenai perbandingan hasil akurasi dari keempat pengujian. Maka dibuatkan grafik perbandingan yang dapat dilihat pada Gambar 4.1:



Gambar 4.1 Grafik perbandingan penelitian

Berdasarkan Gambar 4.1 diketahui beberapa hasil klasifikasi yang telah dilakukan pengujian, dapat kita lihat pengujian menggunakan metode *Naïve Bayes* memiliki akurasi sebesar 84.61% sedangkan metode *Correlated Naïve Bayes* memiliki akurasi sebesar 87.91% memiliki kenaikan akurasi sebesar 3.3% setelah dihitung nilai korelasi antar atribut. Kemudian pengujian menggunakan seleksi fitur *Information Gain* pada *Naïve Bayes* mendapatkan nilai akurasi sebesar 89.01% sedangkan pada *Correlated Naïve Bayes* mendapatkan akurasi sebesar 91.20% memiliki kenaikan akurasi sebesar 2.19% setelah dihitung nilai korelasi antar atribut. Hasil klasifikasi yang tertinggi menggunakan metode *Correlated Naïve Bayes* mengalami peningkatan sebesar 3.29% sesudah diterapkan seleki fitur

menggunakan *Information Gain* dibandingkan sebelum diterapkan seleksi fitur information gain. Peningkatan akurasi metode Correlated naïve bayes disebabkan karena atribut yang digunakan untuk pengujian telah dilakukan seleksi fitur information gain dengan cara menentukan nilai entropy yang tertinggi, sehingga didapatkan beberapa atribut yang memiliki relasi kuat terhadap class pada dataset. Perbandingan hasil penelitian dengan peneliti sebelumnya juga dilakukan untuk dapat mengevaluasi hasil dari keseluruhan penelitian. Penelitian ini merupakan penelitian lanjutan dari penelitian sebelumnya dengan menggunakan data yang sama yaitu dataset penyakit jantung.

4.2.2 Perbandingan hasil penelitian

Berikut merupakan perbandingan hasil pengujian yang dilakukan oleh beberapa penelitian menggunakan dataset penyakit jantung dilihat Tabel 4.8

Tabel 4.8 Perbandingan Hasil Penelitian

Perbandingan beberapa pendekatan yang dilakukan				
No	Peneliti	Dataset	Metode	Akurasi
1	(Bianto, M. A. et al., 2020)	Heart Disease	Naïve Bayes	90.16%
2	(Saputra, D. et al., 2021)	Heart Disease	Naïve Bayes + PSO	85.26%
3	(Mubaroq, T. F. et al., 2019)	Heart Disease	Naïve Bayes + Discretization and Information Gain	85.55%
4	(Reddy, K. V. V. et al., 2021)	Heart Disease	Naïve Bayes + Principal Component Analysis	82.20%
5	(Marzuki, J. I. et al., 2018)	Data Diabetes	Correlated Naïve Bayes	67.15%
6	(Hairani, H. and Innuddin, M., 2020)	Data Thyroid	Correlated Naïve Bayes + Fitur Wrapper	79.38%
7	Pendekatan yang diusulkan	Heart Disease	Correlated Naïve Bayes + Information Gain	91.20%

Tabel 4.8 berisikan hasil dari beberapa penelitian sebelumnya menggunakan penyakit jantung dalam pengujian dilakukan menggunakan metode yang berbeda, karena beberapa metode digabungkan untuk meningkatkan akurasi *Naive Bayes*. Akurasi tertinggi ditemukan pada penelitian yang dilakukan dengan menggunakan fitur seleksi *Information Gain* pada metode *Correlated Naïve Bayes Classifier* mendapatkan akurasi sebesar 91.20%

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian, dapat diambil kesimpulan bahwa:

1. Tingkat akurasi tertinggi yang didapatkan dari metode *Naïve Bayes* sebelum dan sesudah diterapkan seleksi fitur *Information Gain* yaitu 84.61% dan 89.01%
2. Tingkat akurasi tertinggi yang didapatkan dari metode *Correlated Naïve Bayes* sebelum dan sesudah diterapkan seleksi fitur *Information Gain* yaitu 87.91% dan 91.20%
3. Berdasarkan hasil pengujian, metode terbaik untuk klasifikasi penyakit jantung adalah *Correlated Naïve Bayes* dengan seleksi fitur *Information Gain* dengan nilai akurasi sebesar 91.20%
4. Tingkat akurasi tertinggi diperoleh dengan menggunakan 6 atribut yaitu cp, thal, ca, exang, slope, dan num
5. Kenaikan akurasi dari penelitian sebelumnya sebesar 1.04%

5.2. Saran

Penelitian ini dapat dikembangkan dengan memperhatikan hal berikut:

1. Penggunaan metode Correlated Naïve Bayes Classifier dengan fitur seleksi information gain dapat dilakukan pada dataset yang berbeda apakah menghasilkan nilai akurasi yang lebih baik atau tidak, sehingga dapat digunakan sebagai alternatif dalam proses penyempurnaan metode yang sudah ada.
2. Penelitian berikutnya dapat mengabungkan metode *Correlated Naïve Bayes Classifier* dengan yang lain agar mendapatkan nilai akurasi yang lebih optimal.

DAFTAR PUSTAKA

Annisa, R. (2019), *Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung*, *Jurnal Teknik Informatika Kaputama (JTIK)*, *3*(1), 22–28.

Arifin, T. and Syalwah, S. (2020), *Prediksi Keberhasilan Immunotherapy Pada Penyakit Kutil Dengan Menggunakan Algoritma Naïve Bayes*, *Jurnal Responsif*, *2*(1), 38–43.

Bianto, M.A., Kusrini, K. and Sudarmawan, S. (2020), *Perancangan Sistem Klasifikasi Penyakit Jantung Mengunakan Naïve Bayes*, *Creative Information Technology Journal*, *6*(1), 75.

Dulhare, U.N. (2018), *Prediction system for heart disease using Naive Bayes and particle swarm optimization*, *Biomedical Research (India)*, *29*(12), 2646–2649.

Hairani, H. and Innuddin, M. (2020), *Kombinasi Metode Correlated Naive Bayes dan Metode Seleksi Fitur Wrapper untuk Klasifikasi Data Kesehatan*, *Jurnal Teknik Elektro*, *11*(2), 50–55.

Hasran (2020), *Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor*, *Indonesian Journal of Data and Science*, *1*(1), 1–4.

Kusrini and Luthfi, E.T. (2009), *ALGORITMA DATA MINING*, T. A. Prabawati, Ed. Yogyakarta: C.V ANDI OFFSET.

Lestari, M. (2014), *Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) untuk Mendeteksi Penyakit Jantung*, *Faktor Exacta*, *7*(September 2010), 366–371.

Marzuki, J.I., Mataram, K. and Bar, N.T. (2018), *KOMPARASI AKURASI METODE CORRELATED NAIVE BAYES CLASSIFIER DAN NAIVE BAYES CLASSIFIER UNTUK DIAGNOSIS PENYAKIT DIABETES Hairani , Gibran Satyo Nugraha , Mokhammad Nurkholis Abdillah , Muhammad Innuddin InfoTekJar (Jurnal Nasional Informatika dan Teknolog*, *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, *3*(1), 6–11.

Muharoq, T.F., Sugiharti, E. and Akhlis, I. (2019), *Application of Discretization and Information Gain on Naïve Bayes to Diagnose Heart Disease*, , *1*(October), 75–82.

Muktamar, B.A., Setiawan, N.A. and Adji, T.B. (2015), *Pembobotan Korelasi Pada Naïve Bayes Classifier*, *Seminar Nasional Teknologi Informasi Dan Multimedia 2015*, (2), 43–47.

Mustika, Ardila, Y., Manuhutu, A., Ahmad, N., Hasbi, I., Guntoro, ... Ernawati, I. (2021), *DATA MINING DAN APLIKASINYA*, Bandung: Penerbit Widina.

Nawawi, H.M., Purnama, J.J. and Hikmah, A.B. (2019), *Komparasi Algoritma Neural Network Dan Naïve Bayes Untuk Memprediksi Penyakit Jantung*, *Jurnal Pilar Nusa Mandiri*, *15*(2), 189–194.

Palaniappan, S. and Awang, R. (2008), *Intelligent heart disease prediction system using data mining techniques*, *AICCSA 08 - 6th IEEE/ACS International Conference on Computer Systems and Applications*, (December), 108–115.

Pusdatin kementrian Kesehatan RI (2014), *situasi kesehatan jantung*, *Angewandte Chemie International Edition*, *6(11)*, *951–952.*, 5–24.

Reddy, K.V.V., Elamvazuthi, I., Aziz, A.A., Paramasivam, S. and Chua, H.N. (2021), *Heart Disease Risk Prediction using Machine Learning with Principal Component Analysis*, *International Conference on Intelligent and Advanced Systems: Enhance the Present for a Sustainable Future, ICIAS 2021*, (December).

Saputra, D., Irmayani, W., Purwaningtias, D. and Sidauruk, J. (2021), *A Comparative Analysis of C4.5 Classification Algorithm, Naïve Bayes and Support Vector Machine Based on Particle Swarm Optimization (PSO) for Heart Disease Prediction*, *International Journal of Advances in Data and Information Systems*, *2*(2), 84–95.

Sciences, H. (2016), *KOMPARASI ALGORITMA MULTI LAYER PERCEPTRON DAN RADIAL BASIS*, *Jurnal Pilar Nusa Mandiri*, *4*(1), 1–23.

Siregar, A.M. and Puspabhuana, A. (2017), *DATA MINING: Pengolahan Data Menjadi Informasi dengan RapidMiner*, Sukoharjo: CV Kekata Group.

Syafitri Hidayatul AA, Yuita Arum S, A.A. (2018), *Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes*, *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 2546–2554.

Wanto, A. (2020), *Data Mining : Algoritma dan Implementasi*, T. Limbong, Ed. Yayasan Kita MenulisAccessed from https://www.google.co.id/books/edition/_/gAnfDwAAQBAJ?hl=id&sa=X&ved=2ahUKEwixrtGe3vf0AhVvT2wGHcXvD2wQ8fIDegQIBRAE.

Werdiningsih, I., Nuqoba, B. and Muhammadun (2020), *Data Mining Menggunakan Android, Weka, dan SPSS*, Surabaya: Airlangga University Press.

Yessy Nabella, F., Arum Sari, Y. and Cahya Wihandika, R. (2019), *Seleksi Fitur Information Gain Pada Klasifikasi Citra Makanan Menggunakan Hue Saturation Value dan Gray Level Co-Occurrence Matrix*, *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(2), 1892–1900.