Design and Implementation of Multi Knowledge Base Expert System Using the SQL Inference Mechanism for Herbal Medicine

Melina[1], Eddie Krishna Putra[1], Wina Witanti[1], Sukrido[2], Valentina Adimurti Kusumaningtyas[2]

[1]Departement of Informatics, Universitas Jenderal Achmad Yani, Cimahi, Indonesia.
[2]Departement of Chemistry, Universitas Jenderal Achmad Yani, Cimahi, Indonesia.

Melina@lecture.unjani.ac.id

Abstract. The expert system is part of artificial intelligence with basic knowledge which is built to transfer its ability into computer system. In this study, the expert system is a combination of two technologies namely health technology as a basic of knowledge and information technology as interface of patient with health expert. The knowledge of health expert about diseases and herbal medicine is implemented in artificial intelligence to support in making a decision. This multi knowledge expert system is using the SQL Inference mechanism. The multi knowledge concept is made with database design, so that the expert system in this study is able to diagnosis many diseases and has more solutions in health problems. This study is using herbal medicine in order to prevent its users from poisoning and allergic of synthetic medicines.

1. Introduction

Artificial Intelligence (AI) nowadays generally describes the implementation of several aspects of human ability. The most important applied area of AI is the field of expert systems. An expert system (ES) is a system that uses human knowledge captured in a computer to solve problems that ordinarily require human expertise [1]. In order for computers to use heuristic knowledge effectively, it must be arranged in an easily accessible format distinguishing between data, knowledge, and control structures.The strength of an ES derives from its knowledge based an organized collection of facts and heuristics about the system's domain. An ES is built in a process known as knowledge engineering, during which knowledge about the domain is acquired from human experts and other sources by knowledge engineers.

The database and AI communities need to come to terms with each other and learn to work together without imposing unrealistic, requirements on each other [2] to improve their performance**.** The use of the SQL technology in the ES as AI tool is an actual and powerful possibility of elaborating inference mechanisms due to specific advantages [3]. In this study, the design of rule based ES using the Structured Query Language (SQL) database technology. This SQL technology is well known for database environments and was recently integrated in new powerful object oriented. Therefore, in our opinion this technology comes of interest for ES shells, becoming very attractive for ES developers [3]. The paper shows a solution that implementing rule engine using SQL has provided major benefits. One of several strategies can be employed by an inference engine to reach a conclusion. Inferencing engines for rule-based systems generally work by either forward or backward chaining of rules.

The multi knowledge concept is made with database design, so that the expert system in this study can accommodate multi knowledge bases about disease and has more solutions in health problems with solutions that affect consultative. To minimize the effects of synthetic drugs and poisoning of synthetic drugs, the drugs used are natural antibiotics and herbal medicines [4], as an alternative treatment that is safe, healthy and natural. The advantages of using herbal medicines are the preferred side effects such as modern medicines which are completely synthetic and chemical. Another plus is, herbal medicines can still be consumed by taking modern medicines, but must go through the approval and supervision of a doctor needed for a disease suffered by someone. To keep systems secure from irresponsible people, hence the expert consultation system has been designed with system security techniques with a password as the system's control and accountability [5]. Password generator is created automatically by a system that rainbow connected coloring theorems so as to minimize the amount of memory [6].

2. Method

2.1. Inference machanism

Inferencing engines for rule-based systems generally work by either forward or backward chaining of rules. Forward chaining is a data-driven strategy. The inferencing process moves from the facts of the case to a goal (conclusion). The strategy is thus driven by the facts available in the working memory and by the premises that can be satisfied. The inference engine attempts to match the condition (IF) part of each rule in the knowledge base with the facts currently available in the working memory. If several rules match, a conflict resolution procedure is invoked.

SQL, structured query language, is a programming language designed to manage data stored in relational databases. To retrieve data from a database, to create temporary tables and how to manipulate them effectively need query statements for better and faster performance. Example, running the above script in SQL workbench produces.

SQL statemen = “SELECT field A1, field B1, Field C 1, ([fact1 C2]/[fact1 B2])*100 as Percentage
FROM ((table A INNER JOIN Tabel B) INNER JOIN Tabel C)
WHERE field A1 = ‘value’ ORDER BY ’Field A1’”;
The following result set is shown below.

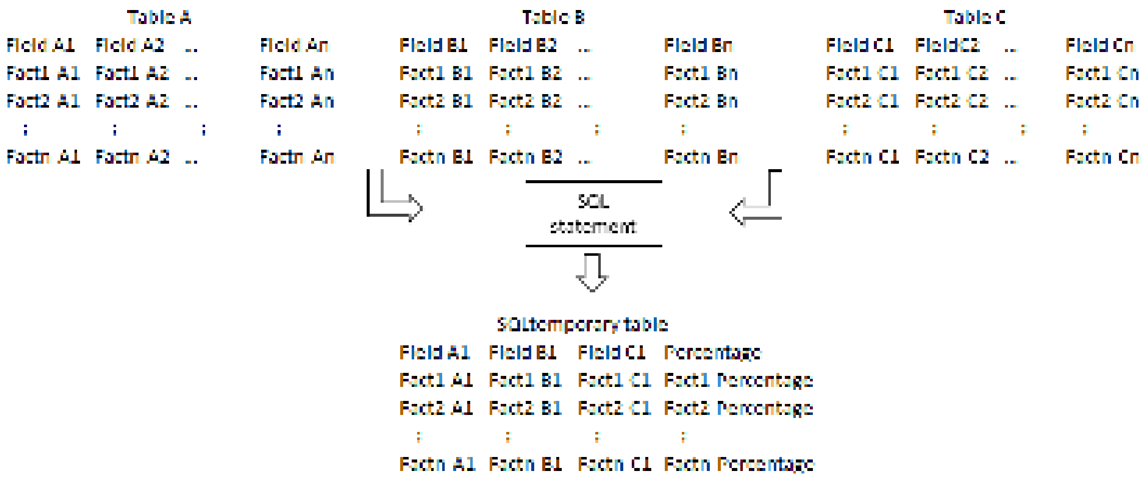

Figure 1. How SQL statements work

If tables contain knowledge, by using several query constructions (SQL statements), every knowledge piece (fact, rule) can be easily found and extracted, especially if knowledge pieces are unique (making them suitable for table index fields). Therefore, the contents searching process, one of the key problems to set up an ES inference mechanism can be solved [3]. The inference mechanism shown in figure 2 can be designed according to the above considerations.

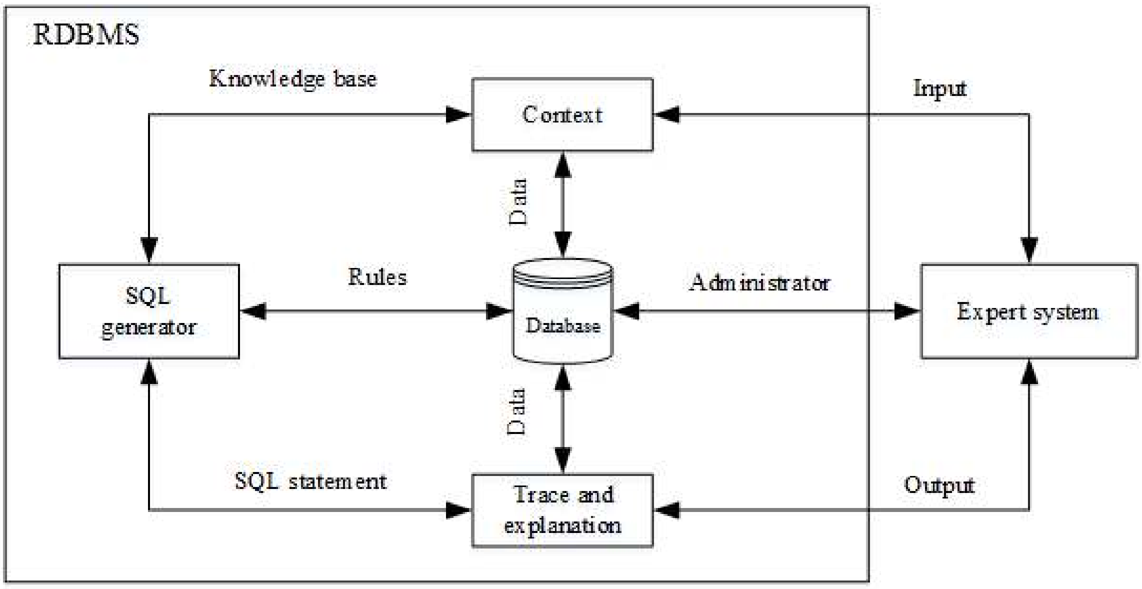

Figure 2. The designed SQL based inference mechanism

for forward chaining inference strategy

2.2. Database design

Database design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. In this paper, we will explain the main phases that create database design.

2.2.1. Collecting and analyzing data

The first step is collecting data with uses interview, literature and questioning method by asking directly to the source of data, experts and the person in charge. After these steps, the data is ready for analysis. We used data analysis methods are Data Flow Diagram (DFD). 0-DFD level for the multi-knowledge base expert system is shown below.

Figure 3. 0-DFD level the multi knowledge base expert system

0-DFD level It is also known as context diagram. It's designed to be an abstraction view, showing the expert consultation system as a single process with its relationship to two external entities, the expert as source of knowledge base and user to query (question) and to receive advice from expert. Process in a level 0 context diagram can be decomposed (exploded) into a level 1 to represent details of the processing activities. A possible level 1 DFD for the expert consultation system is as follows.

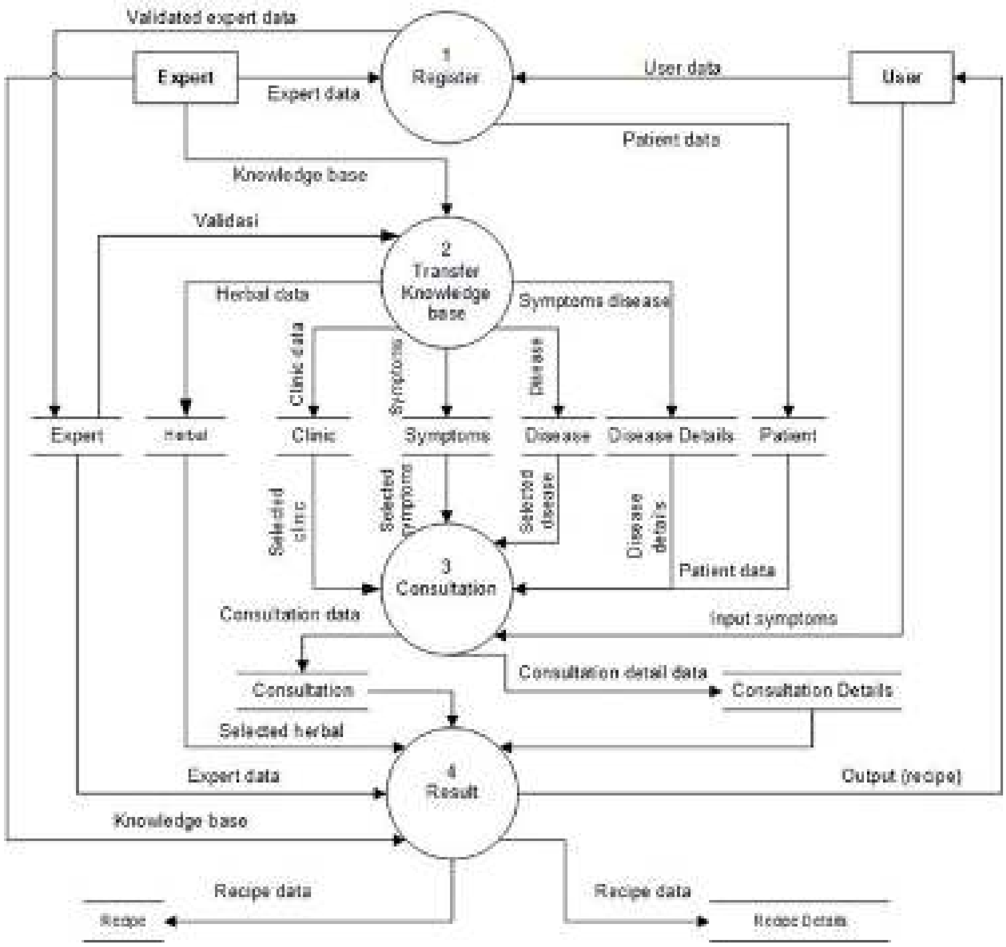

Figure 4. 1-DFD level the multi knowledge base expert system

The figure below shows the level 1 DFD contains four processes, two external entities and eleven data stores. Based on the diagram, for the first time, an expert must register on the system to get validation as an expert then transferring expertise from an expert to the to system. User that need to register and create their account to log into system for consultations and get medical advice. Process in a level 1 DFD can be decomposed (exploded) into a level 2 to represent details of the processing activities. A possible level 2 DFD for the expert consultation system is as follows.

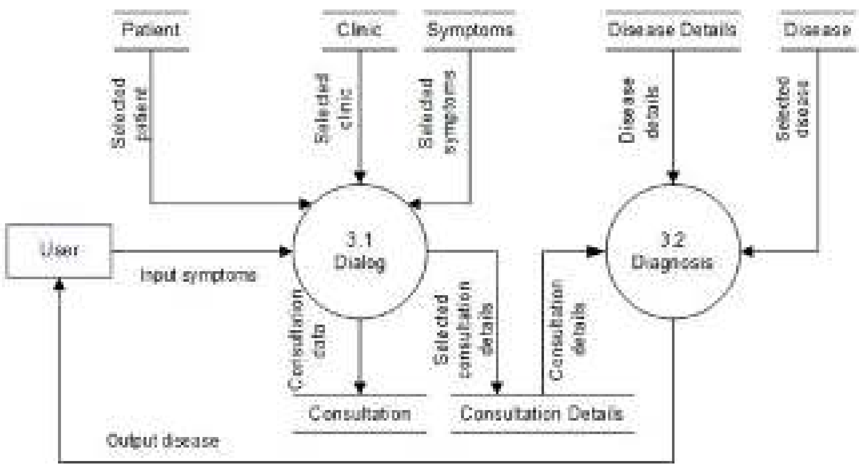

Figure 5. 2-DFD level the multi knowledge base expert system

A level 2 data flow diagram (DFD) offers a more detailed look at the consultation processes that make up an information system than a level 1 DFD does. The level 2 DFD contains two processes, one external entities and seven data stores. First, a patient chooses the clinic and the symptoms they have, the symptoms chosen by the patient will be processed in the diagnosis process. The output of this level is information about the patient's disease.

2.2.2. Conceptual design

When every data requirement is stored and analyzed, then is creating a conceptual database plan. The result of this phase is an Entity-Relationship (ER) diagram, as shown in the figure below.

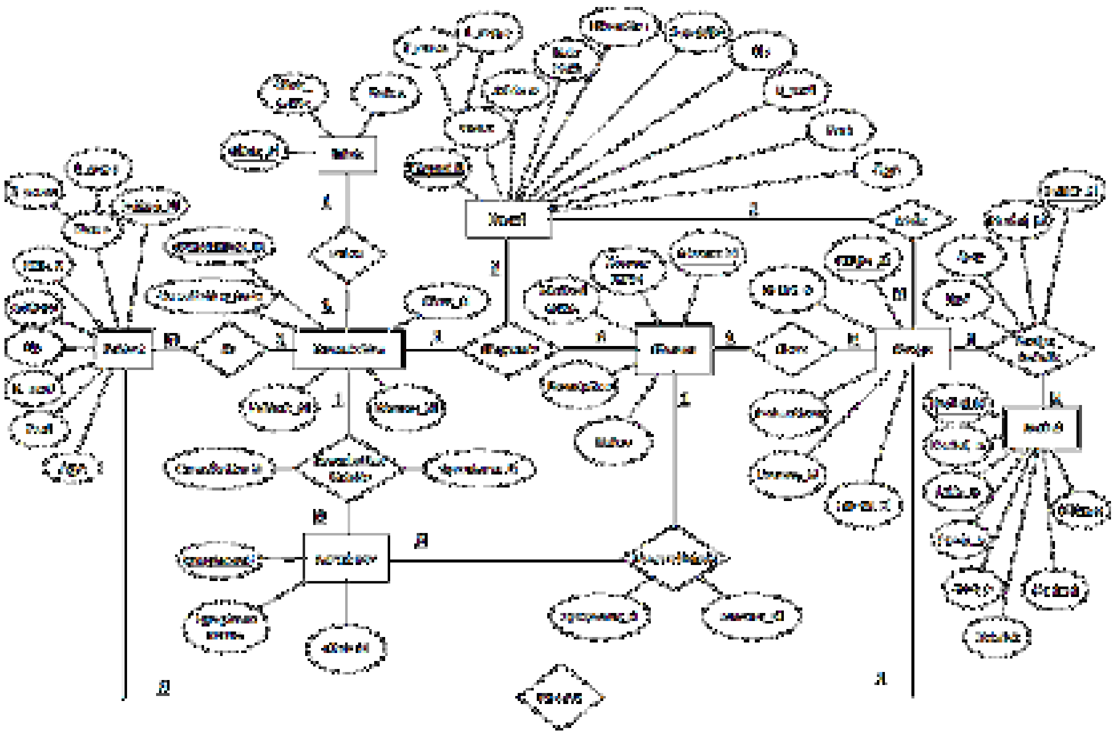

Figure 6. Entity-relationship the multi knowledge base expert system

2.2.3. Logical design

This phase provides the results of the relationship scheme. The basis of this scheme is the ER Diagram. In this study, the rules for transferring the ER model to the relationship scheme are as follows.

- Expert (<u>Expert_id</u>, F_Name, L_Name, Address, Date_birth, Education, Specialist, HP, E_mail, Pwd)
- Patient (<u>Patient_id</u>, F_name, L_name, date_b, Address, hp, E_mail, Pwd)
- Disease (<u>disease_id</u>, Disease_name, medical_name, descripsi. Notes)
- Disease_Details ((Disease_Id, Symptoms_Id)
- Symptoms (<u>Symptoms_Id</u>, Symptoms_Name, Clinic_id)
- Clinic (<u>Clinic_id</u>, Clinic_name, Notes)
- Consultation (<u>Consultation_Id</u>, Consultation_Date, Patient_Id, Clinic_id, Disease_Id)
- Consultation_Details (Consultation_Id, Symptoms_Id)
- Recipe (<u>Recipe_id</u>, recipe_n, Instructions, Disease_id, expert_id)
- Recipe_details (Recipe_id, Herbal_Id, dose, Unit_id)
- Herbal (<u>Herbal_Id</u>, herbal_n, latin_n, foreig_n, area_n, habitat, content, efficacy)
- Unit (<u>Unit_id</u>, Unit, Notes)

2.2.4. Physical Design

Physical database design translates the logical data model into a set of SQL statements that define the database. For relational database systems, it is relatively easy to translate from a logical data model into a physical database. Rules for translation: entities become tables in the physical database. DDL statemens for creating a database and structure in language of the chosen DBMS, that is MySQL, as follows.

- CREATE DATABASE MultyExpertS;
- CREATE TABLE Expert (Expert_id varchar(10) NOT NULL, f_name varchar(30), l_name varchar(30) NOT NULL, address text, date_birth date NOT NULL, Education varchar(30) NOT NULL, Specialist varchar(30) NOT NULL, hp varchar (20) NOT NULL, email varchar (50) NOT NULL, pwd varchar(30) NOT NULL, PRIMARY KEY(Expert_id));
- CREATE TABLE patient (patient_id varchar(10) NOT NULL, f_name varchar(30), l_name varchar(30) NOT NULL, date_b date NOT NULL, address text, hp varchar (20), email varchar (50) NOT NULL, pwd varchar(30) NOT NULL, PRIMARY KEY(patient_id));
- CREATE TABLE disease (disease_id varchar(10) NOT NULL, disease_name varchar(30) NOT NULL, medical_name varchar(30) NOT NULL, descripsi text, notes text, PRIMARY KEY(disease_id)) ENGINE=InnoDB;
- CREATE TABLE clinic (clinic_id varchar(10) NOT NULL, clinic_name varchar(50) NOT NULL, notes varchar(200), PRIMARY KEY(clinic_id)) ENGINE=InnoDB;
- CREATE TABLE symptoms (symptoms_id varchar(10) NOT NULL, symptoms_name varchar(50) NOT NULL, clinic_id varchar(10) NOT NULL, PRIMARY KEY(symptoms_id), FOREIGN KEY(clinic_id) REFERENCES clinic (clinic_id)) ENGINE=InnoDB;
- CREATE TABLE herbal (herbal_id varchar(10) NOT NULL, herbal_n varchar(30) NOT NULL, latin_n varchar(30) NOT NULL, foreig_n varchar(30), area_n varchar(30), habitat varchar(30), content varchar (200) NOT NULL, efficacy varchar (250) NOT NULL, PRIMARY KEY(herbal_id)) ENGINE=InnoDB;
- CREATE TABLE Disease_Details (disease_id varchar(10) NOT NULL, symptoms_id varchar(10) NOT NULL, FOREIGN KEY(disease_id) REFERENCES disease (disease_id), FOREIGN KEY(symptoms_id) REFERENCES symptoms (symptoms_id));
- CREATE TABLE consultation (Consultation_id varchar(10) NOT NULL, consultation_date date NOT NULL, patient_Id varchar(10) NOT NULL, clinic_id varchar(10) NOT NULL, disease_id varchar(10) NOT NULL, PRIMARY KEY(Consultation_id), FOREIGN KEY(patient_Id) REFERENCES patient (patient_Id), FOREIGN KEY(clinic_id) REFERENCES clinic (clinic_id), FOREIGN KEY(disease_id) REFERENCES disease(disease_id)) ENGINE=InnoDB;
- CREATE TABLE consultation_details (consultation_id varchar(10) NOT NULL, symptoms_id varchar(10) NOT NULL, FOREIGN KEY(consultation_id) REFERENCES consultation (consultation_id), FOREIGN KEY(symptoms_id) REFERENCES symptoms (symptoms_id)) ENGINE=InnoDB;
- CREATE TABLE unit (Unit_id varchar(10) NOT NULL, unit varchar(50) NOT NULL, notes varchar(200), PRIMARY KEY(unit_id)) ENGINE=InnoDB;
- CREATE TABLE recipe (Recipe_id varchar(10) NOT NULL, recipe_n varchar(30) NOT NULL, Instructions text NOT NULL, disease_id varchar(10) NOT NULL, expert_id varchar(10) NOT NULL, PRIMARY KEY(recipe_id), FOREIGN KEY(disease_id) REFERENCES disease (disease_id), FOREIGN KEY(expert_id) REFERENCES expert (expert_id)) ENGINE=InnoDB;
- CREATE TABLE recipe_details(recipe_id varchar(10) NOT NULL, herbal_id varchar(10) NOT NULL, dose varchar(30) NOT NULL, unit_id varchar(10), FOREIGN KEY(recipe_id) REFERENCES recipe(Recipe_id), FOREIGN KEY(herbal_id) REFERENCES herbal(herbal_id), FOREIGN KEY(unit_id) REFERENCES unit(unit_id));

This phase shown figure 7 relationship between tables that have been normalized as belows.

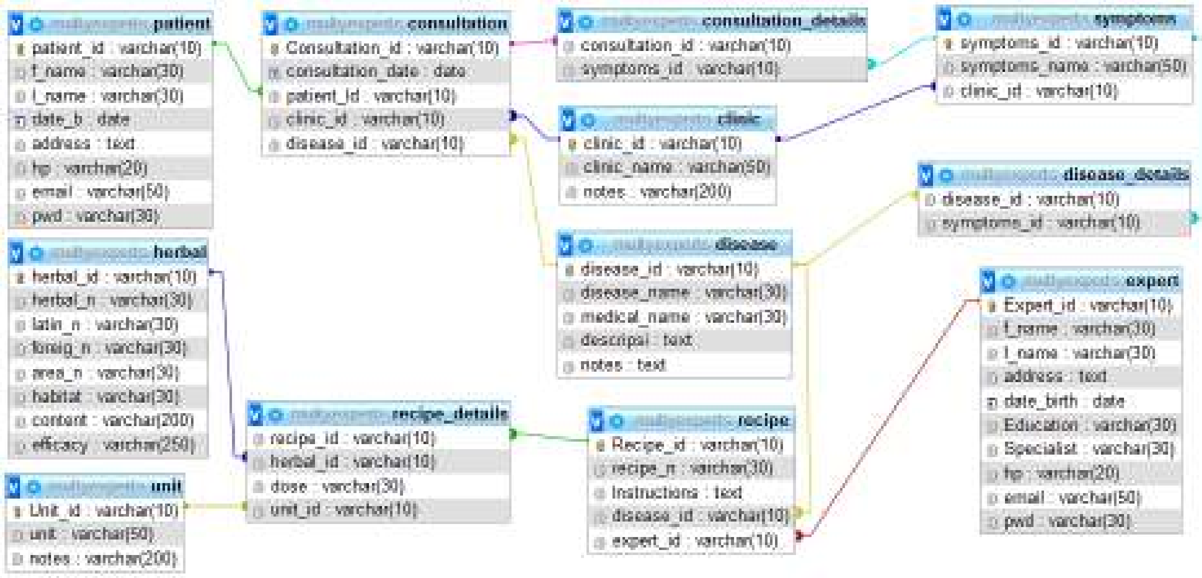

Figure 7. Relationship between tables that have been normalized

2.2.5. Knowledge base design using the SQL inference mechanism for forward chaining

In this study, we designing a knowledge base using the SQL inference mechanism for forward chaining**.** Forward chaining inference mechanisms start deriving knowledge from some initial premises (true facts). The inference process stops either when all possible applies rules are fired or a certain goal was reached. One example of the results of the representation of knowledge about diagnosing one disease based on symptoms, as follow.

If	:	Cough	And	:	Chronic cough
And	:	Bloody sputum	And	:	Blood in stool
And	:	Shortness of breath	And	:	Night sweats
And	:	Weight loss	And	:	Fatigue
And	:	Chest pain	or	:	Pain while breathing

Then : Tuberculosis 100 % recipe tuberculosis remedies by expert Popong Malia,
Asthma 60 % recipe Asthma remides by expert Abdul Mukthy,
Chronic bronchitis 40 % recipe chronic bronchitis remides by expert Tina Alfitria,
Flu 28,60 % recipe flu remides by expert Toto Sukarno.

We will show how to create a new table by using SQL statement to get the results of the diagnostic process, and new table shown in table 1, table 2 and table 3, respectively.

- Query table of disease with total symptoms.
  SELECT Disease_Details.Disease_Id, Count(Disease_Details.Disease_Id) AS TotalSymptoms FROM Disease_Details GROUP BY Disease_Details.Disease_Id;
- Query table of diseases with selected symptoms.
  SELECT DiseaseSelected.Disease_Id, Count(DiseaseSelected.Disease_Id) AS Selected FROM DiseaseSelected GROUP BY DiseaseSelected.Disease_Id;
- Query table of result.
  SELECT Disease.Disease_id, Disease.Disease_name, DiseaseSymptoms.TotalSymptoms, SymptomsSelected.Selected, ([Selected]/[TotalSymptoms])*100 AS Percentage FROM (Disease INNER JOIN DiseaseSymptoms ON Disease.disease_id = DiseaseSymptoms.Disease_Id) INNER JOIN SymptomsSelected ON Disease.Disease_id = SymptomsSelected.Disease_Id ORDER BY ([Selected]/[TotalSymptoms])*100 DESC";

Table 1. Disease with total symptoms

Disease_Id	Total symptoms
D-0001	10
D-0002	5
D-0003	5
D-0005	7
D-0013	10

Table 2. Symptoms selected

Disease_Id	Selected
D-0001	2
D-0002	3
D-0003	2
D-0005	2
D-0013	10

Table 3. Diagnosis result

Disease_id	Disease_name	TotalSymptoms	Selected	Percentage
D-0013	Tuberculosis	10	10	100,00
D-0002	Asthma	5	3	60,00
D-0003	Chronic bronchitis	5	2	40,00
D-0005	Flu	7	2	28,60
D-0001	Allergy	10	2	20,00

2.2.6. Implementation

This phase is input data into the new database, it concerned with the identification of errors in the newly implemented system and also checks the database against requirement specifications. Consultation process and input data using forms consultation disease and form recipe, see figure 8-9, respectively.

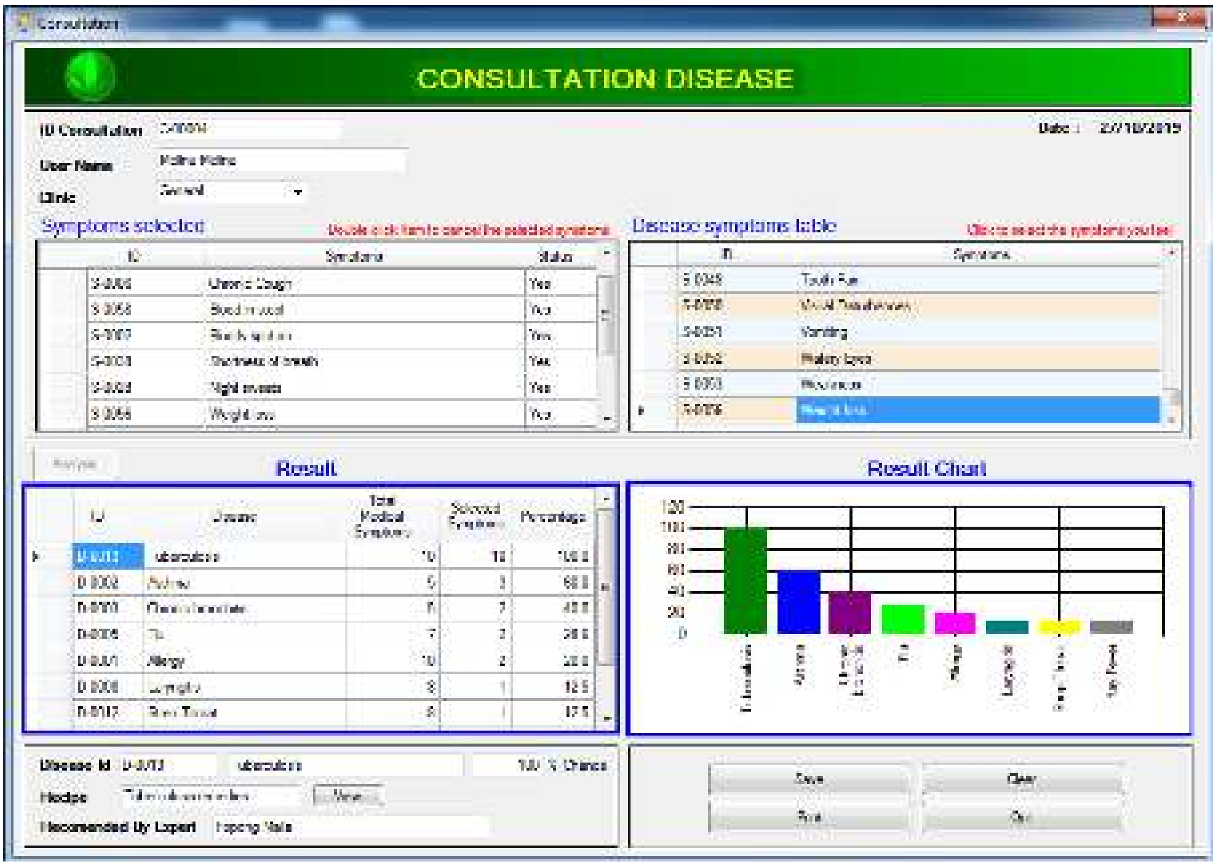

Figure 8. Consultation disease form

Figure 9. Result recipe form

3. Results

The nouvel of expert system from multi knowledge base for many experts using SQL inference mechanism has been successfully implemented in this study so that the expert system is integrated, fast and accurate.

4. Conclusion

From the results of this study, some conclusions which are as follows.

1. The expert system in this study was designed by using the basic concept of multi knowledge in order to be able to store a lot of knowledge base about diseases from many experts.
2. In this study, the expert system uses the SQL inference mechanism, so that the expert system is dynamic, updated, accurately and accesed more quickly.

References

[1] Turban, Effraim 1995 *Decision support and expert system* (London: Prentice-Hall Inc)
[2] O. Friesen and F. Golshani 1989 Database in large AI*, AI Magazine* Volume **10** Number 4
[3] George Dobre, Cosmin Grigorescu, Dorin Carstoiu and Simona Grigoresu, 1998 Applying SQL techniques on inference mechanisme of expert system*, transaction on information and communication technologies* vol **20** ISSN 1743-3517
[4] VA Kusumaningtyas, YM Syah dan DM Agustini 2008 *Proc. Int. Conf. on chemistry* (Bandung) p 229-230
[5] Melina, EK Putra, VA Kusumanintyas and W Witanti 2018 *Proc. seminar nasional sistem informasi Indonesia* (Surabaya) p 231-236
[6] Melina, A.N.M. Salman 2018 *Proc. Int. Conf. on basic science* (Malang) p 273-276
[7] Gavin Powel 2006 *Beginning database design* (Indiana: Wiley Publishing Inc)
[8] Beynon, P and Davies 1991 *Expert database system a gentle introduction* (England: McGraw Hill Inc)
[9] Barber, Richard 1992 Bones an experts system for diagnosis with fault models (England: Ellis Horwood)