BAB III

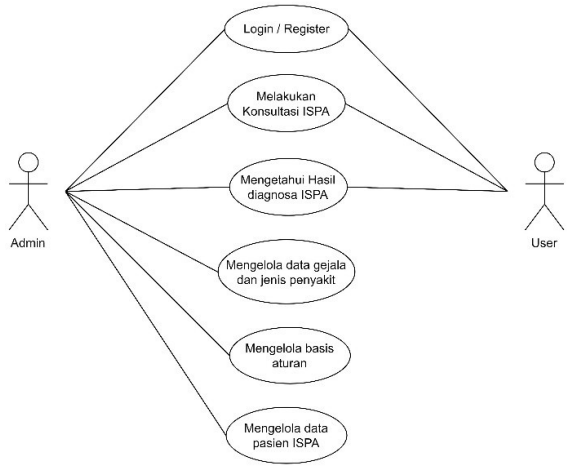
HASIL DAN PEMBAHASAN

3.1 Rancang Bangun UML (*Unified Modeling Language*)

Bagian ini merupakan bentuk model cara kerja sistem yang dirancang.

3.1.1 *Use Case Diagram*

Use Case Diagram adalah teknik pemodelan cara kerja pemograman sistem melalui relasi antara aktor yang menggunakan sistem tersebut. Teknik ini juga menjelaskan fungsi-fungsi pemograman kepada pihak berwenang yang menggunakan sistem tersebut (Irfan, Siregar, and Handoko 2023). Dalam penerapannya teknik pemodelan dapat dijabarkan melalui gambar 3.1.



Gambar 3.1 *User Case Diagram*

Berdasarkan gambar diatas, terdapat interaksi antara *admin* dan *user* dalam melakukan proses *Login* / *Registry* untuk mengakses program sistem pakar dalam mendiagnosa penyakit ISPA. Hal ini dijelaskan melalui Tabel 3.1.

Tabel 3.1 Model *Use Case Scenario Login/Registry*

Use Case Scenario Login/Register		
1	*Objective*	Memungkinkan *Admin* dan *User* mengakses sistem pakar konsultasi jenis penyakit
2	*Actor*	*Admin* dan *User*
3	*Pre-Condition*	Sistem menampilkan halaman *Login* berupa *Form* yang berisi *Username* dan *Password* agar pengguna bisa mengakses sistem pakar tersebut. Selain itu, sistem juga menampilkan halaman *Register* berupa *form* yang berisi *Email*, *Username* dan *Password User* dapat menjadi *member* dari sistem agar bisa mengakses program sistem pakar tersebut.
4	*Main Flow*	Sistem menerima *Input* dan memverifikasi data pengguna berupa *E-mail*, *Username* dan *Password*. Jika hasilnya benar, sistem mengizinkan untuk masuk.
5	*Alternative Flow*	Sistem menampilkan pesan "*Username* dan *Password* salah, Silahkan coba lagi". Jika *Admin* dan *User* salah memasukkan *Username* dan *Password* maka sistem otomatis kembali ke *Login Forms* agar user dapat menginput ulang data *Login* sistem kembali.
6	*Post Condition*	Sistem menampilkan halaman beranda program sistem pakar sehingga *Admin* dan *User* berhasil mengakses sistem pakar diagnosa penyakit ISPA .

Selanjutnya terdapat interaksi antara *admin* dan *user* dalam konsultasi penyakit berdasarkan gejalanya yang ditunjukkan dalam Tabel 3.2.

Tabel 3.2 Model *Use Case Scenario* Konsultasi Penyakit

***Use Case Scenario* Konsultasi Penyakit**		
1	*Objective*	Memungkinkan *Admin* dan *User* melakukan konsultasi penyakit yang diderita
2	*Actor*	*Admin* dan *User*
3	*Pre-Condition*	Sistem menampilkan *Form* yang berisi Nama Pasien beserta *Checkbox* yang berisi beberapa gejala penyakit sehingga *Admin* dan *User* dapat menginput nama dan memilih gejala – gejala penyakit yang dialami melalui *Checkbox Forms* tersebut.
4	*Main Flow*	Sistem menerima *Input* nama pasien beserta gejala yang dialami.
5	*Alternative Flow*	Sistem menampilkan pesan "Silahkan pilih salah satu diantaranya" jika *Admin* dan *User* tidak memasukkan nama serta memilih gejala yang ada di *Checkbox Forms* tersebut.
6	*Post Condition*	Sistem berhasil menyimpan data nama pasien beserta gejala yang dialami.

Kemudian, pada Gambar 3.1 terdapat suatu interaksi antara *Admin* dan *User* dalam mengetahui hasil diagnosa jenis penyakit ISPA melalui Tabel 3.3.

Tabel 3.3 Model *Use Case Scenario* Hasil Diagnosa Penyakit

***Use Case Scenario* Hasil Diagnosa Penyakit**		
1	*Objective*	Memungkinkan *Admin* dan *User* mengetahui hasil diagnosa jenis penyakit yang dialami
2	*Actor*	*Admin* dan *User*
3	*Pre-Condition*	Sistem menampilkan *Form* yang berisi Nama Pasien beserta gejala yang dipilih serta menampilkan jenis penyakit yang dialami pasien dengan hasil persentase yang didapat dan solusi penanganan penyakit tersebut.
4	*Main Flow*	Sistem telah memverifikasi nama pasien dan gejala yang dialami
5	*Post Condition*	Sistem berhasil hasil persentase yang terdapat pada jenis penyakit yang dialami.

Pada Gambar 3.1, terdapat suatu interaksi antara *Admin* dan sistem dalam mengelola data gejala dan jenis penyakit melalui Tabel 3.4.

Tabel 3.4 Model *Use Case Scenario* Kelola Data Gejala dan Jenis Penyakit

***Use Case Scenario* Kelola Gejala dan Jenis Penyakit**		
1	*Objective*	Memungkinkan *Admin* mengelola gejala dan jenis penyakit yang dialami
2	*Actor*	*Admin*
3	*Pre-Condition*	Sistem menampilkan Tabel yang berisi kode gejala dan nama gejala serta fitur Edit dan Hapus gejala. Tiap baris pada Tabel juga berisi data kode penyakit dan nama penyakit beserta keterangan penyakit dan solusi penanganan yang tepat.
4	*Main Flow*	Sistem berhasil menampilkan halaman pengelolaan data gejala dan jenis penyakit.
5	*Post Condition*	Sistem berhasil menambah, mengedit dan menghapus data gejala dan jenis penyakit

Selanjutnya pada Gambar 3.1, terdapat suatu interaksi antara *admin* dan sistem dalam mengelola basis aturan suatu sistem berdasarkan Tabel 3.5.

Tabel 3.5 Model *Use Case Scenario* Kelola Basis Aturan

***Use Case Scenario* Kelola Basis Aturan**		
1	*Objective*	Memungkinkan *admin* mengelola basis aturan dari sistem pakar
2	*Actor*	*Admin*
3	*Pre-Condition*	Sistem memiliki data gejala dan jenis penyakit yang telah diinput.serta Tabel basis aturan berupa nama jenis penyakit beserta gejala – gejalanya. Didalam Tabel tersebut terdapat fitur detail untuk mengetahui basis aturan yang telah dibuat, dan juga fitur mengedit dan menghapus basis aturan.
4	*Main Flow*	Sistem berhasil menampilkan halaman pengelolaan basis aturan
5	*Post Condition*	Sistem berhasil menambah, mengedit dan menghapus basis aturan

Kemudian pada Gambar 3.1, terdapat suatu interaksi antara admin dan sistem dalam mengelola data pasien berdasarkan Tabel 3.6.

Tabel 3.6 Model *Use Case Scenario* Kelola Data Pasien

***Use Case Scenario* Kelola Data Pasien**		
1	*Objective*	Memungkinkan *admin* mengelola data pasien pengidap ISPA
2	*Actor*	*Admin*
3	*Pre-Condition*	Sistem memiliki data nama pasien beserta hasil persentase jenis penyakit yang dialami dan mampu mengedit nama pasien beserta gejala dan jenis penyakit
4	*Main Flow*	Sistem berhasil menampilkan halaman data pasien
5	*Post Condition*	Sistem berhasil menambah, mengedit dan menghapus data pasien

3.1.2 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) merupakan model penjabaran secara terperinci berdasarkan konteks *Use Case Diagram*. Penjabaran *Data Flow Diagram* sendiri dilakukan dalam beberapa *level* yang sesuai dengan perancangan sistem (Sunardi 2023). *Data Flow Diagram* digunakan untuk menggambarkan fungsi dari sistem informasi bekerja (Nozomi and Wadisman 2022).

1) Diagram Konteks (*Data Flow Diagram Level 0*)

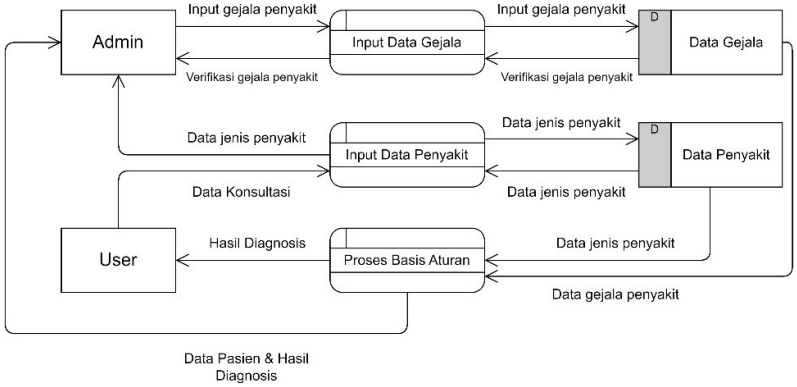
Diagram konteks adalah bentuk pemodelan yang berfungsi menampilkan ruang lingkup sistem yang dapat menerima atau membagikan perancangan data (Sunardi 2023). Diagram konteks juga berisi *input* dan *output* yang dihasilkan sistem secara keseluruhan (Julisawati and Yanti 2022). Bentuk dari pemodelan diagram konteks dapat diuraikan melalui gambar 3.2.

Gambar 3.2 Diagram Konteks (*Data Flow Diagram Level 0*)

Berdasarkan gambar diatas, *User* berperan sebagai pasien yang melakukan pemeriksaan penyakit dengan menentukan gejala – gejala yang dialami. Kemudian, data pasien yang berupa nama pasien beserta gejala – gejalanya akan dimasukkan kedalam *Database*. Setelah proses penginputan data, maka dilakukan proses verifikasi data pasien melalui *Database* yang akan dimasukkan kedalam sistem. Kemudian, sistem akan menentukan serta menampilkan hasil diagnosa penyakit yang dialami pasien.

2) Diagram Tingkat 1 (*Data Flow Diagram Level 1*)

Diagram Tingkat 1 ialah wujud pemodelan diagram yang merepresentasikan visual aliran data dalam sesuatu sistem (Julisawati and Yanti 2022). Diagram Tingkat 1 berfungsi untuk menguraikan proses aliran informasi yang terdapat pada diagram konteks (Ummah, Sodikin, and Susetyo 2019). Hal ini dapt diuraikan melalui gambar 3.3.



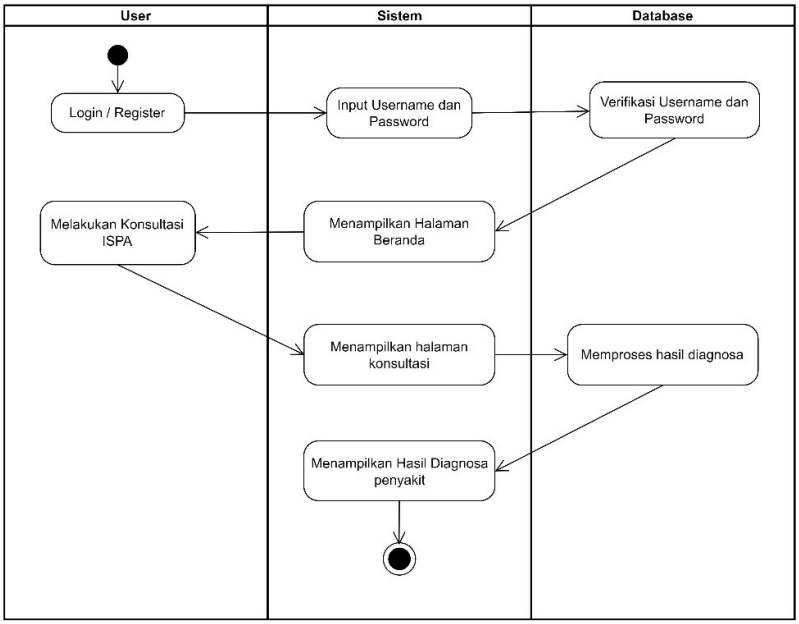
Gambar 3.3 Diagram Tingkat 1 (*Data Flow Diagram Level 1*)

Melalui gambar diatas, terdapat penjabaran aktifitas diagram konteks pada perancangan sistem sebagai berikut :

1. *Admin* memasukkan data gejala penyakit kedalam sistem, kemudian sistem akan memverifikasi data gejala penyakit sehingga sistem dapat menampilkan data gejala penyakit yang diinput.
2. *User* dapat menginput data konsultasi yang disimpan melalui *Database* jenis penyakit. Selain itu, *Admin* juga dapat menginput data jenis penyakit sehingga sistem dapat memverifikasi data yang diinput.
3. Data gejala dan jenis penyakit yang telah disimpan melalui *Database* masing -masing digunakan untuk menentukan basis aturan yang didapat serta menghitung persentase dari kedua *Database* tersebut. Kemudian, sistem akan menampilkan hasil diagnosa penyakit kepada *Admin* dan *User*.

3.1.3 Diagram Aktivitas (*Activity Diagram*)

Diagram Aktivitas merupakan bentuk penggambaran aliran kerja sistem informasi yang tersedia di perangkat lunak (Musthofa and Adiguna 2022). Diagram ini menunjukkan berbagai macam alur aktivitas pemograman yang dirancang kedalam bentuk kumpulan aksi (Rhido Rezwana, Dwinita Arwidiyarti, and Hendri Ramdan 2024). Seperti yang diuraikan pada gambar 3.4.



Gambar 3.4 Diagram Aktivitas (*Activity Diagram*)

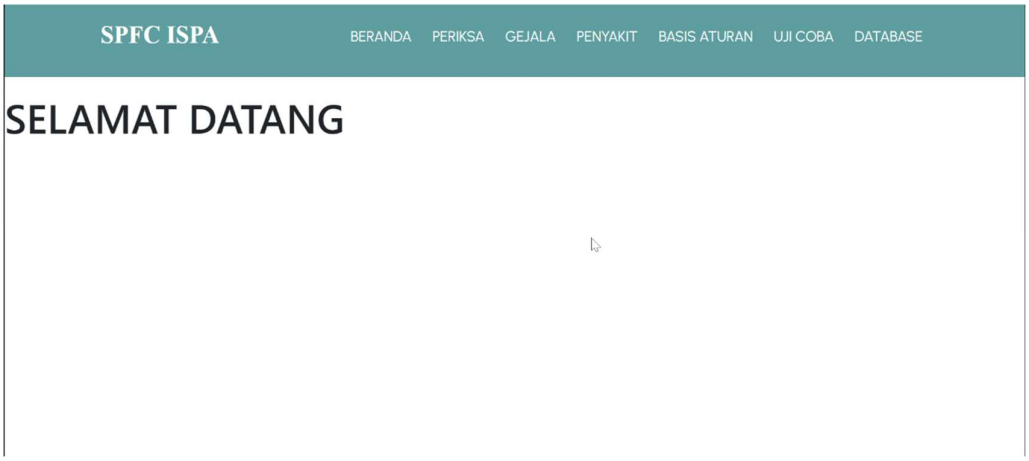
Berdasarkan gambar tersebut, terdapat aktifitas *User d*engan sistem yang dimulai dengan melakukan *Login*, melakukan konsultasi penyakit dan melihat hasil diagnosa penyakit. Berikut ini adalah penjelasan prosesnya :

1. *User* akan melakukan *Login* ke sistem melalui halaman *Website* di mana mereka harus memasukkan *Username* dan *Password* agar sistem dapat memverifikasi informasi *User* untuk mengakses program sistem pakar diagnosa penyakit ISPA.
2. *User* mengklik *Menu* periksa untuk melakukan konsultasi penyakit yang diderita Kemudian sistem akan menampilkan halaman konsultasi, di mana *User* harus memasukkan nama serta mengidentifikasi gejala yang diderita.
3. Kemudian sistem akan memproses hasil konsultasi yang diinput *User* dengan memverifikasi data *User* melalui *Database*. Sehingga sistem dapat menampilkan hasil diagnosa jenis penyakit yang dialami *User* melalui gejala yang ada.

3.2 Perancangan Antarmuka

Perancangan antarmuka merupakan bentuk tampilan atau dikenal *GUI* (*User Interface*) dari program sistem yang dirancang.

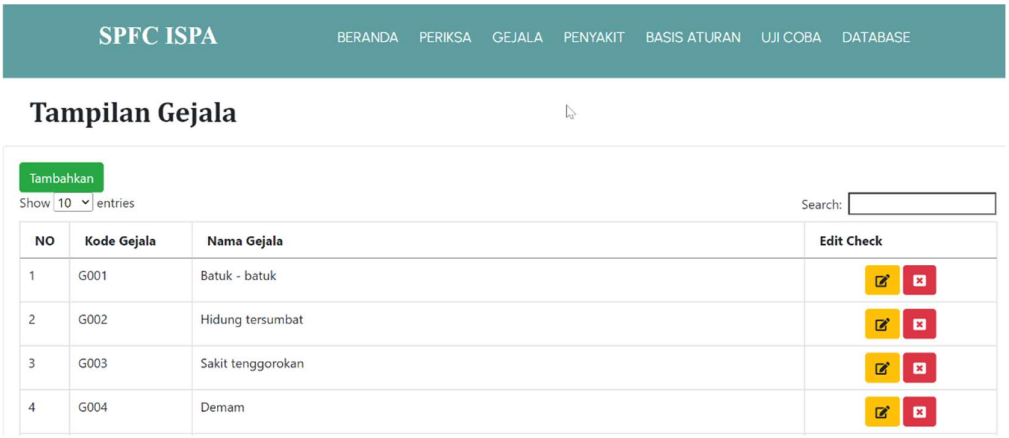
3.2.1 Halaman Beranda



Gambar 3.5 Halaman Beranda Sistem Pakar Diagnosa Penyakit ISPA

Halaman ini merupaka halaman utama sistem yang pertama kali diinput sebelum masuk halaman lain. Berdasarkan Gambar 3.5, halaman beranda Sistem Pakar terdiri dari beberapa Menu seperti halaman konsultasi, halaman gejala, dan halaman basis aturan.

3.2.2 Halaman Gejala



Gambar 3.6 Halaman Gejala Penyakit ISPA

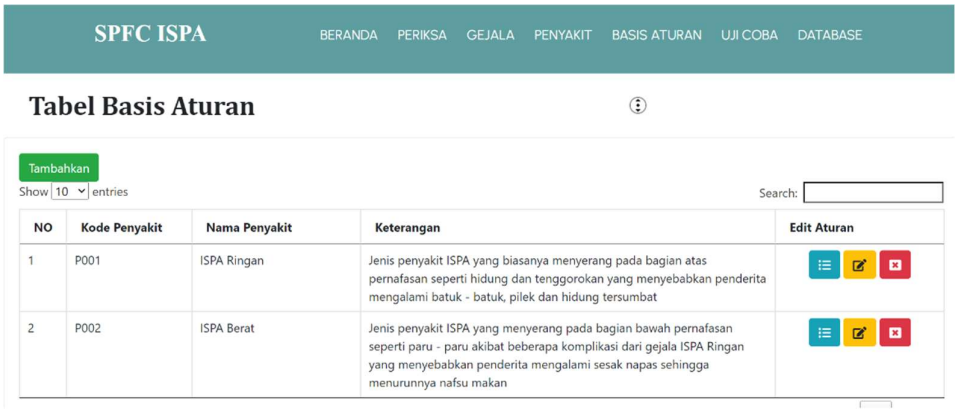
Berdasarkan Gambar 3.6, terdapat Tabel yang berisi data kode dan nama gejala yang disertai fitur *Edit Check* yang memudahkan *Admin* dalam mengedit atau menghapus data gejala penyakit. Diatas Tabel terdapat tombol hijau bertuliskan "Tambahkan" yang berfungsi untuk menambahkan data gejala penyakit.

3.2.3 Halaman Penyakit

Gambar 3.7 Halaman Jenis Penyakit ISPA

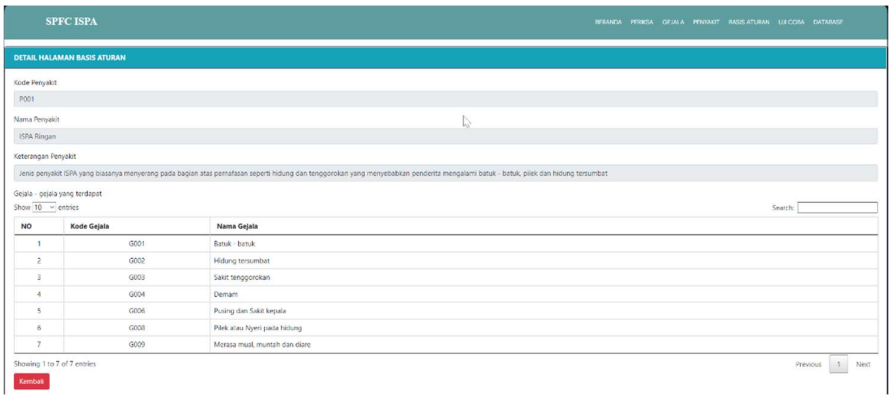
Berdasarkan Gambar 3.7, terdapat Tabel yang berisi data kode dan nama jenis penyakit beserta penjelasan dan solusi penanganan penyakit tersebut. didalam Tabel tersebut *Admin* bisa menambahkan data jenis penyakit, mengedit dan menghapus data tersebut dengan mudah.

3.2.4 Halaman Basis Aturan



Gambar 3.8 Halaman Basis Aturan

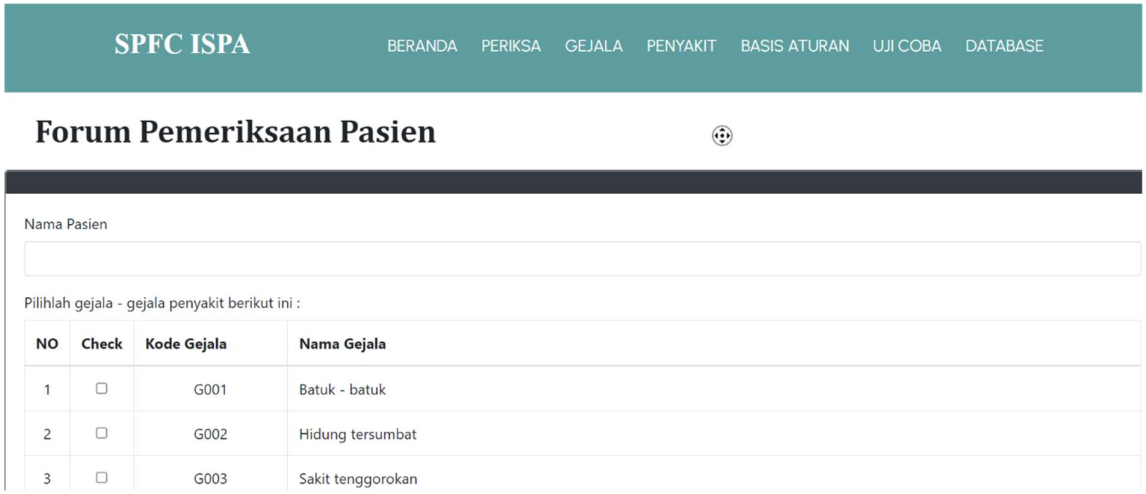
Berdasarkan Gambar 3.8, terdapat Tabel yang berisi data jenis penyakit serta yang disertai fitur untuk menambahkan, mengedit dan menghapus basis aturan dari sistem pakar.



Gambar 3.9 Halaman Detail Basis Aturan

Berdasarkan Gambar 3.9, data dari basis aturan sistem pakar diagnosa penyakit terdiri dari kode penyakit, nama penyakit, penjelasan jenis penyakit yang dialami berserta gejala – gejala penyakit ditentukan oleh *Admin*.

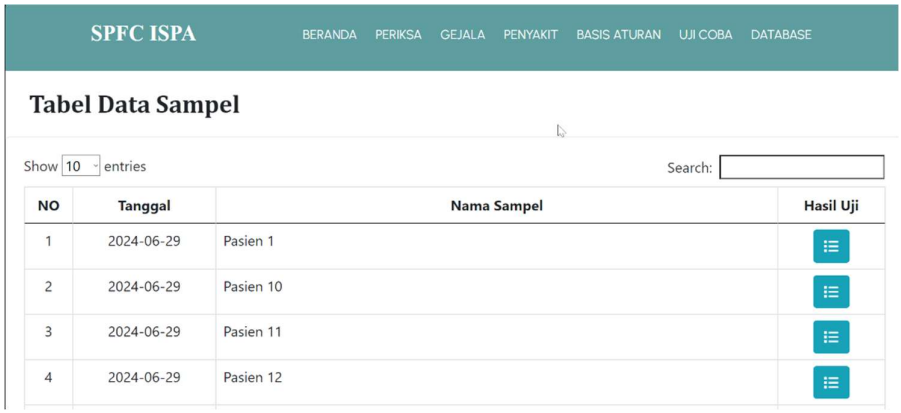
3.2.5 Halaman Pemeriksaan Pasien



Gambar 3.10 Halaman Konsultasi Pasien

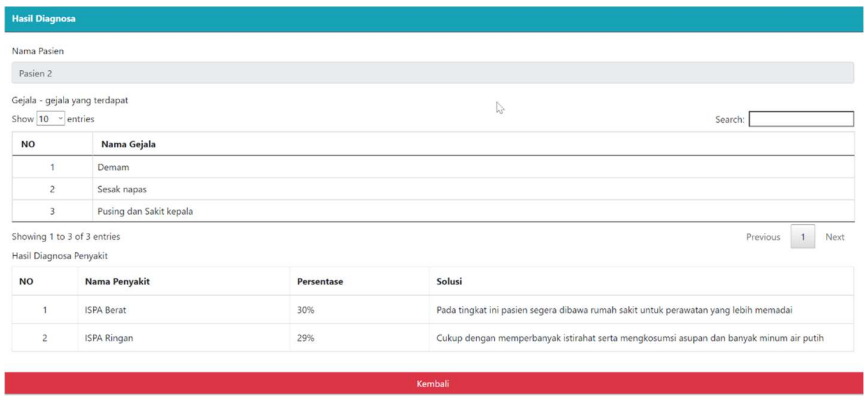
Berdasarkan Gambar 3.10, *User* atau pasien yang ingin melakukan konsultasi bisa memasukkan nama beserta memilihi gejala penyakit yang dialami melalui *Form* yang berada pada halaman konsultasi. Setelah *User* memasukkan nama serta gejala yang dialami, sistem akan menginput dan memverifikasi data konsultasi melalui *Database*.

3.2.6 Halaman Hasil Diagnosa



Gambar 3.11 Halaman Data Pasien

Berdasarkan Gambar 3.11, terdapat Tabel yang berisi nama pasien dan tanggal pasien mengakses sistem dan melakukan konsultasi. Serta terdapat fitur *detail* yang menampilkan hasil diagnosa jenis penyakit ISPA.



Gambar 3.12 Halaman Hasil Diagnosa Pasien ISPA

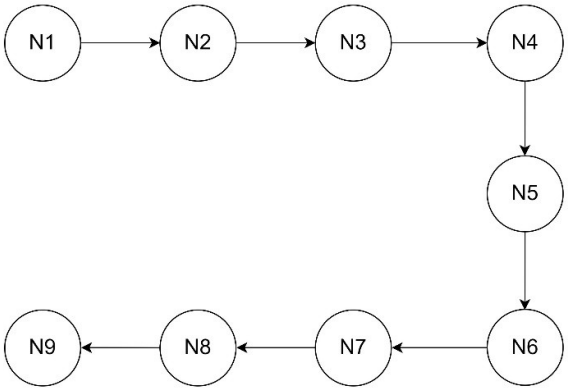
Berdasarkan pada Gambar 3.12, terdapat nama pasien beserta gejala yang dialami, yang ditentukan melalui hasil persentase seberapa besar kemungkinan penyakit yang dialami pasien, serta cara menangani penyakit tersebut. Jika suatu penyakit memiliki hasil persentase yang banyak, maka pasien mengalami penyakit. Dan jika hasilnya sedikit, maka pasien tidak mengalami penyakit.

3.3 Pengujian Hasil

Pengujian ini dilakukan melakukan menentukan hasil kinerja sistem yang dirancang. Pengujian ini dilakukan dengan memasukkan data, menghitung hasil akurasi perbandingan antara data sistem dengan data penelitian. Sehingga menghasilkan hasil *output* yang diinginkan.

3.3.1 *Whitebox Testing*

Whitebox Testing adalah teknik pengujian struktur internal dalam suatu pemograman untuk mengidentifikasi kesalahan sistem dalam bekerja melalui perangkat lunak (Sie, Izmy Alwiah Musdar, and Syamsul Bahri 2022). Pengujian ini melibatkan pengecekan setiap detail *Module* seperti *Input* data gejala dan jenis penyakit, pengelolaan basis aturan hingga pengujian hasil persentase diagnosa penyakit.



Gambar 3.13 *Flowchart Whitebox Testing*

Berdasarkan Gambar 3.13, pengujian *whitebox testing* akan dilakukan dari *Node 01* (N1) hingga *Node 09* (N9) yang menggambarkan alur kerja sistem yang diuji, sehingga setiap jalur akan dianalisis untuk memastikan setiap kondisi dan *node* diuji secara menyeluruh. Hasil uji *Whitebox Testing* pada sistem pakar dapat digambarkan pada Tabel 3.7.

Tabel 3.7 Hasil Uji *Whitebox Testing*

Path	**Jalur**	***Scenario***	**Hasil Pengujian**
1	N1 → N2	N1 : *Input* data gejala N2 : Verifikasi data gejala	Berhasil
2	N3 → N4	N3 : *Input* data jenis penyakit N4 : Verifikasi data jenis penyakit	Berhasil
3	N5 → N6	N5 : *Input* kelola basis aturan N6 : Verifikasi bobot berdasarkan gejala yang ada	Berhasil
4	N7 → N8 → N9	N7 : *Input* data pasien N8 : Verifikasi data pasien melalui basis aturan N9 : Verifikasi hasil diagnosa	Berhasil

3.3.2 *Blackbox Testing*

Blackbox Testing merupakan pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data sistem serta memeriksa fungsional dari perangkat lunak (Iqbal Muhammad 2019). Pengujian ini dimulai dengan pemeriksaan *input* data gejala dan jenis penyakit, penginputan basis aturan hingga penghitungan hasil diagnosa penyakit. Hasil pengujian dapat dijabarkan melalui Tabel 3.8.

Tabel 3.8 Hasil Uji *Blackbox Testing*

ID Test	Deksripsi Test	Hasil yang diantisipasi	Hasil Uji	Kesimpulan
TC – 01	*Input* gejala penyakit	Sistem dapat menerima dan memproses data gejala yang diinput.	Sistem berhasil menginput data gejala penyakit serta dapat mengedit dan menghapus data tersebut.	Sesuai
TC – 02	*Input* jenis penyakit	Sistem dapat menerima dan memproses data jenis penyakit yang diinput.	Sistem berhasil menginput dapat jenis penyakit sehingga *Admin* dapat menambahkan, mengedit dan menghapus data yang ingin diinput.	Sesuai
TC – 03	Kelola Basis Aturan	Sistem dapat mengelola basis aturan dari jenis penyakit berdasarkan gejala yang ada.	Sistem berhasil mengelola basis aturan serta menentukan bobot berdasarkan gejala yang ada	Sesuai
TC – 04	Kelola Basis Aturan	Hasil Diagnosa dihitung berdasarkan basis aturan yang diinput	Data konsultasi berhasil disimpan kedalam *Database*, sehingga sistem dapat memverifikasi hasil persentase diagnosa penyakit.	Sesuai

3.3.3 Perhitungan Akurasi

Pengujian hasil akurasi didapatkan dengan pendekatan algoritma metode *Forward Chaining* untuk mengevaluasi sistem yang dirancang. Besarnya hasil akurasi yang didapatkan berdasarkan jumlah data akurat yang diperoleh melalui sistem dengan data hasil penelitian sebanyak 150 data . Dari hasil pengujian tersebut jumlah data akurat yang diperoleh sebanyak 131 data dari 150 data penelitian beserta data yang tidak akurat sebanyak 19 data. Maka proses penentuan nilai akurasi dapat dihitung sebagai berikut.

$$Akurasi = \frac{131}{150} \times 100\ \% = 87{,}3\ \%$$

Berdasarkan hasil perhitungan diatas, dapat disimpulkan bahwa nilai akurasi yang didapatkan sebesar 87 % yang menunjukkan sistem memiliki performa baik dalam mendiagnosa jenis penyakit.