

**TUGAS AKHIR – KI000000**

**DIAGNOSIS PENYAKIT BERDASARKAN GEJALA MENGGUNAKAN FP TREE**

**KEVIN ALIF FACHREZA**

**NRP 5114100128**

Dosen Pembimbing I

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom

Dosen Pembimbing II

Anny Yuniarti, S.Kom, M.Comp.Sc

Departemen Informatika

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018

***(Halaman ini sengaja dikosongkan)***



**TUGAS AKHIR – KI000000**

**DIAGNOSIS PENYAKIT BERDASARKAN GEJALA MENGGUNAKAN FP TREE**

**KEVIN ALIF FACHREZA**

**NRP 5114100128**

**Dosen Pembimbing I**

**Dr. Eng. Chastine Fatichah, S.Kom., M.Kom**

**Dosen Pembimbing II**

**Anny Yuniarti, S.Kom, M.Comp.Sc**

**Departemen Informatika**

**Fakultas Teknologi Informasi dan Komunikasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2018**

***(Halaman ini sengaja dikosongkan)***



**UNDERGRADUATE THESES – KI000000**

**DISEASE DIAGNOSIS BASED ON SYMPTOMS USING FP TREE**

**KEVIN ALIF FACHREZA**

**NRP 5114100128**

**First Advisor**

**Dr. Eng. Chastine Fatichah, S.Kom., M.Kom**

**Second Advisor**

**Anny Yuniarti, S.Kom, M.Comp.Sc**

**Department of Informatics**

**Faculty of Information Technology and Communication**

**Sepuluh Nopember Institute of Technology**

**Surabaya 2018**

***(Halaman ini sengaja dikosongkan)***

**LEMBAR PENGESAHAN**

**DIAGNOSIS PENYAKIT BERDASARKAN GEJALA MENGGUNAKAN FREQUENT PATTERN TREE**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

pada

Bidang Studi Kecerdasan Komputasional

Program Studi S-1 Departemen Informatika

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Oleh:

**KEVIN ALIF FACHREZA**

**NRP : 5114100128**

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. .....................

(NIP. 198410162008121002) (Pembimbing 1)

1. Anny Yuniarti, S.Kom, M.Comp.Sc ......................

(NIP. ) (Pembimbing 2)

**SURABAYA**

**JUNI, 2018**

***(Halaman ini sengaja dikosongkan)***

**DIAGNOSIS PENYAKIT BERDASARKAN GEJALA MENGGUNAKAN FP TREE**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | **:** | **Kevin Alif Fachreza** |
| **NRP** | **:** | **5114100128** |
| **Departemen** | **:** | **Informatika FTIK-ITS** |
| **Dosen Pembimbing 1** | **:** | **Dr. Eng. Chastine Fatichah, S.Kom., M.Kom** |
| **Dosen Pembimbing 2** | **:** | **Anny Yuniarti, S.Kom, M.Comp.Sc** |

# Abstrak

*Machine learning* merupakan salah satu bidang di teknologi informasi yang sedang naik daun. Teknologi ini dimanfaatkan diberbagai bidang kehidupan manusia, mulai dari perbankan, transportasi, sosial media, termasuk kesehatan. Potensi pemanfaatan *machine learning* pada bidang kesehatan sangatlah besar. Salah satunya adalah diagnosis penyakit. Diagnosis penyakit atau dikenal dengan tahap anamnesis pada kedokteran adalah suatu proses dimana dokter akan menanyakan kepada pasien gejala gejala yang dialami pasien. Sehingga dokter dapat memperkecil kemungkinan penyakit yang mungkin dialami pasien dan melakukan tes penunjang seperti lab atau radiologi untuk mendapatkan keputusan diagnosis final.

Banyak aplikasi yang dapat mendiagnosis penyakit, akan tetapi *user experience* yang buruk menyebabkan diagnosis meleset. Aplikasi tersebut biasanya akan meminta pengguna untuk memasukkan gejala mereka baik teks maupun berupa *checkboxes*. Padahal pasien tidak terlalu mengetahui gejala gejala spesifik atau yang berkorelasi yang dapat membantu dokter dalam mendiagnosis penyakit secara signifikan.

Aplikasi yang dibuat pada tugas akhir ini akan berfokus pada bagaimana pasien dapat memasukkan gejala yang dialami dengan spesifik dan sejelas mungkin. Sehingga dokter dapat memberikan diagnosis yang lebih baik. Untuk menyelesaikan masalah tersebut, akan digunakan algoritma *FP Tree*, yaitu salah satu metode *association rules* dimana algoritma ini dapat memetakan gejala gejala yang berkaitan, sehingga aplikasi dapat menanyakan gejala yang mungkin dialami oleh pasien berdasarkan gejala yang telah dimasukkan sebelumnya oleh pasien.

Dengan jumlah diagnosis penyakit sebanyak 66 penyakit, aplikasi ini memberikan hasil yang cukup baik. Dibuktikan dengan skenario pengujian dengan 198 data menghasilkan rata rata akurasi sebesar 69% dengan *classifier Naïve Bayes.* Dan rata rata akurasi sebesar 67% dengan *classifier SVM.* Aplikasi juga diujikan kepada dokter spesialis yang menghasilkan 80% hasil diagnosis relevan dengan gejala yang dimasukkan.

**Kata kunci: diagnosis, gejala, aplikasi**

**DISEASE DIAGNOSIS BASED ON SYMPTOMS USING FP TREE**

|  |  |  |
| --- | --- | --- |
| **Student’s Name** | **:** | **Kevin Alif Fachreza** |
| **Student’s ID** | **:** | **5114100128** |
| **Departemen** | **:** | **Informatika FTIK-ITS** |
| **First Advisor** | **:** | **Dr. Eng. Chastine Fatichah, S.Kom., M.Kom** |
| **Second Advisor** | **:** | **Anny Yuniarti, S.Kom, M.Comp.Sc** |

# Abstract

*Machine learning is currently one of the promising tech. This technology can be used in any fields, banking, transportation, social media, include health. Machine learning has a huge potency in health industry. One of them is diseases diagnosis. Diseases diagnosis or known as anamnesis in medical is a process where doctor will ask the patient about the symptoms or experience that patient felt. So doctor can narrow the possibility of diseases that the patient may suffer and do some lab and radiology tests to get final diagnosis decision*

*There’s a lot app that able to diagnose diseases, but with bad user experience made the diagnosis went fumble. Those apps will asks user to input their symptoms using text or checkboxes. Whereas patients don’t really know the specific symptoms or the symptoms that correlated to their main symptoms which can help doctor to diagnose disease significantly.*

*The app that made for this final project focused on how patient can input their symptoms specifically and clearly. So doctor can give better diagnosis decision. To solve that problem, FP Tree algorithm will be used, FP Tree is one of the association rules methods where this algorithm can map which symptoms is related to other symptoms, so the app can ask the next symptoms based on the previous symptoms that inputted by user.*

*With 66 diagnoses, this app gave a pretty good result. Proven by scenarios of testing with 198 datas, yielding 69% average accuracy with classifier Naïve Bayes. And 67% average accuracy with classifier SVM. This app also tested to a doctor specialist which came by a good result, 80% of the diagnosis result relevant to the inputted symptoms.*

***Keywords : diagnosis, symptoms, app***

# KATA PENGANTAR

Penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT.
2. Keluarga penulis
3. Saya Sendiri.

Surabaya, 07 Juni 2018

Kevin A Fachreza

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR KODE SUMBER

# BAB I PENDAHULUAN

Bab ini akan membahas latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran tugas akhir secara umum dapat dipahami.

## Latar Belakang

Kemajuan teknologi membuat informasi semakin mudah diakses. Masyarakat kini lebih haus informasi dibanding dengan generasi sebelumnya. Tidak terkecuali dalam bidang kesehatan. Masyarakat kini dapat membaca artikel kesehatan dengan mudah. Mendapatkan obat dengan mudah. Obat obatan kini bukan lagi sebuah rahasia dokter dan apoteker. Masyarakat kini cenderung mengetahui jenis jenis obat dan apa guna obat tersebut.

Sakit dapat terjadi pada siapapun. Mulai dari penyakit ringan hingga penyakit serius. Menurut statistik dari *UK Digital Health Report*, 1 dari 5 orang lebih memilih untuk melakukan diagnosis sendiri dengan bantuan *search engine* [1]. Hal ini tentu saja meningkatkan resiko kesalahan diagnosis pada pasien, dan dapat menyebabkan penyakit pasien semakin memburuk bahkan meninggal dunia.

Untuk mengatasi hal tersebut, diperlukan suatu solusi yang dapat memperkecil kesalahan masyarakat dalam mendiagnosis penyakit. Sehingga pasien lebih waspada dan tidak menganggap remeh gejala yang mereka alami. Solusi tersebut dapat dikemas dalam bentuk aplikasi yang didukung oleh kecerdasan buatan yang dapat mendiagnosis berdasarkan gejala gejala yang diberikan oleh pasien.

Dalam tugas akhir ini, akan digunakan *FP Tree* untuk dapat memberikan gejala gejala berkaitan dengan gejala utama pasien dan dapat memberikan diagnosis yang sesuai.

## Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut :

1. Bagaimana memperoleh data gejala dan diagnosis yang sesuai untuk digunakan pada algoritma *FP Tree*?
2. Bagaimana menentukan diagnosis yang cocok untuk gejala gejala yang telah diinput pasien berdasarkan hasil *FP Tree*?
3. Bagaimana cara menghasilkan pertanyaan gejala yang sesuai dengan jawaban pertanyaan gejala sebelumnya menggunakan *FP Tree*?

## Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Jumlah diagnosis terbatas sebanyak 66 diagnosis.
2. Implementasi dilakukan pada lingkungan kerja berbasis web.

## Tujuan Pembuatan Tugas Akhir

Tujuan pembuatan tugas akhir ini adalah menciptakan aplikasi yang dapat memberikan diagnosis berdasarkan gejala gejala yang diberikan oleh pengguna menggunakan *FP Tree*.

## Manfaat Tugas Akhir

Tugas akhir ini akan berguna untuk memberikan perkiraan penyakit kepada masyarakat awam, serta memberikan masukan diagnosis kepada dokter. Input data gejala dan kondisi pasien dapat dibaca oleh dokter yang ditunjuk oleh pasien sebagai pemeriksa dirinya. Sehingga akan membantu proses pemeriksaan dokter terhadap pasien.

## Metodologi

Metodologi yang akan digunakan pada tugas akhir ini adalah:

* 1. Penyusunan Proposal Tugas Akhir

Proposal akan berisi tentang pendahuluan, latar belakang, tujuan, manfaat, dan rumusan masalah. Yang akan didukung dengan penjelasan berupa tinjauan pustaka dan juga metode serta langkah langkah yang akan dilakukan untuk menciptakan produk.

* 1. Studi Literatur dan Wawancara

Sebelum pembuatan aplikasi, akan dilakukan studi literatur terkait aplikasi, dan juga melakukan wawancara kepada ahli, yang dalam tugas akhir ini adalah dokter untuk menyesuaikan metode yang dilakukan ahli untuk memberikan output yang diinginkan (diagnosis).

* 1. Analisis dan Desain

Pada tahap ini akan dilakukan analisis terkait bagaimana menciptakan model FP Tree yang baik dan juga arsitektur terkait. Serta akan di analisa juga dataset yang sesuai untuk digunakan pada model.

* 1. Implementasi

Model yang telah dihasilkan pada langkah sebelumnya akan di implementasikan pada server yang berisi model, dan juga aplikasi untuk klien berupa web.

* 1. Evaluasi

Aplikasi yang telah dibuat akan dievaluasi oleh dokter dan juga pengguna untuk menilai kesesuaian anamnesis yang dilakukan oleh aplikasi dan kepuasan pengguna terhadap aplikasi. Selain itu aplikasi juga akan di evaluasi secara sistem menggunakan akurasi dengan rumus sebagai berikut

TP : True Positive TN : True Negative

FP : False Positive FN : False Negative

* 1. Penyusunan Buku Tugas Akhir

Pada tahap ini akan dilakukan penyusunan laporan berupa buku tugas akhir yang menjelaskan dasar teori pada tugas akhir ini serta hasil implementasi pada tugas akhir.

## Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu rumusan permasalahan, batasan masalah, dan sistematika penulisan juga merupakan bagian dari bab ini.

1. Bab II Tinjauan Pustaka

Bab ini berisi penjelasan tentang metode, algoritma,dan *library* yang digunakan pada pembuatan aplikasi ini. Pembahasan akan berfokus pada algoritma *associative rules* terutama pada *FP Tree* dan juga metode klasifikasi yang digunakan yaitu *Naïve Bayes* dan *Support Vector Machine*.

1. Bab III Perancangan Perangkat Lunak

Bab ini aka membahas mengenai perancangan, desain, model, dan proses proses lain yang digunakan untuk pembuatan aplikasi juga model *FP Tree* serta model *Classifier*.

1. Bab IV. Implementasi

Bab ini akan menjelaskan proses pembuatan aplikasi dengan bahasa *Python* yang dibantu oleh library *Flask* dan juga *PHP* dengan *framework laravel.* Juga akan membahas pembuatan model menggunakan bahasa *Python* yang didukung dengan *ScikitLearn*.

1. Bab V. Hasil Uji Coba dan Evaluasi

Bab ini akan menjelaskan hasil percobaan yang dilakukan secara sistem dengan *data testing* yang telah diambil dari sumber buku dan web, juga hasil percobaan yang dilakukan oleh dokter.

1. Bab VI. Kesimpulan dan Saran

Bab ini akan membahas kesimpulan yang didapatkan dari hasil ujicoba dan juga akan membahas saran saran yang didapatkan baik dari penulis juga dari dokter yang melakukan ujicoba. Sehingga aplikasi dapat dikembangkan menjadi lebih baik

# BAB II DASAR TEORI

## Anamnesis

### 2.1.1 Pengertian Anamnesis

Proses akumulasi data yang menyangkut data medis pasien, latar belakang pasien, termasuk keluarga, lingkungan, pengalaman, terutama ingatan untuk digunakan dalam menganalisa kondisi [2].

### 2.1.2 Langkah Langkah Anamnesis

Umumnya anamnesis dilakukan sesuai dengan cara cara berikut [3]:

1. Pasien memberikan gejala

Pasien akan memberikan gejala yang dialami, misal : nyeri dada, badan panas.

1. Mendapatkan informasi lebih dalam mengenai gejala

Dokter akan menanyakan lebih lanjut terkait gejala yang dialami pasien. Misal pada nyeri dada, dokter akan menanyakan pertanyaan berikut kepada pasien.

* Dimana tepatnya letak nyeri pada dada?
* Sejak kapan nyeri dada dirasakan?
* Apakah nyeri dada sering sekali muncul, terkadang muncul, atau jarang muncul?
* Apakah rasa nyeri berpindah?
* Jika nyeri dada kambuh, berapa lama biasanya nyeri dada tersebut terasa?
* Apakah nyerinya semakin sakit atau semakin tidak sakit?
* Dalam skala 1-10, seberapa sakit yang anda rasakan?

1. Mencari gejala lain yang dialami pasien

Dokter akan menanyakan apakah pasien mengalami gejala lain yang mungkin berkaitan dengan gejala sebelumnya. Misal : Nyeri dada, maka mungkin pasien akan mengalami sulit bernapas.

1. Menanyakan tindakan/obat yang sudah dilakukan terhadap gejala tersebut

Dokter akan menanyakan tindakan atau obat apa yang telah dikonsumsi pasien. Lebih lanjut dokter akan menanyakan terkait dosis, nama obat, dan seberapa sering pasien mengonsumsi obat atau melakukan tindakan tersebut.

1. Menanyakan informasi kesehatan keluarga

Dokter akan menanyakan informasi keluarga yang mungkin memiliki penyakit yang berkaitan yang bersifat genetik. Misal : diabetes

1. Menanyakan informasi lingkungan keseharian

Dokter akan menanyakan bagaimana keseharian pasien, apakah pasien merokok, atau apakah pasien menggunakan obat obatan terlarang.

1. Menanyakan informasi lain terkait sistem tubuh lain yang tidak tercakup pada gejala

Dokter akan menanyakan apakah ada sistem tubuh lain yang terganggu. Umumnya sistem yang akan ditanyakan dokter adalah sebagai berikut :

* Cardiovaskular
* Pernapasan
* Pencernaan
* Saraf
* Genital
* Muskuloskeletal
* Kejiwaan

1. Mengulas ulang keluhan yang diberikan pasien

Dokter akan mengulas poin poin penting yang diberikan pasien sebelum memberikan diagnosis.

1. Dokter memberikan diagnosis

## 2.2 FP Tree Growth

### 2.2.1 Pengertian FP Tree Growth

FP (Frequent Pattern) Tree Growth merupakan salah satu algoritma associative rules yang sering digunakan pada berbagai permasalahan data mining. Algoritma ini sendiri bertujuan membuat rules yang didasarkan pada tree yang dibuat berdasarkan dataset yang diberikan.

Seperti *tree* pada umumnya, *tree* pada FP Tree Growth juga memiliki root, node dan juga leaf. Pada FP Tree Growh penempatan node akan didasarkan pada support pada setiap attribute pada sebuah data. Sehingga jika dilihat semakin tinggi posisi dari suatu node maka dapat dipastikan node tersebut memiliki support yang lebih tinggi daripada *child*-nya.

### 2.2.2 Support

Support adalah indikasi seberapa sering *item* akan muncul pada dataset. [4]

*support(A→B) = P (A∪ B)*

### 2.2.3 Confidence

Confidence mengindikasikan jumlah dari pernyataan if-else yang bernilai benar.[4]

*confidence(A→B) = P(B|A)*

### 2.2.4 Algoritma FP Tree Growth

FP Tree Growth memiliki algoritma sebagai berikut [5]:

Input : Dataset, dan minimum support

Output : FP Tree

Method :

1. Scan database, dan mengumpulkan kumpulan frequent items, dan support untuk setiap frequent items. Urutkan data tersebut sesuai dengan nilai support secara descending.
2. Buat root dari tree
3. Pilih salah satu frequent item dan buat node untuk setiap item. Lanjutkan hingga item dari set tersebut habis.
4. Jika node telah terbuat untuk item tertentu, maka atribut jumlah akan ditambahkan sesuai dengan frekuensi dia muncul pada node tersebut.
5. Ulangi langkah 3 dan 4 hingga tree terbuat.

## 2.4 Naïve Bayes

Naïve Bayes adalah salah satu algoritma *supervised* pada data mining. Naïve Bayes adalah sebuah *classifier* berbasis probabilitas yang sederhana yang menghitung frekuensi dan kombinasi nilai pada dataset [6]. Algoritma Naïve Bayes berbasiskan pada *Bayes Theorem*. Dengan persamaan sebagai berikut [7] :

## 2.5 Support Vector Machine

Support Vector Machine atau SVM adalah sebuah *classifier* yang menggolongkan dengan cara membagi data menjadi area yang terpisah atau *hyperplane* [8]. Dengan adanya hyperplane tersebut data data yang berbeda dapat dikategorikan menjadi 2 data berbeda. *Hyperplane* terbaik adalah *hyperplane* yang dapat memberikan margin terbesar dari 2 kategori [9].

## 2.6 Scikit Learn

*Scikit Learn* adalah library machine learning gratis berbasiskan bahasa pemrograman *Python*. Library ini memiliki banyak fitur mulai dari klasifikasi, regresi, kluster termasuk SVM, Random Forest, Gradient Boosting, KMeans, DBScan dan didesain untuk bekerja dengan *library* numerik dan ilmiah milik Python, *NumPy* dan *SciPy* [10].

## 2.7 Python – Flask

Flask adalah sebuah microframework untuk python [11]. Flask umumnya digunakan untuk membuat web. Flask tidak memiliki *database abstraction layer, form validation,* dan banyak fitur lainnya yang mana library lain sudah ada dan dapat mengatasi masalah tersebut. Flask mungkin library mikro namun mampu untuk digunakan dalam berbagai kebutuhan [12].

## 2.8 PHP - Laravel

Laravel adalah sebuah framework untuk web dengan syntax yang ekspresif dan elegan. Laravel mudah diakses, kuat, dan memiliki *tools* untuk aplikasi besar dan kuat. [13]. Berbeda dengan flask, laravel berbasiskan bahasa pemrograman PHP.

# BAB III PERANCANGAN

## 3.1 Perancangan Data

Data yang akan digunakan pada aplikasi ini berasal dari berbagai sumber buku dan website. Penulis mengekstrak informasi data dari buku dan website dan memasukkannya pada database.

Data ini terdiri dari 858 data kasus, yang memiliki 66 diagnosis atau kelas. Serta 375 atribut. Data ini akan dibagi menjadi 2 kelompok yaitu data belajar dan data pengujian. Data belajar terdiri dari 10 kasus untuk setiap kelas. Dan data pengujian terdiri dari 3 kasus untuk setiap kelas.

## 3.2 Desain Sistem Secara Umum

Rancangan aplikasi terbagi dalam 3 tahap. Tahap pembuatan model FP Tree, tahap pembuatan model *classifier*, dan tahap pembuatan aplikasi untuk user. Pada tahap pembuatan model FP Tree, data akan difilter dengan support sesuai dengan skenario. Lalu data support tersebut akan di training untuk membuat rules sesuai dengan data.

Pada tahap pembuatan model *classifier,* dataset akan di filter berdasarkan support, yang mana akan di training dengan *Naïve Bayes* dan juga *SVM.* Lalu masing masing hasil model akan di ujikan dengan data pengujian.

Tahap pembuatan aplikasi user akan menggabungkan antara hasil rules dengan model *classifier*. Aplikasi ini akan dibuat dengan backend Laravel dan front-end Javascript Vanilla.

bagan

## 3.4 Perancangan Model FP Tree

Dataset akan diolah dengan algoritma FP Tree sesuai dengan algoritma pada bab sebelumnya. Algoritma akan diimplementasikan dengan bahasa pemrograman Python.

Atribut pada masing masing data akan difilter berdasarkan support. Support yang akan diskenariokan adalah 3, 5, 8, 10, 12, 15, 18, 20, dan 23. Setelah di filter, data akan dibentuk tree berdasarkan urutan support dari yang terbanyak hingga terkecil. Setelah tree terbentuk, maka akan dicari *Frequent Pattern* dari setiap node. Dari *Frequent Pattern* tersebut maka kita dapat mengekstrak informasi rules beserta confidence. Confidence tidak akan di filter mengingat jumlah rules yang tidak begitu banyak.

Rules tersebut akan disimpan pada sebuah tabel pada database yang memiliki atribut id, rules input, rules output dan probability. Sehingga nantinya pada akhir proses akan memunculkan 9 tabel sesuai dengan jumlah skenario.

## 3.5 Perancangan Model Classifier Naïve Bayes

Naïve Bayes yang digunakan pada model adalah Gaussian Naïve Bayes karena dataset memiliki format binary. Model Naïve Bayes akan dibuat berdasarkan dataset yang telah di filter berdasarkan support yang sama dengan skenario. Setelah itu dataset yang telah di filter tersebut akan di masukkan sebagai data belajar dari model Naïve Bayes yang akan dibuat. Model Naïve Bayes akan dibuat menggunakan library *Scikit Learn*. Nantinya model ini akan diujikan dengan data testing yang telah ditetapkan sebelumnya.

Output dari model ini adalah 5 kemungkinan diagnosis yang memiliki *probability* tertinggi. Dengan format diagnosis dan persentase kemungkinan untuk setiap diagnosis.

## 3.6 Perancangan Model Classifier Support Vector Machine

Pada tahap ini akan dibuat model SVM dengan kernel linear. Pada Scikit Learn lebih dikenal dengan SVC. Sama seperti Naïve Bayes, model SVM juga akan dibuat berdasarkan data belajar yang telah di filter berdasarkan support. Model ini dibuat dengan library *Scikit Learn*.

Output dari model ini adalah 5 kemungkinan diagnosis yang memiliki *probability* tertinggi. Dengan format diagnosis dan persentase kemungkinan untuk setiap diagnosis.

## 3.7 Perancangan Metrik

### 3.7.1 Perancangan Metrik Akurasi

Metrik akurasi akan didasarkan pada hasil klasifikasi dengan probabilitas klasifikasi tertinggi dari model classifier dengan hasil yang seharusnya. Dengan rumus sebagai berikut :

### 3.7.2 Perancangan Metrik IR

Metrik ini dihitung berdasarkan hasil pada data training yang akan dibandingkan dengan hasil model yang memiliki 5 hasil probabilitas tertinggi. Dan data benar tersebut akan dihitung berdasarkan peringkat, 100 untuk peringkat pertama dan turun 20 untuk setiap peringkat sehingga peringkat 5 akan mendapatkan nilai 20 dan jika data tidak muncul akan memberikan nilai 0. Atau disederhanakan menjadi rumus berikut :

Selain itu juga akan dihitung nilai kemunculan dari hasil diagnosis. Sehingga jika ada salah satu hasil klasifikasi dari 5 hasil yang sesuai dengan hasil pada data training maka akan dinilai benar.

## 3.8 Perancangan Aplikasi

Pada sub-bab ini akan dijelaskan bagaimana perancangan desain aplikasi. Perancangan aplikasi akan dibagi menjadi 3 yaitu :

* 1. Perancangan Desain Aplikasi Secara Umm
  2. Perancangan Output Tanya Jawab
  3. Perancangan Output Diagnosis

### 3.8.1 Perancangan Desain Aplikasi Secara Umum

Aplikasi akan dimulai dengan menanyakan identitas pasien, nama, usia dan juga jenis kelamin. Setelah itu pasien akan menginputkan salah satu gejala yang dirasa paling mengganggu pasien. Setelah itu aplikasi akan menanyakan pertanyaan gejala yang mungkin berikutnya berdasarkan jawaban dari pertanyaan gejala sebelumnya. Pertanyaan akan memiliki 2 jawaban ya atau tidak.

Aplikasi akan menghentikan pertanyaan dan akan memberikan jawaban jika aplikasi telah menanyakan 20 aplikasi atau pasien telah memberikan 10 jawaban “ya” pada pertanyaan yang ditanyakan.

### 3.8.2 Perancangan *Output* Tanya Jawab

Pertanyaan selanjutnya akan diberikan berdasarkan jawaban jawaban dari pertanyaan sebelumnya. Yang mana jawaban akan diolah oleh server yang akan mengekstrak kemungkinan pertanyaan selanjutnya berdasarkan rules yang telah dibuat dari model FP Tree.

Adapun olahan khusus, seperti gejala yang memiliki sub gejala, seperti demam yang memiliki sub “demam tinggi” dan “demam lebih dari 3 hari”, maka mesin akan langsung menanyakan sub gejala sub gejala tersebut jika telah mendapai salah satu sub gejala ditanyakan. Serta pengolahan data khusus yang didasari dari input identitas pasien yaitu usia dan jenis kelamin. Adapun berikut adalah pembagian kategori menurut usia :

1. Bayi 0 – 1
2. Balita dan Remaja 3 - 18
3. Dewasa 18 – 65
4. Tua > 65

### 3.8.3 Perancangan *Output* Diagnosis

Jika aplikasi sudah selesai menanyakan pertanyaan, maka aplikasi akan mengirimkan data gejala yang dialami pasien ke server *classifer* yang dijalankan menggunakan flask dan akan mengolahnya pada model *classifier*. Setelah itu server ini akan mengirimkan kembali hasil data ke aplikasi. Dan aplikasi akan menampilkan 5 kemungkinan diagnosis tertinggi beserta persentase.

# BAB IV IMPLEMENTASI

## 4.1 Lingkungan Implementas

## 4.2 Implementasi FP Tree

## 4.3 Implementasi Classifier

### 4.3.1 Classifier Naïve Bayes

### 4.3.2 Classifier Support Vector Machine

//

## 4.4 Implementasi Metrik Analisis

### 4.4.1 Implementasi Metrik Akurasi

### 4.4.2 Implementasi Metrik IR

## //

## 4.5 Implementasi Simulasi Aplikasi

### 4.5.1 Output Tanya Jawab

### 4.5.2 Output Classifier

### 4.5.3 Interface User

# BAB V PENGUJIAN DAN EVALUASI

## 5.1 Lingkungan Pengujian

## 5.2 Data Uji Coba

## 5.3 Skenario Uji Coba

## 5.4 Hasil Uji Coba

### 5.4.1 Performa Pada Support = 3

### 5.4.2 Performa Pada Support = 5

### 5.4.3 Performa Pada Support = 8

### 5.4.4 Performa Pada Support = 10

### 5.4.5 Performa Pada Support = 12

### 5.4.6 Performa Pada Support = 15

### 5.4.7 Performa Pada Support = 18

### 5.4.8 Performa Pada Support = 20

### 5.4.9 Performa Pada Support = 23