

## BAB II TINJAUAN PUSTAKA

### 2.1 Tinjauan Pustaka

Penelitian yang berhubungan dengan pengembangan Sistem Pakar serta penerapan *Dempster Shafer* (DS) dan *Certainty Factor* (CF) antara lain dilakukan oleh Atmaja (2014), Mustikadewi (2014), Wahyuni dan Prijodiprojo (2013), Setyarini, Putra, dan Purnawan (2013). Menurut Atmaja (2014), penerapan metode *Certainty Factor* dalam melakukan diagnosis pada penyakit kelinci memiliki persentase keakuratan yang cukup tinggi yakni sebesar 80% dengan nilai rata-rata kepastian 0.74. Dengan persentase yang cukup tinggi tersebut, penerapan metode *Certainty Factor* dalam melakukan diagnosis penyakit kelinci hias cukup dapat diandalkan tingkat keakuratannya.

Pada penelitian lain yang dilakukan oleh Mustikadewi (2014), sistem pakar yang dibangun dapat mengidentifikasi penyakit sapi serta pengambilan kesimpulan diagnosis dihitung menggunakan metode *Dempster Shafer* dengan menggunakan masukan gejala fakta dari pengguna. Penerapan metode *Dempster Shafer* pada kasus diagnosis penyakit sapi ini juga mampu menghasilkan tingkat akurasi sebesar 88,89% dengan tingkat keyakinan pakar yang berbeda-beda setiap diagnosis. Sedangkan menurut Setyarini, Putra, dan Purnawan (2013) disebutkan bahwa terdapat perbandingan yang relevan antara penggunaan metode *Certainty Factor* dan metode *Dempster Shafer*. Metode *Certainty Factor* memiliki perhitungan yang lebih sederhana bila dibandingkan dengan metode *Dempster Shafer*. *Dempster Shafer* merupakan metode yang lebih baik daripada *Certainty Factor* karena metode DS dalam menentukan hasil dari persentase nilai keyakinan mempertimbangkan nilai dari semua variabel yang digunakan dalam kombinasi *Dempster's Rule of Combination* yang menghasilkan perhitungan nilai-nilai yang lebih bervariasi dan akurat. Dari 10 kasus yang dicoba dalam penelitian ini, rata-rata nilai persentase yang diperoleh

dengan menggunakan metode *Dempster Shafer* adalah 76%. Sedangkan dengan menggunakan metode *Certainty Factor* adalah sebesar 73%.

Berdasarkan tinjauan di atas, maka penulis akan mencoba membandingkan tingkat keakuratan sistem dalam penggunaan metode *Certainty Factor* dalam penelitian yang telah dilakukan oleh Atmaja (2014) dengan metode *Dempster Shafer*.

## **2.2 Landasan Teori**

### **2.2.1 Sistem Pakar**

Secara umum sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang dilakukan oleh ahli. Menurut Martin dan Oxman (1998), sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah, yang biasanya hanya dapat diselesaikan oleh seorang pakar dalam bidang tertentu. Sistem pakar adalah salah satu cabang kecerdasan buatan yang menggunakan pengetahuan-pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu (Giarrantano dan Rieley, 2005).

### **2.2.2 Dempster Shafer**

Teori *Dempster-Shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief function* dan *plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer.

Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval

$$[ \textit{Belief}, \textit{Plausability} ] \quad [2.1]$$

*Belief* ( *Bel* ) adalah total ukuran kekuatan *evidence* dalam mendukung suatu himpunan bagian.

Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian.

*Plausability* ( *Pl* ) dinotasikan sebagai :

$$Pl(s) = 1 - Bel(\neg s) \quad [2.2]$$

*Plausability* juga bernilai 0 sampai 1. Jika kita yakin akan  $\neg s$ , maka dapat dikatakan bahwa  $Bel(\neg s) = 1$ , dan  $Pl(\neg s) = 0$  (Kusumadewi, 2003).

Pada teori *Dempster-Shafer* dikenal adanya *frames of discernment* yang dinotasikan dengan simbol ( $\theta$ ) dan *mass function* yang dinotasikan dengan  $m$ . *Frames of discernment* ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment*, dimana :

$$\theta = \{\theta_1, \theta_2, \dots, \theta_N\} \quad [2.3]$$

Dimana :

$\theta$  = *frame of discernment* atau *environment*  
 $\theta_1, \theta_2, \dots, \theta_N$  = elemen/ unsur bagian dalam *environment*

*Environment* mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban, dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan  $P(\theta)$ , setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m: P(\theta) \rightarrow [0,1]$$

Sehingga dapat dirumuskan :

$$\sum_{X \in P(\theta)} m(X) = 1 \quad [2.4]$$

Dimana :

$P(\theta)$  = power set

$m(X)$  = mass function (X)

Sebagai contoh :

$P(\text{hostile}) = 0,7$

$P(\text{non-hostile}) = 1 - 0,7 = 0,3$

Pada contoh di atas *belief* dari *hostile* adalah 0,7 sedangkan *disbelief hostile* adalah sebesar 0,3. Dalam teori *Dhempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan dengan  $m(\theta)$ .

Sedangkan *mass function* ( $m$ ) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan  $m$ .

Pada aplikasi suatu sistem pakar pada suatu penyakit terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosis suatu penyakit. Untuk mengatasi jumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination*.

$$m_1 \oplus m_2(Z) = \sum_{X \cap Y = Z} m_1(X) m_2(Y) \quad [2.5]$$

Dimana :

$m_1 \oplus m_2(Z)$  = mass function dari *evidence* (Z)

$m_1(X)$  = *mass function* dari *evidence* (X), yang diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut.

$m_2(Y)$  = *mass function* dari *evidence* (Y), yang diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut.

$\oplus$  = *operator direct sum*

Secara umum formulasi untuk *Dempster's Rule of Combination* adalah :

$$m_1 \oplus m_2(Z) = \frac{\sum_{X \cap Y = Z} m_1(X)m_2(Y)}{1 - \kappa} \quad [2.6]$$

Dimana :

$\kappa$  = jumlah *evidential conflict*

Besar jumlah *evidential conflict* ( $\kappa$ ) dirumuskan dengan :

$$\kappa = \sum_{X \cap Y = \phi} m_1(X)m_2(Y) \quad [2.7]$$

Sehingga bila persamaan (2.6) disubstitusikan ke persamaan (2.7) akan menjadi :

$$m_1 \oplus m_2(Z) = \frac{\sum_{X \cap Y = Z} m_1(X)m_2(Y)}{1 - \sum_{X \cap Y = \phi} m_1(X)m_2(Y)} \quad [2.8]$$

Dimana :

$m_1 \oplus m_2(Z)$  = *mass function* dari *evidence* (Z)

$m_1(X)$  = *mass function* dari *evidence* (X), yang diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut.

$m_2(Y)$  = *mass function* dari *evidence* (Y), yang diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut.

$\kappa$  = jumlah *evidential conflict*

$\sum_{X \cap Y = Z} m_1(X)m_2(Y)$  = merupakan nilai kekuatan dari *evidence* Z yang diperoleh dari kombinasi nilai keyakinan sekumpulan *evidence*.

Sebagai contoh ada 2 jenis penyakit:

1. Parasit Kulit (P01)

Mempunyai gejala-gejala sebagai berikut :

- G01. Kulit Kemerahan (0,3)
- G02. Merasa Gatal (0,7)
- G03. Kulit Bersisik (0,4)

2. Ring Worm (P02)

Mempunyai gejala-gejala sebagai berikut :

- G02. Merasa Gatal (0,7)
- G05. Iritasi Dibagian Kepala/kaki (0,4)
- G06. Terdapat Bercak Merah (0,6)
- G07. Bulu Menggumpal/kusut (0,7)

Jika kelinci Toby mengalami gejala-gejala seperti :

- 1. Merasa gatal (G02)
- 2. Kulit bersisik (G03)
- 3. Bulu menggumpal (G07)

Dimana dari 3 gejala tersebut saling terkait dengan 2 penyakit diatas, maka

Gejala	Penyakit	
	P01	P02
G02	*	*
G03	*	

Tabel 2.1 Keterkaitan antar gejala dengan penyakit

Sehingga untuk memperoleh nilai keyakinan dengan *Dempster-Shafer* dari gejala di atas adalah:

1. Gejala 1 : Merasa Gatal (G02)

$$\begin{aligned} m_1\{P01,P02\} &= 0.7 \\ m_1\{\theta\} &= 1 - m_1\{P01,P02\} \\ &= 1 - 0,7 = 0,3 \end{aligned}$$

2. Gejala 2 : Kulit Bersisik (G03)

$$\begin{aligned} m_2\{P01\} &= 0,4 \\ m_2\{\theta\} &= 1 - m_2\{P01\} \\ &= 1 - 0,4 = 0,6 \end{aligned}$$

Jika diilustrasikan dalam suatu tabel :

Gejala 1	Gejala 2	$m_2\{P01\}$	(0,4)	$m_2\{\theta\}$	(0,6)
$m_1\{P01,P02\}$	(0,7)	{P01}	(0,28)	{P01.P02}	(0,42)
$m_1\{\theta\}$	(0,3)	{P01}	(0,12)	$\theta$	(0,18)

Selanjutnya menghitung tingkat keyakinan ( $m$ ) *combine* dengan rumus (2.8) maka dapat dihitung nilai keyakinan sebagai berikut :

$$m_1 \oplus m_2 Z\{P01\} = \frac{0,7 * 0,4}{1 - 0} = 0,28$$

$$m_1 \oplus m_2 Z\{P01, P02\} = \frac{0,7 * 0,6}{1 - 0} = 0,42$$

$$m_1 \oplus m_2 Z\{\theta\} = \frac{0,3 * 0,4}{1 - 0} = 0,12$$

$$m_1 \oplus m_2 Z\{\theta\} = \frac{0,3 * 0,6}{1 - 0} = 0,18$$

Dengan adanya 2 nilai { P01 } maka :  $0,28 + 0,12 = 0,40$

Nilai keyakinan paling kuat dari 2 gejala G02 dan G03 adalah terhadap penyakit { P01,P02 } sebesar 0,4

3. Gejala 3 : Bulu Menggumpal (G07)

$$\begin{aligned}
 m_3\{P02\} &= 0,7 \\
 m_3\{\theta\} &= 1 - m_3\{P02\} \\
 &= 1 - 0,7 = 0,3
 \end{aligned}$$

Jika diilustrasikan dalam suatu tabel :

Gejala 1&2	Gejala 3	$m_3\{P02\}$	$(0,7)$	$m_3\{\theta\}$	$(0,3)$
$m_2\{P01\}$	$(0,40)$	k	$(0,28)$	{P01}	$(0,12)$
$m_2\{P01,P02\}$	$(0,42)$	{P02}	$(0,294)$	{P01,P02}	$(0,126)$
$m_2\{\theta\}$	$(0,18)$	{P02}	$(0,126)$	$\theta$	$(0,054)$

Selanjutnya menghitung tingkat keyakinan ( $m$ ) *combine* dengan rumus (2.8) maka dapat dihitung nilai keyakinan sebagai berikut :

$$\begin{aligned}
 m_1 \oplus m_2 Z\{P01\} &= \frac{0.12}{1-k} = \frac{0.12}{1-0.28} = 0.167 \\
 m_1 \oplus m_2 Z\{P02\} &= \frac{0.294 + 0.126}{1-k} = \frac{0.42}{1-0.28} = 0.583 \\
 m_1 \oplus m_2 Z\{P01, P02\} &= \frac{0.126}{1-k} = \frac{0.126}{1-0.28} = 0.175 \\
 m_1 \oplus m_2 Z\{\theta\} &= \frac{0.054}{1-k} = \frac{0.054}{1-0.28} = 0.075
 \end{aligned}$$

Dengan adanya 3 gejala yaitu G02, G03, dan G07 ini, nilai keyakinan paling kuat adalah terhadap penyakit {P02} yaitu sebesar 0.583 atau 58%. Sedangkan untuk penyakit {P01} hanya sebesar 0.167 atau 17%

**2.2.3 Faktor Kepastian (Certainty Factor)**



Faktor Kepastian merupakan cara penggabungan dari kepercayaan (*belief*) dan ketidakpercayaan (*disbelief*) dalam bilangan yang tunggal. Menurut Arhami (2005), dalam *certainty theory* data-data kualitatif direpresentasikan sebagai derajat keyakinan (*degree of belief*). Dalam mengekspresikan derajat keyakinan digunakan suatu nilai yang disebut *certainty factor* (CF) untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data.

Berikut ini adalah formulasi dasar CF (Giarrantano & Riley, 2005) :

$$CF[h, e] = MB[h, e] - MD[h, e] \quad [2.9]$$

CF = *Certainty Factor* (faktor kepastian) dalam hipotesis H yang dipengaruhi oleh fakta E.

MB = *Measure of Belief* (tingkat keyakinan), adalah ukuran kenaikan dari kepercayaan hipotesis H dipengaruhi oleh fakta E.

MD = *Measure of Disbelief* (tingkat ketidakpercayaan), adalah kenaikan dari ketidakpercayaan hipotesis H dipengaruhi fakta E.

E = *Evidence* (peristiwa atau fakta).

H = Hipotesis (Dugaan).

Dari rumus dasar diatas, ada 3 hal yang mungkin terjadi pada *Certainty Factor* (CF):

1. Beberapa *evidence* dikombinasikan untuk menentukan CF dari suatu hipotesis.

Jika e1 dan e2 adalah observasi, maka:

$$MB[h, e1 \wedge e2] = \begin{cases} 0 & MB[h, e1 \wedge e2] = 1 \\ MB[h, e1] + MB[h, e2]. (1 - MB[h, e1]) & \end{cases} \quad [2.10]$$

$$MD[h, e1 \wedge e2] = \begin{cases} 0 & MD[h, e1 \wedge e2] = 1 \\ MD[h, e1] + MD[h, e2]. (1 - MD[h, e1]) & \end{cases} \quad [2.11]$$

2. Perhitungan CF dengan Multiple Antecedent

<i>Evidence, E</i>	<i>Antecedent Certainty</i>	
E <sub>1</sub> AND E <sub>2</sub>	min [CF(H,E <sub>1</sub> ), CF(H,E <sub>2</sub> )]	[2.12]
E <sub>1</sub> OR E <sub>2</sub>	max [CF(H,E <sub>1</sub> ), CF(H,E <sub>2</sub> )]	
NOT E	-CF(H,E)	

$$3. CF_{kombinasi}(CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{keduanya} > 0 \\ \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & \text{salah satu} < 0 \\ CF_1 + CF_2(1 + CF_1) & \text{keduanya} < 0 \end{cases} \quad [2.13]$$

Dengan MB[h,s] adalah ukuran kepercayaan h berdasarkan keyakinan penuh terhadap *validitas* s.

Sebagai contoh dengan kasus yang sama dalam perhitungan menggunakan metode *Dempster Shafer* di atas, yaitu : (P01) Parasit Kulit dengan gejala (G01) kulit kemerahan, (G02) merasa gatal, (G03) kulit bersisik, dan (P02) yang memiliki gejala (G02) merasa gatal, (G05) iritasi kulit di bagian kepala/kaki, (G06) terdapat bercak merah, (G07) bulu menggumpal/kusut dengan gejala penyakit yang dialami oleh kelinci yaitu (G02) merasa gatal, (G03) kulit bersisik, dan (G07) bulu menggumpal. Maka dapat dibentuk aturan seperti berikut :

IF G02 AND G03 AND G07 THEN P01 OR P02

Dimana CF P01 sebesar 0,5 dan CF P02 sebesar 0,7 dengan bobot setiap gejala yang dialami sama dengan besar bobot dalam menggunakan metode DS maka bobot G02 (0,7), G03 (0,4), G07 (0,7). Dari data tersebut dapat dihitung nilai kepastiannya sebagai berikut :

$$\begin{aligned} CF(E,e) &= CF(G02 \cap G03 \cap G07) \\ &= \min [ CF(G02,e) \cap CF(G03,e) \cap CF(G07,e) ] \\ &= \min [ 0,7 , 0,4 , 0,7 ] \\ &= 0,4 \end{aligned}$$

CF Hipotesa dapat dihitung sebagai berikut :

$$CF (P01,e) = CF (E,e) \times CF (P01)$$

$$= 0,4 \times 0,5$$

$$= 0,2$$

$$CF (P02,e) = CF (E,e) \times CF (P02)$$

$$= 0,4 \times 0,7$$

$$= 0,28$$

Sehingga diperoleh nilai kepercayaan kelinci yang didiagnosis menderita (P01) parasit kulit sebesar 20% atau (P02) ringworm sebesar 28%.

#### **2.2.4 Penyakit Pada Kelinci**

Menurut Atmaja (2014), sebagai binatang peliharaan kelinci mulai banyak diminati sejak sekitar tahun 2003. Pada tahun tersebut mulai banyak *pet shop* yang menjual berbagai jenis anakan kelinci. Pamor kelinci sebagai binatang peliharaan makin menanjak sejak banyak digelarnya kontes-kontes yang memamerkan keindahan kelinci. Selain sebagai binatang peliharaan, kelinci juga banyak dijadikan sebagai bahan konsumsi maupun digunakan sebagai bahan percobaan di laboratorium.

Sebagai binatang peliharaan, para pemelihara kelinci pasti berharap jika kelinci-kelinci tersebut sehat. Hal itu memerlukan perawatan yang teratur, termasuk perawatan kesehatan tubuh. Maka dari itu, pemelihara kelinci harus mengetahui penyakit yang umumnya diderita kelinci serta gejala-gejalanya, sehingga pemelihara dapat melakukan pertolongan awal sebelum ke dokter.

Penyakit yang umumnya diderita kelinci adalah sebagai berikut (Hustamin,2006) :

##### **1. Enteritis Kompleks**

Penyakit ini menyerang alat pencernaan, dan menjadi penyebab kematian paling umum pada kelinci di peternakan. Korbannya anak-anak kelinci yang masih

menyusu. Anak kelinci yang sembuh dari penyakit ini pertumbuhan selanjutnya kurang baik.

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu kelinci berdiri dengan posisi membungkuk, kaki depan agak maju, daun telinga turun, mata suram memicing, gigi berkerot menahan sakit, dehidrasi (apabila ditarik kulitnya akan kembali sangat lama, gusi ditekan terlihat putih), kotorannya hijau gelap dan bau, lendir menggantung pada dubur, nafsu makan turun, pertumbuhan terlambat, telinga pucat (keadaan normal berwarna merah muda, jika pucat akan berwarna keputihan bahkan bisa berwarna kuning), jika diraba perutnya terasa sakit, perut kembung, hewan terlihat lesu (hanya diam saja, nafsu makan berkurang), bulu kasar.

## 2. *Mastitis* (Radang Susu)

*Mastitis* merupakan suatu peradangan pada kelenjar susu atau ambing dapat terjadi secara perakut, akut, subakut atau kronis.

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu dehidrasi (apabila ditarik kulitnya akan kembali sangat lama, gusi ditekan terlihat putih), nafsu makan turun, kelinci kesakitan jika disentuh, ambing dan puting berwarna merah, suhu badan naik, nafsu makan turun, puting mengeras, ujung puting berwarna hitam, timbul benjolan pada ambing, kulit mengering dan pecah-pecah serta suhu badan kelinci naik.

## 3. *Coccidiosis* (Parasit Pencernaan)

*Coccidiosis* (Parasit Pencernaan) merupakan infeksi parasit *Eimeria/Isospora* pada saluran pencernaan ternak baik unggas, sapi, kambing, domba, babi serta kelinci yang terjadi secara akut yang diikuti dengan rusaknya mukosa saluran pencernaan.

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu gigi berkerot menahan sakit, nafsu makan turun, hewan terlihat lesu, berat badan berkurang, perut tampak besar, mencret bercampur darah, kotoran berlendir putih, badan kurus, dehidrasi.

#### 4. *Pneumonia (Pasteurellois / Radang Paru-paru)*

Penyebab dari penyakit ini adalah bakteri *pasteurella*, penyakit ini dapat menular.

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu dehidrasi (apabila ditarik kulitnya akan kembali sangat lama, gusi ditekan terlihat putih), nafsu makan turun, mata dan telinga kebiruan, kadang kotoran bernanah, mencret, kepala sering diangkat tinggi-tinggi, hidung keluar nanah, suhu badan dingin / rendah, sesak nafas atau susah bernafas, hidung mengeluarkan lendir berwarna jernih atau keruh, kaki depan selalu berusaha menggaruk hidung, kaki dan bulu ikut basah, bersin-bersin, mata sembab, basah, dan berair, seperti kelinci terlihat stress, agresif, dan meloncat-loncat. Selain itu, kelinci mengalami sesak nafas atau susah bernafas, leher berputar kesamping.

#### 5. *Scabies (Kudis)*

*Scabies* adalah sejenis kutu yang ukurannya mikroskopik. Mereka bersarang di bawah lapisan kulit sehingga menyebabkan gatal. Kelinci yang terjangkit akan sering menggaruk bagian yang terjangkit (pada umumnya kuping, kaki, hidung kelinci) sampai bagian tersebut luka bahkan robek.

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu nafsu makan turun, berat badan berkurang, kulit kemerah-merahan (pada umumnya merah muda atau gelap tergantung jenis kelinci, tidak ada yg berwarna merah), badan penuh keropeng, merasa gatal, badan kurus, bulu rontok, tidak ada yang berwarna merah. Kulit kelinci menjadi gatal dan bersisik.

#### 6. *Ringworm (Jamur)*

*Ringworm* bukan penyakit cacangan, tetapi penyakit akibat infeksi jamur, termasuk 'zoonosis' (Penyakit yang dapat menular ke manusia).

Gejala-gejala yang biasanya ditemui pada penyakit ini yaitu merasa gatal, iritasi di bagian kepala atau kaki, lalu menular ke bagian dalam yang lain, bagian yang terserang menimbulkan bercak-bercak merah, bulu menjadi menggumpal dan

kusut, kulit kepala menimbulkan sisik berbentuk bulat pipih, merah dan keras, kulit kepala tampak pecah-pecah, bulu rontok.

#### 7. Radang Mata

Radang Mata merupakan gejala penyakit ketika kedua mata kelinci mengeluarkan cairan, mata kelinci menjadi radang berwarna merah terutama pada kelopak atau selaput mata, bulu disekitar menjadi kusam dan basah, kelinci mengeluarkan air mata secara terus menerus, kelinci selalu mengusap mata dengan kakinya. Infeksi atau gejala penyakit tersebut kemungkinan karena tercakar atau ada kotoran atau benda asing pada mata kelinci, dalam waktu lebih dari sepuluh hari mata anak kelinci tidak dapat terbuka, mengalami peradangan pada mata, sembab, basah dan berair.

#### 8. *Hairball*

*Hairball* terjadi karena masuknya rambut atau bulu kelinci ke dalam saluran pencernaan, kemudian terakumulasi dalam jangka waktu yang lama hingga mengeras sehingga menimbulkan gejala-gejala penyakit kelinci seperti nafsu makan kelinci yang mulai menurun yang dapat menyebabkan kematian pada 3-4 pekan kemudian, bulu kelinci menjadi kasar, adanya gangguan pada pernafasan (nafas terdengar berbunyi), serta gangguan pencernaan akibat penumpukan bulu di saluran pencernaan sehingga badan kelinci menjadi kurus.

#### 9. Cacingan

Cacingan atau *nematodes* yaitu cacing perut, kremi, dan cacing pita. Kelinci terinfeksi pada saat memakan telur cacing. Gejalanya adalah nafsu makan kelinci menjadi turun, berat badan berkurang, kelinci terlihat lemah, pucat, dan kurus.

#### 10. *Metritis* (Infeksi Kandungan)

*Metritis* merupakan penyakit kandungan yang memiliki gejala-gejala seperti

lendir di vagina kelinci, jika kandungan diraba akan terasa membesar, produktivitas kelinci menurun dan anak kelinci mati di dalam kandungan.

#### 11. *Pinworm (Oxyuris)*

Penyakit *pinworm* atau *oxyuris* ini disebabkan oleh cacing kecil berwarna putih yang hidup di usus belakang. Kelinci sehat dapat tertular karena makan dan minum yang mengandung cacing.

Gelaja-gelajanya adalah kelinci merasa gatal di sekitar lubang lubur, kelinci sering menggaruk dan menjilati bulu di sekitar lubang duburnya, badan kelinci menjadi kurus dan kelinci akan mengalami sesak nafas atau susah bernafas.

#### 12. Bloat (*Tympany*)

Penyakit ini sering menyerang anakan dibawah 2 bulan dan indukan yang hamil atau menyusui, apabila tidak ada penanganan yang cepat akan berakibat pada kematian. Gelaja-gelajanya adalah perut membesar, susah bernafas dan badan ambruk.

### 2.2.5 Android

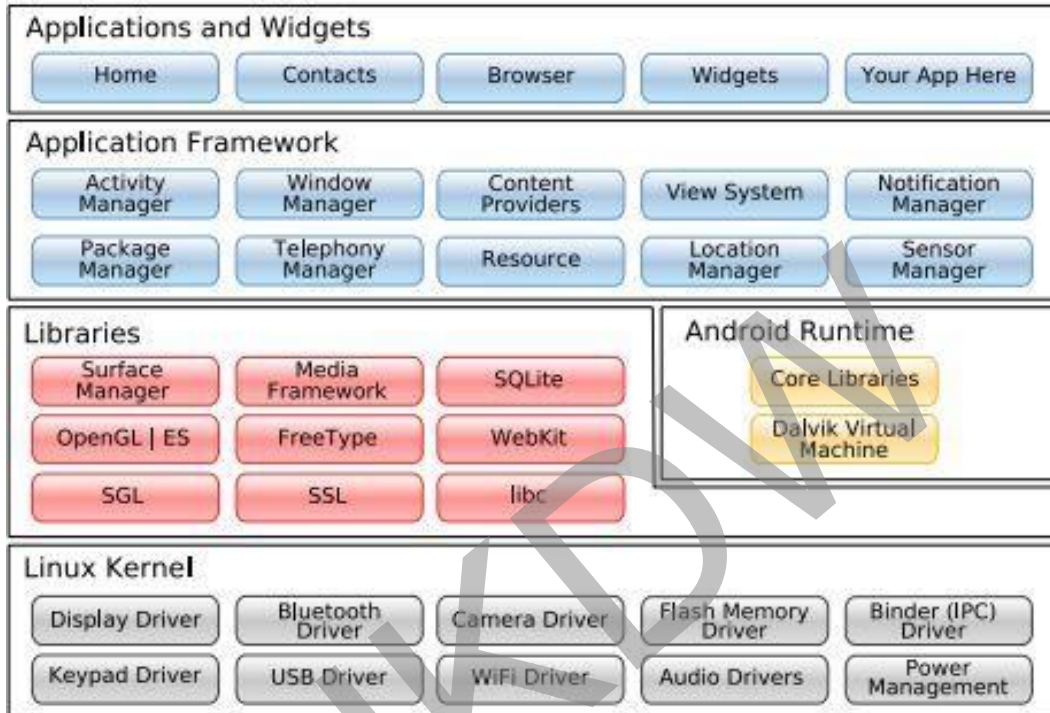
Android merupakan sistem operasi berbasis Linux yang pada dasarnya ditujukan untuk berjalan dengan *hardware smartphone* maupun tablet. Dalam pengembangannya, android merupakan sistem operasi yang terbuka. Ini artinya para pengembang dapat memodifikasi sistem operasi android sesuai keinginan.

Menurut John Wiley dan Sons (2010), android merupakan sistem *opensource*, lengkap dan luas untuk sistem operasi perangkat *mobile*. Yang mencakup sistem informasi, *user interface* dan aplikasi-aplikasi untuk perangkat *mobile* android itu sendiri. Gambar arsitektur sistem operasi android dapat dilihat pada gambar 2.1.

Dalam arsitektur android terdiri dari lima ruang, yaitu :

#### 1. *Layer Application*

Lapisan teratas pada sistem operasi android, pada layer ini android bertemu langsung dengan pengguna android. Aplikasi yang ada pada lapisan ini dapat berupa kontak, sms, browser, main menu.



Gambar 2.1 Arsitektur Android

## 2. *Layer Application Framework*

Tempat para pengembang membuat suatu aplikasi yang akan dipakai untuk melakukan pengembangan sistem operasi android.

## 3. *Layer Libraries*

Lapisan yang berisi kamus-kamus untuk aplikasi yang akan dijalankan atau dikembangkan pada sistem operasi android. Contoh kamus yang ada pada lapisan ini adalah SQLite, SSL, OpenGL, WebKit.

## 4. *Layer Android Runtime*

Bagian yang terdiri dari *Core Libraries* dan *Dalvik Virtual Machine*. Fungsi dari *Core libraries* merupakan bagian yang dipakai untuk memahami *Java Core Library*.

## 5. *Layer Linux Kernel*



Merupakan bagian yang berisi driver-driver yang bekerja untuk sistem operasi android, seperti driver untuk WiFi, kamera, speaker, keypad, GPS.

©UKDWN

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Deskripsi Sistem

Sistem yang akan dibuat ini merupakan sebuah sistem perangkat lunak yang digunakan untuk membandingkan hasil diagnosis penyakit kelinci antara penggunaan metode *Certainty Factor* dengan metode *Dempster-Shafer*. Sistem ini merupakan pengembangan dari sistem yang telah dibuat oleh Atmaja (2014) dalam tugas akhirnya yang berjudul “ Sistem Pakar Diagnosis Penyakit Kelinci dengan Menggunakan Metode *Certainty Factor* ”. Sistem ini menggunakan basis pengetahuan, *user interface*, serta *rule* yang telah dibuat oleh Atmaja (2014). Pada sistem ini penulis melakukan pengembangan dengan menambah perhitungan mengenai *Certainty Factor* dimana bobot gejala di ambil dari basis data yang telah tersimpan di sistem. Selain itu penulis juga menambahkan perhitungan menggunakan metode *Dempster-Shafer* dalam 2 cara yaitu, yang pertama *user* memasukkan nilai dari bobot tiap gejala yang telah dipilih dan cara yang kedua yaitu nilai bobot dari gejala yang telah dipilih *user* diambil dari basis data yang telah tersimpan di dalam sistem.

Dalam sistem ini, nantinya pengguna dapat berinteraksi dengan sistem melalui pernyataan-pernyataan berupa fakta yang telah tersimpan didalam sistem. Hasil akhir dari sistem nantinya akan menampilkan penyakit yang mungkin terjadi berdasarkan masukan yang telah dilakukan oleh pengguna sebelumnya.

Sesuai dengan struktur sistem pakar secara umum, struktur yang digunakan dalam mendiagnosis penyakit pada kelinci terdiri dari dua lingkup kerja, yaitu lingkup pengembangan dan lingkup konsultasi.

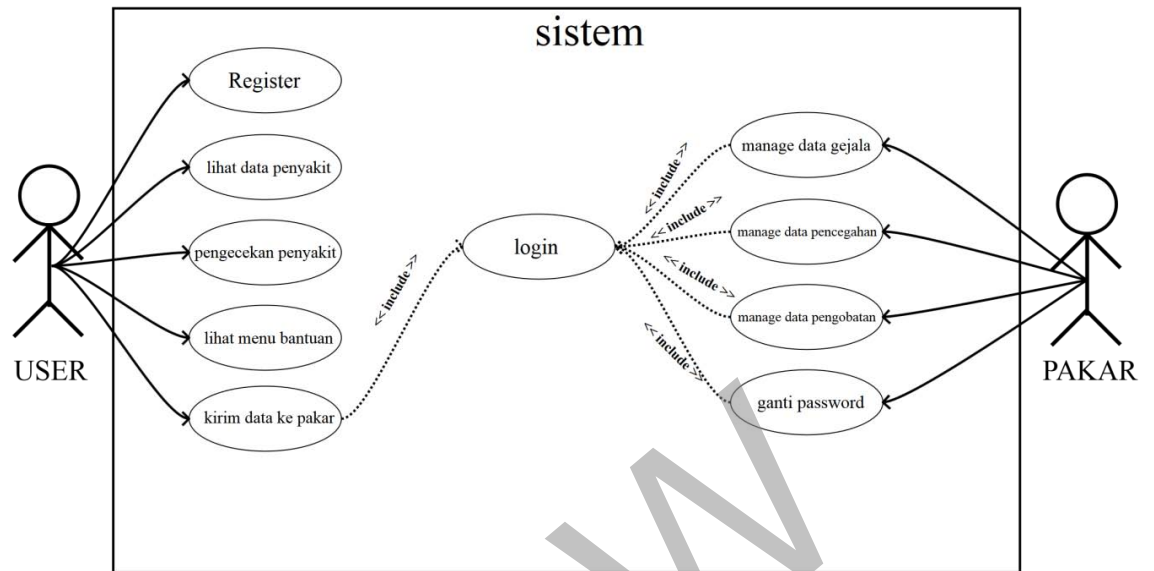
Lingkup pengembangan digunakan untuk memasukkan pengetahuan dari pakar serta melakukan perbaikan pengetahuan. Sedangkan lingkup konsultasi merupakan lingkup yang digunakan pengguna untuk memperoleh informasi.

Sistem pakar ini akan dibangun berbasis android dengan menggunakan bahasa pemrograman *Java* serta menggunakan *web service* untuk pengolahan data. Berikut ini merupakan kebutuhan fungsional dan non-fungsional seperti yang telah dijelaskan oleh Atmaja (2014):

a. Kebutuhan Fungsional

- Sistem dapat menampilkan penyakit-penyakit kelinci
- Sistem dapat melakukan pengecekan berdasarkan gejala yang dipilih oleh pengguna dan menampilkan hasilnya
- Hasil dari pengecekan berupa nama penyakit beserta saran pencegahan dan pengobatan
- Pengguna dapat mengirimkan data baru ke pakar jika penyakit tidak ditemukan
- Use Case

Sistem yang akan dibangun akan memiliki 2 rancangan, yaitu sistem aplikasi android sebagai aplikasi antar muka yang digunakan oleh pengguna dan sistem web service yang digunakan untuk melakukan proses perhitungan gejala – gejala pada penyakit kelinci.



Gambar 3.1 Diagram *Use Case*

- User :

- 1) Register ( Registrasi ) adalah proses *user* membuat akun untuk didaftarkan ke dalam sistem.
- 2) Melihat data penyakit merupakan menu yang menampilkan gejala-gejala penyakit yang akan dipilih, beserta pencegahan dan pengobatan.
- 3) Pengecekan penyakit merupakan menu untuk melakukan pengecekan penyakit dengan gejala yang telah tersedia.
- 4) Menu bantuan merupakan menu penjelasan cara menggunakan aplikasi.
- 5) Kirim data ke pakar merupakan menu dimana *user* yang sudah terdaftar dapat mengirimkan usulan data ke pakar.

- Admin :

- 1) Mengatur data gejala merupakan menu yang digunakan *admin* untuk memasukkan data-data gejala penyakit kelinci ke dalam basis pengetahuan agar dapat digunakan *user*.

- 2) Mengatur data pencegahan yaitu menu yang digunakan *admin* untuk memasukkan data-data pencegahan penyakit kelinci ke dalam basis pengetahuan agar dapat digunakan *user*.
- 3) Mengatur data pengobatan merupakan menu yang digunakan *admin* untuk memasukkan data-data pengobatan ke dalam basis pengetahuan agar dapat digunakan *user*.
- 4) Mengganti password.
- 5) Logout.
- 6) Review data test yaitu menu yang digunakan *admin* untuk melihat data usulan *user*.
- 7) Login

b. Spesifikasi Hardware dan Software

Berikut ini adalah spesifikasi hardware dan software dari sisi server :

- Processor : Intel core I3
- Memory : 4GB
- Kartu grafis : Nvidia GeForce 820m
- Media penyimpanan : 500GB
- Layar : 14" HD LED
- Web Server : Apache versi 2.2 dalam XAMPP versi 1.8.1
- Bahasa pemrograman : PHP versi 5.4.7
- Database : MySQL
- Browser : Google Chrome versi 54.0.2840.87
- OS : Windows 10 Pro 64-bit

Berikut ini adalah spesifikasi hardware dan software dari sisi android :

- Model : Asus\_Z00AD ( Asus Zenfone 2 )
- Processor : Intel Atom Z3580 2.3 GHz quad-core processor
- OS : Android 6.0.1 ( Marshmallow )

- Layar : 5.5" Full HD ( 1080 x 1920 )
- Memory : 32GB User memory + 4GB RAM
- Emulator : Android Studio versi 2.2.2

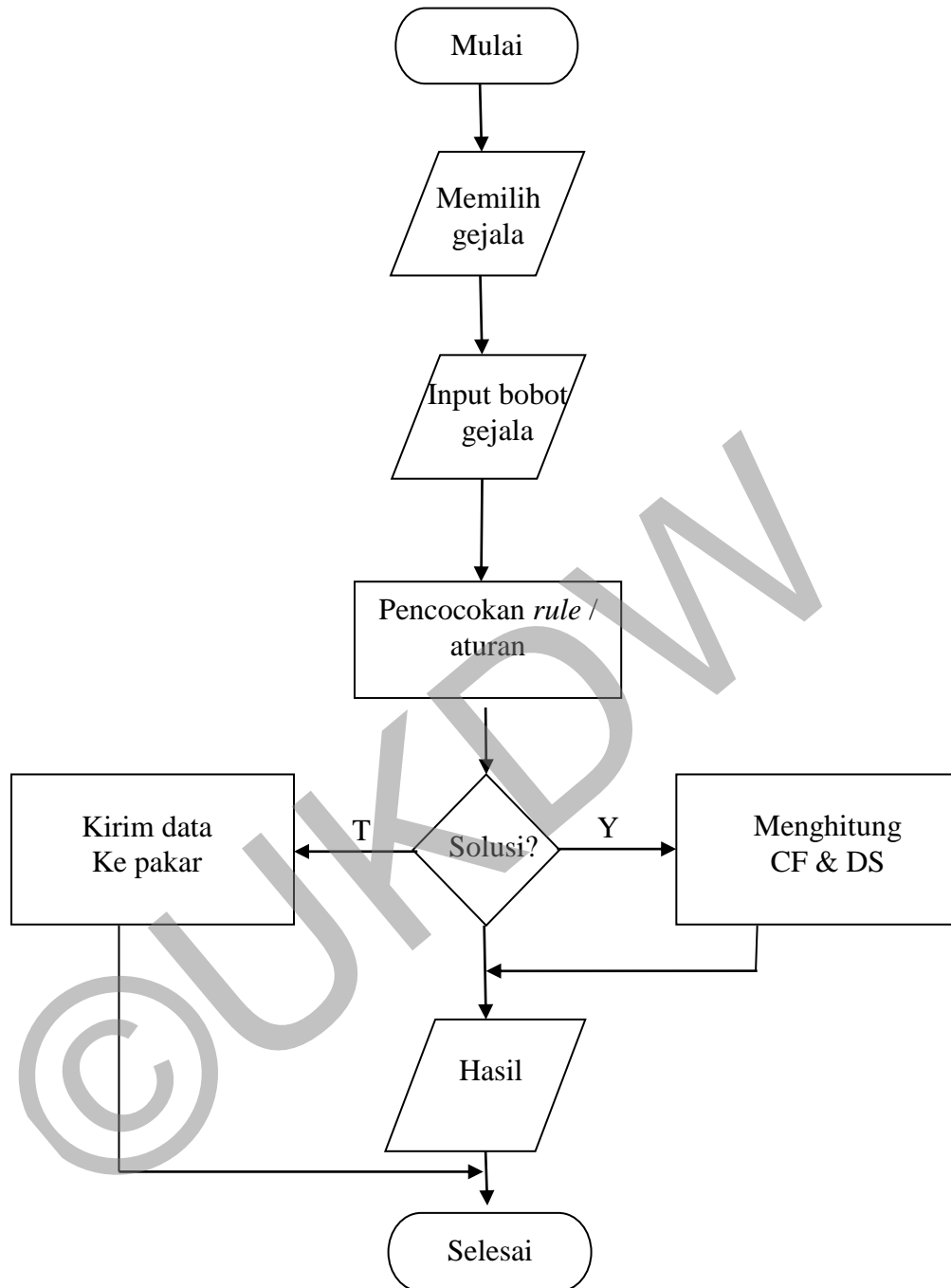
## 3.2 Algoritma dan Flowchart

### 3.2.1 Flowchart Sistem

Sistem akan memiliki 2 flowchart, yaitu flowchart A dimana *user* memasukkan nilai bobot dari tiap gejala yang telah dipilih, dan flowchart B dimana nilai bobot dari tiap gejala yang telah dipilih diambil dari basis data yang telah tersimpan.

Pada flowchart A, proses akan terbagi menjadi beberapa langkah yaitu :

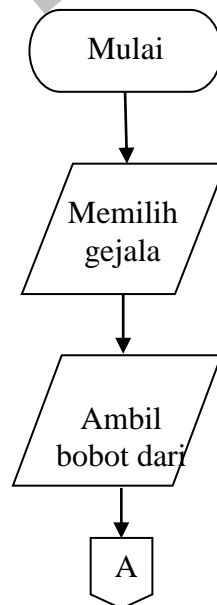
- Sistem aplikasi akan menampilkan gejala-gejala utama/umum dimana *user* dapat memilih 1 gejala.
- Selanjutnya *user* diberi pilihan berupa fakta-fakta mengenai gejala pendukung, disini *user* dapat memilih lebih dari 1 gejala.
- Setelah proses pemilihan gejala utama dan pendukung, *user* diminta untuk memasukkan besar bobot antara 0.1 sampai 0.9 dari tiap-tiap gejala yang telah dipilih.
- Setelah itu, sistem akan melakukan pencocokan *rule* dengan gejala-gejala yang dipilih *user*.
- Apabila terdapat kecocokan antara gejala yang dipilih *user* dengan *rule* yang terdapat di sistem, maka dilanjutkan ke penghitungan nilai CF ( *Certainty Factor* ) dan DS ( *Dempster-Shafer* ). Sedangkan jika tidak ditemukan kecocokan, maka *user* akan diarahkan untuk mengirimkan data tersebut ke pakar.
- Hasil dari penghitungan sistem berupa persentase dari penyakit yang ditemukan beserta cara pengobatannya.



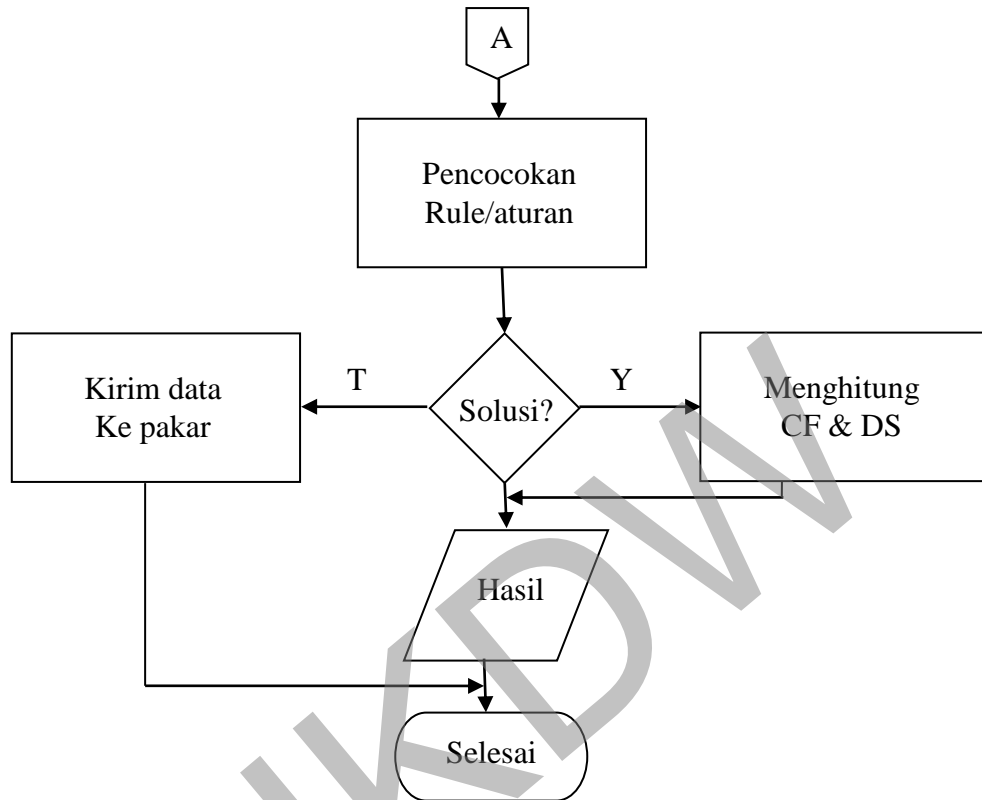
Gambar 3.2 Flowchart A

Pada flowchart B, proses juga terbagi menjadi beberapa langkah yaitu :

- Sistem aplikasi akan menampilkan gejala-gejala utama/umum dimana *user* dapat memilih 1 gejala.
- Selanjutnya *user* diberi pilihan berupa fakta-fakta mengenai gejala pendukung, disini *user* dapat memilih lebih dari 1 gejala.
- Setelah proses pemilihan gejala utama dan pendukung, sistem akan memasukkan besar nilai bobot yang ada di dalam basis data sesuai dengan gejala yang telah dipilih.
- Setelah itu, sistem akan melakukan pencocokan *rule* dengan gejala-gejala yang dipilih *user*.
- Apabila terdapat kecocokan antara gejala yang dipilih *user* dengan *rule* yang terdapat di sistem, maka dilanjutkan ke penghitungan nilai CF ( *Certainty Factor* ) dan DS ( *Dempster-Shafer* ). Sedangkan jika tidak ditemukan kecocokan, maka *user* akan diarahkan untuk mengirimkan data tersebut ke pakar.
- Hasil dari penghitungan sistem berupa persentase dari penyakit yang ditemukan beserta cara pengobatannya







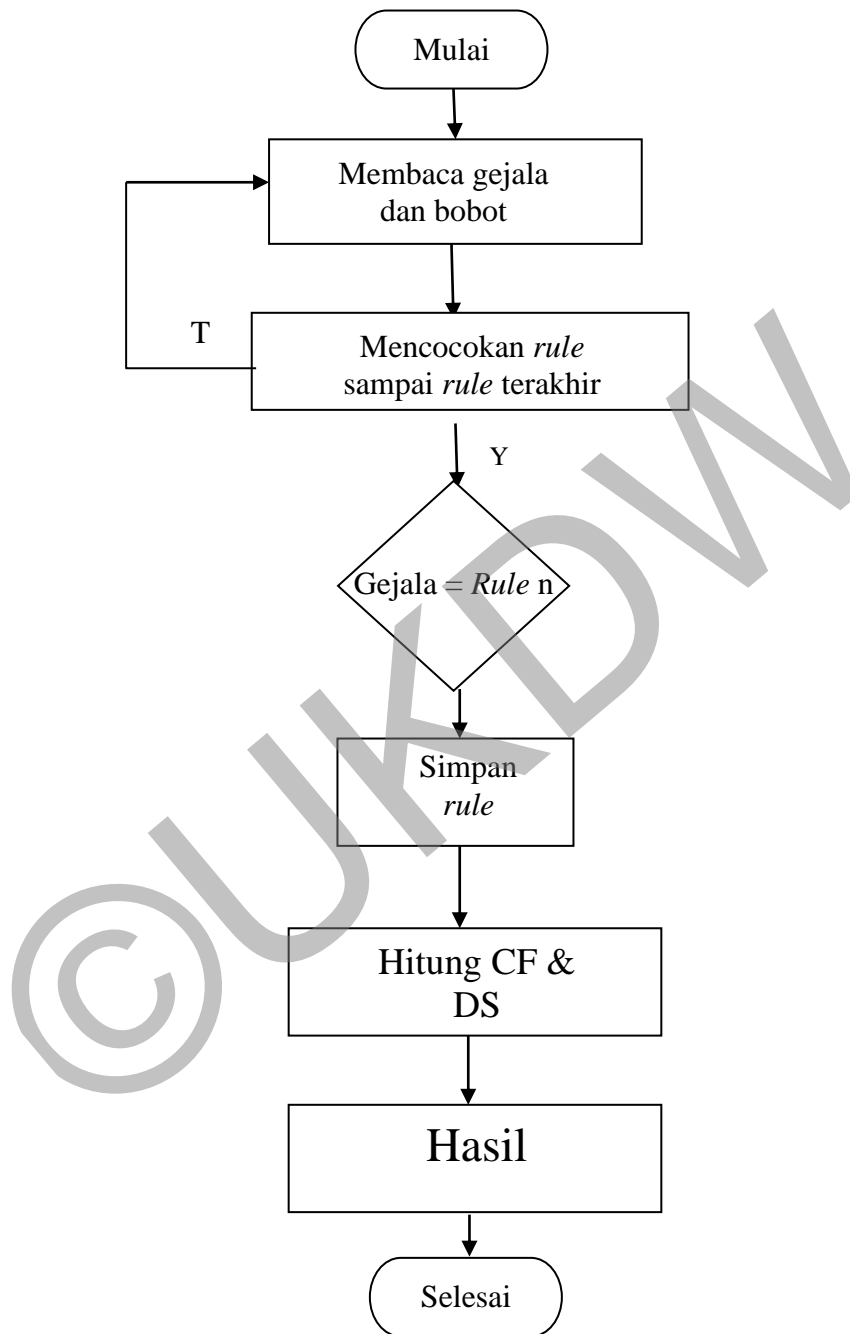
Gambar 3.3 Flowchart B

### 3.2.2 Flowchart *Forward Chaining*

Pada flowchart *forward chaining*, proses terbagi menjadi beberapa langkah :

- Sistem membaca gejala yang telah dipilih oleh *user* beserta dengan bobot dari setiap gejala.
- Sistem akan melakukan pencocokan secara progresif yaitu mencocokkan *rule* yang telah dibuat di dalam sistem sampai pada iterasi dimana *rule* tersebut berakhir.
- Jika gejala dan bobot yang diinputkan *user* berada pada 1 *rule* atau lebih, sistem akan berhenti melakukan pengecekan dan mulai menghitung nilai CF dan DS.

- Hasil akhir yang muncul berupa penyakit beserta persentase dari masing-masing metode.



Gambar 3.4 Flowchart *Forward Chaining*

### 3.2.3 Aturan ( Rule ) Penyakit Kelinci

Berdasarkan data yang diperoleh Atmaja ( 2014 ) mengenai gejala, penyakit dan *rule* pada kelinci, maka dapat dijelaskan lebih lanjut sebagai berikut :

No	ID Gejala	Nama Gejala
1	G1	Suhu Badan Naik
2	G2	Badan Kurus
3	G3	Sesak Nafas atau Susah Bernafas
4	G4	Mata Sembab, Basah dan Berair
5	G5	Bulu Rontok
6	G6	Bulu Menggumpal dan Kusut
7	G7	Lendir di Vagina Berupa Darah/Nanah
8	G8	Daun Telinga Turun
9	G9	Gigi Berkerot Menahan Sakit
10	G10	Daerah Anus Kotor
11	G11	Kotoran Berwarna Hijau Gelap dan Bau
12	G12	Lendir Menggantung Pada Dubur
13	G13	Pertumbuhan Melambat
14	G14	Perut Diraba Terasa Sakit
15	G15	Perut Kembung
16	G16	Nafsu Makan Turun
17	G17	Hewan Kesakitan jika Disentuh Bagian Ambing
18	G18	Ambing dan Putting Berwarna Merah, Mengeras
19	G19	Hewan Terlihat Lesu
20	G20	Berat Badan Berkurang
21	G21	Perut Tampak Besar
22	G22	Mencret Bercampur Darah
23	G23	Kotoran Berlendir Putih
24	G24	Mencret/Diare

25	G25	Kepala Sering Diangkat Tinggi-Tinggi
26	G26	Hidung Keluar Nanah, Lendir Berwarna Putih/Keruh
27	G27	Kulit Kemerah-merahan
28	G28	Badan Penuh Keropeng
29	G29	Merasa Gatal
30	G30	Bersin-Bersin
31	G31	Iritasi di Bagian Kepala/Kaki, Menular ke Bagian Dalam yang Lain
32	G32	Bagian yang Terserang Menimbulkan Bercak Merah
33	G33	Muntah Bulu
34	G34	Kotoran Ada Cacing
35	G35	Radang Berwarna Merah, Terutama pada Kelopak/Selaput Mata
36	G36	Bulu di Sekitar Mata Basah dan Kusam
37	G37	Keluar Air Mata Terus Menerus
38	G38	Hewan Selalu Mengusap Mata dengan Kakinya
39	G39	Kotoran Lembek Cair
40	G40	Gangguan Pencernaan Akibat Penumpukan Bulu di Saluran Pencernaan
41	G41	Hewan Terlihat Lemah
42	G42	Hewan Terlihat Pucat
43	G43	Ujung Puting Berwarna Hitam
44	G44	Susu yang Diperah Rusak/Pecah
45	G45	Mengalami Peradangan pada Mata
46	G46	Kulit Bersisik
47	G47	Hewan Terlihat Stress
48	G48	Hewan Terlihat Agresif
49	G49	Hewan Meloncat Loncat

<b>50</b>	G50	Minum Banyak, Makan Sedikit
<b>51</b>	G51	Jika Kandungan Diraba, Terasa Membesar
<b>52</b>	G52	Gatal disekitar Lubang Dubur
<b>53</b>	G53	Hewan Sering Menggaruk dan Menjilati Bulu disekitar Lubang Dubur
<b>54</b>	G54	Kotoran Mata Lebih Banyak dari Biasanya dan Berwarna Kuning
<b>55</b>	G55	Badan Ambruk Tidak Bisa Berdiri
<b>56</b>	G56	Leher Berputar ke Samping

Tabel 3.1 Tabel Nama Gejala

Berikut ini merupakan nama penyakit yang biasanya dialami oleh kelinci :

<b>No</b>	<b>ID Penyakit</b>	<b>Nama Penyakit</b>
<b>1</b>	P1	Enteritis Kompleks
<b>2</b>	P2	Radang Susu
<b>3</b>	P3	Parasit Pencernaan
<b>4</b>	P4	Radang Paru-Paru
<b>5</b>	P5	Kudis
<b>6</b>	P6	Jamur
<b>7</b>	P7	Radang Mata
<b>8</b>	P8	<i>Hairball</i>
<b>9</b>	P9	Cacingan
<b>10</b>	P10	Infeksi Kandungan
<b>11</b>	P11	<i>Pinworm</i>
<b>12</b>	P12	<i>Bloat</i>

Tabel 3.2 Tabel Nama Penyakit

Berdasarkan table gejala dan table penyakit di atas, kemudian disusunlah *rule* seperti yang tertera pada table di bawah ini yang telah disusun oleh Atmaja (2014) :

<b>NO</b>	<b>Rule</b>	<b>CF</b>
1	IF G21 AND G3 AND G55 THEN P12	0.8
2	IF G3 AND G21 THEN P12	0.7
3	IF G3 AND G55 THEN P12	0.7
4	IF G2 AND G16 AND G34 AND G41 AND G42 THEN P9	0.8
5	IF G2 AND G16 AND G41 THEN P9	0.5
6	IF G2 AND G41 AND G42 THEN P9	0.5
7	IF G2 AND G42 AND G34 THEN P9	0.8
8	IF G2 AND G11 AND G12 AND G19 AND G39 THEN P1	0.8
9	IF G2 AND G11 AND G13 AND G14 THEN P1	0.4
10	IF G2 AND G8 AND G9 AND G10 AND G11 AND G12 AND G13 AND G14 AND G15 AND G19 AND G39 THEN P1	0.9
11	IF G2 AND G10 AND G11 AND G19 AND G39 THEN P1	0.9
12	IF G2 AND G8 AND G11 AND G12 AND G14 AND G15 AND G19 THEN P1	0.8
13	IF G2 AND G16 AND G33 AND G40 THEN P8	0.8
14	IF G2 AND G3 AND G16 AND G33 AND G40 AND THEN P8	0.9
15	IF G2 AND G16 AND G3 THEN P8	0.3
16	IF G2 AND G3 AND G16 AND G40 THEN P8	0.8
17	IF G7 AND G50 AND G51 THEN P10	0.8
18	IF G7 AND G50 THEN P10	0.7
19	IF G7 AND G51 THEN P10	0.7
20	IF G5 AND G6 AND G29 AND G31 AND G32 THEN P6	0.9
21	IF G5 AND G29 AND G32 THEN P6	0.9
22	IF G5 AND G29 AND G31 THEN P6	0.7
23	IF G5 AND G6 AND G29 THEN P6	0.5

24	IF G2 AND G5 AND G20 AND G28 AND G29 THEN P5	0.8
25	IF G2 AND G5 AND G27 AND G28 AND G29 THEN P5	0.9
26	IF G2 AND G20 AND G5 AND G27 THEN P5	0.6
27	IF G2 AND G5 AND G16 THEN P5	0.4
28	IF G2 AND G5 AND G16 AND G20 AND G27 AND G28 AND G29 THEN P5	0.9
29	IF G2 AND G9 AND G19 AND G20 THEN P3	0.4
30	IF G2 AND G10 AND G20 AND G22 AND G23 THEN P3	0.8
31	IF G2 AND G19 AND G20 AND G22 AND G23 THEN P3	0.8
32	IF G2 AND G16 AND G20 AND G22 THEN P3	0.7
33	IF G2 AND G9 AND G10 AND G16 AND G19 AND G22 AND G23 THEN P3	0.9
34	IF G2 AND G52 AND G53 THEN P11	0.7
35	IF G2 AND G52 AND G54 THEN P11	0.7
36	IF G2 AND G52 AND G53 AND G54 THEN P11	0.8
37	IF G2 AND G53 AND G54 THEN P11	0.7
38	IF G4 AND G35 AND G37 AND G38 THEN P7	0.8
39	IF G4 AND G35 AND G36 AND G37 AND G38 AND G45 THEN P7	0.9
40	IF G4 AND G35 AND G36 THEN P7	0.9
41	IF G4 AND G35 AND G36 AND G37 THEN P7	0.8
42	IF G3 AND G4 AND G25 AND G26 AND G31 THEN P4	0.8
43	IF G3 AND G4 AND G26 THEN P4	0.4
44	IF G3 AND G4 AND G25 AND G26 AND G30 THEN P4	0.8
45	IF G3 AND G4 AND G25 AND G26 AND G30 AND G31 AND G56 THEN P4	0.9
46	IF G1 AND G17 AND G18 AND G44 THEN P2	0.8
47	IF G1 AND G10 AND G16 AND G17 AND G18 AND G43	0.9

	AND G44 THEN P2	
48	IF G1 AND G10 AND G16 AND G18 THEN P2	0.5
49	IF G1 AND G10 AND G16 AND G18 AND G43 THEN P2	0.8
50	IF G1 AND G16 AND G18 AND G43 THEN P2	0.7

Tabel 3.3 Tabel Rule Penyakit Kelinci

### 3.3 Kamus Data

#### 3.3.1 Tabel Relasi Diagnosis Penyakit Kelinci

Tabel Penyakit merupakan table yang digunakan untuk menyimpan data penyakit kelici.

Nama field	Tipe Data	Keterangan
id_penyakit	Int	Primary key
penyakit_kode	VARCHAR(45)	Kode penyakit
penyakit_nama	VARCHAR(225)	Nama penyakit
penyakit_namalatin	VARCHAR(150)	Nama ilmiah dari suatu penyakit
penyakit_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)
Penyakit_cf	VARCHAR(45)	Nilai CF pakar

Tabel 3.4 Tabel Penyakit

#### 3.3.2 Tabel Penyakit Gejala

Tabel penyakit gejala merupakan table yang digunakan untuk menyimpan data gejala dari table penyakit yang disertai dengan nilai MB dan MD.



<b>Nama field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_penyakit_gejala	INT	Primary key
Id_penyakit	INT	id penyakit mengacu pada field id_penyakit di tabel penyakit
id_gejala	INT	id gejala mengacu pada field id_gejala di tabel gejala
penyakit_gejala_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.5 Tabel Penyakit Gejala

### 3.3.3 Tabel Gejala

Tabel gejala merupakan tabel yang digunakan untuk menyimpan data gejala-gejala penyakit kelinci.

<b>Nama field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_gejala	INT	Primary key
gejala_kode	VARCHAR(45)	Kode gejala
gejala_nama	VARCHAR(45)	Nama gejala
gejala_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.6 Tabel Gejala

### 3.3.4 Tabel Tes Entry

Table tes entry merupakan table yang berisi gejala-gejala yang dipilih oleh *user* pada saat menggunakan aplikasi.

Nama field	Tipe Data	Keterangan
id_tesentry	INT	Primary key
id_tes	INT	id tes mengacu pada field id_tes di tabel Tes
id_gejala	INT	id gejala mengacu pada field id_gejala di tabel gejala
tesentry_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.7 Tabel Tes Entry

### 3.3.5 Tabel Tes

Tabel tes merupakan table yang digunakan untuk menyimpan data tes yang telah dilakukan oleh *user*.

Nama field	Tipe Data	Keterangan
id_tes	INT	Primary key
tes_tanggal	DATETIME	Tanggal tes
id_pelakutes	INT	id pelakutes mengacu pada field id_pelakutes di tabel pelakutes
tes_status	SMALLINT	Menandakan <i>record</i>

		aktif (1) dan tidak aktif/dihapus (0)
--	--	---------------------------------------

Tabel 3.8 Tabel Tes

### 3.3.6 Tabel Pelaku Tes

Tabel pelaku tes merupakan table yang digunakan untuk menyimpan data-data *user* yang terintegrasi di sistem.

Nama field	Tipe Data	Keterangan
id_pelakutes	INT	Primary key
pelakutes_nomoridentitas	VARCHAR(45)	ID <i>user</i>
pelakutes_nama	VARCHAR(45)	Nama <i>user</i>
pelakutes_alamat	VARCHAR(45)	Alamat <i>user</i>
pelakutes_phone	VARCHAR(45)	Nomor telepon <i>user</i>
pelakutes_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.9 Tabel Pelaku Tes

### 3.3.7 Tabel Penyakit Pencegahan

Tabel penyakit pencegahan merupakan table yang digunakan untuk menyimpan data pencegahan penyakit dari table pencegahan.

Nama field	Tipe Data	Keterangan
id_penyakit_pencegahan	INT	Primary key
id_penyakit	INT	id penyakit mengacu

		pada field id_penyakit di tabel penyakit
id_pencegahan	INT	id pencegahan mengacu pada field id_pencegahan di tabel pencegahan
penyakit_pencegahan_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.10 Tabel Penyakit Pencegahan

### 3.3.8 Tabel Pencegahan

Tabel pencegahan merupakan table yang digunakan untuk menyimpan data pencegahan dari penyakit kelinci.

<b>Nama field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_pencegahan	INT	Primary key
pencegahan_kode	VARCHAR(45)	Kode pencegahan
Pencegahan_nama	VARCHAR(225)	Nama pencegahan
pencegahan_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.11 Tabel Pencegahan

### 3.3.9 Tabel Penyakit Pengobatan

Table penyakit pengobatan merupakan table yang digunakan untuk menyimpan data pengobatan kelinci dari table pengobatan.

Nama field	Tipe Data	Keterangan
id_penyakit_pengobatan	INT	Primary key
id_penyakit	INT	id penyakit mengacu pada field id_penyakit di tabel penyakit
id_pengobatan	INT	id pengobatan mengacu pada field id_pengobatan di tabel pengobatan
penyakit_pengobatan_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.12 Tabel Penyakit Pengobatan

### 3.3.10 Tabel Pengobatan

Table pengobatan merupakan table yang digunakan untuk menyimpan data pengobatan dari penyakit kelinci.

Nama field	Tipe Data	Keterangan
id_pengobatan	INT	Primary key
pengobatan_kode	VARCHAR(45)	Kode pengobatan
pengobatan_nama	VARCHAR(225)	Nama pengobatan
pengobatan_status	SMALLINT	Menandakan <i>record</i> aktif (1) dan tidak aktif/dihapus (0)

Tabel 3.13 Tabel Pengobatan

### 3.3.11 Tabel Admin

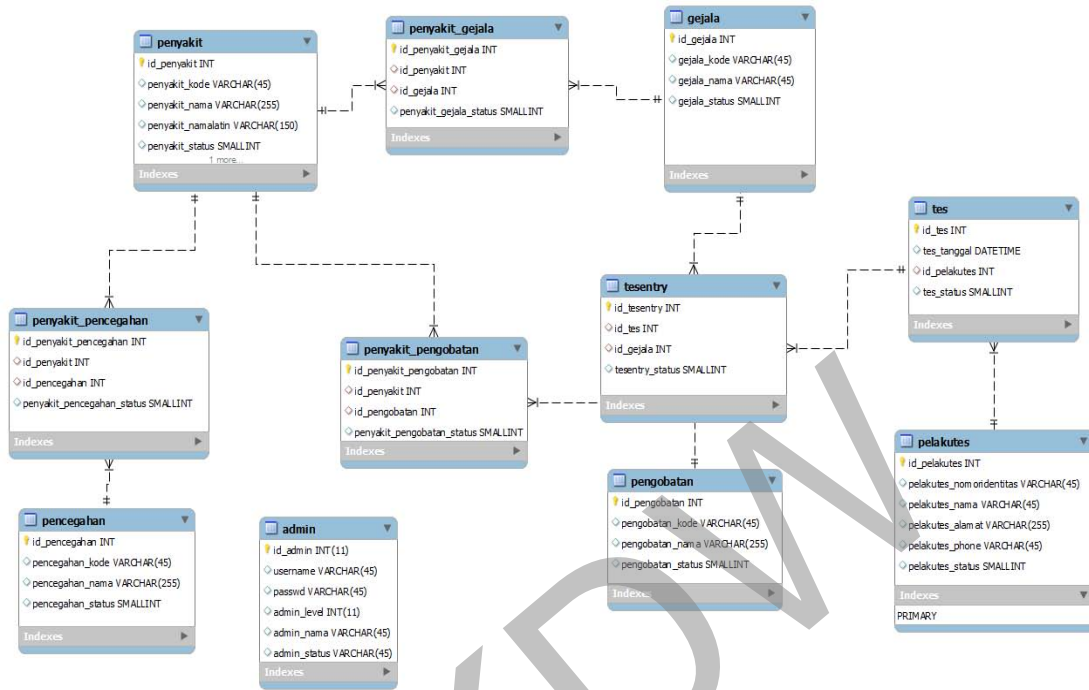
Tabel admin merupakan table yang digunakan untuk menyimpan data pakar berupa *username* dan *password* yang digunakan pakar untuk masuk ke sistem.

<b>Nama field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_admin	INT(11)	Primary key
username	VARCHAR(45)	Nama admin
passwd	VARCHAR(45)	Password admin

Tabel 3.14 Tabel Admin

©UKDW

### 3.4 Diagram Skema



Gambar 3.5 Diagram Skema

### 3.5 Rancangan Antarmuka Sistem

#### 3.5.1 Rancangan Antarmuka Web Service

The screenshot shows a simple login interface with the following elements:

- A text input field labeled "username".
- A text input field labeled "password".
- A button labeled "login".

Gambar 3.6 Halaman Login Pakar

Gambar 3.6 di atas merupakan gambar rancangan halaman *login* pakar. Halaman ini berfungsi sebagai pengaman sistem sehingga tidak semua orang dapat menggunakan sistem. Hak akses hanya diberikan kepada pakar.

Administrator    Gejala    Penyakit    Pencegahan    Pengobatan    Pelaku Tes    Ganti Password    Logout

Form Data Gejala

Gejala Kode :

Gejala Nama :

[kembali ke data gejala](#)

Gambar 3.7 Form Data Gejala

Pada gambar 3.7 di atas merupakan rancangan halaman Form Data Gejala. Halaman ini merupakan halaman dimana pakar dapat memasukkan data-data gejala penyakit yang nantinya digunakan untuk mengolah data yang diinputkan oleh *user*. Pada form ini berisi kode gejala sebagai inisialisasi dan nama gejala dari penyakit kelinci.

Administrator    Gejala    Penyakit    Pencegahan    Pengobatan    Pelaku Tes    Ganti Password    Logout

Daftar Data Gejala

NO	kode gejala	nama gejala	aksi
99	xxx	aaa	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
99	xxx	bbb	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
-	-	-	-
-	-	-	-
99	xxx	zzz	<a href="#">Perbaharui</a> <a href="#">Hapus</a>

[tambah data gejala](#)

Gambar 3.8 Daftar Data Gejala



Pada gambar 3.8 di atas merupakan rancangan halaman Daftar Data Gejala dimana pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang sudah dimasukkan melalui form data gejala.

Administrator    Gejala    Penyakit    Pencegahan    Pengobatan    Pelaku Tes    Ganti Password    Logout

Form Data Penyakit

Gejala Kode :

Gejala Nama :

Penyakit nama ilmiah :

Nilai CF :

**Gejala**

nama gejala

nama gejala

**Pengobatan**

nama pengobatan

nama pengobatan

**Pencegahan**

nama pengobatan

nama pengobatan

[kembali ke data penyakit](#)

Gambar 3.9 Form Data Penyakit

Gambar 3.9 di atas merupakan rancangan halaman Form Data Penyakit dimana pada halaman ini pakar dapat memasukkan data-data penyakit. Pada form ini berisi kode penyakit sebagai inisialisasi dan nama penyakit beserta nama ilmiahnya. Didalam form ini pakar memasukkan nilai MD dan MB untuk perhitungan pada rumus *certainty factor* dan *dempster-shafer*.

Administrator      Gejala   Penyakit   Pencegahan   Pengobatan   Pelaku Tes   Ganti Password   Logout				
Daftar Data Penyakit				
NO	kode penyakit	penyakit nama	penyakit nama ilmiah	aksi
99	xxx	aaa	xxx	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
99	xxx	bbb	xxx	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
99	xxx	zzz	aaa	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
<a href="#">tambah data penyakit</a>				

Gambar 3.10 Daftar Data Penyakit

Gambar 3.10 di atas merupakan rancangan halaman Daftar Data Penyakit dimana pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang sudah dimasukkan melalui form data penyakit.

Administrator      Gejala   Penyakit   Pencegahan   Pengobatan   Pelaku Tes   Ganti Password   Logout	
Form Data Pencegahan	
Pencegahan Kode :	<input type="text"/>
PencegahanNama :	<input type="text"/>
<input type="button" value="Simpan"/>	
<a href="#">kembali ke data pencegahan</a>	

Gambar 3.11 Form Data Pencegahan

Gambar 3.11 di atas merupakan halaman Form Data Pencegahan dimana pada halaman ini pakar dapat memasukkan data-data pencegahan yang nantinya akan diolah. Pada form ini berisi kode pencegahan sebagai inisialisasi dan nama pencegahannya.

NO	kode pencegahan	nama pencegahan	aksi
99	xxx	aaa	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
99	xxx	bbb	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
-	-	-	-
-	-	-	-
-	-	-	-
99	xxx	zzz	<a href="#">Perbaharui</a> <a href="#">Hapus</a>

[tambah data pencegahan](#)

Gambar 3.12 Daftar Data Pencegahan

Gambar 3.12 di atas merupakan rancangan halaman Daftar Data Pencegahan dimana pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang sudah dimasukkan ke dalam sistem melalui form data pencegahan.

Administrator	Gejala	Penyakit	Pencegahan	Pengobatan	Pelaku Tes	Ganti Password	Logout
Form Data Pengobatan							
Pengobatan Kode : <input type="text"/>							
Pengobatan Nama : <input type="text"/>							
<input type="button" value="Simpan"/>							
<a href="#">kembali ke data pengobatan</a>							

Gambar 3.13 Form Data Pengobatan

Gambar 3.13 di atas merupakan rancangan halaman Form Data Pengobatan dimana pada halaman ini pakar dapat memasukkan data-data pengobatan. Pada form ini berisi kode pengobatan sebagai inialisasi dan nama pengobatannya.

Administrator      Gejala    Penyakit    Pencegahan    Pengobatan    Pelaku Tes    Ganti Password    Logout			
Daftar Data Pengobatan			
NO	kode pengobatan	nama pengobatan	aksi
99	xxx	aaa	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
99	xxx	bbb	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
-	-	-	-
-	-	-	-
-	-	-	-
99	xxx	zzz	<a href="#">Perbaharui</a> <a href="#">Hapus</a>
<a href="#">tambah data pengobatan</a>			

Gambar 3.14 Daftar Data Pengobatan

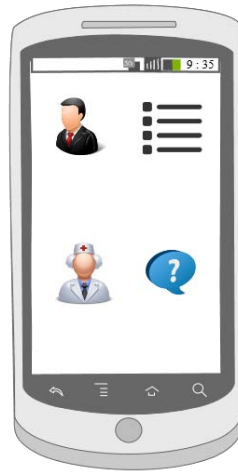
Pada gambar 3.14 di atas merupakan rancangan halaman Daftar Data Pengobatan dimana pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang telah dimasukkan ke dalam sistem.

Administrator      Gejala    Penyakit    Pencegahan    Pengobatan    Pelaku Tes    Ganti Password    Logout						
Daftar Data Tes						
No	Tanggal Tes	Pelaku Tes	No Identitas	Alamat	Phone	Detail
99	dd/mm/yy	aaa	9999	jalan a	xxxxxxxxxxx	<a href="#">Detail</a>
99	dd/mm/yy	aaa	9999	jalan b	xxxxxxxxxxx	<a href="#">Detail</a>
-	-	-	-	-	-	-
-	-	-	-	-	-	-
99	dd/mm/yy	aaa	9999	jalan z	xxxxxxxxxxx	<a href="#">Detail</a>

Gambar 3.15 Daftar Data Tes

Pada gambar 3.15 di atas merupakan rancangan halaman Daftar Data Tes dimana pada halaman ini pakar dapat melihat detail data *user* yang telah melakukan tes melalui aplikasi diagnosis penyakit kelinci melalui android.

### 3.5.2 Rancangan Antarmuka Android



Gambar 3.16 Halaman Utama

Gambar 3.16 di atas merupakan rancangan antarmuka halaman utama pada aplikasi android. Pada halaman ini terdapat 4 buah menu utama yaitu menu akun, menu daftar penyakit, menu tes, dan menu bantuan.



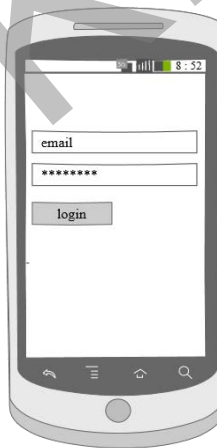
Gambar 3.17 Halaman Akun

Gambar 3.17 merupakan rancangan antarmuka halaman akun dimana pada halaman ini terdapat *login* dan *register*.



Gambar 3.18 Halaman Registrasi

Gambar 3.18 di atas merupakan rancangan antarmuka halaman registrasi. Halaman ini digunakan untuk melakukan registrasi bagi *user* yang belum terdaftar di sistem.



Gambar 3.19 Halaman Login

Gambar 3.19 di atas merupakan rancangan antarmuka halaman login.



Gambar 3.20 Halaman Daftar Penyakit

Gambar 3.20 di atas merupakan rancangan antarmuka halaman daftar penyakit. Pada halaman ini *user* dapat melihat daftar penyakit yang terdapat di dalam aplikasi.



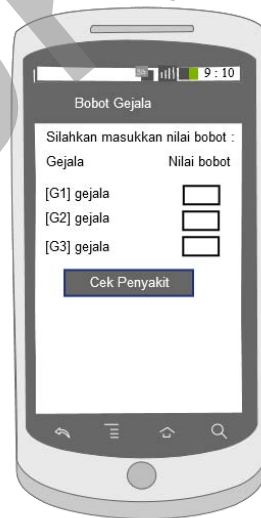
Gambar 3.21 Halaman Pemilihan Gejala Utama

Gambar 3.21 di atas merupakan rancangan antarmuka halaman pemilihan gejala utama.



Gambar 3.22 Halaman Pemilihan Gejala Pendukung

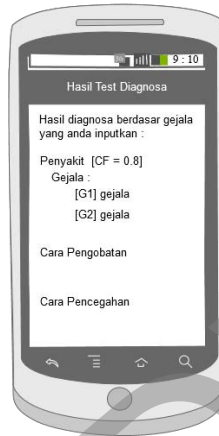
Gambar 3.22 di atas merupakan rancangan halaman antarmuka halaman pemilihan gejala pendukung setelah pemilihan gejala utama.



Gambar 3.23 Halaman Input Bobot Gejala

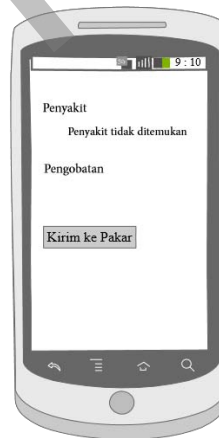


Gambar 3.23 di atas merupakan rancangan antarmuka halaman Input Bobot Gejala. Pada halaman ini *user* mengisi setiap bobot gejala yang telah dipilih pada menu pemilihan gejala sebelumnya yang kemudian digunakan untuk menghitung nilai *certainty factor* dan *dempster-shafer*.



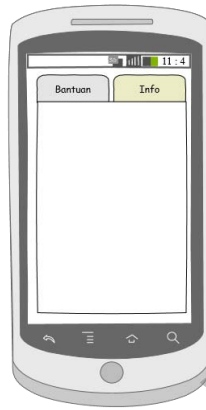
Gambar 3.24 Halaman Hasil Tes

Gambar 3.24 di atas merupakan rancangan antarmuka halaman hasil tes yang berupa penyakit, cara pengobatan serta cara pencegahan.



Gambar 3.25 Halaman Kirim ke Pakar

Gambar 3.25 di atas merupakan rancangan antarmuka halaman kirim ke pakar dimana halaman ini akan muncul apabila dari penghitungan sistem hasil akhirnya tidak ditemukan.



Gambar 3.26 Halaman Bantuan

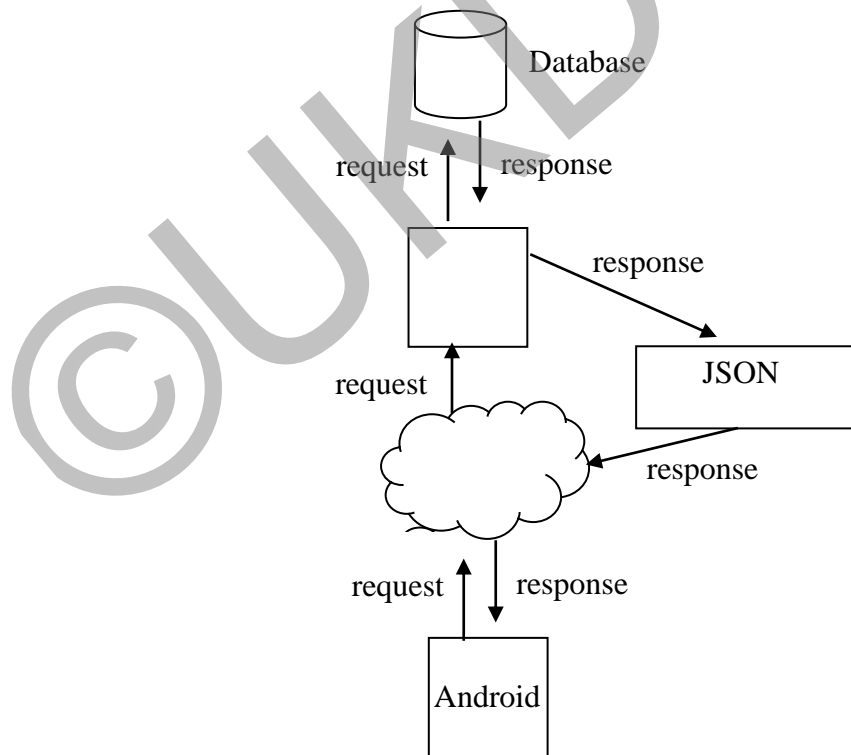
Gambar 3.26 di atas merupakan rancangan antarmuka halaman bantuan. Pada halaman ini memiliki 2 tab yaitu tab bantuan dan tab info dimana tab bantuan berisi cara memakai aplikasi dan pada tab info berisi informasi mengenai aplikasi android ini.

## BAB IV IMPLEMENTASI DAN ANALISIS SISTEM

### 4.1 Implementasi Basis Pengetahuan Sistem

#### 4.1.1 Arsitektur Sistem

Gambar 4.1 merupakan gambar arsitektur sistem untuk Sistem Pakar Diagnosis Penyakit Kelinci dengan menggunakan *web service* sebagai komunikasi data dan android sebagai antarmuka seperti yang telah dikerjakan oleh Atmaja (2014). Dalam sistem ini android terhubung ke internet kemudian meminta data dari server, lalu server mencari data yang ada di *database*. Setelah data didapat, data dikirim ke server dimana data diolah dan diubah bentuknya menjadi JSON kemudian data diterima dan diolah android untuk kemudian ditampilkan.



Gambar 4.1 Arsitektur Sistem

#### 4.1.2 Implementasi Antarmuka

Aplikasi ini memiliki 2 bagian sistem yaitu *web service* sebagai sistem untuk pengolahan data dan Android sebagai aplikasi yang digunakan oleh *user*.

a) Halaman Data Penyakit

Gambar 4.2 di bawah merupakan implementasi halaman data penyakit. Halaman ini berisi data – data penyakit yang ada di *database*. Pada halaman ini admin dapat memperbaharui maupun menghapus data penyakit melalui *link* yang tersedia.



Daftar Data Penyakit

12 ≥

No.	Kode Penyakit	Penyakit Nama	Penyakit Namalatin	Nilai CF	Aksi
1	P12	Bloat	Tympahni	0.7	<a href="#">Perbarui</a> <a href="#">Hapus</a>
2	P12	Bloat	Tympahni	0.7	<a href="#">Perbarui</a> <a href="#">Hapus</a>
3	P12	Bloat	Tympahni	0.8	<a href="#">Perbarui</a> <a href="#">Hapus</a>
4	P9	Cacingan	-	0.8	<a href="#">Perbarui</a> <a href="#">Hapus</a>
5	P9	Cacingan	-	0.8	<a href="#">Perbarui</a> <a href="#">Hapus</a>
6	P9	Cacingan	-	0.8	<a href="#">Perbarui</a> <a href="#">Hapus</a>
7	P9	Cacingan	-	0.5	<a href="#">Perbarui</a> <a href="#">Hapus</a>
8	P1	Enteritis Kompleks	-	0.8	<a href="#">Perbarui</a> <a href="#">Hapus</a>
9	P1	Enteritis Kompleks	-	0.4	<a href="#">Perbarui</a> <a href="#">Hapus</a>
10	P1	Enteritis Kompleks	-	0.9	<a href="#">Perbarui</a> <a href="#">Hapus</a>

[tambah data penyakit](#)

Gambar 4.2 Halaman Data Penyakit

## b) Halaman Input Data Penyakit

Gambar 4.3 di bawah merupakan implementasi dari halaman *input* data penyakit. Pada halaman ini admin dapat memasukan kode penyakit, nama penyakit, nama latin penyakit, nilai CF, gejala, pengobatan, serta pencegahan.

The screenshot shows the 'Form Data Penyakit' interface. At the top, there is a navigation bar with a rabbit icon and the text 'Aekabulatan'. The main form area is titled 'Form Data Penyakit' and contains the following fields and sections:

- Kode**: P12
- Penyakit Nama**: Bloat
- Penyakit Namalatin**: Tympahni
- Nilai CF Penyakit**: 0.8

The form is divided into three main sections, each with a blue header bar:

- Gejala**:
  - [ G1] Suhu badan naik
  - [ G10] Daerah anus kotor
  - [ G11] Kotorannya hijau gelap dan bau
  - [ G12] Lendir menggantung pada dubur
- Pengobatan**:
  - [OB-1] Selama 1 hari hewan diberi pakan hay atau daun kacang kering
  - [OB-10] Olesi dengan salep Belerang
  - [OB-11] Olesi dengan salep Caviam
- Pencegahan**:
  - [PC-1] Berikan pakan cukup serat kasar
  - [PC-10] Beri hijauan segar yang cukup

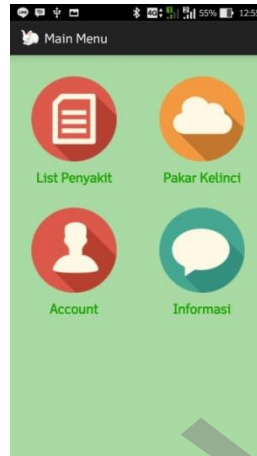
At the bottom of the form, there is a 'Perbarui Data' button and a link 'kembali ke data penyakit'.

Gambar 4.3 Halaman Input Data Penyakit

Pada aplikasi android terdapat 4 menu yang bisa dipilih yaitu menu *list* penyakit, menu pakar penyakit, menu akun, dan menu informasi.

c) Halaman Menu Utama

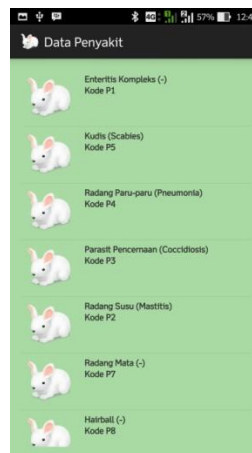
Pada gambar 4.4 di bawah merupakan halaman utama dari aplikasi android.



Gambar 4.4 Halaman Menu Utama

d) Halaman Daftar Penyakit

Pada gambar 4.5 di bawah ini merupakan implementasi dari halaman daftar penyakit. Pada halaman ini *user* dapat melihat daftar penyakit yang telah tersedia di aplikasi.



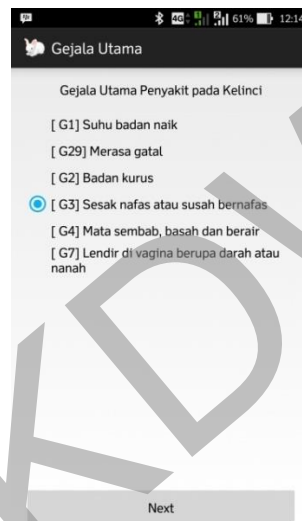
Gambar 4.5 Halaman Daftar Penyakit

e) Halaman Pemilihan Gejala Utama

Gambar 4.6 di bawah merupakan halaman awal saat memasuki menu pakar.

Disini *user*

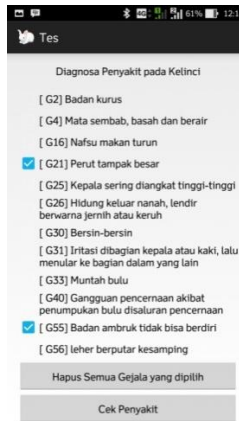
dapat memilih satu gejala utama yang terlihat pada kelinci. Setelah *user* memilih gejala utama, *user* akan diarahkan menuju menu selanjutnya dengan menekan tombol *next*.



Gambar 4.6 Halaman Pemilihan Gejala Utama

f) Halaman Pemilihan Gejala Pendukung

Pada gambar 4.7 di bawah merupakan halaman pemilihan gejala pendukung. Pada halaman ini *user* dapat memilih lebih dari satu gejala pendukung yang dialami kelinci.



Gambar 4.7 Halaman Pemilihan Gejala Pendukung

g) Halaman *Input* Bobot Gejala

Gambar 4.7 di bawah ini merupakan halaman untuk memasukan bobot dari masing – masing gejala yang telah dipilih oleh *user*. Bobot pada gejala utama sudah *default* 1 ( satu ) karena itu merupakan gejala yang dialami oleh kelinci dimana gejala terlihat secara kasat mata sedangkan untuk gejala pendukung *user* dapat memasukan nilai bobot antara 0.1 sampai 0.9. Setelah semua bobot dimasukan, *user* dapat menekan tombol cek penyakit. Bobot yang telah dimasukan oleh *user* kemudian dikirim ke *web service* untuk dicocokkan di *database*. Selanjutnya *web service* memberikan *response* berupa hasil dari perhitungan gejala yang telah dimasukan oleh *user*.

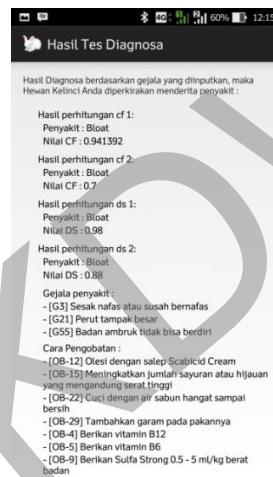


Gambar 4.7 Halaman *Input* Bobot Gejala



h) Halaman Hasil Diagnosis ( Penyakit Ditemukan Semua )

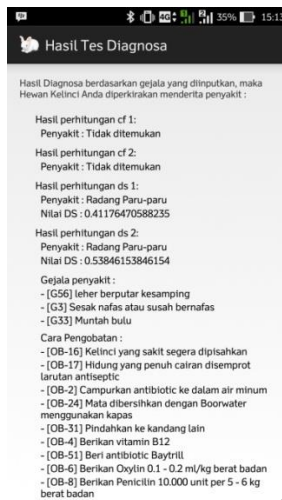
Gambar 4.8 di bawah ini merupakan halaman hasil diagnose penyakit kelinci. Pada halaman ini ditampilkan hasil perhitungan dari gejala – gejala yang telah dimasukan oleh *user* dimana hasilnya terdiri dari 4 cara perhitungan, 2 hasil merupakan perhitungan dengan menggunakan metode *Certainty Factor* dan sisanya merupakan perhitungan menggunakan metode *Dempster Shafer*. Pada halaman ini juga ditampilkan cara pencegahan dan pengobatan mengenai penyakit yang diderita oleh kelinci *user*.



Gambar 4.8 Halaman Hasil Diagnosis ( Penyakit Ditemukan )

i) Halaman Hasil Diagnosis ( Penyakit Ditemukan Sebagian )

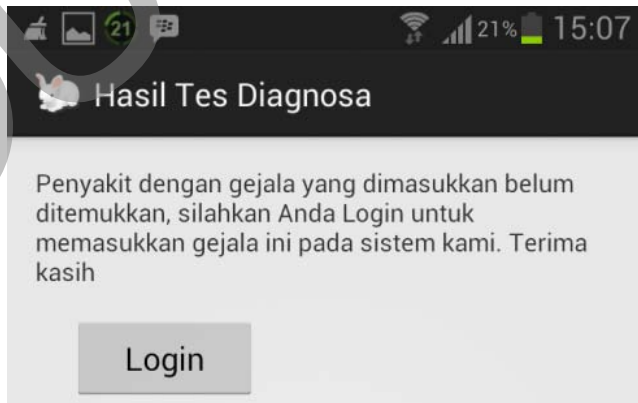
Pada gambar 4.9 di bawah ini merupakan halaman hasil diagnose dimana hasil penyakit hanya ditemukan sebagian, hasil bisa ditemukan hanya pada perhitungan menggunakan metode *Certainty Factor* maupun pada penggunaan metode *Dempster-Shafer*.



Gambar 4.9 Halaman Hasil Diagnosis ( Penyakit Ditemukan Sebagian )

j) Halaman Hasil Diagnosis ( Penyakit Tidak Ditemukan )

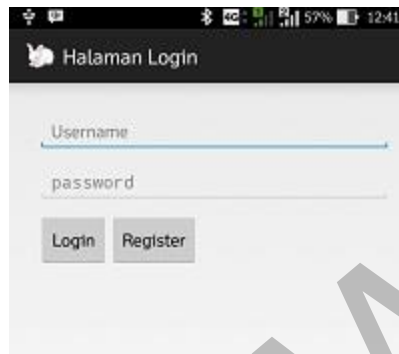
Gambar 4.10 di bawah merupakan halaman hasil diagnosis dimana penyakit tidak berhasil ditemukan. Pada halaman ini *user* akan diarahkan untuk melakukan pendaftaran apabila belum mempunyai akun dan jika *user* sudah memiliki akun maka *user* dapat mengirimkan data gejala tersebut ke *database* agar bisa di *review* oleh pakar.



Gambar 4.10 Halaman Hasil Diagnosis ( Penyakit Tidak Ditemukan )

k) Halaman Akun

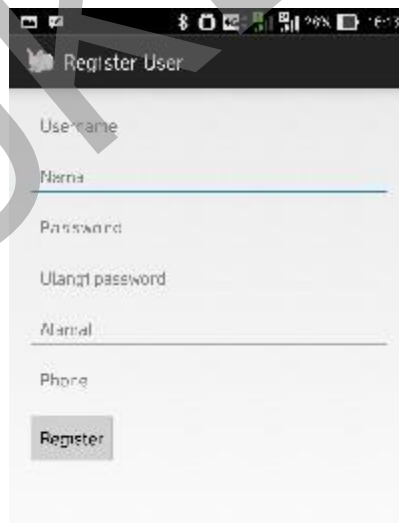
Gambar 4.11 di bawah merupakan halaman akun dari aplikasi dimana *user* dapat melakukan *login* atau melakukan *register*.



Gambar 4.11 Halaman Akun

l) Halaman Daftar

Gambar 4.12 merupakan halaman daftar pada aplikasi dimana *user* dapat melakukan pendaftaran akun pada sistem.



Gambar 4.12 Halaman Daftar

### 4.1.3 Implementasi Algoritma

Dalam membangun aplikasi Perbandingan Penggunaan Metode *Dempster-Shafer* dengan *Certainty Factor* pada Kasus Diagnosis Penyakit Kelinci Hias

menggunakan algoritma *Forward Chaining*, *Certainty Factor*, dan *Dempster-Shafer*. Potongan program beserta pseudocode dari algoritma *forward chaining* dan *Certainty Factor* ini merupakan potongan yang telah dibuat oleh Atmaja (2014), penulis menambahkan untuk metode *Dempster-Shafer*. Berikut merupakan potongan program beserta pseudocode dari algoritma yang penulis terapkan untuk membangun aplikasi ini.

#### 1. Algoritma dan Implementasi *Forward Chaining* Pengambilan Gejala Utama

Pada proses ini dilakukan pemilihan gejala utama yang sering muncul atau gejala yang memiliki keterkaitan dengan lebih dari 1 penyakit serta memiliki gejala pendukung. Berikut adalah kode program dan algoritma serta penerapan *forward chaining* dalam pengambilan penentuan gejala utama :

```
//ambil data gejala utama
public function getGejalaUtama(){
    $data = array();
    $this->db->order_by('gejala_kode');
    $this->db->where('gejala_status',1);
    $q_gejala = $this->db->get('gejala')->result();

    foreach($q_gejala as $r){
        $d_gejala[] = $r->id_gejala;
        $d_gejala_array[$r->id_gejala] = '['.$r->gejala_kode.' '.$r->gejala_nama;
    }

    //memanggil fungsi get_penyakit_gejala() untuk menampilkan gejala-gejala
    yang ada pada data penyakit.
    $d_penyakit = $this->get_penyakit_gejala();
    $hasil = array();

    //fungsi rekursif yang nantinya memanggil fungsi yg berulang dan kemudian
    mjd gejala utama
    $d_gejala_utama = $this->GejalaUtama($d_gejala, $d_penyakit, 0, $hasil);
```

```

foreach($d_gejala_utama as $v){
    $d_gejalanya[$v] = $d_gejala_array[$v];
}
asort($d_gejalanya);
echo json_encode($d_gejalanya);
}

```

### Pseudocode *Forward Chaining* Gejala Utama

```

GET tabel gejala
WHERE gejala_status = 1
ORDER by gejala_kode asc

Foreach var gejala as var r
    SET var array d_gejala[] = var r->id_gejala
    SET var array d_gejala_array[] = 'var r->gejala_kode' . 'var r-
    >gejala_nama'

SET var d_penyakit
GET var get_penyakit_gejala
SET hasivar 1 = array()

SET d_gejala_utama
GET fungsi GejalaUtama(var d_gejala , var d_penyakit, 0 ,var hasil)
foreach var d_gejala_utama as var v
    SET var array d_gejala[var v] = var array d_gejala_array[var v]
Asort var d_gejalanya
Echo json_encode( var d_gejalanya)

```

## 2. Algoritma dan Implementasi *Forward Chaining* Pengambilan Gejala Pendukung

Pada proses ini merupakan lanjutan dari proses pemilihan gejala utama, pengguna memilih gejala pendukung terlebih dahulu sebelum melakukan perhitungan diagnosis penyakit kelinci. Berikut ini merupakan kode program dan algoritma serta penerapan *forward chaining* dalam pengambilan penentuan gejala pendukung :

```
public function getTest($gejalaUtama){
    $data = array();
    //cari penyakit yang memiliki gejala utama
    $this->db->where('pg.penakit_gejala_status', 1);
    $this->db->where('pg.id_gejala', $gejalaUtama);
    $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penakit_gejala pg');

    foreach ($query->result() as $row){
        $penakit[] = $row->id_penakit;
    }
    //cari gejala yang sesuai penyakitnya dengan gejala utama
    $this->db->where('pg.penakit_gejala_status', 1);
    $this->db->where_in('pg.id_penakit', $penakit);
    $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penakit_gejala pg');

    foreach ($query->result() as $row){
        $data[$row->id_gejala] = [ '$row->gejala_kode .' ] '$row->gejala_nama;
    }
    echo json_encode($data);
}
```

### Pseudocode *Forward Chaining* Pengambilan Gejala Pendukung

```
SET var query
WHERE penyakit_gejala_status = 1
WHERE var id_gejala = var gejalaUtama
JOIN var gejala g , var g.id_gejala = var pg.id_gejala

foreach (var query->result() as var row)
    SET var array penyakit[] = var row->var id_penyakit
SET var query
GET var penyakit_gejala
WHERE penyakit_gejala_status = 1
WHERE_IN pg.id_penyakit , penyakit
JOIN gejala g , g.id_gejala = pg.id_gejala

foreach (var query->result() as var row){
    SET var arraydata[var row->id_gejala] = [ '. var row->gejala_kode .' ] '.var
row- >gejala_nama
echo json_encode(var data)
```

### 3. Algoritma dan Implementasi Menentukan Nilai Minimal dan CF Kombinasi

Pada proses ini sistem mengambil nilai minimal dari setiap *rule* agar bisa dihitung dalam rumus **CF Rule = min(a,b) x CF pakar**, dimana hasil dari perhitungan tersebut akan digunakan untuk menghitung CF kombinasi.

Setelah nilai *CF Rule* didapat, sistem akan melakukan pengecekan terlebih dahulu terhadap gejala yang telah dipilih oleh pengguna dengan *rule* yang ada, jika cocok maka akan disimpan dahulu agar dihitung pada CF kombinasi yang terakhir.

Rumus untuk CF kombinasi adalah  $CF_{combine} = CF_1 + CF_2 (1 - CF_1)$ . Rumus tersebut untuk mengetahui bobot sesungguhnya dari penyakit yang akan muncul di aplikasi. Berikut adalah kode program dan algoritma serta penerapan untuk menentukan nilai minimal dan menghitung CF Kombinasi :

```
//comment : fungsi untuk get minimal input
function getMINIMAL($nilai=",$a="){
    $nilai = $nilai;
    $a = $a;
    $b = array_keys($nilai);
    $x = array_diff($b,$a);

    foreach ($x as $k => $v) {
        unset($nilai[$v]);
    }

    return min($nilai);
}

//perhitungan untuk cf kombinasi
function CF_KOMBINASI($query_cocok, $rule, $cf_pakar, $input_user){
    foreach ($rule as $key => $value) {
        $min_input[$key] = getMINIMAL($input_user, $value );
    }

    foreach ($cf_pakar as $key => $value) {
        $CF_rule_index_sort_x[$key] = $min_input[$key] *
$cf_pakar[$key];
        $CF_rule[$key] = $min_input[$key] * $cf_pakar[$key];
        $CF_rule_asli[$key] = $min_input[$key] * $cf_pakar[$key];
    }
}
```



```

unset($CF_rule_index_sort_x[$query_cocok]);
foreach ($CF_rule_index_sort_x as $key => $value) {
    $CF_rule_index_sort[] = $value;
}

unset($CF_rule[$query_cocok]);
$CF_rule_selain_1 = $CF_rule;

unset($rule[$query_cocok]);
$rulesisa = $rule;
$keyrulesisa = array_keys($rulesisa);

$y = $CF_rule_index_sort[0];
for ($i=0;$i<=count($keyrulesisa)-2;$i++){
    if($i==0){
        $CF_LOOPING_COMBINASI = $y +
$CF_rule_selain_1[$keyrulesisa[($i+1)]] * ( 1- $y );
        $z[$i] = $CF_LOOPING_COMBINASI;
    }elseif($i==1){
        $CF_LOOPING_COMBINASI = $z[0] +
$CF_rule_selain_1[$keyrulesisa[($i+1)]] * ( 1- $z[0] );
        $z[1] = $CF_LOOPING_COMBINASI;
    }else{
        $CF_LOOPING_COMBINASI = $z[1] +
$CF_rule_selain_1[$keyrulesisa[($i+1)]] * ( 1- $z[1] );
    }
}

$hasil_kombinasi = $CF_LOOPING_COMBINASI +
$CF_rule_asli[$query_cocok] * ( 1 - $CF_LOOPING_COMBINASI );

```

```

return $hasil_kombinasi;
}    unset($rule[$query_cocok]);
    $rulesisa = $rule;
    $keyrulesisa = array_keys($rulesisa);

    $y = $CF_rule_index_sort[0];
    for ($i=0;$i<=count($keyrulesisa)-2;$i++){
        $CF_LOOPING_COMBINASI    =    $y    +
$CF_rule_selain_1[$keyrulesisa[$i+1]] * ( 1- $y );
    }

    $hasil_kombinasi    =    $CF_LOOPING_COMBINASI    +
$CF_rule_asli[$query_cocok] * ( 1 - $CF_LOOPING_COMBINASI );

    return $hasil_kombinasi;
}

```

### Pseudocode Fungsi untuk menghitung CF Kombinasi

```

// fungsi untuk get minimal input untuk menentukan nilai minimal dari gejala di
setiap rule berdasarkan input dari user

SET nilai minimal dari ( param nilai dan param a )
    SET var nilai = param nilai
    SET var a = param a
    SET var b = array_keys dari(var nilai)
    SET var x = array_diff dari (var b, var a)

    foreach (var x as var k => var v) //k sebagai key dan v sebagai value
        unset (var nilai[var v]) // unset adalah untuk mengeluarkan
        sebuah data dari array untuk tidak diikutsertakan d awal perhitungan

    return min(nilai) //hasil pengembalian nilai dari bobot minimal

```

```

//perhitungan untuk CF kombinasi

SET nilai dari param query_cocok, param rule, param cf_pakar
    foreach (param rule as var key => var value)
        SET array min_input[var key as key] = GET fungsi
getMINIMAL (param          input_user, var value)
        //tujuan foreach ini adalah membentuk sebuah array yang bernama
min_input yang didalamnya terdapat nilai minimal dari masing-masing rule yang
masuk dalam kondisi

//menghitung CF Rule
    foreach (param cf_pakar as var key => var value)
        SET var array CF_rule_index_sort_x[var key] = GET
min_input[var key] * GET cf_pakar[var key];
        SET var array CF_rule[var key] = GET min_input[var key] *
GET cf_pakar[var key];
        SET CF_rule_asli[var key] = GET min_input[var key] *
GET cf_pakar[var key];
        //membuat 3 variabel untuk menentukan nilai CF Rule untuk
nantinya dipanggil pada perhitungan CF Combine
// memisahkan rule yang cocok dengan rule yang lain atau yg tidak cocok
        unset(array CF_rule_index_sort_x[var query_cocok])
        foreach (param CF_rule_index_sort_x as var key => var value)
            SET var array CF_rule_index_sort[] = var value

        unset(var array CF_rule[var query_cocok])
        SET var CF_rule_selain_1 = var CF_rule

```

```

unset(var array rule[var query_cocok])
SET var rulesisa = var rule
SET var keyrulesisa = array_keys($rulesisa)

//untuk melakukan perulangan penghitungan CF kombinasi
SET var y = var array CF_rule_index_sort[0]
for (var i=0;var i<=count(var keyrulesisa)-2;var i++)
    IF var i = 0
        SET var CF_LOOPING_COMBINASI = var y + var array
CF_rule_selain_1[var keyrulesisa[(var i+1)]] * ( 1- var y )
        SET var array z[i] = var CF_LOOPING_COMBINASI
        ELSE var i = 1
            Var CF_LOOPING_COMBINASI = Var array z[0] + var
CF_rule_selain_1[var keyrulesisa[(var i+1)]] * ( 1- var array z[0] );
            Var z[1] = var CF_LOOPING_COMBINASI
        ELSE
            Var CF_LOOPING_COMBINASI = var array z[1] + var
CF_rule_selain_1[var keyrulesisa[(var i+1)]] * ( 1- var z[1] )

SET var hasil_kombinasi = var CF_LOOPING_COMBINASI +
var array CF_rule_asli[var query_cocok] * ( 1 - var
CF_LOOPING_COMBINASI )

return var hasil_kombinasi

```

#### 4. Algoritma dan Implementasi *Dempster-Shafer*

Pada proses ini gejala – gejala yang sudah memiliki bobot, dicari terlebih dahulu hubungan antara gejala yang telah dipilih dengan penyakit yang memiliki keterkaitan dengan gejala tersebut. Setelah itu dicari nilai semesta dari gejala pertama, kemudian hasil dari bobot dan nilai semesta tersebut disimpan di variable sementara.

Setelah itu dicari keterkaitan antara gejala kedua dengan penyakit yang ada di *database*, kemudian dicari nilai semestanya dan disimpan di variable sementara. Lalu dari gejala pertama dan kedua dilakukan perhitungan kombinasi untuk menentukan nilai  $m_3$ , setelah nilai  $m_3$  didapat kemudian disimpan kembali di variabel sementara. Proses untuk mencari keterkaitan antara gejala dan penyakit, bobot dari nilai semesta dan perhitungan kombinasi terus dilakukan hingga gejala terakhir. Berikut adalah kode program untuk mencari keterkaitan antara gejala dan penyakit, nilai semesta dan perhitungan kombinasi antar gejala :

```
function perhitungan_densitas($array_densitas) {  
  
    $history = array();  
    $collection = array();  
  
    $kom = 2;  
    foreach( $array_densitas as $kd => $vd ) {  
        $kd = key($vd);  
        $vd = $vd[$kd];  
  
        if (empty($collection)) {  
            $collection[$kd] = $vd;  
            $collection[""] = 1 - $vd;  
            $history['m'][] = $collection;  
            continue;  
        }  
  
        $hasil = array();  
  
        foreach ($collection as $k => $v) {
```

```

        if ($k != '') {
            $k1 = implode(',', array_intersect(explode(',',
$kd), explode(',', $kd))) ? ':' : '-';
        } else {
            $k1 = implode(',', explode(',', $kd));
        }

        // perhitungan kolom belief
        $history['kombinasi'][$kom][][$k1] = $v * $vd;
        $hasil[$k1] = @$hasil[$k1] + $v * $vd;

        // perhitungan kolom plausibility
        $history['kombinasi'][$kom][][$k] = $v * (1 - $vd);
        $hasil[$k] = @$hasil[$k] + $v * (1 - $vd);
    }

    $pengurang = 0;
    if ( @$hasil['-'] ) {
        $pengurang = $hasil['-'];
        if($pengurang >= 1) $pengurang = 0.999999;
    }
    unset($hasil['-']);

    // perhitungan seluruh
    foreach ( $hasil as $ka => $va ) {
        $hasil[$ka] = $va / (1 - $pengurang);
    }

    $collection = $hasil;
    $history['m'][] = array(
        $kd => $vd,
        " => (1 - $vd),
    );
    $history['m'][] = $collection;
    $kom = $kom + 2;
}

#pre($history);
$result['diagnosa'] = $array_densitas;
$result['perhitungan'] = $history;
$result['hasil'] = array(
    'penyakit' => array_search(max($collection), $collection),

```

```

        'densitas' => max($collection),
    );
    return $result;
}

```

## 4.2 Analisis Sistem

Analisis sistem bertujuan untuk mengetahui perbandingan hasil akhir dari penggunaan metode *Certainty Factor* dengan metode *Dempster-Shafer* pada diagnosis penyakit kelinci hias. Perhitungan dilakukan 2 kali untuk masing – masing metode, dimana perhitungan pertama (  $CF_1$  dan  $DS_1$  ) merupakan perhitungan dimana *user* memasukan bobot dari masing – masing gejala yang dipilih, sedangkan perhitungan kedua (  $CF_2$  dan  $DS_2$  ) merupakan perhitungan dimana bobot dari setiap gejala yang telah dipilih oleh *user* diperoleh dari data bobot gejala yang terdapat pada database. Pengujian terhadap sistem dilakukan sebanyak 20 kasus, hasil diagnosis sistem dapat dilihat pada tabel 4.1. Perincian dan bobot tiap gejala kasus dapat dilihat pada lampiran.

Tabel 4.1 Analisis Data Testing

No	Diagnosis Sistem <i>Certainty Factor</i>			Diagnosis Sistem <i>Dempster-Shafer</i>		
	Penyakit	$CF_1$	$CF_2$	Penyakit	$DS_1$	$DS_2$
1	Enteritis Kompleks	0.91	0.99	Enteritis Kompleks	0.97	0.87
2	Bloat ( <i>Tympany</i> )	0.82	0.7	Bloat ( <i>Tympany</i> )	0.94	0.88
3	Cacing Gilig ( <i>Pinworm</i> )	0.91	0.7	Cacing Gilig ( <i>Pinworm</i> )	0.88	0.79
4	Tidak ditemukan	-	-	Tidak ditemukan	-	-
5	Infeksi Kandungan	0.9	0.7	Infeksi Kandungan	0.99	0.8

Tabel 4.1 (Sambungan)

No	Diagnosis Sistem <i>Certainty Factor</i>			Diagnosis Sistem <i>Dempster-Shafer</i>		
	Penyakit	CF <sub>1</sub>	CF <sub>2</sub>	Penyakit	DS <sub>1</sub>	DS <sub>2</sub>
6	Radang Paru-paru ( <i>pneumonia/pasteurel loises</i> )	0.86	0.8	Radang Paru-paru ( <i>pneumonia/pasteur elloises</i> )	0.97	0.98
7	Jamur ( <i>Ringworm</i> )	0.94	0.9	Jamur ( <i>Ringworm</i> )	0.48	0.46
8	Radang Susu ( <i>Mastitis</i> )	0.96	0.7	Radang Susu ( <i>Mastitis</i> )	0.98	0.92
9	Tidak ditemukan	-	-	Tidak ditemukan	-	-
10	Tidak Ditemukan	-	-	Enteritis Kompleks	0.34	-
11	Parasit Pencernaan ( <i>Coccidiosis</i> )	0.97	0.67	Parasit Pencernaan ( <i>Coccidiosis</i> )	0.8	0.7
12	Tidak ditemukan	-	-	Tidak ditemukan	-	-
13	Kudis ( <i>Scabies</i> )	0.82	0.71	Kudis ( <i>Scabies</i> )	0.9	1
14	Radang Mata	0.94	0.81	Radang Mata	0.84	0.92
15	Tidak ditemukan	-	-	Tidak ditemukan	-	-
16	Radang Mata	0.94	0.81	Radang Mata	0.95	0.98
17	Jamur ( <i>Ringworm</i> )	0.91	0.6	Jamur ( <i>Ringworm</i> )	0.6	-
18	Cacing Gilig ( <i>Pinworm</i> )	0.84	0.84	Cacing Gilig ( <i>Pinworm</i> )	0.91	0.94
19	Cacingan	0.7	0.44	Cacingan	0.8	0.4
20	Makan Bulu ( <i>Hairball</i> )	0.83	-	Makan Bulu ( <i>Hairball</i> )	0.88	0.8
21	Cacingan	0.63	0.52	Cacingan	0.79	0.76
22	Radang Paru-paru ( <i>pneumonia/pasteurel loises</i> )	0.84	0.75	Radang Paru-paru ( <i>pneumonia/pasteur elloises</i> )	0.96	0.98



Tabel 4.1 (Sambungan)

No	Diagnosis Sistem <i>Certainty Factor</i>			Diagnosis Sistem <i>Dempster-Shafer</i>		
	Penyakit	CF <sub>1</sub>	CF <sub>2</sub>	Penyakit	DS <sub>1</sub>	DS <sub>2</sub>
23	Tidak ditemukan	-	-	Enteritis Kompleks	0.9	0.87
24	Radang Susu ( <i>Mastitis</i> )	0.93	0.55	Radang Susu ( <i>Mastitis</i> )	0.98	0.97
25	Hairball	0.3	-	Hairball	0.28	-
	Jumlah	15.95	12.19	Jumlah	16.24	15.02

$$CF_1 \text{ rata - rata} = 15.95/25 = 0.638$$

$$CF_2 \text{ rata - rata} = 12.19/25 = 0.488$$

$$DS_1 \text{ rata - rata} = 16.24/25 = 0.649$$

$$DS_2 \text{ rata - rata} = 15.02/25 = 0.6$$

$$\text{Persentase perbandingan } CF_1 \text{ dan } DS_1 = 0.64 : 0.65$$

$$\text{Persentase perbandingan } CF_2 \text{ dan } DS_2 = 0.49 : 0.6$$

Perbandingan persentase antara CF<sub>1</sub> dan DS<sub>1</sub> sebesar 64% berbanding 65% dan perbandingan persentase antara CF<sub>2</sub> dan DS<sub>2</sub> sebesar 49% berbanding 60%.

$$\text{Persentase kecocokan } CF_1 \text{ dengan Pakar} = \frac{18}{25} \times 100\% = 72\%$$

$$\text{Persentase kecocokan } DS_1 \text{ dengan Pakar} = \frac{19}{25} \times 100\% = 76\%$$

$$\text{Persentase kecocokan } CF_2 \text{ dengan Pakar} = \frac{17}{25} \times 100\% = 68\%$$

$$\text{Persentase kecocokan } DS_2 \text{ dengan Pakar} = \frac{18}{25} \times 100\% = 72\%$$

Tabel 4.2 Perbandingan Tingkat Akurasi

	Sistem				Atmaja(2014)
	CF		DS		80%
Tingkat Akurasi	CF1	CF2	DS1	DS2	
	72%	68%	76%	72%	

Pada penelitian yang dilakukan Atmaja (2014) disebutkan bahwa sistem memiliki tingkat akurasi sebesar 80% dengan rata-rata kepastian sebesar 0.74. Sedangkan pada penelitian yang penulis lakukan didapat bahwa sistem memiliki tingkat akurasi sebesar 72% untuk CF<sub>1</sub> dengan rata-rata kepastian sebesar 0.64. Dan pada penggunaan metode DS<sub>1</sub> didapat tingkat akurasi sebesar 76% dengan rata-rata probabilitas sebesar 0.65. Masing-masing perhitungan dilakukan dengan menggunakan *input* bobot dari *user*.

Sedangkan pada perhitungan dimana bobot gejala didapat dari basis data didapat bahwa sistem memiliki tingkat akurasi sebesar 68% dengan rata-rata kepastian sebesar 0.49 untuk perhitungan dengan CF<sub>2</sub>, sedangkan untuk perhitungan dengan DS<sub>2</sub> didapat tingkat akurasi sebesar 72% dengan nilai probabilitas sebesar 0.6.

Semua data contoh kasus menggunakan data penelitian yang dilakukan oleh Atmaja (2014), namun untuk besaran bobot setiap gejala berbeda antara penulis dengan yang dilakukan Atmaja (2014). Hal ini dikarenakan dalam dokumen yang penulis terima, pada lampiran contoh kasus tidak tercantum besar bobot untuk setiap gejala yang digunakan untuk melakukan analisa penyakit.

### 4.3 Kelebihan dan Kekurangan Sistem

Berdasarkan implementasi sistem maka dapat diketahui kelebihan serta kekurangan dari sistem. Kelebihan sistem adalah sistem mampu melakukan diagnosis penyakit kelinci dan melakukan perbandingan hasil diagnosis dari penggunaan 2 metode. Sistem juga mampu memberi masukan kepada *user* mengenai pencegahan dan pengobatan berbagai penyakit kelinci.

Aplikasi ini memiliki kekurangan yaitu apabila gejala dan bobot yang dimasukkan secara acak atau terlalu kecil maka hasil akhir tidak dapat menemukan penyakit yang akan didiagnosis. Pengobatan dan pencegahan yang diberikan masih bersifat umum sehingga perlu dilakukan penambahan informasi yang lebih rinci agar didapat hasil yang lebih lengkap.

©UKDW

## Listing Program

## Web Service

## a) Admins.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access
allowed');

class Admins extends CI_Controller {

    public function index()
    {
        $data = array();
        if (isset($_POST['b_login'])){
            $this->db->where('username',$_POST['f_username']);
            $this->db->where('passwd',md5($_POST['f_passwd']));
            $cekdata = $this->db->get('admin');
            if ($cekdata->num_rows() > 0){
                $data_admin = $cekdata->row();
                $this->session->set_userdata($data_admin);
                $url = base_url().'admins/home';
                redirect($url);
            }else{
                $data['error'] = 'Kombinasi User dan Password
salah';
            }
        }
        $this->load->view('admin_index',$data);
    }

    public function petugas(){
        $data = array();
        $data['petugas'] = 1;

        if (isset($_POST['b_login'])){
            $this->db->where('nip',$_POST['f_username']);
            $this->db->where('passwd',$_POST['f_passwd']);
            $cekdata = $this->db->get('petugas');
            if ($cekdata->num_rows() > 0){
                $data_admin = $cekdata->row();
                $this->session->set_userdata($data_admin);
                $url = base_url().'admins/home';
                redirect($url);
            }else{
                $data['error'] = 'Kombinasi User dan Password
salah';
            }
        }

        $this->load->view('admin_index',$data);
    }

    public function home(){

```

```

        $data = array();

        $this->load->view('admin_home', $data);
    }

    public function passwd(){
        $data = array();
        if ( isset($_POST['b_simpan']) ){
            $d_p['passwd'] = md5($_POST['f_passwd']);
            $this->db->update('admin', $d_p);
            $url = base_url().'admins/logout';
            $this->session->set_flashdata('pesan', 'Data
berhasil di-Perbarui');
        }

        /* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
        if( isset($url) ){
            redirect($url);
        }

        $this->load->view('admin_passwd', $data);
    }

    public function logout(){
        $this->session->sess_destroy();
        $url = base_url().'admins';
        redirect($url);
    }

    public function sesi(){
        pre($this->session->all_userdata());
    }
}

```

## b) Android.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Android extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel', 'p');
    }

    public function getPenyakit() {
        $data = array();
        $data['result'] = 'false';

        $sql = "SELECT penyakit.*, max((select ";
        $sql .= "count(penyakit_gejala.id_penyakit) from
penyakit_gejala ";
        $sql .= "where penyakit_gejala.id_penyakit =

```

```

penyakit.id_penyakit)) ";
    $sql .= "as j from penyakit where penyakit_status = 1
group by ";
    $sql .= "penyakit_kode order by j desc";

    $query = $this->db->query($sql);
    if ($query->num_rows() > 0) {
        $data['result'] = 'true';
        $data['data'] = $query->result();
    }

    header('Content-Type: application/json');
    echo json_encode($data);
}

public function getTest($gejalaUtama) {
    $data = array();

    $this->db->where('pg.penyakit_gejala_status', 1);
    $this->db->where('pg.id_gejala', $gejalaUtama);
    $this->db->join('gejala g', 'g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penyakit_gejala pg');
    foreach ($query->result() as $row){
        $penyakit[] = $row->id_penyakit;
    }

    $this->db->where('pg.penyakit_gejala_status', 1);
    $this->db->where_in('pg.id_penyakit', $penyakit);
    $this->db->join('gejala g', 'g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penyakit_gejala pg');
    foreach ($query->result() as $row){
        $data[$row->id_gejala] = '['.$row->gejala_kode .' ] '.$row-
>gejala_nama;
    }

    header('Content-Type: application/json');
    echo json_encode($data);
}

public function tesEntry(){
    $data = array();
    $data['result'] = 'false';

    $idPelaku = $this->input->post('pelakuId');
    $data_simpan2 ['id_pelakutes'] = $idPelaku;

    if($this->db->insert('tes', $data_simpan2)){
        $idTes = $this->db->insert_id();
        $gejala = $this->input->post('gejala');
        foreach($gejala as $k => $v){
            $data_simpan3 ['id_tes'] = $idTes;
            $data_simpan3 ['id_gejala'] = $k;
            if($this->db->insert('tesentry', $data_simpan3)){
                $data['result'] = 'true';
            }
        }
    }
}

```

```

        header('Content-Type: application/json');
        echo json_encode($data);
    }

    public function getDetail($id){
        $data = array();

        $data = array_merge($data, $this->getGejalay($id));
        $data = array_merge($data, $this->getPengobatan($id));
        $data = array_merge($data, $this->getPencegahan($id));

        header('Content-Type: application/json');
        echo json_encode($data);
    }

    public function getGejalay($id){
        $data = array();
        $data['result'] = 'false';

        $this->db->where('pg.penyakit_gejala_status', 1);
        $this->db->where('pg.id_penyakit', $id);
        $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
        $query = $this->db->get('penyakit_gejala pg');

        if($query->num_rows() > 0){
            $data['result'] = 'true';
            $data['data'] = $query->result();
        }

        return $data;
    }

    public function getPengobatan($id){
        $data = array();
        $data['resultPengobatan'] = 'false';

        $this->db->where('penyakit_pengobatan_status', 1);
        $this->db->where('id_penyakit', $id);
        $this->db->join('pengobatan','pengobatan.id_pengobatan =
penyakit_pengobatan.id_pengobatan');
        $query = $this->db->get('penyakit_pengobatan');
        if($query->num_rows() > 0){
            $data['resultPengobatan'] = 'true';
            $data['dataPengobatan'] = $query->result();
        }

        return $data;
    }

    public function getPencegahan($id){
        $data = array();
        $data['resultPencegahan'] = 'false';

        $this->db->where('penyakit_pencegahan_status', 1);
        $this->db->where('id_penyakit', $id);
        $this->db->join('pencegahan','pencegahan.id_pencegahan =
penyakit_pencegahan.id_pencegahan');
        $query = $this->db->get('penyakit_pencegahan');
    }

```

```

        if($query->num_rows() > 0){
            $data['resultPencegahan'] = 'true';
            $data['dataPencegahan'] = $query->result();
        }

        return $data;
    }

    public function getGejalaUtama(){
        $data = array();
        $hasil = array();

        $this->db->order_by('gejala_kode');
        $this->db->where('gejala_status',1);
        $q_gejala = $this->db->get('gejala')->result();
        foreach($q_gejala as $r){
            $d_gejala[] = $r->id_gejala;
            $d_gejala_array[$r->id_gejala] = '['.$r->gejala_kode.']
            '.$r->gejala_nama;
        }

        $d_penyakit = $this->get_penyakit_gejala();
        $d_gejala_utama = $this->GejalaUtama($d_gejala, $d_penyakit, 0,
        $hasil);
        foreach($d_gejala_utama as $v){
            $d_gejalanya[$v] = $d_gejala_array[$v];
        }
        asort($d_gejalanya);

        header('Content-Type: application/json');
        echo json_encode($d_gejalanya);
    }

    public function GejalaUtama($d_gejala, $d_penyakit, $rekursif,
    $d_hasil){
        foreach($d_gejala as $v){
            $ketemu = 0;
            $d_ketemu = array();
            foreach($d_penyakit as $k2 => $v2){
                if(in_array($v, $v2)){
                    $ketemu++;
                    $d_ketemu[$v][] = $k2;
                }
            }
            $d_ketemu2[] = $d_ketemu;
        }

        $jml = 0;
        foreach($d_ketemu2 as $k => $v){
            foreach($v as $k2 => $v2){
                if(count($v2) > $jml){
                    $jml = count($v2);
                    $d_ambil = $v;
                }
            }
        }

        foreach($d_ambil as $k => $v){

```



```

        foreach($d_penyakit as $k2 => $v2){
            if(in_array($k, $v2)){
                unset($d_penyakit[$k2]);
            }
        }
        $hasil = $k;
    }
    $d_hasil [$rekursif] = $hasil;
    if(count($d_penyakit) > 0){
        $rekursif++;
        return $this->GejalaUtama($d_gejala, $d_penyakit,
    $rekursif, $d_hasil);
    } else {
        return $d_hasil;
    }
}

public function get_penyakit_gejala(){
    $g = $this->input->post('gejala');

    if($g){
        $this->db->where('pg.penyakit_gejala_status', 1);
        $this->db->join('gejala g', 'g.id_gejala = pg.id_gejala');
        $this->db->join('penyakit p', 'p.id_penyakit =
pg.id_penyakit');
        $this->db->order_by('g.gejala_kode', 'ASC');
        $query = $this->db->get('penyakit_gejala pg');
        if($query->num_rows() > 0){
            foreach($query->result() as $r){
                $penyakit[$r->id_penyakit] = $r->penyakit_nama;
                $d_penyakit[$r->id_penyakit][] = $r->id_gejala;
                $d_gejala[$r->id_gejala] = $r->gejala_nama;
            }
        }
        foreach($g as $k => $v){
            $temp [] [$v] = $d_gejala[$v];
        }
        $d = array();
        foreach($d_penyakit as $k => $v){
            $jmlGejala = count($g);
            $containsSearch = count(array_intersect($v, $g));

            if($jmlGejala == $containsSearch){
                $d[$k] = $penyakit[$k];
            }
        }
        if(count($d) > 0){
            $data['result'] = 'true';
            $data['data'] = $d;
            $data['gejala'] = $temp;
        }else{
            $data['result'] = 'false';
            $data['gejala'] = $temp;
        }
    }

    header('Content-Type: application/json');
    echo json_encode($data);
} else {

```

```

        $this->db->where('penyakit_gejala_status', 1);
        $this->db->join('gejala ', 'gejala.id_gejala =
penyakit_gejala.id_gejala');
        $this->db->join('penyakit', 'penyakit.id_penyakit =
penyakit_gejala.id_penyakit');
        $this->db->where('penyakit_status', 1);
        $this->db->order_by('gejala_kode', 'ASC');
        $query = $this->db->get('penyakit_gejala');
        if($query->num_rows() > 0){
            foreach($query->result() as $r){
                $d_penyakit[$r->id_penyakit][] = $r->id_gejala;
            }
        }
        return $d_penyakit;
    }
}

public function register(){
    $data = array();
    $datax['register'] = 'false';

    $data['pelakutes_nomoridentitas'] = $this->input-
>post('identitas');
    $data['pelakutes_nama'] = $this->input->post('nama');
    $data['pelakutes_password'] = md5($this->input-
>post('password'));
    $data['pelakutes_alamat'] = $this->input->post('alamat');
    $data['pelakutes_phone'] = $this->input->post('phone');

    $insert = $this->db->insert('pelakutes', $data);
    if($insert){
        $datax['register'] = 'true';
    }

    header('Content-Type: application/json');
    echo json_encode($datax);
}

public function login(){
    $data = array();
    $data['login'] = 'false';

    $identitas = $this->input->post('identitas');
    $password = md5($this->input->post('password'));

    $this->db->where('pelakutes_nomoridentitas', $identitas);
    $this->db->where('pelakutes_status', 1);
    $this->db->where('pelakutes_password', $password);
    $query = $this->db->get('pelakutes');
    if ($query->num_rows() > 0) {
        $data['login'] = 'true';
        $data['data'][] = $query->row();
    }

    header('Content-Type: application/json');
    echo json_encode($data);
}

```

```

public function cf () {
    $data = array();
    $data['result'] = 'false';
    $data['result_pengobatan'] = 'false';
    $data['result_pencegahan'] = 'false';

    if ($input_user = $this->input->post('gejala')) {
        // bloat
        /* $input_user = array (
            21 => 0.8,
            3 => 0.5,
            55 => 0.6
        );*/
        // not found
        /* $input_user = array (
            2 => 0.7,
            16 => 0.6,
            15 => 0.5
        );*/

        $data['cf_1'] = $this->cf_1($input_user);
        $data['cf_2'] = $this->cf_2($input_user);
        $data['ds_1'] = $this->ds_1($input_user);
        $data['ds_2'] = $this->ds_2($input_user);

        // data gejala
        $data_gejala = array();
        $this->db->where('gejala_status', '1');
        $q_g = $this->db->get('gejala');
        if ($q_g->num_rows() > 0) {
            foreach($q_g->result() as $rg) {
                $data_gejala[$rg->id_gejala] = $rg;
            }
            foreach($input_user as $k => $v) {
                $tmp_g = $data_gejala[$k];
                $data['gejala'][] = '[' . $tmp_g->gejala_kode . ']' . $tmp_g->gejala_nama;
            }
        }
        #pre($data['gejala']);

        // get penyakit dari kode;
        $kode_4 = array();
        if (array_key_exists('penyakit_kode',
    $data['cf_1'])) {
                $kode_4[$data['cf_1']['penyakit_kode']] =
    $data['cf_1']['penyakit_kode'];
            }
            if (array_key_exists('penyakit_kode',
    $data['cf_2'])) {
                $kode_4[$data['cf_2']['penyakit_kode']] =
    $data['cf_2']['penyakit_kode'];
            }
            if (array_key_exists('penyakit_kode',
    $data['ds_1'])) {
                $kode_4[$data['ds_1']['penyakit_kode']] =
    $data['ds_1']['penyakit_kode'];
            }
        }
    }
}

```

```

        if (array_key_exists('penyakit_kode',
$data['ds_2'])) {
            $kode_4[$data['ds_2']['penyakit_kode']][] =
$data['ds_2']['penyakit_kode'];
        }
        // ambil hasil terbanyak
        $hit_kode = array();
        foreach ($kode_4 as $kp => $vp) {
            $hit_kode[$kp] = count($vp);
        }

        if (! empty($hit_kode)) {
            $data['result'] = 'true';
            arsort($hit_kode);
            $kode_penyakit = array_keys($hit_kode)[0];
            #pre($kode_penyakit);

            // data pengobatan
            $this->db->distinct();
            $this->db->
>where('penyakit_pengobatan_status', 1);
            $this->db->where('penyakit_status', 1);
            $this->db->where('penyakit_kode',
$kode_penyakit);
            $this->db->
>join('pengobatan','pengobatan.id_pengobatan =
penyakit_pengobatan.id_pengobatan');
            $this->db->
>join('penyakit','penyakit.id_penyakit =
penyakit_pengobatan.id_penyakit');
            $this->db->group_by('pengobatan_kode');
            $q_obat = $this->db->
>get('penyakit_pengobatan');
            if ($q_obat->num_rows() > 0) {
                foreach($q_obat->result() as $ro) {
                    $data['result_pengobatan'] =
'true';
                    $data['data_pengobatan'][] =
[' . $ro->pengobatan_kode . ' . $ro->pengobatan_nama;
                }
            }
            #pre($data['pengobatan']);

            // data pencegahan
            $this->db->distinct();
            $this->db->
>where('penyakit_pencegahan_status', 1);
            $this->db->where('penyakit_status', 1);
            $this->db->where('penyakit_kode',
$kode_penyakit);
            $this->db->
>join('pencegahan','pencegahan.id_pencegahan =
penyakit_pencegahan.id_pencegahan');
            $this->db->
>join('penyakit','penyakit.id_penyakit =
penyakit_pencegahan.id_penyakit');
            $this->db->group_by('pencegahan_kode');
            $q_cegah = $this->db->

```

```

>get('penyakit_pencegahan');
        if ($q_cegah->num_rows() > 0) {
            foreach($q_cegah->result() as $rc) {
                $data['result_pencegahan'] =
'true';
                $data['data_pencegahan'][] =
'[' . $rc->pencegahan_kode . ']' . $rc->pencegahan_nama;
            }
        }
        #pre($data['pencegahan']);
    }

    header('Content-Type: application/json');
    echo json_encode($data);
}

private function cf_1 ($input_user) {
    // input + database
    $data = array();
    $data['result'] = 'false';

    foreach($input_user as $k => $v){
        $gejala [] = $k;
        $bobot [$k] = $v;
    }

    $jumlah_gejala = count($gejala);

    $this->db->select('*', count(*) as jumlah_gejala');
    $this->db->join('penyakit p', 'p.id_penyakit=pg.id_penyakit');
    $this->db->where_in('id_gejala', $gejala);
    $this->db->where('p.penyakit_status', 1);
    $this->db->group_by('p.id_penyakit');
    $this->db->having('jumlah_gejala', $jumlah_gejala);
    $query_z = $this->db->get('penyakit_gejala pg');

    $a = array();
    foreach($query_z->result() as $r){
        $a[] = $r->id_penyakit;
    }

    if (! empty($a)) {
        $b = join(', ', $a);

        $query = "SELECT penyakit.*,
        (select count(penyakit_gejala.id_penyakit) from penyakit_gejala
where penyakit_gejala.id_penyakit = penyakit.id_penyakit) as jumlah
FROM penyakit
where penyakit.id_penyakit in ($b)
and penyakit.penyakit_status = 1
having jumlah=$jumlah_gejala";
        $query_a = $this->db->query($query);
        if($query_a->num_rows() > 0){
            $query_cocok = $query_a->row()->id_penyakit;
            $data['result'] = 'true';
            $data['penyakit'] = $query_a->row()-
>penyakit_nama;

```

```

>penyakit_cf;                                # $data['penyakit_cf'] = $query_a->row()-
                                             $kode = $query_a->row()->penyakit_kode;
                                             $data['id_penyakit'] = $query_cocok;
                                             $data['penyakit_kode'] = $kode;

                                             $kode = $query_a->row()->penyakit_kode;
                                             $this->db->where('penyakit_kode', $kode);
                                             $this->db->order_by("penyakit.id_penyakit",
"asc");
                                             $this->db-
>join('penyakit','penyakit.id_penyakit = penyakit_gejala.id_penyakit');
                                             $query_b = $this->db-
>get('penyakit_gejala');
                                             foreach($query_b->result() as $r){
>id_gejala;
                                             $rule[$r->id_penyakit][ ] = $r-
>penyakit_cf;
                                             $cf_pakar[$r->id_penyakit] = $r-
                                             }
                                             $x = CF_KOMBINASI($query_cocok, $rule,
$cf_pakar, $input_user);
                                             $data['nilai_cf'] = $x;
                                             }
                                             }
                                             return $data;
}

private function cf_2 ($input_user) {
// database
$data = array();
$data['result'] = 'false';

foreach($input_user as $k => $v){
$gejala [ ] = $k;
$bobot [ $k ] = $v;
}

$jumlah_gejala = count($gejala);

$this->db->select('*', count(*) as jumlah_gejala');
$this->db->join('penyakit p', 'p.id_penyakit=pg.id_penyakit');
$this->db->where_in('id_gejala', $gejala);
$this->db->where('p.penyakit_status', 1);
$this->db->group_by('p.id_penyakit');
$this->db->having('jumlah_gejala', $jumlah_gejala);
$query_z = $this->db->get('penyakit_gejala pg');

$a = array();
foreach($query_z->result() as $r){
$a[] = $r->id_penyakit;
}

if (!empty($a)) {
$b = join(', ', $a);

$query = "SELECT penyakit.*,

```

```

        (select count(penyakit_gejala.id_penyakit) from penyakit_gejala
where penyakit_gejala.id_penyakit = penyakit.id_penyakit) as jumlah
FROM penyakit
where penyakit.id_penyakit in ($b)
and penyakit.penyakit_status = 1
having jumlah=$jumlah_gejala";
        $query_a = $this->db->query($query);
        if($query_a->num_rows() > 0){
            $query_cocok = $query_a->row()->id_penyakit;
            $data['result'] = 'true';
            $data['penyakit'] = $query_a->row()-
>penyakit_nama;

            #$data['penyakit_cf'] = $query_a->row()-
>penyakit_cf;

            $kode = $query_a->row()->penyakit_kode;
            $data['id_penyakit'] = $query_cocok;
            $data['penyakit_kode'] = $kode;

            $tmp_cf = array();
            $this->db->where('penyakit_kode', $kode);
            $this->db->order_by("penyakit.id_penyakit",
"asc");

            $this->db-
>join('penyakit','penyakit.id_penyakit = penyakit_gejala.id_penyakit');
            $query_b = $this->db-
>get('penyakit_gejala');

            foreach($query_b->result() as $r){
                $rule[$r->id_penyakit][] = $r-
>id_gejala;
                $cf_pakar[$r->id_penyakit] = $r-
>penyakit_cf;
                $tmp_cf[] = $r->penyakit_cf;
            }
            $data['nilai_cf'] = min($tmp_cf);
        }
    }
    return $data;
}

private function ds_1 ($input_user) {
    // input
    $this->load->helper('densitas');

    $data = array();
    $data['result'] = 'false';

    $gejala = format_perhitungan($input_user);
    // ganti nilai sesuai input user
    #pre($input_user);
    #pre($gejala);
    $new_gejala = array();
    foreach($gejala as $k => $v) {
        $new_v = array();
        foreach($v as $ka => $va) {
            $new_v[$ka] = (array_key_exists($k,

```

```

$input_user) ? $input_user[$k] : $va);
    }
    $new_gejala[$k] = $new_v;
}
#pre($new_gejala);

$hitung = perhitungan_densitas($new_gejala);
if ($hitung['hasil']['penyakit'] != '') {
    if (! strpos ($hitung['hasil']['penyakit'], ',')) {
        $data['result'] = 'true';
        $data['penyakit_kode'] =
$hitung['hasil']['penyakit'];
        $data['penyakit'] = ''; // diisi di bawah;
        $data['nilai_ds'] =
$hitung['hasil']['densitas'];

        $this->db->select('penyakit_nama');
        $this->db->where('penyakit_kode',
$hitung['hasil']['penyakit']);
        $q_p = $this->db->get('penyakit');
        if ($q_p->result() > 0) {
            $data['penyakit'] = $q_p->row()-
>penyakit_nama;
        }
    }
}

return $data;
}

private function ds_2 ($input_user) {
    // database
    $this->load->helper('densitas');

    $data = array();
    $data['result'] = 'false';

    $gejala = format_perhitungan($input_user);
    $hitung = perhitungan_densitas($gejala);
    if ($hitung['hasil']['penyakit'] != '') {
        if (! strpos ($hitung['hasil']['penyakit'], ',')) {
            $data['result'] = 'true';
            $data['penyakit_kode'] =
$hitung['hasil']['penyakit'];
            $data['penyakit'] = ''; // diisi di bawah;
            $data['nilai_ds'] =
$hitung['hasil']['densitas'];

            $this->db->select('penyakit_nama');
            $this->db->where('penyakit_kode',
$hitung['hasil']['penyakit']);
            $q_p = $this->db->get('penyakit');
            if ($q_p->result() > 0) {
                $data['penyakit'] = $q_p->row()-
>penyakit_nama;
            }
        }
    }
}

```



```

        return $data;
    }
}

```

### c) gejala.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Gejala extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
     * fungsi ini digunakan untuk menampilkan form utama dari
     managemen table : gejala
     */
    public function index($aksi='tambahdata', $id='') {
        $data = array();
        $data['aksi'] = $aksi;

        /*
         * load all reference data dari [publicmodel]
         * ./application/models/publicmodel.php
         */

        /* validasi data */
        $this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
        $this->form_validation->set_rules('f_gejala_kode', 'gejala
kode', 'required');
        $this->form_validation->set_rules('f_gejala_nama', 'gejala
nama', 'required');

```

```

/* jika $action = ubahdata dan $id tidak kosong */
if ($id != '' ){
    $this->db->where('id_gejala',$id);
    $query_gejala = $this->db->get('gejala');
    $record_gejala = $query_gejala->row();
    $data['record_gejala'] = $record_gejala;
}

/*
 * jika tombol simpan di tekan
 * bisa digunakan untuk proses INPUT dan UPDATE
 */
$b_simpan = $this->input->post('b_simpan');
if ( $b_simpan ){
    if ($this->form_validation->run() !== FALSE) {

        /* siapkan data dari form untuk dimasukkan dalam
table */
        $data_gejala['gejala_kode'] = $this->input-
>post('f_gejala_kode');
        $data_gejala['gejala_nama'] = $this->input-
>post('f_gejala_nama');

        /*
         * proses memasukan data (simpan)
         * jika ada $_POST['f_id_gejala'] maka yang
dilakukan adalah update
         */
        $id_gejala = $this->input-
>post('f_id_gejala');

        if ( $id_gejala ){
            $this->db->where('id_gejala', $this-
>input->post('f_id_gejala'));

            $this->db->update('gejala',
$data_gejala);

```

```

                                $this->session->set_flashdata('pesan',
'Data berhasil di-Perbarui');
                                }else{
                                $this->db-
>insert('gejala',$data_gejala);
                                $this->session->set_flashdata('pesan',
'Data berhasil di-Simpan');
                                }
                                $url = base_url().'gejala/data';

                                }
                                }

                                /* proses hapus, dipisahkan dari proses UPDATE dan INSERT
*/
                                $b_hapus = $this->input->post('b_hapus');
                                if ( $b_hapus ){
                                $data_gejala['gejala_status'] = 0; // 1:active
0:delete
                                $this->db->where('id_gejala', $this->input-
>post('f_id_gejala'));
                                $this->db->update('gejala', $data_gejala);
                                $this->session->set_flashdata('pesan', 'Data
berhasil di-Hapus');
                                $url = base_url().'gejala/data';
                                }
                                /* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
                                if( isset($url) ){
                                redirect($url);
                                }

                                $this->load->view('gejala_index',$data);
                                }

                                /*
                                * fungsi ini digunakan untuk menampilkan data table : gejala

```

```

*/
public function data($halaman=0){
    $data = array();

    /*
    * paging halaman data
    * ambil jumlah data keseluruhan
    */
    $data['no'] = $halaman;
    $datatiaphalaman = 30;
    $this->db->where('gejala_status',1);
    $config['base_url'] = base_url().'gejala/data/';
    $config['total_rows'] = $this->db->count_all_results('gejala');
    $config['per_page'] = $datatiaphalaman;
    $this->pagination->initialize($config);

    /* ambil data yang ingin ditampilkan sebanyak
    $datatiaphalaman */
    $this->db->where('gejala_status',1);
    $this->db->limit($datatiaphalaman,$halaman);
    $query_gejala = $this->db->get('gejala');
    $data['gejala'] = $query_gejala;

    /* tampilkan view */
    $this->load->view('gejala_data',$data);
}
/*
* fungsi melihat detail dari 1 record terpilih
*/
public function details($id){
    $data = array();

    $this->db->where('id_gejala',$id);
    $query_gejala = $this->db->get('gejala');
    $data['gejala'] = $query_gejala;

```

```
        /* tampilkan view */
        $this->load->view('gejala_detail',$data);
    }

    /*
    * fungsi ini digunakan untuk update data table : gejala
    */
    public function updatestatus($id,$status){
        $data_gejala['gejala_status'] = $status; // 1:active
0:delete
        $this->db->where('id_gejala', $id);
        $this->db->update('gejala', $data_gejala);
        switch ($status) {
            case 0:
                $pesan = "Hapus";
                break;
            /*
            case 1:
                $pesan = "sesuaikan dengan status yang lain";
                break;
            */
        }
        $this->session->set_flashdata('pesan', 'Data berhasil di-
'.'. $pesan);
        $url = base_url().'gejala/data';
        redirect($url);
    }
}
```

## d) hewan.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Hewan extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
     * fungsi ini digunakan untuk menampilkan form utama dari
managemen table : hewan
     */
    public function index($aksi='tambahdata', $sid='') {
        $data = array();
        $data['aksi'] = $aksi;

        /* validasi data */
        $this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
        $this->form_validation->set_rules('f_hewan_nama', 'hewan
nama', 'required');

        /* jika $action = ubahdata dan $sid tidak kosong */
        if ($sid!= '' ) {
            $this->db->where('id_hewan',$sid);
            $query_hewan = $this->db->get('hewan');
            $record_hewan = $query_hewan->row();
            $data['record_hewan'] = $record_hewan;
        }

        /*

```

```

* jika tombol simpan di tekan
* bisa digunakan untuk proses INPUT dan UPDATE
*/
$b_simpan = $this->input->post('b_simpan');
if ( $b_simpan ){
    if ($this->form_validation->run() !== FALSE) {

        /* siapkan data dari form untuk dimasukkan dalam
table */

        $data_hewan['hewan_nama'] = $this->input-
>post('f_hewan_nama');
        $data_hewan['hewan_gambar'] = $this->input-
>post('f_hewan_gambar');

        if(isset($_FILES['f_hewan_gambar']['name']) &&
$_FILES['f_hewan_gambar']['name'] != ''){
            $files = $_FILES['f_hewan_gambar'];

            $namafile = md5(date('Ymdhis'));

            move_uploaded_file($files['tmp_name'], "./images/hewan/".$namafile
.'_'.$files['name']);

            $data_hewan['hewan_gambar'] =
$namafile.'_'.$files['name'];
        }

        /*
        * proses memasukan data (simpan)
        * jika ada $_POST['f_id_hewan'] maka yang
dilakukan adalah update
        */
        $id_hewan = $this->input->post('f_id_hewan');
        if ( $id_hewan ){
            $this->db->where('id_hewan', $this-
>input->post('f_id_hewan'));

            $this->db->update('hewan',

```

```

$data_hewan);
                                $this->session->set_flashdata('pesan',
'Data berhasil di-Perbarui');
                                }else{
                                $this->db-
>insert('hewan',$data_hewan);
                                $this->session->set_flashdata('pesan',
'Data berhasil di-Simpan');
                                }
                                $url = base_url().'hewan/data';
                                }
                                }

/* proses hapus, dipisahkan dari proses UPDATE dan INSERT
*/
$b_hapus = $this->input->post('b_hapus');
if ( $b_hapus ){
    $data_hewan['hewan_status'] = 0; // 1:active
0:delete
    $this->db->where('id_hewan', $this->input-
>post('f_id_hewan'));
    $this->db->update('hewan', $data_hewan);
    $this->session->set_flashdata('pesan', 'Data
berhasil di-Hapus');
    $url = base_url().'hewan/data';
}

/* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
if( isset($url) ){
    redirect($url);
}

$this->load->view('hewan_index',$data);
}

/*

```



```

    * fungsi ini digunakan untuk menampilkan data table : hewan
    */
    public function data($halaman=0){
        $data = array();
        /*
        * paging halaman data
        * ambil jumlah data keseluruhan
        */
        $data['no'] = $halaman;
        $datatiaphalaman = 30;
        $this->db->where('hewan_status',1);
        $config['base_url'] = base_url(). 'hewan/data/';
        $config['total_rows'] = $this->db->count_all_results('hewan');
        $config['per_page'] = $datatiaphalaman;
        $this->pagination->initialize($config);

        /* ambil data yang ingin ditampilkan sebanyak
        $datatiaphalaman */
        $this->db->where('hewan_status',1);
        $this->db->limit($datatiaphalaman,$halaman);
        $query_hewan = $this->db->get('hewan');
        $data['hewan'] = $query_hewan;

        /* tampilkan view */
        $this->load->view('hewan_data',$data);
    }
    /*
    * fungsi melihat detail dari 1 record terpilih
    */
    public function details($id){
        $data = array();

        $this->db->where('id_hewan',$id);
        $query_hewan = $this->db->get('hewan');
        $data['hewan'] = $query_hewan;
    }

```

```
        /* tampilkan view */
        $this->load->view('hewan_detail',$data);
    }

    /*
    * fungsi ini digunakan untuk update data table : hewan
    */
    public function updatestatus($id,$status){
        $data_hewan['hewan_status'] = $status; // 1:active
0:delete

        $this->db->where('id_hewan', $id);
        $this->db->update('hewan', $data_hewan);
        switch ($status) {
            case 0:
                $pesan = "Hapus";
                break;

            /*
            case 1:
                $pesan = "sesuaikan dengan status yang lain";
                break;
            */
        }
        $this->session->set_flashdata('pesan', 'Data berhasil di-
'.'. $pesan);
        $url = base_url().'hewan/data';
        redirect($url);
    }
}
```

## e) pelakutes.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Pelakutes extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
     * fungsi ini digunakan untuk menampilkan form utama dari
    managemen table : jenisgejala
     */
    public function index($aksi='tambahdata'){
        //pre($this->session->userdata('gejala'));
        $data = array();
        $data['aksi'] = $aksi;
        /* validasi data */
        //$this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
        //$this->form_validation->set_rules('f_no_identitasx', 'No
Identitas', 'required');
        //$this->form_validation->set_rules('f_nama', 'Nama',
'required');
        //$this->form_validation->set_rules('f_alamat', 'Alamat',
'required');
        //$this->form_validation->set_rules('f_phone', 'Telepon',
'required');

        /* proses pencarian identitas pelaku tes */
        $b_cari = $this->input->post('cari');
        if($b_cari){
            $this->db->where('pelakutes_nomoridentitas', $this-
>input->post('f_no_identitas'));
            $query_pelakutes = $this->db->get('pelakutes');
            $data['record_pelakutes'] = $query_pelakutes->row();
        }
    }
}

```

```

        //pre($query_pelakutes->row());
        //untuk mencari data gejala//
        $gejala_id = $this->session->userdata('gejala');
        $gejala = $this->p->gejalaArrayById($gejala_id);
        //pre($penyakit);
        $data['record_gejala'] = $gejala;
    }

    /*
    * jika tombol simpan di tekan
    * bisa digunakan untuk proses INPUT dan UPDATE
    */
    $b_simpan = $this->input->post('b_simpan');
    if ( $b_simpan ){
        //untuk mencari data gejala//
        $gejala_id = $this->session->userdata('gejala');
        $gejala = $this->p->gejalaArrayById($gejala_id);
        $data['record_gejala'] = $gejala;
        //if ($this->form_validation->run() !== FALSE) {
            //simpan data di tabel tes
            //cek dulu apakah id sudah ada belum
            $id_pelaku = $this->input-
>post('f_id_pelakutes');
            if(!empty($id_pelaku)){
                $id_pelakutes = $id_pelaku;
            }else{
                /* siapkan data dari form untuk
dimasukan dalam table */
                $data_simpan = array(

                    'pelakutes_nomoridentitas' => $this->input-
>post('f_no_identitasx'),
                    'pelakutes_nama' =>
                    $this->input->post('f_nama'),
                    'pelakutes_alamat' =>
                    $this->input->post('f_alamat'),

```

```

'pelakutes_phone' =>
$this->input->post('f_phone'),
    );
    $this->db->
>insert('pelakutes',$data_simpan);
    $id_pelakutes = $this->db->insert_id();
    }

    //simpan data tes
    //pre($id_pelakutes);
    $tanggal = date('Y-m-d H:i:s');
    $data_simpan_tes = array(
        'tes_tanggal' => $tanggal,
        'id_pelakutes' => $id_pelakutes
    );
    $this->db->insert('tes',$data_simpan_tes);

    //simpan data tesentry
    $id_tes = $this->db->insert_id();
    foreach ($gejala as $key => $v){
        $data_simpan_tesentry = array(
            'id_tes' => $id_tes,
            'id_gejala' => $key
        );
        $this->db->
>insert('tesentry',$data_simpan_tesentry);
    }
    $url = base_url().'welcome';
    $this->session->set_flashdata('pesan', 'Data
berhasil di Simpan');

    $this->session->unset_userdata('gejala');

    //}
}

/* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
if( isset($url) ){

```

```

        redirect($url);
    }

    $this->load->view('pelakutes_index',$data);
}

/*
 * fungsi ini digunakan untuk menampilkan data table : tesentry
 */
public function data($halaman=0){
    $data = array();

    /*
     * paging halaman data
     * ambil jumlah data keseluruhan
     */
    $data['no'] = $halaman;
    $datatiaphalaman = 30;
    $this->db->where('tes_status',1);
    $config['base_url'] = base_url().'pelakutes/data/';
    $config['total_rows'] = $this->db->count_all_results('tes');
    $config['per_page'] = $datatiaphalaman;
    $this->pagination->initialize($config);

    /* ambil data yang ingin ditampilkan sebanyak
    $datatiaphalaman */
    $this->db->join('pelakutes','tes.id_pelakutes=pelakutes.id_pelakutes');
    $this->db->where('tes_status',1);
    $this->db->limit($datatiaphalaman,$halaman);
    $this->db->order_by('id_tes DESC');
    $query_tesentry = $this->db->get('tes');
    $data['tesentry'] = $query_tesentry;

    /* tampilkan view */
    $this->load->view('pelakutes_data',$data);
}

```

```
}

public function detail($id_tes='') {
    $data = array();
    /* ambil data yang ingin ditampilkan sebanyak
$datatiaphalaman */

    $this->db->join('tes','tes.id_tes=tesentry.id_tes');
    $this->db-
>join('gejala','gejala.id_gejala=tesentry.id_gejala');
    $this->db->where('tesentry.id_tes',$id_tes);
    $this->db->where('tes_status',1);
    //$this->db->order_by('id_tesentry');
    $query_tesentry = $this->db->get('tesentry');
    $data['tesentry'] = $query_tesentry;
    //pre($query_tesentry );

    //untuk data tanggal tes dan nama pelaku tes
    $this->db-
>join('pelakutes','pelakutes.id_pelakutes=tes.id_pelakutes');
    $this->db->where('tes.id_tes',$id_tes);
    $this->db->where('tes_status',1);
    $query_pelaku = $this->db->get('tes');
    $data['pelakutes'] = $query_pelaku->row();

    /* tampilkan view */
    $this->load->view('pelakutes_detail',$data);
}

}
```

## f) pencegahan.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Pencegahan extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
     * fungsi ini digunakan untuk menampilkan form utama dari
    managemen table : pencegahan
    */
    public function index($aksi='tambahdata', $sid='') {
        $data = array();
        $data['aksi'] = $aksi;

        /* validasi data */
        $this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
        $this->form_validation->set_rules('f_pencegahan_kode',
'pencegahan kode', 'required');
        $this->form_validation->set_rules('f_pencegahan_nama',
'pencegahan nama', 'required');

        /* jika $action = ubahdata dan $id tidak kosong */
        if ($sid != '' ) {
            $this->db->where('id_pencegahan',$sid);
            $query_pencegahan = $this->db->get('pencegahan');
            $record_pencegahan = $query_pencegahan->row();
            $data['record_pencegahan'] = $record_pencegahan;
        }
    }
}

```



```

/*
 * jika tombol simpan di tekan
 * bisa digunakan untuk proses INPUT dan UPDATE
 */
$b_simpan = $this->input->post('b_simpan');
if ( $b_simpan ){
    if ($this->form_validation->run() !== FALSE) {

        /* siapkan data dari form untuk dimasukkan dalam
table */
        $data_pencegahan['pencegahan_kode'] = $this->input-
>post('f_pencegahan_kode');
        $data_pencegahan['pencegahan_nama'] = $this->input-
>post('f_pencegahan_nama');

        /*
        * proses memasukan data (simpan)
        * jika ada $_POST['f_id_pencegahan'] maka
yang dilakukan adalah update
        */
        $id_pencegahan = $this->input-
>post('f_id_pencegahan');
        if ( $id_pencegahan ){
            $this->db->where('id_pencegahan',
$this->input->post('f_id_pencegahan'));
            $this->db->update('pencegahan',
$data_pencegahan);
            $this->session->set_flashdata('pesan',
'Data berhasil di-Perbarui');
        }else{
            $this->db-
>insert('pencegahan',$data_pencegahan);
            $this->session->set_flashdata('pesan',
'Data berhasil di-Simpan');
        }
        $url = base_url().'pencegahan/data';

```

```

    }
}

/* proses hapus, dipisahkan dari proses UPDATE dan INSERT
*/
$b_hapus = $this->input->post('b_hapus');
if ( $b_hapus ){
    $data_pencegahan['pencegahan_status'] = 0; //
1:active 0:delete
    $this->db->where('id_pencegahan', $this->input-
>post('f_id_pencegahan'));
    $this->db->update('pencegahan', $data_pencegahan);
    $this->session->set_flashdata('pesan', 'Data
berhasil di-Hapus');
    $url = base_url().'pencegahan/data';
}

/* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
if( isset($url) ){
    redirect($url);
}

$this->load->view('pencegahan_index',$data);
}

/*
* fungsi ini digunakan untuk menampilkan data table : pencegahan
*/
public function data($halaman=0){
    $data = array();

    /*
    * paging halaman data
    * ambil jumlah data keseluruhan
    */
    $data['no'] = $halaman;
    $datatiaphalaman = 30;

```

```

        $this->db->where('pencegahan_status',1);
        $config['base_url'] = base_url(). 'pencegahan/data/';
        $config['total_rows'] = $this->db-
>count_all_results('pencegahan');
        $config['per_page'] = $datatiaphalaman;
        $this->pagination->initialize($config);

        /* ambil data yang ingin ditampilkan sebanyak
$datatiaphalaman */
        $this->db->where('pencegahan_status',1);
        $this->db->limit($datatiaphalaman,$halaman);
        $query_pencegahan = $this->db->get('pencegahan');
        $data['pencegahan'] = $query_pencegahan;

        /* tampilkan view */
        $this->load->view('pencegahan_data',$data);
    }

    /*
    * fungsi melihat detail dari 1 record terpilih
    */
    public function details($id){
        $data = array();

        $this->db->where('id_pencegahan',$id);
        $query_pencegahan = $this->db->get('pencegahan');
        $data['pencegahan'] = $query_pencegahan;

        /* tampilkan view */
        $this->load->view('pencegahan_detail',$data);
    }

    /*
    * fungsi ini digunakan untuk update data table : pencegahan
    */
    public function updatestatus($id,$status){
        $data_pencegahan['pencegahan_status'] = $status; //

```

```

1:active 0:delete
    $this->db->where('id_pencegahan', $id);
    $this->db->update('pencegahan', $data_pencegahan);
    switch ($status) {
        case 0:
            $pesan = "Hapus";
            break;
        /*
        case 1:
            $pesan = "sesuaikan dengan status yang lain";
            break;
        */
    }
    $this->session->set_flashdata('pesan', 'Data berhasil di-
'. $pesan);
    $url = base_url(). 'pencegahan/data';
    redirect($url);
}
}

```

#### g) pengobatan.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Pengobatan extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
    * fungsi ini digunakan untuk menampilkan form utama dari
    managemen table : pengobatan
    */
}

```

```

public function index($aksi='tambahdata', $sid='') {
    $data = array();
    $data['aksi'] = $aksi;

    /* validasi data */
    $this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
    $this->form_validation->set_rules('f_pengobatan_kode',
'pengobatan kode', 'required');
    $this->form_validation->set_rules('f_pengobatan_nama',
'pengobatan nama', 'required');

    /* jika $action = ubahdata dan $id tidak kosong */
    if ($sid != '' ) {
        $this->db->where('id_pengobatan',$sid);
        $query_pengobatan = $this->db->get('pengobatan');
        $record_pengobatan = $query_pengobatan->row();
        $data['record_pengobatan'] = $record_pengobatan;
    }

    /*
    * jika tombol simpan di tekan
    * bisa digunakan untuk proses INPUT dan UPDATE
    */
    $b_simpan = $this->input->post('b_simpan');
    if ( $b_simpan ) {
        if ($this->form_validation->run() !== FALSE) {

            /* siapkan data dari form untuk dimasukkan dalam
table */

            $data_pengobatan['pengobatan_kode'] = $this->input-
>post('f_pengobatan_kode');
            $data_pengobatan['pengobatan_nama'] = $this->input-
>post('f_pengobatan_nama');

```

```

        /*
        * proses memasukan data (simpan)
        * jika ada $_POST['f_id_pengobatan'] maka
yang dilakukan adalah update
        */
        $id_pengobatan = $this->input-
>post('f_id_pengobatan');
        if ( $id_pengobatan ){
            $this->db->where('id_pengobatan',
$this->input->post('f_id_pengobatan'));
            $this->db->update('pengobatan',
$this->session->set_flashdata('pesan',
'Data berhasil di-Perbarui');
        }else{
            $this->db-
>insert('pengobatan',$data_pengobatan);
            $this->session->set_flashdata('pesan',
'Data berhasil di-Simpan');
        }
        $url = base_url().'pengobatan/data';
    }
}

/* proses hapus, dipisahkan dari proses UPDATE dan INSERT
*/
    $b_hapus = $this->input->post('b_hapus');
    if ( $b_hapus ){
        $data_pengobatan['pengobatan_status'] = 0; //
1:active 0:delete
        $this->db->where('id_pengobatan', $this->input-
>post('f_id_pengobatan'));
        $this->db->update('pengobatan', $data_pengobatan);
        $this->session->set_flashdata('pesan', 'Data
berhasil di-Hapus');
        $url = base_url().'pengobatan/data';
    }
}

```

```

    }

    /* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
    if( isset($url) ){
        redirect($url);
    }

    $this->load->view('pengobatan_index',$data);
}

/*
 * fungsi ini digunakan untuk menampilkan data table : pengobatan
 */
public function data($halaman=0){
    $data = array();

    /*
     * paging halaman data
     * ambil jumlah data keseluruhan
     */
    $data['no'] = $halaman;
    $datatiaphalaman = 30;
    $this->db->where('pengobatan_status',1);
    $config['base_url'] = base_url().'pengobatan/data/';
    $config['total_rows'] = $this->db->
>count_all_results('pengobatan');
    $config['per_page'] = $datatiaphalaman;
    $this->pagination->initialize($config);

    /* ambil data yang ingin ditampilkan sebanyak
$datatiaphalaman */
    $this->db->where('pengobatan_status',1);
    $this->db->limit($datatiaphalaman,$halaman);
    $query_pengobatan = $this->db->get('pengobatan');
    $data['pengobatan'] = $query_pengobatan;

```

```

        /* tampilkan view */
        $this->load->view('pengobatan_data',$data);
    }

    /*
    * fungsi melihat detail dari 1 record terpilih
    */
    public function details($id){
        $data = array();

        $this->db->where('id_pengobatan',$id);
        $query_pengobatan = $this->db->get('pengobatan');
        $data['pengobatan'] = $query_pengobatan;

        /* tampilkan view */
        $this->load->view('pengobatan_detail',$data);
    }

    /*
    * fungsi ini digunakan untuk update data table : pengobatan
    */
    public function updatestatus($id,$status){
        $data_pengobatan['pengobatan_status'] = $status; //
1:active 0:delete
        $this->db->where('id_pengobatan', $id);
        $this->db->update('pengobatan', $data_pengobatan);
        switch ($status) {
            case 0:
                $pesan = "Hapus";
                break;
            /*
            case 1:
                $pesan = "sesuaikan dengan status yang lain";
                break;
            */
        }
        $this->session->set_flashdata('pesan', 'Data berhasil di-
' . $pesan);
    }

```



```

        $url = base_url().'pengobatan/data';
        redirect($url);
    }
}

```

## h) penyakit.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct
script access allowed');

class Penyakit extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->model('publicmodel','p');
    }

    /*
    * fungsi ini digunakan untuk menampilkan form utama dari
    managemen table : penyakit
    */
    public function index($aksi='tambahdata', $id='') {
        $data = array();
        $data['aksi'] = $aksi;

        $rs_gejala = $this->p->gejalaArray();
        $data['gejalaArray'] = $rs_gejala;
        $data['pengobatanArray'] = $this->p->pengobatanArray();
        $data['pencegahanArray'] = $this->p->pencegahanArray();

        /* validasi data */
        $this->form_validation->set_error_delimiters('<br /><span
class="error">', '</span>');
        $this->form_validation->set_rules('f_penyakit_kode',
'penyakit kode', 'required');
        $this->form_validation->set_rules('f_penyakit_nama',
'penyakit nama', 'required');
        $this->form_validation->set_rules('f_penyakit_cf', 'penyakit
Nilai CF', 'required');
        $this->form_validation->set_rules('f_penyakit_namalatin',
'penyakit namalatin', 'required');

        /* jika $action = ubahdata dan $id tidak kosong */
        if ($id != '' ) {
            $this->db->where('id_penyakit',$id);
            $query_penyakit = $this->db->get('penyakit');
            $record_penyakit = $query_penyakit->row();
            $data['record_penyakit'] = $record_penyakit;
        }
    }
}

```

```

        $this->db->where('id_penyakit',$id);
        $query_penyakit_gejala = $this->db-
>get('penyakit_gejala');
        $record_penyakit_gejala = $query_penyakit_gejala;
        $data['record_penyakit_gejala'] =
$record_penyakit_gejala;

        foreach ($record_penyakit_gejala->result() as $r){
            $data['array_db_penyakit_gejala'][$r-
>id_gejala] = $r->id_gejala;
            // $data['array_db_penyakit_gejala_mb'][$r-
>id_gejala] = $r->penyakit_gejala_mb;
            // $data['array_db_penyakit_gejala_md'][$r-
>id_gejala] = $r->penyakit_gejala_md;
        }
        //pre($data['array_db_penyakit_gejala']);

        $this->db->where('id_penyakit',$id);
        $query_penyakit_pengobatan = $this->db-
>get('penyakit_pengobatan');
        $record_penyakit_pengobatan =
$query_penyakit_pengobatan;
        $data['record_penyakit_pengobatan'] =
$record_penyakit_pengobatan;
        foreach ($record_penyakit_pengobatan->result() as
$r){
            $data['array_db_penyakit_pengobatan'][$r-
>id_pengobatan] = $r->id_pengobatan;
        }

        $this->db->where('id_penyakit',$id);
        $query_penyakit_pencegahan = $this->db-
>get('penyakit_pencegahan');
        $record_penyakit_pencegahan =
$query_penyakit_pencegahan;
        foreach ($record_penyakit_pencegahan->result() as
$r){
            $data['array_db_penyakit_pencegahan'][$r-
>id_pencegahan] = $r->id_pencegahan;
        }

        /*$this->db->where('id_penyakit',$id);
        $query_penyakit_hewan = $this->db-
>get('hewan_penyakit');
        #pre($this->db->last_query());
        $record_penyakit_hewan = $query_penyakit_hewan;
        foreach ($record_penyakit_hewan->result() as $r){
            $data['array_db_penyakit_hewan'][$r-
>id_hewan] = $r->id_hewan;
        }*/

    }

    /*
    * jika tombol simpan di tekan
    * bisa digunakan untuk proses INPUT dan UPDATE

```

```

        */
        $b_simpan = $this->input->post('b_simpan');
        if ( $b_simpan ) {
            if ( $this->form_validation->run() !== FALSE ) {

                /*if(isset($_FILES['file']['name']) &&
                $_FILES['file']['name'] != '') {
                    $files = $_FILES['file'];

                    $namafile = md5(date('Ymdhis'));

                    move_uploaded_file($files['tmp_name'], "./images/penyakit/".$namaf
                    ile.'_'.$files['name']);
                    $data_penakit['penyakit_gambar'] =
                    $namafile.'_'.$files['name'];
                }*/

                /* siapkan data dari form untuk dimasukkan dalam
                table */
                $data_penakit['penyakit_kode'] = $this->input-
                >post('f_penakit_kode');
                $data_penakit['penyakit_nama'] = $this->input-
                >post('f_penakit_nama');
                $data_penakit['penyakit_cf'] = $this->input-
                >post('f_penakit_cf');
                $data_penakit['penyakit_namalatin'] = $this->input-
                >post('f_penakit_namalatin');

                /*
                * proses memasukan data (simpan)
                * jika ada $_POST['f_id_penakit'] maka yang
                dilakukan adalah update
                */
                $id_penakit = $this->input-
                >post('f_id_penakit');
                if ( $id_penakit ) {

                    /* penyakit gejala */
                    $this->db-
                >where('id_penakit',$id_penakit);
                    $this->db->delete('penyakit_gejala');
                    /* penyakit pengobatan */
                    $this->db-
                >where('id_penakit',$id_penakit);
                    $this->db-
                >delete('penyakit_pengobatan');

                    /* penyakit pencegahan */
                    $this->db-
                >where('id_penakit',$id_penakit);
                    $this->db-
                >delete('penyakit_pencegahan');

                    /* penyakit hewan */
                    // $this->db-
                >where('id_penakit',$id_penakit);
                    // $this->db-
                >delete('hewan_penakit');
            }
        }
    }
}

```

```

    $this->db->where('id_penyakit', $this->input->post('f_id_penyakit'));
    $this->db->update('penyakit',
    $data_penyakit);

    foreach ($this->input->post('gejala')
    as $k=>$v) {

        $data_penyakit_gejala['id_penyakit'] = $id_penyakit;

        $data_penyakit_gejala['id_gejala'] = $k;
        $mb = $this->input->post('mb');
        $md = $this->input->post('md');

        //$data_penyakit_gejala['penyakit_gejala_mb'] = $mb[$k];

        //$data_penyakit_gejala['penyakit_gejala_md'] = $md[$k];
        $this->db->
    >insert('penyakit_gejala',$data_penyakit_gejala);
        //pre($data_penyakit_gejala);
    }

    foreach ($this->input->
    >post('pencegahan') as $k=>$v) {

        $data_penyakit_pencegahan['id_penyakit'] = $id_penyakit;

        $data_penyakit_pencegahan['id_pencegahan'] = $k;
        $this->db->
    >insert('penyakit_pencegahan',$data_penyakit_pencegahan);
        //pre($data_penyakit_gejala);
    }

    foreach ($this->input->
    >post('pengobatan') as $k=>$v){

        $data_penyakit_pengobatan['id_penyakit'] = $id_penyakit;

        $data_penyakit_pengobatan['id_pengobatan'] = $k;
        $this->db->
    >insert('penyakit_pengobatan',$data_penyakit_pengobatan);
        //pre($data_penyakit_pengobatan);
    }

    $this->session->set_flashdata('pesan',
    'Data berhasil di-Perbarui');
    }else{
        $this->db->
    >insert('penyakit',$data_penyakit);
        $id = $this->db->insert_id();

        foreach ($this->input->post('gejala')
    as $k=>$v) {

            $data_penyakit_gejala['id_penyakit'] = $id;

```

```

        $data_penyakit_gejala['id_gejala'] = $k;
        $mb = $this->input->post('mb');
        $md = $this->input->post('md');

        //$data_penyakit_gejala['penyakit_gejala_mb'] = $mb[$k];
        //$data_penyakit_gejala['penyakit_gejala_md'] = $md[$k];
        $this->db-
>insert('penyakit_gejala',$data_penyakit_gejala);

        //pre($data_penyakit_gejala);
    }

    foreach ($this->input-
>post('pencegahan') as $k=>$v){
        $data_penyakit_pencegahan['id_penyakit'] = $id;
        $data_penyakit_pencegahan['id_pencegahan'] = $k;
        $this->db-
>insert('penyakit_pencegahan',$data_penyakit_pencegahan);
        //pre($data_penyakit_gejala);
    }

    foreach ($this->input-
>post('pengobatan') as $k=>$v){
        $data_penyakit_pengobatan['id_penyakit'] = $id;
        $data_penyakit_pengobatan['id_pengobatan'] = $k;
        $this->db-
>insert('penyakit_pengobatan',$data_penyakit_pengobatan);
        //pre($data_penyakit_pengobatan);
    }

    $this->session->set_flashdata('pesan',
'Data berhasil di-Simpan');
    }
    $url = base_url().'penyakit/data';
    }
}

// $b_hapus = $this->input->post('b_hapus');
// if ( $b_hapus ){
//     $data_penyakit['penyakit_status'] = 0; // 1:active
0:delete
//     $this->db->where('id_penyakit', $this->input-
>post('f_id_penyakit'));
//     $this->db->update('penyakit', $data_penyakit);
//     $this->session->set_flashdata('pesan', 'Data
berhasil di-Hapus');
//     $url = base_url().'penyakit/data';
// }

```

```

/* jika terdapat data URL, maka arahkan halaman ke lokasi
yang bersangkutan */
    if( isset($url) ){
        redirect($url);
    }

    $this->load->view('penyakit_index',$data);
}

/*
 * fungsi ini digunakan untuk menampilkan data table : penyakit
 */
public function data($halaman=0){
    $data = array();

    /*
     * paging halaman data
     * ambil jumlah data keseluruhan
     */
    $data['no'] = $halaman;
    $datatiaphalaman = 30;
    $this->db->where('penyakit_status',1);
    $this->db->order_by('penyakit_nama');
    $config['base_url'] = base_url().'penyakit/data/';
    $config['total_rows'] = $this->db-
>count_all_results('penyakit');
    $config['per_page'] = $datatiaphalaman;
    $this->pagination->initialize($config);

    /* ambil data yang ingin ditampilkan sebanyak
    $datatiaphalaman */
    $this->db->where('penyakit_status',1);
    $this->db->limit($datatiaphalaman,$halaman);
    $this->db->order_by('penyakit_nama');
    $query_penyakit = $this->db->get('penyakit');
    $data['penyakit'] = $query_penyakit;

    /* tampilkan view */
    $this->load->view('penyakit_data',$data);
}

/*
 * fungsi ini digunakan untuk update data table : penyakit
 */
public function updatestatus($id,$status){
    $data_penyakit['penyakit_status'] = $status; // 1:active
0:delete

    $this->db->where('id_penyakit', $id);
    $this->db->update('penyakit', $data_penyakit);
    switch ($status) {
        case 0:
            $pesan = "Hapus";
            break;

        /*
        case 1:
            $pesan = "sesuaikan dengan status yang lain";
            break;

```

```
        */
    }
    $this->session->set_flashdata('pesan', 'Data berhasil di-
'.$pesan);
    $url = base_url().'penyakit/data';
    redirect($url);
}

}
```

©UKDW

**Android xrb.kelinci**

## a) AboutActivity.java

```
package xrb.kelinci;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;

public class AboutActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
it is present.
        getMenuInflater().inflate(R.menu.about, menu);
        return true;
    }
}
```

## b) AccountActivity.java

```
package xrb.kelinci;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class AccountActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account);
    }
}
```



```

    }

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        // Inflate the menu; this adds items to the action bar if
it is present.

        getMenuInflater().inflate(R.menu.account, menu);

        return true;

    }

}

```

## c) BobotActivity.java

```

package xrb.kelinci;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Set;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.Penyakit;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.InputType;
import android.util.Log;

```

```

import android.view.View;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.HorizontalScrollView;
import android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class BobotActivity extends Activity {

    public static ArrayList<Penyakit> data;
    public static ArrayList<NameValuePair> nameValuePair;
    private JSONObject jsonRespon;
    private LinearLayout llPelakuTes;
    private HorizontalScrollView hsView;
    private TextView tvJudul;
    private Button btnCari;
    public static HashMap<String, String> dataTes;
    public static HashMap<String, String> dataPenyakit;
    public static HashMap<String, String> dataGejala;
    public TableLayout tabel;
    private List<EditText> listPenyakitGejala;
    private Button btnCekPenyakit;
    private String gejalaUtama;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bobot);

        setupView();
        gejalaUtama = getIntent().getStringExtra("gejalaUtama");

        if (TesActivity.dataTes.size() > 0) {
            tvJudul.setVisibility(View.VISIBLE);
            tvJudul.setText("Silahkan masukkan Nilai Bobot :
");

            dataTes = new HashMap<String, String>();

            TableRow tr = new TableRow(getBaseContext());
            TextView tv = new TextView(getBaseContext());
            tv.setText(" Gejala");
            tv.setTextColor(Color.BLACK);
            tv.setPadding(10, 0, 0, 10);
            tr.addView(tv);

```

```

        TextView tv2 = new TextView(getBaseContext());
        tv2.setText("Nilai MB");
        tv2.setTextColor(Color.BLACK);
        tv2.setPadding(10, 0, 0, 10);
        tr.addView(tv2);

        tabel.addView(tr);

        listPenyakitGejala = new ArrayList<EditText>();
        for (Map.Entry<String, String> entryGejala :
TesActivity.dataTes
                .entrySet()) {
            TableRow trGejala = new
TableRow(getBaseContext());
            TextView tvGejala = new
TextView(getBaseContext());
            tvGejala.setText(entryGejala.getValue());
            tvGejala.setTextColor(Color.BLACK);
            tvGejala.setPadding(10, 3, 10, 3);
            trGejala.addView(tvGejala);
            EditText txtGejala = new
EditText(getBaseContext());
            String tag =
entryGejala.getKey().toString();

            Log.i("gp", tag);
            txtGejala.setTag(tag);

            txtGejala.setInputType(InputType.TYPE_NUMBER_FLAG_DECIMAL
|
InputType.TYPE_CLASS_NUMBER);

            txtGejala.setImeOptions(EditorInfo.IME_ACTION_NEXT);
            tvGejala.setTextColor(Color.BLACK);
            tvGejala.setHintTextColor(Color.BLACK);
            trGejala.addView(txtGejala);
            String gejalaUtamax = "gejala[" +
gejalaUtama + "]";

            if(gejalaUtamax.equalsIgnoreCase(tag)){
                txtGejala.setText("1");
            }
            listPenyakitGejala.add(txtGejala);

            tabel.addView(trGejala);

        }

        btnCekPenyakit.setText("Cek Penyakit");
        btnCekPenyakit.setPadding(5, 5, 5, 5);
        TableRow trTombol = new

```

```

TableRow(getContext());
        trTombol.addView(btnCekPenyakit);
        tabel.addView(trTombol);

    } else {
        RbHelper.pesan(getContext(),
            "Data Tes penyakit Tidak Boleh
kosong.");
    }

    btnCekPenyakit.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // cek setiap edittext sudah diisi belum
            for (EditText et : listPenyakitGejala) {
                if (RbHelper.isEmpty(et)) {

                    RbHelper.pesan(getContext(),
                        "Semua bobot
harus diisi!");
                    return;
                }
                double value =
Double.parseDouble(et.getText().toString());
                if (value > 1) {

                    RbHelper.pesan(getContext(),
                        "Bobot nilainya
tidak boleh lebih dari 1!");
                    return;
                }
                if (value < 0) {

                    RbHelper.pesan(getContext(),
                        "Bobot nilainya
tidak boleh negatif!");
                    return;
                }
            }

            // kirim data keserver
            // format data gp[idpenyakit][idgejala] =
nilai bobot
            dataTes = new HashMap<String, String>();
            for (EditText et : listPenyakitGejala) {
                dataTes.put(et.getTag().toString(),
et.getText().toString());
                Log.i(et.getTag().toString(),

```

```

et.getText().toString());
    }

    startActivity(new Intent(getBaseContext(),
        TesHasilActivity.class));

    });
}

private void setupView() {
    hsView = (HorizontalScrollView)
findViewById(R.id.horizontalScrollLBobot);
    tvJudul = (TextView) findViewById(R.id.tvBobotJudul);
    tvJudul.setVisibility(View.GONE);
    llPelakuTes = (LinearLayout)
findViewById(R.id.LLPelakuTesBobot);
    llPelakuTes.setVisibility(View.GONE);
    btnCari = (Button) findViewById(R.id.btnBobotCari);
    btnCari.setVisibility(View.GONE);
    tabel = (TableLayout) findViewById(R.id.tbLLayoutBobot);
    btnCari.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            Intent i = new Intent(getBaseContext(),
PelakuTesActivity.class);
            MainActivity.tesEntry = true;
            startActivity(i);
        }
    });

    btnCekPenyakit = new Button(getBaseContext());
}

private class GetDataAsyncTask extends AsyncTask<String, Void,
String> {
    JSONParser jParser;

    public GetDataAsyncTask() {
        jParser = new JSONParser();
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        jParser.loadingIn(BobotActivity.this);
    }
}

```



```

        dataPenyakit.put(kunci, x.getString(kunci));
    }
}

for (Map.Entry<String, String>
e : dataPenyakit
        .entrySet()) {
    new TextView(getBaseContext());
    TextView tvPenyakit =
        new TextView(getBaseContext());
        tvPenyakit.setText(e.getValue());
        tvPenyakit.setPadding(10, 0, 10, 0);
        tvPenyakit.setTextColor(Color.BLACK);
        tr.addView(tvPenyakit);
    }
    tabel.addView(tr);
    JSONArray jGejala =
    jsonRespon.getJSONArray("gejala");
    for (int i = 0; i <
jGejala.length(); i++) {
        jGejala.getJSONObject(i);
        JSONObject x =
        (String) key.next();
        Iterator key = x.keys();
        while (key.hasNext()) {
            dataGejala.put(kunci, x.getString(kunci));
            String kunci =
        }
    }
    listPenyakitGejala = new
    ArrayList<EditText>();
    for (Map.Entry<String, String>
    entryGejala : dataGejala
        .entrySet()) {
        TableRow trGejala = new
        TableRow(getBaseContext());
        TextView tvGejala = new
        TextView(getBaseContext());
        tvGejala.setText(entryGejala.getValue());
    }
}

```

```

        tvGejala.setTextColor(Color.BLACK);
3, 10, 3);
        tvGejala.setPadding(10,
        trGejala.addView(tvGejala);
String> entryPenyakit : dataPenyakit
        .entrySet()) {
        EditText
txtGejala = new EditText(
        getContext());
        String tag =
"gp["
        entryPenyakit.getKey().toString()
        "]"
        entryGejala.getKey().toString() + "]";
        Log.i("penyakit
        gejala ", "penyakit "
        entryPenyakit.getKey().toString()
        gejala "
        entryGejala.getKey().toString());
        Log.i("gp", tag);
        txtGejala
        .setInputType(InputType.TYPE_NUMBER_FLAG_DECIMAL
        | InputType.TYPE_CLASS_NUMBER);
        txtGejala
        .setImeOptions(EditorInfo.IME_ACTION_NEXT);
        tvGejala.setTextColor(Color.BLACK);
        tvGejala.setHintTextColor(Color.BLACK);
        trGejala.addView(txtGejala);
        listPenyakitGejala.add(txtGejala);
    }

```





```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_help);  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if  
it is present.  
    getMenuInflater().inflate(R.menu.help, menu);  
    return true;  
}  
}
```

## e) GejalaUtamaActivity.java

```
package xrb.kelinci;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Iterator;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import xrb.kelinci.helper.JSONParser;  
import xrb.kelinci.helper.Penyakit;  
import xrb.kelinci.helper.RbHelper;  
import android.app.Activity;
```

```
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;

public class GejalaUtamaActivity extends Activity {

    public static ArrayList<Penyakit> data;
    private JSONObject jsonRespon;
    private LinearLayout layoutLinear;
    private Button btnNext;
    private RadioGroup rgGejala;
    private String gejalaUtama = "0";
```

```
@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_gejala_utama);

    setupView();

    if (!isOnline()) {

        alertMessageNoInternet();

        btnNext.setVisibility(View.GONE);

    } else {

        btnNext.setVisibility(View.VISIBLE);

        String url = "";

        url = MainActivity.URL_SERVICE + "getGejalaUtama/";

        Log.i("url", url);

        new GetDataAsyncTask().execute(url);

    }

}

private void setupView() {

    LinearLayout = (LinearLayout)

findViewById(R.id.lineartesgejala);

    btnNext = (Button) findViewById(R.id.btnGejalaNext);

    btnNext.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View arg0) {

            //RbHelper.pesan(getContext(),

gejalaUtama);
```

```
        if (!gejalaUtama.equalsIgnoreCase("0")) {  
            Intent i = new Intent(getApplicationContext(),  
TesActivity.class);  
            i.putExtra("gejalaUtama", gejalaUtama);  
            startActivity(i);  
        } else {  
            RbHelper.pesan(getApplicationContext(),  
                "Silahkan pilih Gejala  
Penyakit Utama");  
        }  
    }  
});  
rgGejala = (RadioGroup) findViewById(R.id.rgGejala);  
}  
private class GetDataAsyncTask extends AsyncTask<String, Void,  
String> {  
    JSONParser jParser;  
  
    public GetDataAsyncTask() {  
        jParser = new JSONParser();  
    }  
  
    @Override  
    protected void onPreExecute() {
```

```

        super.onPreExecute();

        jParser.loadingIn(GejalaUtamaActivity.this);

    }

    @Override
    protected String doInBackground(String... params) {
        String url = params[0], getData = "true";
        jsonRespon = jParser.sendDataax(url);

        return getData;
    }

    @Override
    protected void onPostExecute(String result) {
        jParser.loadingOut();

        Object value = jsonRespon;
        Iterator<String> iter2 = ((JSONObject)
value).keys();

        while (iter2.hasNext()) {
            String key2 = iter2.next();
            // Log.i("key2", key2);
            try {
                String value2 = ((JSONObject)
value).getString(key2);
                RadioButton (getBaseContext());
                rb.setText(value2);
                rb.setTag(key2);
                rb.setTextColor(Color.BLACK);
                // rb.setBackgroundColor(Color.WHITE);
                rb.setOnClickListener(new
OnClickListener() {

                    @Override
                    public void onClick(View arg0) {
                        gejalaUtama =
rb.getTag().toString();

                    }

                });
                rgGejala.addView(rb);
            } catch (JSONException e) {
                e.printStackTrace();
            }

        }

    }

}

// untuk check koneksi internet
public boolean isOnline() {

```

```

        ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo netInfo = cm.getActiveNetworkInfo();
        if (netInfo != null && netInfo.isConnectedOrConnecting()) {
            return true;
        }
        return false;
    }

    // peringatan jika internet tidak konek
    private void alertMessageNoInternet() {
        final AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        builder.setMessage(
            "Anda tidak terkoneksi dengan internet,
Silahkan Aktifkan Internet Anda terlebih dahulu.")
            .setCancelable(false)
            .setTitle("Informasi Internet")
            .setNegativeButton("Tutup",
                new
DialogInterface.OnClickListener() {
                    public void onClick(final
DialogInterface dialog,
                        @SuppressWarnings("unused") final int id) {
                            dialog.cancel();
                        }
                });
        final AlertDialog alert = builder.create();
        alert.show();
    }
}

```

#### f) InfoActivity.java

```

package xrb.kelinci;

import android.app.TabActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;

public class InfoActivity extends TabActivity {
    public static TabHost tabHost;
    public static int tabAktif = 0;

    public static String idHp;
    public static int kategori;
}

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_info);

    idHp = getIntent().getStringExtra("hpId");
    kategori = getIntent().getIntExtra("kategori", 1);

    tabHost = getTabHost();
    //
    TabSpec settingSpec = tabHost.newTabSpec("Bantuan");
    settingSpec.setIndicator("Bantuan");
    settingSpec.setContent(new Intent(getBaseContext(),
        HelpActivity.class));

    //
    TabSpec transaksiSpec = tabHost.newTabSpec("About");
    transaksiSpec.setIndicator("About");
    transaksiSpec.setContent(new Intent(getBaseContext(),
        AboutActivity.class));

    tabHost.addTab(settingSpec);
    tabHost.addTab(transaksiSpec);
}
}
```



## g) LoginActivity.java

```
package xrb.kelinci;

import java.util.ArrayList;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.PelakuTes;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class LoginActivity extends Activity {

    private EditText txtIdentitas, txtPassword;
    private ArrayList<NameValuePair> dataLogin;
    private Button bLogin, bRegister;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        setupView();

        if (MainActivity.isLogin) {
            bRegister.setVisibility(View.GONE);
            txtIdentitas.setVisibility(View.GONE);
            txtPassword.setVisibility(View.GONE);
        }
    }
}
```

```

        bLogin.setText("Logout");
    } else {
        txtIdentitas.setVisibility(View.VISIBLE);
        txtPassword.setVisibility(View.VISIBLE);
        bLogin.setText("Login");
    }
}

private void setupView() {
    txtIdentitas = (EditText)
findViewById(R.id.txtLoginIdentitas);
    txtPassword = (EditText)
findViewById(R.id.txtLoginPassword);
    bLogin = (Button) findViewById(R.id.btnLoginSubmit);
    bRegister = (Button) findViewById(R.id.btnLoginRegister);
}

public void submit(View v) {
    switch (v.getId()) {
        case R.id.btnLoginRegister:
            Intent i = new Intent(getApplicationContext(),
RegisterActivity.class);
            startActivity(i);
            break;
        case R.id.btnLoginSubmit:
            if (MainActivity.isLogin) {
                MainActivity.isLogin = false;
                RbHelper.pesan(getApplicationContext(), "Anda Sudah
Logout.");
                Intent ix = new Intent(getApplicationContext(),
MainActivity.class);
                startActivity(ix);
            } else {
                if (RbHelper.isEmpty(txtIdentitas)) {
                    RbHelper.pesan(getApplicationContext(),
"Username Harus Diisi");

```

```

        } else if (RbHelper.isEmpty(txtPassword)) {
            RbHelper.pesan(getContext(),
"Password Harus Diisi");
        } else {
            String url = MainActivity.URL_SERVICE +
"login";

            // Log.i("tombol kirim", "diklik");
            dataLogin = new
ArrayList<NameValuePair>(2);

            try {

                dataLogin.add(new
BasicNameValuePair("identitasa",
                txtIdentitas.getText().toString()));
                dataLogin.add(new
BasicNameValuePair("password",
                txtPassword.getText().toString()));

            } catch (Exception e) {
            }

            new LoginAsyncTask().execute(url);
        }
    }
    break;
}

}

private class LoginAsyncTask extends AsyncTask<String, Void,
String> {
    JSONParser jParser;
    JSONObject jsonRespon;

```

```
public LoginAsyncTask() {
    jParser = new JSONParser();
}

@Override
protected void onPreExecute() {
    super.onPreExecute();
    jParser.loadingIn(LoginActivity.this);
}

@Override
protected String doInBackground(String... params) {
    String url = params[0];
    String resultx = "false";
    jsonRespon = jParser.sendDatax(dataLogin, url);
    try {
        resultx = jsonRespon.getString("login");
        if (resultx.equalsIgnoreCase("true")) {
            MainActivity.isLogin = true;
            JSONArray jArray =
jsonRespon.getJSONArray("data");

            // definisikan arraylist jurusan
            for (int i = 0; i < jArray.length();
i++) {
                JSONObject jobject =
jArray.getJSONObject(i);

                MainActivity.pelaku = new
PelakuTes();

                MainActivity.pelaku.setId(jobject
.getString("id_pelakutes"));

                MainActivity.pelaku.setNama(jobject
```

```

        .getString("pelakutestes_nama"));

        MainActivity.pelaku.setNoIdentitas(jObject

        .getString("pelaku_nomoridentitas"));

        MainActivity.pelaku.setPassword(jObject

        .getString("pelakutes_password"));

        MainActivity.pelaku.setAlamat(jObject

        .getString("pelakutes_alamat"));

        MainActivity.pelaku.setTelp(jObject

        .getString("pelakutes_phone"));
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
return resultx;
}

@Override
protected void onPostExecute(String result) {
    jParser.loadingOut();
    if (result.equalsIgnoreCase("true")) {
        if (MainActivity.tesEntry) {
            jParser.pesan(
                "Login berhasil, si;ahkan
mengirim tes Entry Anda",
                getBaseContext());
            Intent i = new Intent(getBaseContext(),
                PelakuTesActivity.class);
            MainActivity.tesEntry = false;

```



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

public void mainMenu(View v) {
    final Intent i;
    switch (v.getId()) {
        case R.id.ivMainPenyakit:
            i = new Intent(getApplicationContext(),
PenyakitActivity.class);
            startActivity(i);
            break;
        case R.id.ivMainPakar:
            i = new Intent(getApplicationContext(),
GejalaUtamaActivity.class);
            startActivity(i);
            break;
        case R.id.ivMainAccount:
            i = new Intent(getApplicationContext(),
LoginActivity.class);
            startActivity(i);
            break;
        case R.id.ivMainInfo:
            i = new Intent(getApplicationContext(),
InfoActivity.class);
            startActivity(i);
            break;
    }
}
}
```

## i) PelakuTesActivity.java

```
package xrb.kelinci;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.PelakuTes;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;

public class PelakuTesActivity extends Activity {

    private ArrayList<NameValuePair> nameValuePairs;
    private JSONObject jsonRespon;
    private LinearLayout llPelakuTes;
    private Button btnKirim;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```



```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_pelaku_tes);

setupView();

if (MainActivity.isLogin) {
    if (TesActivity.dataTes.size() > 0) {

        Set set =
TesActivity.dataPenyakit.entrySet();
        Iterator i = set.iterator();
        int no = 0;
        while (i.hasNext()) {
            no++;
            Map.Entry me = (Map.Entry) i.next();
            TextView tv = new
TextView(getBaseContext());
            tv.setText(no + ". " +
me.getValue().toString());
            tv.setTextColor(Color.BLACK);
            llPelakuTes.addView(tv);
        }
    }
} else {
    RbHelper.pesan(getBaseContext(),
        "Silahkan Anda Login dulu untuk
mengakses halaman ini.");
    Intent i = new Intent(getBaseContext(),
LoginActivity.class);
    startActivity(i);
}
}
```

```

private void setupView() {
    llPelakuTes = (LinearLayout) findViewById(R.id.llGejala);

    btnKirim = (Button) findViewById(R.id.btnPelakuTesKirim);
    btnKirim.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            if (TesActivity.dataTes.size() > 0) {
                nameValuePairs = new
ArrayList<NameValuePair>();

                // Get a set of the entries
                Set set =
TesActivity.dataTes.entrySet();

                // Get an iterator
                Iterator i = set.iterator();
                // Display elements
                while (i.hasNext()) {
                    Map.Entry me = (Map.Entry)
i.next();

                    try {
                        nameValuePairs.add(new
BasicNameValuePair(me
.getKey().toString(), me.getValue()
.toString()));

                    } catch (Exception e) {
                        e.printStackTrace();
                    }

                    nameValuePairs.add(new
BasicNameValuePair("pelakuId",
MainActivity.pelaku.getId()));

```

```
String url = MainActivity.URL_SERVICE +
"tesentry";

Log.i("url", url);
new GetDataAsyncTask().execute(url);

    }

    }

});

}

private class GetDataAsyncTask extends AsyncTask<String, Void,
String> {

    JSONParser jParser;

    public GetDataAsyncTask() {
        jParser = new JSONParser();
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        jParser.loadingIn(PelakuTesActivity.this);
    }

    @Override
    protected String doInBackground(String... params) {
        String url = params[0], getData = "true";
        Log.i("valuepairs", nameValuePairs.toString());
        jsonRespon = jParser.sendData(nameValuePairs, url);
        Log.i("json", jsonRespon.toString());

        try {
            getData = jsonRespon.getString("result");

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

```
        return getData;
    }

    @Override
    protected void onPostExecute(String result) {
        jParser.loadingOut();

        if (result.equals("true")) {
            RbHelper.pesan(
                getBaseContext(),
                "Kirim Data ke pakar Berhasil,
silahkan tunggu respon dari pakar untuk gejala yang anda inputkan");
            Intent i = new Intent(getBaseContext(),
                GejalaUtamaActivity.class);
            startActivity(i);
        } else {
            RbHelper.pesan(getBaseContext(),
                "Kirim Data ke pakar Gagal,
silahkan coba kembali.");
        }
    }
}
```

## j) PenyakitActivity.java

```
package xrb.kelinci;

import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.Penyakit;
import xrb.kelinci.helper.RbHelper;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class PenyakitActivity extends Activity {
    private ListView lvList;
    public static ArrayList<Penyakit> data;

    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_penyakit);
    }
}
```

```

        setupView();

        String url = MainActivity.URL_SERVICE + "getPenyakit/";
        //Log.i("url", url);
        new GetDataAsyncTask().execute(url);
    }

    private void setupView() {
        lvList = (ListView) findViewById(R.id.lvListPenyakit);
        lvList.setOnItemClickListener(new OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> arg0, View v,
int posisi,
                                long id) {
                Intent i = new Intent(getApplicationContext(),
                                        PenyakitDetailActivity.class);
                i.putExtra("posisiData", posisi);
                startActivity(i);
            }
        });
    }

    private class GetDataAsyncTask extends AsyncTask<String, Void,
String> {
        JSONParser jParser;
        JSONObject jsonRespon;

        public GetDataAsyncTask() {
            jParser = new JSONParser();
        }

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            jParser.loadingIn(PenyakitActivity.this);
        }
    }
}

```

```
    }

    @Override
    protected String doInBackground(String... params) {
        String url = params[0], getData = "false";
        jsonRespon = jParser.sendDataX(url);

        try {
            getData = jsonRespon.getString("result");
            data = new ArrayList<Penyakit>();

            JSONArray jsonArray =
jsonRespon.getJSONArray("data");
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject ar =
jsonArray.getJSONObject(i);
                Penyakit x = new Penyakit();
                try {

                    x.setId(ar.getString("id_penyakit"));

                    x.setNama(ar.getString("penyakit_nama"));

                    x.setNamaLatin(ar.getString("penyakit_namalatin"));

                    x.setKode(ar.getString("penyakit_kode"));

                    data.add(x);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        } catch (JSONException e) {
```

```

        e.printStackTrace();
    }
    return getData;
}

@Override
protected void onPostExecute(String result) {
    jParser.loadingOut();
    // RbHelper.pesan(getBaseContext(), "hasil data");
    if (result.equals("true")) {
        CustomAdapter adapter = new
CustomAdapter(getBaseContext(),
                data);
        lvList.setAdapter(adapter);
    } else {
        RbHelper.pesan(getBaseContext(), "data belum
tersedia");
    }
}

}

// subclass untuk custom adapter pada listview
private class CustomAdapter extends BaseAdapter {
    private Context context;
    private ArrayList<Penyakit> dataz;
    private LayoutInflater inflater = null;

    public CustomAdapter(Context c, ArrayList<Penyakit> data) {
        context = c;
        dataz = data;
        inflater = (LayoutInflater) context

.getService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {

```



```

        return dataz.size();
    }

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View view, ViewGroup
parent) {
        View vi = view;
        if (view == null)
            vi =
inflater.inflate(R.layout.custom_adapter_listview, null);
        TextView judul = (TextView) vi
            .findViewById(R.id.tvCustomAdapterLvJudul);
        TextView title = (TextView) vi
            .findViewById(R.id.tvCustomAdapterLvTitle);
        TextView desc = (TextView) vi
            .findViewById(R.id.tvCustomAdapaterLvDesc);
        //ImageView gambar = (ImageView) vi
            //.findViewById(R.id.ivCustomLvAdapter);

        Penyakit x = dataz.get(position);

        judul.setText(x.getNama() + " (" + x.getNamaLatin()
+ ")");
    }

```

```
        title.setVisibility(View.GONE);
        desc.setText("Kode " + x.getKode());
        //gambar.setVisibility(View.GONE);
        return vi;
    }
}
}
```

### k) PenyakitDetailActivity.java

```
package xrb.kelinci;

import java.util.ArrayList;
import java.util.Iterator;

import org.apache.http.NameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.Gejala;
import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.Pencegahan;
import xrb.kelinci.helper.Pengobatan;
import xrb.kelinci.helper.Penyakit;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;

import com.koushikdutta.urlimageviewhelper.UrlImageViewHelper;

public class PenyakitDetailActivity extends Activity {
    public static ArrayList<Gejala> data;
    public static ArrayList<Pengobatan> dataPengobatan;
    public static ArrayList<Pencegahan> dataPencegahan;
    private TextView tvJudul;
    private ImageView ivCover;
    private Penyakit penyakit;
    private String getData = "false", getDataPengobatan = "false",
        getDataPencegahan = "false";
    private LinearLayout llGejala, llPengobatan, llPencegahan;

    private ArrayList<NameValuePair> dataCari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_penyakit_detail);
        setupView();

        int posisi = getIntent().getIntExtra("posisiData", 1);

        penyakit = PenyakitActivity.data.get(posisi);
        tvJudul.setText("Penyakit " + penyakit.getNama() + " ("
            + penyakit.getNamaLatin() + ")");
        ivCover.setVisibility(View.GONE);
    }
}
```

```
        String url = MainActivity.URL_SERVICE + "getDetail/" +
penyakit.getId();
        Log.i("url", url);
        new GetDataAsyncTask().execute(url);
    }

    private void setupView() {

        tvJudul = (TextView) findViewById(R.id.tvDetailKasusJudul);
        ivCover = (ImageView) findViewById(R.id.ivDetailGb);

        llGejala = (LinearLayout) findViewById(R.id.linearGejala);
        llPengobatan = (LinearLayout)
findViewById(R.id.linearPengobatan);
        llPencegahan = (LinearLayout)
findViewById(R.id.linearPencegahan);
    }

    private class GetDataAsyncTask extends AsyncTask<String, Void,
String> {
        JSONParser jParser;
        JSONObject jsonRespon;

        public GetDataAsyncTask() {
            jParser = new JSONParser();
        }

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            jParser.loadingIn(PenyakitDetailActivity.this);
        }

        @Override
        protected String doInBackground(String... params) {
            String url = params[0];
            jsonRespon = jParser.sendDataX(url);
        }
    }
}
```

```
try {

    // for data gejala
    getData = jsonRespon.getString("result");
    if (getData.equalsIgnoreCase("true")) {
        JSONArray jArray =
jsonRespon.getJSONArray("data");
        data = new ArrayList<Gejala>();
        for (int i = 0; i < jArray.length();
i++) {
            JSONObject ar =
jArray.getJSONObject(i);
            Gejala x = new Gejala();
            try {

                x.setGejalaId(ar.getString("id_gejala"));

                x.setGejalaNama(ar.getString("gejala_nama"));

                x.setGejalaKode(ar.getString("gejala_kode"));

                data.add(x);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        // for data pengobatan
        getDataPengobatan =
jsonRespon.getString("resultPengobatan");
        if
(getDataPengobatan.equalsIgnoreCase("true")) {
            JSONArray jArray = jsonRespon

                .getJSONArray("dataPengobatan");
```

```

dataPengobatan = new
ArrayList<Pengobatan> ();
    for (int i = 0; i < jsonArray.length();
i++) {
        JSONObject ar =
jArray.getJSONObject(i);
        Pengobatan x = new Pengobatan();
        try {
            x.setNama(ar.getString("pengobatan_nama"));
            x.setKode(ar.getString("pengobatan_kode"));
            dataPengobatan.add(x);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
// for data pencegahan
getDataPencegahan =
jsonRespon.getString("resultPencegahan");
    if
(getDataPencegahan.equalsIgnoreCase("true")) {
        JSONArray jArray = jsonRespon
        .getJSONArray("dataPencegahan");
        dataPencegahan = new
ArrayList<Pencegahan> ();
        for (int i = 0; i < jArray.length();
i++) {
            JSONObject ar =
jArray.getJSONObject(i);
            Pencegahan x = new Pencegahan();
            try {

```



```

llGejala.addView(tv);

    }
    } else {
        RbHelper.pesan(getBaseContext(), "data Gejala
belum tersedia");
    }

    if (getDataPencegahan.equalsIgnoreCase("true")) {
        int no = 0;
        for (Iterator iterator =
dataPencegahan.iterator(); iterator
        .hasNext();) {
            Pencegahan x = (Pencegahan)
iterator.next();
            no++;
            TextView tv = new
TextView(getBaseContext());
            tv.setText(no + ". " + "[" +
x.getKode() + "]" +
                + x.getNama());
            tv.setPadding(15, 0, 0, 0);
            tv.setTextColor(Color.BLACK);
            llPencegahan.addView(tv);
        }
    } else {
        RbHelper.pesan(getBaseContext(),
            "data Pencegahan belum
tersedia");
    }

    if (getDataPengobatan.equalsIgnoreCase("true")) {
        int no = 0;
        for (Iterator iterator =
dataPengobatan.iterator(); iterator
        .hasNext();) {

```





```
import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class RegisterActivity extends Activity {

    private EditText txtNama, txtIdentitas, txtPassword,
txtRepassword, txtAlamat,
        txtPhone;
    private ArrayList<NameValuePair> dataRegister;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        setupView();
    }

    private void setupView() {
        txtNama = (EditText) findViewById(R.id.txtRegisterNama);
        txtIdentitas = (EditText)
findViewById(R.id.txtRegisterNoIdentitas);
        txtAlamat = (EditText)
findViewById(R.id.txtRegisterAlamat);
        txtPhone = (EditText) findViewById(R.id.txtRegisterPhone);
        txtPassword = (EditText)
findViewById(R.id.txtRegisterPassword);
        txtRepassword = (EditText)
findViewById(R.id.txtRegisterRePassword);

    }

    public void submit(View v) {
        switch (v.getId()) {
```

```

        case R.id.btnRegisterSubmit:
            if (RbHelper.isCompare(txtPassword, txtRepassword))
            {
                RbHelper.pesan(getBaseContext(),
                    "Ulangi password Harus Diisi dan
harus sama dengan password");
                txtRepassword.setText("");
            } else if (RbHelper.isEmpty(txtNama)) {
                RbHelper.pesan(getBaseContext(), "Nama Harus
diisi");
            } else if (RbHelper.isEmpty(txtIdentitas)) {
                RbHelper.pesan(getBaseContext(), "Username
Harus Diisi");
            } else if (RbHelper.isEmpty(txtPassword)) {
                RbHelper.pesan(getBaseContext(), "Password
Harus Diisi");
            } else if (RbHelper.isEmpty(txtRepassword)) {
                RbHelper.pesan(getBaseContext(), "Ulangi
Password Harus Diisi");
            } else if (RbHelper.isEmpty(txtAlamat)) {
                RbHelper.pesan(getBaseContext(), "Alamat
Harus Diisi");
            } else if (RbHelper.isEmpty(txtPhone)) {
                RbHelper.pesan(getBaseContext(), "Phone Asal
Harus Diisi");
            } else {
                // RbHelper.pesan(getBaseContext(), "Sudah
diisi semua");
                String url = MainActivity.URL_SERVICE +
"register";
                // Log.i("tombol kirim", "diklik");
                dataRegister = new
ArrayList<NameValuePair>(5);
                try {
                    dataRegister.add(new
BasicNameValuePair("identitas", txtIdentitas
                    .getText().toString()));

```

```

        dataRegister.add(new
BasicNameValuePair("password",
        txtPassword.getText().toString());
        dataRegister.add(new
BasicNameValuePair("nama", txtNama
        .getText().toString());
        dataRegister.add(new
BasicNameValuePair("alamat", txtAlamat
        .getText().toString());
        dataRegister.add(new
BasicNameValuePair("phone", txtPhone
        .getText().toString());
    } catch (Exception e) {
        e.printStackTrace();
    }
    new registerAsyncTask().execute(url);
}
break;
}
}

private class registerAsyncTask extends AsyncTask<String, Void,
String> {
    JSONParser jParser;
    JSONObject jsonRespon;

    public registerAsyncTask() {
        jParser = new JSONParser();
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        jParser.loadingIn(RegisterActivity.this);

```

```
}

@Override
protected String doInBackground(String... params) {
    String url = params[0];
    String resultx = "false";
    jsonRespon = jParser.sendDataax(dataRegister, url);
    try {
        resultx = jsonRespon.getString("register");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return resultx;
}

@Override
protected void onPostExecute(String result) {
    jParser.loadingOut();
    if (result.equalsIgnoreCase("true")) {
        jParser.pesan(
            "Register berhasil, \nSilhakan
Anda Login dengan akun baru Anda",
            RegisterActivity.this);

        // pindah ke halaman main menu
        Intent i = new Intent(RegisterActivity.this,
            LoginActivity.class);
        startActivity(i);

    } else if (result.equalsIgnoreCase("false")) {
        jParser.pesan("Register gagal, Silahkan coba
lagi...",
            RegisterActivity.this);
    }
}
}
}
```

```
}
```

## m) SplashActivity.java

```
package xrb.kelinci;

import android.app.Activity;
import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;

public class SplashActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_splash);
        // Generates a Handler to launch the About Screen
        // after 2 seconds
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable()
        {
            public void run()
            {
                // Starts the About Screen Activity
                startActivity(new Intent(getApplicationContext(),
                    MainActivity.class));
            }
        }, 3000L);
    }

    public void onConfigurationChanged(Configuration newConfig)
    {
        // Manages auto rotation for the Splash Screen Layout
    }
}
```

```
        super.onConfigurationChanged(newConfig);
        setContentView(R.layout.activity_splash);
    }
}
```

## n) TesActivity.java

```
package xrb.kelinci;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;

import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.Penyakit;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.LinearLayout;
import android.widget.TextView;
```

```

public class TesActivity extends Activity {
    public static ArrayList<Penyakit> data;
    private JSONObject jsonRespon;
    private LinearLayout layoutLinear;
    public static HashMap<String, String> dataTes;
    public static HashMap<String, String> dataPenyakit;
    private Button btnTesCek, btnTesClear;
    private ArrayList<CheckBox> dataCb;
    private String gejalaUtama;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tes);

        setupView();
        if (!isOnline()) {
            alertMessageNoInternet();
            btnTesCek.setVisibility(View.GONE);
            btnTesClear.setVisibility(View.INVISIBLE);
        } else {
            btnTesCek.setVisibility(View.VISIBLE);
            String url = "";
            gejalaUtama =
getIntent().getStringExtra("gejalaUtama");
            url = MainActivity.URL_SERVICE + "getTest/" +
gejalaUtama;

            Log.i("url", url);
            new GetDataAsyncTask().execute(url);
        }
    }

    private void setupView() {
        layoutLinear = (LinearLayout) findViewById(R.id.lineartes);
        btnTesCek = (Button) findViewById(R.id.btnTestCekPenyakit);
        btnTesCek.setOnClickListener(new View.OnClickListener() {

            @Override

```



```

        public void onClick(View arg0) {
            if (dataTes.size() > 0) {
                Intent i = new Intent(getApplicationContext(),
BobotActivity.class);

                i.putExtra("gejalaUtama", gejalaUtama);
                startActivity(i);

            } else {
                RbHelper.pesan(getApplicationContext(),
                    "Silahkan pilih penyakit
dahulu.");
            }
        }
    });

    btnTesClear = (Button) findViewById(R.id.btnTestCekClear);
    btnTesClear.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            if (dataCb.size() > 0) {
                for (Iterator iterator =
dataCb.iterator(); iterator
                    .hasNext();) {
                        CheckBox cb = (CheckBox)
iterator.next();

                        cb.setChecked(false);
                    }
            }
        }
    });
}

private class GetDataAsyncTask extends AsyncTask<String, Void,

```

```

String> {
    JSONParser jParser;

    public GetDataAsyncTask() {
        jParser = new JSONParser();
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        jParser.loadingIn(TesActivity.this);
    }

    @Override
    protected String doInBackground(String... params) {
        String url = params[0], getData = "true";
        jsonRespon = jParser.sendDataax(url);

        return getData;
    }

    @Override
    protected void onPostExecute(String result) {
        jParser.loadingOut();

        // RbHelper.pesan(getBaseContext(), "hasil data");
        dataTes = new HashMap<String, String>();
        // tambahkan data gejala utama"
        // dataTes.put("gejala[" + gejalaUtama + "]",
gejalaUtama);

        Iterator<String> iter = jsonRespon.keys();
        dataCb = new ArrayList<CheckBox>();
        dataPenyakit = new HashMap<String, String>();
        btnTesClear.setVisibility(View.VISIBLE);

        Object value = jsonRespon;
        Iterator<String> iter2 = ((JSONObject)

```

```

value).keys();
        while (iter2.hasNext()) {
            String key2 = iter2.next();
            // Log.i("key2", key2);
            try {
                String value2 = ((JSONObject)
value).getString(key2);
                if (key2.equalsIgnoreCase(gejalaUtama))
{
                    dataTes.put("gejala[" + key2 +
"]", value2);
                    dataPenyakit.put("gejala[" +
key2 + "]", value2);
                } else {
                    final CheckBox cb = new
CheckBox(getContext());
                    cb.setText(value2);
                    cb.setTag(key2);
                    cb.setTextColor(Color.BLACK);
                    //
cb.setBackgroundColor(Color.WHITE);

                    cb.setOnCheckedChangeListener(new OnCheckedChangeListener() {

                        @Override
                        public void
onCheckedChanged(
                            CompoundButton buttonView, boolean isChecked) {
                            String value =
cb.getText().toString();
                            String idGejala =
cb.getTag().toString();
                            String idTes =
"gejala[" + idGejala + "];

                            if (isChecked) {

```

```
//
RbHelper.pesan (getBaseContext (),
//
cb.getTag ().toString ());
//
RbHelper.pesan (getBaseContext (),
//
idPenyakit);

    dataTes.put (idTes, value);

    dataPenyakit.put (idTes, cb.getText ()
        .toString ());
        } else {
            if
(dataTes.size () > 0) {
                if
(dataTes.containsKey (idTes)) {
                    dataTes.remove (idTes);
                }
            }
            if
(dataPenyakit.containsKey (idTes)) {
                dataPenyakit.remove (idTes);
            }
        }
    }
}

});

layoutLinear.addView (cb);
dataCb.add (cb);
}
```

```

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }

}

// untuk check koneksi internet
public boolean isOnline() {
    ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo netInfo = cm.getActiveNetworkInfo();
    if (netInfo != null && netInfo.isConnectedOrConnecting()) {
        return true;
    }
    return false;
}

// peringatan jika internet tidak konek
private void alertMessageNoInternet() {
    final AlertDialog.Builder builder = new
AlertDialog.Builder(this);
    builder.setMessage(
        "Anda tidak terkoneksi dengan internet,
Silahkan Aktifkan Internet Anda terlebih dahulu.")
        .setCancelable(false)
        .setTitle("Informasi Internet")
        .setNegativeButton("Tutup",
            new
DialogInterface.OnClickListener() {
                public void onClick(final
DialogInterface dialog,

                @SuppressWarnings("unused") final int id) {
                    dialog.cancel();
                }
            }
        );
}

```

```

    });
    final AlertDialog alert = builder.create();
    alert.show();
}
}

```

## o) TesHasilActivity.java

```

p) package xrb.kelinci;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import xrb.kelinci.helper.JSONParser;
import xrb.kelinci.helper.Penyakit;
import xrb.kelinci.helper.RbHelper;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;

public class TesHasilActivity extends Activity {
    public static ArrayList<Penyakit> data;
    private ArrayList<NameValuePair> nameValuePairs;
    public static HashMap<String, String> dataGejala;
    private JSONObject jsonRespon;
    private LinearLayout layoutLinear, llPelakuTes;
    private TextView tvJudul;
    private EditText txtNoIdentitas;
    private Button btnCari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tes_hasil);
    }
}

```

```

setupView();

    if (BobotActivity.dataTes.size() > 0) {
        nameValuePairs = new ArrayList<NameValuePair>();

        for (Map.Entry<String, String> me :
BobotActivity.dataTes.entrySet()) {
            try {
                nameValuePairs.add(new
BasicNameValuePair(me.getKey()
                    .toString(), me.getValue().toString()));
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        String url = MainActivity.URL_SERVICE + "cf";
        Log.i("url", url);
        new GetDataAsyncTask().execute(url);
    } else {
        RbHelper.pesan(getBaseContext(),
            "Data Tes penyakit Tidak Boleh kosong.");
    }
}

private void setupView() {
    layoutLinear = (LinearLayout)
findViewById(R.id.linearKet);
    tvJudul = (TextView) findViewById(R.id.tvHasilTesJudul);
    tvJudul.setVisibility(View.GONE);
    llPelakuTes = (LinearLayout)
findViewById(R.id.llPelakuTes);
    llPelakuTes.setVisibility(View.GONE);
    txtNoIdentitas = (EditText)
findViewById(R.id.txtHasilNoIdentitas);
    txtNoIdentitas.setVisibility(View.GONE);
    btnCari = (Button) findViewById(R.id.btnHasilCari);
    btnCari.setVisibility(View.GONE);
    btnCari.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            Intent i = new Intent(getBaseContext(),
PelakuTesActivity.class);
            MainActivity.tesEntry = true;
            startActivity(i);
        }
    });
}

private class GetDataAsyncTask extends AsyncTask<String,
Void, String> {
    JSONParser jParser;

```

```

public GetDataAsyncTask() {
    jParser = new JSONParser();
}

@Override
protected void onPreExecute() {
    super.onPreExecute();
    jParser.loadingIn(TesHasilActivity.this);
}

@Override
protected String doInBackground(String... params) {
    String url = params[0], getData = "true";
    Log.i("valuepairs", nameValuePair.toString());
    jsonRespon = jParser.sendDataax(nameValuePair, url);
    Log.i("json", jsonRespon.toString());
    return getData;
}

@Override
protected void onPostExecute(String hasil) {
    jParser.loadingOut();

    try {
        hasil = jsonRespon.getString("result");
        if (hasil.equalsIgnoreCase("true")) {
            // double akhir =
            jsonRespon.getDouble("nilai_cf");
            tvJudul.setVisibility(View.VISIBLE);
            tvJudul.setText("Hasil Diagnosa berdasarkan
            gejala yang diinputkan, "
            + "maka Hewan Kelinci Anda diperkirakan
            menderita penyakit : ");
            try {
                // String jObj2 =
                jsonRespon.getString("penyakit");
                // //RbHelper.pesan(getContext(),
                jsonRespon.getString("id_penyakit"));
                //
                // TextView tvPenyakit = new
                TextView(getContext());
                // tvPenyakit.setTextColor(Color.BLACK);
                // tvPenyakit.setText(jObj2 + " (CF = " +
                String.valueOf(akhir) + ")");
                // layoutLinear.addView(tvPenyakit);

                // info cf_1
                TextView tvInfoCf1 = new
                TextView(getContext());
                tvInfoCf1.setTextColor(Color.BLACK);
                tvInfoCf1.setText("Hasil perhitungan cf 1:
                ");

                tvInfoCf1.setPadding(0, 2, 0, 0);
                layoutLinear.addView(tvInfoCf1);

                JSONObject cf1 =
                jsonRespon.getJSONObject("cf_1");
            }
        }
    }
}

```



```

        if
        (cf1.getString("result").equalsIgnoreCase("true")) {
            String cf1Penyakit =
            cf1.getString("penyakit");

            TextView tvCf11 = new
            TextView(getBaseContext());
            tvCf11.setTextColor(Color.BLACK);
            tvCf11.setText("Penyakit : " +
            cf1Penyakit);

            tvCf11.setPadding(20, 2, 0, 0);
            layoutLinear.addView(tvCf11);

            String cf1Nilai =
            cf1.getString("nilai_cf");

            TextView tvCf12 = new
            TextView(getBaseContext());
            tvCf12.setTextColor(Color.BLACK);
            tvCf12.setText("Nilai CF : " + cf1Nilai);
            tvCf12.setPadding(20, 2, 0, 20);
            layoutLinear.addView(tvCf12);

        } else {
            TextView tvCf1 = new
            TextView(getBaseContext());
            tvCf1.setTextColor(Color.BLACK);
            tvCf1.setText("Penyakit : Tidak
            ditemukan");

            tvCf1.setPadding(20, 2, 0, 20);
            layoutLinear.addView(tvCf1);
        }

        // info cf_2
        TextView tvInfoCf2 = new
        TextView(getBaseContext());
        tvInfoCf2.setTextColor(Color.BLACK);
        tvInfoCf2.setText("Hasil perhitungan cf 2:
        ");

        tvInfoCf2.setPadding(0, 2, 0, 0);
        layoutLinear.addView(tvInfoCf2);

        JSONObject cf2 =
        jsonRespon.getJSONObject("cf_2");
        if
        (cf2.getString("result").equalsIgnoreCase("true")) {
            String cf2Penyakit =
            cf2.getString("penyakit");

            TextView tvCf21 = new
            TextView(getBaseContext());
            tvCf21.setTextColor(Color.BLACK);
            tvCf21.setText("Penyakit : " +
            cf2Penyakit);

            tvCf21.setPadding(20, 2, 0, 0);
            layoutLinear.addView(tvCf21);

            String cf2Nilai =

```

```

cf2.getString("nilai_cf");

        TextView tvCf22 = new
TextView(getBaseContext());
        tvCf22.setTextColor(Color.BLACK);
        tvCf22.setText("Nilai CF : " + cf2Nilai);
        tvCf22.setPadding(20, 2, 0, 20);
        layoutLinear.addView(tvCf22);

    } else {
        TextView tvCf2 = new
TextView(getBaseContext());
        tvCf2.setTextColor(Color.BLACK);
        tvCf2.setText("Penyakit : Tidak
ditemukan");

        tvCf2.setPadding(20, 2, 0, 20);
        layoutLinear.addView(tvCf2);
    }

    // info ds_1
    TextView tvInfoDs1 = new
TextView(getBaseContext());
    tvInfoDs1.setTextColor(Color.BLACK);
    tvInfoDs1.setText("Hasil perhitungan ds 1:
");

    tvInfoDs1.setPadding(0, 2, 0, 0);
    layoutLinear.addView(tvInfoDs1);

    JSONObject dsl =
jsonRespon.getJSONObject("ds_1");
    if
(dsl.getString("result").equalsIgnoreCase("true")) {
        String dslPenyakit =
dsl.getString("penyakit");

        TextView tvDs11 = new
TextView(getBaseContext());
        tvDs11.setTextColor(Color.BLACK);
        tvDs11.setText("Penyakit : " +
dslPenyakit);

        tvDs11.setPadding(20, 2, 0, 0);
        layoutLinear.addView(tvDs11);

        String dslNilai =
dsl.getString("nilai_ds");

        TextView tvDs12 = new
TextView(getBaseContext());
        tvDs12.setTextColor(Color.BLACK);
        tvDs12.setText("Nilai DS : " + dslNilai);
        tvDs12.setPadding(20, 2, 0, 20);
        layoutLinear.addView(tvDs12);

    } else {
        TextView tvDs1 = new
TextView(getBaseContext());
        tvDs1.setTextColor(Color.BLACK);
        tvDs1.setText("Penyakit : Tidak

```

```

ditemukan");
        tvDs1.setPadding(20, 2, 0, 20);
        LinearLayout.addView(tvDs1);
    }

    // info ds_2
    TextView tvInfoDs2 = new
    TextView(getBaseContext());
    tvInfoDs2.setTextColor(Color.BLACK);
    tvInfoDs2.setText("Hasil perhitungan ds 2:
");
    tvInfoDs2.setPadding(0, 2, 0, 0);
    LinearLayout.addView(tvInfoDs2);

    JSONObject ds2 =
jsonRespon.getJSONObject("ds_2");
    if
(ds2.getString("result").equalsIgnoreCase("true")) {
        String ds2Penyakit =
ds2.getString("penyakit");

        TextView tvDs21 = new
        TextView(getBaseContext());
        tvDs21.setTextColor(Color.BLACK);
        tvDs21.setText("Penyakit : " +
ds2Penyakit);
        tvDs21.setPadding(20, 2, 0, 0);
        LinearLayout.addView(tvDs21);

        String ds2Nilai =
ds2.getString("nilai_ds");

        TextView tvDs22 = new
        TextView(getBaseContext());
        tvDs22.setTextColor(Color.BLACK);
        tvDs22.setText("Nilai DS : " + ds2Nilai);
        tvDs22.setPadding(20, 2, 0, 20);
        LinearLayout.addView(tvDs22);

    } else {
        TextView tvDs2 = new
        TextView(getBaseContext());
        tvDs2.setTextColor(Color.BLACK);
        tvDs2.setText("Penyakit : Tidak
ditemukan");

        tvDs2.setPadding(20, 2, 0, 20);
        LinearLayout.addView(tvDs2);
    }
}

```

```

        // info gejala
        TextView tvInfo = new
TextView(getBaseContext());
        tvInfo.setTextColor(Color.BLACK);
        tvInfo.setText("Gejala penyakit : ");
        tvInfo.setPadding(20, 2, 0, 0);
        LinearLayout.addView(tvInfo);

        // buat linear layout baru
        LinearLayout llayout = new LinearLayout(
            getBaseContext());

        llayout.setOrientation(LinearLayout.VERTICAL);
        llayout.setPadding(20, 1, 0, 13);
        LinearLayout.addView(llayout);

        JSONArray jGejala =
jsonRespon.getJSONArray("gejala");
        for (int i = 0; i < jGejala.length(); i++) {
            TextView tvGejala = new
TextView(getBaseContext());
            tvGejala.setTextColor(Color.BLACK);
            tvGejala.setText("- " +
jGejala.getString(i));
            llayout.addView(tvGejala);
        }

        // info pengobatan
        String cekPengobatan =
jsonRespon.getString("result_pengobatan");
        if(cekPengobatan.equalsIgnoreCase("true")){

            TextView tvInfo2 = new
TextView(getBaseContext());
            tvInfo2.setTextColor(Color.BLACK);
            tvInfo2.setText("Cara Pengobatan : ");
            tvInfo2.setPadding(20, 2, 0, 0);
            LinearLayout.addView(tvInfo2);

            // buat linear layout baru
            LinearLayout llayout2 = new LinearLayout(
                getBaseContext());

            llayout2.setOrientation(LinearLayout.VERTICAL);
            llayout2.setPadding(20, 1, 0, 13);
            LinearLayout.addView(llayout2);

            JSONArray jPengobatan =
jsonRespon.getJSONArray("data_pengobatan");
            for (int i = 0; i < jPengobatan.length();
i++) {

                String x = jPengobatan.getString(i);

                TextView tv = new

```

```

TextView(getBaseContext());
        tv.setTextColor(Color.BLACK);
        tv.setText("- " + x);
        llayout2.addView(tv);
    }
} else {
    RbHelper.pesan(getBaseContext(), "Belum
ada data cara pengobatannya.");
}

// info pencegahan
String cekPencegahan =
jsonRespon.getString("result_pencegahan");
if(cekPencegahan.equalsIgnoreCase("true")) {

    TextView tvInfo3 = new
TextView(getBaseContext());
    tvInfo3.setTextColor(Color.BLACK);
    tvInfo3.setText("Cara pencegahan : ");
    tvInfo3.setPadding(20, 2, 0, 0);
    layoutLinear.addView(tvInfo3);

    // buat linear layout baru
    LinearLayout llayout3 = new LinearLayout(
        getBaseContext());

    llayout3.setOrientation(LinearLayout.VERTICAL);
    llayout3.setPadding(20, 1, 0, 13);
    layoutLinear.addView(llayout3);

    JSONArray jPencegahan =
jsonRespon.getJSONArray("data_pencegahan");
    for (int i = 0; i < jPencegahan.length();
i++) {
        String x = jPencegahan.getString(i);

        TextView tv = new
TextView(getBaseContext());
        tv.setTextColor(Color.BLACK);
        tv.setText(" - " + x);
        llayout3.addView(tv);
    }
} else {
    RbHelper.pesan(getBaseContext(), "Belum
ada data cara pencegahannya.");
}

} catch (JSONException e) {
    e.printStackTrace();
}

} else {
    if (MainActivity.isLogin) {
        Intent i = new Intent(getBaseContext(),
            PelakuTesActivity.class);
        MainActivity.tesEntry = false;
    }
}

```



**Android xrb.kelinci .helper**

## a) Gejala.java

```
package xrb.kelinci.helper;

public class Gejala extends Jenis {
    private String gejalaId, gejalaKode, gejalaNama;

    public String getGejalaId() {
        return gejalaId;
    }

    public void setGejalaId(String gejalaId) {
        this.gejalaId = gejalaId;
    }

    public String getGejalaKode() {
        return gejalaKode;
    }

    public void setGejalaKode(String gejalaKode) {
        this.gejalaKode = gejalaKode;
    }

    public String getGejalaNama() {
        return gejalaNama;
    }

    public void setGejalaNama(String gejalaNama) {
        this.gejalaNama = gejalaNama;
    }
}
```

## b) Jenis.java

```
package xrb.kelinci.helper;

public class Jenis {
    private String jenisId, jenisNama;

    public String getJenisId() {
        return jenisId;
    }

    public void setJenisId(String jenisId) {
        this.jenisId = jenisId;
    }

    public String getJenisNama() {
        return jenisNama;
    }

    public void setJenisNama(String jenisNama) {
        this.jenisNama = jenisNama;
    }
}
```



## c) JsonHelper.java

```
package xrb.kelinci.helper;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class JsonHelper {
    public static Object toJSON(Object object) throws
    JSONException {
        if (object instanceof Map) {
            JSONObject json = new JSONObject();
            Map map = (Map) object;
            for (Object key : map.keySet()) {
                json.put(key.toString(),
                toJSON(map.get(key)));
            }
            return json;
        } else if (object instanceof Iterable) {
            JSONArray json = new JSONArray();
            for (Object value : ((Iterable) object)) {
                json.put(value);
            }
            return json;
        } else {
            return object;
        }
    }

    public static boolean isEmptyObject(JSONObject object) {
        return object.names() == null;
    }
}
```

```
        public static Map<String, Object> getMap(JSONObject object,
String key)
            throws JSONException {
            return toMap(object.getJSONObject(key));
        }

        public static Map<String, Object> toMap(JSONObject object)
            throws JSONException {
            Map<String, Object> map = new HashMap();
            Iterator keys = object.keys();
            while (keys.hasNext()) {
                String key = (String) keys.next();
                map.put(key, fromJson(object.get(key)));
            }
            return map;
        }

        public static List toList(JSONArray array) throws
JSONException {
            List list = new ArrayList();
            for (int i = 0; i < array.length(); i++) {
                list.add(fromJson(array.get(i)));
            }
            return list;
        }

        private static Object fromJson(Object json) throws
JSONException {
            if (json == JSONObject.NULL) {
                return null;
            } else if (json instanceof JSONObject) {
                return toMap((JSONObject) json);
            } else if (json instanceof JSONArray) {
                return toList((JSONArray) json);
            } else {
                return json;
            }
        }
    }
}
```

```
}  
}
```

## d) JSONParser.java

```
package xrb.kelinci.helper;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.UnsupportedEncodingException;  
import java.util.ArrayList;  
  
import org.apache.http.HttpEntity;  
import org.apache.http.HttpResponse;  
import org.apache.http.NameValuePair;  
import org.apache.http.client.ClientProtocolException;  
import org.apache.http.client.HttpClient;  
import org.apache.http.client.entity.UrlEncodedFormEntity;  
import org.apache.http.client.methods.HttpPost;  
import org.apache.http.impl.client.DefaultHttpClient;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.util.Log;  
import android.widget.Toast;
```

```
public class JSONParser {

    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    private ProgressDialog pDialog;

    public JSONParser() {

    }

    public JSONObject AmbilJson(String url) {

        try {
            DefaultHttpClient httpClient = new
DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse =
httpClient.execute(httpPost);
            HttpEntity httpEntity =
httpResponse.getEntity();
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            BufferedReader reader = new
BufferedReader(new InputStreamReader(
                is, "utf-8"), 8);
            StringBuilder sb = new StringBuilder();
```

```
String line = null;
while ((line = reader.readLine()) != null) {
    sb.append(line + "\n");
}
is.close();
json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting
result " + e.toString());
}

try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " +
e.toString());
}

return jsonObj;
}

//loading pada saat data json sedang diambil
public void loadingIn(Context c){
    progressDialog = new ProgressDialog(c);
    progressDialog.setMessage("Loading...");
    progressDialog.setIndeterminate(false);
    progressDialog.setCancelable(false);
    progressDialog.show();
}

//mengakhiri loading setelah data json selesai diambil
public void loadingOut(){
    progressDialog.dismiss();
}

public String sendData(ArrayList<NameValuePair> data,
String server)
```

```
{
    // 1) Connect via HTTP. 2) Encode data. 3) Send data.
    String respon="";
    try
    {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(server);
        httpPost.setEntity(new UrlEncodedFormEntity(data));
        HttpResponse response =
httpClient.execute(httpPost);
        Log.i("postData",
response.getStatusLine().toString());
        //Could do something better with response.
        is = response.getEntity().getContent();
    }
    catch(Exception e)
    {
        Log.e("log_tag", "Error: "+e.toString());
    }
    try {
        BufferedReader reader = new
BufferedReader(new InputStreamReader(
is, "utf-8"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line);
        }
        is.close();
        respon = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting
result " + e.toString());
    }

    Log.i("respon", respon);
}
```

```
        return respon;
    }

    public JSONObject sendDataX(ArrayList<NameValuePair> data,
String server)
    {
        // 1) Connect via HTTP. 2) Encode data. 3) Send data.
        String respon="";
        jObj = null;
        try
        {
            HttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(server);
            httpPost.setEntity(new UrlEncodedFormEntity(data));
            HttpResponse response =
httpClient.execute(httpPost);
            //Log.i("postData",
response.getStatusLine().toString());
            //Could do something better with response.
            is = response.getEntity().getContent();
        }
        catch(Exception e)
        {
            Log.e("log_tag", "Error: "+e.toString());
        }

        try {
            BufferedReader reader = new
BufferedReader(new InputStreamReader(
is, "utf-8"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
        }
    }
}
```

```
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting
result " + e.toString());
    }

    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " +
e.toString());
    }

    return jsonObj;
}
public JSONObject sendDataax(String server)
{
    // 1) Connect via HTTP. 2) Encode data. 3) Send data.
    String respon="";
    try
    {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(server);
        //httpPost.setEntity(new
        UrlEncodedFormEntity(data));
        HttpResponse response =
httpClient.execute(httpPost);
        //Log.i("postData",
response.getStatusLine().toString());
        //Could do something better with response.
        is = response.getEntity().getContent();

    }
    catch(Exception e)
    {
        Log.e("log_tag", "Error: "+e.toString());
    }
}
```



```

    }

    try {

        BufferedReader reader = new
BufferedReader(new InputStreamReader(
                                is, "utf-8"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting
result " + e.toString());
    }

    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " +
e.toString());
    }

    return jsonObj;
}

public void pesan(String msg, Context c) {
    // TODO Auto-generated method stub
    Toast.makeText(c, msg, Toast.LENGTH_SHORT).show();
}
}

```

## e) PelakuTes.java

```

package xrb.kelinci.helper;

```

```
public class PelakuTes {  
    private String id, nama, alamat, noIdentitas, telp,  
password;  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String password) {  
        this.password = password;  
    }  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return nama;  
    }  
  
    public void setName(String nama) {  
        this.nama = nama;  
    }  
  
    public String getAddress() {  
        return alamat;  
    }  
  
    public void setAddress(String alamat) {  
        this.alamat = alamat;  
    }  
  
    public String getNoIdentitas() {  
        return noIdentitas;  
    }  
}
```

```
    }  
  
    public void setNoIdentitas(String noIdentitas) {  
        this.noIdentitas = noIdentitas;  
    }  
  
    public String getTelp() {  
        return telp;  
    }  
  
    public void setTelp(String telp) {  
        this.telp = telp;  
    }  
  
}
```

## f) Pencegahan.java

```
package xrb.kelinci.helper;  
  
public class Pencegahan {  
    private String id, nama, kode;  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getNama() {
```

```
        return nama;
    }

    public void setName(String nama) {
        this.nama = nama;
    }

    public String getKode() {
        return kode;
    }

    public void setKode(String kode) {
        this.kode = kode;
    }
}
```

## g) Pengobatan.java

```
package xrb.kelinci.helper;

public class Pengobatan {
    private String id, kode, nama;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getKode() {
        return kode;
    }

    public void setKode(String kode) {
        this.kode = kode;
    }
}
```

```

    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }
}

```

## h) Penyakit.java

```

package xrb.kelinci.helper;

public class Penyakit {
    private String id, nama, namaLatin, pengendalian, kode;

    public String getKode() {
        return kode;
    }

    public void setKode(String kode) {
        this.kode = kode;
    }

    public String getPengendalian() {
        return pengendalian;
    }

    public void setPengendalian(String pengendalian) {
        this.pengendalian = pengendalian;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {

```

```

        this.nama = nama;
    }

    public String getNamaLatin() {
        return namaLatin;
    }

    public void setNamaLatin(String namaLatin) {
        this.namaLatin = namaLatin;
    }
}

```

### Analisis Sistem

Kasus	Gejala	Diagnosa Pakar	
		Penyakit	CF
1	G2 ( Badan Kurus) 0.5 G11 (Kotorannya hijau gelap dan bau) 0.8 G12 ( Lendir menggantung pada dubur) 0.6	Enteritis Kompleks	0.9

	G19 (Hewan terlihat lesu) 0.7 G39 (Kotoran lembek cair) 0.7		
2	G3 (Sesak nafas atau susah bernafas), 0.6 G21(Perut tampak besar), 0.8 G55 (Badan ambruk tidak bias berdiri) 0.7	Bloat(Tympany)	0.8
3	G2 (Badan kurus), 0.7 G53 (Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya), 0.7 G54 (Kotoran mata lebih banyak dari biasanya dan berwarna kuning) 0.6	Cacing Gilig (Pinworm/Oxyuris)	0.8
4	G2 (Badan kurus), 0.5 G3 (Sesak nafas atau susah bernafas), 0.5 G10 (Daerah anus kotor), G25 (Kepala sering diangkat tinggi-tinggi). 0.7	Radang Paru-paru (Pneumonia/Pasteur elloises)	0.7
5	G7 (Lendir di vagina berupa darah atau nanah), 0.9 G50 (Minum banyak, makan sedikit), 0.8 G51 (Jika kandungan diraba terasa membesar). 0.7	Infeksi Kandungan (Metritis)	0.9
6	G3 (Sesak nafas atau susah bernafas), 0.7 G25 (Kepala sering diangkat tinggi-tinggi), 0.8 G26 (Hidung keluar nanah, lender berwarna jernih atau keruh), 0.8 G4 (Mata sembab, basah dan berair), 0.6	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.9

	G31 (Iritasi dibagian kulit, lalu menular kebagian dalam yang lain). 0.5		
7	G5 (Bulu rontok), 0.8 G29 (Merasa gatal), 0.8 G31 (Iritasi dibagian kulit, lalu menular kebagian dalam yang lain). 0.5	Jamur(Ring Worm)	0.7
8	G1 (Suhu badan naik), 0.8 G16 (Nafsu makan turun), 0.6 G18 (Ambing dan puting berwarna merah, mengeras), 0.7 G43 (Ujung puting berwarna hitam) 0.7	Radang Susu(Mastitis)	0.9
9	G3 (Sesak nafas atau susah bernafas), 0.6 G50 (Minum banyak, makan sedikit), 0.6 G25 (Kepala sering diangkat tinggi-tinggi), 0.7 G29 (Merasa gatal). 0.4	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.6
10	G2 (Badan kurus), 0.5 G16 (Nafsu makan turun), 0.5 G15 (Perut kembung) 0.5	Kudis(Scabies)	0.3
11	G2 (Badan kurus), 0.7 G16 (Nafsu makan turun), 0.7 G20 (Berat badan berkurang), 0.8 G22 (Mencret bercampur darah) 0.8	Parasit Pencernaan (Coccidiosis)	0.8
12	G3 (Sesak nafas atau susah bernafas), 0.7 G15 (Perut kembung), 0.5 G20 (Berat badan berkurang).0.5	Bloat(Tympany)	0.5
13	G2 (Badan kurus), 0.4 G5 (Bulu rontok), 0.7	Kudis(Scabies)	0.9



	G27 (Kulit kemerah-merahan), 0.6 G28 (Badan penuh keropeng), 0.6 G29 (Merasa gatal). 0.7		
14	G4 (Mata sembab, basah, dan berair), 0.7 G35 (Radang berwarna merah, terutama pada kelopak atau selaput mata), 0.6 G36 (Bulu sekitar mata basah dan kusam). 0.6	Radang Mata	0.8
15	G3 (Sesak nafas atau susah bernafas), 0.5 G29 (Merasa gatal), 0.7 G15 (Perut kembung), 0.6 G16 (Nafsu makan turun).0.6	Bloat(Tympany)	0.5
16	G4 (Mata sembab, basah, dan berair), 0.7 G35 (Radang berwarna merah, terutama pada kelopak atau selaput mata), 0.6 G36 (Bulu disekitar mata basah dan kusam), 0.6 G37 (Keluar air mata terus-menerus) 0.7	Radang Mata	0.9
17	G5 (Bulu rontok), 0.6 G6 (Bulu jadi menggumpal dan kusut), 0.6 G29 (Merasa gatal) 0.7	Jamur(Ring Worm)	0.4
18	G2 (Badan kurus), 0.5 G52 (Gatal di sekitar lubang dubur), 0.7 G53 (Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya) 0.7	Cacing Gilig (Pinworm/Oxyuris)	0.7
19	G2 (Badan kurus), 0.4	Cacingan	0.5

	G41 (Hewan terlihat lemah), 0.6 G42 (Hewan terlihat pucat) 0.5		
20	G2 (Badan kurus), 0.5 G40 (Gangguan pencernaan akibat penumpukan bulu disaluran pencernaan), 0.6 G33 (Muntah bulu), 0.7 G16 (Nafsu makan turun) 0.6	Makan Bulu(Hairball)	0.9
21	G2 (Badan kurus), 0.5 G42 (Hewan terlihat pucat), 0.3 G34 (Kotoran ada cacing). 0.7	Cacingan	0.9
22	G3 (Sesak nafas atau susah bernafas), 0.7 G4 (Mata sembab, basah, dan berair), 0.5 G25 (Kepala sering diangkat tinggi-tinggi), 0.7 G26 (Hidung keluar nanah, lender berwarna jernih atau keruh), 0.6 G30 (Bersin-bersin) 0.5	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.8
23	G2 (Badan kurus), 0.2 G11 (Kotorannya hijau gelap dan bau), 0.6 G12 (Lendir menggantung pada dubur), 0.5 G19 (Hewan terlihat lesu), 0.3 G39 (Kotoran lembek cair), 0.5 G10 (Daerah anus kotor) 0.5	Enteritis Kompleks	0.9
24	G1 (Suhu badan naik), 0.6 G17 (Hewan kesakitan jika disentuh)	Radang Susu(Mastitis)	0.9

	bagian ambing), 0.6 G18 (Ambing dan puting berwarna merah, mengeras), 0.7 G44 (Susu diperah rusak/pecah) 0.5		
25	G2 (Badan kurus), 0.5 G16 (Nafsu makan turun), 0.12 G3 (Sesak nafas atau susah nafas). 0.5	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.3

**Bobot Gejala Tiap Rule**

*Rule 1 = IF G21 AND G3 AND G55 THEN P12*

CF=0.8

Rule 2 = IF G3 AND G21 THEN P12

CF = 0.7

Penyakit = P12 (BLOAT)		MB	MD
G21	Perut tampak besar	0.7	0.3
G3	Sesak nafas/susah bernafas	0.5	0.5
G55	Badan ambruk tidak bisa berdiri	0.6	0.4
Penyakit = P12 (BLOAT)		MB	MD
G3	Sesak Nafas/ susah bernafas	0.5	0.5
G21	Perut tampak besar	0.7	0.3

Rule 3 = IF G3 AND G55 THEN P12

CF=0.7

PENYAKIT = P12 (BLOAT)		MB	MD
G3	Sesak nafas/ susah bernafas	0.5	0.5
G55	Badan ambruk tidak bisa berdiri	0.6	0.4

Rule 4 = IF G2 AND G16 AND G34 AND G41 AND G42 THEN P9

CF= 0.8

PENYAKIT = P9 (CACINGAN)		MB	MD
G2	Badan kurus	0.5	0.5
G16	Nafsu makan turun	0.3	0.6
G34	Kotoran ada cacing	0.7	0.3
G41	Hewan terlihat lemah	0.25	0.75
G42	Hewan terlihat pucat	0.2	0.8

Rule 5 = IF G2 AND G16 AND G41 THEN P9

CF= 0.5

PENYAKIT = P9 (CACINGAN)		MB	MD
--------------------------	--	----	----

G2	Badan kurus	0.5	0.5
G16	Nafsu makan turun	0.3	0.7
G41	Hewan terlihat lemah	0.25	0.75

*Rule 6 = IF G2 AND G41 AND G42 THEN P9*

*CF = 0.5*

PENYAKIT = P9 (CACINGAN)		MB	MD
G2	Badan kurus	0.5	0.5
G41	Hewan terlihat lemah	0.25	0.75
G42	Hewan terlihat pucat	0.2	0.8

*Rule 7 = IF G2 AND G42 AND G34 THEN P9*

*CF = 0.8*

PENYAKIT = P9 (CACINGAN)		MB	MD
G2	Badan kurus	0.5	0.5
G42	Hewan terlihat pucat	0.2	0.8
G34	Kotoran ada cacing	0.7	0.3

*Rule 8 = IF G2 AND G11 AND G12 AND P19 AND P39 THEN P1*

*CF = 0.8*

PENYAKIT = P1 (ENTERITIS KOMPLEKS)		MB	MD
G2	Badan kurus	0.33	0.66
G11	Kotoran hijau gelap & bau	0.5	0.5
G12	Lendir menggantung pada dubur	0.5	0.5
G19	Hewan terlihat lesu	0.2	0.8
G39	Kotoran lembek cair	0.5	0.5

*Rule 9 = IF G2 AND G11 AND G13 AND G14 THEN P1*

*CF = 0.4*

PENYAKIT = P1 (ENTERITIS KOMPLEKS)		MB	MD
G2	Badan kurus	0.33	0.66

G11	Kotoran hijau & gelap	0.5	0.5
G13	Pertumbuhan melambat	0.5	0.5
G14	Perut diraba terasa sakit	0.25	0.75

*Rule 10* = G2 AND G8 AND G9 AND G10 AND G11 AND G12 AND G13 AND G14 AND G15 AND G19 AND G39 THEN P1

CF = 0.9

PENYAKIT = P1(ENTERITIS KOMPLEKS)		MB	MD
G2	Badan kurus	0.33	0.66
G8	Daun telinga turun	0.5	0.5
G9	Gigi berkerot menahan sakit	0.2	0.8
G10	Daerah anus kotor	0.5	0.5
G11	Kotoran hijau&bau	0.5	0.5
G12	Lender menggantung pada dubur	0.5	0.5
G13	Pertumbuhan melambat	0.5	0.5
G14	Perut diraba sakit	0.25	0.75
G15	Perut kembung	0.2	0.8
G19	Hewan terlihat lesu	0.2	0.8
G39	Kotoran lembek cair	0.5	0.5

*Rule 11* = G2 AND G10 AND G11 AND G19 AND G39 THEN P1      CF = 0.9

PENYAKIT = P1(ENTERITIS KOMPLEKS)		MB	MD
G2	Badan kurus	0.33	0.66
G10	Darah anus kotor	0.5	0.5
G11	Kotoran hijau&bau	0.5	0.5
G19	Hewan terlihat lesu	0.2	0.8
G39	Kotoran lembek cair	0.5	0.5

*Rule 12 = G2 AND G8 AND G11 AND G12 AND G14 AND G15 AND G19 THEN*

*P1*

*CF = 0.8*

PENYAKIT = P1(ENTERITIS KOMPLEKS)		MB	MD
G2	Badan kurus	0.33	0.66
G8	Daun telinga turun	0.5	0.5
G11	Kotoran hijau&bau	0.5	0.5
G12	Lender menggantung pada dubur	0.5	0.5
G14	Perut diraba terasa sakit	0.25	0.75
G15	Perut kembung	0.2	0.8
G19	Hewan terlihat lesu	0.2	0.8

*Rule 13 = IF G2 AND G16 AND G33 AND G40 THEN P8*

*CF = 0.8*

PENYAKIT = P8 (HAIRBALL)		MB	MD
G2	Badan kurus	0	1
G16	Nafsu makan turun	0.1	0.9
G33	Mutah bulu	0.5	0.5
G40	Gangguan pencernaan	0.6	0.4

*Rule 14 = G2 AND G3 AND G16 AND G33 AND G40 THEN P8*

*CF = 0.9*

PENYAKIT = P8 (HAIRBALL)		MB	MD
G2	Badan kurus	0	1
G3	Sesak nafas	0.2	0.8
G16	Nafsu makan turun	0.1	0.9
G33	Muntah bulu	0.5	0.5
G40	Gangguan pencernaan	0.6	0.4

*Rule 15 = G2 AND G16 AND G3 THEN P8*

*CF = 0.3*

PENYAKIT = P8 (HAIRBALL)		MB	MD
--------------------------	--	----	----

G2	Badan kurus	0	1
G16	Nafsu makan turun	0.1	0.9
G3	Sesak nafas	0.2	0.8

*Rule 16 = IF G2 AND G3 AND G16 AND G40 THEN P8*

*CF = 0.8*

PENYAKIT = P8 (HAIRBALL)		MB	MD
G2	Badan kurus	0	1
G3	Sesak nafas	0.2	0.8
G16	Nafsu makan turun	0.1	0.9
G40	Gangguan pencernaan	0.6	0.4

*Rule 17 = IF G7 AND G50 AND G51 THEN P10*

*CF = 0.8*

PENYAKIT = P10 (INFEKSI KANDUNGAN)		MB	MD
G7	Lendir di vagina berupa darah/nanah	0.33	0.66
G50	Minum banyak, makan dikit	0.4	0.6
G51	Kandungaan diraba terasa membesar	0.5	0.5

*Rule 18 = IF G7 AND G50 THEN P10*

*CF = 0.7*

PENYAKIT = P10 (INFEKSI KANDUNGAN)		MB	MD
G7	Lendir di vagina berupa darah/nanah	0.33	0.66
G50	Minum banyak, makan dikit	0.4	0.6

*Rule 19 = IF G7 AND G51 THEN P10*

*CF = 0.7*

PENYAKIT = P10 (INFEKSI KANDUNGAN)		MB	MD
------------------------------------	--	----	----



G7	Lendir di vagina berupa darah/nanah	0.33	0.66
G51	Kandungannya diraba terasa membesar	0.5	0.5

*Rule 20 = IF G5 AND G6 AND G29 AND 31 AND G32 THEN P6 CF = 0.9*

PENYAKIT = P6 ( JAMUR )		MB	MD
G5	Bulu rontok	0.4	0.6
G6	Bulu menggumpal	0.1	0.9
G29	Merasa gatal	0.7	0.3
G31	Iritasi di bag kepala kaki	0.8	0.2
G32	Bagian terserang timbul bercak merah	0.6	0.4

*Rule 21 = IF G5 AND G29 AND G32 THEN P6 CF = 0.9*

PENYAKIT = P6 ( JAMUR )		MB	MD
G5	Bulu rontok	0.4	0.6
G29	Merasa gatal	0.7	0.3
G32	Bagian terserang timbul bercak merah	0.6	0.4

*Rule 22 = IF G5 AND G29 AND G31 THEN P6 CF = 0.7*

PENYAKIT = P6 ( JAMUR )		MB	MD
G5	Bulu rontok	0.4	0.6
G29	Merasa gatal	0.7	0.3
G31	Iritasi di bag kepala kaki	0.8	0.2

*Rule 23 = IF G5 AND G6 AND G29 THEN P6 CF = 0.5*

PENYAKIT = P6 ( JAMUR )		MB	MD
G5	Bulu rontok	0.4	0.6
G6	Bulu menggumpal	0.1	0.9
G29	Merasa gatal	0.7	0.3

*Rule 24 = IF G2 AND G5 AND G20 AND G28 AND G29 THEN P5      CF = 0.8*

PENYAKIT = P5 ( KUDIS )		MB	MD
G2	Badan kurus	0.5	0.5
G5	Bulu rontok	0.3	0.7
G20	Berat badan berkurang	0.2	0.8
G28	Badan penuh keropeng	1	0
G29	Merasa gatal	0.8	0.2

*Rule 25 = IF G2 AND G5 AND G27 AND G28 AND G29 THEN P5      CF = 0.9*

PENYAKIT = P5 ( KUDIS )		MB	MD
G2	Badan kurus	0.5	0.5
G5	Bulu rontok	0.3	0.7
G27	Kulit kemerahan	0.8	0.2
G28	Badan penuh keropeng	1	0
G29	Merasa gatal	0.8	0.2

*Rule 26 = IF G2 AND G20 AND G5 AND G27 THEN P5      CF = 0.6*

PENYAKIT = P5( KUDIS )		MB	MD
G2	Badan kurus	0.5	0.5
G20	Berat badan berkurang	0.2	0.8
G5	Bulu rontok	0.3	0.7
G27	Kulit kemerahan	0.8	0.2

*Rule 27 = IF G2 AND G5 AND G16 THEN P5*

CF = 0.4

PENYAKIT = P5 ( KUDIS )		MB	MD
G2	Badan kurus	0.5	0.5
G5	Bulu rontok	0.3	0.7
G16	Nafsu makan turun	0.1	0.9

*Rule 28 = IF G2 AND G5 AND G16 AND G20 AND G27 AND G28 AND G29 THEN P5*

CF = 0.9

PENYAKIT = P5 ( KUDIS )		MB	MD
G2	Badan kurus	0.5	0.5
G5	Bulu rontok	0.3	0.7
G16	Nafsu makan turun	0.1	0.9
G20	Berat badan berkurang	0.2	0.8
G27	Kulit kemerahan	0.8	0.2
G28	Badan penuh keropang	1	0
G29	Merasa gatal	0.8	0.2

*Rule 29 = IF G2 AND G9 AND G19 AND G20 THEN P3*

CF = 0.4

PENYAKIT = P3 ( PARASIT PENCERNAAN )		MB	MD
G2	Badan kurus	0.5	0.5
G9	Gigi berkerot menahan sakit	0.2	0.8
G19	Hewan terlihat lesu	0.33	0.66
G20	Berat badan berkurang	0.33	0.66

*Rule 30 = IF G2 AND G10 AND G20 AND G22 AND G23 THEN P3*

CF = 0.8

PENYAKIT = P3 ( PARASIT PENCERNAAN )		MB	MD
G2	Badan kurus	0.5	0.5
G10	Daerah anus kotor	0.25	0.75
G20	Berat badan berkurang	0.33	0.66
G22	Mencret bercampur darah	0.7	0.3
G23	Kotoran berlendir putih	0.33	0.66

*Rule 31 = IF G2 AND G19 AND G20 AND G22 AND G23 THEN P3      CF = 0.8*

PENYAKIT = P3 ( PARASIT PENCERNAAN )		MB	MD
G2	Badan kurus	0.5	0.5
G19	Hewan terlihat lesu	0.33	0.66
G20	Berat badan berkurang	0.33	0.66
G22	Mencret campur darah	0.7	0.3
G23	Kotoran berlendir putih	0.33	0.66

*Rule 32 = IF G2 AND G16 AND G20 AND G22 THEN P3      CF = 0.7*

PENYAKIT = P3 ( PARASIT PENCERNAAN )		MB	MD
G2	Badan kurus	0.5	0.5
G16	Nafsu makan turun	0.2	0.8
G20	Berat badan berkurang	0.33	0.66
G22	Mencret campur darah	0.7	0.3

*Rule 33 = IF G2 AND G9 AND G10 AND G16 AND G19 AND G22 AND G23  
THEN P3*

*CF = 0.9*

PENYAKIT = P3 ( PARASIT PENCERNAAN )		MB	MD
G2	Badan kurus	0.5	0.5
G9	Gigi berkerot menahan sakit	0.2	0.8

G10	Daearah anus kotor	0.25	0.75
G16	Nafsu makan turun	0.2	0.8
G19	Hewan terlihat lesu	0.33	0.66
G22	Mencret campur darah	0.7	0.3
G23	Kotoran berlendir putih	0.33	0.66

*Rule 34 = IF G2 AND G52 AND G53 THEN P11* CF = 0.7

PENYAKIT = P11 ( PINWORM )		MB	MD
G2	Badan kurus	0.5	0.5
G52	Gatal disekitar lubang dubur	0.8	0.2
G53	Hewan sering menggaruk sekitar dubur	0.7	0.3

*Rule 35 = IF G2 AND G52 AND G54 THEN P11* CF = 0.7

PENYAKIT = P11 ( PINWORM )		MB	MD
G2	Badan kurus	0.5	0.5
G52	Gatal disekitar lubang dubur	0.8	0.2
G54	Kotoran mata banyak dan warna kuning	0.3	0.7

*Rule 36 = IF G2 AND G52 AND G53 AND G54 THEN P11* CF = 0.8

PENYAKIT = P11 ( PINWORM )		MB	MD
G2	Badan kurus	0.5	0.5
G52	Gatal disekitar dubur	0.8	0.2
G53	Hewan sering menggaruk sekitar dubur	0.7	0.3
G54	Kotoran mata banyak dan warna	0.3	0.7

	kuning		
--	--------	--	--

*Rule 37 = IF G2 AND G53 AND G54 THEN P11*

CF = 0.7

PENYAKIT = P11 ( PINWORM )		MB	MD
G2	Badan kurus	0.5	0.5
G53	Hewan sering menggaruk sekitar dubur	0.7	0.3
G54	Kotoran mata banyak dan kuning	0.3	0.7

*Rule 38 = IF G4 AND G35 AND G37 AND G38 THEN P7*

CF = 0.8

PENYAKIT = P7 ( RADANG MATA )		MB	MD
G4	Mata sembab, basah , berair	0.4	0.6
G35	Radang berwarna merah	0.8	0.2
G37	Keluar air mata terus menerus	0.7	0.3
G38	Hewan selalu mengusap mata dg kaki	0.33	0.66

*Rule 39 = IF G4 AND G35 AND G36 AND G37 AND G38 AND G45 THEN P7*

CF = 0.9

PENYAKIT = P7 ( RADANG MATA )		MB	MD
G4	Mata sembab basah berair	0.4	0.6
G35	Radang berwarna merah	0.8	0.2
G36	Bulu sekitar mata basah dan kusam	0.6	0.4
G37	Keluar air mata terus menerus	0.7	0.3
G38	Hewan selalu mengusap mata dg kaki	0.33	0.66

G45	Mengalami peradangan pada mata	0.7	0.3
-----	--------------------------------	-----	-----

*Rule 40 = IF G4 AND G35 AND G36 THEN P7*

CF = 0.9

PENYAKIT = P7 ( RADANG MATA )		MB	MD
G4	Mata sembab basah berair	0.4	0.6
G35	Radang berwarna merah	0.8	0.2
G36	Bulu sekitar mata basah dan kusam	0.6	0.4

*Rule 41 = IF G4 AND G35 AND G36 AND G37 THEN P7*

CF = 0.8

PENYAKIT = P7 ( RADANG MATA )		MB	MD
G4	Mata sembab basah berair	0.4	0.6
G35	Radang berwarna merah	0.8	0.2
G36	Bulu sekitar mata basah dan kusam	0.6	0.4
G37	Keluar air mata terus menerus	0.7	0.3

*Rule 42 = IF G3 AND G4 AND G25 AND G26 AND G31 THEN P4*

CF = 0.8

PENYAKIT = P4 ( RADANG PARU-PARU )		MB	MD
G3	Sesak nafas	0.5	0.5
G4	Mata sembab basah berair	0.4	0.6
G25	Kepala sering diangkat tinggi"	0.6	0.4
G26	Hidung keluar nanah	0.9	0.1
G31	Iritasi bag kepala/kaki	0.3	0.7

*Rule 43 = IF G3 AND G4 AND G26 THEN P4*

CF = 0.4

PENYAKIT = P4 ( RADANG PARU-PARU )		MB	MD
G3	Sesak nafas	0.5	0.5
G4	Mata sembab basah berair	0.4	0.6
G26	Hidung keluar nanah	0.9	0.1

*Rule 44 = IF G3 AND G4 AND G25 AND G26 AND G30 THEN P4*      CF = 0.8

PENYAKIT = P4 ( RADANG PARU-PARU )		MB	MD
G3	Sesak nafas	0.5	0.5
G4	Mata sembab basah berair	0.4	0.6
G25	Kepala sering diangkat tinggi"	0.6	0.4
G26	Hidung keluar nanah	0.9	0.1
G30	Bersin"	0.5	0.5

*Rule 45 = IF G3 AND G4 AND G25 AND G26 AND G30 AND G31 AND G56  
THEN P4*

CF = 0.9

PENYAKIT = P4 ( RADANG PARU-PARU )		MB	MD
G3	Sesak nafas	0.5	0.5
G4	Mata sembab basah berair	0.4	0.6
G25	Kepala sering diangkat tinggi"	0.6	0.4
G26	Hidung keluar nanah	0.9	0.1
G30	Bersin"	0.5	0.5
G31	Iritasi bag kepala/kaki	0.3	0.7
G56	Leher berputar ke samping	0.7	0.3

*Rule 46 = IF G1 AND G17 AND AND G18 AND G44 THEN P2*

CF = 0.8



PENYAKIT = P2 ( RADANG SUSU )		MB	MD
G1	Suhu badan naik	0.2	0.8
G17	Hewan sakit jika disentuh bag ambing	0.4	0.6
G18	Ambing putting merah, mengeras	0.8	0.2
G44	Susu yg diperah rusak/pecah	0.7	0.3

*Rule 47 = IF G1 AND G10 AND G16 AND G17 AND G18 AND G43 AND G44*

*THEN P2*

*CF = 0.9*

PENYAKIT = P2 ( RADANG SUSU )		MB	MD
G1	Suhu badan naik	0.2	0.8
G10	Daerah anus kotor	0.5	0.5
G16	Nafsu makan turun	0.2	0.8
G17	Hewan sakit jika disentuh bag ambing	0.4	0.6
G18	Ambing merah mengeras	0.8	0.2
G43	Ujung putting berwarna hitam	0.5	0.5
G44	Susu yang diperah pecah/rusak	0.7	0.3

*Rule 48 = IF G1 AND G10 AND G16 AND G18 THEN P2*

*CF = 0.5*

PENYAKIT = P2 ( RADANG SUSU )		MB	MD
G1	Suhu badan naik	0.2	0.8
G10	Daerah anus kotor	0.5	0.5
G16	Nafsu makan turun	0.2	0.8
G18	Ambing merah mengeras	0.8	0.2

*Rule 49 = IF G1 AND G10 AND G16 AND G18 AND G43 THEN P2*

*CF = 0.8*

PENYAKIT = P2 ( RADANG SUSU )		MB	MD
G1	Suhu badan naik	0.2	0.8
G10	Daerah anus kotor	0.5	0.5
G16	Nafsu makan turun	0.2	0.8
G18	Ambing merah mengeras	0.8	0.2
G43	Ujung puting berwarna hitam	0.5	0.5

*Rule 50 = IF G1 AND G16 AND G18 AND G43 THEN P2*

*CF = 0.7*

PENYAKIT = P2 ( RADANG SUSU )		MB	MD
G1	Suhu badan naik	0.2	0.8
G16	Nafsu makan turun	0.2	0.8
G18	Ambing merah mengeras	0.8	0.2
G43	Ujung puting berwarna hitam	0.5	0.5



Program Studi Teknik Informatika  
Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana Yogyakarta  
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

**FORMULIR PERBAIKAN (REVISI) SKRIPSI**  
Strata-1 Program Studi Teknik Informatika

Yang bertanda tangan di bawah ini:

Nama : CHRISTIAN HARTANTO  
N I M : 71150144  
Judul Skripsi : PERBANDINGAN PENGGUNAAN METODE DEMPSTER SHAFER  
DENGAN CERTAINTY FACTOR PADA KASUS DIAGNOSA PENYAKIT  
KELINCI HIAS  
Tanggal Pendadaran : 20 Desember 2016 13:00 WIB

Telah melakukan perbaikan tugas akhir dengan lengkap.

Demikian pernyataan kami agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 21 Desember 2016

Dosen Pembimbing I

  
Gloria Virginia, S.Kom., MAI, Ph.D.

Dosen Pembimbing II

  
Joko Purwadi, M.Kom



### Kartu Konsultasi Tugas Akhir

Program Studi Teknik Informatika Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana Yogyakarta  
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929



NIM : CHRISTIAN HARTANTO  
Judul : **PERBANDINGAN PENGGUNAAN METODE DEMPSTER SHAFER DENGAN CERTAINTY FACTOR PADA KASUS DIAGNOSA PENYAKIT KELINCI HIAS**  
Dosen Pembimbing I : Gloria Virginia, S.Kom., MAI, Ph.D.

1	Tanggal: 11 Nov. 2015	Paraf: <i>Jeni</i>	2	Tanggal: 7 September 2016	Paraf: <i>Jeni</i>
Implementasi Metode Dempster-Shafer pada sistem Atmora			Diskusi implementasi DS broken. Beta CF. Q : 1). Logika sama? 2). Contoh kasus 3). Literatur teori		
3	Tanggal: 27 September 2016	Paraf: <i>Jeni</i>	4	Tanggal: 1 Oktober 2016	Paraf: <i>Jeni</i>
Analisis 2 metod : CF & DS. → perlu perhitungan lebih banyak supaya karakteristik masing2 metode lebih jelas. → Mulai coding!			Tanya : ①. Contoh rules selain "AND" dgn operator ②. Cari perhitungan yg berat on "difficulty in DS" di buku. → Mulai coding!		
5	Tanggal: 1 Nov 2016	Paraf: <i>Jeni</i>	6	Tanggal: 24 Nov 2016	Paraf: <i>Jeni</i>
Diskusi lebih lanjut tentang perbedaan CF & DS. - CF → mempertimbangkan keyakinan pakar. - DS → hanya brk. keyakinan user.			Diskusi proses 1). Atmora : MB user & CF pakar. 2). Atmora : CF pakar & MB pakar. 3). DS : Bel. user 4). DS : MB pakar nya CF		
7	Tanggal: 2 Des 2016	Paraf: <i>Jeni</i>	8	Tanggal:	Paraf:
Acc pedadaran!					



**Kartu Konsultasi Tugas Akhir**

Program Studi Teknik Informatika Fakultas Teknologi Informasi  
 Universitas Kristen Duta Wacana Yogyakarta  
 Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

NIM : CHRISTIAN HARTANTO  
 Judul : **PERBANDINGAN PENGGUNAAN METODE DEMPSTER SHAFER DENGAN CERTAINTY FACTOR PADA KASUS DIAGNOSA PENYAKIT KELINCI HIAS**  
 Dosen Pembimbing II : Joko Purwadi, M.Kom

<p>1 Tanggal: 26/10/2016</p>	<p>Paraf: [Signature]</p>	<p>2 Tanggal: 17/11/2016</p>	<p>Paraf: [Signature]</p>
<p>✓ Penggunaan data yg sudah ada ✓ (CF)                  → nuncas DS                  ✓ Bx 1-2 ds + awal tabel/label yang penting.                  ✓ gunakan pada keputran / tree structure.                  Bx 3 tahap Basis pengetahuan. ✓ ✓</p>		<p>Kontribusi Bx 3 → revisi                  ✓ Bx 3 perbaikan                  ✓ flowchart                  ✓ Perbaikan tabel -                  ✓ Referensi penggunaan data tabel dari ...?</p>	
<p>3 Tanggal: 25/11/2016</p>	<p>Paraf: [Signature]</p>	<p>4 Tanggal: 29/11/2016</p>	<p>Paraf: [Signature]</p>
<p>Kontribusi Program → cek parahnya.                  → aturatis cek input.                  Laporan → Bx 3 diperbaiki di awal, bahwa ini merupakan penyusunan TGA sudah selesai.</p>		<p>Bx 3 direvisi                  ✓ penggunaan dan skripsi ds                  ✓ penataan struktur laporan.                  ✓ cek hasil flowchart                  ✓ perbaikan tabel / gambar</p>	
<p>5 Tanggal: 01/12/2016</p>	<p>Paraf: [Signature]</p>	<p>6 Tanggal: 02/12/2016</p>	<p>Paraf: [Signature]</p>
<p>Kontribusi hasil program:                  ✓ Be (konsepkan lg web service) saat presentasi                  ✓ Lantir kom dari sumber lain -                  ✓ Laporan terse di inputan laporan</p>		<p>Perbaikan Laporan Akhir.                  ✓ Bx 4 median analitis di matangkan kean mumpung kom → di Bx 4                  ✓ DS → analitis ketetapan su</p>	
<p>7 Tanggal:</p>	<p>Paraf:</p>	<p>8 Tanggal:</p>	<p>Paraf:</p>
<p>App pembuatannya di ...?</p>		<p>...</p>	