

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Masalah

3.1.1 Identifikasi Masalah

Menurut Wahyudi, Panggabean, dan Pujiyanto (2008), secara umum, rata-rata produktivitas kakao/cokelat di Indonesia sebesar 900 kg/ha/tahun, angka ini masih jauh di bawah rata-rata potensi yang diharapkan, yakni sebesar 2.000 kg/ha/tahun. Rendahnya produktivitas tanaman kakao, mayoritas disebabkan antara lain karena penggunaan bahan tanaman yang kurang baik, teknologi budi daya yang kurang optimal, umur tanaman, serta masalah serangan hama dan penyakit sehingga kerugian yang disebabkan oleh hama penggerek buah kakao (PBK) tersebut bisa mencapai 5-80%.

Faktor utama yang menyebabkan rendahnya kualitas cokelat di Indonesia terutama pada perkebunan PT. Panglungan yang terletak di Kec. Wonosalam, adalah karena selalu diancam oleh hama dan penyakit tumbuhan yang menyerang pada bagian akar, batang, daun, buah dan tanaman cokelat yang masih kecil dengan timbulnya beberapa gejala di masing-masing bagian pada tanaman cokelat tersebut. Kurangnya informasi yang diketahui oleh pihak perkebunan cokelat tentang jenis penyakit yang menyerang tanaman cokelat, menyebabkan banyaknya tanaman cokelat yang tidak tertangani dengan benar sehingga mengakibatkan banyak tanaman cokelat yang seharusnya bisa diselamatkan menjadi mati dan kualitas cokelat tersebut menurun dan akan berimbas pada tingkat produktivitas tanaman cokelat tersebut.

Oleh sebab itu, pihak perkebunan coklat membutuhkan sebuah sistem pakar yang dapat memberikan informasi mengenai penyakit yang menyerang tanaman coklat dan memberikan solusi untuk menangani penyakit tersebut. Sehingga dibuatlah sebuah aplikasi sistem pakar yang bertujuan untuk membantu pihak perkebunan coklat agar dapat mengetahui jenis penyakit yang sedang menyerang tanaman coklat berdasarkan pada gejala-gejala penyakit yang terlihat di masing-masing bagian tanaman coklat dan dengan adanya aplikasi sistem pakar ini, dapat menghasilkan solusi untuk mengambil sebuah tindakan yang tepat dalam menangani tanaman coklat yang terserang penyakit.

3.2 Perancangan Sistem

Dalam pembuatan aplikasi sistem pakar untuk menentukan jenis penyakit pada tanaman coklat, diperlukan langkah awal, yaitu menganalisis masalah. Pada bagian ini, akan diuraikan kebutuhan dasar sistem dalam membuat aplikasi sistem pakar ini agar dapat membentuk suatu sistem berbasis aturan yang memenuhi syarat.

Langkah pertama yang harus dilakukan adalah membuat *dependency diagram*. Dalam aplikasi sistem pakar ini, *dependency diagram* dibuat atau direpresentasikan dalam bentuk *tree view*. *Dependency diagram* berguna untuk menggambarkan susunan parameter yang ada. Di dalam sebuah *dependency diagram* minimal terdapat dua parameter (setiap parameter tidak boleh memiliki nama yang sama) dan sebuah parameter minimal memiliki dua *possible value* yang berbeda. Parameter yang memiliki cabang dibawahnya secara otomatis disebut sebagai set. Pengguna harus memasukan sebuah pertanyaan pada setiap parameter yang bukan set (parameter yang tidak memiliki cabang dibawahnya).

Pertanyaan ini nantinya akan ditampilkan kepada pengguna yang melakukan konsultasi untuk memperoleh hasil untuk memperoleh hasil penentuan penyakit pada tanaman cokelat.

Setelah pembuatan *dependency diagram* selesai dilakukan, dilanjutkan dengan mengisi *decision table*, setiap set dalam *dependency diagram* memiliki *decision table* tersendiri. *Decision table* pada aplikasi ini akan dibuat otomatis oleh sistem berdasarkan set yang telah dipilih oleh pengguna. Jumlah baris pada setiap *decision table* diperoleh dengan cara mengalikan jumlah *possible value* dari setiap parameter yang berada dalam satu set yang dipilih pengguna (lihat Tabel 2.1, *step 1*).

Isi dari *decision table* kemudian oleh sistem dibangkitkan jadi beberapa *rule*. Karena proses kombinasi *possible value* pada *decision table* dilakukan oleh sistem, maka tidak mungkin terjadi *redundant rule*, *subsumed rule* dan *circular rule*. Setiap baris pada *decision table* hanya biasa diisi dengan satu *conclusion* atau kesimpulan, sehingga tidak mungkin terdapat *conflicting rule*, kesalahan yang mungkin terjadi adalah *unnecessary if condition*. Namun hal ini dapat diatasi dengan adanya proses *reduction decision table* yang ada pada setiap pengisian *decision table*. Proses *reduction* ini secara otomatis mencari dan menghapus *premise* atau kondisi yang tidak perlu dikondisikan. Jadi *rule* hasil *generate* dari aplikasi ini akan bebas dari kesalahan yang ada.

Setelah proses pembuatan *rule*, dilanjutkan dengan mengisi solusi atau penjelasan dari setiap kesimpulan akhir yang mungkin. Kesimpulan akhir ini diambil dari *possible rule* yang dimiliki oleh parameter paling atas pada *rule view* atau parameter yang paling ujung pada sebuah *dependency diagram*. Dan pada

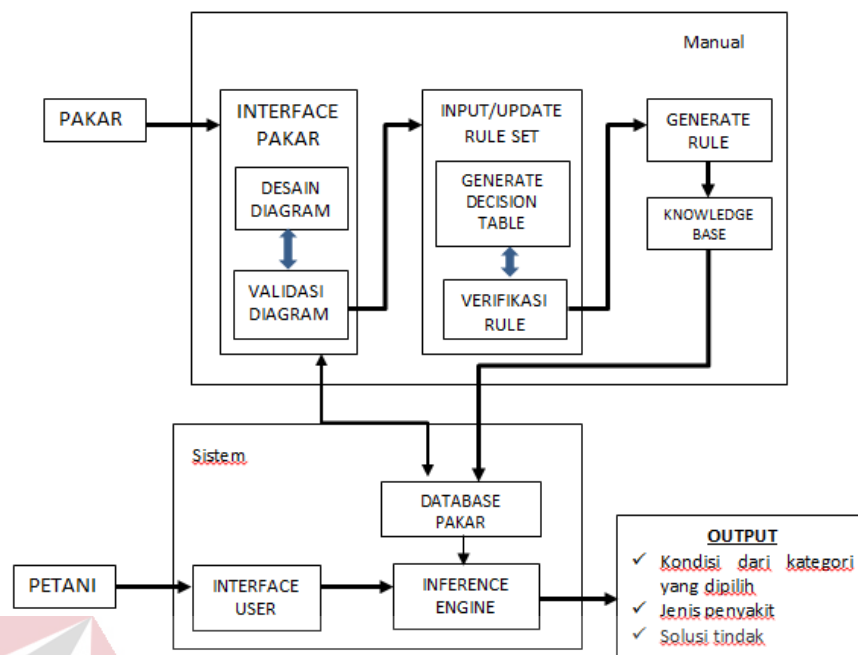
aplikasi sistem pakar ini, kesimpulannya adalah jenis penyakit tanaman cokelat dan penjelasan yang diisi adalah penyebab dari penyakit tersebut, gejala-gejala dan cara pengendaliannya.

Proses selanjutnya adalah pengguna dapat melakukan konsultasi untuk mendapatkan hasil akhir atau output dari aplikasi sistem pakar ini. Untuk melakukan konsultasi, pengguna harus memilih terlebih dahulu bagian dari tanaman cokelat yang tampak gejalanya, misalnya daun, maka sistem akan menampilkan secara otomatis semua gejala-gejala yang terlihat pada bagian daun. Selanjutnya adalah proses menentukan jenis penyakit pada tanaman cokelat untuk menampilkan hasil akhir, yaitu hasil menentukan jenis penyakit, sesuai dengan gejala yang telah dipilih.

Aplikasi sistem pakar untuk menentukan jenis penyakit pada tanaman cokelat ini berbasis *mobile application android* dan memiliki satu pengguna, yaitu pengguna (*user*).

3.2.1 Desain Arsitektur

Desain arsitektur aplikasi sistem ini tidak bisa dirubah *rule* yang terdapat pada aplikasi sistem pakar karena bersifat statis. Pada desain arsitektur *case* yang menjelaskan elemen-elemen utama dalam sistem dan hubungan antara elemen-elemen tersebut untuk menghasilkan *output* berupa *rule based system*. Desain arsitektur secara umum ini dapat dilihat pada Gambar 3.1 di halaman 48. Desain *case* yang dibuat secara garis besar terdiri dari dua desain utama, yaitu desain untuk pakar dilakukan secara manual dan desain sistem untuk user.



Gambar 3.1 Desain Arsitektur Diagnosis Penyakit Pada Tanaman Cokelat

Berikut penjelasan dari desain arsitektur :

1. *Interface* pakar : media yang digunakan oleh pakar untuk mengembangkan sistem. Dimana dalam merancang sistem dengan membuat desain *Dependency Diagram* pada kanvas gambar secara manual. *Interface* pakar terdiri dari dua proses utama, yaitu :
 - a. Desain diagram, merupakan proses pembuatan *dependency diagram* pada kanvas gambar.
 - b. Validasi diagram, merupakan proses pengecekan terhadap relasi dari tiap komponen dalam desain diagram.
2. *Input/update rule set* : proses penentuan parameter-parameter yang digunakan dalam tiap *rule set*. Proses ini meliputi dua subproses yaitu :
 - a. *Generate decision table*, ini merupakan proses untuk membentuk tabel keputusan (*decision table*) dari setiap *rule set* pada desain *dependency diagram*.

- b. Verifikasi *rule*, merupakan proses pengecekan terhadap *rule-rule* yang terbentuk pada *decision table* apakah sudah memenuhi konstrain-konstrain yang telah ditetapkan dalam sistem.
3. *Generate rule* : proses *generate rule* dijalankan untuk membentuk *rule* dalam tiap *rule set* berdasarkan *decision table* yang terbentuk.
4. *Database pakar* : digunakan untuk menampung sementara informasi *rule* pada *project* yang sedang dijalankan.
5. *Interface user* : sebagai media oleh *user* untuk melihat dan berinteraksi dengan sistem pada saat proses konsultasi.
6. *Inference engine* : mekanisme inferensi yang digunakan adalah *forward chaining* karena sistem lebih dahulu mengetahui fakta-fakta yang ada, kemudian mencari kesimpulan sementara sampai akhirnya berhenti setelah menghasilkan sebuah kesimpulan akhir.
Proses *forward chaining* diperlukan dalam mencari solusi berdasarkan *goal* konsultasi dan *rule base* yang ada dalam *working memory*.
7. *Knowledge base* : kumpulan fakta dan aturan (*rule*) serta *working memory* yang merupakan fakta yang diperoleh oleh sistem selama proses berlangsung.
8. *Output* :
 - a. *Output* dari desain user adalah hasil akhir dari proses *inference* berupa laporan hasil konsultasi user, yaitu jenis penyakit dan tidak penanganan pada tanaman coklat

3.2.2 Perancangan Sistem Pakar

Dalam melakukan perancangan sistem pakar ini, beberapa tahap yang harus dilakukan agar aplikasi sistem pakar yang dibuat dapat berfungsi sesuai dengan yang diharapkan. Adapun tahap-tahap dalam perancangan sistem pakar adalah perancangan *block diagram*, *dependency diagram*, *decision table*, *reduced decision table* dan *use case*, *activity diagram*,.

A. Perancangan *Block Diagram*

Block diagram diperlukan untuk mengetahui urutan kerja sistem dalam mencari suatu keputusan. Dalam aplikasi sistem pakar ini terdapat sebuah *block diagram*, yaitu *block diagram* untuk menentukan jenis penyakit pada tanaman coklat.

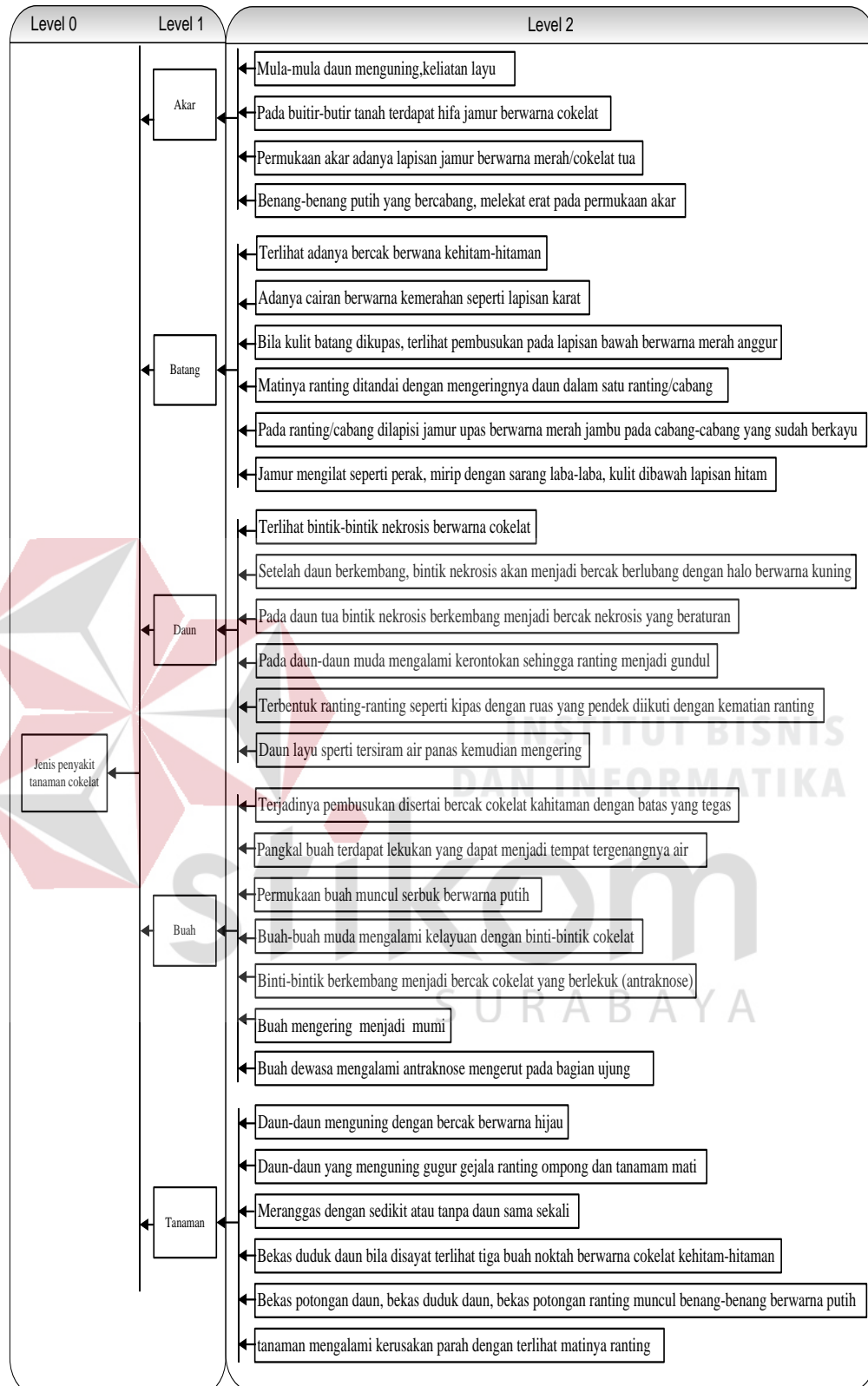
Block diagram ini terdiri dari tiga level, yaitu level 0, level 1 dan level 2. Level 0 berisi hasil akhir berupa jenis penyakit yang menyerang tanaman coklat. Pada level 1 terdapat 5 parameter, yaitu akar, batang, daun, buah dan tanaman. Terakhir pada level 2 dijelaskan sub parameter masing-masing parameter berupa gejala-gejala penyakit yang menyerang tanaman coklat dapat dilihat pada Gambar 3.2 di halaman 52.

B. Perancangan *Dependency Diagram*

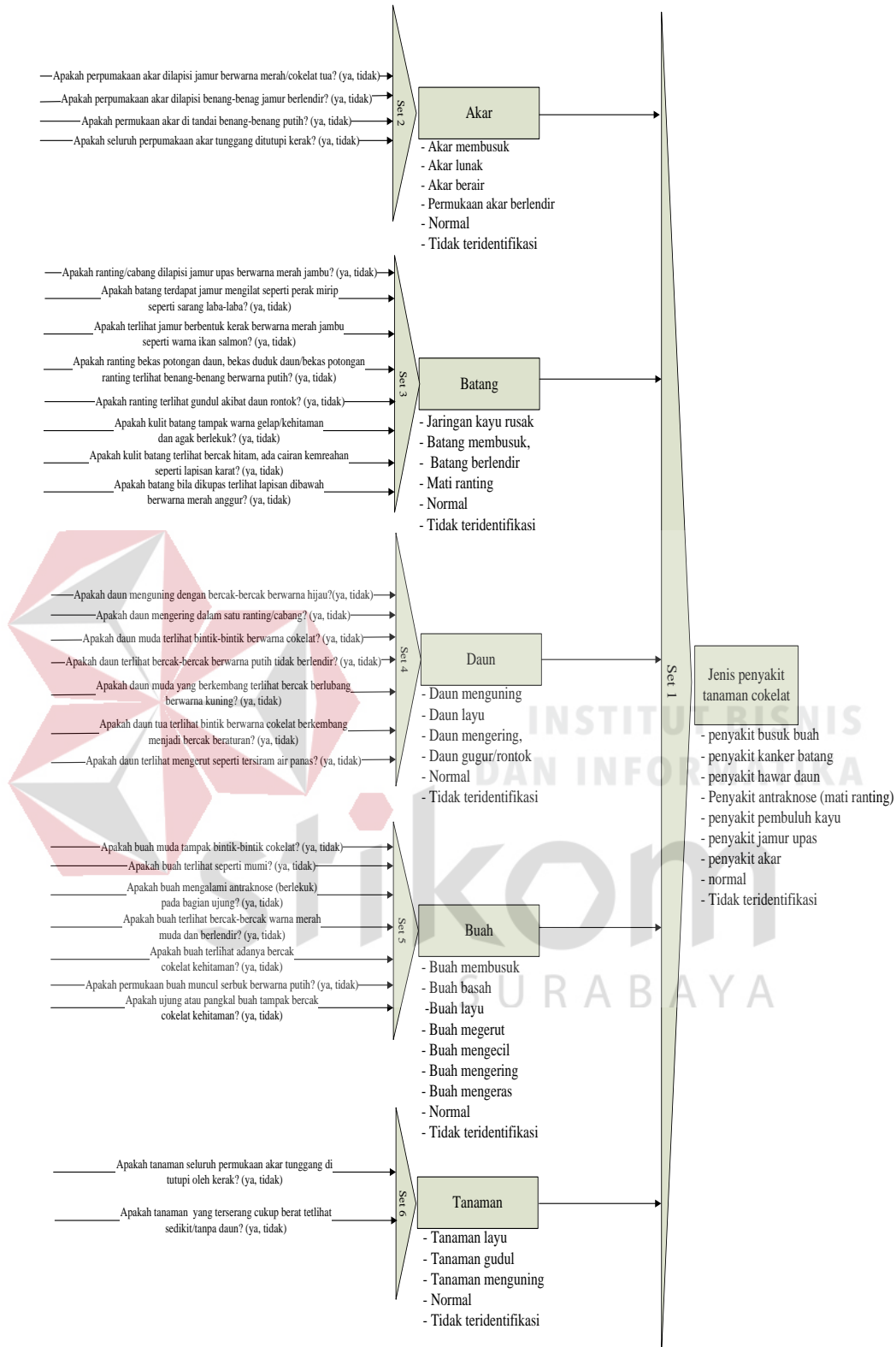
Dependency diagram menunjukkan hubungan atau ketergantungan antara inputan pertanyaan, rule, nilai dan rekomendasi yang dibuat oleh *prototype Knowledge Based System (KBS)*. *Dependency diagram* untuk aplikasi sistem pakar ini dapat dilihat pada Gambar 3.3 di halaman 53.

Dependency diagram pada Gambar 3.3 memiliki enam set. Set 1 yang merupakan konklusi akhir jenis penyakit dari tanaman cokelat. Jenis penyakit ini adalah penyakit busuk buah, penyakit kanker batang, penyakit hawar daun, penyakit *antraknose* (mati ranting), penyakit pembuluh kayu, penyakit jamur upas dan penyakit akar. Set 2 merupakan set dari parameter akar, set 3 merupakan parameter batang, set 4 merupakan parameter daun, set 5 merupakan set dari parameter buah, set 6 merupakan set dari parameter tanaman.





Gambar 3.2 Block Diagram
Menentukan Penyakit Pada Tanaman Cokelat



Gambar 3.3 *Dependency Diagram*
Menentukan Penyakit Pada Tanaman Cokelat

C. Perancangan Decision Table

Decision table diperlukan untuk menunjukkan hubungan timbal balik antara nilai-nilai pada hasil *fase* atau rekomendasi akhir *knowledge based base system* (KBS). Table 3.1 menunjukkan salah satu perancangan *decision table* aplikasi sistem pakar untuk menentukan penyakit pada tanaman cokelat dengan *set 2*, yaitu *set* parameter akar berdasarkan pada perancangan *dependency diagram*.

Tabel 3.1 adalah rangkaian aturan akhir yang terkait dengan enam kondisi, yaitu akar membusuk, akar lunak, akar berair, permukaan akar berlendir, normal dan tidak teridentifikasi. Masing-masing kondisi dapat memiliki sejumlah nilai yang berbeda. permukaan akar dilapisi jamur berwarna merah/cokelat tua adalah kondisi pertama yang memiliki dua nilai, yaitu ya dan tidak. Permukaan akar dilapisi benang-benang jamur berlendir adalah kondisi kedua yang memiliki dua nilai, yaitu ya dan tidak. Permukaan akar di tandai benang-benang putih adalah kondisi ketiga yang memiliki dua nilai, yaitu ya dan tidak. Seluruh permukaan akar tunggang ditutupi kerak adalah kondisi keempat yang memiliki dua nilai, yaitu ya dan tidak. Untuk lebih jelas dapat dilihat pada Tabel 3.1 di halaman 55.

Tabel 3.1 *Decision Table Rule Set 2*

Step 1: <i>Plan</i>					
Kondisi	perpustakaan akar dilapisi jamur berwarna merah/cokelat tua (ya. tidak)				2
	perpustakaan akar dilapisi benang-benang jamur berlendir (ya. tidak)				2
	permukaan akar di tandai benang-benang putih (ya. tidak)				2
	seluruh perpustakaan akar tunggang ditutupi kerak (ya. tidak)				2
Baris	$2 \times 2 \times 2 \times 2 = 16$				
Step 2: <i>Completed decision table</i>					
Rule	Permukaan akar dilapisi jamur berwarna merah/cokelat tua Kolom 1	Permukaan akar dilapisi benang-benang jamur berlendir Kolom 2	Permukaan akar di tandai benang-benang putih Kolom 3	Tanaman seluruh permukaan akar tunggang ditutupi oleh kerak Kolom 4	Akar Kolom 5
A1	Y	Y	Y	Y	Tidak teridentifikasi
A2	Y	Y	Y	T	Tidak teridentifikasi
A3	Y	Y	T	Y	Tidak teridentifikasi
A4	Y	Y	T	T	Tidak teridentifikasi
A5	Y	T	Y	Y	Tidak teridentifikasi
A6	Y	T	Y	T	Tidak teridentifikasi
A7	Y	T	T	Y	Tidak teridentifikasi
A8	Y	T	T	T	Berair
A9	T	y	Y	Y	Tidak teridentifikasi
A10	T	y	Y	T	Tidak teridentifikasi

Tabel 3.1 *Decision Table Rule Set 2* (Lanjutan)

Rule	Permukaan akar dilapisi jamur berwarna merah/cokelat tua Kolom 1	Permukaan akar dilapisi benang-benang jamur berlendir Kolom 2	Permukaan akar ditandai benang-benang putih Kolom 3	Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak Kolom 4	Akar Kolom 5
A11	T	y	T	Y	Tidak teridentifikasi
A12	T	y	T	T	Berlendir
A13	T	T	Y	Y	Tidak teridentifikasi
A14	T	T	Y	T	Lunak
A15	T	T	T	Y	Membusuk
A16	T	T	T	T	Normal

Complete decision table merupakan penjabaran atau langkah selanjutnya setelah langkah 1 yaitu *plan* pada bab III. Tabel 3.2 menjelaskan *decision table rule* akar. Tabel 3.2 merupakan contoh dari *Complete decision table*, untuk *Complete decision table* lainnya dapat dilihat pada lampiran di halaman 118-138.

D. Perancangan *Reduced Decision Table*

Reduced decision table adalah pembuatan tabel yang nilainya didapat dari mereduksi *decision table*. Setelah didapatkan nilai dari kondisi terakhir. Pada sistem ini perancangan *reduced decision table* untuk setiap *decision table* dilakukan secara manual. Perancangan *reduced decision table* pada Tabel 3.1 menghasilkan parameter seperti yang dapat dilihat pada Tabel 3.2. Untuk lebih jelas dapat dilihat pada Tabel 3.2 di halaman 57.

Tabel 3.2 *Reduced Decision Table Rule Set 2*

Rule	Permukaan akar dilapisi jamur berwarna merah/cokelat tua Kolom 1	Permukaan akar dilapisi benang-benang jamur berlendir Kolom 2	Permukaan akar di tandai benang-benang putih Kolom 3	Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak Kolom 4	Kesimpulan Kolom 5
B1	Y	Y	-	-	Tidak teridentifikasi
B2	Y	T	Y	-	Tidak teridentifikasi
B3	Y	T	T	Y	Tidak teridentifikasi
B4	Y	T	T	T	Berair
B5	T	Y	Y	-	Tidak teridentifikasi
B6	T	Y	T	Y	Tidak teridentifikasi
B7	T	Y	T	T	Berlendir
B8	T	T	Y	Y	Tidak teridentifikasi
B9	T	T	Y	T	Lunak
B10	T	T	T	Y	Membusuk
B11	T	T	T	T	Normal

Proses verifikasi merupakan proses pengecekan aturan yang bertujuan untuk menghindari kesalahan sehingga dapat diimplementasikan dengan benar. Pada sistem ini proses verifikasi dilakukan secara manual yaitu pada proses reduksi *decision table*.

Proses penyederhanaan dari *decision table* ke *reduced decision table* adalah sebagai berikut :

1. *Rule* A1, A2, A3 dan A4 dari *decision table* memiliki kondisi akar yang sama yaitu tidak teridentifikasi, memiliki gejala permukaan akar di tandai benang-benang putih dan seluruh permukaan akar tunggang ditutupi kerak yang sama, dapat di jawab ya atau tidak. Variasi ya dan tidak pada kolom 3 dan 4 tidak mempengaruhi kondisi akar sehingga dapat di sederhanakan menjadi *reduced decision table rule* = B1.
2. *Rule* A5 dan A6 dari *decision table* memiliki kondisi yang sama yaitu tidak teridentifikasi memiliki gejala seluruh permukaan akar tunggang ditutupi kerak yang sama dan dapat di jawab ya atau tidak. Variasi ya dan tidak pada kolom 4 tidak mempengaruhi kondisi akar sehingga dapat di sederhanakan menjadi *reduced decision table rule* = B2.
3. *Rule* A7 dari *decision table* tidak dapat di sederhanakan karena tidak memiliki jawaban yang berpasangan dan tidak mempengaruhi hasil kondisi akar yang sama dengan baris yang berdekatan sehingga dapat disederhanakan menjadi *reduced decision table rule* = B3.
4. *Rule* A8 dari *decision table* tidak dapat di sederhanakan karena tidak memiliki hasil kondisi akar yang sama dengan baris yang berdekatan yaitu menghasilkan kondisi akar berair sehingga dapat disederhanakan menjadi *reduced decision table rule* = B4.
5. *Rule* A9 dan A10 dari *decision table* memiliki kondisi yang sama yaitu tidak teridentifikasi memiliki gejala seluruh permukaan akar tunggang ditutupi kerak yang sama dan dapat di jawab ya atau tidak. Variasi ya dan tidak pada kolom 4 tidak mempengaruhi kondisi akar

sehingga dapat di sederhanakan menjadi *reduced decision table rule* = B5.

6. *Rule A11* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki jawaban yang berpasangan dan tidak mempengaruhi hasil kondisi akar yang sama dengan baris yang berdekatan sehingga dapat disederhanakan menjadi *reduced decision table rule* = B6.
7. *Rule A12* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki hasil kondisi akar yang sama dengan baris yang berdekatan yaitu menghasilkan kondisi akar berlendir sehingga dapat disederhanakan menjadi *reduced decision table rule* = B7.
8. *Rule A13* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki jawaban yang berpasangan dan tidak mempengaruhi hasil kondisi akar yang sama dengan baris yang berdekatan sehingga dapat disederhanakan menjadi *reduced decision table rule* = B8.
9. *Rule A14* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki hasil kondisi akar yang sama dengan baris yang berdekatan yaitu menghasilkan kondisi akar linak sehingga dapat disederhanakan menjadi *reduced decision table rule* = B9.
10. *Rule A15* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki hasil kondisi akar yang sama dengan baris yang berdekatan yaitu menghasilkan kondisi akar busuk sehingga dapat disederhanakan menjadi *reduced decision table rule* = B10.
11. *Rule A116* dari *decision table* tidak dapat di sederhanakan karena tidak memiliki hasil kondisi akar yang sama dan memiliki jawaban yaitu

tidak dan menghasilkan kondisi akar normal sehingga dapat disederhanakan menjadi *reduced decision table rule* = B11

E. Perancangan Rule Base

Pada pengembangan *rule base* telah direpresentasikan dalam bentuk *block diagram* di atas kemudian diimplementasikan dalam bentuk *list* aturan yaitu struktur berbasis pengetahuan. Suatu *rule* terdiri dari dua bagian utama, yaitu *IF* yang sering disebut premis atau kondisi dan bagian *THEN* yang sering disebut *konklusi* atau kesimpulan. Berikut ini merupakan contoh struktur basis pengetahuan yang sebelumnya telah dirancang menggunakan *block diagram*.

Rule base dari tabel 3.2 *Reduced decision table* pada set 2 akar *rule* B1-B11 yang akan diimplementasikan pada program sebagai berikut:

1. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua ==“Ya”

AND Permukaan akar dilapisi benang-benang jamur berlendir ==“Ya”)

//B1

THEN ==”Tidak teridentifikasi”

2. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua ==“Ya”

AND Permukaan akar dilapisi benang-benang jamur berlendir ==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Ya”)/B2

THEN= “Tidak teridentifikasi”

3. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Ya”

AND Pemukaan akar dilapisi benang-benang jamur berlendir
 ==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Tidak”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
 ==”Ya”)//B3

THEN = ”Tidak teridentifikasi”

4. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Ya”

AND Pemukaan akar dilapisi benang-benang jamur berlendir
 ==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Tidak”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
 ==”Tidak”)//B4

THEN= ”Berair”

5. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Tidak”

AND Pemukaan akar dilapisi benang-benang jamur berlendir ==”Ya”

AND Permukaan akar di tandai benang-benang putih ==”Ya”)//B5

THEN=”Tidak teridentifikasi”

6. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Tidak”

AND Pemukaan akar dilapisi benang-benang jamur berlendir ==”Ya”

AND Permukaan akar di tandai benang-benang putih ==”Tidak”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
==”Ya”)//B6

THEN=”Tidak teridentifikasi”

7. *IF* Permukaan akar dilapisi jamur berwarna merah/cokelat tua = Tidak

AND Permukaan akar dilapisi benang-benang jamur berlendir = Ya

AND Permukaan akar di tandai benang-benang putih = Tidak

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
= Tidak

THEN Berlendir /B7

8. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua

==”Tidak”

AND Permukaan akar dilapisi benang-benang jamur berlendir

==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Ya”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
==”Ya”)//B8

THEN=”Tidak teridentifikasi”

9. *IF* (Permukaan akar dilapisi jamur berwarna merah/cokelat tua

==”Tidak”

AND Permukaan akar dilapisi benang-benang jamur berlendir

==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Ya”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
 ==”Tidak”)//B9
 THEN=”Lunak”

10. IF (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Tidak”

AND Permukaan akar dilapisi benang-benang jamur berlendir
 ==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Tidak”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
 ==”Ya”)//B10

THEN =”Membusuk”

11. IF (Permukaan akar dilapisi jamur berwarna merah/cokelat tua
 ==”Tidak”

AND Permukaan akar dilapisi benang-benang jamur berlendir
 ==”Tidak”

AND Permukaan akar di tandai benang-benang putih ==”Tidak”

AND Tanaman seluruh permukaan akar tunggang di tutupi oleh kerak
 ==”Tidak”) //B11

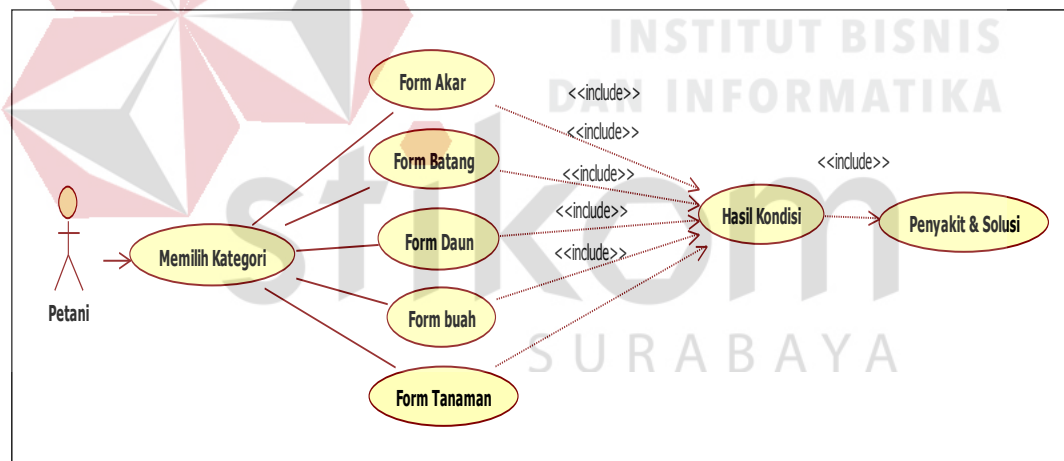
THEN=”Normal”

3.2.3 Use Case Diagram

Use case diagram digunakan untuk menspesifikasikan apa yang dapat dilakukan oleh sistem atau untuk menspesifikasikan kebutuhan fungsional utama dari sistem. Berikut akan dijelaskan *use case diagram* aplikasi *user* pada *mobile android* untuk masing-masing sistem.

1. Use Case Diagram Aplikasi User pada *Mobile Android*

Berikut ini adalah diagram *use case* dari aplikasi *user* yang dapat dilihat pada gambar 3.4. Dalam gambar tersebut bisa dilihat terdapat 1 aktor memiliki sifat yang sama sehingga digeneralisasi menjadi aktor *User* (petani) serta memiliki 8 *use case* yang berinteraksi dengan *User*. *Use case* tersebut yaitu, *use case* pilih kategori yang dimana akan ditampilkan *use case form* akar, *form* batang, *form* daun, *form* buah dan *form* tanaman yang masing-masing *form* berisi gejala atau pertanyaan yang akan di jawab ya atau tidak selanjutnya sistem akan menampilkan hasil kondisi dari kategori yang dipilih beserta nama penyakit dan hasil akhir *use case* jenis penyakit dan solusi.



Gambar 3.4 Use Case diagram aplikasi *user* pada *mobile android*

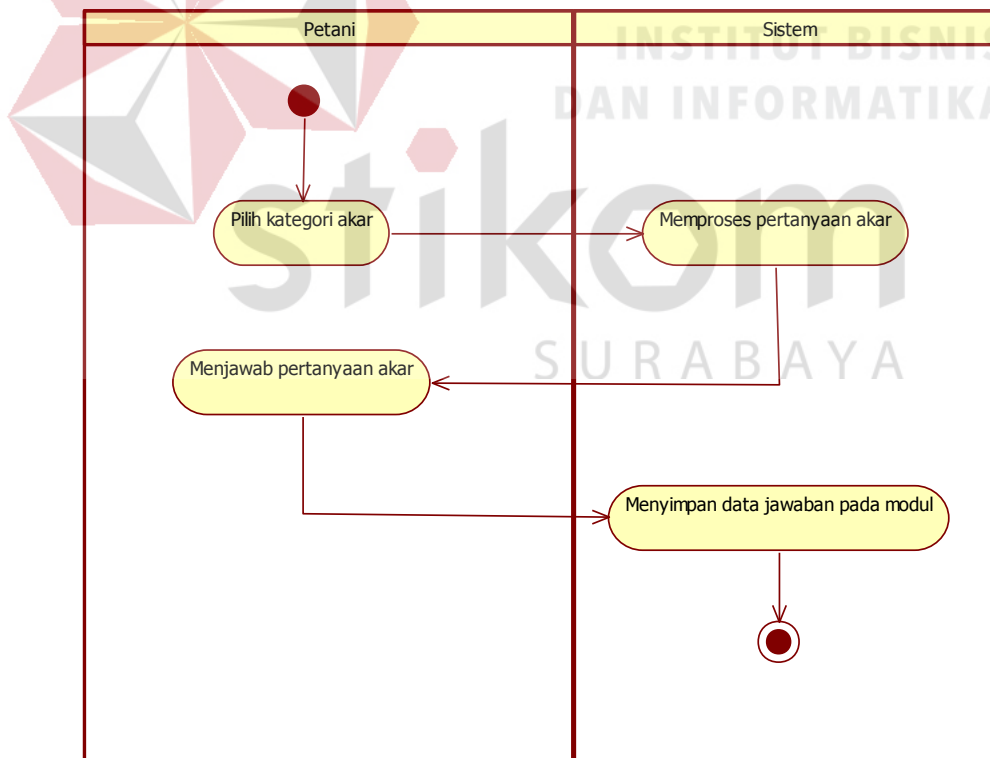
3.2.4 Activity Diagram

Dari *use case* yang ada, dibutuhkan *activity diagram* untuk menjelaskan proses/aliran yang terjadi pada tiap *use case*. *Activity diagram* adalah salah satu bentuk diagram UML yang paling mudah dimengerti dikarenakan diagram ini

memiliki simbol yang menyerupai simbol *flowchart*, yang sangat berguna untuk menerangkan langkah-langkah proses ke pihak lain.

1. Activity Diagram Akar

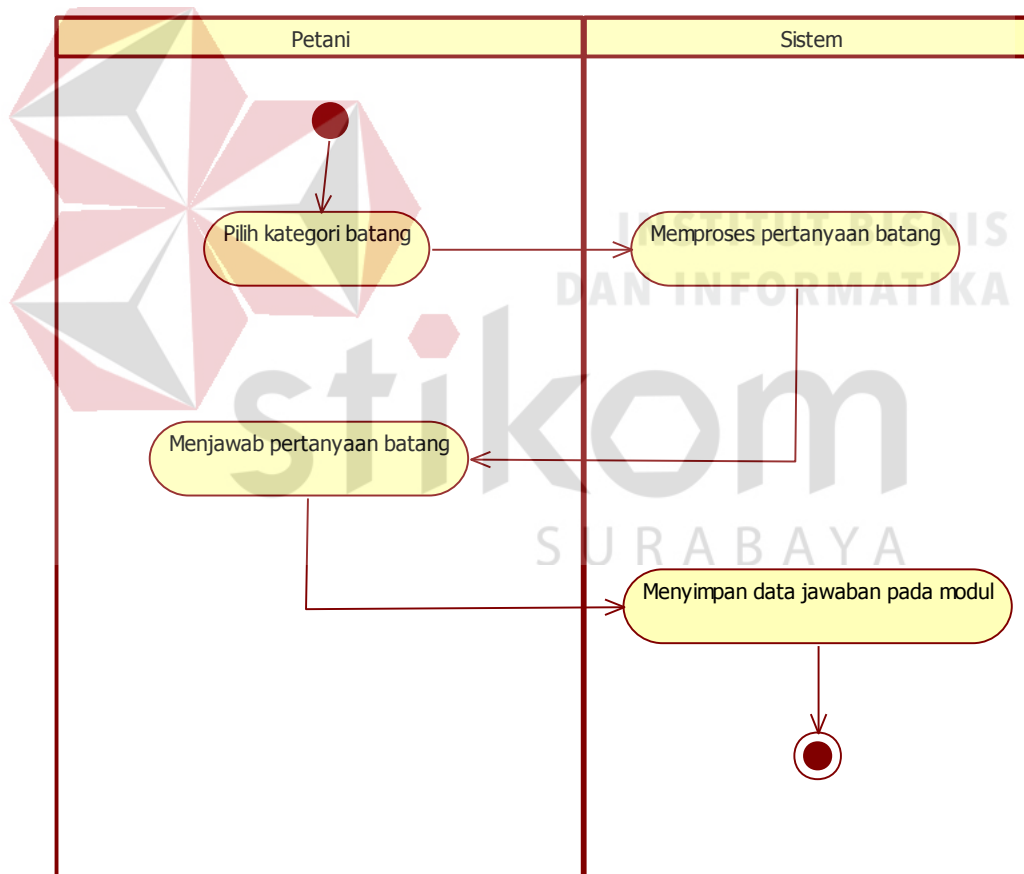
Proses ini dimulai dari *Mobile Android* yang telah menerima permintaan *user* (perani) mengenai pilih kategori dari aplikasi sistem yang ada pada *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan informasi tentang pilihan kategori akar sistem akan memproses pertanyaan gejala pada akar lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem dan jawaban dari *user* disimpan oleh sistem pada modul. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity diagram akar

2. Activity Diagram Batang

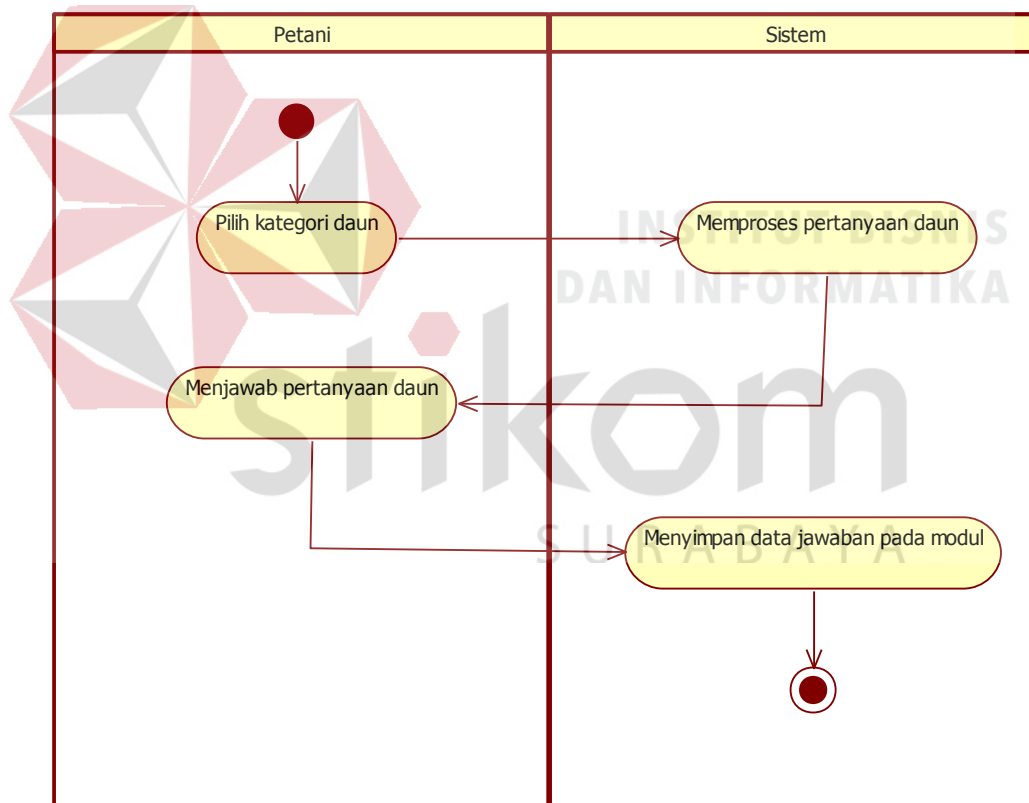
Proses ini dimulai dari *Mobile Android* yang telah menerima permintaan *user* (perani) mengenai pilih kategori dari aplikasi sistem yang ada pada *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan informasi tentang pilihan kategori batang sistem akan memproses pertanyaan gejala pada batang lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem dan jawaban dari *user* disimpan oleh sistem pada modul. Untuk lebih jelasnya dapat dilihat pada Gambar 3.6.



Gambar 3.6 Activity diagram batang

3. Activity Diagram Daun

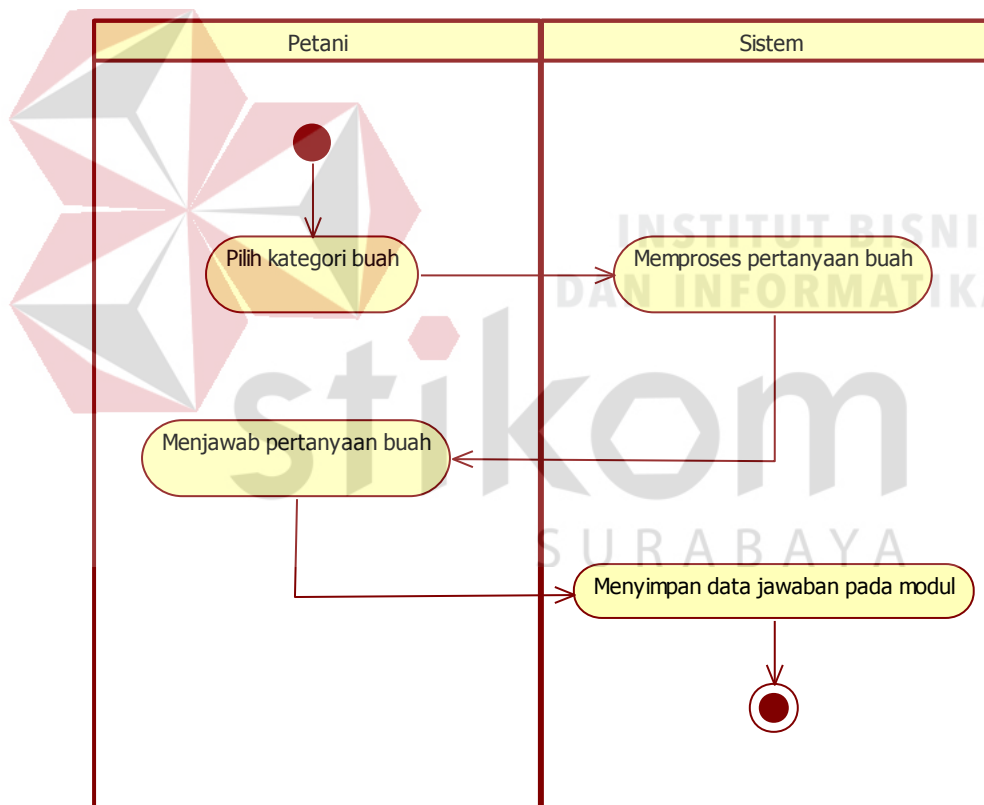
Proses ini dimulai dari *Mobile Android* yang telah menerima permintaan *user* (perani) mengenai pilih kategori dari aplikasi sistem yang ada pada *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan informasi tentang pilihan kategori daun sistem akan memproses pertanyaan gejala pada daun lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem dan jawaban dari *user* disimpan oleh sistem pada modul. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity diagram daun

4. Activity Diagram Buah

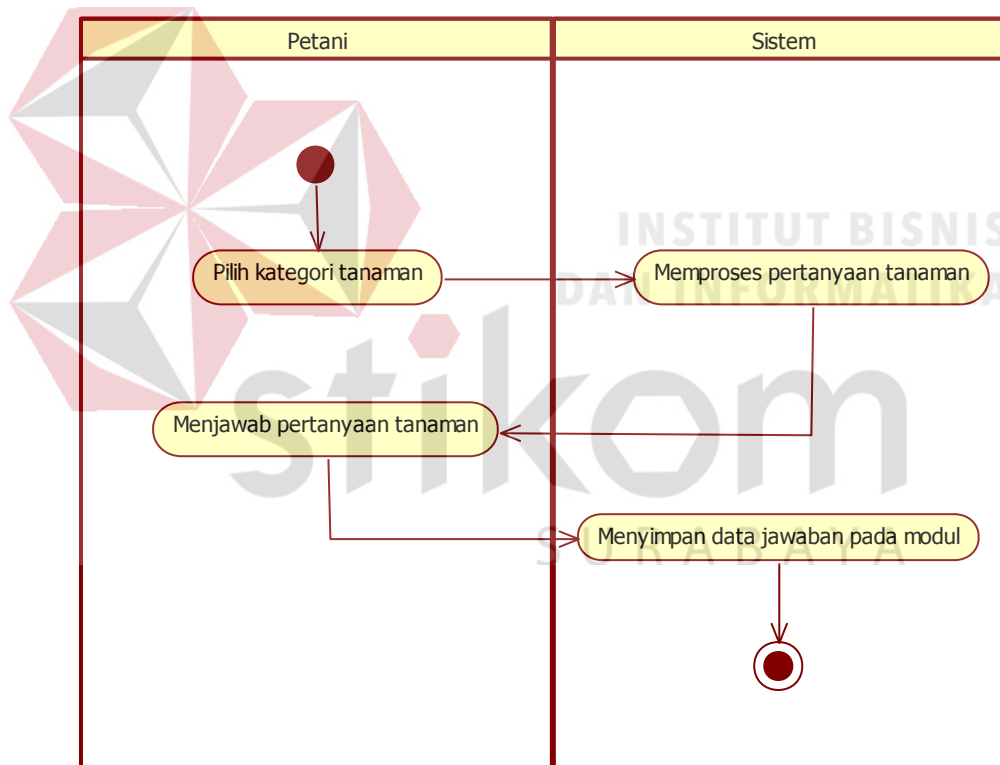
Proses ini dimulai dari *Mobile Android* yang telah menerima permintaan *user* (perani) mengenai pilih kategori dari aplikasi sistem yang ada pada *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan informasi tentang pilihan kategori buah sistem akan memproses pertanyaan gejala pada buah lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem dan jawaban dari *user* disimpan oleh sistem pada modul. Untuk lebih jelasnya dapat dilihat pada Gambar 3.8.



Gambar 3.8 Activity diagram buah

5. Activity Diagram Tanaman

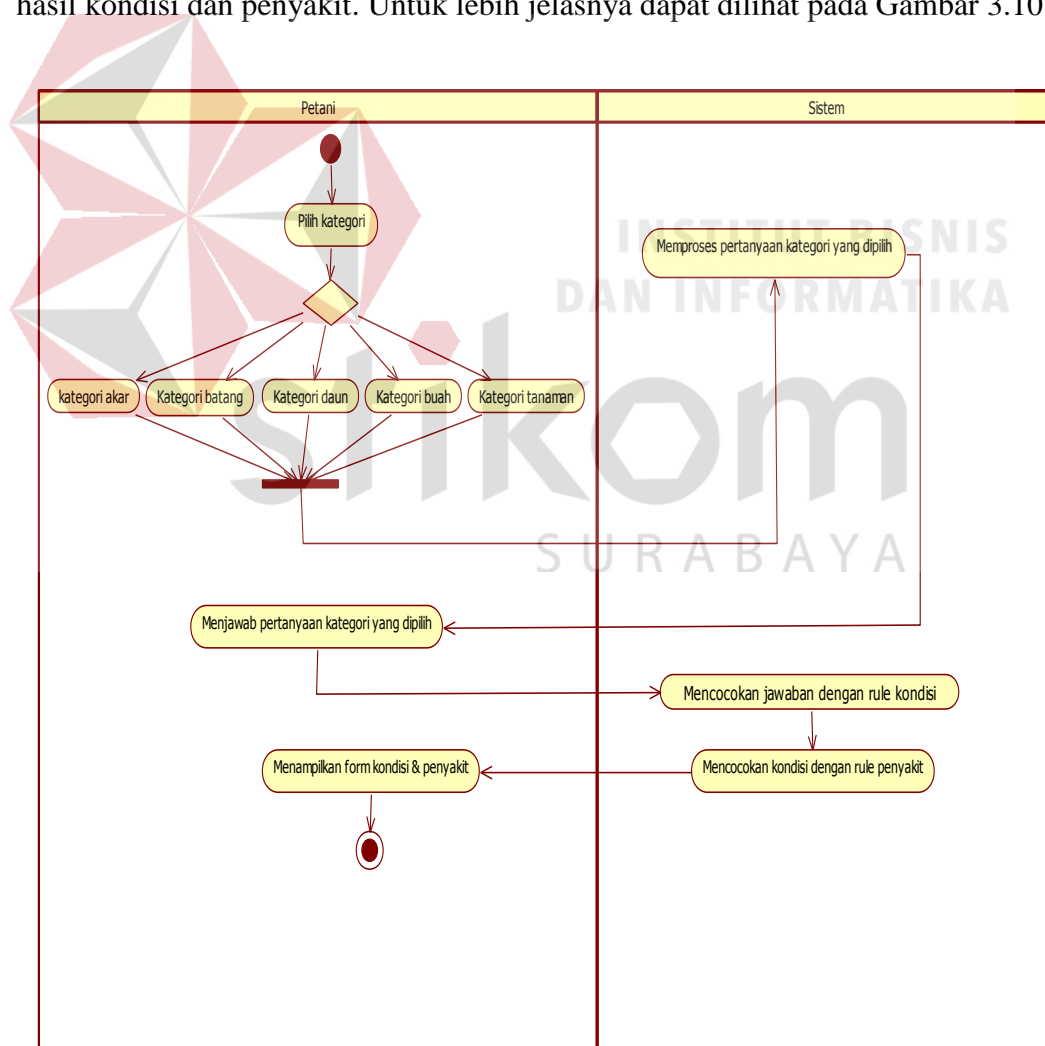
Proses ini dimulai dari *Mobile Android* yang telah menerima permintaan *user* (perani) mengenai pilih kategori dari aplikasi sistem yang ada pada *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan informasi tentang pilihan kategori tanaman sistem akan memproses pertanyaan gejala pada tanaman lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem dan jawaban dari *user* disimpan oleh sistem pada modul. Untuk lebih jelasnya dapat dilihat pada Gambar 3.9.



Gambar 3.9 Activity diagram tanaman

6. Activity Diagram Kondisi & Penyakit

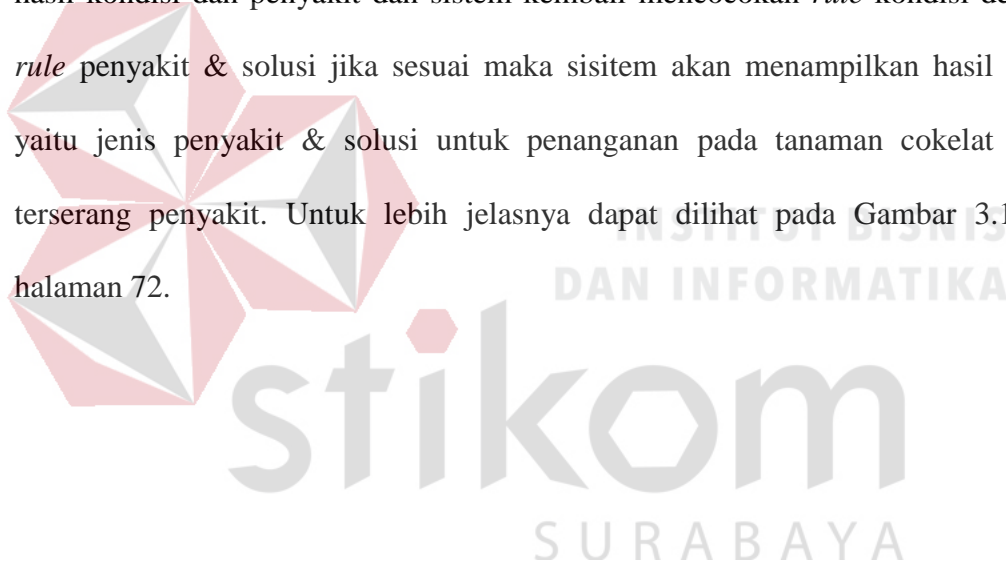
Proses ini dimulai dari *Mobile Application* yang telah menerima permintaan dari *User* (petani) mengenai kategori yang dipilih yaitu akar, batang, daun, buah dan tanaman dari aplikasi sistem pakar yang ada pada *smartphone android*. Selanjutnya *smartphone android* menerima *request* tersebut dan memproses pertanyaan kategori yang dijawab oleh *user* berdasarkan kategori yang telah dipilih dan sistem mencocokkan jawaban dengan *rule* kondisi dan selanjutnya *rule* kondisi dengan *rule* penyakit jika sesuai maka sistem akan menampilkan hasil kondisi dan penyakit. Untuk lebih jelasnya dapat dilihat pada Gambar 3.10.

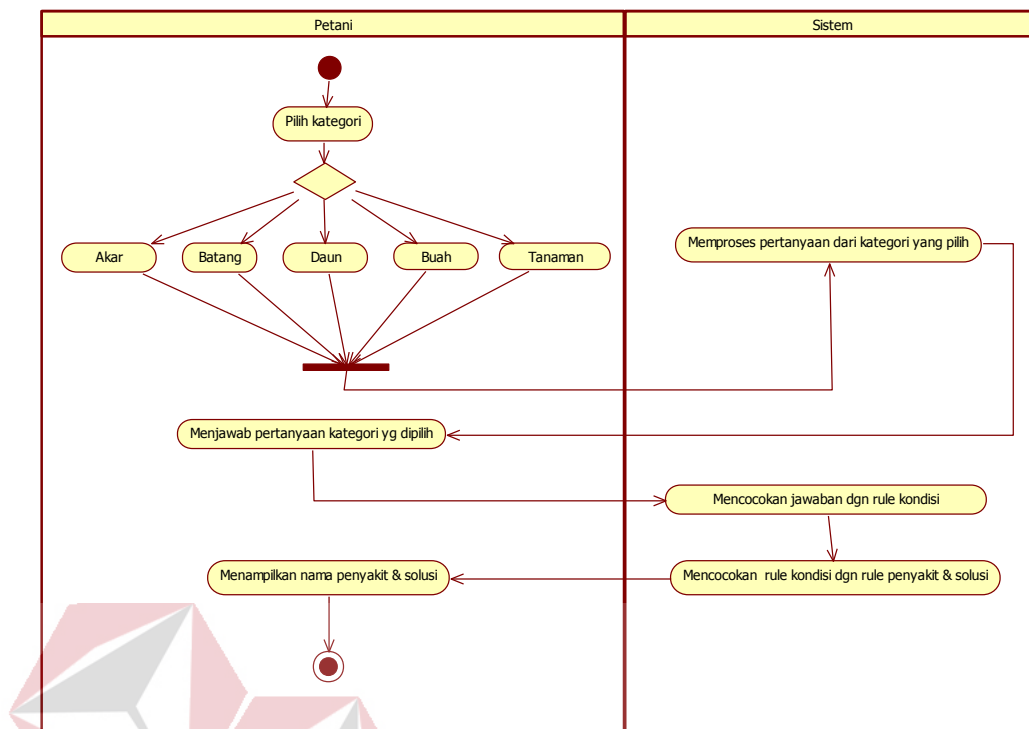


Gambar 3.10 Activity diagram kondisi & penyakit

7. Activity Diagram Penyakit dan Solusi

Proses ini dimulai dari *Mobile Application* yang telah menerima permintaan dari *User* (petani) mengenai kategori yang dipilih yaitu akar, batang, daun, buah dan tanaman dari aplikasi sistem pakar yang ada pada *smartphone android*. Selanjutnya *smartphone android* menerima *request* tersebut dan memproses pertanyaan kategori yang dijawab oleh *user* berdasarkan kategori yang telah dipilih dan sistem mencocokkan jawaban dengan *rule* kondisi dan selanjutnya *rule* kondisi dengan *rule* penyakit jika sesuai maka sistem akan menampilkan hasil kondisi dan penyakit dan sistem kembali mencocokkan *rule* kondisi dengan *rule* penyakit & solusi jika sesuai maka sistem akan menampilkan hasil akhir yaitu jenis penyakit & solusi untuk penanganan pada tanaman cokelat yang terserang penyakit. Untuk lebih jelasnya dapat dilihat pada Gambar 3.11 di halaman 72.





Gambar 3.11 Activity diagram penyakit dan solusi

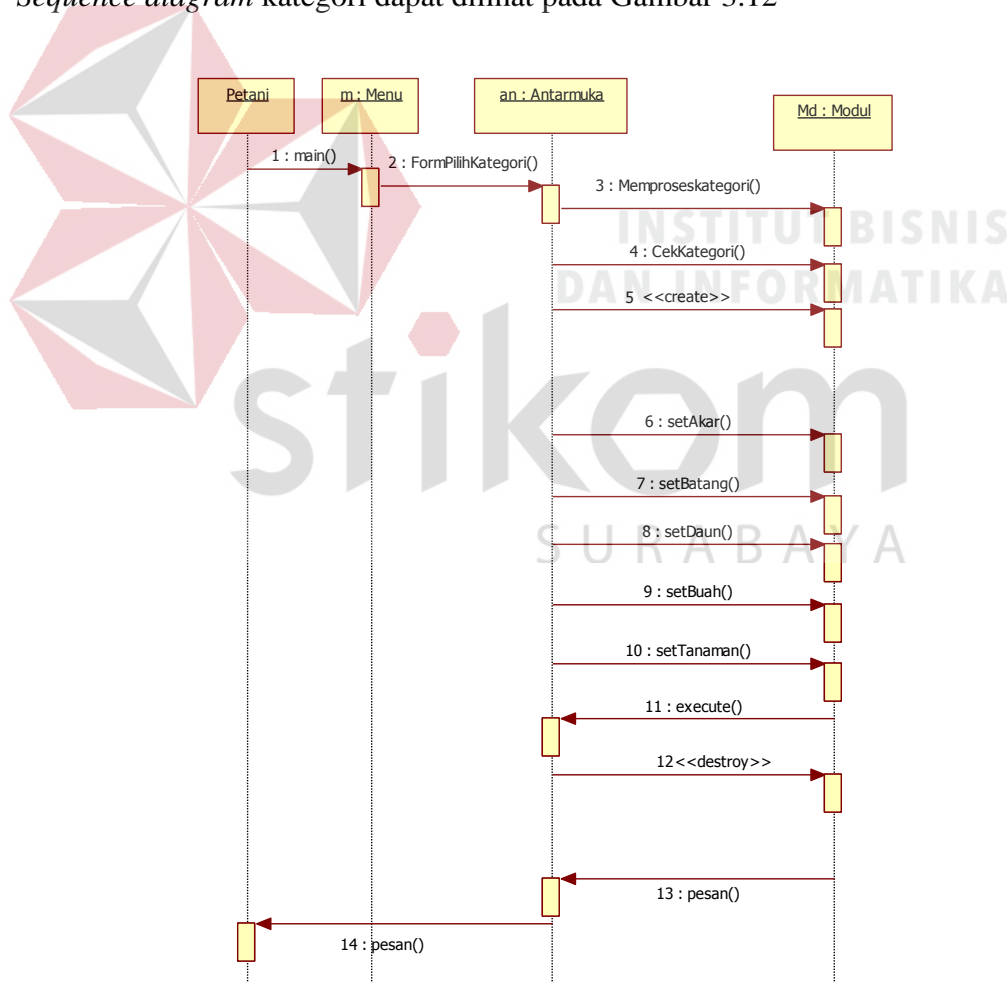
3.2.5 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan interaksi objek berdasarkan urutan waktu yang digambarkan dari atas ke bawah. Untuk melihat tahap ke tahap kejadian masing-masing *use case* pada *use case diagram*, maka dapat digunakan *Sequence diagram*. Diagram ini akan menjelaskan interaksi objek-objek yang disusun ke dalam satu urutan waktu. *Sequence diagram* memperlihatkan tahapan-tahapan yang terjadi untuk menghasilkan suatu di dalam *use case*.

1. Sequence diagram kategori

Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses kategori pada modul, sistem mengecek kategori yang telah di *create* oleh *user* seperti *setAkar*, *setBatang*, *setDaun*, *setBuah*, *setTanaman* pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem.

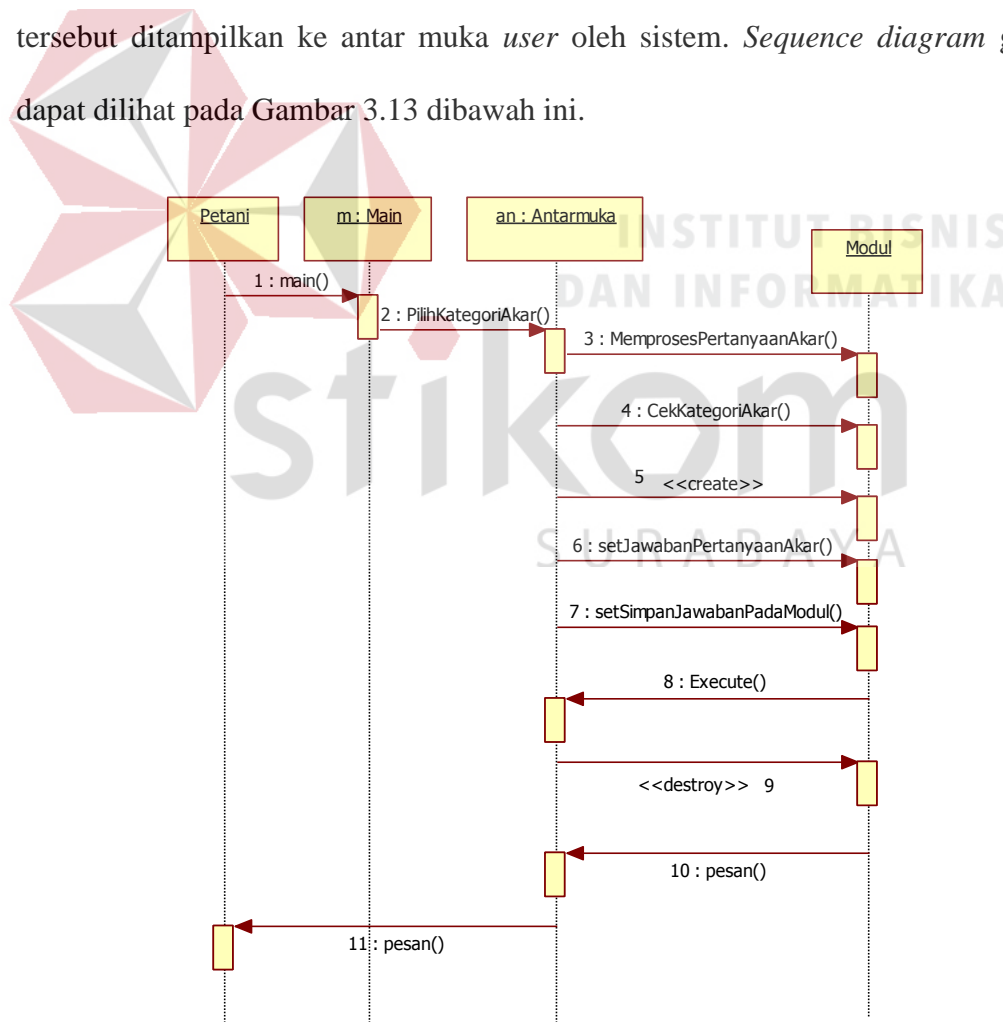
Sequence diagram kategori dapat dilihat pada Gambar 3.12



Gambar 3.12 *Sequence diagram* kategori

2. Sequence diagram akar

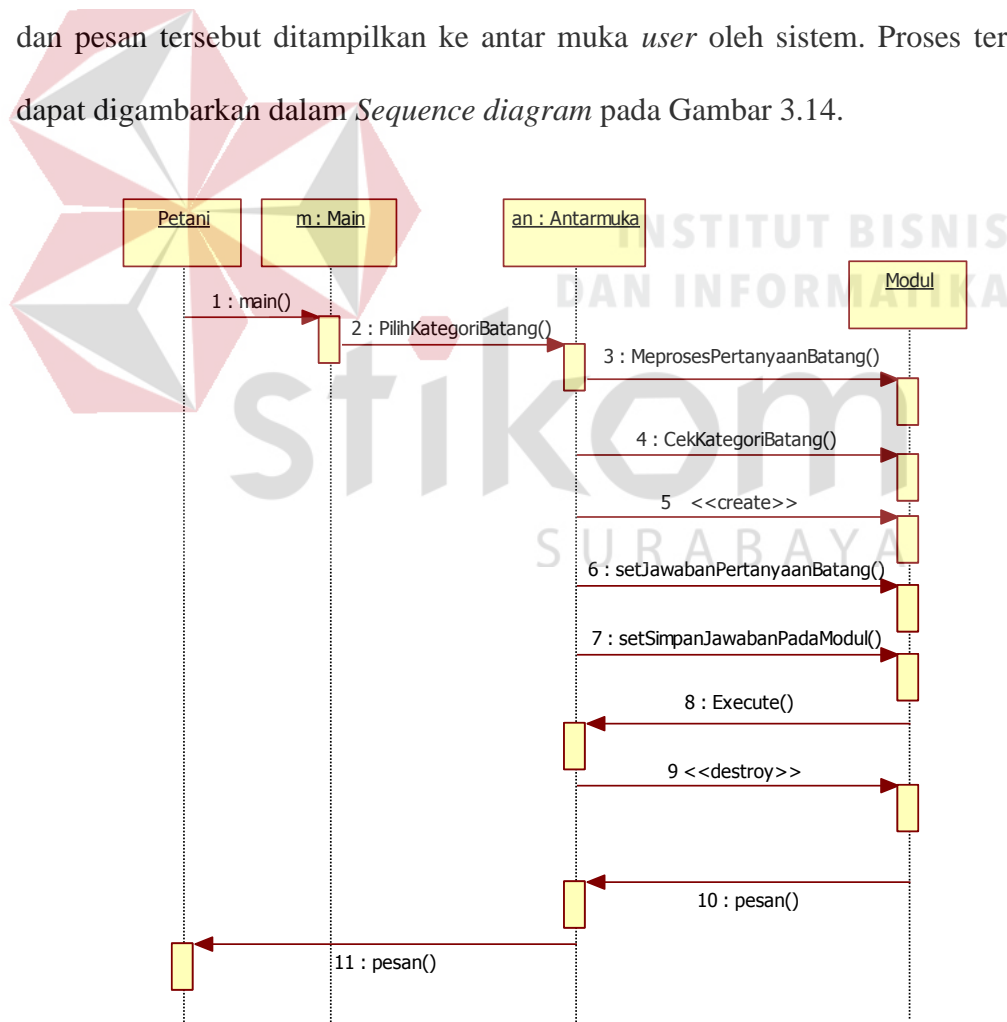
Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori akar di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses pertanyaan akar pada modul, sistem mengecek kategori akar yang telah di *create* oleh *user* seperti *set* jawaban dari pertanyaan yang dipilih oleh *user* dan jawaban tersebut disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori akar yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem. *Sequence diagram* gejala dapat dilihat pada Gambar 3.13 dibawah ini.



Gambar 3.13 Sequence diagram akar

3. Sequence diagram batang

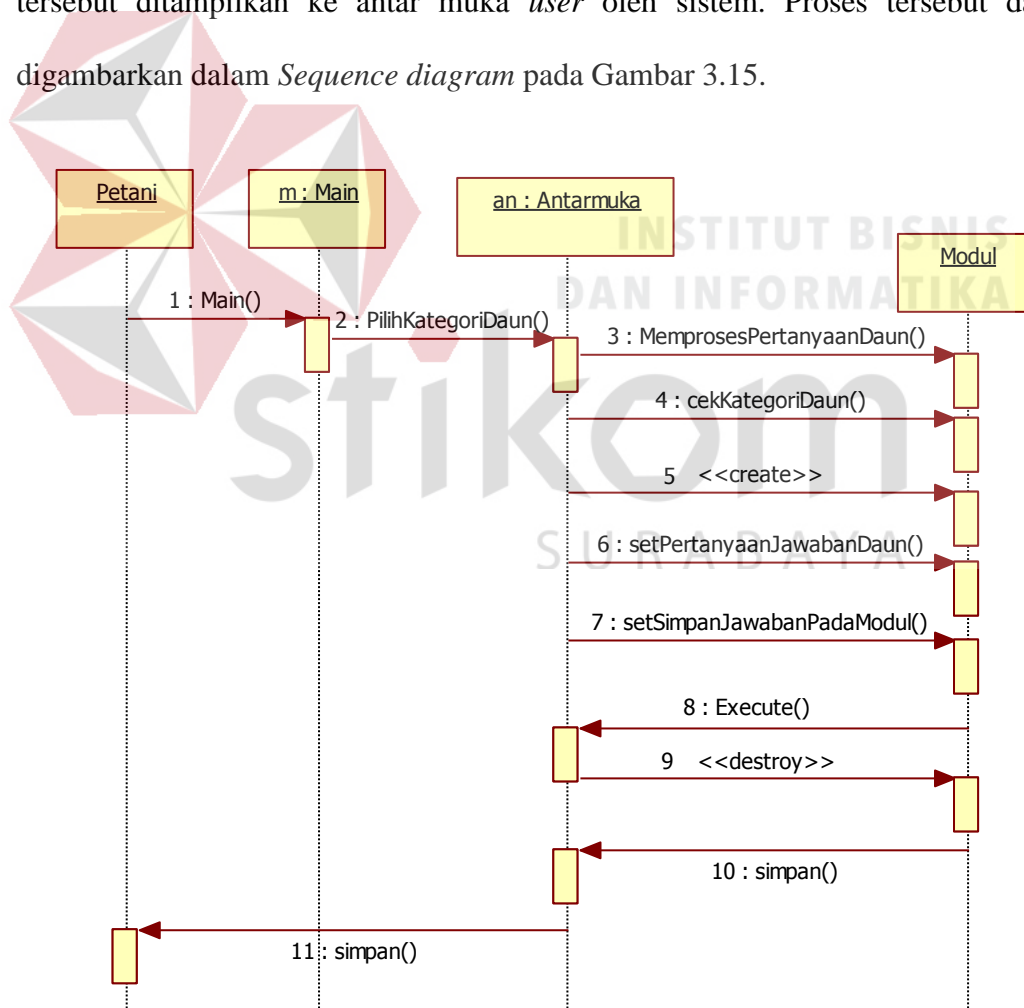
Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori batang di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses pertanyaan batang pada modul, sistem mengecek kategori batang yang telah di *create* oleh *user* seperti *set* jawaban dari pertanyaan yang dipilih oleh *user* dan jawaban tersebut disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori batang yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.14.



Gambar 3.14 Sequence diagram batang

4. Sequence diagram daun

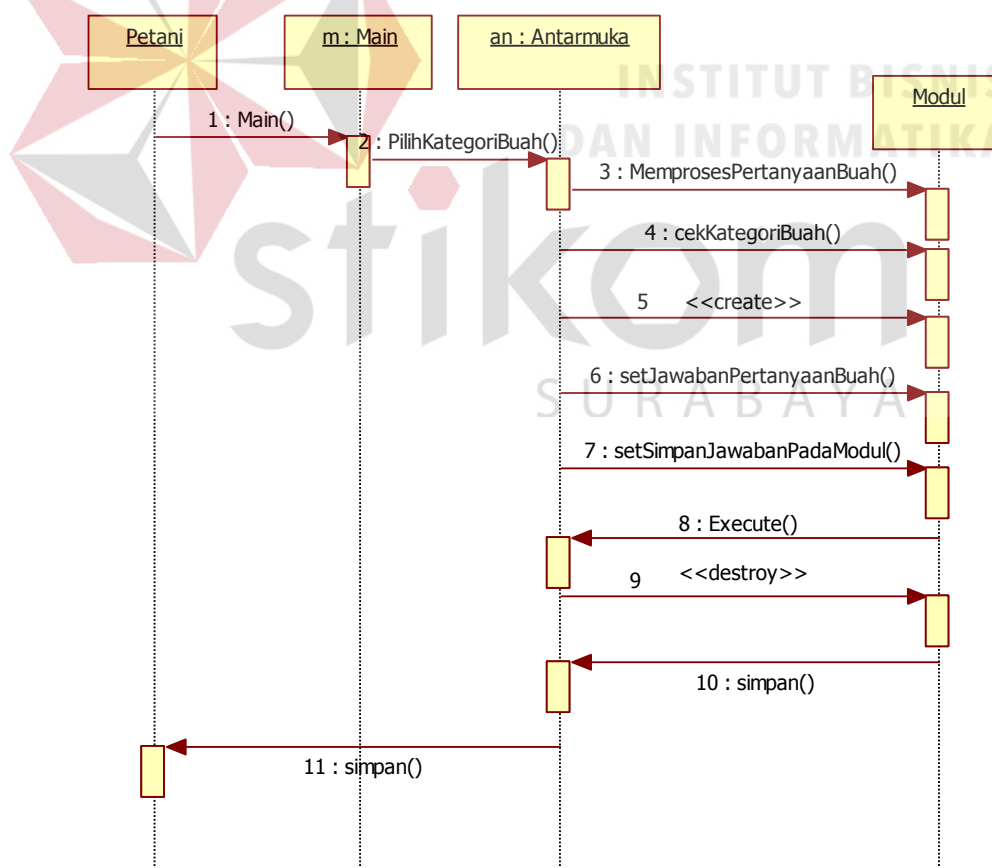
Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori daun di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses pertanyaan daun pada modul, sistem mengecek kategori daun yang telah di *create* oleh *user* seperti *set* jawaban dari pertanyaan yang dipilih oleh *user* dan jawaban tersebut disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori daun yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.15.



Gambar 3.15 Sequence diagram daun

5. Sequence diagram buah

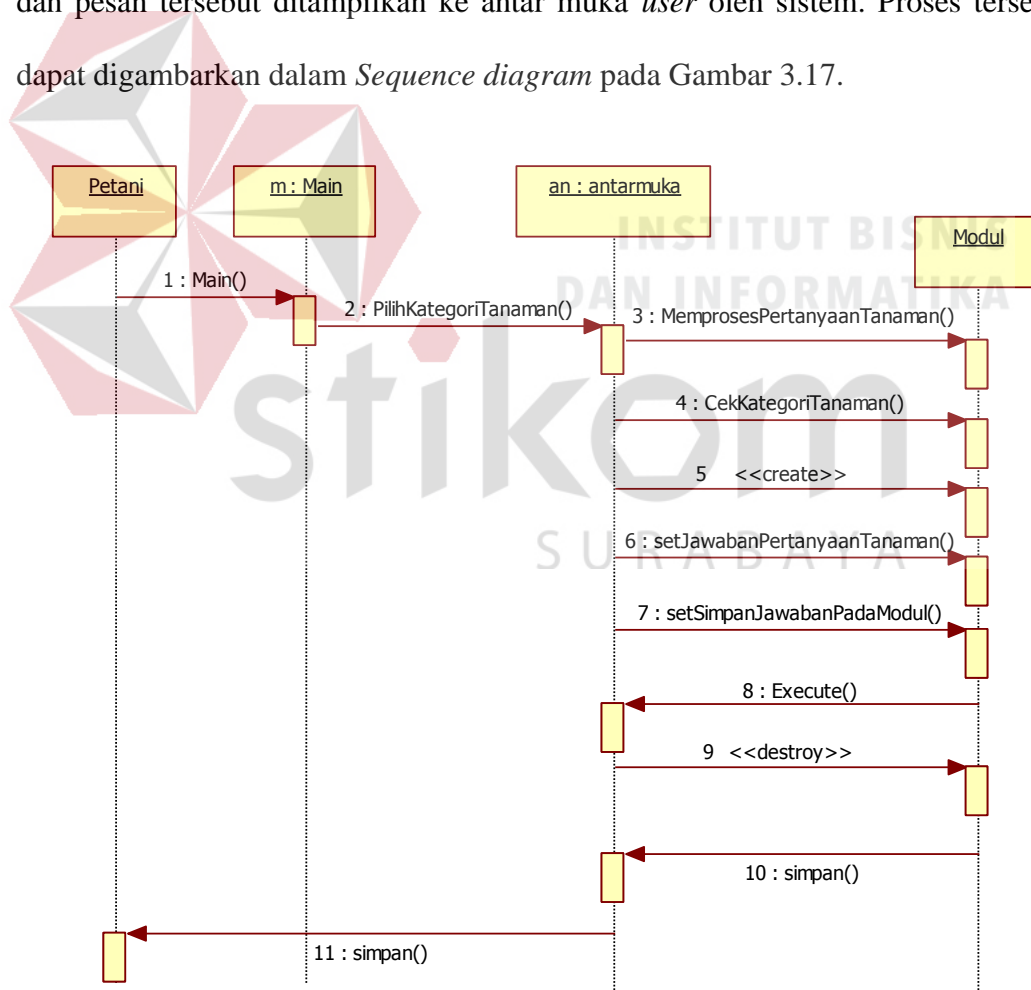
Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori buah di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses pertanyaan buah pada modul, sistem mengecek kategori buah yang telah di *create* oleh *user* seperti *set* jawaban dari pertanyaan yang dipilih oleh *user* dan jawaban tersebut disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori buah yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.16.



Gambar 3.16 Sequence diagram buah

6. Sequence diagram tanaman

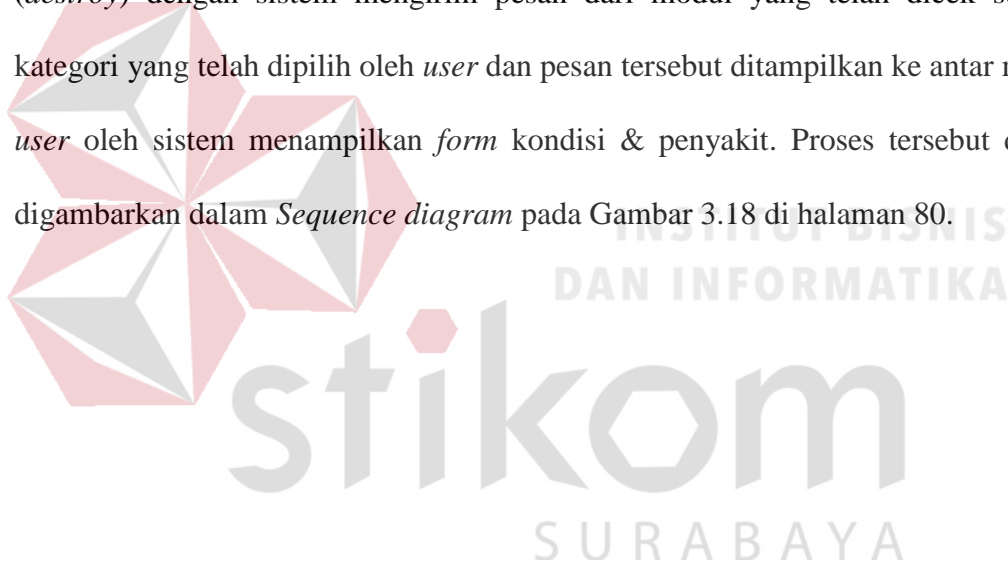
Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori tanaman di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memproses pertanyaan tanaman pada modul, sistem mengecek kategori tanaman yang telah di *create* oleh *user* seperti *set* jawaban dari pertanyaan yang dipilih oleh *user* dan jawaban tersebut disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori tanaman yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.17.

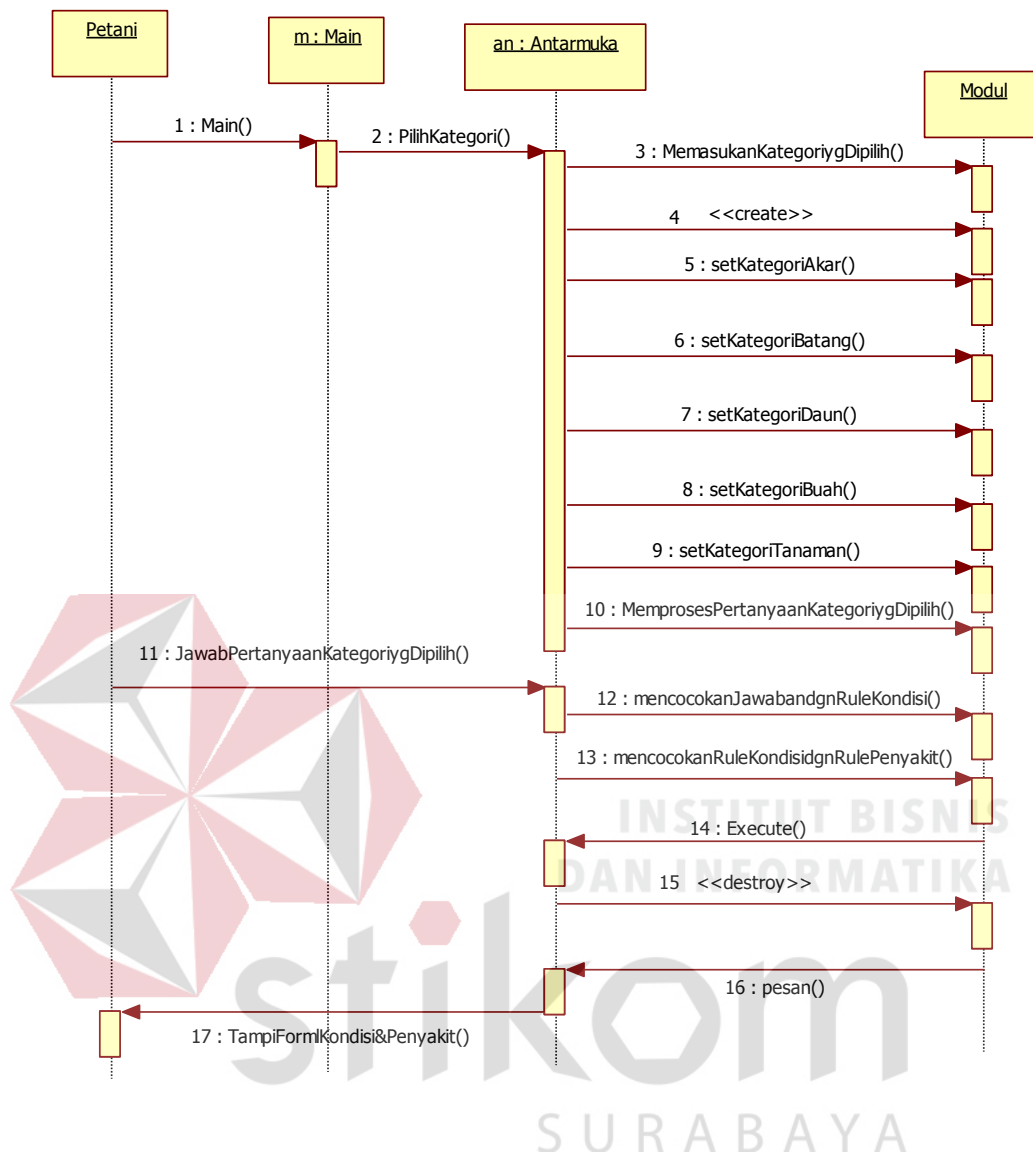


Gambar 3.17 Sequence diagram tanaman

7. *Sequence diagram* kondisi

Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memasukan kategori yang pilih pada modul, sistem mengecek kategori tanaman yang telah di *create* oleh *user* seperti *set* kategori akar, batang, daun, buah dan tanaman dan sistem memproses pertanyaan kategori dari jawaban yang dipilih oleh *user* dan mencocokkan jawaban dengan *rule* kondisi dengan *rule* penyakit dan disimpan pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*) dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh sistem menampilkan *form* kondisi & penyakit. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.18 di halaman 80.





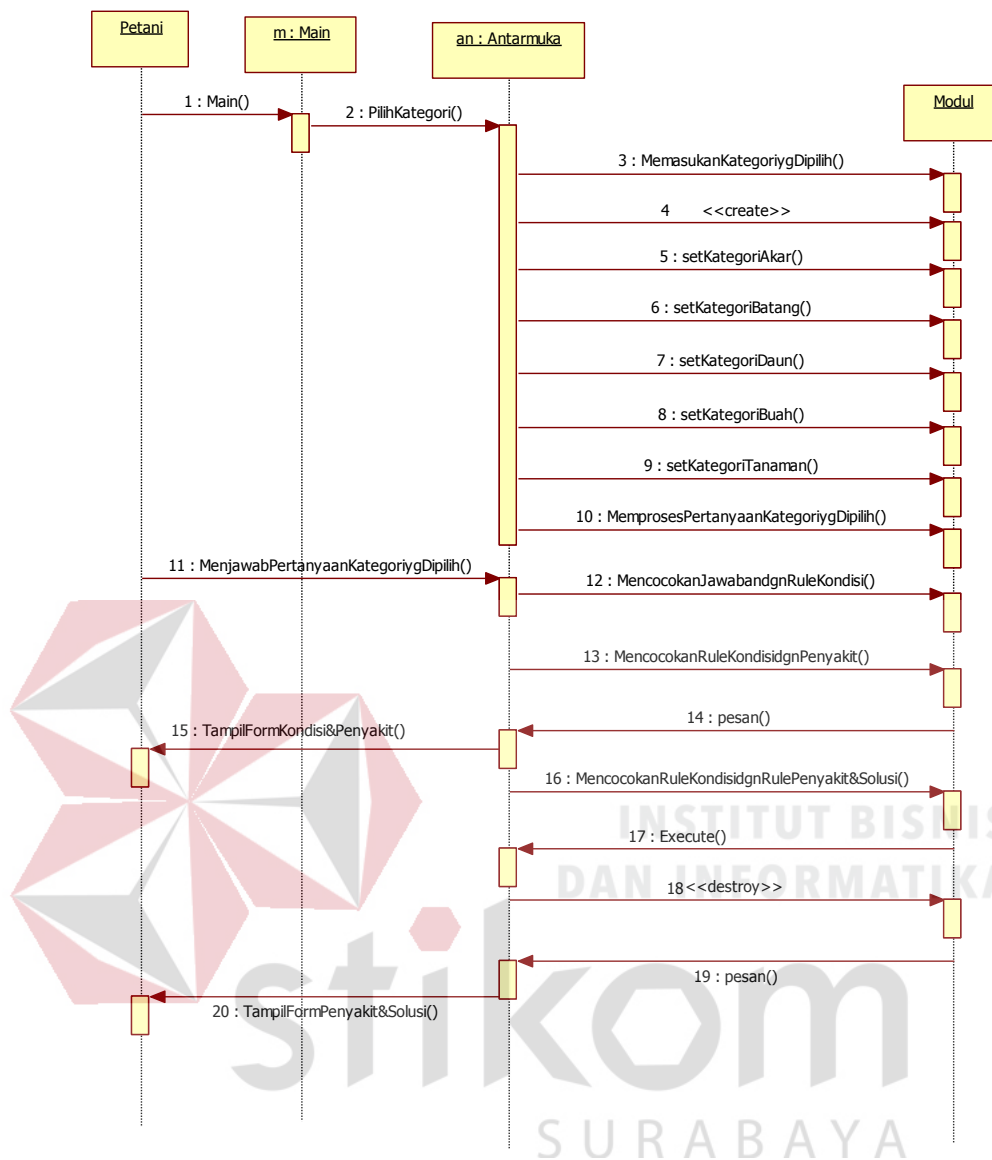
Gambar 3.18 *Sequence diagram* kondisi

8. *Sequence diagram* penyakit & solusi

Proses dimulai ketika *user* (petani) menjalankan sistem dan memilih kategori di menu utama yang ditampilkan ke antarmuka oleh sistem dan sistem memasukkan kategori yang pilih pada modul, sistem mengecek kategori tanaman yang telah di *create* oleh *user* seperti *set* kategori akar, batang, daun, buah dan tanaman dan sistem memproses pertanyaan kategori dari jawaban yang dipilih

oleh *user* dan mencocokkan jawaban dengan *rule* kondisi dengan *rule* penyakit dan dengan sistem mengirim pesan dari modul yang telah dicek sesuai kategori yang telah dipilih oleh *user* dan pesan tersebut ditampilkan ke antar muka *user* oleh system dan menampilkan *form* kondisi & penyakit selanjutnya sistem akan mencocokkan kembali *rule* kondisi dengan *rule* penyakit & solusi pada modul lalu di *execute* ke antarmuka *user* dari sistem dan diakhiri (*destroy*). Sistem akan mengirim pesan ke antarmuka *user* dan menampilkan hasil akhir penyakit & solusi untuk penanganan penyakit pada tanaman coklat. Proses tersebut dapat digambarkan dalam *Sequence diagram* pada Gambar 3.19 di halaman 82.

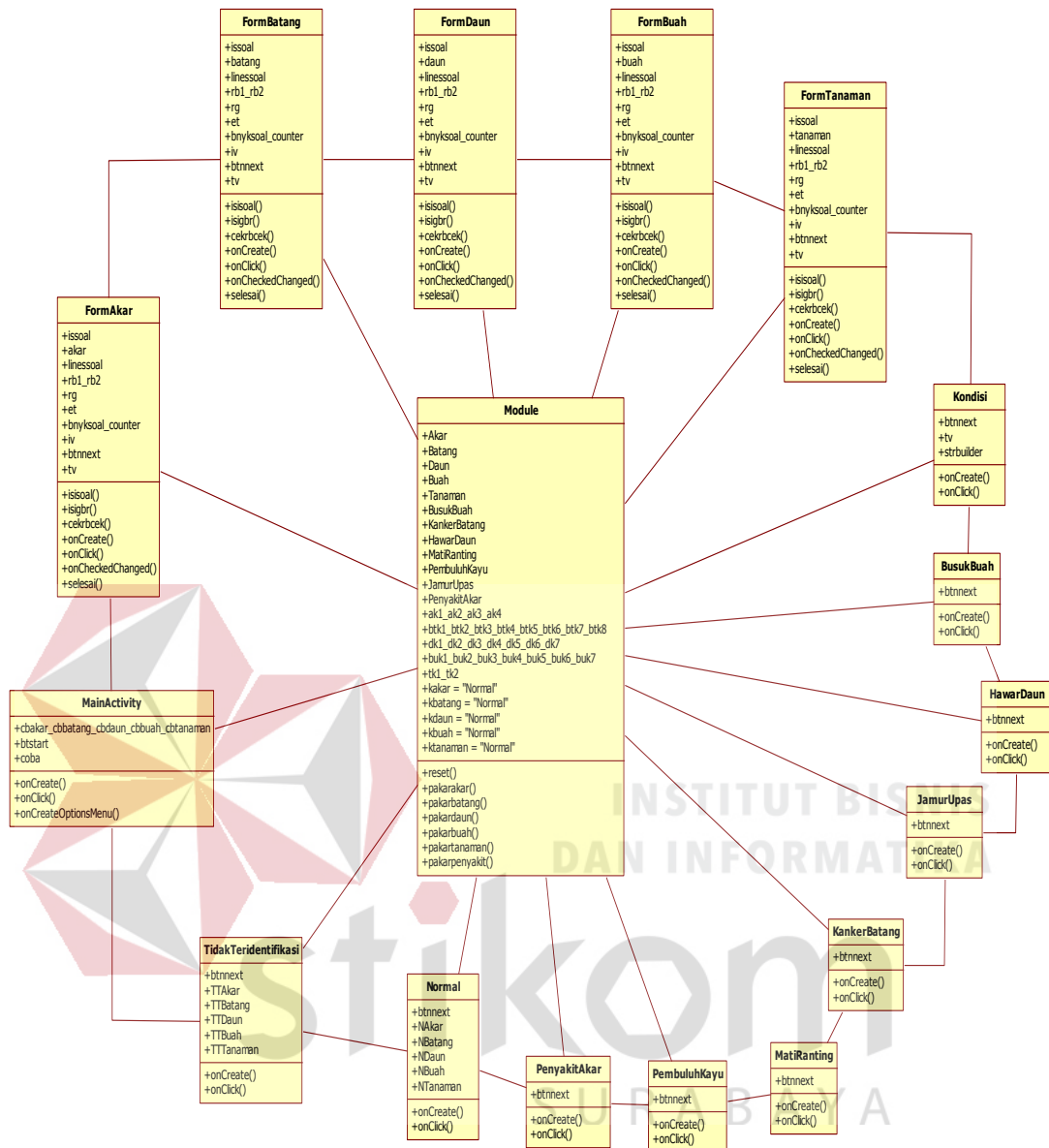




Gambar 3.19 *Sequence diagram* penyakit & solusi

3.2.6 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa, 2011). Diagram kelas memberikan gambaran sistem secara statis dan relasi antar kelas yang ada pada sistem. Diagram kelas sistem pakar untuk menentukan penyakit pada tanaman coklat ini dapat dilihat pada gambar 3.20 di halaman 83.



Gambar 3.20 Diagram Kelas Sistem Pakar Untuk Menentukan Penyakit Pada Tanaman Cokelat

Tabel 3.3 Penjelasan Diagram Kelas Sistem Pakar

NAMA KELAS	KETERANGAN
Modul	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> modul mengelola atau mencocokkan susai kategori yang dipilih oleh petani dan hasil jawaban kondisi dicocokkan dengan

NAMA KELAS	KETERANGAN
	modul yang berisi <i>form</i> akar, batang, daun, buah, tanaman dan menghasilkan kondisi selanjutnya dicocokkan kembali kondisi-kondisi yang telah teridentifikasi dengan <i>rule</i> penyakit yang ada bila tidak ditemukan maka dianggap tidak teridentifikasi
MainActivity	Merupakan kelas menu
FormAkar	Kelas yang menangani tampilan dan proses yang diambil dari pendefinisian <i>use case</i> berisi gejala atau pertanyaan tentang akar yang akan di jawab ya atau tidak
FormBatang	Kelas yang menangani tampilan dan proses yang diambil dari pendefinisian <i>use case</i> berisi gejala atau pertanyaan tentang batang yang akan di jawab ya atau tidak
FormDaun	Kelas yang menangani tampilan dan proses yang diambil dari pendefinisian <i>use case</i> berisi gejala atau pertanyaan tentang daun yang akan di jawab ya atau tidak
FormBuah	Kelas yang menangani tampilan dan proses yang diambil dari pendefinisian <i>use case</i> berisi gejala atau pertanyaan tentang buah yang akan di jawab ya atau tidak
FormTanaman	Kelas yang menangani tampilan dan proses yang diambil dari pendefinisian <i>use case</i> berisi gejala atau pertanyaan tentang tanaman yang akan di jawab ya atau tidak
Kondisi	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses jawaban yang teridentifikasi dengan <i>rule</i> penyakit yang ada pada modul
BusukBuah	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit busuk buah yang teridentifikasi
HawarDaun	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit hawar daun yang teridentifikasi
JamurUpas	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit jamur upas yang teridentifikasi
KankerBatang	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit kanker batang yang teridentifikasi
MatiRanting	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit mati ranting yang teridentifikasi

NAMA KELAS	KETERANGAN
PembuluhKayu	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit pembuluh kayu yang teridentifikasi
PenyakitAkar	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang memproses penyakit akar yang teridentifikasi
Normal	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang telah dicocokkan dengan <i>rule</i> penyakit yang ada bila tidak teridentifikasi dan tidak ditemukan kondisi atau penyakit maka dianggap normal
TidakTeridentifikasi	Merupakan kelas proses yang diambil dari pendefinisian <i>use case</i> yang telah dicocokkan dengan <i>rule</i> penyakit yang ada bila tidak ditemukan kondisi atau penyakit maka dianggap tidak teridentifikasi

3.2.7 Desain Input Output

Desain *input output* yang dibuat berfungsi untuk memudahkan *user* pakar dan *user* umum dalam menggunakan sistem. *Forms* yang dirancang meliputi konsep interaksi manusia dan sistem, dimana seorang *user* hanya dengan melihat *form* dapat mengerti dengan mudah langkah yang dilakukan selanjutnya.

1. Desain Form Konsultasi

Desain ini merupakan desain pertama kali yang ditampilkan ke *user*. Pada desain ini berisikan data kategori diantaranya yaitu: akar, batang, daun, buah dan tanaman yang didukung oleh sistem, diperuntukkan oleh *user* saat memilih kategori yang sesuai dengan masalah pada tanaman coklat *user* dan tombol *Next* untuk masuk ke *form* selanjutnya yang berisi pertanyaan dari masing-masing kategori yang telah dipilih dan akan dijawab oleh *user* dengan mengklik *radio button* pada jawaban ya atau tidak. Pada Gambar 3.21

menunjukkan seluruh kategori yang didukung oleh sistem. Menampilkan kategori dan fungsi setiap obyeknya dapat dilihat pada tabel 3.4

DATA

... Pilih Kategori ...

- AKAR
- BATANG
- DAUN
- BUAH
- TANAMAN

Gambar 3.21 Desain *Form* Konsultasi

Tabel 3.4 Fungsi Obyek Desain *Form* Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Akar, Batang, Daun, Buah,	<i>CheckBox</i>	Berisi pilihan kategori yang dibutuhkan oleh <i>user</i>
2	Next	<i>Button</i>	Tampil ke <i>form</i> selanjutnya

2. Desain Sub Kategori

Pada desain ini, sistem akan menampilkan seluruh data sub kategori yang berhasil dipilih oleh *user*, yang berisi pertanyaan dari masing-masing kategori yang telah dipilih dan akan dijawab oleh *user* dengan mengklik *radio button* pada jawaban ya atau tidak. Tombol *Next* untuk masuk ke *form* selanjutnya yang berisi pertanyaan berikutnya dan Tombol Hasil untuk masuk ke *form* hasil kondisi dari proses sistem yang berisi kesimpulan penyakit dan solusi, tidak teridentifikasi dan normal. Gambar di bawah ini, *user* berhasil memilih kategori tanaman pada *form* pilih kategori. Untuk lebih jelasnya ada pada

desain *input output* Gambar 3.22. menampilkan gejala serta fungsi setiap obyeknya dapa dilihat pada tabel 3.5.

Gambar 3.22 Desain *Form* Pilih Gejala

Tabel 3.5 Fungsi Obyek Desain *Form* Gejala

No	Nama Obyek	Tipe Obyek	Fungsi
1	Nama gejala	<i>Rich textbox</i>	Berisi nama gejala
2	Gambar	<i>PictureBox</i>	Tampilan gambar gejala
3	Ya	<i>Radio Button</i>	Berisi pilihan jawaban Ya
4	Tidak	<i>Radio Button</i>	Berisi pilihan jawaban Tidak
5	Back	<i>Button</i>	Kembali ke halaman sebelumnya
6	Next	<i>Button</i>	Tampil ke <i>form</i> selanjutnya
7	Hasil	<i>Button</i>	Tampil ke <i>form</i> kesimpulan

3. Desain Tampil Jenis Penyakit dan Solusi, Tidak Teridentifikasi dan Normal

Pada desain ini, sistem menampilkan jenis penyakit dan solusi dari kategori yang sudah dipilih oleh *user* dan menjawab setiap pertanyaan gejala pada masing-masing kategori yang berhasil diproses oleh sistem. Desain ini akan menampilkan seluruh informasi yang dibutuhkan oleh *user* seperti jenis penyakit dan solusi, Tidak teridentifikasi atau normal. Sebagai contoh, pada

Gambar 3.23 menampilkan jenis penyakit dan solusi serta fungsi setiap obyeknya dapat dilihat pada tabel 3.6

The image shows three side-by-side screenshots of a web form. Each screenshot represents a different disease category. The first is 'Nama Penyakit', the second is 'Tidak Teridentifikasi', and the third is 'Normal'. Each form has a title bar, a 'Gambar' (Image) field, a 'Solusi' (Solution) field, and 'Back' and 'New' buttons at the bottom.

Gambar 3.23 Desain *Form* Menampilkan Kesimpulan

Tabel 3.6 Fungsi Obyek Desain *Form* Kosultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Nama penyakit	<i>Rich textbox</i>	Berisi nama penyakit
2	Gambar	<i>PictureBox</i>	Tampilan gambar penyakit
3	Solusi	<i>Rich textbox</i>	Berisi informasi solusi penyakit pada tanaman coklat
4	Tidak Teridentifikasi	<i>Rich textbox</i>	Berisi informasi tidak teridentifikasi
5	Normal	<i>Rich textbox</i>	Berisi informasi normal
6	Back	<i>Button</i>	Kembali ke halaman sebelumnya
7	New	<i>Button</i>	Konsultasi baru

3.3 Perancangan Evaluasi Sistem

3.3.1 Perancangan Uji Coba Sistem

Perancangan uji coba sistem ini berisi tentang rancangan uji coba pada *form* yang telah dibuat. Uji coba ini dilakukan dengan menggunakan *Block Box Testing*. Aplikasi ini akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang dibuat sesuai dengan tujuan.

1. Tes Case

Pada tabel dibawah ini merupakan tabel *test case* yang telah direncanakan pada saat uji coba desain. Tabel *test case* ini terdiri dari tujuan, *input*, *output* yang diharapkan dan status.

Tabel 3.7 uji coba *test case*

No	Tujuan	Input	Output Diharapkan
1	Uji coba sistem pakar penyakit busuk buah	<u>Buah</u> : Ujung atau pangkal buah tampak bercak coklat kehitaman (Y), permukaan buah muncul serbuk berwarna putih (Y), Pertanyaan lain dijawab (T)	mengeluarkan jawaban penyakit busuk buah dan penanganannya.
2	Uji coba sistem pakar penyakit kanker batang	<u>Batang</u> : Kulit batang tampak warna gelap/kehitaman dan agak berlekuk(Y), Kulit batang terlihat bercak hitam, ada cairan kemreahan seperti lapisan karat (Y), Batang bila dikupas terlihat lapisan dibawah berwarna merah anggur (Y), Pertanyaan lain	mengeluarkan jawaban penyakit Kanker Batang dan penanganannya.

No	Tujuan	Input	Output Diharapkan
		dijawab (T)	
3	Uji coba sistem pakar penyakit hawar daun	<p><u>Daun</u>: Daun terlihat mengerut seperti tersiram air panas (Y),</p> <p><u>Batang</u>: Kulit batang tampak warna gelap/kehitaman dan agak berlekuk (Y), Kulit batang terlihat bercak hitam, ada cairan kemreahan seperti lapisan karat (Y), Batang bila dikupas terlihat lapisan dibawah berwarna merah anggur (Y),</p> <p>Pertanyaan lain dijawab (T)</p>	mengeluarkan jawaban penyakit hawar daun dan penanganannya.
4	Uji coba sistem pakar penyakit mati ranting (<i>Anthraco</i>)	<p><u>Daun</u>: Daun muda terlihat bintik-bintik berwarna coklat (Y), daun muda yang berkembang terlihat bercak berlubang berwarna kuning (Y);</p> <p><u>Buah</u>: Buah muda tampak bintik-bintik coklat (Y);</p> <p><u>Tanaman</u>: Tanaman yang terserang cukup berat terlihat sedikit/tanpa daun (Y).</p> <p>Pertanyaan lain dijawab (T)</p>	mengeluarkan jawaban penyakit mati ranting (<i>Anthraco</i>) dan penanganannya
5	Uji coba sistem pakar penyakit pembuluh kayu	<p><u>Daun</u>: Daun menguning dengan bercak-bercak berwarna hijau (Y);</p> <p><u>Batang</u>: Ranting</p>	mengeluarkan jawaban penyakit pembuluh kayu dan penanganannya

No	Tujuan	Input	Output Diharapkan
		bekas potongan daun, bekas duduk daun/bekas potongan ranting terlihat benang-benang berwarna putih (Y); Pertanyaan lain dijawab (T)	
7	Uji coba sistem pakar penyakit jamur upas	<u>Daun</u> : Daun terlihat mengerut seperti tersiram air panas (Y); <u>Batang</u> : Batang terdapat jamur mengilat seperti perak mirip seperti sarang laba-laba (Y). Pertanyaan lain dijawab (T)	mengeluarkan jawaban penyakit jamur upas dan penanganannya
8	Uji coba sistem pakar penyakit Akar	<u>Akar</u> : Perpumakaan akar dilapisi jamur berwarna merah/cokelat tua (Y). Pertanyaan lain dijawab (T)	mengeluarkan jawaban penyakit Akar dan penanganannya
9	Uji coba sistem pakar tidak teridentifikasi	<u>Akar</u> : Perpumakaan akar dilapisi jamur berwarna merah/cokelat tua (y). Pertanyaan lain dijawab (y) (Y); <u>Batang</u> : Batang terdapat jamur mengilat seperti perak mirip seperti sarang laba-laba (Y). Pertanyaan lain dijawab (T)	mengeluarkan jawaban Tidak teridentifikasi

No	Tujuan	Input	Output Diharapkan
10	Uji coba sistem pakar Normal	<p><u>Buah</u> :</p> <p>Ujung atau pangkal buah tampak bercak coklat kehitaman (T), permukaan buah muncul serbuk berwarna putih (T), Pertanyaan lain dijawab (T),</p> <p><u>Batang</u>:</p> <p>Kulit batang tampak warna gelap/kehitaman dan agak berlekuk(T), Kulit batang terlihat bercak hitam, ada cairan kemreahan seperti lapisan karat (T), Batang bila dikupas terlihat lapisan dibawah berwarna merah anggur (T)</p>	mengeluarkan jawaban Normal

