

Rekomendasi Makanan Sehat Berbasis AI Menggunakan metode *Forward Chaining* dan NLP Stanza

Fedrick^{1*}, Nadya Balqis Nasution², Gita Noviani Hutagalung³

^{1,2,3} Universitas Satya Terra Bhinneka, Indonesia

Alamat: Jl. Sunggal Gg. Bakul, Sunggal, Kec. Medan Sunggal, Kota Medan, Sumatera Utara 20128

*Korespondensi penulis: fedrickhuang@gmail.com

Abstract. Artificial Intelligence (AI) is currently increasingly used in the health sector, including to help determine food choices that are appropriate for a person's physical condition. In this study, we developed a healthy food recommendation system that can read the symptoms felt by users through text, then provide food suggestions based on those conditions. This system uses natural language processing technology with the Stanza library, and applies the Forward Chaining method to draw conclusions from the symptoms given. Uniquely, this system does not require direct personal data, only from the description of symptoms typed by the user. We tested this system using 10 types of diseases and 40 common symptoms, taken from a total of 400 disease and symptom data in the knowledge base. As a result, the system is able to provide accurate recommendations, with a success rate of between 90% and 100%. With this capability, the system is expected to be a practical solution in helping people make healthier decisions regarding their daily food and nutrition.

Keywords: Artificial Intelligence, Forward Chaining, Healthy Food Recommendations, Natural Language Processing, User Health.

Abstrak. Kecerdasan Buatan (AI) saat ini semakin banyak digunakan dalam bidang kesehatan, termasuk untuk membantu menentukan pilihan makanan yang sesuai dengan kondisi tubuh seseorang. Dalam penelitian ini, kami mengembangkan sistem rekomendasi makanan sehat yang bisa membaca gejala yang dirasakan pengguna melalui teks, lalu memberikan saran makanan berdasarkan kondisi tersebut. Sistem ini menggunakan teknologi pemrosesan bahasa alami (*Natural Language Processing*) dengan pustaka Stanza, serta menerapkan metode *Forward Chaining* untuk menarik kesimpulan dari gejala yang diberikan. Uniknya, sistem ini tidak membutuhkan data pribadi secara langsung cukup dari deskripsi gejala yang diketik pengguna. Kami menguji sistem ini menggunakan 10 jenis penyakit dan 40 gejala yang umum, diambil dari total 400 data penyakit dan gejala yang ada dalam basis pengetahuan. Hasilnya, sistem mampu memberikan rekomendasi yang akurat, dengan tingkat keberhasilan antara 90% hingga 100%. Dengan kemampuan ini, sistem diharapkan bisa menjadi solusi praktis dalam membantu masyarakat mengambil keputusan yang lebih sehat terkait makanan dan gizi harian mereka.

Kata Kunci : Kecerdasan Buatan, Rekomendasi Makanan Sehat, *Forward Chaining*, *Natural Language Processing*, Kesehatan Pengguna.

1. LATAR BELAKANG

Kecerdasan Buatan (*Artificial Intelligence*) adalah suatu cabang dari ilmu computer yang memungkinkan komputer untuk melaksanakan tugas-tugas yang biasanya dilakukan oleh manusia (Yansah, I. R., & Sumijan, S. (2021)). Perkembangan ilmu Pengetahuan dan teknologi terus mengalami kemajuan di berbagai sector, terutama dalam bidang kesehatan, dengan tujuan untuk meningkatkan kualitas pelayanan kesehatan. Tugas dokter menjadi jauh lebih berat agar dapat melayani seluruh pasien yang mengalami masalah kesehatan (Darsin, D., Kurniawan, A., Sesuna, M.F. (2022)). Pelayanan kesehatan di Indonesia perlu ditingkatkan, mengingat negara ini memiliki populasi yang sangat besar. Banyak rumah sakit di Indonesia tidak mampu menyediakan jumlah tenaga ahli kesehatan yang memadai, sehingga masalah kesehatan tetap

menjadi isu utama, mengingat terdapat banyak jenis penyakit yang bermunculan. (Putra, B. P., & Yunus, Y.(2021)). Sebagian besar penyakit tersebut seharusnya dapat dicegah dan diatasi dengan baik jika tenaga medis mampu melakukan diagnosis dengan cepat dan akurat. Penanganan yang cepat dan akurat akan berkontribusi dalam mengurangi angka kematian di Indonesia. Oleh karena itu, dalam konteks ini masyarakat memerlukan seorang ahli yang dapat membantu dalam mendiagnosa jenis penyakit dengan lebih cepat, sehingga masyarakat dapat melakukan tindakan pencegahan sejak dini. Untuk itu diperlukan sistem pakar dimana sistem pakar adalah suatu sistem yang berupaya untuk menerapkan pengetahuan manusia ke dalam computer sehingga computer dapat menyelesaikan masalah yang biasanya ditangani oleh para ahli. Sistem ini mengajukan pertanyaan mengenai fakta-fakta yang akan menggambarkan gejala dari penyakit tertentu dan mampu memberikan penjelasan berdasarkan hasil konsultasi yang telah dilakukan (Putra, B. P., & Yunus, Y.(2021)).

Salah satu metode yang umum digunakan dalam sistem pakar ini adalah *Forward Chaining*, yaitu pendekatan berpikir yang dimulai dari fakta sudah diketahui dan kemudian mencari aturan untuk mencapai kesimpulan. Metode ini sangat tepat untuk menganalisis masalah kesehatan dan memberikan rekomendasi makanan sesuai dengan gejala yang dialami (Yansah, I. R., & Sumijan, S. (2021)). Melalui sistem pakar, seseorang yang bukan ahlinya dapat memperoleh jawaban untuk menyelesaikan permasalahan dan mengambil keputusan yang biasanya dilakukan oleh seorang pakar (Darsin, D., Kurniawan,A., &Sesuna, M.F.(2022)). Selain itu, NLP (*Natural Language Processing*) juga membantu computer untuk memahami Bahasa manusia contohnya dengan mengenali kata-kata penting dalam komentar tentang kesehatan dan meningkatkan analisis sentimen menggunakan teknologi pembelajaran mendalam. Kami menerapkan kombinasi metode ini dalam sebuah situs website yang dapat mendiagnosis 400 jenis penyakit. Situs web ini juga dapat merekomendasikan makanan sehat yang sesuai bagi pengguna berdasarkan kondisi penyakit yang mereka alami. Dengan demikian, pengguna dapat mengatur pola mereka untuk menjadi lebih sehat dan penyakit dapat dicegah dan dikelola dengan baik.

Masalah gizi dan pola makan sehat telah menjadi perhatian utama dalam kesehatan masyarakat modern. Data Organisasi Kesehatan Dunia (WHO) menunjukkan bahwa lebih dari 1,9 miliar orang dewasa mengalami kelebihan berat badan, dengan 650 juta di antaranya mengalami obesitas. Di sisi lain, jutaan orang juga menderita kekurangan gizi dan malnutrisi. Kondisi ini semakin memprihatinkan dengan minimnya pengetahuan masyarakat tentang komposisi makanan sehat yang sesuai dengan kebutuhan individu (Husin, S. & Danang

,W,U.(2023)). Sistem Rekomendasi makanan konvensional sering kali terbatas dalam hal personalisasi dan akurasi. Sebagaimana dalam penelitian - penelitian sebelumnya yaitu

- Penelitian yang dilakukan oleh Yayang Eluis Bali Mawartika , dan Mohammad Guntur menjelaskan tentang “Aplikasi Sistem Pakar Pemilihan Makanan Berdasarkan Kebutuhan Gizi Menggunakan Metode *Forward Chaining*” Sistem pakar ini menyebutkan untuk mendiagnosis gizi buruk berdasarkan gejala-gejala yang diketahui. Hasil penelitian menunjukkan bahwa aplikasi sistem pakar menggunakan metode *forward chaining* dapat menentukan status gizi buruk pada balita dengan pengukuran antropometri berdasarkan berat badan per umur(BB/U) dan dapat mengetahui tipe gizi buruk tersebut berdasarkan gejala-gejala yang telah dimasukkan.
- Penelitian yang dilakukan oleh Andi Yulia Muniar dan Ashari mengenai “ Implementasi makanan pokok bagi penderita penyakit diabetes mellitus “ Sistem pakar ini menyebutkan bahwa pemilihan makanan pokok bagi penderita diabetes mellitus sangat membantu dokter dan masyarakat dalam menentukan makanan pokok yang sehat bagi penderita. Informasi yang dihasilkan dari aplikasi sistem pakar dalam pemilihan makanan pokok bagi penderita diabetes mellitus menggunakan metode *forward chaining* dapat berfungsi dengan baik sehingga mudah dimengerti dan dipahami oleh pengguna.
- Penelitian yang dilakukan oleh Bintang dan Maulana mengenai “Pengembangan sistem pakar untuk menentukan menu makanan bagi penderita darah rendah dengan metode *retinet*“ yang dimana sistem pakar ini memberikan rekomendasi makanan yang sesuai dengan penderita gula rendah. Sistem ini dapat mengidentifikasi dan menganalisis gambar makanan berdasarkan informasi nutrisi yang dibutuhkan oleh penderita darah rendah untuk meningkatkan kadar darah (Bintang, H, N, & Maulana, E, P. (2023))

Dari penelitian terkait ini masih diperlukan pengembangan sistem pakar untuk lebih fleksibel dengan menggunakan kombinasi metode *forward chaining* dan NLP karena meskipun Sistem berbasis aturan sederhana dengan metode *forward chaining* telah berhasil’ digunakan untuk memberikan rekomendasi menu makanan sehat .Namun pendekatan ini masih memiliki keterbatasan dalam memahami Bahasa alami dan konteks khusus pengguna. Perkembangan Kecerdasan Buatan (AI) dan pengolahan Bahasa alami (NLP) membuka peluang baru dalam pengembangan sistem rekomendasi makanan yang lebih cerdas. Teknologi NLP seperti *Stanza* memungkinkan ekstraksi informasi nutrisi dan preferensi makanan dari teks berbasis Bahasa alami. Sementara itu, metode *forward chaining* yang dikombinasikan

dengan teknologi *machine learning* dapat memberikan rekomendasi yang lebih personal dan dinamis.

2. KAJIAN TEORITIS

Kecerdasan Buatan (*Artificial Intelligence / AI*)

Semakin banyak dimanfaatkan dalam bidang kesehatan untuk meningkatkan akurasi, efisiensi, dan personalisasi layanan medis. Salah satu penerapannya adalah system pakar yang mampu melakukan diagnosis awal penyakit berdasarkan gejala yang dimasukkan oleh pengguna, dengan metode inferensi seperti *forward chaining*. Selain itu, AI juga digunakan dalam pemberian rekomendasi makanan sehat yang disesuaikan dengan kondisi kesehatan individu, seperti untuk penderita diabetes, hipertensi, atau gangguan pencernaan. Di sisi lain, teknologi *Natural Language Processing* (NLP), seperti *library Stanza*, memungkinkan system memahami input teks bebas dari pengguna sehingga mempermudah interaksi dalam proses konsultasi awal. Dengan berbagai penerapan ini, AI berperan penting dalam menciptakan sistem kesehatan yang lebih adaptif, *responsive*, dan mendukung pengambilan keputusan medis secara cepat dan akurat.

Sistem Pakar dan Metode Forward Chaining

Sistem pakar adalah aplikasi AI yang meniru kemampuan pengambilan keputusan seorang ahli dalam bidang tertentu. Metode *forward chaining* adalah teknik inferensi data- driven yang digunakan dalam sistem pakar. Proses ini dimulai dengan fakta yang tersedia dan menerapkan aturan inferensi untuk menghasilkan kesimpulan atau rekomendasi. Contohnya Dalam system pakar diagnosis penyakit, jika suatu pasien memberikan gejala-gejala yang dialami, sistem akan menggunakan aturan untuk menentukan kemungkinan penyakit yang diderita.

Pemrosesan Bahasa Alami (NLP) dengan *Stanza*

NLP *Stanza* yang dirancang untuk memproses teks dalam berbagai Bahasa. Dalam konteks kesehatan, *Stanza* dapat digunakan untuk menganalisis catatan medis digital dan memahami interaksi pasien dengan sistem kesehatan. Penerapan NLP seperti *Stanza* membantu dalam mengubah data teks tidak terstruktur menjadi informasi yang dapat digunakan untuk pengambilan keputusan yang tepat dan akurat.

Pendekatan Sistem dalam memberi Rekomendasi Makanan

Pendekatan sistem dalam memberikan rekomendasi makanan yang sehat berdasarkan kondisi kesehatan pengguna dapat dilakukan dengan menggabungkan metode *forward chaining* dan NLP. Sistem ini dapat bekerja sebagai berikut:

- Pengumpulan Data Pengguna: Menggunakan NLP untuk menganalisis input teks dari pengguna mengenai preferensi makanan, alergi, dan kondisi kesehatan.
- Penerapan aturan Pakar: Menggunakan metode *forward chaining* untuk menerapkan aturan berdasarkan data yang dikumpulkan, seperti “Jika pengguna memiliki diabetes, maka hindari makanan tinggi gula.”
- Rekomendasi Makanan: Sistem menghasilkan daftar makanan yang sesuai dengan kondisi kesehatan dan preferensi pengguna.

3. METODE PENELITIAN

Pengumpulan Data

Pengumpulan data sekunder dalam penelitian ini dilakukan dengan mengakses sumber-sumber informasi yang telah tersedia secara publik dan terpercaya melalui internet. Sumber data tersebut meliputi situs resmi lembaga pemerintah seperti ALODOKTER, HELLOSEHAT, dan *KAGGLE* menyediakan berbagai dataset terkait kesehatan dan Rekomendasi makanan sehat yang relevan dengan topik penelitian ini. Selain itu, studi literatur dari jurnal ilmiah dan artikel dari media massa yang kredibel juga digunakan untuk memperoleh informasi tambahan dan perspektif yang lebih luas mengenai pemanfaatan kecerdasan buatan dalam rekomendasi makanan sehat. Sumber-sumber ini dipilih berdasarkan kredibilitasnya untuk memastikan validitas dan reliabilitas data yang digunakan dalam penelitian ini.

Perancangan Sistem

- **Tahapan Analisis dan Perancangan sistem**

Pada tahapan ini yaitu analisis serta perancangan akan dijabarkan mengenai bagaimana sistem yang akan dibangun untuk kebutuhan sistem yang akan dibangun untuk kebutuhan sistem pakar berbentuk *website*, yang mana sistem pakar ini akan menggunakan *Google Colab Notebook* Berbasis bahasa pemrograman *Python*.

- **Analisa Sistem**

Setiap aturan dalam basis pengetahuan berisi kombinasi gejala tertentu yang mengarah pada suatu penyakit. Jika semua gejala dalam suatu aturan terpenuhi, maka

sistem akan menampilkan hasil diagnosis berupa jenis penyakit yang sesuai. Selain hasil diagnosis, sistem juga memberikan rekomendasi makanan sehat yang disesuaikan dengan penyakit yang terdeteksi. Dengan alur kerja ini, sistem mampu membantu proses identifikasi penyakit secara cepat dan informatif tanpa perlu konsultasi langsung dengan tenaga medis.

- **Basis Pengetahuan**

Basis pengetahuan dalam sistem ini berisi informasi mengenai relasi antara gejala dan penyakit yang digunakan untuk proses diagnosis. Sistem dibangun menggunakan bahasa pemrograman *Python* melalui platform *Google Colab Notebook*. Data gejala dan penyakit diperoleh dari berbagai sumber referensi kesehatan terpercaya di internet, kemudian diolah dan disusun dalam bentuk aturan berbasis gejala yang digunakan dalam proses inferensi. Secara keseluruhan, terdapat sekitar 400 jenis penyakit dan ratusan gejala dalam data asli yang telah di kumpulkan. Namun, untuk keperluan pengujian dan pembahasan dalam jurnal ini, hanya digunakan 10 penyakit umum yang sering dialami manusia, dengan total 40 gejala yang relevan. Pengetahuan ini disusun dalam bentuk aturan-aturan yang menjadi dasar dalam penalaran sistem pakar untuk memberikan diagnosis penyakit berdasarkan gejala yang dipilih pengguna

4. HASIL DAN PEMBAHASAN

Analisis Data Penyakit dan Gejala

Data yang digunakan dalam penelitian ini diperoleh melalui studi literatur dengan mengumpulkan informasi dari berbagai jurnal ilmiah dan sumber *website* terpercaya yang relevan dengan topik. Pengumpulan ini dilakukan dengan analisis berdasarkan relevansi dan kredibilitas sumber untuk memastikan validitas dan keakuratan informasi yang diperoleh. Dengan demikian, data yang terkumpul dari jurnal dan *website* memberikan landasan yang kuat dan komprehensif untuk mendukung hasil dan kesimpulan. Data penyakit dan data gejala dapat kita lihat di table 1 dan 2

Tabel 1. Daftar Penyakit

NO	KODE PENYAKIT	JENIS PENYAKIT
1	P1	Pneumonia
2	P2	Asma
3	P3	Usus Buntu
4	P4	Thalassemia
5	P5	Bisul
6	P6	Folikulitis
7	P7	DBD
8	P8	Chikungunya
9	P9	GERD
10	P10	ISP (Infeksi Saluran Pencernaan)

Tabel 2. Daftar Gejala

No	Kode Gejala	Gejala
1	G1	Batuk
2	G2	Sesak Nafas
3	G3	Nyeri di dada
4	G4	Mudah Lelah
5	G5	Demam
6	G6	Nafsu Makan menurun
7	G7	Muntah
8	G8	Mual
9	G9	Muncul Benjolan Merah
10	G10	Bengkak
11	G11	Rasa Nyeri
12	G12	Keluar Nanah
13	G13	Gatal
14	G14	Jerawat kecil merah

15	G15	Nyeri Pada Tulang
16	G16	Jerawat kecil putih
17	G17	Lemas
18	G18	Sakit kepala
19	G19	Sakit otot dan sendi
20	G20	Nyeri di belakang mata
21	G21	Suara serak
22	G22	Sakit saat Menelan
23	G23	Tenggorokan sakit
24	G24	Perut kembung
25	G25	Nyeri diatas perut
26	G26	Mulut pahit
27	G27	Bau mulut
28	G28	Kram Perut
29	G29	Nyeri Otot
30	G30	Ruam pada kulit
31	G31	Sakit perut
32	G32	Sendi bengkak
33	G33	Nyeri di ulu hati
34	G34	Bersin
35	G35	Nyeri saat Berjalan
36	G36	Nyeri di perut
37	G37	Kulit Pucat
38	G38	Sulit berkonsentrasi
39	G39	Mudah marah
40	G40	Jantung berdebar cepat

Pengujian metode *forward chaining*

Berdasarkan data jenis gejala yang didapatkan dari pakar selanjutnya dilakukan penyusunan rule atau aturan.

Rule 1:

IF Batuk is True AND Demam is True AND Nyeri di dada is True AND Nafsu akan menurun is True AND Mudah lelah THEN Penyakit Pneumonia

Rule 2:

IF Batuk is True AND Nyeri di dada is True AND Sesak nafas is True AND Mudah lelah THEN Penyakit Asma

Rule 3:

IF Nyeri di perut is True AND Nyeri di ulu hati is True AND Batuk is True AND Bersin is True AND Mual is True AND Muntah THEN Penyakit Usus Buntu

Rule 4:

IF Mudah lelah is True AND Kulit pucat is True AND Sakit kepala is True AND Nafsu makan menurun is True AND Sulit berkonsentrasi is True AND Mudah marah THEN Penyakit Thalassemia

Rule 5:

IF Muncul benjolan merah is True AND Keluar nanah is True AND Bengkak is True AND Rasa nyeri THEN Penyakit Bisul

Rule 6:

IF Gatal is True AND Jerawat kecil putih is True AND Jerawat kecil merah is True AND Ruam pada kulit THEN Penyakit Folikulitis

Rule 7:

IF Sakit kepala is True AND Lemas is True AND Nyeri di belakang mata is True AND Sakit otot dan sendi is True AND Nafsu makan menurun is True AND mual is True AND Muntah THEN Penyakit DBD

Rule 8:

IF Demam is True AND Nyeri Pada Tulang is True AND Mual is True AND Lemas is True AND Sakit Kepala is True AND Nyeri otot AND Ruam pada Kulit is True AND Sendi bengkak THEN Penyakit Chikugunya

Rule 9:

IF Mulut pahit is True AND Mual is True AND Muntah is True AND Bau Mulut is True AND Nyeri Dada is True AND Sakit saat Menelan is True AND Tenggorokan Sakit is True AND Perut Kembang THEN Penyakit GERD

Rule 10:

IF Mual is True AND Muntah is True AND Sakit perut is True AND Kram perut is True AND Nafsu makan menurun THEN Penyakit ISP (Infeksi Saluran Pencernaan)

Setelah dilakukan pengujian forward chaining (*Rule Based*) hasilnya dapat kita lihat pada Tabel 3.

Tabel 3. Aturan

NO	GEJALA	PENYAKIT
1	G1, G5, G3, G6, G4	P1
2	G1, G3, G2, G4	P2
3	G36, G33 G1, G34, G8, G7	P3
4	G4, G37, G18, G6, G38, G39	P4
5	G9, G12, G10, G11	P5
6	G13, G16, G14, G15	P6
7	G18, G17, G20, G19, G6, G8, G7	P7
8	G5, G15, G8, G18, G29, G30, G32, G17	P8
9	G25, G8, G7, G26, G28, G27, G3, G24,	P9
10	G8, G7, G31, G29, G6	P10

Penerapan Metode NLP (*Natural Language Processing*)

Contoh penerapan dari NPL Stanza: Jika Pengguna mengetik: "Saya merasa mual, muntah, dan nyeri di perut sejak kemarin malam."

Maka NLP Stanza akan memproses Bahasa sesuai kalimat yang dimasukkan :

- mual, muntah, nyeri, dan perut terdeteksi sebagai token dengan POS (*Part of Speech*) yang relevan.
- Token ini dapat dimasukkan ke dalam sistem pakar berbasis forward chaining untuk mencocokkan dengan gejala penyakit.

Contoh Pemanfaatan: Jika nyeri, mual, perut, dan muntah muncul, system dapat mengaitkannya dengan penyakit seperti: Usus Buntu dan Magh

Evaluasi dan Hasil

Setelah melakukan pengujian aturan (*rule*) maka didapatkan hasil rekap seperti pada table 4.

Tabel 4. Rekap Diagnosa Sistem

No	Gejala	Diagnosis	Persentase	Rekomendasi Makanan Sehat
1	Batuk, Demam, Nyeri di dada, Nafsu makan menurun, Mudah lelah	Pneumonia	100%	Jeruk, Beras merah, Jahe, Madu, Ikan salmon, Yoghurt, Brokoli, Bawang putih
2	Batuk, Nyeri di dada, Sesak nafas, Mudah lelah	Asma	100%	Ikan salmon, Apel, Pisang, Jeruk, Bayam, Wortel, Brokoli, Ubi jalar
3	Nyeri di perut, Nyeri di ulu hati, Batuk, Bersin, Mual, Muntah	Usus Buntu	100%	Bubur, Nasi putih, Sup wortel, Yoghurt, Roti putih, Melon, Pisang, Kentang rebus, Pepaya, Ikan salmon
4	Mudah lelah, Kulit pucat, Sakit kepala, Nafsu makan menurun, Sulit berkonsentrasi, Mudah marah	Thalassemia	100%	Bayam, Brokoli, Alpukat, Minyak Sayur, Susu, Sereal, telur, Bawang putih, Teh Hijau, Kubis, Tembakau, Yogurt, dan keju
5	Muncul benjolan merah, Keluar nanah, Bengkak, Rasa nyeri	Bisul	100%	Stroberi, Kiwi, Pepaya, Ikan tuna, Telur, Dada ayam, Kacang kenari, Ikan sarden, Jeruk,
6	Gatal, Jerawat kecil putih, Jerawat kecil merah, Ruam pada kulit	Folikulitis	100%	Apel, Blueberry, Nanas, Ikan salmon, Ikan tuna, Ikan makarel, Yoghurt.

7	Sakit kepala, Lemas, Nyeri dibelakang mata, Sakit otot dan sendi, Nafsu makan menurun, Mual, Muntah	DBD	100%	Bubur, Jahe, Bawang putih, Oatmeal, Yoghurt, Jambu biji, Buah delima, Kurma, Hati ayam/ sapi
8	Demam, nyeri tulang, lemas, mual, sakit kepala, nyeri otot, ruam pada kulit, sendi bengkak.	Chikugunya	100%	Nasi putih, Jeruk, Pepaya, Anggur, Jeruk nipis, Wortel, Sup ayam, Telur, Susu, Kacang almond, Jambu biji, Buah naga
9	Mulut pahit, mual, muntah, bau mulut, nyeri dada, sakit saat menelan, tenggorokan sakit, perut kembung.	GERD	90%	Bubur, Pisang, Kacang hijau, Brokoli, Asparagus, Kembang kol, Kentang, Mentimun, Oatmeal, Pisang, Apel, Pepaya, Alpukat, Daging ayam tanpa lemak
10	Mual, Muntah, Sakit perut, Kram perut, Nafsu makan menurun	ISP (infeksi saluran pencernaan)	100%	Kentang rebus, Pisang, Sup kaldu, Telur, Nasi merah, Ubi jalar, Oatmeal, Jagung

Tahapan Pembuatan Codingan dalam Sistem

- Melakukan instalasi pustaka Stanza



Gambar 1 Instalasi Pustaka Stanza

- OS: Digunakan untuk menangani direktori dan file, terutama untuk cek keberadaan model stanza
 - Pandas as pd: Digunakan untuk membaca dan memproses file CSV (data gejala dan rekomendasi makanan)
 - *Get_close_matches*: Fungsi dari difflib untuk menemukan string yang mirip (digunakan saat pengguna salah ketik gejala/makanan)
 - Stanza: Library NLP dari *Stanford* untuk tokenisasi, POS *tagging*, *lemmatization*, dll. digunakan di sini agar bisa memproses bahasa alami jika dibutuhkan.

```
import os
import pandas as pd
from difflib import get_close_matches
import stanza
```

Gambar 2 Import Library

- Fungsi ini mengecek apakah model *stanza* untuk bahasa Indonesia (lang-‘id’) sudah di download, jika belum ada, model akan di unduh

```
def ensure_stanza_model(lang='id'):
    model_dir = os.path.expanduser(f"~/stanza_resources/{lang}")
    if not os.path.exists(model_dir):
        print(f"Model bahasa '{lang}' belum ditemukan, mulai download...")
        stanza.download(lang)
    else:
        print(f"Model bahasa '{lang}' sudah ada, tidak perlu download ulang.")
|
ensure_stanza_model('id')
```

Gambar 3 Pengecek Stanza

- Setelah model tersedia, stanza.Pipeline akan dijalankan dengan proses:
- Tokenize: Memisahkan kata dalam kalimat.
- Pos: Part-of-speech (kata benda, kerja, dll).
- Lemma: Mengubah kata ke bentuk dasarnya (misal:”memakan” jadi “makan”).

```
nlp = stanza.Pipeline(lang='id', processors='tokenize,pos,lemma')
```

Gambar 4 Tokenisasi Stanza

- Mengambil data CSV dari Google Drive:
- link gejala: Daftar penyakit dan gejalanya- link rekomendasi: Daftar penyakit dan makanan yang dianjurkan. Menghapus kolom No jika ada,karena tidak dibutuhkan

```
link_gejala = "https://drive.google.com/uc?export=download&id=12X1QpHj-xN7fMu7tHhLsIqSV27rwFQhC"
link_rekomendasi = "https://drive.google.com/uc?export=download&id=1_FcdKJL07iC1e5w982mL180ZyrWtI-My"

df_gejala = pd.read_csv(link_gejala)
df_rekomendasi = pd.read_csv(link_rekomendasi)

df_gejala = df_gejala.drop(columns=['No'], errors='ignore')
df_rekomendasi = df_rekomendasi.drop(columns=['No'], errors='ignore')
```

Gambar 5 Pembacaan dan Pengambilan Data

- Menyatukan semua kolom Gejala1, Gejala2,... menjadi satu kolom (pakai.melt())
- Semua nilai diubah ke huruf kapital agar seragam
- Kolom “penyakit” dibersihkan dari karakter aneh (menggunakan regex)

- Output: rule_gejala= Dictionary {penyakit: [daftar gejala]}
- df_gejala_melted= Bentuk panjang (long format) dari data gejala

```
def preprocess_gejala(df):
    gejala_cols = [col for col in df.columns if col.startswith('Gejala')]
    df_melted = df.melt(id_vars=["Penyakit"], value_vars=gejala_cols, var_name="Gejala", value_name="Gejala_Values")
    df_melted = df_melted.dropna(subset=['Gejala_Values'])
    df_melted['Gejala_Values'] = df_melted['Gejala_Values'].str.upper().str.strip()
    df_melted['Penyakit'] = df_melted['Penyakit'].str.upper().str.strip().str.replace(r'\W+', '', regex=True)
    rule = df_melted.groupby("Penyakit")['Gejala_Values'].apply(list).to_dict()
    return rule, df_melted

rule_gejala, df_gejala_melted = preprocess_gejala(df_gejala)
```

Gambar 6 Preprocessing Data Gejala

- Proses serupa: kolom makanan1, makanan2, ... diubah menjadi satu kolom
- Dibuat dict {penyakit: [daftar makanan yang disarankan]}

```
def preprocess_rekomendasi(df):
    makanan_cols = [col for col in df.columns if col.startswith('Makanan')]
    df_melted = df.melt(
        id_vars=["Penyakit"],
        value_vars=makanan_cols,
        var_name="Makanan",
        value_name="Makanan_Values"
    ).dropna(subset=['Makanan_Values'])
    df_melted['Makanan_Values_Original'] = df_melted['Makanan_Values'].str.strip()
    df_melted['Penyakit'] = df_melted['Penyakit'].str.upper().str.strip().str.replace(r'\W+', '', regex=True)
    rekomendasi = df_melted.groupby("Penyakit")['Makanan_Values_Original'].apply(list).to_dict()
    return rekomendasi, df_melted

rekomendasi_makanan, df_rekomendasi_melted = preprocess_rekomendasi(df_rekomendasi)
```

Gambar 7 Preprocessing Data Rekomendasi Makanan

- Kumpulan semua gejala dan makanan yang akan digunakan untuk validasi input dari pengguna

```
all_gejala = list({g for gejalas in rule_gejala.values() for g in gejalas})
all_makanan = list({m.upper() for rekom in rekomendasi_makanan.values() for m in rekom})
```

Gambar 8 Validasi Input Pengguna

- Jika pengguna salah ketik, misalnya Diare ditulis Diaree, maka akan dicari kemungkinan paling mirip dan dikonfirmasi. Menyimpan input gejala/alergi ke dalam list

```
def konfirmasi_input(possible_match):
    while True:
        print(f"Apakah maksud Anda '{possible_match}'? (Ya/Tidak)")
        jawaban = input().strip().lower()
        if jawaban in ['ya', 'tidak']:
            return jawaban == 'ya'
```

Gambar 9 Kemungkinan Paling Dekat Logic NLP

- Jika pengguna salah ketik, misalnya Diare ditulis Diaree, maka akan dicari kemungkinan paling mirip dan dikonfirmasi.

```

def input_list(prompt, valid_list):
    hasil = []
    print(prompt)
    print("Ketik 'selesai' jika sudah selesai.")
    while True:
        entry = input("> ").strip().upper()
        if entry == 'SELESAI':
            break
        if entry in valid_list:
            hasil.append(entry)
            print(f"✅ '{entry}' ditambahkan.")
        else:
            close_match = get_close_matches(entry, valid_list, n=1, cutoff=0.6)
            if close_match and konfirmasi_input(close_match[0]):
                hasil.append(close_match[0])
                print(f"✅ '{close_match[0]}' ditambahkan.")
            else:
                print(f"Tidak ditemukan kecocokan untuk '{entry}'. Silakan coba lagi atau ketik 'selesai' untuk selesai.")
    return hasil

```

Gambar 10 Formulir Inputan

- Menampilkan daftar makanan yang dianjurkan untuk penyakit tertentu
- Jika makanan ada dalam alergi pengguna, akan diberi peringatan

```

def tampilkan_rekomendasi(penyakit, alergi, rekomendasi_makanan):
    rekom_list = rekomendasi_makanan.get(penyakit, [])
    if not rekom_list:
        print(f"Tidak ada data rekomendasi untuk penyakit {penyakit}.")
        return
    for makanan in rekom_list:
        if makanan.upper() in alergi:
            print(f"⚠️ Hindari: {makanan} (karena alergi)")
        else:
            print(f"✅ {makanan}")

```

Gambar 11 Tampilkan Daftar Makanan

- Fungsi main() memulai percakapan dan meminta data dasar pengguna (nama, jenis kelamin, usia, berat badan) Validasi dilakukan agar input benar (angka untuk usia dan berat, serta pilihan L/P) Pengguna juga ditanya tentang alergi dan gejala yang dialami.

```

def main():
    print("AI: Halo! Saya akan bantu rekomendasi makanan sehat berdasarkan kondisi Anda.\n")

    nama = input("Nama: ").strip().capitalize()
    while True:
        jenis_kelamin = input("Jenis Kelamin (L/P): ").strip().upper()
        if jenis_kelamin in ["L", "P"]:
            break
        print("Input tidak valid. Harap masukkan 'L' untuk Laki-Laki atau 'P' untuk Perempuan.")
    while True:
        try:
            usia = int(input("Usia: ").strip())
            berat_badan = int(input("Berat badan (kg): ").strip())
            break
        except ValueError:
            print("Usia dan berat badan harus berupa angka. Silakan coba lagi.")

    alergi = input_list("AI: Sebutkan alergi Anda satu per satu.", all_makanan)
    keluhan = input_list("\nAI: Sebutkan gejala yang Anda alami satu per satu.", all_gejala)

```

Gambar 12 Inputan Data Pengguna

- Gejala yang disebutkan dikirim ke fungsi `forward_chaining()` untuk dilakukan proses diagnosa penyakit.

```
hasil_analisis = forward_chaining(keluhan, rule_gejala)
```

Gambar 13 Hasil Analisis Logic Forward Chaining

- Jika tidak ada gejala, sistem memberi rekomendasi makanan secara umum. Jika ada
- hasil analisis, maka sistem mencari penyakit dengan persentase kecocokan tertinggi, lalu menampilkan:
- Nama penyakit
- Persentase kecocokan
- Gejala yang cocok
- Rekomendasi makanan untuk penyakit tersebut Jika tidak cocok dengan penyakit manapun, maka sistem tetap memberi rekomendasi makanan umum.

```
print("\n--- HASIL ANALISIS ---")
if not keluhan:
    print("Anda tidak menyebutkan gejala apa pun. Berikut rekomendasi makanan sehat secara umum:")
    for penyakit in rekomendasi_makanan:
        print(f"\nUntuk penyakit {penyakit}:")
        tampilkan_rekomendasi(penyakit, alergi, rekomendasi_makanan)
else:
    if hasil_analisis:
        penyakit_tertinggi = max(hasil_analisis, key=lambda x: hasil_analisis[x]["match"])
        persen_tertinggi = hasil_analisis[penyakit_tertinggi]["match"]
        gejala_terpenuhi = hasil_analisis[penyakit_tertinggi]["gejala_terpenuhi"]

        print(f"\nKemungkinan Anda mengalami: {penyakit_tertinggi} ({persen_tertinggi:.2f}% cocok)")
        print(f"Gejala yang terpenuhi: {' '.join(gejala_terpenuhi)}")

        print("\nRekomendasi makanan sehat untuk penyakit ini:")
        tampilkan_rekomendasi(penyakit_tertinggi, alergi, rekomendasi_makanan)
    else:
        print("Tidak ada penyakit yang terdeteksi berdasarkan gejala Anda. Berikut rekomendasi makanan umum:")
        for penyakit in rekomendasi_makanan:
            print(f"\nUntuk penyakit {penyakit}:")
            tampilkan_rekomendasi(penyakit, alergi, rekomendasi_makanan)
```

Gambar 14 Hasil Analisis

- Menyusun informasi pengguna kembali sebagai ringkasan jenis kelamin.

```
jenis_kelamin_full = "Laki-Laki" if jenis_kelamin == "L" else "Perempuan"
```

Gambar 15 Koreksi Inputan Jenis Kelamin

- Menyusun informasi pengguna kembali sebagai ringkasan: nama, jenis kelamin, usia, berat badan, alergi, dan gejala

```
print("\n--- RINGKASAN PROFIL ---")
print(f"👤 Nama: {nama} | Jenis Kelamin: {jenis_kelamin_full} | Usia: {usia} | Berat Badan: {berat_badan} kg")
print(f"🚫 Alergi: {' '.join(alergi) if alergi else 'Tidak ada'}")
print(f"📋 Gejala yang disebutkan: {' '.join(keluhan) if keluhan else 'Tidak ada'}")
```

Gambar 16 Ringkasan Profil Pengguna

- Baris ini menandakan bahwa ketika file *Python* dijalankan secara langsung, makan fungsi `main()` akan dieksekusi

```
if __name__ == "__main__":
    main()
```

Gambar 17 Fungsi AI Berjalan

- Menggunakan *Stanza*, sebuah pustaka NLP dari *Stanford*, untuk memproses input dalam Bahasa Indonesia.
- Modul yang diunduh dan digunakan: *tokenize*, *mwt*, *pos*, *lemma* – yaitu *tokenizer*, *multi-word tokenizer*, *part-of-speech tagger*, dan *lemmatizer*.
- Semua proses ini bertujuan untuk mempersiapkan teks yang diketik pengguna agar bisa dipahami secara semantik.

```
INFO:stanza:Checking for updates to resources.json in case models have been updated. Note: this behavior can be turned off with download_method=None or downl
Model bahasa 'id' sudah ada, tidak perlu download ulang.
Downloading https://raw.githubusercontent.com/stanfordnlp/stanza-resources/main/resources_1.10.0.json: ██████████ 432k/? [00:00<00:00, 18.0MB/s]
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
WARNING:stanza:Language id package default expects mwt, which has been added
INFO:stanza:Loading these models for language: id (Indonesian):
-----
| Processor | Package |
-----
| tokenize  | gsd     |
| mwt       | gsd     |
| pos       | gsd_charlm |
| lemma     | gsd_nocharlm |
-----
INFO:stanza:Using device: cpu
INFO:stanza:Loading: tokenize
INFO:stanza:Loading: mwt
INFO:stanza:Loading: pos
INFO:stanza:Loading: lemma
INFO:stanza:Done loading processors!
AI: Halo! Saya akan bantu rekomendasi makanan sehat berdasarkan kondisi Anda.
```

Gambar 18 Download Library

- Sistem menerima input data pribadi dari pengguna: nama, jenis kelamin, usia, berat badan. Digunakan untuk mencetak ringkasan profil dan kemungkinan relevansi dalam rekomendasi makanan. Pengguna tidak memiliki alergi makanan, sehingga input 'SELESAI' menunjukkan tidak ada alergi.

```
Nama: nadya
Jenis Kelamin (L/P): P
Usia: 20
Berat badan (kg): 50
AI: Sebutkan alergi Anda satu per satu.
Ketik 'selesai' jika sudah selesai.
> SELESAI
```

Gambar 19 Penginputan Data Diri Dan Alergi

- Sistem menerima input data pribadi dari pengguna: nama, jenis kelamin, usia, berat badan.
- Digunakan untuk mencetak ringkasan profil dan kemungkinan relevansi dalam rekomendasi makanan.
- Pengguna tidak memiliki alergi makanan, sehingga input 'SELESAI'

```
AI: Sebutkan gejala yang Anda alami satu per satu.
Ketik 'selesai' jika sudah selesai.
> batuk
  ✓ 'BATUK' ditambahkan.
> demam
  ✓ 'DEMAM' ditambahkan.
> kesulitan bernafas
  ✓ 'KESULITAN BERNAFAS' ditambahkan.
> nyeri di dada
  ✓ 'NYERI DI DADA' ditambahkan.
> gampang lelah
  ✓ 'GAMPANG LELAH' ditambahkan.
> nafsu makan menurun
  ✓ 'NAFSU MAKAN MENURUN' ditambahkan.
> muntah
  ✓ 'MUNTAH' ditambahkan.
> mual
  ✓ 'MUAL' ditambahkan.
> diare
  ✓ 'DIARE' ditambahkan.
> kurang konsentrasi
  ✓ 'KURANG KONSENTRASI' ditambahkan.
> selesai
```

Gambar 20 Penginputan Gejala Pada AI

- Sistem menggunakan metode *forward chaining* untuk mencocokkan gejala.
- Hasil analisis menunjukkan penyakit *PNEUMONIA* karena seluruh gejala cocok 100% dengan data penyakit dalam sistem.
- Rekomendasi ini disesuaikan dengan penyakit dan tanpa alergi pengguna.

```
--- HASIL ANALISIS ---
Kemungkinan Anda mengalami: PNEUMONIA (100.00% cocok)
Gejala yang terpenuhi: MUNTAH, MUAL, KURANG KONSENTRASI, BATUK, GAMPANG LELAH, NAFSU MAKAN MENURUN, NYERI DI DADA, DEMAM, KESULITAN BERNAFAS, DIARE

Rekomendasi makanan sehat untuk penyakit ini:
  ✓ Jeruk
  ✓ Beras merah
  ✓ Jahe
  ✓ Madu
  ✓ Ikan salmon
  ✓ Yogurt
  ✓ Brokoli
  ✓ Bawang putih

--- RINGKASAN PROFIL ---
👤 Nama: Nadya | 🧑 Jenis Kelamin: Perempuan | 🎂 Usia: 20 | ⚖️ Berat Badan: 50 kg
🚫 Alergi: Tidak ada
📋 Gejala yang disebutkan: BATUK, DEMAM, KESULITAN BERNAFAS, NYERI DI DADA, GAMPANG LELAH, NAFSU MAKAN MENURUN, MUNTAH, MUAL, DIARE, KURANG KONSENTRASI
```

Gambar 21 Hasil Dari Analisis / Output AI

5. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan sebuah sistem pakar berbasis *Python* yang dibangun menggunakan platform *Google Colab Notebook* untuk melakukan diagnosis penyakit pengguna berdasarkan gejala yang dirasakan oleh pengguna, serta dilengkapi dengan fitur rekomendasi makanan sehat yang disesuaikan dengan jenis penyakit yang terdeteksi. Penerapan teknologi *Natural Language Processing* (NLP) dengan library *Stanza* memungkinkan sistem memahami dan memproses input teks dari pengguna secara lebih fleksibel. Dari hasil pengembangan dan pengujian sistem, dapat disimpulkan bahwa sistem pakar berbasis *forward chaining* dapat digunakan secara efektif untuk mendiagnosis penyakit berdasarkan gejala yang dimasukkan pengguna, dan fitur rekomendasi makanan sehat yang terintegrasi dalam sistem memberikan nilai tambah dalam edukasi kesehatan pengguna. Selain

itu, penerapan NLP Stanza berhasil mempermudah pengguna dalam memasukkan gejala melalui teks bebas, sehingga sistem menjadi lebih interaktif dan fleksibel. Sistem menunjukkan performa yang baik dari segi akurasi diagnosis, kecepatan respon, dan kemudahan penggunaan, berdasarkan pengujian terhadap data uji yang disediakan. Untuk pengembangan lebih lanjut, sistem ini dapat ditambahkan dengan jenis penyakit dan gejala yang lebih beragam agar cakupan diagnosa dan rekomendasi makanan menjadi lebih luas dan lebih akurat. Penyempurnaan basis pengetahuan juga perlu dilakukan agar sistem dapat menangani gejala yang lebih kompleks atau kombinasi gejala yang mirip antar penyakit. Rekomendasi makanan sehat dapat ditingkatkan dengan mempertimbangkan faktor tambahan seperti usia, jenis kelamin, atau riwayat kesehatan pengguna, dan penggunaan NLP *Stanza* dapat terus dikembangkan untuk mengenali lebih banyak variasi kalimat dan istilah yang digunakan masyarakat dalam menggambarkan gejala.

DAFTAR REFERENSI

- Aprilia, S. A. ((2024)). SISTEM pakar rekomendasi obat berdasarkan gejala penyakit menular umum di masyarakat menggunakan metode forward chaining. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(2).
- Budiyati, E. &. ((2022)). Penerapan Metode Forward Chaining Pada Aplikasi Daring Untuk Mendeteksi Penyakit Anemia. *J. Media Inform. Budidarma*, 6(3), 1667.
- Darsin, D. K. ((2022)). Sistem pakar untuk mendiagnosa penyakit gigi dan mulut menggunakan metode forward chaining (Studi di RSUD Menggala). *J. Sist. Inf. dan Sains Teknol*, 4(2), 1-10.
- Hakim, M. ((2020)). Sistem Pakar Mengidentifikasi Penyakit Alat Reproduksi Manusia Menggunakan Metode Forward Chaining. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 1(1), 59-67.
- Hasugian, A. H. ((2025)). Sistem Pakar Menentukan Menu Makanan Sehat Berdasarkan Penyakit Menggunakan Metode Certainty Factor Dan Forward Chaining Berbasis Website. *Techno Xplore: Jurnal Ilmu Komputer dan Teknologi Informasi*, 10(1), 29-36.
- Kalua, A. L. ((2023)). Sistem Pakar Diagnosa Penyakit Malaria dengan Certainty Factor dan Forward Chaining. *Journal of Information Technology, Software Engineering and Computer Science*, 1(1), 22-33.
- Mukhroji, M. &. ((2022)). Perancangan Aplikasi Sistem Pakar Doagnosa Awal Penyakit Batu Ginjal Berbasis Web Dengan Menggunakan Metode Forward Chaining. *Jurnal Ilmiah Informatika*, 10(01), 15-19.
- Putra, B. D. ((2022)). Aplikasi Sistem Pakar Diagnosa Penyakit Stunting Pada Anak Dengan Metode Forward Chaining. *Jurnal Pustaka Paket*, 1(1), 6-15.

- Putra, B. P. ((2021)). Sistem Pakar dalam Mendiagnosis Penyakit Mata dengan Menggunakan Metode Forward Chaining. *Jurnal Informasi Dan Teknologi*, 128-133.
- Rahmi, F. A. ((2025)). Pengaruh Penerapan Pola Makan Sehat Terhadap Tingkat Kesehatan pada Mahasiswa Kedokteran. *Scientific Journal*, 4(1), 41-45.
- Ramadhani, T. F. ((2020)). Sistem pakar diagnosa penyakit ISPA berbasis web dengan metode forward chaining. *JOINTECS (Journal Inf. Technol. Comput. Sci., vol. 5, no. 2, p. 81, 2020, doi: 10.31328/jointecs. v5i2. 1243.*
- Sari, H. A. ((2025)). Sistem Pakar Diagnosa Dini Penyakit pada Lambung dengan Metode Forward Chaining di Klinik dr. Yolanda. *In Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK) (Vol. 9, No. 1, pp, 496-502.*
- Sawitri, E. S. ((2024)). PENERAPAN METODE FORWARDCHAINING DIAGNOSA PENYAKIT INFEKSI SALURAN KEMIH BERBASIS WEB. *Jurnal Review Pendidikan dan Pengajaran (JRPP)*, 7(1), 2248-2253.
- Sembiring, A. &. ((2021)). Sistem pakar berbasis mobile untuk diagnosis penyakit ginjal menggunakan metode forward chaining. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 6(1), 139-148.
- Septiana, Y. W. ((2024)). SISTEM PAKAR PENENTUAN JUMLAH KALORI PROGRAM DIET BERBASIS WEB MENGGUNAKAN METODE FORWARD CHAINING (Studi kasus: RSA UGM Yogyakarta). *Indonesian Journal of Business Intelligence (IJUBI)*, 7(1), 47-55.
- Sufi, H. U. ((2023)). Sistem pakar rekomendasi menu makanan sehat penderita penyakit dengan metode forward chaining. *Jurnal KomtekInfo*, 8-14.
- Utami, Y. P. ((2021)). Sistem pakar deteksi penyakit diabetes mellitus (dm) menggunakan metode forward chaining dan certainty factor berbasis android. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5(1), 49-55.
- Yansyah, I. R. ((2021)). Sistem Pakar Metode Forward Chaining untuk Mengukur Keparahan Penyakit Gigi dan Mulut. *Jurnal Sistim Informasi dan Teknologi*, 41-47.
- Maksum, M. E. P. ((2023)). Pengembangan Sistem Pakar Untuk Menentukan Menu Makanan bagi Penderita Darah rendah dengan Metode *RetinaNet*. *Journal of Research and Publication Innovation*, 1(3), 590 - 593