

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian yang berhubungan dengan pengembangan Sistem pakar (*Expert System*) dan penerapan Faktor Kepastian (*Certainty Factor*) antara lain dilakukan oleh Rohajawati dan Supriyati (2010), Iskandar (2007), Witari, Gandhiadi, dan Kencana (2013). Dalam sebuah penelitian yang dilakukan oleh Rohajawati dan Supriyati (2010), Sistem Pakar merupakan *tools* yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar (*human expert*). Ada beberapa komponen-komponen yang harus dipenuhi oleh sistem pakar yaitu, basis pengetahuan (*knowledge base*), mesin inferensi (*inference engine*), dan antar muka pemakai (*user interface*). Menurut Iskandar (2007), teori *Certainty Factor* sangat membantu pasien agar mudah mendiagnosa suatu gejala penyakit yang dialami oleh pasien. Sistem pakar juga akan mencatat gejala-gejala dari pasien dan akan mendiagnosa jenis penyakit yang berbasis pada komputer sehingga dapat menyelesaikan masalah yang spesifik seperti yang dilakukan oleh para ahli atau pakar (Iskandar, 2007). Untuk merancang suatu sistem pakar dibutuhkan teknologi berbasis *online* yang dikelola melalui sebuah halaman *website* (Witari, Gandhiadi & Kencana, 2013). Begitu pula dengan *Certainty Factor* yang sangat dibutuhkan untuk mengatasi ketidakpastian terhadap gejala-gejala penyakit yang dialami oleh pasien. Sistem Pakar (Witari, Gandhiadi & Kencana, 2013) juga dapat memuat kesimpulan akhir lebih dari satu (*multi output*) dengan mengelompokkan penyakit-penyakit yang memiliki satu atau lebih gejala yang sama dan dapat mengetahui besarnya tingkat kepercayaan dari penyakit yang didiagnosa dengan menggunakan rumus faktor kepastian (*Certainty factor*)

Metode-metode yang digunakan seperti menggumpulkan sumber data yang meliputi data penyakit dan data gejala-gejala penyakit yang menyerang pasien. Dalam proses pembangunan aplikasi dan pembuatan konsep, peneliti menggunakan *System Development Life Cycle* (Rohajawati & Supriyati, 2010) dengan menekankan pada proses identifikasi masalah, analisis perancangan serta pengujian aplikasi. Rohajawati dan

Supriyati (2010), menggunakan basis data dengan analisis dan perancangan menggunakan *Data Flow Diagram* dan *Entity Relationship Diagram*.

Selain itu, konsep *inference engine* dilakukan dengan penggunaan *production rule* (*if...then*) dengan melalui mekanisme *forward chaining* serta menggunakan model *certainty factor* (*CFs*). Contohnya, ada beberapa masalah yang diseleksi untuk sampel dalam diagnosis tersebut dilakukan dengan menentukan jenis-jenis penyakit yang dapat menyerang ternak ayam disertai gejala yang ditimbulkannya. Tipe pelacakan dan penalaran ditentukan dengan menggunakan mekanisme *Forward Chaining* (Witari, Gandhiadi & Kencana, 2013) dimana diawali dari fakta-fakta yang diberikan *user* kemudian dicari dibasis pengetahuan lalu dicari *rule* yang sesuai dengan fakta-fakta, langkah selanjutnya diadakan hipotesa untuk memperoleh suatu kesimpulan. Oleh karena itu, untuk setiap jenis penyakit dan gejala yang ditimbulkan dibuat kode dengan P1.Pn serta G1.Gn. Setiap gejala diberikan bobot sesuai dengan referensi yang didapat. Proses penelitian tersebut Sistem Pakar dikembangkan dengan menggunakan bahasa LIPS ataupun Prolog (Rohajawati & Supriyati, 2010) dan konstruksi *tools language* yang digunakan pada penelitian tersebut adalah PHP yang ditujukan bagi kemudahan dalam penyebarluasan melalui website untuk penggunaanya.

Dalam penelitian yang dilakukan oleh Iskandar (2007) berbeda karena peneliti menggunakan sejumlah *form* untuk menginput data. Menggunakan metode perancangan sistem dengan menentukan *input* dan *output* berdasarkan data yang diperoleh bertujuan untuk pembuatan model dan program yang menyakut hal-hal penetapan *input* dan *output* serta implementasi sistem yang merupakan presentasi dan visualisasi hasil (*running program*) yang bertujuan untuk mengecek kesalah-kesalahan yang mungkin terjadi. Iskandar (2007) menggunakan sepuluh *form*: *form* menu utama, input data pasien, edit data pasien, pencarian data pasien, input data penyakit, data gejala penyakit, konsultasi, laporan pasien perorangan, laporan pasien keseluruhan dan laporan data penyakit untuk mendiagnosa penyakit ISPA. Dengan menggunakan *form-form* tersebut data dan hasil yang diperoleh sesuai dengan keakuratan yang dihendaki oleh pasien.

Menurut penelitian dari Witari, Gandhiadi, dan Kencana (2013), adalah penyakit menular yang diderita oleh anjing dapat ditanggulangi dini dan mengurangi dampak buruk bagi manusia. Media yang digunakan adalah media *android*, karena *android* sistem operasi yang dapat ditambahkan aplikasi secara bebas dan terdapat pada beberapa jenis

ponsel.. Witari, Gandhiadi, dan Kencana (2013) menggunakan Pohon Keputusan yang merupakan grafik yang akan menjelaskan antara objek-objek yang dihubungkan dengan garis-garis berlabel (“Ya” atau “Tidak) untuk menguji kepercayaan anjing yang didiagnosa menderita penyakit jamur atau tidak.

Menurut Rohajawati dan Supriyati (2010), aplikasi-aplikasi dengan menggunakan metode Faktor Kepastian yang dapat memberikan hasil yang akurat dari perhitungan bobot yang diberikan dan dikalkulasikan berdasarkan fakta-fakta yang muncul sebagai gejala. *Certainty Factor* (CFs) ini dengan kombinasi rule yang lebih kompleks sehingga kompleksitas diagnosis dapat memberikan hasil yang lebih memuaskan. CFs digunakan untuk menangani ketidakpercayaan dalam knowledge based systems yaitu dengan cara menggunakan 1.0 atau 100 untuk kepercayaan absolut (keyakinan penuh) dan 0 untuk kesalahan yang pasti. Iskandar (2007) menyimpulkan bahwa metode *Certainty Factor* merupakan metode yang dapat dikembangkan untuk mencapai keakuratan karena pasien sangat memerlukan keakuratan untuk mendiagnosa penyakit yang dialaminya.

Berdasarkan penelitian-penelitian di atas, hasil atau data yang diperoleh dapat memberikan suatu kepercayaan dan keakuratan pada pasien namun, untuk menjaga dan memelihara keakuratan maka perlu diadakan pengecekan data secara berkala dan untuk mengembangkan penelitian tersebut perlu seorang programmer yang dapat melakukan riset baik melalui seorang ahli yang sudah menguasai di bidang kesehatan atau studi pustaka. Dari ketiga penelitian di atas, penelitian yang dilakukan oleh Iskandar (2007) merupakan rancangan yang paling sederhana dengan berbasis *clientserver* sehingga penelitian tersebut masih dapat dikembangkan untuk mencapai keakuratan data.

2.2 Landasan Teori

2.2.1 Sistem Pakar

Sistem pakar secara umum adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti biasa yang dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Ada beberapa definisi mengenai sistem pakar antara lain yaitu 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 (Martin dan Oxman, 1998), 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).

Arsitektur sistempakar dapat dilihat pada gambar 2.1, pada gambar tersebut dapat dikatakan bahwa sistem pakar memiliki beberapa komponen yaitu :

1. Knowledge acquisition subsystem
2. Knowledge base
3. Inference engine
4. Blackboard (workplace)
5. User
6. User Interface
7. Explanation subsystem (justifier)
8. Knowledge refining system

Pada Umumnya expert system tidak termasuk komponen *knowledge refinement*. Terdapat pula variasi-variasi besar di dalam konten dan kemampuan setiap komponen. Pada Gambar 2.1 menunjukkan hubungan antara komponen-komponen. Berikut adalah penjelasan dari komponen-komponen yang diberikan pada Gambar 2.1 :

1. Knowledge Acquisition Subsystem

Knowledge Acquisition adalah akumulasi, transfer, dan transformasi dari pemecahan masalah kepakaran dari beberapa sumber pengetahuan (knowledge) ke sebuah program komputer untuk mengkontruksi atau memperluas basis pengetahuan (knowledge base) . Sumber potensial dari pengetahuan yaitu pakar, buku-buku, basis data, laporan ilmiah khusus, dan gambar –gambar.

2. Knowledge Base

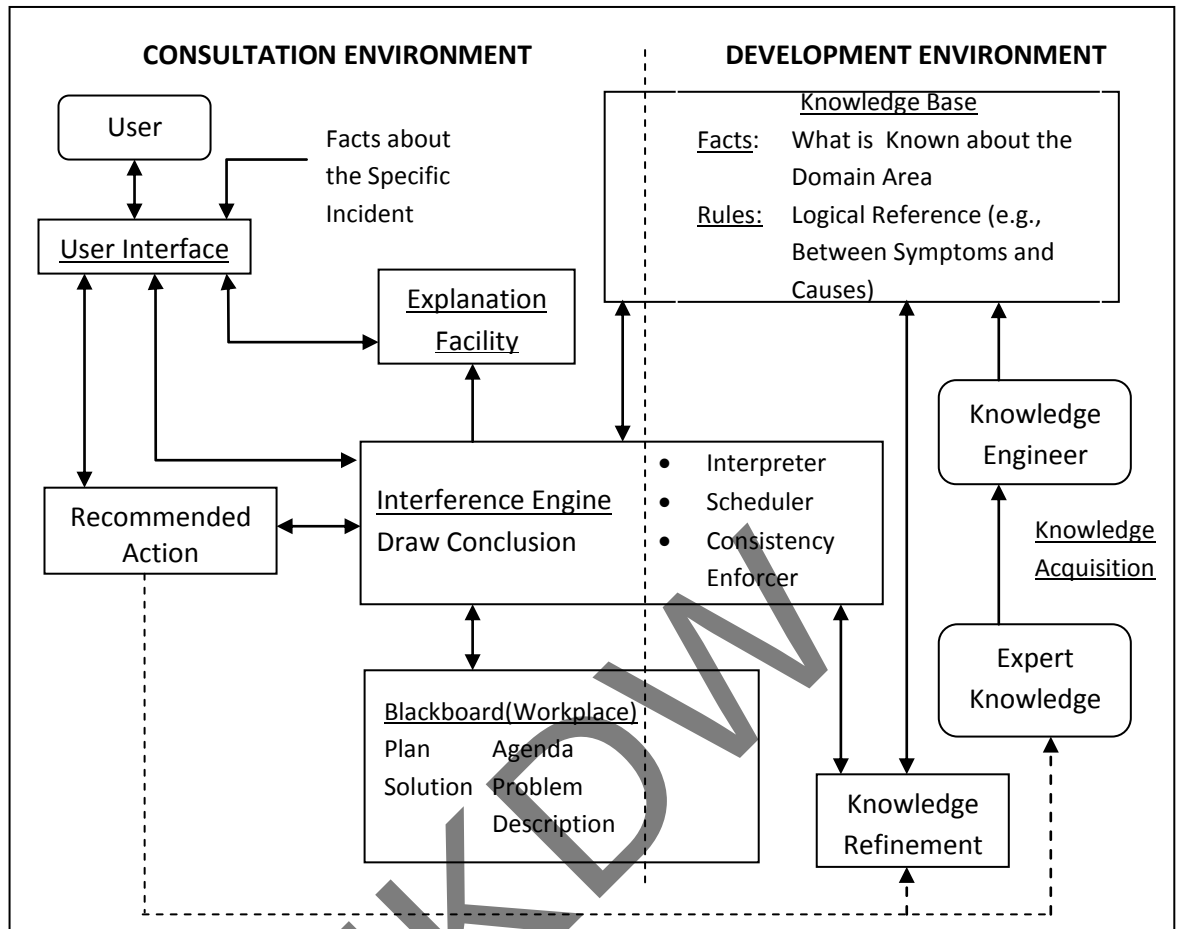
Knowledge base (basis pengetahuan) mengandung pengetahuan yang diperlukan untuk memahami, merumuskan, dan menyelesaikan masalah. Hal tersebut termasuk dua dasar elemen yaitu 1) fakta, contohnya situasi masalah dan teori dalam area permasalahan dan 2) heuristik khusus atau aturan yang secara langsung penggunaan pengetahuan untuk menyelesaikan masalah-masalah yang spesifik di dalam dominan tertentu. Informasi di dalam basis pengetahuan tergabung ke dalam sebuah program komputer oleh sebuah proses yang dinamakan representasi pengetahuan.

3. Knowledge Base

Knowledge base (basis pengetahuan) mengandung pengetahuan yang diperlukan untuk memahami, merumuskan, dan menyelesaikan masalah. Hal tersebut termasuk dua dasar elemen yaitu 1) fakta, contohnya situasi masalah dan teori dalam area permasalahan dan 2) heuristik khusus atau aturan yang secara langsung penggunaan pengetahuan untuk menyelesaikan masalah-masalah yang spesifik di dalam dominan tertentu. Informasi di dalam basis pengetahuan tergabung ke dalam sebuah program komputer oleh sebuah proses yang dinamakan representasi pengetahuan.

4. Inference Engine

inference engine (mesin inferensi) adalah sebuah program komputer yang memberikan sebuah metodologi untuk penalaran tentang informasi di dalam basis pengetahuan dan di dalam “blackboard” dan untuk memformulasikan kesimpulan-kesimpulan. Komponen ini memberikan arahan-arahan tentang bagaimana menggunakan sistem pengetahuan dengan mengembangkan agenda yang mengorganisasikan dan mengontrol langkah-langkah yang diambil untuk menyelesaikan masalah masalah kapanpun konsultasi berlangsung.



Gambar 2.1 Struktur Sistem Pakar (Turban, 1995)

Di dalam Inference Engine terdapat 3 elemen penting:

- Interpreter (aturan umum interpreter) yaitu, menjalankan bagian agenda yang terpilih dengan mengaplikasikan aturan-aturan yang sesuai basis pengetahuan
- Scheduler yaitu, mengambil alih kontrol agenda. Hal tersebut dapat memperkirakan efek dari mengaplikasikan aturan inference di dalam prioritas item atau kriteria lainnya dalam agenda
- Consistency enforcer yaitu, upaya mempertahankan representasi / perwakilan yang konsisten di dalam solusi-solusi yang muncul.

5. Blackboard (Workplace)

The Blackboard adalah sebuah area memori yang bekerja untuk menyisihkan untuk deskripsi permasalahan yang ada akhir-akhir ini, seperti yang dispesifikasikan oleh input data. Hal ini juga digunakan untuk merekam peralihan hasil akhir. The Blackboard merekam peralihan hipotesis dan keputusan. Tiga tipe dari keputusan yang dapat terekam oleh blackboard: a. *plan* – bagaimana untuk menyerang masalah, b. *agenda* – langkah-langkah potensial menunggu eksekusi dan, c. *solution* – kandidat hipotesis dan langkah pelatihan alternatif yang ditimbulkan oleh sistem sejauh ini. Penggunaan blackboard menjadi terkenal pada saat beberapa pakar terbentuk untuk menyelesaikan satu masalah.

6. User Inference

Sistem pakar (*Expert Systems*) terdapat sebuah perangkat bahasa untuk ramah, berorientasi pada masalah komunikasi antara pengguna atau user dan komputer. Komunikasi ini dapat menjadi sebuah kelancaran di dalam bahasa yang bersifat natural, dan di beberapa kasus hal ini dapat disuplementasikan oleh menu dan grafik.

7. Explanation Subsystem (Justifier)

Kemampuan untuk melacak pertanggung jawaban untuk kesimpulan-kesimpulan terhadap sumber-sumber adalah hal yang penting baik dalam mentransfer keahlian dan menyelesaikan masalah. Penjelasan bagian sistem dapat melacak sebuah pertanggung jawaban dan menjelaskan sikap ES dengan secara interaktif menjawab pertanyaan-pertanyaan sebagai berikut:

- a) Mengapa sebuah pertanyaan tertentu ditanyakan oleh sistem pakar?
- b) Bagaimana sebuah kesimpulan tertentu dapat diraih?
- c) Mengapa sebuah alternative tertentu ditolak?
- d) Apa rencana untuk mencapai sebuah solusi? Seperti halnya, Apa yang tetap didirikan sebelum final diagnosis ditentukan?

8. Knowledge Refining System

Pakar manusia memiliki sebuah pengetahuan memperbaiki sistem yang dapat menganalisa kinerja sendiri, belajar dari kinerja tersebut, dan meningkatkan kinerja tersebut untuk konsultasi masa depan. Evaluasi juga merupakan kebutuhan di dalam komputerisasi sehingga program dapat menganalisa alasan-alasan untuk mencapai kesuksesan atau kegagalan. Hal tersebut dapat menjadi sebuah peningkatan terhadap hasil di dalam sebuah basis pengetahuan yang lebih baik dan beralasan yang lebih efektif. Seperti sebuah komponen yang tidak ada dalam sistem pakar komersial pada saat itu, tapi sistem pakar itu sendiri dapat berkembang dalam eksperimen ES dalam beberapa universitas-universitas dan instansi-instansi percobaan ilmiah.

2.2.2 Faktor Kepastian(*Certainty Factor*)

Faktor Kepastian merupakan cara dari penggabungan kepercayaan (*belief*) dan ketidakpercayaan (*unbelief*) dalam bilangan yang tunggal. Dalam *certainty theory*, data-data kualitatif direpresentasikan sebagai derajat keyakinan (*degree of belief*) (Arhami, 2005). Didalam faktor kepastian ada tahapan dalam merepresentasikan data-data kualitatif, yaitu:

- a. Kemampuan untuk mengekspresikan derajat keyakinan sesuai dengan metode yang sudah dibahas sebelumnya.
- b. Kemampuan untuk menempatkan dan mengkombinasikan derajat keyakinan tersebut dalam sistem pakar.

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 atau rumus dasar *certainty factor* (Giarrantano & Riley, 2005):

$$CF[h, e] = MB[h, e] - MD[h, e] \quad [2.1]$$

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 ketidakyakinan), 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:

$$MB[h, e1 \wedge e2] = \begin{cases} 0 & MB[h, e1 \wedge e2] = 1 \\ MB[h, e1] + MB[h, e2] \cdot (1 - MB[h, e1]) & \end{cases} \quad [2.2]$$

$$MD[h, e1 \wedge e2] = \begin{cases} 0 & MB[h, e1 \wedge e2] = 1 \\ MD[h, e1] + MD[h, e2] \cdot (1 - MD[h, e1]) & \end{cases} \quad [2.3]$$

2. Perhitungan CF dengan Multiple Antecedent

Eidence, E	Antecedent Certainty
E ₁ AND E ₂	min [CF(H,E ₁), CF(H,E ₂)]
E ₁ OR E ₂	max [CF(H,E ₁), CF(H,E ₂)]
NOT E	-CF(H,E)

[2.4]

$$3. CF_{kombinasi} (CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{keduanya} > 0 \\ \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & \text{salah satu} < 0 \\ CF_1 + CF_2(1 + CF_1) & \text{keduanya} < 0 \end{cases} \quad [2.5]$$

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

Berdasar kepada rumus *Certainty Factor* yang sudah dituliskan diatas, berikut adalah beberapa contoh perhitungan yang diterapkan :

Penyakit pada kelinci cukup banyak, untuk studi kasus ini saya mengambil salah satu contoh penyakit yaitu (P11)Pinworm(Oxyuris) dengan gejala (G2)badan kurus, (G52) Gatal di sekitar lubang dubur, (G53) Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya , (G54) Kotoran mata lebih banyak dari biasanya dan berwarna kuning. Kemudian didalam diagnosanya diberikan beberapa aturan yaitu :

R1 = IF G2 AND G52 THEN P11 CF1 pakar= 0.5

R2 = IF G2 AND G52 AND G53 THEN P11 CF2 pakar=0.7

R3 = IF G2 AND G52 AND G53 AND G54 THEN P11 CF3 pakar=0.8

Pada kasus pertama ada seekor kelinci yang didiagnosa dokter dengan gejala (G2)badan kurus dengan memberikan nilai kepastian 0.6 dan (G52)gatal disekitar lubang dubur 0.7 dengan indikasi mengalami penyakit Pinworm. Dari data tersebut dapat dihitung nilai kepastiannya sebagai berikut :

$$\begin{aligned} CF &= \text{Min}(G1, G2) \times CF \text{ Pakar} \\ &= \text{Min}(0.6, 0.7) \times 0.6 \\ &= 0.6 \times 0.6 \\ &= 0.36 \end{aligned}$$

Maka, kelinci yang didiagnosa tersebut mengalami sakit yang bernama Pinworm dengan nilai kepastian (CF) sebesar 0.36 .

Pada kasus yang kedua ada seekor kelinci yang didiagnosa dokter dengan gejala (G2)badan kurus dengan nilai 0.7, (G52) Gatal di sekitar lubang dubur dengan nilai 0.6, (G53) Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya dengan nilai 0.7 , (G54) Kotoran mata lebih banyak dari biasanya dan berwarna kuning dengan nilai 0.8 dengan indikasi mengalami penyakit Pinworm. Mengacu pada aturan yang sudah diberikan diatas melalui R1, R2 dan R3, maka masuk pada aturan R3, nilai CF nya dapat dihitung sebagai berikut :

$$\begin{aligned} CF R1 &= \text{min}(G1, G2) \times CF \text{ pakar R1} \\ &= 0.6 \times 0.5 = 0.3 \end{aligned}$$

$$\begin{aligned} CF R2 &= \text{min}(G1, G2, G3) \times CF \text{ pakar R2} \\ &= 0.6 \times 0.7 = 0.42 \end{aligned}$$

$$\begin{aligned} CF R2 &= \min(G1, G2, G3, G4) \times CF \text{ pakar } R3 \\ &= 0.6 \times 0.8 = 0.48 \end{aligned}$$

Karena menggunakan nilai antara 0 – 1 maka menggunakan rumus :

$$CF_{gab} = CF1 + CF2 (1 - CF1)$$

Maka :

CF gabungan R1 dan R2

$$\begin{aligned} CF_{gab1} &= CF R1 + CF2 R2(1 - CF R1) \\ &= 0.3 + 0.42 (1 - 0.3) \\ &= 0.3 + 0.42 (0.7) \\ &= 0.3 + 0.29 \\ &= 0.59 \end{aligned}$$

$$\begin{aligned} CF_{gab2} &= CF_{gab1} + CF R3 (1 - CF_{gab1}) \\ &= 0.59 + 0.48 (1 - 0.41) \\ &= 0.59 + 0.48 (0.41) \\ &= 0.59 + 0.2 \\ &= 0.79 \end{aligned}$$

Maka, kelinci yang didiagnosa tersebut mengalami sakit yang bernama Pinworm dengan nilai kepastian(CF) sebesar 0.79 .

2.2.3 Penyakit Pada Kelinci

Kelinci dengan penampilan yang unik, bersih, lucu dan memiliki warna serta corak yang beragam sangat disukai orang, terlebih lagi anak-anak. Selain itu, bulunya yang halus, lunak, sifatnya yang jinak dan mudah dipelihara mendorong orang untuk memilikinya. Kelinci-kelinci dengan kriteria seperti diatas, umumnya dipelihara sebagai ternak atau hewan peliharaan.

Sebagai binatang peliharaan, kelinci mulai banyak diminati sejak tahun 2003. Beberapa *pet shop* di seluruh kota di Indonesia menjual berbagai jenis anakan kelinci. Pamor kelinci sebagai binatang mulai naik sejak digelarnya

beberapa kontes. Prospek kelinci masih terus berkembang, terutama untuk menernakannya. Selain untuk binatang peliharaan, kelinci seringkali dijadikan sebagai konsumsi dan dapat juga sebagai bahan percobaan di laboratorium.

Kelinci yang sehat adalah dambaan setiap pemeliharanya. Untuk itu diperlukan perawatan yang teratur, termasuk pemeriksaan kondisi kesehatan tubuhnya. Selain itu pendeteksian penyakit sejak dini juga sangat penting dan sangat diperlukan untuk menentukan langkah selanjutnya. Oleh karena itu, para pemelihara kelinci harus mengetahui penyakit yang umumnya diderita oleh kelinci, sehingga dapat melakukan pertolongan 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 infestasi 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 (Pasteurellosis/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. Ring Worm (Jamur)

Ringworm bukan penyakit cacingan, 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 lender 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. Gejala-gejalanya 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. Gejala-gejalanya adalah perut membesar, susah bernafas dan badan ambruk.

2.2.4 Android

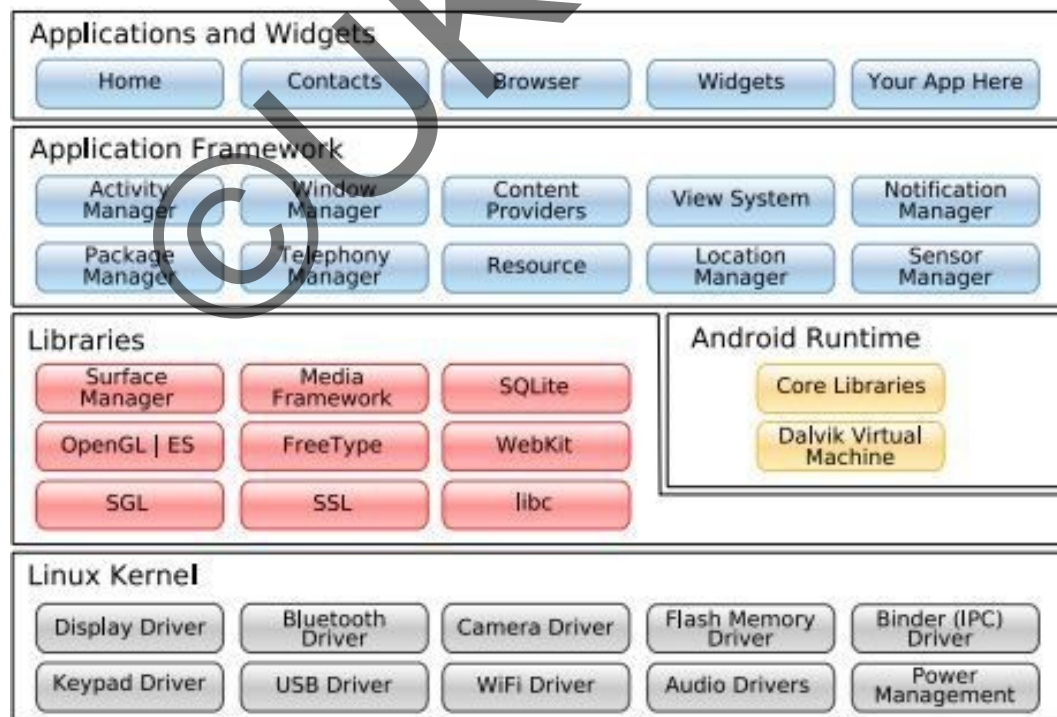
Android merupakan sistem operasi yang dibangun dengan berbasis Linux, sistem operasi android pada dasarnya ditujukan untuk berjalan dengan *hardware smartphone* ataupun tablet. Dalam pengembangan Android menggunakan sistem operasi yang terbuka.

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. (John Wiley dan Sons, 2010). Gambar arsitektur sistem operasi android dapat dilihat pada gambar 2.2.

Dalam arsitektur android tersebut 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 2.2 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.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Deskripsi Sistem

Sistem ini adalah sebuah sistem perangkat lunak yang digunakan untuk mendiagnosa penyakit pada kelinci. Dalam sistem diagnosa penyakit kelinci ini pengguna dapat berkomunikasi dengan sistem melalui pertanyaan-pertanyaan yang berupa fakta-fakta yang telah disimpan dalam sistem berupa basis pengetahuan. Hasil jawaban dari pertanyaan-pertanyaan tersebut akan memunculkan penyakit yang mungkin terjadi beserta saran pengobatannya.

Sesuai dengan struktur sistem pakar secara umum, struktur sistem pakar yang digunakan dalam mendiagnosa penyakit pada hewan peliharaan terdiri dari dua lingkungan kerja, yaitu lingkungan pengembangan dan lingkungan konsultasi.

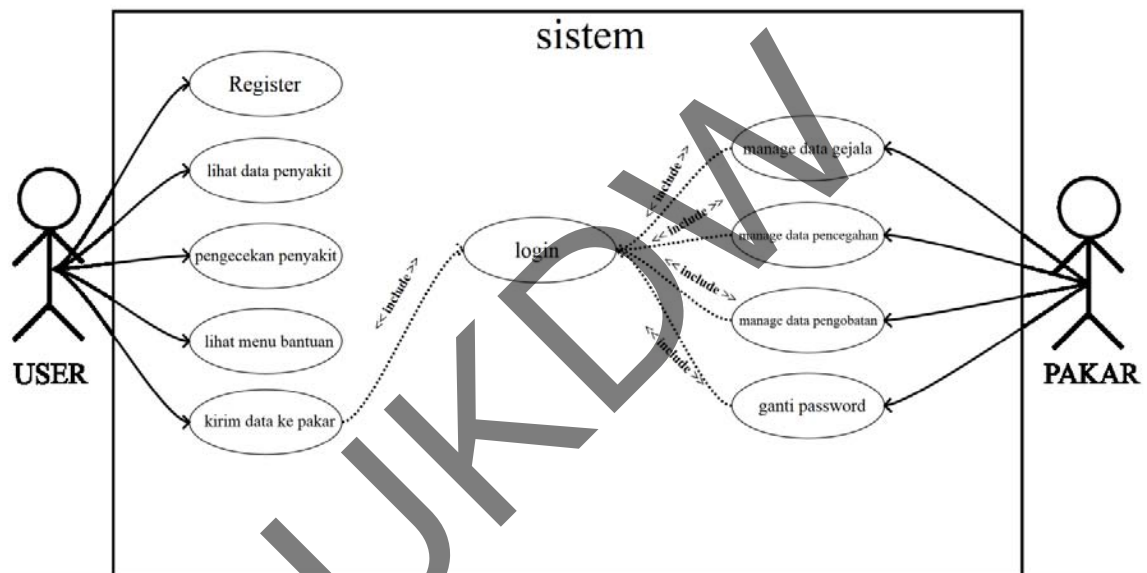
Pada lingkungan pengembangan digunakan untuk memasukkan pengetahuan dari pakar dan melakukan perbaikan pengetahuan. Sedangkan pada lingkungan konsultasi merupakan lingkungan kerja yang digunakan untuk pengguna dalam memperoleh informasi.

Sistem Pakar akan dibangun berbasis android dengan menggunakan bahasa pemrograman *Java* dan untuk pengolahan data menggunakan *web service* dengan menggunakan bahasa pemrograman PHP. Berikut ini adalah kebutuhan fungsional dan non – fungsional :

1. Kebutuhan Fungsional
 - a. Sistem dapat menampilkan penyakit kelinci
 - b. Sistem dapat melakukan pengecekan berdasar gejala yang dipilih dan menampilkan hasilnya.
 - c. Hasil dari pengecekan berupa nama penyakit dan saran pencegahan beserta pengobatan

- d. Pengguna dapat mengirimkan usulan data baru jika penyakit tidak ditemukan
- e. Use case

Sistem yang akan dibangun ada 2 rancangan, yaitu sistem aplikasi di android dan sistem web service. Pengguna merupakan aktor yang akan menggunakan sistem diagnosa penyakit kelinci. Aplikasi Android adalah program yang akan digunakan oleh pengguna (*user*).



Gambar 3.1 Diagram *Use Case*

- User:
 - 1) Register (Registrasi) adalah proses dimana user membuat data pribadi dan mendaftarkan di sistem
 - 2) Melihat data penyakit berupa gejala-gejala penyakit yang akan dipilih, pencegahan, dan pengobatan.
 - 3) Melakukan pengecekan gejala dengan memilih gejala yang telah tersedia.
 - 4) Melihat menu bantuan yaitu user dapat mengakses menu bantuan apabila mengalami kesulitan dalam menggunakan aplikasi.

- 5) Mengirimkan data ke pakar, yaitu *user* dapat memberikan usulan data ke pakar dan sebelumnya harus login terlebih dahulu.

- Admin:

- 1) Mengolah data gejala yaitu admin memasukkan data-data gejala penyakit kelinci ke dalam basis data agar dapat dikonsumsi *user*.
- 2) Mengolah data pencegahan yaitu admin memasukkan data-data pencegahan ke dalam basis data agar dapat dikonsumsi *user*.
- 3) Mengolah data pengobatan yaitu admin memasukkan data-data pengobatan ke dalam basis data agar dapat dikonsumsi *user*.
- 4) Mengganti password
- 5) Logout
- 6) Mereview data test yaitu admin melihat data yang diusulkan oleh *user* kemudian mengkonsultasikan dengan pakar, apakah dapat menjadi sebuah gejala atau penyakit baru dan mungkin cara pencegahan serta pengobatan yang baru.
- 7) Login

2. Spesifikasi Hardware dan Software

Berikut ini adalah spesifikasi hardware dan software dari sisi server :

1. Processor : Intel core I5
2. Memori : 4GB
3. Kartu grafis : Intel HD Graphic
4. Media penyimpanan : 640GB
5. Layar : 14" HD LED
6. Web server : Apache versi 2.2 dalam XAMPP versi 1.8.1
7. Bahasa pemrograman : PHP versi 5.4.7
8. Database : MySQL
9. Browser : Google Chrome version 25.0.1364.152

10. OS : Windows 7 Ultimate 64bit

Berikut ini adalah spesifikasi hardware dan software dari sisi android :

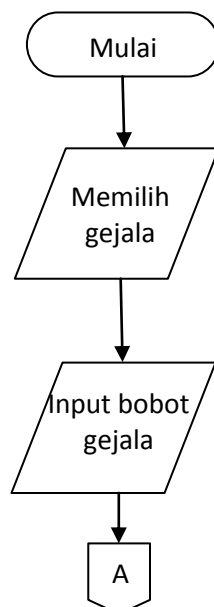
1. Model : GT-N7100 (Samsung Galaxy Note II)
2. Processor : 1.6 GHz quad-core processor
3. OS : Android 4.4.2 (KitKat)
4. Layar : 140.9 mm (5.5") HD Super AMOLED (1,280 x 720)
5. Memori : 16GB User memory + 2GB (RAM)
6. Emulator : Eclipse 3.8

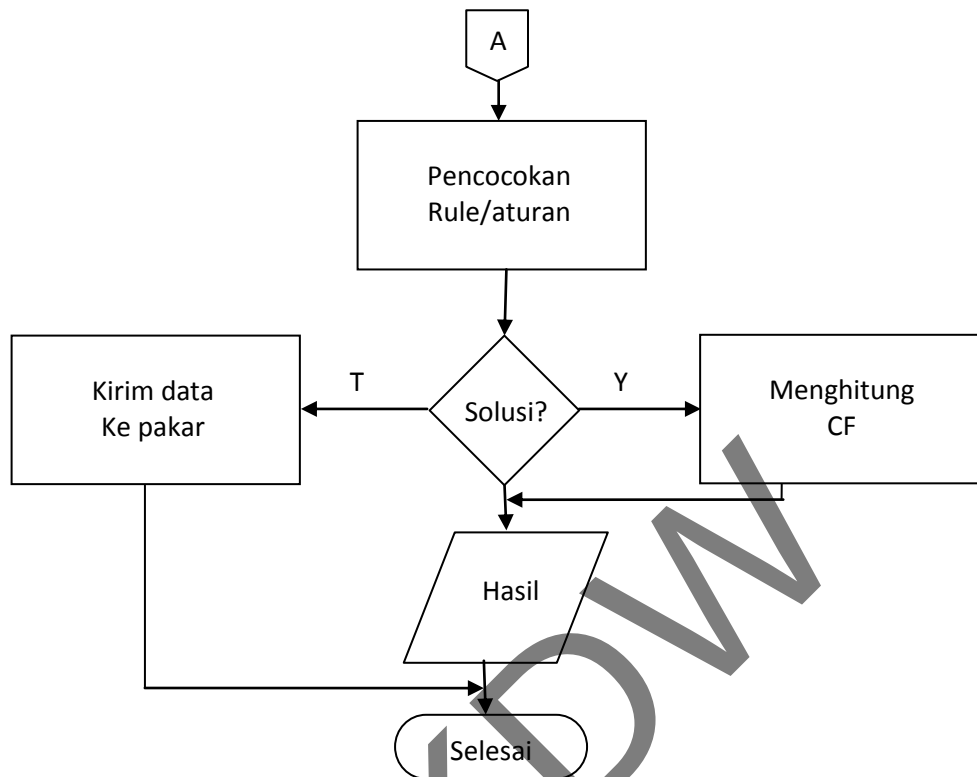
3.2. Algoritma dan Flowchart

3.2.1 Flowchart Keseluruhan Sistem

Pada proses awal seorang *user* memilih gejala yang telah disediakan oleh aplikasi yang dimulai dari gejala umum dan berlanjut ke gejala pendukung, kemudian *user* memasukkan bobot dari setiap gejala yang telah dipilih dengan ketentuan nilai dari 0.1 sampai 0.9 kemudian cek penyakit, maka sistem akan melakukan pengecekan terhadap penyakit yang ada sesuai dengan aturan yang telah dibuat pada sistem. Jika terdapat solusi, maka sistem akan menghitung nilai CF(*certainty factor*) dari penyakit yang ditemukan agar diketahui seberapa nilai kepercayaan penyakit tersebut.

Kemudian jika solusi tidak ditemukan, *user* akan diarahkan untuk mengirimkan data ke pakar.

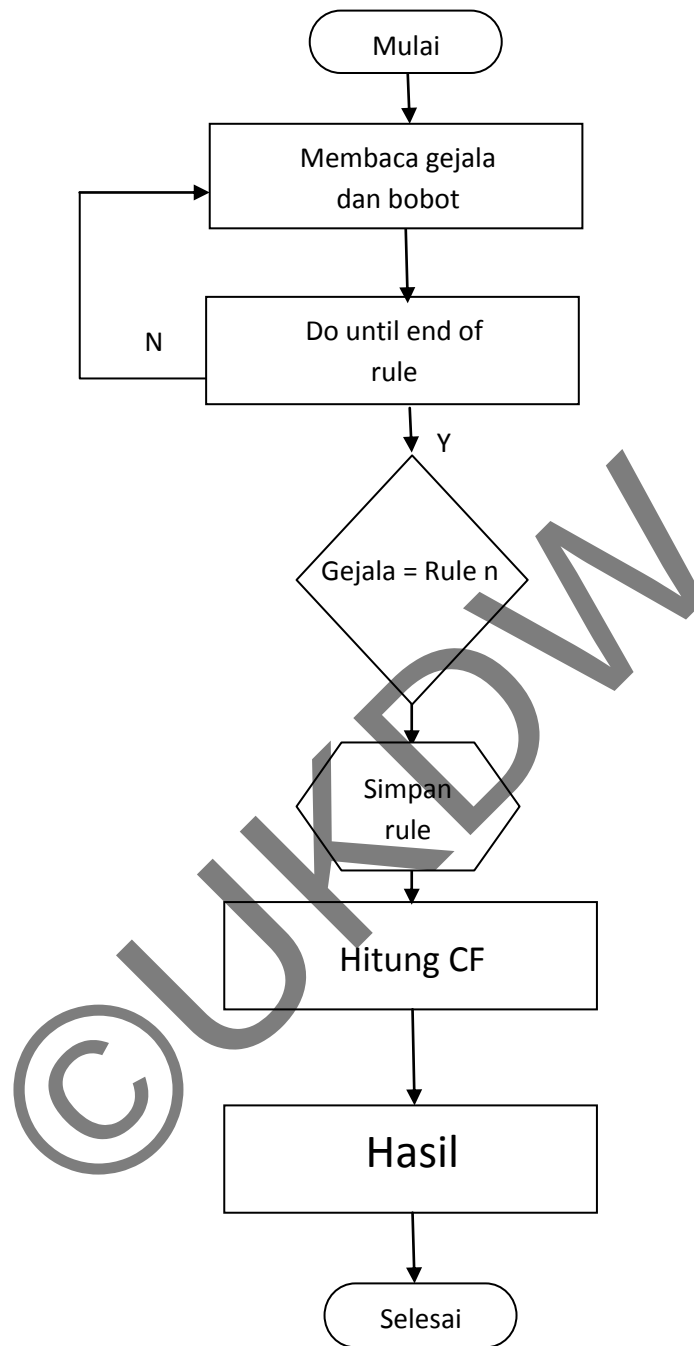




Gambar 3.2 Flowchart Keseluruhan Sistem

3.2.2 Flowchart Forward Chaining

Pada proses forward chaining, sistem membaca gejala yang dipilih oleh user beserta dengan bobot pada setiap gejalanya. Setelah terbaca oleh sistem, maka sistem akan melakukan pencocokan secara progresif yaitu mencocokkan dengan aturan(rule) yang telah dibuat di dalam sistem sampai pada iterasi dimana rule itu berakhir. Jika gejala dan bobot yang di inputkan berada pada satu rule atau lebih, maka sistem akan berhenti mengecek dan kemudian menghitung nilai CF(certainty factor)nya dan akan muncul hasil nya berupa penyakit dan nilai CF-nya, apabila sampai rule ke-n tidak didapatkan kecocokan maka sistem akan berhenti dan kemudia user diarahkan kepada sebuah halaman yang dmakan user akan melaporkan bahwa gejala yang dipilih tidak terdapat solusinya kemudian dikirim ke pakar untuk di evaluasi.



Gambar 3.3 Flowchart Forward Chaining

3.2.3 Aturan (Rule) Penyakit Kelinci

Tabel 3.1 Tabel Rule Penyakit Kelinci

NO	Rule	CF
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 AND G44 THEN P2	0.9
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

3.3 Kamus Data

3.3.1 Tabel Relasi Diagnosa Penyakit Kelinci

Tabel Penyakit adalah tabel yang digunakan untuk menyimpan data penyakit kelinci.

Tabel 3.2 Tabel penyakit

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

3.3.2 Tabel penyakit_gejala

Tabel penyakit_gejala adalah tabel yang digunakan untuk menyimpan data gejala dari tabel penyakit dengan ditambahkan nilai MB dan MD.

Tabel 3.3 Tabel penyakit_gejala

Nama field	Tipe Data	Keterangan
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)

3.3.3 Tabel gejala

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

Tabel 3.4 Tabel gejala

Nama field	Tipe Data	Keterangan
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)

3.3.4 Tabel Tes Entry

Tabel tes *entry* adalah tabel yang berisi gejala-gejala yang di pilih oleh *user* pada saat melakukan tes.

Tabel 3.5 Tabel tes entry

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)

3.3.5 Tabel tes

Tabel tes adalah tabel untuk menyimpan data tes yang dilakukan oleh *user*.

Tabel 3.6 Tabel tes

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)

3.3.6. Tabel pelakutes

Tabel pelakutes adalah tabel untuk menyimpan data-data *user* yang teregistrasi di sistem.

Tabel 3.7 Tabel pelakutes

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)

3.3.7. Tabel penyakit_pencegahan

Tabel penyakit_pencegahan adalah tabel yang digunakan untuk menyimpan data pencegahan dari tabel pencegahan.

Tabel 3.8 Tabel penyakit_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)
----------------------------	----------	--

3.3.8. Tabel pencegahan

Tabel Pencegahan adalah tabel yang digunakan untuk menyimpan data pencegahan dari penyakit kelinci.

Tabel 3.9 Tabel pencegahan

Nama field	Tipe Data	Keterangan
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)

3.3.9. Tabel penyakit_pengobatan

Tabel penyakit_pengobatan adalah tabel yang digunakan untuk menyimpan data pengobatan dari tabel pengobatan

Tabel 3.10 Tabel penyakit_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)

3.3.10. Tabel pengobatan

Tabel Pengobatan adalah tabel yang digunakan untuk menyimpan data pengobatan dari penyakit kelinci.

Tabel 3.11 Tabel pengobatan

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)

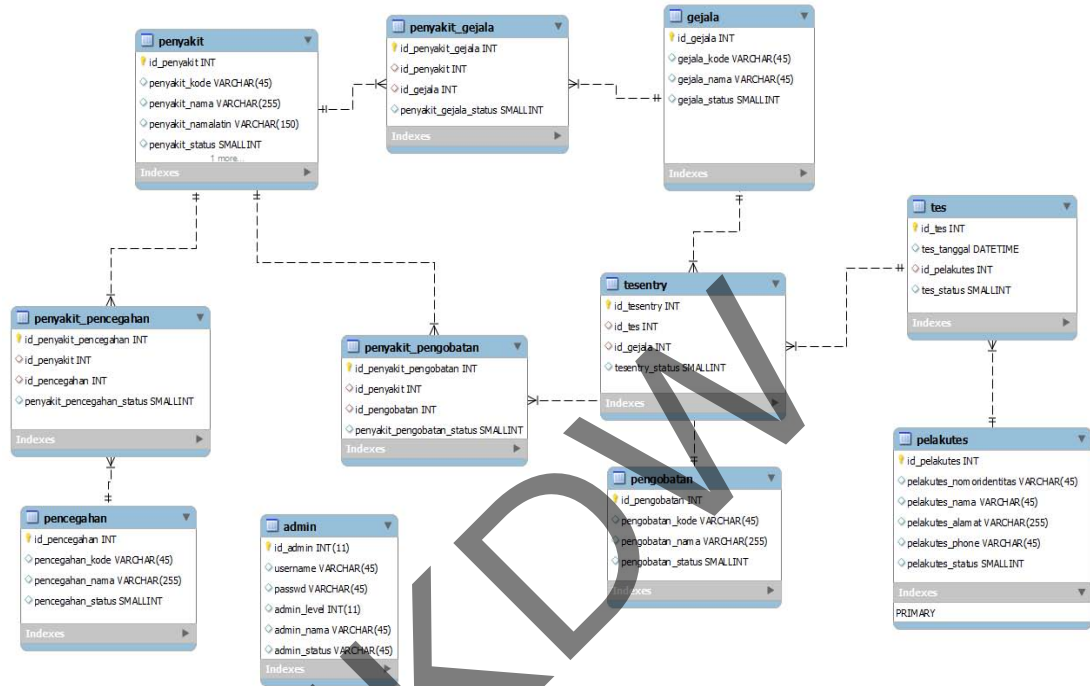
3.3.11. Admin

Tabel Admin adalah tabel yang digunakan untuk menyimpan data pakar berupa *username* dan *password* yang nantinya digunakan untuk *login* ke sistem.

Tabel 3.12 Tabel Admin

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

3.4 Diagram Skema



Gambar 3.4 Diagram Skema

3.5 Rancangan Antarmuka Sistem

3.5.1. Rancangan Antarmuka *Web Service*

Pada Gambar 3.5 di bawah ini merupakan rancangan halaman *login* pakar. Halaman ini berfungsi sebagai keamanan sistem sehingga tidak semua orang dapat menggunakan sistem ini. Hak akses hanya diberikan kepada pakar saja. Jika *login* berhasil, maka akan menuju ke halaman utama pakar.

A simple login form with two input fields labeled 'username' and 'password', and a 'login' button below them.

Gambar 3.5 Halaman Login Pakar

The 'Form Data Gejala' page features a navigation bar with links: Administrator, Gejala, Penyakit, Pencegahan, Pengobatan, Pelaku Tes, Ganti Password, and Logout. The form includes two input fields: 'Gejala Kode' and 'Gejala Nama', followed by a 'Simpan' button. A link labeled 'kembali ke data gejala' is positioned below the form.

Gambar 3.6 Form Data Gejala

Pada Gambar 3.6 di bawah ini merupakan rancangan halaman Form Data Gejala. Pada halaman ini pakar dapat memasukkan data-data gejala yang nantinya akan diolah. Pada form ini berisikan kode gejala sebagai inisialisasi dan nama gejalanya.

The 'Daftar Data Gejala' page displays a table with the following data:

NO	kode gejala	nama gejala	aksi
99	xxx	aaa	Perbaharui Hapus
99	xxx	bbb	Perbaharui Hapus
-	-	-	-
-	-	-	-
99	xxx	zzz	Perbaharui Hapus

Below the table, there is a link labeled 'tambah data gejala'.

Gambar 3.7 Daftar Data Gejala

Gambar 3.7 di atas merupakan rancangan halaman Daftar Data Gejala. Pada halaman ini pakar dapat melihat,memperbaharui serta menghapus data yang sudah dimasukkan melalui form data gejala sebelumnya.

Gambar 3.8 Form Data Penyakit

Gambar 3.8 di atas merupakan rancangan halaman Form Data Penyakit. Pada halaman ini pakar dapat memasukkan data-data penyakit yang nantinya akan diolah. Pada form ini berisikan kode penyakit sebagai inisialisasi dan nama penyakit serta nama ilmiah nya. Didalam form ini juga memasukkan nilai MD dan MB untuk perhitungan pada rumus *certainty factor*.

NO	kode penyakit	penyakit nama	penyakit nama ilmiah	aksi
99	xxx	aaa	xxx	Perbaharui Hapus
99	xxx	bbb	xxx	Perbaharui Hapus
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
99	xxx	zzz	aaa	Perbaharui Hapus

[tambah data penyakit](#)

Gambar 3.9 Daftar Data Penyakit

Gambar 3.9 di atas merupakan rancangan halaman Daftar Data Penyakit. Pada halaman ini pakar dapat melihat,memperbaharui serta menghapus data yang sudah dimasukkan melalui form data penyakit sebelumnya.

Gambar 3.10 Form Data Pencegahan

Gambar 3.10 di atas merupakan rancangan halaman Form Data Pencegahan. Pada halaman ini pakar dapat memasukkan data-data pencegahan yang nantinya akan diolah. Pada form ini berisikan kode pencegahan sebagai inisialisasi dan nama pencegahanya.

NO	kode pencegahan	nama pencegahan	aksi
99	xxx	aaa	Perbaharui Hapus
99	xxx	bbb	Perbaharui Hapus
-	-	-	-
99	xxx	zzz	Perbaharui Hapus

Gambar 3.11 Daftar Data Pencegahan

Gambar 3.11 di atas merupakan rancangan halaman Daftar Data Pencegahan. Pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang sudah dimasukkan melalui form data pencegahan sebelumnya.

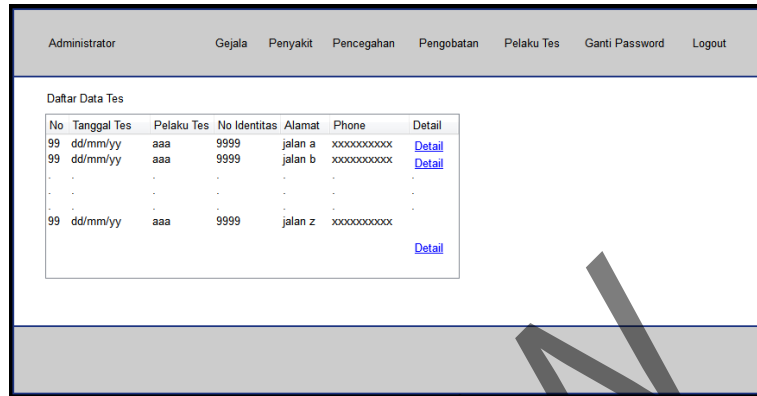
Gambar 3.12 Form Data Pengobatan

Gambar 3.12 di atas merupakan rancangan halaman Form Data Pengobatan. Pada halaman ini pakar dapat memasukkan data-data pengobatan yang nantinya akan diolah. Pada form ini berisikan kode pengobatan sebagai inisialisasi dan nama pengobatannya.

NO	kode pengobatan	nama pengobatan	aksi
99	xxx	aaa	Perbaharui Hapus
99	xxx	bbb	Perbaharui Hapus
.	.	.	.
.	.	.	.
99	xxx	zzz	Perbaharui Hapus

Gambar 3.13 Daftar Data Pengobatan

Gambar 3.13 di atas merupakan rancangan halaman Daftar Data Pengobatan. Pada halaman ini pakar dapat melihat, memperbaharui serta menghapus data yang sudah dimasukkan melalui form data pengobatan sebelumnya.

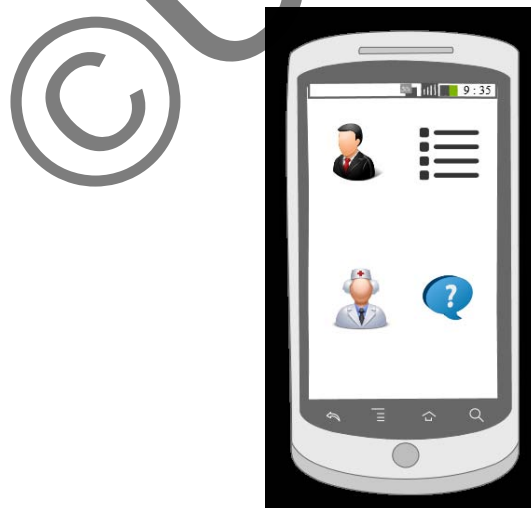


No	Tanggal Tes	Pelaku Tes	No Identitas	Alamat	Phone	Detail
99	dd/mm/yy	aaa	9999	jalan a	xxxxxxxxxx	Detail
99	dd/mm/yy	aaa	9999	jalan b	xxxxxxxxxx	Detail
.	.	.	.	.	.	.
.	.	.	.	.	.	.
99	dd/mm/yy	aaa	9999	jalan z	xxxxxxxxxx	Detail

Gambar 3.14 Daftar Data Tes

Gambar 3.14 di atas merupakan rancangan halaman Daftar Data Tes. Pada halaman ini pakar dapat melihat detail data user yang sudah melakukan tes melalui aplikasi diagnosa penyakit kelinci di android.

3.5.2. Rancangan Antarmuka Android



Gambar 3.15 Halaman Utama

Gambar 3.15 di atas merupakan rancangan antarmuka halamann utama pada aplikasi diagnosa penyakit kelinci di android. Pada halaman diatas terdapat 4 menu utama yaitu menu akun, menu daftar penyakit, menu tes dan menu bantuan.



Gambar 3.16 Halaman Akun

Gambarn 3.16 di atas merupakan rancangan antarmuka halamann akun, pada halaman ini terdapat *login* dan *register*, dimana *user* yang sudah pernah membuat akun akan bisa *login* sesuai yang sudah didaftarkan ke sistem dan kalau belum bisa *register* terlebih dahulu.



Gambar 3.17 Halaman Registrasi

Gambar 3.17 di atas merupakan rancangan antarmuka halamann registrasi untuk pengguna baru yang belum terdaftar di sitem agar bisa *login* pada aplikasi ini.



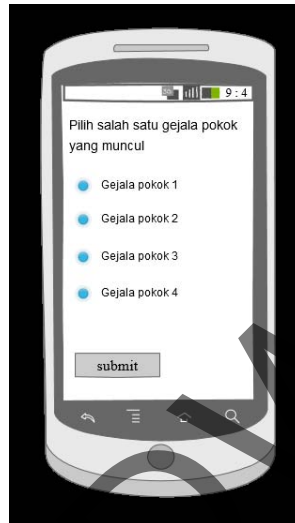
Gambar 3.18 Halaman Login

Pada Gambar 3.18 di bawah ini merupakan rancangan antarmuka halamann login, bagi yang sudah mempunyai akun akan dapat login di aplikasi ini.



Gambar 3.19 Halaman Daftar Penyakit

Gambar 3.19 di atas merupakan rancangan antarmuka halamann daftar penyakit, dimana pada halaman ini seorang user dapat melihat daftar penyakit apa saja yang terdapat pada aplikasi ini.



Gambar 3.20 Halaman Pemilihan Gejala Utama

Gambar 3.20 di atas merupakan rancangan antarmuka halamann pemilihan gejala utama atau gejala pokok.



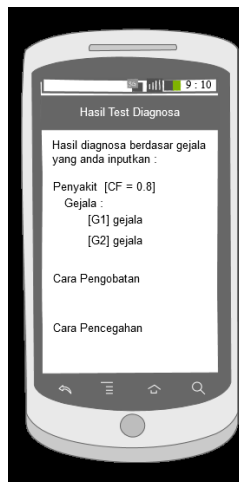
Gambar 3.21 Halaman Pemilihan Gejala Pendukung

Gambar 3.21 di atas merupakan rancangan antarmuka halamann pemilihan gejala pendukung setelah pemilihan gejala pokok.



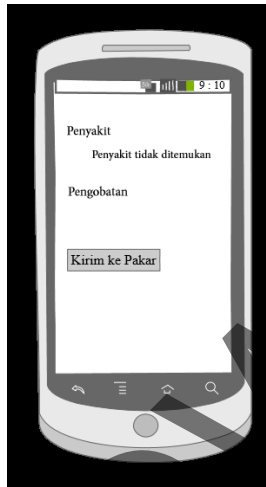
Gambar 3.22 Halaman Input Bobot Gejala

Pada Gambar 3.22 di bawah ini merupakan rancangan antarmuka halaman Input Bobot Gejala yang diperuntukkan mengisi setiap bobot penyakit yang telah dipilih oleh *user* yang kemudian digunakan untuk menghitung nilai CF dari penyakit yang akan muncul.



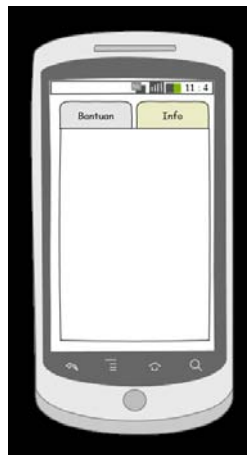
Gambar 3.23 Halaman Hasil Tes

Gambar 3.23 di bawah merupakan rancangan antarmuka halaman hasil tes yang berupa penyakit, pengobatan serta pencegahan setelah user memilih gejala pokok dan pendukung.



Gambar 3.24 Halaman Kirim ke Pakar

Gambar 3.24 di atas merupakan rancangan antarmuka halaman kirim ke pakar. Halaman ini akan muncul apabila penyakit tidak ditemukan ke sistem dan *user* dapat mengirimkan usulan ke pakar mengenai masalah tersebut.



Gambar 3.25 Halaman Bantuan

Gambar 3.25 di atas merupakan rancangan antarmuka halaman bantuan yang terdapat tab bantuan dan tab info. Pada tab bantuan terdapat tuntunan bagaimana cara memakai aplikasi ini, kemudian di tab info terdapat informasi mengenai aplikasi ini.

3.5.3. Rancangan Web Service

Pada perancangan *web service* sistem pakar diagnosa penyakit kelinci ini menggunakan format URL(*uniform resource locator*) sebagai berikut :

```
<protocol>://<portal-domain>/<controller>/<service>/
```

```
URL_SERVICE = "http://andrekusumaa.web.id/android/"
```

Berikut ini adalah detail perancangan web service yang akan digunakan dalam sistem ini :

- 1 http://andrekusumaa.web.id/android/getGejalaUtama

```
request{
```

```
url = <nama activity>.<url> + "getGejalaUtama/"
```

```
}
```

```
response{
```

```
"1": "[ G1] Suhu badan naik",
```

```
"2": "[ G2] Badan kurus"
```

```
}
```

- 2 http://andrekusumaa.web.id/android/ login

```
request{
```

```
url = <nama activity>.<url > + "getGejalaUtama/"
```

```
}
```

```
response{
```

```
"login": "true",
```

```
"data": [
```

```

{
    "id_pelakutes": "23",
    "pelakutes_nomoridentitas": "0",
    "pelakutes_nama": "0",
    "pelakutes_alamat": "0",
    "pelakutes_phone": "0",
    "pelakutes_status": "1",
    "pelakutes_password": "d41d8"
}
]
}
3 http://andrekusumaa.web.id/android/ tesEntry
request{
String url = <nama activity>.<url> + "tesentry"
}
response{
{"result":"true"}
}
4 http://andrekusumaa.web.id/android/ getPenyakit
request{
String url = <nama activity>.<url> + "getPenyakit/"
}
response{
"result": "true",
"data": [
{
    "id_penyakit": "2",
    "penyakit_kode": "P1",
    "penyakit_nama": "Enteritis Kompleks",

```

```

        "penyakit_namalatin": "-",
        "penyakit_cf": "0.4",
        "penyakit_status": "1",
        "j": "11"
    }
}
]
}

```

5 <http://andrekusumaa.web.id/android/getDetail>

```

request{
String url = <nama activity>.<url> + " getDetail/" + <id penyakit>
}
response{
"result": "true",
"data": [
{
    "id_penyakit_gejala": "1",
    "id_penyakit": "1",
    "id_gejala": "1",
    "penyakit_gejala_status": "1",
    "gejala_kode": "G1",
    "gejala_nama": "Suhu badan naik",
    "gejala_status": "1"
}
]
}

```

6 <http://andrekusumaa.web.id/android/register>

```

request{
String url = <nama activity>.<url> + "register"

```

```

    }
    response{
        "register": "true"
    }
}
7 http://andrekusumaa.web.id/android/ getTest/gejalaUtama
request{
    url = <nama activity>.<url> + "getTest/" + <id gejala utama>
}
response{
    "1": "[ G1] Suhu badan naik",
    "2": "[ G2] Badan kurus",
    "3": "[ G3] Sesak nafas atau susah bernafas"
}
8 http://andrekusumaa.web.id/android/cf
request{
    String url = <nama activity>.<url> + "cf"
}
response{
    "result": "true",
    "penyakit": "Cacingan",
    "bobot_minimal": 0.7,
    "penyakit_cf": "0.5",
    "id_penyakit": "45",
    "gejala": [
        "[ G16] Nafsu makan turun",
        "[ G2] Badan kurus",
        "[ G41] Hewan terlihat lemah"
    ],
    "nilai_cf": 0.96568,

```

```

"result_pengobatan": "true",
"data_pengobatan": [
  {
    "id_penyakit_pengobatan": "99",
    "id_penyakit": "45",
    "id_pengobatan": "16",
    "penyakit_pengobatan_status": "1",
    "pengobatan_kode": "OB-15",
    "pengobatan_nama": "Meningkatkan jumlah sayuran atau hijauan yang
mengandung serat tinggi",
    "pengobatan_status": "1"
  }
],
"result_pencegahan": "true",
"data_pencegahan": [
  {
    "id_penyakit_pencegahan": "98",
    "id_penyakit": "45",
    "id_pencegahan": "12",
    "penyakit_pencegahan_status": "1",
    "pencegahan_kode": "PC-11",
    "pencegahan_nama": "Beri pakan yang bergizi",
    "pencegahan_status": "1"
  }
]
}

```

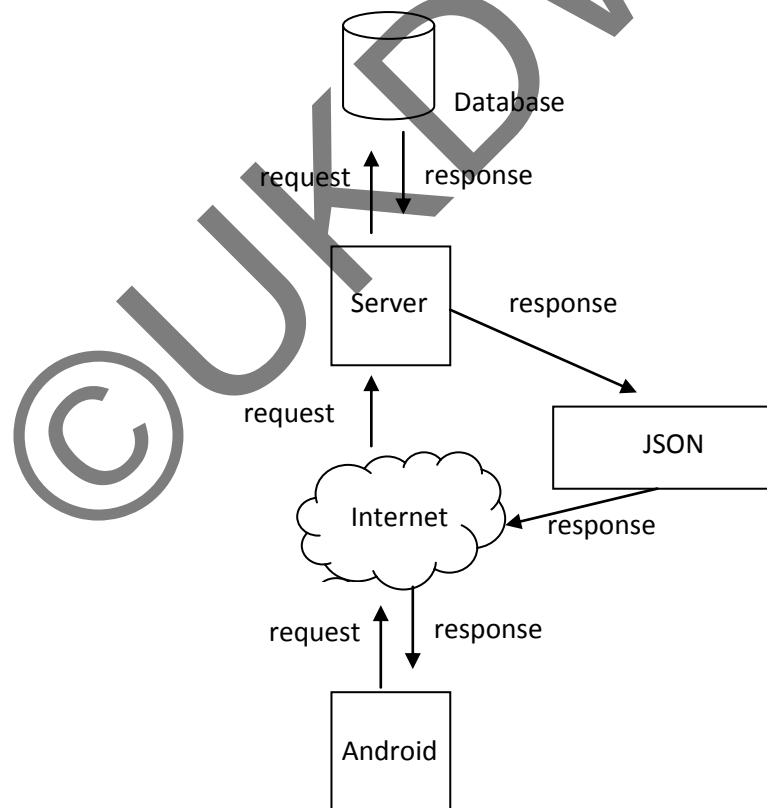
BAB IV

IMPLEMENTASI DAN ANALISIS SISTEM

4.1 Implementasi Basis Pengetahuan Sistem

4.1.1 Arsitektur Sistem

Pada Gambar 4.1 merupakan gambar arsitektur sistem untuk Sistem Pakar Diagnosa Penyakit Klelinci yang menggunakan *web service* sebagai komunikasi data dan android sebagai antarmuka. Dalam sistem ini android koneksi ke internet, kemudian meminta data dari server, kemudian server mencari data di database, setelah mendapat data dikirim ke server, kemudian diolah kebentuk JSON kemudian data diterima dan diolah oleh android kemudian ditampilkan.



Gambar 4.1 Arsitektur Sistem

4.1.2 Implementasi Antarmuka

Aplikasi ini terdiri dari 2 bagian sistem yaitu sistem untuk pengolahan data dalam bentuk *Web service* dan Android yang merupakan aplikasi utama yang digunakan oleh *user*.

Pada *Web service*, admin memiliki akses untuk memenejemen data pengobatan, mengolah data pencegahan, manajemen data gejala, manajemen data penyakit, melihat data *user* sertamelihat data tes.

a) Halaman Data Penyakit



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

[tambah data penyakit](#)

Gambar 4.2 Halaman Data Penyakit

Gambar 4.2 di atas merupakan implementasi halaman data penyakit. Halaman ini merupakan halaman dari data-data penyakit yang ada pad datababase. Pada halaman ini terdapat link perrbaharui untuk melakukan pembaruan data dan link hapus untuk menghapus data penyakit.

b) Halaman Input Data Penyakit

The screenshot shows the 'Form Data Penyakit' interface. At the top, there's a navigation bar with links: 'Gejala', 'Penyakit', 'Pencegahan', 'Pengobatan', 'Pelaku Tes', 'Ganti Password', and 'Logout'. The form itself has a title 'Form Data Penyakit' and several input fields: 'Kode' (P12), 'Penyakit Nama' (Bloat), 'Penyakit Namalatin' (Tympahni), and 'Nilai CF Penyakit' (0.8). Below these are three main sections, each with a blue header bar: 'Gejala' (Symptoms) with checkboxes for [G1] Suhu badan naik, [G10] Daerah anus kotor, [G11] Kotorannya hijau gelap dan bau, and [G12] Lendir menggantung pada dubur; 'Pengobatan' (Treatment) with checkboxes for [OB-1] Selama 1 hari hewan diberi pakan hay atau daun kacang kering, [OB-10] Olesi dengan salep Belerang, and [OB-11] Olesi dengan salep Caviam; and 'Pencegahan' (Prevention) with checkboxes for [PC-1] Berikan pakan cukup serat kasar and [PC-10] Beri hijauan segar yang cukup. At the bottom, there's a 'Perbarui Data' button and a link 'kembali ke data penyakit'.

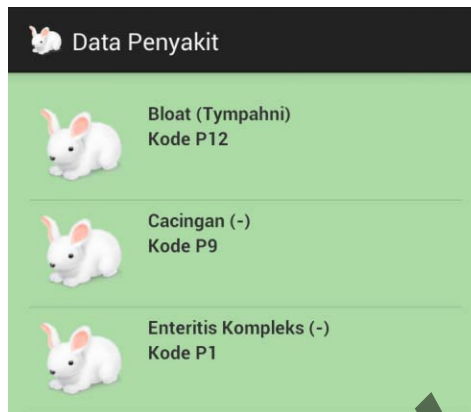
Gambar 4.3 Halaman Input Data Penyakit

Gambar 4.3 di atas merupakan implementasi dari halaman input data penyakit. Pada halaman ini yang diinputkan adalah kode penyakit, nama penyakit, nama latin, nilai CF, gejala, pengobatan, dan pencegahan.

Pada aplikasi android terdapat 4 menu yang bias dipilih yaitu menu list penyakit, menu pakar kelinci, menu account, dan menu informasi.

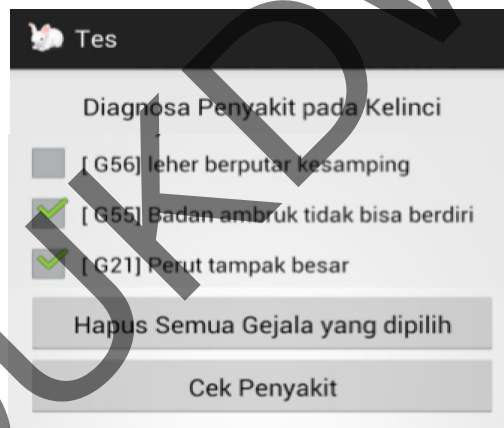
c). Halaman Daftar Penyakit

Pada Gambar 4.4 di atas merupakan implementasi dari halaman daftar penyakit. Pada halaman ini *user* bisa melihat daftar penyakit yang telah tersedia di aplikasi diagnosa penyakit kelinci.



Gambar 4.4 Halaman List Daftar Penyakit

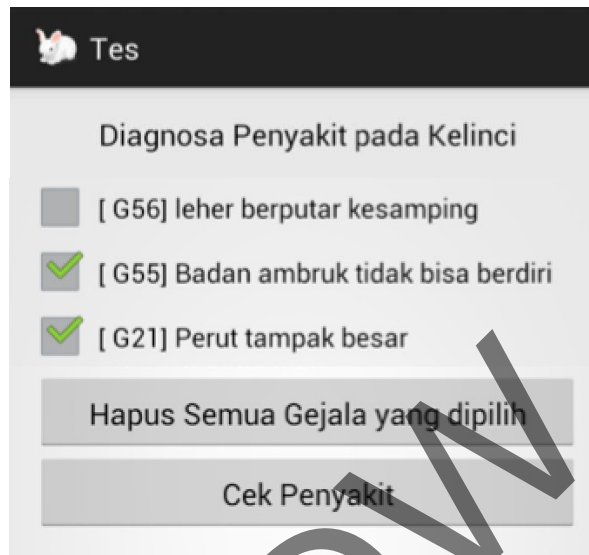
d). Halaman Pemilihan Gejala Utama



Gambar 4.5 Halaman Pemilihan Gejala Utama

Gambar 4.4 di atas merupakan halaman awal saat kita memasuki menu pakar dan kemudian kita dapat memilih gejala utama/umum yang terlihat pada kelinci kita. Gejala utama yang ditampilkan berupa pilihan *radio button*, jadi *user* hanya bisa memilih satu pilihan gejala utama. Setelah memilih salah satu gejala kita dapat memasuki ke langkah selanjutnya dengan menekan tombol *next*.

e). Halaman Pilih Gejala Pendukung



Gambar 4.6 Halaman Pilih Gejala Pendukung

Gambar 4.6 merupakan halaman yang muncul setelah gejala utama dipilih. Pada halaman ini gejala yang ditampilkan berupa pilihan *checkbox*, *user* dapat memilih lebih dari 1 gejala sesuai dengan apa yang terdapat pada kelinci yang sakit. Setelah memilih gejala pendukung, *user* dapat melanjutkan diagnosa dengan menekan tombol 'cek penyakit' atau jika akan mengulang pemilihan dapat menekan tombol 'hapus semua gejala yang dipilih'.

f). Halaman Input Bobot Gejala

Gambar 4.7 merupakan halaman untuk menginputkan bobot dari masing-masing gejala yang telah kita pilih pada dua halaman sebelumnya. Bobot yang pada gejala utama sudah *default* dengan nilai 1(satu) karena itu adalah gejala utama atau gejala umum yang sudah pasti, untuk gejala pendukung dapat diisikan dengan nilai dari 0.1 sampai dengan 0.9. Setelah semua bobot diinputkan, *user* dapat mengetahui hasil diagnosa dengan menekan tombol cek

penyakit. Bobot yang diinputkan akan dikirim ke sistem web via *web service* untuk dicocokkan di database. Web service akan memberikan respon berupa hasil dari perhitungan berdasarkan gejala dan bobot yang diinputkan.

Gejala	Nilai MB
[G21] Perut tampak besar	0.9
[G3] Sesak nafas atau susah bernafas	1
[G55] Badan ambruk tidak bisa berdiri	0.8

Cek Penyakit

Gambar 4.7 Halaman Input Bobot Gejala

g). Halaman Hasil Diagnosa (penyakit ditemukan)

Pada Gambar 4.8 di bawah ini merupakan halaman hasil dari perhitungan jika gejala yang diinputkan *user* ada dalam basis pengetahuan. Pada halaman ini akan ditampilkan data penyakit yang cocok, data gejala yang diinputkan *user*, cara pengobatan, dan cara pencegahan.

Hasil Diagnosa berdasarkan gejala yang diinputkan, maka Hewan Kelinci Anda diperkirakan menderita penyakit :

[P12] Bloat (CF = 0.9676)

Gejala penyakit :

- [G21] Perut tampak besar
- [G3] Sesak nafas atau susah bernafas
- [G55] Badan ambruk tidak bisa berdiri

Cara Pengobatan :

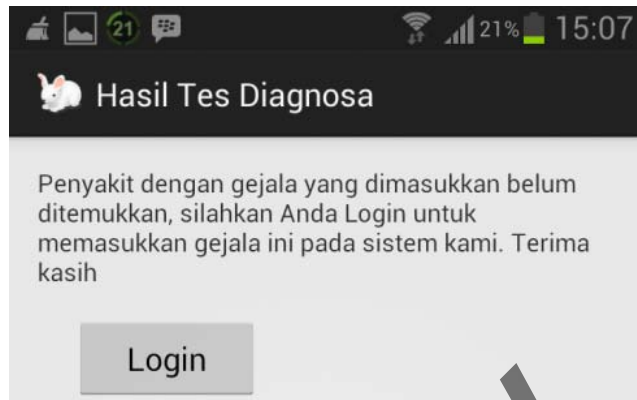
- Hidung yang penuh cairan disemprot larutan antiseptic

Cara pencegahan :

- Sanitasi kandang

Gambar 4.8 Halaman Hasil Diagnosa (penyakit ditemukan)

- h). Halaman Hasil Diagnosa (penyakit tidak ditemukan)

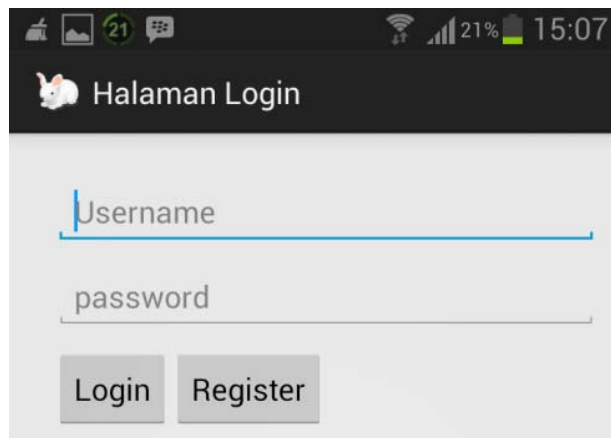


Gambar 4.9 Halaman Hasil Diagnosa (penyakit tidak ditemukan)

Gambar 4.9 merupakan halaman hasil diagnosa jika gejala-gejala yang diinputkan *user* tidak ada yang cocok dalam basis pengetahuan. Pada halaman ini *user* akan diarahkan untuk melakukan *register* (apabila belum mempunyai akun) atau *login* (jika sudah mempunyai akun) ke aplikasi untuk dapat mengirimkan data gejala yang nanti diinputkan ke database agar bisa di *review* oleh pakar di sistem aplikasi web.

- i). Halaman *Account*

Gambar 4.10 dibawah ini merupakan halaman Akun dari *user* yang akan melakukan pengujian terhadap penyakit kelinci yang dimana pada halaman ini terdapat *login* dan *register*, dimana *user* yang sudah pernah membuat akun akan bisa *login* sesuai yang sudah didaftarkan ke sistem dan apabila belum, dapat melakukan *register* terlebih dahulu.



Gambar 4.10 Halaman *Account*

4.1.3 Implementasi Algoritma

Di dalam membangun aplikasi Diagnosa Penyakit Kelinci ini, menggunakan algoritma *Forward Chaining* dan *Certainty Factor*. Berikut adalah potongan program beserta dengan pseudocode dari algoritma yang saya terapkan untuk membangun aplikasi ini.

1. Algoritma dan Implementasi Forward Chaining Pengambilan Gejala Utama

Pada proses ini dilakukan pemilihan terhadap gejala umum yang sering muncul atau gejala yang sering dipakai lebih dari 1 penyakit dan mempunyai gejala pendukung. Berikut ini adalah kode program dan algoritma dan penerapan forward chaining pengambilan penentuan gejala umum :

```
//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 ad
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 Utaama

```

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 l = 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 adalah lanjutan dari proses pemilihan gejala utama, pengguna akan memilih gejala pendukung pada proses ini sebelum menuju pada halaman penghitungan nilai kepastian terhadap penyakit yang nantinya akan muncul. Berikut ini adalah kode program dan algoritma dan penerapan forward chaining pengambilan penentuan gejala pendukung :

```

public function getTest($gejalaUtama){
    $data = array();
    //cari penyakit yang memiliki gejala utama
    $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;
}
//cari gejala yang sesuai penyakitnya dengan gejala utama
$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;
}
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 dahulu nilai minimal dari setiap rule agar bisa dihitung dalam rumus **CF Rule = min(a,b) x CF pakar** , yang nantinya hasil dari perhitungan tersebut akan digunakan untuk menghitung CF kombinasi.

Kemudian setelah nilai CF Rule didapatkan, sistem akan terlebih dahulu melakukan pengecekan terhadap gejala yang telah dipilih oleh pengguna dengan dengan *rule* yang telah ada, jika ada yang cocok akan segera di simpan dahulu agar dihitung pada CF kombinasi yg 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 pada aplikasi. Berikut ini adalah kode program dan algoritma dan penerapan menentukan nilai minimal dan menghitung CFkombinasi :

```
//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

```

4.2 Analisis Sistem

Analisa sistem bertujuan untuk mengetahui seberapa besar kecocokan antara diagnosa Pakar sebenarnya dengan pengetahuan dalam sistem yang telah dibuat. Pengujian terhadap pakar dilakukan sebanyak 25 kasus, hasil diagnosa pakar terdapat dalam Tabel 4.1 . Perincian gejala pada setiap kasus dapat dilihat pada lampiran analisis sistem.

Tabel 4.1 Analisis Data Testing

Kasus	Diagnosa Pakar		Diagnosa Sistem		Keterangan (ya/tidak)
	Penyakit	CF	Penyakit	CF	
1	Enteritis Kompleks	0.9	Enteritis Kompleks	0.8	ya
2	Bloat(Tympany)	0.8	Bloat(Tympany)	0.8	ya
3	Cacing Gilig (Pinworm/Oxyuris)	0.8	Cacing Gilig (Pinworm/Oxyuris)	0.7	ya
4	Radang Paru-paru (Pneumonia/Pasteurelloises)	0.7	-	-	tidak

5	Infeksi Kandungan (Metritis)	0.9	Infeksi Kandungan (Metritis)	0.8	ya
6	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.9	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.8	ya
7	Jamur(Ring Worm)	0.7	Jamur(Ring Worm)	0.7	ya
8	Radang Susu(Mastitis)	0.9	Radang Susu(Mastitis)	0.7	ya
9	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.6	-	-	tidak
10	Kudis(Scabies)	0.3	Kudis(Scabies)	0.4	ya
11	Parasit Pencernaan (Coccidiosis)	0.8	Parasit Pencernaan (Coccidiosis)	0.7	ya
12	Bloat(Tympany)	0.5	-	-	tidak
13	Kudis(Scabies)	0.9	Kudis(Scabies)	0.9	ya
14	Radang Mata	0.8	Radang Mata	0.9	ya
15	Bloat(Tympany)	0.5	-	-	tidak
16	Radang Mata	0.9	Radang Mata	0.8	ya
17	Jamur(Ring Worm)	0.4	Jamur(Ring Worm)	0.5	ya
18	Cacing Gilig (Pinworm/Oxyuris)	0.7	Cacing Gilig (Pinworm/Oxyuris)	0.7	ya
19	Cacingan	0.5	Cacingan	0.5	ya
20	Makan Bulu(Hairball)	0.9	Makan Bulu(Hairball)	0.8	ya
21	Cacingan	0.9	Cacingan	0.8	ya
22	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.8	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.8	ya
23	Enteritis Kompleks	0.9	Enteritis Kompleks	0.9	ya

24	Radang Susu(Mastitis)	0.9	Radang Susu(Mastitis)	0.8	ya
25	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.3	Makan Bulu(Hairball)	0.3	tidak
	Jumlah	15.6	Jumlah	14.8	

CF rata-rata Pakar = $15.6 / 20 = 0.78$

CF rata-rata Sistem = $14.8 / 20 = 0.74$

Prosentase kecocokan = $\frac{20}{25} \times 100\% = 80\%$

Ketepatan sistem 80% dengan CF rata-rata 0.74 dan hanya selisih 0.04 dari CF pakar rata-rata.

4.3 Kelebihan dan Kekurangan Sistem

Berdasarkan implementasi sistem didapat kelebihan dan kekurangan sistem. Kelebihan dari aplikasi diagnosa penyakit kelinci ini telah mampu melakukan diagnosa penyakit pada kelinci dengan baik. Aplikasi ini juga membantu *users* untuk mengetahui bagaimana cara pencegahan dan pengobatan berbagai gejala penyakit kelinci yang muncul pada kelinci peliharaan, namun penanganan lebih lanjut dibutuhkan konsultasi dengan dokter.

Aplikasi ini juga memiliki kekurangan antara lain yaitu pengobatan dan pencegahan yang diberikan masih bersifat umum, masih harus ditambahkan agar lebih rinci dan lengkap. Selain itu, aplikasi ini masih perlu adanya pengembangan di bidang basis pengetahuannya, sehingga ketepatan diagnosa akan lebih optimal.

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');
    }

    //get list penyakit
    public function getPenyakit(){
        $data = array();

        $query = $this->db->query("SELECT  penyakit.*,  max((select
count(penyakit_gejala.id_penyakit)  from      penyakit_gejala      where
penyakit_gejala.id_penyakit = penyakit.id_penyakit)) as j
from penyakit
where penyakit_status = 1

```

```

        group by penyakit_kode
        order by j desc");
    if($query->num_rows() > 0){
        $data['result'] = 'true';
        $data['data'] = $query->result();
    }else{
        $data['result'] = 'false';
    }

    echo json_encode($data);
}

//get gejala berdasarkan id penyakit
public function getGejala($id){
    $data = array();
    $this->db->where('pg.penyeakit_gejala_status', 1);
    $this->db->where('pg.id_penyakit', $id);
    $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penyakit_gejala pg');
    #pre($this->db->last_query());

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

    echo json_encode($data);
}

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

    foreach ($query->result() as $row){
        $penyakit[] = $row->id_penyakit;
    }
    #pre($penyakit);
    //cari gejala yang sesuai penyakitnya dengan gejala utama
    $this->db->where('pg.penyeakit_gejala_status', 1);
    $this->db->where_in('pg.id_penyakit', $penyakit);
    $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
    $query = $this->db->get('penyakit_gejala pg');
    #pre($this->db->last_query());

    foreach ($query->result() as $row){
        $data[$row->id_gejala] = '[ ' . $row->gejala_kode . ' ] ' . $row->gejala_nama;
    }
    //unset($data[$gejalaUtama]);

```

```

        //$data['data'] = $data;
        echo json_encode($data);
    }

    public function getTestx(){
        $data = array();
        $this->db->where('jenisgejala_status', 1);
        $rs_jenisgejala = $this->db->get('jenisgejala')->result();
        foreach($rs_jenisgejala as $r){
            $data[$r->jenisgejala_nama] = $this->getGejalax($r-
>id_jenisgejala);
        }

        #pre($data);

        echo json_encode($data);
    }

    public function getGejalax($id){
        $this->db->order_by('gejala_kode');
        $this->db->where('gejala_status', 1);
        $this->db->where('id_jenisgejala', $id);
        return $this->db->get('gejala')->result();
    }

    public function getPelakuTes(){
        $data = array();
        $this->db->where('pelakutes_status', 1);
        $this->db->where('pelakutes_nomoridentitas', $this->input-
>post('noIdentitas'));
        $query = $this->db->get('pelakutes');
        #pre($this->db->last_query());

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

        echo json_encode($data);
    }

    public function tesEntry(){
        $data = array();

        //ambil id pelaku tes
        $idPelaku = $this->input->post('pelakuId');

        //masukkan ketabel tes
        $data_simpan2 ['id_pelakutes'] = $idPelaku;
        if($this->db->insert('tes', $data_simpan2)){
            $idTes = $this->db->insert_id();

            //ambil semua gejala

```

```

        $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';
            }else{
                $data['result'] = 'false';
            }
        }
    }else{
        $data['result'] = 'false';
    }

    echo json_encode($data);
}

//detail penyakit, setelah diklik pada
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));

    echo json_encode($data);
}

public function getGejalay($id){
    $data = array();
    $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 = $this->db->get('penyakit_gejala pg');
    #pre($this->db->last_query());

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

    return $data;
}

public function getPencegahan($id){
    $data = array();
    $this->db->where('pp.penyakit_pencegahan_status', 1);
    $this->db->where('pp.id_penyakit', $id);
    $this->db->join('pencegahan p', 'p.id_pencegahan
pp.id_pencegahan');
    $query = $this->db->get('penyakit_pencegahan pp');
    #pre($this->db->last_query());

    if($query->num_rows() > 0){

```

```

        $data['resultPencegehan'] = 'true';
        $data['dataPencegehan'] = $query->result();
    }else{
        $data['resultPencegehan'] = 'false';
    }

    return $data;
}

public function getPengobatan($id){
    $data = array();
    $this->db->where('pp.penyakit_pengobatan_status', 1);
    $this->db->where('pp.id_penyakit', $id);
    $this->db->join('pengobatan p', 'p.id_pengobatan =
pp.id_pengobatan');
    $query = $this->db->get('penyakit_pengobatan pp');
    #pre($this->db->last_query());

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

    return $data;
}

public function tesPenyakitCF(){
    $data = array();

    $gp_input = $this->input->post('gp');
    foreach($gp_input as $k => $v){
        foreach($v as $k2 => $v2){
            $md = 1 - $v2;

            $gp[$k][$k2][1] = $v2;
            $gp[$k][$k2][2] = $md;
        }
    }
    #pre($gp);

    //panggil fungsi perhitungan CF yang ada di helper
    $cf = certainty_factor($gp);
    #pre($cf);

    $rs_gejalax = $this->p->gejalaArray();
    #$data['gejalaArrayX'] = $rs_gejalax;

    $rs_penyakit = $this->p->penyakitArray();
    #$data['penyakitArray'] = $rs_penyakit;
    #$data['gejala_penyakit'] = $gp;

    $data['hasil_cf'] = $cf;
    $data['nilai_cf'] = $cf['cf'];
    #pre($cf);
}

```

```

//ambil id penyakit dari cf untuk mencari pengobatan dan
pencegahan
$id_penakit = $cf['penyakit'];
$data['id_penakit'] = $cf['penyakit'];
$data['penyakit'] = $rs_penakit[$cf['penyakit']];
$penakit_gejala = $gp[$id_penakit];
foreach($penakit_gejala as $k => $v){
    $data['gejala'][] = $rs_gejalax[$k];
}

//cari cara pengobatan
$this->db->where('penyakit_pengobatan.id_penakit',
$id_penakit);
$this->db->where('pengobatan_status', 1);
$this->db->
>join('pengobatan', 'pengobatan.id_pengobatan=penyakit_pengobatan.id_peng
obatan');
$query = $this->db->get('penyakit_pengobatan');
//pre($this->db->last_query());
if($query->num_rows() > 0){
    $data['result_pengobatan'] = 'true';
    $data['data_pengobatan'] = $query->result();
}else{
    $data['result_pengobatan'] = 'false';
}

//cari cara pencegahan
$this->db->where('penyakit_pencegahan.id_penakit',
$id_penakit);
$this->db->where('pencegahan_status', 1);
$this->db->
>join('pencegahan', 'pencegahan.id_pencegahan=penyakit_pencegahan.id_penc
egahan');
$query2 = $this->db->get('penyakit_pencegahan');
if($query2->num_rows() > 0){
    $data['result_pencegahan'] = 'true';
    $data['data_pencegahan'] = $query2->result();
}else{
    $data['result_pencegahan'] = 'false';
}

#pre($data);
echo json_encode($data);

}

//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;
    }

    $d_penyakit = $this->get_penyakit_gejala();

    $hasil = array();
    $d_gejala_utama = $this->GejalaUtama($d_gejala, $d_penyakit, 0,
$hasil);
    #pre("--- gejala utamanya ---");
    #pre($d_gejala_utama);
    foreach($d_gejala_utama as $v){
        $d_gejalanya[$v] = $d_gejala_array[$v];
    }
    asort($d_gejalanya);
    echo json_encode($d_gejalanya);

    #pre($d_gejalanya);
}

//rekursif pencarian gejala utama
public function GejalaUtama($d_gejala, $d_penyakit, $rekursif,
$d_hasil){
    #pre($rekursif);
    #pre($d_gejala);
    #pre($d_penyakit);

    //cari gejala utama berdasarkan gejala terbanyak di penyakit
    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] = $k2;
            }
        }
        $d_ketemu2[] = $d_ketemu;
    }
    #pre($d_ketemu2);
    #ambil kategori terbanyak
    $jml = 0;
    foreach($d_ketemu2 as $k => $v){
        #pre(count($v));
        foreach($v as $k2 => $v2){
            if(count($v2) > $jml){
                $jml = count($v2);
                $d_ambil = $v;
            }
        }
    }
    #pre($d_ambil);
    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] = $hasil;
    #pre($d_penyakit);
    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 = $this->input->post('gejala');
    //pre($g);
    //    $g = Array
    //        (
    //            '2' => '2',
    //        );

    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 = $this->db->get('penyakit_gejala pg');
        #pre($this->db->last_query());

        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];
        }

        //check semua gaejala ada dipenyakit tersebut
        $d = array();
        foreach($d_penyakit as $k => $v){
            $jmlGejala = count($g);
            $containsSearch = count(array_intersect($v, $g));

            if($jmlGejala == $containsSearch){
                $d[] [$k] = $penyakit[$k];
            }
        }
    }
}

```

```

        #pre($d);
        if(count($d) > 0){
            $data['result'] = 'true';
            $data['data'] = $d;
            $data['gejala'] = $temp;
        }else{
            $data['result'] = 'false';
            $data['gejala'] = $temp;
        }

        echo json_encode($data);

    }else{
        $this->db->where('pg.penyakit_gejala_status', 1);
        $this->db->join('gejala g','g.id_gejala = pg.id_gejala');
        $this->db->order_by('g.gejala_kode','ASC');
        $query = $this->db->get('penyakit_gejala pg');
        #pre($this->db->last_query());

        if($query->num_rows() > 0){
            foreach($query->result() as $r){
                $d_penyakit[$r->id_penyakit][] = $r->id_gejala;
            }
        }
        #pre($d_penyakit);
        return $d_penyakit;
    }
}

public function register(){
    $data['pelakutes_nama'] = $this->input->post('nama');
    $data['pelakutes_password'] = md5($this->input->post('password'));
    $data['pelakutes_nomoridentitas'] = $this->input->post('identitas');
    $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';
    }else{
        $datax['register'] = 'false';
    }

    echo json_encode($datax);
}

public function login(){
    $data = array();
    $email = $this->input->post('identitasa');
    $password = md5($this->input->post('password'));

    //query cek user terdaftar

```

```

$this->db->where('pelakutes_nomoridentitas',$email);
$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();
}else{
    $data['login'] = 'false';
}

echo json_encode($data);
}

public function gejala_by_id($id_penyakit){
    $this->db->where('penyakit_gejala_status',1);
    $this->db->where('id_penyakit',$id_penyakit);
    $q_gej = $this->db->get('penyakit_gejala');
    foreach($q_gej->result() as $r){
        $data[] = $r->id_gejala;
    }
    return $data;
}

public function cf_by_id($id_penyakit){
    $this->db->where('id_penyakit',$id_penyakit);
    $q_gej = $this->db->get('penyakit_gejala');
    foreach($q_gej->result() as $r){
        $data[] = $r->id_gejala;
    }
    return $data;
}

//perhitungan
public function cf(){
    $input_user = $this->input->post('gejala');

    //bloat
    /*$input_user = array(
        25 => 0.8,
        3 => 1,
        68 => 0.6,
    );*/

    $input_user = array(
        20 => 0.7,
        2 => 1,
        50 => 0.8,
    );

    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');

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

    //pre($query_a->result());

    //pre($this->db->last_query());
    //pre($query_a->result());

    if($query_a->num_rows() > 0){
        $query_cocok = $query_a->row()->id_penyakit;
        //pre($query_cocok);
        $data['result'] = 'true';
        $data['penyakit'] = $query_a->row()->penyakit_nama;

        $data['bobot_minimal'] = min($input_user);
        $data['penyakit_cf'] = $query_a->row()->penyakit_cf;

        $data['id_penyakit'] = $query_cocok;
        $rs_gejalax = $this->p->gejalaArray();
        foreach($input_user as $k => $v){
            $data['gejala'][] = $rs_gejalax[$k];
        }

        //query gejala
        //$this->db->where();

        $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){

```

```

        $rule[$r->id_penyakit][] = $r->id_gejala;

        $cf_pakar[$r->id_penyakit] = $r->penyakit_cf;
    }
    //pre($cf_pakar);
    //pre($query_b->result());
    #pre($this->db->last_query());
    #pre($query_a->result());

    $x = CF_KOMBINASI($query_cocok, $rule, $cf_pakar,
$input_user);

    //pre($x);
    $data['nilai_cf'] = $x;

    //CARA PENGobatan QUERY
    $this->db->where('penyakit_pengobatan.id_penyakit',
$query_cocok);

    $this->db->where('pengobatan_status', 1);
    $this->db->
>join('pengobatan','pengobatan.id_pengobatan=penyakit_pengobatan.id_peng
obatan');

    $query = $this->db->get('penyakit_pengobatan');
    //pre($this->db->last_query());
    if($query->num_rows() > 0){
        $data['result_pengobatan'] = 'true';
        $data['data_pengobatan'] = $query->result();
    }else{
        $data['result_pengobatan'] = 'false';
    }

    //CARA PENCEGAHAN QUERY
    $this->db->where('penyakit_pencegahan.id_penyakit',
$query_cocok);

    $this->db->where('pencegahan_status', 1);
    $this->db->
>join('pencegahan','pencegahan.id_pencegahan=penyakit_pencegahan.id_penc
egahan');

    $query2 = $this->db->get('penyakit_pencegahan');
    if($query2->num_rows() > 0){
        $data['result_pencegahan'] = 'true';
        $data['data_pencegahan'] = $query2->result();
    }else{
        $data['result_pencegahan'] = 'false';
    }
    }else{
        $data['result'] = 'false';
    }

    echo json_encode($data);

}

public function tesPenyakitCFRevisi(){
    $data = array();
    // $g = $this->input->post('gejala');

```

```

$g = array(
    25 => 1,
    3 => 0.5,
    68 => 0.6,
);

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

$jumlah_gejala = count($gejala);

//list penyakit yg ada gejala yg dicentang
$this->db->select('*', (select count(id_penyakit) from
penyakit_gejala where id_penyakit = pg.id_penyakit) 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.penakit_status', 1);
$this->db->group_by('p.id_penyakit');
$this->db->having('jumlah_gejala', $jumlah_gejala);
$query_a = $this->db->get('penyakit_gejala pg');
//pre($query_a->result());
if($query_a->num_rows() > 0){
    foreach($query_a->result() as $r){
        $penyakit_masuk[] = $r->id_penyakit;
        $penyakit_nilai_cf[$r->id_penyakit][$r->id_penyakit] =
        $r->penyakit_cf; //get nilai CF
    }
    //pre($penyakit_nilai_cf);

    //Query get list gejala
    $this->db->where_in('id_penyakit', $penyakit_masuk);
    $query_b = $this->db->get('penyakit_gejala');
    //pre($query_b->result());
    foreach($query_b->result() as $rx){
        $penyakit_gejala_masuk[$rx->id_penyakit][] = $rx-
>id_gejala;
    }
    //pre($penyakit_gejala_masuk);

    //proses pencocokan gejala yg diinput dengan gejala pada
    sebuah penyakit
    foreach($penyakit_gejala_masuk as $ke=>$va){
        $x = array_diff($gejala, $va);
        if(empty($x)){
            $hasil[$ke] = $va;
            $id_penyakit_terpilih = $ke;
        }
    }

    //cek pencocokan (KONDISI 'AND' TERPENUHI) -> $gejala =
    $hasil[value]
    if(isset($hasil)){

```

```

//pre($hasil);
$data['result'] = 'true';
$bobot_minimal = min($g); //bobot minimal
$data['bobot_minimal'] = $bobot_minimal;
$data['penyakit_cf'] =
$penyakit_nilai_cf[$id_penyakit_terpilih];
$data['nilai_cf'] =
$penyakit_nilai_cf[$id_penyakit_terpilih][$id_penyakit_terpilih];
$data['id_penyakit'] = $id_penyakit_terpilih;
$rs_penyakit = $this->p->penyakitArray();
$data['penyakit'] = $rs_penyakit[$id_penyakit_terpilih];
$rs_gejalax = $this->p->gejalaArray();
foreach($g as $k => $v){
    $data['gejala'][] = $rs_gejalax[$k];
}

//HASIL DARI RUMUS : CF = min input x nilai CF
//cek cf gabungan
$rs_penyakitKode = $this->p->penyakitKodeArray();
$id_dan_kode= $rs_penyakitKode[$id_penyakit_terpilih];
$cek_gabungan = $this->cek_gabungan($id_dan_kode,
$penyakit_nilai_cf[$id_penyakit_terpilih][$id_penyakit_terpilih], $bobot_
minimal);
//pre($cek_gabungan);
$data['hasil_perhitungan'] = $cek_gabungan;

//CARA PENGobatan QUERY
$this->db->where('penyakit_pengobatan.id_penyakit',
$id_penyakit_terpilih);
$this->db->where('pengobatan_status', 1);
$this->db->
>join('pengobatan', 'pengobatan.id_pengobatan=penyakit_pengobatan.id_peng
obatan');
$query = $this->db->get('penyakit_pengobatan');
//pre($this->db->last_query());
if($query->num_rows() > 0){
    $data['result_pengobatan'] = 'true';
    $data['data_pengobatan'] = $query->result();
}else{
    $data['result_pengobatan'] = 'false';
}

//CARA Pencegahan QUERY
$this->db->where('penyakit_pencegahan.id_penyakit',
$id_penyakit_terpilih);
$this->db->where('pencegahan_status', 1);
$this->db->
>join('pencegahan', 'pencegahan.id_pencegahan=penyakit_pencegahan.id_penc
egahan');
$query2 = $this->db->get('penyakit_pencegahan');
if($query2->num_rows() > 0){
    $data['result_pencegahan'] = 'true';
    $data['data_pencegahan'] = $query2->result();
}else{

```

```

        $data['result_pencegahan'] = 'false';
    }

    }else{
        //ga ada yg cocok dari hasil pencocokan
        $data['result'] = 'false';
    }
}
}
}
}

    $data['result'] = 'false';
}

    echo json_encode($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
    manajemen 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 dimasukan 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', $id=''){
        $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 $id tidak kosong */
        if ($id != '' ){
            $this->db->where('id_hewan',$id);
            $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 dimasukan 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', $id='') {
        $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 $id tidak kosong */
        if ($id != '') {
            $this->db->where('id_pencegahan',$id);
            $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 dimasukan 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',
$this->input->post('f_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', $id=''){

        $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 $id tidak kosong */
if ($id != '' ){
    $this->db->where('id_pengobatan',$id);
    $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 dimasukan 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',
$data_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', $sid='') {
        $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_penyakit['penyakit_gambar'] =
$namafile.'_'.$files['name'];
            }
        }
    }

```

```

        */
        /* siapkan data dari form untuk dimasukan dalam
table */
        $data_penyakit['penyakit_kode'] = $this->input-
>post('f_penyakit_kode');
        $data_penyakit['penyakit_nama'] = $this->input-
>post('f_penyakit_nama');
        $data_penyakit['penyakit_cf'] = $this->input-
>post('f_penyakit_cf');
        $data_penyakit['penyakit_namalatin'] = $this->input-
>post('f_penyakit_namalatin');

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

                /* penyakit gejala */
                $this->db-
>where('id_penyakit',$id_penyakit);
                $this->db->delete('penyakit_gejala');
                /* penyakit pengobatan */
                $this->db-
>where('id_penyakit',$id_penyakit);
                $this->db-
>delete('penyakit_pengobatan');
                /* penyakit pencegahan */
                $this->db-
>where('id_penyakit',$id_penyakit);
                $this->db-
>delete('penyakit_pencegahan');
                /* penyakit hewan */
                // $this->db-
>where('id_penyakit',$id_penyakit);
                // $this->db->delete('hewan_penyakit');

                $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);
    }
}

```

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> nameValuePairs;
    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(getBaseContext());
        trTombol.addView(btnCekPenyakit);
        tabel.addView(trTombol);

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

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

                    RbHelper.pesan(getBaseContext(),
                        "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());
            }
        }
    }
}

```

```

        et.getText().toString());
        Log.i(et.getTag().toString(),
    }

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

    }

    });

}

private void setupView() {
    hsView = (HorizontalScrollView)
    findViewById(R.id.horizontalScrollBobot);
    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.tbLayoutBobot);
    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);
    }
}

```

```

    }

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

        Log.i("json", jsonRespon.toString());
        return getData;
    }

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

        try {
            String hasil = jsonRespon.getString("result");
            dataGejala = new HashMap<String, String>();

            if (hasil.equalsIgnoreCase("true")) {
                tvJudul.setVisibility(View.VISIBLE);
                tvJudul.setText("Silahkan masukkan bobot untuk setiap penyakit : ");
                dataTes = new HashMap<String, String>();
                dataPenyakit = new HashMap<String, String>();

                try {
                    TableRow tr = new TableRow(getContext());
                    TextView tv = new TextView(getContext());
                    tv.setText("Gejala / Penyakit");
                    tv.setTextColor(Color.BLACK);
                    tv.setPadding(10, 0, 0, 10);
                    tr.addView(tv);
                    JSONArray jPenyakit = jsonRespon.getJSONArray("data");

                    for (int i = 0; i < jPenyakit.length(); i++) {
                        JSONObject x = jPenyakit.getJSONObject(i);
                        Iterator key = x.keys();
                        while (key.hasNext()) {
                            String kunci = (String) key.next();
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

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

for (Map.Entry<String, String>
e : dataPenyakit
        .entrySet()) {
    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++) {
    JSONObject x =
jGejala.getJSONObject(i);

    Iterator key = x.keys();
    while (key.hasNext()) {
        String kunci =
(String) key.next();
        dataGejala.put(kunci, x.getString(kunci));
    }
}

listPenyakitGejala = new
ArrayList<EditText>();
entryGejala : dataGejala

TableRow(getBaseContext());
TextView(getBaseContext());

        .entrySet()) {
    TableRow trGejala = new
    TextView tvGejala = new

```

```

        tvGejala.setText(entryGejala.getValue());

        tvGejala.setTextColor(Color.BLACK);
        tvGejala.setPadding(10,
3, 10, 3);

        trGejala.addView(tvGejala);
        for (Map.Entry<String,
String> entryPenyakit : dataPenyakit
        .entrySet()) {
            EditText
            txtGejala = new EditText(
                getBaseContext());
            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.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);

            listPenyakitGejala.add(txtGejala);

```

```

    }
    tabel.addView(trGejala);
    }
    btnCekPenyakit.setText("Cek
Penyakit");
    btnCekPenyakit.setPadding(5,
5, 5, 5);
    TableRow trTombol = new
TableRow(getBaseContext());
    trTombol.addView(btnCekPenyakit);
    tabel.addView(trTombol);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    } else {
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
    }
}

```

d) HelpActivity.java

```

package xrb.kelinci;

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

public class HelpActivity extends Activity {

```

```

@Override

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(getBaseContext(),
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.sendDataX(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);

                final RadioButton rb = new
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(getBaseContext(),
RegisterActivity.class);
            startActivity(i);
            break;
        case R.id.btnLoginSubmit:
            if (MainActivity.isLogin) {
                MainActivity.isLogin = false;
                RbHelper.pesan(getBaseContext(), "Anda Sudah
Logout.");
                Intent ix = new Intent(getBaseContext(),
MainActivity.class);
                startActivity(ix);
            } else {
                if (RbHelper.isEmpty(txtIdentitas)) {
                    RbHelper.pesan(getBaseContext(),

```

```

"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.sendDataax(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;
startActivity(i);
    } else {
        // pindah ke halaman main menu
        jParser.pesan("Login berhasil",
getBaseContext());

        Intent i = new Intent(getBaseContext(),
MainActivity.class);

        startActivity(i);
    }

    } else if (result.equalsIgnoreCase("false")) {
        jParser.pesan("Login gagal, Silahkan coba
lagi",
getBaseContext());
    }
}

}
}

```

h) MainActivity.java

```

package xrb.kelinci;

import xrb.kelinci.helper.PelakuTes;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends Activity {
    public static String URL_SERVICE =
"http://andrekusumaa.web.id/android/";
    public static boolean isLogin = false;
    public static boolean tesEntry = false;
    public static PelakuTes pelaku;

```

```
@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(getApplicationContext());
            tv.setText(no + ". " +
            me.getValue().toString());
            tv.setTextColor(Color.BLACK);
            llPelakuTes.addView(tv);
        }
    }
    else {
        RbHelper.pesan(getApplicationContext(),
        "Silahkan Anda Login dulu untuk
        mengakses halaman ini.");
        Intent i = new Intent(getApplicationContext(),
        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.sendDataax(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.sendDataax(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(getContext(), "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> dataaz;
    private LayoutInflater inflater = null;

    public CustomAdapter(Context c, ArrayList<Penyakit> data) {
        context = c;
        dataaz = data;
        inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return dataaz.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.sendData(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 < jArray.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 {
            x.setNama(ar.getString("pencegahan_nama"));

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

            dataPencegahan.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(),
String.valueOf(dataPencegahan.size()));
    if (result.equalsIgnoreCase("true")) {
        int no = 0;
        for (Iterator iterator = data.iterator();
iterator.hasNext();) {
            Gejala x = (Gejala) iterator.next();
            no++;

            TextView tv = new
TextView(getBaseContext());
            tv.setText(no + ". " + "[" +
x.getGejalaKode() + "]" +
                + x.getGejalaNama());
            tv.setPadding(15, 0, 0, 0);
            tv.setTextColor(Color.BLACK);
            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()); {
        Pengobatan x = (Pengobatan)
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);

```

```

llPengobatan.addView(tv);

    }

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

}

}

}

```

l) RegisterActivity.java

```

package xrb.kelinci;

import java.util.ArrayList;

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

import xrb.kelinci.helper.JSONParser;
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.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(getContext(), "Username  

Harus Diisi");

        } else if (RbHelper.isEmpty(txtPassword)) {

            RbHelper.pesan(getContext(), "Password  

Harus Diisi");

        } else if (RbHelper.isEmpty(txtRepassword)) {

            RbHelper.pesan(getContext(), "Ulangi  

Password Harus Diisi");

        } else if (RbHelper.isEmpty(txtAlamat)) {

            RbHelper.pesan(getContext(), "Alamat  

Harus Diisi");

        } else if (RbHelper.isEmpty(txtPhone)) {

            RbHelper.pesan(getContext(), "Phone Asal  

Harus Diisi");

        } else {

            // RbHelper.pesan(getContext(), "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, \nSilahkan
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.lineartest);
    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.sendDataX(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(getBaseContext());

        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

```
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(getApplicationContext(),
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", nameValuePairs.toString());
        jsonRespon = jParser.sendDataax(nameValuePairs, 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 gejala
                    TextView tvInfo = new

```

```

TextView(getContext());

        tvInfo.setTextColor(Color.BLACK);

        tvInfo.setText("Gejala penyakit
: ");

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

        // buat linear layout baru
        LinearLayout llayout = new
LinearLayout(
                                getContext());

        llayout.setOrientation(LinearLayout.VERTICAL);
        llayout.setPadding(20, 1, 0,
13);
        layoutLinear.addView(llayout);

        JSONArray jGejala =
jsonRespon.getJSONArray("gejala");
        for (int i = 0; i <
jGejala.length(); i++) {
                                TextView tvGejala = new
TextView(getContext());

                                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(getContext());

        tvInfo2.setTextColor(Color.BLACK);

        tvInfo2.setText("Cara
Pengobatan : ");

        tvInfo2.setPadding(20, 2,
0, 0);

        layoutLinear.addView(tvInfo2);

// buat linear layout
baru
        LinearLayout llayout2 =
new LinearLayout(

        getContext());

        llayout2.setOrientation(LinearLayout.VERTICAL);

        llayout2.setPadding(20,
1, 0, 13);

        layoutLinear.addView(llayout2);

        JSONArray jPengobatan =
jsonRespon.getJSONArray("data_pengobatan");

        for (int i = 0; i <
jPengobatan.length(); i++) {

                JSONObject x =

                TextView tv = new

                TextView(getContext());

                tv.setTextColor(Color.BLACK);

                tv.setText("- " +
x.getString("pengobatan_nama"));

```

```

        layout2.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++) {

JSONObject x =
jPencegahan.getJSONObject(i);

TextView tv = new
TextView(getBaseContext());

tv.setTextColor(Color.BLACK);

tv.setText("- " +
x.getString("pencegahan_nama"));

layout3.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;
startActivity(i);
} else {
/*dataGejala = new

```

```

HashMap<String, String>();

JSONArray jGejala =
jsonRespon.getJSONArray("gejala");

for (int i = 0; i <
jGejala.length(); i++) {

JSONObject x =

Iterator key = x.keys();
while (key.hasNext()) {
String kunci =

(String) key.next();

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

}*/
tvJudul.setText("Penyakit dengan
gejala yang dimasukkan belum ditemukan, "
+ "silahkan Anda
Login untuk memasukkan gejala ini pada sistem kami. Terima kasih ");

tvJudul.setVisibility(View.VISIBLE);

llPelakuTes.setVisibility(View.VISIBLE);

btnCari.setVisibility(View.VISIBLE);
}

}

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

}

}

}

```

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 jObj = 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 getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(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 setNama(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 getName() {  
    return nama;  
}  
  
public void setName(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 getName() {
        return nama;
    }

    public void setName(String nama) {
        this.nama = nama;
    }

    public String getNameLatin() {
        return namaLatin;
    }

    public void setNameLatin(String namaLatin) {
        this.namaLatin = namaLatin;
    }
}

```

Analisis Sistem

Kasus	Gejala	Diagnosa Pakar	
		Penyakit	CF
1	G2 (Badan Kurus) G11 (Kotorannya hijau gelap dan bau) G12 (Lendir menggantung pada dubur) G19 (Hewan terlihat lesu) G39 (Kotoran lembek cair)	Enteritis Kompleks	0.9
2	G3 (Sesak nafas atau susah bernafas),	Bloat(Tympany)	0.8

	G21(Perut tampak besar), G55 (Badan ambruk tidak bias berdiri)		
3	G2 (Badan kurus), G53 (Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya), G54 (Kotoran mata lebih banyak dari biasanya dan berwarna kuning)	Cacing Gilig (Pinworm/Oxyuris)	0.8
4	G2 (Badan kurus), G3 (Sesak nafas atau susah bernafas), G10 (Daerah anus kotor), G25 (Kepala sering diangkat tinggi-tinggi).	Radang Paru-paru (Pneumonia/Pasteurelloises)	0.7
5	G7 (Lendir di vagina berupa darah atau nanah), G50 (Minum banyak, makan sedikit), G51 (Jika kandungan diraba terasa membesar).	Infeksi Kandungan (Metritis)	0.9
6	G3 (Sesak nafas atau susah bernafas), G25 (Kepala sering diangkat tinggi-tinggi), G26 (Hidung keluar nanah, lender berwarna jernih atau keruh), G4 (Mata sembab, basah dan berair), G31 (Iritasi dibagian kulit, lalu menular kebagian dalam yang lain).	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.9
7	G5 (Bulu rontok), G29 (Merasa gatal),	Jamur(Ring Worm)	0.7

	G31 (Iritasi dibagian kulit, lalu menular kebagian dalam yang lain).		
8	G1 (Suhu badan naik), G16 (Nafsu makan turun), G18 (Ambing dan puting berwarna merah, mengeras), G43 (Ujung puting berwarna hitam)	Radang Susu(Mastitis)	0.9
9	G3 (Sesak nafas atau susah bernafas), G50 (Minum banyak, makan sedikit), G25 (Kepala sering diangkat tinggi-tinggi), G29 (Merasa gatal).	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.6
10	G2 (Badan kurus), G16 (Nafsu makan turun), G15 (Perut kembung)	Kudis(Scabies)	0.3
11	G2 (Badan kurus), G16 (Nafsu makan turun), G20 (Berat badan berkurang), G22 (Mencret bercampur darah)	Parasit Pencernaan (Coccidiosis)	0.8
12	G3 (Sesak nafas atau susah bernafas), G15 (Perut kembung), G20 (Berat badan berkurang).	Bloat(Tympany)	0.5
13	G2 (Badan kurus), G5 (Bulu rontok), G27 (Kulit kemerah-merahan), G28 (Badan penuh keropeng), G29 (Merasa gatal).	Kudis(Scabies)	0.9

14	G4 (Mata sembab, basah, dan berair), G35 (Radang berwarna merah, terutama pada kelopak atau selaput mata), G36 (Bulu sekitar mata basah dan kusam).	Radang Mata	0.8
15	G3 (Sesak nafas atau susah bernafas), G29 (Merasa gatal), G15 (Perut kembung), G16 (Nafsu makan turun).	Bloat(Tympany)	0.5
16	G4 (Mata sembab, basah, dan berair), G35 (Radang berwarna merah, terutama pada kelopak atau selaput mata), G36 (Bulu disekitar mata basah dan kusam), G37 (Keluar air mata terus-menerus)	Radang Mata	0.9
17	G5 (Bulu rontok), G6 (Bulu jadi menggumpal dan kusut), G29 (Gigi berkerot menahan sakit)	Jamur(Ring Worm)	0.4
18	G2 (Badan kurus), G52 (Gatal di sekitar lubang dubur), G53 (Hewan sering menggaruk dan menjilati bulu di sekitar lubang duburnya)	Cacing Gilig (Pinworm/Oxyuris)	0.7
19	G2 (Badan kurus), G41 (Hewan terlihat lemah), G42 (Hewan terlihat pucat)	Cacingan	0.5
20	G2 (Badan kurus), G40 (Gangguan pencernaan akibat	Makan	0.9

	penumpukan bulu disaluran pencernaan), G33 (Muntah bulu), G16 (Nafsu makan turun)	Bulu(Hairball)	
21	G2 (Badan kurus), G42 (Hewan terlihat pucat), G34 (Kotoran ada cacing).	Cacingan	0.9
22	G3 (Sesak nafas atau susah bernafas), G4 (Mata sembab, basah, dan berair), G25 (Kepala sering diangkat tinggi-tinggi), G26 (Hidung keluar nanah, lender berwarna jernih atau keruh), G30 (Bersin-bersin)	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.8
23	G2 (Badan kurus), G11 (Kotorannya hijau gelap dan bau), G12 (Lendir menggantung pada dubur), G19 (Hewan terlihat lesu), G39 (Kotoran lembek cair), G10 (Daerah anus kotor)	Enteritis Kompleks	0.9
24	G1 (Suhu badan naik), G17 (Hewan kesakitan jika disentuh bagian ambing), G18 (Ambing dan puting berwarna merah, mengeras), G44 (Susu diperah rusak/pecah)	Radang Susu(Mastitis)	0.9
25	G2 (Badan kurus), G16 (Nafsu makan turun), G3 (Sesak nafas atau susah nafas).	Radang Paru-paru (Pneumonia/ Pasteurelloises)	0.3

©UKDW