

Perancangan Modul Pendukung Keputusan Klinis Dalam Peresepan

Yosep Parlindungan Naibaho¹, Lukas Chrisantyo², Yuan Lukito³

Program Studi Informatika, Universitas Kristen Duta Wacana,

Jl. Dr. Wahidin Sudirohusodo 5-25, Yogyakarta

¹yosep.naibaho@ti.ukdw.ac.id

²lukaschris@staff.ukdw.ac.id

³yuanlukito@ti.ukdw.ac.id

Abstract— *The manual prescription process has a potential of error, which can lead to undesirable consequences, particularly for patients. As the advancement of technology, the use of decision support systems is necessary to minimize errors in prescription delivery. In this study, a decision support system module will be developed to assist prescription process by transferring knowledge possessed by experts as the knowledge base of the system. This module will use rule-based method as the decision-making method by applying validated rules based on the acquired knowledge base. To test the accuracy of the module's output, test case design will be used as the method. The result of the research showed that all tested scenarios have the same results as the expected results, indicating that the use of rule-based expert system is effective in developing this decision support module. However, the module can still be improved by adding more data as the knowledge base or integrating the module directly with existing system in clinic or hospital.*

Intisari— Pemberian resep secara manual dalam bidang kesehatan memiliki potensi terjadinya kesalahan yang dapat menimbulkan beberapa dampak yang tidak diinginkan khususnya bagi pasien. Seiring perkembangan teknologi, penggunaan sistem pendukung keputusan diperlukan untuk meminimalisir kesalahan saat pemberian resep. Pada penelitian ini, sebuah modul pendukung keputusan dibuat untuk membantu peresepan dengan memindahkan pengetahuan yang dimiliki oleh pakar sebagai *knowledge base* dari sistem dan kemudian menggunakan metode *rule based* sebagai metode pengambilan keputusan oleh modul dengan menerapkan aturan-aturan yang sudah divalidasi oleh pakar berdasarkan *knowledge base* yang didapatkan. Untuk menguji akurasi dari *output* yang dihasilkan modul, dilakukan pengujian menggunakan *test case design*. Hasil dari penelitian yang dilakukan yaitu hasil pengujian memperlihatkan bahwa seluruh skenario yang diuji memberikan hasil yang sesuai dengan ekspektasi sehingga dapat disimpulkan bahwa penggunaan *rule based expert system* efektif dalam membangun modul pendukung keputusan kali ini. Modul ini masih dapat dikembangkan dengan menambahkan data untuk membangun *knowledge base* atau mengintegrasikan modul dengan sistem yang ada pada klinik atau rumah sakit yang ada secara langsung.

Kata Kunci— *Rule based, Expert system, Pendukung keputusan, test case design*

I. PENDAHULUAN

Pemilihan obat atau resep yang sesuai dengan diagnosa penyakit pasien sangat perlu dilakukan di dalam dunia

pengobatan. Pemberian resep yang tidak sesuai dapat menjadi pengaruh yang tidak baik untuk pasien baik secara kesehatan maupun finansial pasien itu sendiri. Salah satu faktor yang paling sering menyebabkan kesalahan dalam peresepan adalah karena penulisan resep dokter yang masih manual dan terkadang sulit untuk dibaca. Seiring dengan perkembangan teknologi saat ini, belum ada sebuah sistem yang dapat menilai kecocokan antara obat dan gejala dimana diharapkan teknologi bisa membantu dalam melakukan peresepan agar dapat mengurangi kesalahan dalam pemberian resep obat sehingga dapat meminimalisir kerugian yang diterima pasien khususnya dalam hal pengobatan simtomatik. Pengobatan simtomatik sendiri merupakan pengobatan yang bertujuan untuk meredakan keluhan atau gejala yang timbul tanpa melihat penyakit utama yang menyebabkan keluhan atau gejala tersebut timbul. Terdapat beberapa kasus dimana pemberian resep yang tidak sesuai menyebabkan pasien mengalami gangguan Kesehatan lainnya karena resep yang diberikan tidak sesuai dengan keadaan atau gejala yang dialami pasien [1].

International Classification for Diseases (ICD) merupakan salah satu standar pengkodean internasional yang digunakan untuk rekam medis elektronik di Indonesia. Di dalam standar pengkodean tersebut telah diklasifikasikan gejala-gejala penyakit yang sering dialami oleh pasien. Data Obat di Indonesia (DOI) merupakan keterangan mengenai obat-obat yang beredar di Indonesia saat ini. Dengan adanya standar klasifikasi yang digunakan di Indonesia ini, maka permasalahan terkait peresepan dapat dibuat menjadi lebih sederhana.

Electronic Prescribing (eRx) merupakan suatu sistem yang memungkinkan seorang dokter untuk melakukan peresepan secara elektronik dan bahkan dokter juga dapat memasukkan informasi yang lebih spesifik mengenai informasi pasien seperti alergi terhadap obat [2].

Expert System merupakan suatu pendekatan yang berasal dari pengetahuan ahli dalam bidang tertentu untuk membantu mengambil keputusan yang cukup kompleks [3]. *Rule Based* merupakan salah satu metode dari *Expert System* yang bertujuan untuk mencapai tujuan tertentu dalam sebuah data dan memiliki bentuk aturan IF-THEN. Metode ini

memiliki pendekatan dalam pengambilan keputusan dengan mengandalkan sekumpulan aturan yang sudah ditentukan [4]. Dalam metode *rule based* aturan ditentukan melalui pengetahuan dari pakar dalam bidang tertentu yang kemudian direpresentasikan ke dalam bentuk rule yang terdiri atas *premises* (kondisi) dan *conclusions* (tindakan), dimana kondisi menggambarkan parameter atau informasi yang relevan untuk mengidentifikasi suatu permasalahan dan Tindakan menggambarkan solusi dari permasalahan yang ada.

Englebert [5] mengimplementasikan peresepan secara elektronik di Amerika yang memiliki hasil bahwa tingkat *error* terhadap peresepan berhasil berkurang, tingkat efisiensi untuk meresepkan menjadi lebih tinggi, dan mengurangi biaya perawatan Kesehatan. Sehingga peneliti menyimpulkan bahwa implementasi dari peresepan secara elektronik memiliki banyak sekali keuntungan akan tetapi masih banyak rintangan dan kesulitan untuk diimplementasi sehingga diharapkan bisa lebih berkembang di kemudian hari. Dengan demikian, penulis ingin melakukan hal yang serupa sehingga diharapkan tingkat *error* terhadap peresepan bisa berkurang.

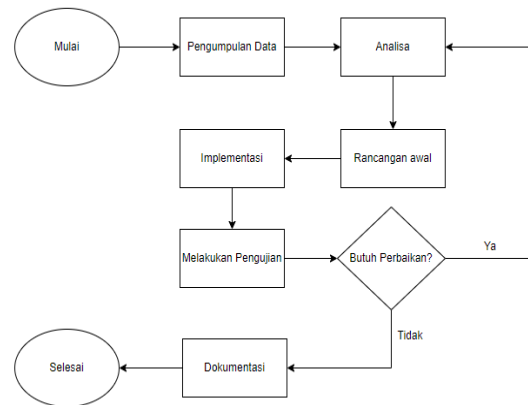
Marfuah [6] merancang sistem pendukung keputusan dengan menggunakan metode *Rule Based Expert System* dalam perancangan sistemnya. Pada penelitiannya, *Rule Based Expert System* dinilai dapat digunakan dan mampu menjadi jawaban untuk penyelesaian masalah pada perancangan sistem pendukung keputusan, sehingga penulis menggunakan metode *Rule Based Expert System* juga yang akan diterapkan pada penyakit yang umum terjadi kepada anak dan diharap dapat mendapatkan hasil dan akurasi yang sama seperti penelitian yang sudah dilakukan pada penelitian tersebut.

Aprilia [7] melakukan penelitian untuk diagnosa penyakit yang terjadi pada anak. Disini penulis menggunakan data-data penyakit yang digunakan oleh peneliti sebagai salah satu acuan pengumpulan dan pengklasifikasian data penyakit untuk kemudian digunakan sebagai data untuk menguji modul yang akan dibuat.

Berdasarkan permasalahan tersebut, peneliti ingin merancang sebuah modul *Electronic Prescribing* menggunakan metode *Rule Based* yang dapat membantu pengguna (petugas kesehatan) untuk mendukung keputusan dalam melakukan peresepan. Penelitian ini dikhususkan untuk pengobatan simptomatik dengan mencocokkan resep ataupun yang akan diberikan berdasarkan gejala yang diderita oleh pasien yang telah ter klasifikasi, maupun melakukan pengecekan obat yang diberikan berdasarkan formularium nasional. Pada modul yang akan dibuat, modul akan memberikan peringatan-peringatan (*warning system*) untuk mengkonfirmasi peresepan yang dilakukan oleh petugas Kesehatan dan status BPJS dari pasien.

II. METODOLOGI PENELITIAN

Alur dari pengembangan modul dapat dilihat melalui Gambar 1. Rancangan modul ini dimulai dengan pengumpulan data terlebih dahulu contohnya seperti data obat dan gejala yang akan digunakan pada modul. Setelah mendapat data yang diperlukan, dilakukan analisa untuk melihat permasalahan serta merumuskan solusi untuk memecahkan permasalahan tersebut. Rancangan awal dilakukan sebagai pedoman pengerjaan modul nantinya seperti rancangan UI maupun rancangan database, hingga rancangan pengujian modul. Selanjutnya, implementasi dilakukan berdasarkan rancangan awal dari modul yang sudah dibuat sebelumnya. Modul yang sudah diimplementasikan diuji berdasarkan metode yang sudah ditentukan untuk melihat apakah modul yang diimplementasi sudah sesuai dengan yang direncanakan ataupun diinginkan. Jika belum sesuai, maka dilakukan analisa kembali pada modul untuk kemudian dilakukan perbaikan sampai dengan pengujian hingga sesuai. Setelah modul dirasa sesuai, dilakukan dokumentasi ke dalam bentuk laporan.



Gambar 1. Diagram Alir Rancangan Pengembangan Modul

A. Perancangan penelitian

- 1) *Data collection (pengumpulan data)*
 - a) Wawancara

Melakukan wawancara kepada dokter (pakar) yang bekerja di salah satu klinik dalam hal ini yaitu kepada Dr. Daniel C.A Nugroho, MPH, untuk mendapatkan basis pengetahuan yang diperlukan berupa jenis-jenis penyakit yang tergolong ke dalam penyakit anak dan obat apa yang sesuai untuk diberikan sebagai basis pengetahuan dari sistem. Contoh data gejala dapat dilihat pada Tabel 1 dan data obat dapat dilihat pada Tabel 2.

TABEL I.
DATA GEJALA YANG DIGUNAKAN

Kode Gejala	Nama Gejala
R50	Demam
R12	Nyeri ulu hati
R11	Muntah
R56	Kejang
R05	Batuk

Kode Gejala	Nama Gejala
L29	Gatal
R51	Nyeri kepala
R50	Demam

TABEL II
DATA OBAT YANG DIGUNAKAN

Golongan Obat	Kode Obat	Kode BPJS	Nama Obat
A02	BC01	1	Omeprazole
A02	BC02	0	Pantoprazole
A03	FA03	1	Domperidon
A04	AA02	0	Granisteron
N02	BE01	1	Paracetamol
N02	BE05	0	Propacetamol
N05	BA01	1	Diazepam
N03	AD01	0	Ethosuximide
R05	DA04	1	Codein
R05	DA09	0	Dextromethorphan
R06	AE07	1	Cetirizine
R06	AX27	0	Dexloratadine
A01	AC02	1	Dexamethasone
A07	EA04	0	Betametason

b) Studi literatur

Menggunakan beberapa referensi dari literatur ilmiah seperti jurnal, buku, ataupun data yang sudah ada sebagai acuan untuk pengembangan modul. Dengan berpedoman kepada ICD, daftar penyakit dan kode dari penyakit yang berada dalam lingkup penyakit anak (berdasarkan konsultasi dengan pakar/dokter) diambil dan dimasukkan ke dalam database yang telah dibuat, serta memasukkan obat yang sesuai dengan penyakit tersebut yang terdaftar ke dalam DOI untuk kemudian di proses.

B. Membangun sistem pendukung keputusan

Sistem Pendukung Keputusan atau *Decision Support System* (DSS) merupakan bagian dari sistem informasi yang memiliki basis komputerisasi yang berperan untuk mendukung aktivitas pengambilan keputusan.

Berikut merupakan tahapan-tahapan pengambilan keputusan menurut Miquel [8] :

1. Tahap identifikasi (*identification phase*)
Tahap *intelligence* merupakan tahapan dimana dilakukan perumusan masalah untuk mengetahui apa yang akan dilakukan dan informasi apa yang dibutuhkan.
2. Tahap pengembangan (*development phase*)
Tahap pengembangan merupakan tahapan dimana dilakukan analisa untuk merancang atau merumuskan model pemecahan masalah.
3. Tahap pemilihan (*selection phase*)
Tahap pemilihan merupakan tahapan pemilihan alternatif solusi dari permasalahan.

Clinical Decision Support System merupakan perangkat lunak dengan knowledge base yang dirancang untuk digunakan petugas kesehatan sebagai pendukung pengambilan keputusan klinis [9]. *Clinical Decision Support System* sendiri terdiri dari 2 kategori yaitu *knowledge based* dan *non-knowledge based*. Pada penelitian kali ini, penulis menggunakan *knowledge based* karena menggunakan basis pengetahuan yang sudah ada untuk mendukung kinerja dari modul dalam hal ini yaitu penentuan jenis obat yang sesuai dengan gejala dan gejala yang sudah terklasifikasi berdasarkan ICD. Kategori *knowledge based* ini sendiri terbagi menjadi beberapa subsistem, yakni:

a) Knowledge base

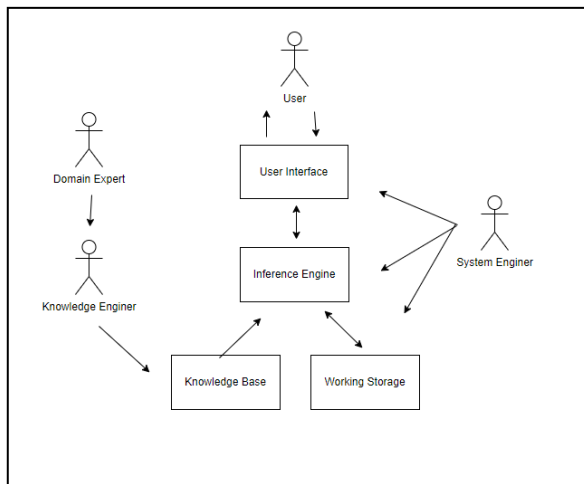
Menggunakan pengetahuan yang didapatkan langsung oleh pakar yang berpedoman kepada standar pengkodean gejala penyakit yang digunakan di Indonesia yaitu International Classification for Diseases (ICD) dan obat yang terdaftar di Indonesia yaitu DOI (Data Obat di Indonesia) berdasarkan hasil data yang dikumpulkan melalui pakar yang dapat dilihat pada Tabel 3. Sebagai contoh pasien memiliki gejala Muntah (R11) maka obat yang cocok adalah Granisteron (A03).

TABEL III
DATA KECOCOKAN OBAT DAN GEJALA

Id Gejala	Golongan Gejala
R05	R05
R05	R06
R11	A03
R11	A04
R12	A02
R50	N02
R51	N02
R51	A01
R56	N03
R56	N05
L29	R06
L29	A01
L29	A07

b) Reasoning engine

Reasoning engine berisi formula untuk menggabungkan aturan dari *Expert system* dan data yang sebenarnya. *Expert System* sendiri merupakan suatu pendekatan yang berasal dari pengetahuan ahli dalam bidang tertentu untuk membantu mengambil keputusan yang cukup kompleks [3]. Menurut Merritt [10], di dalam metode *Expert System*, terdapat struktur yang membentuk suatu *Expert System* yang dapat dilihat pada Gambar 2 [10].



Gambar 2. Struktur Expert System

Struktur *Expert System* terbagi menjadi 2 bagian, yaitu bagian sistem dan bagian yang mengacu pada orang yang mengemban peran untuk pengembangan sistem tersebut dalam berbagai posisi. Dalam bagian sistem sendiri terdapat 4 komponen yaitu *Knowledge Base*, *Working Storage*, *Inference Engine*, *User Interface*. *Knowledge base* sendiri merupakan bagian dimana pengetahuan dari pakar disimpan dan digunakan untuk menilai sesuatu. Selanjutnya *Working Storage* merupakan data spesifik yang digunakan untuk memecahkan masalah yang sedang terjadi. *Inference Engine* merupakan tempat terjadinya pemrosesan data berdasarkan *knowledge base* dan permasalahan yang sedang terjadi untuk memunculkan rekomendasi yang diharapkan akurat. Pada *Inference Engine* sendiri, terdapat beberapa metode yang dapat digunakan untuk memproses data, salah satunya adalah *Rule-based method*. Terakhir yaitu *User Interface* yang merupakan sarana interaksi antara *user* dan sistem itu sendiri. Selanjutnya, orang yang berperan dalam pengembangan sistem itu sendiri terbagi menjadi 3 bagian yaitu *Domain Expert*, yang mengacu pada orang yang berperan sebagai pakar dan memiliki pengetahuan (*knowledge base*) yang digunakan pada sistem. Kedua *Knowledge Engineer*, merupakan orang yang berperan dalam melakukan konversi dari pengetahuan pakar ke dalam bentuk tertentu untuk keperluan komunikasi atau pemrosesan data dalam komputasi. Terakhir merupakan *User*, yaitu orang yang menggunakan sistem dan berinteraksi langsung dengan *User Interface* demi mendapatkan hasil dari *Inference Engine* berdasarkan *Knowledge Base* dan permasalahan yang ada.

Pada *reasoning engine* ini, akan dilakukan pengecekan dari input data tentang gejala pasien dan input obat yang dimasukkan oleh petugas kesehatan untuk kemudian dilakukan pengecekan. Setelah pengecekan dilakukan, modul akan memberikan warning terkait kecocokan antara gejala pasien dan obat yang diberikan. Sebagai contoh pasien memiliki gejala muntah (R11) dan Kejang (R56) dan dokter memasukkan obat Granisteron (A03). Maka sistem akan mengeluarkan suatu peringatan karena obat untuk Kejang (R56) belum dimasukkan. Output dari modul ini diproses

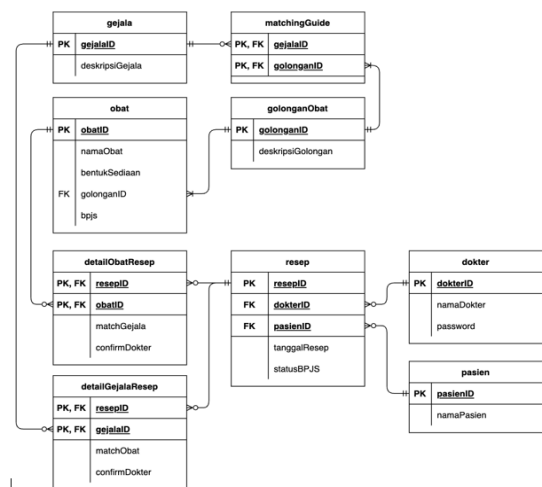
dengan menggunakan *Rule-based method* berdasarkan *knowledge* yang sudah didapatkan menggunakan metode *Expert System*.

c) *User interface*

Berisi tentang cara interaksi antara pengguna dan modul dimana pengguna bisa memasukkan data gejala pasien dan memasukkan obat yang akan diberikan. Kemudian, modul memberikan *feedback* berupa warning kepada pengguna setelah input diproses.

C. Perancangan basis data

Tabel resep merupakan tabel yang berisi data hasil input obat dan gejala untuk pasien dari dokter. Tabel resep sendiri memiliki *primary key* bernama *resepID*. Pada tabel resep, terdapat kolom *dokterID* dan *pasienID* dengan relasi *optional-many to mandatory-one* dimana kolom *dokterID* dan *pasienID* pada tabel resep merupakan *foreign key* dari tabel dokter dan tabel pasien yang menyimpan detail dari data pasien dan dokter. Tabel resep ini juga terhubung dengan tabel *detailObatResep* dan *detailGejalaResep* dengan relasi *mandatory-one to optional-many* dimana kedua tabel ini menyimpan data detail dari obat dan gejala serta konfirmasi yang dimasukkan oleh dokter. pada tabel *detailObatResep*, terdapat kolom *obatID* yang merupakan *primary key* dan juga sebagai *foreign key* dari tabel obat yang menyimpan detail data obat seperti *namaObat*, *bentukSediaan*, *golonganID*, dan *BPJS*. Kolom *golonganID* pada tabel obat ini diambil dari tabel *golonganObat* yang menyimpan detail data golongan obat seperti *deskripsiGolongan*. Pada tabel *detailGejalaResep* juga terdapat kolom *gejalaID* yang merupakan *primary key* sekaligus *foreign key* dari tabel gejala yang menyimpan detail data dari gejala seperti *deskripsiGejala*. Terakhir, terdapat tabel *matchingGuide* yang berisi kecocokan antara gejala dan obat dengan data *gejalaID* dan *golonganID* yang terhubung dengan tabel *golonganObat* dan tabel gejala. Struktur database dapat dilihat pada Gambar 3.



Gambar 3 Rancangan basis data

D. Perancangan pengujian sistem

Pengujian sistem dirancang dengan menggunakan *Black Box Testing* yang bertujuan untuk memastikan bahwa perangkat lunak berfungsi seperti yang diharapkan baik dari fitur, *input* dan *output*, interaksi pengguna, maupun logika bisnisnya (*Functional Testing*). *Test Case Testing* dilakukan untuk mengukur keakuratan output dari modul yang dibuat berdasarkan input. Rancangan skenario dari pengujian dapat dilihat pada Tabel 4.

TABEL IV
RANCANGAN SKENARIO PENGUJIAN SISTEM

Test Scenario	Input	Expected Output
Input obat BPJS sesuai dengan gejala pada pasien BPJS	Gejala: Demam Obat: Paracetamol	Sudah sesuai
Terdapat gejala yang belum diresepkan obat nya	Gejala: Batuk, Demam Obat: Paracetamol	Ada gejala yang belum diresepkan obat nya Gejala: Batuk
Terdapat Obat yang tidak sesuai dengan gejala	Gejala: Batuk Obat: paracetamol, Cetrizine	Ada obat yang tidak sesuai dengan gejala, obat: Cetrizine
Input obat non BPJS kepada pasien BPJS, obat dan gejala sesuai	Gejala: Demam Obat: Propacetamol	Obat tidak termasuk dalam BPJS
Input obat non BPJS kepada pasien non BPJS	Gejala: Demam Obat: Propacetamol	Sudah sesuai

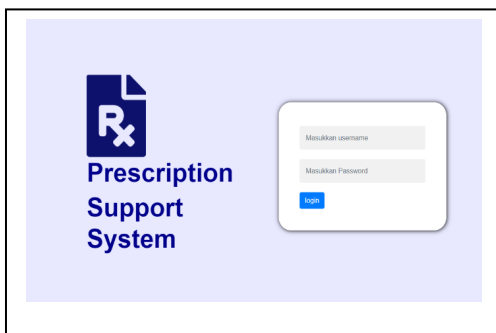
III. HASIL DAN ANALISIS

A. Implementasi

1) Implementasi antarmuka

a) Antarmuka login

Pada antarmuka *login* terdapat *textfield* untuk mengisi *username* dan *password*, *button*, logo, dan nama modul. Antarmuka login dapat dilihat pada Gambar 4.

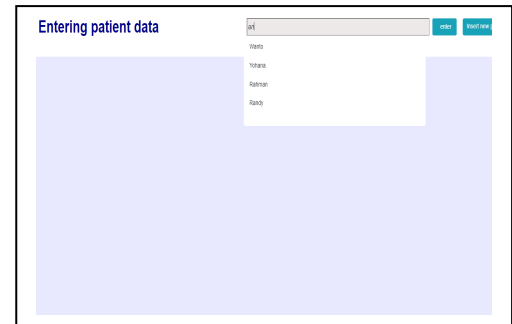


Gambar 4. Implementasi Antarmuka Login

b) Antarmuka pilih pasien

Pada antarmuka pilih pasien, *user* bisa memilih pasien dengan melakukan *search* berdasarkan *keyword*, Antarmuka

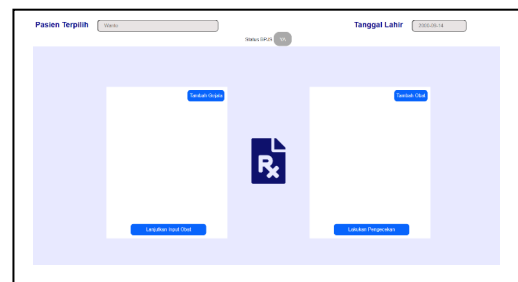
akan secara otomatis menampilkan nama yang sesuai dengan *input user*. Setelah melakukan pemilihan pasien, antarmuka akan berpindah ke *input* gejala dan obat. Antarmuka pilih pasien dapat dilihat pada Gambar 5.



Gambar 5. Implementasi Antarmuka Pilih Pasien

c) Antarmuka input data gejala dan obat

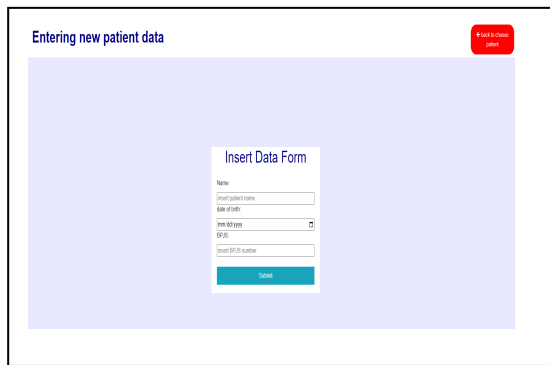
Pada antarmuka *input* gejala dan obat, *user* diharapkan memasukkan data gejala dan obat dari pasien yang sudah dipilih untuk kemudian dilakukan pengecekan kesesuaian antara obat dan gejala berdasarkan *input* sebelumnya. Pada antarmuka ini, terdapat juga informasi dari pasien yang sudah dipilih sebelumnya yang berisi nama pasien, tanggal lahir, dan status BPJS dari pasien tersebut pada bagian atas antarmuka. Antarmuka input data gejala dan obat dapat dilihat pada Gambar 6.



Gambar 6 Implementasi antarmuka input gejala dan obat

d) Antarmuka menambah pasien baru

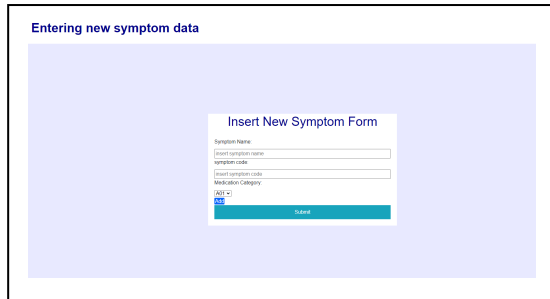
Pada antarmuka ini, dokter dapat menambahkan pasien baru apabila pasien belum terdaftar. Menambahkan pasien baru dilakukan dengan mengisi data nama pasien, tanggal lahir, dan BPJS dari pasien tersebut. Pada antarmuka ini juga terdapat tombol untuk kembali ke antarmuka pilih pasien. Setelah submit data baru pasien, akan ada sebuah notifikasi yang menunjukkan bahwa data pasien yang baru sudah disimpan ke dalam database. Antarmuka tambah pasien dapat dilihat pada Gambar 7.



Gambar 7. Implementasi Antarmuka Tambah Pasien Baru

e) Antarmuka menambahkan data gejala baru

Pada antarmuka ini, dokter dapat menambahkan data gejala yang baru apabila data gejala belum terdaftar. Disini dokter dapat menambahkan gejala yang baru dengan mengisi nama gejala, kode gejala (gejalaID) dan kategori obat (golonganID). Antarmuka tambah data gejala dapat dilihat pada Gambar 8.



Gambar 8. Implementasi Antarmuka Tambah Gejala Baru

f) Antarmuka tambah data obat baru

Selain tambah gejala, terdapat juga antarmuka tambah obat dimana dokter dapat menambahkan obat baru apabila obat belum terdaftar. Pada antarmuka ini, dokter menambahkan data obat dengan mengisi nama obat, status BPJS, kode obat, kategori obat (golonganID), dan bentuk sediaannya seperti pada Gambar 9.



Gambar 9. Implementasi Antarmuka Tambah Data Obat Baru

g) Alert

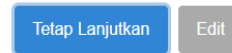
Contoh dari *output* yang diberikan jika data gejala atau obat tidak sesuai, dimana terlihat informasi yang

menunjukkan kekeliruan pada *input* gejala atau obat dan menunjukkan detail gejala atau obat yang belum sesuai. sedangkan apabila data sudah sesuai, akan muncul notifikasi yang menunjukkan bahwa data cocok dan sudah disimpan ke dalam *database* seperti yang dapat dilihat pada Gambar 10 dan Gambar 11.



Gejala Tidak Sesuai

Gejala yang belum diberi obat: Batuk



Gambar 10. Implementasi alert error



Input Sesuai

Data Sudah Tersimpan



Gambar 1. Implementasi *alert* sesuai

h) Antarmuka rekomendasi obat

Setelah dilakukan pengecekan *input* obat dan gejala, apabila *input* tidak sesuai maka akan ada rekomendasi obat yang sesuai dengan gejala yang dimasukkan. Rekomendasi tersebut diberikan dalam bentuk tabel seperti pada Gambar 12.

The suitable medicine for the symptoms you inputted

Symptoms	Medicines
Itchy	Dexamethasone
Itchy	test
Itchy	Betamethasone
Itchy	Calcitriol
Itchy	Dexamethasone
Cough	Sodium
Cough	Dexamethasone
Cough	Calcitriol
Cough	Dexamethasone
Nasal	Dexamethasone
Nasal	Griseofulvin

Gambar 2. Implementasi antarmuka rekomendasi obat

2) Implementasi modul pendukung keputusan

Modul pendukung keputusan dirancang dengan menggunakan *rule based expert system* dengan melakukan pengecekan antara obat dan gejala yang dimasukkan oleh *user*. Implementasi dari modul pendukung keputusan diawali dengan pengambilan ID dari gejala dan obat dari pasien yang telah dimasukkan sebelumnya oleh dokter. Kemudian, dilakukan pengecekan antara gejala dan obat yang telah dimasukkan berdasarkan *knowledge base* yang telah didapatkan sebelumnya yang terdapat pada tabel *matchingguide* pada *database*. *Database* dapat dilihat pada Gambar 13.

gejalaID	golongan
L29	A01
L29	A07
L29	R06
R05	R05
R05	R06
R11	A03
R11	A04
R12	A02
R50	N02
R51	A01
R51	N02
R56	N03
R56	N05

Gambar 13. Database matchingguide Kecocokan Obat dan Gejala

Struktur dari aturan yang digunakan yakni menggunakan 4 item sebagai elemen pengujianya dan 6 aturan. Elemen yang digunakan yakni *obatanpagejala* (ada obat yang tidak sesuai dengan gejala yang dialami pasien), *gejalatanpaobat* (ada gejala yang obatnya tidak sesuai), *pasienYobatN* (pasien BPJS diberikan obat non-BPJS), dan *pasienNobatY* (pasien non-BPJS diberikan obat BPJS). Aturan-aturan tersebut dapat dilihat pada Tabel 5.

TABEL V
STRUKTUR ATURAN YANG DIGUNAKAN

No	Deskripsi Aturan	Aturan
1	Gejala dan obat tidak sesuai	Gejala tidak sesuai: Yes Obat tidak sesuai: Yes Pasien non-BPJS obat BPJS: null Pasien BPJS obat non-BPJS: null
2	Hanya ada gejala yang belum	Gejala tidak sesuai: yes Obat tidak sesuai: null

	diresepkan obat nya	Pasien non-BPJS obat BPJS: null Pasien BPJS obat non-BPJS: null
3	Hanya ada obat tidak sesuai dengan gejala	Gejala tidak sesuai: null Obat tidak sesuai: yes Pasien non-BPJS obat BPJS: null Pasien BPJS obat non-BPJS: null
4	Gejala dan obat sesuai, Pasien BPJS diberikan obat non-BPJS	Gejala tidak sesuai: null Obat tidak sesuai: null Pasien non-BPJS obat BPJS: null Pasien BPJS obat non-BPJS: yes
5	Gejala dan obat sesuai, Pasien non-BPJS diberikan obat BPJS	Gejala tidak sesuai: null Obat tidak sesuai: null Pasien non-BPJS obat BPJS: yes Pasien BPJS obat non-BPJS: null
6	Gejala dan obat sesuai, status BPJS obat sama dengan pasien (non-BPJS atau BPJS)	Gejala tidak sesuai: null Obat tidak sesuai: null Pasien non-BPJS obat BPJS: null Pasien BPJS obat non-BPJS: null

B. Pengujian

Pengujian dilakukan untuk mengukur tingkat keakuratan dari *output* yang dihasilkan oleh modul. Pada pengujian akurasi modul, digunakan pendekatan *blackbox testing* dengan menggunakan metode *test case design*. Hasil pengujian dapat dilihat pada Tabel 6.

TABEL VI
HASIL SKENARIO PENGUJIAN

#	Scenario	Input	Expected Result	Actual Result
1	Ada gejala yang belum diresepkan obat nya dan ada obat yang tidak sesuai dengan gejala	Symptom: Fever, cough Medication: Domperidon	Mismatched between Symptoms and medication. Symptoms: Fever, Cough Medication: Domperidon	Mismatched between Symptoms and medication. Symptoms: Fever, Cough Medication: Domperidon
2	Ada gejala yang belum diresepkan obat nya	Symptom: Fever, Vomitting Medication: Paracetamol	Symptoms exist without corresponding medication. Symptoms: Vomitting	Symptoms exist without corresponding medication. Symptoms: Vomitting
3	Ada obat yang tidak	Symptom: Vomiting	Medication is provided	Medication is provided

#	Scenario	Input	Expected Result	Actual Result
	sesuai dengan gejala	Medication: Granisteron, Codein	without matching symptoms. Medication: Codein	without matching symptoms. Medications: Codein
4	Input gejala dan obat sudah sesuai, obat BPJS diberikan kepada pasien berstatus non-BPJS	Patient = non-bpjs Symptom: Fever Medication: Paracetamol (BPJS)	Accurate Input. Data saved	Accurate Input. Data saved
5	Input gejala dan obat sudah sesuai, obat non-BPJS diberikan kepada pasien BPJS	Patient = BPJS Symptom: Fever Medication: Propacetamol (non-BPJS)	Non-BPJS medication is given to BPJS patient. Medication:	Non-BPJS medication is given to BPJS patient. Medication:
6	Input gejala dan obat sudah sesuai, obat yang diberikan berstatus BPJS sesuai dengan pasien berstatus BPJS	Patient = BPJS Symptom: Itchy Medication: Dexamethasone (BPJS)	Accurate Input. Data saved	Accurate Input. Data saved
7	Input gejala dan obat sudah sesuai, obat yang diberikan berstatus non-BPJS sesuai dengan pasien berstatus non-BPJS	Patient = non-BPJS Symptom: Itchy Medication: Betametasone (non-BPJS)	Accurate Input. Data saved	Accurate Input. Data saved

sebuah modul pendukung keputusan. Pengujian dari modul yang sudah dibangun ini dilakukan dengan menggunakan metode *test case design* dimana beberapa skenario dibuat untuk memastikan seluruh aturan dari *rule based* berdasarkan *knowledge base* yang sudah diimplementasikan sudah sesuai. Pada pengujian ini, terdapat 7 skenario untuk menguji aturan tersebut yang berisi input (data input gejala dan obat dari dokter), *expected result* (hasil yang diinginkan), dan *result* (hasil sebenarnya).

Pada skenario pertama hingga ketiga, berisi pengujian apakah input data gejala dan obat belum sesuai. Jika iya, maka *expected result* dari skenario ini adalah “*Mismatched between Symptoms and medication*” apabila ada gejala dan obat yang tidak sesuai, “*Symptoms exist without corresponding medication*” apabila hanya ada data gejala yang belum diresepkan obat nya, dan “*Medication is provided without matching symptoms*” apabila hanya ada data obat yang tidak sesuai dengan input gejala. Untuk skenario pertama hingga ketiga, hasil skenario *input* gejala dan obat memberikan *result* (hasil) yang sesuai dengan apa yang diharapkan (*expected result*) sehingga skenario pertama hingga ketiga dari pengujian dinyatakan lolos. Setelah itu, pengujian pada skenario keempat hingga ketujuh dilakukan untuk menguji apakah status BPJS dari data obat yang sudah sesuai cocok dengan status BPJS dari pasien. Apabila status BPJS dari pasien dan obat sama baik itu BPJS maupun non-BPJS, maka *expected result* modul adalah “*Accurate input*” seperti pada pengujian skenario 6 dan 7. Berdasarkan hasil uji skenario input, dapat terlihat bahwa *result* dari modul sama dengan *expected result* sehingga skenario 6 dan 7 dinyatakan lolos. Apabila status BPJS dari pasien dan obat tidak sesuai yakni pasien dengan status non-BPJS dan obat dengan status BPJS seperti pada skenario 4, *expected result* Nya juga sama dengan pengujian pada skenario 6 dan 7. Hasil dari skenario *input* pada pengujian skenario 4 menunjukkan *result* yang sama dengan *expected result* sehingga pengujian skenario 4 dinyatakan lolos. Apabila pasien dengan berstatus BPJS dan obat berstatus non-BPJS seperti pada skenario kelima, maka *expected result* Nya adalah “*Non-BPJS medication is given to BPJS patient.*”. *Result* dari skenario input untuk menguji skenario kelima ini sesuai dengan *expected result* sehingga skenario 5 juga dinyatakan lolos. Semua *test* skenario yang sudah dilakukan menunjukkan *result* yang sama dengan *expected result* dan ini menunjukkan bahwa metode *rule based* dapat digunakan untuk membuat sebuah modul pendukung keputusan antara gejala dan obat.

C. Pembahasan

Setelah dilakukan implementasi pada antarmuka dan pendukung keputusan dari modul dengan metode *rule based*, aturan-aturan *rule based* yang sebelumnya ditetapkan berdasarkan *knowledge base* yang didapatkan melalui pakar, dapat diimplementasikan ke modul untuk membangun

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil dari implementasi sistem yang telah dilakukan, sebuah modul pendukung keputusan untuk melihat kecocokan antara gejala dan obat berhasil tercipta. Dengan menggunakan metode *Rule Based Expert System*, modul pendukung keputusan dapat menilai kecocokan antara

obat dan gejala yang dimasukkan pengguna dan memberikan *output* berupa *alert*. Modul juga dapat memberikan rekomendasi obat yang sesuai.

Hasil dari pengujian memperlihatkan bahwa *output* pada setiap skenario menghasilkan hasil yang sesuai dengan apa yang diharapkan (*expected result*).

Modul pendukung keputusan yang dibuat sudah dapat menghasilkan *output* yang sesuai dengan apa yang diharapkan, tetapi terdapat beberapa hal yang dapat diperbaiki ataupun dikembangkan seperti misalnya:

1. menambahkan data sebagai *knowledge base* agar dapat mencakup permasalahan yang lebih besar.
2. modul dapat diintegrasikan dengan sistem Kesehatan yang ada pada klinik atau rumah sakit. Dengan demikian, modul dapat diintegrasikan dengan data yang ada pada klinik atau rumah sakit tersebut secara langsung.
3. Menambahkan aturan yang baru untuk pengembangan modul selanjutnya.

DAFTAR PUSTAKA

- [1] Kristine Ruggiero PhD MSN RN CPNP and Michael Ruggiero MHS PA-C, *Fast Facts Handbook for Pediatric Primary Care: A Guide for Nurse Practitioners and Physician Assistants*, 1st ed. Springer Publishing Company, 2020.
- [2] Gerardus Blokdyk, "Electronic prescribing A Complete Guide," Kindle.5STARCOOKS, 2018.
- [3] A. Barr, P. R. Cohen, and E. A. Feigenbaum, "The Handbook of Artificial Intelligence," 1982.
- [4] Steven C. Shaffer, *CODE YOUR OWN RULE-BASED EXPERT SYSTEM: A step-by-step explanation*, Kindle. Shaffer Media Enterprises LLC, 2020.
- [5] M. Digital, K. Englebert, A. Porterfield, W. K. Willis, and A. Coustasse, "Electronic prescribing and its implementation in the United States," 2014.
- [6] Marfuah, "Algoritma Sistem Pendukung Keputusan dalam Monitoring Penyelesaian Berkas Perkara Pidana menggunakan Rule Based Expert Systems dan Teknologi Radio Frequency Identification (RFID)," *Jurnal Saintekom : Sains, Teknologi, Komputer dan Manajemen*, vol. 07, 2017.
- [7] Aprilia, "Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Dengan Inference Forward Menerapkan Metode Dempster Shafer Berbasis WEB," *Coding Jurnal Komputer dan Aplikasi*, 2018.
- [8] Miquel Sánchez-Marrè, *Intelligent Decision Support Systems*, Kindle. Springer, 2022.
- [9] Robert Greenes and Guilherme Del Fiol, *Clinical Decision Support and Beyond: Progress and Opportunities in Knowledge-Enhanced Health and Healthcare*, 3rd ed. Academic Press, 2023.
- [10] D. Merritt, *Expert Systems in Prolog*. 2017.