# BAB IV

# PEMBAHASAN

## 4.1 Analisis Studi Kasus

Klinik Aulia adalah Klinik yang meyalani Penyakit ibu dan anak terletak di Desa Kait Kait kecamatan Bati Bati Kabupaten Tanah Laut, Provinsi Kalimantan Selatan, adalah salah 1 Klinik yang ada di desa Kait kait, di desa tersebut ada sekitar 6000 kepala keluarga tetapi untuk pelayanan kesehatan seperti Puskesmas dan Posyandu masih kurang untuk melayani kesehatan penduduk disana, berjarak 40 KM dari Kota terdekat (kota Banjarbaru)

Tetangga desa Kait kait meliputi desa Martadah (disebelah utara), Imban (sebelah selatan), juga memiliki masalah serupa yaitu masalah tentang layanan kesehatan yang kurang baik dikarenakan kurangnya klinik, puskesmas maupun hal lain untuk meningkatkan kesehatan di masing masing desa, sehingga klinik terdekat yang ada di Kait kait ini menjadi incaran dari warga di desa lain.

### 4.1.1 Analisis Pakar

Menjelaskan tentang garis besar seorang pakar yang menjadi sumber dari data yang digunakan penulis dalam merancang sistem ini dan kenapa penulis memilih pakar dalam membangun sistem pakar penyakit anak.

### 4.1.2 Siti Fitriah

Siti fitriah Lahir di Kota Banjarmasin, 5 Oktober 1975, anak dari 4 bersaudara yang sekarang bekerja sebagai Pegawai Negeri Sipil di Kabupaten Tanah Laut, Kalimantan Selatan, Sembari menjadi Pegawai Negeri Sipil beliau juga membuka usaha sebuah klinik Kesehatan bernama Klinik Aulia, beliau menyelesaikan Studi dengan strata terakhir Diplomat 4 dan memiliki pengelaman 20 tahun sebagai pemilik Klinik sejak berdiri tahun 1997 di Desa kait kait sebagai Petugas kesehatan yang memahami seluk beluk penyakit ibu dan anak, karena itulah penulis memilih beliau sebagai sumber data untuk membuat sistem pakar penyakit anak, dalam sehari Klinik Aulia bisa kedatangan lebih dari 50 Pasien yang 60% periksa tentang kesehatan anak.

### 4.1.3 Analisis Penyakit

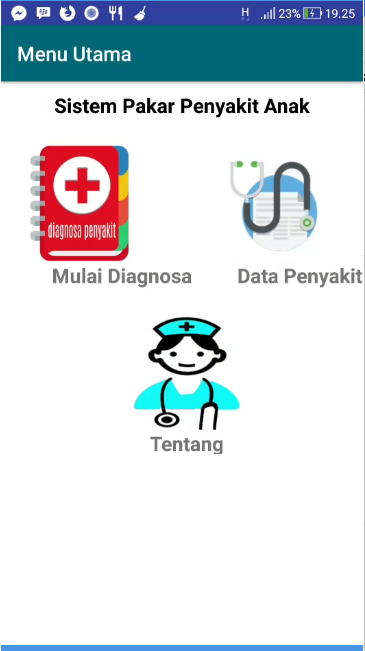
Dalam Sistem ini ada 6 penyakit yang dikenali sistem dari penyakit yang dikenali sistem ini adalah penyakit yang selain umum menyerang anak anak di Studi Kasus penulis juga menyerang banyak daerah di seluruh Indonesia, karena penyakit ini memang umum sekali menyerang anak anak, dan sebagai besar penyakit tak perlu harus kedokter atau kerumah sakit, penyakit yang menyerang anak biasanya bisa sembuh bila anak istirahat cukup dan mendapatkan perawatan oleh pihak keluarga, tetapi penyakit anak ini akan tetap berbahaya jika dibiarkan karena gejala gejala penyakit anak biasanya juga sebagai awal penyakit yang lebih parah, oleh sebab itulah diperlukan penangan bila penyakit tersebut menyerang.

## 4.2 Desain Antar Muka

Desain antar muka menampilkan, tampilan langsung dari sistem pakar penyakit anak :

1. Halaman Menu Utama

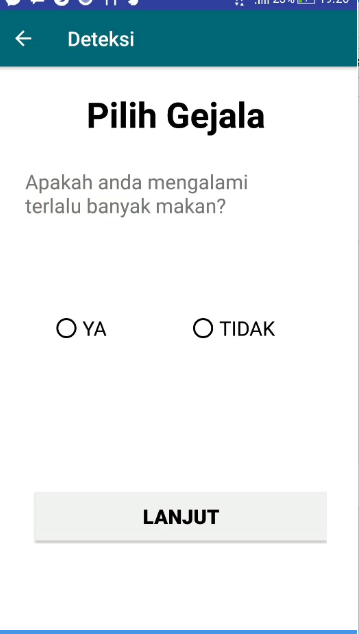
Menu Utama adalah menu awal saat pertama kali menjalankan aplikasi sistem pakar penyakit anak



**Gambar 4.1 Halaman Menu Utama**

1. Halaman Diagnosa

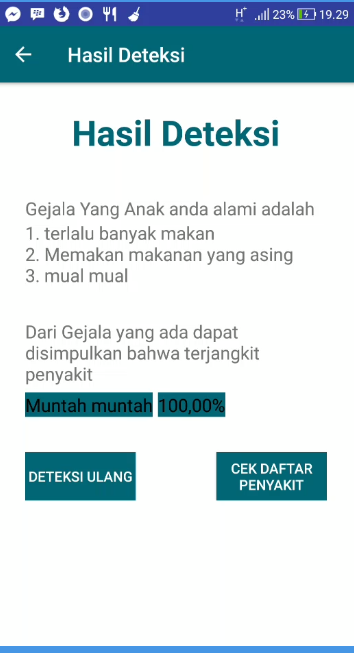
Halaman Diagnosa adalah halaman dimana pengguna diminta memilih gelaja sesuai kondisi anak yang dimana jika menekan YA atau TIDAK akan mengganti gejalanya sesuai metode *Best First Search*



**Gambar 4.2 halaman diagnosa**

1. Halaman Hasil Diagnosa

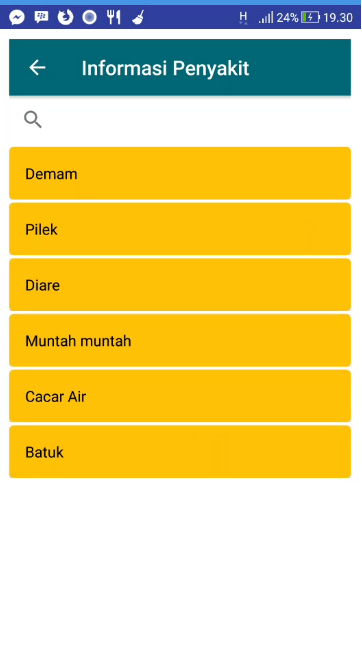
Dari Hasil diagnosa yang dilakukan oleh pengguna maka akan memberikan kesimpulan berupa hasil diagnosa untuk menampilkan nama penyakit dari gejala gejala yang telah dipilih user



**Gambar 4.3 Halaman hasil diagnosa**

1. Halaman Daftar Penyakit

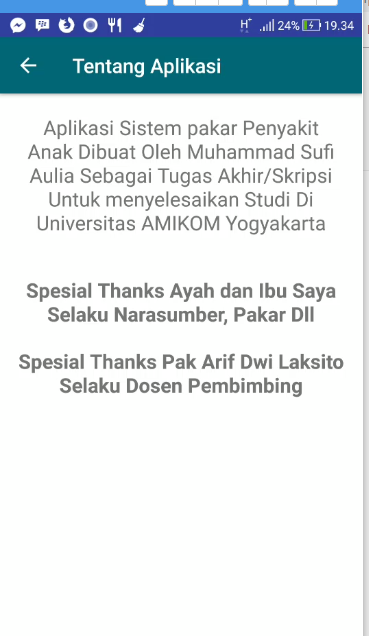
Halaman daftar penyakit menampilkan nama nama penyakit yang dikenali oleh sistem



**Gambar 4.4 Halaman Daftar penyakit**

1. Halaman Tentang

Halaman Tentang berisi informasi tentang tentang aplikasi sistem pakar penyakit anak



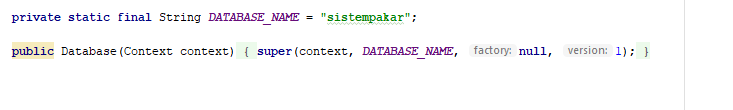
**Gambar 4.5 Halaman Tentang**

## Implementasi *SQlite*

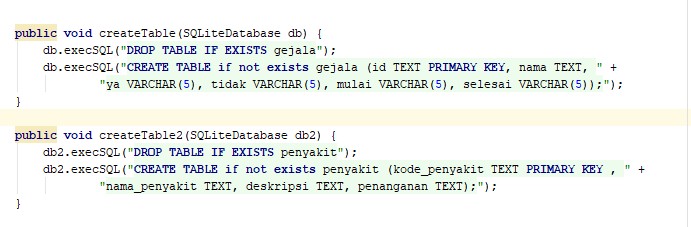
Dalam Proses penyimpanan data Penyakit dan gejala semua dikerjakan dengan menggunakan *SQlite* yang dimana *SQlite* adalah sebuah aplikasi yang termasuk kedalam paket dari aplikasi *Android Studio* , untuk menggunakan SQlite di *Android Studio* dengan membuka *Class Java* baru dengan nama Database (nama tak harus database) lalu include kan *SQliteOpenHelper* untuk deklarasi penggunaan SQlite dalam *Android Studio* seperti gambar 4.6 

**Gambar 4.6 Deklarasi *SQliteOpenHelper***

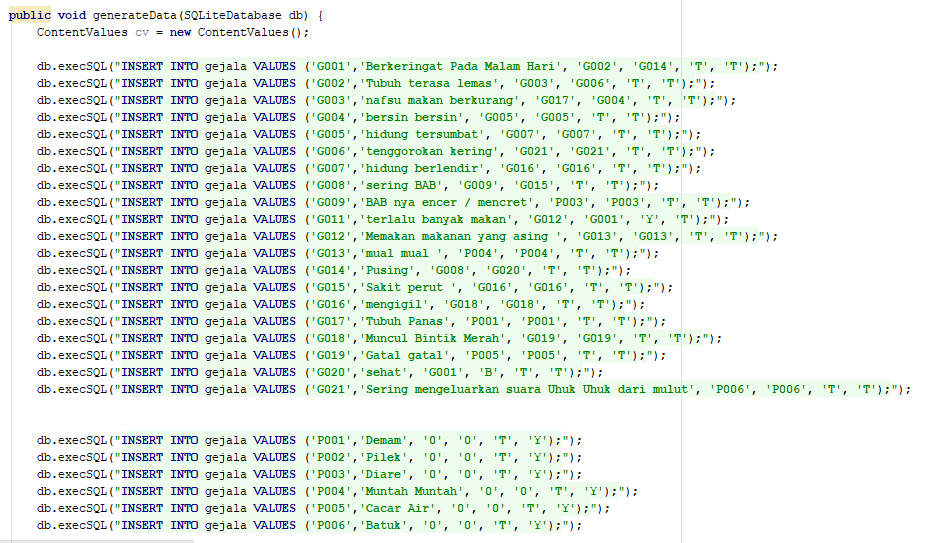
Lalu dilanjutkan dengan membuat method dan variable penamaan database, sistem pakar adalah nama databasenya seperti gambar4.7

,

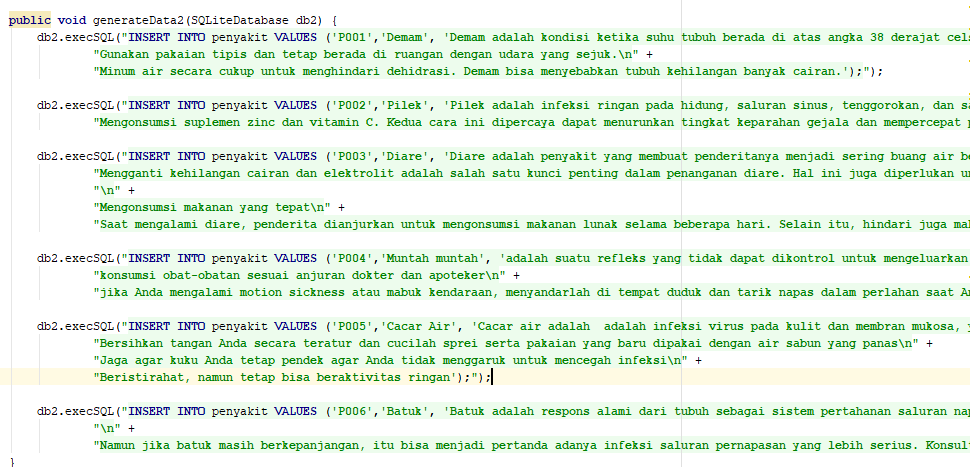
**Gambar 4.7 Deklarasi *Variabel* dan penamaan *Database***

Dilanjutkan dengan Coding SQL Pembuatan Table databasenya seperti gambar 4.8 *create Table*

**Gambar 4.8 *Create Table***

Selesai membuat table dilanjutkan dengan Input data datanya seperti gambar 4.9 dan 4.10**

**Gambar 4.9 *Insert* Data Gejala**

**

**Gambar 4.10 *Insert* data penyakit**

## 4.4 Implementasi Metode Best First Search

Metode Best First Search dalam perancangan sistem pakar penyakit anak membagi gejala gejala dalam sebuah Node Level yang dimana Level 0 belum dimulai, gejala yang paling banyak memiliki penyakit yaitu Gejala dengan Kode Penyakit G011, G001 dan G014 karena memiliki lebih dari 1 penyakit sehingga masuk kedalam Node Level 1, maka pencarian berikutnya adalah pencarian dengan melihat pohon keputusannya seperti terlihat dalam tabel gejala yang memilki *Field* yaitu YA, TIDAK, MULAI dan SELESAI.

*Field* Ya memiliki Value adalah Kode Gejala berapa bila Pengguna menekan YA, misal Gejala G011 di *FIELD* YA memiliki Value G012 maka diagnosa berikutnya akan memunculkan gejala dari kode G012.

TIDAK berisi value kode gelaja saat pengguna dalam melakukan diagnosa memilih TIDAK maka, gejala selanjutnya yang ditampilkan Gejala dengan Kode sesuai Value di field Tidak, missal G011, dalam field TIDAK memiliki value kode gejala G001, maka saat pengguna memilih tidak akan dilanjutkan ke gejala G001

Mulai adalah *Field* adalah *field* untuk menentukan kode gejala mana yang akan ditampilkan di awal diagnosa, disini G011 adalah awal diagnosa maka dari itu hanya gejala dengan kode G011 yang nilai Mulainya adalah “Y”

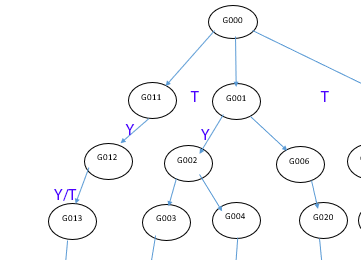
Selesai adalah field akhir, jadi saat diagnosa mencapai di Node dimana memiliki gejala yang memiliki *Value field* selesainya adalah “Y” maka diagnosa selesai dilanjutkan ke halaman hasil diagnosa.

Hasil persentase dari setiap penyakit ditentukan dari berapa banyak dalam 1 penyakit menyimpan berapa gejala, jumlah gejala sesuai pohon keputusan dan juga pembobotan misal Penyakit Pilek di Pohon keputusan ada 5 gejala dan di Tabel pembobotan 1 Gejala bernilai 20% jadi untuk diagnosa hanya memilih 4 gejala saja muncul hasil 80% dengan nama penyakit pilek, untuk menampilkan penyakit mana yang akan ditampilkan adalah penyakit dengan persentase tertinggi misal pengguna memilih gejala yang dimana awalnya ada di Penyakit A tetapi pada akhir diagnosa ternyata memilih banyak penyakit B maka hasil diagnosa hanya menampilkan gejala penyakit B dan menghasilkan Persentase penyakit B saja.

### 4.4.1 Implementasi Metode dalam Program

Implementasi Metode dimulai dari membuat Pohon keputusan sebagai basis pengetahuan untuk menentukan arah arah Node yang akan berjalan, implementasi Metode di Program sistem pakar penyakit anak dilihat dari gambar 4. Bisa dilihat bahwa G000 adalah saat diagnosa belum berjalan, jika diagnosa berjalan dimulai dari G011 jika YA maka dilanjutkan ke G012 jika tidak G001, untuk Node Level selanjutnya dengan Keterangan Y/T maka pengguna memilih apapun akan tetap lanjut ke gejala yang sama, jika pengguna memilih T tetapi dalam node sebelumnya punya cabang ke Node lain maka pilihan gejala akan berpindah ke cabang lainnya perhatikan ilustari gambar 4.11.

Gambar 4.11 Sebagian Pohon Keputusan



**Gambar 4.12 Pohon Keputusan**

Dari pohon keputusan diatas didapatkan basis pengetahuan sebagai berikut

**Tabel 4.1 Basis Pengetahuan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode | Gejala | Ya | Tidak |
| G001 | Berkeringat pada malam hari | G02 | G14 |
| G002 | Tubuh terasa lemas | G03 | G06 |
| G003 | nafsu makan berkurang | G17 | G04 |
| G004 | bersin bersin | G05 | G05 |
| G005 | hidung tersumbat | G07 | G07 |
| G006 | tenggorokan kering | G14 | G14 |
| G007 | hidung berlendir | G16 | G16 |
| G008 | sering BAB | G09 | G15 |
| G009 | BAB nya encer / mencret | P03 | P03 |
| G011 | terlalu banyak makan | G12 | G01 |
| G012 | Memakan makanan yang asing | G13 | G13 |
| G013 | mual mual | P04 | P04 |
| G014 | Pusing | G08 | G08 |
| G015 | Sakit perut | G16 | G16 |
| G016 | mengigil | P02 | P02 |
| G017 | Tubuh Panas | P01 | P01 |
| G018 | muncul bintik merah | G19 | G19 |
| G019 | gatal gatal | P05 | P05 |
| G021 | sering terdengar suara uhuk uhuk dari anak | P06 | P06 |

Sehingga dibuat tabel gejala dengan source coding

**Tabel 4.2 Source Coding Gejala**

|  |  |
| --- | --- |
| NO | Source code |
| 1 | db.execSQL("CREATE TABLE if not exists gejala (id TEXT PRIMARY KEY, nama TEXT, " +  "ya VARCHAR(5), tidak VARCHAR(5), mulai VARCHAR(5), selesai VARCHAR(5));"); |
| 2 | db.execSQL("INSERT INTO gejala VALUES ('G001','Berkeringat Pada Malam Hari', 'G002', 'G014', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G002','Tubuh terasa lemas', 'G003', 'G006', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G003','nafsu makan berkurang', 'G017', 'G004', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G004','bersin bersin', 'G005', 'G005', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G005','hidung tersumbat', 'G007', 'G007', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G006','tenggorokan kering', 'G021', 'G021', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G007','hidung berlendir', 'G016', 'G016', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G008','sering BAB', 'G009', 'G015', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G009','BAB nya encer / mencret', 'P003', 'P003', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G011','terlalu banyak makan', 'G012', 'G001', 'Y', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G012','Memakan makanan yang asing ', 'G013', 'G013', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G013','mual mual ', 'P004', 'P004', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G014','Pusing', 'G008', 'G020', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G015','Sakit perut ', 'G016', 'G016', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G016','mengigil', 'G018', 'G018', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G017','Tubuh Panas', 'P001', 'P001', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G018','Muncul Bintik Merah', 'G019', 'G019', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G019','Gatal gatal', 'P005', 'P005', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G020','sehat', 'G001', 'B', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('G021','Sering mengeluarkan suara Uhuk Uhuk dari mulut', 'P006', 'P006', 'T', 'T');");  db.execSQL("INSERT INTO gejala VALUES ('P001','Demam', '0', '0', 'T', 'Y');");  db.execSQL("INSERT INTO gejala VALUES ('P002','Pilek', '0', '0', 'T', 'Y');");  db.execSQL("INSERT INTO gejala VALUES ('P003','Diare', '0', '0', 'T', 'Y');");  db.execSQL("INSERT INTO gejala VALUES ('P004','Muntah Muntah', '0', '0', 'T', 'Y');");  db.execSQL("INSERT INTO gejala VALUES ('P005','Cacar Air', '0', '0', 'T', 'Y');");  db.execSQL("INSERT INTO gejala VALUES ('P006','Batuk', '0', '0', 'T', 'Y');"); |

Source Coding no 1 dalam tabel 4. Adalah source coding create tabel dan source code no 2 adalah meinginputkan data dari tabel

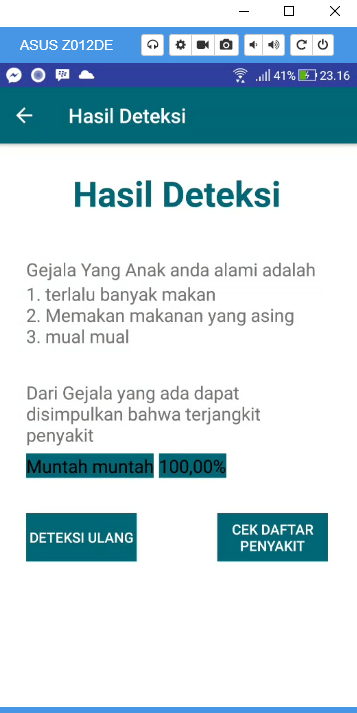
**Tabel 4.3 Keterangan Field Gejala**

|  |  |  |
| --- | --- | --- |
| NO | Nama Field | Keterangan |
| 1 | ID | Menyimpan kode gejala, ID ini yang akan menjadi value dalam penelusuran Best First Search nya |
| 2 | Nama | Nama gejala |
| 3 | ya | Field yang berfungsi sebagai arah gejala selanjutnya bila pengguna memilih ya di radio button lalu lanjut maka akan menampilkan gejala sesuai dengan value kode gejala penyakit lain di fieldnya |
| 4 | tidak | Field yang berfungsi sebagai arah gejala selanjutnya bila pengguna memilih tidak di radio button lalu lanjut maka akan menampilkan gejala sesuai dengan value kode gejala penyakit lain di fieldnya |
| 5 | mulai | Bagi gejala yang memiliki value Y di gejala penyakitnya maka akan menjadi gejala pertama yang akan ditampilkan dalam program ini hanya G011 yang memiliki value Y di fieldnya selain itu maka value field lain adalah T |
| 6 | selesai | Jika didalam field tersebut memiliki value Y maka diagnosa berakhir dan akan menuju ke hasi diagnosa, jika T maka masih ada diagnosa yang tersisa, field yang miliki value Y di program sistem pakar penyakit anak ini adalah kode penyakitnya |

## 4.5 Uji Diagnosa penyakit

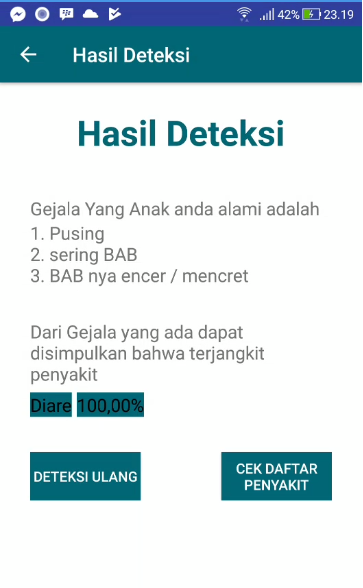
Uji Coba Diagnosa ini akan memilih beberapa pilihan gejala berbeda sebanyak 3 kali untuk membandingkan masing masing hasil persentase.

Percobaan pertama saat melakukan diagnosa hanya memilih Pilihan YA sebanyak 3 kali sehingga didapatkan hasil diagnoa sebagai berikut :



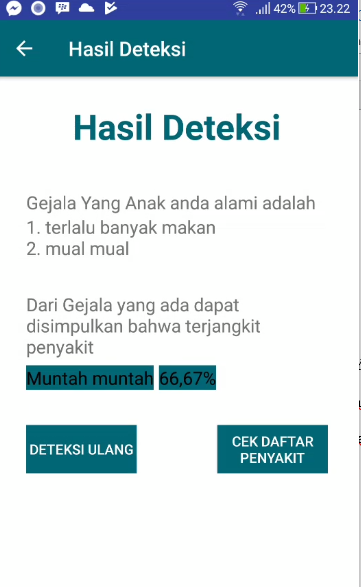
**Gambar 4.11 Percobaan pencarian pertama**

Pada Percobaan Kedua memilih TIDAK sebanyak 2 kali lalu dilanjutkan dengan Memilih YA sampai diagnosa selesai, didapatkan hasil diagnosa sesuai gambar 4.12



**Gambar 4.12 Percobaan pencarian kedua**

Untuk percobaan terakhir akan dilakukan diagnose dengan menekan YA untuk gejala pertama lalu TIDAK untuk gejala kedua, dan YA lagi di gejala ketiga dan TIDAK di gejala keempat dan seterusnya sehingga mendapatkan hasil diagnose, didapatkan hasil diagnosa seperti gambar 4.13



**Gambar 4.13 Percobaan pencarian ketiga**

## 4.6 Pengujian Sistem

Pengujian sistem dilakukan untuk mencari kelemahan, atau kesalahan sistem, sehingga memastikan sistem bebas dari kesalahan maupun meminimalisir kesalahan yang ada. Metode pengujian yang akan dilakukan terdiri dari *black box testing* dan *white box testing*[7].

### 4.6.1 Black Box Testing

Metode *black box* testing dilakukan untuk mencari kesalahan dengan melakukan uji coba pada *interface software*. Pegujian kotak hitam (*black box Testing*) mendemonstrasikan fungsi dari perangkat lunak yang beroperasi, dengan mengecek apakah input sudah bisa diterima dengan baik, dan hasil outputnya sesuai dengan apa yang diharapkan. Uji coba menggunakan metode *black box* ini melakukan pengecekan pada integritas informasi eksternal dimana pengujian hanya memeriksa hasil output yang dihasilkan apakah sudah sesuai dengan apa yang diharapkan dan dinyatakan benar tanpa perlu mengecek logika dari perangkat lunak tersebut[7].

1. Black Box testing Alur Diagnosa

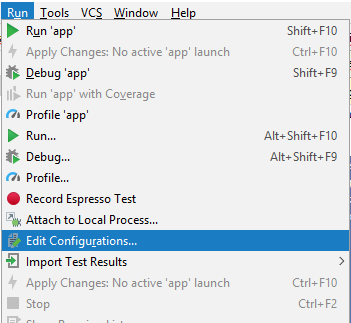
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Pengujian** | **Kondisi Pengujian** | **Harapan** | **Hasil** |
| 1 | Halaman Diagnosa | Memilih ya pada saat diagnosa | Ketika pengguna memilih YA maka akan dilanjutkan ke gejala sesuai pohon keputusan | Valid |
|  | Halaman Diagnosa | Memilih tidak pada saat diagnosa | Ketika pengguna memilih Tidak maka akan dilanjutkan ke gejala sesuai pohon keputusan | Valid |

1. Black Box testing Hasil Diagnosa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Pengujian** | **Kondisi Pengujian** | **Harapan** | **Hasil** |
| 1 | Halaman Hasil Diagnosa | Menyelesaikan semua gejala diagnosa | Hasil diagnosa menampilkan nama penyakit sesuai gejala yang telah dipilih | Valid |
| 2 | Halaman Hasil Diagnosa | Menyelesaikan semua gejala diagnosa | Hasil diagnosa memberikan persentase sesuai dengan kondisi saat pemilihan gejala | Valid |

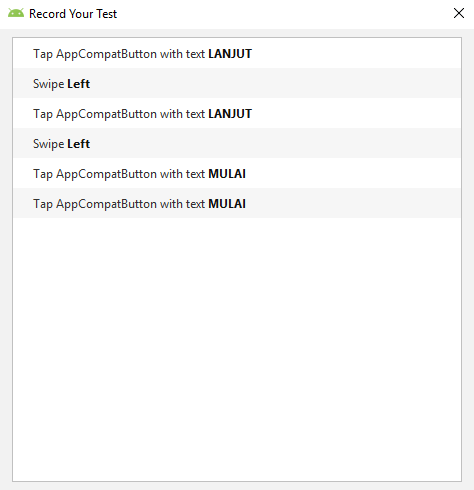
### 4.6.2 White Box Testing

White box testing disini akan menggunakan Tools yang disediakan di *Android Studio yaitu, Ekspresso unit testing*  dari tools ini akan melakukan test terhadap sistem secara langsung saat digunakan, lalu menampilkan dalam sebuah window apa apa saja yang sudah bisa digunakan. Untuk melakukan testing ini cukup RUN program dengan Tools *Record Eskpresso Test*  di Menu RUN



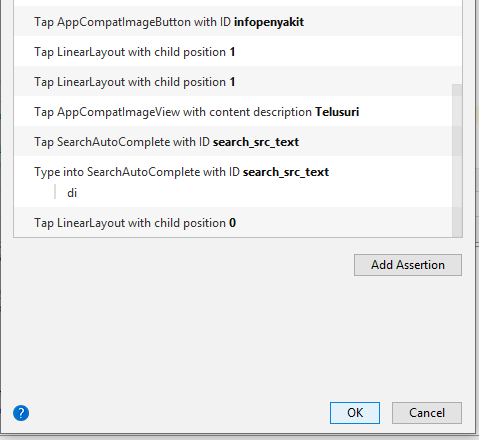
**Gambar 4.14 *Run Record Espresso Test***

Hasil Testing dengan Ekspresso untuk pengujian Home Welcome, terlihat pada gambar dibawah ini bahwa Pengujian Tombol Lanjut, Swipe dan Mulai Berjalan



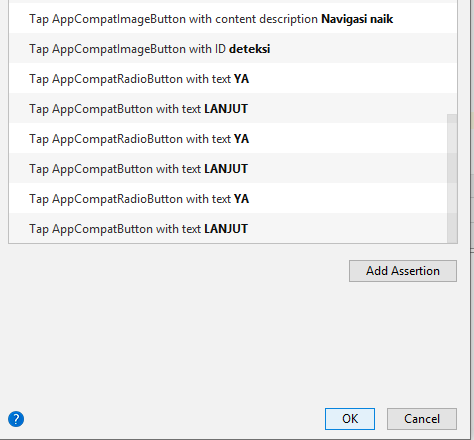
**Gambar 4.15 *Unit Testing Welcome Home***

Pada Pengujian di Data penyakit Tombol Info penyakit, terlihat seperti gambar dibawah ini ImageButton info penyakit berjalan, tombol pencarian juga berjalan.



**Gambar 4.16 *Unit Testing* Data Penyakit**

Pada pengujian Ekspresso saat melakukan diagnosa. Terlihat ImageButtonNavigasiDiNaik berfungsi untuk melakukan kembali ke Page sebelumnya, lalu melakukan diagnosa dengan cara memilih YA di setiap Gejala, seperti terlihat pada gambar dibawah ini.



**Gambar 4.17 *Unit Testing Ekspresso* Diagnosa Penyakit**

### 4.6.3 Kesimpulan Hasil Pengujian

Dari Hasil Pengujian *Black box* dan *White box* Testing didapatkan hasil bahwa sistem pakar penyakit anak, berhasil berjalan dengan metode *Best First Search ,* terlihat dari tak ditemukan error saat melakukan Pengujian dari sisi *Black box* dan *White box Testing*.