Application of Recommendations for Diagnosis of Diseases in the Field of Hematology and Oncology With Forward Chaining Algorithm

Abiel Filetus*1, Ricky Raymond2, Prittania Friska Octavia3, Halim Agung4
1,2,3,4Fakultas Teknologi dan Desain Universitas Bunda Mulia
E-mail:*1**abielfiletus@gmail.com**, 2rickyraymond71@gmail.com, 3prittaniafriska@gmail.com,
4halimagungubm@gmail.com

Abstract

The COVID-19 pandemic has made many changes in everything, both from the industrial world and human habits. Illness is one of the things that cannot be postponed in its completion, so that people can consult a doctor to be able to recognize the disease suffered by the community In the past, consultations had to come to the hospital, nowadays there are innovations in the field of Information Technology that make it easier for people to consult. The expert system is one of the fastest growing innovations. The expert system is one of the fastest growing innovations. Forward chaining is one of the algorithms of an expert system to get a conclusion through existing facts / data based on predetermined rules. Forward chaining can provide good disease conclusions with a 100% percentage of 30 diseases and 15 test cases. However, a diagnosis based on symptoms according to doctors only has a confidence level of 50-60%. So other assessment factors are needed to increase the percentage of these beliefs such as gender, age, medical history, supporting examinations, and others.

Keywords *— Forward Chaining, Expert System, Haversein Formula*

1. INTRODUCTION

The COVID-19 pandemic has made drastic changes in people's lives, both in industry and others, to the psychological condition of the community. Many in the community are increasingly afraid and wary of going and wanting to get out of their homes.

While the health problem is one thing that can not be delayed, so it requires quick and appropriate treatment. Consultation with a doctor is one of the best solutions to be able to recognize the health problems experienced.

The Indonesian Ministry of Health has recommended for the public / patients who are lightly ill to postpone to the hospital to be able to break the chain of spread that is feared to occur in hospitals[1]. So that activities in the hospital are recommended to be restricted and make consultations to the doctor to be reduced.

Expert system is one of the solutions that can be offered to the community to be able to recognize the disease suffered by simply choosing and filling out what symptoms are felt and experienced without having to chat with a doctor.

Every disease must have symptoms felt by the sufferer, as well as diseases in the field of hematology and oncology, which is a disease that attacks blood cells and cancer in the human body.

In this study, researchers intend to see the effectiveness of forward chaining algorithms with the implementation of a recommendation application for disease diagnosis in the field of hematology and oncology that can help the public know what diseases they suffer through the symptoms that the community provides / input in the application.

2. RESEARCH METHOD

2.1. Expert System

Expert systems are again programs that advise automatically that are designed to mimic thought processes and know from experts where to place for victims of major problems[2]. Figure 1. the large components in the expert system. With a short sound[2] :

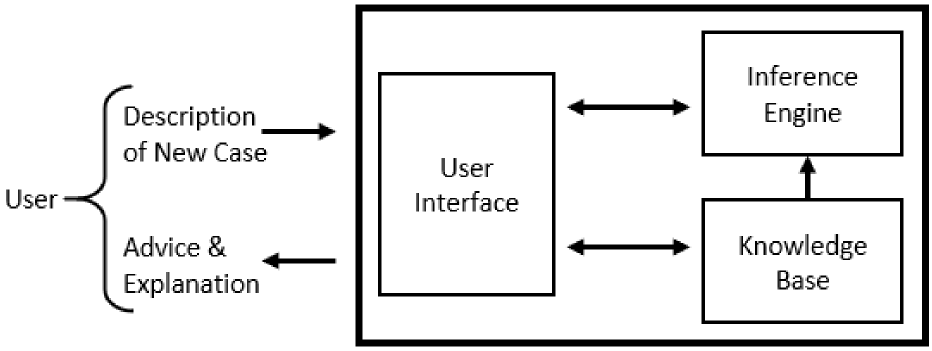


Figure 1. Expert System Components [2].

1) **Knowledge Base**, contains specific facts and knowledge, as well as the rules used for those facts.
2) **Working Memory**, contains facts of the problems to be searched during the session. Consists of 2 information, which is provided by the user and collected by the system.
3) **Inference Engine**, as a giver of permission to retrieve data from the knowledge base and to run the system, and provide conclusions from the facts provided by the user with the rules on the knowledge base.
4) **User Interface**, which is the view seen by the user.

2.2. Forward Chaining

Forward chaining is one of the two main methods of reasoning when using engine inference and can be logically described as a repeat application of ponens mode (a set of inference rules and valid arguments). The opponent of forward chaining is backward-chaining[3].

The basic concept of this forward chaining algorithm, where data is known in the form of facts will be tested using logical operators to give a new decision / conclusion. So onwards until the results are considered correct / completed.

The working system of this method is from the data / facts known / given, will be compared with every rule in the knowledge base that has been entered previously by both the administrator and predefine data.

2.3. Haversein Formula

The Haversein formula is an important equation in navigation, giving a large circular distance between 2 points on the surface of the ball (Earth) based on longitude and latitude [4].

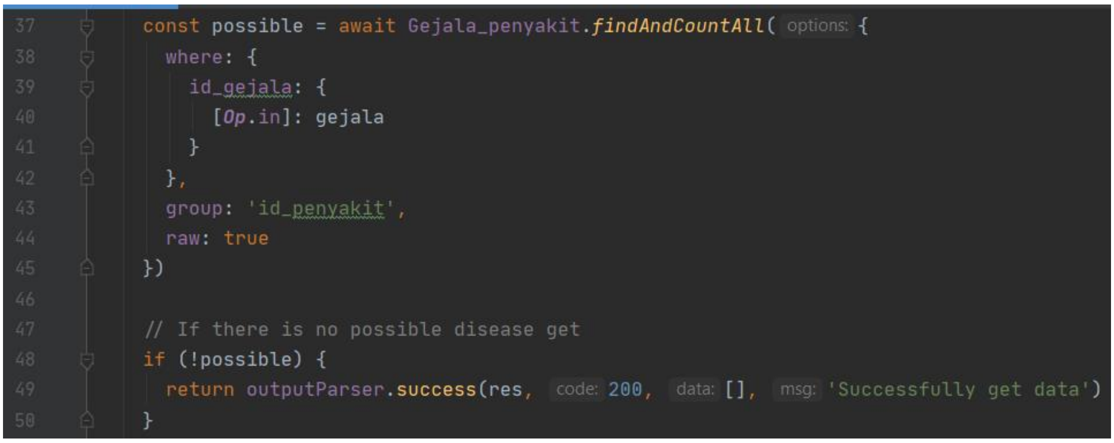
$$D = \operatorname{acos}(\, sin(lat1) * sin(lat2) + cos(lat1) * cos(lat2) * cos(long2 - long1)\,) * R$$

<u>Haversein Formula</u>

Description:
D : The calculation of haversein formula in the form of distance (km)
lat1 : latitude 1
long1 : longitude 1
lat2 : latitude 2
lat2 : latitude 2
R : radius of the earth 6371 (km)

3. RESULTS AND DISCUSSION

The forward chaining process in the system of disease diagnosis experts in the field of Hematology and Oncology is shown in the source code below:

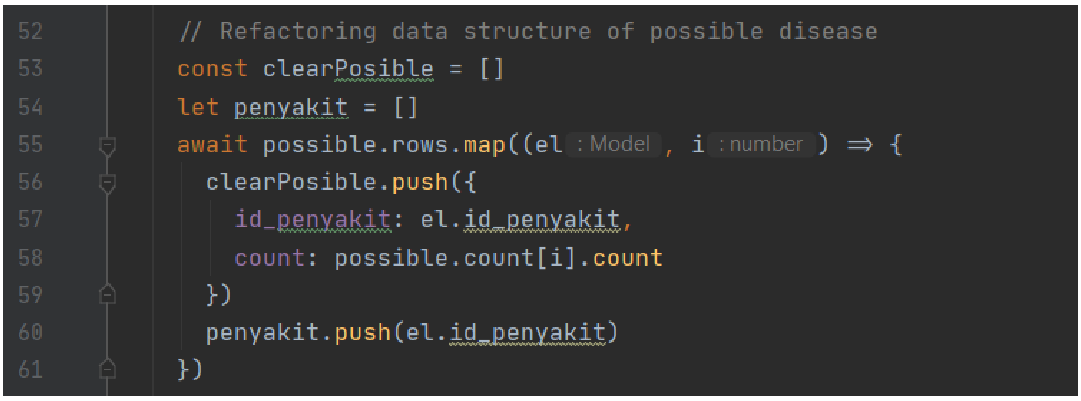


```
const possible = await Gejala_penyakit.findAndCountAll( options: {
  where: {
    id_gejala: {
      [Op.in]: gejala
    }
  },
  group: 'id_penyakit',
  raw: true
})

// If there is no possible disease get
if (!possible) {
  return outputParser.success(res,  code: 200,  data: [],  msg: 'Successfully get data')
}
```

Figure 2. Implementation of Source code forward chaining (1)

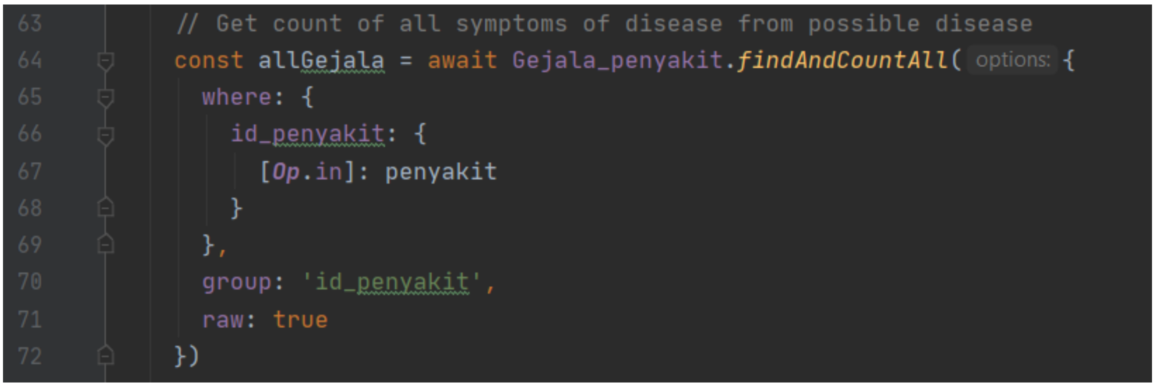
In Figure 2, the program looks for diseases that patients may experience according to the symptoms included by the user. Then if no corresponding disease is found then the program will immediately return a JSON containing an empty array which means no data is appropriate.



```
// Refactoring data structure of possible disease
const clearPosible = []
let penyakit = []
await possible.rows.map((el : Model , i : number ) ⇒ {
  clearPosible.push({
    id_penyakit: el.id_penyakit,
    count: possible.count[i].count
  })
  penyakit.push(el.id_penyakit)
})
```

Figure 3. Implementation of Source code forward chaining (2)

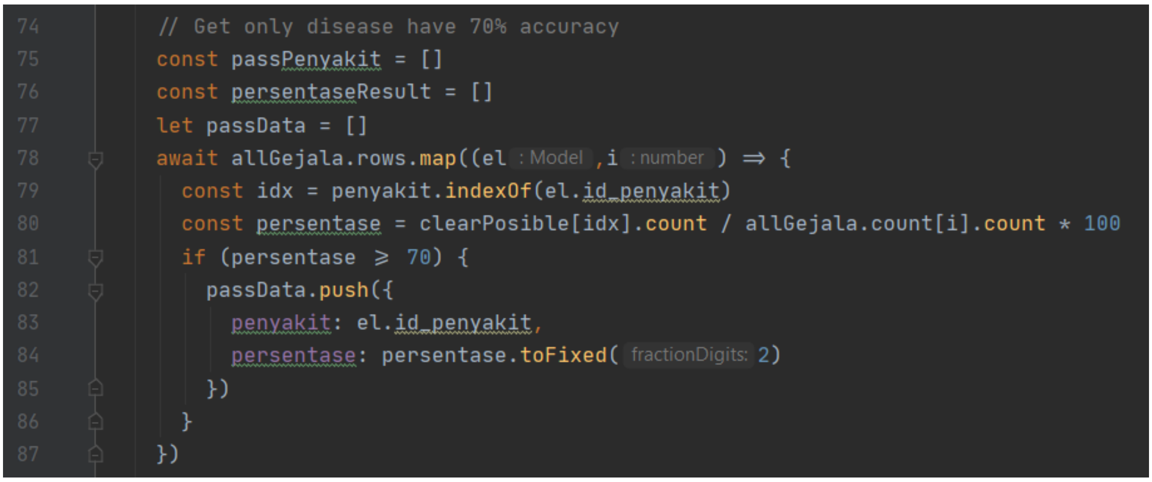
In Figure 3, the program takes only the disease id data and the appropriate number of diseases.



```
// Get count of all symptoms of disease from possible disease
const allGejala = await Gejala_penyakit.findAndCountAll( options: {
  where: {
    id_penyakit: {
      [Op.in]: penyakit
    }
  },
  group: 'id_penyakit',
  raw: true
})
```

Figure 4. Implementation of Source code forward chaining (3)

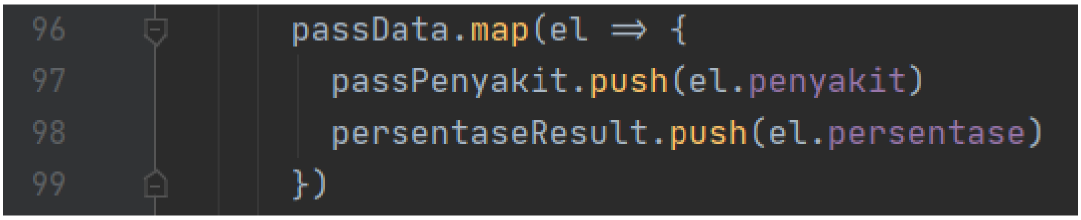
In Figure 4, the program retrieves data back to the database to retrieve the rule data that is all the symptoms that exist for a disease and the number of symptoms in the disease. Which will then be processed to calculate the percentage in the next image.



```
// Get only disease have 70% accuracy
const passPenyakit = []
const persentaseResult = []
let passData = []
await allGejala.rows.map((el :Model ,i :number ) => {
  const idx = penyakit.indexOf(el.id_penyakit)
  const persentase = clearPosible[idx].count / allGejala.count[i].count * 100
  if (persentase >= 70) {
    passData.push({
      penyakit: el.id_penyakit,
      persentase: persentase.toFixed( fractionDigits: 2)
    })
  }
})
```

Figure 5. Implementation of Source code forward chaining (4)

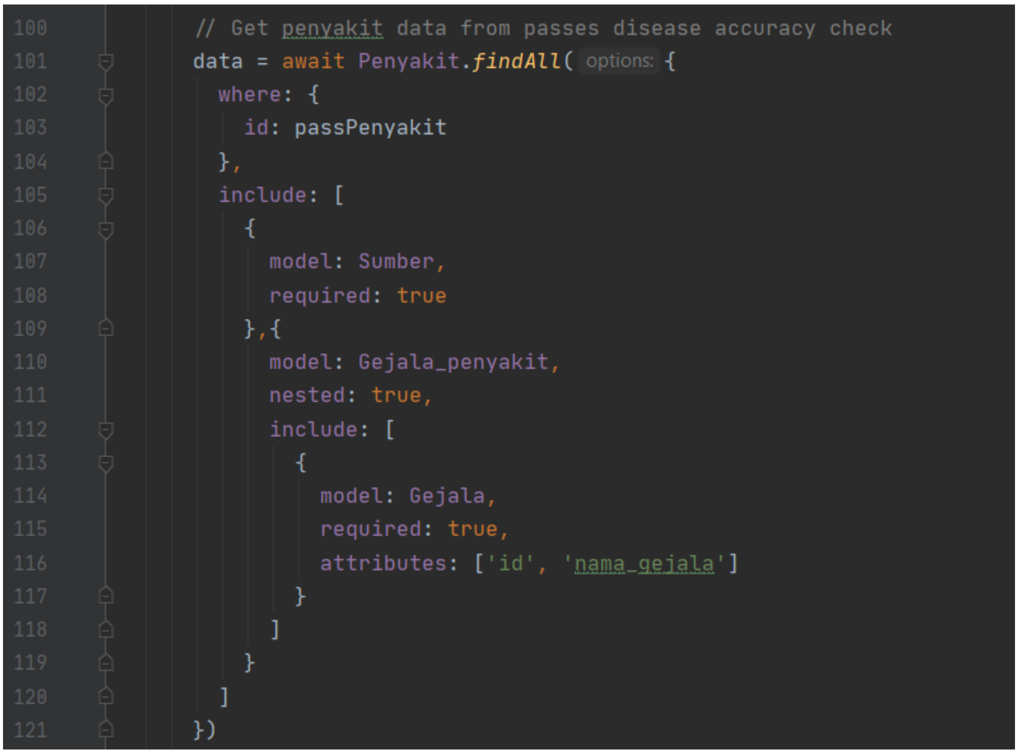
In Figure 5, the program performs the process of calculating the percentage - percentage of a disease in accordance with the predetermined rule, namely the number of symptoms in a disease compared to the number of symptoms that are included by the user. Then filter the data and take only data that has a percentage rate of more than 70% only.



```
passData.map(el => {
  passPenyakit.push(el.penyakit)
  persentaseResult.push(el.persentase)
})
```

Figure 6. Implementation of Source code forward chaining (5)

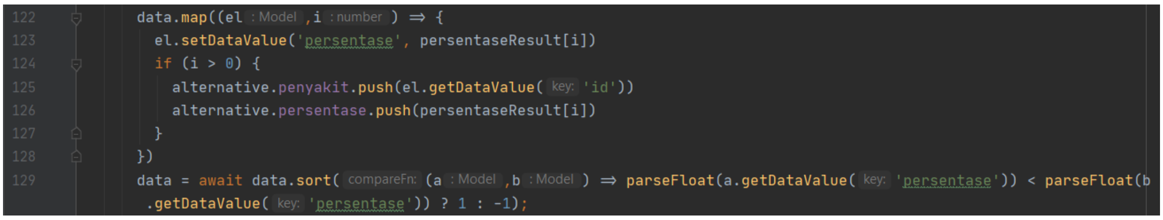
In Figure 6, the program mapping data by separating the disease id that has passed the data filter process into separate variables with the percentage of disease. This is used for the process of retrieving data to the database which will be described in the next image.



```
// Get penyakit data from passes disease accuracy check
data = await Penyakit.findAll( options: {
  where: {
    id: passPenyakit
  },
  include: [
    {
      model: Sumber,
      required: true
    },{
      model: Gejala_penyakit,
      nested: true,
      include: [
        {
          model: Gejala,
          required: true,
          attributes: ['id', 'nama_gejala']
        }
      ]
    }
  ]
})
```

Figure 7. Implementation of Source code forward chaining (6)

In Figure 7, the program again retrieves the data to the database for details of disease data that has gone through the data filter process. At this time the data was taken in as much detail as possible, ranging from the disease, the source, to all the symptoms of the disease..

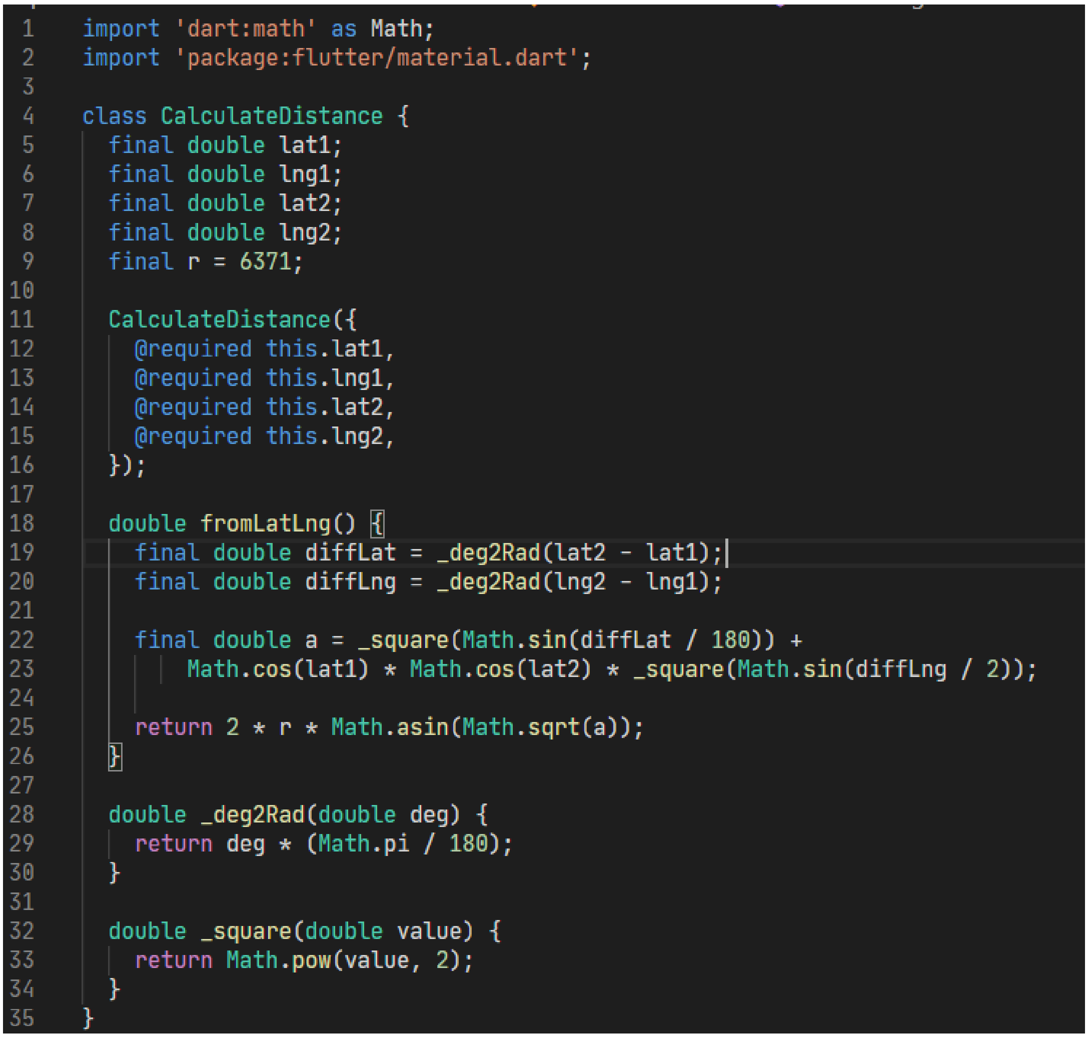


```
122  data.map((el :Model ,i :number ) => {
123    el.setDataValue('persentase', persentaseResult[i])
124    if (i > 0) {
125      alternative.penyakit.push(el.getDataValue( key: 'id'))
126      alternative.persentase.push(persentaseResult[i])
127    }
128  })
129  data = await data.sort( compareFn: (a :Model ,b :Model ) => parseFloat(a.getDataValue( key: 'persentase')) < parseFloat(b
      .getDataValue( key: 'persentase')) ? 1 : -1);
```

Figure 8. Implementation of Source code forward chaining (7)

In Figure 8, the program performs the process of entering percentage data that can be seen on line 123. And in line 124 - 127 done the process of entering disease id data and percentage to one variable in the form of arrays that aim to be entered into the database as data from user consultation. And on line 129, the process of sorting data so that the data with the largest percentage is in the first array of the return JSON data API.

The process of Haversein Formula in finding the nearest hospital in this system is shown in the source code below:

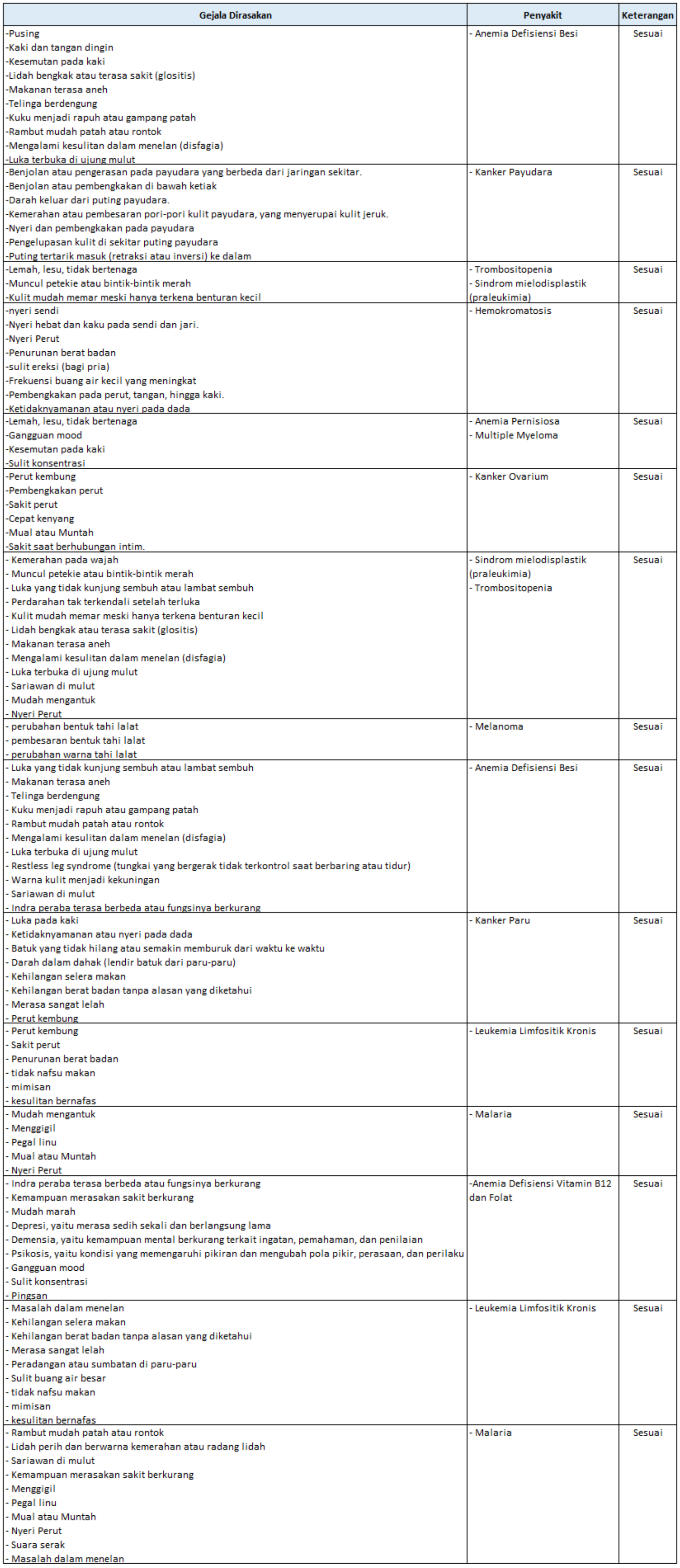


```
1   import 'dart:math' as Math;
2   import 'package:flutter/material.dart';
3
4   class CalculateDistance {
5     final double lat1;
6     final double lng1;
7     final double lat2;
8     final double lng2;
9     final r = 6371;
10
11    CalculateDistance({
12      @required this.lat1,
13      @required this.lng1,
14      @required this.lat2,
15      @required this.lng2,
16    });
17
18    double fromLatLng() {
19      final double diffLat = _deg2Rad(lat2 - lat1);
20      final double diffLng = _deg2Rad(lng2 - lng1);
21
22      final double a = _square(Math.sin(diffLat / 180)) +
23          Math.cos(lat1) * Math.cos(lat2) * _square(Math.sin(diffLng / 2));
24
25      return 2 * r * Math.asin(Math.sqrt(a));
26    }
27
28    double _deg2Rad(double deg) {
29      return deg * (Math.pi / 180);
30    }
31
32    double _square(double value) {
33      return Math.pow(value, 2);
34    }
35  }
```

Figure 9. Implementation of Source code haversein distance

In Figure 9, the program performs the distance calculation process using the Haversein Formula algorithm starting from line 5 – 9 is the initialization of global variables in the class thrown by the one calling this function. Then on lines 28 – 30 is to change latitude or longitude to radians. In lines 32 – 34 is to do the calculation of rank 2 on the value thrown. And on line 18 - 26 which is the core of this algorithm, that is, first look for the difference from the distance latitude and longitude user with the hospital and converted into radians after that is done calculation Haversein Formula which is then returned in the form of double value.

Table 1. System trial data results



Gejala Dirasakan	Penyakit	Keterangan
-Pusing -Kaki dan tangan dingin -Kesemutan pada kaki -Lidah bengkak atau terasa sakit (glositis) -Makanan terasa aneh -Telinga berdengung -Kuku menjadi rapuh atau gampang patah -Rambut mudah patah atau rontok -Mengalami kesulitan dalam menelan (disfagia) -Luka terbuka di ujung mulut	- Anemia Defisiensi Besi	Sesuai
-Benjolan atau pengerasan pada payudara yang berbeda dari jaringan sekitar. -Benjolan atau pembengkakan di bawah ketiak -Darah keluar dari puting payudara. -Kemerahan atau pembesaran pori-pori kulit payudara, yang menyerupai kulit jeruk. -Nyeri dan pembengkakan pada payudara -Pengelupasan kulit di sekitar puting payudara -Puting tertarik masuk (retraksi atau inversi) ke dalam	- Kanker Payudara	Sesuai
-Lemah, lesu, tidak bertenaga -Muncul petekie atau bintik-bintik merah -Kulit mudah memar meski hanya terkena benturan kecil	- Trombositopenia - Sindrom mielodisplastik (praleukimia)	Sesuai
-nyeri sendi -Nyeri hebat dan kaku pada sendi dan jari. -Nyeri Perut -Penurunan berat badan -sulit ereksi (bagi pria) -Frekuensi buang air kecil yang meningkat -Pembengkakan pada perut, tangan, hingga kaki. -Ketidaknyamanan atau nyeri pada dada	- Hemokromatosis	Sesuai
-Lemah, lesu, tidak bertenaga -Gangguan mood -Kesemutan pada kaki -Sulit konsentrasi	- Anemia Pernisiosa - Multiple Myeloma	Sesuai
-Perut kembung -Pembengkakan perut -Sakit perut -Cepat kenyang -Mual atau Muntah -Sakit saat berhubungan intim.	- Kanker Ovarium	Sesuai
- Kemerahan pada wajah - Muncul petekie atau bintik-bintik merah - Luka yang tidak kunjung sembuh atau lambat sembuh - Perdarahan tak terkendali setelah terluka - Kulit mudah memar meski hanya terkena benturan kecil - Lidah bengkak atau terasa sakit (glositis) - Makanan terasa aneh - Mengalami kesulitan dalam menelan (disfagia) - Luka terbuka di ujung mulut - Sariawan di mulut - Mudah mengantuk - Nyeri Perut	- Sindrom mielodisplastik (praleukimia) - Trombositopenia	Sesuai
- perubahan bentuk tahi lalat - pembesaran bentuk tahi lalat - perubahan warna tahi lalat	- Melanoma	Sesuai
- Luka yang tidak kunjung sembuh atau lambat sembuh - Makanan terasa aneh - Telinga berdengung - Kuku menjadi rapuh atau gampang patah - Rambut mudah patah atau rontok - Mengalami kesulitan dalam menelan (disfagia) - Luka terbuka di ujung mulut - Restless leg syndrome (tungkai yang bergerak tidak terkontrol saat berbaring atau tidur) - Warna kulit menjadi kekuningan - Sariawan di mulut - Indra peraba terasa berbeda atau fungsinya berkurang	- Anemia Defisiensi Besi	Sesuai
- Luka pada kaki - Ketidaknyamanan atau nyeri pada dada - Batuk yang tidak hilang atau semakin memburuk dari waktu ke waktu - Darah dalam dahak (lendir batuk dari paru-paru) - Kehilangan selera makan - Kehilangan berat badan tanpa alasan yang diketahui - Merasa sangat lelah - Perut kembung	- Kanker Paru	Sesuai
- Perut kembung - Sakit perut - Penurunan berat badan - tidak nafsu makan - mimisan - kesulitan bernafas	- Leukemia Limfositik Kronis	Sesuai
- Mudah mengantuk - Menggigil - Pegal linu - Mual atau Muntah - Nyeri Perut	- Malaria	Sesuai
- Indra peraba terasa berbeda atau fungsinya berkurang - Kemampuan merasakan sakit berkurang - Mudah marah - Depresi, yaitu merasa sedih sekali dan berlangsung lama - Demensia, yaitu kemampuan mental berkurang terkait ingatan, pemahaman, dan penilaian - Psikosis, yaitu kondisi yang memengaruhi pikiran dan mengubah pola pikir, perasaan, dan perilaku - Gangguan mood - Sulit konsentrasi - Pingsan	-Anemia Defisiensi Vitamin B12 dan Folat	Sesuai
- Masalah dalam menelan - Kehilangan selera makan - Kehilangan berat badan tanpa alasan yang diketahui - Merasa sangat lelah - Peradangan atau sumbatan di paru-paru - Sulit buang air besar - tidak nafsu makan - mimisan - kesulitan bernafas	- Leukemia Limfositik Kronis	Sesuai
- Rambut mudah patah atau rontok - Lidah perih dan berwarna kemerahan atau radang lidah - Sariawan di mulut - Kemampuan merasakan sakit berkurang - Menggigil - Pegal linu - Mual atau Muntah - Nyeri Perut - Suara serak - Masalah dalam menelan	- Malaria	Sesuai

From table 1, it can be seen that the accuracy of expert systems for diagnosing diseases in the field of Hematology and Oncology is 100% where out of 15 test data there are 15 data that correspond to expert test results.

But although the results of the expert system showed that 100% of the diagnosis results can not be simply caught raw that a person suffers from a disease diagnosed by the system, because according to the diagnosis expert only with symptoms can only convince experts by 50-60% only.

4. CONCLUSION

The conclusions of the authors through this study are as follows:

1. System Of Disease Diagnosis Experts In the Field of Hematology and Oncology using Android-based Forward Chaining algorithm that has been created has a diagnosis accuracy rate of 100% with details of 30 diseases and 136 symptoms.
2. This system will be optimal if one or a group of experts in this case the expert doctor has clearly defined every symptom of the disease.
3. Haversein formula algorithm can calculate the distance between 2 locations based on latitude and longitude but only to a straight line. So when compared to Google Maps the results of distance calculations are not the same (for long distances). Because Google Maps calculates the distance based on the path that humans / vehicles can travel.

5. SUGGESTED

The authors' suggestions on this study are as follows:

1. To be able to try methods from other expert systems such as certainty factor to see which method is best in providing diagonsa.
2. Other factors are needed in analyzing data such as age, gender, previous history of the disease, follow-up examination and so on. Because according to experts that diagnosis only using symptoms can only convince a doctor in making a diagnosis of 50-60%.
3. To find the nearest hospital can use other algorithms.

6. REFERENCES

[1.] Kompas.com, “Kemenkes Imbau RS Kurangi Praktik Rawat Jalan: Pasien Sakit Ringan Sebaiknya Tunda ke Rumah Sakit,” 2020. https://newsmaker.tribunnews.com/2020/04/17/kemenkes-imbau-rs-kurangi-praktik-rawat-jalan-pasien-sakit-ringan-sebaiknya-tunda-ke-rumah-sakit (accessed Sep. 30, 2020).

[2.] A. Al-Ajlan, “The Comparison between Forward and Backward Chaining,” Int. J. Mach. Learn. Comput., vol. 5, no. 2, pp. 106–113, 2015, doi: 10.7763/ijmlc.2015.v5.492.

[3.] I. Akil, “Analisa Efektifitas Metode Forward Chaining Dan Backward Chaining Pada Sistem Pakar,” J. Pilar Nusa Mandiri, vol. 13, no. 1, pp. 35–42, 2017.

[4.] F. Farid and Y. Yunus, “Analisa Algoritma Haversine Formula Untuk Pencarian Lokasi Terdekat Rumah Sakit Dan Puskesmas Provinsi Gorontalo,” Ilk. J. Ilm., vol. 9, no. 3, pp. 353–355, 2017, doi: 10.33096/ilkom.v9i3.178.353-355.