

Sistem Pakar

Deskripsi Modul

Mata Kuliah	Kecerdasan Buatan
Kode Mata Kuliah / SKS	RPL3404 / 3
Semester	4
Kelas	TRPL 2A, 2B, 2D
Capaian Pembelajaran	
Deskripsi Singkat Mata Kuliah	Mata kuliah ini menjelaskan metode untuk membuat perangkat lunak menjadi lebih pintar, lebih menalar dan lebih berguna ntuk membantu manusia menyelesaikan permasalahan menggunakan teknik kecerdasan buatan, yang meliputi pencarian/teknik-teknik pelacakan, representasi pengetahuan dan logika, algoritma genetika, sistem pakar serta komputasi lunak (soft computing) yaitu neural network.
Bahan Kajian Modul	Pengetahuan dan Penalaran
Bentuk dan Metode Pembelajaran	
Waktu pembelajaran	120 Menit
Rekomendasi buku teks:	<ol style="list-style-type: none">1. S. Russell dan P. Norvig, Artificial Intelligence: A Modern Approach, 4 penyunt., US: Pearson Education, 20212. Suyanto, Artificial Intelligence : Searching - Reasoning - Planning - Learning, 2 penyunt., Bandung: Informatika, 2021
Petunjuk khusus	-

1. Materi Pembelajaran

Sistem pakar merupakan bagian penting dari kecerdasan buatan (AI) yang menirukan kemampuan pengambilan keputusan seorang pakar manusia. Sistem ini menggunakan basis pengetahuan yang berisi informasi dan aturan khusus domain untuk menafsirkan dan memecahkan masalah yang rumit. Misalnya, sistem pakar medis dapat menganalisis gejala pasien dan menyarankan kemungkinan diagnosis atau perawatan. Demikian pula, sistem pakar keuangan dapat mengevaluasi tren pasar dan merekomendasikan strategi investasi.

1.1 Pengenalan Sistem Pakar

Konsep dasar sistem pakar mencakup beberapa persoalan mendasar, antara lain apa yang dimaksud dengan keahlian, siapa yang disebut pakar, bagaimana keahlian dapat ditransfer, dan bagaimana sistem bekerja. Kepakaran (expertise) adalah pengetahuan yang ekstensif dan spesifik yang diperoleh melalui rangkaian pelatihan, membaca, dan pengalaman. Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih cepat daripada non-pakar dalam memecahkan problem yang kompleks.

Konsep dasar sistem pakar yaitu pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Tujuan Sistem Pakar adalah untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (yang bukan pakar). Proses ini membutuhkan 4 aktivitas, yaitu tambahan pengetahuan (dari para ahli atau sumber lainnya), representasi pengetahuan (ke komputer), inferensi pengetahuan dan pengalihan pengetahuan ke pengguna.

Terdapat tiga orang yang terlibat dalam lingkungan sistem pakar, yaitu :

1. Pakar

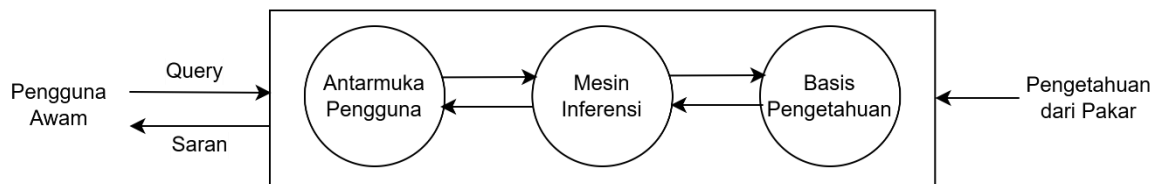
Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

2. Knowledge engineer (Perekayasa Sistem)

Knowledge engineer adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan counter example dan menerangkan kesulitan-kesulitan konseptual.

3. Pengguna

Sistem pakar memiliki beberapa pemakai, yaitu: pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan, dan pakar.



Mekanisme kerja sistem pakar, yaitu:

- Dimulai ketika pengguna non-pakar mengajukan pertanyaan melalui antarmuka pengguna
- Pertanyaan ini kemudian diproses oleh mesin inferensi, yang menerapkan aturan logika dan teknik penalaran untuk menganalisis masukan
- Mesin inferensi berinteraksi dengan basis pengetahuan, mengambil fakta, aturan, dan heuristik relevan yang disumbangkan oleh pengguna ahli
- Berdasarkan pengetahuan terstruktur ini, sistem menarik kesimpulan dan merumuskan respons yang tepat
- Terakhir, sistem pakar memberikan saran atau rekomendasi kepada pengguna, membantu dalam pengambilan keputusan atau pemecahan masalah

1.2 Basis Pengetahuan

Basis pengetahuan dalam sistem pakar adalah kumpulan data, fakta, aturan (rules), dan heuristik yang digunakan untuk mengidentifikasi solusi atau mengambil keputusan dalam menyelesaikan masalah. Basis pengetahuan berisi informasi yang diorganisir secara sistematis dan digunakan oleh mesin inferensi (Inference Engine) untuk menganalisis masalah dan mencari solusi.

Komponen Utama Basis Pengetahuan

Basis pengetahuan pada sistem pakar terdiri dari dua komponen utama, yaitu:

1. Fakta (Facts)

Fakta adalah informasi dasar tentang masalah yang ada. Fakta bisa berupa data konkret yang relevan dengan domain masalah. Misalnya, dalam sistem pakar medis, fakta bisa berupa "pasien mengalami demam" atau "pasien batuk."

2. Aturan (Rules)

Aturan adalah instruksi berbentuk IF-THEN yang menghubungkan fakta dengan keputusan atau solusi. Aturan ini merupakan cara sistem pakar untuk berpikir dan menyelesaikan masalah. Misalnya, aturan bisa berbunyi, "IF pasien demam AND batuk, THEN kemungkinan besar pasien menderita flu."

Selain fakta dan aturan, basis pengetahuan juga bisa menyertakan heuristik (pedoman atau aturan praktis yang berbasis pengalaman) yang lebih fleksibel dan tidak terlalu kaku dibandingkan dengan aturan yang berbentuk IF-THEN.

Jenis-Jenis Representasi Pengetahuan dalam Basis Pengetahuan

1. Penalaran berbasis aturan (rule-based reasoning)

Pada penalaran berbasis aturan pengetahuan direpresentasikan dalam bentuk aturan logika seperti IF-THEN, yang menghubungkan kondisi (fakta) dengan hasil (solusi). Metode ini digunakan jika terdapat sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Metode ini juga digunakan jika dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

Contoh : Aturan identifikasi hewan

- 1) Rule 1 : IF hewan berambut dan menyusui THEN hewan mamalia
- 2) Rule 2 : IF hewan mempunyai sayap dan bertelur THEN hewan jenis burung
- 3) Rule 3 : IF hewan mamalia dan memakan daging THEN hewan karnivora

2. Penalaran berbasis kasus (case-based reasoning)

Basis pengetahuan pada penalaran berbasis kasus akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Metode ini digunakan jika user menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Metode ini juga digunakan jika telah terdapat sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

2.3 Mesin Inferensi

Mesin inferensi adalah otak dari sistem pakar. Mesin inferensi merupakan struktur kontrol atau penerjemah aturan (dalam sistem pakar berbasis-aturan). Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan blackboard, dan merumuskan kesimpulan. Komponen ini menyediakan arahan bagaimana menggunakan pengetahuan sistem, yakni dengan mengembangkan agenda yang mengatur dan mengontrol langkah yang diambil untuk memecahkan persoalan kapan pun konsultasi berlangsung.

Fungsi Mesin Inferensi

1) Mesin inferensi berfungsi untuk

Menganalisis Fakta: Memeriksa fakta yang diberikan oleh pengguna dan mencocokkannya dengan aturan-aturan dalam basis pengetahuan.

2) Menarik Kesimpulan

Berdasarkan fakta yang tersedia dan aturan yang relevan, mesin inferensi akan menarik kesimpulan yang digunakan untuk menentukan solusi atau diagnosis.

3) Memberikan Solusi

Berdasarkan kesimpulan yang ditarik, mesin inferensi akan memberikan hasil berupa rekomendasi atau solusi kepada pengguna.

Terdapat dua pendekatan dalam menyusun mekanisme inferensi berbasis aturan, yaitu Forward Chaining dan Backward Chaining.

2.5.1. Forward Chaining

Forward Chaining merupakan pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

Proses

- Mesin inferensi mulai dengan fakta-fakta yang diketahui (misalnya, gejala yang dialami pasien)
- Mesin kemudian mencari aturan yang sesuai dengan fakta-fakta tersebut
- Jika aturan cocok, mesin akan mengambil tindakan atau menghasilkan kesimpulan
- Proses ini diulang sampai mesin menemukan solusi atau tidak ada aturan lebih lanjut yang dapat diterapkan

Contoh:

Terdapat aturan-aturan sebagai berikut:

No	Aturan
R-1	IF A & B THEN C
R-2	IF C THEN D
R-3	IF A & E THEN F
R-4	IF A THEN G
R-5	IF F & G THEN D
R-6	IF G & E THEN H
R-7	IF C & H THEN I
R-8	IF I & A THEN J

R-9	IF G THEN J
R-10	IF J THEN K

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal adalah A & F (artinya : A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis : K)?

Langkah-langkah inferensi adalah sebagai berikut :

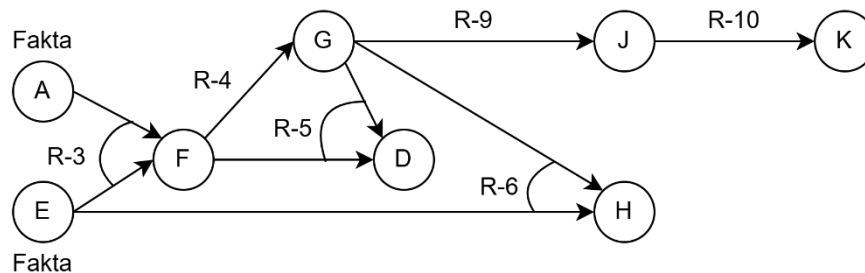
1. Dimulai dari R-1. A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa diketahui kebenarannya, sehingga C-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-2
2. Pada R-2, kita tidak mengetahui informasi apapun tentang C, sehingga kita juga tidak bisa memastikan kebenaran D. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-3
3. Pada R-3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu F. Karena F bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-4
4. Pada R-4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu G. Karena G bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-5
5. Pada R-5, baik F maupun G bernilai benar berdasarkan aturan R-3, dan R-4. Dengan demikian D sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu D. Karena D bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-6
6. Pada R-6, baik A maupun G adalah benar berdasarkan fakta dan R-4. Dengan demikian H sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu H. Karena H bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-7
7. Pada R-7, meskipun H benar berdasarkan R-6, namun kita tidak tahu kebenaran C, sehingga I-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-7 ini. Sehingga kita menuju ke R-8
8. Pada R-8, meskipun A benar karena fakta, namun kita tidak tahu kebenaran I, sehingga J-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-8 ini. Sehingga kita menuju ke R-9
9. Pada R-9, J bernilai benar karena G benar berdasarkan R-4. Karena J bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-10

10. Pada R-10, K bernilai benar karena J benar berdasarkan R-9. Karena H sudah merupakan hipotesis yang hendak kita buktikan ($=K$), maka terbukti bahwa K adalah benar

Tabel munculnya fakta baru pada saat inferensi adalah sebagai berikut:

Aturan	Fakta Baru
R-3	F
R-4	G
R-5	D
R-6	H
R-9	J
R-10	K

Alur inferensi Forward Chaining:



2.5.2. Backward Chaining

Backward Chaining merupakan pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut dicari harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Contoh:

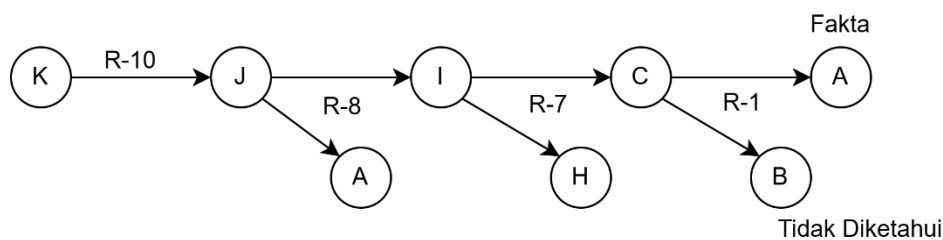
Terdapat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya : A & F (artinya : A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis : K)? Langkah-langkah inferensi adalah sebagai berikut:

- Pertama-tama kita cari terlebih dahulu mulai dari R-1, aturan mana yang memiliki konsekuen K. Ternyata setelah ditelusur, aturan dengan benar, maka perlu dibuktikan bahwa J benar
- Kita cari aturan yang memiliki konsekuen J. Kita mulai dari R-1, dan ternyata kita baru akan menemukan aturan dengan konsekuen J pada R-8. Untuk membuktikan bahwa J benar, maka perlu dibuktikan bahwa I dan A benar. Untuk membuktikan kebenaran I, kita perlu cari aturan dengan konsekuen I, ternyata ada di R-7

- Untuk membuktikan I benar di R-7, kita perlu buktikan bahwa C dan H benar. Untuk itu kitapun perlu mencari aturan dengan konsekuen C, dan ada di R-1
- Untuk membuktikan C benar di R-1, kita perlu buktikan bahwa A dan B benar. A jelas benar karena A merupakan fakta. Sedangkan B kita tidak bisa membuktikan kebenarannya, karena selain bukan fakta, di dalam basis pengetahuan juga tidak ada aturan dengan konsekuen B. Dengan demikian maka dari penalaran ini kita tidak bisa buktikan kebenaran dari hipotesis K. Namun demikian, kita masih punya alternatif lain untuk melakukan penalaran
- Kita lakukan backtracking. Kita ulangi lagi dengan pembuktian kebenaran C dengan mencari aturan lain dengan konsekuen C. Ternyata tidak ditemukan
- Kita lakukan backtracking lagi dengan mencari aturan dengan konsekuen I, ternyata juga tidak ada
- Kita lakukan backtracking lagi dengan mencari aturan dengan konsekuen J, ternyata kita temukan pada R-9. Sehingga kita perlu buktikan kebenaran G
- Kita mendapatkan R-4 dengan konsekuen G. Kita perlu buktikan kebenaran A. Karena A adalah fakta, maka terbukti bahwa G benar. Dengan demikian berdasarkan penalaran ini bisa dibuktikan bahwa K bernilai benar

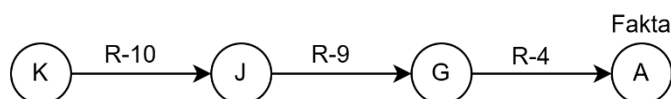
Alur inferensi Backward Chaining:

Kemungkinan 1:



Gagal: Hipotesis K tidak dapat dibuktikan karena terdapat satu node bukan fakta (tidak diketahui)

Kemungkinan 2:



Berhasil: Hipotesis K dapat dibuktikan dengan ditemukan fakta A

2.4 User Interface

Antarmuka pengguna (User Interface/UI) pada sistem pakar adalah komponen yang memungkinkan interaksi antara pengguna (seperti seorang dokter, teknisi, atau pengguna biasa)

dengan sistem pakar. UI ini memungkinkan pengguna untuk memasukkan informasi atau gejala yang mereka miliki dan menerima diagnosis atau solusi yang diberikan oleh sistem pakar.

Fungsi Antarmuka Pengguna pada Sistem Pakar

1. Mengumpulkan Input dari Pengguna

Antarmuka memungkinkan pengguna untuk memasukkan informasi atau data terkait masalah yang ingin mereka pecahkan. Misalnya, dalam sistem pakar medis, ini bisa berupa gejala yang dialami pasien

2. Menyampaikan Hasil Diagnosis atau Solusi

Setelah mesin inferensi selesai menganalisis data, UI akan menampilkan hasil diagnosis atau rekomendasi yang dihasilkan oleh sistem pakar

3. Menunjukkan Penjelasan

UI juga dapat memberikan penjelasan tentang proses inferensi atau bagaimana sistem mencapai kesimpulan atau solusi tertentu

4. Mempermudah Interaksi dengan Sistem

UI yang baik harus mempermudah pengguna dalam berinteraksi dengan sistem tanpa perlu memahami detail teknisnya. Pengguna dapat berfokus pada masukan yang relevan dan hasil yang diinginkan

Jenis Antarmuka Pengguna dalam Sistem Pakar

1. Antarmuka Pengguna Berbasis Teks (Text-Based UI)

Antarmuka berbasis teks menggunakan tampilan terminal atau konsol untuk berinteraksi dengan pengguna. Pengguna memasukkan informasi dalam bentuk teks atau memilih dari beberapa opsi yang ditampilkan di layar.

Kelebihan: Sederhana dan mudah diterapkan. Dapat digunakan di hampir semua sistem, bahkan sistem dengan sumber daya terbatas.

Kekurangan: Tidak begitu menarik secara visual dan kurang interaktif. Pengguna harus memasukkan data secara manual dan kadang-kadang rawan kesalahan ketik.

2. Antarmuka Pengguna Berbasis Grafik (Graphical User Interface - GUI)

GUI memungkinkan pengguna untuk berinteraksi dengan sistem menggunakan elemen-elemen visual seperti tombol, menu dropdown, dan input form. GUI memberikan pengalaman pengguna yang lebih baik karena lebih intuitif dan menarik dibandingkan UI berbasis teks.

Kelebihan: Memudahkan interaksi dengan menggunakan elemen visual yang lebih familiar dan ramah pengguna.

Kekurangan: Memerlukan lebih banyak sumber daya dan pengembangan, serta bisa lebih kompleks dalam hal implementasi.

3. Antarmuka Pengguna Berbasis Web (Web-Based UI)

Antarmuka berbasis web memungkinkan pengguna untuk mengakses sistem pakar melalui browser web. Antarmuka ini bisa berupa formulir online yang dapat diisi oleh pengguna atau aplikasi berbasis web yang menyediakan pengalaman pengguna yang lebih interaktif.

Kelebihan: Dapat diakses di berbagai perangkat dan lebih fleksibel. Pengguna hanya membutuhkan koneksi internet dan browser untuk mengakses sistem.

Kekurangan: Memerlukan pengembangan dan pemeliharaan server, serta terkadang tergantung pada koneksi internet.

2.5 Ketidakpastian

Jika sistem kecerdasan buatan yang dikembangkan memiliki pengetahuan yang lengkap, maka sistem dapat dengan mudah memberikan solusi menggunakan pendekatan logika. Tetapi, jika sistem hampir tidak pernah bisa mengakses seluruh fakta yang ada dalam lingkungan permasalahan yang akan diselesaikan, sehingga sistem harus bekerja dalam ketidakpastian dan kesamaran. Sehingga, sistem harus menggunakan teknik-teknik khusus yang dapat menangani ketidakpastian dan kesamaran dalam menyelesaikan permasalahan yang ditanganinya.

Contoh:

Potongan pengetahuan berikut yang direpresentasikan dalam aturan produksi menggambarkan gejala-gejala dari suatu penyakit.

Rule 1:

IF Has_fever (Patient) AND

Has_rash (Patient) AND

Has_high_body_ache(Patient)

THEN Bears-Typhoid (Patient)

Berdasarkan aturan diatas, terlihat bahwa jika seorang pasien mengalami ketiga gejala yang disebutkan dalam aturan, maka sistem akan mendiagnosa pasien tersebut menderita tifus (typhoid). Aturan ini akan diterapkan pada seluruh pasien yang mengalami ketiga gejala tersebut. Jika sistem ini diterapkan untuk kasus nyata, apakah gejala-gejala yang disebutkan telah mewakili seluruh gejala dari penyakit tifus? Bagaimana jika ada penyakit lain yang memiliki gejala yang sama dengan ketiga gejala tersebut? Bagaimana jika derajat gejala yang dialami seorang pasien dengan pasien yang lainnya berbeda? Pertanyaan-pertanyaan ini tidak dapat ditangani oleh aturan tersebut karena jawaban dari pasien sebagai pengguna tidak hanya “ya” atau “tidak”, sehingga muncul ketidakpastian dan kesamaran pengetahuan dalam permasalahan ini.

Ada tiga teknik yang dapat digunakan untuk menangani ketidakpastian dan kesamaran pengetahuan, yaitu:

2.5.1. Teknik Probabilitas

Teknik Probabilitas dikembangkan dengan memanfaatkan teorema Bayes yang menyajikan hubungan sebab akibat yang terjadi diantara evidenceevidence yang ada. Pendekatan alternatif lainnya yang dapat digunakan adalah teori Dempster-Shafer.

1. Teorema Bayes

Thomas Bayes menemukan pendekatan penalaran statistik yang jauh lebih maju dibandingkan dengan pola pikir matematis tradisional pada saat itu. Fokus matematika pada saat itu adalah pada tingkah laku sampel dari populasi yang diketahui. Akan tetapi, Bayes mengemukakan ide untuk menentukan properti dari populasi berdasarkan sampel tersebut.

Teorema Bayes diperoleh dari aturan produksi konjungsi dengan notasi:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

Dimana :

$P(Y|X)$: kemungkinan hipotesis Y jika diberi bukti X

$P(X|Y)$: kemungkinan munculnya bukti X jika diketahui hipotesis Y

$P(Y)$: kemungkinan hipotesis Y tanpa memandang bukti apapun

$P(X)$: kemungkinan bukti X

Contoh:

Seorang dokter mengetahui bahwa penyakit meningitis menyebabkan "stiff neck" adalah 50%. Probabilitas pasien menderita meningitis adalah 1/50000 dan probabilitas pasien menderita stiff neck adalah 1/20 dari nilai-nilai tersebut. Didapatkan :

$$P(stiffneck|meningitis) = 50\% = 0,5$$

$$P(meningitis) = 1/50000$$

$$P(stiffneck) = 1/20$$

Maka :

$$\begin{aligned} P(meningitis|stiffneck) &= \frac{P(stiffneck|meningitis) P(meningitis)}{P(stiffneck)} \\ &= \frac{5/10 \times 1/50000}{1/20} \\ &= \frac{1}{5000} = 0,0002 \end{aligned}$$

Hasil diatas menunjukkan bahwa hanya 1 diantara 5000 pasien yang mengalami stiff neck.

2. Teori Dempster-Shaffer

Secara umum teori Dempster-Shafer dapat ditulis dalam suatu interval: [Belief, Plausability]. Belief (Bel) adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 maka menunjukkan adanya kepastian. Plausability (Pl) dinotasikan sebagai:

$$Pl(s) = 1 - Bel(\neg s)$$

Dimana:

$Pl(s)$ = Fungsi kemungkinan

$Bel(\neg s)$ = keyakinan bahwa hipotesis s tidak benar

Plausability juga bernilai antara 0 dan 1. Jika kita yakin terhadap $\neg s$, maka dapat dikatakan bahwa $Bel(\neg s) = 1$ dan $Pl(s) = 0$.

Salah satu fitur utama dari teori Dempster-Shafer adalah kemampuannya untuk menggabungkan bukti dari dua atau lebih sumber yang berbeda. Dempster mengusulkan Aturan Penggabungan untuk mengkombinasikan informasi yang diberikan oleh dua fungsi massa pembobot m_1 dan m_2 .

Aturan Penggabungan Dempster adalah sebagai berikut:

$$m(s) = \frac{\sum_{x \cap y = s} m_1(x) \times m_2(y)}{1 - \sum_{x \cap y = \emptyset} m_1(x) \times m_2(y)}$$

Dimana:

- x dan y adalah subset dari ruang hipotesis yang relevan dari dua sumber informasi
- Bagian numerator menghitung kontribusi bukti dari kedua sumber untuk himpunan bagian s
- Bagian denominator adalah faktor normalisasi yang memastikan bahwa jumlah semua massa bukti tetap 1 (menghindari kontradiksi atau ketidakselarasan)

2.5.2. Teknik Kepastian (Certainty Factor)

Faktor kepastian (Certainty Factor-CF) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN. Faktor kepastian merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan.

Ada dua macam faktor kepastian yang dapat digunakan, yaitu faktor kepastian yang diisikan oleh pakar bersama aturan dan faktor kepastian yang diberikan oleh pengguna. Faktor kepastian yang diisikan oleh pakar menggambarkan kepercayaan pakar terhadap hubungan antara antecedent dan consequent pada aturan kaidah produksi. Faktor kepastian dari pengguna menunjukkan besarnya kepercayaan terhadap keberadaan masing-masing elemen dalam antecedent.

Faktor kepastian didefinisikan sebagai berikut.

$$CF[h, e] = MB[h, e] - MD[h, e]$$

Dimana:

$CF[h, e]$ = faktor kepastian

$MB[h, e]$ = ukuran kepercayaan terhadap hipotesis h, jika diberikan evidence e (antara 0 dan 1)

$MD[h, e]$ = ukuran ketidakpercayaan terhadap hipotesis h, jika diberikan evidence e (antara 0 dan 1)

Pengguna akan memberikan faktor kepastian terhadap setiap premis yang ada dalam aturan. Untuk menentukan faktor kepastian dari suatu aturan yang didalamnya terdapat beberapa premis dengan faktor kepastiannya masing-masing, maka perlu dilakukan perhitungan CF paralel. CF paralel merupakan CF yang diperoleh dari beberapa premis pada sebuah aturan.

Besarnya CF paralel dipengaruhi oleh CF pengguna untuk masing-masing premis dan operator dari premis. Rumus untuk masing-masing operator diberikan sebagai berikut.

$$CF(x \text{ and } y) = \text{Min}(CF(x), CF(y))$$

$$CF(x \text{ or } y) = \text{Max}(CF(x), CF(y))$$

$$CF(\text{not } x) = -CF(x)$$

Suatu hipotesis pada kenyataannya bisa dihasilkan dari beberapa aturan yang berbeda, dimana setiap aturan memiliki faktor kepastian masing-masing, sehingga perlu dilakukan perhitungan CF gabungan dari seluruh aturan yang ada untuk suatu hipotesis.

CF gabungan merupakan CF akhir dari sebuah calon kesimpulan. CF ini dipengaruhi oleh semua CF sekuensial dari aturan yang menghasilkan kesimpulan tersebut. Rumus untuk melakukan perhitungan CF gabungan adalah sebagai berikut.

- Jika $CF(x) > 0$ dan $CF(y) > 0$, maka:

$$CF(x, y) = CF(x) + CF(y) - (CF(x) \times CF(y))$$

- Jika salah satu, $CF(x)$ atau $CF(y) < 0$, maka:

$$CF(x, y) = \frac{CF(x) + CF(y)}{1 - (\text{MIN}(|CF(x)|, |CF(y)|))}$$

- Jika $CF(x) < 0$ dan $CF(y) < 0$, maka:

$$CF(x, y) = CF(x) + (CF(y) \times (1 + CF(x)))$$

2.5.3. Logika Fuzzy (Fuzzy Logic)

Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika fuzzy adalah teori himpunan fuzzy. Pada teori himpunan fuzzy, peranan derajat keanggotaan/nilai keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting.

Logika fuzzy digunakan sebagai suatu cara untuk memecahkan permasalahan dari input menuju ke output yang diharapkan. Pada himpunan fuzzy, nilai keanggotaan terletak pada rentang 0 sampai 1. Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy, misalnya umur, temperature, dan lain-lain.

2. Praktikum

Aplikasi Sistem Pakar Untuk Simulasi Diagnosa Hama Dan Penyakit Tanaman Bawang Merah Menggunakan Forward Chaining Dan Pendekatan Berbasis Aturan.

2.1. Jenis Hama dan Penyakit

a. Ulat bawang

Serangga dewasa merupakan ngengat dengan sayap depan berwarna kelabu gelap dan sayap belakang berwarna agak putih. Imago betina meletakkan telur secara berkelompok pada ujung daun. Satu kelompok biasanya berjumlah 50 – 150 butir telur. Telur dilapisi oleh bulu-bulu putih yang berasal dari sisik tubuh induknya. Telur berwarna putih, berbentuk bulat atau bulat telur (lonjong) dengan ukuran sekitar 0,5 mm. Larva *S. exigua* berukuran panjang 2,5 cm dengan warna yang bervariasi. Ketika masih muda, larva berwarna hijau muda dan jika sudah tua berwarna hijau kecoklatan gelap dengan garis kekuningan-kuningan.

b. Ulat grayak

Ngengat berwarna agak gelap dengan garis putih pada sayap depannya, sedangkan sayap belakang berwarna putih dengan bercak hitam. Telur berwarna putih diletakkan berkelompok dan berbulu halus seperti diselimuti kain laken. Larva mempunyai warna yang bervariasi, tetapi mempunyai kalung hitam pada segmen abdomen yang keempat dan kesepuluh. Pada sisi lateral dan dorsal terdapat garis kuning. Pupa berwarna coklat gelap terbentuk dalam tanah.

c. Trips

Tubuhnya tipis sepanjang ± 1 mm dan dengan sayap berumbai-umbai. Warna tubuh kuning dan berubah menjadi coklat sampai hitam jika sudah dewasa. Telur berwarna kekuningan, lama hidup 4 – 5 hari. Nimfa berwarna putih kekuningan lama hidupnya sekitar 9 hari. Pupa terbentuk dalam tanah, lama hidup sekitar 9 hari. Gejala serangan daun berwarna putih keperak-perakan. Pada serangan hebat, seluruh areal pertanian berwarna putih dan akhirnya tanaman mati. Serangan hebat terjadi pada suhu udara rata-rata di atas normal dan kelembaban lebih dari 70%.

d. Lalat pengorok daun

Liriomyza sp. menyerang tanaman bawang merah dari umur 15 hari setelah tanam sampai menjelang panen. Spesies yang menyerang tanaman bawang merah adalah *L. chinensis*. Gejala daun bawang merah yang terserang, berupa bintik-bintik putih akibat tusukan ovipositor, dan berupa liang korokan larva yang berkelok-kelok. Pada keadaan serangan berat, hampir seluruh helaian daun penuh dengan korokan, sehingga menjadi kering dan berwarna coklat seperti terbakar

e. Penyakit trotol atau bercak ungu (Purple blotch)

Infeksi awal pada daun menimbulkan bercak berukuran kecil, melekok ke dalam, berwarna putih dengan pusat yang berwarna ungu (kelabu). Jika cuaca lembab, serangan berlanjut dengan cepat, bercak berkembang hingga menyerupai cincin dengan bagian tengah yang berwarna ungu dengan tepi yang kemerahan dikelilingi warna kuning yang dapat meluas ke bagian atas maupun bawah bercak. Ujung daun mengering, sehingga daun patah. Permukaan bercak tersebut akhirnya berwarna coklat kehitaman. Serangan dapat berlanjut ke umbi, yang menyebabkan umbi membusuk, berwarna kuning lalu merah kecoklatan. Semula umbi membusuk dan berair yang dimulai dari bagian leher, kemudian jaringan umbi yang terinfeksi mengering dan berwarna lebih gelap.

f. Penyakit antraknose (Antracnose)

Serangan awal ditandai dengan terlihatnya bercak berwarna putih pada daun, selanjutnya terbentuk lekukan ke dalam (invaginasi), berlubang dan patah karena terkulai tepat pada bercak tersebut. Jika infeksi berlanjut, maka terbentuklah koloni konidia yang berwarna merah muda, yang kemudian berubah menjadi coklat muda, coklat tua, dan akhirnya kehitam-hitaman. Dalam kondisi kelembaban udara yang tinggi terutama pada musim penghujan, konidia berkembang dengan cepat membentuk miselia yang tumbuh menjalar dari helaian daun, masuk menembus sampai ke umbi, seterusnya menyebar di permukaan tanah, berwarna putih, dan menginfeksi inang di sekitarnya. Umbi kemudian membusuk, daun mengering dan sebaran serangan yang bersifat sporadis tersebut, pada hamparan tanaman akan terlihat gejala botak-botak di beberapa tempat.

g. Penyakit embun bulu

Pada kondisi yang lembab, berkabut atau curah hujan tinggi, cendawan akan membentuk masa spora yang sangat banyak, yang terlihat sebagai bulu-bulu halus berwarna ungu (violet) yang menutupi daun bagian luar dan batang (umbi). Gejala akibat infeksi cendawan ini dapat bersifat sistemik dan lokal. Jika infeksi terjadi pada awal pertumbuhan tanaman, dan tanaman mampu bertahan hidup, maka pertumbuhan tanaman terhambat

dan daun berwarna hijau pucat. Bercak infeksi pada daun mampu menyebar ke bawah hingga mencapai umbi lapis, kemudian menjalar ke seluruh lapisan, Akibatnya, umbi menjadi berwarna coklat. Serangan lanjut akan mengakibatkan umbi membusuk, tetapi lapisan luarnya mengering dan berkerut, daun layu dan mengering, sering dijumpai anyaman miselia yang berwarna hitam. Gejala lokal biasanya merupakan akibat infeksi sekunder, yang mengakibatkan bercak pada daun yang berwarna pucat dan berbentuk lonjong, yang mampu menimbulkan gejala sistemik seperti tersebut di atas.

h. Penyakit moler

Gejala visual adalah daun yang menguning dan cenderung terpelintir (terputar). Tanaman sangat mudah tercabut karena pertumbuhan akar terganggu bahkan membusuk. Pada dasar umbi terlihat cendawan yang berwarna keputih-putihan, sedangkan jika umbi lapis dipotong membujur terlihat adanya pembusukan, yang berawal dari dasar umbi meluas ke atas maupun ke samping. Serangan lanjut akan mengakibatkan tanaman mati, yang dimulai dari ujung daun dan dengan cepat menjalar ke bagian bawahnya.

i. Penyakit ngelumpruk

Bercak-bercak berwarna putih kekuning-kuningan, tumbuh sangat banyak dan cepat sesuai dengan arah bertiupnya angin di awal pertanaman. Cendawan tersebut mampu mematikan tanaman secara serentak dan kumpulan tanaman yang mati serentak tersebut terlihat seperti pada kelembaban udara yang tinggi dan berangin. Cendawan ditemukan baik menginfeksi secara tunggal maupun berasosiasi dengan cendawan *A. porri*.

j. Penyakit bercak daun *Serkospora*

Bercak klorosis kebanyakan terkumpul pada ujung daun dan sering tampak terpisah dengan yang menginfeksi pangkal daun, sehingga gejala visualnya terlihat daun tampak belang-belang. Bercak klorosis yang berbentuk bulat tersebut berwarna kuning pucat, bergaris tengah sekitar 3-5 mm. Serangan lebih lanjut menyebabkan pusat bercak berwarna coklat karena jaringannya mati. Di bagian tersebut terdapat bintik-bintik yang sebenarnya terdiri atas berkas-berkas konidiofora yang mengandung konidia, yang tampak jelas jika cuaca lembab.

2.2. Penyusunan Tabel

a. Penyusunan tabel hama dan penyakit

No	Hama
H1	Ulat Bawang
H2	Ulat Grayak
H3	Trips
H4	Lalat Pengorok Daun

No	Penyakit
P1	Trotol atau Bercak Ungu
P2	Antraknose
P3	Embun Bulu
P4	Layu Fusarium
P5	Ngelumpruk
P6	Serkospora

b. Penyusunan tabel gejala

No	Gejala
GH1	Ngengat
GH2	Sayap depan berwarna kelabu gelap
GH3	Telur terletak pada ujung daun
GH4	Telur berwarna putih
GH5	Telur berbentuk bulat
GH6	Sayap depan berwarna gelap dengan garis putih
GH7	Pupa berwarna coklat gelap terbentuk dalam tanah
GH8	Sayap berumbai-umbai
GH9	Telur berwarna kekuningan
GH10	Daun berwarna putih keperak-perakan
GH11	Bintik-bintik putih
GH12	Adanya liang korokan larva yang berkelok-kelok
GP1	Bercak berukuran kecil
GP2	Bercak melekok ke dalam
GP3	Berwarna putih dengan pusat yang berwarna ungu (kelabu)
GP4	Suhu lembab
GP5	Umbi membusuk dan berair
GP6	Terlihatnya bercak berwarna putih pada daun
GP7	Terbentuklah koloni konidia yang berwarna merah muda
GP8	Adanya miselia yang tumbuh menjalar dari helaian daun
GP9	Spora yang terlihat sebagai bulu-bulu halus berwarna ungu
GP10	Daun berwarna hijau pucat
GP11	Umbi menjadi berwarna coklat
GP12	Daun yang menguning
GP13	Daun terpelintir (terputar)
GP14	Pada dasar umbi terlihat cendawan yang berwarna keputih-putihan
GP15	Bercak-bercak berwarna putih kekuning-kuningan
GP16	Daun tampak belang-belang
GP17	Bercak klorosis yang berbentuk bulat
GP18	Bercak klorosis berwarna kuning pucat
GP19	Terdapat bintik-bintik

c. Penyusunan tabel aturan (Basis Pengetahuan)

No	Aturan
A1	IF GH1 and GH2 and GH3 and GH4 and GH5 THEN H1
A2	IF GH1 and GH4 and GH6 and GH7 THEN H2
A3	IF GH8 and GH9 and GH 10 THEN H3

A4	IF GH11 and GH12 THEN H4
A5	IF GP1 and GP2 and GP3 and GP4 and GP5 THEN P1
A6	IF GP6 and GP7 and GP4 and GP8 THEN P2
A7	IF GP9 and GP10 and GP11 THEN P3
A8	IF GP12 and GP13 and GP14 THEN P4
A9	IF GP15 and GP4 THEN P5
A10	IF GP16 and GP17 and GP18 and GP4 and GP19 THEN P6

2.3. Pembuatan Program

Pembuatan program sistem pakar dibagi menjadi dua jenis. Jenis yang pertama hanya menggunakan python saja dan jenis yang kedua dengan menerapkan sistem pakar berbasis website.

2.3.1. Sistem Pakar berbasis Python

Contoh metode yang digunakan pada modul ini adalah metode Forward Chaining, sehingga penelusuran inferensi akan dimulai dari gejala yang kemudian akan menghasilkan sebuah kesimpulan. Terdapat bagian yang harus diperhatikan dalam pembuatan program sistem pakar, yaitu aturan (rules), modul yang berisi metode dan UI.

Pada bagian aturan berisi gejala dan penyakit yang dibuat dalam bentuk percabangan IF THEN.

Kode program aturan ditunjukkan oleh gambar berikut:

```

1 rules = [
2     {"if": {"ngengat": True, "sayap depan kelabu": True, "sayap depan garis putih": False, "sayap berumbai": False, "telur warna putih": True, "te
3     {"if": {"ngengat": True, "sayap depan kelabu": False, "sayap depan garis putih": True, "sayap berumbai": False, "telur warna putih": True, "te
4     {"if": {"ngengat": False, "sayap depan kelabu": False, "sayap depan garis putih": False, "sayap berumbai": True, "telur warna putih": False, "
5     {"if": {"ngengat": False, "sayap depan kelabu": False, "sayap depan garis putih": False, "sayap berumbai": False, "telur warna putih": False, "
6     {"if": {"suhu lembab": True, "bercak ukuran kecil": True, "bercak putih pada daun": False, "berwarna putih dengan pusat yang berwarna ungu": T
7     "then": "Tanaman Bawang Merah Terkena Penyakit Trotol atau Bercak Ungu"},
8     {"if": {"suhu lembab": True, "bercak ukuran kecil": False, "bercak putih pada daun": True, "bercak-bercak putih kekuning-kuningan": False, "da
9     "then": "Tanaman Bawang Merah Terkena Penyakit Antraknose"},
10    {"if": {"suhu lembab": False, "bercak ukuran kecil": False, "bercak putih pada daun": False, "bercak-bercak putih kekuning-kuningan": False, "i
11    "then": "Tanaman Bawang Merah Terkena Penyakit Embun Bulu"},
12    {"if": {"suhu lembab": False, "bercak ukuran kecil": False, "bercak putih pada daun": False, "bercak-bercak putih kekuning-kuningan": False, "i
13    "then": "Tanaman Bawang Merah Terkena Penyakit Layu Fusarium"},
14    {"if": {"suhu lembab": True, "bercak ukuran kecil": False, "bercak putih pada daun": False, "bercak-bercak putih kekuning-kuningan": True, "da
15    "then": "Tanaman Bawang Merah Terkena Penyakit NgeLumpruk"},
16    {"if": {"suhu lembab": True, "bercak ukuran kecil": False, "bercak putih pada daun": False, "bercak-bercak putih kekuning-kuningan": False, "d
17    "then": "Tanaman Bawang Merah Terkena Penyakit Serkospora"}
18 ]

```

Pada bagian modul metode berisi kode program metode forward chaining untuk membuat sebuah kesimpulan. Kode program modul metode ditunjukkan oleh gambar berikut:

```

20 def infer(symptoms):
21     for rule in rules:
22         if all(symptoms.get(k) == v for k, v in rule["if"].items()):
23             return rule["then"]
24     return "Maaf Gejala Tidak Ditemukan"

```

Pada bagian UI berisi kode program desain tampilan program sistem pakar. Kode program UI ditunjukkan oleh gambar berikut:

```

26 def main():
27     print("Selamat Datang di Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Bawang Merah")
28     symptoms = {}
29
30     # Gather user input
31     symptoms["ngengat"] = input("Apakah serangga hama berbentuk ngengat? (ya/tidak): ").lower() == "ya"
32     symptoms["sayap depan kelabu"] = input("Apakah serangga hama memiliki sayap depan berwarna kelabu gelap? (ya/tidak): ").lower() == "ya"
33     symptoms["sayap depan garis putih"] = input("Apakah serangga hama memiliki sayap garis putih? (ya/tidak): ").lower() == "ya"
34     symptoms["sayap berumbai"] = input("Apakah serangga hama memiliki sayap berumbai? (ya/tidak): ").lower() == "ya"
35     symptoms["telur warna putih"] = input("Apakah serangga hama memiliki telur berwarna putih? (ya/no): ").lower() == "ya"
36     symptoms["telur kuning"] = input("Apakah serangga hama memiliki telur kuning? (ya/no): ").lower() == "ya"
37     symptoms["suhu lembab"] = input("Apakah tanaman bawang terletak di suhu lembab? (ya/no): ").lower() == "ya"
38     symptoms["bercak ukuran kecil"] = input("Apakah tanaman bawang memiliki bercak berukuran kecil? (ya/no): ").lower() == "ya"
39     symptoms["bercak putih pada daun"] = input("Apakah tanaman bawang memiliki bercak putih pada daun? (ya/no): ").lower() == "ya"
40     symptoms["bercak-bercak putih kekuning-kuningan"] = input("Apakah tanaman bawang memiliki bercak-bercak putih kekuning-kuningan? (ya/no): ").lower() == "ya"
41     symptoms["daun yang menguning"] = input("Apakah tanaman bawang memiliki daun yang menguning? (ya/no): ").lower() == "ya"
42     symptoms["daun berwarna hijau pucat"] = input("Apakah tanaman bawang memiliki daun berwarna hijau pucat? (ya/no): ").lower() == "ya"
43     symptoms["daun tampak belang-belang"] = input("Apakah tanaman bawang memiliki daun tampak belang-belang? (ya/no): ").lower() == "ya"
44     symptoms["Umbi menjadi berwarna coklat"] = input("Apakah tanaman bawang memiliki Umbi menjadi berwarna coklat? (ya/no): ").lower() == "ya"
45     symptoms["dasar umbi terlihat cendawan putih"] = input("Apakah tanaman bawang memiliki dasar umbi terlihat cendawan putih? (ya/no): ").lower() == "ya"
46
47     diagnosis = infer(symptoms)
48     print("Diagnosis: " + diagnosis)
49
50 if __name__ == "__main__":
51     main()

```

Full kode program sistem pakar menggunakan metode forward chaining berbasis python yaitu:

```

1 rules = [
2     {"if": {"ngengat": True, "sayap depan kelabu": True, "sayap depan garis putih": False, "sayap berumbai": False, "telur warna putih": True, "telur kuning": False, "suhu lembab": True, "bercak ukuran kecil": True, "bercak putih pada daun": True, "bercak-bercak putih kekuning-kuningan": False, "daun yang menguning": False, "daun berwarna hijau pucat": False, "daun tampak belang-belang": False, "Umbi menjadi berwarna coklat": False, "dasar umbi terlihat cendawan putih": False}, "then": "Tanaman Bawang Merah Terkena Penyakit Antraknose"},
3     {"if": {"ngengat": True, "sayap depan kelabu": False, "sayap depan garis putih": True, "sayap berumbai": False, "telur warna putih": True, "telur kuning": False, "suhu lembab": True, "bercak ukuran kecil": True, "bercak putih pada daun": True, "bercak-bercak putih kekuning-kuningan": False, "daun yang menguning": False, "daun berwarna hijau pucat": False, "daun tampak belang-belang": False, "Umbi menjadi berwarna coklat": False, "dasar umbi terlihat cendawan putih": False}, "then": "Tanaman Bawang Merah Terkena Penyakit Embun Bulu"},
4     {"if": {"ngengat": False, "sayap depan kelabu": False, "sayap depan garis putih": False, "sayap berumbai": True, "telur warna putih": False, "telur kuning": True, "suhu lembab": True, "bercak ukuran kecil": True, "bercak putih pada daun": True, "bercak-bercak putih kekuning-kuningan": False, "daun yang menguning": False, "daun berwarna hijau pucat": False, "daun tampak belang-belang": False, "Umbi menjadi berwarna coklat": False, "dasar umbi terlihat cendawan putih": False}, "then": "Tanaman Bawang Merah Terkena Penyakit Layu Fusarium"},
5     {"if": {"ngengat": False, "sayap depan kelabu": False, "sayap depan garis putih": False, "sayap berumbai": False, "telur warna putih": False, "telur kuning": True, "suhu lembab": True, "bercak ukuran kecil": True, "bercak putih pada daun": True, "bercak-bercak putih kekuning-kuningan": True, "daun yang menguning": True, "daun berwarna hijau pucat": True, "daun tampak belang-belang": True, "Umbi menjadi berwarna coklat": True, "dasar umbi terlihat cendawan putih": True}, "then": "Tanaman Bawang Merah Terkena Penyakit Ngelumprik"},
6     {"if": {"ngengat": False, "sayap depan kelabu": False, "sayap depan garis putih": False, "sayap berumbai": False, "telur warna putih": False, "telur kuning": False, "suhu lembab": True, "bercak ukuran kecil": False, "bercak putih pada daun": False, "bercak-bercak putih kekuning-kuningan": False, "daun yang menguning": False, "daun berwarna hijau pucat": False, "daun tampak belang-belang": False, "Umbi menjadi berwarna coklat": False, "dasar umbi terlihat cendawan putih": False}, "then": "Tanaman Bawang Merah Terkena Penyakit Serkospora"}
7 ]
8
9 def infer(symptoms):
10     for rule in rules:
11         if all(symptoms.get(k) == v for k, v in rule["if"].items()):
12             return rule["then"]
13     return "Maaf Gejala Tidak Ditemukan"
14
15 if __name__ == "__main__":
16     main()

```

```

26 def main():
27     print("Selamat Datang di Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Bawang Merah")
28     symptoms = {}
29
30     # Gather user input
31     symptoms["ngengat"] = input("Apakah serangga hama berbentuk ngengat? (ya/tidak): ").lower() == "ya"
32     symptoms["sayap depan kelabu"] = input("Apakah serangga hama memiliki sayap depan berwarna kelabu gelap? (ya/tidak): ").lower() == "ya"
33     symptoms["sayap depan garis putih"] = input("Apakah serangga hama memiliki sayap garis putih? (ya/tidak): ").lower() == "ya"
34     symptoms["sayap berumbai"] = input("Apakah serangga hama memiliki sayap berumbai? (ya/tidak): ").lower() == "ya"
35     symptoms["telur warna putih"] = input("Apakah serangga hama memiliki telur berwarna putih? (ya/no): ").lower() == "ya"
36     symptoms["telur kuning"] = input("Apakah serangga hama memiliki telur kuning? (ya/no): ").lower() == "ya"
37     symptoms["suhu lembab"] = input("Apakah tanaman bawang terletak di suhu lembab? (ya/no): ").lower() == "ya"
38     symptoms["bercak ukuran kecil"] = input("Apakah tanaman bawang memiliki bercak berukuran kecil? (ya/no): ").lower() == "ya"
39     symptoms["bercak putih pada daun"] = input("Apakah tanaman bawang memiliki bercak putih pada daun? (ya/no): ").lower() == "ya"
40     symptoms["bercak-bercak putih kekuning-kuningan"] = input("Apakah tanaman bawang memiliki bercak-bercak putih kekuning-kuningan? (ya/no): ").lower() == "ya"
41     symptoms["daun yang menguning"] = input("Apakah tanaman bawang memiliki daun yang menguning? (ya/no): ").lower() == "ya"
42     symptoms["daun berwarna hijau pucat"] = input("Apakah tanaman bawang memiliki daun berwarna hijau pucat? (ya/no): ").lower() == "ya"
43     symptoms["daun tampak belang-belang"] = input("Apakah tanaman bawang memiliki daun tampak belang-belang? (ya/no): ").lower() == "ya"
44     symptoms["Umbi menjadi berwarna coklat"] = input("Apakah tanaman bawang memiliki Umbi menjadi berwarna coklat? (ya/no): ").lower() == "ya"
45     symptoms["dasar umbi terlihat cendawan putih"] = input("Apakah tanaman bawang memiliki dasar umbi terlihat cendawan putih? (ya/no): ").lower() == "ya"
46
47     diagnosis = infer(symptoms)
48     print("Diagnosis: " + diagnosis)
49
50 if __name__ == "__main__":
51     main()

```

Hasil Running program yaitu:

```

Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/FX506HC/Sistem Pakar/pakar.py', wdir='C:/Users/FX506HC/
Sistem Pakar')
Selamat Datang di Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Bawang Merah
Apakah serangga hama berbentuk ngengat? (ya/tidak):

```

Pada hasil running setiap pertanyaan yang tampil harus dijawab untuk menghasilkan sebuah kesimpulan.

```
Sistem Pakar')
Selamat Datang di Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Bawang Merah
Apakah serangga hama berbentuk ngengat? (ya/tidak): ya
Apakah serangga hama memiliki sayap depan berwarna kelabu gelap? (ya/tidak): ya
Apakah serangga hama memiliki sayap garis putih? (ya/tidak): tidak
Apakah serangga hama memiliki sayap berumbai? (ya/tidak): tidak
Apakah serangga hama memiliki telur berwarna putih? (ya/no): ya
Apakah serangga hama memiliki telur kuning? (ya/no): tidak
Apakah tanaman bawang terletak di suhu lembab? (ya/no): tidak
Apakah tanaman bawang memiliki bercak berukuran kecil? (ya/no): tidak
Apakah tanaman bawang memiliki bercak putih pada daun? (ya/no): tidak
Apakah tanaman bawang memiliki bercak-bercak putih kekuning-kuningan? (ya/no): tidak
Apakah tanaman bawang memiliki daun yang menguning? (ya/no): tidak
Apakah tanaman bawang memiliki daun berwarna hijau pucat? (ya/no): tidak
Apakah tanaman bawang memiliki daun tampak belang-belang? (ya/no): tidak
Apakah tanaman bawang memiliki Umbi menjadi berwarna coklat? (ya/no): tidak
Apakah tanaman bawang memiliki dasar umbi terlihat cendawan putih? (ya/no): tidak
Diagnosis: Tanaman Bawang Merah Terkena Hama Ulat Bawang
```

2.3.2. Sistem Pakar berbasis Website

Cara kedua untuk merancang program sistem pakar adalah melalui sebuah website. Perancangan website dengan menggunakan python sebagai backend program adalah dengan menggunakan library Flask ataupun Django. Pada modul ini, library yang digunakan untuk merancang python sebagai backend program adalah library Flask. Sehingga sebelum merancang kerangka kerja, library Flask harus diinstal terlebih dahulu, karena library Flask bukan library bawaan dari python. Cara instalasi library Flask bisa dilihat di link berikut:

<https://flask.palletsprojects.com/en/stable/installation/>

Setelah library Flask diinstall kemudian langkah berikutnya adalah merancang kerangka kerja python. Terdapat tiga file yang akan dirancang, yaitu file metode.py, app.py dan index.html. Susunan file dan folder dari ketiga file ini ditunjukkan oleh gambar berikut:

```
your_project/
├─ app.py
├─ metode.py
└─ templates/
    └─ index.html
```

<div> <div> <div></div> <div>This PC</div> </div> <div> <div>OS (C:)</div> <div>Users</div> <div>FX506HC</div> <div>Sistem Pakar</div> </div> </div>			
<div> <div> <div></div> <div>Sort</div> </div> <div> <div></div> <div>View</div> </div> <div>...</div> </div>			
Name	Date modified	Type	Size
<div> <div></div> <div>_pycache_</div> </div>	15/04/2025 00:09	File folder	
<div> <div></div> <div>templates</div> </div>	14/04/2025 23:34	File folder	
<div> <div></div> <div>pakar_web.py</div> </div>	13/04/2025 16:46	PY File	20 KB
<div> <div></div> <div>app.py</div> </div>	15/04/2025 00:08	PY File	2 KB
<div> <div></div> <div>metode.py</div> </div>	15/04/2025 00:09	PY File	3 KB
<div> <div></div> <div>pakar.py</div> </div>	15/04/2025 00:39	PY File	6 KB

Pada file metode.py berisi metode sistem pakar yang digunakan. Dalam file ini juga berisi aturan yang digunakan untuk membuat kesimpulan. Kode program file metode.py yaitu:

```

1 rules = [
2     {"if": {"ngengat", "sayap depan berwarna kelabu gelap", "telur terletak pada ujung daun", "telur berwarna putih", "telur berbentuk bulat"}, "then":
3     {"if": {"ngengat", "telur berwarna putih", "sayap depan berwarna gelap dengan garis putih", "pupa berwarna coklat gelap terbentuk dalam tanah"},
4     {"if": {"sayap berumbai-umbai", "telur berwarna kekuningan", "daun berwarna putih keperak-perakan"}, "then": "Trips"},
5     {"if": {"bintik-bintik putih", "adanya liang korokan larva yang berkelok-kelok", "daun berwarna putih keperak-perakan"}, "then": "LaLat Pengorok D
6     {"if": {"bercak berukuran kecil", "bercak melekok ke dalam", "berwarna putih dengan pusat yang berwarna ungu", "suhu lembab", "umbi membusuk dan b
7     {"if": {"suhu lembab", "terlihatnya bercak berwarna putih pada daun", "terbentuklah koloni konidia yang berwarna merah muda", "adanya miselia yang
8     {"if": {"spora yang terlihat sebagai bulu-bulu halus berwarna ungu", "daun berwarna hijau pucat", "umbi menjadi berwarna coklat"}, "then": "Embun
9     {"if": {"daun yang menguning", "daun terpelintir", "pada dasar umbi terlihat cendawan yang berwarna keputih-putihan"}, "then": "Layu Fusarium"},
10    {"if": {"suhu lembab", "bercak-bercak berwarna putih kekuning-kuningan"}, "then": "Ngelumpruk"},
11    {"if": {"suhu lembab", "daun tampak belang-belang", "bercak klorosis yang berbentuk bulat", "bercak klorosis berwarna kuning pucat", "terdapat bin
12 ]
13
14 def forward_chaining(facts):
15     known_facts = set(facts)
16     inferred = set()
17
18     changed = True
19     while changed:
20         changed = False
21         for rule in rules:
22             if rule["if"].issubset(known_facts) and rule["then"] not in known_facts:
23                 known_facts.add(rule["then"])
24                 inferred.add(rule["then"])
25                 changed = True
26     return list(inferred)

```

Pada file app.py berisi perancangan backend program yang berfungsi sebagai jembatan antara file metode dengan file index. Kode program file app.py yaitu:

```

1 from flask import Flask, render_template, request
2 from metode import forward_chaining
3
4 app = Flask(__name__)
5
6 symptom_list = ["ngengat", "sayap depan berwarna kelabu gelap", "telur terletak pada ujung daun", "telur berwarna putih", "telur berbentuk bulat", "sa
7 "pupa berwarna coklat gelap terbentuk dalam tanah", "sayap berumbai-umbai", "telur berwarna kekuningan", "daun berwarna putih keperak
8 "adanya liang korokan larva yang berkelok-kelok", "daun berwarna putih keperak-perakan", "bercak berukuran kecil", "bercak melekok ke
9 "suhu lembab", "umbi membusuk dan berair", "terlihatnya bercak berwarna putih pada daun", "terbentuklah koloni konidia yang berwarna m
10 "spora yang terlihat sebagai bulu-bulu halus berwarna ungu", "daun berwarna hijau pucat", "umbi menjadi berwarna coklat", "daun yang m
11 "pada dasar umbi terlihat cendawan yang berwarna keputih-putihan", "bercak-bercak berwarna putih kekuning-kuningan", "daun tampak bela
12 "bercak klorosis berwarna kuning pucat", "terdapat bintik-bintik"]
13
14 @app.route("/", methods=["GET", "POST"])
15 def index():
16     result = None
17     if request.method == "POST":
18         selected_symptoms = request.form.getlist("symptoms")
19         result = forward_chaining(selected_symptoms)
20         return render_template("index.html", symptoms=symptom_list, result=result)
21
22 if __name__ == "__main__":
23     app.run(debug=True)

```

Pada file index.html berisi kode desain program. Kode program file index.html yaitu:

```

1 <!-- templates/index.html -->
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Expert System - Ulat Grayak</title>
6 </head>
7 <body>
8   <h1>Identifikasi Ulat Grayak</h1>
9   <form method="POST">
10    <h3>Pilih Gejala yang Diketahui:</h3>
11    {% for symptom in symptoms %}
12      <label>
13        <input type="checkbox" name="symptoms" value="{{ symptom }}"> {{ symptom }}
14      </label><br>
15    {% endfor %}
16    <br>
17    <button type="submit">Diagnosa</button>
18  </form>
19
20  {% if result %}
21    <h2>Hasil Inferensi:</h2>
22    <ul>
23      {% for item in result %}
24        <li>{{ item }}</li>
25      {% endfor %}
26    </ul>
27  {% endif %}
28 </body>
29 </html>

```

Proses eksekusi program dimulai dengan menjalankan file metode.py terlebih dahulu

```

In [2]: runfile('C:/Users/FX506HC/Sistem Pakar/
metode.py', wdir='C:/Users/FX506HC/Sistem Pakar')

```

Jika file metode.py telah berhasil dijalankan, kemudian file app.py dijalankan

```

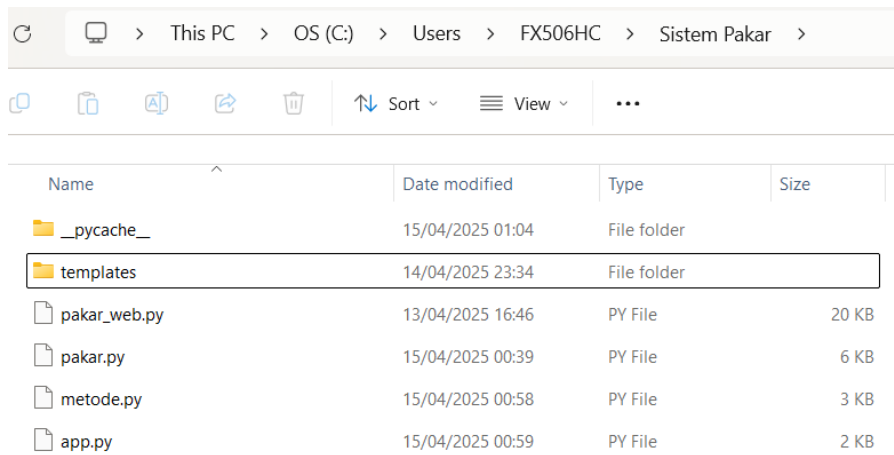
In [3]: runfile('C:/Users/FX506HC/Sistem Pakar/
app.py', wdir='C:/Users/FX506HC/Sistem Pakar')
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it
in a production deployment. Use a production WSGI
server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
An exception has occurred, use %tb to see the full
traceback.

SystemExit: 1

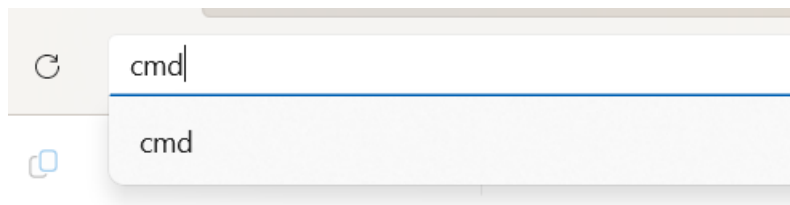
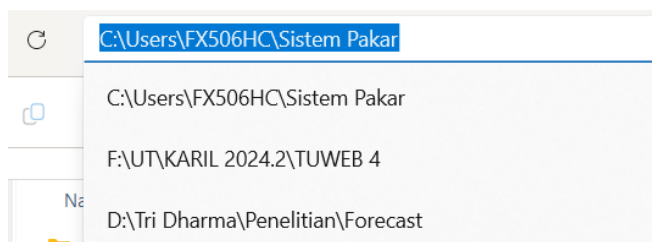
```

Hasil running file app.py menyatakan jika program dijalankan melalui alamat <http://127.0.0.1:5000> Sebelum menjalankan alamat diatas melalui browser, sebuah command prompt harus dijalankan terlebih dahulu. Untuk menjalankan command prompt sesuai alamat folder sistem pakar adalah dengan cara berikut:

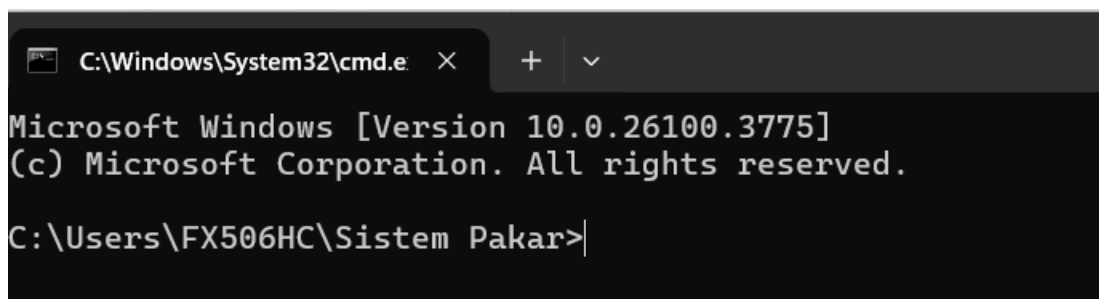
- Masuk ke folder sistem pakar melalui file explorer



- Kemudian pada bagian alamat file ketik cmd



- Klik enter dan sebuah command prompt akan muncul dengan alamat folder yang diinginkan



Pada command prompt ketik python app.py kemudian klik enter.


```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\FX506HC\Sistem Pakar>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
n WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 845-849-055
```

Kemudian buka alamat <http://127.0.0.1:5000> melalui browser

Sistem Pakar Hama dan P

VPN 127.0.0.1:5000

Modul Math for RPL Forecast RPL Akreditasi Jobsheet DTS NLP Image

Identifikasi Gejala

Pilih Gejala yang Diketahui:

- ☐ ngengat
- ☐ sayap depan berwarna kelabu gelap
- ☐ telur terletak pada ujung daun
- ☐ telur berwarna putih
- ☐ telur berbentuk bulat
- ☐ sayap depan berwarna gelap dengan garis putih
- ☐ pupa berwarna coklat gelap terbentuk dalam tanah
- ☐ sayap berumbai-umbai
- ☐ telur berwarna kekuningan
- ☐ daun berwarna putih keperak-perakan
- ☐ bintik-bintik putih
- ☐ adanya liang korokan larva yang berkelok-kelok
- ☐ daun berwarna putih keperak-perakan
- ☐ bercak berukuran kecil
- ☐ bercak melekuk ke dalam
- ☐ berwarna putih dengan pusat yang berwarna ungu
- ☐ suhu lembab
- ☐ umbi membusuk dan berair
- ☐ terlihatnya bercak berwarna putih pada daun
- ☐ terbentuklah koloni konidia yang berwarna merah muda
- ☐ adanya miselia yang tumbuh menjalar dari helaian daun
- ☐ spora yang terlihat sebagai bulu-bulu halus berwarna ungu
- ☐ daun berwarna hijau pucat
- ☐ umbi menjadi berwarna coklat
- ☐ daun yang menguning
- ☐ daun terpelintir
- ☐ pada dasar umbi terlihat cendawan yang berwarna keputih-putihan
- ☐ bercak-bercak berwarna putih kekuning-kuningan
- ☐ daun tampak belang-belang

Pilih gejala yang ditemukan pada objek penelitian (pada modul ini adalah tanaman bawang merah).

- ☒ suhu lembab
- ☐ umbi membusuk dan berair
- ☐ terlihatnya bercak berwarna putih pada daun
- ☐ terbentuklah koloni konidia yang berwarna merah muda
- ☐ adanya miselia yang tumbuh menjalar dari helaian daun
- ☐ spora yang terlihat sebagai bulu-bulu halus berwarna ungu
- ☐ daun berwarna hijau pucat
- ☐ umbi menjadi berwarna coklat
- ☐ daun yang menguning
- ☐ daun terpelintir
- ☐ pada dasar umbi terlihat cendawan yang berwarna keputih-putihan
- ☐ bercak-bercak berwarna putih kekuning-kuningan
- ☒ daun tampak belang-belang
- ☒ bercak klorosis yang berbentuk bulat
- ☒ bercak klorosis berwarna kuning pucat
- ☒ terdapat bintik-bintik

Kemudian hasil inferensi akan tampil setelah tombol diagnosa di klik.

Diagnosa

Hasil Inferensi:

- Serkospora

3. Rangkuman

Sistem pakar merupakan salah satu cara untuk menampilkan pengetahuan (representasi pengetahuan). Sistem pakar menggunakan aturan produksi untuk merancang aturan dalam basis pengetahuan. Sistem pakar terdiri dari tiga bagian utama, yaitu interface, mesin inferensi dan basis pengetahuan. Secara umum sistem pakar memiliki dua metode, yaitu forward chaining dan backward chaining. Pada forward chaining proses penalaran dimulai dari gejala yang kemudian akan menghasilkan kesimpulan. Sedangkan pada metode backward chaining proses penalaran dilakukan dari kesimpulan yang akan tertuju pada gejala yang telah ada sebelumnya. Aplikasi sistem pakar berfungsi untuk membantu orang awam (non pakar) untuk mendiagnosa atau menghasilkan sebuah kesimpulan dari gejala-gejala yang dialami.

4. Tes Formatif

5. Daftar Pustaka

- [1] S. Suyanto, Artificial Intelligence: Searching, Reasoning, Planning, Learning, Bandung: Informatika, 2014.
- [2] R. A. Putri, "Modul Kuliah Sistem Pakar," Universitas Islam Negeri Sumatera Utara, Medan, 2020.

- [3] M. "Pengenal Sistem Pakar," -, -, 2020.
- [4] S. "Kecerdasan Buatan Sistem Pakar," Politeknik Elektronika Negeri Surabaya, Surabaya, 2017.
- [5] Tessy, "Modul Kuliah Sistem Pakar," Politeknik Elektronika Negeri Surabaya, Surabaya, 2022.
- [6] B. K. Udiarto, W. Setiawati and E. Suryaningsih, Pengenal Hama dan Penyakit pada Tanaman Bawang Merah dan Pengendaliannya, Bandung: Balai Penelitian Tanaman Sayuran, 2005.
- [7] R. Rosnelly, Sistem Pakar Konsep dan Teori, Yogyakarta: ANDI, 2012.
- [8] Teams, "Flask Installation," Flask, 2010. [Online]. Available: <https://flask.palletsprojects.com/en/stable/installation/>. [Accessed 15 04 2025].