

LAPORAN PROYEK AKHIR

Sistem Pakar Diagnostik Penyakit Umum



Anggota:

11S21003 - Benhard Yudha

11S21034 - Enrico Hezekiel Sirait

11S21039 - Hanny Yosephine Br.Kaban

11S21042 - Pany Irene Matondang

11S21047 - Corry Betriks Sitorus

11S21050 - Ella Tasya Marito Silaban

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

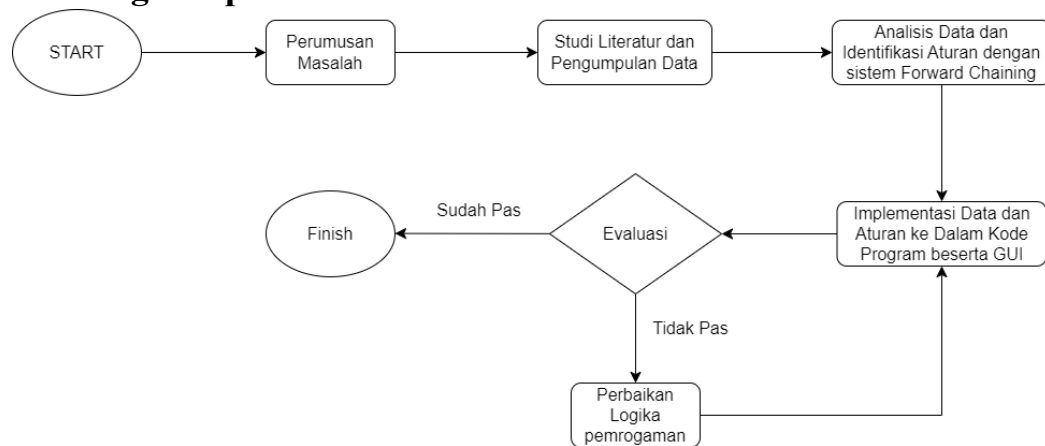
Role Anggota:

Nama	Peran
Benhard Yudha	<ul style="list-style-type: none">• Kode program bagian <i>GUI(Graphic User Interface)</i> dan Dokumen bagian rancangan aplikasi
Enrico Hezkiel Sirait	<ul style="list-style-type: none">• Kode program bagian <i>knowledge base</i> (dasar pengetahuan) dan <i>rule base</i> (dasar aturan), dan dokumen bagian metode dan aturan
Hanny Yosephine Br.Kaban	<ul style="list-style-type: none">• Kode program bagian Algoritma <i>Forward Chaining</i> (Maju Berantai) dan Dokumen bagian pengaplikasian program
Pany Irene Matondang	<ul style="list-style-type: none">• Kode program bagian <i>GUI(Graphic User Interface)</i> dan Dokumen bagian tampilan aplikasi
Corry Betriks Sitorus	<ul style="list-style-type: none">• Kode program bagian Algoritma <i>Forward Chaining</i> (Maju Berantai) dan Dokumen bagian rancangan aplikasi
Ella Tasya Marito Silaban	<ul style="list-style-type: none">• Kode program bagian <i>knowledge base</i> (dasar pengetahuan) dan <i>rule base</i> (dasar aturan), dan dokumen bagian deskripsi

I. Deskripsi

Layanan kesehatan merupakan hal yang sangat diperlukan oleh masyarakat luas untuk mengetahui tentang kesehatan. seperti puskesmas, rumah sakit ataupun tempat tempat medis dapat memberikan layanan kesehatan kepada masyarakat. oleh sebab itu tempat-tempat medis ini berusaha memberikan kenyamanan dalam hal konsultasi, kebersihan tempat, dan pelayanan yang memuaskan. untuk mewujudkan itu kami membuat suatu sistem pakar diagnostik diagnosa penyakit umum pada anak-anak menggunakan metode Forward Chaining. Dimana sistem ini, diharapkan dapat membantu dokter dalam menangani pasien yang berkonsultasi. sistem pakar ini dibuat untuk pengolahan data konsultasi penyakit umum pada anak-anak yang dapat dilakukan secara digital, sehingga waktu yang diperlukan lebih singkat dalam membantu melayani pasien yang berkonsultasi.

II. Rancangan Aplikasi



1. Perumusan Masalah

Pada tahap ini, kami melakukan identifikasi dan penentuan fokus pada permasalahan yang akan diselidiki. Perhatian kami tertuju pada pengembangan sebuah sistem pakar diagnostik untuk diagnosis penyakit umum pada anak-anak.

2. Studi Literatur dan Pengumpulan Data

Pada tahapan ini, kami memulai dengan melakukan studi literatur yang komprehensif yaitu dengan mencari referensi terbaru dan relevan melalui berbagai sumber seperti jurnal ilmiah yang terkemuka, database online, dan juga artikel.

3. Analisis Data dan Identifikasi Aturan dengan sistem Forward Chaining

4. Implementasi Data dan Aturan ke Dalam Kode Program Beserta GUI

Implementasi aplikasi yang dilakukan dimulai dengan pembuatan kode terkait Sistem Pakar Diagnostik dengan menggunakan metode Forward Chaining.

a. Mendefinisikan class aturan

Dalam class ini akan mengatur kombinasi gejala yang diperlukan untuk menarik hasil tertentu yang dimana setiap aturan terdiri dari sekumpulan gejala dan hasil yang terkait dengan gejala tersebut.

Dalam code programming dapat dilihat:

```
4 class Aturan:  
5     def __init__(self, gejala, hasil):  
6         self.gejala = gejala  
7         self.hasil = hasil  
8
```

- b. Mendefinisikan class GUI Expert System Diagnostik
- Dalam class ini akan mengatur antarmuka aplikasi serta logika untuk menampilkan gejala dan logika untuk menampilkan gejala diagnosa.
- Untuk code programming dapat dilihat:

```
9 class GUIExpertSystemDiagnostik:  
10     def __init__(self, master, aturan):  
11         self.master = master  
12         self.master.title("Sistem Pakar Diagnostik")  
13         self.master.geometry("550x500") # Menyesuaikan ukuran jendela awal  
14  
15         self.aturan = aturan  
16         self.var_gejala = [tk.BooleanVar() for _ in range(len(set(gejala for aturan in aturan for gejala in aturan.gejala)))]  
17         self.buat_gui()  
18  
19     def buat_gui(self):  
20         background_color = "#f0f0f0" # Warna latar belakang  
21         widget_color = "#ffffff" # Warna widget  
22         text_color = "#333333" # Warna teks  
23  
24         self.master.configure(bg=background_color) # Set latar belakang utama  
25  
26         label_judul = tk.Label(self.master, text="Sistem Pakar Diagnostik", font="Arial", 14, "bold", bg=background_color, fg="black")  
27         label_judul.pack(pady=10)  
28  
29         frame_gejala = tk.Frame(self.master, bg=background_color)  
30         frame_gejala.pack(pady=10, padx=10)  
31  
32         col_count = 2 # Jumlah kolom yang diinginkan  
33         row = 0  
34         col = 0  
35  
36         for i, gejala in enumerate(set(gejala for aturan in self.aturan for gejala in aturan.gejala)):  
37             checkbox = tk.Checkbutton(frame_gejala, text=gejala, variable=self.var_gejala[i], font="Arial", 12, bg=background_color, fg=text_color, selectcolor=widget_color, activebackground=background_color, activecolor=background_color)  
38             checkbox.grid(row=row, column=col, pady=10, padx=10, sticky="w")  
39  
40             col += 1  
41             if col == col_count:  
42                 col = 0  
43                 row += 1  
44  
45         tombol_diagnosa = tk.Button(self.master, text="Diagnosa", command=self.diagnosa, font="Arial", 14, "bold", bg="black", fg="white", relief=tk.RAISED)  
46         tombol_diagnosa.pack(pady=20)  
47  
48         self.label_hasil_diagnosa = tk.Label(self.master, text="", font="Arial", 12, wraplength=500, justify="left", bg=background_color, fg=text_color)  
49         self.label_hasil_diagnosa.pack(pady=10)  
50  
51     def tanya_gejala(self):  
52         return [var.get() for var in self.var_gejala]  
53  
54     def faktanya(self, gejala_terpilih):  
55         fakta_terpenuhi = set()  
56         for i, gejala in enumerate(set(gejala for aturan in self.aturan for gejala in aturan.gejala)):  
57             if gejala_terpilih[i]:  
58                 fakta_terpenuhi.add(gejala)  
59         return fakta_terpenuhi  
60  
61     def maju_berantai(self, gejala_terpilih):  
62         hasil = []  
63         fakta_terpenuhi = self.faktanya(gejala_terpilih)  
64         for aturan in self.aturan:  
65             if set(aturan.gejala).issubset(fakta_terpenuhi):  
66                 hasil.append(aturan.hasil)  
67         return hasil  
68  
69     def tampilkan_hasil_diagnosa(self, hasil):  
70         if hasil:  
71             teks_hasil = f"Berdasarkan gejala yang Anda sampaikan, hasil yang mungkin adalah: {', '.join(set(hasil))}"  
72         else:  
73             teks_hasil = "Tidak dapat menentukan hasil berdasarkan gejala yang diberikan."  
74  
75         self.label_hasil_diagnosa.config(text=teks_hasil)  
76         messagebox.showinfo("Hasil Diagnosa", teks_hasil)  
77  
78     def diagnosa(self):  
79         gejala_terpilih = self.tanya_gejala()  
80  
81         # Memeriksa apakah jumlah gejala yang dipilih kurang dari 3  
82         if sum(gejala_terpilih) < 3:  
83             messagebox.showerror("Error", "Pilih setidaknya 3 gejala untuk melakukan diagnosa.")  
84             return  
85  
86         hasil_diagnosa = self.maju_berantai(gejala_terpilih)  
87         self.tampilkan_hasil_diagnosa(hasil_diagnosa)  
88
```

```
43         col = 0  
44         row += 1  
45  
46         tombol_diagnosa = tk.Button(self.master, text="Diagnosa", command=self.diagnosa, font="Arial", 14, "bold", bg="black", fg="white", relief=tk.RAISED)  
47         tombol_diagnosa.pack(pady=20)  
48  
49         self.label_hasil_diagnosa = tk.Label(self.master, text="", font="Arial", 12, wraplength=500, justify="left", bg=background_color, fg=text_color)  
50         self.label_hasil_diagnosa.pack(pady=10)  
51  
52     def tanya_gejala(self):  
53         return [var.get() for var in self.var_gejala]  
54  
55     def faktanya(self, gejala_terpilih):  
56         fakta_terpenuhi = set()  
57         for i, gejala in enumerate(set(gejala for aturan in self.aturan for gejala in aturan.gejala)):  
58             if gejala_terpilih[i]:  
59                 fakta_terpenuhi.add(gejala)  
60         return fakta_terpenuhi  
61  
62     def maju_berantai(self, gejala_terpilih):  
63         hasil = []  
64         fakta_terpenuhi = self.faktanya(gejala_terpilih)  
65         for aturan in self.aturan:  
66             if set(aturan.gejala).issubset(fakta_terpenuhi):  
67                 hasil.append(aturan.hasil)  
68         return hasil  
69  
70     def tampilkan_hasil_diagnosa(self, hasil):  
71         if hasil:  
72             teks_hasil = f"Berdasarkan gejala yang Anda sampaikan, hasil yang mungkin adalah: {', '.join(set(hasil))}"  
73         else:  
74             teks_hasil = "Tidak dapat menentukan hasil berdasarkan gejala yang diberikan."  
75  
76         self.label_hasil_diagnosa.config(text=teks_hasil)  
77         messagebox.showinfo("Hasil Diagnosa", teks_hasil)  
78  
79     def diagnosa(self):  
80         gejala_terpilih = self.tanya_gejala()  
81  
82         # Memeriksa apakah jumlah gejala yang dipilih kurang dari 3  
83         if sum(gejala_terpilih) < 3:  
84             messagebox.showerror("Error", "Pilih setidaknya 3 gejala untuk melakukan diagnosa.")  
85             return  
86  
87         hasil_diagnosa = self.maju_berantai(gejala_terpilih)  
88         self.tampilkan_hasil_diagnosa(hasil_diagnosa)  
89
```

c. Penggunaan metode Forward Chaining

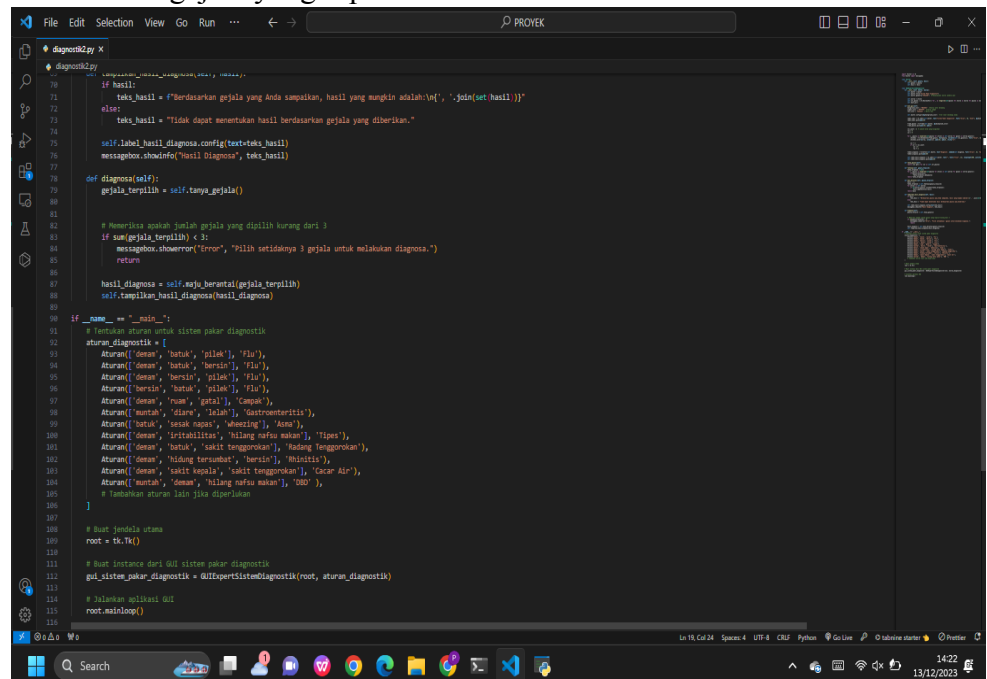
Metode Forward Chaining digunakan untuk menentukan hasil diagnosa berdasarkan gejala yang diberikan oleh pengguna. Yang dimana asalkan dapat melibatkan pengujian setiap gejala pada aturan yang ada dan menyimpan hasil yang diperbolehkan.

Implementasi dalam code programming:

```
def maju_berantai(self, gejala_terpilih):
    hasil = []
    fakta_terpenuhi = self.faktanya(gejala_terpilih)
    for aturan in self.aturan:
        if set(aturan.gejala).issubset(fakta_terpenuhi):
            hasil.append(aturan.hasil)
    return hasil
```

d. Penggunaan Tkinter untuk membuat antarmuka aplikasi

Tkinter digunakan untuk membuat antarmuka aplikasi, termasuk tombol “Diagnosa” yang akan menghasilkan hasil diagnosa berdasarkan gejala yang dipilih user.



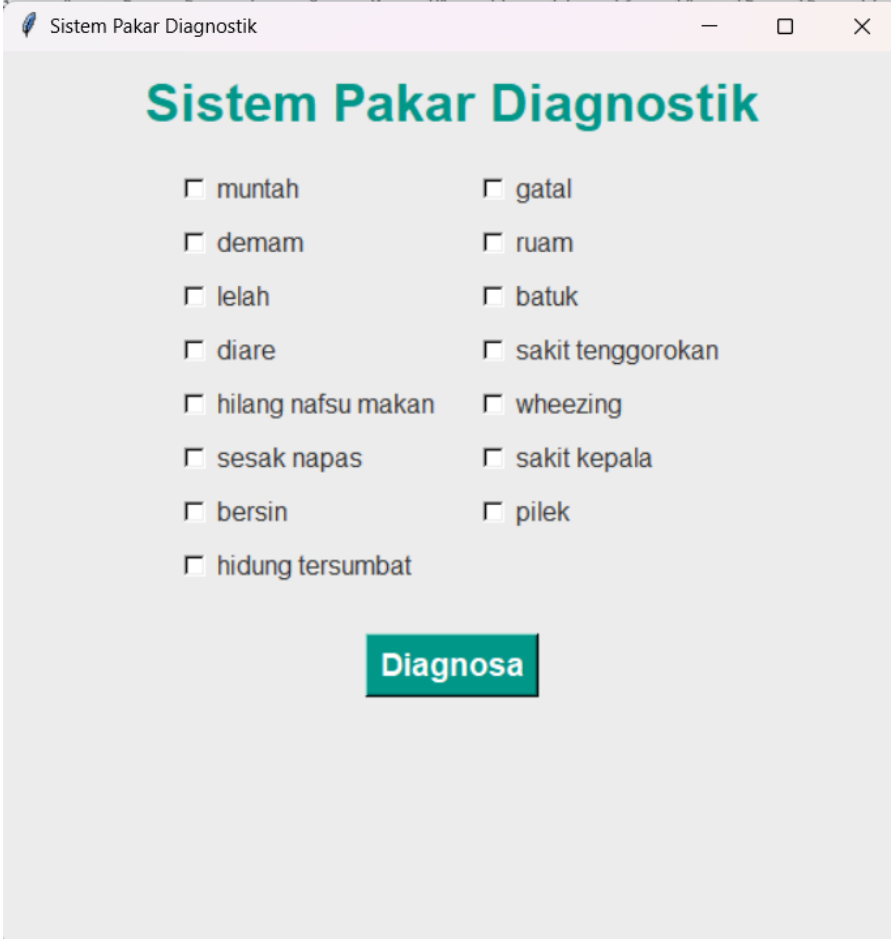
5. Evaluasi

Setelah menerapkan sistem pakar diagnostik ke dalam kode program, kami akan melakukan serangkaian pengujian sederhana. Uji coba ini akan menggunakan input acak guna memverifikasi kinerja metode Forward Chaining. Jika output dari program tidak sejalan dengan data dan aturan yang telah disusun, hal ini menunjukkan kemungkinan kesalahan dalam logika pemrograman. Kami akan melakukan pemeriksaan mendalam dan perbaikan terhadap kode program.

Setelah melalui proses evaluasi yang cermat dan memastikan kesesuaian dengan data dan aturan yang telah ditetapkan, program akan siap untuk digunakan.

III. Tampilan Aplikasi (Desain Interface)

Berikut adalah tampilan GUI (Graphical User Interface) untuk aplikasi yang dibangun.



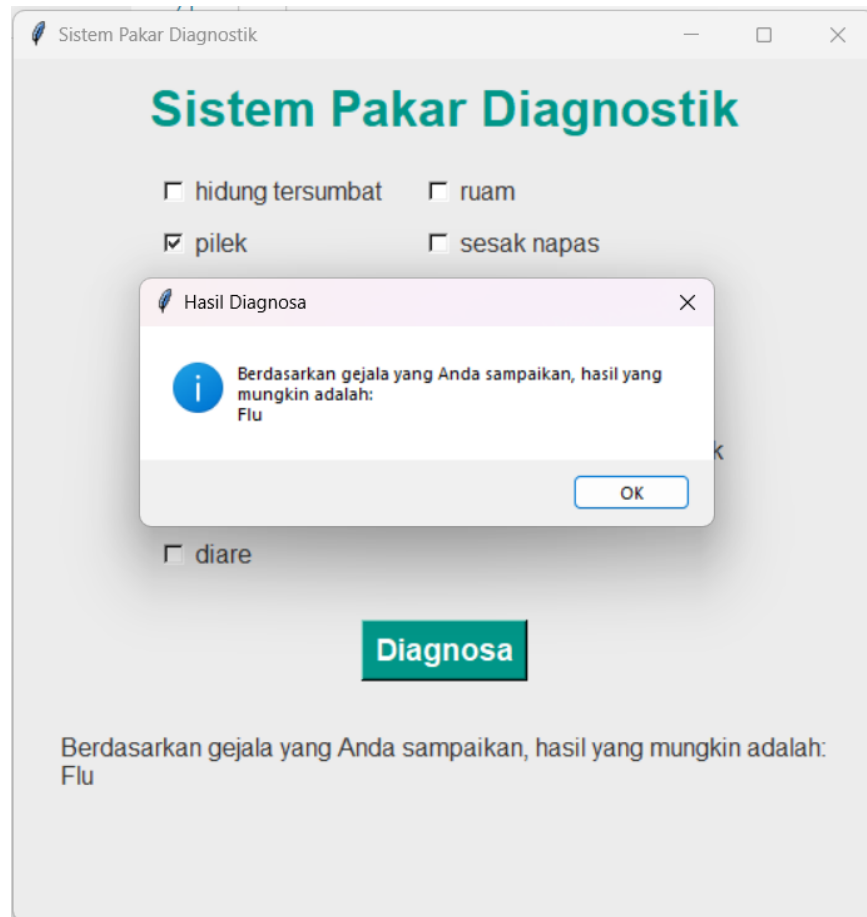
Sistem Pakar Diagnostik

Sistem Pakar Diagnostik

- ☐ muntah
- ☐ demam
- ☐ lelah
- ☐ diare
- ☐ hilang nafsu makan
- ☐ sesak napas
- ☐ bersin
- ☐ hidung tersumbat
- ☐ gatal
- ☐ ruam
- ☐ batuk
- ☐ sakit tenggorokan
- ☐ wheezing
- ☐ sakit kepala
- ☐ pilek

Diagnosa

Dapat dilihat dari tampilan GUI di atas ada tampilan GUI sederhana. Pembuatan aplikasi ini digunakan 'tkinter as tk' yang memungkinkan kemudahan untuk pengguna karena tampilan yang sederhana dan interaksi dengan pengguna memungkinkan sangat responsif. Pada tampilan GUI ini terdapat tampilan antarmuka pengguna yang menggunakan elemen grafis seperti jendela, tombol, kotak centang, label, dan lainnya untuk berinteraksi dengan pengguna secara visual.



Jendela aplikasi ini akan menampilkan beberapa elemen antarmuka pengguna:

- ❖ **Judul Aplikasi:** Terdapat judul "Sistem Pakar Diagnostik" yang ditampilkan dengan font yang besar dan mencolok.
- ❖ **Daftar Gejala:** Berisi daftar kotak centang yang mewakili gejala-gejala yang terkait dengan penyakit pada sistem pakar diagnostik. Pengguna dapat memilih gejala yang relevan dengan kondisi yang diamati.
- ❖ **Tombol "Diagnosa":** Sebuah tombol yang akan memicu proses diagnosa saat diklik. Setelah pengguna memilih gejala, mereka dapat menekan tombol ini untuk memulai diagnosa.
- ❖ **Hasil Diagnosa:** Area di bawah tombol "Diagnosa" yang akan menampilkan hasil diagnosa berdasarkan gejala yang dipilih pengguna. Hasil akan ditampilkan dalam bentuk teks di area ini.
- ❖ **Pesan Kotak Dialog:** Setelah menampilkan hasil diagnosa di dalam aplikasi, akan muncul sebuah kotak pesan (message box) yang memberikan informasi yang sama kepada pengguna dalam jendela terpisah.
- ❖ **Warna dan Tata Letak:** Aplikasi menggunakan kombinasi warna latar belakang, teks, dan widget agar mudah dibaca dan menarik secara visual.

Aplikasi ini memungkinkan pengguna untuk memilih gejala dan mendapatkan hasil diagnosa berdasarkan aturan yang telah ditentukan sebelumnya dalam program.

IV. Pengaplikasian Program

Langkah-langkah dalam penggunaan program

1. Klik kolom centang yang berisikan gejala-gejala apa saja yang dialami oleh pasien

Berikut hasil tampilannya :

Sistem Pakar Diagnostik

<input type="checkbox"/> diare	<input type="checkbox"/> demam
<input type="checkbox"/> wheezing	<input type="checkbox"/> hidung tersumbat
<input type="checkbox"/> pemberian makan buruk	<input type="checkbox"/> gatal
<input type="checkbox"/> sesak napas	<input type="checkbox"/> bersin-bersin
<input type="checkbox"/> pilek	<input type="checkbox"/> muntah
<input type="checkbox"/> lelah	<input type="checkbox"/> batuk
<input type="checkbox"/> ruam	<input type="checkbox"/> iritabilitas
<input type="checkbox"/> sakit tenggorokan	

Diagnosa

Tidak dapat menentukan hasil berdasarkan gejala yang diberikan.

Sistem Pakar Diagnostik

<input type="checkbox"/> diare	<input checked="" type="checkbox"/> demam
<input type="checkbox"/> wheezing	<input type="checkbox"/> hidung tersumbat
<input type="checkbox"/> pemberian makan buruk	<input type="checkbox"/> gatal
<input type="checkbox"/> sesak napas	<input type="checkbox"/> bersin-bersin
<input checked="" type="checkbox"/> pilek	<input type="checkbox"/> muntah
<input type="checkbox"/> lelah	<input checked="" type="checkbox"/> batuk
<input type="checkbox"/> ruam	<input type="checkbox"/> iritabilitas
<input type="checkbox"/> sakit tenggorokan	

Diagnosa

Berdasarkan gejala yang Anda sampaikan, hasil yang mungkin adalah:
Flu

2. Selanjutnya Klik Button “Diagnosa”

Berikut Hasil Tampilannya :

Sistem Pakar Diagnostik

<input type="checkbox"/> diare	<input checked="" type="checkbox"/> demam
<input type="checkbox"/> wheezing	<input type="checkbox"/> hidung tersumbat
<input type="checkbox"/> pemberian makan buruk	<input type="checkbox"/> gatal
<input type="checkbox"/> sesak napas	<input type="checkbox"/> bersin-bersin
<input checked="" type="checkbox"/> pilek	<input type="checkbox"/> muntah
<input type="checkbox"/> lelah	<input checked="" type="checkbox"/> batuk
<input type="checkbox"/> ruam	<input type="checkbox"/> iritabilitas
<input type="checkbox"/> sakit tenggorokan	

Diagnosa

Tidak dapat menentukan hasil berdasarkan gejala yang diberikan.

3. Saat button diklik maka analisa diagnosa akan dilakukan dan nantinya akan menampilkan hasil perkiraan penyakitnya

Berikut Hasil yang Ditampilkan :

Sistem Pakar Diagnostik

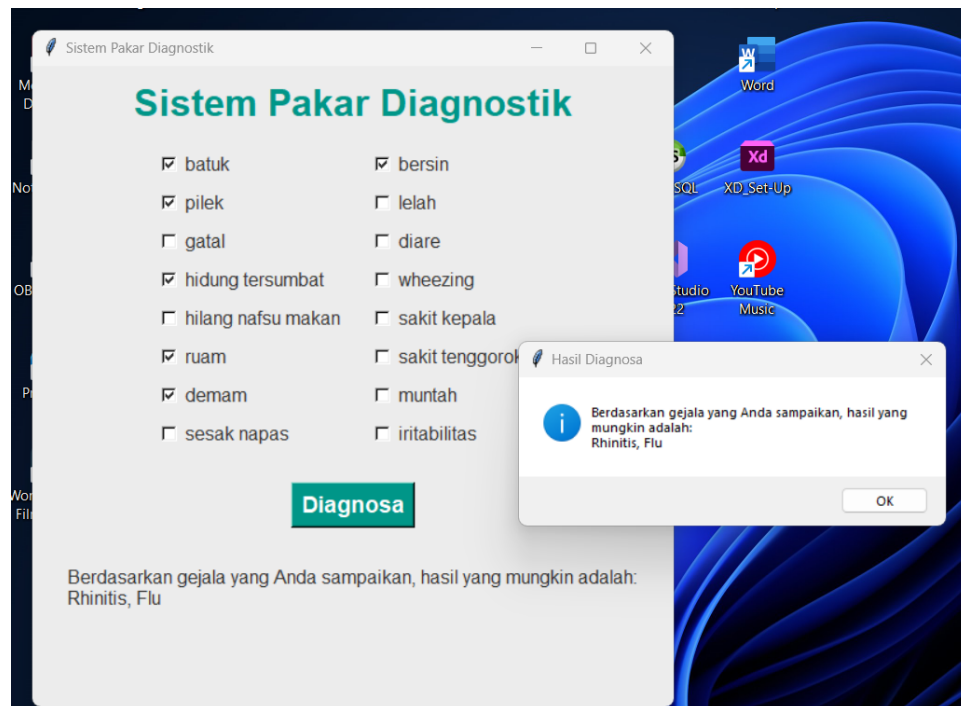
<input type="checkbox"/> diare	<input checked="" type="checkbox"/> demam
<input type="checkbox"/> wheezing	<input type="checkbox"/> hidung tersumbat
<input type="checkbox"/> pemberian makan buruk	<input type="checkbox"/> gatal
<input type="checkbox"/> sesak napas	<input type="checkbox"/> bersin-bersin
<input checked="" type="checkbox"/> pilek	<input type="checkbox"/> muntah
<input type="checkbox"/> lelah	<input checked="" type="checkbox"/> batuk
<input type="checkbox"/> ruam	<input type="checkbox"/> iritabilitas
<input type="checkbox"/> sakit tenggorokan	

Hasil Diagnosa

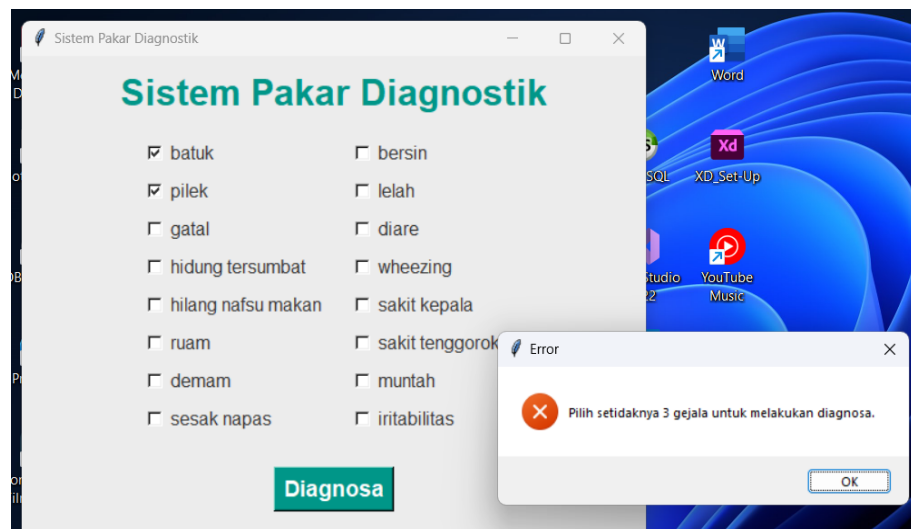
Berdasarkan gejala yang Anda sampaikan, hasil yang mungkin adalah:
Flu

OK

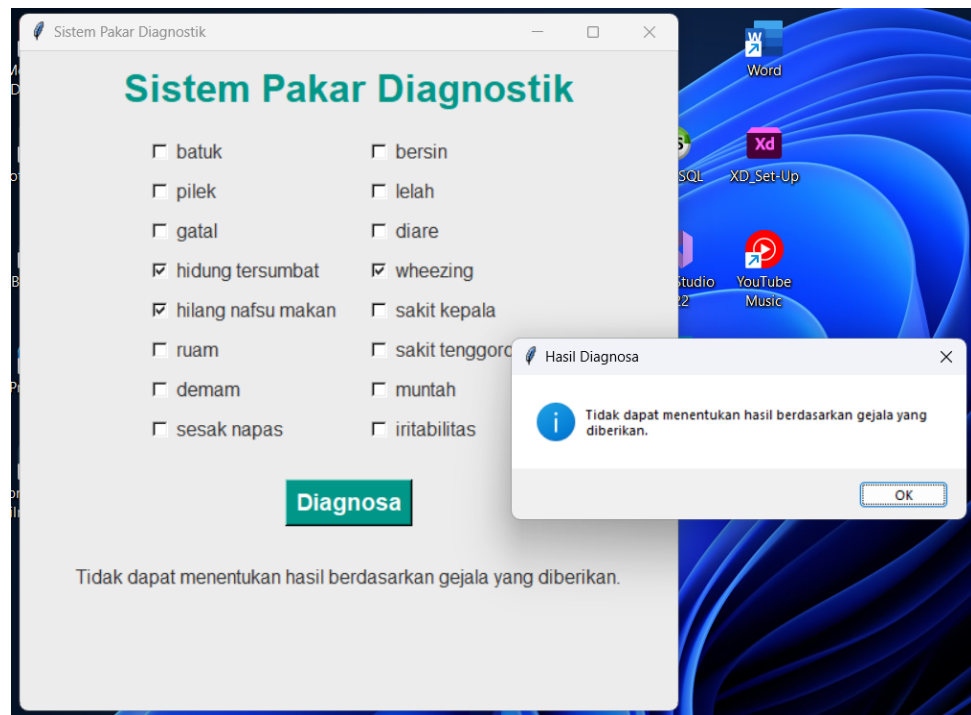
4. Saat gejala yang dipilih sampai beberapa dan ternyata memenuhi aturan yang lain maka akan menampilkan gejala penyakitnya



5. Saat gejala yang dipilih tidak sampai 3 atau lebih maka akan menampilkan pesan error



6. Saat tidak ada gejala yang terpenuhi maka akan menampilkan pesan seperti pada gambar dibawah ini



V. Metode dan Aturan

Tabel 1. Kode dan Nama Penyakit

Kode Penyakit	Nama Penyakit
P01	Flu
P02	Campak
P03	Gastroenteritis
P04	Asma
P05	Tipes
P06	Radang Tenggorokan
P07	Rhinitis
P08	Cacar Air
P09	DBD

Tabel 3. Logika aturan yang dipakai dalam program

No	RULE
1	IF G1(demam) AND G2(batuk) AND G3(pilek) THEN P1(Flu)
2	IF G1(demam) AND G4(ruam) AND G5(gatal) THEN P2Campak
3	IF G6(muntah) AND G7(diare) AND G8(lelah) THEN P3(Gastroenteritis)
4	IF G2(batuk) AND G9(sesak napas) AND G10(wheezing) THEN P4(Asma)
5	IF G1(demam) AND G11(iritabilitas) AND G12(hilang nafsu makan) THEN P5(Tipes)
6	IF G1(demam) AND G2(batuk) AND G13(sakit tenggorokan) THEN P6(Radang Tenggorokan)
7	IF G1(demam) AND G14(hidung tersumbat) AND G15(bersin) THEN P7(Rhinitis)
8	IF G1(demam) AND G13(sakit tenggorokan AND G4(ruam) THEN P8(Cacar Air)
9	IF G6(muntah) AND G1(demam) AND G12(hilang nafsu makan) THEN P9(DBD)