

Identifikasi Kombinasi Penyakit Jerawat pada Wajah Berdasarkan Jumlah Diagnosis Menggunakan Metode Forward Chaining

Muhammad Abiyyu Habibi (G64180066), Muhammad Faris Waliyuddin (G64180067),
Annisa Faradila (G64180074), Reza Achmad Naufal (G64180078), dan Alvin Ferdiansyah (G64180079)

*Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut
Pertanian Bogor*

Abstraksi

Permasalahan pada kulit salah satunya berkaitan dengan penyakit jerawat. Jerawat merupakan salah satu jenis penyakit yang umum terjadi pada wajah manusia. Dibangunnya sistem pakar yang mengidentifikasi kombinasi penyakit jerawat pada wajah manusia memudahkan individu untuk lebih mengenali permasalahan pada kulit wajah manusia. Penelitian ini dibagi menjadi tiga tahap. Tahap pertama adalah pengumpulan data dengan mengambil data mentah dari jurnal lain. Pada tahap ini, didapatkan hasil mencakup gejala gejala, jenis penyakit jerawat, dan gejala pada tiap jenis penyakit. Tahap kedua ialah proses analisis pada data yang diperoleh. Data tersebut diatur sebagai variabel, kemudian disusun dalam sebuah aturan untuk mencapai tujuan dari kesimpulan. Aturan disusun dengan menggunakan metode *forward-chaining*, dan diimplementasikan menggunakan bahasa pemrograman python. Tahap ketiga adalah implementasi yang dibuat dalam bentuk tampilan pengguna aplikasi berbasis *mobile*. Implementasi hanya menampilkan contoh kasus tampilan pengguna ketika *user* menggunakan aplikasi dan ketika program menampilkan kesimpulan setelah data input diproses pada sistem inferensi.

Kata Kunci : Identifikasi Kombinasi. *forward chaining*, Penyakit jerawat, *Knowledge Base*

1. Pendahuluan

Kulit merupakan organ terluar yang melapisi tubuh manusia. Kulit menyumbang 15%

dari total berat badan. Ada pori-pori (lesung pipit) di permukaan luar kulit, tempat keluarnya keringat. Kulit memiliki banyak fungsi, diantaranya sebagai pelindung tubuh, sebagai alat peraba atau komunikasi, dan sebagai alat untuk mengontrol suhu (Puteri, A. G., & Bhakti, R. 2019).

Permasalahan pada kulit salah satunya berkaitan dengan penyakit jerawat. Jerawat merupakan salah satu jenis penyakit yang umum terjadi pada wajah manusia. Jerawat sendiri memiliki banyak jenis penyakit dengan efek dan dampak yang berbeda dan tidak jarang wajah manusia mengalami lebih dari 1 jenis penyakit secara bersamaan.

Dibangunnya sistem pakar yang dapat mengidentifikasi kombinasi penyakit jerawat pada wajah manusia guna memudahkan individu untuk lebih mengenali permasalahan pada kulit wajah manusia. Dengan menggunakan metode *forward chaining* disertai set aturan inferensi dan argumen yang valid, aplikasi diharapkan dapat memberikan kesimpulan berdasarkan gejala gejala yang diinput oleh pengguna.

2. Landasan Teori

Inferensi dapat diartikan sebagai sebuah proses yang digunakan dalam sistem pakar untuk menghasilkan sebuah kesimpulan logis dari *Knowledge Base* atau premis premis terhadap informasi yang diperoleh.

Knowledge Base (Basis pengetahuan) berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Basis pengetahuan ini juga berisi tentang

aturan-aturan yang berkaitan dengan pengetahuan tersebut. Dalam proses ini, pengetahuan direpresentasikan menjadi basis pengetahuan dan basis aturan selanjutnya dikodekan, dikumpulkan, dan dibentuk secara sistematis. (Sasmito dan Khomsah 2011).

Mesin inferensi untuk basis pengetahuan memiliki komponen dasar sebagai berikut :

Atribut: $X_1, X_2, X_3, \dots, X_{n1}$
 Kondisi: $C_1, C_2, C_3, \dots, C_{n2}$
 Aturan: $R_1, R_2, R_3, \dots, R_{n3}$
 Aksi: $A_1, A_2, A_3, \dots, A_{n4}$

Atribut adalah sebuah variabel yang digunakan untuk menyimpan sebuah nilai. Atribut dapat memberikan nilai dalam kesimpulan sebuah aturan. Atribut juga memberikan nilai pada input dari pengguna. Atribut dapat dibandingkan dengan setiap nilai atau setiap premis (dasar pikiran) dengan menggunakan operator relasi.

Sedangkan, Aturan dinyatakan dalam bentuk:

IF <kondisi> THEN <aksi>
IF <premis> THEN <konklusi>
IF <premis 1> AND <premis 2> AND <premis ke-n> THEN <kesimpulan>
 (Adriyendi. 2017)

Terdapat beberapa metode dan algoritma yang secara umum digunakan dalam sebuah proses inferensi, yaitu *Resolution Algorithm* , *Forward Chaining*, dan *Backward Chaining*.

Resolution Algorithm

Resolution merupakan sebuah *rule* dari inference yang mengarah ke pembuktian menggunakan konsep *contradiction*.

Sebuah *sentence* dapat kita katakan valid, apabila bernilai benar dalam semua model. Sebuah *sentence* dapat kita katakan *satisfiable*, apabila bernilai benar dalam sebagian model. dan sebuah *sentence* dapat kita katakan *unsatisfiable*, apabila bernilai tidak bernilai benar dalam model apapun. Dengan konsep *proof by contradiction*, alpha dikatakan sebagai intel dari *Knowledge Base* jika dan hanya jika *Knowledge Base* dan negasi alpha *is unsatisfiable* .

Dengan kata lain, kita bisa secara interaktif menerapkan *resolution rule* dengan cara yang sesuai untuk menunjukkan apakah sebuah *propositional formula is satisfiable* dan membuktikan *first-order formula is unsatisfiable*.

Algoritma resolution rule dapat kita tulis sebagai pseudocode berikut :

```
function PL_Resolution (KB, alpha) returns true or false
  clauses <- the set of clauses in the CNF representation of  $KB \wedge \neg \alpha$ 
  new <- {}
  loop do
    for each  $C_i, C_j$  in clauses do
      resolvents <- PL-Resolve( $C_i, C_j$ )
      if resolvents contains the empty clause
        then return true
    new <- new  $\cup$  resolvents
  if new C clauses then return false
  clauses <- clauses  $\cup$  new
```

Forward Chaining

Algoritma *forward-chaining* sebagai salah satu metode inferensi secara logis dapat dideskripsikan sebagai aplikasi pengulangan dari modus ponens (satu set aturan inferensi dan argumen yang valid).

Forward-chaining mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Mesin inferensi yang menggunakan *forward-chaining* mencari aturan-aturan inferensi sampai menemukan satu dari antecedent (dalil hipotesa atau klausa IF -THEN) yang benar. Ketika aturan tersebut ditemukan maka mesin pengambil keputusan dapat membuat kesimpulan, atau konsekuensi (klausa THEN), yang menghasilkan informasi tambahan yang baru dari data yang disediakan. Mesin akan mengulang melalui proses ini sampai sasaran ditemukan. *Forward-chaining* adalah contoh konsep umum dari pemikiran yang dikendalikan oleh data (data-driven) yaitu, pemikiran yang mana fokus perhatiannya dimulai dari data yang diketahui (Akil 2017)

Pseudocode *Forward Chaining* dapat adalah sebagai berikut :

```
function PL-FC(KB, q) return true or false
  local variables :

  while agenda is not empty do
    p <- POP(agenda)
    unless inferred[p] do
      inferred[p] <- true
```

```

    for each Horn clause c in whose
    premise p appears do
        decrement count[c]
        if count[c] = 0 then do
            if HEAD[c] = q then return
true
        push(HEAD[c],agenda)
return false

```

Backward Chaining

Algoritma *backward-chaining* merupakan metode lainnya dalam menerapkan sebuah proses inferensi. Algoritma ini bekerja mundur dari query-nya. Jika query q diketahui adalah benar, maka tak ada yang perlu dikerjakan selanjutnya. Selain itu, algoritmanya akan mencari implikasi-implikasi di dalam basis data pengetahuan atau *Knowledge Base* yang kesimpulannya adalah q . Jika semua premis-premis dari salah satu implikasi-implikasi tersebut bisa dibuktikan benar, maka q adalah benar (Russel dan Norvig 2010).

Jelas sekali disini, bahwa *backward-chaining* menggunakan algoritma pencarian depth-first. *Backward-chaining* adalah sebuah bentuk pemikiran yang dikendalikan oleh tujuan atau goal (Akil 2017).

Pseudocode *Backward Chaining* dapat adalah sebagai berikut :

```

function FOL-BC-ASK(KB, query) returns a
generator of substitutions return FOL-BC-OR (
KB, query, F)

generator FOL-BC-OR(KB, goal, 0) yields a
substitution
    for each rule (/is rhs) in
    FETCH-RULES-FOR-GOAL(KB, goal) do
        (lhs, rise) ← STANDARDIZE
        VARIABLES((lhs, rhs))
        for each 0' in FOL-BC-AND(KB, the,
        UNIFY(rhs, goad, 0)) do
            yield 9'

generator FOL-BC-AND(KB, goals, 9) yields a
substitution
    if 0 = failure then return
    else if LENGTH(goals) = 0 then yield 0
    else do
        first,rest FIRST(goalS).
        REST(goals)

```

```

for each 0' in
FOL-BC-OR(KB, SUBST(0, first), 0)
do
    for each 0'' in
    FOL-BC-AND(KB, rest, 0') do
        yield 8''

```

3. Metode

Penelitian dibagi menjadi 3 tahap, yaitu :
(1) Pengumpulan data, (2) Analisa dan Perancangan, (3) Implementasi.

Pada tahap pertama, pengumpulan data dilakukan dengan mengambil data mentah dari jurnal lain terkait jenis penyakit jerawat pada wajah. (Kusbianto *et al.* 2017). Pada tahap ini, didapatkan hasil mencakup gejala gejala, jenis penyakit jerawat, dan gejala pada tiap jenis penyakit.

Pada tahap kedua, dilakukan analisis pada data data yang diperoleh. Data tersebut diatur sebagai variabel, kemudian disusun dalam sebuah aturan (*rule*) untuk mencapai tujuan dari kesimpulan. Aturan disusun dengan menggunakan metode *forward-chaining* , dan diimplementasikan menggunakan bahasa pemrograman python. Terakhir, pada tahap ini disusun rancangan alur program untuk implementasi sistem.

Pada tahap ketiga implementasi dibuat dalam bentuk tampilan pengguna aplikasi berbasis *mobile*. Implementasi hanya menampilkan contoh kasus tampilan pengguna ketika *user* menggunakan aplikasi dan ketika program menampilkan kesimpulan setelah data input diproses pada sistem inferensi.

4. Hasil dan Pembahasan

A. Data dan Hasil Analisa

Kombinasi penyakit jerawat pada wajah (K1) memiliki 28 variabel berupa gejala gejala pasien sebagai *Knowledge Base* (KB).

Kode Gejala	Gejala
G1	Muncul papul
G2	Muncul pustule
G3	Muncul nodul
G4	Muncul sikatrik atau scar

	acne
G5	Predileksinya pada area wajah, dada dan punggung
G6	Pada umumnya muncul pada usia remaja
G7	Kulit perih dan sensasi terbakar
G8	Kemerahan pada kulit yang permanen
G9	Permukaan kulit menjadi kasar, seperti membengkak
G10	Masalah pada mata (mata bengkak, kelopak mata merah)
G11	Predileksinya pada sentral wajah yaitu hidung, pipi, dagu, kening dan alis
G12	Pada umumnya ditemukan pada usia 30-40 tahun
G13	Bintil kecil pada lipatan dagu atau bagian bibir atas
G14	Kulit berwarna merah dan bersisik
G15	Predileksinya pada area mulut, bisa menyebar disekitar hidung dan mata
G16	Pada umumnya ditemukan pada wanita muda
G17	Ada keluhan gatal
G18	Muncul warna kemerahan yang menyebar ke alis dan glabella
G19	Sekumpulan benjolan merah atau benjolan benjolan kecil berisi nanah yang berkembang di sekitar folikel rambut
G20	Predileksinya pada area punggung, bahu, dan dada bagian atas, bisa meluas sampai ke leher, lengan atas dan wajah

G21	Pada umumnya ditemukan pada laki-laki atau perempuan usia 13-45 tahun
G22	Permukaan kulit kasar, tidak rata atau bersisik
G23	Predileksinya pada area kulit lengan, paha, pipi, bokong. Bisa muncul di wajah, alis atau kulit kepala
G24	Pada umumnya ditemukan pada anak-anak dan remaja
G25	Benjolan yang bengkak, besar dan bernanah
G26	Pada umumnya ditemukan pada laki-laki dewasa
G27	Peradangan pada wajah dan leher
G28	Pada umumnya ditemukan pada segala umur

Tabel 1.0 Kode Gejala

Kombinasi penyakit jerawat pada wajah (K1) memiliki variabel lanjutan berupa jenis penyakit, yang terdiri dari 7 jenis penyakit. Variabel variabel ini didiagnosis berdasarkan gejala gejala yang dialami pasien atau *Knowledge Base*.

Kode Penyakit	Nama Penyakit
P1	Acne Vulgaris
P2	Rosacea
P3	Perioral dermatitis
P4	Pityrosporum folliculitis
P5	Keratosis pilaris
P6	Gram-negative bacterial folliculitis
P7	Pseudofolliculitis

Tabel 1.1 Nama Penyakit

Diagnosis penyakit berdasarkan gejala dapat dilihat pada tabel berikut :

Kode Penyakit	Gejala yang dialami
P1	G01,G02, G03,G04,G05,G06
P2	G01, G02, G07, G08, G09, G10, G11, G12
P3	G01, G02, G13, G14, G15, G16
P4	G01, G02, G17, G18, G19, G20, G21
P5	G01, G22, G23, G24
P6	G01, G02, G13, G25, G26
P7	G01, G02, G27, G28

Tabel 1.1 Gejala pada tiap penyakit

B. Penerapan metode Forward Chaining

Dalam mencapai goal, yaitu kombinasi penyakit jerawat pada wajah (K1), disusun aturan aturan pada tabel 1.2 dengan metode forward chaining :

No.	Aturan (Rule)
1.	If G01 and G02 and G03 and G04 and G05 and G06 then P1
2.	If G01 and G02 and G07 and G08 and G09 and G10 and G11 and G12 then P2
3.	If G01 and G02 and G13 and G14 and G15 and G16 then P3
4.	If G01 and G02 and G17 and G18 and G19 and G20 and G21 then P4
5.	If G01 and G22 and G23 and G24 then P5
6.	If G01 and G02 and G13 and G25 and G26 then P6
7.	If G01 and G02 and G27 and G28 then P7
8.	If (P1 and P2) or (P1 and P3) or (P1 and P4) then K1
9.	If (P1 and P5) or (P1 and P6) or

	(P1 and P7) then K1
10.	If (P2 and P3) or (P2 and P4) or (P2 and P5) then K1
11.	If (P3 and P4) or (P3 and P5) or (P3 and P6) then K1
12.	If (P4 and P5) or (P4 and P6) or (P4 and P7) then K1
13.	If (P2 and P6) or (P5 and P6) or (P5 and P7) then K1
14.	If (P2 and P7) or (P3 and P7) or (P6 and P7) then K1

Tabel 1.2 Rule

Kombinasi penyakit jerawat akan mencapai goal ketika ditemukan lebih dari 1 penyakit yang terbukti benar.

Untuk mengimplementasikan pada program, kami mengimplementasikan aturan aturan tersebut menggunakan bahasa python, dengan source code sebagai berikut :

```

from itertools import combinations
global Gejala
global Penyakit

def tanyakan_gejala():
    global Gejala
    for G in range(len(Gejala)):
        inp = int(raw_input("G["+str(G+1)+"] : "))
        if(inp == 1):
            Gejala[G] = 1

def diagnosis(G):
    global Penyakit
    if ( G[1-1] and G[2-1] and G[3-1] and G[4-1]
    and G[5-1] and G[6-1] ) :
        Penyakit[1-1][1] = 1
    if ( G[1-1] and G[2-1] and G[7-1] and G[8-1]
    and G[9-1] and G[1-1] and G[11-1] and G[12-1]
    ) :
        Penyakit[2-1][1] = 1
    if ( G[1-1] and G[2-1] and G[13-1] and
    G[14-1] and G[15-1] and G[16-1] ) :
        Penyakit[3-1][1] = 1
    if ( G[1-1] and G[2-1] and G[17-1] and
    G[18-1] and G[19-1] and G[2-1] and G[21-1] ) :
        Penyakit[4-1][1] = 1
    if ( G[1-1] and G[22-1] and G[23-1] and
    G[24-1] ) :

```

```

Penyakit[5-1][1] = 1
if ( G[1-1] and G[2-1] and G[13-1] and
G[25-1] and G[26-1] ) :
    Penyakit[6-1][1] = 1
    if ( G[1-1] and G[2-1] and G[27-1] and
G[28-1] ) :
        Penyakit[7-1][1] = 1

def Cek_Kombinasi_Penyakit(P):
    global Kesimpulan
    kombinasi = combinations(P,2)
    for k in kombinasi :
        if(k[0][1] and k[1][1]):
            Kesimpulan = 1

Gejala = [0]* 28
Penyakit = [ ["Acne Vulgaris",0],["Rosacea",0],["Perioral dermatitis",0],["Pityrosporum folliculitis",0],["Keratosis pilaris",0],["Gram-negative bacterial folliculitis",0],["Pseudofolliculitis",0] ]
Kesimpulan = 0

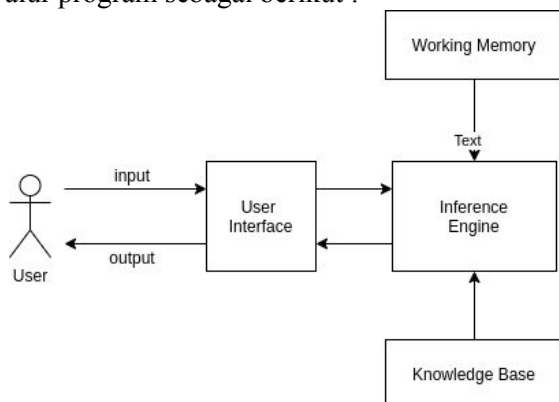
tanyakan_gejala()
diagnosis(Gejala)
for i in Penyakit:
    if(i[1]):
        print("Anda di diagnosis memiliki penyakit "+i[0])

Cek_Kombinasi_Penyakit(Penyakit)
if(Kesimpulan):
    print("Anda MEMILIKI kombinasi penyakit jerawat pada wajah")
else:
    print("Anda TIDAK MEMILIKI kombinasi penyakit jerawat pada wajah")

```

C. Perancangan Sistem

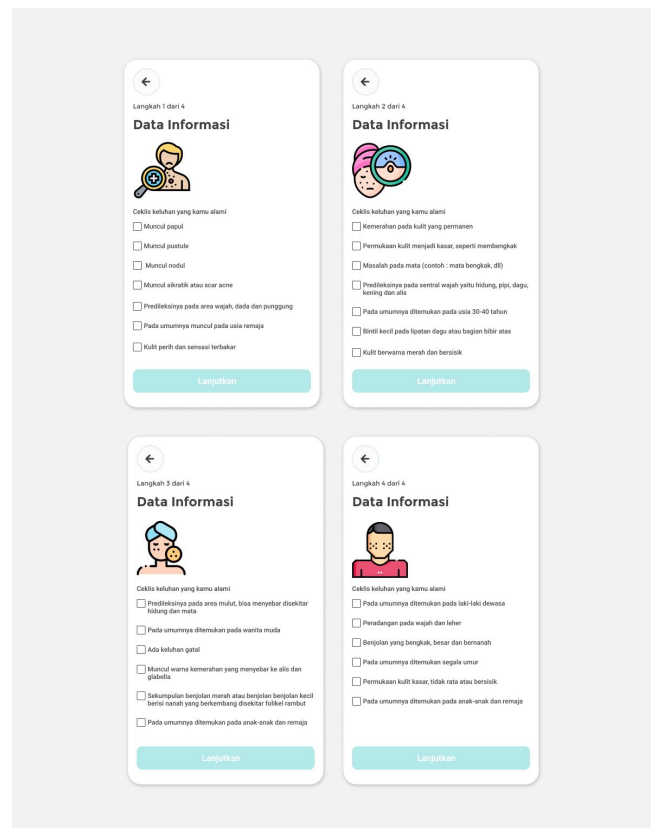
Aturan inferensi yang telah diimplementasikan pada program kemudian dilanjut kembangkan menjadi sebuah aplikasi berbasis *mobile*, dengan alur program sebagai berikut :



Gambar 1.0 Alur Program

User diminta untuk memasukkan input dengan melakukan ceklis pada gejala-gejala yang dialami melalui tampilan pengguna. Program kemudian menjalankan aturan-aturan inferensi dengan input user sebagai fakta. Nilai sementara disimpan pada *working memory* untuk kembali diolah melalui sistem inferensi. Hasil akhir program berupa kesimpulan yang telah didapatkan, yang kemudian ditampilkan sebagai output melalui tampilan pengguna.

D. Hasil Implementasi



Gambar 1.0 tampilan pengguna aplikasi

The input form consists of four steps, each titled 'Data Informasi' and 'Langkah X dari 4'.

- Langkah 1 dari 4:**
 - Ceklis keluhan yang kamu alami:
 - ☒ Muncul papul
 - ☒ Muncul pustule
 - ☐ Muncul nodul
 - ☐ Muncul ulserasi atau scar acne
 - ☐ Predileksinya pada area wajah, dada dan punggung
 - ☐ Pada umumnya muncul pada usia remaja
 - ☐ Kulit perih dan sensasi terbakar
- Langkah 2 dari 4:**
 - Ceklis keluhan yang kamu alami:
 - ☐ Kemerahan pada kulit yang permanen
 - ☐ Perusakan kulit menjadi kasar, seperti membingkai
 - ☐ Masalah pada mata (contoh: mata bengkak, dll)
 - ☐ Predileksinya pada sekitar wajah yaitu hidung, pipi, dagu, kening dan alis
 - ☐ Pada umumnya ditemukan pada usia 30-40 tahun
 - ☒ Benit kecil pada lipatan dagu atau bagian bibir atas
 - ☒ Kulit berwarna merah dan berisik
- Langkah 3 dari 4:**
 - Ceklis keluhan yang kamu alami:
 - ☒ Predileksinya pada area mulut, bisa menyebar disekitar hidung dan mata
 - ☒ Pada umumnya ditemukan pada wanita muda
 - ☐ Ada keluhan gatal
 - ☐ Muncul warna kemerahan yang menyebar ke alis dan gigitan
 - ☐ Sekelompok benjolan merah atau benjolan benjolan kecil berisi nanah yang berkembang disekitar folikel rambut
 - ☐ Pada umumnya ditemukan pada anak-anak dan remaja
- Langkah 4 dari 4:**
 - Ceklis keluhan yang kamu alami:
 - ☒ Pada umumnya ditemukan pada laki-laki dewasa
 - ☐ Peradangan pada wajah dan leher
 - ☐ Benjolan yang bengkak, besar dan bernanah
 - ☐ Pada umumnya ditemukan segala umur
 - ☐ Perusakan kulit kasar, tidak rata atau berisik
 - ☐ Pada umumnya ditemukan pada anak-anak dan remaja

Gambar 1.1 User memasukkan input

The output form consists of two screens, each titled 'Selesai'.

- Screen 1:**
 - Kamu memiliki **kombinasi** penyakit jerawat pada wajah
 - Kamu memiliki kombinasi dari penyakit jerawat seperti *Acne Vulgaris*, *Keratosis pilaris* dan *Pseudofolliculitis*
- Screen 2:**
 - Kamu **tidak memiliki** kombinasi penyakit jerawat pada wajah
 - Kamu tidak memiliki kombinasi dari penyakit jerawat dan kamu hanya memiliki satu penyakit yaitu *Pseudofolliculitis*

Gambar 1.2 Program menampilkan output

5. Kesimpulan

Beberapa kesimpulan yang dapat diambil dari penelitian ini. Pertama, *forward chaining*

dapat dijadikan model dalam identifikasi kombinasi penyakit jerawat pada wajah dengan logika *and* dan *or*. Kedua, penelitian ini menghasilkan 14 *rule* dalam memodelkan proses identifikasi kombinasi penyakit jerawat pada wajah menggunakan model penelusuran *forward chaining*. Ketiga, penelitian ini menggunakan tujuh diagnosis penyakit, yaitu *Acne Vulgaris*, *Rosacea*, *Perioral dermatitis*, *Pityrosporum folliculitis*, *Keratosis pilaris*, *Gram-negative bacterial folliculitis*, dan *Pseudofolliculitis* dengan 28 gejala pasien.

6. Saran

Penelitian ini hanya mengidentifikasi kombinasi penyakit jerawat pada wajah berdasarkan jumlah diagnosis. Penelitian ini masih jauh dari kata sempurna untuk menggambarkan keadaan sebenarnya. Sebagai akhir dari penelitian, kami memiliki saran-saran yang mungkin dapat berguna jika ada yang ingin menggunakan sistem ini. Diharapkan dikembangkan lagi sistem ini dengan menggunakan data yang lebih banyak disertai perbandingan dengan diagnosa dari para ahli sehingga bisa mendapatkan hasil diagnosa yang lebih baik.

7. Daftar Pustaka

- [1] Adriyendi. 2017. Inference menggunakan forward chaining pada food affordability. *Journal of saintek*. 9(2): 108-12.
- [2] Agus Sasmito Aribowo, Siti Khomsah. 2011. Sistem pakar dengan beberapa knowledge base menggunakan probabilitas bayes dan mesin inferensi forward chaining. *Seminar Nasional Informatika 2011 (semnasIF 2011)*. 1(4).
- [3] Kusbianto, Ardiansyah, Hamadi. 2017. Implementasi sistem pakar forward chaining untuk identifikasi dan tindakan perawatan jerawat wajah. *Jip*. 4(1): 71.
- [4] Ibnu Akil. 2017. Analisa efektifitas metode forward chaining dan backward chaining pada sistem pakar. *Jurnal Pilar Nusa Mandiri*. 13(1).
- [5] Indyah Hartami Santi, Bina Andari. 2019. Expert systems to identify facial skin types with the certainty factor method. *INTENSIF*. 3(2).
- [6] Febi Nur Salisah, Leony Lidya, Sarjon Defit. 2015. Sistem pakar penentuan bakat anak dengan menggunakan metode forward

chaining. *Jurnal Rekayasa dan Manajemen Sistem Informasi*. 1(1): 62-66.

- [7] Puteri, A. G., Bhakti, R.. 2019. Penggunaan certainty factor dalam sistem pakar diagnosa penyakit jerawat. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*. 1(2): 86-96.
- [8] Russel, S., Norvig, P.. 2010. Artificial intelligent a modern approach third edition. New Jersey: Pearson Education.
- [9] Setiadi. 2007. *Anatomi dan Fisiologi Manusia*. Yogyakarta (ID): Graha Ilmu.