



**PENERAPAN *BREADTH FIRST SEARCH ALGORITHM* DAN  
*CERTAINTY FACTOR THEORY* PADA DIAGNOSA  
PENYAKIT GIGI DAN MULUT**

**SKRIPSI**

Oleh

**Falviana Yulia Dewi  
NIM 121810101026**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2016**



**PENERAPAN *BREADTH FIRST SEARCH ALGORITHM* DAN  
*CERTAINTY FACTOR THEORY* PADA DIAGNOSA  
PENYAKIT GIGI DAN MULUT**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Matematika (S1)  
dan mencapai gelar Sarjana Sains

Oleh

**Falviana Yulia Dewi**  
**NIM 121810101026**

**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS JEMBER**  
**2016**

## PERSEMBAHAN

Alhamdulillah, dengan memanjatkan puji syukur kehadiran Allah SWT, dan dengan segala ketulusan serta kerendahan hati, saya persembahkan skripsi ini kepada:

- a. Ayahanda Hari Jujud Cahyono dan Ibunda Enik Erna Wati, yang telah dengan ikhlas memberikan untaian dzikir dan doa, dukungan dan curahan kasih sayang, serta pengorbanan sejak kecil sampai sekarang ini;
- b. Kakak Ayu Priyantini A.W, adik-adik Novia Tri R.W, Mawaddatur R dan A. Rizki Rahman yang telah memberikan hiburan maupun untaian doa;
- c. M. Ziaul Arif, S.Si., M.Sc selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni S.Si., M.Kom selaku Dosen Pembimbing Anggota yang telah memberikan bantuan dan bimbingan secara intensif untuk penyempurnaan skripsi ini;
- d. May Hendra Satria, yang dengan sabar mendidik saya untuk menjadi pribadi yang kuat dan tidak mudah mengeluh;
- e. Unit Kegiatan Mahasiswa Seni Titik dan HIMATIKA “GeoKomStat” yang telah memberi dukungan moril dan pengalaman;
- f. Guru-guru sejak taman kanak-kanak sampai dengan perguruan tinggi yang telah mendidik dengan penuh kesabaran dan keikhlasan;
- g. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;

## MOTTO

*Complaining will never solve the problem, stop complaining and take action!*<sup>\*)</sup>

Jangan melihat setiap kendala merupakan halangan, namun lihatlah sebagai peluang yang menantang.<sup>\*)</sup>

---

<sup>\*)</sup> <http://dewikata.blogspot.co.id/2015/04/kata-bijak-bahasa-inggris.html>

## PERNYATAAN

Saya yang bertanda tangan dibawah ini :

nama : Falviana Yulia Dewi

NIM : 121810101026

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan *Breadth First Search Algorithm* dan *Certainty Factor Theory* pada Diagnosa Penyakit Gigi dan Mulut” adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isisnya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2016

Yang menyatakan,

Falviana Yulia Dewi

NIM 121810101026

**SKRIPSI**

**PENERAPAN *BREADTH FIRST SEARCH ALGORITHM* DAN  
*CERTANTY FACTOR TEORY* PADA DIAGNOSA  
PENYAKIT GIGI DAN MULUT**

Oleh

Falviana Yulia Dewi  
NIM 121810101026

Pembimbing

Dosen Pembimbing Utama : M. Ziaul Arif, S.Si., M.Sc

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom

**PENGESAHAN**

Skripsi berjudul “Penerapan *Breadth First Search Algorithm* dan *Certainty Factor Theory* pada Diagnosa Penyakit Gigi dan Mulut” telah diuji dan disahkan pada :

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,

Sekretaris,

M. Ziaul Arif, S.Si., M.Sc  
NIP. 198501112008121002

Ahmad Kamsyakawuni S.Si., M.Kom  
NIP. 197211291998021001

Anggota I,

Anggota II,

Drs. Rusli Hidayat, M.Sc  
NIP. 196610121993031001

Dian Anggaraeni, S.Si., M.Si  
NIP. 198202162006042002

Mengesahkan  
Dekan,

Drs. Sujito, Ph.D  
NIP.196102041987111001



## RINGKASAN

**Penerapan Breadth First Search Algorithm dan Certainty factor Theory pada Diagnosa Penyakit Gigi dan Mulut;** Falviana Yulia Dewi, 121810101026; 2016; 61 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Sistem pakar merupakan cabang dari kecerdasan buatan yang dapat meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu. Sistem pakar digunakan dalam berbagai bidang, diantaranya yaitu dalam bidang kesehatan, manajerial, ekonomi dan robotika. Dalam bidang kesehatan, sistem pakar digunakan untuk melakukan diagnosa suatu penyakit.

Tujuan dari penelitian ini yaitu merancang sebuah aplikasi untuk mendiagnosa penyakit gigi dan mulut yang menggunakan *Breadth First Search Algorithm* sebagai metode pencarian. Dan menggunakan *Certainty Factor Theory* sebagai metode yang digunakan untuk menghitung tingkat keyakinan *output* yang diperoleh oleh *Breadth First Search Algorithm*.

Penelitian dilakukan melalui beberapa langkah, yaitu diawali dengan *study literature*, mengolah data yang berupa gejala-gejala penyakit gigi dan mulut, kemudian menerapkan dua metode sebagai metode pencarian dan metode perhitungan. Selanjutnya membuat program sesuai algoritma yang digunakan dengan menggunakan *software* MATLAB. Selanjutnya membuat kesimpulan dari hasil yang didapat dari penerapan kedua metode tersebut.

Hasil penelitian yang didapat dapat dilihat bahwa sistem pakar dapat dijadikan sarana untuk menyimpan sebuah pengetahuan tentang penyakit gigi dan mulut dari dokter spesialis gigi dan mulut. Selain itu juga memberikan informasi terakurat terhadap pasien mengenai penyakit yang diderita berdasarkan gejala yang dirasakan karena adanya faktor kepastian.



## PRAKARTA

Alhamdulillah, dengan memanjatkan puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia Nya sehingga skripsi yang berjudul “Penerapan *Breadth First Search Algorithm* dan *Certainty Factor Theory* pada Diagnosa Penyakit Gigi dan Mulut” dapat terselesaikan. Tujuan disusunnya skripsi ini adalah untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan dtrata 1 (S1) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

- a. M. Ziaul Arif, S.Si., M.Sc selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni S.Si., M.Kom selaku Dosen Pembimbing Anggota yang telah memberikan bantuan dan bimbingan secara intensif untuk penyempurnaan skripsi ini;
- b. Drs. Rusli Hidayat, M.Sc dan Dian Anggaraeni. S.Si., M.Si, selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun;
- c. Seluruh dosen dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
- d. Ayahanda Hari Jujud Cahyono dan Ibunda Enik Erna Wati, yang telah dengan ikhlas memberikan untaian dzikir dan doa, dukungan dan curahan kasih sayang, serta pengorbanan sejak kecil sampai sekarang ini;
- e. Kakak Ayu Priyantini A.W, adik-adik Novia Tri R.W, Mawaddatur R dan A. Rizki Rahman yang telah memberikan hiburan maupun untaian doa;
- f. May Hendra Satria yang dengan sabar mendidik saya untuk menjadi pribadi yang kuat dan tidak mudah mengeluh;
- g. Unit kegiatan Mahasiswa Seni Titik dan HIMATIKA “GeoKomStat” yang telah memberi dukungan moril dan pengalaman;

- h. Keluarga besar Bathics'12 yang selalu member dukungan dalam hal apapun;
- i. Guru-guru sejak taman kanak-kanak sampai dengan perguruan tinggi yang telah mendidik dengan penuh kesabaran dan keikhlasan;
- j. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;

Penulis menyadari bahwa dalam menyusun skripsi ini masih terdapat kekurangan baik isi maupun susunannya. Oleh karena itu, penulis menerima segala kritik dan saran demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat member manfaat bagi pembaca.

Jember, Juni 2016

Penulis

DAFTAR ISI

	<b>Halaman</b>
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSEMBAHAN</b> .....	ii
<b>HALAMAN MOTTO</b> .....	iii
<b>HALAMAN PERNYATAAN</b> .....	iv
<b>HALAMAN PEMBIMBING</b> .....	v
<b>HALAMAN PENGESAHAN</b> .....	vi
<b>RINGKASAN</b> .....	vii
<b>PRAKARTA</b> .....	viii
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiii
<b>DAFTAR LAMPIRAN</b> .....	xiv
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	3
<b>1.3 Batasan Masalah</b> .....	3
<b>1.4 Tujuan Penelitian</b> .....	4
<b>1.5 Manfaat Penelitian</b> .....	4
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	5
<b>2.1 Penyakit Gigi dan Mulut</b> .....	5
<b>2.2 Kecerdasan Buatan (<i>Artificial Intelligent</i>)</b> .....	6
2.2.1 Sistem Pakar .....	7
2.2.2 Komponen Sistem Pakar .....	9
<b>2.3 Pohon (<i>Tree</i>)</b> .....	11
<b>2.4 <i>Breadth First Search Algorithm</i></b> .....	12
<b>2.5 Peluang Komplemen Suatu Kejadian</b> .....	17

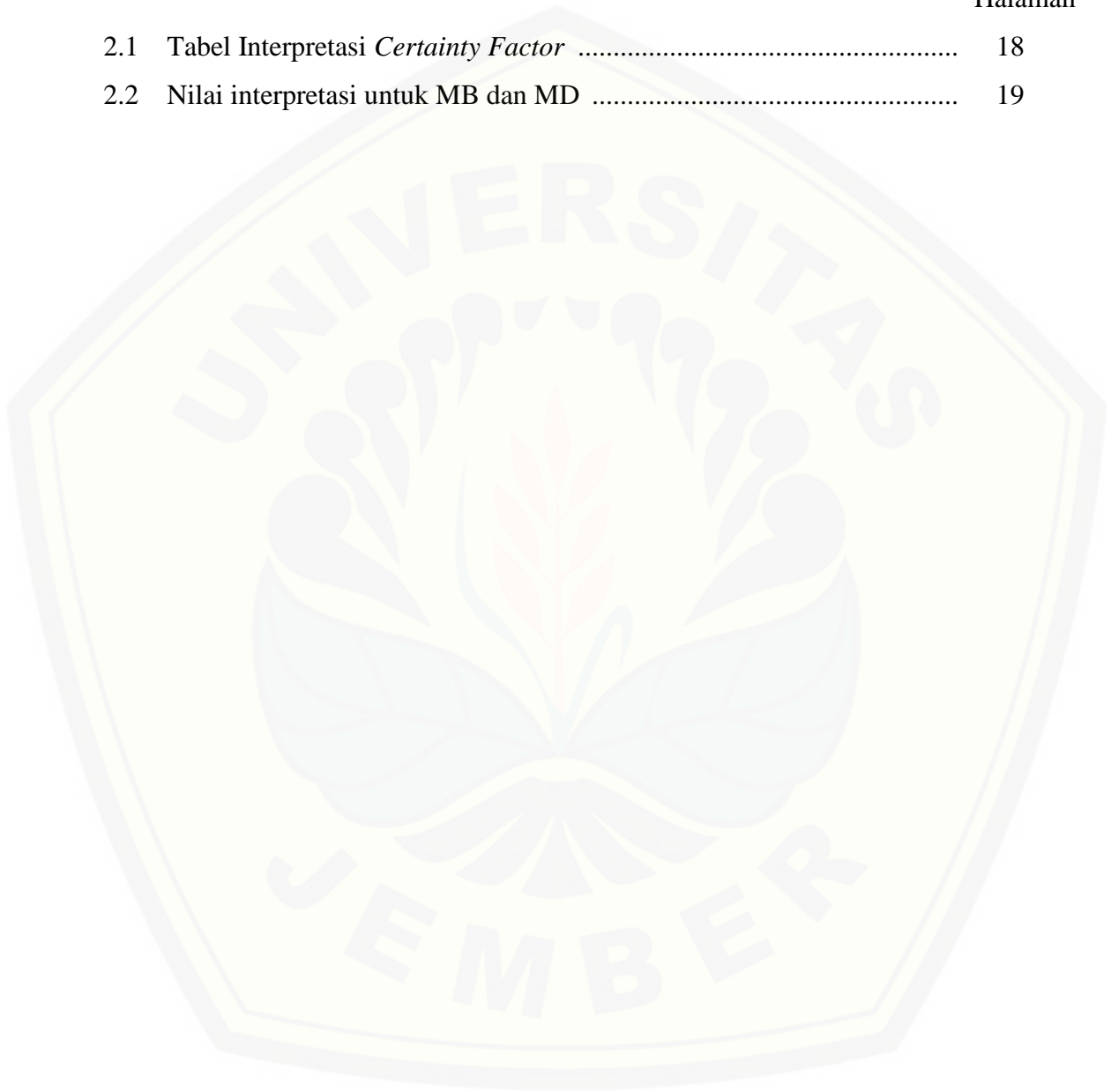
2.6 <i>Certainty Factor Theory</i> .....	17
<b>BAB 3. METODE PENELITIAN</b> .....	21
3.1 Metode Pengumpulan Data .....	21
3.2 Langkah-langkah Penyelesaian .....	21
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	23
4.1 Hasil .....	23
4.2 Pembahasan .....	27
4.2.1 Penerapan <i>Breadth First Search Algorithm</i> .....	28
4.2.2 Penerapan <i>Certainty Factor Theory</i> .....	30
<b>BAB 5. PENUTUP</b> .....	33
5.1 Kesimpulan .....	33
5.2 Saran .....	33
<b>DAFTAR PUSTAKA</b> .....	34

DAFTAR GAMBAR

	Halaman
2.1 Penerapan Konsep <i>Artificial Intelligence</i> Dalam Komputer .....	7
2.2 Komponen Sistem Pakar .....	9
2.3 Pohon ( <i>Tree</i> ) .....	11
2.4 Pencarian Melebar <i>Breadth First Search Algorithm</i> .....	12
2.5 Pohon Pencarian ( <i>Search Tree</i> ) .....	13
2.6 Langkah Awal <i>Breadth First Search</i> .....	13
2.7 Langkah Kedua <i>Breadth First Search</i> .....	14
2.8 Langkah Ketiga <i>Breadth First Search</i> .....	14
2.9 Langkah Keempat <i>Breadth First Search</i> .....	15
2.10 Langkah Keenam <i>Breadth First Search</i> .....	16
3.1 Skematik Langkah-Langkah Penelitian .....	22
4.1 Tampilan Awal Aplikasi Diagnosa Penyakit Gigi dan Mulut .....	23
4.2 Tampilan 13 Gejala Pertama .....	24
4.3 Tampilan 13 Gejala Kedua .....	25
4.4 Tampilan 13 Gejala Ketiga .....	25
4.5 Tampilan 13 Gejala Keempat .....	26
4.6 Tampilan 13 Gejala Kelima .....	26
4.7 Tampilan Hasil Aplikasi Diagnosa Penyakit Gigi dan Mulut .....	27
4.8 Pohon BFS .....	29

**DAFTAR TABEL**

	Halaman
2.1 Tabel Interpretasi <i>Certainty Factor</i> .....	18
2.2 Nilai interpretasi untuk MB dan MD .....	19



DAFTAR LAMPIRAN

	Halaman
A. Tabel Data 35 Penyakit Gigi dan Mulut Beserta Gejala .....	36
B. Tabel Data 10 Penyakit Gigi dan Mulut Beserta Cara Pengobatannya .....	39
C. Keterangan Notasi Penyakit dan Gejala .....	41
D. Bobot Beberapa Penyakit yang Memiliki Gejala Sama .....	44
E. Pohon <i>Breadth First Search</i> .....	45
F. <i>Script</i> MATLAB Aplikasi Diagnosa Penyakit Gigi dan Mulut .....	51
F.1 <i>Script</i> tombol lihat hasil .....	51
F.2 <i>Script</i> fungsi algoritma BFS .....	54
F.3 <i>Script</i> cabang fungsi algoritma BFS .....	55
F.4 <i>Script</i> fungsi CF .....	60



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Kesehatan kondisi gigi dan mulut sangatlah penting karena keberadaannya yang sangat vital. Gigi dan mulut adalah organ-organ tubuh yang ada pada wajah. Pada umumnya masyarakat umum kurang memperhatikan kesehatan gigi dan mulut karena mereka lebih mementingkan kesehatan organ-organ tubuh yang lain. Padahal penyakit gigi dan mulut juga dapat menimbulkan efek yang sangat berarti terutama saat makan atau saat berbicara. Bahkan berawal dari penyakit yang ada pada kedua organ inilah akan timbul penyakit-penyakit membahayakan yang akan menyerang organ-organ tubuh lainnya.

Keberadaan seorang pakar/ahli sangat diperlukan untuk mereka yang peduli dengan kondisi gigi dan mulut mereka. Seorang pakar/ahli dalam penyakit gigi dan mulut yaitu seorang dokter spesialis gigi dan mulut. Ada beberapa permasalahan yang terkadang dialami oleh dokter spesialis gigi dan mulut salah satunya adalah permasalahan dalam mendiagnosa penyakit. Ada beberapa penyakit gigi dan mulut yang memiliki gejala yang hampir sama sehingga terkadang terdapat perbedaan diagnosa dari beberapa dokter spesialis gigi dan mulut.

Permasalahan diagnosa penyakit gigi dan mulut saat ini dapat diselesaikan dengan bantuan komputer. Komputer merupakan perangkat yang sangat membantu manusia. Hampir semua bidang memanfaatkan komputer untuk menyelesaikan pekerjaan manusia seiring dengan kebutuhannya yang semakin banyak. Menurut Kusumadewi (2003), kecerdasan buatan atau *artificial intelligence* merupakan teknik untuk membuat komputer agar melakukan pekerjaan hampir sebaik yang dilakukan oleh manusia.

Menurut Mardiah (2015), salah satu cabang dari kecerdasan buatan yang sering digunakan yaitu sistem pakar. Sistem pakar akan meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu. Sistem pakar memiliki kemampuan untuk berfikir sendiri dan memberikan solusi terbaik dalam

domain permasalahan tertentu sesuai dengan spesifikasi yang diberikan oleh pengguna. Sistem pakar digunakan dalam berbagai bidang, diantaranya yaitu dalam bidang kesehatan, manajerial, ekonomi dan robotika. Implementasi sistem pakar diterapkan dalam dunia kesehatan sebagai media informasi bagi masyarakat awam dan juga dapat memudahkan dokter spesialis gigi dan mulut untuk mendiagnosa penyakit gigi dan mulut.

Kemampuan sistem pakar diperoleh dari seorang pakar dan diimplementasikan menjadi suatu algoritma. Terdapat sejumlah algoritma sistem pakar yang akan berpengaruh terhadap kemampuan sistem pakar tersebut dalam memberikan solusi. Algoritma yang dapat digunakan diantaranya yaitu algoritma *Depth First Search* (DFS) dan algoritma *Hill Climbing* yang pernah dibahas oleh Mardiah (2015). Dari penelitian yang telah dilakukan oleh Mardiah (2015) didapatkan bahwa apabila ditinjau dari perhitungan kompleksitas waktu yang dihasilkan, algoritma DFS dan algoritma *Hill Climbing* memiliki kompleksitas waktu yang setara yaitu kompleksitas waktu linier  $O(nm)$ . Dan algoritma *Hill Climbing* dianggap mendapatkan hasil yang lebih baik dari pada algoritma *Depth First Search* karena hasil diagnosa menggunakan algoritma *Hill Climbing* hampir sama dengan hasil diagnosa pakarnya.

Selain algoritma *Depth First Search* (DFS) dan algoritma *Hill Climbing* yang digunakan sebagai teknik pencarian, masih banyak algoritma lain yang dapat digunakan sebagai teknik pencarian untuk menyelesaikan permasalahan sistem pakar yaitu *Breadth First Search Algorithm*, metode *forward chaining*, metode *backward chaining*, metode AHP, metode *Best First Search* dan lain-lain. *Breadth First Search Algorithm* adalah teknik pencarian yang akan digunakan dalam penelitian ini. Teknik pencarian merupakan proses penelusuran yang digunakan untuk mencapai keadaan tujuan atau solusi. Proses ini berupa jalur yang menggambarkan keadaan awal sebuah masalah menuju kepada penyelesaian masalah yang diinginkan. Jalur-jalur ini menggambarkan langkah-langkah penyelesaian masalah, yang mana dalam penggunaannya digambarkan dalam visualisasi *tree*.

Selain teknik pencarian, untuk menyelesaikan masalah sistem pakar juga membutuhkan teknik perhitungan. Teknik perhitungan digunakan untuk menghitung berapa besar keyakinan *output*/hasil yang didapat dari teknik pencarian. Teknik perhitungan yang digunakan dalam penelitian ini yaitu menggunakan *Certainty Factor Theory*. Faktor kepastian (*Certainty Factor*) menyatakan kepercayaan dalam sebuah kejadian (atau fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. *Certainty factor* menggunakan suatu nilai untuk mengasumsi derajat keyakinan seorang pakar terhadap suatu data.

Kelebihan dari *Breadth First Search Algorithm* adalah tidak akan menemukan jalan buntu artinya tidak akan terperangkap untuk mengeksplorasi sebuah jalan yang salah dan jika ada satu solusi, maka *Breadth First Search Algorithm* akan menemukannya. Dan jika ada lebih dari satu solusi maka solusi minimum akan ditemukan artinya solusi yang ditemukan adalah solusi yang paling baik (Indrawaty, dkk, 2011). Sedangkan *Certainty Factor Theory* adalah salah satu teori dari beberapa teori yang mampu bekerja dalam ketidakpastian dan kelebihan dari *Certainty Factor Theory* adalah dapat menunjukkan ukuran kepastian terhadap suatu fakta atau aturan (Rohman dan Fauziah, 2008). Sesuai dengan penjelasan sebelumnya, untuk itu penulis tertarik untuk memilih sebuah judul “Penerapan *Breadth First Search Algorithm* dan *Certainty Factor Theory* pada Diagnosa Penyakit Gigi dan Mulut”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka, permasalahan yang akan dibahas dalam penelitian ini adalah bagaimanakah penerapan *Breadth First Search Algorithm* dan *Certainty Factor Theory* untuk mendiagnosa penyakit gigi dan mulut.

## 1.3 Batasan Masalah

Agar tidak terjadi perluasan dalam pembahasan masalah, maka diperlukan adanya pembatasan masalah untuk dapat memberikan gambaran yang lebih terarah dan fokus. Adapun batasan masalah yang diambil adalah sebagai berikut :

1. Penyakit gigi dan mulut yang dibahas hanya terbatas 35 penyakit dengan 65 gejala yang jelas;
2. data yang digunakan berupa gejala dan cara pengobatan dari ke 35 penyakit tersebut;

#### **1.4 Tujuan Penelitian**

Berdasarkan latar belakang dan rumusan masalah di atas, maka tujuan dari penelitian ini yaitu menerapkan algoritma *Breadth First Search Algorithm* dan *Certainty Factor Theory* untuk mendiagnosa penyakit gigi dan mulut.

#### **1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah dapat memberi kemudahan kepada dokter spesialis gigi dan mulut untuk mendiagnosa penyakit gigi dan mulut dengan menggunakan *software* yang digunakan dalam penelitian ini selain itu juga dapat dijadikan bahan evaluasi di bidang teknologi kedokteran untuk meningkatkan mutu dan fungsinya.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Penyakit Gigi dan Mulut

Mulut merupakan bagian utama dari saluran cerna. Gigi merupakan salah satu alat pencernaan yang bertugas sebagai pelembut makanan agar mudah dicerna oleh lambung. Apabila gigi dan mulut mengalami masalah, kinerja pencernaan akan terhambat. Oleh karena itu kesehatan gigi dan mulut sangatlah penting.

Pemeliharaan kesehatan gigi dan mulut merupakan salah satu upaya meningkatkan kesehatan. Salah satu penyebab seseorang mengabaikan masalah kesehatan gigi dan mulutnya adalah faktor pengetahuan tentang kebersihan gigi dan mulut yang kurang. Masalah kesehatan gigi dan mulut seperti karies, gingivitis, radang, stomatis dan lain-lain saat ini menjadi perhatian yang penting dalam pembangunan kesehatan. Hal itu dilandasi oleh kurangnya pengetahuan dan kesadaran akan pentingnya pemeliharaan gigi dan mulut.

Berikut akan disebutkan 35 macam penyakit gigi dan mulut manusia yang akan digunakan sebagai data dalam penelitian ini:

1. Abses gigi
2. *Glossitis*
3. Kanker gusi gigi
4. Kanker mulut
5. *Geographic tongue*
6. Maloklusi
7. *Periodontitis*
8. *Pulpitis*
9. Radang gusi
10. Sariawan
11. *Fordyce spot*
12. *Retrocuspid papila*
13. *Fissure tongue*



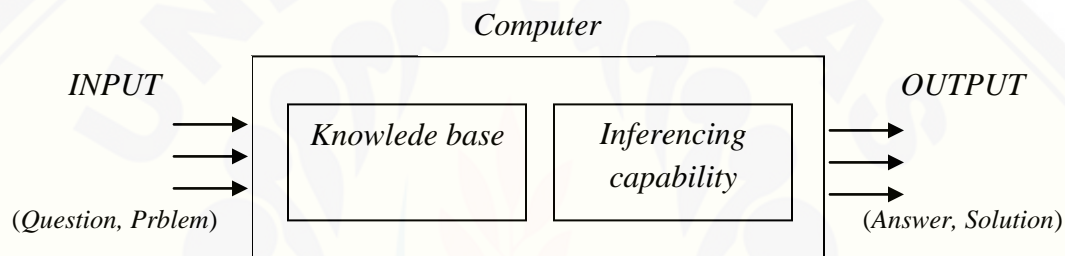
14. *White sponge nevus*
15. *Amalgam tattoo*
16. *Oral melanotic macule*
17. *Melanocytic nevus*
18. *Hematoma*
19. *Ulser aphtosa*
20. *Ulser traumatik*
21. *Infeksi herpes simplex*
22. *Angular cheilitis*
23. *Papilloma*
24. *Verruca vulgaris*
25. *Irritation fibroma*
26. *Peripheral ossifying fibroma*
27. *Peripheral giant cell granuloma*
28. *Limfangioma*
29. *Neurofibroma*
30. *Epulis kongenital*
31. Radang mulut
32. Hipersensitifitas gigi
33. *Cocksackle virus*
34. *Caries dentis*
35. *Calculus*

## **2.2 Kecerdasan Buatan (*Artificial Intelligent*)**

Kecerdasan buatan menurut Paul Y. Gloess didefinisikan sebagai ilmu yang mempelajari bagaimana membuat suatu mesin seolah-olah memiliki kecerdasan dalam memecahkan suatu masalah yang diberikan kepadanya. Teknologi kecerdasan buatan dipelajari dalam bidang-bidang, seperti masalah pembuktian teorema, masalah permainan, sistem pakar, pengolahan bahasa alami, masalah pengolahan citra,

masalah robotika, *modeling human performance* dan bahasa dan lingkungan untuk kecerdasan buatan. (Kristanto, 2004).

Bagian utama *artificial intelligence* adalah pengetahuan atau *knowledge*, suatu pengertian tentang beberapa wilayah subjek yang diperoleh melalui pendidikan dan pengalaman. Walaupun komputer tidak mungkin mendapat pengalaman atau belajar dan meneliti seperti manusia, tetapi ia dapat memperoleh pengetahuan yang dibutuhkannya itu, melalui upaya yang diberikan oleh seorang pakar manusia (Kusumadewi, 2003).



Gambar 2.1 Penerapan Konsep *Artificial Intelligence* Dalam Komputer

### 2.2.1 Sistem Pakar

Secara umum sistem pakar (*expert system*) adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor atau mobil, psikolog adalah orang yang ahli dalam memahami kepribadian seseorang, dan lain-lain (Kusrini, 2008).

Sistem pakar merupakan program-program praktis yang menggunakan strategi



heuristic yang dikembangkan oleh manusia untuk menyelesaikan permasalahan-permasalahan yang khusus, disebabkan oleh keheuristikannya dan sifatnya yang berdasarkan pada pengetahuan sehingga umumnya sistem pakar mempunyai ciri-ciri sebagai berikut :

1. Terbatas pada domain tertentu.
2. Berdasarkan pada kaidah/*rule* tertentu.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Mudah dimodifikasi, yaitu dengan menambah atau menghapus suatu kemampuan dari basis pengetahuannya.
5. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pemakai (Kusrini, 2006).

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain :

1. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
2. Meningkatkan produktivitas akibat meningkatnya kualitas hasil pekerjaan, meningkatnya kualitas pekerjaan ini disebabkan meningkatnya efisiensi kerja.
3. Menghemat waktu kerja.
4. Merupakan arsip terpercaya dari sebuah keahlian, sehingga bagi pemakai sistem pakar seolah-olah beekonsultasi langsung dengan sang pakar, meskipun sang pakar tidak ada.
5. Memperluas jangkauan dari keahlian seorang pakar. Dimana sebuah sistem pakar yang telah disahkan, akan sama saja artinya dengan seorang pakar yang tersedia dalam jumlah besar (dapat diperbanyak dengan kemampuan yang persis sama), dapat diperoleh dan dipakai dimana saja (Kusumadewi, 2003).

Disamping memiliki beberapa manfaat, sistem pakar juga memiliki beberapa kelemahan, antara lain :

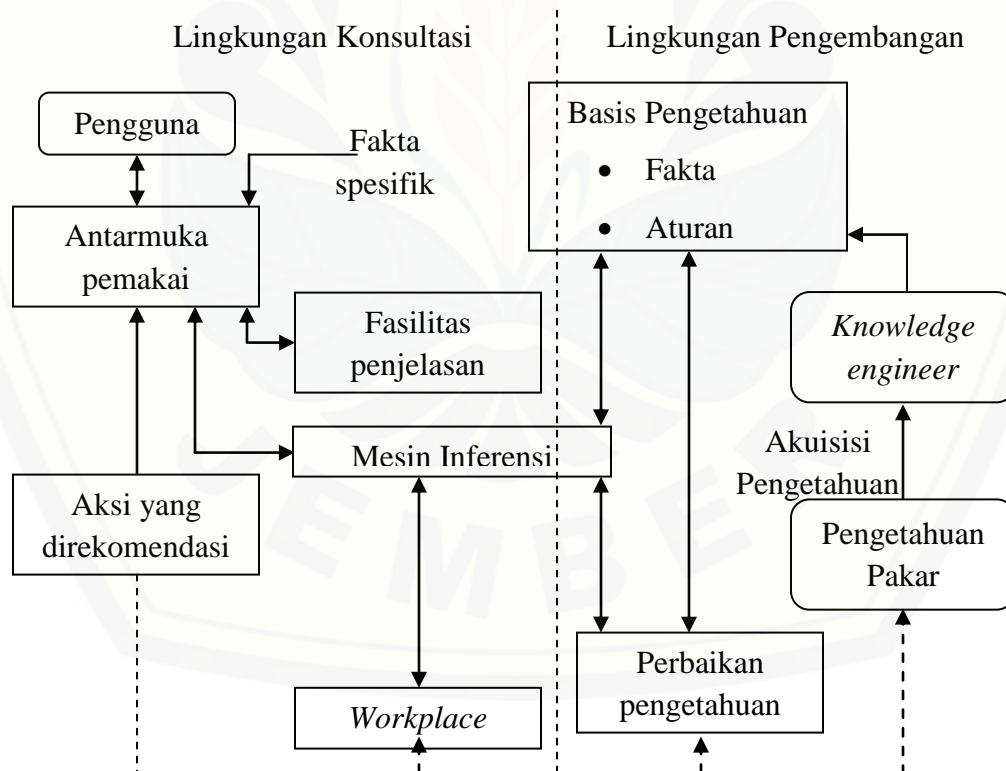
1. Biaya yang diperlukan untuk membuat dan memeliharanya tidak murah.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar dibidangnya.

3. Sistem pakar tidak 100% bernilai benar (Kusumadewi, 2003).

### 2.2.2 Komponen Sistem Pakar

Sebuah sistem pakar memiliki dua komponen utama yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori komputer, dimana pengetahuan ini diambil dari pengetahuan pakar. Sedangkan mesin inferensi merupakan otak dari aplikasi sistem pakar. Bagian inilah yang menuntun user untuk memasukkan fakta sehingga diperoleh suatu kesimpulan. Apa yang dilakukan oleh mesin inferensi ini didasarkan pada pengetahuan yang ada dalam basis pengetahuan (Kusrini, 2008).

Komponen penyusun struktur sistem pakar dapat dilihat pada Gambar 2.2 (Kusumadewi, 2003).



Gambar 2.2 Komponen Sistem Pakar

Kesimpulan dari gambar diatas adalah sistem pakar dibagi menjadi dua lingkungan utama, yaitu lingkungan konsultasi dan pengembangan. Pada lingkungan konsultasi, user berinteraksi dengan sistem pakar melalui user interface. Sistem pakar mengumpulkan informasi tentang masalah user melalui Tanya jawab. Input dari user interface diteruskan ke mesin inferensi. Fasilitas penjelasan berfungsi untuk melacak respon dan menjelaskan tentang kelakuan sistem secara interaktif untuk meningkatkan kinerja sistem. Blackboard merekam setiap aktifitas dari sistem pakar (Kusrini, 2006).

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk manipulasi dalam rangka mencapai solusi kesimpulan (Kusrini, 2006).

Berbeda dengan lingkungan konsultasi, pada lingkungan pengembangan dikhususkan untuk pakar yang mengembangkan sistem pakar dengan menambahkan informasi ke dalam basis pengetahuan maupun mengembangkan komponen mesin inferensinya. Terdapat basis aturan (*rule based*) seperti pengetahuan yang dipresentasikan dalam bentuk *IF-THEN (production rules)* yang akan ditentukan aturan awalnya oleh mesin inferensi dan diaktivasi. Bagian *rule* yang terdiri dari klausa-klausa disebut dengan bagian *IF*, kemudian bagian konklusi yang berisikan konklusi disebut bagian *THEN*. Dalam lingkungan pengembangan *knowledge engineer* menyerap pengetahuan pakar untuk ditransfer ke dalam basis pengetahuan sehingga sistem mampu menjawab permasalahan yang dihadapi oleh *user* (Kusrini, 2006).

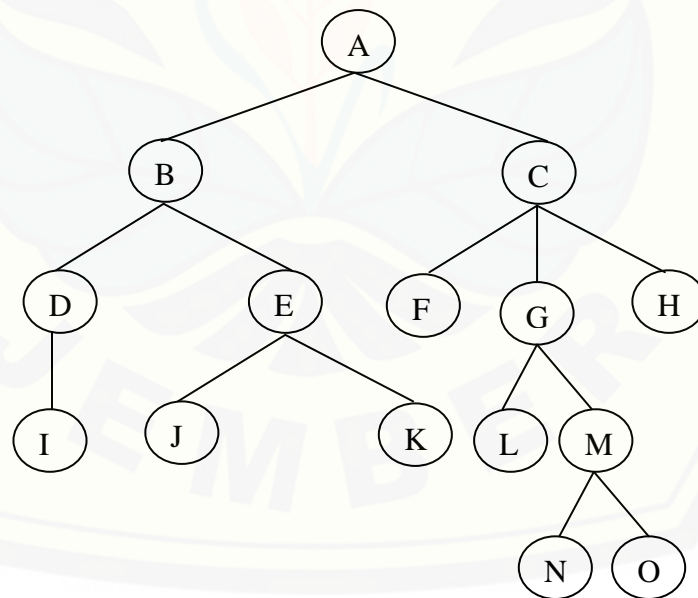
Metode inferensi dibutuhkan untuk membuat sebuah mesin inferensi. Metode inferensi dalam sistem pakar adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Metode inferensi akan menganalisis masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Metode inferensi akan memulai pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada

dalam basis data (Merlina dan Hidayat, 2012). Dalam penelitian ini metode inferensi yang digunakan yaitu *Breadth First Search Algorithm*.

### 2.3 Pohon (*Tree*)

Pohon (*Tree*) adalah suatu struktur hirarki dari simpul (*node*) yang menyimpan informasi atau pengetahuan dan cabang (*edge*) yang menghubungkan *node* (Girriantono & Rilley, 2004). Simpul yang berada paling atas dari hirarki dinamakan *root*, yang biasanya juga mempunyai anak dibawahnya (Karso, 2013). Level suatu simpul dari *root* sebagai level 1 atau level tertinggi. Apabila *node* dinyatakan sebagai level  $n$ , maka *node-node* yang merupakan anaknya berada pada level  $n+1$ .

Pohon pada Gambar 2.3 menunjukkan bahwa *A* merupakan *root*. *B*, *C*, *D* sampai *O* merupakan *node-node* nya. *A* adalah level tertinggi karena *A* merupakan *root*. Sedangkan *B*, *C* berada pada level kedua, *D* berada pada level ketiga, *I* berada pada level keempat, begitu seterusnya.

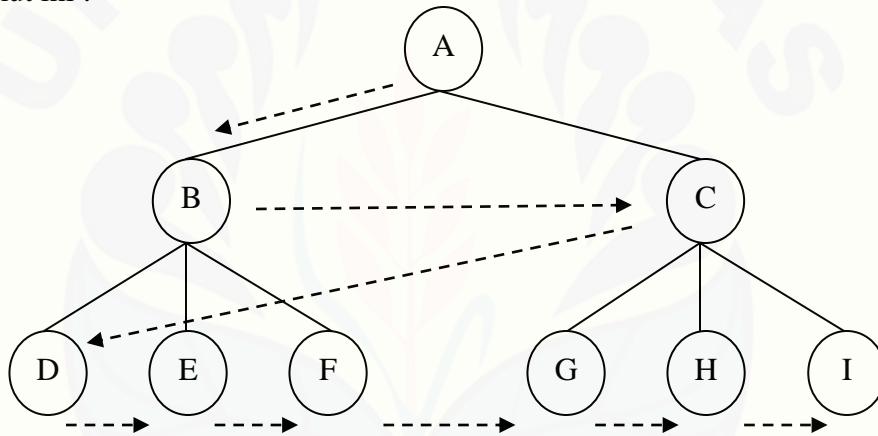


Gambar 2.3 Pohon (*Tree*)

## 2.4 Breadth First Search Algorithm

*Breadth First Search* adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara *preorder* yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu (Klockner, 2005). Pada *Breadth First Search Algorithm* semua *node* pada *level n* akan dikunjungi terlebih dahulu sebelum mengunjungi *nodes* pada *level n+1*. Pencarian dimulai dari *node* akar terus ke level pertama dari kiri ke kanan (Indrawaty, Hermana dan Rinanto, 2011).

Gambaran pencarian *Breadth First Search Algorithm* dapat dilihat pada gambar 2.4 berikut ini :



Gambar 2.4 Pencarian Melebar *Breadth First Search Algorithm*

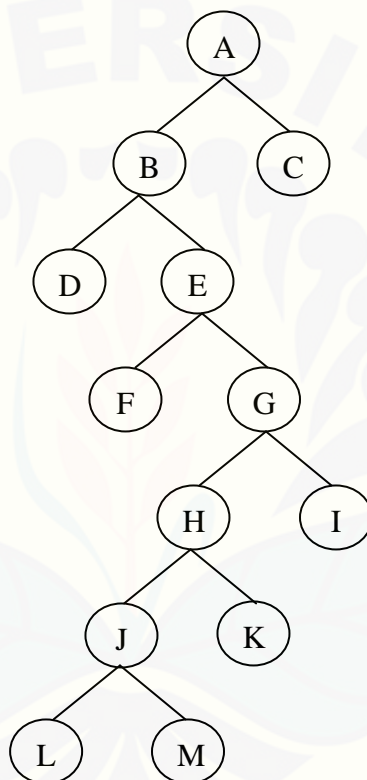
Pada gambar 2.4 terlihat bahwa pencarian dimulai dari keadaan awal (*node A*). Dari *node A* dikembangkan dua *node* baru yang menjadi anaknya, yaitu *node B* dan *node C*. Penelusuran dilanjutkan ke *node B* (menghasilkan *node D, E, F*) dan *node C* (menghasilkan *node G, H, I*). Demikian seterusnya hingga ditemukan solusi (Wijaya, 2013).

Algoritma ini memerlukan sebuah antrian *q* untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan *table*



*Boolean* untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali (Kl'ockner, 2005).

Berikut ini adalah demonstrasi yang dapat membantu kita memahami cara kerja *Breadth First Search*. Misalkan kita ingin mencari lintasan antara simpul A dan E pada pohon pencarian dibawah ini :



Gambar 2.5 Pohon Pencarian (*Search Tree*)

maka langkah-langkahnya adalah :

- langkah awal : dimulai dari simpul akar, atau simpul tujuan yang kita pilih yaitu :

A



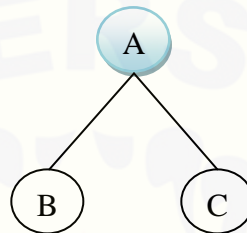
Gambar 2.6 Langkah Awal *Breadth First Search*

kita akan menggunakan dua daftar info untuk menyimpan langkah yang kita lakukan, yaitu *Open* dan *Close*. Sekarang, tambahkan A ke *Open*.

*Open* : A

*Close* : <kosong>

- b. langkah kedua : selanjutnya periksa tetangga-tetangga dari simpul A;



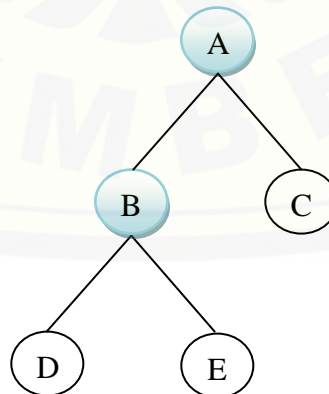
Gambar 2.7 Langkah Kedua *Breadth First Search*

menghapus A dari *Open* dan menambahkannya ke *Close*. Tetangga A, simpul B dan C, ditambahkan ke *Open*. Kemudian, B dan C ditambahkan ke akhir dari *open*. Maka *Open* dan *Close* akan berisi data sebagai berikut :

*Open* : B, C

*Close* : A

- c. langkah ketiga : kini, kita melihat pada simpul B karena muncul pertama kali pada *Open*. Karena B bukan tujuan kita, kita menelusuri tetangganya;



Gambar 2.8 Langkah Ketiga *Breadth First Search*

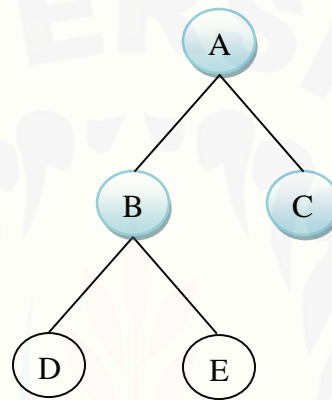


B kini dipindah ke *Close*, tetapi tetangga dari B, simpul D dan E ditambahkan ke akhir dari *Open*.

*Open* : C, D, E

*Close* : A, B

- d. langkah keempat : selanjutnya meluaskan simpul C :



Gambar 2.9 Langkah Keempat *Breadth First Search*

karena C tidak memiliki tetangga, yang kita lakukan adalah menghapus C dari *Open* dan maju lagi, sehingga didapat data *Open* dan *Close* seperti dibawah ini :

*Open* : D, E

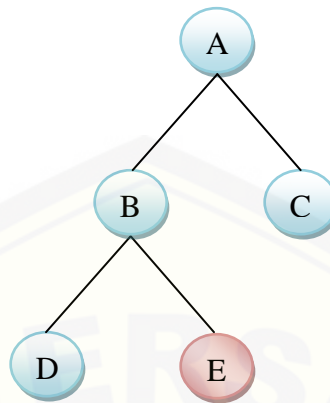
*Close* : A, B, C

- e. langkah kelima : serupa dengan langkah keempat, meluaskan simpul D. Karena bukan tujuan kita, dan juga tidak memiliki tetangga, maka hapus D dari *Open*, lalu menambakkannya ke *Close*, dan lanjut lagi, sehingga didapat data seperti berikut ini :

*Open* : E

*Close* : A, B, C, D

- f. langkah keenam : karena *Open* kita hanya memiliki satu simpul, kita tidak memiliki pilihan lain selain melihat pada simpul E. Karena simpul E adalah tujuan kita, maka kita dapat berhenti di sini :



Gambar 2.10 Langkah Keenam *Breadth First Search*

*Open* dan *Close* akhir kita akan berisi data sebagai berikut :

*Open* : < kosong >

*Close* : A, B, C, D, E

Berjalan dari A ke E akan melalui lintasan B, C dan D bila menggunakan *Breadth First Search Algorithm*. Berdasarkan pada demonstrasi *Breadth First Search Algorithm* yang telah kita lakukan, pseudocode untuk *Breadth First Search Algorithm* dapat dijelaskan sebagai berikut:

- a. mendeklarasikan dua list kosong : *Open* dan *Close*;
- b. menambahkan simpul awal ke *Open*;
- c. *while* <*Open* tidak kosong> *do* <lakukan loop sebagai berikut :>
  1. hapus simpul awal dari *Open*;
  2. cek dan lihat apakah simpul yang dihapus adalah tujuan
    - i. *if* -- jika simpul yang dihapus adalah tujuan, keluar dari *loop*, tambahkan simpul ke *Close*, dan kembali ke nilai dari *Close*;
    - ii. *Else* – jika simpul yang dihapus bukan tujuan, lanjutkan *loop* ( ke langkah c)
  3. ekstrak tetangga dari simpul yang dihapus di atas;
  4. tambahkan tetangga ke awal dari *Open*, dan tambahkan simpul yang dihapus ke *Close*; kemudian lanjutkan *looping*.

## 2.5 Peluang Komplemen Suatu Kejadian

Probabilitas suatu kejadian adalah angka yang menunjukkan kemungkinan terjadinya suatu kejadian. Nilainya diantara 0 dan 1. Kejadian yang mempunyai nilai probabilitas 1 adalah kejadian yang pasti terjadi atau sesuatu yang telah terjadi, misalnya matahari yang terbit di timur sampai sekarang. Sedangkan suatu kejadian yang mempunyai nilai probabilitas 0 adalah kejadian yang mustahil atau tidak mungkin terjadi. Misalnya seekor kambing yang melahirkan seekor sapi (Ross, 1976).

Probabilitas atau peluang suatu kejadian  $A$  terjadi dilambangkan dengan notasi  $P(A)$ . Sebaliknya, probabilitas bukan  $A$  (komplemen  $A$ ), atau probabilitas suatu kejadian  $A$  tidak akan terjadi adalah  $1-P(A)$ . Sebagai contoh, peluang untuk tidak munculnya mata dadu enam bila dilempar sebuah dadu adalah  $1 - \frac{1}{6} = \frac{5}{6}$  (Ross, 1976).

Jika  $A'$  adalah komplemen kejadian  $A$ , maka secara umum peluang kejadian  $A'$  dapat ditulis dengan aturan :

$$A' = 1 - P(A) \quad (2.1)$$

## 2.6 Certainty Factor Teory

Karena sering ditemukan jawaban yang tidak memiliki kepastian penuh dalam menghadapi suatu permasalahan maka dalam menyelesaikan masalah sistem pakar tidak hanya membutuhkan metode inferensi akan tetapi juga membutuhkan metode perhitungan untuk menghitung sebuah kepastian hasil dari metode inferensi. Ketidakpastian ini dapat berupa probabilitas atau kebolehjadian yang tergantung dari hasil suatu kejadian. Hasil yang tidak pasti disebabkan oleh dua faktor, yaitu aturan yang tidak pasti dan jawaban pengguna yang tidak pasti atas suatu pertanyaan yang diajukan oleh sistem. Hal ini sangat mudah dilihat pada sistem diagnosis penyakit, dimana pakar tidak dapat mendefinisikan hubungan antara gejala dengan

penyebabnya secara pasti, dan pasien tidak dapat merasakan suatu gejala dengan pasti pula. Pada akhirnya akan ditemukan banyak kemungkinan diagnosis (Rohman dan Fauziah, 2008).

Sistem pakar harus mampu bekerja dalam ketidakpastian. Sejumlah teori telah ditemukan untuk menyelesaikan ketidakpastian, termasuk diantaranya probabilitas klasik, probabilitas bayes, teori *Hartley* berdasarkan himpunan klasik, teori *Shannon* berdasarkan pada probabilitas, teori *depnster-shafer*, teori *fuzzy zadeh*, dan factor kepastian (*certainty factor*) (Rohman dan Fauziah, 2008).

Faktor kepastian (*certainty factor*) diperkenalkan oleh *Shortliffe Buchanan* dalam pembuatan MYCIN. *Certainty factor* (CF) merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. *Certainty factor* (CF) menunjukkan ukuran kepastian terhadap suatu fakta atau aturan (Kusumadewi, 2003).

Team pengembang MYCIN mencatat bahwa dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti mungkin, kemungkinan besar dan hamper pasti. Untuk mengakomodasi halini tim MYCIN menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. Faktor kepastian ini juga berguna untuk mengatasi ketidakpastian dalam menentukan penyakit yang mempunyai gejala sama. Dengan ukuran kepercayaan (MB) dan ketidakpercayaan (MD) suatu gejala terhadap penyakit yang sama, maka didapat sebuah nilai kepastian atau factor kepastian (CF). Nilai yang digunakan sebagai level kepercayaan pada suatu hipotesis yang memberikan suatu informasi yang tersedia adalah *certainty factor*. Berikut dalam table adalah aturan nilai-nilai kepercayaan yang diberikan oleh MYCIN (Cahyono dan Riadi, 2013).

Tabel 2.1 Tabel Interpretasi *Certainty Factor*

No.	<i>Certainty Term</i>	$CF_{akhir}$
1	Pasti tidak	-1,0

2	Hampir pasti tidak	-0,8
3	Kemungkinan besar tidak	-0,6
4	Mungkin tidak	-0,4
5	Tidak tahu / tidak yakin	-0.2 ... 0.2
6	Mungkin	0,4
7	Kemungkinan besar	0,6
8	Hampir pasti	0,8
9	Pasti	1,0

Ada dua tahap model yang sering digunakan untuk menghitung tingkat keyakinan (CF) dari sebuah *rule* adalah sebagai berikut :

- a. Dengan menggali dari hasil wawancara dengan pakar. Nilai CF didapat dari interpretasi term dari pakar menjadi nilai MD atau MB tertentu yang digunakan untuk menghitung nilai CF dari suatu *rule* dengan beberapa metode.

Tabel 2.2 Nilai interpretasi untuk MB dan MD

No.	Kepercayaan	MB/MD
1	Tidak tahu / tidak ada	0.00 ... 0.29
2	Mungkin	0.30 ... 0.49
3	Kemungkinan besar	0.50 ... 0.69
4	Hampir pasti	0.70 ... 0.89
5	Pasti	0.90 ... 1.00

- b. Menggunakan metode perhitungan. Faktor kepastian menunjukkan ukuran kepastian suatu fakta atau aturan. Rumus umum factor kepastian adalah sebagai berikut :

$$CF[h, e] = MB[h, e] - MD[h, e] \quad (2.2)$$

dengan,

$CF[h, e]$  = faktor kepastian;

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

$MD[h,e]$  = ukuran ketidakpercayaan terhadap *hipotesis*  $h$ , jika diberi *evidence*  $e$  (antara 0 dan 1) (Cahyono dan Riadi, 2013).

Sistem pakar ini menggunakan nilai kepercayaan (MB) dan nilai ketidakpercayaan (MD) dalam mengkombinasikan beberapa *evidence* untuk menentukan nilai CF dari suatu *hipotesis*. Konsep ini diformulasikan dengan rumusan dasar sebagai berikut :

$$MB[h, e_1 \wedge e_2] = MB[h, e_1] + MB[h, e_2]. (1 - MB[h, e_1]) \quad (2.3)$$

$$MD[h, e_1 \wedge e_2] = MD[h, e_1] + MD[h, e_2]. (1 - MD[h, e_1]) \quad (2.4)$$

(Cahyono dan Riadi, 2013).



## BAB 3. METODE PENELITIAN

### 3.1 Metode Pengumpulan Data

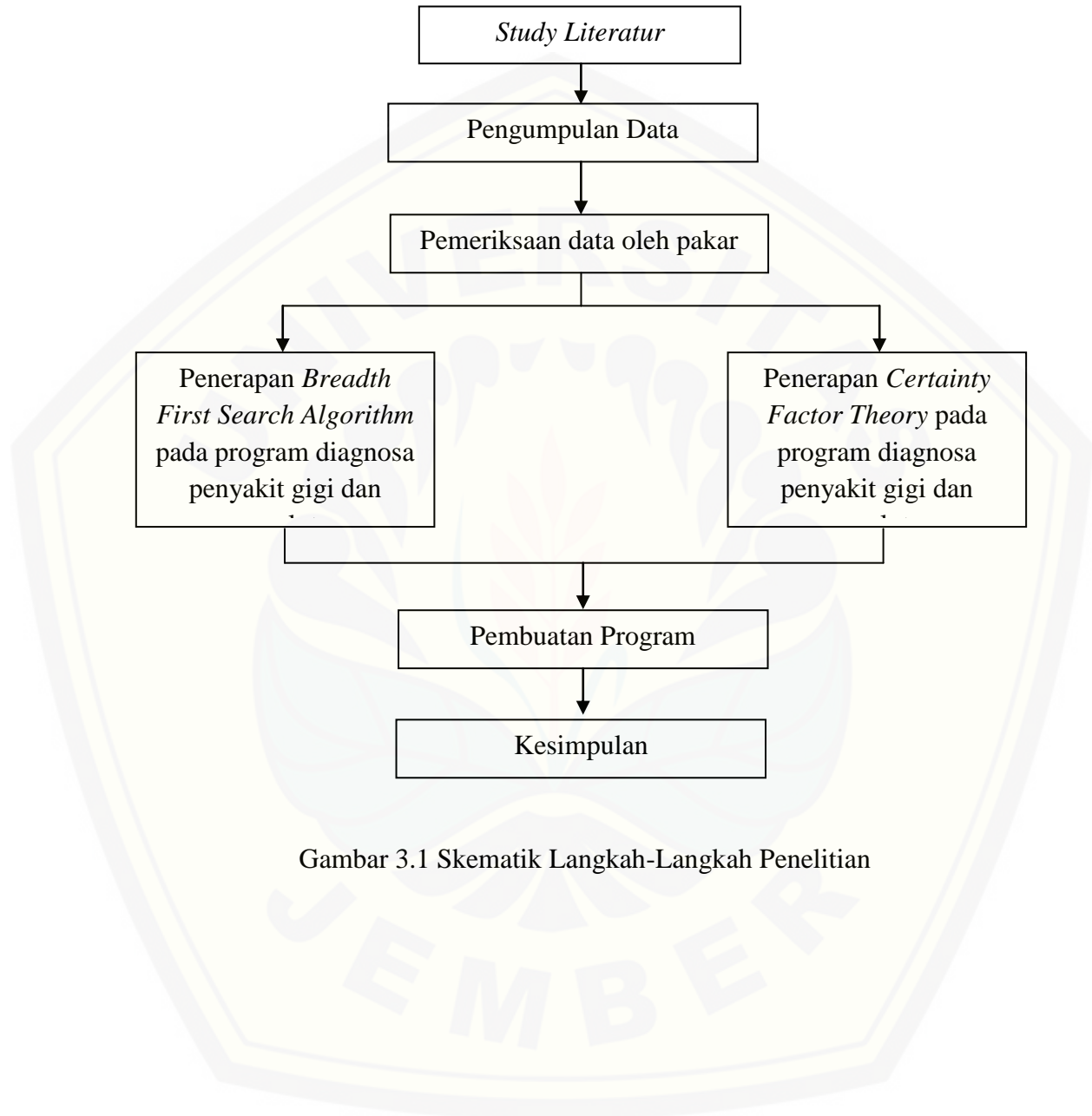
Data yang digunakan dalam penelitian ini adalah data yang diambil dari *google* yang kemudian hasil akhirnya telah dievaluasi dan diperiksa oleh seorang pakar kesehatan gigi dan mulut yaitu Drg. Elya Adibah yang membuka praktek di Jl. Srikoyo No 150 Patrang Jember. Data 35 macam penyakit gigi dan mulut beserta gejala dan pengobatannya dapat dilihat pada Lampiran A, Lampiran B dan Lampiran C.

### 3.2 Langkah-langkah Penyelesaian

Langkah-langkah yang dilakukan untuk memilih sebuah mobil dengan menggunakan *Breadth First Search Algorithm* dan *Certainty Factor Teory* adalah sebagai berikut:

- a. Mengumpulkan data yang berupa gejala dan cara pengobatan dari penyakit gigi dan mulut dari internet;
- b. Hasil akhir dievaluasi dan diperiksa oleh seorang pakar yaitu seorang dokter spesialis gigi dan mulut;
- c. Menerapkan *Breadth First Search Algorithm* dan *Certainty Factor Teory* pada data yang telah diperoleh pada langkah *a* untuk membangun program diagnosa penyakit gigi dan mulut;
- d. Membuat program diagnosa penyakit gigi dan mulut dengan menggunakan *Breadth First Search Algorithm* dan *Certainty Factor Teory* dengan *software* MATLAB R2009a;
- e. Membuat kesimpulan dari hasil yang diperoleh.

Secara sistematis, langkah-langkah yang akan dilakukan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skematik Langkah-Langkah Penelitian

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Kesimpulan yang didapat dari tugas akhir ini yaitu :

- a. Aplikasi sistem pakar diagnosa penyakit gigi dan mulut mampu mengenali diagnosa penyakit dengan menggunakan *Breadth First Search Algorithm* sebagai metode pencarian dan *Certainty Factor Theory* sebagai metode perhitungan.
- b. *Breadth First Search Algorithm* membutuhkan sebuah bobot untuk beberapa penyakit yang memiliki gejala yang sama agar hanya didapat satu diagnosa saja.
- c. Dengan menggunakan *Breadth First Search Algorithm*, pencarian selalu ditemukan solusi karena semua *nodes* akan dilalui.
- d. *Certainty Factor Theory* dapat menyimpan representasi pengetahuan pakar berdasarkan nilai keyakinan *MB* dan ketidakyakinan *MD* dan mampu menyelesaikan masalah yang kompleks dengan menggunakan nilai kepastian *Certainty Factor*, semakin besar nilai kepastian, semakin besar pula potensi penyakit yang menyerang pasien, begitupun sebaliknya.

### 5.2 Saran

Aplikasi sistem pakar diagnosa penyakit gigi dan mulut masih jauh dari kata sempurna untuk itu pengembangan aplikasi ini sangat disarankan untuk :

1. membuat halaman *edit* yang dapat digunakan untuk menambah, mengupdate dan menghapus penyakit, gejala dan pengobatannya;
2. mencari metode pencarian dan metode perhitungan lain selain *Breadth First Search Algorithm* dan *Certainty Factor Theory* yang dapat diaplikasikan untuk kasus sistem pakar agar aplikasi menjadi lebih efektif dan efisien seperti Metode *Dempster Shafer*, *Fuzzy Clustering*, Metode *Bayes*, Metode *Case Based Reasoning*, Metode *Fuzzy Logic* dan lain-lain.

**DAFTAR PUSTAKA**

- Aini, N. 2014. *Perbandingan Algoritma Ho-Chang dan Algoritma Pour pada Optimasi Penjadwalan Produksi Suwar-Suwir*. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Cahyono, G.R & Riadi, J. 2013. Implementasi *Certainty Factor* pada Sistem Pakar untuk Diagnosa Hama dan Penyakit Tanaman Jagung Menggunakan SMS Gateway. *Jurnal INTEKNA*, (2) : 131-136.
- Girriantano, J & Riley, G. 2004. *Expert System: Principle and Programing*, 4<sup>th</sup> Edition. PWS Kent.
- Indrawaty, Y, Hermana, A.N, Rinanto, V.S. 2011. Simulasi Pergerakan Langkah Kuda Menggunakan Metode *Breadth First Search*. *Media Informatika*, 2 (3) : 1-7.
- Kl'ockner. K. 2005. *Depth and Breadth-First Processing of Search Result Lists: An Example Paper in the SIGCHI Style*.
- Kristanto. A. 2004. *Kecerdasan Buatan*. Yogyakarta : Graha Ilmu.
- Kusrini. 2006. *Sistem Pakar Teori dan Aplikasi*. Yogyakarta : Andi Offset.
- Kusrini, 2008. *Aplikasi Sistem Pakar*. Yogyakarta : Andi Offset.
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Merlina, N & Hidayat, R. 2012. *Perancangan Sistem Pakar*. Bogor : Ghalia Indonesia.
- Munir, R. 2009. *Matematika Diskrit, Edisi 3*. Bandung: Informatika.

- N. Mardiah, A. 2015. *Diagnosa Penyakit Kucing dengan Algoritma Depth Firsh Search dan Hill Climbing*. Tidak Diterbitkan. Skripsi. Jember: Fakultas MIPA Jurusan Matematika Universitas Jember.
- Nugraha, D.W. 2012. Penerapan Kompleksitas Waktu Algoritma Prim untuk Menghitung Kemampuan Komputer dalam Melaksanakan Perintah. *Jurnal Ilmiah Foristek*, **2** (2) : 195-207.
- Rohman, F.F, Fauziah, A. 2008. Rancang Bangun Aplikasi Sistem Pakar untuk Menentukan Jenis Gangguan Perkembangan pada Anak. *Media Informatika*, **6** (1) : 1-23.
- Ross, S. 1976. a First Course in Probability. [https://id.wikipedia.org/wiki/Peluang\\_%28matematika%29](https://id.wikipedia.org/wiki/Peluang_%28matematika%29) [19 Mei 2015].
- Wijaya, E. 2013. Analisis Penggunaan Algoritma Breadth First Search dalam Konsep Artificial Intellegencia. *Jurnal TIME*, **2** (2) : 18-26.
- Yanti, R. 2015. Metode Certainty Factor (CF) Serta Proses Hitung Manual Contoh Kasus Diagnosa Penyakit Paru-Paru. <http://rekayantianwar.blogspot.co.uk/2015/09/metode-certainty-factor-cf-serta-proses.html> [06 September 2015].









B. Tabel Data 10 Penyakit Gigi dan Mulut Beserta Cara Pengobatannya

No.	Penyakit	Pengobatan
1	$P_3$	Diobati menggunakan obat penghilang rasa sakit seperti parasetamol atau ibuprofen sampai abses dikeringkan atau diberi antibiotik, pasien diharapkan berkunjung lagi untuk pembersihan abses.
2	$P_1$	Taa (tidak perlu dilakukan terapi, karena penyakit akan hilang dengan sendirinya)
3	$P_4$	Melakukan metode radioterapi, kemoterapi serta pembedahan.
4	$P_2$	Kanker mulut stadium awal dapat diatasi dengan operasi menggunakan sinar laser atau yang disebut <i>photodynamic therapy</i> (PDT). Kanker mulut stadium lanjut dilakukan pengangkatan tumor beserta jaringan sehat di sekitarnya, misalnya pengangkatan sebagian jaringan pada dinding mulut, lidah, atau bahkan rahang. Dilanjutkan dengan operasi rekonstruksi selanjutnya radioterapi. Kanker mulut stadium akhir dilakukan kemoterapi.
5	$P_7$	Taa (tidak perlu dilakukan terapi, karena penyakit akan hilang dengan sendirinya) Obat yang dapat digunakan untuk mengelola ketidaknyamanan atau sensitivitas yang merupakan pengobatan simptomatik, antara lain: <ul style="list-style-type: none"> <li>• Obat penghilang nyeri (analgesik)</li> <li>• Larutan kumur dengan anestesi (obat bius)</li> <li>• Larutan kumur yang mengandung antihistamin</li> <li>• Salep atau larutan kumur kortikosteroid</li> </ul>
6	$P_5$	Gigi bisa disusun ulang dengan menggunakan alat gigi, seperti bingkai penyangga gigi (kawat dan pegas dibawa oleh pengurung yang disesuaikan dengan gigi dengan lem gigi) atau sebuah penyangga (bingkai penahan gigi yang bisa dipindahkan dipadukan dengan kawat dan piringan plastik yang diselipkan ke dalam atap mulut).
7	$P_9$	Menggunakan pemeriksaan X-ray.
8	$P_{10}$	Bila pulpitis masih berada pada stadium awal, maka nyeri bisa diredakan dengan penambalan sementara dengan kandungan obat penenang saraf. Tambalan ini dapat dibiarkan selama 6-8 minggu lalu diganti dengan tambalan permanen. Bila kerusakan pulpa sudah meluas dan tidak bisa diperbaiki lagi maka nyeri hanya bisa dihilangkan dengan mencabut pulpa, baik dengan pencabutan gigi atau melalui pengobatan saluran akar.
9	$P_8$	Membersihkan plak dan karang gigi apabila penyebabnya adalah plak dan karang gigi untuk mencegah komplikasi. Lakukan penambalan serta perbaikan apabila ada gigi pasien yang berlubang atau gigi palsu yang rusak. Dianjurkan untuk melakukan pembersihan gigi secara rutin
10	$P_6$	Memberi obat anti-jamur.
11	$P_{11}$	Taa (tidak perlu dilakukan terapi, hilang sendiri)
12	$P_{12}$	Taa (tidak perlu dilakukan terapi, hilang sendiri)
13	$P_{13}$	menyikat kotoran sisa makanan pada bagian lidah yang mengalami fissure tongue

14	$P_{14}$	Taa (tidak perlu dilakukan terapi, hilang sendiri)
15	$P_{15}$	Taa (tidak perlu dilakukan terapi, hilang sendiri)
16	$P_{16}$	Taa (tidak perlu dilakukan terapi, hilang sendiri)
17	$P_{17}$	melakukan pembedahan dengan cara memotong benjolan karena berpotensi menjadi tumor ganas
18	$P_{18}$	Taa (tidak perlu dilakukan terapi, hilang sendiri) → sembuh 7-14 hari
19	$P_{19}$	kalau sakit dikasih obat kumur antiseptik
20	$P_{20}$	kalau sakit dikasih obat kumur antiseptik
21	$P_{21}$	kalau sakit dikasih obat kumur antiseptik
22	$P_{22}$	diberi antifungi, antibiotik dan vitamin
23	$P_{23}$	Taa (tidak perlu dilakukan terapi, hilang sendiri) atau eksisi (dibedah, di potong benjolan)
24	$P_{24}$	eksisi (dibedah, di potong benjolan)
25	$P_{25}$	eksisi (dibedah, di potong benjolan) dan menghilangkan penyebab iritasi
26	$P_{26}$	pembedahan dengan pemotongan melibatkan jaringan penyangga gigi
27	$P_{27}$	pembedahan dengan pemotongan melibatkan jaringan penyangga gigi
28	$P_{28}$	eksisi (dibedah, di potong benjolan)
29	$P_{29}$	Taa (tidak perlu dilakukan terapi, hilang sendiri) atau eksisi (dibedah, di potong benjolan) apabila mengganggu estetika gigi
30	$P_{30}$	eksisi (dibedah, di potong benjolan), dilakukan secara hati-hati agar tidak mengganggu perkembangan gigi
31	$P_{31}$	Menggunakan tablet hisap antifungal
32	$P_{32}$	Melakukan ekstraksi atau pencabutan gigi geraham
33	$P_{33}$	Pasien disarankan untuk diisolasi dan diberi makanan yang mengandung banyak protein serta istirahat yang cukup
34	$P_{34}$	Melakukan penambalan gigi atau pencabutan yang kemudian diganti dengan gigi tiruan
35	$P_{35}$	Membersihkan karang giginya

## C. Keterangan Notasi Penyakit dan Gejala

$P_1$ = <i>Glossitis</i>	$G_5$ = Serak pada suara secara berkepanjangan
$P_2$ = Kanker mulut	$G_6$ = Sariawan yang tidak kunjung sembuh
$P_3$ = Abses gigi	$G_7$ = Sakit tenggorokan
$P_4$ = Kanker gusi gigi	$G_8$ = Kaku dan sakit yang menyebar ke telinga, rahang bawah, leher dan pada gigi yang terkena
$P_5$ = Maloklusi	$G_9$ = Sakit berdenyut intens di gigi atau gusi
$P_6$ = Sariawan	$G_{10}$ = Gigi tiba-tiba sakit dan memburuk untuk beberapa jam atau hari
$P_7$ = <i>Geographic tongue</i>	$G_{11}$ = Pembengkakan parah di wajah
$P_8$ = Radang gusi	$G_{12}$ = Terjadi pendarahan di dalam mulut tetapi tidak diketahui penyebabnya
$P_9$ = <i>Periodontitis</i>	$G_{13}$ = Terasa nyeri pada bagian yang ditumbuhi oleh sel kanker
$P_{10}$ = <i>Pulpitis</i>	$G_{14}$ = Terjadi pengenduran atau patahan pada kelainan baris gigi
$P_{11}$ = <i>Fordyce spot</i>	$G_{15}$ = Luka berwarna putih yang biasa muncul di lidah atau dinding mulut
$P_{12}$ = <i>Retrocuspid papila</i>	$G_{16}$ = Ketidaknyamanan, rasa sakit atau sensasi terbakar
$P_{13}$ = <i>Fissure tongue</i>	$G_{17}$ = Permukaan lidah dan samping lidah halus kemerahan dengan pola yang tidak teratur
$P_{14}$ = <i>White sponge nevus</i>	$G_{18}$ = Sering terjadi perubahan lokasi, ukuran dan bentuk dari pola tersebut
$P_{15}$ = <i>Amalgam tattoo</i>	$G_{19}$ = Gusi membengkak dan berwarna merah keunguan
$P_{16}$ = <i>Oral melanotic macule</i>	$G_{20}$ = Mulut terasa tidak enak dan napas menjadi bau (bau mulut)
$P_{17}$ = <i>Melanocytic nevus</i>	$G_{21}$ = Gusi yang rentan mengalami pendarahan
$P_{18}$ = <i>Hematoma</i>	$G_{22}$ = Jika disentuh, gusi terasa lunak
$P_{19}$ = <i>Ulser aphtosa</i>	$G_{23}$ = Penyusutan gusi sehingga ukuran gigi terlihat lebih tinggi dari biasanya
$P_{20}$ = <i>Ulser traumatik</i>	$G_{24}$ = Keluarnya nanah pada bagian yang membatasi gigi dan gusi
$P_{21}$ = <i>Infeksi herpes simplex</i>	$G_{25}$ = Jarak antara satu gigi dan gigi lainnya terasa renggang
$P_{22}$ = <i>Angular cheilitis</i>	
$P_{23}$ = <i>Papilloma</i>	
$P_{24}$ = <i>Verruca vulgaris</i>	
$P_{25}$ = <i>Irritation fibroma</i>	
$P_{26}$ = <i>Peripheral ossifying fibroma</i>	
$P_{27}$ = <i>Peripheral giant cell granuloma</i>	
$P_{28}$ = <i>Limfangioma</i>	
$P_{29}$ = <i>Neurofibroma</i>	
$P_{30}$ = <i>Epulis kongenital</i>	
$P_{31}$ = Radang mulut	
$P_{32}$ = Hipersensitifitas gigi	
$P_{33}$ = <i>Cocksackle virus</i>	
$P_{34}$ = <i>Caries dentis</i>	
$P_{35}$ = <i>Calculus</i>	
$G_1$ = Kesulitan mengunyah, menelan atau berbicara	
$G_2$ = Peradangan pada lidah	
$G_3$ = Permukaan lidah menjadi halus	
$G_4$ = Warna lidah sangat merah atau putih	



- $G_{26}$  = Gigi tanggal
- $G_{27}$  = Menyebabkan sakit gigi yang luar biasa
- $G_{28}$  = Terdapat bintik-bintik kecil kekuningan dan peninggian yang sangat banya di pipi bagian dalam, bibir dan bagian belakang rahang
- $G_{29}$  = Asimptomatik (Tidak terdapat gejala seperti sakit, nyeri, demam dan lain-lain)
- $G_{30}$  = Terdapat benjolan lunak pada gusi gigi taring rahang bawah bagian dalam
- $G_{31}$  = Terdapat garis-garis berlipat pada bagian belakang lidah
- $G_{32}$  = Terasa sakit saat ada peradangan
- $G_{33}$  = Terdapat lipatan putih yang tidak jelas atau samar menyerupai spons pada bagian dalam pipi, bibir, gusi, dasar mulut, lidah dan langit-langit mulut
- $G_{34}$  = Terdapat lipatan keabu-abuan dan samar pada permukaan lunak rongga mulut seperti gusi dan bagian dalam pipi
- $G_{35}$  = Terdapat bulatan datar warna coklat, hitam dan abu-abu pada bibir, gusi, langit-langit mulut dan bagian dalam pipi
- $G_{36}$  = Terdapat sekitar 20 bintik hitam ditubuh
- $G_{37}$  = Terdapat benjolan seperti kubah, berwarna samar pada langit-langit mulut bagian depan dan belakang, bagian dalam pipi, bibir dan dasar mulut
- $G_{38}$  = Terdapat riwayat trauma
- $G_{39}$  = Terdapat luka seperti sariawan yang sangat banyak, memilik batas yang jelas dan berwarna kemerahan
- $G_{40}$  = Luka seperti sariawan yang tertutup selaput putih
- $G_{41}$  = Sakit pada bagian dalam pipi, bibir, langit-langit mulut bagian belakang, lidah dan dasar mulut
- $G_{42}$  = Terdapat benjolan-benjolan kecil yang berdekatan, banyak dan samar, hampir sama seperti sariawan terdapat pada bagian tepi lidah, bagian dalam pipi dan bibir
- $G_{43}$  = Terdapat benjolan kecil yang sangat banyak ditutupi dengan selaput putih pada bibir, langit-langit mulut bagian depan, gusi dan bagian belakang lidah
- $G_{44}$  = Terdapat garis-garis dalam, kering pada sudut mulut
- $G_{45}$  = Terdapat benjolan bertangkai kasar yang saling berdekatan, berwarna putih atau merah muda terdapat pada bibir, lidah, bagian belakang mulut dan langit-langit mulut.
- $G_{46}$  = Terdapat benjolan bertangkai kasar yang saling berdekatan, berwarna putih atau merah muda terdapat pada bibir
- $G_{47}$  = permukaan halus, dan berwarna normal pada bibir lidah, langit-langit mulut dan bagian dalam pipi
- $G_{48}$  = Terdapat pembesaran dengan batasan yang sangat jelas, permukaan halus, dan berwarna normal pada gusi dan diantara dua gigi terutama didaerah bagian gigi geraham
- $G_{49}$  = Terdapat pembesaran dengan batasan yang sangat jelas, permukaan halus, dan berwarna normal pada gusi bagian depan rahang atas maupun rahang bawah
- $G_{50}$  = Terdapat benjolan seperti kubah berwarna putih pudar atau terawang dan agak berwarna merah muda kebiruan pada lidah bagian dalam pipi dan dasar mulut
- $G_{51}$  = Terdapat benjolan dengan batas yang jelas, bisa juga samar, bergerombol, halus dan banyak, berwarna normal terdapat pada lidah, gusi, bagian dalam pipi dan langit-langit mulut
- $G_{52}$  = Terdapat benjolan dengan batas yang jelas, bertangkai, kenyal, berwarna



normal, dapat menjadi sariawan bagian depan rahang atas

$G_{53}$  = Lidah berwarna pucat

$G_{54}$  = Terdapat bercak kuning dan mudah dikeruk pada lidah

$G_{55}$  = Terasa perih saat digunakan untuk makan dan gosok gigi

$G_{56}$  = Rasa nyeri pada gigi

$G_{57}$  = Gigi dan gusi membengkak

$G_{58}$  = Terdapat bintik-bintik merah pada mulut

$G_{59}$  = Demam dan dapat mengakibatkan rasa menggigil

$G_{60}$  = Terjadi perubahan warna pada gigi, seperti coklat, kehitaman atau terlalu putih

$G_{61}$  = Terdapat plak atau gidal yang mengadakan pengapuran atau pengerasan dan melekat pada permukaan gigi disekeliling gusi

$G_{62}$  = Mudah berdarah pada gigi yang ompong

$G_{63}$  = Terasa agak terbakar pada sudut mulut

$G_{64}$  = Terasa seperti terbakar pada bibir, langit-langit mulut bagian depan, gusi dan bagian belakang lidah

$G_{65}$  = Terdapat pembesaran dengan batasan yang sangat jelas pada bibir, lidah, langit-langit mulut dan bagian dalam pipi

## D. Bobot Beberapa Penyakit yang Memiliki Gejala Sama

$G_1$  = Kesulitan mengunyah, menelan atau berbicara

1.  $P_1$  = *Glossitis* (90%)
2.  $P_2$  = Kanker mulut (70%)
3.  $P_4$  = Kanker gusi gigi (20%)
4.  $P_5$  = Maloklusi (2%)
5.  $P_6$  = Sariawan (50%)

$G_4$  = Warna lidah sangat merah atau putih

1.  $P_1$  = *Glossitis* (80%)
2.  $P_2$  = Kanker mulut (10%)

$G_5$  = Mengalami serak pada suara secara berkepanjangan sehingga menyebabkan perubahan suara

1.  $P_2$  = Kanker mulut (50%)
2.  $P_4$  = Kanker gusi gigi (30%)

$G_8$  = Rasa kaku dan sakit yang menyebar ke telinga, rahang bawah dan leher, di sisi yang sama dengan gigi yang terkena

1.  $P_2$  = Kanker mulut (85%)
2.  $P_3$  = Abses gigi (80%)

$G_{16}$  = Ketidaknyamanan, rasa sakit atau sensasi terbakar

1.  $P_6$  = Sariawan (90%)
2.  $P_7$  = *Geographic tongue* (60%)

$G_{19}$  = Gusi membengkak dan berwarna merah keunguan

1.  $P_8$  = Radang gusi (60%)

2.  $P_9$  = *Periodontitis* (80%)

$G_{20}$  = Mulut terasa tidak enak dan napas menjadi bau (bau mulut)

1.  $P_8$  = Radang gusi (60%)

2.  $P_9$  = *Periodontitis* (80%)

$G_{29}$  = Asimptomatik (Tidak terdapat gejala seperti sakit, nyeri, demam dan lain-lain)

1.  $P_{11}$  = *Fordyce spot* (5%)

2.  $P_{12}$  = *Retrocuspid papila* (10%)

3.  $P_{14}$  = *White sponge nevus* (12%)

4.  $P_{15}$  = *Amalgam tattoo* (20%)

5.  $P_{16}$  = *Oral melanotic macule* (25%)

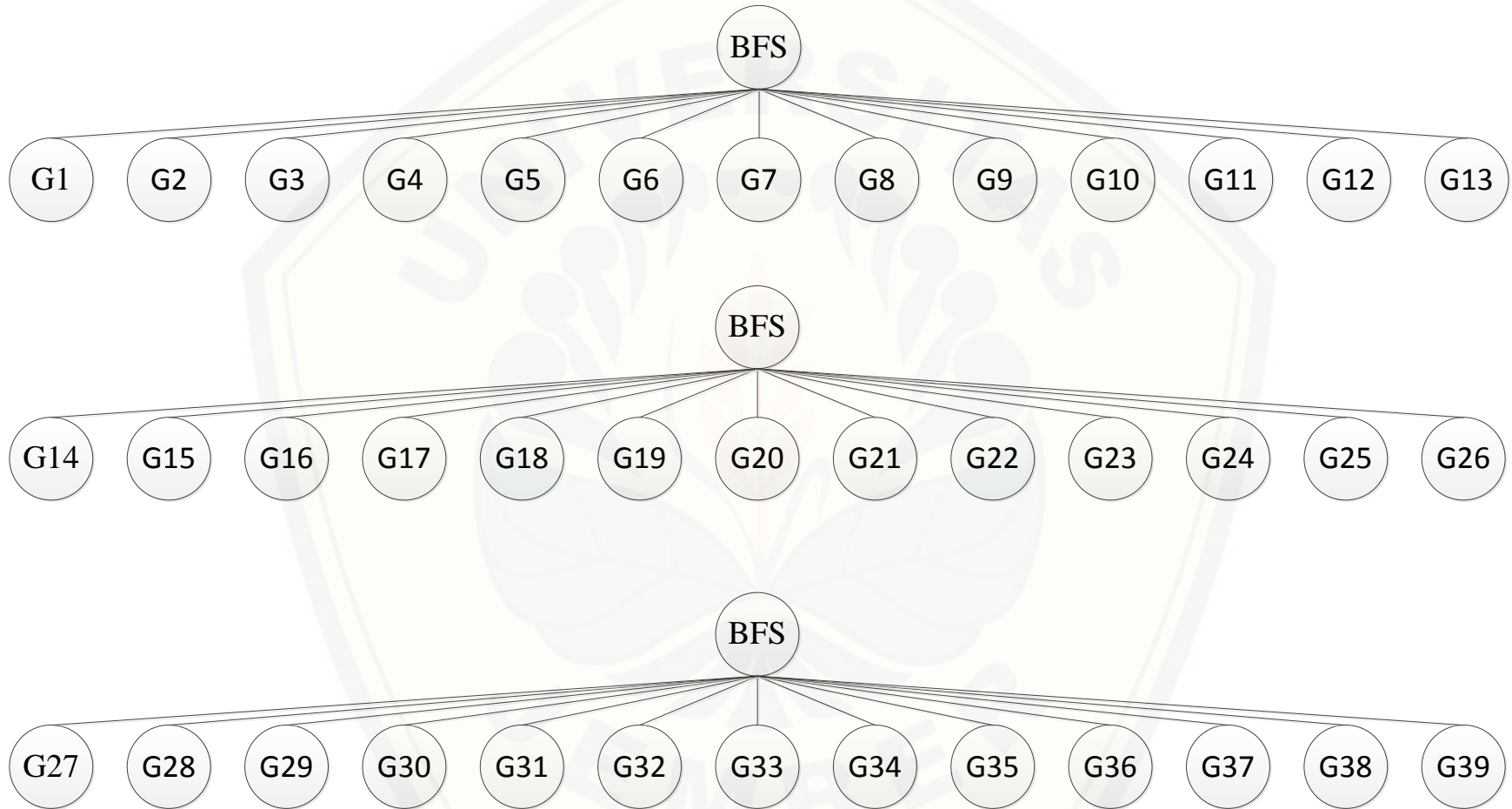
6.  $P_{17}$  = *Melanocytic nevus* (15%)

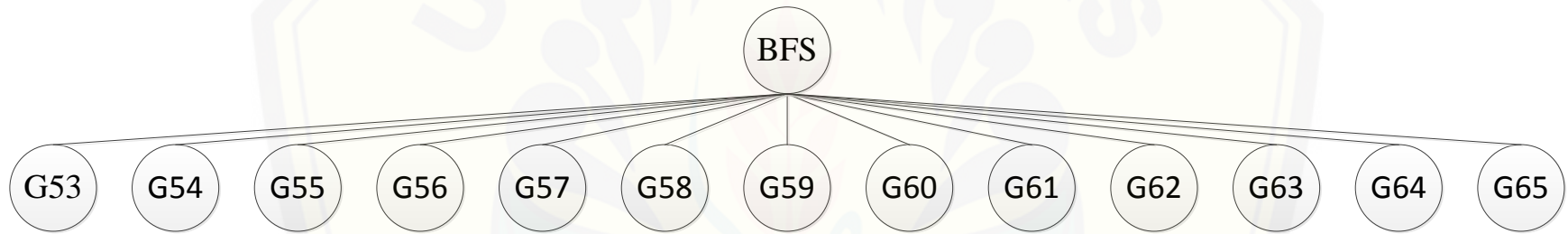
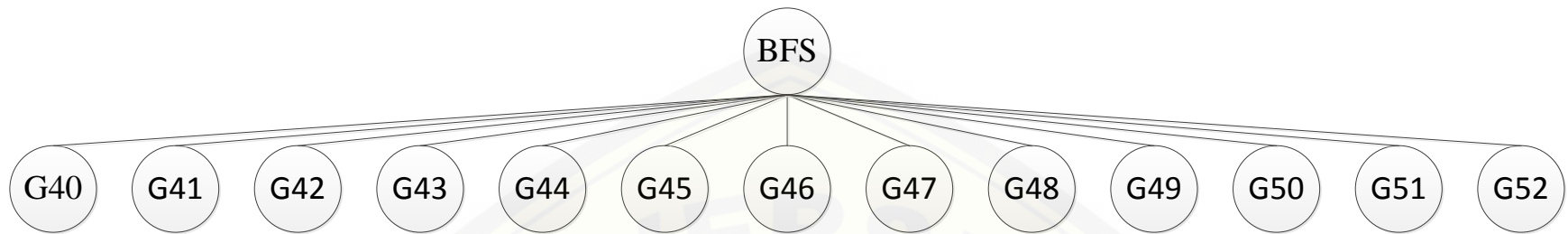
$G_{37}$  = Terdapat benjolan seperti kubah, berwarna samar pada langit-langit mulut bagian depan dan belakang, bagian dalam pipi, bibir dan dasar mulut

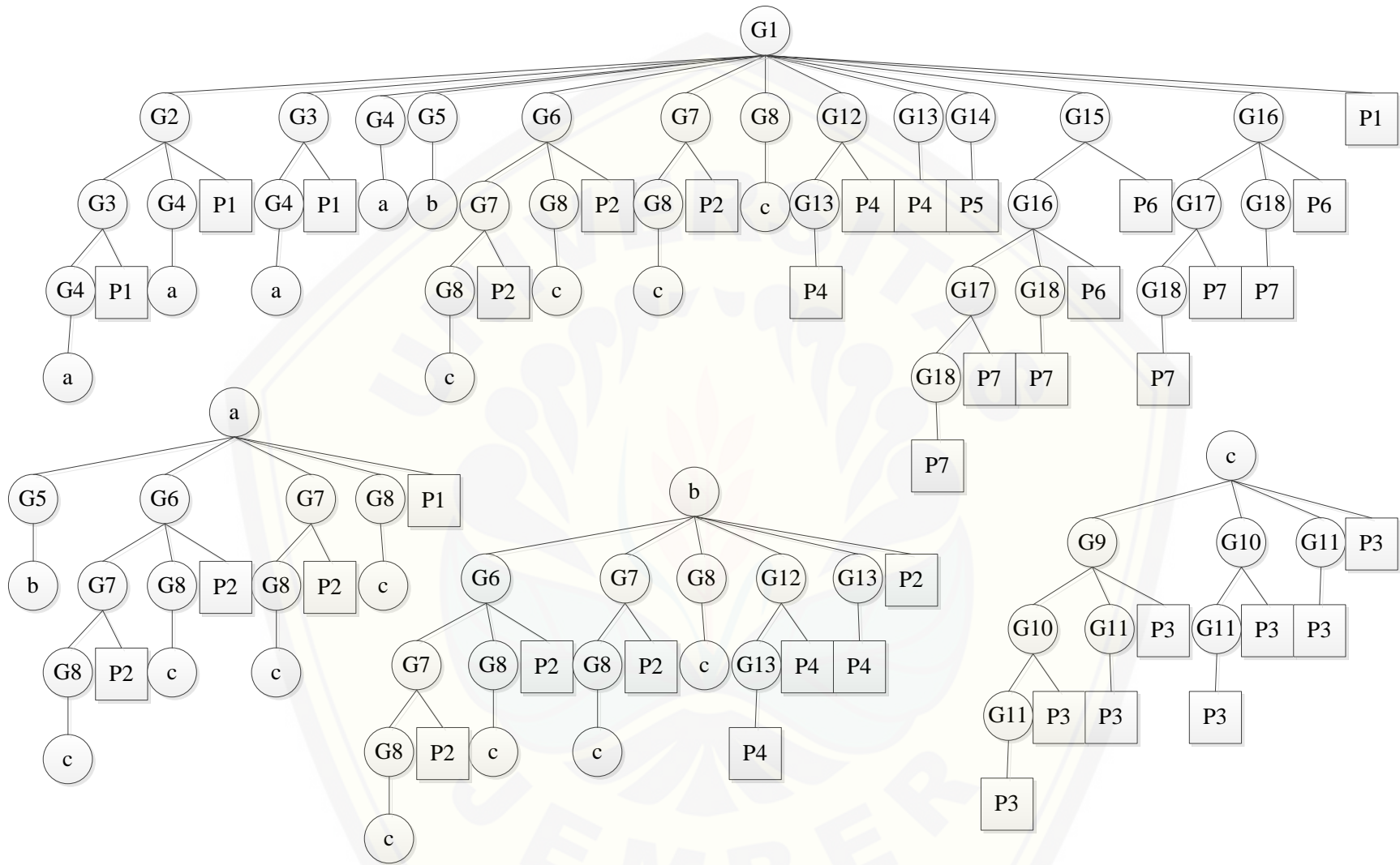
1.  $P_{17}$  = *Melanocytic nevus* (5%)

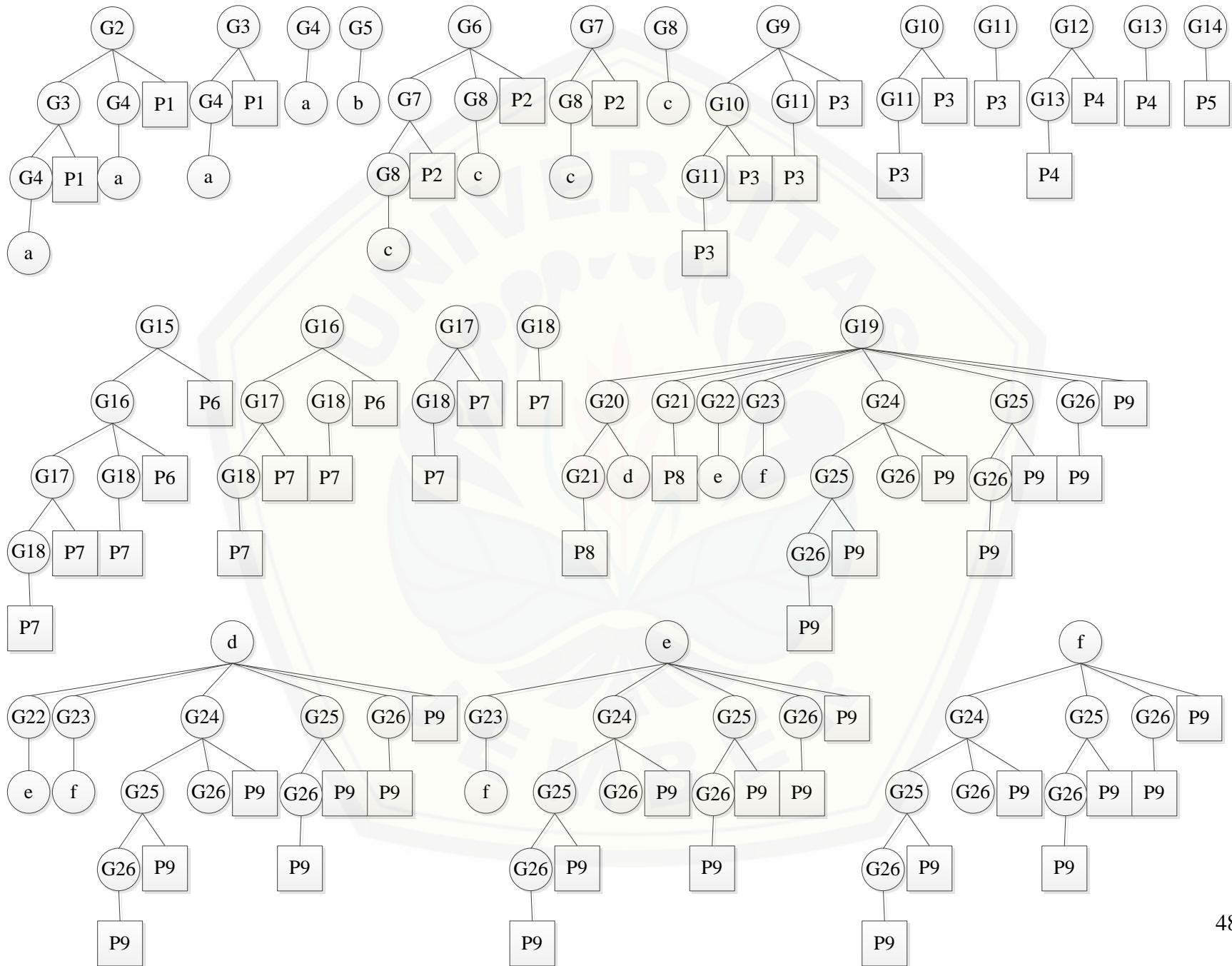
2.  $P_{18}$  = *Hematoma* (2%)

E. Pohon Breadt First Search

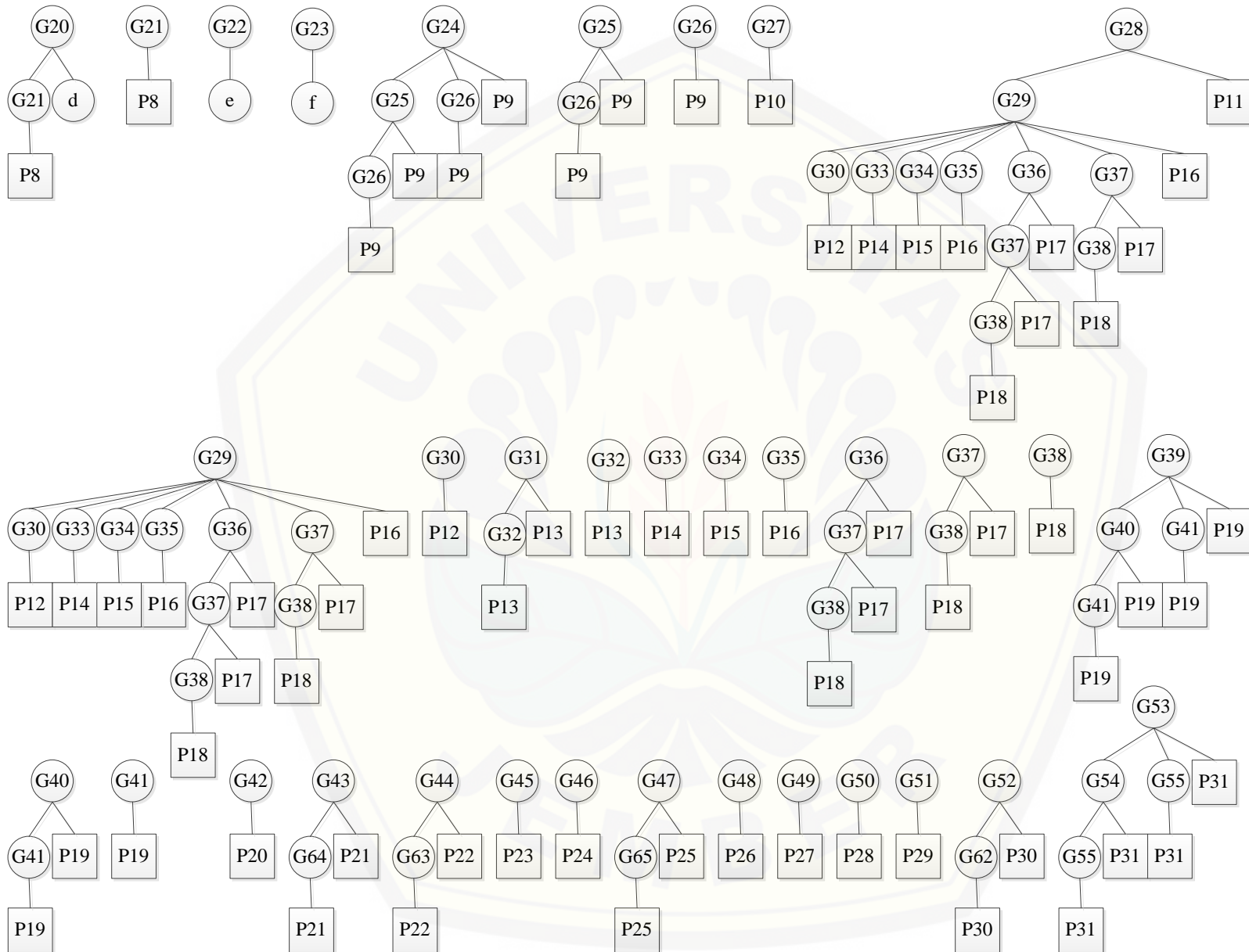


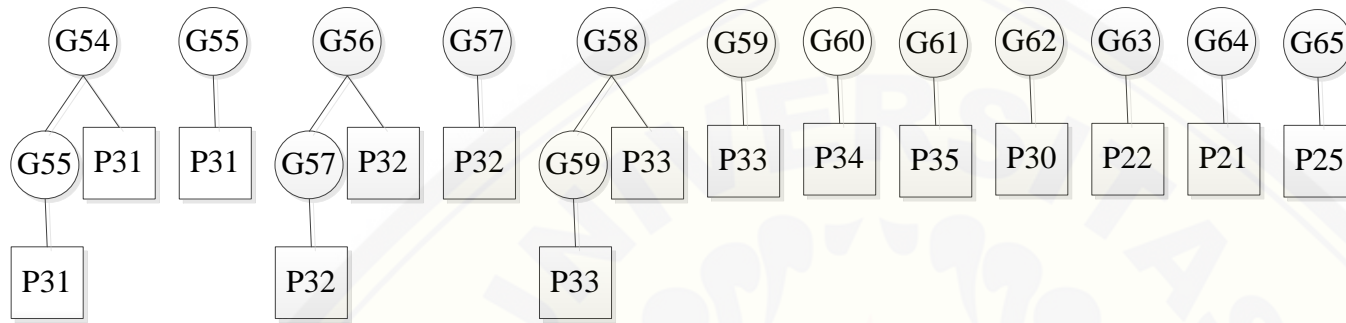












## F. Script MATLAB Aplikasi Diagnosa Penyakit Gigi dan Mulut

## F.1 Script tombol lihat hasil

```

gjl=get(handles.uipanel2,'userdata');
gj=get(handles.text1,'userdata');
t=0;
for i=1:length(gj)
    if gj(i)~=0
        t=t+1;
        mb(t)=gj(i);
    end
end
penyakit=bfs(gjl);
if penyakit==1
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Glossitis']);
    data=importdata('pengobatan1.txt');
    set(handles.text15,'string',data)
elseif penyakit==2
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Kanker Mulut' ]);
    data=importdata('pengobatan2.txt');
    set(handles.text15,'string',data)
elseif penyakit==3
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Abses Gigi' ]);
    data=importdata('pengobatan3.txt');
    set(handles.text15,'string',data)
elseif penyakit==4
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Kanker Gusi Gigi'
]);
    data=importdata('pengobatan4.txt');
    set(handles.text15,'string',data)
elseif penyakit==5
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Maloklusi' ]);
    data=importdata('pengobatan5.txt');
    set(handles.text15,'string',data)
elseif penyakit==6
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Sariawan' ]);
    data=importdata('pengobatan6.txt');
    set(handles.text15,'string',data)
elseif penyakit==7
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Geographic Tongue'
]);
    data=importdata('pengobatan7.txt');
    set(handles.text15,'string',data)
elseif penyakit==8
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Radang Gusi' ]);
    data=importdata('pengobatan8.txt');
    set(handles.text15,'string',data)
elseif penyakit==9
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Periodontitis' ]);
    data=importdata('pengobatan9.txt');
    set(handles.text15,'string',data)
elseif penyakit==10
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Pulpitis' ]);
    data=importdata('pengobatan10.txt');
    set(handles.text15,'string',data)
elseif penyakit==11
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Fordyce Spot' ]);
    data=importdata('pengobatan11.txt');
    set(handles.text15,'string',data)

```

```
elseif penyakit==12
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Retrocuspid
Papila' ]);
    data=importdata('pengobatan12.txt');
    set(handles.text15,'string',data)
elseif penyakit==13
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Fissure Tongue']);
    data=importdata('pengobatan13.txt');
    set(handles.text15,'string',data)
elseif penyakit==14
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit White Sponge
Nevus' ]);
    data=importdata('pengobatan14.txt');
    set(handles.text15,'string',data)
elseif penyakit==15
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Amalgam Tattoo']);
    data=importdata('pengobatan15.txt');
    set(handles.text15,'string',data)
elseif penyakit==16
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Oral Melanotic
Macule' ]);
    data=importdata('pengobatan16.txt');
    set(handles.text15,'string',data)
elseif penyakit==17
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Melanocytic
Nevus' ]);
    data=importdata('pengobatan17.txt');
    set(handles.text15,'string',data)
elseif penyakit==18
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Hematoma']);
    data=importdata('pengobatan18.txt');
    set(handles.text15,'string',data)
elseif penyakit==19
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Ulser Aftosa']);
    data=importdata('pengobatan19.txt');
    set(handles.text15,'string',data)
elseif penyakit==20
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Ulser
Traumatik' ]);
    data=importdata('pengobatan20.txt');
    set(handles.text15,'string',data)
elseif penyakit==21
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Infeksi Herpes
Simplex' ]);
    data=importdata('pengobatan21.txt');
    set(handles.text15,'string',data)
elseif penyakit==22
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Angular
Cheilitis' ]);
    data=importdata('pengobatan22.txt');
    set(handles.text15,'string',data)
elseif penyakit==23
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Papilloma']);
    data=importdata('pengobatan23.txt');
    set(handles.text15,'string',data)
```

```
elseif penyakit==24
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Verruca
Vulgaris']);
    data=importdata('pengobatan24.txt');
    set(handles.text15,'string',data)
elseif penyakit==25
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Irritation
Fibroma']);
    data=importdata('pengobatan25.txt');
    set(handles.text15,'string',data)
elseif penyakit==26
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Peripheral
Ossifying Fibroma']);
    data=importdata('pengobatan26.txt');
    set(handles.text15,'string',data)
elseif penyakit==27
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Peripheral Giant
Cell Granuloma']);
    data=importdata('pengobatan27.txt');
    set(handles.text15,'string',data)
elseif penyakit==28
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Limfangioma']);
    data=importdata('pengobatan28.txt');
    set(handles.text15,'string',data)
elseif penyakit==29
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Neurofibroma']);
    data=importdata('pengobatan29.txt');
    set(handles.text15,'string',data)
elseif penyakit==30
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Epulis
Kongenital']);
    data=importdata('pengobatan30.txt');
    set(handles.text15,'string',data)
elseif penyakit==31
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Radang Mulut']);
    data=importdata('pengobatan31.txt');
    set(handles.text15,'string',data)
elseif penyakit==32
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Hipersensitifitas
Gigi']);
    data=importdata('pengobatan32.txt');
    set(handles.text15,'string',data)
elseif penyakit==33
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Coksackle
Virus']);
    data=importdata('pengobatan33.txt');
    set(handles.text15,'string',data)
elseif penyakit==34
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Caries Dentis']);
    data=importdata('pengobatan34.txt');
    set(handles.text15,'string',data)
elseif penyakit==35
    set(handles.text14,'string',['Anda ' CF(mb) ' menderita penyakit Calculus']);
    data=importdata('pengobatan35.txt');
    set(handles.text15,'string',data)
end
```



```
[r e]=CF(mb);
set(handles.text16,'string',['Faktor Keyakinan (CF) = ' num2str(e)])
data=[{'-1.0 <= CF <= -0.8'}, {'-0.8 <= CF <= -0.6'}, {'-0.6 <= CF <= -0.4'},...
      {'-0.4 <= CF <= -0.2'}, {'-0.2 <= CF <= 0.2'}, {' 0.2 <= CF <= 0.4'},...
      {' 0.4 <= CF <= 0.6'}, {' 0.6 <= CF <= 0.8'}, {' 0.8 <= CF <= 1'}];
CT=[{'Pasti Tidak'},{'Hampir Pasti Tidak'},{'Kemungkinan Besar Tidak'},{'Mungkin
Tidak'},...
    {'Tidak Yakin'},{'Mungkin'},{'Kemungkinan Besar'},{'Hampir Pasti'},{'Pasti'}];
set(handles.uitable2,'data',[ CT' data'],'columnname',{'Certainty Term','CF
Akhir'],'columnwidth',{125,82})
```

## F.2 Script fungsi algoritma BFS

```
function penyakit=bfs(gj1)
for i=1:65
    if gj1(i)==1
        if i==1
            penyakit=cabang1(gj1);
        elseif i==2
            penyakit=cabang2(gj1);
        elseif i==3
            penyakit=cabang3(gj1);
        elseif i==4
            penyakit=cabang4(gj1);
        elseif i==5
            penyakit=cabang5(gj1);
        elseif i==6
            penyakit=cabang6(gj1);
        elseif i==7
            penyakit=cabang7(gj1);
        elseif i==8
            penyakit=cabang8(gj1);
        elseif i==9
            penyakit=cabang9(gj1);
        elseif i==10
            penyakit=cabang10(gj1);
        elseif i==11
            penyakit=cabang11(gj1);
        elseif i==12
            penyakit=cabang12(gj1);
        elseif i==13
            penyakit=cabang13(gj1);
        elseif i==14
            penyakit=cabang14(gj1);
        elseif i==15
            penyakit=cabang15(gj1);
        elseif i==16
            penyakit=cabang16(gj1);
        elseif i==17
            penyakit=cabang17(gj1);
        elseif i==18
            penyakit=cabang18(gj1);
        elseif i==19
            penyakit=cabang19(gj1);
        elseif i==20
            penyakit=cabang20(gj1);
        elseif i==21
            penyakit=cabang21(gj1);
        elseif i==22
            penyakit=cabang22(gj1);
        elseif i==23
            penyakit=cabang23(gj1);
        elseif i==24
            penyakit=cabang24(gj1);
        elseif i==25
            penyakit=cabang25(gj1);
        elseif i==26
            penyakit=cabang26(gj1);
        elseif i==27
            penyakit=cabang27(gj1);
        elseif i==28
            penyakit=cabang28(gj1);
        elseif i==29
            penyakit=cabang29(gj1);
        elseif i==30
            penyakit=cabang30(gj1);
        elseif i==31
            penyakit=cabang31(gj1);
        elseif i==32
            penyakit=cabang32(gj1);
        elseif i==33
            penyakit=cabang33(gj1);
        elseif i==34
            penyakit=cabang34(gj1);
        elseif i==35
            penyakit=cabang35(gj1);
        elseif i==36
            penyakit=cabang36(gj1);
        elseif i==37
            penyakit=cabang37(gj1);
        elseif i==38
            penyakit=cabang38(gj1);
        elseif i==39
            penyakit=cabang39(gj1);
        elseif i==40
            penyakit=cabang40(gj1);
        elseif i==41
            penyakit=cabang41(gj1);
```



```

elseif i==42
    penyakit=cabang42 (gjl);
elseif i==43
    penyakit=cabang43 (gjl);
elseif i==44
    penyakit=cabang44 (gjl);
elseif i==45
    penyakit=cabang45 (gjl);
elseif i==46
    penyakit=cabang46 (gjl);
elseif i==47
    penyakit=cabang47 (gjl);
elseif i==48
    penyakit=cabang48 (gjl);
elseif i==49
    penyakit=cabang49 (gjl);
elseif i==50
    penyakit=cabang50 (gjl);
elseif i==51
    penyakit=cabang51 (gjl);
elseif i==52
    penyakit=cabang52 (gjl);
elseif i==53
    penyakit=cabang53 (gjl);
elseif i==54
    penyakit=cabang54 (gjl);

elseif i==55
    penyakit=cabang55 (gjl);
elseif i==56
    penyakit=cabang56 (gjl);
elseif i==57
    penyakit=cabang57 (gjl);
elseif i==58
    penyakit=cabang58 (gjl);
elseif i==59
    penyakit=cabang59 (gjl);
elseif i==60
    penyakit=cabang60 (gjl);
elseif i==61
    penyakit=cabang61 (gjl);
elseif i==62
    penyakit=cabang62 (gjl);
elseif i==63
    penyakit=cabang63 (gjl);
elseif i==64
    penyakit=cabang64 (gjl);
elseif i==65
    penyakit=cabang65 (gjl);
end
end
end

```

### F.3 Script cabang fungsi algoritma BFS

#### Cabang 1

```

function y=cabang1 (gjl)
if gjl (2)==1
    y=cabang2 (gjl);
end
if gjl (3)==1
    y=cabang3 (gjl);
end
if gjl (4)==1
    y=cabang4 (gjl);
end
if gjl (5)==1
    y=cabang5 (gjl);
end
if gjl (6)==1
    y=cabang6 (gjl);
end
if gjl (7)==1
    y=cabang7 (gjl);
end
if gjl (8)==1
    y=cabang8 (gjl);
end

if gjl (12)==1
    y=cabang12 (gjl);
end
if gjl (13)==1
    y=cabang13 (gjl);
end
if gjl (14)==1
    y=cabang14 (gjl);
end
if gjl (15)==1
    y=cabang15 (gjl);
end
if gjl (16)==1
    y=cabang16 (gjl);
end
if gjl (2)==0 && gjl (3)==0 &&
gjl (4)==0 && gjl (5)==0 && gjl (6)==0
&& gjl (7)==0 && gjl (8)==0 &&
gjl (12)==0 && gjl (13)==0 &&
gjl (14)==0 && gjl (15)==0 &&
gjl (16)==0
    y=1;
end

```

```

Cabang 2
function y=cabang2(gjl)
if gjl(3)==1
    y=cabang3(gjl);
end
if gjl(4)==1
    y=cabang4(gjl);
end
if gjl(3)==0 && gjl(4)==0
    y=1;
end
Cabang 3
function y=cabang3(gjl)
if gjl(4)==1
    y=cabang4(gjl);
else
    y=1;
end
cabang 4
function y=cabang4(gjl)
if gjl(5)
    y=cabang5(gjl);
end
if gjl(6)
    y=cabang6(gjl);
end
if gjl(7)
    y=cabang7(gjl);
end
if gjl(8)
    y=cabang8(gjl);
end
if gjl(5)==0 && gjl(6)==0 && gjl(7)==0
&& gjl(8)==0
    y=1;
end
cabang 5
function y=cabang6(gjl)
if gjl(7)==1
    y=cabang7(gjl);
end
if gjl(8)==1
    y=cabang8(gjl);
end
if gjl(7)==0 && gjl(8)==0
    y=2;
end
cabang 6
function y=cabang6(gjl)
if gjl(7)==1
    y=cabang7(gjl);
end
if gjl(8)==1
    y=cabang8(gjl);
end
if gjl(7)==0 && gjl(8)==0
    y=2;
end
cabang 7
function y=cabang7(gjl)
if gjl(8)==1
    y=cabang8(gjl);
else
    y=2;
end
cabang 8
function y=cabang8(gjl)
if gjl(9)==1
    y=cabang9(gjl);
end
if gjl(10)==1
    y=cabang10(gjl);
end
if gjl(11)==1
    y=cabang11(gjl);
end
if gjl(9)==0 && gjl(10)==0 && gjl(10)==0
    y=2;
end
cabang 9
function y=cabang9(gjl)
if gjl(10)==1
    y=cabang10(gjl);
end
if gjl(11)==1
    y=cabang11(gjl);
end
if gjl(10)==0 && gjl(11)==0
    y=3;
end
cabang 10
function y=cabang10(gjl)
if gjl(11)==1
    y=cabang11(gjl);
else
    y=3;
end
cabang 11
function y=cabang11(gjl)
y=3;
end
cabang 12
function y=cabang12(gjl)
if gjl(13)==1
    y=cabang13(gjl);
else
    y=4;
end

```

**Cabang 13**

```
function y=cabang13(gjl)
y=4;
```

**cabang 14**

```
function y=cabang14(gjl)
y=5;
```

**cabang 15**

```
function y=cabang15(gjl)
if gjl(16)==1
    y=cabang16(gjl);
else
    y=6;
```

```
end
```

**cabang 16**

```
function y=cabang16(gjl)
if gjl(17)==1
    y=cabang17(gjl);
end
if gjl(18)==1
    y=cabang18(gjl);
end
if gjl(17)==0 && gjl(18)==0
    y=6;
```

```
end
```

**cabang 17**

```
function y=cabang17(gjl)
if gjl(18)==1
    y=cabang18(gjl);
else
    y=7;
```

```
end
```

**cabang 18**

```
function y=cabang18(gjl)
y=7;
```

**cabang 19**

```
function y=cabang19(gjl)
if gjl(20)==1
    y=cabang20(gjl);
end
if gjl(21)==1
    y=cabang21(gjl);
end
if gjl(22)==1
    y=cabang22(gjl);
end
if gjl(23)==1
    y=cabang23(gjl);
end
if gjl(24)==1
    y=cabang24(gjl);
end
if gjl(25)==1
    y=cabang25(gjl);
end
if gjl(26)==1
    y=cabang26(gjl);
end
```

```
if gjl(20)==0 && gjl(21)==0 &&
gjl(22)==0 && gjl(23)==0 && gjl(24)==0
&& gjl(25)==0 && gjl(26)==0
    y=9;
```

```
end
```

**cabang 20**

```
function y=cabang20(gjl)
if gjl(21)==1
    y=cabang21(gjl);
end
if gjl(22)==1
    y=cabang22(gjl);
end
if gjl(23)==1
    y=cabang23(gjl);
end
if gjl(24)==1
    y=cabang24(gjl);
end
if gjl(25)==1
    y=cabang25(gjl);
end
if gjl(26)==1
    y=cabang26(gjl);
```

```
end
```

```
if gjl(21)==0 && gjl(22)==0 &&
gjl(23)==0 && gjl(24)==0 && gjl(25)==0
&& gjl(26)==0
    y=9;
```

```
end
```

**cabang 21**

```
function y=cabang21(gjl)
y=8;
```

**cabang 22**

```
function y=cabang22(gjl)
if gjl(23)==1
    y=cabang23(gjl);
end
if gjl(24)==1
    y=cabang24(gjl);
end
if gjl(25)==1
    y=cabang25(gjl);
end
if gjl(26)==1
    y=cabang26(gjl);
```

```
end
```

```
if gjl(23)==0 && gjl(24)==0 &&
gjl(25)==0 && gjl(26)==0
    y=9;
```

```
end
```

**cabang 23**

```
function y=cabang23(gjl)
if gjl(24)==1
    y=cabang24(gjl);
end
if gjl(25)==1
    y=cabang25(gjl);
end
```

```

if gjl(26)==1
    y=cabang26(gjl);
end
if gjl(24)==0 && gjl(25)==0 &&
gjl(26)==0
    y=9;
end
cabang 24
function y=cabang24(gjl)
if gjl(25)==1
    y=cabang25(gjl);
end
if gjl(26)==1
    y=cabang26(gjl);
end
if gjl(25)==0 && gjl(26)==0
    y=9;
end
cabang 25
function y=cabang25(gjl)
if gjl(26)==1
    y=cabang26(gjl);
else
    y=9;
end
cabang 26
function y=cabang26(gjl)
y=9;
cabang 27
function y=cabang27(gjl)
y=10;
cabang 28
function y=cabang28(gjl)
if gjl(29)==1
    y=cabang29(gjl);
else
    y=11;
end
cabang 29
function y=cabang29(gjl)
if gjl(30)==1
    y=cabang30(gjl);
end
if gjl(33)==1
    y=cabang33(gjl);
end
if gjl(34)==1
    y=cabang34(gjl);
end
if gjl(35)==1
    y=cabang35(gjl);
end
if gjl(36)==1
    y=cabang36(gjl);
end
if gjl(37)==1
    y=cabang37(gjl);
end

if gjl(30)==0 && gjl(33)==0 &&
gjl(34)==0 && gjl(35)==0 && gjl(36)==0
&& gjl(37)==0
    y=16;
end
cabang 30
function y=cabang30(gjl)
y=12;
cabang 31
function y=cabang31(gjl)
if gjl(32)==1
    y=cabang32(gjl);
else
    y=13;
end
cabang 32
function y=cabang32(gjl)
y=13;
cabang 33
function y=cabang33(gjl)
y=14;
cabang 34
function y=cabang34(gjl)
y=15;
cabang 35
function y=cabang35(gjl)
y=16;
cabang 36
function y=cabang36(gjl)
if gjl(37)==1
    y=cabang37(gjl);
else
    y=17;
end
cabang 37
function y=cabang37(gjl)
if gjl(38)==1
    y=cabang38(gjl);
else
    y=17;
end
cabang 38
function y=cabang38(gjl)
y=18;
cabang 39
function y=cabang39(gjl)
if gjl(40)==1
    y=cabang40(gjl);
end
if gjl(41)==1
    y=cabang41(gjl);
end
if gjl(40)==0 && gjl(41)==0
    y=19;
end

```

**Cabang 40**

```

function y=cabang40(gjl)
if gjl(41)==1
    y=cabang41(gjl);
else
    y=19;
end
cabang 41
function y=cabang41(gjl)
y=19;
cabang 42
function y=cabang42(gjl)
y=20;
cabang 43
function y=cabang43(gjl)
if gjl(64)==1
    y=cabang64(gjl);
else
    y=21;
end
cabang 44
function y=cabang44(gjl)
if gjl(63)==1
    y=cabang63(gjl);
else
    y=22;
end
cabang 45
function y=cabang45(gjl)
y=23;
cabang 46
function y=cabang46(gjl)
y=24;
cabang 47function y=cabang47(gjl)
if gjl(65)==1
    y=cabang65(gjl);
else
    y=25;
end
cabang 48
function y=cabang48(gjl)
y=26;
cabang 49
function y=cabang49(gjl)
y=27;
cabang 50
function y=cabang50(gjl)
y=28;
cabang 51
function y=cabang51(gjl)
y=29;
cabang 52
function y=cabang52(gjl)
if gjl(62)==1
    y=cabang62(gjl);
else
    y=30;
end

```

**Cabang 53**

```

function y=cabang53(gjl)
if gjl(54)==1
    y=cabang54(gjl);
end
if gjl(55)==1
    y=cabang55(gjl);
end
if gjl(54)==0 && gjl(55)==0
    y=31;
end
cabang 54
function y=cabang54(gjl)
if gjl(55)==1
    y=cabang55(gjl);
else
    y=31;
end
cabang 55
function y=cabang55(gjl)
y=31;
cabang 56
function y=cabang56(gjl)
if gjl(57)==1
    y=cabang57(gjl);
else
    y=32;
end
cabang 57
function y=cabang57(gjl)
y=32;
cabang 58
function y=cabang58(gjl)
if gjl(59)==1
    y=cabang59(gjl);
else
    y=33;
end
cabang 59
function y=cabang59(gjl)
y=33;
cabang 60
function y=cabang60(gjl)
y=34;
cabang 61
function y=cabang61(gjl)
y=35;
cabang 62
function y=cabang62(gjl)
y=30;
cabang 63
function y=cabang63(gjl)
y=22;
cabang 64
function y=cabang64(gjl)
y=21;
cabang 65
function y=cabang65(gjl)
y=25;

```

## F.4 Script fungsi CF

```
function [Cf Cef]=CF(MB)
if length(MB)==1
    MD=1-MB;
    Cef=MB-MD;
else
    for i=1:length(MB)
        MD(i)=1-MB(i);
    end
    p=MB(1);
    q=MD(1);
    for i=1:length(MB)
        if i~=length(MB)
            C(i)=p+MB(i+1)*(1-p);
            C1(i)=q+MD(i+1)*(1-q);
            p=C(i);
            q=C1(i);
        end
    end
    Cef=C(i-1)-C1(i-1);
end
if Cef>=-1.0 && Cef <=-0.8
    Cf='Pasti Tidak';
elseif Cef>=-0.8 && Cef <=-0.6
    Cf='Hampir Pasti Tidak';
elseif Cef>=-0.6 && Cef <=-0.4
    Cf='Kemungkinan Besar Tidak';
elseif Cef>=-0.4 && Cef <=-0.2
    Cf='Mungkin Tidak';
elseif Cef>=-0.2 && Cef <=0.2
    Cf='Tidak Yakin';
elseif Cef>=0.2 && Cef <=0.4
    Cf='Mungkin';
elseif Cef>=0.4 && Cef <=0.6
    Cf='Kemungkinan Besar';
elseif Cef>=0.6 && Cef <=0.8
    Cf='Hampir Pasti';
elseif Cef>=0.8 && Cef <=1.0
    Cf='Pasti';
end
```



