BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Terkadang seorang penderita vertigo itu sendiri pada awalnya tidak mengetahui jenis gejala maupun penyakit yang dideritanya karena minimnya informasi yang mereka ketahui. Bila seorang penderita vertigo ingin mengetahui tentang gejala-gejala, penyebab serta cara penanggulangan yang baik maka mereka akan mendatangi dokter untuk berkonsultasi. Akan tetapi, hal tersebut tidak dapat dilakukan oleh semua orang mungkin karena faktor perekonomian yang kurang mencukupi ataupun karena tuntutan kesibukan dan aktifitas mereka yang padat sehingga tidak bisa menyesuaikan dengan jadwal praktek dokter. Terkadang ketika berkonsultasi pun informasi hasil diagnosa yang disampaikan oleh dokter tidak sepenuhnya dipaparkan secara mendetail.

Berdasarkan hal tersebut, maka diperlukan suatu alat bantu yang mampu merepresentasikan keahlian seorang pakar (dokter spesialis) dan dapat diakses dimanapun dan kapanpun.

3.2 Analisis Sistem Pakar

Dalam membangun sistem pakar dilakukan beberapa tahapan analisis :

1. Informasi menentukan masalah yang akan dibangun sistem pakarnya.
2. Mengumpulkan data yang diperlukan untuk membangun sistem berupa jenis-jenis penyakit, gejala-gejala penyakit, pengertian penyakit, dan saran terapi penanggulangan penyakit melalui studi literatur, penelitian, dan

wawancara kepada pihak yang bersangkutan, dalam hal ini dokter spesialis yang akan digunakan sebagai knowledge base.

3. Merepresentasikan pengetahuan yang didapat.
4. Menentukan metode inferensi yang akan digunakan.
5. Menentukan target *user* yang akan menggunakan sistem pakar ini
6. Usulan sistem pakar yang telah dibangun.

3.3 Analisis Kebutuhan Data

Data yang diperoleh ialah data mengenai pengertian vertigo, jenis-jenis vertigo, gejala-gejala vertigo, dan terapi penangulangan vertigo. Data tersebut didapat dari beberapa sumber, seperti buku-buku mengenai vertigo, internet, dan dari seorang dokter spesialis saraf yang telah memberikan bimbingan dan pengarahan serta informasi yang dibutuhkan selengkapnya dalam proses pembuatan sistem pakar untuk mendiagnosa penyakit vertigo ini. Adapun dokter yang menjadi sumber dalam sistem pakar ini adalah dr. Zaenal Arifin, Sp. S.

Adapun jenis vertigo yang didiagnosa adalah sebagai berikut :

1. Vertigo Periferal

   Vertigo perifer (*peripheral vertigo*) disebabkan oleh disfungsi struktur perifer hingga ke batang otak (*brain stem*). Vertigo periferal terjadi jika terdapat gangguan disaluran yang disebut *kanalis semisirkularis*, yaitu telinga bagian tengah yang bertugas mengontrol keseimbangan.

   Gejala :

   1) Mual.
   2) Muntah.
   3) Keringat dingin.

4) Pandangan gelap.
5) Lelah dan stamina menurun.
6) Memori dan daya pikir menurun.
7) Tidak mampu berkonsentrasi.
8) Pusing.
9) Telinga Berdenging (*tinnitus*).
10) Posisi kepala terfiksasi kearah tertentu.
11) Sensitif pada cahaya terang dan suara.
12) Rasa seolah-olah akan terjatuh pada permulaan tidur.
13) Hilang keseimbangan, seperti tidak bisa tegak dan tidak bisa stabil ketika berdiri dan atau duduk jalan sempoyongan dan jalan membelok.
14) Ketika berbaring miring atau ketika kepalanya menengadah ke atas keadaan sekitar terasa berputar selama 15-20 detik.
15) Pandangan tidak fokus (gerakan mata ulang-alik diluar kemauan (*nistagimus* atau *nystagmoid jerks*)).

Terapi :

1) Terapi Kausal
2) Terapi Simtomatik
3) Terapi Rehabilitatif, yaitu latihan *Vestibular* dengan menggunakan metode Brandt-Daroff.

2. Vertigo Sentral

Vertigo sentral (*central vertigo*) terjadi jika ada sesuatu yang tidak normal di dalam otak, khususnya di bagian saraf keseimbangan, yaitu daerah percabangan otak dan *serebelum* (otak kecil).

Gejala :

1) Mual.
2) Muntah.
3) Keringat dingin.
4) Pandangan gelap.
5) Lelah dan stamina menurun.
6) Memori dan daya pikir menurun.
7) Tidak mampu berkonsentrasi.
8) Pusing.
9) Sulit berdiri.
10) Lemas (contoh : tidak mampu menggenggam erat benda dengan telapak tangan)
11) Keadaan sekitar terasa berputar, benda diam di sekitar nampak bergoyang (*Oscillpsia*) dan nampak ganda (*Diplopia*).
12) Sukar menelan
13) Kelumpuhan otot-otot wajah
14) Sakit kepala yang parah
15) Kesadaran terganggu
16) Hilangnya koordinasi

Terapi :

1) Terapi Kausal
2) Terapi Simtomatik
3) Terapi Rehabilitatif

Yaitu dengan latihan *Visual-Vestibular* dan latihan berjalan (*Gait Exercise*)

a. Pada pasien yang harus berbaring :
    a) Melirik ke atas, ke bawah, ke samping kanan, ke samping kiri. Selanjutnya gerakan serupa sambil menatap jari yang digerakkan pada jarak 30 cm, mula-mula gerakkannya lambat, makin lama makin cepat.
    b) Gerakkan kepala fleksi dan ekstensi, makin lama makin cepat. Lalu diulangi dengan mata tertutup. Setelah itu, gerakkan kepala ke kiri dan ke kanan dengan urutan yang sama.
b. Untuk pasien yang sudah bisa duduk :
    a) Gerakkan kepala dengan cepat ke atas dan ke bawah, seperti sedang manggut, sebanyak 5 kali, lalu tunggu 10 detik atau lebih lama sampai vertigo menghilang. Ulangi latihan tersebut sebanyak 5 kali.
    b) Gerakkan kepala menatap ke kiri atau ke kanan atas selama 30 detik, kembali ke posisi biasa selama 30 detik, lalu menatap ke atas sisi lain selama 30 detik dan seterusnya. Ulangi latihan sebanyak 3 kali.
    c) Sambil duduk membungkuk dan mengambil benda yang diletakkan di lantai.
c. Untuk pasien yang sudah bisa berdiri (berjalan) :
    a) Sambil berdiri gerakkan mata, kepala seperti pada latihan a-a), a-b), b-a), dan b-b).

b) Duduk di kursi lalu berdiri dengan mata terbuka dan tertutup.

a. Jalan menyebrang ruangan dengan mata terbuka dan tertutup bergantian.
b. Berjalan tandem dengan mata terbuka dan tertutup bergantian. Lalu jalan tandem dengan kepala menghadap ke atas.
c. Jalan turun-naik pada lantai miring atau undakan dengan mata terbuka dan tertutup bergantian.
d. Jalan mengelilingi seseorang sambil melempar bola dengannya.
e. *Physical conditioning* dengan melakukan olahraga *bowling*, basket, *jogging*, dan *rowing*.

3.4 Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Perepresentasian dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi itu dapat diakses oleh prosedur pemecahan problema.

Setelah data (pengetahuan) yang dibutuhkan terkumpul, maka data tersebut direpresentasikan ke dalam format tertentu yang kemudian dihimpun dalam suatu basis pengetahuan. Representasi ini akan digunakan untuk menentukan proses pencarian atau menentukan kesimpulan yang didapat (Bunafit, 2008 : 29).

3.4.1 Tabel Penyakit

Berikut ini adalah daftar jenis-jenis vertigo berdasarkan saluran *vestibular* yang mengalami kerusakan :

Tabel 3.1 Jenis-jenis vertigo

Kode Penyakit	Jenis vertigo
0001	Vertigo Periferal
0002	Vertigo Sentral

3.4.2 Tabel Gejala

Berikut ini adalah daftar gejala vertigo secara umum :

Tabel 3.2 Gejala-Gejala Vertigo

Kode gejala	Gejala
0001	Mual.
0002	Muntah.
0003	Keringat dingin.
0004	Pandangan gelap.
0005	Lelah dan stamina menurun.
0006	Memori dan daya pikir menurun.
0007	Tidak mampu berkonsentrasi.
0008	Pusing.
0009	Sulit berdiri.
0010	Lemas (contoh : tidak mampu menggenggam erat benda dengan telapak tangan).
0011	Keadaan sekitar terasa berputar, benda diam di sekitar nampak bergoyang (*Oscillpsia*) dan nampak ganda (*Diplopia*).

0012	Sukar menelan
0013	Kelumpuhan otot-otot wajah.
0014	Sakit kepala yang parah
G015	Kesadaran terganggu.
0016	Hilangnya koordinasi.
0017	Telinga Berdenging (*tinnitus*) atau hilangnya pendengaran.
0018	Posisi kepala terfiksasi kearah tertentu.
0019	Sensitif pada cahaya terang dan suara.
0020	Rasa seolah-olah akan terjatuh pada permulaan tidur.
0021	Hilang keseimbangan, seperti tidak bisa tegak dan tidak bisa stabil ketika berdiri dan atau duduk jalan sempoyongan dan jalan membelok.
0022	Ketika berbaring miring atau ketika kepalanya menengadah ke atas keadaan sekitar terasa berputar selama 15-20 detik.
0023	Pandangan tidak fokus (gerakan mata ulang-alik diluar kemauan (*nistagimus* atau *nystagmoid jerks*)).

Berikut ini adalah daftar gejala vertigo periferal :

Tabel 3.3 Gejala Vertigo Periferal

Kode Gejala	**Gejala**
0001	Mual.
0002	Muntah.
0003	Keringat dingin.
0004	Pandangan gelap.
0005	Lelah dan stamina menurun.
0006	Memori dan daya pikir menurun.

0007	Tidak mampu berkonsentrasi.
0008	Pusing.
0017	Telinga Berdenging (*tinnitus*) atau hilangnya pendengaran.
0018	Posisi kepala terfiksasi kearah tertentu.
0019	Sensitif pada cahaya terang dan suara.
0020	Rasa seolah-olah akan terjatuh pada permulaan tidur.
0021	Hilang keseimbangan, seperti tidak bisa tegak dan tidak bisa stabil ketika berdiri dan atau duduk jalan sempoyongan dan jalan membelok.
0022	Ketika berbaring miring atau ketika kepalanya menengadah ke atas keadaan sekitar terasa berputar selama 15-20 detik.
0023	Pandangan tidak fokus (gerakan mata ulang-alik diluar kemauan (*nistagimus* atau *nystagmoid jerks*)).

Berikut ini adalah daftar gejala vertigo sentral :

Tabel 3.4 Gejala Vertigo Sentral

Kode Gejala	Gejala
0001	Mual.
0002	Muntah.
0003	Keringat dingin.
0004	Pandangan gelap.
0005	Lelah dan stamina menurun.
0006	Memori dan daya pikir menurun.
0007	Tidak mampu berkonsentrasi.
0008	Pusing.
0009	Sulit berdiri.

0010	Lemas (contoh : tidak mampu menggenggam erat benda dengan telapak tangan).
0011	Keadaan sekitar terasa berputar, benda diam di sekitar nampak bergoyang (*Oscillpsia*) dan nampak ganda (*Diplopia*).
0012	Sukar menelan
0013	Kelumpuhan otot-otot wajah.
0014	Sakit kepala yang parah
0015	Kesadaran terganggu.
0016	Hilangnya koordinasi.

3.4.3 Tabel Relasi Penyakit dan Gejala

Berikut ini adalah daftar relasi penyakit dengan gejalanya :

Tabel 3.5 Relasi Penyakit dan Gejala

		PENYAKIT	
	KODE	A	B
GEJALA	0001	*	*
	0002	*	*
	0003	*	*
	0004	*	*
	0005	*	*
	0006	*	*
	0007	*	*
	0008	*	*
	0009		*
	0010		*
	0011		*
	0012		*
	0013		*
	0014		*
	0015		*
	0016		*
	0017	*	
	0018	*	
	0019	*	
	0020	*	
	0021	*	
	0022	*	
	0023	*	

Keterangan :

Penyakit A : Vertigo Periferal

Penyakit B : Vertigo Sentral

3.4.4 Atruran Kaidah Produksi

Berdasarkan tabel 3.5, dapat disimpulkan kaidah produksinya adalah :

IF Mual

AND Muntah

AND Keringat dingin

AND Pandangan gelap

AND Lelah dan stamina menurun

AND Memori dan daya pikir menurun

AND Tidak mampu berkonsentrasi

AND Pusing

AND Telinga Berdenging (*tinnitus*) atau hilangnya pendengaran

AND Posisi kepala terfiksasi kearah tertentu.

AND Sensitif pada cahaya terang dan suara.

AND Rasa seolah-olah akan terjatuh pada permulaan tidur.

AND Hilang keseimbangan, seperti tidak bisa tegak dan tidak bisa stabil ketika berdiri dan atau duduk jalan sempoyongan dan jalan membelok.

AND Ketika berbaring miring atau ketika kepalanya menengadah ke atas keadaan sekitar terasa berputar selama 15-20 detik.

AND Pandangan tidak fokus (gerakan mata ulang-alik diluar kemauan (*nistagimus* atau *nystagmoid jerks*)).

THEN Vertigo Periferal

IF Mual

AND Muntah

AND Keringat dingin

AND Pandangan gelap

AND Lelah dan stamina menurun

AND Memori dan daya pikir menurun.

AND Tidak mampu berkonsentrasi.

AND Pusing

AND Sulit berdiri.

AND Lemas (contoh : tidak mampu menggenggam erat benda dengan telapak tangan).

AND Keadaan sekitar terasa berputar, benda diam di sekitar nampak bergoyang (*Oscillpsia*) dan nampak ganda (*Diplopia*).

AND Sukar menelan

AND Kelumpuhan otot-otot wajah.

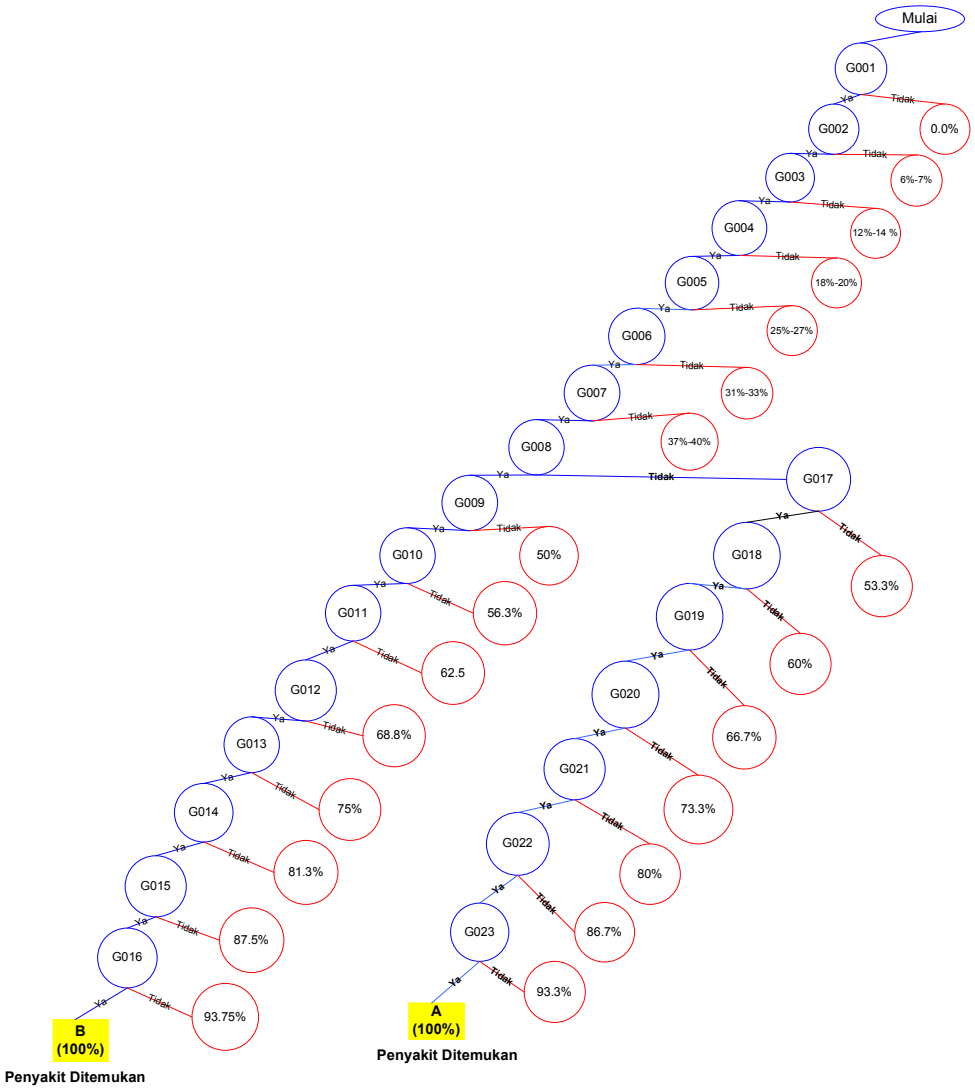
AND Sakit kepala yang parah

AND Kesadaran terganggu.

AND Hilangnya koordinasi.

THEN Vertigo Sentral

3.4.5 Pohon Pelacakan dan Penelusuran Penyakit



Gambar 3.1 Pohon Pelacakan

Keterangan :

G001, G002, G003,…G023	:	Kode gejala
A	:	Vertigo Periferal
B	:	Vertigo Sentral
Kesimpulan Hasil Akhir	:	1. Dipastikan (=100%)
		2. Kemungkinan (>=56% AND <100%)
		3. Diragukan (< 56%)

3.5 Metode Inferensi

Metode inferensi adalah mekanisme berfikir dan pola-pola penalaran yang digunakan oleh sistem untuk mencapai suatu kesimpulan. Metode ini akan menganalisis masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Penalaran dimulai dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Adapun metode yang digunakan dalam pembangunan sistem pakar ini adalah metode *backward chaining* sebagai metode penalarannya dan *depth first search* sebagai metode penelusurannya.

3.5.1 Backward Chaining

Runut mundur (*backward chaining*) adalah penalaran dimulai dari kesimpulan dan akan dibuktikan kebenarannya (*goal driven*). Metode merupakan cara yang efesien untuk memecahkan masalah yang dimodelkan sebagai masalah yang terstruktur. Metode *backward chaining* ini cocok digunakan untuk memecahkan masalah diagnosis (Schnupp, 1989).

3.5.2 Depth First Search

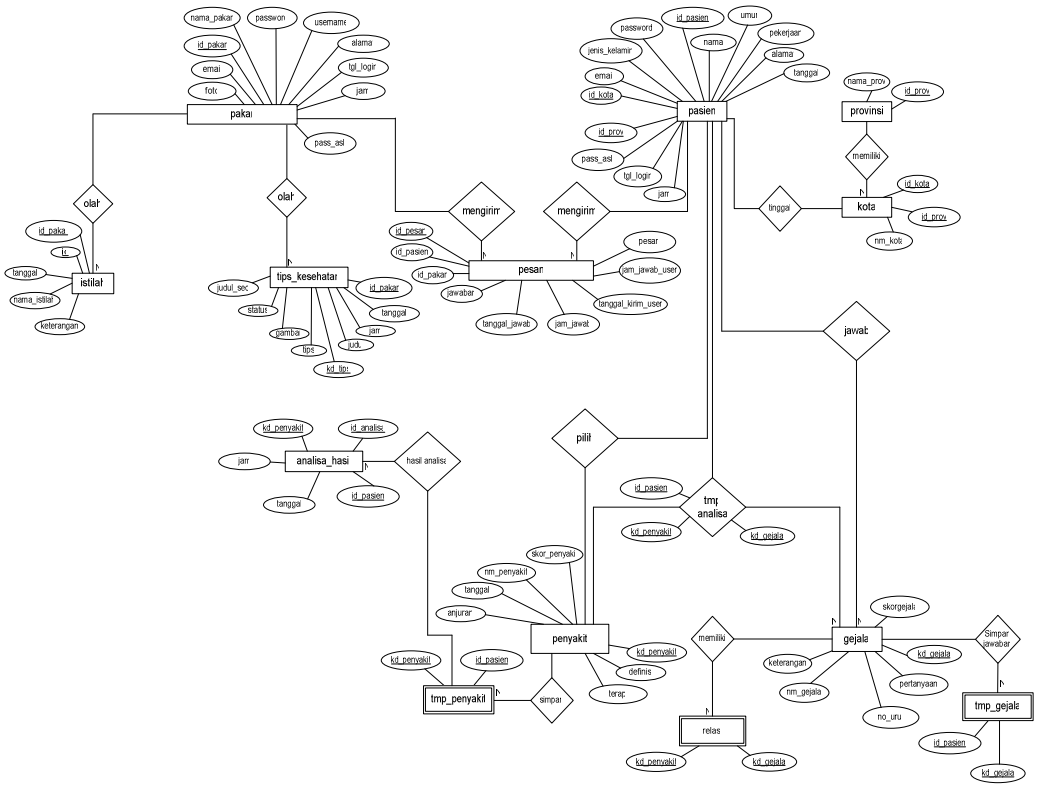
Depth first search merupakan teknik penelusuran dari node ke node bergerak menurun ke tingkat dalam yang berurutan. Proses pencarian dilakukan dengan mengunjungi cabang terlebih dahulu hingga tiba di simpul terakhir. Jika tujuan yang diinginkan belum tercapai maka pencarian dilanjutkan ke cabang sebelumnya, turun ke bawah jika memang masih ada cabangnya. Begitu seterusnya hingga diperoleh tujuan akhir (*goal*).

3.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini memaparkan analisis fungsional yang meliputi *Data Context Diagram*, *Data Flow Diagram*, spesifikasi proses, kamus data, *Entity Relationship Diagram*, dan skema relasi.

3.6.1 Entity Relationship Diagram (ERD)

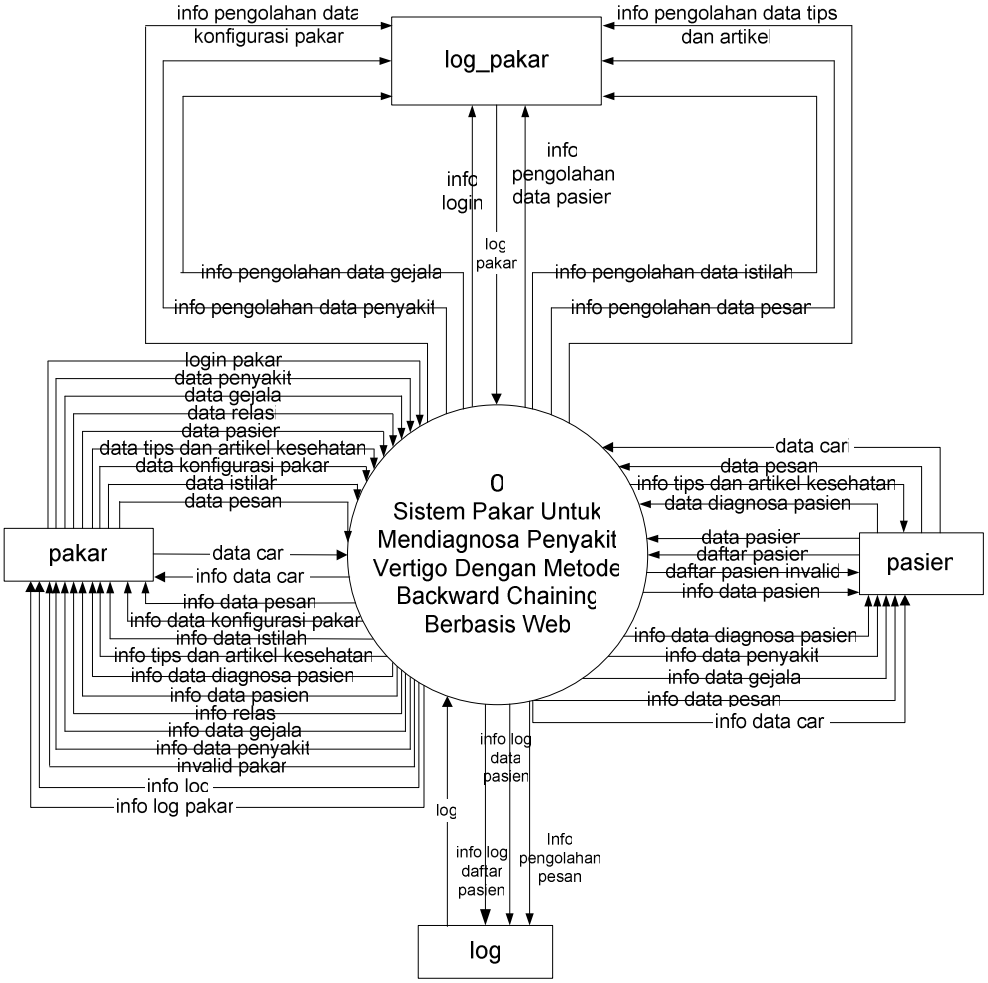
ERD digunakan untuk menggambarkan relasi antartabel dengan tujuan untuk memperjelas hubungan antartabel penyimpanan. ERD terdiri atas sekumpulan objek dasar yaitu entitas dan hubungan antar entitas-entitas yang saling berhubungan. ERD dalam aplikasi sistem pakar ini dapat ditunjukan pada gambar berikut.



Gambar 3.2 *Entity Relationship Diagram*

3.6.2 Data Context Diagram (DCD)

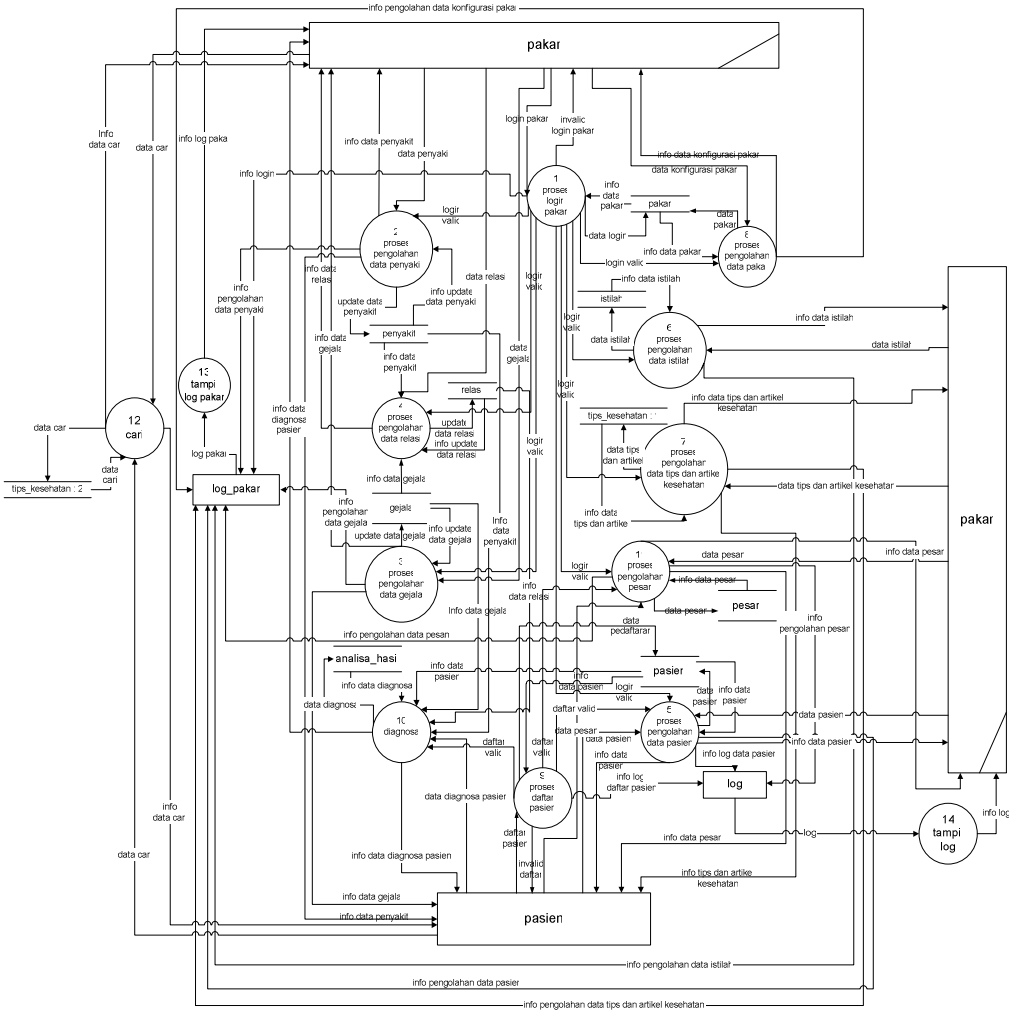
Diagram Konteks (*Data Context Diagram*) adalah kasus *data flow diagram* yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Dalam sistem ini terdapat dua entitas, yaitu entitas pakar dan entitas pasien. Adapun diagram konteks sistem pakar ini dapat dilihat pada gambar 3.3 sebagai berikut :



Gambar 3.3 *Data Context Diagram*

3.6.3 Data Flow Diagram (DFD) Level 1

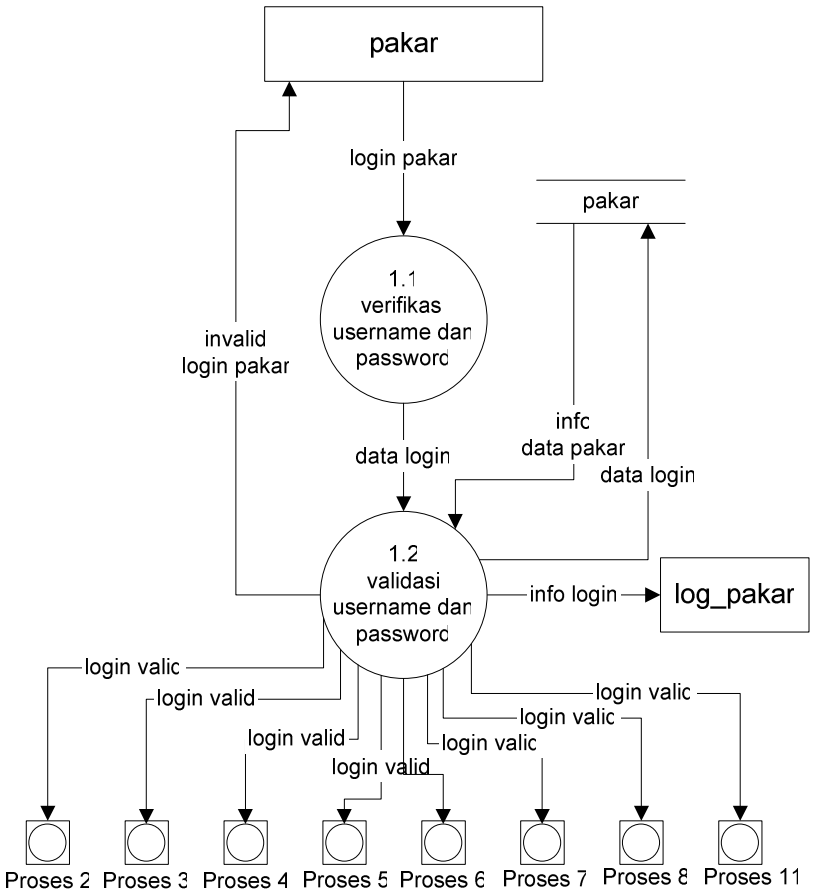
Data Flow Diagram digunakan untuk menggambarkan sistem sebagai sebuah jaringan dari proses-proses secara fungsional yang dihubungkan satu dengan yang lainnya oleh aliran data. Dalam *Data Flow Diagram* terdiri atas entitas luar, aliran data, proses, dan penyimpanan data.



Gambar 3.4 *Data Flow Diagram* Level 1

3.6.3 DFD Level 2 Proses 1 (Login Pakar)

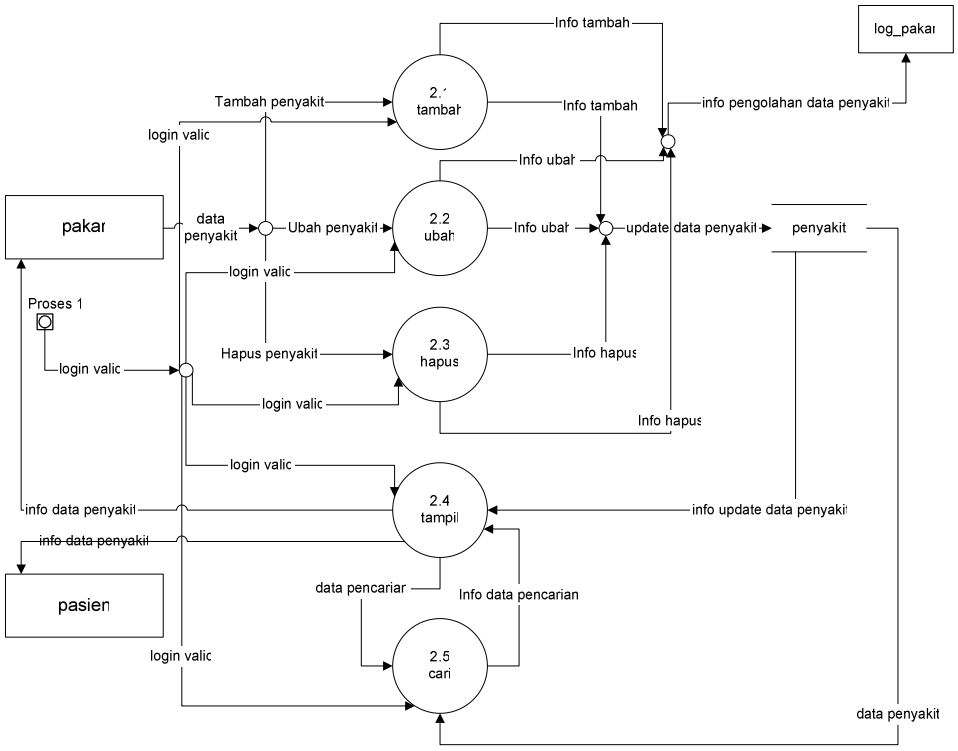
Proses 1 *login* pakar di DFD level 1 dirincikan lagi menjadi dua subproses yaitu verifikasi *username* dan *password* serta validasi *username* dan *password*. Model penurunan proses tersebut dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Data Flow Diagram* Level 2 Proses 1

3.6.4 DFD Level 2 Proses 2 (Pengolahan Data Penyakit)

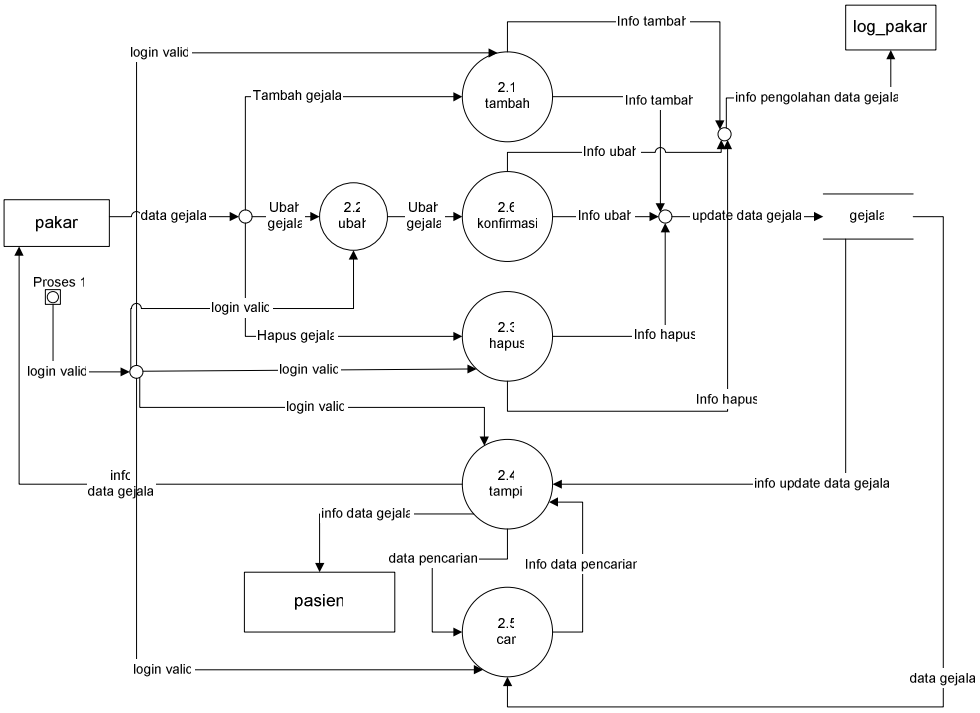
Proses 2 pengolahan data penyakit di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Data Flow Diagram* Level 2 Proses 2 (Pengolahan Data Penyakit)

3.6.5 DFD Level 2 Proses 3 (Pengolahan Data Gejala)

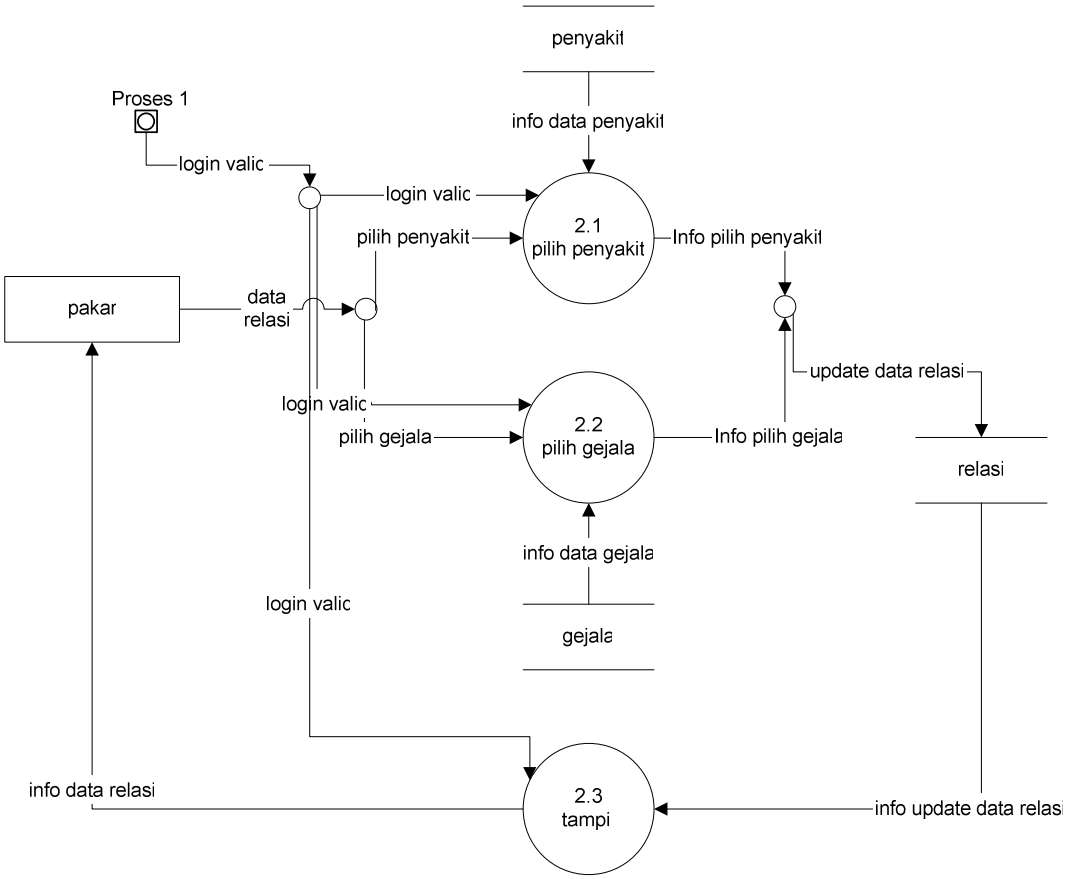
Proses 3 pengolahan data gejala di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.7.



Gambar 3.7 *Data Flow Diagram* Level 2 Proses 3 (Pengolahan Data Gejala)

3.6.6 DFD Level 2 Proses 4 (Pengolahan Data Relasi)

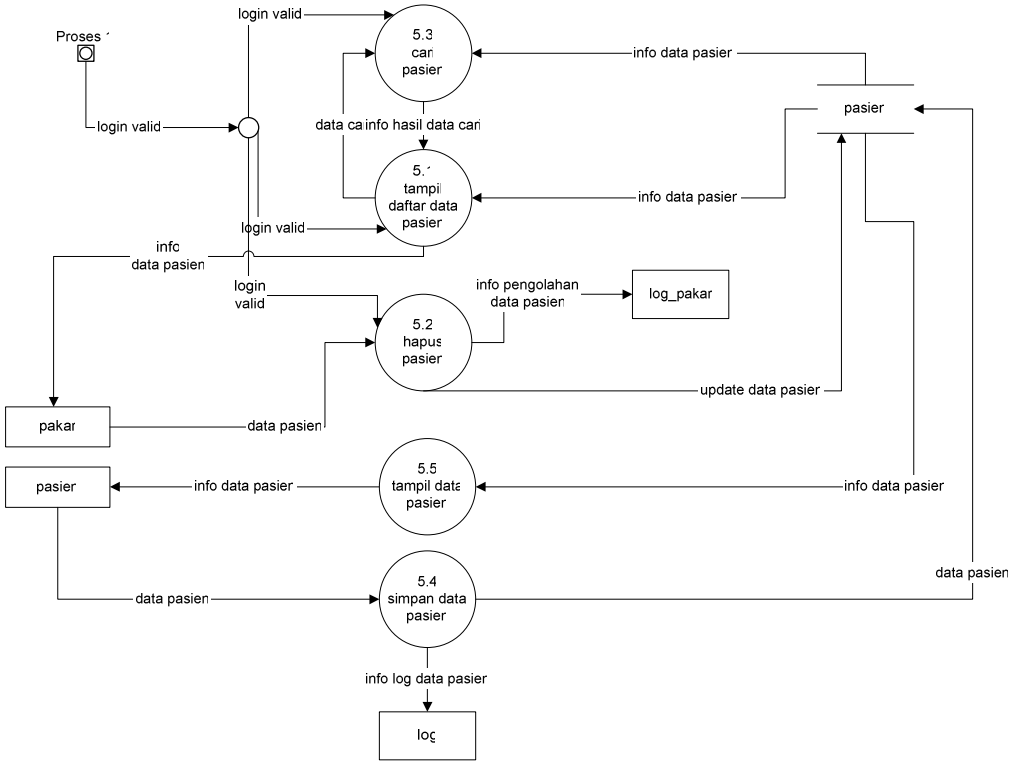
Proses 4 pengolahan data relasi di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Data Flow Diagram* Level 2 Proses 4 (Pengolahan Data Relasi)

3.6.7 DFD Level 2 Proses 5 (Pengolahan Data Pasien)

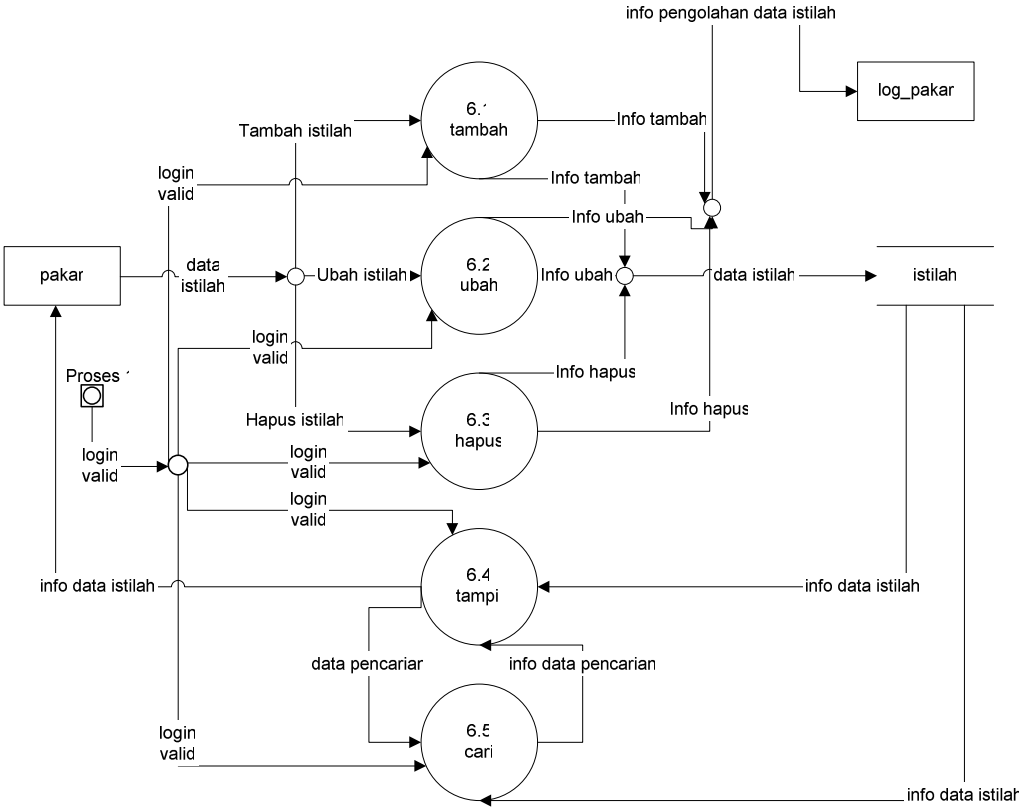
Proses 5 pengolahan data pasien di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Data Flow Diagram* Level 2 Proses 5 (Pengolahan Data Pasien)

3.6.8 DFD Level 2 Proses 6 (Pengolahan Data Istilah)

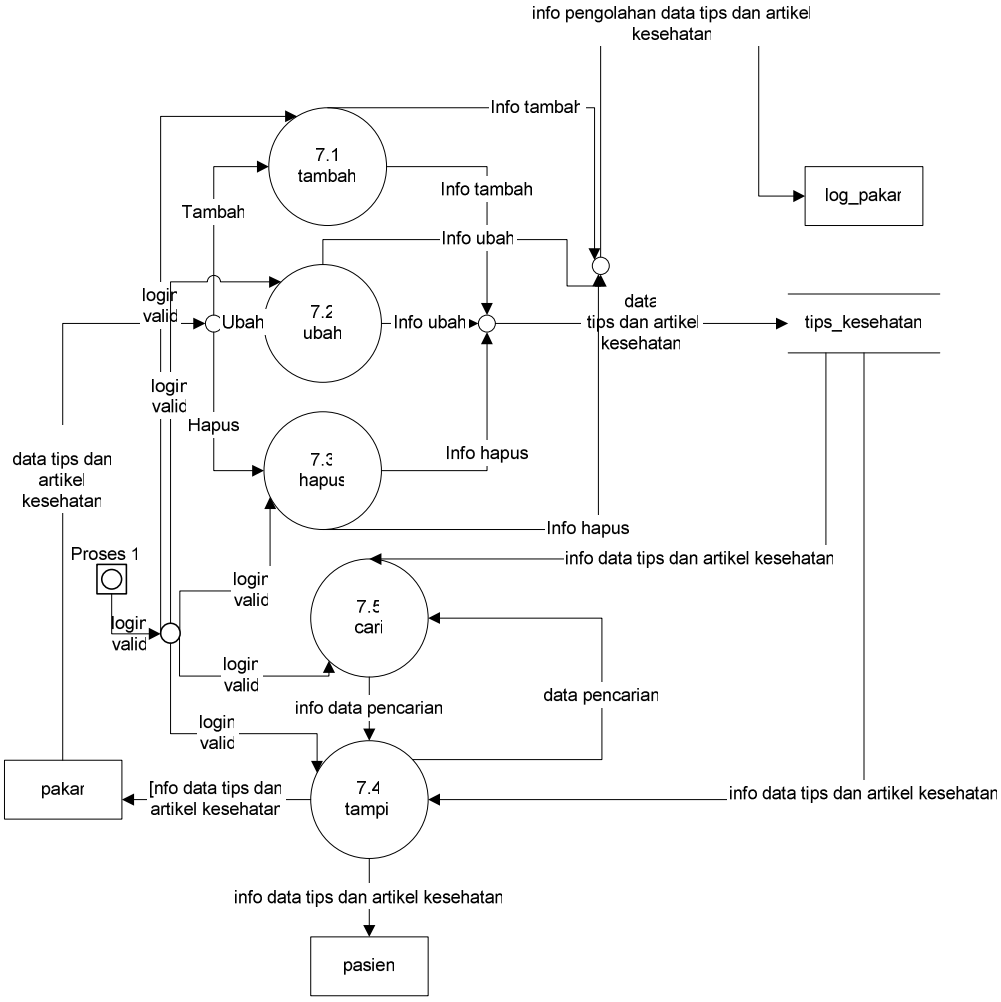
Proses 6 pengolahan data istilah di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.10.



Gambar 3.10 *Data Flow Diagram* Level 2 Proses 6 (Pengolahan Data Istilah)

3.6.9 DFD Level 2 Proses 7 (Pengolahan Data Tips dan Artikel Kesehatan)

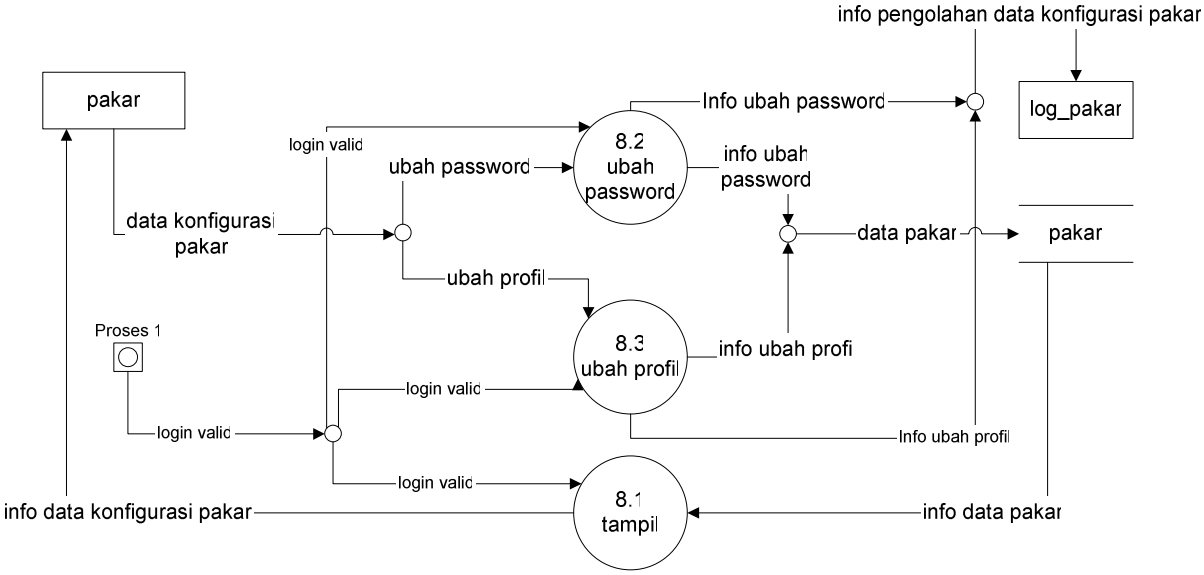
Proses 7 pengolahan data tips dan artikel kesehatan di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.11.



Gambar 3.11 *Data Flow Diagram* Level 2 Proses 7
(Pengolahan Data Tips dan Artikel Kesehatan)

3.6.10 DFD Level 2 Proses 8 (Pengolahan Data Pakar)

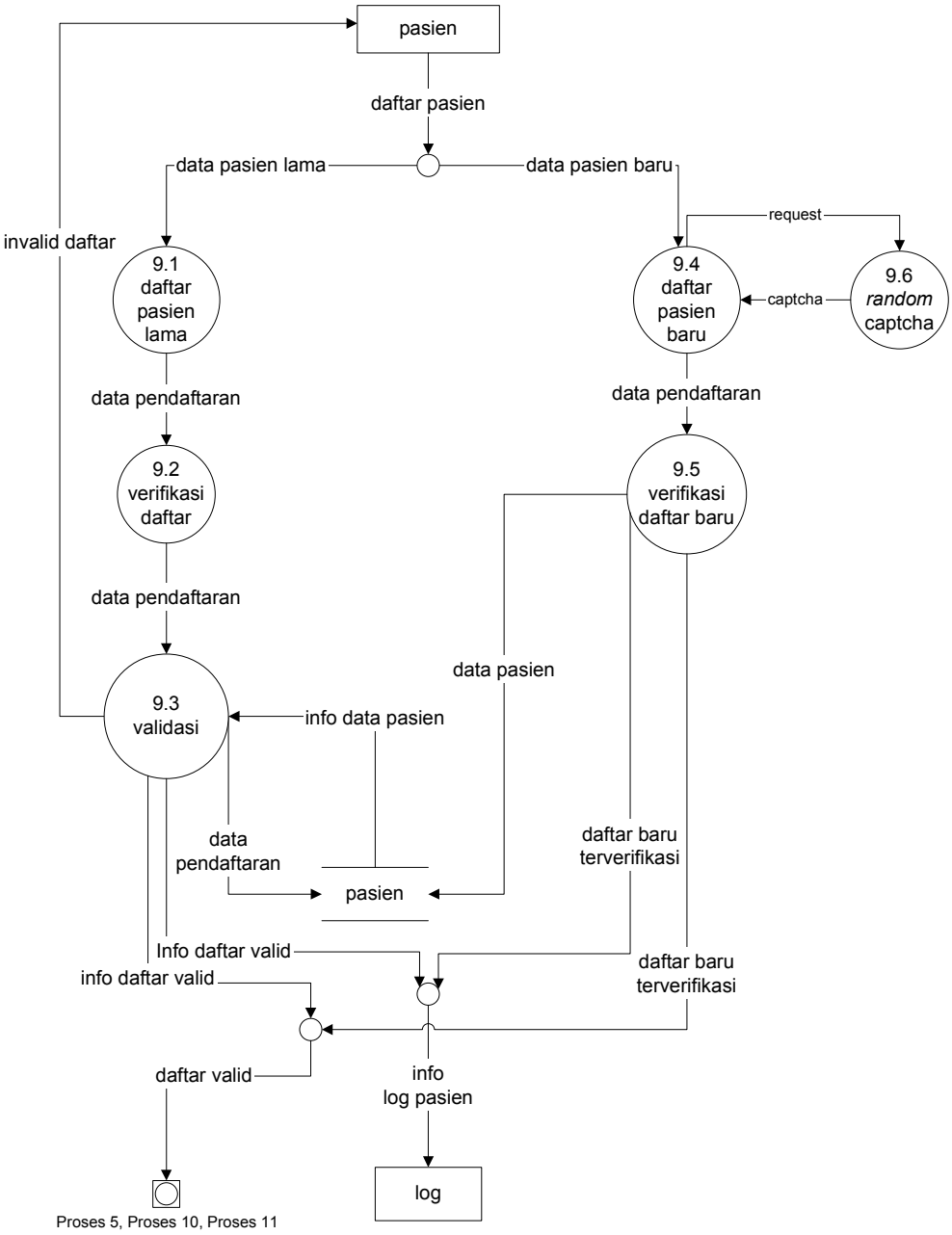
Proses 8 pengolahan data pakar di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Data Flow Diagram* Level 2 Proses 8 (Pengolahan Data Pakar)

3.6.11 DFD Level 2 Proses 9 (Daftar Pasien)

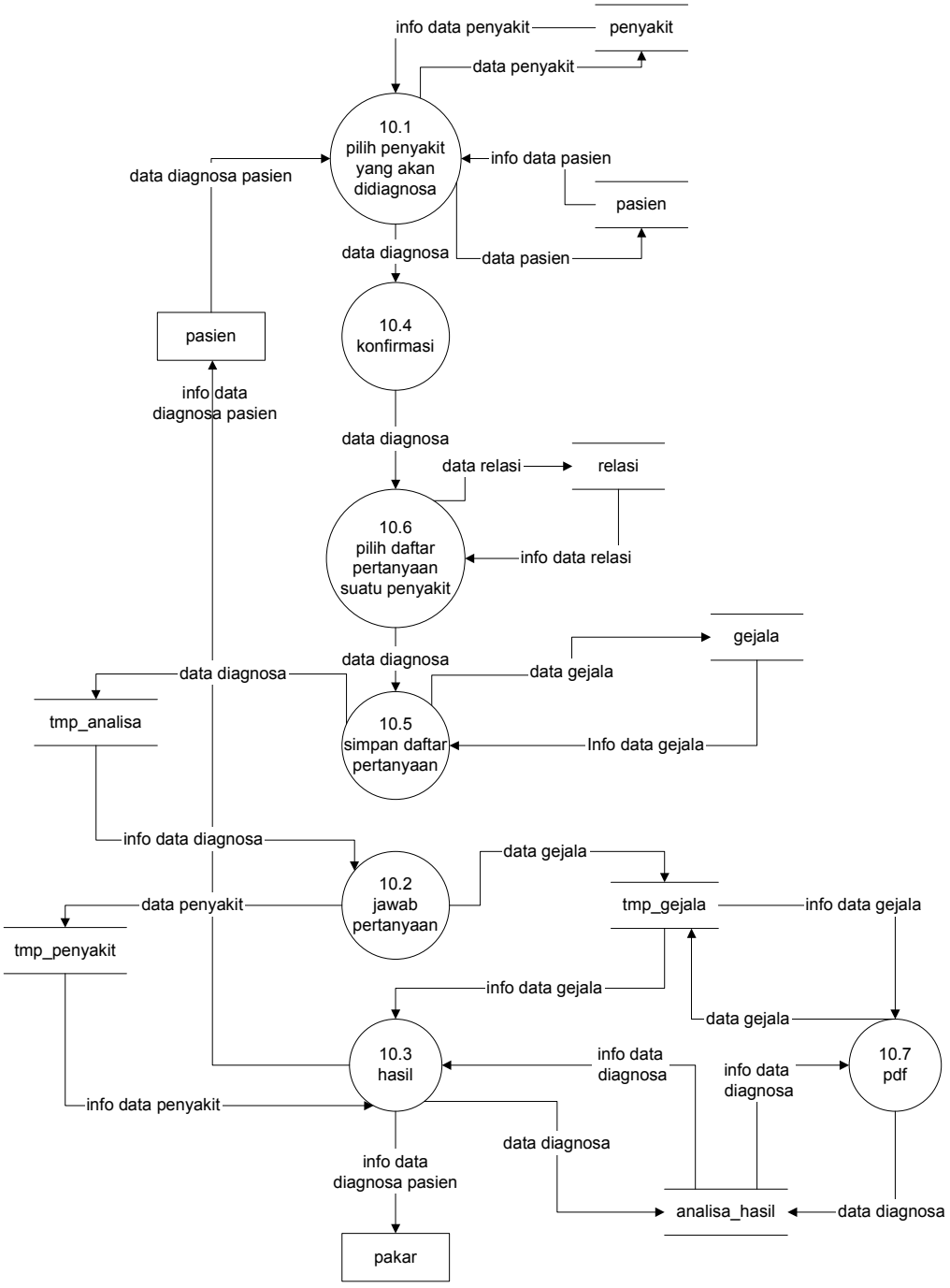
Proses 9 daftar pasien di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.13.



Gambar 3.13 *Data Flow Diagram* Level 2 Proses 9 (Daftar Pasien)

3.6.12 DFD Level 2 Proses 10 (Diagnosa)

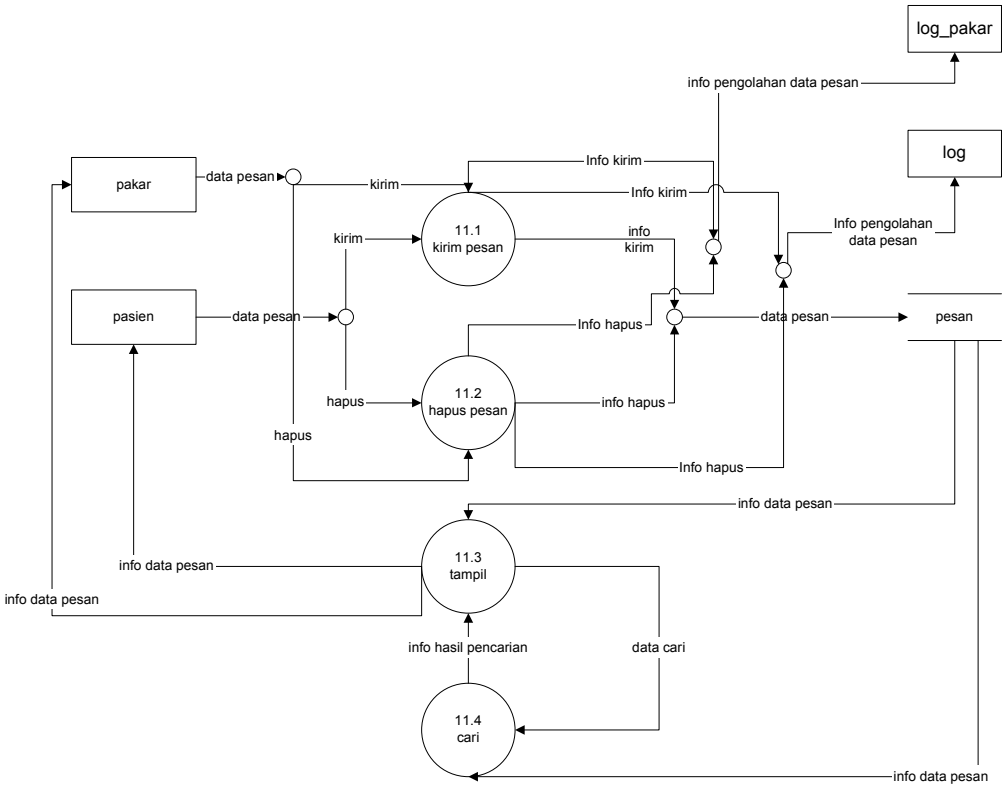
Proses 10 diagnosa di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Data Flow Diagram* Level 2 Proses 10 Diagnosa

3.6.13 DFD Level 2 Proses 11 (Pengolahan Data Pesan)

Proses 11 pengolahan data pasien di DFD level 1 dirincikan lagi menjadi beberapa subproses. Model penurunan proses tersebut dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Data Flow Diagram* Level 2 Proses 11 Pengolahan Data Pesan

3.6.14 Spesifikasi Proses

Spesifikasi proses digunakan untuk menjelaskan proses-proses yang ada dalam diagram aliran data secara terperinci. Hal tersebut bermanfaat untuk mengetahui lebih jelas bagaimana logika dari proses-proses yang ada.

Tabel 3.6 Spesifikasi Proses

No.	Proses	Keterangan
1	No. Proses	1
	Nama Proses	proses *login* pakar
	Sumber	pakar
	Input	*login* pakar
	Output	Informasi *login* pakar (*valid* atau *invalid*)
	Deskripsi	Menampilkan halaman *login* pakar
	Logika Proses	Pakar memasukkan data *login* berupa *username* dan *password* IF *login* pakar benar THEN *Login Valid,* Masuk ke halaman pakar ELSE *Login Invalid,* tampil *invalid* data pakar
2	No. Proses	2
	Nama Proses	proses pengolahan data penyakit
	Sumber	pakar
	Input	data penyakit, info *update* penyakit
	Output	info data penyakit, *update* data penyakit
	Deskripsi	Menampilkan halaman daftar penyakit
	Logika Proses	Pakar menambah, merubah, dan menghapus data penyakit
3	No. Proses	3
	Nama Proses	proses pengolahan data gejala
	Sumber	pakar
	Input	data gejala, info *update* gejala
	Output	info data gejala, *update* data gejala
	Deskripsi	Menampilkan halaman daftar gejala
	Logika Proses	Pakar menambah, merubah, dan menghapus data gejala
4	No. Proses	4
	Nama Proses	proses pengolahan data relasi
	Sumber	pakar
	Input	data relasi, info data penyakit dan info data gejala

	Output	info data relasi
	Deskripsi	Menampilkan halaman daftar relasi
	Logika Proses	Pakar memanipulasi data relasi
5	No. Proses	5
	Nama Proses	proses pengolahan data pasien
	Sumber	pakar, *user*
	Input	data pasien, info data pasien
	Output	info data pasien, info pengolahan data pasien
	Deskripsi	Menampilkan halaman daftar pasien
	Logika Proses	1. Pakar dapat menghapus data pasien 2. Pasien memasukkan data pada form pendaftaran
6	No. Proses	6
	Nama Proses	proses pengolahan data istilah
	Sumber	pakar
	Input	data istilah
	Output	info data istilah, info pengolahan data istilah
	Deskripsi	Menampilkan halaman daftar istilah
	Logika Proses	Pakar menambah, merubah, dan menghapus data istilah
7	No. Proses	7
	Nama Proses	proses pengolahan data tips dan artikel kesehatan
	Sumber	pakar
	Input	data data tips dan artikel kesehatan
	Output	info data tips dan artikel kesehatan, info pengolahan data tips dan artikel kesehatan
	Deskripsi	Menampilkan halaman daftar tips dan artikel kesehatan
	Logika Proses	Pakar menambah, merubah, dan menghapus data tips dan artikel kesehatan
8	No. Proses	8
	Nama Proses	proses pengolahan data pakar
	Sumber	pakar
	Input	data konfigurasi pakar, info data pakar
	Output	info data konfigurasi pakar, info pengolahan data konfigurasi pakar
	Deskripsi	Menampilkan halaman profil pakar
	Logika Proses	Pakar merubah (mengkonfigurasi) data pakar
9	No. Proses	9
	Nama Proses	proses daftar pasien
	Sumber	pasien
	Input	data pasien
	Output	daftar *valid*, daftar *invalid*, dan info data pasien
	Deskripsi	Menyimpan data pasien dan menampilkan halaman pasien

	Logika Proses	Pasien memasukkan data pasien ke dalam proses melalui form pendaftaran IF daftar benar THEN daftar *valid,* masuk ke halaman pasien ELSE daftar *invalid,* tampil daftar *invalid*
10	No. Proses	10
	Nama Proses	diagnosa
	Sumber	pasien
	Input	data diagnosa pasien, info data gejala, info data penyakit, info data relasi, info data pasien, info data diagnosa
	Output	info data diagnosa pasien,data diagnosa
	Deskripsi	menampilkan halaman konsultasi
	Logika Proses	1. Pasien memilih penyakit yang akan didiagnosa 2. Pasien menjawab pertanyaan yang didasarkan dari gejala-gejala yang timbul
11	No. Proses	11
	Nama Proses	proses pengolahan data pesan
	Sumber	pakar, pasien
	Input	data pesan
	Output	info data pesan
	Deskripsi	menampilkan halaman kontak
	Logika Proses	1. Pakar mengirim pesan kepada pasien, menghapus dan merubah pesan yang dikirim oleh pakar 2. Pasien mengirim pesan kepada pakar, menghapus dan merubah pesan yang dikirim oleh pasien
12	No. Proses	12
	Nama Proses	cari
	Sumber	pakar, pasien
	Input	data cari
	Output	info data cari, data cari
	Deskripsi	menampilkan halaman hasil pencarian
	Logika Proses	/* cari data tips dan artikel kesehatan berdasarkan kata kunci yang dicari*/ IF ada data penyakit di dalam *database* THEN tampil daftar data tips dan artikel kesehatan ELSE data tips dan artikel kesehatan yang dicari tidak ada

13	No. Proses	13
	Nama Proses	tampil log pakar
	Sumber	pakar
	Input	info pengolahan data konfigurasi pakar, info pengolahan data gejala, info pengolahan data penyakit, info *login*, info pengolahan data pasien, info pengolahan data istilah, info pengolahan data pesan, info pengolahan data tips dan artikel kesehatan
	Output	log pakar
	Deskripsi	menampilkan halaman log pakar
	Logika Proses	
14	No. Proses	14
	Nama Proses	tampil log
	Sumber	pasien
	Input	info log daftar pasien, info pengolahan pesan, info log data pasien
	Output	log
	Deskripsi	menampilkan halaman log pasien
	Logika Proses	
15	No. Proses	1.1
	Nama Proses	verifikasi *username* dan *password*
	Sumber	pakar
	Input	*login* pakar
	Output	data *login*
	Deskripsi	data *login* yang dimasukkan diverifikasi
	Logika Proses	IF *login* pakar terverifikasi THEN lanjutkan ke proses validasi data *login* ELSE *Login Invalid,* tampil *invalid login* pakar
16	No. Proses	1.2
	Nama Proses	validasi *username* dan *password*
	Sumber	pakar
	Input	data *login*
	Output	*login* valid, *invalid login* pakar
	Deskripsi	verifikasi *login* pakar
	Logika Proses	IF data *login valid* THEN *login valid, tampil halaman pakar* ELSE *Login Invalid,* tampil *invalid* data pakar
17	No. Proses	2.1
	Nama Proses	tambah
	Sumber	pakar
	Input	data penyakit

	Output	*update* data penyakit, info pengolahan data penyakit
	Deskripsi	menampilkan halaman tambah data penyakit
	Logika Proses	/*Pakar menambah data penyakit*/ IF tambah data penyakit *valid* THEN tambah data penyakit ELSE tambah data penyakit gagal
18	No. Proses	2.2
	Nama Proses	ubah
	Sumber	pakar
	Input	data penyakit
	Output	*update* data penyakit, info pengolahan data penyakit
	Deskripsi	menampilkan halaman ubah data penyakit
	Logika Proses	/*Pakar merubah data penyakit*/ IF ubah data penyakit *valid* THEN ubah data penyakit ELSE ubah data penyakit gagal
19	No. Proses	2.3
	Nama Proses	hapus
	Sumber	pakar
	Input	data penyakit
	Output	*update* data penyakit, info pengolahan data penyakit
	Deskripsi	menampilkan halaman daftar data penyakit
	Logika Proses	/*Pakar menghapus data penyakit*/ IF hapus data penyakit *valid* THEN hapus data penyakit ELSE hapus data penyakit gagal
20	No. Proses	2.4
	Nama Proses	tampil
	Sumber	pakar
	Input	info *update* data penyakit, info data pencarian
	Output	info data penyakit, data pencarian
	Deskripsi	menampilkan halaman daftar data penyakit
	Logika Proses	IF ada data penyakit di dalam *database* THEN tampil daftar data penyakit ELSE

		gagal tampil
21	No. Proses	2.5
	Nama Proses	cari
	Sumber	pakar
	Input	data pencarian
	Output	info data pencarian, info data penyakit
	Deskripsi	menampilkan halaman daftar data penyakit yang dicari
	Logika Proses	/* cari data penyakit berdasarkan data penyakit yang dicari */ IF ada data penyakit di dalam *database* THEN tampil daftar data penyakit ELSE data penyakit yang dicari tidak ada
22	No. Proses	3.1
	Nama Proses	tambah
	Sumber	pakar
	Input	data gejala
	Output	*update* data gejala, info pengolahan data gejala
	Deskripsi	menampilkan halaman tambah data gejala
	Logika Proses	/*Pakar menambah data gejala*/ IF tambah data gejala *valid* THEN tambah data gejala ELSE tambah data gejala gagal
23	No. Proses	3.2
	Nama Proses	ubah
	Sumber	pakar
	Input	data gejala
	Output	*update* data gejala, info pengolahan data gejala
	Deskripsi	menampilkan halaman ubah data gejala
	Logika Proses	/*Pakar mengubah data gejala*/ IF ubah data gejala *valid* THEN ubah data gejala ELSE ubah data gejala gagal
24	No. Proses	3.3
	Nama Proses	hapus
	Sumber	pakar

	Input	data gejala
	Output	*update* data gejala, info pengolahan data gejala
	Deskripsi	menampilkan halaman daftar data gejala
	Logika Proses	/*Pakar menghapus data gejala*/ IF hapus data gejala *valid* THEN hapus data gejala ELSE hapus data gejala gagal
25	No. Proses	3.4
	Nama Proses	tampil
	Sumber	pakar
	Input	info *update* data gejala, info data pencarian
	Output	info data gejala, data pencarian
	Deskripsi	menampilkan halaman daftar data gejala
	Logika Proses	IF ada data gejala di dalam *database* THEN tampil daftar data gejala ELSE gagal tampil
26	No. Proses	3.5
	Nama Proses	cari
	Sumber	pakar
	Input	data gejala, data pencarian
	Output	data gejala
	Deskripsi	menampilkan halaman daftar data gejala yang dicari
	Logika Proses	/* cari data gejala berdasarkan data gejala yang dicari */ IF ada data gejala di dalam *database* THEN tampil daftar data gejala ELSE data gejala yang dicari tidak ada
27	No. Proses	4.1
	Nama Proses	pilih penyakit
	Sumber	pakar
	Input	data relasi, data penyakit
	Output	*update* data relasi
	Deskripsi	menampilkan halaman daftar data relasi
	Logika Proses	/* pakar menambah data relasi, penambahan data relasi berupa data penyakit yang diambil dari tabel penyakit, kemudian disimpan dalam tabel relasi sebagai kunci

		tamu (*foreign key*)*/ IF data relasi ditambah THEN data berhasil disimpan ELSE Data gagal disimpan
28	No. Proses	4.2
	Nama Proses	pilih gejala
	Sumber	pakar
	Input	data relasi, data gejala
	Output	*update* data relasi
	Deskripsi	menampilkan halaman daftar data relasi
	Logika Proses	/* pakar menambah data relasi, penambahan data relasi berupa data gejala yang diambil dari tabel gejala, kemudian disimpan dalam tabel relasi sebagai kunci tamu (*foreign key*)*/ IF data relasi ditambah THEN data berhasil disimpan ELSE Data gagal disimpan
29	No. Proses	4.3
	Nama Proses	tampil
	Sumber	pakar
	Input	info *update* data relasi
	Output	info data relasi
	Deskripsi	menampilkan halaman daftar data relasi
	Logika Proses	IF ada data relasi di dalam *database* THEN tampil daftar data relasi ELSE gagal tampil
30	No. Proses	5.1
	Nama Proses	tampil
	Sumber	pakar
	Input	info data pasien, info data hasil cari
	Output	data cari, info data cari
	Deskripsi	menampilkan halaman daftar data pasien
	Logika Proses	IF ada data pasien di dalam *database* THEN tampil daftar data pasien ELSE

		gagal tampil
31	No. Proses	5.2
	Nama Proses	hapus pasien
	Sumber	pakar
	Input	data pasien
	Output	info pengolahan data pasien, *update* data pasien
	Deskripsi	menampilkan halaman daftar data pasien
	Logika Proses	/*Pakar menghapus data pasien*/ IF hapus data pasien *valid* THEN hapus data pasien ELSE hapus data pasien gagal
32	No. Proses	5.3
	Nama Proses	cari
	Sumber	pakar
	Input	info data pasien, data cari
	Output	info data hasil cari
	Deskripsi	menampilkan halaman daftar data pasien yang dicari
	Logika Proses	/* cari data pasien berdasarkan data pasien yang dicari di dalam *database*/ IF ada data pasien di dalam *database* THEN tampil daftar data pasien ELSE data pasien yang dicari tidak ada
33	No. Proses	5.4
	Nama Proses	simpan data pasien
	Sumber	pasien
	Input	data pasien
	Output	info log data pasien
	Deskripsi	menampilkan halaman pendaftaran pasien
	Logika Proses	/*pasien memasukkan data ke dalam sistem melalui form pendaftaran*/ IF simpan data pasien *valid* THEN simpan data pasien ELSE data pasien yang dicari tidak ada
34	No. Proses	5.5
	Nama Proses	tampil data pasien
	Sumber	pasien
	Input	info data pasien
	Output	info data pasien

	Deskripsi	menampilkan halaman pasien
	Logika Proses	IF tampil data pasien *valid* THEN tampil halaman data pasien ELSE gagal tampil
35	No. Proses	6.1
	Nama Proses	tambah
	Sumber	pakar
	Input	data istilah
	Output	data istilah, info pengolahan data istilah
	Deskripsi	menampilkan halaman tambah data istilah
	Logika Proses	/*Pakar menambah data istilah*/ IF tambah data istilah *valid* THEN tambah data istilah ELSE tambah data istilah gagal
36	No. Proses	6.2
	Nama Proses	ubah
	Sumber	pakar
	Input	data istilah
	Output	data istilah, info pengolahan data istilah
	Deskripsi	menampilkan halaman ubah data istilah
	Logika Proses	/*Pakar mengubah data istilah */ IF ubah data istilah *valid* THEN ubah data istilah ELSE ubah data istilah gagal
37	No. Proses	6.3
	Nama Proses	hapus
	Sumber	pakar
	Input	data istilah
	Output	data istilah, info pengolahan data istilah
	Deskripsi	menampilkan halaman daftar data istilah
	Logika Proses	/*Pakar menghapus data istilah */ IF hapus data istilah *valid* THEN hapus data istilah ELSE hapus data istilah gagal
38	No. Proses	6.4

	Nama Proses	tampil
	Sumber	pakar
	Input	info data istilah, info data pencarian
	Output	info data istilah, data pencarian
	Deskripsi	menampilkan halaman daftar data istilah
	Logika Proses	IF ada data istilah di dalam *database* THEN tampil daftar data istilah ELSE gagal tampil
39	No. Proses	6.5
	Nama Proses	cari
	Sumber	pakar
	Input	info data istilah
	Output	info data pencarian
	Deskripsi	menampilkan halaman daftar data istilah yang dicari
	Logika Proses	/* cari data istilah berdasarkan data istilah yang dicari */ IF ada data istilah di dalam *database* THEN tampil daftar data istilah ELSE data istilah yang dicari tidak ada
40	No. Proses	7.1
	Nama Proses	tambah
	Sumber	pakar
	Input	data tips dan artikel kesehatan
	Output	data tips dan artikel kesehatan, info pengolahan data tips dan artikel kesehatan
	Deskripsi	menampilkan halaman tambah data tips dan artikel kesehatan
	Logika Proses	/*Pakar menambah data tips dan artikel kesehatan */ IF tambah data tips dan artikel kesehatan *valid* THEN tambah data tips dan artikel kesehatan ELSE tambah data tips dan artikel kesehatan gagal
41	No. Proses	7.2
	Nama Proses	ubah
	Sumber	pakar

	Input	data tips dan artikel kesehatan
	Output	data tips dan artikel kesehatan, info pengolahan data tips dan artikel kesehatan
	Deskripsi	menampilkan halaman ubah data tips dan artikel kesehatan
	Logika Proses	/*Pakar mengubah data tips dan artikel kesehatan*/ IF ubah data tips dan artikel kesehatan *valid* THEN ubah data tips dan artikel kesehatan ELSE ubah data tips dan artikel kesehatan gagal
42	No. Proses	7.3
	Nama Proses	hapus
	Sumber	pakar
	Input	data tips dan artikel kesehatan
	Output	data tips dan artikel kesehatan, info pengolahan data tips dan artikel kesehatan
	Deskripsi	menampilkan halaman daftar data tips dan artikel kesehatan
	Logika Proses	/*Pakar menghapus data tips dan artikel kesehatan*/ IF hapus data tips dan artikel kesehatan *valid* THEN hapus data tips dan artikel kesehatan ELSE hapus data tips dan artikel kesehatan gagal
43	No. Proses	7.4
	Nama Proses	tampil
	Sumber	pakar
	Input	info data tips dan artikel kesehatan, info data pencarian
	Output	info data tips dan artikel kesehatan, data pencarian
	Deskripsi	menampilkan halaman daftar data tips dan artikel kesehatan
	Logika Proses	IF ada data tips dan artikel kesehatan di dalam *database* THEN tampil daftar data tips dan artikel kesehatan ELSE gagal tampil
44	No. Proses	7.5
	Nama Proses	cari

	Sumber	pakar
	Input	info data tips dan artikel kesehatan, data pencarian
	Output	info data pencarian
	Deskripsi	menampilkan halaman daftar data tips dan artikel kesehatan yang dicari
	Logika Proses	/* cari data tips dan artikel kesehatan berdasarkan data tips dan artikel kesehatan yang dicari */ IF ada data tips dan artikel kesehatan di dalam *database* THEN tampil daftar data tips dan artikel kesehatan ELSE data tips dan artikel kesehatan yang dicari tidak ada
45	No. Proses	8.1
	Nama Proses	tampil
	Sumber	pakar
	Input	info data pakar
	Output	info data konfigurasi pakar
	Deskripsi	menampilkan halaman profil pakar
	Logika Proses	IF ada pakar di dalam *database* THEN tampil daftar profil data pakar ELSE gagal tampil
45	No. Proses	8.2
	Nama Proses	ubah
	Sumber	pakar
	Input	data pakar (*password* lama, *password* baru, dan konfirmasi *password* baru)
	Output	validasi (pesan data pakar diubah)
	Deskripsi	menampilkan halaman ubah *password* pakar
	Logika Proses	/*Pakar mengubah data *password* pakar */ IF cek *password* lama *valid* THEN IF *password* baru = konfirmasi *password* baru THEN SET *password* lama = *password* baru ELSE gagal, *password* baru != konfirmasi *password* baru ELSE

		ubah *password* pakar gagal
47	No. Proses	8.3
	Nama Proses	ubah
	Sumber	pakar
	Input	data konfigurasi pakar
	Output	data pakar, info pengolahan data konfigurasi pakar
	Deskripsi	menampilkan halaman ubah profil pakar
	Logika Proses	/*Pakar mengubah data profil pakar */ IF ubah data profil pakar *valid* THEN ubah data profil pakar ELSE ubah data profil pakar gagal
48	No. Proses	9.1
	Nama Proses	daftar pasien lama
	Sumber	*user* (pasien)
	Input	daftar pasien lama
	Output	data pendaftaran
	Deskripsi	menampilkan halaman daftar pasien
	Logika Proses	/*Pasien memasukkan *email* dan *password* pada form pendaftaran*/ IF *email valid* AND *password valid* THEN daftar *valid*, tampil halaman pasien ELSE daftar pasien *invalid*
59	No. Proses	9.1
	Nama Proses	daftar pasien lama
	Sumber	pasien
	Input	data pasien lama
	Output	data pendaftaran
	Deskripsi	menampilkan halaman daftar pasien
	Logika Proses	Pasien memasukkan *email* dan *password* pada form pendaftaran
60	No. Proses	9.2
	Nama Proses	verifikasi daftar
	Sumber	pasien
	Input	data pendaftaran
	Output	data pendaftaran
	Deskripsi	verifikasi *email* dan *password*

	Logika Proses	IF *email* terverifikasi AND *password* terverifikasi THEN lanjutkan ke proses berikutnya ELSE daftar pasien *invalid*
51	No. Proses	9.3
	Nama Proses	validasi
	Sumber	pasien
	Input	data pendaftaran, info data pasien
	Output	data pendaftaran, Info daftar valid
	Deskripsi	validasi data pendaftaran
	Logika Proses	/*Pasien memasukkan data pasien pada form pendaftaran*/ IF data pasien *valid* THEN daftar *valid*, tampil halaman pasien ELSE daftar pasien *invalid*
52	No. Proses	9.4
	Nama Proses	daftar pasien baru
	Sumber	pasien
	Input	data pasien baru, captcha
	Output	data pendaftaran
	Deskripsi	menampilkan halaman daftar pasien
	Logika Proses	Pasien memasukkan data pasien pada form pendaftaran
53	No. Proses	9.5
	Nama Proses	verifikasi daftar baru
	Sumber	pasien
	Input	data pendaftaran
	Output	info daftar baru, data pasien
	Deskripsi	verifikasi data pedaftaran pasien baru
	Logika Proses	IF data pendaftaran terverifikasi THEN lanjutkan ke proses berikutnya ELSE daftar pasien *invalid*
54	No. Proses	9.6
	Nama Proses	*random captcha*
	Sumber	sistem
	Input	*request*
	Output	*captcha*
	Deskripsi	menghasilkan kode captcha sebagai kode

		verifikasi sebelum masuk ke halaman pasien

	Logika Proses	/* menampilkan kode dan gambar yang dibuat secara acak*/
		$captcha_image = imagecreatefrompng("images/captcha.png");
		$captcha_font = imageloadfont("images/font.gdf");
		$captcha_text = substr(md5(uniqid("")),-6,6);
		
		$_SESSION['captcha_session'] = $captcha_text;
		
		$captcha_color = imagecolorallocate($captcha_image,0,0,0);
		imagepng($captcha_image);
		imagedestroy($captcha_image);

55	No. Proses	9.6.1
	Nama Proses	gambar *backgraound captcha*
	Sumber	sistem
	Input	gambar
	Output	info gambar
	Deskripsi	menghasilkan gambar *background*
	Logika Proses	$captcha_image = imagecreatefrompng("images/captcha.png");
56	No. Proses	9.6.2
	Nama Proses	tentukan huruf
	Sumber	sistem
	Input	huruf
	Output	info info huruf
	Deskripsi	menghasilkan jenis huruf yang digunakan untuk menampilkan *captcha*
	Logika Proses	$captcha_font = imageloadfont("images/font.gdf");
57	No. Proses	9.6.3
	Nama Proses	*random text captcha*
	Sumber	sistem
	Input	*random*
	Output	info *random*
	Deskripsi	menghasilkan jenis huruf yang digunakan untuk menampilkan *captcha*
	Logika Proses	$captcha_text = substr(md5(uniqid("")),-6,6); $_SESSION['captcha_session'] = $captcha_text;
58	No. Proses	9.6.4
	Nama Proses	tampil
	Sumber	sistem

	Input	info *request*
	Output	*captcha*
	Deskripsi	menghasilkan kode verifikasi *captcha*
	Logika Proses	
59	No. Proses	9.6.5
	Nama Proses	hapus *text captcha* sebelumnya
	Sumber	sistem
	Input	hapus
	Output	info hapus
	Deskripsi	menghapus kode *captcha* sebelumnya
	Logika Proses	imagedestroy($captcha_image);
60	No. Proses	10.1
	Nama Proses	pilih penyakit yang akan didiagnosa
	Sumber	pasien
	Input	data diagnosa pasien, info data penyakit, info data pasien
	Output	data diagnosa, data penyakit, data pasien
	Deskripsi	menampilkan halaman awal diagnosa dan menampilkan jenis-jenis penyakit yang ada
	Logika Proses	pasien memilih data penyakit yang akan didiagnosa
61	No. Proses	10.2
	Nama Proses	jawab pertanyaan
	Sumber	pasien
	Input	info data diagnosa
	Output	data penyakit, data gejala
	Deskripsi	menampilkan halaman diagnosa dan menampilkan pertanyaan berdasarkan gejala
	Logika Proses	pasien menjawab pertanyaaan yang diberikan. Pertanyaan yang diberikan berdasarkan data penyakit yang telah dipilih
62	No. Proses	10.3
	Nama Proses	hasil
	Sumber	pasien
	Input	info data diagnosa, info data penyakit, info data gejala
	Output	data diagnosa, info data diagnosa pasien
	Deskripsi	menampilkan halaman hasil diagnosa
	Logika Proses	IF data diagnosa pasien = salah satu data penyakit yang ada di dalam *database* THEN penyakit diketahui (terdiagnosa) ELSE penyakit tidak diketahui
63	No. Proses	10.4

	Nama Proses	konfirmasi
	Sumber	pasien
	Input	data diagnosa
	Output	data diagnosa
	Deskripsi	menampilkan halaman diagnosa
	Logika Proses	
64	No. Proses	10.5
	Nama Proses	simpan daftar pertanyaan
	Sumber	pasien
	Input	data diagnosa, info data gejala
	Output	data diagnosa, data gejala
	Deskripsi	menyimpan daftar pertanyaan (data gejala) yang terkait dengan penyakit yang dipilih untuk didiagnosa ke tmp_analisa

	Logika Proses	function AddTmpAnalisa($kdgejala, $ip, $idpas) {
		$kdpenyakit=addslashes($_GET['kdpenyakit']);
		$sql_sakit = "SELECT relasi.kd_gejala, nm_gejala, pertanyaan
		FROM gejala, relasi
		WHERE gejala.kd_gejala = relasi.kd_gejalaAND relasi.kd_penyakit = '$kdpenyakit' order by relasi.kd_gejala ASC";
		$qry_sakit = mysql_query($sql_sakit);
		while ($data_sakit = mysql_fetch_array($qry_sakit)) {
		$sqltmp = "INSERT INTO tmp_analisa (id_pasien, noip, kd_penyakit,kd_gejala)
		VALUES ('$idpas', '$ip','$kdpenyakit','$data_sakit[kd_gejala]')";
		mysql_query($sqltmp);
		}
		}

65	No. Proses	10.6
	Nama Proses	pilih daftar pertanyaan suatu penyakit
	Sumber	pasien
	Input	data diagnosa, info data relasi
	Output	data diagnosa, data gejala
	Deskripsi	memilih gejala berdasarkan penyakit yang akan didiagnosa
	Logika Proses	SELECT relasi.kd_gejala, nm_gejala, pertanyaan FROM gejala, relasi WHERE gejala.kd_gejala = relasi.kd_gejala

		AND relasi.kd_penyakit = '$kdpenyakit' order by relasi.kd_gejala ASC
66	No. Proses	11.1
	Nama Proses	kirim pesan
	Sumber	pasien, pakar
	Input	data pesan, info data pasien
	Output	data pesan, info pengolahan data pesan
	Deskripsi	menampilkan halaman daftar pesan
	Logika Proses	Pasien mengirim pesan, kemudian pakar menjawab pesan tersebut
67	No. Proses	11.2
	Nama Proses	hapus Pesan
	Sumber	pakar, pakar
	Input	data pesan
	Output	data pesan, info pengolahan data pesan
	Deskripsi	menampilkan halaman daftar data pesan
	Logika Proses	/*Pakar atau pasien menghapus data pesan*/ IF hapus data tips dan artikel kesehatan *valid* THEN hapus data pesan ELSE hapus data pesan gagal
68	No. Proses	11.3
	Nama Proses	tampil
	Sumber	pakar, pasien
	Input	info data pesan, info data pencarian
	Output	data pencarian, info data pesan
	Deskripsi	menampilkan halaman data pesan
	Logika Proses	IF ada data pesan di dalam *database* THEN tampil daftar data pesan sesuai id_pasien ELSE gagal tampil
69	No. Proses	11.4
	Nama Proses	cari
	Sumber	pakar
	Input	data pesan, data pasien
	Output	info data pencarian
	Deskripsi	menampilkan halaman daftar data pesan yang dicari
	Logika Proses	/* cari data pesan berdasarkan data pesan dan data pasien yang ada di dalam *database**/

		IF ada data pesan di dalam *database* THEN tampil daftar data pesan ELSE data pesan yang dicari tidak ada

3.6.15 Kamus Data

Kamus data dapat mendefinisikan dengan lengkap data yang mengalir diantara proses maupun penyimpanan data. Data yang mengalir tersebut dapat berupa masukan untuk sistem atau hasil dari proses sistem. Adapun kamus data untuk sistem ini adalah sebagai berikut :

Tabel 3.7 Kamus Data

Nama	**data pakar**
Where Used/Who used	Pakar – Proses 8 “proses pengolahan data pakar”
Deskripsi	Pakar adalah seseorang yang memiliki hak akses sebagai admin dalam sistem ini
id_pakar	{0..9}
username	{a..z | A..Z | 0..9}
nama_pakar	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}
alamat	{a..z | A..Z | 0..9}
foto	{a..z | A..Z | 0..9}
tgl_login	{0..9}
jam	{0..9}
Nama	**data pasien**
Where Used/Who used	Pasien – Proses 5 ”proses pengolahan data pasien”
Deskripsi	Pasien merupakan pengguna sistem yang melakukan proses pendaftaran dan konsultasi
id_pasien	{0..9}
password	{a..z | A..Z | 0..9}
nama	{a..z | A..Z | 0..9}
jenis_kelamin	{a..z | A..Z }
pekerjaan	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}
alamat	{a..z | A..Z | 0..9}
kota	{a..z | A..Z | 0..9}
provinsi	{a..z | A..Z | 0..9}
noip	{0..9}
tanggal	{0..9}

tgl_login	{0..9}
jam	{0..9}

Nama	**data gejala**
Where Used/Who used	Proses 3 ”proses pengolahan data gejala”, proses 4 ”proses pengolahan data relasi”, proses 10 ”diagnosa”
Deskripsi	Data klinis mengenai gejala suatu penyakit
kd_gejala	{0..9}
nm_gejala	{a..z | A..Z | 0..9}
pertanyaan	{a..z | A..Z | 0..9}
skorgejala	[0..9]
keterangan	{a..z | A..Z | 0..9}
Nama	**data istilah**
Where Used/Who used	Proses 6 “proses pengolahan data istilah”
Deskripsi	Data istilah yang terkait dengan data gejala maupun data penyakit
id	{0..9}
id_pakar	{0..9}
nama_istilah	{a..z | A..Z | 0..9}
keterangan	{a..z | A..Z | 0..9}
tanggal	{0..9}
Nama	**data penyakit**
Where Used/Who used	Proses 2 “proses pengolahan data penyakit”, proses 4 “proses pengolahan data relasi”, proses 10 “diagnosa”
Deskripsi	Salah satu komponen basis pengetahuan, yang akan menjadi hasil diagnosa
kd_penyakit	{0..9}
nm_penyakit	{a..z | A..Z | 0..9}
definisi	{a..z | A..Z | 0..9}
anjuran	{a..z | A..Z | 0..9}
terapi	{a..z | A..Z | 0..9}
skorpenyakit	{0..9}
tanggal	{0..9}
Nama	**data pesan**
Where Used/Who used	Proses 11 “Proses pengolahan pesan”
Deskripsi	Data pesan merupakan jembatan komunikasi antara pakar dan pasien
id_pesan	{0..9}
id_pasien	{0..9}
id_pakar	{0..9}
pesan	{a..z | A..Z | 0..9}
jawaban	{a..z | A..Z | 0..9}
tanggal_jawab	{0..9}
jam_jawab	{0..9}
tanggal_kirim_user	{0..9}

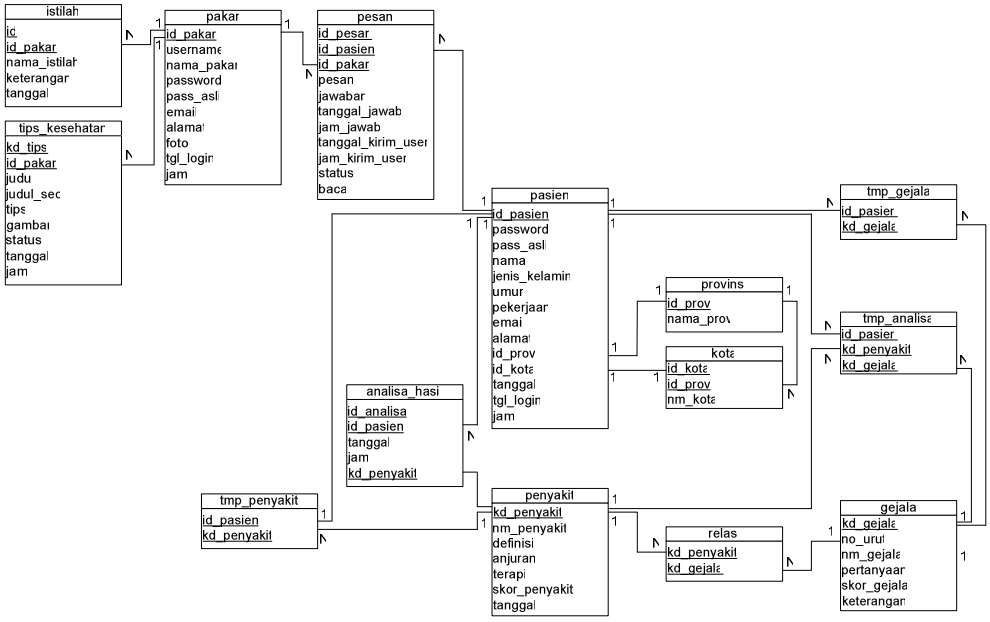
jam_kirim_user	{0..9}
status	{a..z | A..Z | 0..9}
baca	{a..z | A..Z | 0..9}
Nama	**data relasi**
Where Used/Who used	Proses 4 ”proses pengolahan data relasi”, proses 10 ”diagnosa”
Deskripsi	Data relasi merupakan suatu aturan atau keterkaitan antara data gejala dengan data penyakit
kd_penyakit	{0..9}
kd_gejala	{0..9}
Nama	**login pakar**
Where Used/Who used	Pakar - Proses 1 “proses *login* pakar”
Deskripsi	*Login* pakar merupakan data yang diverifikasi pada saat proses *login* pakar dilakukan
username	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
Nama	**data login**
Where Used/Who used	Pakar - Proses 1 “proses *login* pakar”
Deskripsi	*Login* pakar merupakan data yang diverifikasi pada saat proses *login* pakar dilakukan
username	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
Nama	**data tips dan artikel**
Where Used/Who used	Pakar - Proses 4 “proses pengolahan data tips dan artikel kesehatan”
Deskripsi	Data tips dan artikel merupakan data yang terkait pembahasan bidang kesehatan yang dikelola oleh admin (pakar)
kd_tips	{0..9}
id_pakar	{0..9}
judul	{a..z | A..Z | 0..9}
tips	{a..z | A..Z | 0..9}
gambar	{a..z | A..Z | 0..9}
status	[P | Y | T]
tanggal	{0..9}
jam	{0..9}
Nama	**data diagnosa pasien**
Where Used/Who used	Pasien - Proses 10 “Diagnosa”
Deskripsi	Data diagnosa merupakan data yang didapat dari proses diagnosa
kd_gejala	{0..9}
kd_penyakit	{0..9}
id_pasien	{0..9}
noip	{0..9}
Nama	**data diagnosa**
Where Used/Who used	Pasien - Proses 10 “Diagnosa”

Deskripsi	Data diagnosa merupakan data yang didapat dari proses diagnosa
kd_gejala	{0..9}
kd_penyakit	{0..9}
id_pasien	{0..9}
noip	{0..9}
Nama	**data konfigurasi pakar**
Where Used/Who used	Pasien – Proses 8 ”proses pengolahan data pakar”
Deskripsi	Data konfigurasi merupakan data yang diolah pada proses pengolahan data pakar
username	{a..z | A..Z | 0..9}
nama_pakar	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}
gambar	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
Nama	**daftar pasien**
Where Used/Who used	Pasien - Proses 9 ”Daftar pasien”
Deskripsi	Daftar pasien merupakan data yang didapat dari proses daftar pasien yang dikelompokkan menjadi daftar dan daftar baru.
id_pasien	{0..9}
username	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
nama	{a..z | A..Z | 0..9}
jenis kelamin	{a..z | A..Z | 0..9}
umur	{0..9}
pekerjaan	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}
kota	{a..z | A..Z | 0..9}
provinsi	{a..z | A..Z | 0..9}
noip	{0..9}
tanggal	{0..9}
tgl_login	{0..9}
jam	{0..9}
Nama	**data pasien baru**
Where Used/Who used	Pasien - Proses 9 ”Daftar pasien”
Deskripsi	Daftar pasien merupakan data yang didapat dari proses daftar pasien yang dikelompokkan menjadi daftar dan daftar baru.
id_pasien	{0..9}
username	{a..z | A..Z | 0..9}
password	{a..z | A..Z | 0..9}
nama	{a..z | A..Z | 0..9}
jenis kelamin	{a..z | A..Z | 0..9}
umur	{0..9}
pekerjaan	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}

kota	{a..z | A..Z | 0..9}
provinsi	{a..z | A..Z | 0..9}
noip	{0..9}
tanggal	{0..9}
tgl_login	{0..9}
jam	{0..9}
Nama	**data pasien lama**
Where Used/Who used	Pasien - Proses 9 ”Daftar pasien”
Deskripsi	Daftar pasien merupakan data yang didapat dari proses daftar pasien yang dikelompokkan menjadi daftar dan daftar baru.
password	{a..z | A..Z | 0..9}
email	{a..z | A..Z | 0..9}

3.6.16 Skema Relasi

Skema relasi merupakan keterkaitan antartabel yang dihubungkan dengan kunci utama (*primary key*), sehingga tabel-tabel tersebut menjadi satu-kesatuan. Skema relasi Sistem Pakar Untuk Mendiagnosa Penyakit Vertigo Dengan Metode Backward Chaining Berbasis Web dapat dilihat pada gambar 3.16 sebagai berikut:



Gambar 3.16 Skema Relasi

3.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional ini meliputi analisis kebutuhan pengguna (*user*), perangkat keras (*hardware*), dan perangkat lunak (*software*) yang digunakan dalam membangun sistem.

3.7.1 Analisis Kebutuhan Pengguna (*User*)

Pengguna (*user*) sistem pakar ini ialah administrator dan pengguna biasa. Administrator yaitu seseorang yang memiliki keahlian (pakar) dalam bidang kedokteran sehingga mampu memelihara isi (*content*) sistem berdasarkan pengetahuan yang dimilikinya, dalam hal ini adalah dokter spesialis saraf. Sedangkan pengguna biasa yaitu masyarakat umum (pasien) sebagai pengakses informasi dan fasilitas sistem pakar ini. Adapun Spesifikasi pengguna sistem pakar ini adalah sebagai berikut :

Tabel 3.8 Karakteristik Pengguna

Pengguna	Usia (tahun)	HakAkses	Pendidikan	Pengalaman
Administrator (pakar)	25-50	Akses informasi, login sebagai pakar, dan manipulasi (tambah, ubah, dan hapus) data di dalam sistem.	Kedokteran (dokter spesialis saraf).	Mampu mengoperasikan komputer, pernah mengakses, dan memelihara sistem berbasis web
Masyarakat (pasien)	13-60	Akses informasi, daftar pasien, login sebagai pasien, dan konsultasi	Minimal Sekolah Menengah Pertama (SMP)	Mampu mengoperasikan komputer dan pernah mengakses sistem berbasis web

3.7.2 Analisis Kebutuhan Perangkat Keras (Hardware)

Spesifikasi perangkat keras yang dibutuhkan antara lain sebagai berikut :

1. Monitor 14" dengan resolusi layer 1024x768 pixels.
2. Kapasitas hardisk (*free memory*) 2 GB.
3. RAM 512 MB
4. *Processor* Pentium 4.

3.7.3 Analisis Kebutuhan Perangkat Lunak (Software)

Spesifikasi perangkat lunak yang dibutuhkan antara lain sebagai berikut :

1. Untuk membangun sistem
    1) Sistem operasi Microsoft Windows XP Service Pack 2
    2) Wampp 2.0
    3) Dreamwaver MX 2004
    4) Photoshop CS 2
    5) Mozilla FireFox 3.0
    6) Bahasa pemrograman PHP
    7) Database MySQL
    8) Bahasa pemrograman JavaScript
2. Untuk mengakses sistem
    1) Sistem operasi Microsoft Windows
    2) *Software browser* Mozilla FireFox

3.8 Struktur Tabel

Tabel-tabel yang digunakan dalam membangun Sistem Pakar Untuk Mendiagnosa Penyakit Vertigo Dengan Metode Backward Chaining Berbasis Web ini terdiri atas tabel analisa_hasil, tabel counter, tabel gejala, tabel istilah, tabel pakar, tabel pasien, tabel pesan, tabel penyakit, tabel relasi, tabel tips_kesehatan, tabel tmp_analisa, tabel tmp_gejala dan tabel tmp_penyakit. Adapun struktur tabelnya adalah sebagai berikut :

Tabel 3.9 Tabel analisis_hasil

Nama Field	Tipe Data	Atribut	Deskripsi
id_analisa	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY REFERENCES pasien (id_pasien) ON DELETE CASCADE
tanggal	DATE	NOT NULL	
jam	TIME	NOT NULL	
kd_penyakit	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_penyakit) REFERENCES penyakit(kd_penyakit) ON DELETE CASCADE

Tabel 3.10 Tabel counter

Nama Field	Tipe Data	Atribut	Deskripsi
ip	VARCHAR (60)	NOT NULL	
tanggal	DATE	NOT NULL	
hits	INTEGER (10)	UNSIGNED NOT NULL	
online	VARCHAR (2550\)	NOT NULL	

Tabel 3.11 Tabel gejala

Nama Field	Tipe Data	Atribut	Deskripsi
kd_gejala	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTOINCREMENT	PRIMARY KEY
no_urut	INTEGER (4)	UNSIGNED NOT NULL	
nm_gejala	VARCHAR (100)	NOT NULL	
pertanyaan	VARCHAR (200)	NOT NULL	
skorgejala	INTEGER (1)	UNSIGNED NOT NULL DEFAULT 1	
keterangan	TEXT	NOT NULL	

Tabel 3.12 Tabel istilah

Nama Field	Tipe Data	Atribut	Deskripsi
id	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
nama_istilah	VARCHAR (30)	NOT NULL	
keterangan	TEXT	NOT NULL	
tanggal	DATE	NOT NULL	
id_pakar	INTEGER (2)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_pakar) REFERENCES pakar(id_pakar) ON DELETE CASCADE

Tabel 3.13 Tabel pakar

Nama Field	Tipe Data	Atribut	Deskripsi
id_pakar	INTEGER (2)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
*user*name	VARCHAR (30)	NOT NULL UNIQUE	

nama_pakar	VARCHAR (40)	NOT NULL	
password	VARCHAR (60)	NOT NULL UNIQUE	
pass_asli	VARCHAR (60)	NOT NULL UNIQUE	
email	VARCHAR (80)	NOT NULL UNIQUE	
alamat	VARCHAR (150)	NOT NULL	
foto	VARCHAR (150)	NOT NULL DEFAULT 'KOSONG'	
tgl_login	DATE	NOT NULL	
jam	TIME	NOT NULL	

Tabel 3.14 Tabel pasien

Nama Field	Tipe Data	Atribut	Deskripsi
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
password	VARCHAR (60)	NOT NULL UNIQUE	
nama	VARCHAR (60)	NOT NULL	
jenis_kelamin	ENUM ('Laki-laki','Perempuan')	NOT NULL	
email	VARCHAR (80)	NOT NULL UNIQUE	
alamat	VARCHAR (150)	NOT NULL	
kota	VARCHAR (20)	NOT NULL	
provinsi	VARCHAR (30)	NOT NULL	
tanggal	DATE	NOT NULL	
tgl_login	DATE	NOT NULL	
jam	TIME	NOT NULL	

Tabel 3.15 Tabel penyakit

Nama Field	Tipe Data	Atribut	Deskripsi
kd_penyakit	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTOINCREMENT	PRIMARY KEY
nm_penyakit	VARCHAR (80)	NOT NULL	
definisi	TEXT	NOT NULL	

anjuran	TEXT	NOT NULL	
terapi	TEXT	NOT NULL	
skorpenyakit	INTEGER (4)	UNSIGNED NOT NULL	
tanggal	DATE		

Tabel 3.16 Tabel pesan

Nama Field	Tipe Data	Atribut	Deskripsi
id_pesan	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_pasien) REFERENCES pasien (id_pasien) ON DELETE CASCADE
id_pakar	INTEGER (2)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_pakar) REFERENCES pakar (id_pakar) ON DELETE CASCADE
pesan	TEXT	NOT NULL	
jawaban	TEXT	NOT NULL	
tanggal_jawab	DATE	NOT NULL	
jam_jawab	TIME	NOT NULL	
tanggal_kirim_*user*	DATE	NOT NULL	
jam_kirim_*user*	TIME	NOT NULL	
status	VARCHAR (15)	NOT NULL DEFAULT ‘belum dijawab’	
baca	VARCHAR (15)	NOT NULL DEFAULT ‘belum dibaca’	

Tabel 3.17 Tabel relasi

Nama Field	Tipe Data	Atribut	Deskripsi
kd_penyakit	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_penyakit) REFERENCES penyakit (kd_penyakit) ON DELETE CASCADE
kd_gejala	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_gejala) REFERENCES gejala (kd_gejala) ON DELETE CASCADE

Tabel 3.18 Tabel tips_kesehatan

Nama Field	Tipe Data	Atribut	Deskripsi
kd_tips	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
judul	VARCHAR (100)	NOT NULL	
judul_seo	VARCHAR (150)	NOT NULL	
tips	TEXT	NOT NULL	
gambar	VARCHAR (150)	NOT NULL	
status	ENUM ('P','Y','T')	NOT NULL	
tanggal	DATE	NOT NULL	
jam	TIME	NOT NULL	
id_pakar	INTEGER (2)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_pakar) REFERENCES pakar(id_pakar) ON DELETE CASCADE

Tabel 3.19 Tabel tmp_analisa

Nama Field	Tipe Data	Atribut	Deskripsi
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_pasien) REFERENCES pasien (id_pasien) ON DELETE CASCADE
kd_penyakit	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_penyakit) REFERENCES penyakit(kd_penyakit) ON DELETE CASCADE
kd_gejala	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_gejala) REFERENCES gejala(kd_gejala) ON DELETE CASCADE

Tabel 3.20 Tabel tmp_gejala

Nama Field	Tipe Data	Atribut	Deskripsi
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY REFERENCES tmp_analisa (id_pasien) ON DELETE CASCADE
kd_gejala	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_gejala) REFERENCES gejala(kd_gejala) ON DELETE CASCADE

Tabel 3.21 Tabel tmp_penyakit

Nama Field	Tipe Data	Atribut	Deskripsi
id_pasien	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY REFERENCES tmp_analisa (id_pasien) ON DELETE CASCADE
kd_penyakit	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (kd_penyakit) REFERENCES penyakit(kd_penyakit) ON DELETE CASCADE

Tabel 3.22 Tabel kota

Nama Field	Tipe Data	Atribut	Deskripsi
id_kota	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
id_prov	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL	FOREIGN KEY (id_prov) REFERENCES provinsi(id_prov) ON DELETE CASCADE
nm_kota	VARCHAR (150)	NOT NULL	

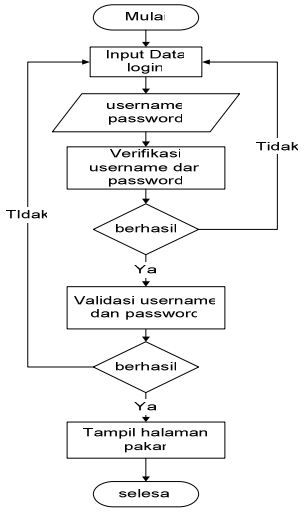
Tabel 3.23 Tabel provinsi

Nama Field	Tipe Data	Atribut	Deskripsi
id_prov	INTEGER (4)	UNSIGNED ZEROFILL NOT NULL AUTO INCREMENT	PRIMARY KEY
nama_prov	VARCHAR (150)	NOT NULL	

3.9 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

3.9.1 Flowchart Login Pakar

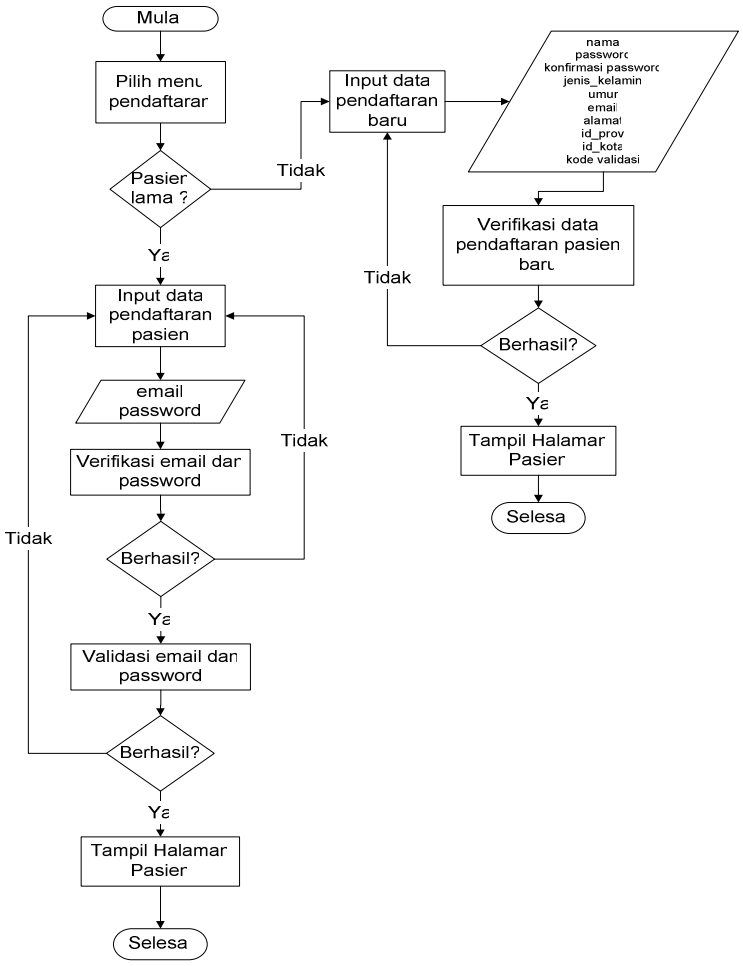


Gambar 3.17 *Flowchart Login* Pakar

Dari *flowchart* di atas, dapat dijelaskan langkah-langkah proses yang dilakukan adalah sebagai berikut :

Untuk mendapatkan hak akses sebagai pakar, *user* memulai prosesnya dengan memasukkan *username* dan *password* yang sesuai yang kemudian masukkan tersebut akan diverifikasi. Setelah *username* dan *password* terverifikasi, maka akan dilanjutkan dengan proses validasi dan jika *username* dan *password* tersebut valid maka akan dilanjutkan dengan proses menampilkan halaman pakar.

3.9.2 Flowchart Daftar Pasien



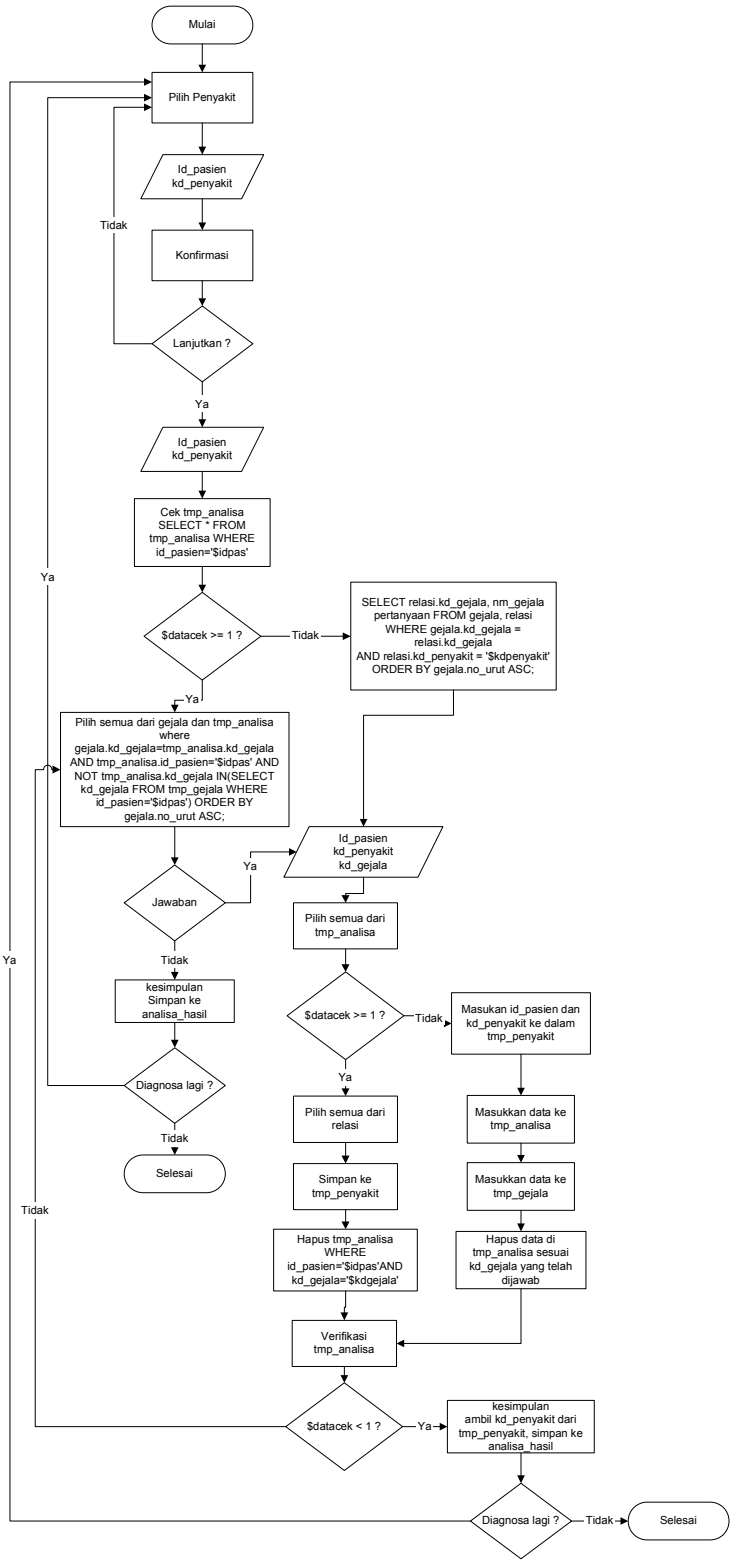
Gambar 3.18 *Flowchart* Daftar Pasien

Dari *flowchart* di atas, dapat dijelaskan langkah-langkah proses yang dilakukan adalah sebagai berikut :

Untuk memulai konsultasi, *user* diharuskan melakukan pendaftaran terlebih dahulu. Jika *user* telah memiliki akun, maka proses pendaftarannya dimulai dengan memasukkan *username* dan *password* yang sesuai yang kemudian masukkan tersebut akan diverifikasi. Setelah *username* dan *password* terverifikasi, maka akan dilanjutkan dengan proses validasi dan jika *username* dan *password* tersebut valid, maka akan dilanjutkan dengan proses menampilkan

halaman *user* (halaman konsultasi pasien). Sedangkan, jika *user* belum memiliki akun, maka proses pendaftarannya dilakukan dengan memasukkan data pendaftaran seperti nama, *password,* konfirmasi *password,* jenis kelamin, umur, alamat *email,* alamat, nama provinsi, nama kota, dan kode validasi *captcha* yang kemudian masukkan tersebut akan diverifikasi. Selanjutnya, data pendaftaran tersebut terverifikasi, maka akan dilanjutkan dengan proses validasi dan jika data pendaftaran tersebut valid, maka akan dilanjutkan dengan proses menampilkan halaman *user* (halaman konsultasi pasien).

3.9.3 Flowchart Diagnosa Penyakit



Gambar 3.19 *Flowchart* Diagnosa Penyakit

Dari *flowchart* di atas, dapat dijelaskan langkah-langkah proses yang dilakukan adalah sebagai berikut :

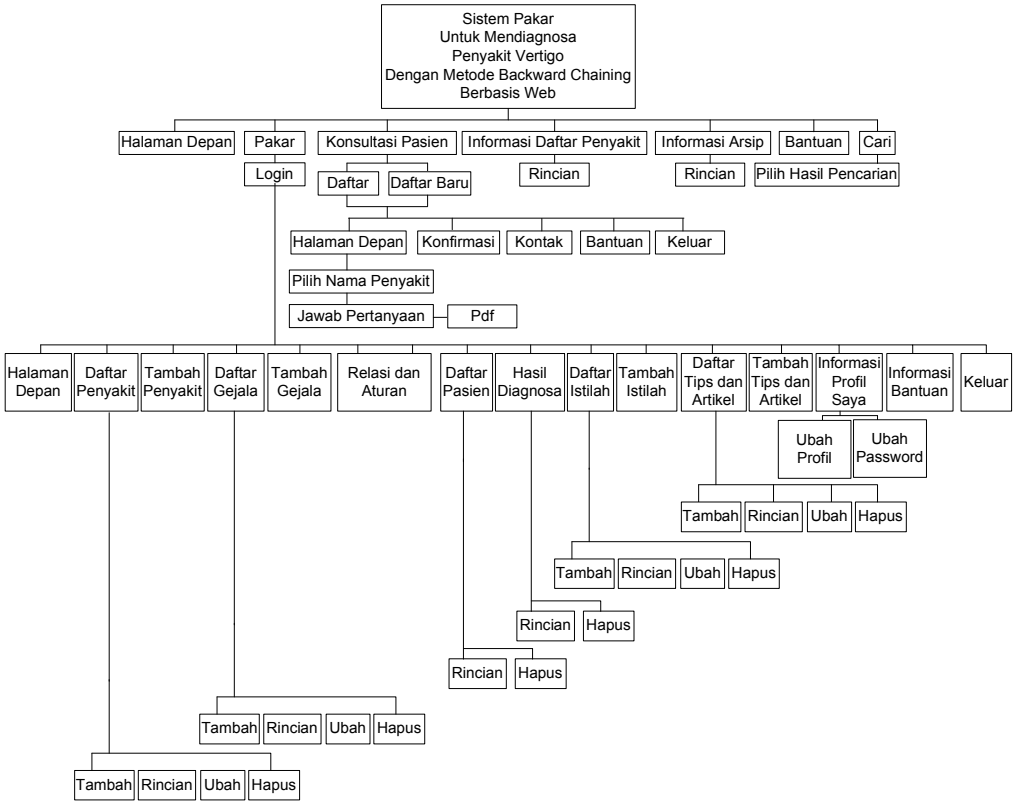
Proses diagnosa penyakit dimulai dengan pemilihan jenis penyakit yang dilakukan oleh *user.* Setelah itu, *user* diberikan pertanyaan konfirmasi untuk melanjutkan proses diagnosa. Jika *user* menjawab "Ya", maka proses diagnosa akan dilanjutkan, sedangkan jika menjawab "Tidak", maka proses akan kembali ke halaman depan *user* (pasien).

Di dalam sistem, tahap awal diagnosa dilakukan dengan memeriksa tabel tmp_analisa, jika sudah terdapat data dengan id *user* (pasien) yang menjawab, maka proses akan berlanjut dengan memilih semua pertanyaan yang ada di dalam tabel tmp_analisa, jika tidak terdapat data, maka proses akan berlanjut dengan memilih pertanyaan ke tabel relasi berdasarkan kd_penyakit yang dipilih pasien.

Pada sesi pertanyaan, jika pasien menjawab "Tidak" sebelum semua pertanyaan terjawab, maka proses diagnosa akan terhenti dan memberikan kesimpulan berdasarkan gejala yang telah dijawab. Dan jika pasien menjawab "Ya" hingga pertanyaan telah terjawab semua, maka sistem akan memberikan kesimpulan yang sesuai dengan penyakit yang dipilihnya pada awal sesi diagnosa.

3.10 Struktur Menu

Dalam perancangan sebuah program, dibutuhkan suatu struktur menu yang memaparkan mengenai menu dan submenu yang akan dibangun.

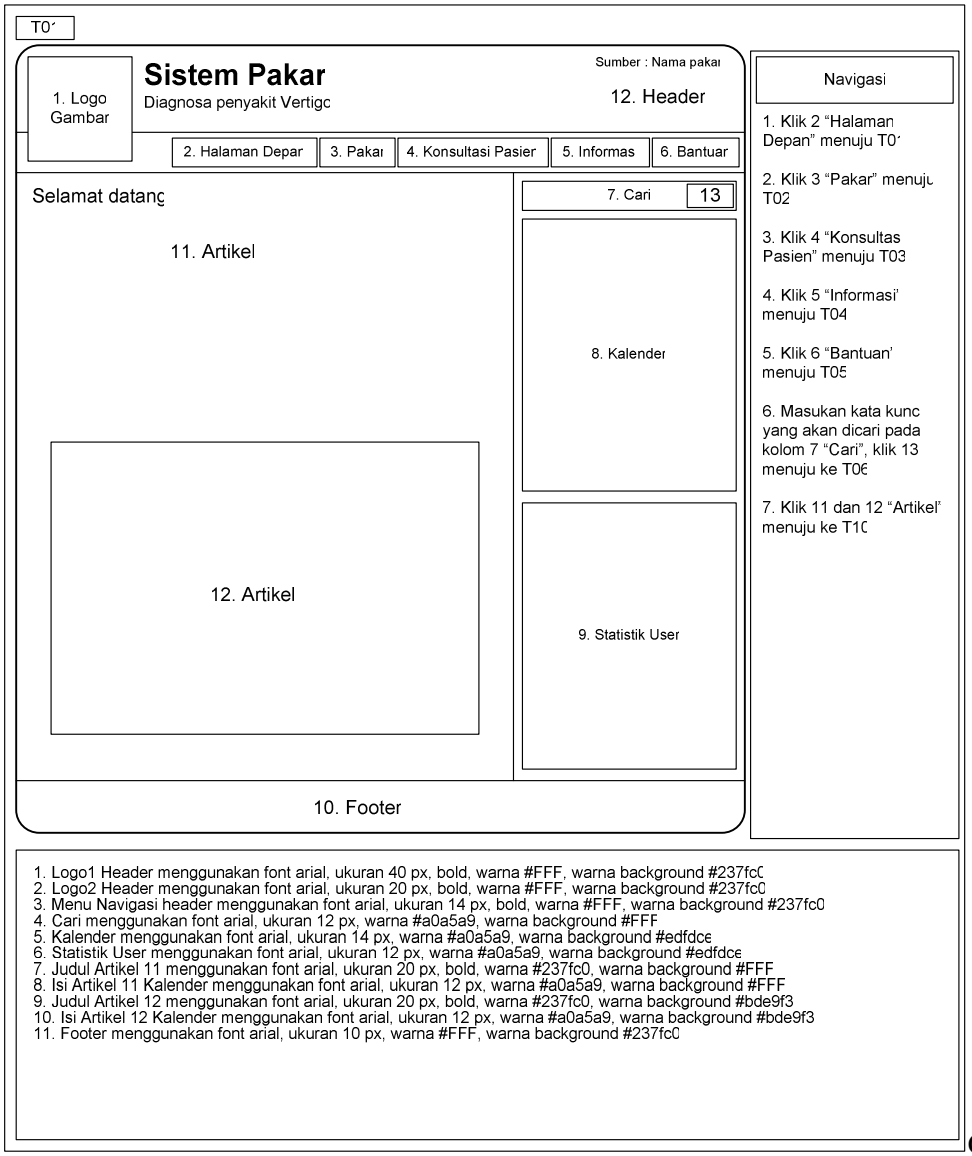


Gambar 3.20 Struktur Menu

3.11 Perancangan Antarmuka

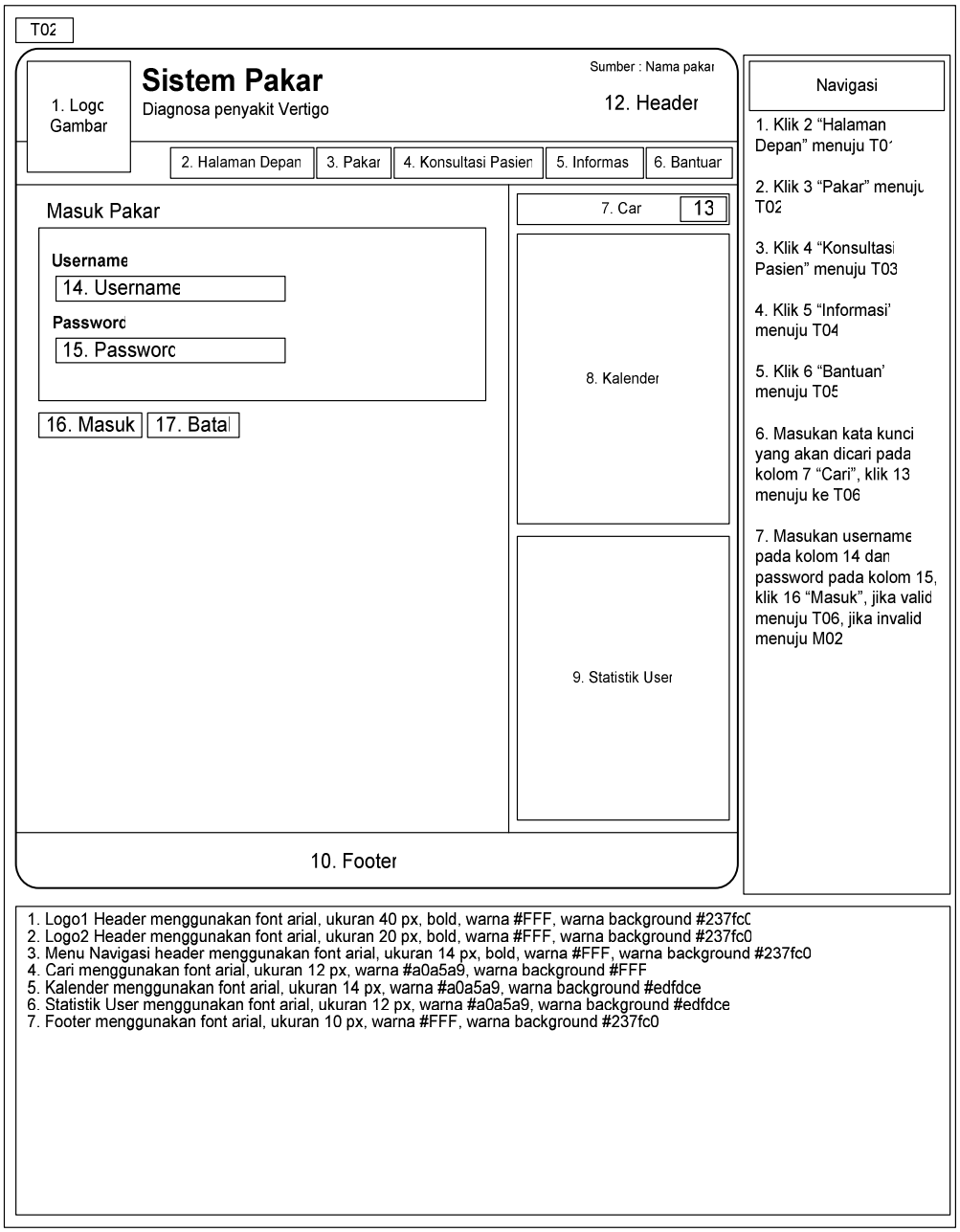
Antarmuka (*interface*) merupakan suatu media komunikasi antara *user* dengan aplikasi yang dibangun. Oleh karena itu, dalam perancangan sebuah aplikasi, mutlak dibutuhkan suatu perancangan antarmuka guna memberikan gambaran umum bagi *user*.

3.11.1 Perancangan Halaman Depan Aplikasi



Gambar 3.21 Perancangan Halaman Depan Aplikasi

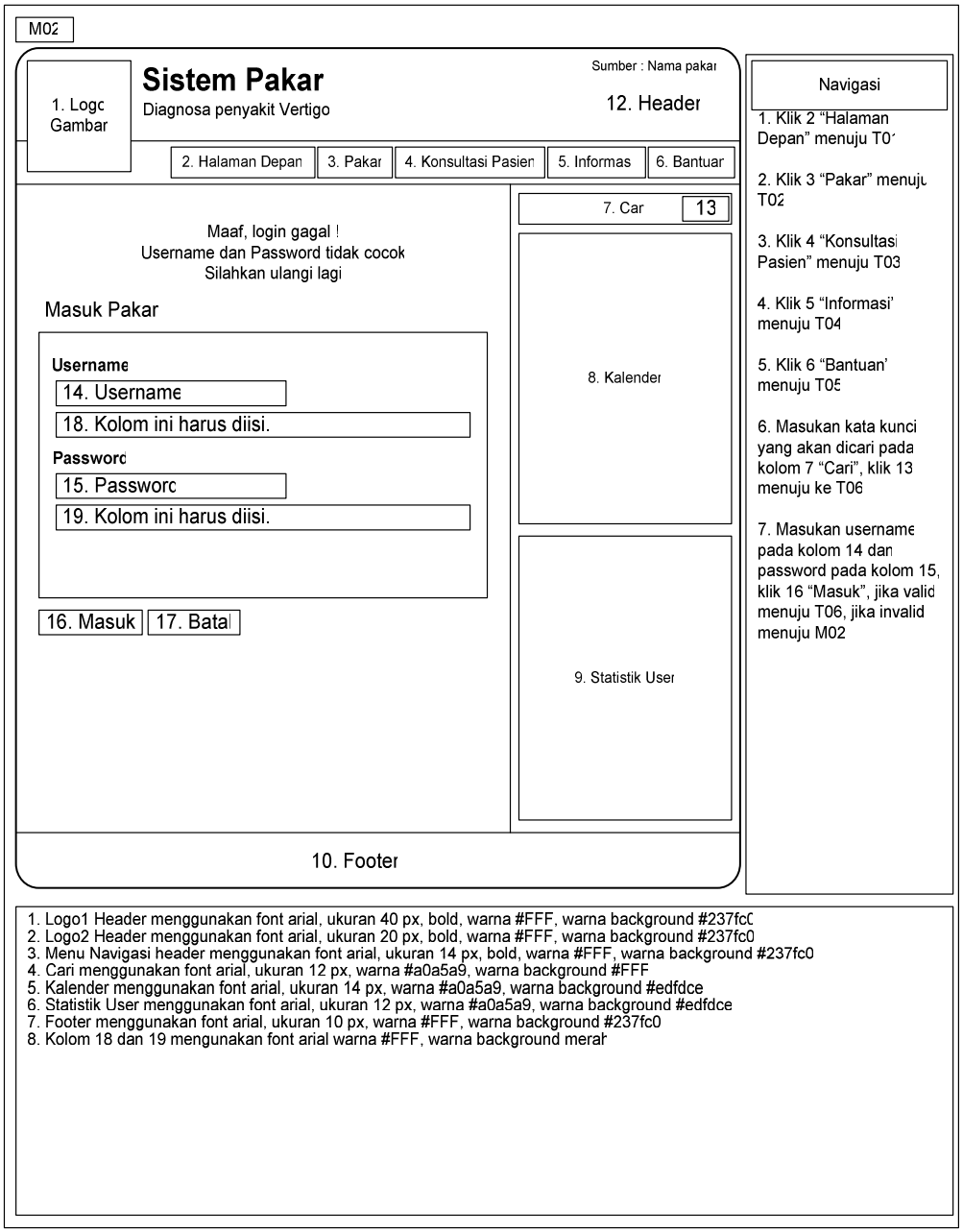
3.11.2 Perancangan Halaman Login Pakar



Gambar 3.22 Perancangan Halaman Login Pakar

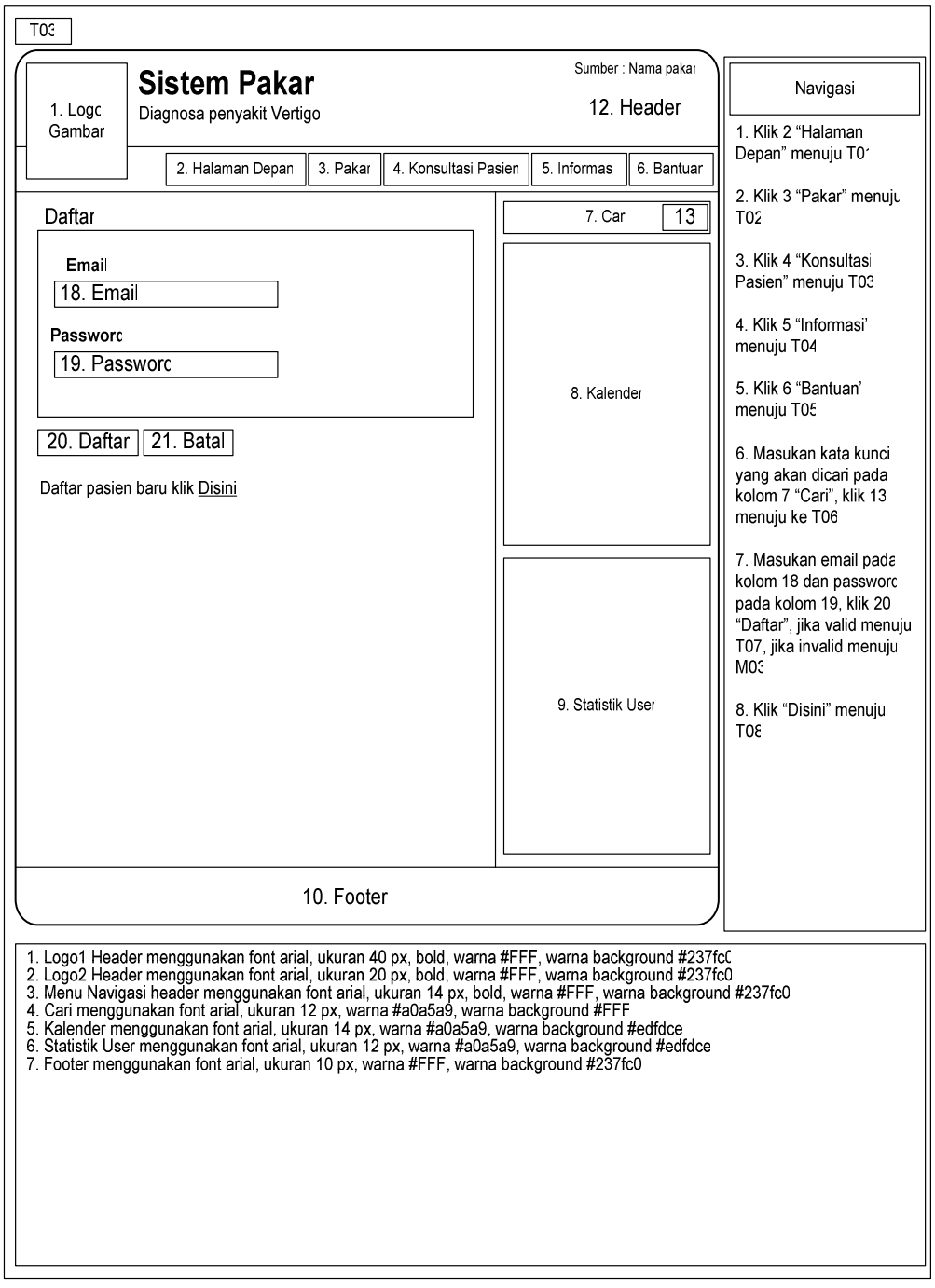
3.11.3 Perancangan Halaman Validasi Login Pakar

Kondisi dimana *user* salah memasukkan *username* dan *password* serta kolom *username* dan *password* tidak diisi.



Gambar 3.23 Perancangan Halaman Validasi Login Pakar

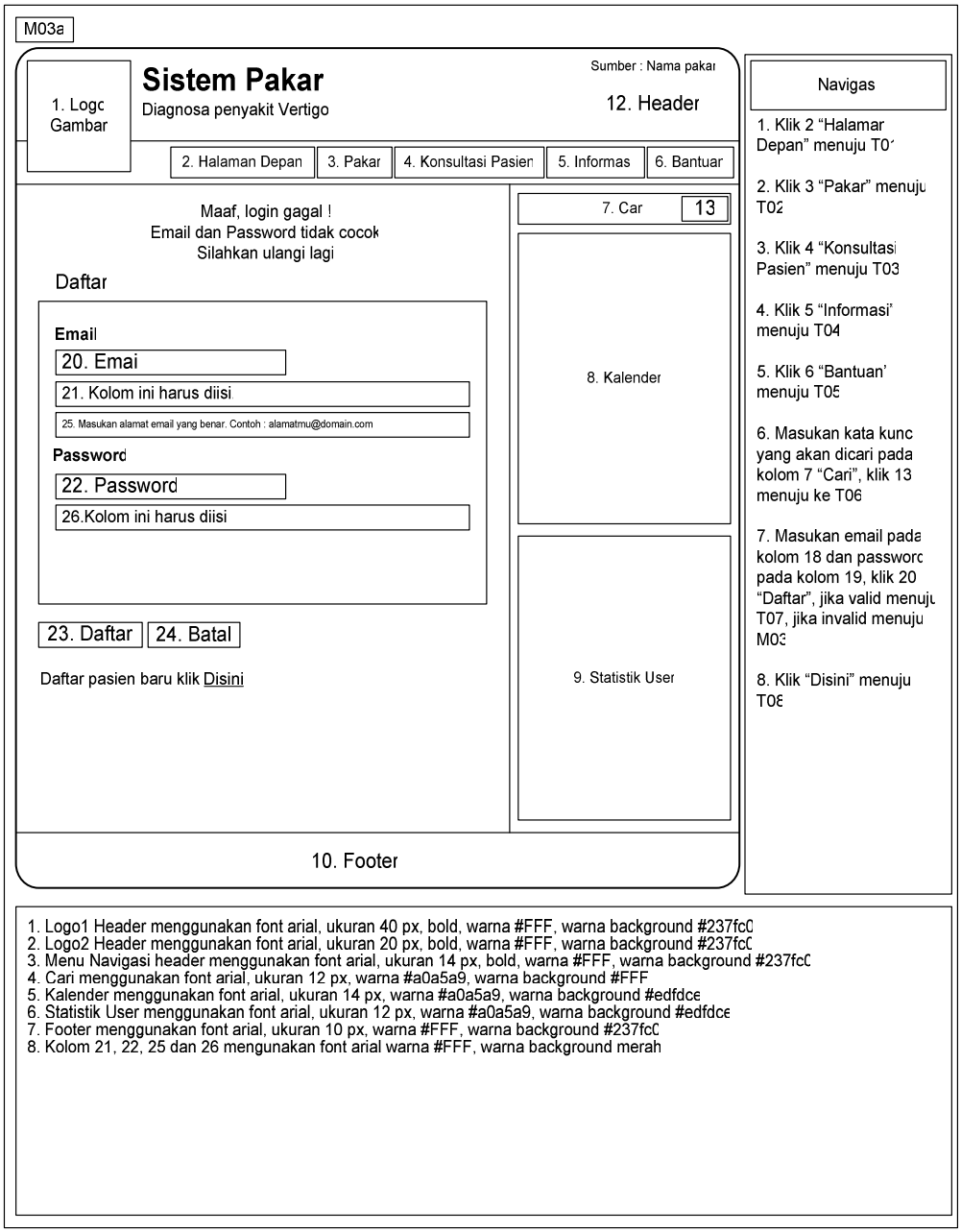
3.11.4 Perancangan Halaman Login Pasien



Gambar 3.24 Perancangan Halaman *Login* Pasien

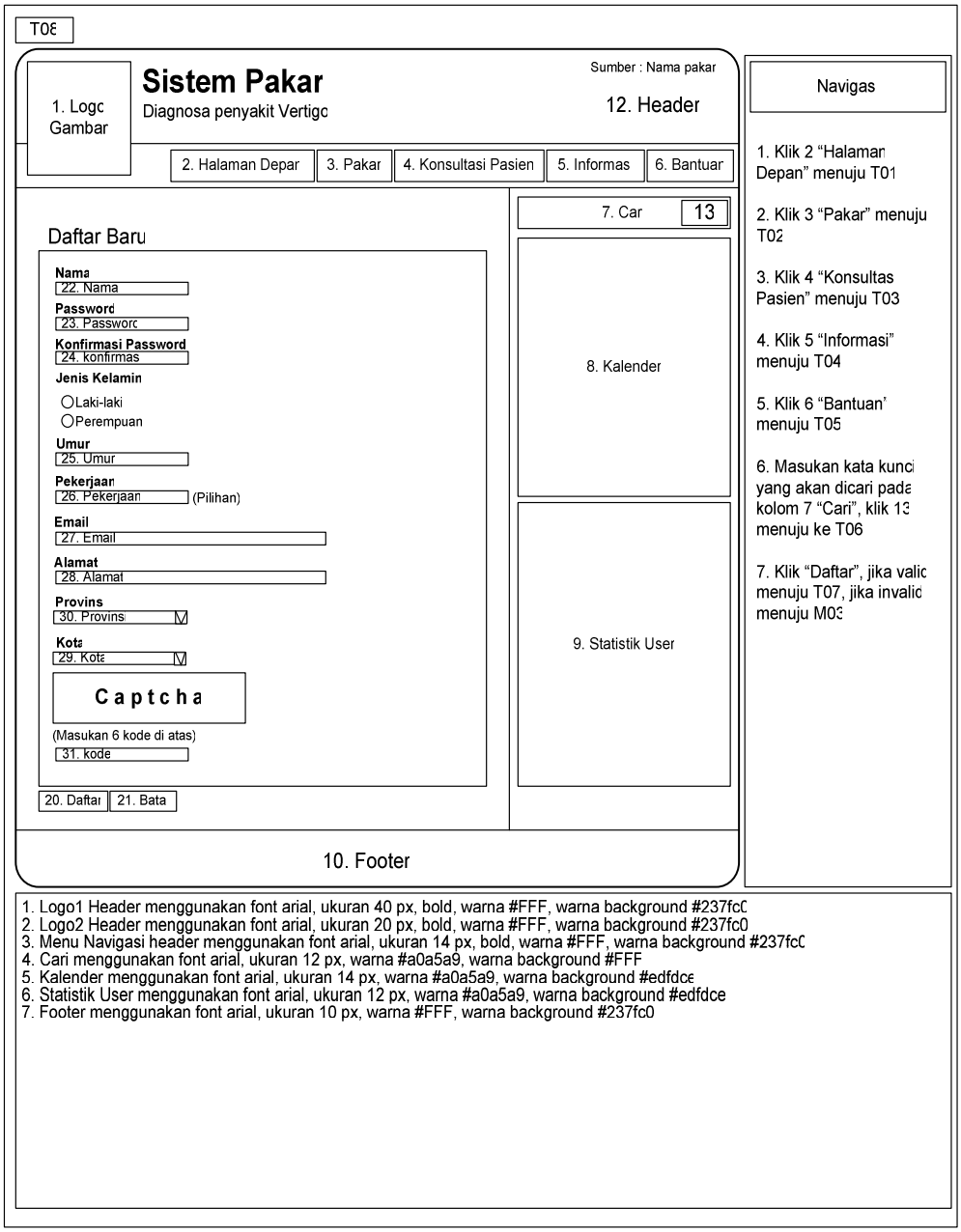
3.11.5 Perancangan Halaman Validasi Login Pasien

Kondisi dimana *user* salah memasukkan *email* dan *password* serta kolom *username* dan *password* tidak diisi.



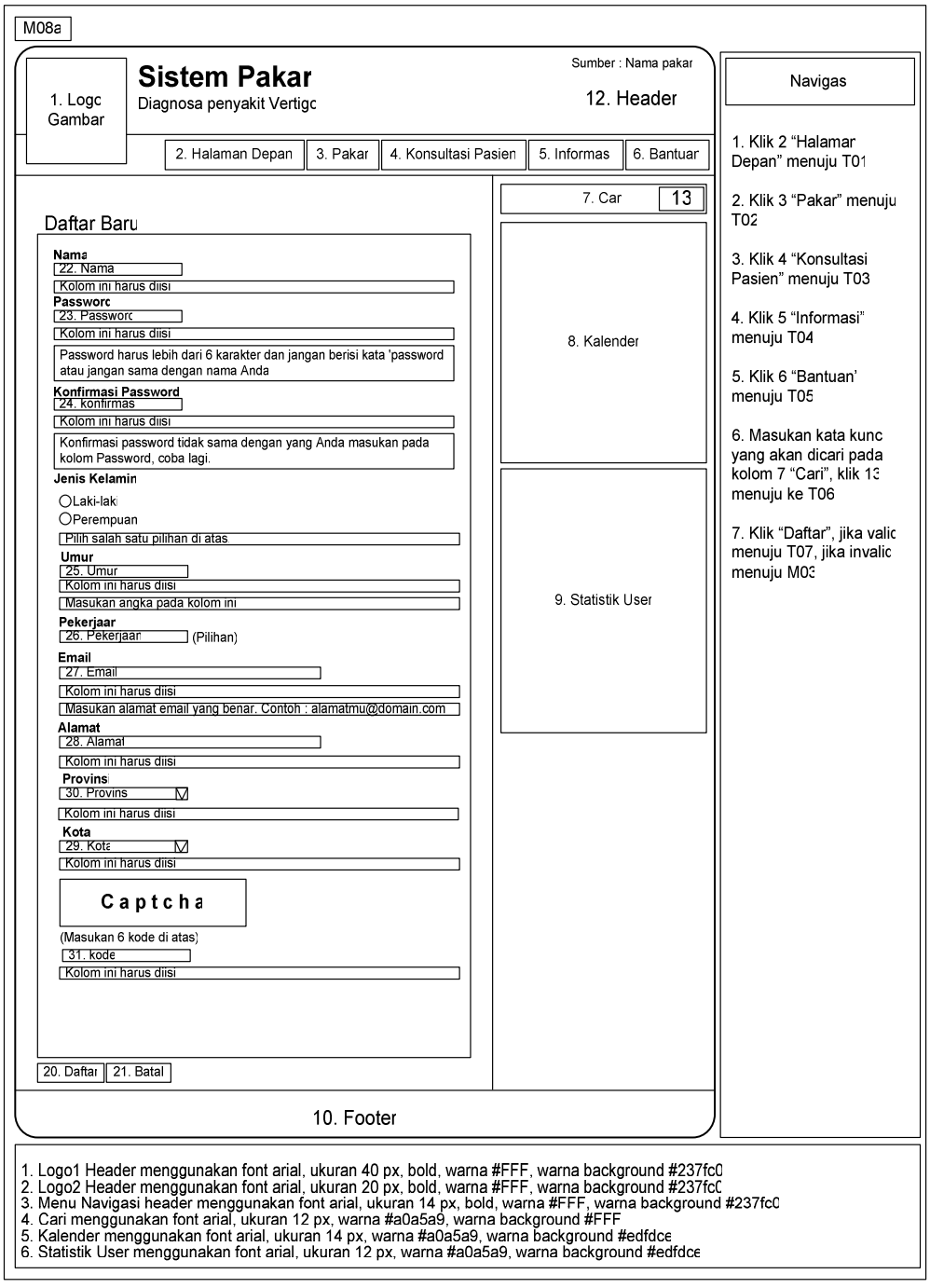
Gambar 3.25 Perancangan Halaman Validasi *Login* Pasien

3.11.6 Perancangan Halaman Daftar Pasien Baru

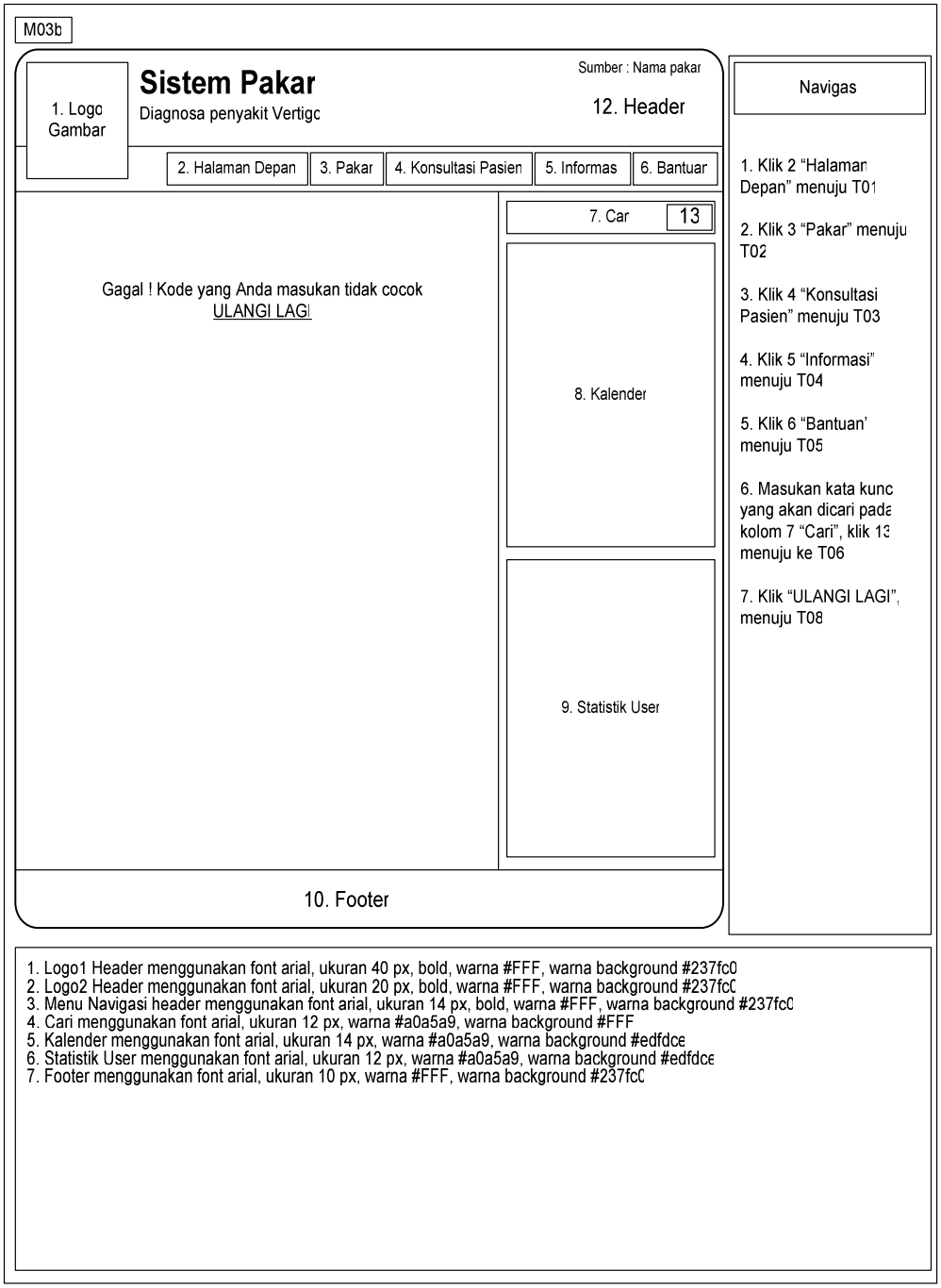


Gambar 3.26 Perancangan Halaman Daftar Pasien Baru

3.11.7 Perancangan Halaman Daftar Pasien Baru

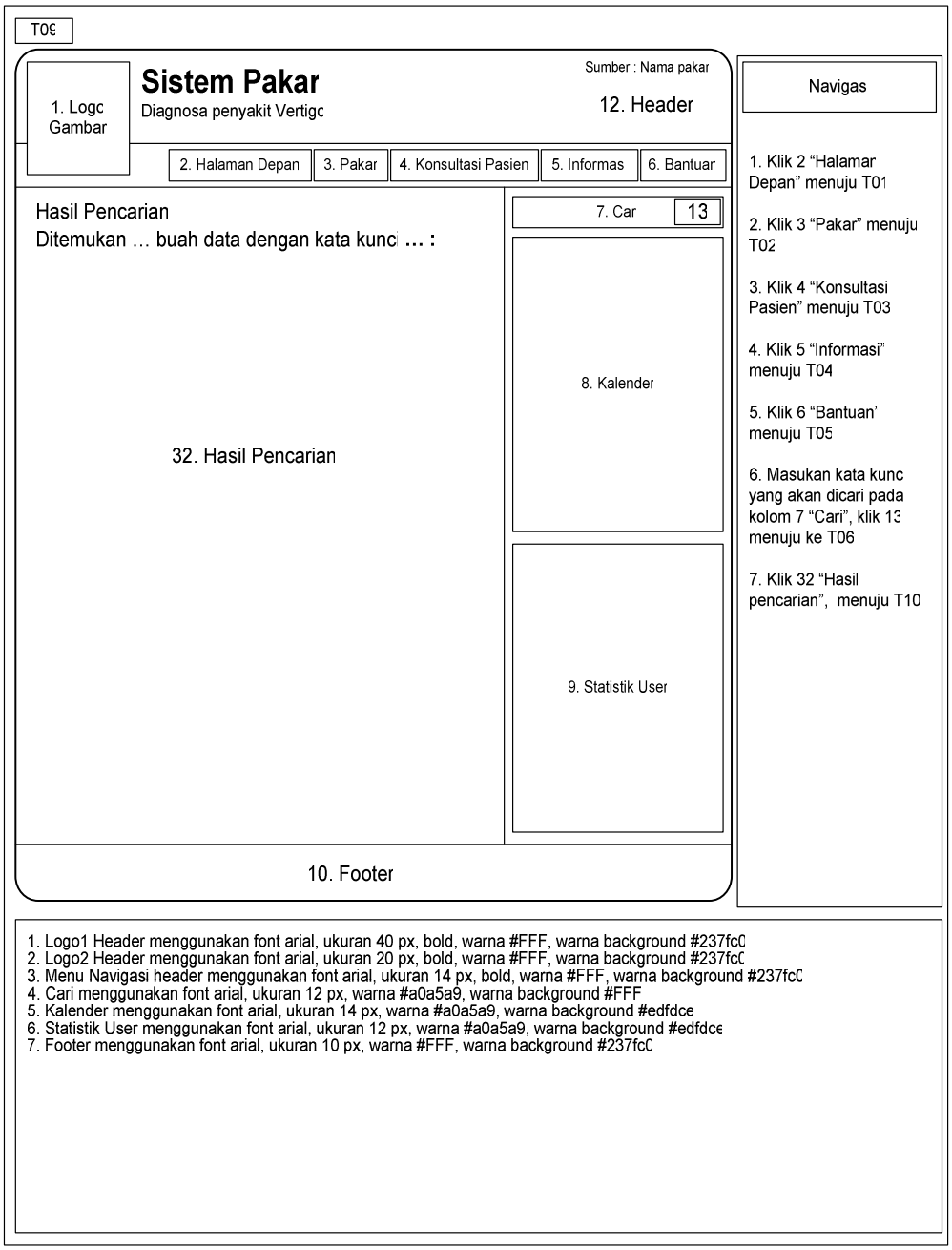


Gambar 3.27 Perancangan Halaman Validasi Daftar Pasien Baru



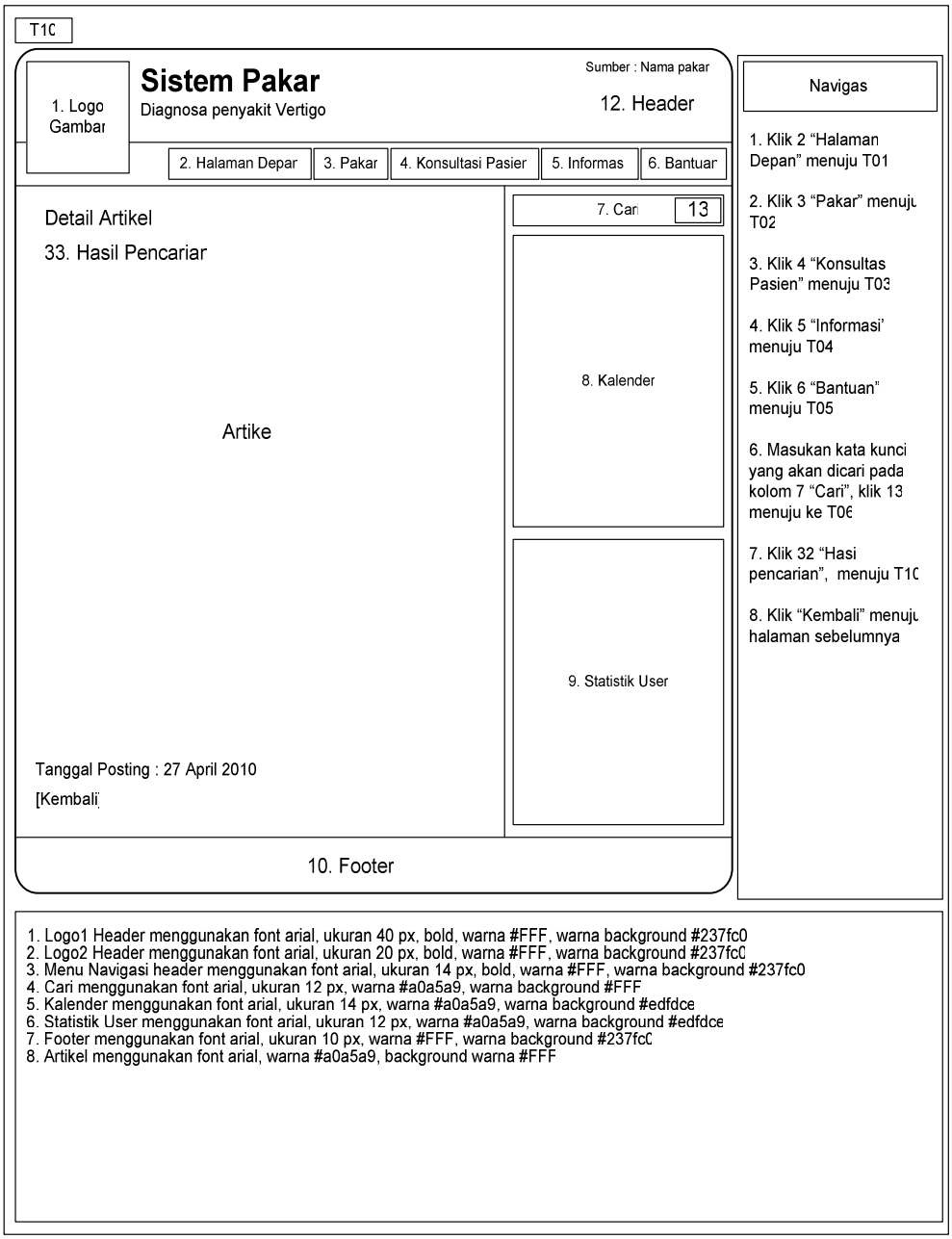
Gambar 3.28 Perancangan Halaman Validasi Daftar Pasien Baru *(Captcha)*

3.11.8 Perancangan Halaman Hasil Pencarian



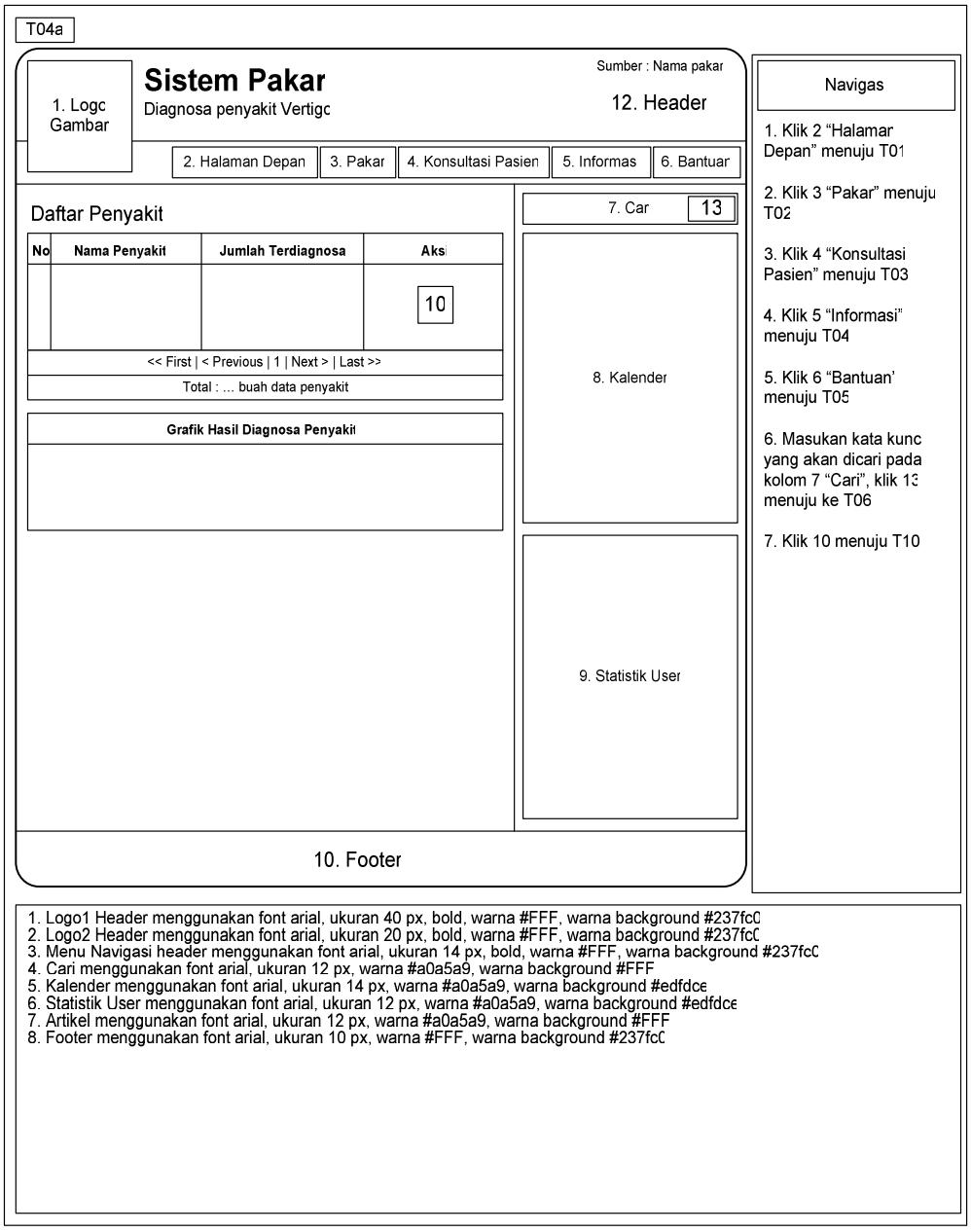
Gambar 3.29 Perancangan Halaman Hasil Pencarian

3.11.9 Perancangan Halaman Detail Artikel



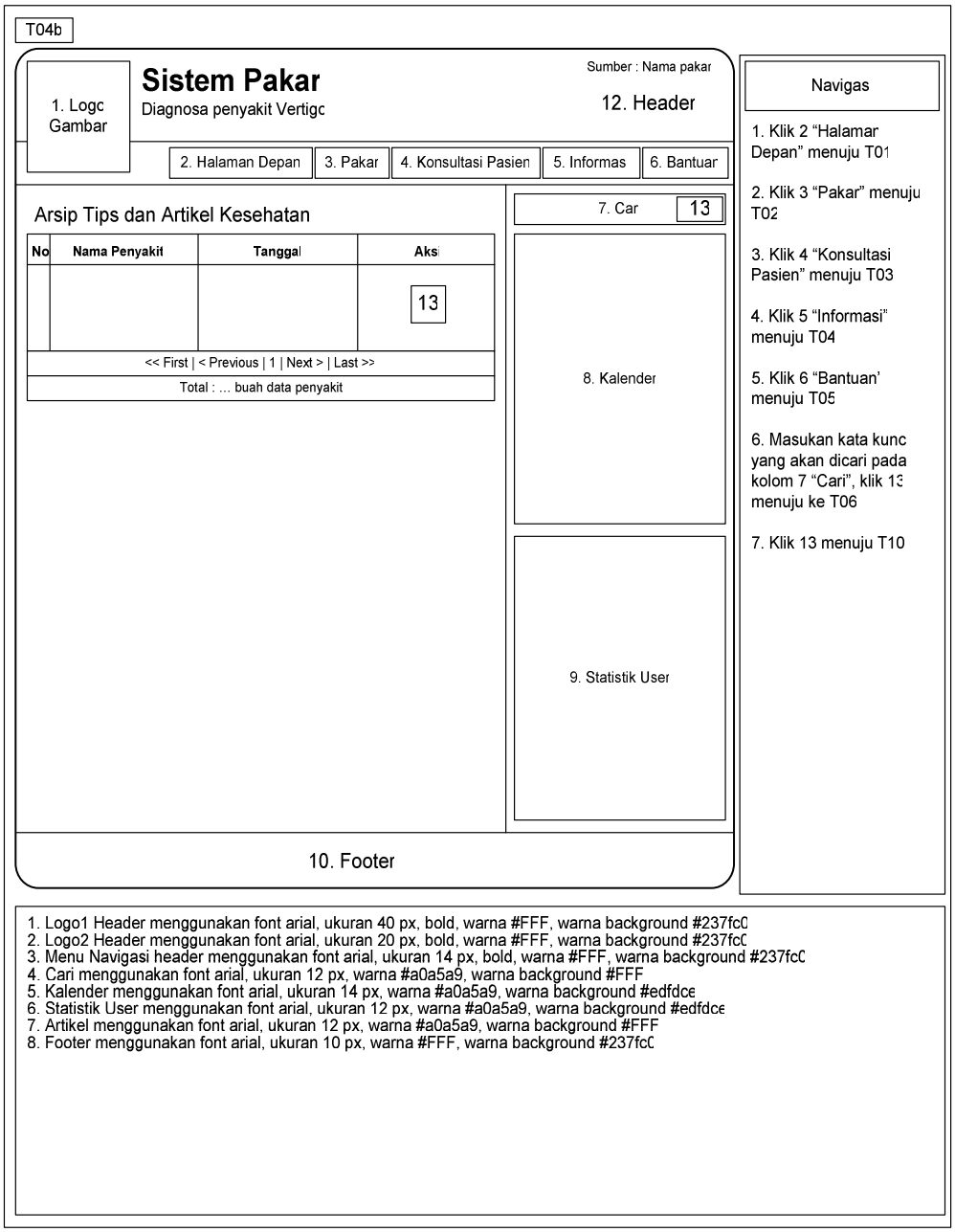
Gambar 3.30 Perancangan Halaman Detail Artikel

3.11.10Perancangan Halaman Informasi Daftar Penyakit



Gambar 3.31 Perancangan Halaman Informasi Daftar Penyakit

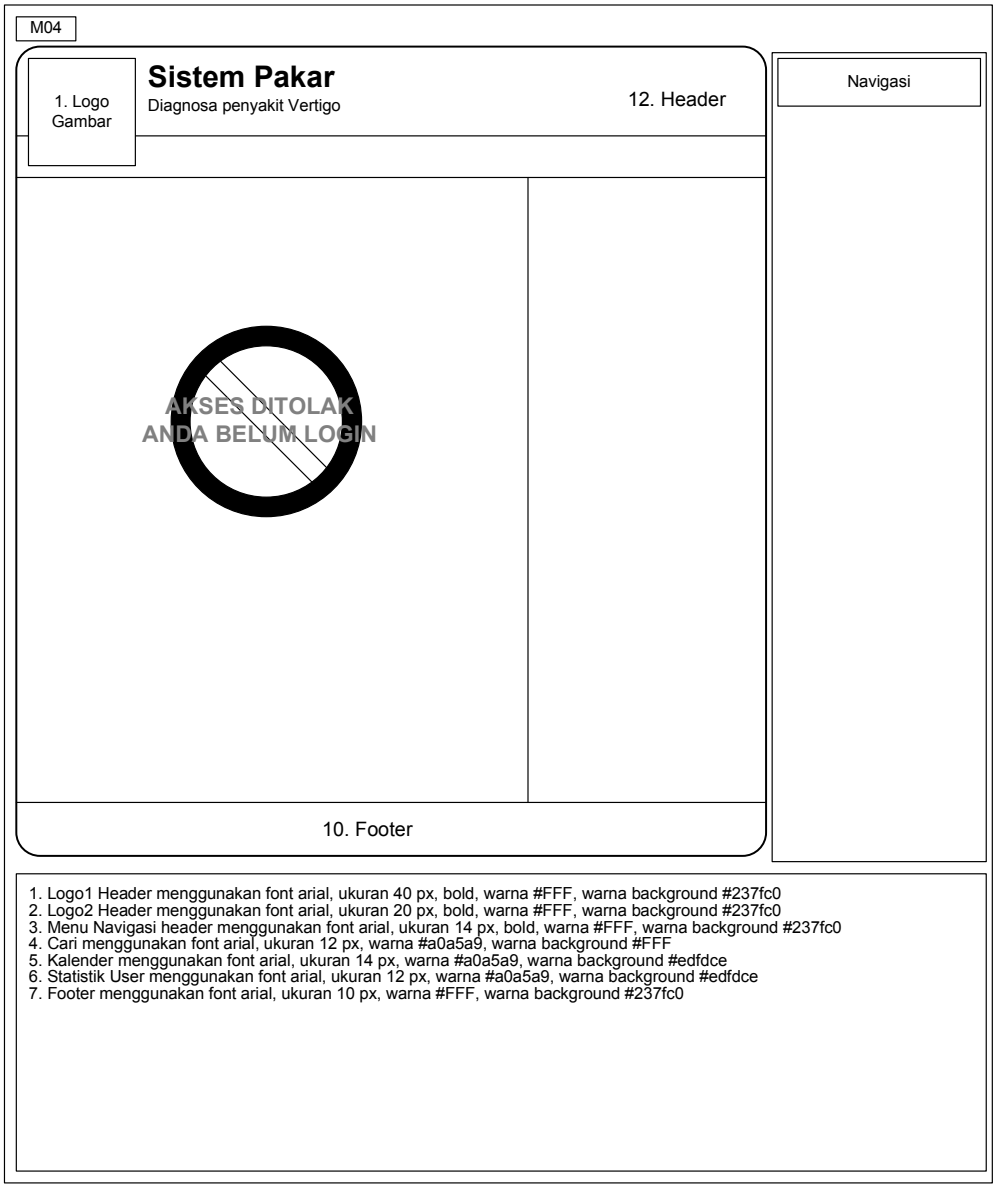
3.11.11Perancangan Halaman Informasi Arsip



Gambar 3.32 Perancangan Halaman Informasi Arsip

3.11.12Perancangan Halaman Peringatan

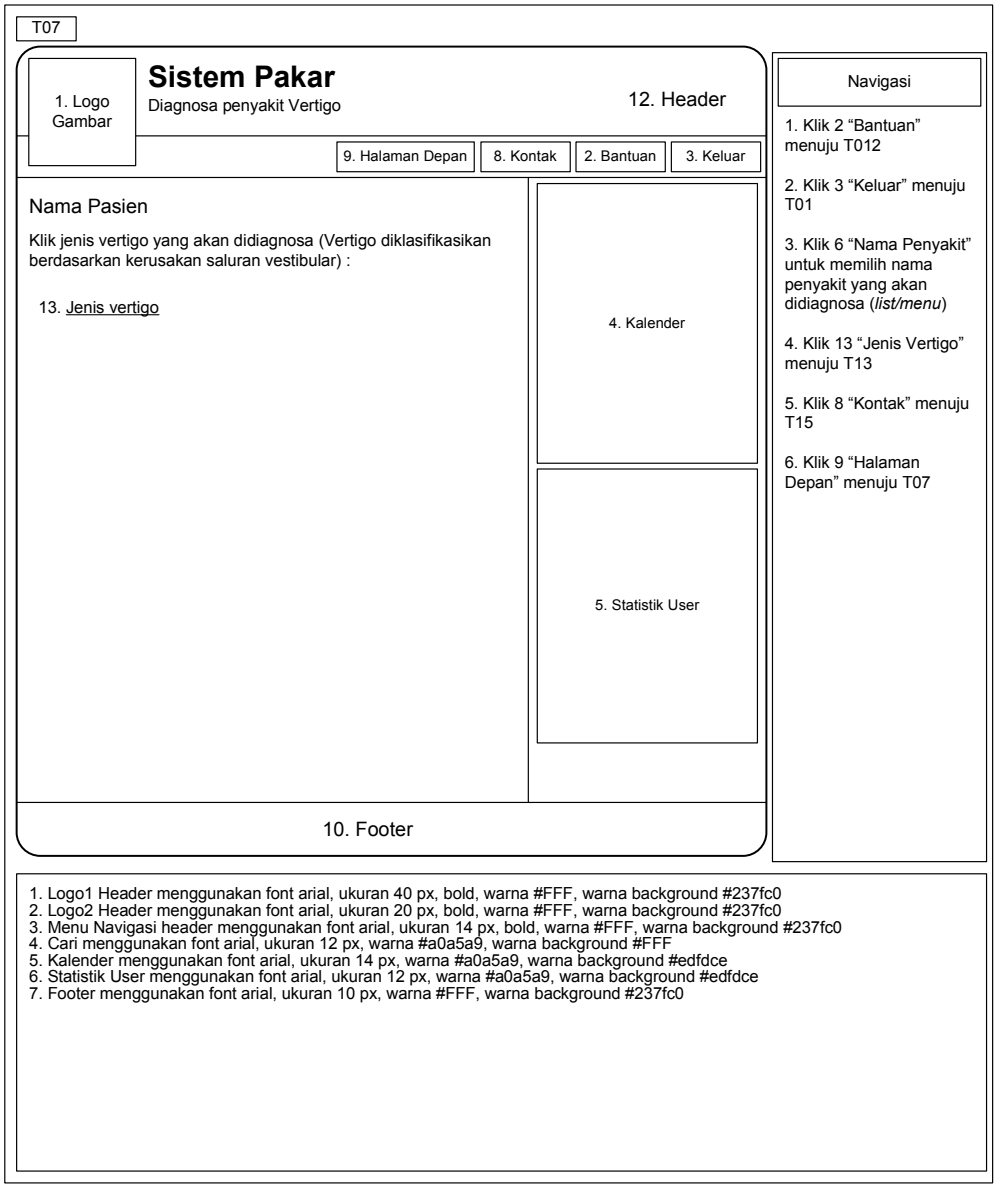
Kondisi ketika *user* mencoba masuk ke halaman pakar maupun pasien tanpa melakukan proses login atau daftar terlebih dahulu.



Gambar 3.33 Perancangan Halaman Peringatan

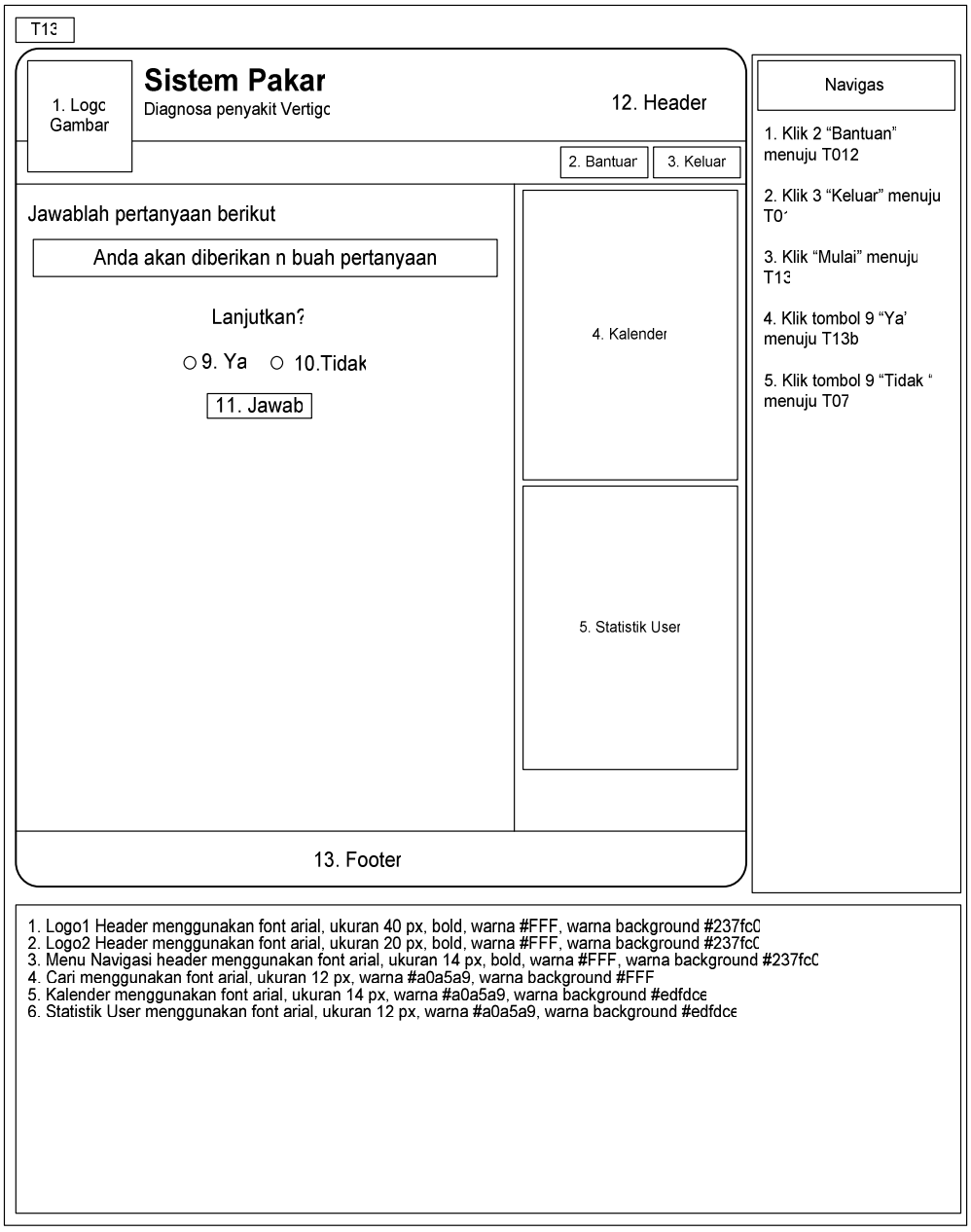
3.11.13Perancangan Halaman Pasien

Setelah proses daftar selesai dilakukan, maka *user* akan masuk ke halaman pasien. Perancangan halaman pasien dapat dilihat pada gambar 3.34 di bawah ini :



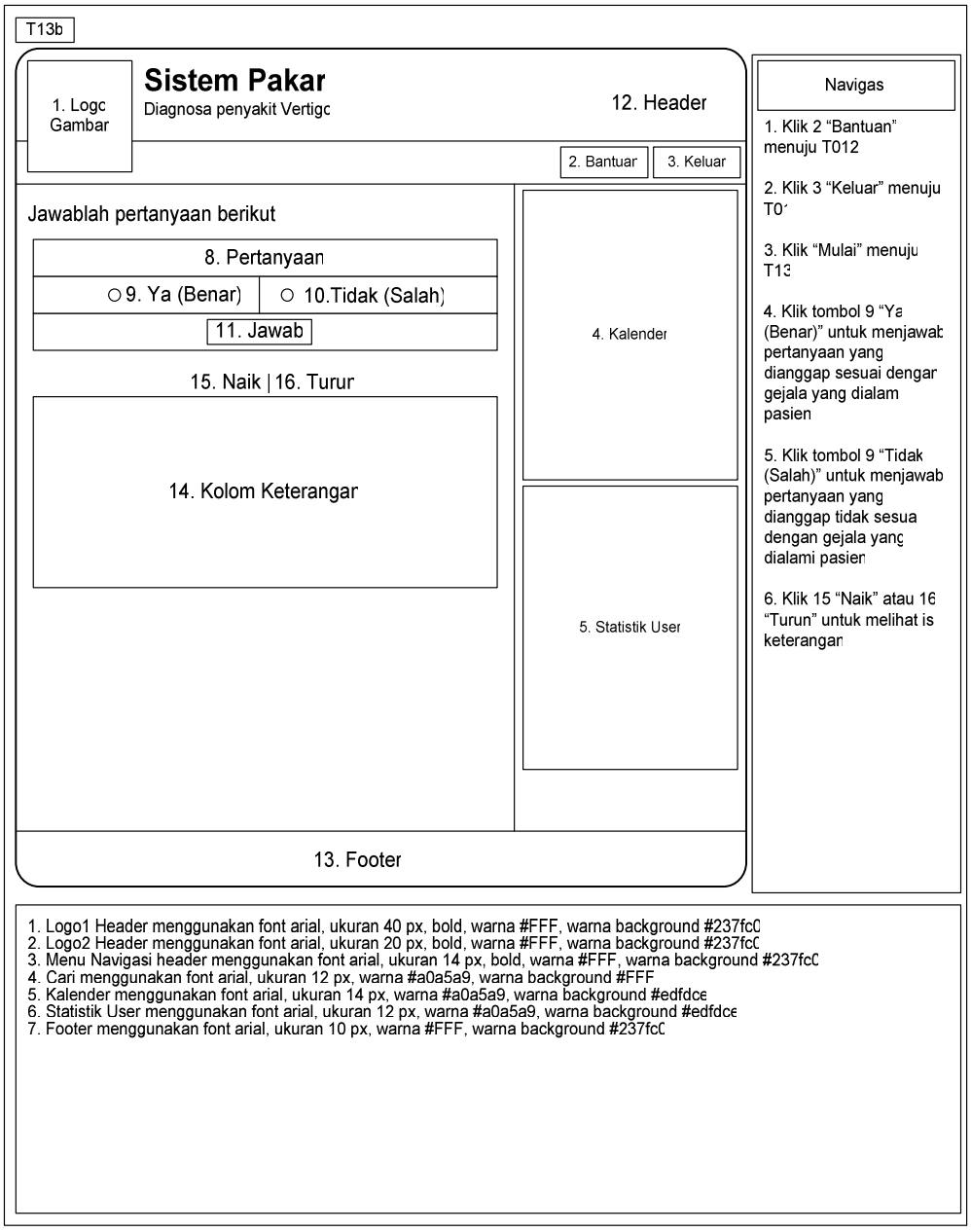
Gambar 3.34 Perancangan Halaman Pasien

3.11.14Perancangan Halaman Konfirmasi



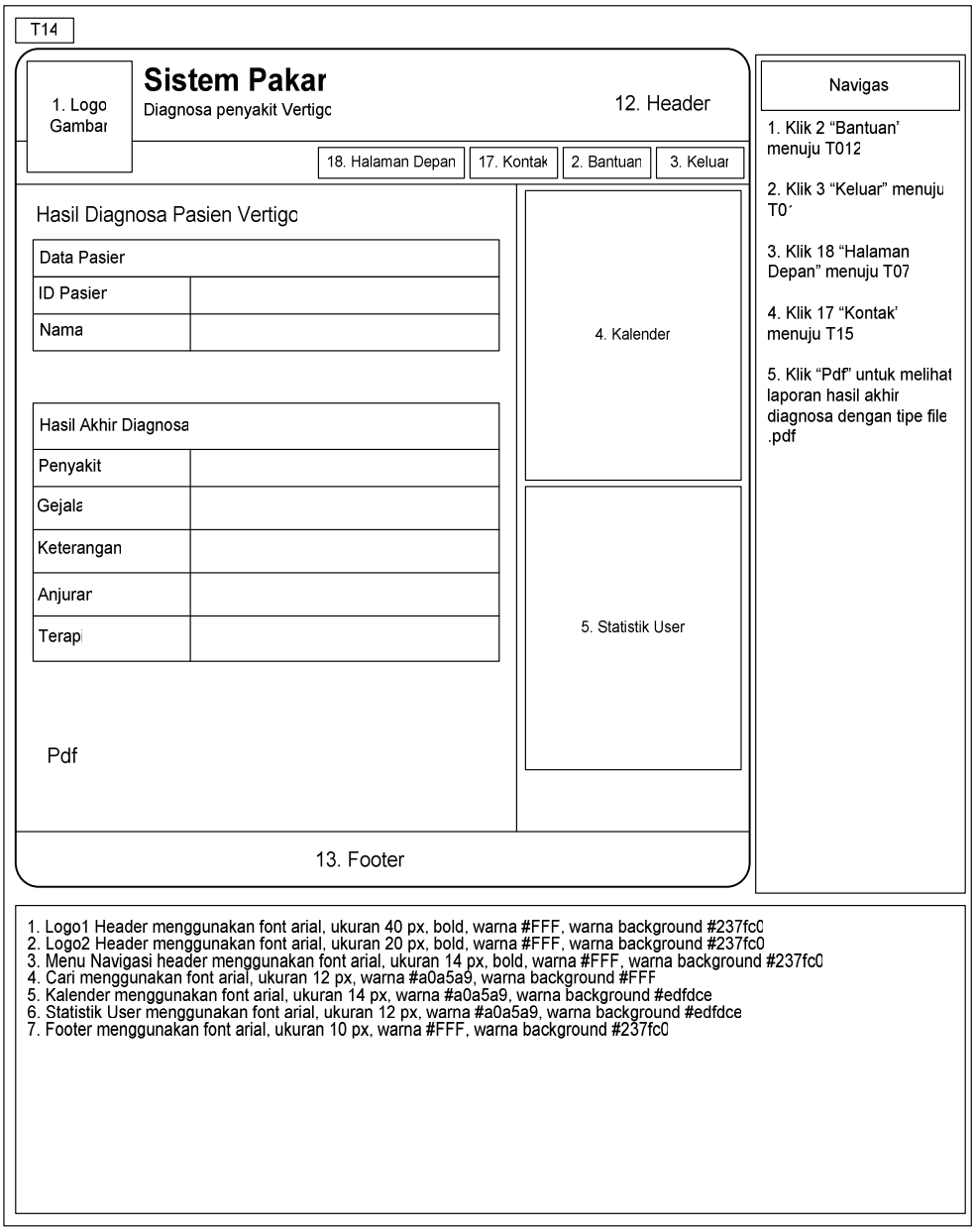
Gambar 3.35 Perancangan Halaman Pasien

3.11.15Perancangan Halaman Diagnosa (Pertanyaan)



Gambar 3.36 Perancangan Halaman Diagnosa (Pertanyaan)

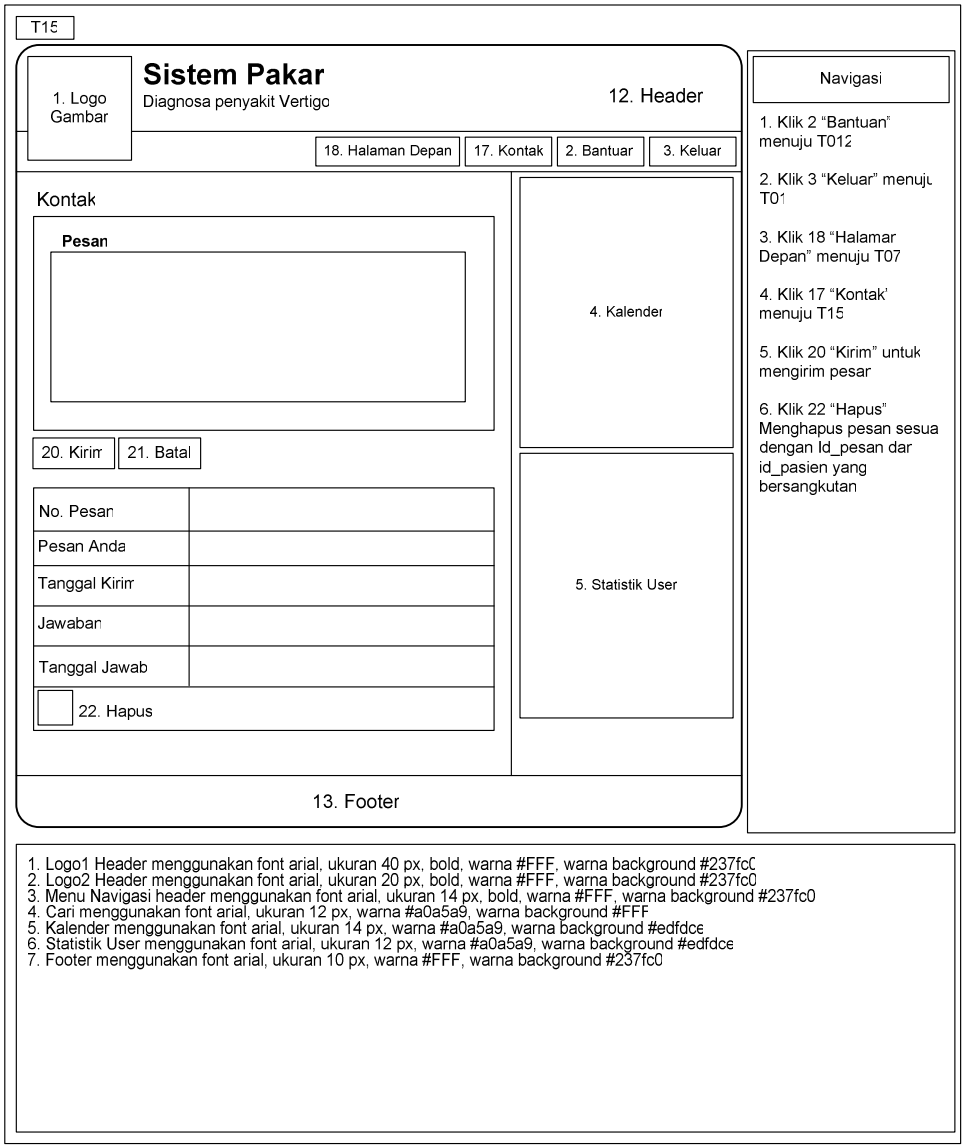
3.11.16Perancangan Halaman Hasil Diagnosa



Gambar 3.37 Perancangan Halaman Hasil Diagnosa

3.11.17Perancangan Halaman Kontak

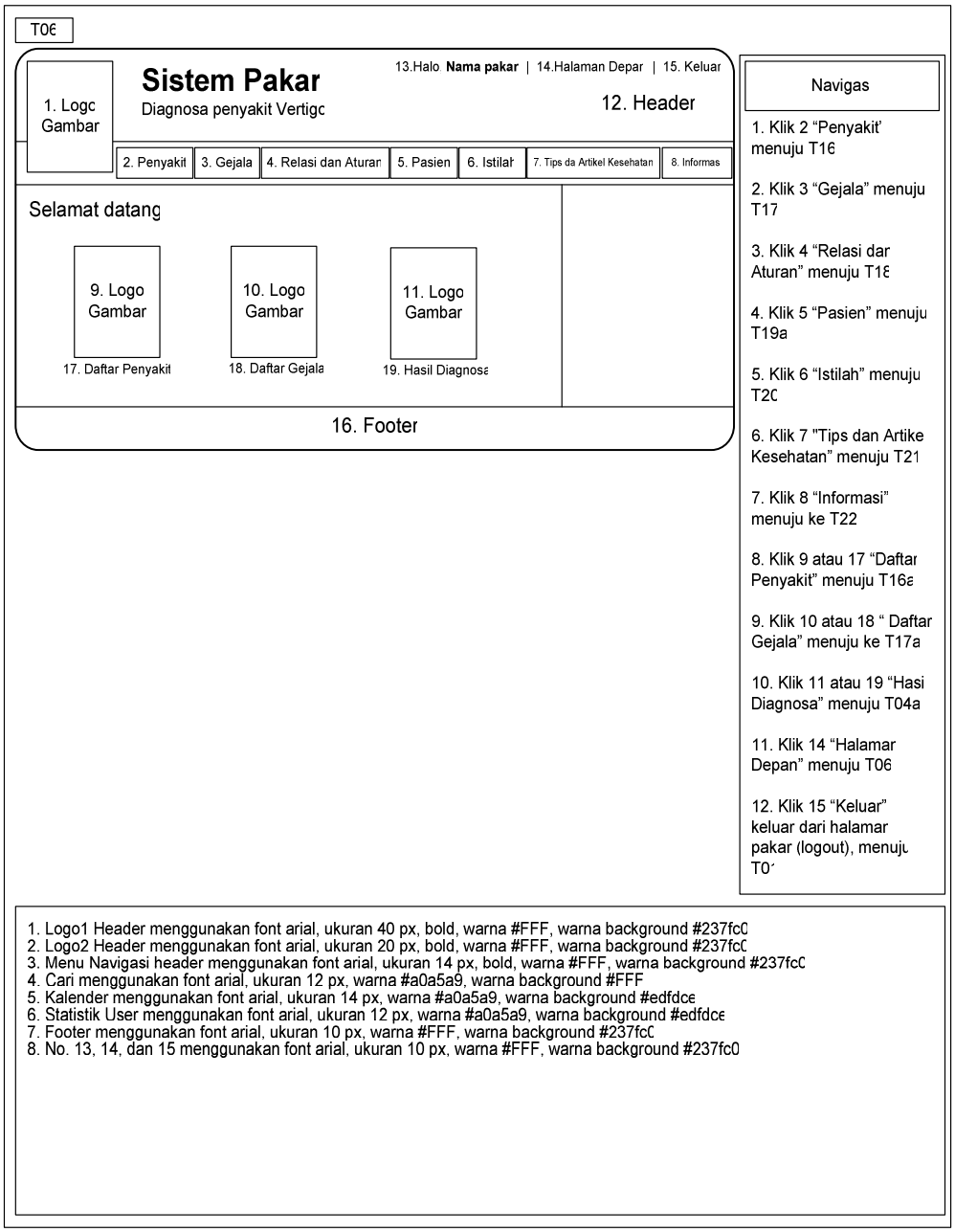
Ketika *user* mengklik menu kontak, maka akan tampil halaman seperti di bawah ini :



Gambar 3.38 Perancangan Halaman Kontak

Menu kontak ini merupakan layanan interaksi antara pasien dengan pakar yang sifat personal, artinya pesan yang disampaikan hanya tampil pada halaman kontak pasien yang terkait.

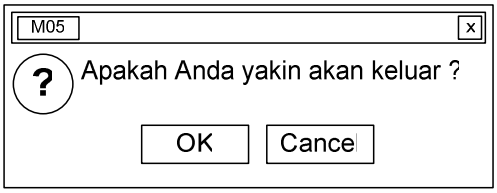
3.11.18Perancangan Halaman Depan Pakar



Gambar 3.39 Perancangan Halaman Depan Pakar

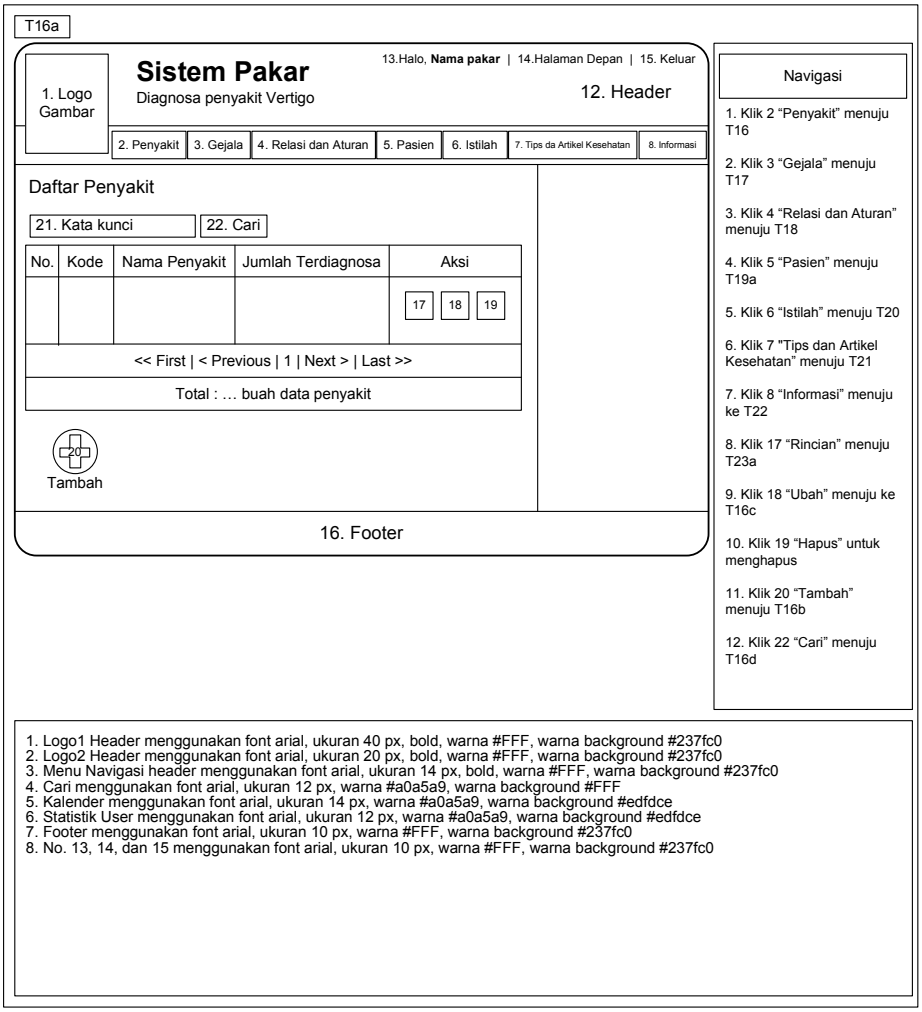
3.11.19Perancangan Pesan Validasi Keluar

Ketika *user* mengklik menu keluar baik pakar maupun pasien, maka akan tampil pesan sebagai berikut :



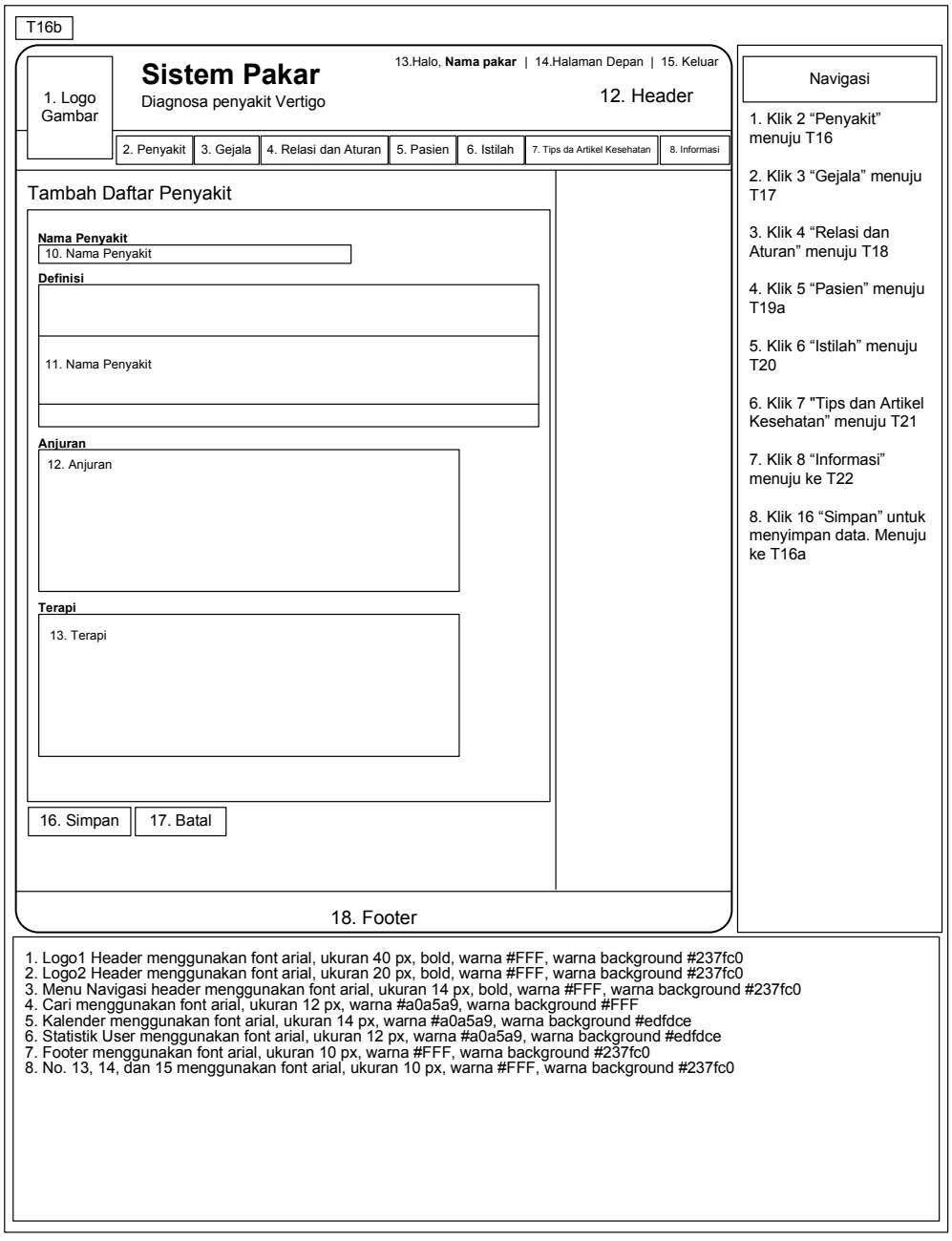
Gambar 3.40 Perancangan Pesan Validasi Keluar

3.11.20Perancangan Halaman Daftar Penyakit



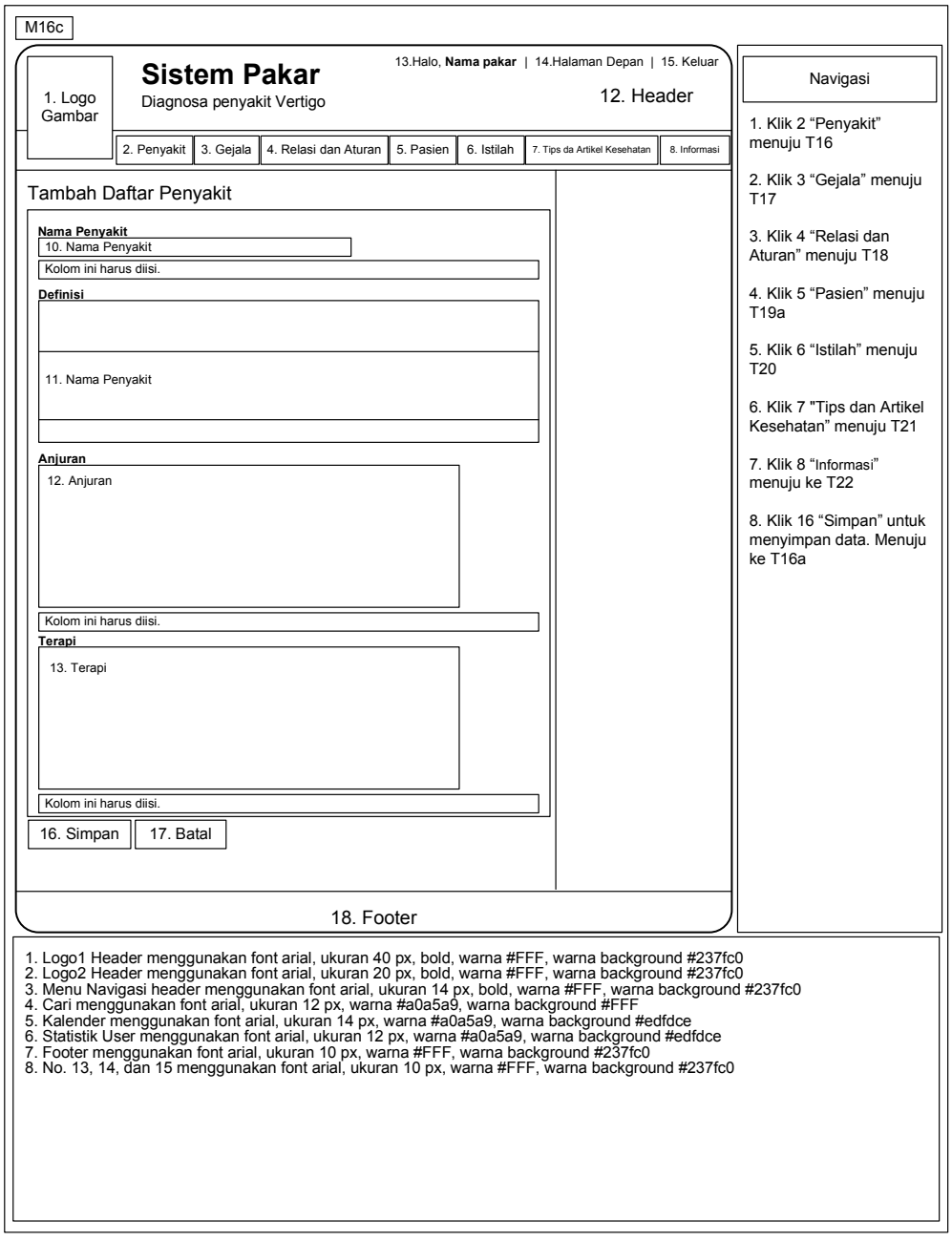
Gambar 3.41 Perancangan Halaman Daftar Penyakit

3.11.21Perancangan Halaman Tambah Penyakit



Gambar 3.42 Perancangan Halaman Tambah Daftar Penyakit

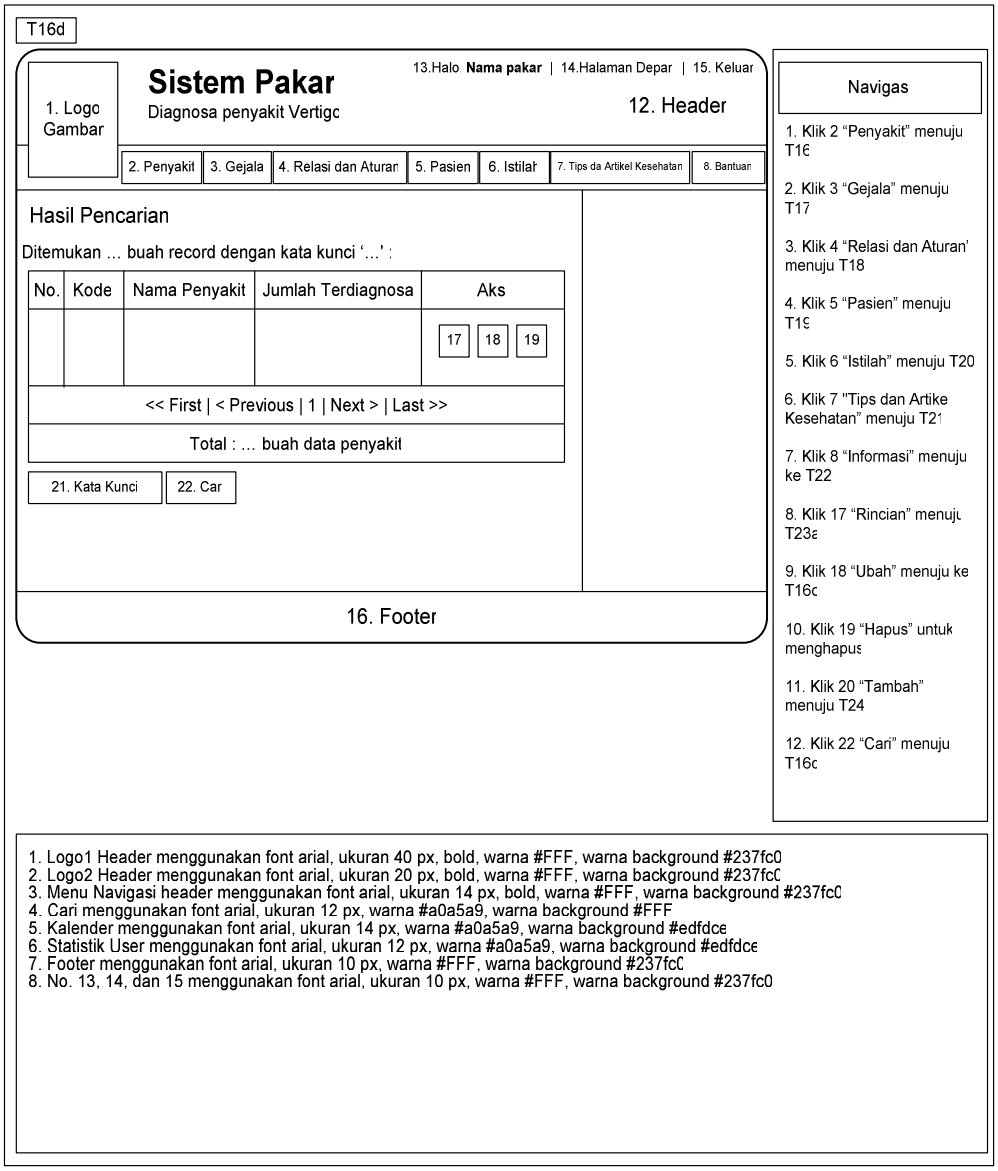
3.11.22Perancangan Validasi Halaman Tambah Penyakit



Gambar 3.43 Perancangan Validasi Halaman Tambah Daftar Penyakit

3.11.23Perancangan Halaman Hasil Pencarian Penyakit

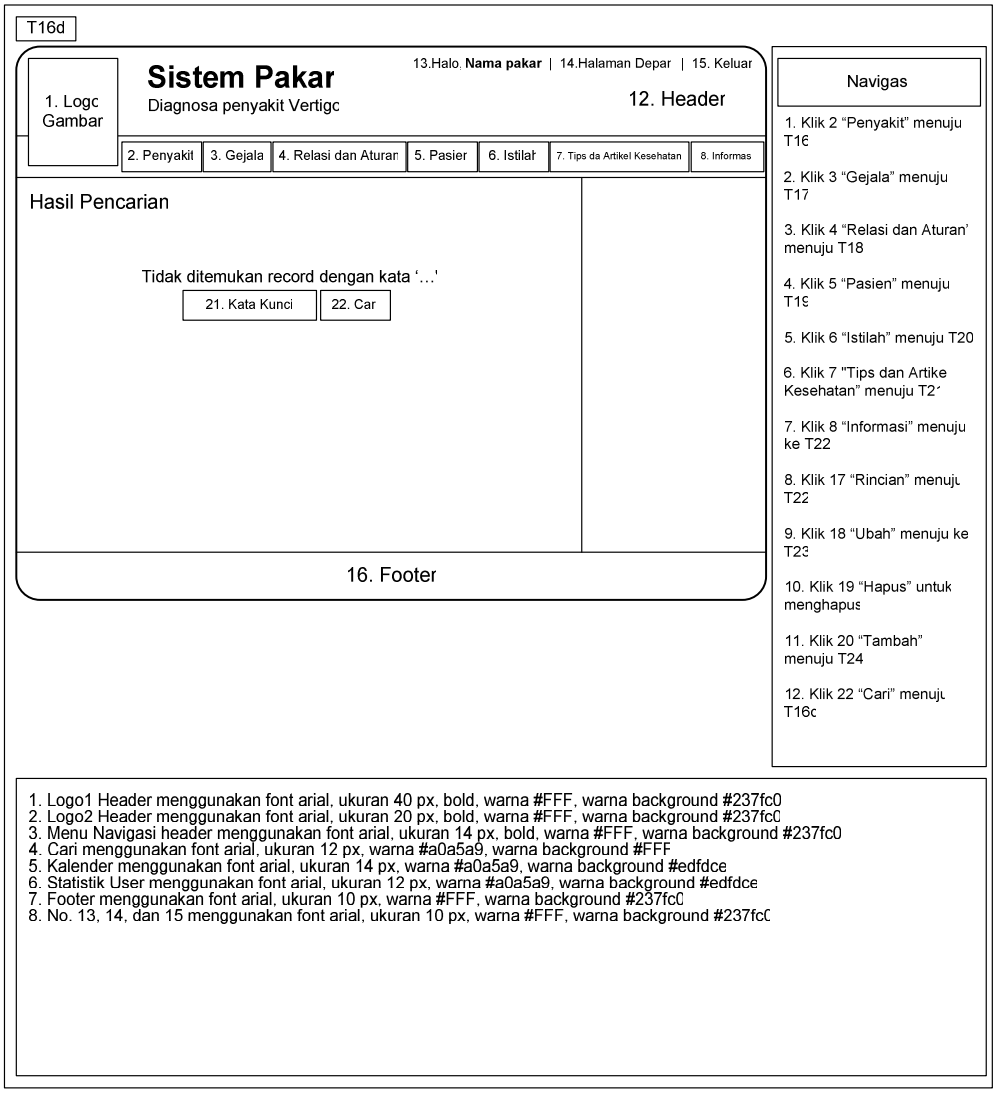
Berikut ini adalah perancangan halaman pencarian penyakit jika kata kunci yang dimasukkan telah ditemukan.



Gambar 3.44 Perancangan Validasi Halaman Hasil Pencarian Penyakit

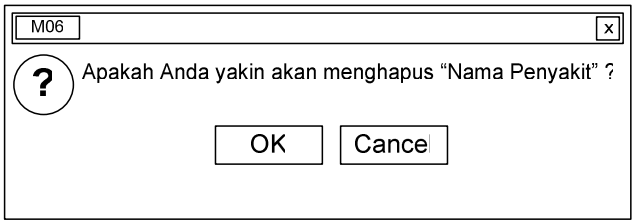
3.11.24Perancangan Halaman Hasil Pencarian Penyakit

Berikut ini adalah perancangan halaman pencarian penyakit jika kata kunci yang dimasukkan tidak ditemukan.



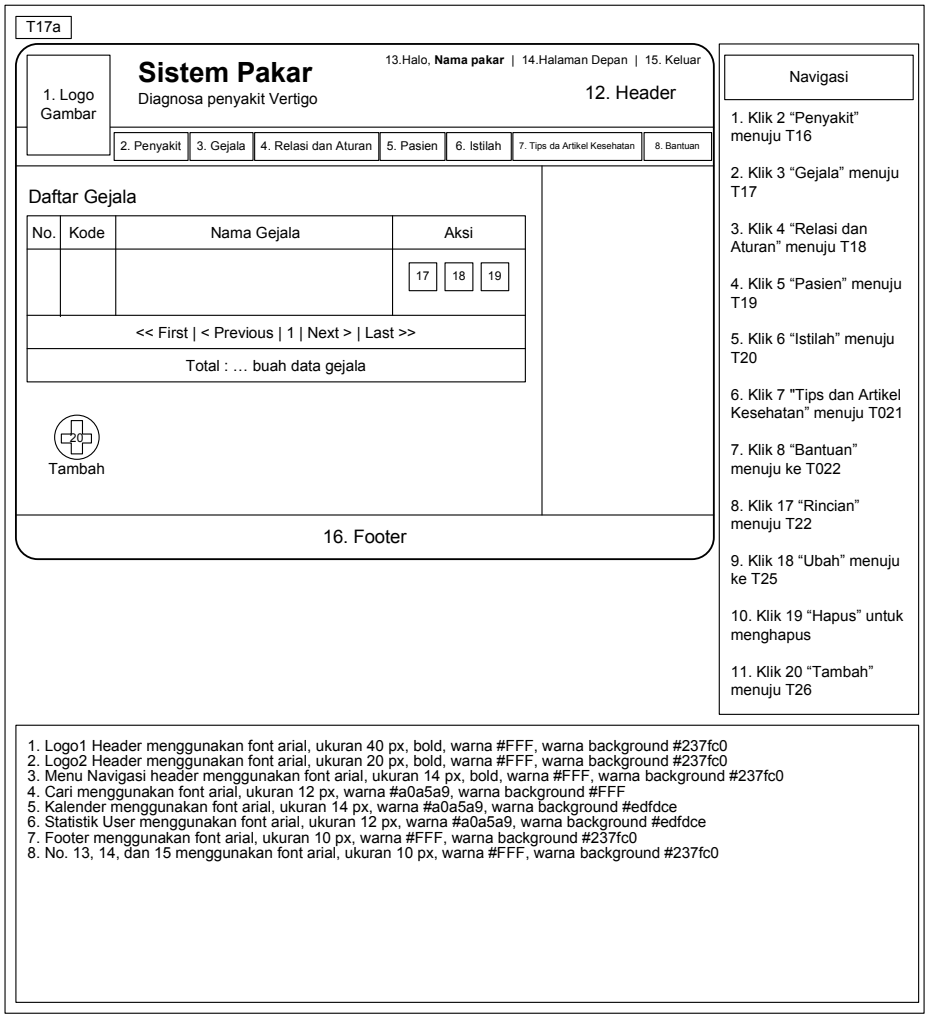
Gambar 3.45 Perancangan Validasi Halaman Hasil Pencarian Penyakit

3.11.25Perancangan Halaman Hasil Pencarian Penyakit



Gambar 3.46 Perancangan Pesan Validasi Hapus Penyakit

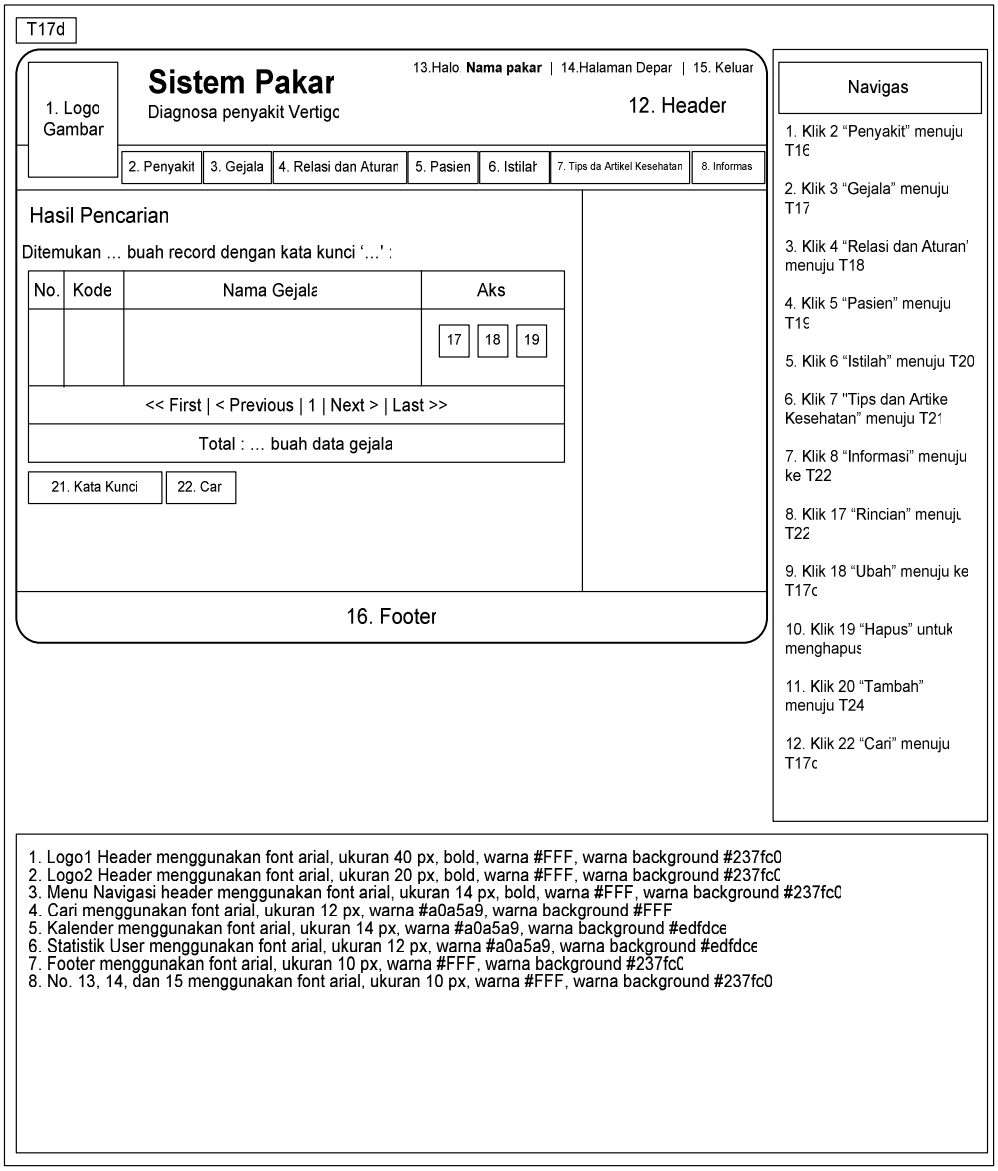
3.11.26Perancangan Halaman Daftar Gejala



Gambar 3.47 Perancangan Halaman Daftar gejala

3.11.27Perancangan Halaman Hasil Pencarian Gejala

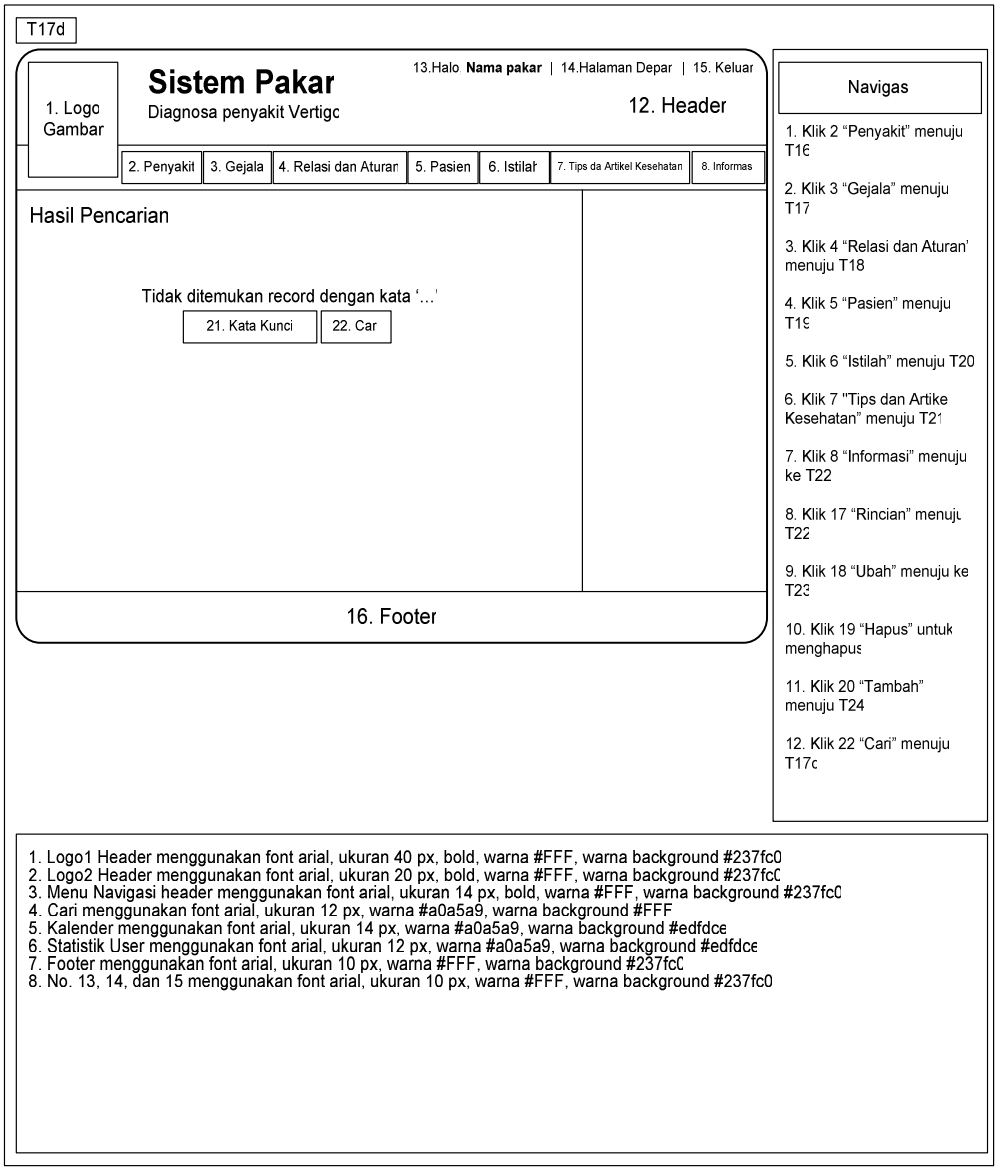
Berikut ini adalah perancangan halaman pencarian gejala jika kata kunci yang dimasukkan telah ditemukan.



Gambar 3.48 Perancangan Validasi Halaman Hasil Pencarian Gejala

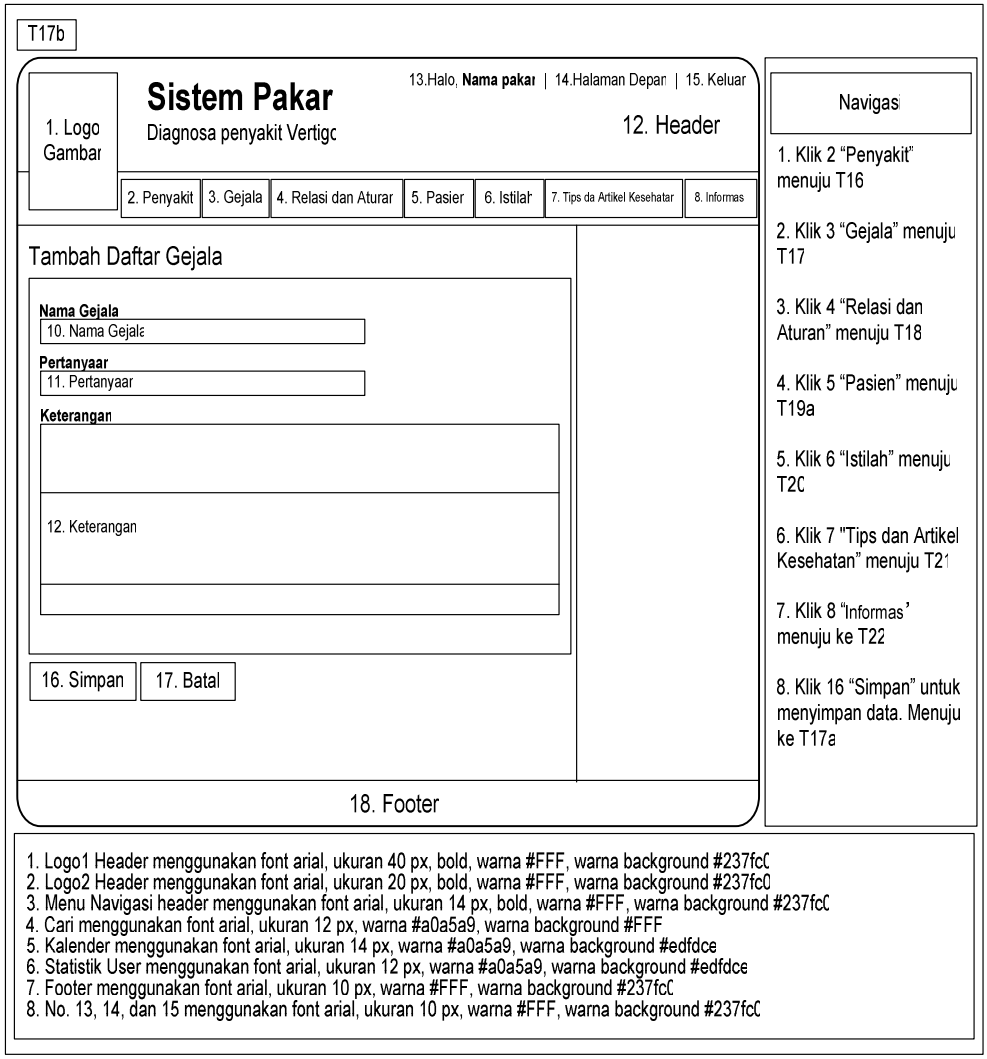
3.11.28Perancangan Halaman Hasil Pencarian Gejala

Berikut ini adalah perancangan halaman pencarian gejala jika kata kunci yang dimasukkan tidak ditemukan.



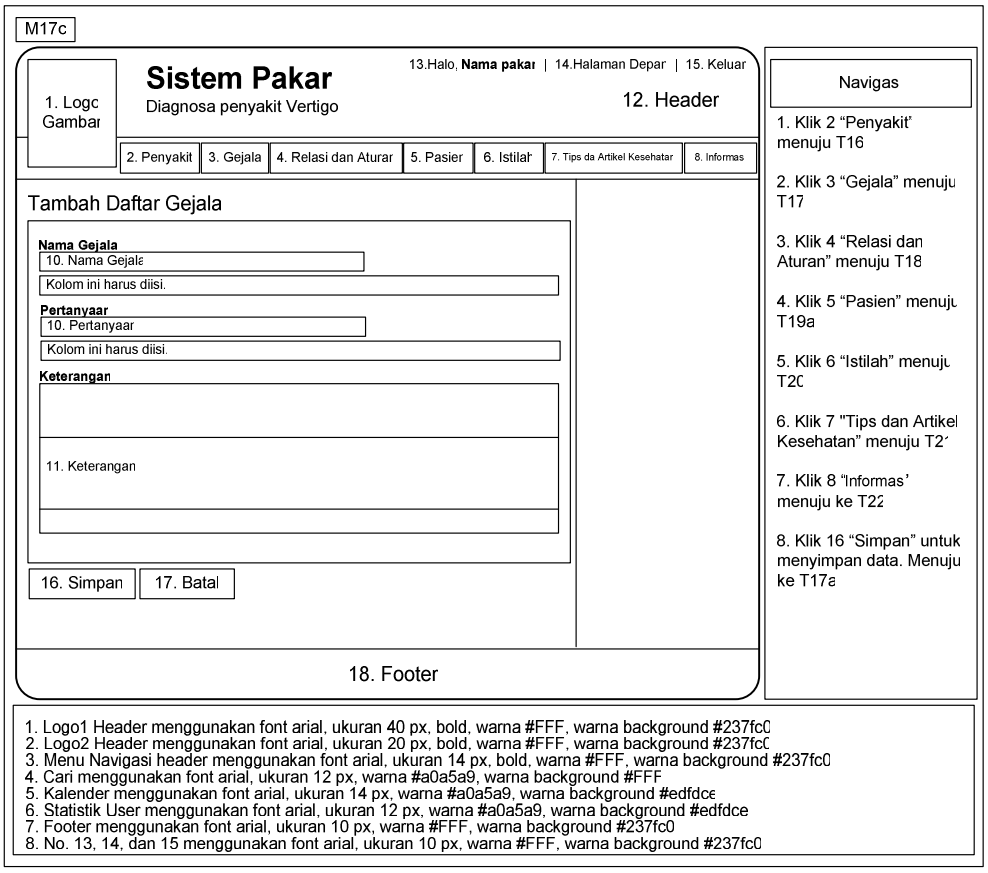
Gambar 3.49 Perancangan Validasi Halaman Hasil Pencarian Gejala

3.11.29Perancangan Halaman Tambah Gejala



Gambar 3.50 Perancangan Halaman Tambah gejala

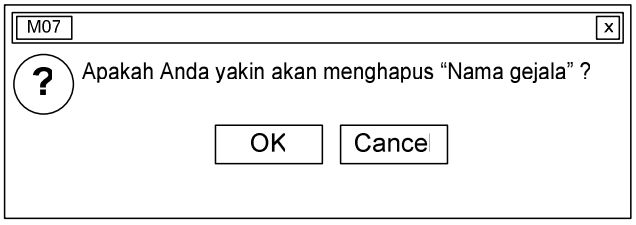
3.11.30Perancangan Halaman Validasi Tambah Gejala



Gambar 3.51 Perancangan Validasi Halaman Tambah gejala

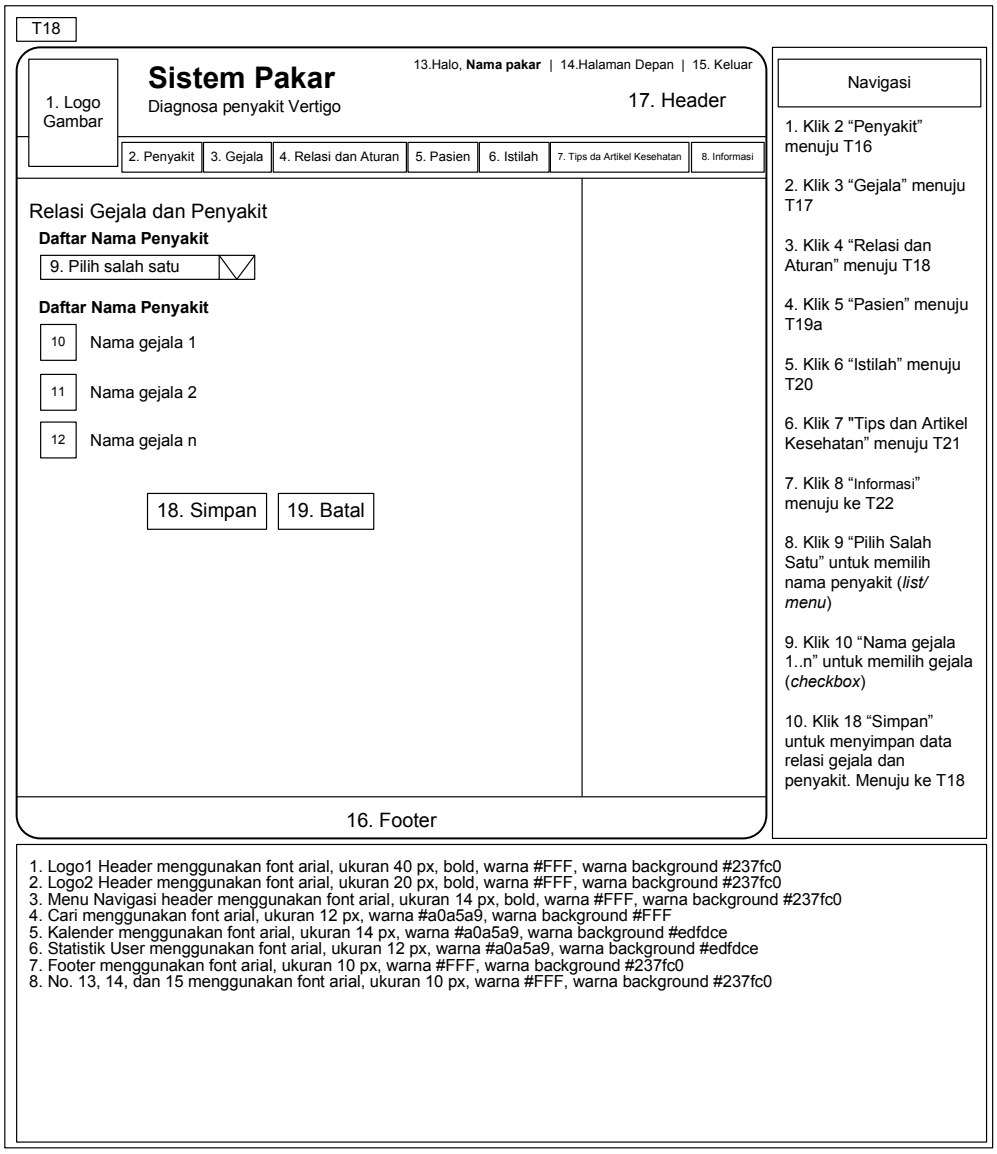
3.11.31Perancangan Pesan Hapus Gejala

Ketika *user* mengklik menu hapus, maka akan tampil pesan sebagai berikut :



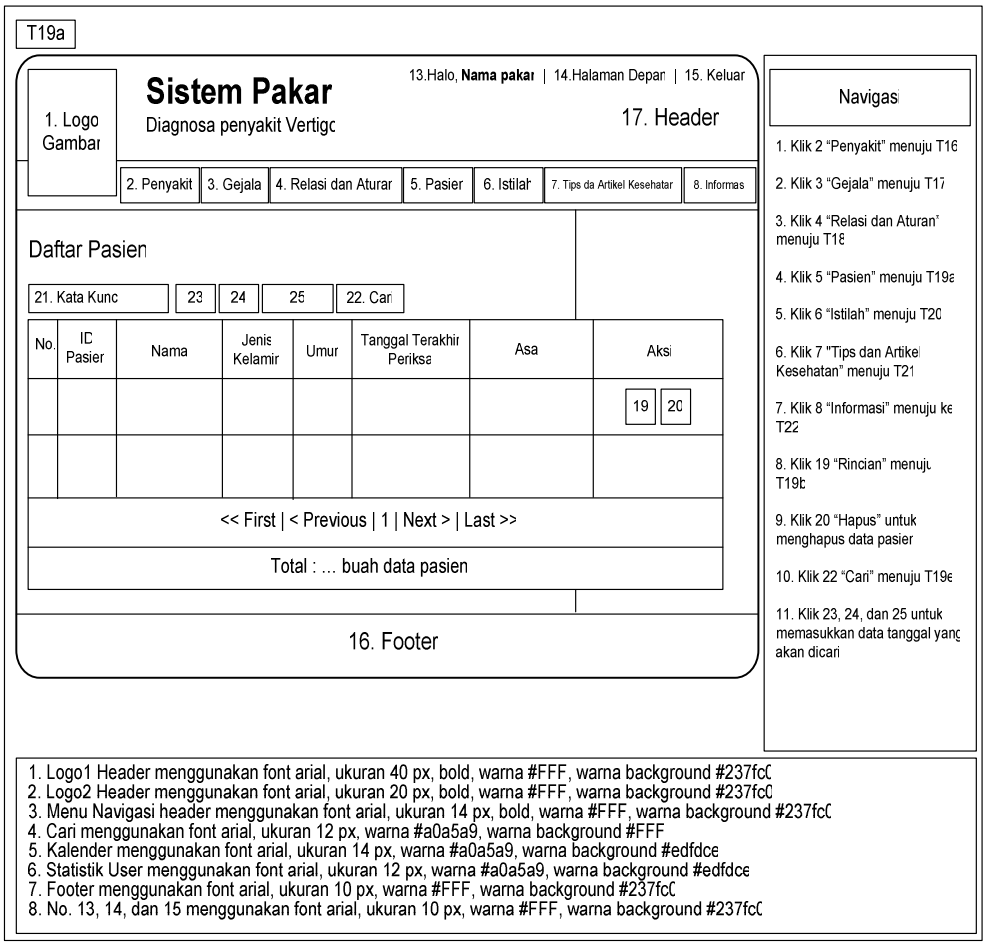
Gambar 3.52 Perancangan Pesan Hapus Gejala

3.11.32Perancangan Halaman Relasi dan Aturan



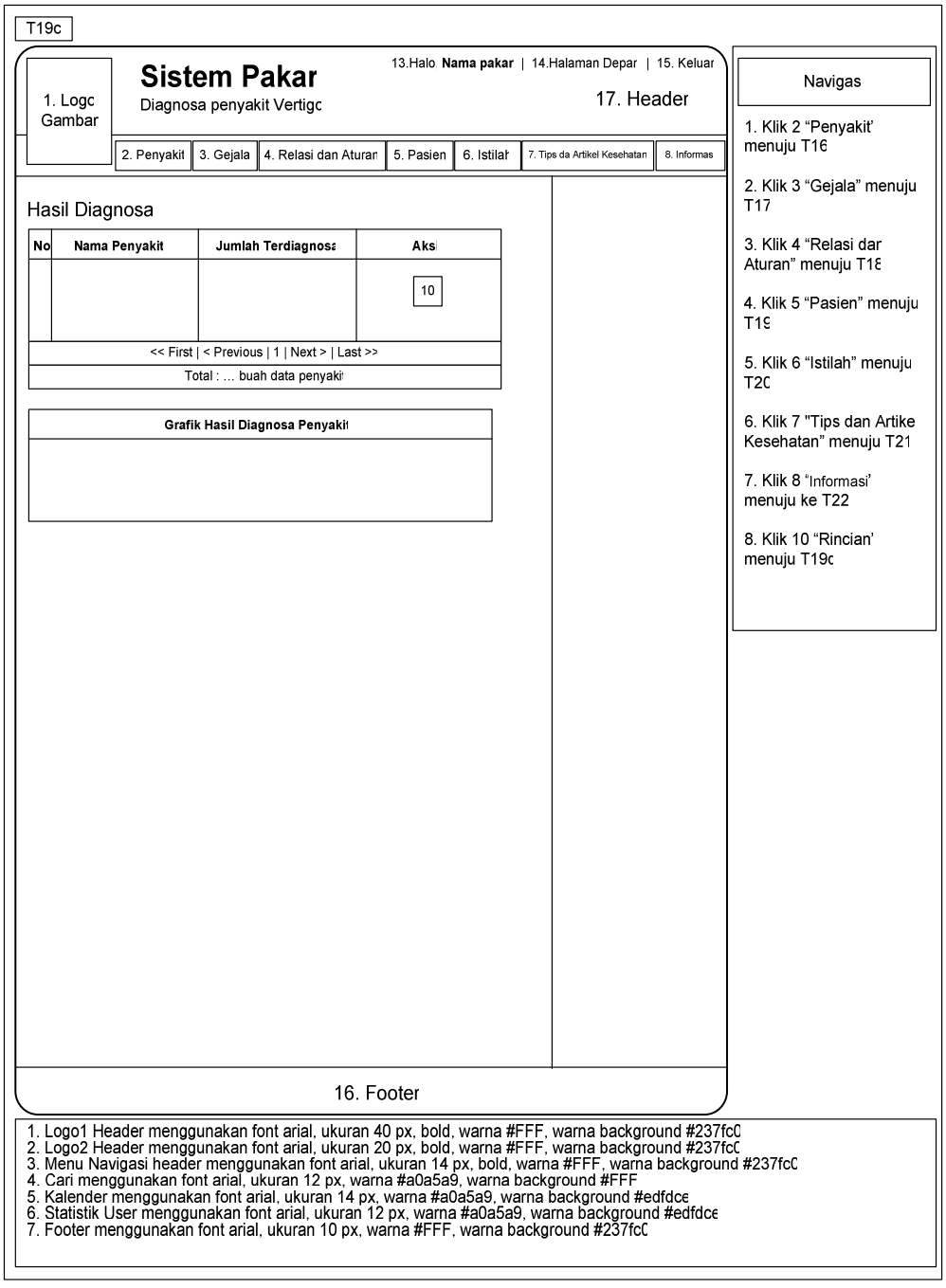
Gambar 3.53 Perancangan Halaman Relasi dan Aturan

3.11.33Perancangan Halaman Daftar Pasien



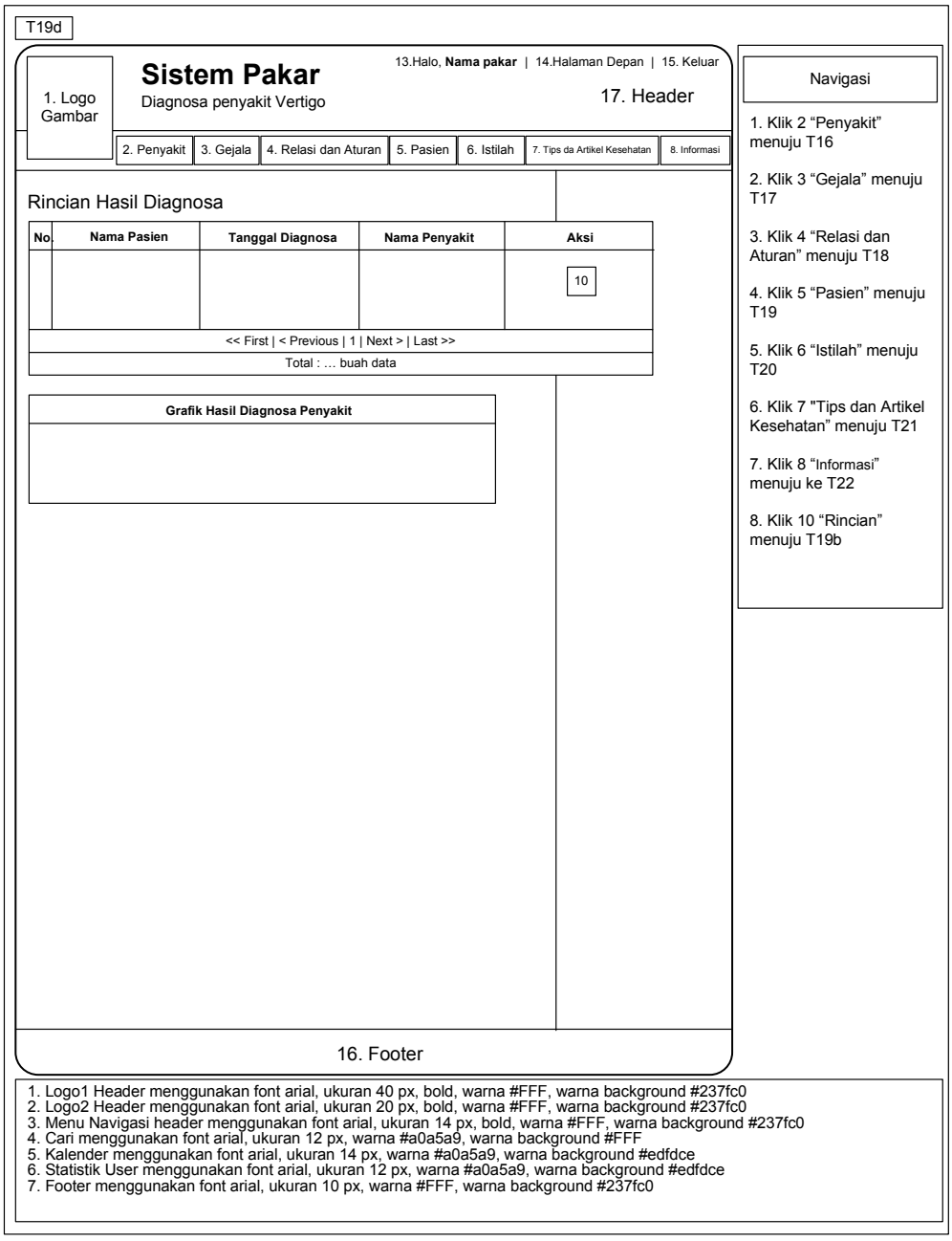
Gambar 3.54 Perancangan Halaman Daftar Pasien

3.11.34Perancangan Halaman Hasil Diagnosa



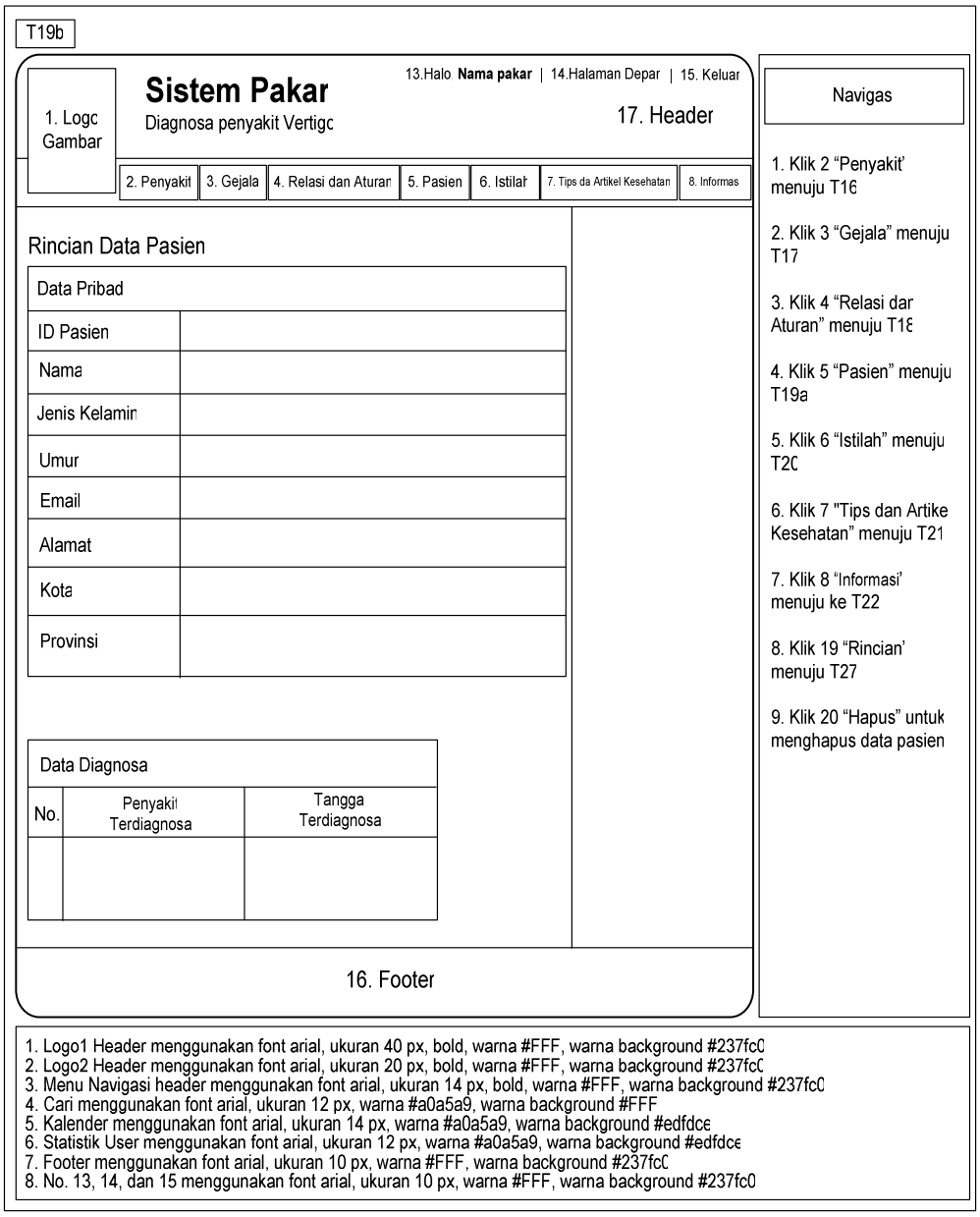
Gambar 3.55 Perancangan Halaman Hasil Diagnosa

3.11.35Perancangan Halaman Rincian Hasil Diagnosa



Gambar 3.56 Perancangan Rincian Halaman Hasil Diagnosa

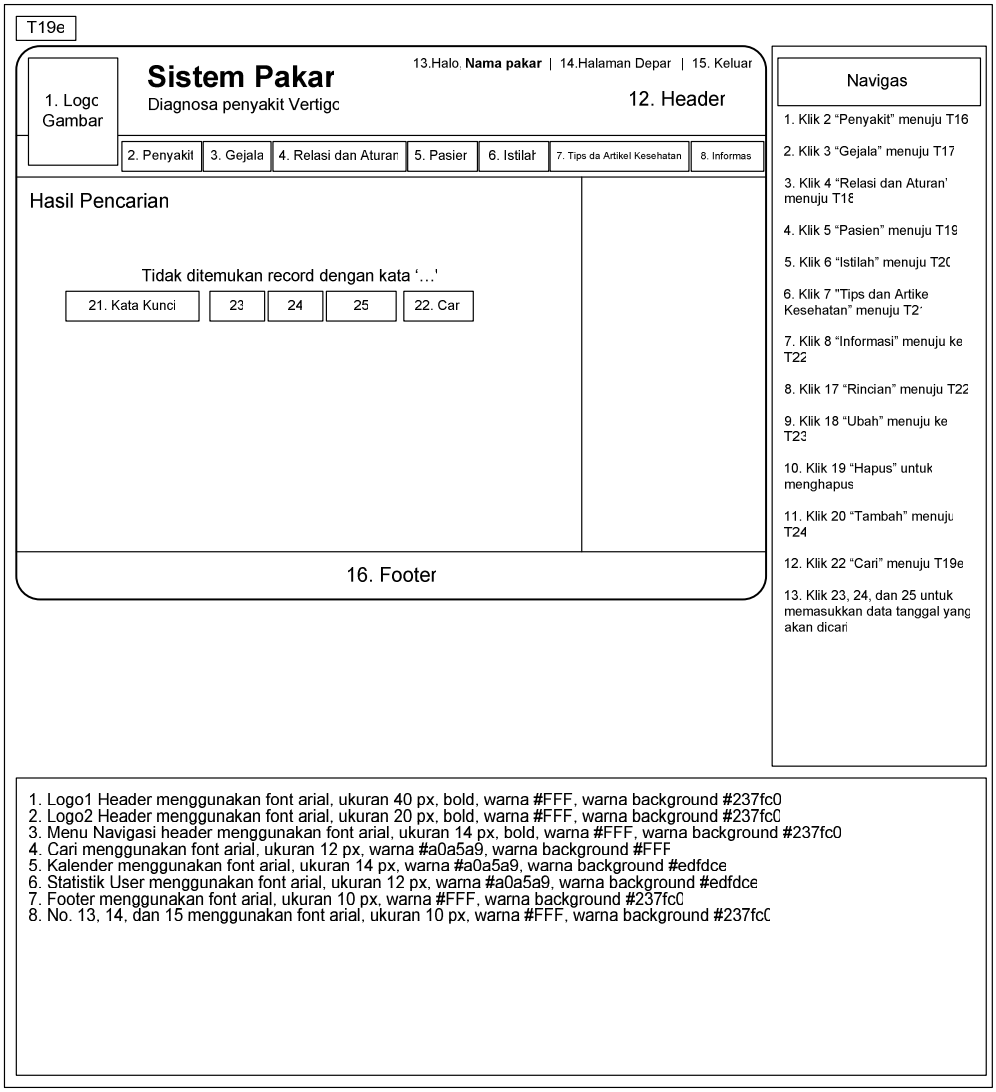
3.11.36Perancangan Halaman Rincian Data Pasien



Gambar 3.57 Perancangan Halaman Rincian Data Pasien

3.11.37Perancangan Halaman Hasil Pencarian Pasien

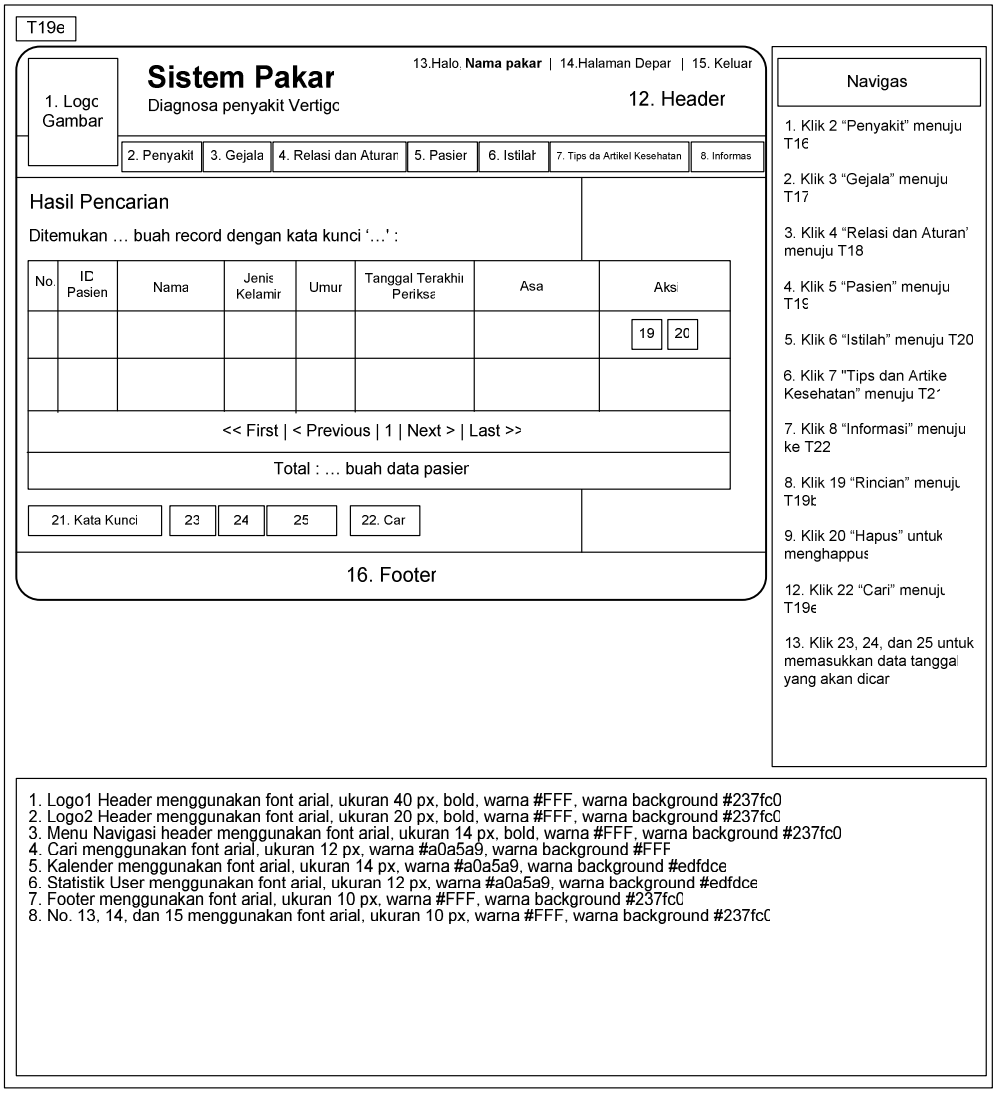
Berikut ini adalah perancangan halaman pencarian pasien jika kata kunci yang dimasukkan tidak ditemukan.



Gambar 3.58 Perancangan Validasi Halaman Hasil Pencarian Pasien

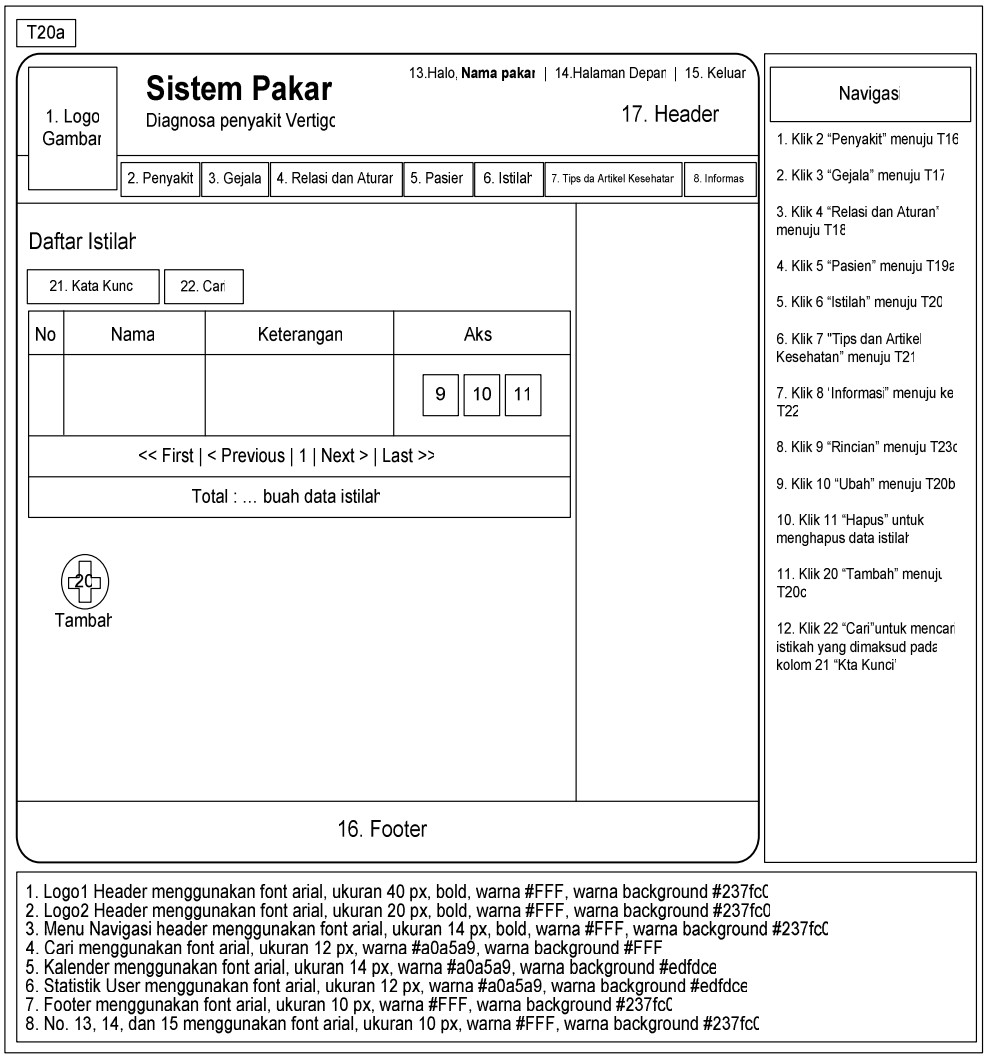
3.11.38Perancangan Halaman Hasil Pencarian Pasien

Berikut ini adalah perancangan halaman pencarian pasien jika kata kunci yang dimasukkan telah ditemukan.



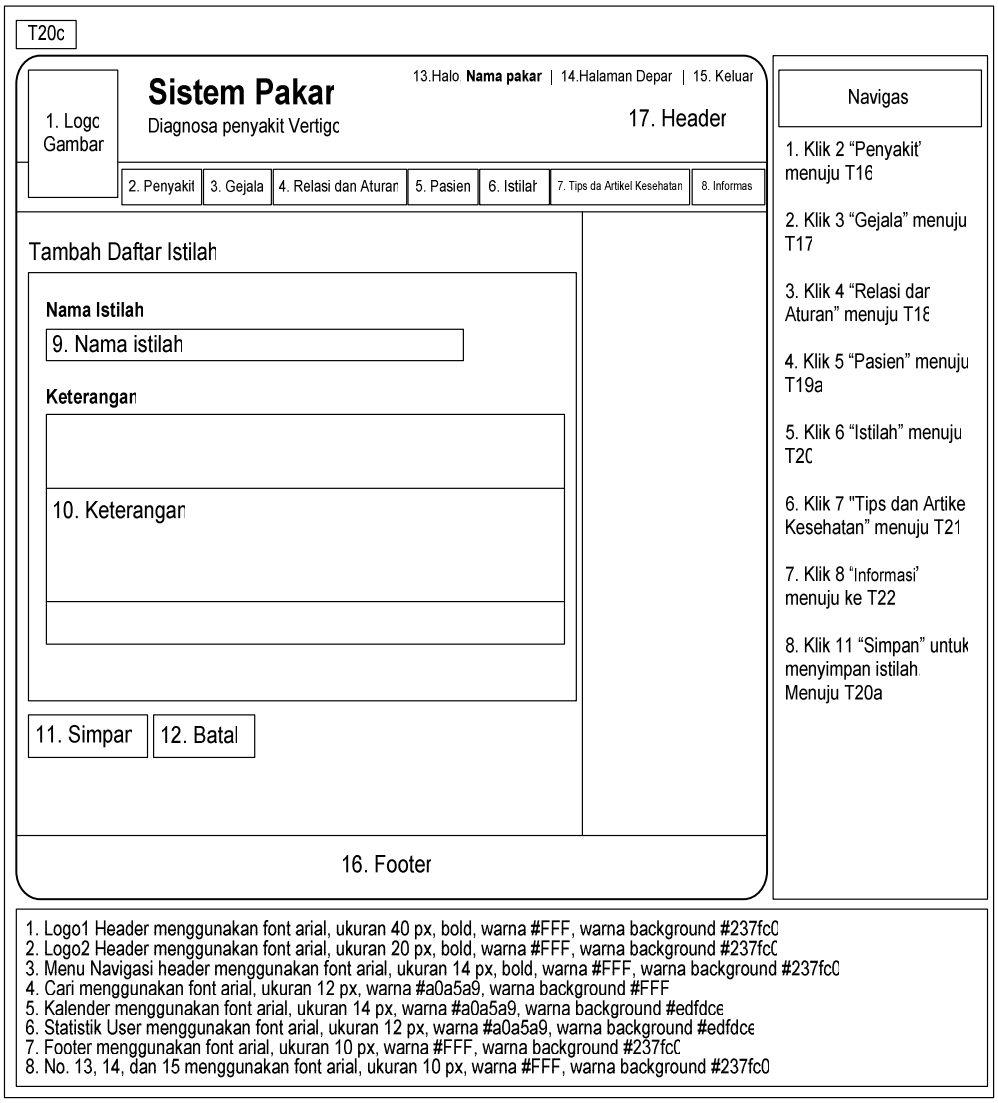
Gambar 3.59 Perancangan Validasi Halaman Hasil Pencarian Pasien

3.11.39Perancangan Halaman Daftar Istilah



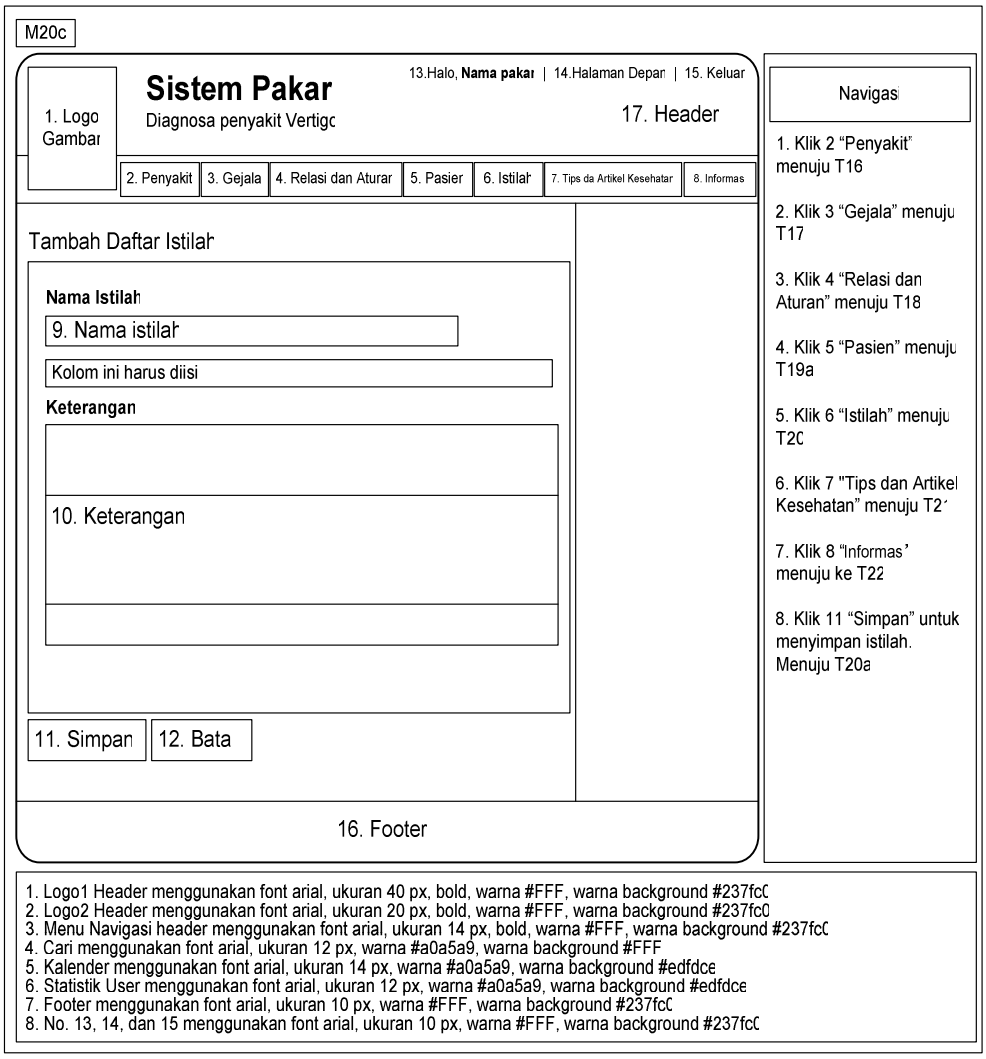
Gambar 3.60 Perancangan Halaman Daftar Istilah

3.11.40Perancangan Halaman Tambah Daftar Istilah



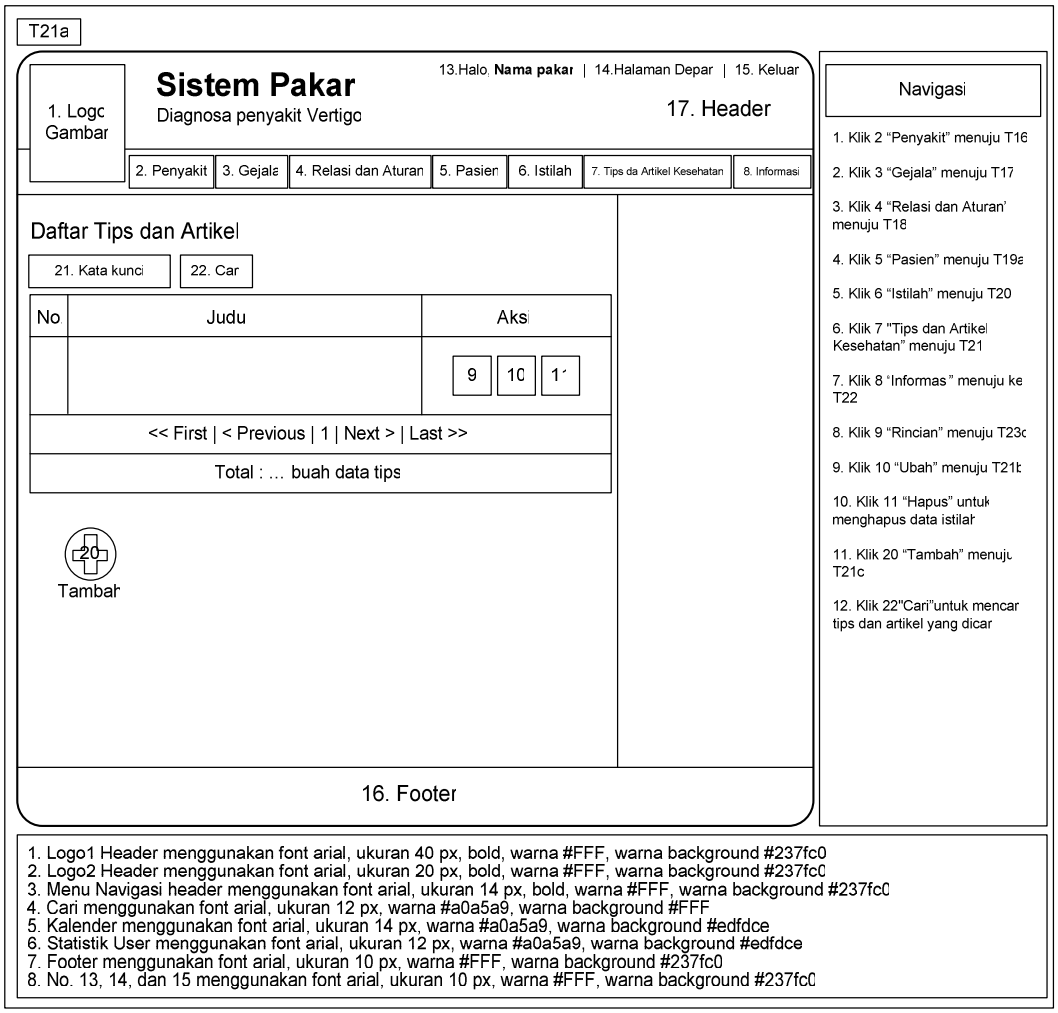
Gambar 3.61 Perancangan Halaman Tambah Daftar Istilah

3.11.41Perancangan Halaman Pesan Validasi Tambah Daftar Istilah



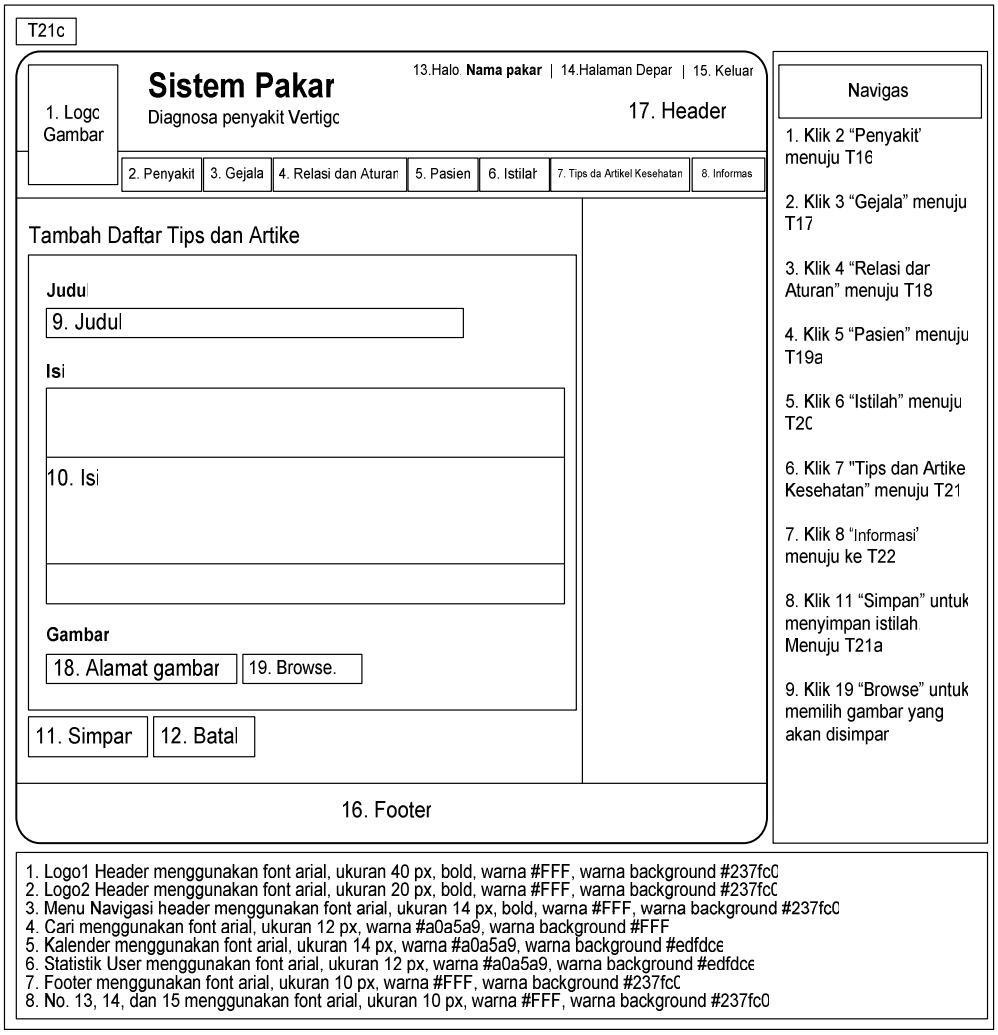
Gambar 3.62 Perancangan Halaman Pesan Validasi Tambah Daftar Istilah

3.11.42Perancangan Halaman Daftar Tips dan Artikel



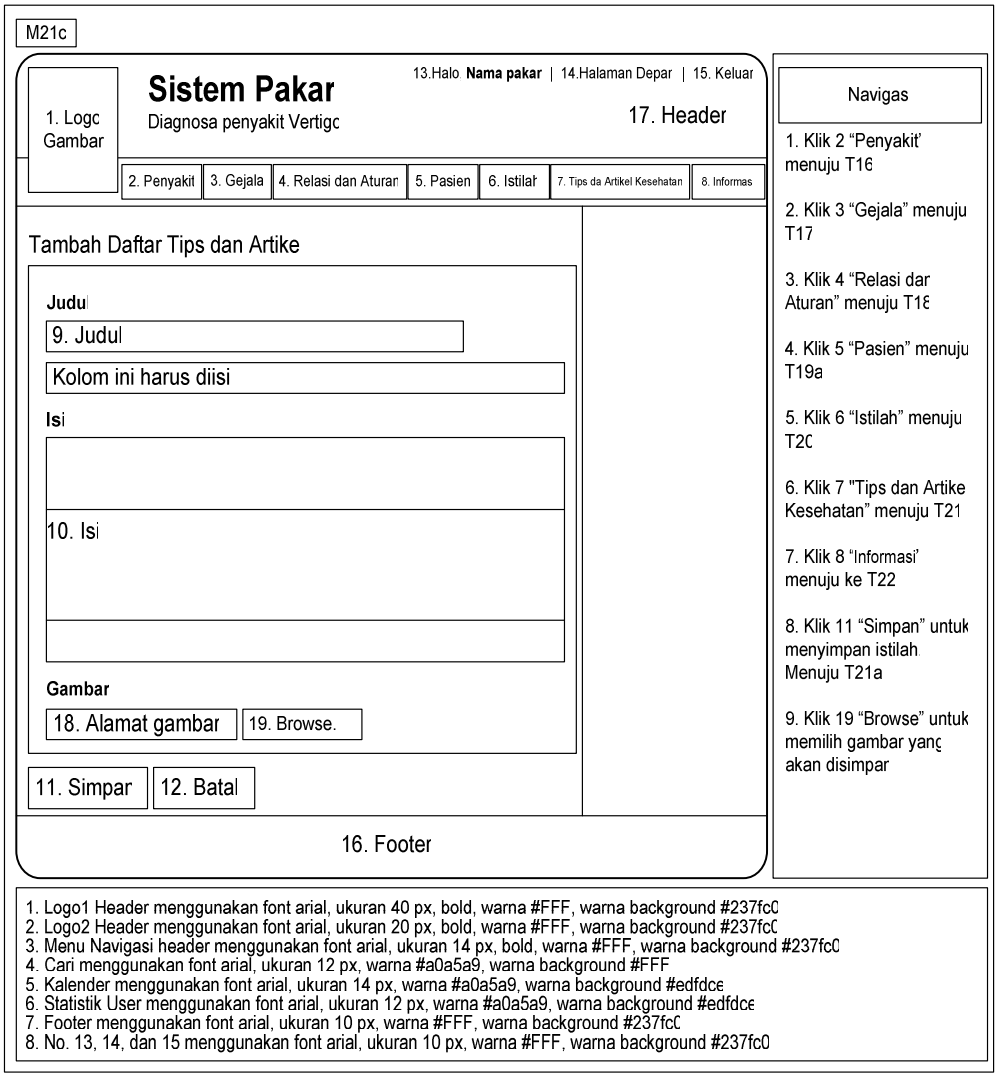
Gambar 3.63 Perancangan Halaman Daftar Tips dan Artikel

3.11.43Perancangan Halaman Tambah Daftar Tips dan Artikel



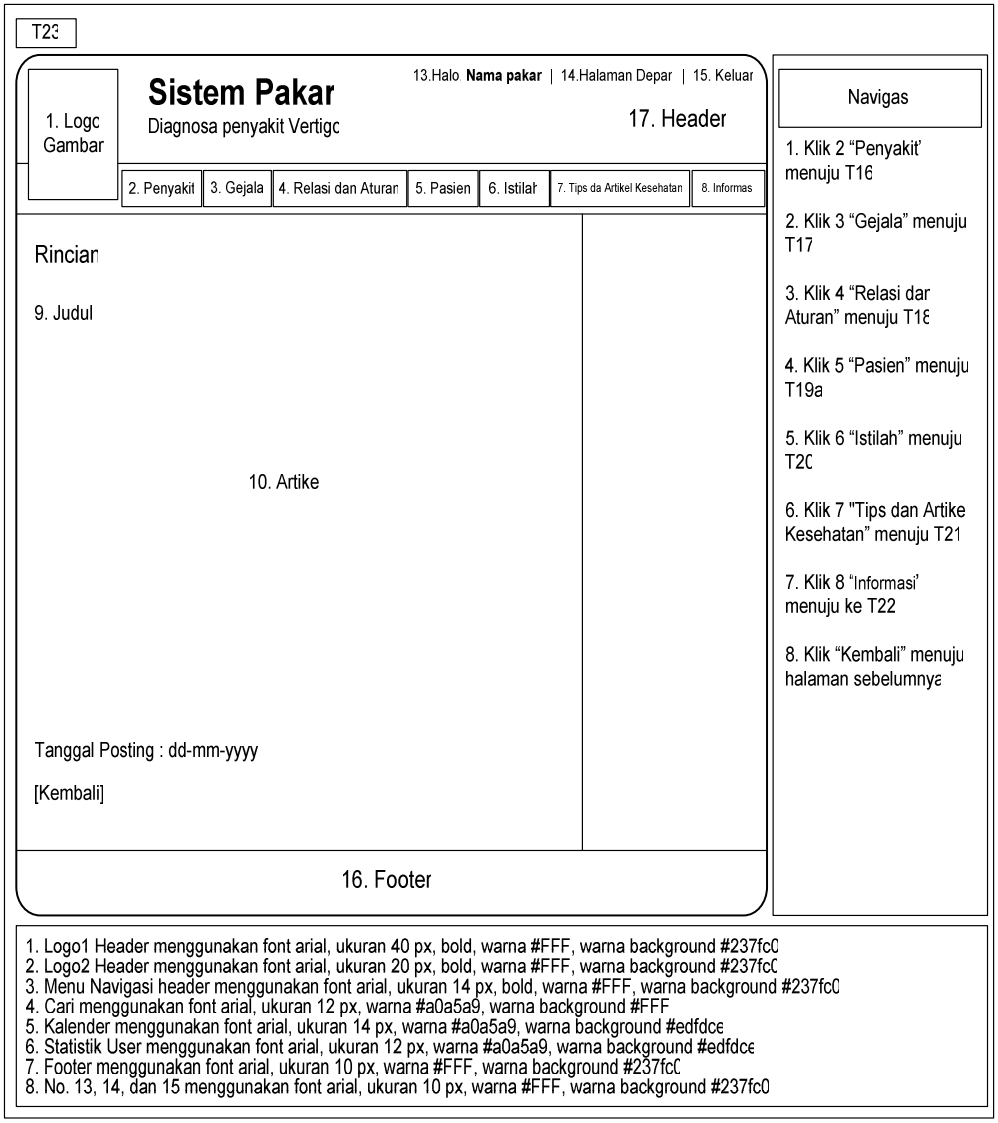
Gambar 3.64 Perancangan Tambah Halaman Daftar Tips dan Artikel

3.11.44Perancangan Halaman Pesan Validasi Tambah Daftar Tips dan Artikel



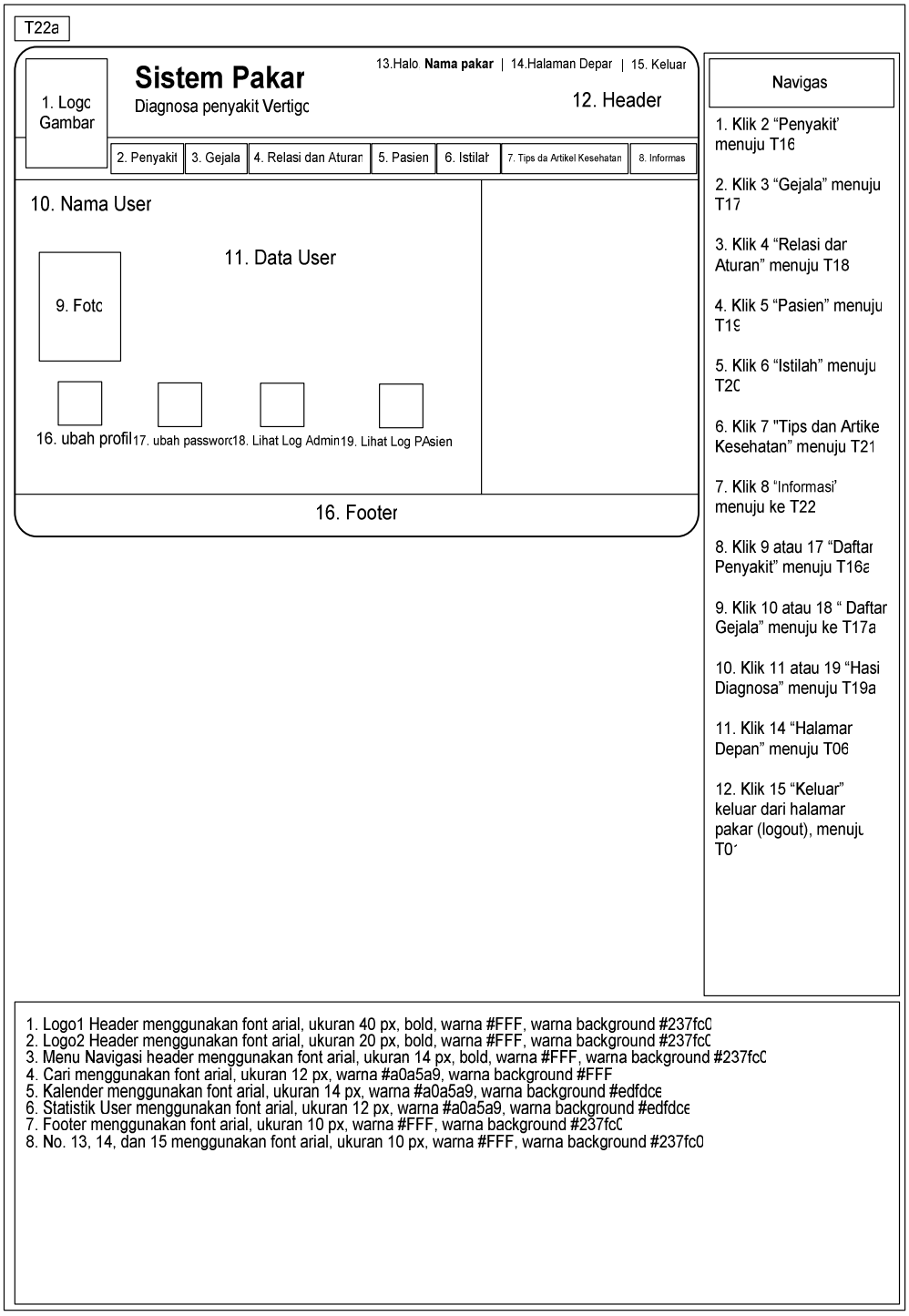
Gambar 3.65 Perancangan Pesan ValidasiTambah Halaman Daftar Tips dan Artikel

3.11.45Perancangan Halaman Detail



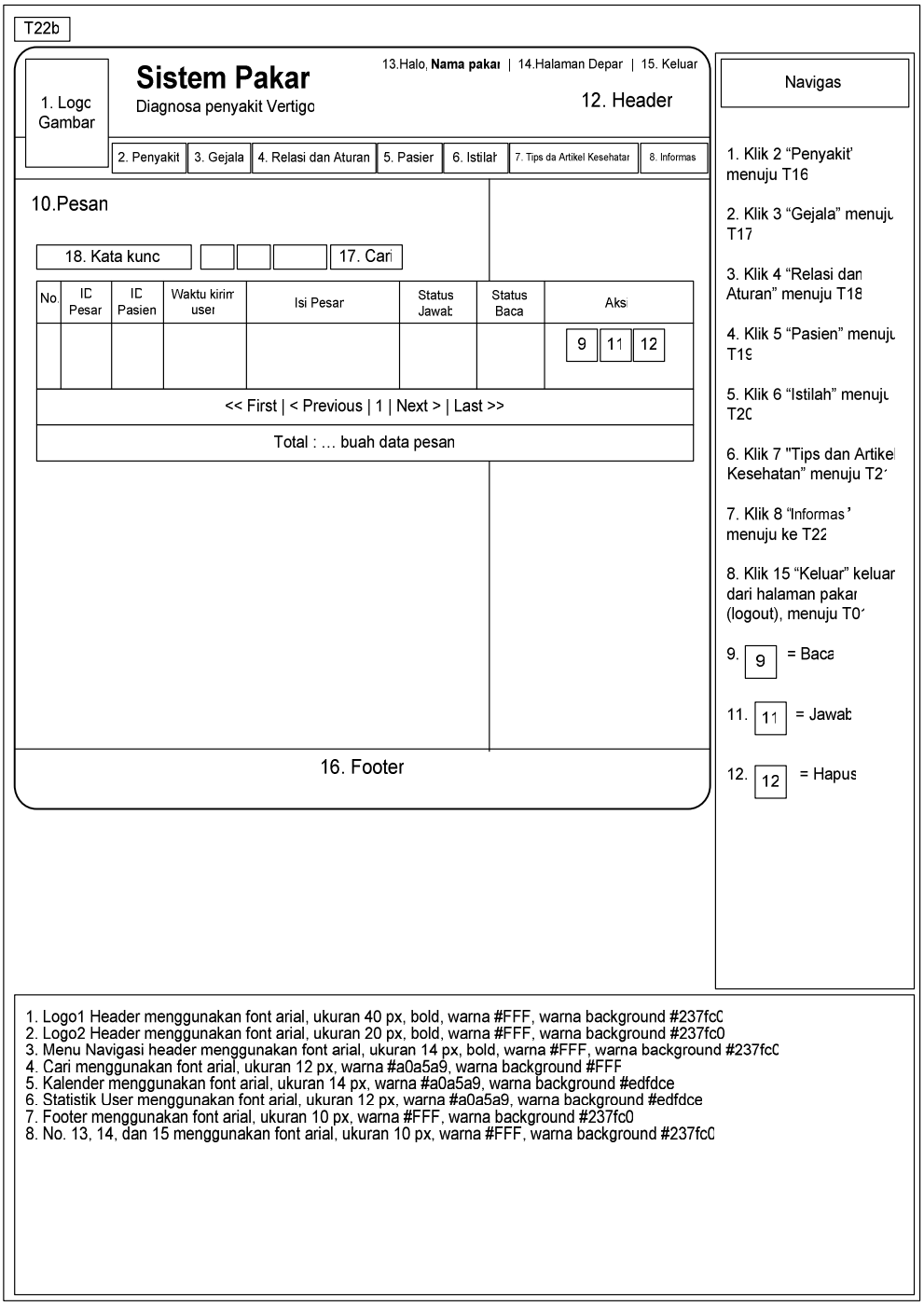
Gambar 3.66 Perancangan Pesan ValidasiTambah Halaman Daftar Tips dan Artikel

3.11.46Perancangan Halaman Informasi Profil



Gambar 3.67 Perancangan Halaman Informasi Profil

3.11.47Perancangan Halaman Informasi Pesan



Gambar 3.68 Perancangan Halaman Informasi Pesan

3.12 Persentase Keputusan

Persentase keputusan merupakan suatu kesimpulan hasil diagnosis berupa nominal atau nilai dalam bentuk persen (%) yang didapat dari jawaban *user* terhadap pertanyaan-pertanyaan yang diberikan. Sistematika perhitungannya adalah dengan menghitung total jumlah pertanyaan yang dijawab kemudian dibagi total keseluruhan pertanyaan yang terkait dengan penyakit yang didiagnosis kemudian hasil pembagian tersebut dikali 100. Berikut ini adalah rumus yang digunakan untuk menghitung kesimpulan hasil diagnosis :

$$PK = (JJ/JSG)*100$$

Keterangan :

PK : Persentase Keputusan hasil diagnosis dalam bentuk persen (%).

JJ : Jumlah Pertanyaan yang dijawab ”Ya”.

JSG : Jumlah Seluruh Pertanyaan yang terkait dengan penyakit yang didiagnosis.

Setelah persentase keputusan didapat, maka dapat ditentukan kesimpulan hasil diagnosis berdasarkan aturan sebagai berikut :

1. Jika persentase keputusan < 56%, maka kesimpulan yang didapat adalah ”Diragukan”.
2. Jika persentase keputusan >= 56% dan < 100%, maka kesimpulan yang didapat adalah ”Kemungkinan”.
3. Jika persentase keputusan = 100%, maka kesimpulan yang didapat adalah ”Dipastikan”.

3.13 Algoritma Tree

1. Penyisipan Data Penyakit (Tujuan/Akar)

```
{I.S. : Diberikan sejumlah data penyakit oleh user (admin)}
{F. S. : Menyisipkan data penyakit}
Kamus :
type
        DataPyt = record
        <kd_penyakit : integer,
        nm_penyakit, definisi, anjuran, terapi : string,
        skorpenyakit : integer,
        tanggal : date>
        penyakit : DataPyt
        i : integer

Algoritma :
if (penyakit.definisi=0) then
        output(”Kolom Definisi Belum diisi”)
 else
        input (penyakit(i).kd_penyakit)
        input (penyakit(i).nm_penyakit)
        input (penyakit(i).definisi)
        input (penyakit(i).anjuran)
        input (penyakit(i).terapi)
endif
```

2. Penyisipan Data Gejala (Node/Daun)

Teknik yang digunakan untuk menyisipkan data gejala baru adalah sebagai berikut :

1) Objek data yang akan disisipkan harus diciptakan.
2) Lakukan pemeriksaan, pohon itu masih kosong atau sudah berisi.
    a. Jika pohon itu masih kosong, lakukan langkah sedemikian rupa sehingga yang akan disisipkan itu menjadi sebuah node baru.
    b. Jika Pohon itu sudah berisi, lakukan konfirmasi dan selanjutnya lakukan peng-*update*-an terhadap urutan node yang sudah berisi

tersebut dengan cara menambahkan urutan node yang berisi dimana node yang berisi >= node baru yang akan disisipkan.

Berikut ini adalah algoritma yang diterapkan dari teknik yang digunakan untuk menyisipkan data :

```
{I.S. : Diberikan sejumlah data gejala oleh user (admin)}
{F. S. : Menyisipkan data gejala}
Kamus :
type
        DataGjl = record
        <kd_gejala, no_urut,skorgejala : integer,
        nm_gejala, pertanyaan, keterangan : string,
        >
        gejala : DataGjl
        posisi, i, cekno_urut : integer
        ketemu : boolean

Algoritma :
if (gejala(i).keterangan=0) then
        output(”Kolom Definisi Belum diisi”)
 else
        input(posisi)
        gejala(i).no_urut←posisi
        cekno_urut←gejala(i).no_urut
        ketemu←true
        if(cekno_urut=ketemu) then
            output(“Nomor Urut Gejala yang Anda masukkan sudah ada.”)
            output(“Anda ingin menyisipkan gejala yang ini ?”)
            if(jawab=”ya”) then
                while(gejala(i).no_urut>=posisi)do
                    gejala.no_urut← gejala.no_urut+1
                        input (gejala(i).kd_gejala)
                        input (gejala(i).no_urut)
                        input (gejala(i).nm_gejala)
                        input (gejala(i).skorgejala)
                        input (gejala(i).pertanyaan)
                        input (gejala(i).keterangan)
                endwhile
            else
                output(“Penyisipan dibatalkan”)
            endif
        else
                input (gejala(i).kd_gejala)
                input (gejala(i).no_urut)
                input (gejala(i).nm_gejala)
```

```
            input (gejala(i).skorgejala)
            input (gejala(i).pertanyaan)
            input (gejala(i).keterangan)
        endif
```

3. Hapus Data Gejala (Node/Daun)

Teknik yang digunakan untuk menghapus data gejala baru adalah sebagai berikut :

ii. Pilih objek data yang akan dihapus.

iii. Lakukan pengurangan (no. urut – 1) terhadap node yang no. urutnya > no. urut node yang akan dihapus.

iv. Lakukan penghapusan terhadap objek (node) yang dipilih.

Berikut ini adalah algoritma yang diterapkan dari teknik yang digunakan untuk menghapus data :

```
{I.S. : Dilakukan data gejala oleh user (admin)}
{F. S. : Menghapus data gejala}
Kamus :
type
        DataGjl = record
        <kd_gejala, no_urut,skorgejala : integer,
        nm_gejala, pertanyaan, keterangan : string,>
        gejala : DataGjl
        kd, posisi, i, cek : integer
        ketemu : boolean

Algoritma :
input(kd)
gejala(i).kd_gejala←kd
cek←gejala(i).kd_gejala
ketemu←true
if (cek=ketemu) then
    posisi←gejala(i).no_urut
    while(gejala(i).no_urut>posisi)do
        gejala(i).no_urut← gejala(i).no_urut-1
    endwhile
dealloc(gejala(i).kd_gejala)
endif
```

4 Ubah Data Gejala (Node/Daun)

Berikut ini adalah algoritma yang diterapkan dari teknik yang digunakan untuk mengubah data :

```
{I.S. : Dilakukan edit data gejala oleh user (admin)}
{F. S. : Mengubah data gejala}
Kamus :
type
        DataGjl = record
        <kd_gejala, no_urut,skorgejala : integer,
        nm_gejala, pertanyaan, keterangan : string,>
        gejala : DataGjl
        posisi, posisigejala, i, kd, tmp, tmp2, kdtmp2,tmk, cekno_urut : integer
        ketemu : boolean

Algoritma :
if (gejala(i).keterangan=0) then
        output(”Kolom Definisi Belum diisi”)
 else
        input(posisi)
        gejala(i).no_urut←posisi
        cekno_urut←gejala(i).no_urut
        ketemu←true
        if(cekno_urut=ketemu) then
            output(“Nomor Urut Gejala yang Anda masukkan sudah ada.”)
            output(“Anda ingin menyisipkan gejala yang ini ?”)
            if(jawab=”ya”) then
                input(kd)
                gejala(i).kd_gejala←kd
                posisigejala←gejala(i).no_urut
                if(posisigejala=posisi) then
                   input (gejala(i).kd_gejala)
                   input (gejala(i).no_urut)
                   input (gejala(i).nm_gejala)
                   input (gejala(i).skorgejala)
                   input (gejala(i).pertanyaan)
                   input (gejala(i).keterangan)
                 elseif(posisigejala>posisi) then
                   gejala(i).kd_gejala←kd
                   tmp← gejala(i).no_urut
                   if(tmp-posisi=1)then
                      gejala(i).no_urut←posisi
                      tmp2←gejala(i).no_urut
                      kdtmp2←gejala(i).kd_gejala

                      gejala(i).kd_gejala←kd
```

```
                    tmp←gejala(i).no_urut
                    kdtmp←gejala(i).kd_gejala
                    gejala(i+1).no_urut←tmp
                    gejala(i).no_urut←tmp2
                  endif
                  else
                    gejala(i).kd_gejala←kd
                    tmp←tmp
                    while(gejala(i).no_urut>=posisi)AND
                          (gejala(i).no_urut<=tmp)do
                          gejala(i).no_urut← gejala(i).no_urut+1
                    endwhile
                    while (gejala(i).kd_gejala=kd)do
                          gejala(i).no_urut←posisi
                    endwhile
                endif
                if(posisigejala<posisi)then
                    gejala(i).kd_gejala←kd
                    tmk←gejala(i).no_urut
          while(gejala(i).no_urut>tmk)AND(gejala(i).no_urut<=posisi)do
              gejala(i).no_urut← gejala(i).no_urut-1
          endwhile
          while(gejala(i).kd_gejala=kd)do
              gejala(i).no_urut=posisi
          endwhile
              endif
          else
              output("Penyisipan dibatalkan")
          endif
    else
          input (gejala(i).kd_gejala)
          input (gejala(i).no_urut)
          input (gejala(i).nm_gejala)
          input (gejala(i).skorgejala)
          input (gejala(i).pertanyaan)
          input (gejala(i).keterangan)
    endif
```

5. Relasi (Batang)

Berikut ini adalah algoritma yang diterapkan dari teknik yang digunakan untuk membuat batang :

{I.S. : Dilakukan proses penyimpan relasi antara data gejala dan data penyakit oleh *user* (admin)}
{F. S. : Menyimpan relasi antara data gejala dan data penyakit}

```
Kamus :
type
        DataRelasi = record
        <kd_gejala, kd_penyakit : integer>
        relasi : DataRelasi
        kdsakit, cekgejala, jum, i, n : integer
        cocok : boolean
Algoritma :
jum←cekgejala
if(jum=0)then
    output("Ulangi lagi")
endif
else{
    while(relasi.kd_penyakit=kdsakit)do
        for 1 to n do
            relasi.kd_penyakit←kdsakit
            relasi.kd_gejala<>cekgejala(i)
            dealloc(relasi.kd_penyakit)
        endfor
    endwhile
    for 1 to n do
        relasi.kd_penyakit←kdsakit
        relasi.kd_gejala←cekgejala(i)
    cocok←true
    if(cocok=false) then
        input(relasi.kd_penyakit←kdsakit)
        input(relasi.kd_gejala←cekgejala(i))
        gejala← cekgejala(i)
        output(gejala)
}
```