BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

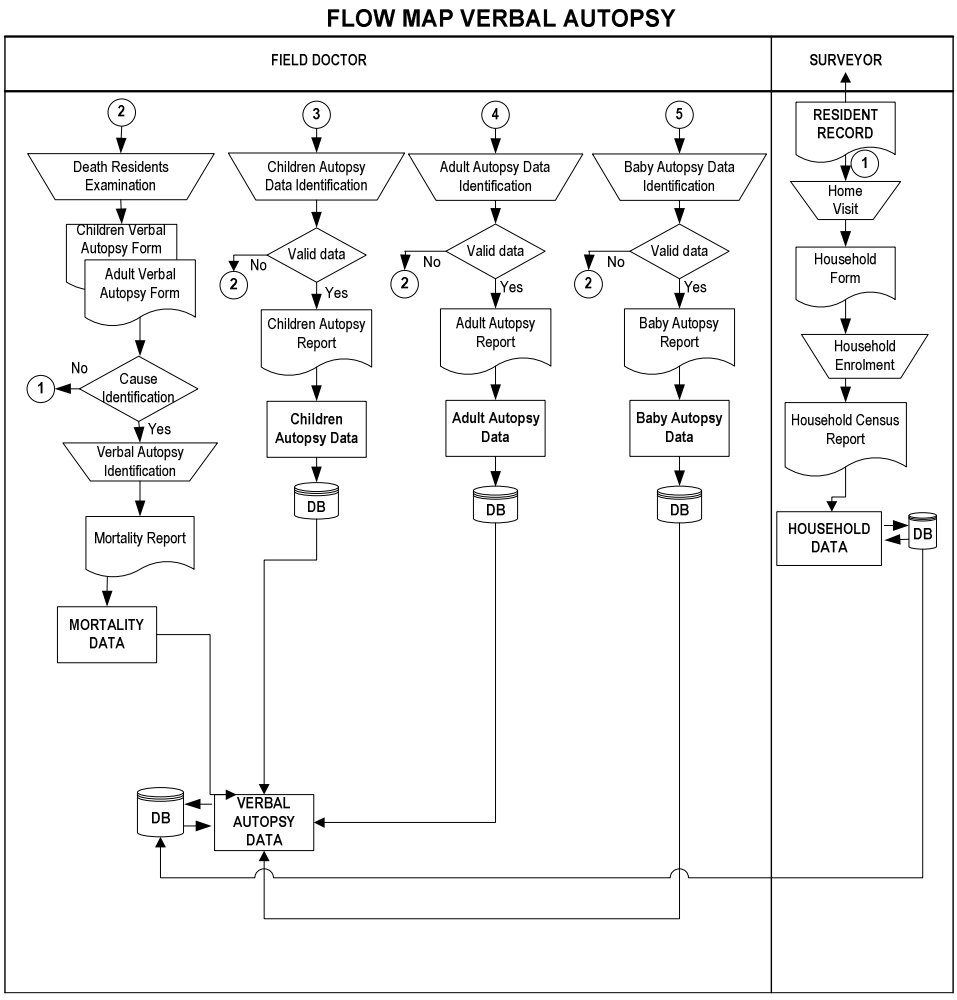
Analisis sistem diperlukan untuk menganalisa sistem penelitian CIRAI yang sudah berjalan, menganalisa permasalahan yang ada pada sistem penelitian dan menganalisa kebutuhan yang diperlukan oleh penelitian saat ini.

3.1.1 Analisis Sistem Penelitian

Penelitian CIRAI yang merupakan kependekan dari *Community-based and Integrated Research of Influenza* adalah salah satu kelompok kerja yang dilakukan oleh Unit Penelitian Kesehatan (UPK) FK UNPAD - Rumah Sakit Hasan Sadikin tentang penyakit influenza. Seperti penelitian kesehatan pada umumnya CIRAI mempunyai kegiatan utama mengumpulkan data objek penelitian, dimana data tersebut dikelola menjadi beberapa modul aplikasi yang terintregasi satu sama lain dalam sebuah sistem berbasis web.

Dengan berkembangnya data penelitian, saat ini CIRAI melakukan pengambilan data kematian didaerah penelitian dengan metode otopsi verbal, tujuannya untuk mengidentifikasi penyebab kematian oleh *symptoms* influenza. Dari kegiatan ini akan dihasilkan sebuah data otopsi verbal yang harus dikelola oleh sistem untuk menghasilkan sebuah informasi berupa pendukung diagnosa tentang identifikasi penyebab kematian oleh *symptoms* influenza, dengan acuan data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10.

Dimana pada modul aplikasi yang sudah ada, untuk membuat sebuah laporan setiap modulnya, masih dilakukan secara manual oleh dokter penelitian. Manual disini maksudnya, setelah data masuk ke database sistem, dokter akan mendownloadnya dalam bentuk file excel, dan akan mengolahnya secara manual untuk menghasilkan informasi yang dibutuhkan. Berikut ini adalah alur prosedur dari sistem otopsi verbal yang sedang berjalan:



Gambar 3.1 Alur Proses Pengolahan Data Otopsi verbal

3.1.2 Analisis Masalah

Dari analisis sistem yang sedang berjalan, dapat dilihat bahwa permasalahan dalam sistem penelitian CIRAI adalah dibutuhkan sebuah modul aplikasi yang dapat mengelola data otopsi verbal dan menghasilkan informasi berupa hasil identifikasi penyebab kematian oleh *symptoms* influenza untuk setiap kasus otopsi verbal yang ada dengan acuan data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10.

3.1.3 Solusi yang ditawarkan

Dari permasalahan yang telah dipaparkan dalam analisis masalah, maka dapat disimpulkan bahwa saat ini pihak penelitian CIRAI membutuhkan sebuah modul aplikasi otopsi verbal yang dapat mengelola data kematian dan mengolahnya menjadi sebuah informasi dalam bentuk prediksi identifikasi penyebab kematian dengan menggunakan acuan data korespondensi daftar penyebab kematian untuk otopsi verbal dan hubungannya dengan kode ICD-10.

Untuk aplikasi otopsi verbalnya sendiri, berdasarkan bentuk form manual otopsi verbal standar WHO yang komplek dengan pengkodean serta kondisi disetiap pertanyaannya, penulis memberikan solusi sebuah aplikasi yang *user friendly* dengan validasi disetiap pertanyaannya untuk meminimalisasi kesalahan input data. Sedangkan untuk penyajian informasi dalam bentuk identifikasi penyebab kematian, salah satu solusi yang dapat diterapkan adalah dengan *data mining*, karena *data mining* dapat ‘menggali’ *knowledge* yang tersimpan dalam data, dalam hal ini *knowledge* mengenai *symptoms* influenza pada data otopsi

verbal. Teknik *mining* yang akan digunakan adalah teknik *Classification* berupa *decision tree*. Dan jenis *tree* yang akan diterapkan adalah *forward chaining tree,* dimana algoritma *forward chaining* ini dapat menjawab kebutuhan penelitian, karena menggunakan data *symptoms* influenza sebagai fakta dan data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10 sebagai kaidah untuk memperoleh suatu kesimpulan.

3.1.4 Analisis Data

Dari analisa sistem yang akan dibangun, dapat diusulkan struktur data yang akan digunakan oleh sistem. Dalam hal ini data dibagi menjadi dua kategori, yaitu data pengolahan otopsi verbal, dan data siap *mining* dari data pengolahan otopsi verbal tersebut. Usulan struktur data pengolahan otopsi verbal tersebut terdiri dari tabel t_vo, t_vo_kematian, t_vo_observasi, t_vo11,t_vo12, t_vo12 , t_vo22, t_vo31, t_vo32 dan t_vo33. Pembagian tabel tersebut didasarkan pada pola form dan design form aplikasi yang sesuai dengan form manual otopsi verbal. Berikut tabel deskripsi dari pembagian tabel – tabel tersebut :

Tabel 3.1 Deskripsi masing – masing tabel otopsi verbal

No	Tabel	Deskripsi
1	t_vo	Tabel utama, berisi informasi umum dari form otopsi verbal
2	t_vo_kematian	Tabel yang berisi catatan riwayat kematian termasuk surat kematian
3	t_vo_observasi	Tabel yang berisi informasi hasil observasi dari form otopsi verbal

4	t_vo11	Tabel yang berisi informasi Penyakit atau kejadian yang mengarah pada kematian, untuk kategori umur 1 (usia kematian dibawah 4 minggu)
5	t_vo12	Tabel yang berisi informasi gejala sebelum kematian, untuk kategori umur 1 (usia kematian dibawah 4 minggu)
6	t_vo21	Tabel yang berisi informasi Penyakit atau kejadian yang mengarah pada kematian, untuk kategori umur 2 (usia kematian 4 minggu - 14 tahun)
7	t_vo22	Tabel yang berisi informasi gejala sebelum kematian, untuk kategori umur 2 (usia kematian 4 minggu - 14 tahun)
8	t_vo31	Tabel yang berisi informasi Penyakit atau kejadian yang mengarah pada kematian dan cidera/kecelakaan, untuk kategori umur 3 (usia kematian 15 tahun keatas)
9	t_vo32	Tabel yang berisi informasi gejala penyakit pada wanita dan gejala yang berkaitan dengan kehamilan sebelum kematian, untuk kategori umur 3 (usia kematian 15 tahun keatas)
10	t_vo33	Tabel yang berisi informasi perawatan dan pelayanan kesehatan saat masa akhir hidup almarhum, untuk kategori umur 3(usia kematian 15 tahun keatas)

Sedangkan untuk aplikasi data miningnya sendiri akan membutuhkan tabel berdasarkan skenario mining yang akan dibahas pada subbab berikut.

3.1.5 Analisis Data Mining

Metode yang digunakan dalam proses *mining* dalam melakukan pencarian informasi mengenai identifikasi penyebab kematian disini menggunakan metode *Decision Tree.* Dimana jenis *decision tree* yang digunakan adalah *forward chaining tree*. Dalam penulisan tugas akhir ini akan dilakukan proses pencarian informasi dari sekumpulan data atau fakta berupa *influenza symptoms*, untuk menemukan suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi yaitu jenis *verbal autopsy themes* sebagai hasil identifikasi penyebab kematian. Proses akan mencari kaidah-kaidah dalam basis pengetahuan yang premisnya sesuai dengan data-data tersebut, kemudian dari kaidah-kaidah tersebut diperoleh suatu kesimpulan. Sesuai landasan teori *data mining* yang dijelaskan pada bab sebelumnya, berikut tahapan proses *mining* identifikasi penyebab kematian pada penelitian CIRAI:

1. **Pembersihan Data (*Data Cleaning*)**

   Proses *data mining* dalam aplikasi ini terdiri dari dua sumber data, yaitu data otopsi verbal hasil dari modul pengolahan otopsi verbal dalam aplikasi ini sendiri dan data korespondensi hasil penyebab kematian dengan ICD – 10. Kedua data tersebut mempunyai atribut-atribut data yang tidak relevan yang perlu dibersihkan, berikut keterangan proses pembersihan data tersebut :

   1.1 Pembersihan Data Korespondensi Hasil Penyebab Kematian dengan ICD – 10

Dari batasan masalah yng ada, bahwa aplikasi ini dibatasi pada *symptoms* influenza saja, maka proses pembersihan data dalam proses mining disini adalah menghilangkan *noise* untuk field *symptoms* influenza. Data yang digunakan untuk proses *mining* disini terdiri dari dua sumber data, yaitu data otopsi verbal dan data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10. Penghilangan *noise* pada data ICD-10, dilakukan dengan melakukan pencocokan dengan *symptoms* influenza yang akan digunakan sesuai keterangan pakar. Data ICD-10 dalam bentuk PDF, contohnya untuk kasus tema otopsi verbal Campak, *symptoms* fotopobia dan tidak mendapat vaksin campak, tidak terdapat pada data otopsi verbal, maka dihilangkang, seperti terlihat pada gambar berikut:

Tabel 3.2 Contoh proses pembersihan data ICD-10

Kode otopsi verbal	Tema otopsi verbal	Kode ICD	Tema ICD	Keterangan
VA-01	Penyakit infeksi dan parasit			Perbedaan yang penting pada ICD-10 Pada ICD-10 , meningitis ditemukan pada pada bagian 1 (penyakit infeksi dan parasit) dan bagian VI (penyakit sistem saraf) tergantung penyebab infeksinya. Infeksi saluran napas akut masuk ke bagian X (penyakit sistem pernapasan) Beberapa penyakit infeksi timbul akibat penyakit infeksi yang lain. Untuk kombinasi yang sesuai terdapat kategori yang sesuai pada ICD-10.
VA-01.01	Penyakit infeksi usus (termasuk diare)	A029	Infeksi salmonela, tidak terperinci	**Informasi yang mendukung** Peningkatan kekerapan feses < 3 minggu. Darah di feses. Mukus di feses Demam Nyeri perut atau tenesmus Penurunan berat badan Muntah
		A039	Shigelosis, tidak terperinci	
		A049	Infeksi bakteri intestinum, tidak terperinci	
		A059	Intoksikasi bakteri foodborne, tidak terperinci	
		A069	Amoebiasis, tidak terperinci.	
		A079	Penyakit infeksi protozoa, tidak terperinci	
		A084	Infeksi virus intestinum, tidak terperinci	
		A09	Diare dan gastroenteritis yang disebabkan infeksi.	
Kode otopsi verbal	**Tema otopsi verbal**	**Kode ICD**	**Tema ICD**	**Keterangan**
			terperinci	
VA-01.07	Campak	B059	Campak	Campak, NOS **Informasi yang mendukung** Ruam kulit (makuloapapular, diawali di wajah kemudisan turun dan keluar) diperlukan untuk diagonsis. Demam, tinggi Batuk Merah mata bangkak Tidak mendapat vaksin campak Fotofobia.
VA-01.08	Hepatitis virus	B159	Hepatitis A	Tanda dan gejalan hepatitis virus tidak spesifik; tanda klinis sangat bervariasi; dan,
		B169	Hepatitis B akut	
		B179	Hepatitis akut lainya	
		B189	Hepatitis kronis viral, tidak terperinci	
		B199	Hepatitis viral tidak terperinci	
VA-01.09	HIV/AIDS	B209	HIV yang menyebabkan infeksi yang tidak terperinci atau penyakit parasi	HIV / AIDS dapat menyebabkan tumor ganas tertentu (Misalnya, sarcoma Kaposi). Dalam kasus-kasus seperti itu harus kode di sini dan di ICD-10 dengan menggunakan kategori yang sesuai di bawah B21.
		B219	HIV yang menyebabkan neoplasma maligna tidak terperinci	
		B227	HIV yang menyebabkan penyakit multipel yang masuk banyak klasifikasi	
		B238	HIV yang menyebabkan kondisi lain yang terperinci	
		B24	HIV yang tidak terperindi	
VA-01.10	Malaria	B509	Malaria *P. Falciparum* tidak terperinci	**Informasi yang mendukung** Demam tinggi Demam naik turun Penurunan kesadaran
		B50.0	Malaria cerebral	
		B519	Malaria *P Vivax*	

1.2 Pembersihan Data Otopsi Verbal

Data symptoms otopsi verbal menggunakan pengkodean disetiap kontennya dimana pengkodean tersebut adalah sebagai berikut:

Tabel 3.3 Keterangan Pengkodean Data Symptoms Otopsi Verbal

No	Kode	Keterangan
1	1	Ya
2	2	Tidak
3	998	Tidak Tahu

Dari pengkodean tersebut, maka data *symptoms* yang akan dipakai hanya data dengan isian 1 yang artinya Ya. Contohnya data *symptoms* dapat dilihat pada isi tabel t_vo33 yang berisi *symptoms* untuk kematian kategori umur diatas 15 tahun berikut ini:

Tabel 3.4 Contoh Data Symptoms Sebelum Dibersihkan

id_vo	vo901	vo902	vo903	vo904	vo905	vo906	vo907	vo908	vo909	vo910
1	1	2	0	0	0	0	2	0	0	0
2	1	2	0	0	0	0	1	2	2	1
3	2	2	0	0	0	0	2	0	0	0
4	1	2	0	0	0	0	1	1	2	2
5	998	2	0	0	0	0	2	0	0	0
6	1	1	1	2	1	2	2	0	0	0
7	1	2	0	0	0	0	1	1	1	1
8	1	1	1	2	2	2	2	0	0	0
9	1	2	0	0	0	0	1	1	2	2

2. Seleksi Data (*Data Selection)*

Proses ini merupakan proses pemilihan data penelitian CIRAI yang akan di-*mining* dari sumber data yang sudah disebutkan sebelumnya. Dari analisa

penentuan *symptoms* influenza yang telah dilakukan, maka hanya beberapa *field* dari beberapa tabel yang akan digunakan, yaitu:

1. Tabel t_vo(id_vo, vo301)
2. Tabel t_vo12 (vo913, vo914,vo917,vo919,vo921,vo923)
3. Tabel t_vo22 (vo512,vo803, vo804, vo808, vo809, vo824)
4. Tabel t_vo33 (vo512,vo902, vo903,vo907,vo908,vo933)

Pemilihan atribut tersebut merupakan hasil analisis dari data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10 dengan tabel otopsi verbal terpilih. Dari data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10 dipelajari dan dipilih *symptoms* influenza dan kemudian di cocokkan dengan *attribute* dari tabel t_vo12, t_vo22 dan t_vo33. Setelah dianalisis, *symptoms* influenza tersebut tidak sama antara masing - masing tabel t_vo12, t_vo22 dan t_vo33, karena *symptoms* untuk masing – masing kategori umur akan berbeda. Dan berikut keterangan dari setiap atribut tersebut:

Tabel 3.5 Keterangan Atribut

Tabel	Atribut	Keterangan
t_vo	id_vo	Id otopsi verbal
t_vo	vo301	Nama almarhum
t_vo12	vo906	kejang
t_vo12	vo911	penurunan_kesadaran
t_vo12	vo913	demam
t_vo12	vo917	batuk
t_vo12	vo919	napas_cepat
t_vo12	vo921	dispnea
t_vo12	vo936	ruam

t_vo22	vo803	demam
t_vo22	vo806	demam_turun_naik
t_vo22	vo807	menggigil
t_vo22	vo808	batuk
t_vo22	vo809	batuk_3mgg
t_vo22	vo812	napas_cepat
t_vo22	vo814	dispnea
t_vo22	vo815	dispnea_3mgg
t_vo22	vo818	mengi
t_vo22	vo841	penurunan_kesadaran
t_vo22	vo844	kejang
t_vo22	vo851	urin_sedikit
t_vo22	vo852	ruam
t_vo22	vo856	mata_merah
t_vo22	vo858	penurunan_berat
t_vo22	vo863	bengkak
t_vo33	vo902	demam
t_vo33	vo905	demam_turun_naik
t_vo33	vo907	batuk
t_vo33	vo908	batuk_3mgg
t_vo33	vo989	penurunan_berat
t_vo33	vo982	ruam
t_vo33	vo960	penurunan_kesadaran
t_vo33	vo963	kejang
t_vo33	vo981	urin_sedikit
t_vo33	vo906	menggigil
t_vo33	vo913	dispnea
t_vo33	vo914	dispnea_3mgg
t_vo33	vo918	nyeri_dada
t_vo33	vo910	dahak
t_vo33	vo986	mata_merah
t_vo33	vo991	bengkak
t_vo33	vo917	mengi

3. Integrasi Data (*Data Integration)*

Dalam penulisan tugas akhir kali ini diasumsikan bahwa data yang diambil sudah berupa tabel-tabel dalam satu server. Dari sumber data yang sudah dijabarkan sebelumnya, berikut integrasi data yang diusulkan untuk penyelesaian *mining* identifikasi penyebab kematian, dari data korespondensi penyebab kematian untuk otopsi verbal dengan kode ICD-10 terbentuk tabel t_tema, dari tabel otopsi verbal terpilih terbentuk tabel t_symptoms dimana hubungan keduanya adalah *many to many*, maka menghasilkan tabel t_temasymp sebagai tabel referensi. Berikut struktur tabel – tabel tersebut :

Tabel 3.6 Struktur Tabel t_symptoms

Field	Type	Null	Default
id_symp	int(11)	No	
Symptoms	varchar(50)	No	
Keterangan	varchar(100)	Yes	*NULL*

Tabel 3.7 Isi Tabel t_symptoms

id_symp	symptoms	keterangan
1	demam	Demam
2	demam_turun_naik	Demam Turun Naik
3	batuk	Batuk
4	batuk_3mgg	Batuk kurang dari 3 minggu
5	penurunan_berat	Penurunan Berat Badan
6	ruam	Ruam
7	penurunan_kesadaran	Penurunan Kesadaran
8	kejang	Kejang
9	urin_sedikit	Produksi Urin Sedikit
10	menggigil	Menggigil / Kedinginan

11	dispnea	Dispnea / Sesak Napas
12	dispnea_3mgg	Dispnea kurang dari 3 minggu
13	nyeri dada	Nyeri Dada
14	dahak	Produksi dahak meningkat
15	mata_merah	Matah merah
16	bengkak	Pembengkakan / Edema
17	mengi	Mengi

Tabel 3.8 Struktur Tabel t_tema

Field	Type	Null	Default
id_va	int(11)	No	
kode_va	varchar(8)	Yes	*NULL*
tema_va	varchar(100)	Yes	*NULL*

Tabel 3.9 Isi Tabel t_symptoms

id_va	kode_va	tema_va
1	VA-01.07	Campak
2	VA-01.10	Malaria
3	VA-01.13	Infeksi Saluran Napas Bawah (termasuk pneumonia dan bronkitis akut)
4	VA-05.01	Penyakit Paru Obstruktif Kronis
5	VA-05.02	Asma

Tabel 3.10 Struktur Tabel t_temasymp

Field	Type	Null	Default	Links to
id_symp	int(11)	Yes	*NULL*	t_symptoms -> id_symp
id_va	int(11)	Yes	*NULL*	t_tema -> id_va

Tabel 3.11 Isi Tabel t_temasymp

id_symp	id_va
1	1
1	2
1	3
2	2
3	1
3	3
3	4
3	5
4	3
4	4
5	4
6	1
7	2
8	2
9	2
10	2
11	3
11	4
11	5
12	3
13	3
14	4
15	1
16	3
17	5

4. Tranformasi Data (*Data Transformation)*

Dalam proses ini data akan ditransformasikan ke dalam data yang siap untuk di *mining*, hasil dari transformasi data ini menghasilkan tabel t_kasus. Proses tranformasi dimulai dari proses pembersihan tabel t_kasus untuk user sedang login (admin/*data manager)* dilanjutkan dengan proses pencarian *symptoms* ke

semua kategori umur untuk setiap kematian pada data otopsi verbal dan hasilnya akan disimpan dalam tabel t_kasus. Berikut struktur tabel t_kasus :

Tabel 3.12 Struktur Tabel t_kasus

Field	Type	Null	Default	Links to
id_vo	int(11)	No		t_vo -> id_vo
id_symp	int(11)	No		t_symptoms -> id_symp
id_user	int(11)	No		user -> id_user

Tabel 3.13 Isi Tabel t_kasus

id_vo	id_symp	id_user
1	5	61
2	14	61
2	11	61
2	1	61
2	5	61
2	3	61
3	8	61
3	5	61
4	11	61
4	8	61
4	5	61
4	4	61
4	3	61
5	1	61
6	16	61
6	8	61
6	1	61
6	5	61
6	1	61
7	17	61
7	14	61
7	11	61
7	5	61

7	4	61
7	3	61
8	2	61
8	1	61
9	11	61
9	5	61
9	4	61
9	3	61
10	16	61
10	5	61
10	2	61
10	1	61
12	11	61
12	5	61
13	16	61
13	10	61
13	5	61
13	2	61
13	1	61
14	5	61
15	16	61
15	1	61
15	5	61
17	11	61
17	6	61
17	5	61
18	9	61
18	1	61
19	11	61
19	9	61
19	5	61
19	4	61
19	3	61
20	1	61
24	16	61
24	11	61
24	1	61

24	5	61
24	4	61
24	3	61
24	1	61
25	1	61
25	5	61
26	16	61
26	14	61
26	10	61
26	5	61
26	3	61
26	2	61
26	1	61
27	5	61
28	17	61
28	16	61
28	11	61
28	5	61
29	15	61
29	5	61
30	1	61
30	5	61

5. Proses Mining

Setelah didapatkan data siap *mining* yang disimpan dalam tabel t_kasus, langkah selanjutnya adalah melakukan proses *mining,* yaitu penerapan metode untuk menemukan pengetahuan dalam data. Karena metode klasifikasi yang diterapkan disini adalah metode *decision tree* dengan jenis *tree* yang merupakan *forward chaining tree,* maka algoritma yang diterapkan adalah algoritma *forward chaining.* Proses mining dimulai dari tari tabel t_kasus, dimana dapat kita ambil *symptoms* yang paling sering muncul setiap kasusnya:

Tabel 3.14 Tabel jumlah kemunculan symptoms

Id_symp	Symtomps	Jumlah
5	Penurunan Berat Badan	21
1	Demam	15
11	Dispnea	9
16	Bengkak	7
3	Batuk	7
4	Batuk < 3 minggu	5
2	Demam Naik Turun	4
8	Kejang	3
14	Produksi Dahak Meningkat	3
9	Produksi Urin Sedikit	2
17	Mengi	2
10	Menggigil	2
6	Ruam	1
15	Mata Merah	1

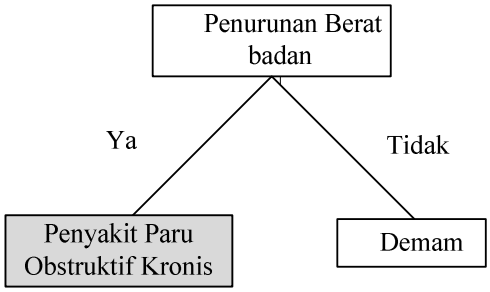
Dari tabel tersebut, terlihat bahwa *symptoms* penurunan berat badan dengan id_symp = 5 dipilih sebagai atribut terbaik karena mempunyai kemunculan terbanyak setiap kasusnya, maka *symptoms* tersebut dapat kita jadikan sebagai *root* atau akar dari pohon *decision tree* yang akan kita bangun. Sedangkan untuk penanganan *symptoms* dengan jumlah kemunculan sama, digunakan tabel acuan pendekatan *symptoms* dengan tema influenza dari pakar, berikut tabel acuan tersebut:

Tabel 3.15 Tabel acuan urutan symptoms

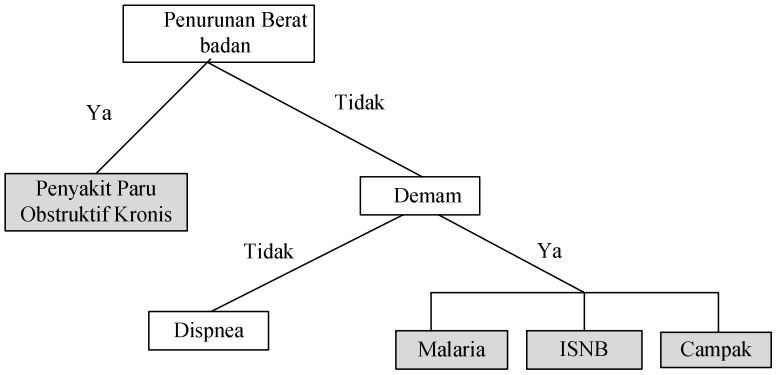
Id_symp	Symtomps	Urutan
1	Demam	1
2	Demam Naik Turun	2
3	Batuk	3

4	Batuk < 3 minggu	4
10	Menggigil	5
6	Ruam	6
8	Kejang	7
11	Dispnea	8
14	Produksi Dahak Meningkat	9
9	Produksi Urin Sedikit	10
15	Mata Merah	11
17	Mengi	12
5	Penurunan Berat Badan	13
16	Bengkak	14

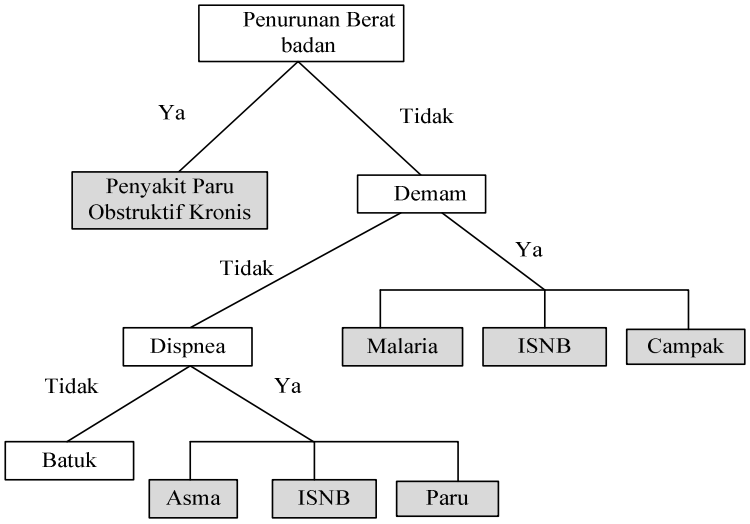
Berdasarkan tabel kemunculan *symptoms* dan data acuan urutan *symptoms* yang paling berpengaruh terhadap kasus influenza, berikut langkah pembentukan *decision tree* yang dibangun:



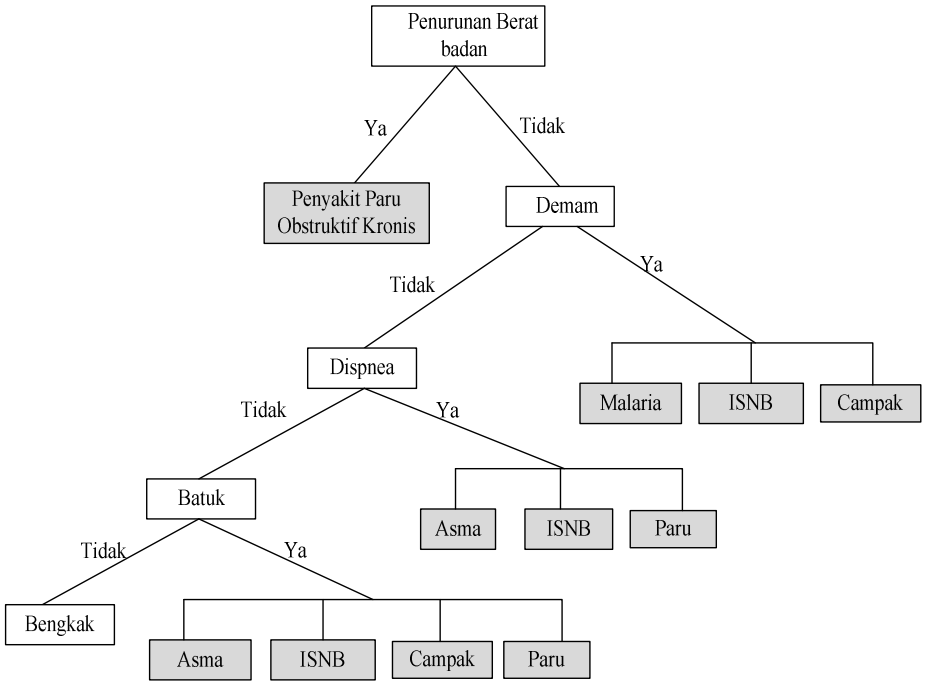
Gambar 3.2 *Tree* Tahap 1



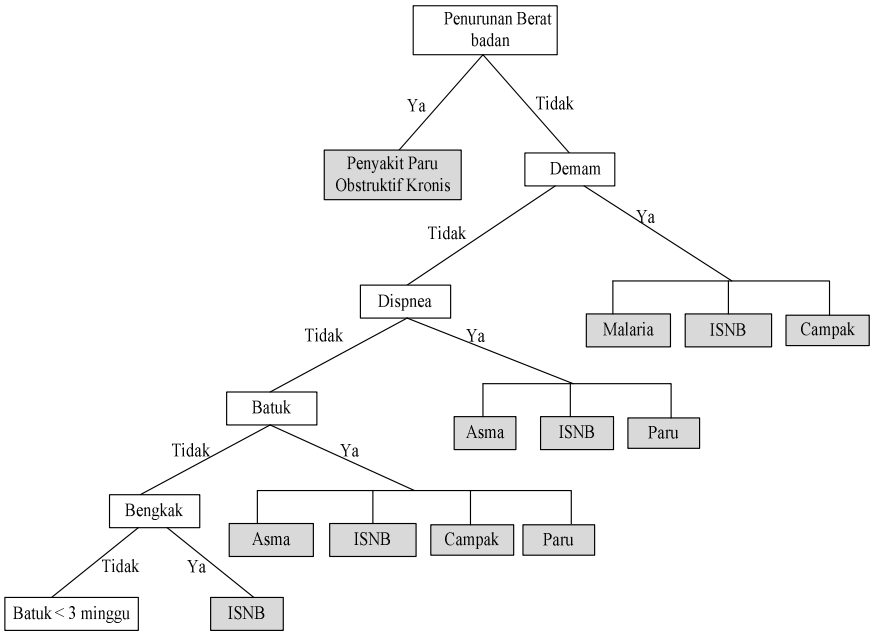
Gambar 3.3 *Tree* Tahap 2



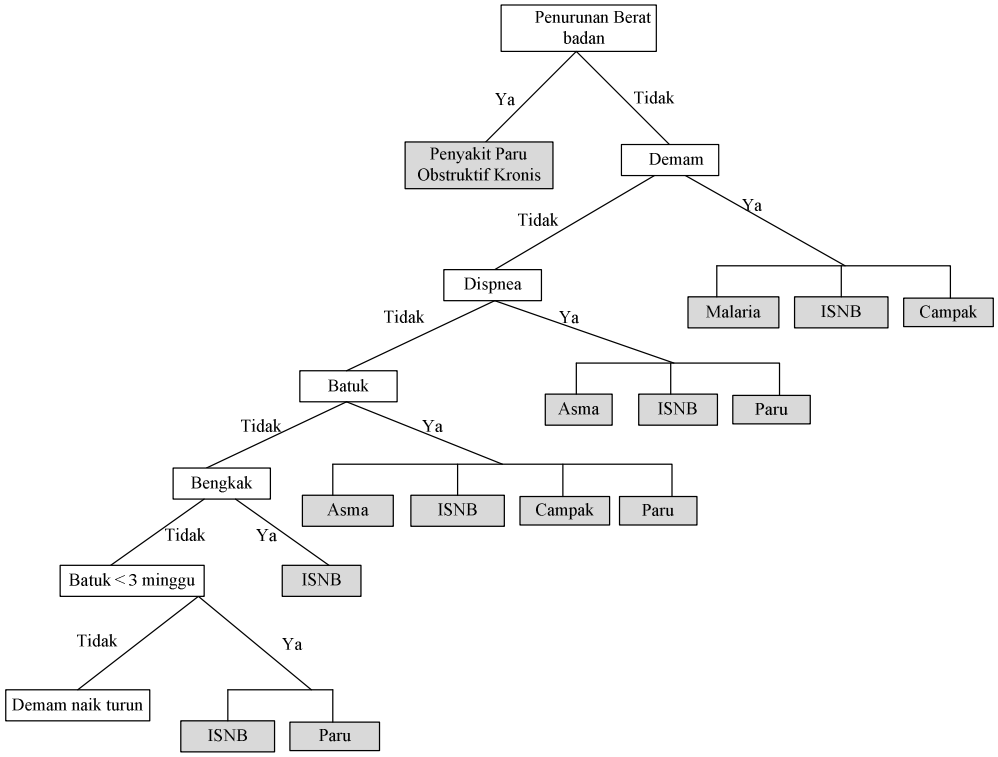
Gambar 3.4 *Tree* Tahap 3



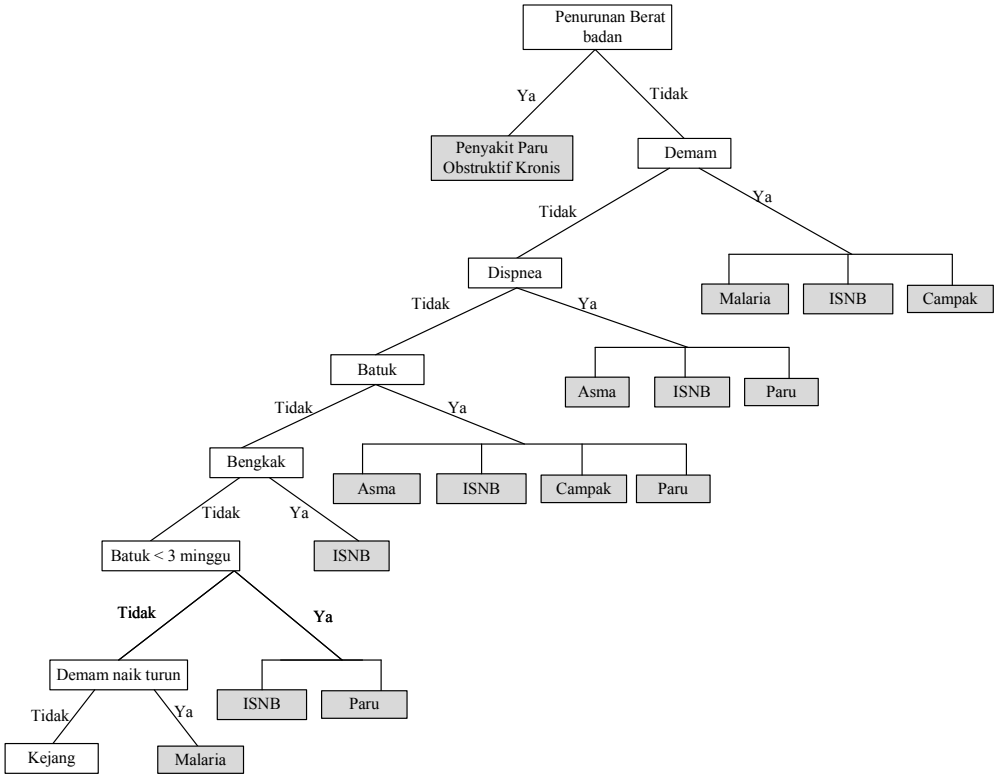
Gambar 3.5 *Tree* Tahap 4



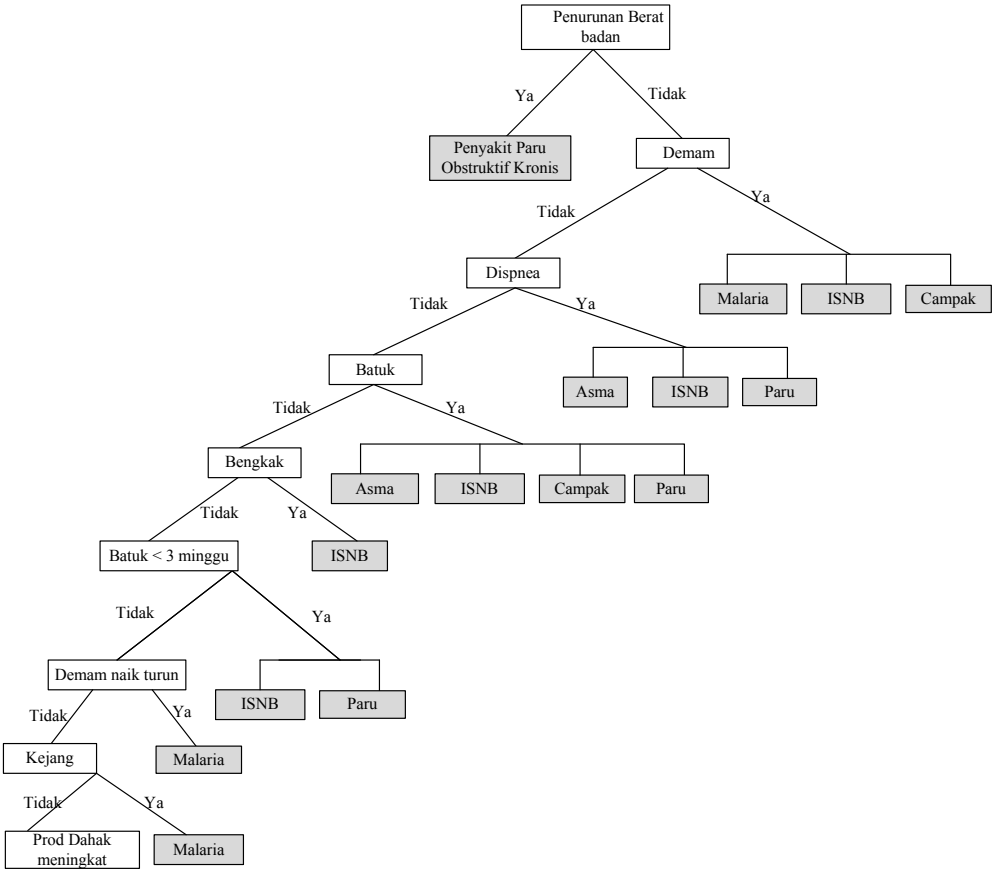
Gambar 3.6 *Tree* Tahap 5



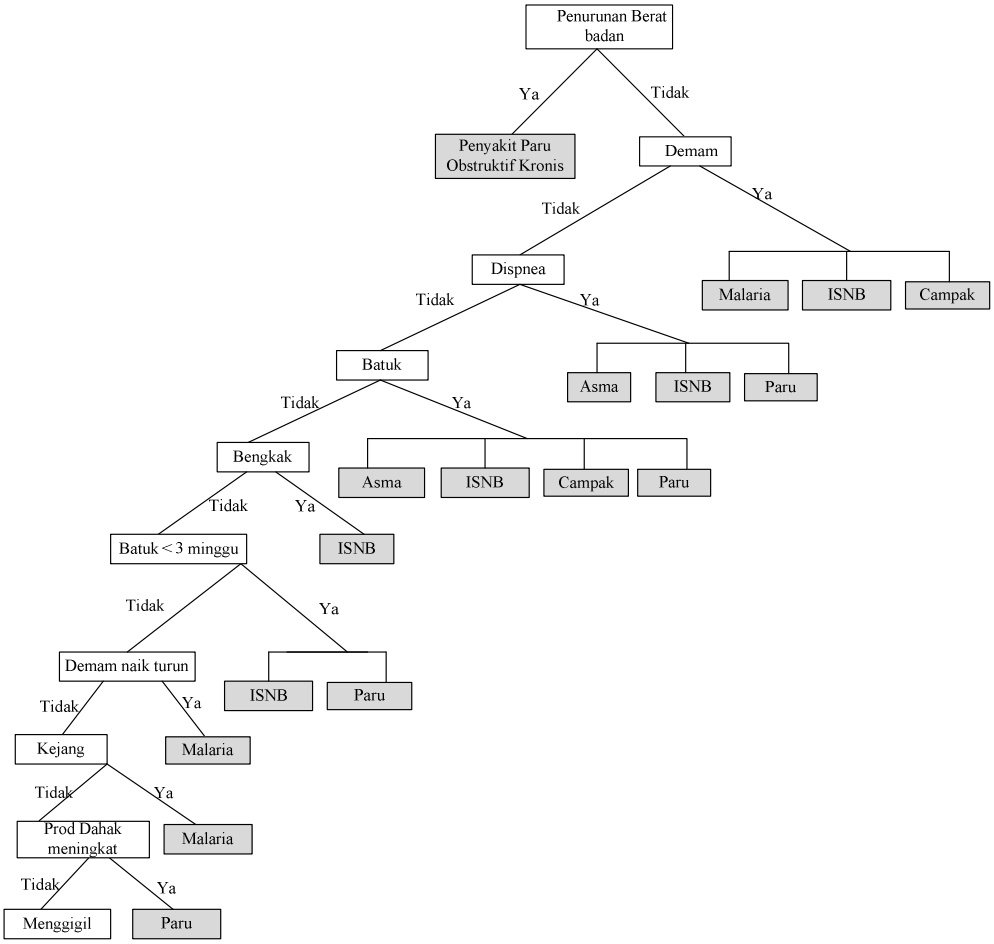
Gambar 3.7 *Tree* Tahap 6



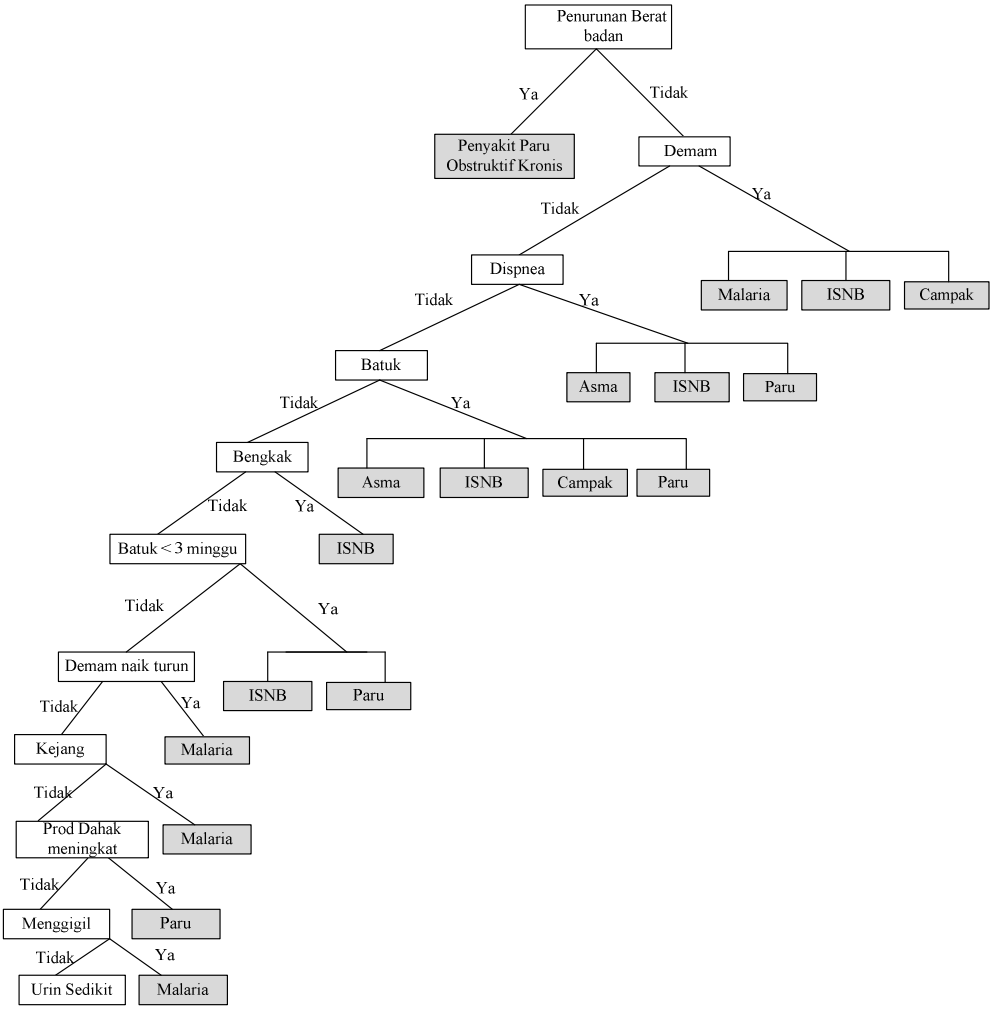
Gambar 3.8 *Tree* Tahap 7



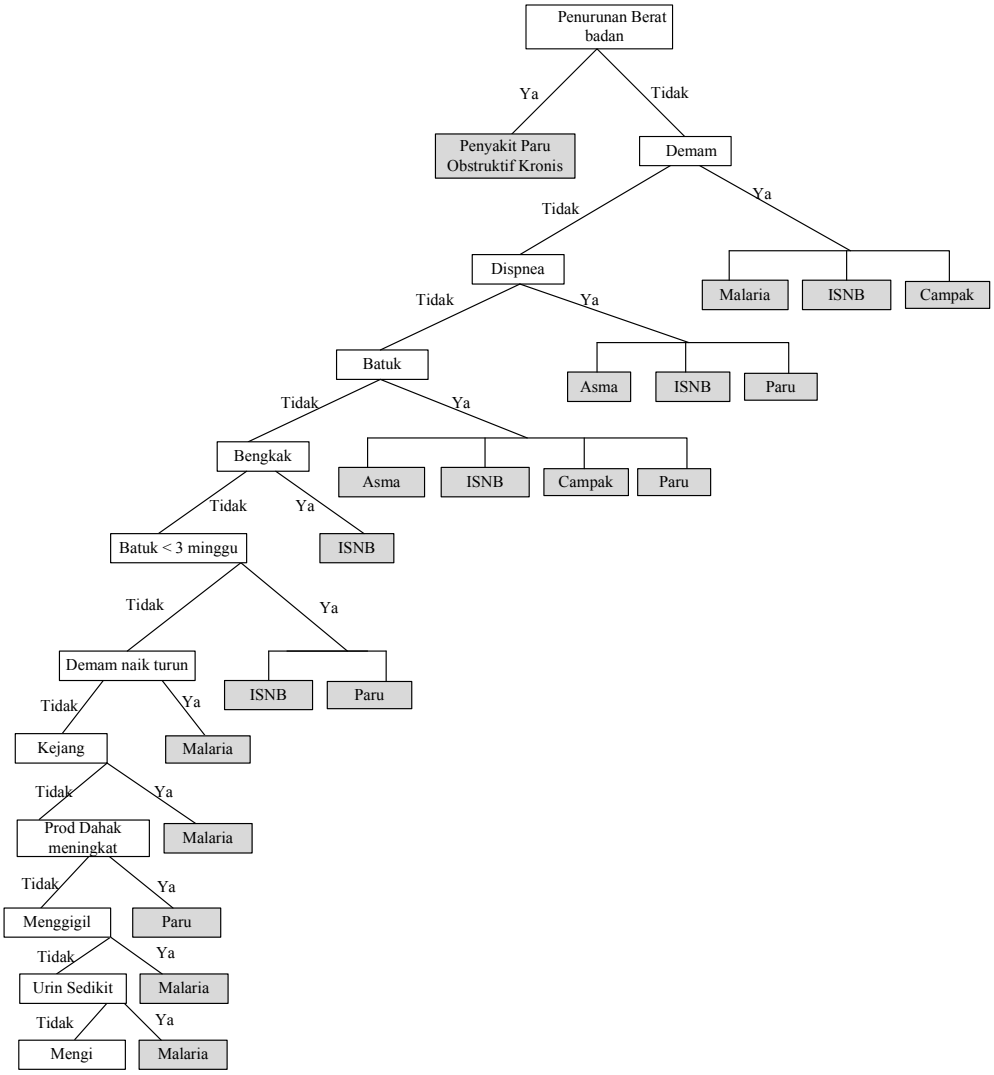
Gambar 3.9 *Tree* Tahap 8



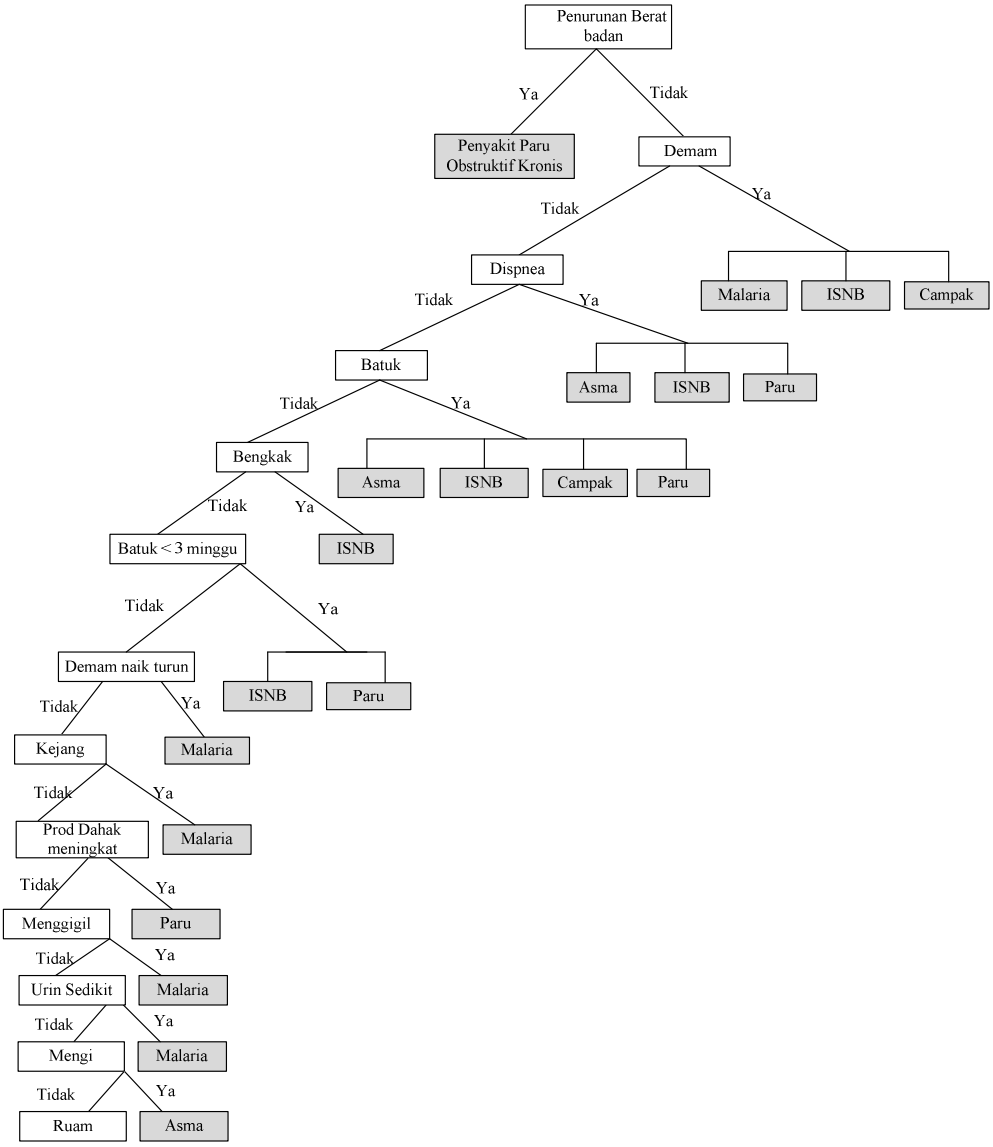
Gambar 3.10 *Tree* Tahap 9



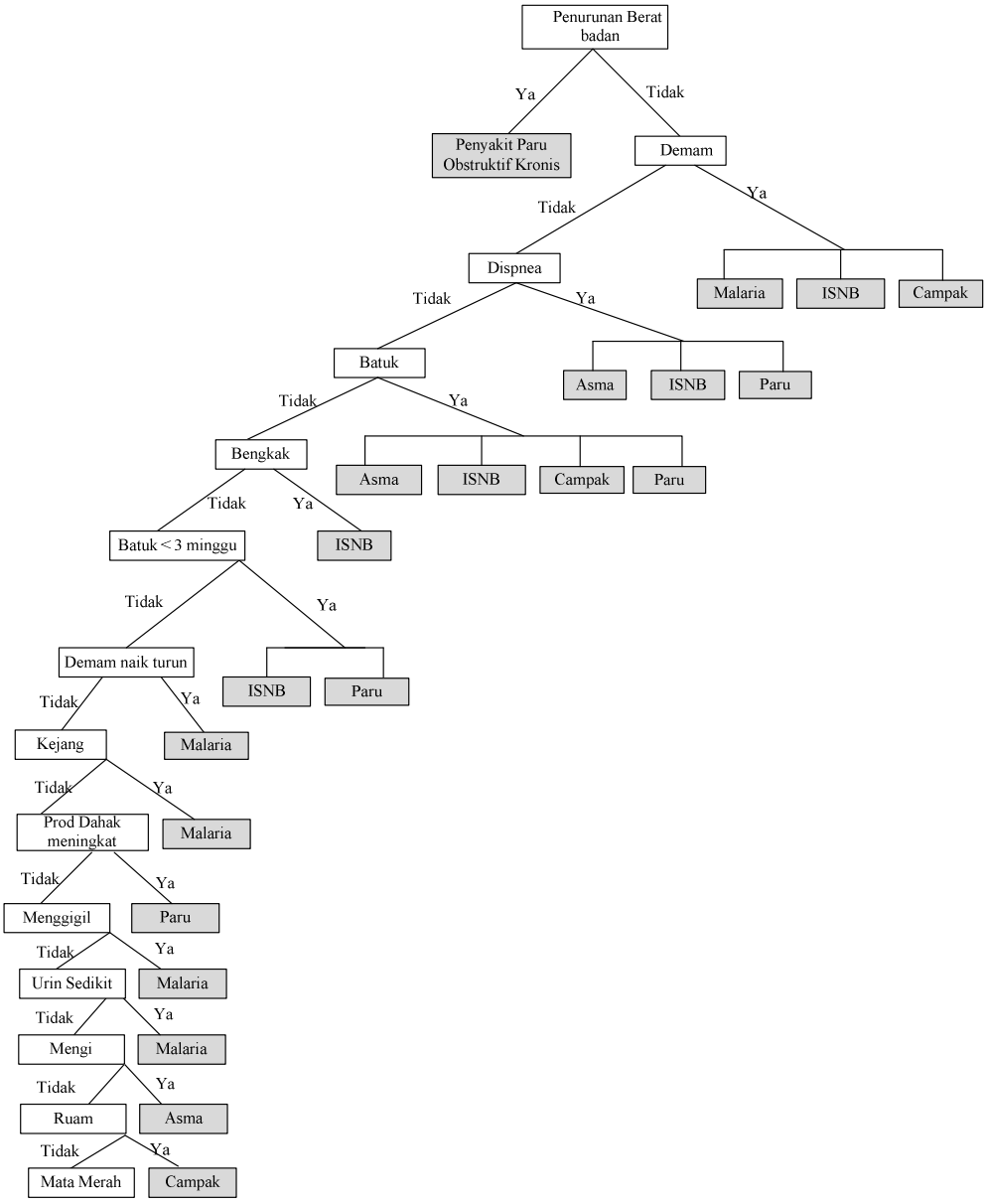
Gambar 3.11 *Tree* Tahap 10



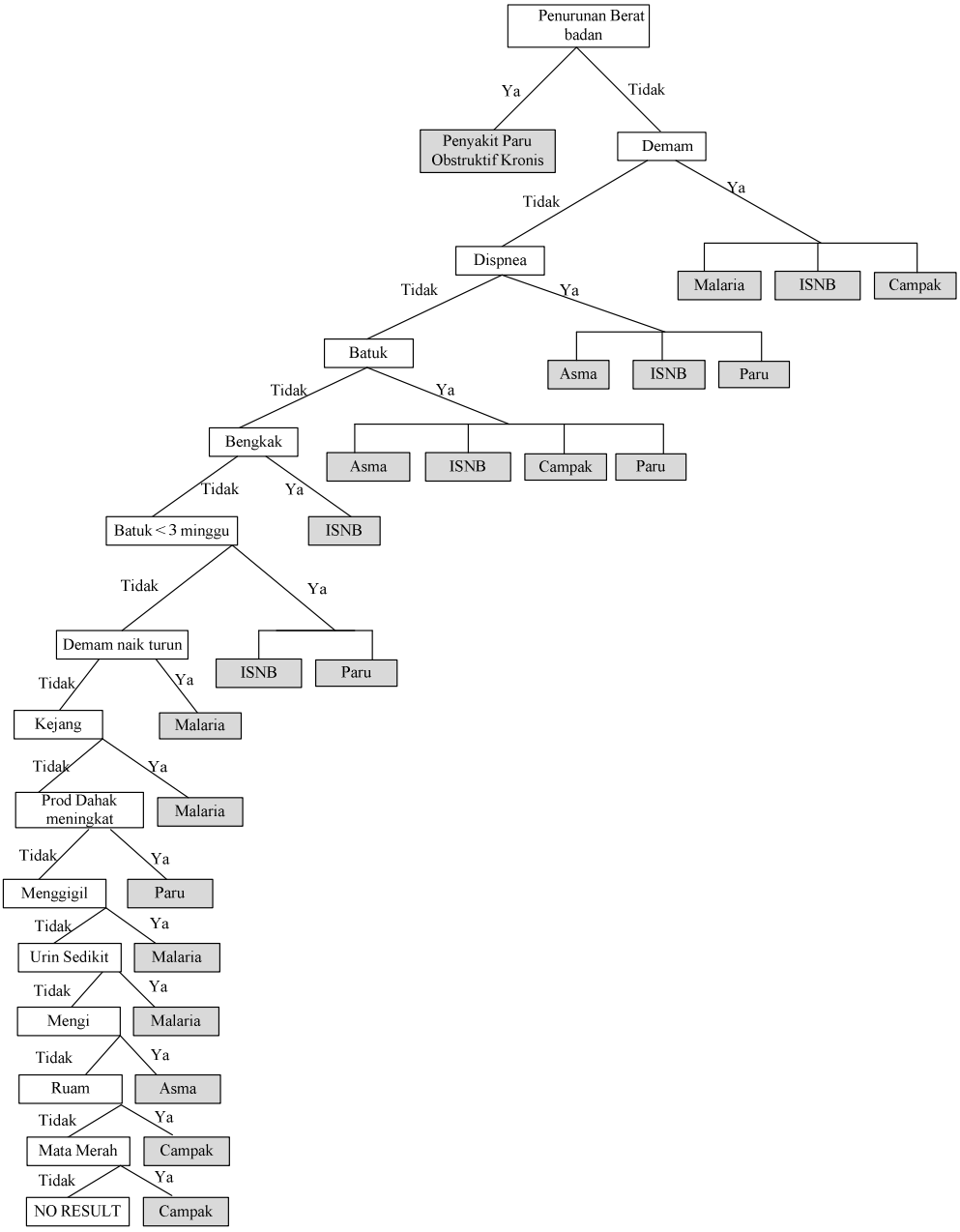
Gambar 3.12 *Tree* Tahap 11



Gambar 3.13 *Tree* Tahap 12



Gambar 3.14 *Tree* Tahap 13



Gambar 3.15 Hasil Akhir *Tree*

Dari proses pembentukan tree yang digambarkan, terlihat bahwa alur pada *decision tree* di telusuri dari simpul akar terpilih ke simpul daun yang

memegang prediksi kelas untuk contoh tersebut. Dimana penelusuran simpul menerapkan algoritma runut maju atau *forward chaining* yaitu strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta dalam tabel t_kasus, dan dari data-data tersebut dicari suatu kesimpulan yang menjadi solusi permasalahan yang dihadapi berdasarkan kaidah-kaidah basis pengetahuatan dalam tabel t_temasymp. Basis pengetahuan dalam t_temasymp menjelaskan bahwa sebuah *symptoms* bisa merupakan bagian dari satu atau lebih tema otopsi verbal (*verbal autopsy themes).* Berikut kaidah dari akuisisi pengetahuan yang dijabarkan dalam susunan pohon *forward chaining*:

Tabel 3.16 Kaidah Identifikasi penyebab Kematian

Symptoms	**Campak**	**Malaria**	**ISNB**	**Paru**	**Asma**
Penurunan berat badan				√	
Demam	√	√	√		
Dispnea					√
Batuk	√		√	√	√
Bengkak			√		
Batuk < 3 Minggu			√	√	
Demam naik turun		√			
Kejang		√			
Prod Dahak Meningkat				√	
Menggigil		√			
Urin Sedikit		√			
Mengi					√
Ruam	√				
Mata Merah	√				

6. Presentasi Pengetahuan

Sesuai tahapan decision tree yaitu dari dati diubah ke *tree* dan langkah selanjutnya adalah merepresentasikan tree tersebut ke dalam *rule* atau aturan. Aturan ditulis dalam struktur IF – THEN, berikut aturan-aturan tersebut :

```
Rule_1:
IF
    symptoms = demam
THEN
    tema otopsi verbal =  Campak
    tema otopsi verbal =  Malaria
    tema otopsi verbal = Infeksi saluran napas bawah
Rule_2:
IF
    symptoms = demam_turun_naik
THEN
    kode otopsi verbal =  Malaria
Rule_3:
IF
    symptoms = batuk
THEN
    tema otopsi verbal =  Campak
    tema otopsi verbal =  Infeksi saluran napas bawah
    tema otopsi verbal =  Penyakit paru obstruktif kronis
    tema otopsi verbal =  Asma
Rule_4:
IF
    symptoms = batuk < 3 minggu
THEN
    tema otopsi verbal =  Infeksi saluran napas bawah
    tema otopsi verbal =  Penyakit paru obstruktif kronis
Rule_5:
IF
    symptoms = penurunan berat badan
THEN
    tema otopsi verbal =  Penyakit paru obstruktif kronis
Rule_6:
IF
    symptoms = ruam
THEN
```

```
    tema otopsi verbal =  Campak
Rule_7:
IF
    symptoms = penurunan kesadaran
THEN
    Tema otopsi verbal =  Malaria
Rule_8:
IF
    symptoms = kejang
THEN
    tema otopsi verbal =  Malaria
Rule_9:
IF
    symptoms = produksi urin sedikit
THEN
    tema otopsi verbal =  Malaria
Rule_10:
IF
    symptoms = menggigil/kedinginan
THEN
    tema otopsi verbal =  Malaria
Rule_11:
IF
    symptoms = dispnea/sesak napas
THEN
    tema otopsi verbal =  Infeksi saluran napas bawah
    tema otopsi verbal =  Penyakit paru obstruktif kronis
    tema otopsi verbal =  Asma
Rule_12:
IF
    symptoms = dispnea < 3 minggu
THEN
    tema otopsi verbal =  Infeksi Saluran Napas Bawah
Rule_13:
IF
    symptoms = nyeri dada
THEN
    tema otopsi verbal =  Infeksi Saluran Napas Bawah
Rule_14:
IF
    symptoms = dahak
THEN
    tema otopsi verbal =  Penyakit paru obstruktif kronis
```

```
Rule_15:
IF
    symptoms = mata_merah
THEN
    tema otopsi verbal =  Campak
Rule_16:
IF
    symptoms = bengkak/edema
THEN
    tema otopsi verbal =  Infeksi saluran napas bawah
Rule_17:
IF
    symptoms = mengi
THEN
    tema otopsi verbal =  Asma
```

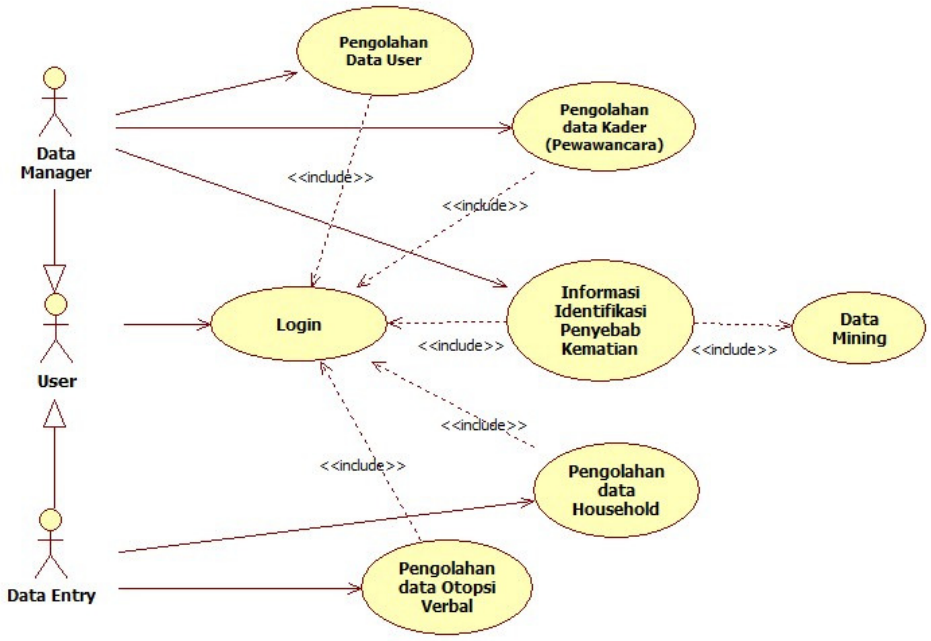
Presentasi pengetahuan dari proses *mining* disini adalah sebuah kesimpulan apakah sebuah kasus kematian mempunyai satu atau lebih *symptoms* influenza dan teridentifikasi disebabkan oleh satu atau lebih tema otopsi verbal (*verbal autopsy themes*) atau tidak mempunyai *symptoms* influenza yang berarti tidak akan teridentifikasi penyebabnya.

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem. Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*), dengan tahapan antara lain, use case diagram, skenario, activity diagram dan sequence diagram

3.1.6.1 *Use Case Diagram*

Use case diagram menggambarkan secara grafis perilaku aplikasi identifikasi penyebab kematian pada penelitian CIRAI. Dimana perilaku aplikasi terdiri dari dua aktor, yaitu *Data Manager* dan *Data Entry*.



Gambar 3.16 *Use Case Diagram*

3.1.6.1.1 *Use Case* dan Skenario

1. **Use Case Login**

Identifikasi	
Nama Use case	Login
Deskripsi	Melakukan proses login ke aplikasi/system.
Aktor	Data Entry, Data Manager

Kondisi Awal		Pengguna memasukkan user id dan password ke halaman login aplikasi.	
Kondisi Akhir		Menampilkan halaman utama aplikasi CIRAI, dimana menu yang ditampilkan sesuai dengan level pengguna (admin/entry).	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	Memasukkan user dan password		-
2	Memilih login	3	Sistem akan mengecek apakah user dan password yang diinputkan telah terdaftar dan valid. Jika user , password valid, halaman utama akan ditampilkan.
Komentar			
Untuk proses login akan dilakukan 2 kali proses pengecekan/validasi, pertama apakah user yang dimasukkan sudah terdaftar, kedua apakah user dan password sesuai.			

2. Use Case Pengolahan Data User

Identifikasi			
Nama Use case		Pengolahan Data User	
Deskripsi		Mengelola data pengguna sistem.	
Aktor		Data Manager	
Kondisi Awal		Pengguna telah login sebagai admin dan memilih menu otopsi verbal.	
Kondisi Akhir		Menampilkan halaman list data pengguna setelah terjadi proses pengolahan data baik input, edit, ataupun delete	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	memilih proses add user	2	Sistem akan menampilkan Form add user
3	Memilih tombol submit	4	sistem akan menyimpan data hasi proses input pada data base

5	memilih proses edit dan memilih data yang akan di edit	6	sistem akan menampilkan form edit user
7	Memilih menu update	8	sistem akan menyimpan data yang telah di edit pada data base
9	Memilih tombol delete pada data terpilih	10	sistem akan menghapus data yang dipilih pada database

3. Use Case Pengolahan Data Kader (Pewawancara)

Identifikasi			
Nama Use case		Pengolahan Data Kader (Pewawancara)	
Deskripsi		Mengelola data kader untuk digunakan pada form input data otopsi verbal sebagai pewawancara	
Aktor		Data Manager	
Kondisi Awal		Pengguna telah login sebagai admin dan memilih menu Kader.	
Kondisi Akhir		Menampilkan halaman list data kader setelah terjadi proses pengolahan data baik input, edit, ataupun delete	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	memilih proses add kader	2	Sistem akan menampilkan Form isian kader
3	Memilih tombol Submit	4	sistem akan menyimpan data hasi proses input pada data base
5	memilih proses edit dan memilih data yang akan di edit	6	sistem akan menampilkan form edit otopsi verbal
7	Memilih tombol submit	8	sistem akan menyimpan data yang telah di edit pada data base
9	Memilih tombol delete pada data terpilih	10	sistem akan menghapus data yang dipilih pada database

4. Use Case Pengolahan Data VO

Identifikasi			
Nama Use case		Pengolahan Data VO	
Deskripsi		Mengelola data otopsi verbal baik input, delete , maupun edit untuk data.	
Aktor		Data Entry	
Kondisi Awal		Pengguna telah login sebagai entry dan memilih menu otopsi verbal.	
Kondisi Akhir		Menampilkan halaman list data otopsi verbal setelah terjadi proses pengolahan data baik input, edit, ataupun delete	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	memilih proses input	2	Sistem akan menampilkan Form isian otopsi verbal
3	Memilih tombol save	4	sistem akan menyimpan data hasi proses input pada data base
5	memilih proses edit dan memilih data yang akan di edit	6	sistem akan menampilkan form edit otopsi verbal
7	Memilih tombol save	8	sistem akan menyimpan data yang telah di edit pada data base
9	Memilih tombol delete pada data terpilih	10	sistem akan menghapus data yang dipilih pada database
Komentar			
Untuk proses Input dan Update sistem memiliki proses validasi data dimana jika ada kesalahan dalam pengisian maupun perubahan data, maka sistem akan menampilkan sebuah pesan kesalahan agar entry dapat memperbaiki kesalahan tersebut			

5. Use Case Data Mining

Identifikasi	
Nama Use case	Data Mining
Deskripsi	Mengeluarkan laporan identifikasi penyebab kematian.
Aktor	Data Manager
Kondisi Awal	Pengguna telah login sebagai admin dan memilih menu otopsi verbal.

Kondisi Akhir		Menampilkan halaman laporan data otopsi verbal, berikut detail mengenai identifikasi penyebab kematian setiap kasusnya. Menyiman hasil mining ke database	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	memilih menu otopsi verbal	2	Sistem akan menghapus t_kasus untuk user yang sedang login dan menginsertkan data siap mining terbaru .
3	Memilih link *identify* dari salah satu kasus	4	sistem akan menjalankan proses mining dan penerapan algoritma *decision tree* serta menampilkan informasi detail identifikasi penyebab kematian dari kasus
			terpilih, berupa prosentase masing – masing penyebab yang mungkin.
Komentar			
Untuk proses data mining menerapkan teknik *classification* dengan metode *decision tree* dan algoritma *forward chaining*.			

6. Use Case Informasi Identifikasi Penyebab Kematian

Identifikasi			
Nama Use case		Laporan Identifikasi Penyebab Kematian	
Deskripsi		Mengeluarkan laporan identifikasi penyebab kematian.	
Aktor		Data Manager	
Kondisi Awal		Pengguna telah login sebagai admin, memilih menu otopsi verbal dan mengakses link identify atau view pada kasus terpilih	
Kondisi Akhir		Menampilkan halaman informasi hasil identifikasi penyebab kematian untuk kasus terpilih.	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	Memilih menu otopsi verbal	2	Sistem akan menampilkan list data otopsi verbal yang masuk ke

			sistem
3	Memilih link identify pada kasus terpilih	4	sistem akan menampilkan informasi identifikasi penyebab kematian yang didapat dari proses data mining.
5	Memilih link view pada kasus terpilih yang sudah teridentifikasi	6	sistem akan menampilkan informasi identifikasi penyebab kematian yang didapat dari proses data mining besert informasi tambahan berupa note jika kita mengisinya ketika proses identifikasi.
7	Memilih tombol print	8	System menampilkan jendela popup print hasil identifikasi penyebab kematian
Komentar			
Untuk proses pelaporan aplikasi identifikasi penyebab kematian disini menggunakan metode data mining dengan decision tree.			

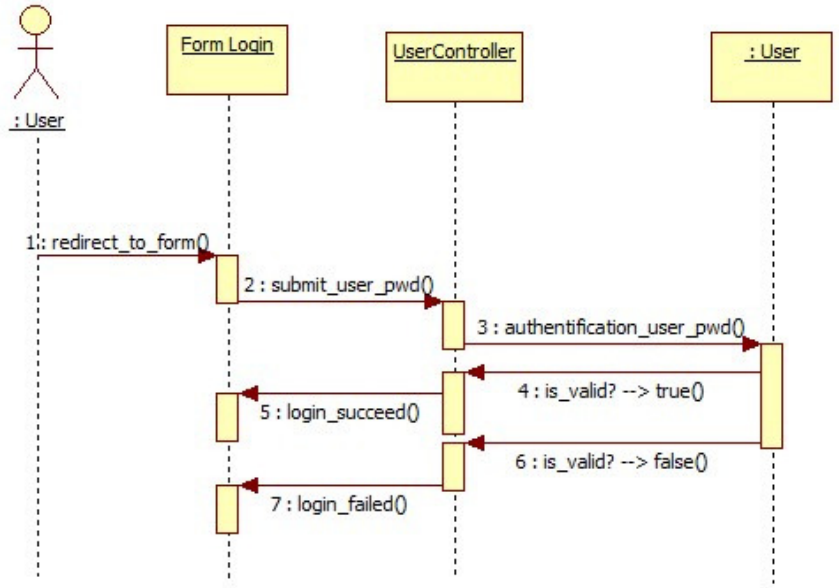
7. Use Case Pengolahan Data Household

Identifikasi			
Nama Use case		Pengolahan Data Household	
Deskripsi		Mengolah data household (input, edit, delete) jika dibutuhkan untuk aplikasi verbal otopsi.	
Aktor		Data Entry	
Kondisi Awal		Pengguna telah login sebagai data entry, memilih menu household.	
Kondisi Akhir		Data household berhasil ditambahkan, diubah atau dihapus.	
Skenario			
No	Aksi Aktor	No	Aksi Sistem
1	Memilih modul Household	2	Sistem akan menampilkan list data household yang masuk ke sistem
3	Memilih tombol “add household”	4	sistem akan menampilkan halaman form tambah data household.
5	Memasukkan data household yang ingin	6	Data household akan tersimpan dalam basis data system.

	ditambahkan pada form tambah data household dan memilih tombol submit		
7	Kembali ke aplikasi otopsi verbal untuk menginputkan id household dari data household terinput kedalam data otopsi verbal terpilih	8	System akan menerima id household yang diinputkan dan tidak akan menampilkan pesan error bahwa id household belum ada di data household
Komentar			
Pengolahan data household pada aplikasi otopsi verbal ini *tentative,* dalam artian pengolahan hanya akan dilakukan jika data otopsi verbal membutuhkan atau memerlukan, karena tidak semua atau belum semua data otopsi verbal memiliki data household.			

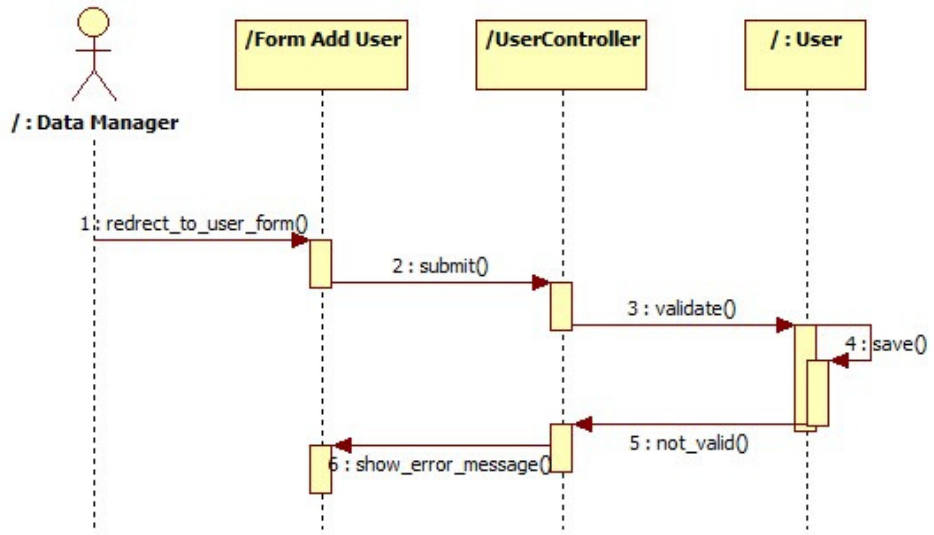
3.1.6.2 *Sequence Diagram*

1. ***Sequence Diagram*** **Login**

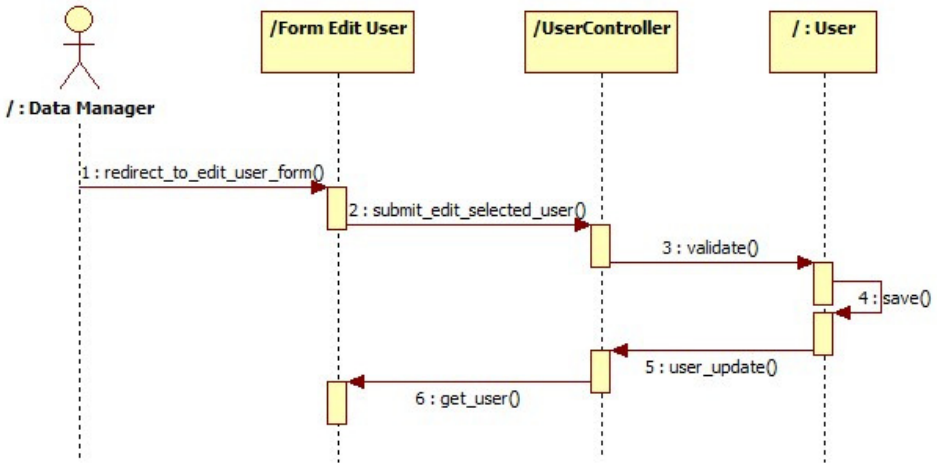


Gambar 3.17 *Sequence Diagram* Login

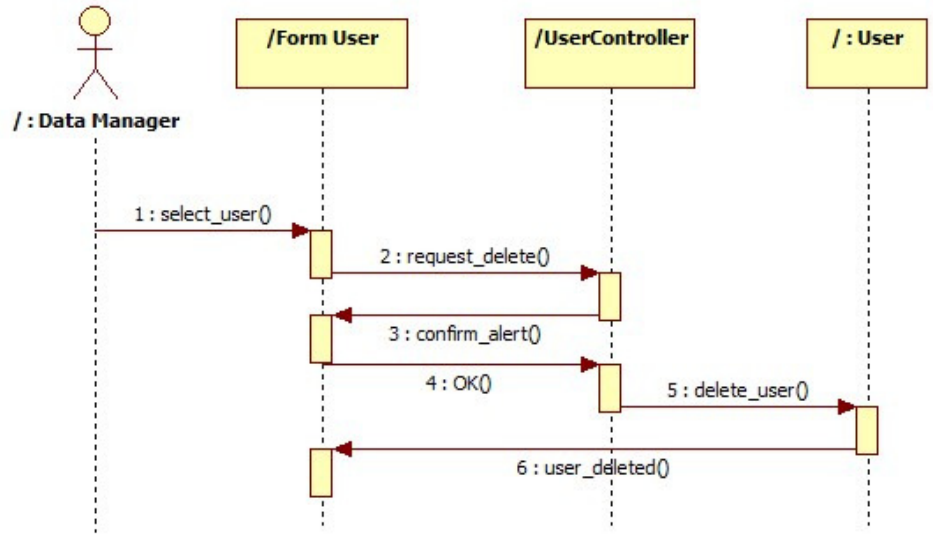
2. *Sequence Diagram* Pengolahan User



Gambar 3.18 *Sequence Diagram* Add User

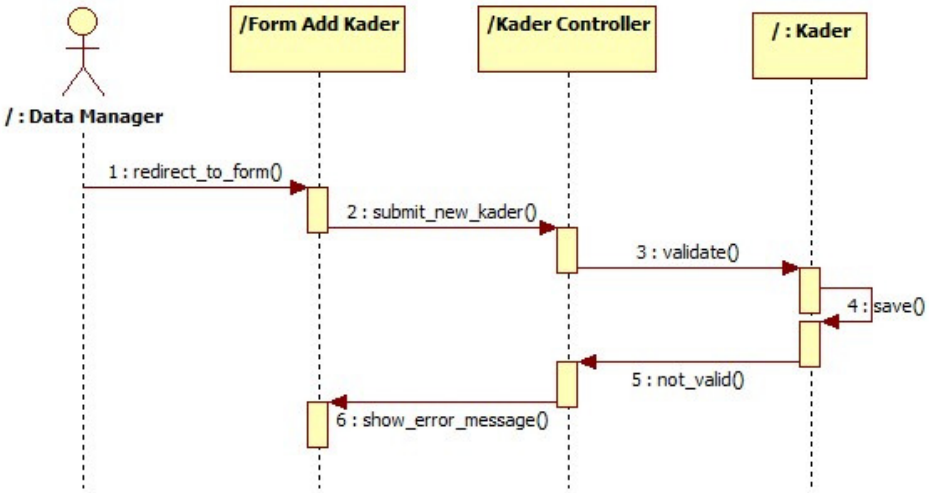


Gambar 3.19 *Sequence Diagram* Update User

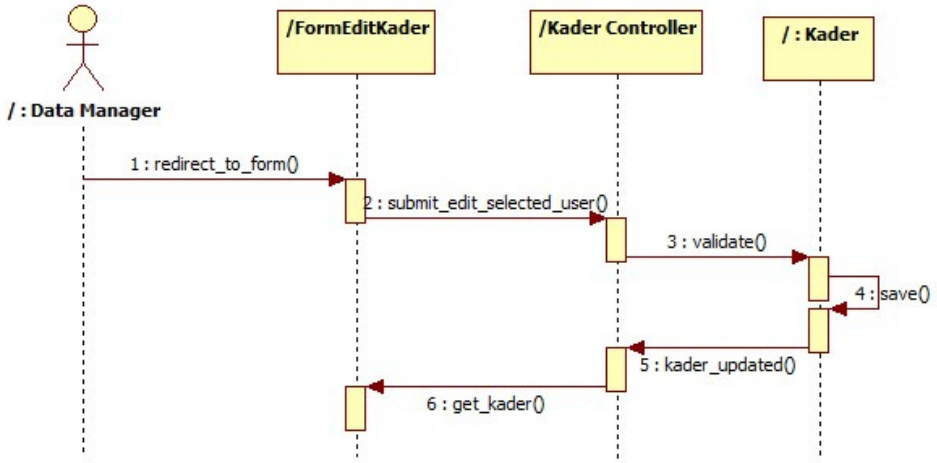


Gambar 3.20 *Sequence Diagram* Delete User

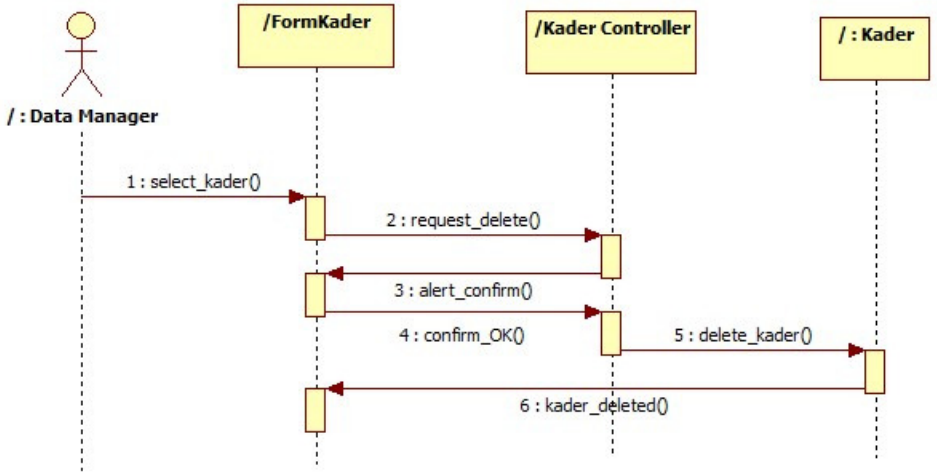
3. *Sequence Diagram* Pengolahan Kader



Gambar 3.21 *Sequence Diagram* Tambah Kader

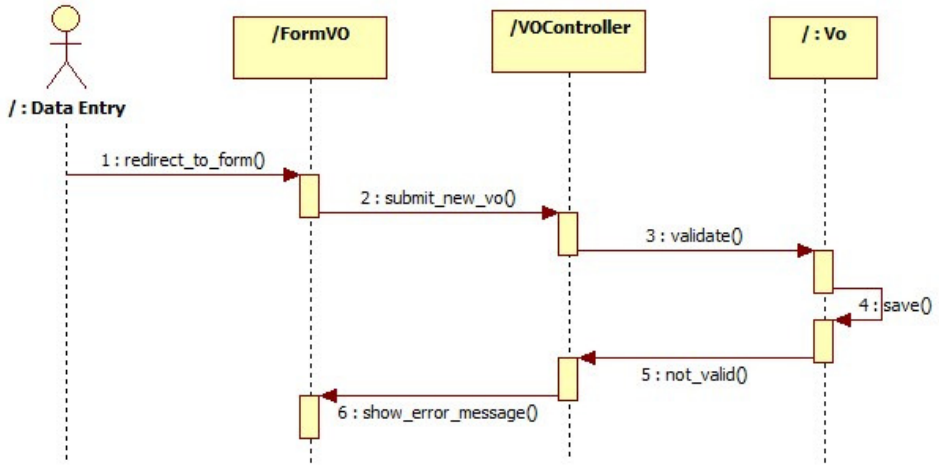


Gambar 3.22 *Sequence Diagram* Ubah Kader

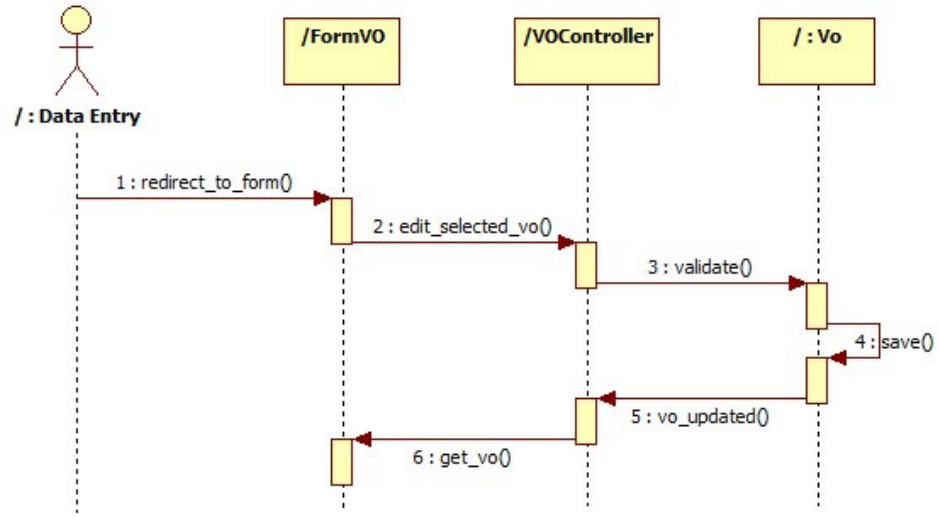


Gambar 3.23 *Sequence Diagram* Hapus Kader

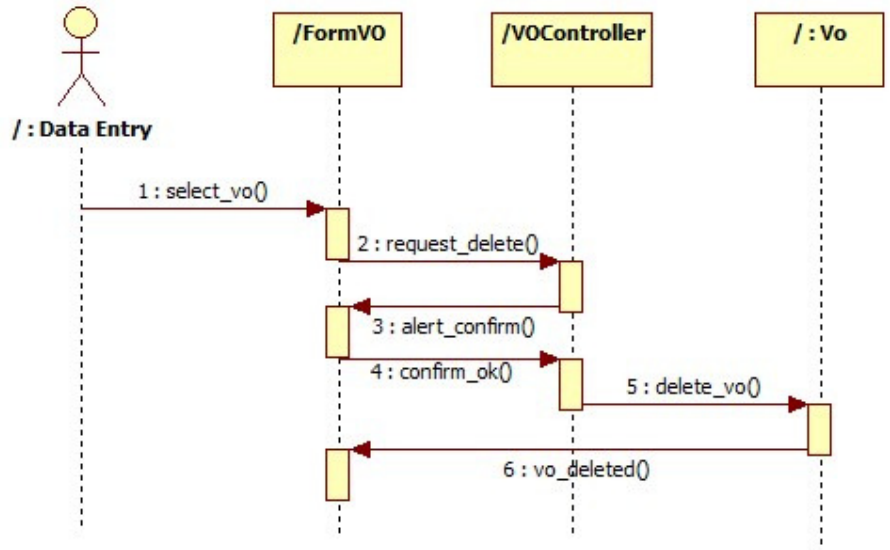
4. *Sequence Diagram* Pengolahan VO



Gambar 3.24 *Sequence Diagram* Tambah VO

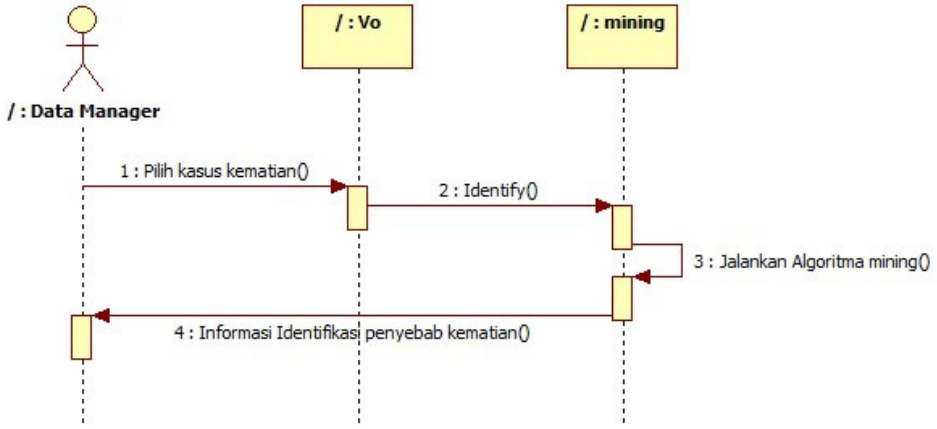


Gambar 3.25 *Sequence Diagram* Ubah VO



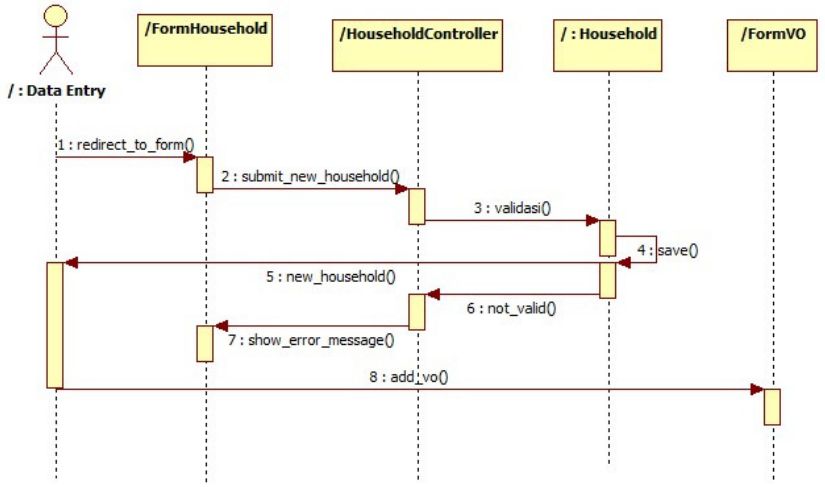
Gambar 3.26 *Sequence Diagram* Hapus VO

5. *Sequence Diagram* Identifikasi Penyebab Kematian



Gambar 3.27 *Sequence Diagram* Identifikasi Penyebab Kematian

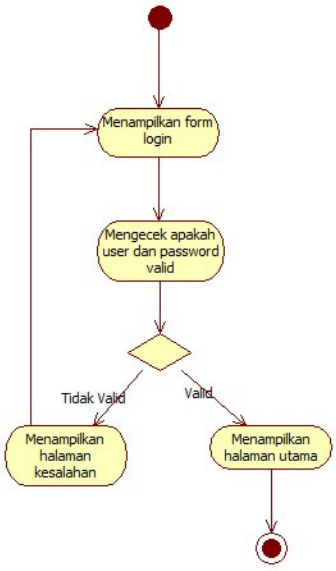
6. *Sequence Diagram* Pengolahan Data Household



Gambar 3.28 *Sequence Diagram* Pengolahan Household

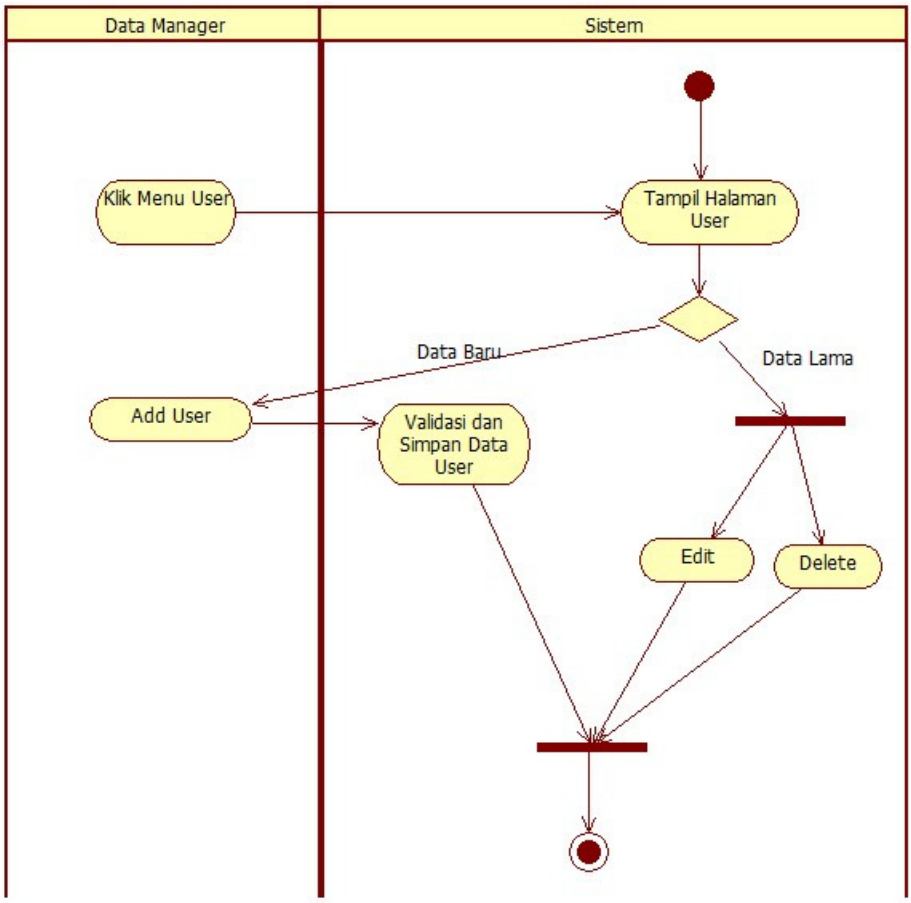
3.1.6.3 *Activity Diagram*

1. *Activity Diagram* proses Login



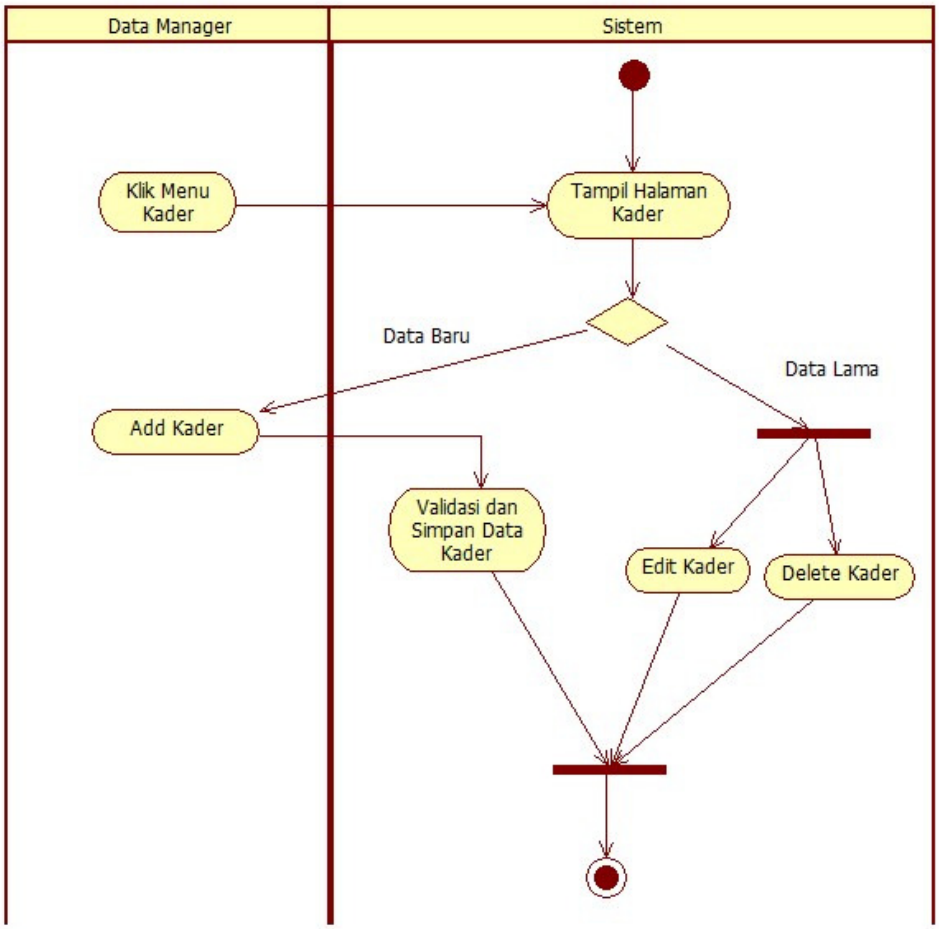
Gambar 3.29 Activity Diagram proses Login

2. *Activity Diagram* proses Pengolahan Data User



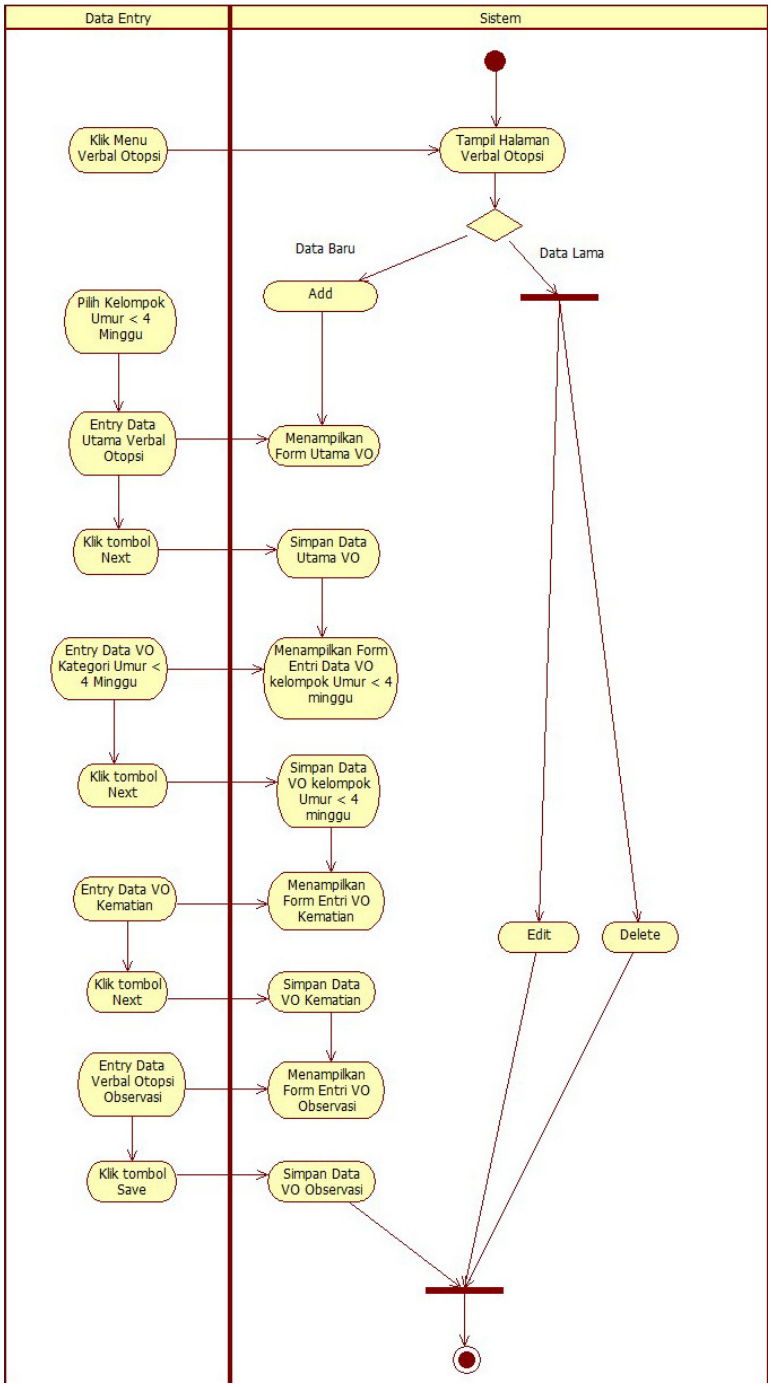
Gambar 3.30 Activity Diagram proses Pengolahan Data User

3. *Activity Diagram* proses Pengolahan Data Kader



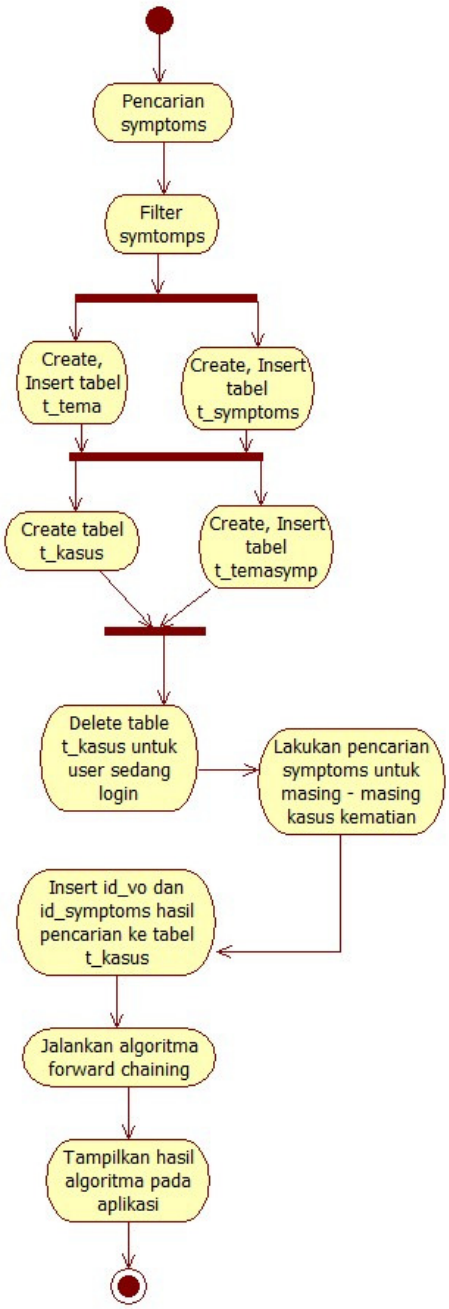
Gambar 3.31 Activity Diagram proses Pengolahan Data Kader

4. *Activity Diagram* proses Pengolahan Data VO



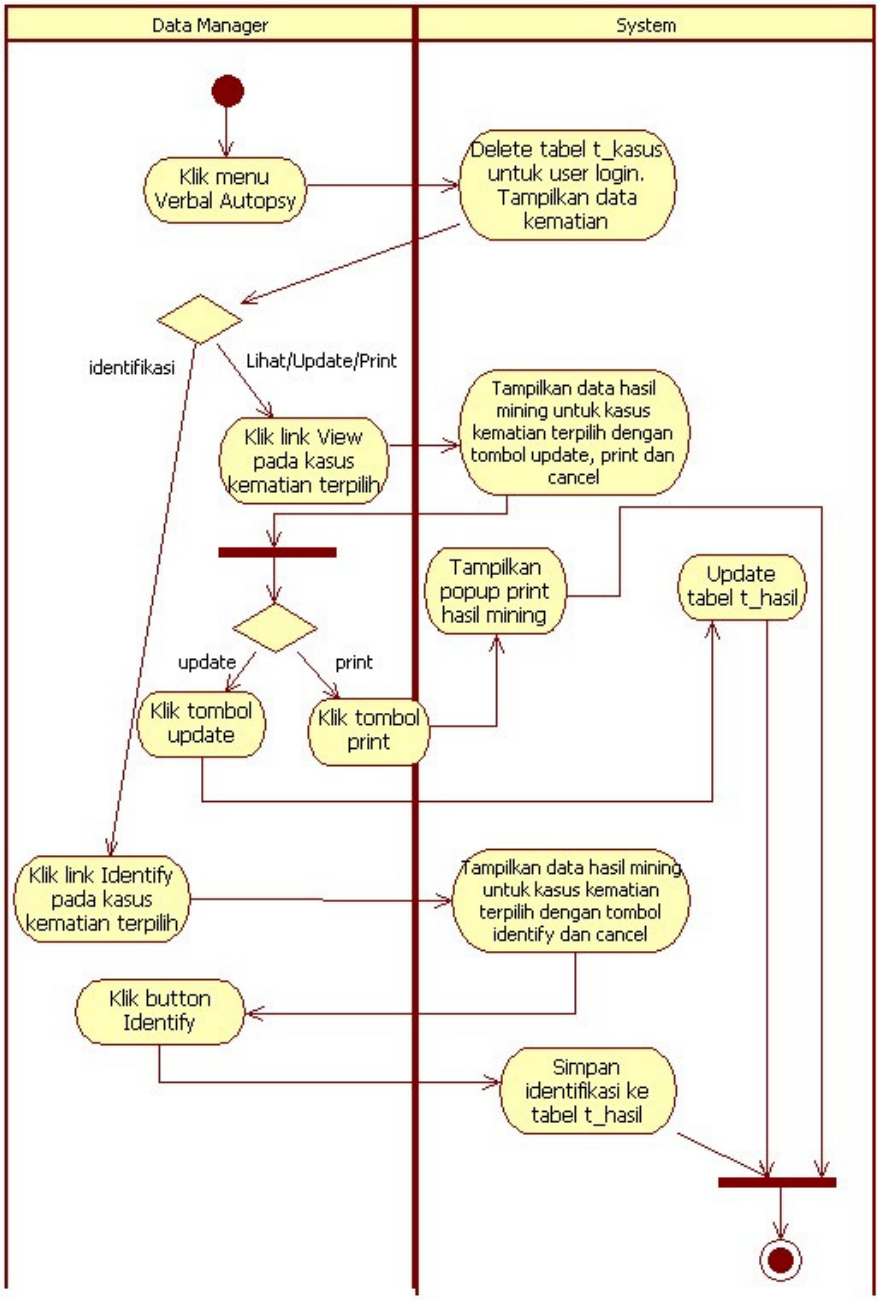
Gambar 3.32 Activity Diagram proses Pengolahan Data VO

5. *Activity Diagram* proses Data Mining

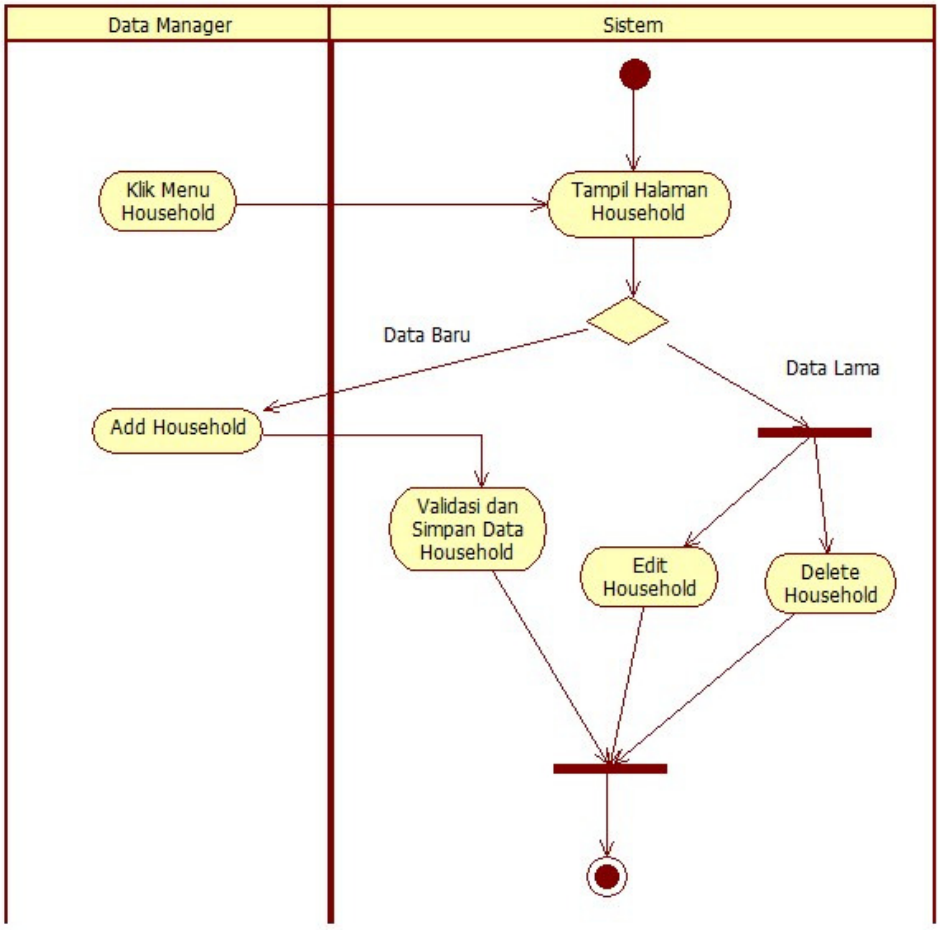


Gambar 3.33 Activity Diagram proses *data mining*

6. *Activity Diagram* proses Identifikasi Penyebab Kematian

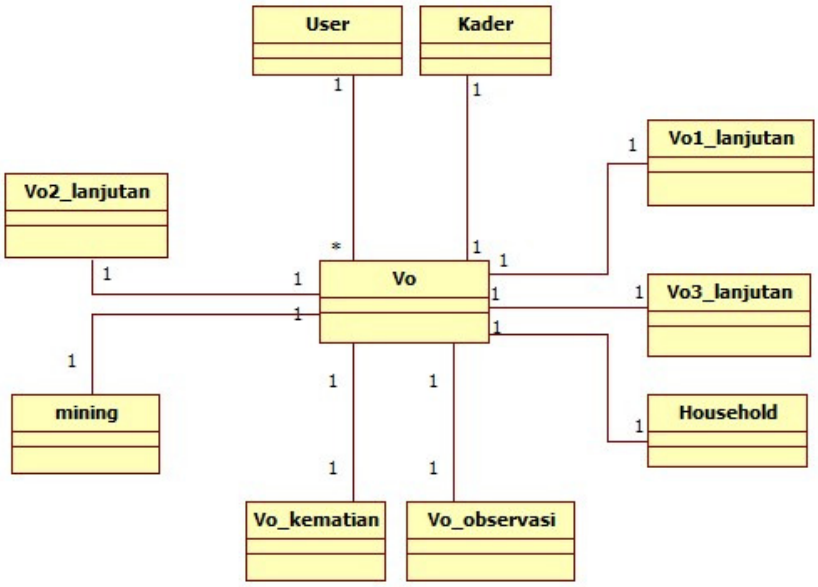


Gambar 3.34 Activity Diagram Identifikasi Penyebab Kematian



Gambar 3.35 Activity Diagram Pengolahan Data Household

3.1.6.4 *Class Diagram*



Gambar 3.36 *Class Diagram*

Karena atribut dari kelas yang ada sangat banyak, maka berikut keterangan mengenai spesifikasi dan deskripsi dari masing – masing kelas yang ada:

3.1.6.4.1 Spesifikasi *Class*

Tabel 3.17 Spesifikasi *Class*

No	Nama Kelas	Jenis
1	user	kontrol
2	kader	kontrol
3	vo	kontrol
4	vo1_lanjutan	data
5	vo2_lanjutan	data
6	vo3_lanjutan	data
7	vo_observasi	data
8	vo_kematian	data

9	mining	kontrol
10	Household	kontrol

3.1.6.4.2 Deskripsi *Class Diagram* User

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_user	+	Integer	
nama	+	Varchar	
level	+	Varchar	
password	+	Varchar	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
login	+	void	Fungsi login
validasi	-	void	Fungsi validasi login
Add_user	-	void	Fungsi tambah user
Update_user	-	Id_user	Fungsi update user
Delete_user	-	Id_user	Fungsi delete user

3.1.6.4.3 Deskripsi Class *Diagram* Kader

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_kader	+	Integer	
nama_kader	-	Varchar	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
Add_kader	-	void	Fungsi tambah user
Update_kader	-	Id_user	Fungsi update user
Delete_kader	-	Id_user	Fungsi delete user

3.1.6.4.4 *Class Diagram* VO

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	-	int(11)	
no_vo	+	varchar(10)	
id_household	-	varchar(10)	
tanggal1	-	date	
tanggal2	-	date	
tanggal3	-	date	
tanggal_akhir	-	date	
nama_pewawancara1	-	varchar(50)	
nama_pewawancara2	-	varchar(50)	
nama_pewawancara3	-	varchar(50)	
tanggal_kunj_berikut1	-	date	
waktu_kunj_berikut1	-	time	
tanggal_kunj_berikut2	-	date	
waktu_kunj_berikut2	-	time	
tanggal_kunj_berikut3	-	date	
waktu_kunj_berikut3	-	time	
hasil1	-	int(11)	
hasil2	-	int(11)	
hasil3	-	int(11)	
hasil_akhir	-	int(11)	
hasil_lainnya	-	varchar(100)	
total_kunjungan	-	int(11)	
no_int	-	int(11)	
nama_pengawas	-	varchar(50)	
nama_pemeriksa	-	varchar(50)	
tanggal_pengawas	-	date	
tanggal_pemeriksa	-	date	
nama_tempat	-	varchar(25)	
alamat_jalan	-	varchar(100)	
propinsi	-	varchar(25)	
area	-	varchar(25)	
no_rumah	-	int(11)	
nama_narasumber	-	varchar(50)	

status_tinggal	-	int(11)	
status_tinggallainnya	-	varchar(100)	
main201_jam	-	time	
main202	-	varchar(50)	Nama Responden
main203	-	int(11)	Hubungan dengan almarhum
main203_lainnya	-	varchar(50)	Hubungan lainnya dengan almarhum
main204	-	int(11)	Status tinggal dengan almarhum
main301	-	varchar(50)	Nama Almarhum
main302	-	int(11)	Jenis Kelamin Almarhum
main303_tanggal	-	int(11)	
main303_bulan	-	int(11)	
main303_tahun	-	int(11)	
main304_hari	-	int(11)	
main304_bulan	-	int(11)	
main304_tahun	-	int(11)	
main305	-	varchar(50)	Pekerjaan Almarhum
main306	-	int(11)	Pekerjaan lainnya almarhum
main307	-	int(11)	Status pernikahan almarhum
main308_tanggal	-	int(11)	
main308_bulan	-	int(11)	
main308_tahun	-	int(11)	
main309	-	int(11)	
main309_lainnya	-	varchar(50)	
kategori	-	int(11)	Kategori umur kasus kematian
entry	-	varchar(100)	nama entry data
postdate	-	date	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getRecordmain	+	($id_vo)	Fungsi select table t_vo

checkhousehold	-	($id_household)	Fungsi cek keberadaan id_household pada t_household
checkhouseholdvo	-	($id_household)	Fungsi cek keberadaan id_household pada t_vo
delvo1	-	($id_vo)	Fungsi hapus data otopsi verbal untuk kategori umur 1
delvo2	-	($id_vo)	Fungsi hapus data otopsi verbal untuk kategori umur 2
delvo3	-	($id_vo)	Fungsi hapus data otopsi verbal untuk kategori umur 3
getAllFieldsmain	-	void	Fungsi mendapatkan semua field pada tabel t_vo
insert	-	void	Fungsi insert data pada table t_vo
updatemain	-	($id_vo)	
total	+	void	

3.1.6.4.5 *Class Diagram* Vo1_Lanjutan

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	+	int(11)	
vo401	+	text	
vo402	-	text	
vo403	-	text	
vo502	-	int(11)	
vo502_ket	-	int(11)	
vo503	-	int(11)	
vo503_ket	-	int(11)	

vo504	-	int(11)	
vo505	-	int(11)	
vo505_ket	-	int(11)	
vo506_1	-	int(11)	
vo506_2	-	int(11)	
vo506_3	-	int(11)	
vo506_4	-	int(11)	
vo506_5	-	int(11)	
vo506_lain	-	varchar(50)	
vo507_1	-	int(11)	
vo507_2	-	int(11)	
vo507_3	-	int(11)	
vo507_4	-	int(11)	
vo507_5	-	int(11)	
vo507_6	-	int(11)	
vo507_7	-	int(11)	
vo507_8	-	int(11)	
vo507_9	-	int(11)	
vo507_10	-	int(11)	
vo507_lain	-	varchar(50)	
vo508	-	int(11)	
vo509	-	int(11)	
vo601	-	int(11)	
vo601_lain	-	varchar(50)	
vo602	-	int(11)	
vo602_lain	-	varchar(50)	
vo603	-	int(11)	
vo604	-	int(11)	
vo605	-	int(11)	
vo606	-	int(11)	
vo607	-	int(11)	
vo608	-	int(11)	
vo609	-	int(11)	
vo610	-	int(11)	
vo611	-	int(11)	
vo612	-	int(11)	
vo613	-	int(11)	

vo614_lain	-	varchar(50)	
vo615	-	int(11)	
vo615_lain	-	varchar(50)	
vo616	-	int(11)	
vo701	-	int(10)	
vo702	-	int(10)	
vo703	-	int(10)	
vo703_ket	-	int(10)	
vo704	-	int(10)	
vo704_ket	-	int(10)	
vo705	-	int(10)	
vo706	-	text	
vo707	-	int(10)	
vo708	-	text	
vo709	-	int(10)	
vo710	-	int(10)	
vo711	-	int(10)	
vo711_lain	-	varchar(45)	
vo712	-	int(10)	
vo713	-	int(10)	
vo714	-	int(10)	
vo715	-	int(10)	
vo716	-	int(10)	
vo717	-	int(10)	
vo718	-	int(10)	
vo719	-	int(10)	
vo801	-	int(10)	
vo802	-	int(10)	
vo802_lain	-	varchar(45)	
vo803	-	int(10)	
vo804	-	int(10)	
vo805	-	int(10)	
vo805_lain	-	varchar(45)	
vo901	-	int(10)	
vo902	-	int(10)	
vo902_ket	-	int(10)	
vo903	-	int(10)	

vo904	-	int(10)	
vo904_ket	-	int(10)	
vo905	-	int(10)	
vo906	-	int(10)	
vo907	-	int(10)	
vo907_ket	-	int(10)	
vo908	-	int(10)	
vo909	-	int(10)	
vo910	-	int(10)	
vo910_ket	-	int(10)	
vo911	-	int(10)	
vo912	-	int(10)	
vo912_ket	-	int(10)	
vo913	-	int(10)	
vo914	-	int(10)	
vo914_ket	-	int(10)	
vo915	-	int(10)	
vo916	-	int(10)	
vo916_ket	-	int(10)	
vo917	-	int(10)	
vo918	-	int(10)	
vo918_ket	-	int(10)	
vo919	-	int(10)	
vo920	-	int(10)	
vo920_ket	-	int(10)	
vo921	-	int(10)	
vo922	-	int(10)	
vo922_ket	-	int(10)	
vo923	-	int(10)	
vo924	-	int(10)	
vo925	-	int(10)	
vo926	-	int(10)	
vo927	-	int(10)	
vo927_ket	-	int(10)	
vo928	-	int(10)	
vo928_ket	-	int(10)	
vo929	-	int(10)	

vo930	-	int(10)	
vo931	-	int(10)	
vo931_ket	-	int(10)	
vo932	-	int(10)	
vo932_ket	-	int(10)	
vo933	-	int(10)	
vo934	-	int(10)	
vo934_ket	-	int(10)	
vo935	-	int(10)	
vo936	-	int(10)	
vo937	-	int(10)	
vo938	-	int(10)	
vo938_ket	-	int(10)	
vo939	-	int(10)	
vo939_ket	-	int(10)	
vo1001	-	int(10)	
vo1001_ket	-	int(10)	
vo1002	-	int(10)	
vo1003	-	int(10)	
vo1004	-	int(10)	
vo1004_ket	-	int(10)	
vo1005	-	int(10)	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getAllRecordsVo1	-	void	Fungsi select table t_vo11
getAllRecordsVo2	-	void	Fungsi select table t_vo12
getAllFields1	-	void	Fungsi mendapatkan semua field pada tabel t_vo11
getAllFields2	-	void	Fungsi mendapatkan semua field pada tabel t_vo12
insert1	-	void	Fungsi insert data pada table t_vo11
insert2	-	void	Fungsi insert data pada table t_vo12
update1	-	($id_vo)	Fungsi update tabel

			t_vo11
update2	-	($id_vo)	Fungsi update tabel t_vo12

3.1.6.4.6 *Class Diagram* Vo2_Lanjutan

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	-	int(11)	
vo401	+	text	
vo402	-	varchar(100)	
vo403	-	varchar(100)	
vo502	-	int(11)	
vo503	-	int(11)	
vo504	-	int(11)	
vo505	-	int(11)	
vo506	-	int(11)	
vo507	-	int(11)	
vo508	-	varchar(100)	
vo509	-	int(11)	
vo510	-	int(11)	
vo511	-	int(11)	
vo512	-	varchar(100)	
vo601	-	int(11)	
vo602	-	int(11)	
vo602_lainnya	-	varchar(25)	
vo603	-	int(11)	
vo604	-	int(11)	
vo605	-	int(11)	
vo606	-	int(11)	
vo607	-	int(11)	
vo607_lainnya	-	varchar(25)	
vo608	-	int(11)	
kategori	-	int(11)	
vo701	-	int(11)	
vo702	-	int(11)	

vo703	-	int(11)	
vo703_ket	-	int(11)	
vo704	-	int(11)	
vo705	-	int(11)	
vo706	-	int(11)	
vo706_ket	-	int(11)	
vo801	-	int(11)	
vo802	-	int(11)	
vo802_ket	-	int(11)	
vo803	-	int(11)	
vo804	-	int(11)	
vo804_ket	-	int(11)	
vo805	-	int(11)	
vo806	-	int(11)	
vo807	-	int(11)	
vo808	-	int(11)	
vo809	-	int(11)	
vo809_ket	-	int(11)	
vo810	-	int(11)	
vo811	-	int(11)	
vo812	-	int(11)	
vo813	-	int(11)	
vo813_ket	-	int(11)	
vo814	-	int(11)	
vo815	-	int(11)	
vo815_ket	-	int(11)	
vo816	-	int(11)	
vo817	-	int(11)	
vo817_ket	-	int(11)	
vo818	-	int(11)	
vo819	-	int(11)	
vo820	-	int(11)	
vo821	-	int(11)	
vo821_ket	-	int(11)	
vo822	-	int(11)	
vo822_ket	-	int(11)	
vo823	-	int(11)	

vo824	-	int(11)	
vo825	-	int(11)	
vo825_ket	-	int(11)	
vo826	-	int(11)	
vo826_ket	-	int(11)	
vo827	-	int(11)	
vo828	-	int(11)	
vo828_ket	-	int(11)	
vo829	-	int(11)	
vo830	-	int(11)	
vo831	-	int(11)	
vo831_ket	-	int(11)	
vo832	-	int(11)	
vo833	-	int(11)	
vo834	-	int(11)	
vo835	-	int(11)	
vo835_ket	-	int(11)	
vo836	-	int(11)	
vo837	-	int(11)	
vo837_ket	-	int(11)	
vo838	-	int(11)	
vo839	-	int(11)	
vo840	-	int(11)	
vo840_ket	-	int(11)	
vo841	-	int(11)	
vo842	-	int(11)	
vo842_ket	-	int(11)	
vo843	-	int(11)	
vo844	-	int(11)	
vo845	-	int(11)	
vo845_ket	-	int(11)	
vo846	-	int(11)	
vo847	-	int(11)	
vo847_ket	-	int(11)	
vo848	-	int(11)	
vo849	-	int(11)	
vo850	-	int(11)	

vo850_ket	-	int(11)	
vo851	-	int(11)	
vo852	-	int(11)	
vo853	-	int(11)	
vo853_ket	-	int(11)	
vo854_1	-	int(11)	
vo854_2	-	int(11)	
vo854_3	-	int(11)	
vo854_4	-	int(11)	
vo854_lain	-	varchar(45)	
vo855	-	int(11)	
vo855_lain	-	varchar(45)	
vo856	-	int(11)	
vo857	-	int(11)	
vo858	-	int(11)	
vo859	-	int(11)	
vo859_ket	-	int(11)	
vo860	-	int(11)	
vo861	-	int(11)	
vo862	-	int(11)	
vo862_ket	-	int(11)	
vo863	-	int(11)	
vo864	-	int(11)	
vo864_ket	-	int(11)	
vo865_1	-	int(11)	
vo865_2	-	int(11)	
vo865_3	-	int(11)	
vo865_4	-	int(11)	
vo865_5	-	int(11)	
vo865_lain	-	varchar(45)	
vo866	-	int(11)	
vo867	-	int(11)	
vo867_ket	-	int(11)	
vo868_1	-	int(11)	
vo868_2	-	int(11)	
vo868_3	-	int(11)	
vo868_4	-	int(11)	

vo868_lain	-	varchar(45)	
vo869	-	int(11)	
vo869_ket	-	varchar(45)	
vo870	-	int(11)	
vo870_ket	-	int(11)	
vo871	-	int(11)	
vo872	-	int(11)	
vo872_ket	-	int(11)	
vo873	-	int(11)	
vo874	-	int(11)	
vo874_ket	-	int(11)	
vo875	-	int(11)	
vo876	-	int(11)	
vo876_ket	-	int(11)	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getAllRecordsVo1	-	void	Fungsi select table t_vo21
getAllRecordsVo2	-	void	Fungsi select table t_vo22
getAllFields1	-	void	Fungsi mendapatkan semua field pada tabel t_vo21
getAllFields2	-	void	Fungsi mendapatkan semua field pada tabel t_vo22
insert1	-	void	Fungsi insert data pada table t_vo21
insert2	-	void	Fungsi insert data pada table t_vo22
update1	-	($id_vo)	Fungsi update tabel t_vo21
update2	-	($id_vo)	Fungsi update tabel t_vo22

3.1.6.4.7 *Class Diagram* Vo3_Lanjutan

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	+	int(11)	
vo401	-	text	
vo402	-	varchar(100)	

vo403	-	varchar(100)	
vo502	-	int(11)	
vo503	-	int(11)	
vo504	-	int(11)	
vo505	-	int(11)	
vo506	-	int(11)	
vo507	-	int(11)	
vo508	-	varchar(100)	
vo509	-	int(11)	
vo510	-	int(11)	
vo511	-	int(11)	
vo512	-	varchar(100)	
vo601	-	int(11)	
vo602	-	int(11)	
vo602_lainnya	-	varchar(25)	
vo603	-	int(11)	
vo604	-	int(11)	
vo605	-	int(11)	
vo606	-	int(11)	
vo606_lainnya	-	varchar(25)	
vo607	-	int(11)	
vo701	-	int(11)	
vo702	-	int(11)	
vo702_ket	-	int(11)	
vo703	-	int(11)	
vo704	-	int(11)	
vo704_ket	-	int(11)	
vo705	-	int(11)	
vo706	-	int(11)	
vo706_ket	-	int(11)	
vo707	-	int(11)	
vo708	-	int(11)	
vo708_ket	-	int(11)	
vo801	-	int(11)	
vo802	-	int(11)	
vo802_ket	-	int(11)	
vo803	-	int(11)	

vo803_ket	-	int(11)	
vo8041	-	int(11)	
vo8042	-	int(11)	
vo8043	-	int(11)	
vo8044	-	int(11)	
vo8045	-	int(11)	
vo8046	-	int(11)	
vo8047	-	int(11)	
vo8048	-	int(11)	
vo8049	-	int(11)	
vo804_lainnya	-	varchar(50)	
vo805	-	int(11)	
vo806	-	int(11)	
vo807	-	int(11)	
vo807_ket	-	int(11)	
vo808	-	int(11)	
vo809	-	int(11)	
vo810	-	int(11)	
vo811	-	int(11)	
vo812	-	int(11)	
vo813	-	int(11)	
vo814	-	int(11)	
vo814_lainnya	-	varchar(50)	
vo815	-	int(11)	
vo816	-	int(11)	
vo816_lainnya	-	varchar(50)	
vo817	-	int(11)	
vo817_lainnya	-	varchar(50)	
vo818	-	int(11)	
vo819	-	int(11)	
vo820	-	int(11)	
vo820_ket	-	int(11)	
vo821	-	int(11)	
vo821_ket	-	int(11)	
vo822	-	int(11)	
vo823	-	int(11)	
vo824	-	int(11)	

vo901	-	int(11)	
vo901_ket	-	int(11)	
vo902	-	int(11)	
vo903	-	int(11)	
vo903_ket	-	int(11)	
vo904	-	int(11)	
vo905	-	int(11)	
vo906	-	int(11)	
vo907	-	int(11)	
vo908	-	int(11)	
vo908_ket	-	int(11)	
vo909	-	int(11)	
vo910	-	int(11)	
vo911	-	int(11)	
vo912	-	int(11)	
vo913	-	int(11)	
vo914	-	int(11)	
vo914_ket	-	int(11)	
vo915	-	int(11)	
vo916	-	int(11)	
vo917	-	int(11)	
vo918	-	int(11)	
vo919	-	int(11)	
vo919_ket	-	int(11)	
vo920	-	int(11)	
vo921	-	int(11)	
vo922	-	int(11)	
vo923	-	int(11)	
vo924	-	int(11)	
vo925	-	int(11)	
vo926	-	int(11)	
vo927	-	int(11)	
vo928	-	int(11)	
vo929	-	int(11)	
vo929_ket	-	int(11)	
vo930	-	int(11)	
vo931	-	int(11)	

vo932	-	int(11)	
vo932_ket	-	int(11)	
vo933	-	int(11)	
vo934	-	int(11)	
vo934_ket	-	int(11)	
vo935	-	int(11)	
vo935_lainnya	-	varchar(100)	
vo936	-	int(11)	
vo936_ket	-	int(11)	
vo937	-	int(11)	
vo938	-	int(11)	
vo939	-	int(11)	
vo940	-	int(11)	
vo940_ket	-	int(11)	
vo941	-	int(11)	
vo942	-	int(11)	
vo942_ket	-	int(11)	
vo943	-	int(11)	
vo944	-	int(11)	
vo945	-	int(11)	
vo946	-	int(11)	
vo946_ket	-	int(11)	
vo947	-	int(11)	
vo948	-	int(11)	
vo949	-	int(11)	
vo949_ket	-	int(11)	
vo950	-	int(11)	
vo951	-	int(11)	
vo951_ket	-	int(11)	
vo952	-	int(11)	
vo953	-	int(11)	
vo953_ket	-	int(11)	
vo954	-	int(11)	
vo955	-	int(11)	
vo956	-	int(11)	
vo956_ket	-	int(11)	
vo957	-	int(11)	

vo958	-	int(11)	
vo958_ket	-	int(11)	
vo959	+	int(11)	
vo960	-	int(11)	
vo961	-	int(11)	
vo961_ket	-	int(11)	
vo962	-	int(11)	
vo963	-	int(11)	
vo964	-	int(11)	
vo964_ket	-	int(11)	
vo965	-	int(11)	
vo966	-	int(11)	
vo966_ket	-	int(11)	
vo967	-	int(11)	
vo968	-	int(11)	
vo968_ket	-	int(11)	
vo969	-	int(11)	
vo970	-	int(11)	
vo970_ket	-	int(11)	
vo971	-	int(11)	
vo972	-	int(11)	
vo973	-	int(11)	
vo973_ket	-	int(11)	
vo974	-	int(11)	
vo975	-	int(11)	
vo975_ket	-	varchar(100)	
vo976	-	int(11)	
vo976_ket	-	int(11)	
vo977	-	int(11)	
vo978	-	int(11)	
vo978_ket	-	int(11)	
vo979	-	int(11)	
vo980	-	int(11)	
vo980_ket	-	int(11)	
vo981	-	int(11)	
vo982	-	int(11)	
vo983	-	int(11)	

vo983_ket	-	int(11)	
vo984_wajah	-	int(11)	
vo984_badan	-	int(11)	
vo984_lengan	-	int(11)	
vo984_lain	-	int(11)	
vo984_sebutkan	-	varchar(100)	
vo985	-	int(11)	
vo985_lain	-	varchar(100)	
vo986	-	int(11)	
vo987	-	int(11)	
vo988	-	int(11)	
vo989	-	int(11)	
vo989_1	-	int(11)	
vo989_1_ket	-	int(11)	
vo989_2	-	int(11)	
vo990	-	int(11)	
vo990_1	-	int(11)	
vo990_1_ket	-	int(11)	
vo991	-	int(11)	
vo991_1	-	int(11)	
vo991_1_ket	-	int(11)	
vo991_2_wajah	-	int(11)	
vo991_2_sendi	-	int(11)	
vo991_2_tumit	-	int(11)	
vo991_2_badan	-	int(11)	
vo991_2_lain	-	int(11)	
vo991_2_sebutkan	-	varchar(100)	
vo992	-	int(11)	
vo992_1	-	int(11)	
vo992_1_ket	-	int(11)	
vo992_2_leher	-	int(11)	
vo992_2_ketiak	-	int(11)	
vo992_2_lipat	-	int(11)	
vo992_2_lain	-	int(11)	
vo992_2_sebutkan	-	varchar(100)	
vo993	-	int(11)	
vo993_ket	-	varchar(100)	

vo993_1	-	int(11)	
vo993_1_ket	-	int(11)	
vo994	-	int(11)	
vo994_1	-	int(11)	
vo994_1_ket	-	int(11)	
vo995	-	int(11)	
vo995_1	-	int(11)	
vo995_1_ket	-	int(11)	
vo995_2_jelaskan	-	varchar(100)	
vo1101	-	int(11)	
vo1102	-	int(11)	
vo1102_ket	-	int(11)	
vo1103	-	int(11)	
vo1104	-	int(11)	
vo1105	-	int(11)	
vo1105_ket	-	int(11)	
vo1106	-	int(11)	
vo1107	-	int(11)	
vo1107_ket	-	int(11)	
vo1108	-	int(11)	
vo1109	-	int(11)	
vo1109_ket	-	int(11)	
vo1110	-	int(11)	
vo1111	-	int(11)	
vo1111_ket	-	int(11)	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getAllRecordsVo1	-	void	Fungsi select table t_vo31
getAllRecordsVo2	-	void	Fungsi select table t_vo32
getAllRecordsVo2	-	void	Fungsi select table t_vo33
getAllFields1	-	void	Fungsi mendapatkan semua field pada tabel t_vo31
getAllFields2	-	void	Fungsi mendapatkan semua field pada tabel t_vo32
getAllFields2	-	void	Fungsi mendapatkan semua field pada tabel t_vo33
insert1	-	void	Fungsi insert data pada table t_vo31

insert2	-	void	Fungsi insert data pada table t_vo32
insert3	-	void	Fungsi insert data pada table t_vo33
update1	-	($id_vo)	Fungsi update tabel t_vo31
update2	-	($id_vo)	Fungsi update tabel t_vo32
update3	-	($id_vo)	Fungsi update tabel t_vo33

3.1.6.4.8 *Class Diagram* Vo_Kematian

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	+	int(11)	
vaksin	-	int(11)	
per1	-	int(11)	
per2	-	varchar(255)	
per3_rehidrasi	-	int(11)	
per3_tranfusi	-	int(11)	
per3_ngt	-	int(11)	
per3_lain	-	int(11)	
per3_sebutkan	-	varchar(100)	
per4_rumah	-	int(11)	
per4_tradisional	-	int(11)	
per4_puskesmas	-	int(11)	
per4_rs	-	int(11)	
per4_klinik	-	int(11)	
per4_swasta	-	int(11)	
per4_apotek	-	int(11)	
per4_lain	-	int(11)	
per4_lainnya	-	varchar(100)	
per5	-	int(11)	
per5_ket	-	int(11)	
per6	-	int(11)	
per7	-	text	
per8	-	int(11)	
per9	-	int(11)	
per9_ket	-	int(11)	

per10	-	int(11)	
per10_sebutkan	-	varchar(100)	
sur1	-	int(11)	
sur2_hari	-	int(11)	
sur2_bulan	-	int(11)	
sur2_tahun	-	int(11)	
sur3_hari	-	int(11)	
sur3_bulan	-	int(11)	
sur3_tahun	-	int(11)	
sur4	-	text	
sur5	-	text	
sur6	-	text	
sur7	-	text	
cat1	-	int(11)	
cat3	-	text	
cat4	-	text	
cat5	-	text	
cat6	-	text	
cat7	-	text	
cat8	-	text	
cat9	-	text	
cat10	-	text	
cat11	-	time	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getAllRecord	-	void	Fungsi select table t_vo_kematian
getAllFields1	-	void	Fungsi mendapatkan semua field pada tabel t_vo_kematian
insert1	-	void	Fungsi insert data pada table t_vo_kematian
update1	-	($id_vo)	Fungsi update tabel t_vo_kematian

3.1.6.4.9 *Class Diagram* Vo_Observasi

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	+	int(11)	
observasi	-	text	
komentar_khusus	-	text	
komentar_lain	-	text	
pengamatan	-	text	
nama	-	varchar(25)	
tanggal	-	date	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**
getAllRecord	-	void	Fungsi select table t_vo_observasi
getAllFields	-	void	Fungsi mendapatkan semua field pada tabel t_vo_observasi
insert	-	void	Fungsi insert data pada table t_vo_observasi
update	-	($id_vo)	Fungsi update tabel t_vo_observasi

3.1.6.4.10 *Class Diagram* Mining

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_vo	+	int(11)	
id_symp	+	int(11)	
id_user	+	int(11)	
symptoms	-	varchar(50)	
keterangan	-	varchar(100)	
kode_va	-	varchar(8)	
tema_va	-	varchar(100)	
id_symp	-	int(11)	
id_va	-	int(11)	
Methode			
Nama	**Visibility**	**Type**	**Keterangan**

getAllRecordsMining	-	void	Fungsi select table t_mining
getAllFieldsMining	-	void	Fungsi mendapatkan semua field pada tabel mining
geDelete	-	($iduser)	Hapus isi tabel t_kasus untuk id user yang sedang login
getSelect	-	void	Fungsi pencarian symptoms dengan filtering
getInsert	-	void	Isi tabel t_kasus dengan data uptodate
getAlgo	-	void	Fungsi yang menjalankan algoritma

3.1.6.4.11 *Class Diagram* Household

Atribut			
Nama	**Visibility**	**Type Data**	**Keterangan**
id_household	+	int(11)	
kode_house	+	varchar(10)	
nma_responde	-	varchar(50)	
hub_keluarga	-	varchar(100)	
kerja_household	-	varchar(25)	
pend_household	-	varchar(15)	
pendtingkathh	-	varchar(15)	
unggas_skt_mati	-	varchar(20)	
vaksinasi1	-	int(11)	
vaksinasi2	-	int(11)	
vaksinasi3	-	int(11)	
vgroup_vnd	-	varchar(5)	
id_district	-	int(4)	
tgl_survey	-	date	
geocoding	-	varchar(100)	
pola11	-	varchar(100)	
pola12	-	varchar(100)	
pola13	-	varchar(100)	
pola14	-	varchar(100)	
pola15	-	varchar(100)	
pola21	-	varchar(100)	

pola22	-	varchar(100)	
pola23	-	varchar(100)	
pola24	-	varchar(100)	
pola25	-	varchar(100)	
pola31	-	varchar(100)	
pola32	-	varchar(100)	
pola33	-	varchar(100)	
pola34	-	varchar(100)	
pola35	-	varchar(100)	
ditemukan	-	varchar(11)	
pindah	-	varchar(11)	
pindah_ya	-	text	
huni_ada	-	varchar(11)	
tolak_wwcr	-	varchar(11)	
posting_date	-	date	
posting_day	-	varchar(20)	
posting_hour	-	time	
entry	-	varchar(50)	
kader	-	varchar(50)	
valid	-	varchar(10)	
cetak	-	int(11)	
edit_date	-	date	
edit_day	-	varchar(10)	
edit_hour	-	time	
editor	-	varchar(50)	

3.1.7 Analisis Kebutuhan Perangkat Keras

Pada dasarnya kebutuhan perangkat keras disini tidak ada hal spesifik seperti komputer biasa pada umumnya, hanya karena data yang dikelola system ini termasuk banyak, maka dibutuhkan sebuah server dengan *spec* yang memadai. Dalam prakteknya, berikut spesifikasi server yang ada dan digunakan saat ini oleh sistem:

Processor type	*Intel® Xeon® X3330 Quad Core Processor 2.66 GHz*
Processor speed	*2.66 GHz*
Number of processors	*1 processor*
Processor core available	*Quad*
Processor front side bus	*1333 MHz Front Side Bus*
Memory	*4 GB (2 x 2 GB)*
Internal cache	*6 MB (2 x 3 MB) Level 2 cache*
Memory type	*PC2-6400 unbuffered DDR2 ECC 800MHz*
Maximum memory	*8 GB*
Memory slots	*4 DIMM slots*
Internal drives	
Internal drives	*250 GB SATA Non Hot Plug 3.5" Hard Drive*
Hard disk drive speed	*7,200 rpm*
Storage controller	*HP Embedded 6 Port SATA Controller with Embedded RAID (4 ports for HDD)*
Internal drive bays	*Four 3.5" Non Hot Plug SATA*
Optical drives	*16x SATA DVD-ROM*

3.1.8 Analisis Kebutuhan Perangkat Lunak

Dalam analisis kebutuhan perangkat lunak, dibutuhkan beberapa software yang mendukung untuk digunakan dalam implementasi. Implementasi pembuatan aplikasi ini menggunakan software dan bahasa pemograman berikut :

1. Untuk bahasa pemograman :

    1.1. HTML

1.2. PHP

1.3. JavaScript

1.4. CSS

1.5. SQL

2. Untuk RDBMS

2.1. MySQL

3. Untuk *development tools*

3.1. Macromedia Dreamweaver

3.2. Notepad++

4. Untuk *web server*

4.1. Apache

3.1.9 Analisis Kebutuhan Pengguna

Pengguna dalam sistem ini dibagi menjadi dua level yaitu level admin dan level entri. Dimana admin disini yang dimaksud adalah dokter penelitian yang bertugas sebagai *data manager* untuk mengidentifikasi data hasil penelitian yang sudah di-*input*-kan oleh pengguna entri. Dan pengguna entri yang memasukkan data ke dalam system sesuai form manual *hardcopy* yang didapat dari dokter lapangan dan kader.

Tabel 3.18 Kebutuhan Pengguna

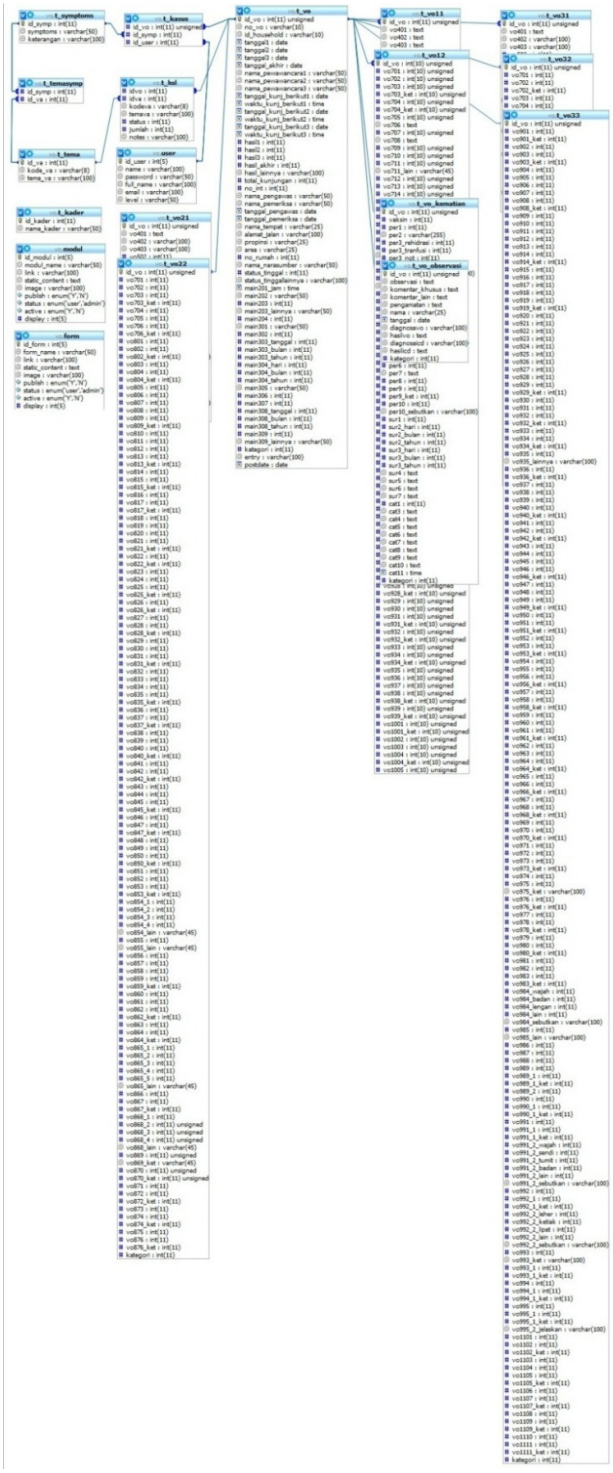
No	Pengguna	Deskripsi
1	*Data Manager*	Dokter penelitian yang berkepentingan untuk melakukan laporan diagnosa identifikasi penyebab kematian.
2	*Data Entry*	Beberapa staf yang bertugas memasukkan data otopsi verbal kedalam sistem.

3.2 Perancangan Sistem

Perancangan sistem dibuat berdasarkan hasil analisa yang telah dilakukan sebelumnya. Tahap perancangan merupakan tahap yang menentukan hasil akhir dari sebuah sistem.

3.2.1 Perancangan Basis Data

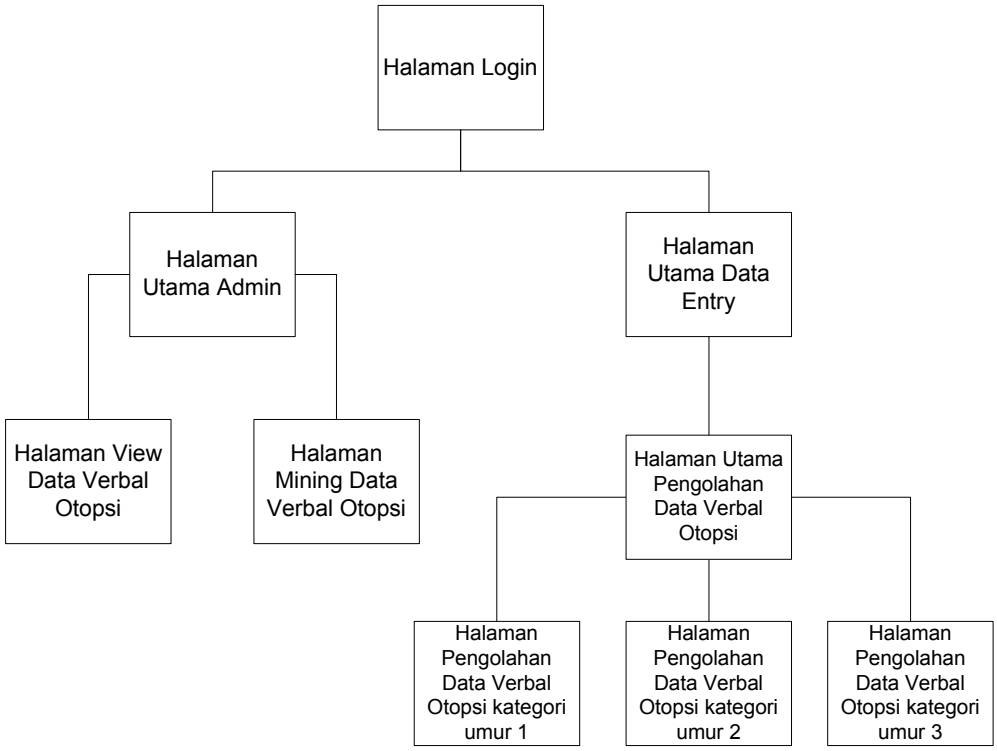
Berdasarkan perancangan *class diagram,* berikut usulan skema relasi dari aplikasi identifikasi penyebab kematian pada penelitian CIRAI UPK - FKUnpad. Pembuatan tabel entitas selalu berpedoman pada ketentuan-ketentuan tentang cara penyusunan tabel. Sehingga akan diperoleh susunan tabel entitas yang merupakan tabel yang terbebas dari adanya data rangkap (data redudancy). Untuk mencapai tujuan tersebut, dalam penyusunan skema relasi harus memperhatikan hubungan antar entitas dengan kardinalitas dan partisipasi hubungan. Dan berikut skema relasi dari aplikasi identifikasi penyebab kematian pada penelitian CIRAI UPK – FK Unpad:



Gambar 3.37 Skema relasi otopsi verbal

3.2.2 Perancangan Arsitektur menu

Arsitektur menu disini dibagi kedalam beberapa halaman yang bertujuan untuk mempermudah pemahaman dan pengoperasian aplikasi Identifikasi Penyebab Kematian pada penelitian CIRAI di Unit Penelitian Kesehatan – FK Unpad. Halaman-halaman yang digunakan dapat dilihat pada struktur halaman di bawah ini :

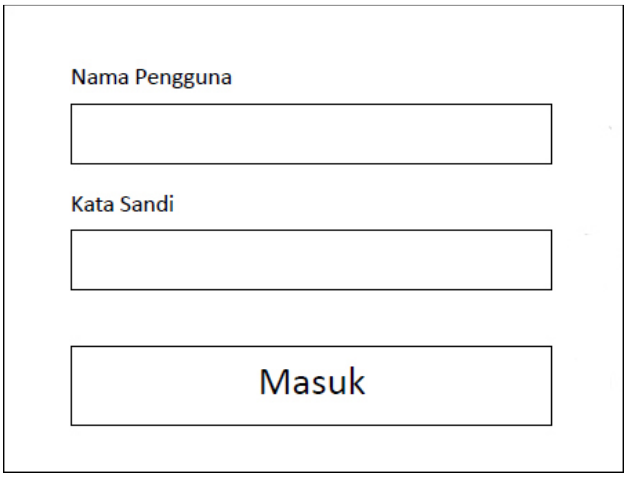


Gambar 3.38 Struktur menu aplikasi Identifikasi penyebab Kematian

3.2.3 Perancangan Antarmuka

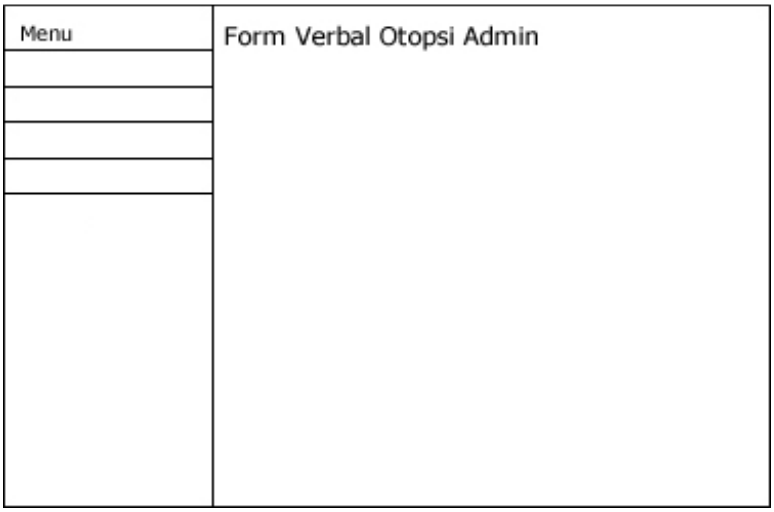
Dalam membuat aplikasi dibutuhkan tampilan yang sangat menarik, dimana dalam perancangan antar muka ini akan dituangkan beberapa imajinasi dalam merancang tampilan program dan tampilan pesan yang nantinya akan dipakai untuk membuat aplikasi sesungguhnya. Perancangan antar muka ini terdiri dari perancangan tampilan program, sedangkan untuk mengantisipasi kesalahan dalam aplikasi ini penulis menggunakan pencegahan langsung dengan *javascript* untuk setiap kondisi inputan yang boleh diinputkan atau tidak.

3.2.3.1 Perancangan Antarmuka Halaman Login



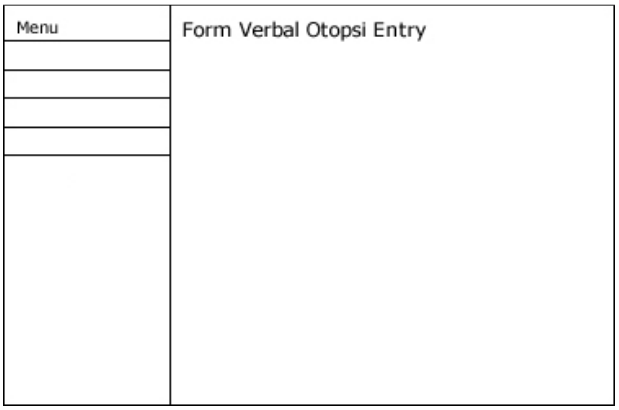
Gambar 3.39 Rancangan Antarmuka Halaman Login Aplikasi Identifikasi Penyebab Kematian

3.2.3.2 Perancangan Antarmuka Halaman Utama Admin



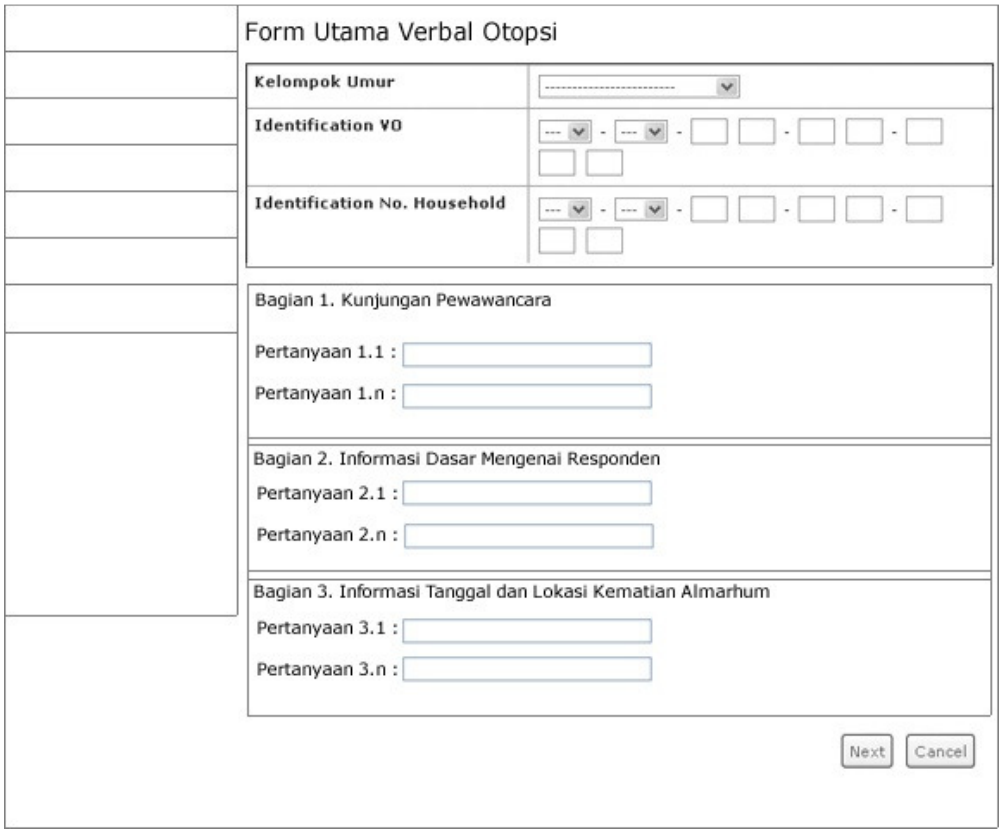
Gambar 3.40 Rancangan Antarmuka Halaman Utama Admin

3.2.3.3 Perancangan Antarmuka Halaman Utama *Data Entry*



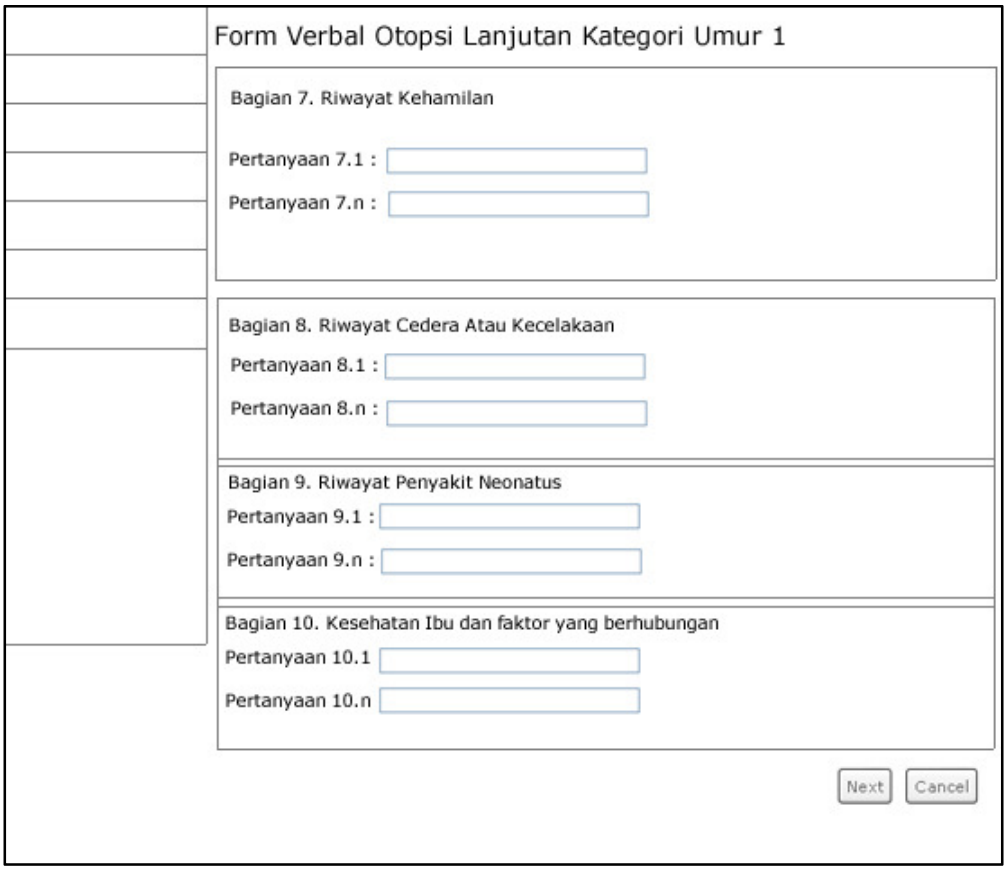
Gambar 3.41 Rancangan Antarmuka Halaman Utama *Data Entry*

3.2.3.4 Perancangan Antarmuka Halaman Pengolahan Data Otopsi verbal



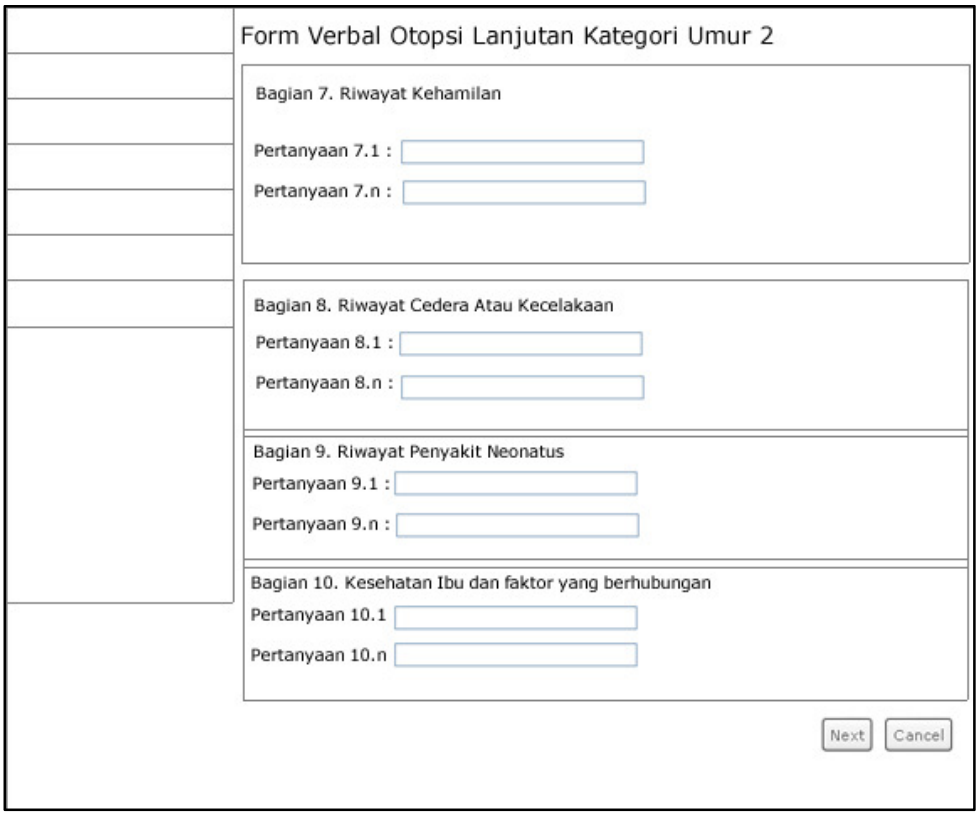
Gambar 3.42 Rancangan Antarmuka Halaman Utama Pengolahan Data Otopsi verbal

3.2.3.5 Perancangan Antarmuka Halaman Pengolahan Data Otopsi verbal Kategori Umur 1



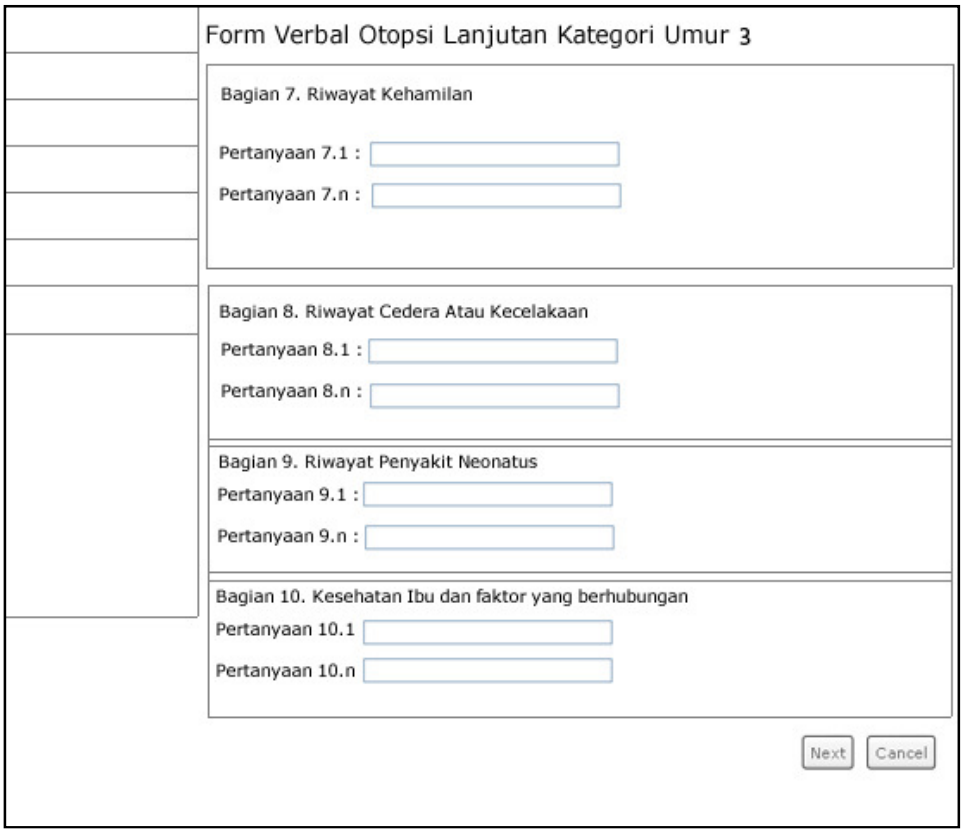
Gambar 3.43 Rancangan Antarmuka Halaman Utama Pengolahan Data Otopsi verbal Lanjutan kategori Umur 1

3.2.3.6 Perancangan Antarmuka Halaman Pengolahan Data Otopsi verbal Kategori Umur 2



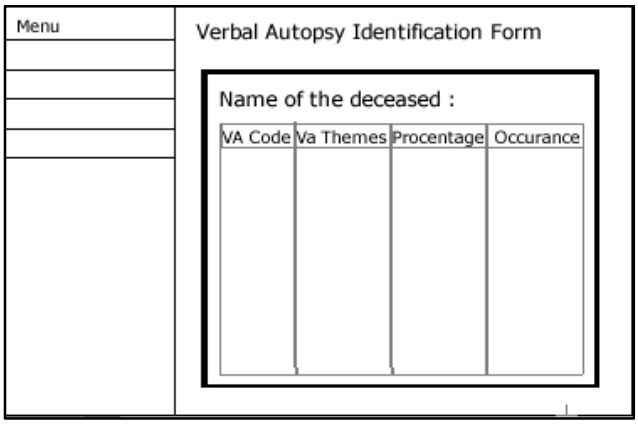
Gambar 3.44 Rancangan Antarmuka Halaman Utama Pengolahan Data Otopsi verbal Lanjutan kategori Umur 2

3.2.3.7 Perancangan Antarmuka Halaman Pengolahan Data Otopsi verbal Kategori Umur 3



Gambar 3.45 Rancangan Antarmuka Halaman Utama Pengolahan Data Otopsi verbal Lanjutan kategori Umur 3

3.2.3.8 Perancangan Antarmuka Halaman Informasi Identifikasi Penyebab Kematian



Gambar 3.46 Rancangan Antarmuka Halaman Informasi Identifikasi penyebab kematian

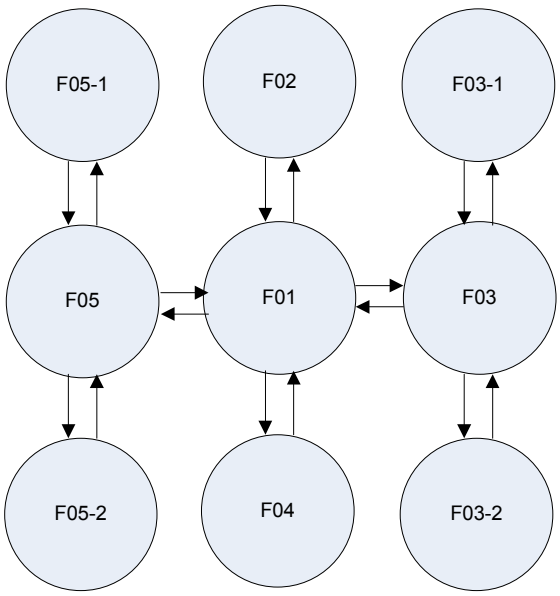
3.2.4 Perancangan Antarmuka Pesan Kesalahan



Gambar 3.47 Rancangan Antarmuka Pesan Kesalahan

3.2.5 Jaringan Semantik

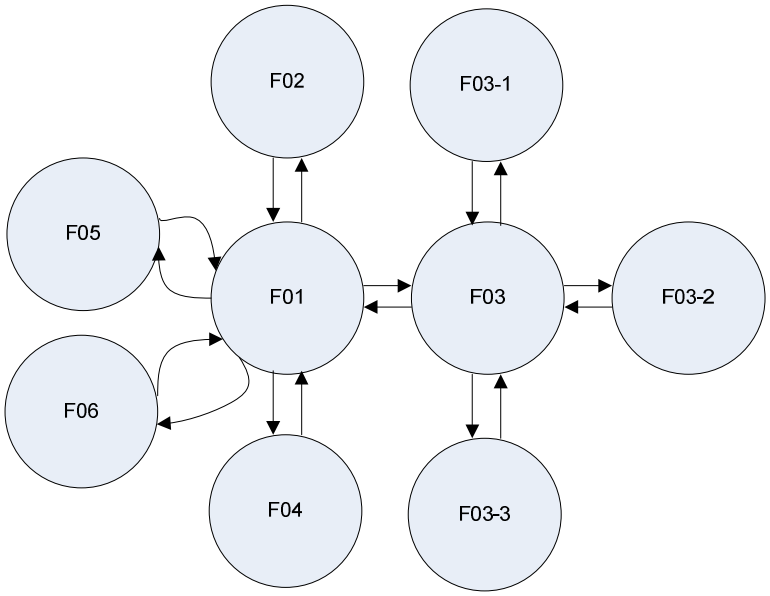
Jaringan semantik adalah alat yang efektif untuk merepresentasikan pemetaan data, yang bertujuan mencegah terjadinya duplikasi data. Berikut adalah uraian mengenai jaringan semantik pada sistem ini.



Gambar 3.48 Jaringan Semantik Entry

Keterangan :

F01 Halaman Login
F02 Halaman Menu Utama Data Entry
F03 Halaman Menu Utama Otopsi Verbal
F03-1 Halaman Menu Input Data Otopsi Verbal
F03-2 Halaman Menu Edit Data Otopsi Verbal
F04 Halaman Ubah Profil Data Entry
F05-1 Halaman Menu Input Data Household
F05-2 Halaman Menu Edit Data Household



Gambar 3.49 Jaringan Semantik Data Manager/Admin

Keterangan :

F01 Halaman Login
F02 Halaman Menu Utama Data Manager
F03 Halaman Menu Utama Otopsi Verbal
F03-1 Halaman Menu Identify Laporan Otopsi Verbal
F03-2 Halaman Menu View Laporan Otopsi Verbal
F03-3 Halaman Menu Print Laporan Otopsi Verbal
F04 Halaman Ubah Profil Data Manager
F05 Halaman Form Kader
F06 Halaman Form User