

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Secara umum sistem yang akan dibangun pada penelitian ini adalah aplikasi yang digunakan untuk menganalisa pola hasil diagnosa penyakit pasien puskesmas dari *database* yang telah dihimpun oleh Dinas Kesehatan Kabupaten Lamongan. Data tersebut terlebih dahulu dilakukan *preprocessing* data yaitu pembersihan data dari data yang tidak relevant atau data yang memiliki nilai tidak konsisten.

Setelah dilakukan proses *preprocessing*, data tersebut kemudian dilakukan query untuk menentukan dataset dari database tersebut. Aplikasi ini selanjutnya Melakukan penggalan data dengan algoritma GSP (Generalized Sequence Pattern) dengan mengoptimalkan query Database Management System (DBMS) untuk menghasilkan *frequent-n-sequence* beserta *sequential rule* untuk mengetahui urutan dari diagnosa penyakit pasien pada rekam medis elektronik berbasis web.

Sistem akan melakukan penggalan informasi bermanfaat (*interesting information*) dari dataset database rekam medis Dinas Kesehatan Kabupaten Lamongan. Dataset akan dilakukan analisa berdasarkan rentan waktu dan lokasi yang diinginkan, Misalnya data akan dianalisa berdasarkan dataset mulai tanggal 1 Januari 2009 hingga 31 Oktober 2009 karena data untuk sistem ini berdasarkan pada 1 Januari 2009 hingga 31 Oktober 2009 atau menganalisa dataset dari kecamatan di Kabupaten Lamongan. Sistem akan meminta dua buah inputan nilai, yaitu nilai minimal support dan nilai confidence, nilai tersebut dapat berupa nilai prosentase atau nilai 1-100. Sistem akan membuat tabel *sequential* diagnosa pasien yang kemudian diproses untuk mendapatkan *frequent-n-sequence* yaitu informasi pola urutan diagnosa penyakit yang memiliki nilai sama atau diatas batas nilai *support*, dan menggunakan *frequent-n-sequence* tersebut untuk mendapatkan *sequential rule* yang memiliki nilai sama atau diatas batas nilai *confidence*,

3.2 Analisa Kebutuhan Sistem

3.2.1 Kebutuhan Fungsional Sistem

Dari gambaran umum sistem tersebut, dapat diketahui kebutuhan-kebutuhan fungsional untuk aplikasi analisa pola penyakit pada rekam medis elektronik yang dikembangkan yaitu :

1. Sistem harus mampu menghasilkan *frequent sequence* dan *sequential rule* dari dataset.
2. Sistem harus melakukan penggalian informasi *bermanfaat(interesting information)* dari kedua itemset.
3. Sistem harus mampu memvisualisasikan hasil analisa dari penggalian informasi dalam bentuk report.

3.2.2 Kebutuhan Nonfungsional Sistem

1. Usability.

Usability adalah kebutuhan non fungsional terkait dengan kemudahan penggunaan sistem atau perangkat lunak oleh *user*.

2. Reliability.

Reliability yaitu kebutuhan terkait kehandalan sistem atau perangkat lunak termasuk juga faktor keamanan (*security*) sistem.

3. Portability.

Portability adalah kemudahan dalam pengaksesan sistem khususnya terkait dengan faktor waktu dan lokasi pengaksesan, serta perangkat atau teknologi yang digunakan untuk mengakses. Perangkat atau teknologi tersebut meliputi perangkat lunak, perangkat keras, dan perangkat jaringan.

4. Supportability.

Supportability adalah kebutuhan terkait dengan dukungan dalam penggunaan sistem atau perangkat lunak.

3.2.3 Kebutuhan Pembuatan Sistem

3.2.3.1 Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang diperlukan dalam pembangunan sistem adalah :

1. Sistem operasi Windows XP
2. PHP Versi 5.2 + Apache 2.2.0, tools untuk pengembangan aplikasi
3. MySQL Server 5, DBMS untuk media penyimpanan data.
4. Mozilla FireFox 3.1.6, browser untuk mengakses aplikasi
5. Power Designer 15, untuk desain DFD

3.2.3.2 Kebutuhan Perangkat Keras

Adapun perangkat keras yang diperlukan berdasarkan kebutuhan perangkat lunak diatas adalah :

1. Prosesor minimum Pentium 4 2.8 GHz
2. Memori minimum DDRAM 512 MB
3. Hardisk dengan kapasitas penyimpanan minimum 40 GB
4. Monitor SVGA (1024 x 800)
5. Keyboard, Mouse
6. Printer

3.3 Daftar Tabel

Tabel–tabel yang digunakan antara lain:

3.3.1 Tabel sekuensial

Tabel ini digunakan untuk menyimpan data sekuensial penyakit pasien. Tabel ini akan menyimpan no reg pasien (no_reg) dan diagnosa penyakit pasien keseluruhan(sekuensi). Deskripsi tabel sekuensial dapat dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi Tabel Sekuensial

Field	Type	Key	Extra	Keterangan
S_id	Int(11)	PRI	Auto_increment	Id pengenal sekuensi
No_reg	Varchar(30)			Id pasien
sekuensi	Varchar(50)			Diagnosa penyakit pasien keseluruhan
count	Int(11)			Nilai jumlah sekuensi

3.3.2 Tabel frequent

Tabel ini digunakan untuk menyimpan data candidate sequence beserta nilai supportnya. Deskripsi tabel frequent dapat dilihat pada Tabel 3.2.

Tabel 3.2 Deskripsi Tabel frequent

Field	Type	Key	Extra	Keterangan
C_id	Int(11)	PRI	Auto_increment	Id pengenal candidate
candidate	Varchar(50)			Candidate sequence
L	Int(11)			Nilai n-sequence
Support	Int(11)			Nilai Frekuensi kemunculan id_dx
frequent	Enum('Y','T')			Nilai apakah lolos menjadi frequent

3.3.3 Tabel rule

Tabel ini digunakan untuk menyimpan rule yang didapatkan. Tiap rule yang disimpan terdapat nilai confidencenya beserta nilai yang menandai apakah rule tersebut memenuhi dari nilai minimal confidence yang di masukkan oleh user. Penjelasan struktur tabel laporan asosiasi dapat dilihat pada Tabel 3.3.

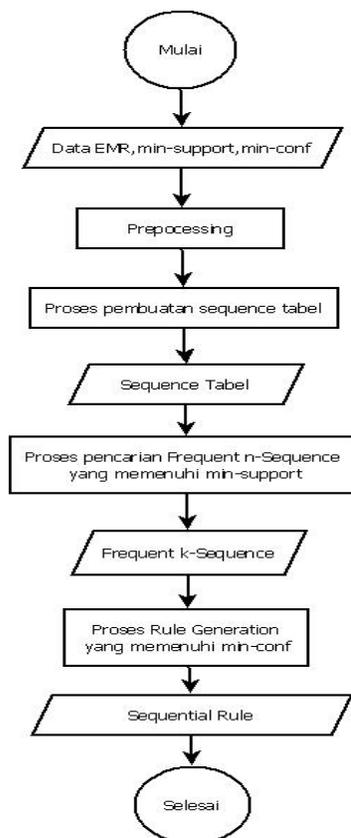
Tabel 3.3 Struktur Tabel rule

Field	Type	Key	Extra	Keterangan
R_id	Int(11)	PRI	Auto_increment	Id rule
rule	Varchar(50)			Sekuensial rule
confidence	float			Nilai confidence rule
memenuhi	Enum('Y','T')			Nilai apakah rule memenuhi nilai minimal confidence

3.4 Perancangan Sistem

3.4.1 Diagram Alir Sistem

Berdasarkan dari gambaran umum beserta data yang telah disiapkan diatas, maka sistem dapat dijelaskan dengan proses – proses sebagai berikut :



Gambar 3.1.Diagram Alir Proses Aplikasi Analisa pola urutan penyakit pasien Puskesmas dengan metode GSP (Generalized Sequence Pattern)

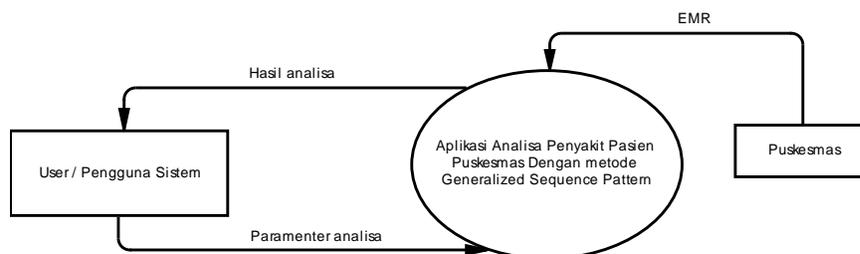
Adapun keterangan dari Gambar 3.1 adalah sebagai berikut :

Aplikasi ketika dimulai membutuhkan 3 buah inputan yaitu Data EMR, nilai min-support dan nilai min-conf. Kemudian Data EMR pada tabel transaksi dilakukan preprocessing dengan query database untuk menemukan data yang relevan dan juga user dapat memberikan parameter data yang berupa rentan waktu atau batas sekuensial pasien yang akan diuji atau diproses. Dari hasil data yang telah dilakukan pembatasan sebelumnya kemudian dibuatlah tabel sekuensial penyakit pasien. Dari tabel tersebut diproses dengan metode GSP untuk mendapatkan frequent sequence yang memenuhi nilai minimum support yang telah ditentukan. Selanjutnya dilakukan proses rule generation dari hasil sebelumnya yaitu frequent sequence dengan algoritma sequential rule. Selanjutnya rule yang dihasilkan diseleksi berdasarkan yang memenuhi nilai minimal confidence yang telah ditentukan. Hasil akhir berupa sequential rule yang merupakan aturan kemungkinan terkena suatu penyakit yang bersamaan atau setelahnya. Misalnya, Difteria→Otitis Media \Rightarrow Difteria→Otitis Media→Hipertensi (80%) maka jika seseorang terkena Difteria dan selanjutnya terkena Otitis Media mempunyai peluang 80% selanjutnya akan terkena Hipertensi. Dengan pengetahuan tersebut dapat dijadikan informasi tambahan bagi dokter untuk memperingatkan pasien.

3.4.2 Data Flow Diagram

3.4.2.1 Diagram Konteks

Berdasarkan dari arus proses data yang telah disiapkan diatas, maka sistem dapat dijelaskan dengan diagram konteks sebagai berikut :

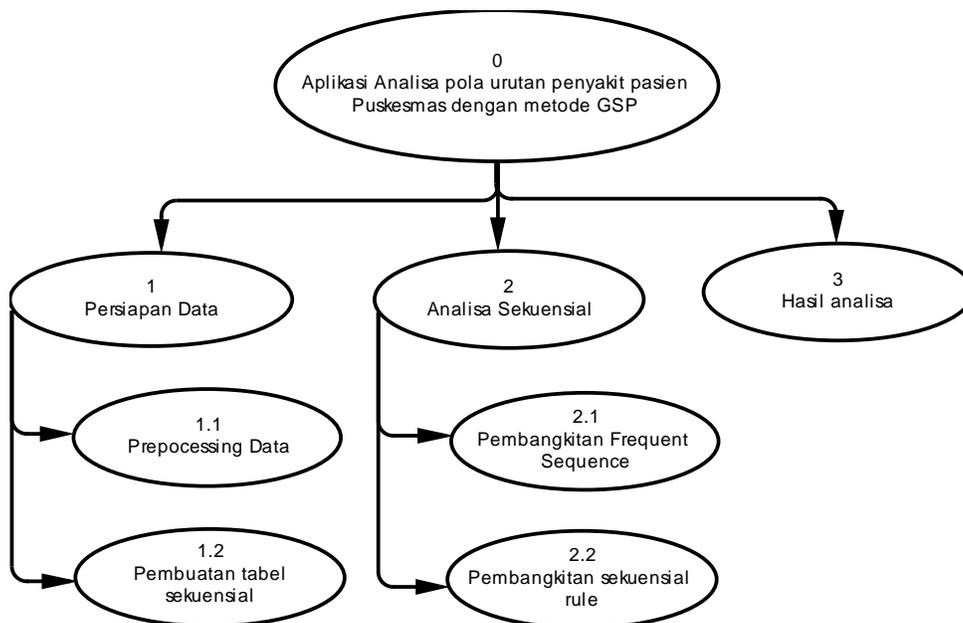


Gambar 3.2 Diagram Konteks Aplikasi Analisa pola urutan penyakit pasien Puskesmas dengan metode GSP (Generalized Sequence Pattern)

Adapun keterangan dari gambar 3.2 adalah sebagai berikut :

1. Dari entitas Puskesmas memberikan data EMR kepada sistem.
2. Dari entitas User atau pengguna sistem memberikan inputan berupa parameter analisa. Dengan data EMR yang diberikan oleh entitas Puskesmas dan parameter analisa yang dimasukkan *User* sistem akan memprosesnya dengan metode GSP. Kemudian sistem memberikan hasil dari analisa kepada user berupa *sequential rule* penyakit pasien yang memenuhi nilai *confidence*.

3.4.2.2 Diagram Hirarki



Gambar 3.3. Diagram Hirarki Aplikasi Analisa pola urutan penyakit pasien Puskesmas dengan metode GSP (Generalized Sequence Pattern)

Gambar 3.3 dapat dijelaskan sebagai berikut:

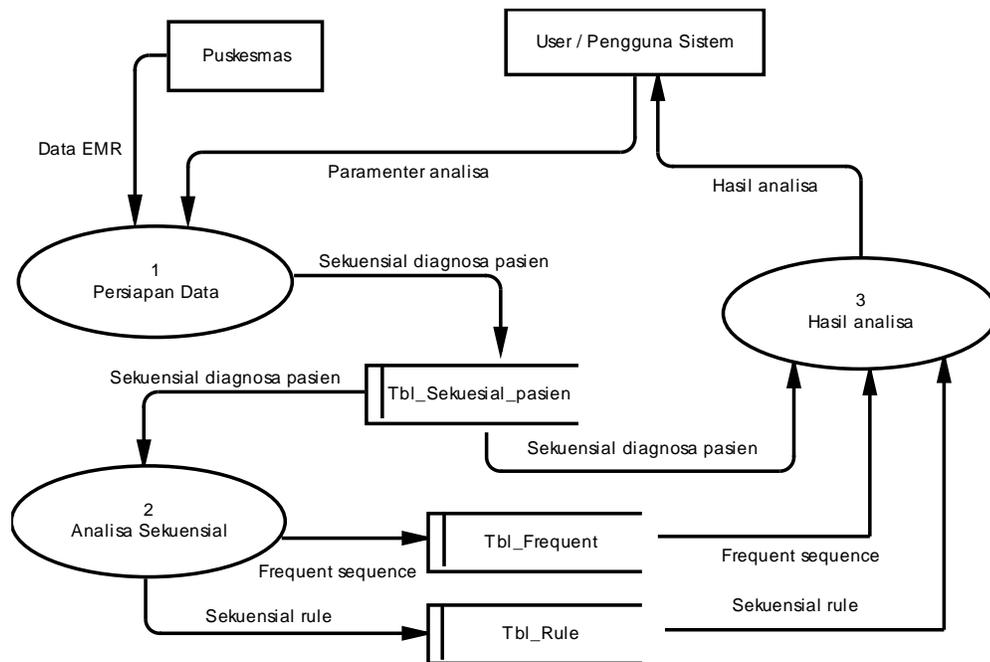
- Top Level : Aplikasi Analisa pola urutan penyakit pasien Puskesmas dengan metode GSP (Generalized Sequence Pattern)
- Level 0 : 1. Persiapan Data
- ❖ Preprocessing Data
 - ❖ Pembuatan table sekuensial

2. Analisa Sekuensial

- ❖ Pembangkitan Frequent Sequence
- ❖ Pembangkitan Sequential rule

3. Hasil analisa

3.4.2.3 DFD Level 0



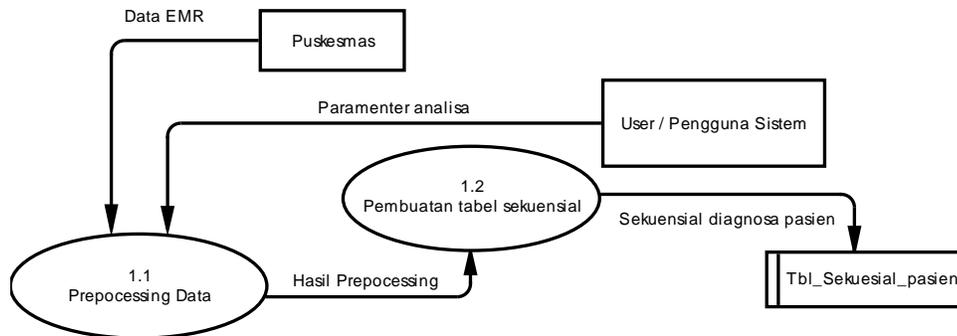
Gambar 3.4.DFD Level 0

Adapun keterangan dari gambar 3.4 adalah sebagai berikut :

1. Entitas Puskesmas merupakan entitas luar yang memberikan masukan berupa data EMR kemudian dengan parameter analisa yang diberikan oleh entitas *User* menghasilkan data sekuensial pasien disimpan di tabel sekuensial pasien dan digunakan untuk proses berikutnya.
2. Data sekuensial diagnosa pasien dari tabel sekuensial pasien dilakukan proses analisa sekuensial yang menghasilkan data *frequent sequence* yang disimpan di tabel frequent dan data sekuensial *rule* yang disimpan di tabel rule.
3. Data sekuensial diagnosa pasien, *frequent sequence* dan sekuensial *rule* diproses untuk ditampilkan dalam bentuk laporan kepada *user*.

3.4.2.4 DFD Level 1

3.4.2.4.1 DFD Level 1 Proses Persiapan Data

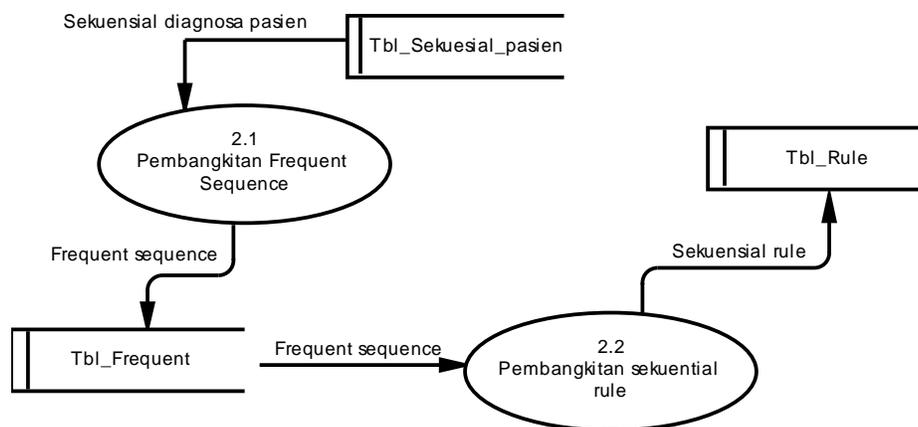


Gambar 3.5 DFD Level 1 Proses Preprocessing Data

Adapun keterangan dari gambar 3.5 adalah sebagai berikut :

- 1.1 Entitas Puskesmas yang merupakan entitas luar memberikan masukan berupa data EMR diproses dengan parameter analisa yang diberikan oleh entitas *User*.
- 1.2 Hasil dari Preprocessing Data diolah kedalam bentuk sekuensial yang berupa data sekuensial diagnosa pasien dan disimpan kedalam tabel sekuensial pasien.

3.4.2.4.2 DFD Level 1 Proses Analisa Sekuensial



Gambar 3.6 DFD Level 1 Proses Analisa Sekuensial

Adapun keterangan dari gambar 3.6 adalah sebagai berikut :

- 2.1 Sistem membangkitkan frequent sequence dari tabel sekuensial yang memenuhi nilai support yang merupakan parameter analisa yang dimasukkan oleh entitas *User* dan hasilnya disimpan dalam tabel frequent.
- 2.2 Sistem mengolah frequent sequence dari table frequent untuk membangkitkan sekuensial rule yang kemudian diseleksi berdasarkan nilai confidence yang merupakan parameter analisa yang dimasukkan oleh entitas *User* dan hasilnya disimpan dalam tabel rule.

3.5 Representasi Data

3.5.1 Proses pembuatan tabel *Sequence*

Sebelum dibuat tabel sequence data transaksi awal dilakukan preprocessing data yaitu menghilangkan data-data yang kurang relevan atau data transaksi yang berisi informasi yang kurang. Setelah itu dilakukan proses pembuatan tabel *sequence* merupakan langkah awal agar algoritma GSP dapat dilakukan. Dikarenakan tabel tersebut yang dijadikan dasar dari setiap proses.

Langkah pembuatan tabel *sequence* adalah sebagai berikut :

1. Tabel transaksi awal dikelompokkan berdasarkan pasien dan kemudian diurutkan berdasarkan tanggal transaksi.
2. Dari pengelompokkan tersebut dibuatlah tabel *sequence* , yaitu kode diagnosa penyakit pasien di gabung dan dengan tanda hubung '→' untuk pemisah tiap transaksi yang berarti juga urutan transaksi pasien.

Tabel transaksi yang diambil secara acak dalam data set yang telah dikelompokkan berdasarkan pasien dan juga diurutkan berdasarkan waktu transaksinya diperlihatkan pada **Tabel 3.4**.

Tabel 3.4. Tabel Transaksi dikelompokkan berdasar pasien dan diurutkan berdasarkan waktu transaksi

id_reg	no_reg	tgl_kunj	dx_utama
367	KARANGBINANGUNC0084	4/23/2009	2100
2341	KARANGBINANGUNC0084	6/5/2009	2200
3322	KARANGBINANGUNC0084	6/19/2009	1302

id_reg	no_reg	tgl_kunj	dx_utama
4730	KARANGBINANGUNC0084	7/24/2009	2100
5248	KARANGBINANGUNC0084	8/14/2009	1005
122	KARANGBINANGUND0008	4/15/2009	4102
1104	KARANGBINANGUND0008	5/8/2009	1302
1214	KARANGBINANGUND0008	5/8/2009	1302
2827	KARANGBINANGUND0008	6/5/2009	4102
3047	KARANGBINANGUND0008	6/5/2009	4102
1241	KARANGBINANGUNO0010	5/7/2009	2100
1994	KARANGBINANGUNO0010	5/29/2009	1504
1526	KARANGBINANGUNO0129	5/13/2009	1302
1874	KARANGBINANGUNO0129	5/23/2009	1504
6491	MANTUP00020	2/23/2009	1504
6492	MANTUP00020	3/24/2009	2001
6493	MANTUP00020	8/25/2009	4801
7239	MANTUP00160	2/25/2009	2100
7240	MANTUP00160	4/22/2009	2100
7241	MANTUP00160	6/13/2009	1200
7242	MANTUP00160	6/16/2009	1303
7243	MANTUP00160	8/22/2009	1302
9544	MANTUP00689	3/13/2009	102
9545	MANTUP00689	6/9/2009	1303
9546	MANTUP00689	6/22/2009	1504
10401	MANTUP00968	3/20/2009	1504
10402	MANTUP00968	5/22/2009	1503
10519	MANTUP00968	8/13/2009	1005
11284	MANTUP01226	3/30/2009	1504
11285	MANTUP01226	6/30/2009	1504
11286	MANTUP01226	7/6/2009	1502
11432	MANTUP01336	4/1/2009	1002
11433	MANTUP01336	4/15/2009	2100

id_reg	no_reg	tgl_kunj	dx_utama
11434	MANTUP01336	6/29/2009	1005
11435	MANTUP01336	9/16/2009	1303
11436	MANTUP01336	10/16/2009	1405
12064	MANTUP01550	4/6/2009	1303
12065	MANTUP01550	4/22/2009	1005
12066	MANTUP01550	5/22/2009	2001
12067	MANTUP01550	8/20/2009	1303
13147	MANTUP02053	4/20/2009	1005
13149	MANTUP02053	4/29/2009	1405
13148	MANTUP02053	5/2/2009	1303
13155	MANTUP02053	6/25/2009	1505
21555	MANTUPe0038	7/27/2009	4102
21556	MANTUPE0038	8/18/2009	1302
21557	MANTUPE0038	8/26/2009	102
21558	MANTUPE0038	9/3/2009	4102
21559	MANTUPE0038	10/18/2009	1303
22988	MOD000247	3/12/2009	1504
23424	MOD000247	4/3/2009	1302
23959	MOD000247	6/1/2009	1200
24392	MOD000247	7/1/2009	1302
25468	MOD000247	9/28/2009	1504
23931	MOD000292	5/22/2009	1302
24295	MOD000292	6/19/2009	1302
24384	MOD000292	6/29/2009	102
24864	MOD000292	8/5/2009	1302
25515	MOD000292	10/7/2009	1303
23249	MOD000352	3/27/2009	102
23471	MOD000352	4/6/2009	1302
23938	MOD000352	5/11/2009	4102
23900	MOD000352	5/20/2009	1101

id_reg	no_reg	tgl_kunj	dx_utama
26187	MODO00352	10/30/2009	2100
22947	MODO00358	3/10/2009	1403
23683	MODO00358	4/16/2009	1302
23682	MODO00358	4/16/2009	1302
23679	MODO00358	4/16/2009	1504
23956	MODO00358	5/25/2009	1302
28670	SUKODADI00897	9/1/2009	6814
28849	SUKODADI00897	9/2/2009	1302
29427	SUKODADI00897	10/14/2009	2001
29616	SUKODADI00897	10/28/2009	1302
29803	SUMBERAJI00124	7/13/2009	1302
29973	SUMBERAJI00124	7/14/2009	5104
29776	SUMBERAJI00124	7/22/2009	1302
30950	SUMBERAJI00124	9/14/2009	2100
29938	SUMBERAJI00144	7/13/2009	1302
29794	SUMBERAJI00144	7/18/2009	2100
30325	SUMBERAJI00144	8/6/2009	1302
30487	SUMBERAJI00144	8/10/2009	1302

Dari Tabel transaksi pada **Tabel 3.4**, dapat diketahui jumlah transaksi = 81 dan hanya 1 buah item pertransaksi. Setelah terkelompok maka dapat dilanjutkan dengan membuat tabel *sequence* seperti yang terlihat pada **Tabel 3.5**.

Tabel 3.5 Tabel *sequence*

S_id	no_reg	sequential
1	KARANGBINANGUNC0084	1005→2100→1302→2200→2100
2	KARANGBINANGUND0008	4102→4102→1302→1302→4102
3	KARANGBINANGUNO0010	2100→1504
4	KARANGBINANGUNO0129	1302→1504
5	MANTUP00020	1504→2001→4801

S_id	no_reg	sequential
6	MANTUP00160	1302→1303→1200→2100→2100
7	MANTUP00689	1504→1303→0102
8	MANTUP00968	1005→1503→1504
9	MANTUP01226	1502→1504→1504
10	MANTUP01336	1405→1303→1005→2100→1002
11	MANTUP01550	1303→2001→1005→1303
12	MANTUP02053	1505→1303→1405→1005
13	MANTUPe0038	4102→1302→0102→4102→1303
14	MODO00247	1504→1302→1200→1302→1504
15	MODO00292	1303→1302→0102→1302→1302
16	MODO00352	2100→1101→4102→1302→0102
17	MODO00358	1302→1504→1302→1302→1403
18	SUKODADI00897	6814→1302→2001→1302
19	SUMBERAJI00124	1302→5104→1302→2100
20	SUMBERAJI00144	1302→1302→2100→1302

3.5.2 Proses pencarian *Frequent n-Sequence*

Proses pencarian *frequent n-sequence* merupakan tujuan dari algoritma GSP. Memproses tabel *sequence* berulang-ulang untuk mendapatkan *frequent n-sequence* yang sesuai dengan nilai *min-support*.

Langkah pencarian *frequent n-sequence* adalah sebagai berikut :

1. Dimulai dengan $n=1$, yaitu mencari *frequent 1-sequence*.
2. Melakukan perhitungan *support* dari *candidate 1-sequence*.
3. Seleksi *candidate 1-sequence* yang memenuhi nilai *min-support* yang kemudian akan menjadi *frequent 1-sequence*.
4. Setelah *frequent 1-sequence* ditemukan maka akan dilanjutkan dengan *candidate generation 2-sequence*.
5. *Candidate* didapatkan dengan dengan *join* item *frequent 1-sequence* sebagai *candidate 2-sequence*.

6. Seleksi *candidate 2-sequence* yang memenuhi nilai *min-support* yang kemudian akan menjadi *frequent 2-sequence*.
7. Untuk mendapatkan *candidate ke-n* maka dilakukan *join frequent n-1-sequence*, dan kemudian dilakukan seleksi *min-support*.
8. Proses tersebut berulang sampai tidak ditemukan *candidate sequence* atau *frequent sequence*.

Dari tabel *sequence* sebelumnya dilakukan perhitungan *support candidate 1-sequence* seperti pada **Tabel 3.6**.

Tabel 3.6. *Candidate 1-sequence* beserta nilai *support*

Item	Support	Item	Support
0102	4	1503	1
1002	1	1504	8
1005	5	1505	1
1101	1	2001	3
1200	2	2100	7
1302	12	2200	1
1303	7	4102	3
1403	1	4801	1
1405	2	5104	1
1502	1	6814	1

Jika nilai *min-support* yang dimasukkan oleh user adalah 20% yang berarti 4 maka dari **Tabel 3.6** yang memenuhi nilai *min-support* seperti pada **Tabel 3.7**.

Tabel 3.7 *Frequent 1-sequence*

Item	support	Item	support
102	4	1303	7
1005	5	1504	8
1302	13	2100	7

Pada **Tabel 3.7** menunjukkan *candidate 1-sequence* yang memenuhi nilai *min-support* akan menjadi *frequent 1-sequence*. Proses dilanjutkan yaitu mencari

candidate 2-sequence yang diperoleh dari menggabungkan item *frequent 1-sequence*. Hasil dari penggabungan seperti terlihat pada **Tabel 3.8**.

Tabel 3.8 *Candidate 2-sequence* beserta nilai support.

Item	Support	Item	Support
(0102 , 1005)	0	1005 → 2100	2
(0102 , 1302)	0	1302 → 1303	2
(0102 , 1303)	0	1302 → 1504	3
(0102 , 1504)	0	1302 → 2100	4
(0102 , 2100)	0	1303 → 1504	0
(1005 , 1302)	0	1303 → 2100	2
(1005 , 1303)	0	1504 → 2100	0
(1005 , 1504)	0	1005 → 102	0
(1005 , 2100)	0	1302 → 102	2
(1302 , 1303)	0	1303 → 102	2
(1302 , 1504)	0	1504 → 102	1
(1302 , 2100)	0	2100 → 102	1
(1504 , 2100)	0	1302 → 1005	0
102 → 1005	0	1303 → 1005	2
102 → 1302	1	1504 → 1005	0
102 → 1303	1	2100 → 1005	0
102 → 1504	0	1303 → 1302	1
102 → 2100	0	1504 → 1302	2
1005 → 1302	1	2100 → 1302	3
1005 → 1303	1	1504 → 1303	1
1005 → 1504	1	2100 → 1303	0
		2100 → 1504	1

Pada **Tabel 3.8** terdapat 43 *candidate 2-sequence*, yang diperoleh dengan menggabungkan tiap item yang terdapat pada *frequent 1-sequence*. Berdasarkan nilai *min-support* sebelumnya maka akan ditemukan 1 *frequent 2-sequence* seperti pada **Tabel 3.9**.

Tabel 3.9 *Frequent 2-sequence.*

Item	Support
1302 → 2100	4

Proses berhenti dikarenakan tidak bisa ditemukan candidate untuk 3-sequence. Pada **Tabel 3.10** ditunjukkan hasil pencarian Frequent n-sequence.

Tabel 3.10 *Frequent k-sequence.*

N-Freq Itemset	Frequent Sequence dengan support
L1	{(0102)[4],(1005)[5],(1302)[13],(1303)[7],(1504)[8],(2100)[7]}
L2	{(1302→2100)[4]}

3.5.3 Proses Rule Generation

Setelah ditemukan Frequent n-sequence, maka dapat digunakan untuk mencari *rule* dengan syarat minimal terdapat *frequent 2-sequence* atau L2.

Rule diperoleh dengan menghubungkan *subsequence* dari *frequent sequence* dengan *frequent-sequence*. Hasil dari rule generation dari proses sebelumnya terlihat pada **Tabel 3.11**.

Tabel 3.11 *Rule Generation dengan confidence.*

rule_id	Rule	Confidence
1	1302=>1302→2100	4 / 13 = 31 %

Pada **Tabel 3.11** hanya terdapat 1 rule dengan tingkat confidence 31%. Jika user memasukkan nilai min-conf diatas 31% maka tidak ada rule yang dihasilkan.

3.6 Persiapan Data (*Data Preparation*)

Data yang didapat dari Dinas Kesehatan Kabupaten Lamongan berupa database Status_penderita berupa database acces yang memiliki 55 Tabel, namun tidak semua Tabel yang digunakan dalam penelitian ini. Beberapa Tabel diambil

dari database tersebut kemudian dikonvesitkan menjadi Tabel dalam database Mysql, Tabel yang digunakan dalam penelitian ini yaitu :

1. Tabel Master Diagnosa

Tabel master diagnosa menyimpan kode diagnosa dari Dinas Kesehatan Kabupaten Lamongan dengan nama field kode_dx, kode ICD X pada field kode_icdx dan nama penyakit. Field kode_dx ini yang akan dijadikan bahan analisa diagnosa penyakit untuk mengetahui pola penyakit. Struktur dari Tabel master Diagnosa dapat dilihat di **Tabel 3.12**

Tabel 3.12. Struktur penyimpanan data diagnosa pada Tabel master diagnosa

Nama Field	Type	Null	Key
Id_idx	int(11)	No	
Kode_dx	int(4)	No	Pri
kode_icdx	Varchar(20)	Yes	
Nama_penyakit	Varchar(200)	Yes	

2. Tabel Master Pasien

Data mengenai pasien yang menjadi stakeholder dari data rekam medis Dinas Kesehatan Kabupaten Lamongan terekam dalam Tabel master pasien. Struktur dari Tabel master pasien dapat dilihat di **Tabel 3.13.**

Tabel 3.13. Struktur penyimpanan data pasien pada Tabel master pasien

Nama Field	Type	Null	Key
Id_pas	Int (11)	No	
Puskesmas	Varchar(50)	Yes	
No_reg	Varchar(30)	No	Pri
Nama_pas	Varchar(50)	Yes	
Nama_kk	Varchar(50)	Yes	
Kecamatan	Varchar(20)	Yes	
Desa	Varchar(20)	Yes	
Dusun	Varchar(30)	Yes	

Nama Field	Type	Null	Key
Tgl_lahir	Varchar(10)	Yes	
Jk	Char(1)	Yes	
Tgl_daftar	Varchar(10)	Yes	
Petugas	Varchar(11)	Yes	
Pendidikan	Varchar(20)	Yes	
Pekerjaan	Varchar(15)	Yes	

3. Tabel Register Harian

Tabel register harian merupakan Tabel yang menyimpan transaksi rekam medis harian dari setiap puskesmas di Kabupaten Lamongan. Tabel ini sebagai Tabel acuan untuk proses analisa diagnosa penyakit. Struktur dari Tabel register harian dapat dilihat di **Tabel 3.14**.

Tabel 3.14 Struktur Tabel Register Harian Rekam Medis

Nama Field	Type	Null	Key
Id_reg	Int(11)	No	Pri
No_reg	Varchar(30)	Yes	
Tgl_kunj	Date	Yes	
Rujukan	Varchar(100)	Yes	
Tempat_pel	Varchar(50)	Yes	
Ruangan	Varchar(5)	Yes	
Cara_bayar	Varchar(5)	Yes	
Kasus	Varchar(5)	Yes	
Anamnesa	Varchar(255)	Yes	
Pemeriksaan_fisik	Varchar(255)	Yes	
Dx_utama	Varchar(50)	Yes	
Dx_kedua	Varchar(50)	Yes	
Tindakan	Varchar(255)	Yes	
Injeksi	Varchar(20)	Yes	
Tx1	Varchar(15)	Yes	

Nama Field	Type	Null	Key
Tx2	Varchar(15)	Yes	
Tx3	Varchar(15)	Yes	
Tx4	Varchar(15)	Yes	
Tx5	Varchar(15)	Yes	
Tx6	Varchar(15)	Yes	
Tx7	Varchar(15)	Yes	
Tx8	Varchar(15)	Yes	
Tx9	Varchar(15)	Yes	
Tx10	Varchar(15)	Yes	
Kode_pet	Varchar(20)	Yes	

4. Tabel *User*

Untuk menjaga keamanan dari Aplikasi ini maka dibuatlah *user* sendiri untuk bisa secara khusus mengakses aplikasi ini. Data daripada *user* tersebut tersimpan dalam Tabel *user*. Adapun Struktur dari Tabel *user* dapat dilihat di **Tabel 3.15**.

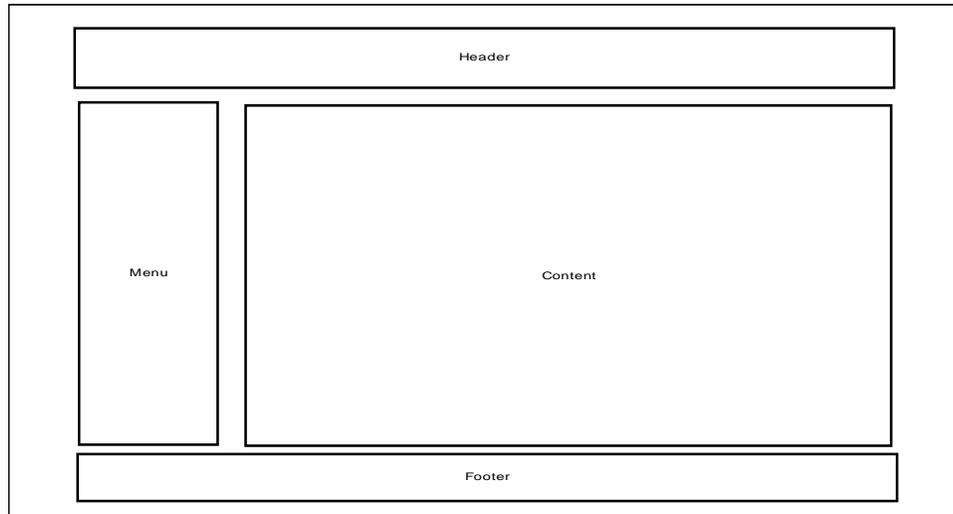
Tabel 3.15 Struktur Tabel *User*

Nama Field	Type	Null	Key
<i>Id_user</i>	int(11)	No	Pri
Nama	Varchar(30)	No	
<i>Username</i>	Varchar(25)	No	
Pass	Varchar(25)	No	

3.7 Desain Antarmuka Sistem

Antarmuka merupakan bagian yang menghubungkan user dengan sistem untuk melakukan proses analisa dengan menggunakan metode Generalized Sequence Pattern untuk didapatkan informasi baru berupa pola urutan penyakit pasien dari dataset yang diambil dari data EMR. Sistem ini berbasis Web dengan bahasa pemrograman yang dipakai menggunakan PHP. Interface juga

memberikan kemudahan pada user untuk menjalankan sistem, dan halaman yang akan dibuat adalah sebagai berikut :



Gambar 3.7 Layout desain Aplikasi Analisa pola urutan penyakit pasien Puskesmas dengan metode GSP (Generalized Sequence Pattern)

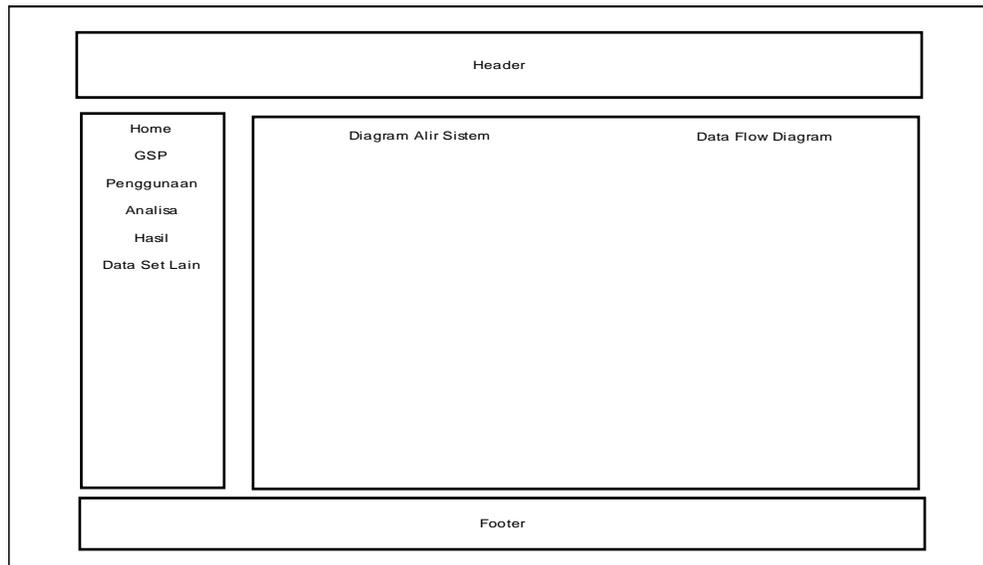
Adapun penjelasan dari gambar diatas sebagai berikut:

1. Header, merupakan tempat yang berisi gambar atau tulisan yang mempunyai arti yang terkait dengan aplikasi ini.
2. Menu, merupakan tempat yang berisi pilihan untuk menjalankan perintah berdasarkan menu tersebut.
3. Content, merupakan tempat yang berisi sesuai dengan menu yang dipilih.
4. Footer, merupakan tempat yang berisi dengan sesuatu yang berhubungan dengan pembuat sistem.

3.7.1 Halaman Home

Pada halaman ini merupakan halaman awal yang muncul setelah user sukses melakukan login. Rancangan antar muka halaman home dapat dilihat pada

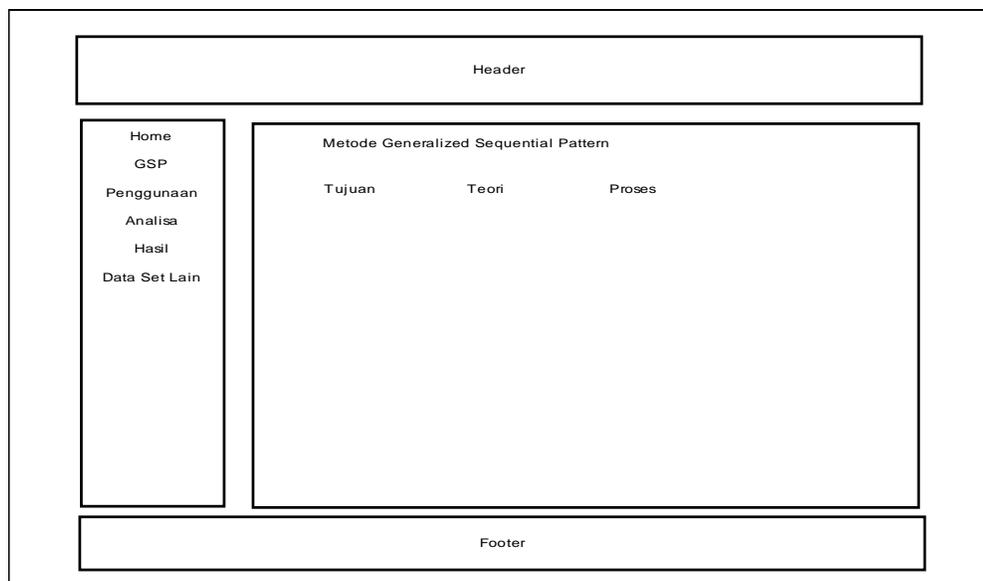
Gambar 3.8.



Gambar 3.8 Halaman Home

3.7.2 Halaman Generalized Sequential Pattern

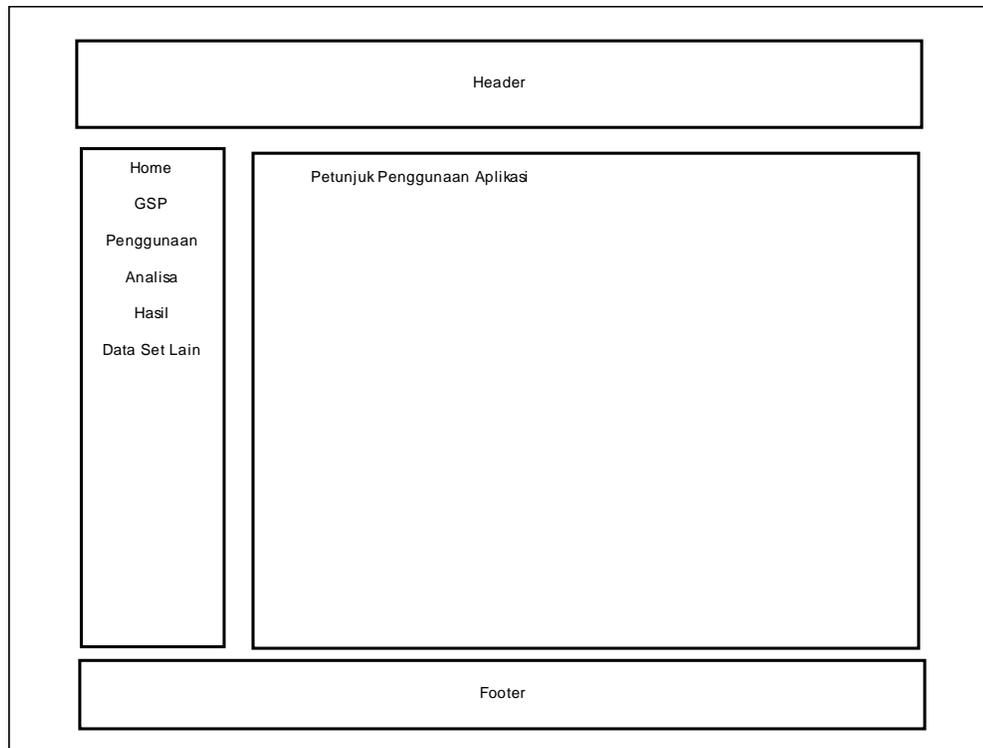
Pada halaman ini berisi penjelasan tentang metode generalized sequential pattern. Rancangan antar muka dari halaman generalized sequential pattern dapat dilihat pada **Gambar 3.9**.



Gambar 3.9 Halaman Generalized Sequential Pattern

3.7.3 Halaman Penggunaan

Pada halaman ini berisi cara penggunaan aplikasi ini. Petunjuk dalam menggunakan aplikasi ini terdapat pada halaman ini. Rancangan halaman penggunaan dapat dilihat pada **Gambar 3.10**.



Gambar 3.10 Halaman Penggunaan

3.7.4 Halaman Analisa Sekuensial

Pada halaman ini merupakan halaman utama untuk melakukan proses analisa, berupa halaman input parameter analisa dan kemudian sistem melakukan pemrosesan data. Rancangan antar muka dari halaman analisa sekuensial dapat dilihat pada **Gambar 3.11**.

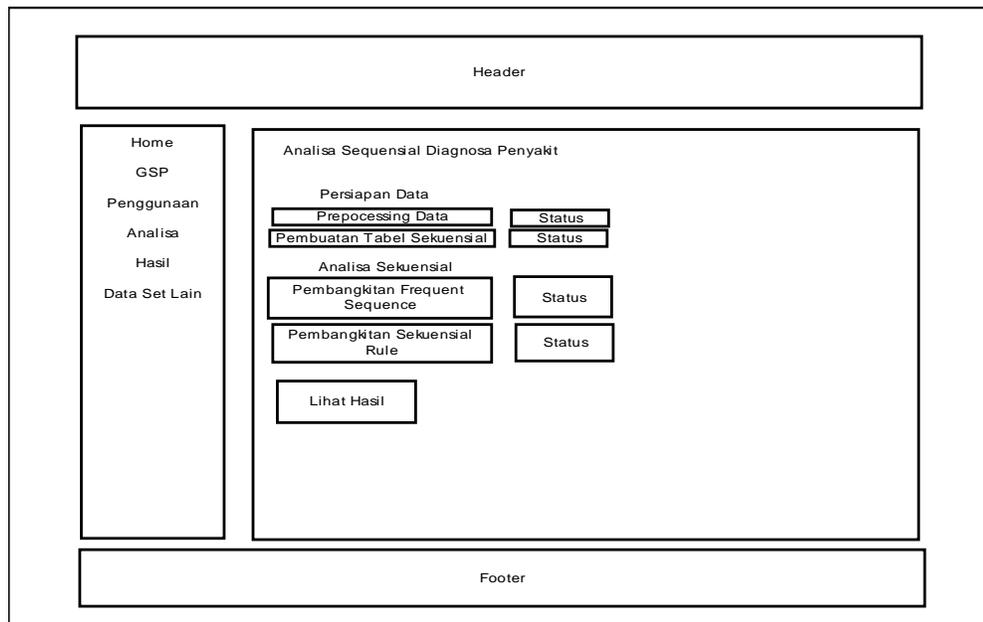
Gambar 3.11 Halaman Analisa Sekuensial

Adapun penjelasan dari gambar diatas sebagai berikut:

1. Rentan waktu : Dengan mengklik *checkbox* maka rentan waktu dapat diisi. Akan muncul sebuah kalender yang membantu pengisian ketika user mencoba untuk mengisi di *textfield*. Rentan waktu ini diisi untuk membatasi data yang akan diproses.
2. Batas sekuensi: Dengan mengklik *checkbox* maka Batas sekuensi dapat diisi. Diisi dengan minimal angka 1 dan maksimal sesuai dengan yang terdapat pada Data Uji. Batas sekuensi adalah pembatasan data yang akan diproses berdasarkan sekuensi atau banyaknya kunjungan berobat ke puskesmas.
3. Input minimal support, merupakan tempat untuk memasukkan nilai dari minimal support. Nilai tersebut berkisar dari 1-100, menggunakan titik untuk nilai yang desimal.
4. Input minimal confidence, merupakan tempat untuk memasukkan nilai minimal confidence. Nilai tersebut berkisar dari 1-100, menggunakan titik untuk nilai yang desimal.

3.7.4.1 Halaman Proses Analisa Sekuensial

Halaman ini merupakan halaman yang ditampilkan setelah memproses pada halaman analisa sekuensial. Halaman ini berisi rangkuman singkat proses analisa sekuensial. Rancangan antarmuka halaman proses analisa sekuensial terdapat pada **Gambar 3.12**.



Gambar 3.12 . Desain halaman proses analisa sekuensial

3.7.5 Halaman Hasil

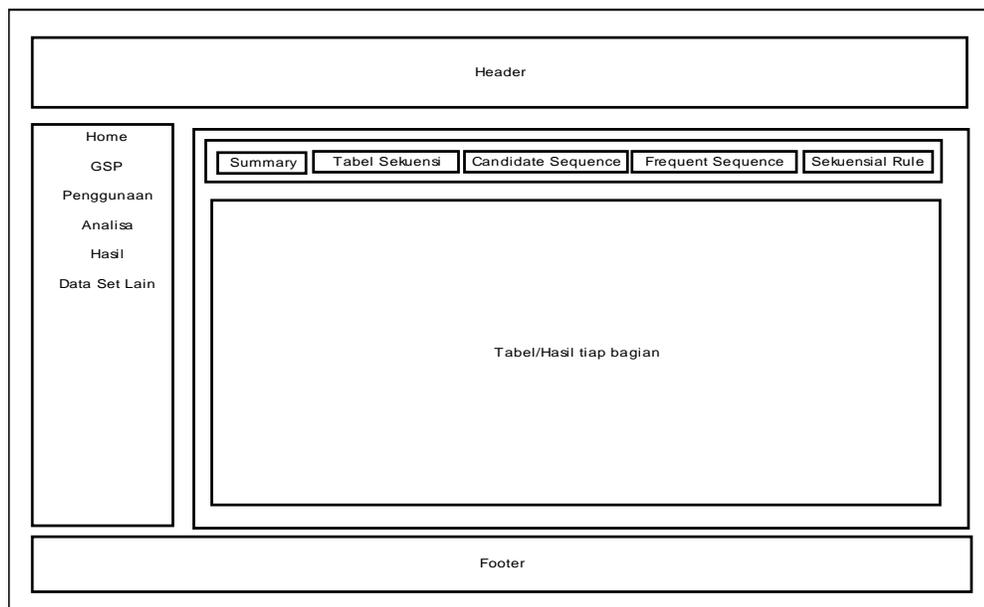
Pada halaman Hasil merupakan laporan dari aplikasi ini. Terdapat beberapa bagian yang membagi keseluruhan hasil dari analisa.

Halaman hasil terdiri dari :

1. Halaman Summary : Halaman yang menampilkan rangkuman hasil analisa berupa nilai statistik hasil analisa.
2. Halaman Tabel Sekuensial : Halaman yang menampilkan keseluruhan dari tabel sekuensi yang merupakan hasil dari proses persiapan data.
3. Halaman Candidate Sequence : Halaman ini menampilkan semua candidate - candidate sequence yang ditemukan selama proses analisa.

4. Halaman Frequent Sequence : Halaman ini menampilkan semua frequent sequence yang dihasilkan selama proses analisa.
5. Halaman Sequential Rule : Halaman ini menampilkan sekuensial rule yang dihasilkan oleh sistem.

Rancangan antarmuka halaman hasil terdapat pada **Gambar 3.13**.



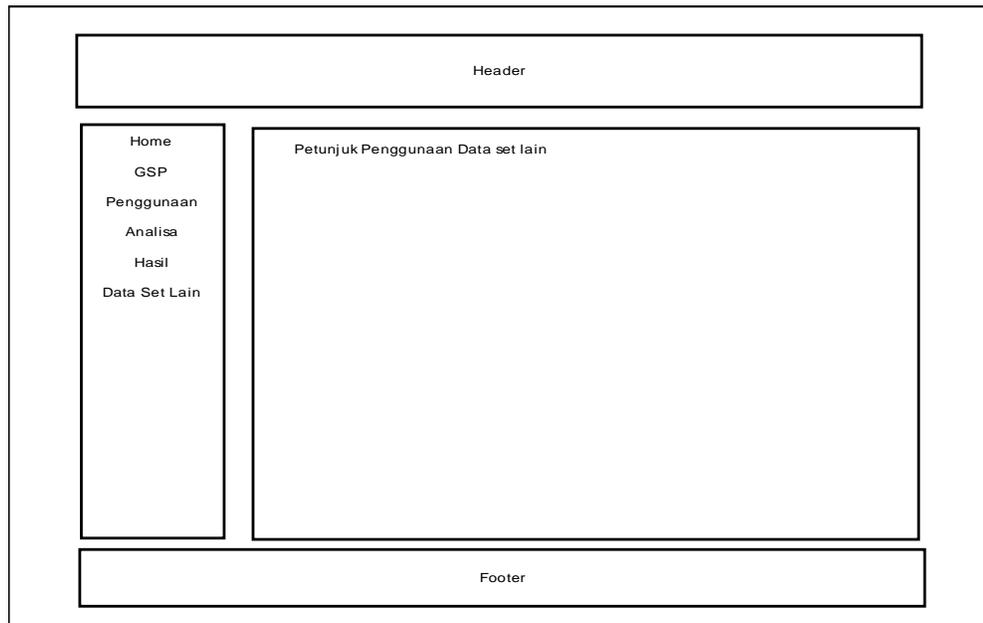
Gambar 3.13 . Desain halaman hasil

Adapun penjelasan dari gambar diatas sebagai berikut:

- Terdapat sub menu yang berupa tab yang dapat di klik untuk menampilkan hasil yang sesuai dengan sub menu.
- Tabel/hasil tiap bagian akan berisi sesuai dengan sub menu hasil yang dipilih.

3.7.6 Halaman Data Set Lain

Pada halaman ini berisi petunjuk bagaimana menggunakan data set lain untuk diuji pada aplikasi ini. Rancangan antarmuka halaman data set lain terdapat pada **Gambar 3.14**.



Gambar 3.14 . Desain halaman data set lain

3.8 Skenario Pengujian

Pada proses pengujian sistem ini digunakan metode korelasi *LIFT*. Dimana metode ini menguji korelasi dari rule yang telah terbentuk. Adapun korelasi lift dapat dihitung dengan rumus seperti dibawah ini

$$Lift(X, Y) = \frac{P(X \cup Y)}{P(X)P(Y)}$$

Dimana :

$$Lift(X, Y) = \text{korelasi Lift } X, Y$$

$P(X \cup Y)$ = Jumlah Kemunculan X dan Y dibagi dengan Total Transaksi.

$P(X)P(Y)$ = Jumlah Kemunculan X dikali dengan Jumlah Kemunculan Y pada total transaksi.

Apabila dari perhitungan tersebut menghasilkan nilai dibawah 1 maka terdapat korelasi *negatif* yang berarti rule tersebut salah, untuk perhitungan yang menghasilkan nilai diatas 1 maka terdapat korelasi *positif* yang berarti rule tersebut ada korelasinya. Namun apabila menghasilkan nilai sama dengan 1 maka tidak ada korelasi antara X dan Y.

Sebagai contoh, ditemukan sebuah rule yaitu $2100 \Rightarrow 2100 \rightarrow 1700$ dari total 31250 transaksi. Dimana nilai support dari 2100 adalah 1750 dan nilai support dari $2100 \rightarrow 1700$ adalah 425. Sehingga nilai $P(X) = 1750/31250 = 0,056$ dan nilai $P(Y) = 425/31250 = 0,0136$. Untuk nilai $P(X \cup Y) = 425/31250 = 0,0136$ dan nilai $P(X)P(Y) = 0,056 \times 0,0136 = 0,0007616$. Dan didapatkan nilai $Lift(X, Y) = 0,0136 / 0,0007616 = 17,857$. Karena nilai dari lift dari rule tersebut diatas 1 maka rule tersebut adalah korelasi positif yang berarti ada korelasi.