



tif

Program Studi Teknik Informatika  
Universitas Ahmad Dahlan

ISBN 978-979-3812-25-1

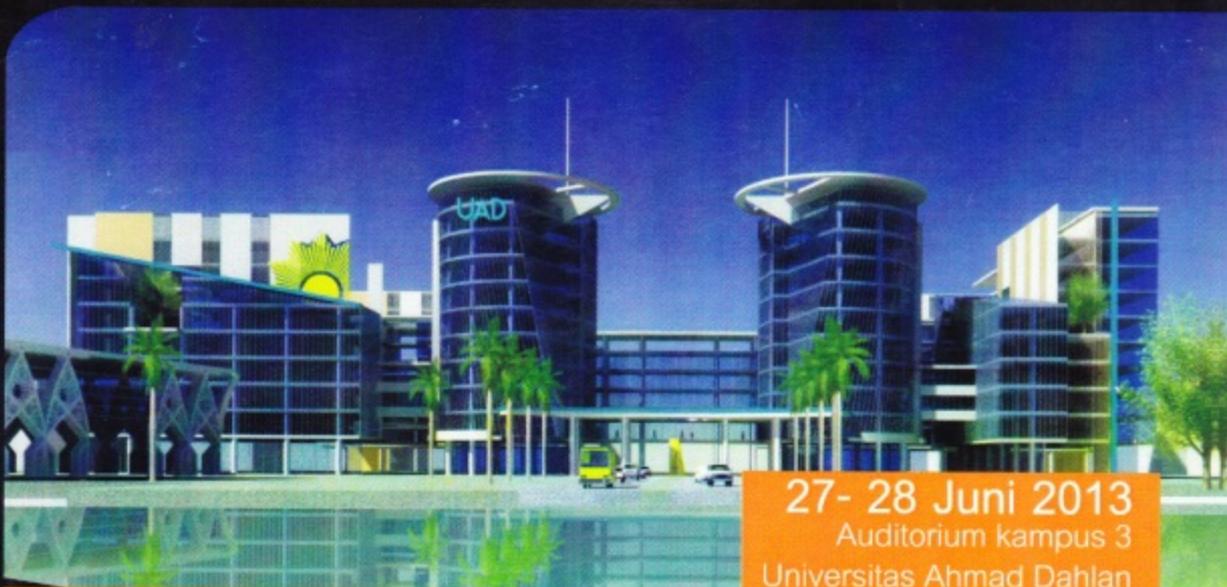


# PROSIDING

## STI 2013

Seminar Nasional Teknik Informatika

“Prospek dan Tantangan *Mobile Application*”



27- 28 Juni 2013  
Auditorium kampus 3  
Universitas Ahmad Dahlan

Universitas Ahmad Dahlan

# PROSIDING

STI 2013

SEMINAR TEKNIK INFORMATIKA

PROSPEK DAN TANTANGAN MOBILE APPLICATION

Yogyakarta, 27 Juni 2013

Penulis dan Pembicara

Herman Yuliansyah dkk.

Reviewer

Dr. Abdul Fadlil, M.T  
Anton Yudhana, Ph. D  
Eko Aribowo, S.T, M.Kom

Editor

Fifin Noviyanto, S.T, M.Cs  
Arfiani Nur Khusna, S.T  
Lisna Zahrotun, S.T

Penerbit

UAD PRESS

Jl. Kapas No.9 Semaki Yogyakarta 55166

Telp. 0274-563515, Fax. 0274-564604

## Kata Pengantar

Bismillahirrahmanirrahim Warahmatullahi Wabarakatuh,

Alhamdulillah kita panjatkan kehadiran Allah SWT, karena atas rahmat berkat dan izin-Nya maka Seminar Nasional Teknik Informatika (STI) 2013 yang diselenggarakan di Jurusan Studi Teknik Informatika Fakultas Teknologi Industri Universitas Ahmad Dahlan (UAD) pada hari Kamis, 27 Juni 2013 dapat diselenggarakan dengan baik.

STI merupakan agenda seminar tahunan yang diselenggarakan oleh TIF-UAD. Tujuan dari seminar ini antara lain menyediakan forum dan sarana komunikasi bagi para peneliti dan praktisi dari kalangan dosen, mahasiswa, maupun masyarakat umum dan memberikan kesempatan untuk mempublikasikan hasil penelitian.

Perkembangan pasar *mobile phone/smart phone* mengalami peningkatan yang sangat pesat. Perkembangan teknologi *hardware* dan *software* yang semakin pesat memungkinkan untuk berbagai aplikasi dapat dijalankan pada *mobile phone*. Selain itu, *mobile phone* dapat memberikan dukungan dalam menyelesaikan pekerjaan-pekerjaan, memberikan fasilitas permainan *games*, *chatting*, presentasi, berbelanja, bertransaksi melalui *social network*, serta masih banyak dukungan dan fasilitas yang ada. Perkembangan teknologi dan permintaan pasar yang semakin pesat ini menjadikan *mobile phone/smartphone* memberikan *platform* terbuka untuk mengembangkan aplikasi sesuai dengan yang diinginkan. Produsen *smartphone* menyediakan lingkungan pengembangan yang kondusif dan menawarkan prospek yang besar untuk berbisnis bagi para pengembang. Selain itu, para produsen menyediakan *application market* untuk para pengguna men-download aplikasi dan para programmer dapat memasang aplikasi mereka ke *application market* tersebut. Tantangannya adalah bagaimana aplikasi tersebut dapat diterima dan sukses di pasaran. Selain itu, keamanan dan keberagaman platform merupakan tantangan bagi pengembang aplikasi. Hal ini melatarbelakangi panitia pengantar tema STI 2013 yaitu "**Prospek dan Tantangan Mobile Application**".

STI 2013 berhasil mengumpulkan 42 paper yang merupakan kontribusi peserta seminar yang berasal baik dari berbagai perguruan tinggi maupun instansi dari beberapa provinsi di Indonesia.

Panitia STI 2013 mengucapkan terima kasih kepada pimpinan UAD yang telah memberikan support sehingga kegiatan ini terlaksana, para reviewer yang telah memberikan masukan untuk perbaikan, keynote speaker dan seluruh peserta yang telah berpartisipasi aktif. Panitia juga memohon maaf jika ada hal-hal yang kurang baik selama penyelenggaraan.

Bismillahirrahmanirrahim Warahmatullahi Wabarakatuh,

Jakarta, 27 Juni 2013

Peny. Prakara

Widhi Kusniawaty, S.T.,M.T.

## DAFTAR ISI

Kata Pengantar .....		i
Daftar Isi .....		ii
Daftar Panitia .....		iii
Daftar Isi .....		iv
SISTEM PAKAR MONITORING	Junaidi	1
INVENTORY CONTROL UNTUK	Khanna Tiara	
MENGHITUNG HARGA JUAL EFEKTIF	Nenden Dewi Y.	
DALAM MENINGKATKAN KEUNTUNGAN		
RANCANG BANGUN <i>WEB SERVICE</i>	Ardiansyah	10
APLIKASI MANAJEMEN <i>EVENT</i>	Dewangga Ari Putra	
MODEL DATA WAREHOUSE UNTUK	Junaidi	21
MENDUKUNG SISTEM INFORMASI	Herlina	
EKSEKUTIF PENDIDIKAN DALAM	Nina Karolina	
MENDUNGGUNAKAN KEPUTUSAN		
PENGUNAAN METODE QUANTITATIVE	Arif Arizal	32
ASSOCIATION RULES UNTUK Mencari		
MUTRAN ASOSIAIF DARI DATABASE		
PENGOBATAN		
PENGUNGANAN WEB MONITORING	Tedy Setiadi	44
COMPUTER-BASED TEST DENGAN	Edi Zuviyanto	
MENGUNAKAN BASIS DATA		
TERDISTRIBUSI PADA UJIAN		
KOMPETENSI DOKTER GIGI INDONESIA		
PERANCANGAN SISTEM INFORMASI	Teguh Cahyomo	53
PENJADWALAN SEMINAR UJIAN AKHIR		
DAN KERJA PRAKTEK (STUDI KASUS		
PRODI TEKNIK INFORMATIKA JURUSAN		
TEKNIK UNIVERSITAS JENDERAL		
SOEDIRMAN )		

IMPLEMENTASI SINGLE SIGN ON PADA CLOUD COMPUTING : STUDI KASUS FACEBOOK	Marvin Chandra W	64
PERBANDINGAN COSINE COEFFICIENT DENGAN EXTENDED JACCARD COEFFICIENT PADA CASE-BASED REASONING	Murien Nugraheni	72
IMPLEMENTASI WEB SERVICES PADA SISTEM INTEGRASI PELAPORAN DATA PENERIMAAN PESERTA DIDIK BARU (STUDI KASUS DINAS PENDIDIKAN KABUPATEN BANYUMAS)	Yuli Purwati Ema Utami Armadyah A	79
SISTEM INFORMASI PENDUKUNG KEPUTUSAN PENENTUAN DESA SIAGA DI PUSKESMAS KALIBAGOR KABUPATEN BANYUMAS	Linda P Abidarin Rosidi Syamsul A Syahdan	89
SISTEM PENDUKUNG KEPUTUSAN UNTUK MENENTUKAN SISWA PENERIMA BEASISWA PRESTASI DI SMA NEGERI 5 MAGELANG	Astri Wuragil Anastasia Meyliana	99
ANALISIS <i>CUSTOMER RELATIONSHIP MANAGEMENT</i> DI PT. JC INDONESIA BOGORYAKARTA (Studi Kasus Penanganan Pelanggan)	Irfan Santiko Abidarin Rosidi Sudarmawan	111
KERANGKA KERJA ZACHMAN UNTUK PERENCANAAN STRATEGI SISTEM INFORMASI PENANGGULANGAN BENCANA	Arfiani Nur Khusna	123
PERANCANGAN DATA MINING PENGELOMPOKAN JUMLAH PENUMPANG BUS TRANS JOGJA MENGGUNAKAN METODE <i>CLUSTERING K-MEANS</i>	Lisna Zahrotun	134
LAYANAN PEMBUATAN LAPORAN BERBASIS <i>FTP</i>	Wisnu Manupraba Novi Setiani	143
PERANCANGAN APLIKASI MANAJEMEN DOKUMENTASI DENGAN MEMANFAATKAN KRIPTOGRAFI	I Made Mustika K A Agung Nugraha	152

<b>PENYANCANGAN SISTEM PAKAR UNTUK          MENYUDIKAN KEPOLISIAN          (Studi Kasus pada tindak pelanggaran lalulintas          dan angkutan jalan)</b>	Ika Arfiani	164
<b>PENYEMING DOKUMEN TEKS BAHASA          INDONESIA MENGGUNAKAN ALGORITMA          FUZZY</b>	Lasmedi Afuan	181
<b>PENERAPAN METODE ASSOCIATION          RULES DAN COSINE SIMILARITY UNTUK          PENGEMBANGAN FITUR DETEKSI DINI          PENYAKIT</b>	Aldion Renata Ristu Saptono Meiyanto Eko Sulistyo	189
<b>MODEL RENCANA STRATEGIS          PENGEMBANGAN E-GOVERMENT (Studi          Kasus : Dinas Kelautan Dan Perikanan DIY)</b>	Sri Handayaningsih Rahmat Rian H	206
<b>MODEL SISTEM INFORMASI          E-GOVERNMENT MENGGUNAKAN          KERANGKA KERJA TOGAF ADM          BERBASIS SOA (<i>Service Oriented Architecture</i>)          DI DINAS KELAUTAN DAN PERIKANAN          PROVINSI DIY</b>	Sri Handayaningsih Ina Mufiatun	220
<b>PENGEMBANGAN MODUL INVERTER          TIGA FASE DENGAN PERBANDINGAN          FUZZY KONSTAN</b>	Muhammad Arrofiq R. Arif Tri R	232
<b>IMPLEMENTASI TEKNIK PENGOLAHAN          CITRA UNTUK MENENTUKAN TINGKAT          KEMERUHAN AIR (Studi Kasus di PDAM          Kota Yogyakarta)</b>	Murinto Herlina Nurhidayah	243
<b>SISTEM PENDUKUNG KEPUTUSAN          PERENCANAAN PERUBAHAN ANGGARAN          BIAYA UNTUK MENGANTISIPASI          KETERLAMBATAN (Studi Kasus Pada PT.          Wicari Arsita Graha)</b>	Annie Purwani Putut Janu W	254
<b>PENENTUAN TINGKAT STRES PADA          MAHASISWA MENGGUNAKAN METODE          TSUKAMOTO</b>	Ardi Pujiyanta Septi Nur Chasanah	268

<p>UJIAN ALGORITMA RIVEST CODE 5 UNTUK ENKRIPSI STRUKTUR FILE DOKUMEN</p>	<p>Hamdani Sayekti Harits S Anindita Septiarini</p>	<p>280</p>
<p>RANCANGAN MODIFIKASI KRIPTOGRAFI MODERN CBC UNTUK PENGAMANAN DATA/FILE TEXT</p>	<p>Nur Rochmah D.P.A</p>	<p>288</p>
<p>APLIKASI TERJEMAHAN BAHASA INDONESIA KE BAHASA TERNATE BERSERTAI DENGAN PEMERIKSAAN KALIMAT AMBIGU</p>	<p>Dewi Soyusiawaty Rafdi</p>	<p>297</p>
<p>RANCANGAN CASE BASED REASONING UNTUK DIAGNOSA GANGGUAN KEPRIBADIAN</p>	<p>Anna Hendri S J</p>	<p>307</p>
<p>VISUALISASI PROSES ENKRIPSI DAN DEKRIPSI METODE KRIPTOGRAFI AES/GVX PRODUCT DAN MODERN CBC</p>	<p>Eko Aribowo Heru Kurniawan</p>	<p>313</p>
<p>ANALISIS KINERJA WEBSITE UNIVERSITAS AHMAD DAHLAN DENGAN TEKNIK SEO (<i>SEARCH ENGINE OPTIMIZATION</i>) <i>OFF PAGE</i></p>	<p>Mellia Marselina Ali Tarmuji</p>	<p>322</p>
<p>RANCANG BANGUN SISTEM SIMULASI URBAN NASIONAL BERBASIS WEB STUDI KASUS SMA NEGERI 1 TANDUN LUNGBATU RIAU</p>	<p>Hasanuddin Fitriani Muttakin Tri Ida Sri Ningsih</p>	<p>332</p>
<p>RANCANGAN APLIKASI INFORMASI JADWAL PERKULIAHAN BERBASIS MOBILE (STUDI KASUS FAKULTAS TEKNOLOGI INDUSTRI UAD)</p>	<p>Herman Yuliansyah Merlinda Wibowo Wisnu Arisandy Arif Budiarti</p>	<p>343</p>
<p>RANCANGAN PENGAMANAN SISTEM INFORMASI PADA PERANGKAT ANDROID</p>	<p>Prasetyo Adi W P</p>	<p>355</p>
<p>SISTEM <i>BROADCAST</i> PROSES BELAJAR MENGAJAR dengan <i>SYNCHRONOUS</i> dan <i>ASYNCHRONOUS</i></p>	<p>Arif Budiyanto Wahyu Pujiyono</p>	<p>361</p>
<p><i>DECISION SUPPORT SYSTEM</i> UNTUK PEMILIHAN SUPPLIER MENGGUNAKAN METODE PROMETHEE</p>	<p>Elfitri Dwi R</p>	<p>375</p>

<p> <b>SISTEM PENDUKUNG KEPUTUSAN DALAM          PEMILIHAN LOKASI BENGKEL DENGAN          METODE PROMETHEE</b> </p>	<p>           Agus Setiawan            Sri Winiarti         </p>	<p>383</p>
<p> <b>SISTEM PENGUKUR VOLUME BARANG          MENGGUNAKAN WEBCAM</b> </p>	<p>           Kartika Firdausy            Rizki Hidayat         </p>	<p>388</p>
<p> <b>ANALISIS KINERJA WEBSITE          UNIVERSITAS AHMAD DAHLAN DENGAN          TEKNIK SEO (<i>SEARCH ENGINE          OPTIMIZATION</i>) ON PAGE</b> </p>	<p>           Anggi Tyas P            Ali Tarmuji         </p>	<p>399</p>
<p> <b>RANCANGAN APLIKASI ANTI PENCURIAN          SMARTPHONE BERBASIS ANDROID</b> </p>	<p>Taufiq Ismail</p>	<p>409</p>
<p> <b>PENGENALAN AKSARA DAERAH          MENGGUNAKAN DETEKSI TEPI          OPERATOR PREWITT</b> </p>	<p>Fradika Indrawan</p>	<p>417</p>
<p> <b>INTEGRASI SMS GATEWAY DAN          PEMANFAATAN RECEIVER GPS UNTUK          PENELUSURAN LOKASI</b> </p>	<p>Muhammad Aziz</p>	<p>422</p>

# PENERAPAN METODE ASSOCIATION RULES DAN COSINE SIMILARITY UNTUK PENGEMBANGAN FITUR DETEKSI DINI PENYAKIT

**Aldion Renata, Ristu Saptono, Meiyanto Eko Sulisty**

Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Sebelas Maret

Jl. Ir. Sutami 36A Kentingan Surakarta 57126 Jawa Tengah.

di\_aldion@yahoo.co.id, r\_saptono@uns.ac.id, mekosulistyo@uns.ac.id

## *Abstrak*

*Pendeteksian penyakit dapat dilakukan dengan membandingkan gejala yang dialami dengan data gejala penyakit yang dimiliki. Untuk meminimalisir jumlah gejala yang ditanyakan guna memperoleh feedback, diperlukan pengelompokan gejala dan pengeliminasian penyakit dalam beberapa tahap.*

*Similarity merupakan metode yang digunakan untuk menghitung kemiripan dua objek. Association rule adalah bentuk paling umum dalam menemukan pola dari suatu kumpulan data. Pengelompokan gejala akan dilakukan menggunakan metode association rules. Gejala-gejala yang telah dikelompokkan dijadikan sebagai acuan dalam mengajukan pertanyaan untuk memperoleh feedback. Kemudian feedback akan dihitung nilai similarity-nya dengan gejala penyakit menggunakan rumus Cosine. Penyakit dengan nilai similarity yang tidak memenuhi batas minimum akan dieliminasi. Nilai similarity terbesar menjadi kesimpulan hasil deteksi.*

*Ada dua macam rules yang digunakan dalam penelitian ini yaitu rules implikasi dan rules gabungan antara rules implikasi dan rules kontraposisi. Dalam logika matematika, implikasi memiliki ekuivalensi dengan bentuk kontraposisinya. Kedua rules tersebut memiliki hasil pengujian yang berbeda. Masing-masing rules mampu mengurangi jumlah pertanyaan yang harus dijawab rata-rata mencapai 70.55% dan 69.490%. tetapi akurasi yang dihasilkan dari kedua bentuk rules tersebut menunjukkan bahwa rules implikasi menghasilkan nilai akurasi dan nilai similarity yang lebih baik dibandingkan dengan rules gabungan.*

**Kata kunci:** *association rule, cosine similarity, pendeteksian penyakit, feedback*

## 1. PENDAHULUAN

Pada umumnya, untuk mendeteksi penyakit yang diderita oleh seseorang adalah dilakukan dengan cara meminta *feedback* gejala yang dirasakan kemudian mencocokkannya dengan gejala-gejala dari setiap penyakit yang terdapat di dalam daftar.

Fungsi *similarity* adalah fungsi yang menerima dua buah objek dan mengembalikan nilai kemiripan (*similarity*) antara kedua objek tersebut berupa bilangan riil [1]. Eska Sebayu [2] memanfaatkan metode *Cosine Similarity* untuk pendeteksian dini penyakit pada toko *online* obat herbal. Deteksi penyakit dilakukan dengan cara meminta *feedback* gejala yang dirasakan oleh seseorang dari 128 daftar gejala yang ada kemudian mencocokkannya dengan daftar 33 penyakit. Untuk meminimalisir jumlah pertanyaan pada proses input gejala, diperlukan pengelompokan gejala dan pengeliminasian penyakit dalam beberapa tahap, sehingga gejala-gejala dari penyakit yang tidak mungkin diderita tidak akan ditanyakan lagi oleh aplikasi. Pengeliminasian penyakit dilakukan berdasarkan perhitungan nilai *similarity* positif dan negatif pada tahap awal, di mana *similarity* negatif diperoleh dari perhitungan dengan negasi dari *feedback*. Apabila nilai *similarity* negatif tinggi, berarti penyakit tersebut tidak mungkin diderita oleh pengguna aplikasi. Dengan cara demikian rata-rata pengurangan jumlah pertanyaan yang diajukan mencapai 64,24%.

Dalam penelitian ini ditambahkan metode *association rule* untuk proses pengelompokan gejala yang akan dijadikan acuan dalam mengajukan pertanyaan. *Association rule* berusaha menemukan aturan-aturan tertentu yang mengasosiasikan data yang satu dengan data yang lain. *Rules* yang dihasilkan merupakan bentuk pernyataan implikasi yang dalam logika matematika memiliki nilai kebenaran yang sama dengan bentuk kontraposisinya.

Tujuan dari penelitian ini adalah mencari hasil yang paling optimal dari kedua bentuk *rules* tersebut dengan tolok ukur efektifitas jumlah pertanyaan gejala yang diajukan kepada pengguna dan akurasi hasil pendeteksian penyakit.

Secara umum, fungsi *similarity* adalah fungsi yang menerima dua buah objek dan mengembalikan nilai kemiripan (*similarity*) antara kedua objek tersebut berupa bilangan riil [1]. Umumnya, nilai yang dihasilkan oleh fungsi *similarity* berkisar pada interval [0...1]. Namun ada juga beberapa fungsi *similarity* yang menghasilkan nilai yang berada di luar interval tersebut. Untuk memetakan hasil fungsi tersebut pada interval [0...1] dapat dilakukan normalisasi.

Semakin besar hasil fungsi *similarity*, maka kedua objek yang dievaluasi dianggap semakin mirip. Sebaliknya, semakin kecil hasil fungsi *similarity*, maka kedua objek tersebut dianggap semakin berbeda. Pada fungsi yang menghasilkan nilai pada jangkauan [0...1], nilai 1 melambangkan kedua objek persis sama, sedangkan nilai 0 melambangkan kedua objek sama sekali berbeda.

*Cosine Similarity* adalah metode *similarity* yang paling banyak digunakan untuk menghitung *similarity* dua buah dokumen [3]. Rumus *Cosine Similarity* adalah sebagai berikut:

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

di mana

$x \cdot y$  = vektor *dot product* dari  $x$  dan  $y$ , dihitung dengan  $\sum_{k=1}^n x_k y_k$

$\|x\|$  = panjang vektor  $x$ , dihitung dengan  $\sqrt{\sum_{k=1}^n x_k^2}$

$\|y\|$  = panjang vektor  $y$ , dihitung dengan  $\sqrt{\sum_{k=1}^n y_k^2}$

Pola penyakit di Indonesia (bahkan di dunia) telah mengalami pergeseran dari penyakit infeksi (yang terjadi sekitar tahun 1970 ke bawah) ke penyakit-penyakit metabolik degeneratif (sesudah tahun 1970 hingga sekarang) [4]. Hal ini seiring dengan laju perkembangan tingkat ekonomi dan peradaban manusia yang ditandai dengan pesatnya perkembangan ilmu dan teknologi dengan berbagai penemuan baru yang bermanfaat dalam pengobatan dan peningkatan kesejahteraan umat manusia.

Akan tetapi timbul penyakit baru yang bukan disebabkan oleh jasad renik, melainkan oleh gangguan metabolisme tubuh akibat konsumsi berbagai jenis makanan yang tidak terkendali serta gangguan faal tubuh sejalan dengan proses degenerasi. Penyakit ini dikenal dengan sebutan penyakit metabolik dan degeneratif. Yang termasuk penyakit metabolik antara lain: diabetes (kencing manis), *hiperlipidemia* (kolesterol tinggi), asam urat, batu ginjal dan hepatitis; sedangkan penyakit degeneratif diantaranya: rematik (radang persendian), asma (sesak nafas), *ulser* (tukak lambung), *haemorrhoid* (*ambeien*/wasir) dan pikun (*lost of memory*).

*Association rule* (aturan asosiatif) adalah salah satu teknik utama dalam *data mining* dan merupakan bentuk yang paling umum dipakai dalam menemukan *pattern* atau pola dari suatu kumpulan data [5]. *Association rule* (aturan asosiatif) berusaha menemukan aturan-aturan tertentu yang mengasosiasikan data yang satu dengan data yang lain.

Untuk mencari *association rule* dari suatu kumpulan data, pertama kita harus mencari lebih dulu yang disebut *frequent itemset* atau sekumpulan item yang sering muncul bersamaan. Setelah semua pola *frequent itemset* ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat yang telah ditentukan.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam database dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan asosiatif.

Aturan asosiatif yang memenuhi syarat *minimum* untuk *support* dan *confidence*. Nilai *support* dari 2 *item* diperoleh dari rumus berikut :

$$\text{support}(A \cap B) = \frac{\text{jumlah Transaksi mengandung A dan B}}{\text{total Transaksi}}$$

Setelah semua *support* ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat *minimum* untuk *confidence* dengan menghitung *confidence*

aturan assosiatif  $A \rightarrow B$  di mana A adalah *antecedent* dan B adalah *consequent*. Nilai *confidence* dari aturan  $A \rightarrow B$  diperoleh dari rumus berikut:

$$\text{confidence}(A \rightarrow B) = \frac{\text{jumlah Transaksi mengandung A dan B}}{\text{total Transaksi mengandung A}}$$

## 2. METODE PENELITIAN

### 2.1 Pengumpulan Data

Pengumpulan data yaitu melakukan observasi, mencari dan mengumpulkan data-data yang ada relevansinya dengan judul Tugas Akhir ini dari buku yang ditulis oleh Prayogo Utomo [6] yang melibatkan 128 gejala dan 33 penyakit.

### 2.2 Pemodelan Data

#### 2.2.1 Proses Pembentukan *Rules*

*Rules* yang potensial dalam penelitian ditemukan dengan cara mencari hubungan dari setiap gejala yang diderita suatu penyakit, sehingga *rules* yang diperoleh akan memiliki makna jika gejala A diderita oleh suatu penyakit maka gejala B juga diderita oleh penyakit tersebut. *Rules* yang akan digunakan dalam pengolahan data ditentukan dengan menghitung nilai *support* dan *confidence* masing-masing hubungan gejala tadi. Karena *support* yang diperoleh rendah, sehingga *rules* yang dipilih adalah yang memiliki nilai *confidence* di atas 0,75. *Rules* yang diperoleh merupakan bentuk implikasi yang ekuivalen dengan bentuk kontraposisinya. Untuk mengkonfirmasi bahwa *rules* tersebut dapat dipakai dalam proses selanjutnya, maka digunakan *rules* kontraposisi yang diperoleh dari hasil penegasian gejala-gejala setiap penyakit. Jika *rules* kontraposisi tersebut memiliki nilai *support* dan *confidence* yang tinggi maka *rules* yang telah ditemukan sebelumnya tadi dapat digunakan.

#### 2.2.2 Proses Pengelompokan Ulang Gejala

Sebelumnya gejala dibagi ke dalam dua kelompok yaitu kelompok gejala umum yaitu gejala-gejala yang diderita oleh lebih dari satu penyakit dan gejala khusus yaitu gejala yang hanya diderita oleh penyakit-penyakit tertentu. *Rules* yang telah diperoleh digunakan untuk mengelompokkan ulang gejala. Tiap kelompok gejala akan dibagi lagi ke dalam kelompok yang lebih kecil, yaitu kelompok gejala yang merupakan *antecedent* dari suatu *rules*, gejala yang merupakan *consequent*, dan gejala yang tidak memiliki *rule*. Dua bentuk *rules* akan menghasilkan pengelompokan ulang gejala yang berbeda. Hasil dari pengelompokan ulang gejala inilah yang akan digunakan sebagai acuan dalam mengajukan pertanyaan.

#### 2.2.3 Proses Eliminasi Penyakit dan Penemuan Penyakit yang mungkin dialami

*Feedback* yang diperoleh dari pengguna akan dicocokkan dengan gejala-gejala dari penyakit yang terdapat di dalam daftar. Nilai akhir *similarity* diperoleh dari perbandingan seluruh *feedback* yang telah diperoleh dalam beberapa tahap pengajuan pertanyaan dengan daftar gejala penyakit. Pada tahap akhir proses pendeteksian penyakit akan diperoleh beberapa penyakit dengan nilai akhir

*similarity* yang lebih besar dari 0,5. Penyakit-penyakit itulah yang akan menjadi keluaran proses yang menunjukkan kemungkinan penyakit yang diderita oleh pengguna berdasarkan gejala yang dirasakannya.

### 2.3 Pengembangan Aplikasi

Aplikasi deteksi dini penyakit ini akan diterapkan menggunakan bahasa pemrograman PHP serta *database* MySQL sebagai tempat penyimpanan data-data yang diperlukan.

### 2.4 Pengujian dan Analisa Hasil

Pada penelitian ini, dilakukan pengujian untuk mengukur akurasi aplikasi deteksi dini penyakit. Pengujian akan dilakukan dengan empat cara yang berbeda, antara lain:

1) Pengujian dengan *feedback* lengkap

Setiap penyakit (dari 33 penyakit) akan diujicobakan pada aplikasi dengan memberikan *feedback* gejala penyakit tersebut secara lengkap sesuai dengan gejala penyakit.

2) Pengujian dengan *feedback random* 20%

Pengujian ini dilakukan dengan memberikan jawaban ‘tidak’ pada 20% dari gejala penyakit yang seharusnya bernilai ‘ya’. Pemilihan gejala yang dijawab ‘tidak’ ini dilakukan secara acak. Pengujian ini akan dilakukan sebanyak tiga kali untuk memperoleh nilai akurasi rata-rata.

3) Pengujian dengan *feedback random* 40%

Pengujian ini sama dengan pengujian *feedback random* 20%, tapi dengan 40% gejala penyakit yang dijawab ‘tidak’. Pengujian ini dilakukan sebanyak tiga kali untuk memperoleh nilai akurasi rata-rata.

4) Pengujian dengan data real

Pada pengujian ini, dilakukan wawancara terprogram terhadap beberapa responden untuk memperoleh kasus penyakit yang diderita dalam kurun waktu tidak lebih dari tiga bulan. Responden diberi pertanyaan mengenai penyakit yang pernah atau yang sedang diderita disertai dengan gejala yang dirasakan. Pertanyaan gejala yang dirasakan disesuaikan dengan data penyakit dan gejala yang sudah ada di dalam aplikasi.

Setiap pengujian tersebut akan dihitung nilai akurasinya. Nilai akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{\text{jumlah per}}{\text{jumlah percoba}}$$

Selain menghitung akurasi, di setiap percobaan pada masing-masing pengujian akan dihitung jumlah pertanyaan yang diajukan, hal ini berguna untuk mengetahui bahwa jumlah pertanyaan yang harus dijawab di setiap percobaan tidak perlu jumlah yang sama dengan jumlah gejala yang ada, yang artinya aplikasi sudah dapat meminimalisir jumlah pertanyaan yang diperlukan. Untuk menghitungnya menggunakan rumus sebagai berikut :

$$\text{rata - rata} = \frac{\text{jumlah pertanyaan yang diajukan setiap percobaan}}{\text{jumlah percobaan}} \times 100\%$$

Setelah jumlah rata-rata di setiap percobaan pada masing-masing pengujian dihitung, kemudian dilakukan perhitungan efektifitas pengurangan jumlah pertanyaan. Perhitungan ini dilakukan untuk mengetahui sejauh mana kemampuan aplikasi mengurangi jumlah pertanyaan yang harus dijawab oleh pengguna.

$$\text{Efektifitas} = \left(1 - \frac{\text{rata - rata}}{\text{jumlah gejala}}\right) \times 100\%$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Gambaran Umum Aplikasi

Aplikasi deteksi dini penyakit merupakan aplikasi digunakan untuk membantu mendeteksi secara dini penyakit yang mungkin diderita oleh seseorang. Aplikasi ini meminta input gejala yang dialami dalam bentuk input radio. Sejumlah gejala akan diajukan sebagai pertanyaan, dan pengguna cukup memberikan jawaban 'ya' pada gejala yang sedang atau pernah dirasakannya. *Feedback* yang diberikan dan data gejala setiap penyakit diubah dalam bentuk biner. *Feedback* dari pengguna akan dicari pasangan *consequent*-nya dalam daftar *rules* yang ada dalam *database*. Perhitungan nilai kemiripannya menggunakan metode *Cosine Similarity*. Semakin tinggi nilai *similarity* suatu penyakit terhadap *feedback* yang diperoleh dari pengguna aplikasi, maka semakin besar pula peluang penyakit tersebut muncul sebagai penyakit yang mungkin diderita.

#### 3.2 Pemodelan

##### 3.2.1 Pembentukan Rules

Berikut ini adalah tahapan-tahapan yang dilakukan dalam pembentukan *rules*

1. Merubah daftar penyakit beserta gejalanya ke dalam bentuk matriks  
Membuat matriks yang menunjukkan hubungan antara penyakit dengan gejalanya. Tahap berikutnya adalah transformasi biner setiap gejala yang dimiliki oleh setiap penyakit.  
Jika gejala tersebut terdapat pada suatu penyakit, maka diberi kode biner '1' jika tidak maka '0' seperti yang ditunjukkan pada gambar 1 berikut.

	G1	G2	G3	...	G128
P1	1	1	1	...	0
P2	0	0	0	...	0
P3	0	0	0	...	0
:	:	:	:	:	:
P33	0	0	0	...	1

**Gambar 1. Matriks hubungan penyakit dengan gejala**

2. Mencari nilai *support* dan *confidence* dari setiap pasangan gejala.  
Mencari nilai *support* dan *confidence* semua kombinasi berpasangan dari *item* gejala-gejala di semua daftar penyakit. Sebagai contoh dicari nilai *support* dari

$G8 \rightarrow G6$ , setelah nilai *support* dari  $G8 \rightarrow G6$  ditemukan, selanjutnya mencari nilai *confidence* dari  $G8 \rightarrow G6$ . Begitu juga untuk pasangan gejala lainnya. Hal ini adalah tahapan awal untuk menentukan *rules* yang akan digunakan dalam tahapan berikutnya.

3. Menentukan *association rules* (aturan asosiasi)

Menentukan *association rules* (aturan asosiasi) yang akan digunakan, diambil dari pasangan *antecedent* dan *consequent* yang telah ditemukan pada langkah pertama sebelumnya. Karena *support* yang diperoleh memiliki nilai yang rendah, sehingga *rules* yang dipilih adalah yang memiliki nilai *confidence* di atas 75%.

4. Membentuk kontraposisi dari *rules* yang telah diperoleh

*Rule* tersebut dalam logika matematika merupakan bentuk implikasi di mana nilainya memiliki kesamaan (ekuivalen) dengan kontraposisinya. Untuk mengkonfirmasi bahwa *rules* tersebut dapat dipakai dalam proses selanjutnya, maka digunakan *rules* kontraposisi yang diperoleh dari hasil penegasian gejala-gejala setiap penyakit. Jika *rules* kontraposisi tersebut memiliki nilai *support* dan *confidence* yang tinggi maka *rules* yang telah ditemukan sebelumnya tadi dapat digunakan.

Berikut ini adalah contoh *rules* yang diperoleh pada proses awal ini.

**Tabel 2. Contoh *rules* yang diperoleh pada proses pertama**

No	<i>Rules</i> implikasi		<i>Rules</i> kontraposisi		
	<i>Antecedent</i> $\rightarrow$ <i>consequent</i>	<i>Conf</i>	<i>Antecedent</i> $\rightarrow$ <i>consequent</i>	<i>Supp</i>	<i>Conf</i>
1	$G8 \rightarrow G6$	0.757 6	$-G6 \rightarrow -G8$	0.757 6	0.961 5
2	$G12 \rightarrow G11$	0.818 2	$-G11 \rightarrow -G12$	0.818 2	1.00
3	$G14 \rightarrow G9$	0.697 0	$-G9 \rightarrow -G14$	0.697 0	0.958 3
4	$G21 \rightarrow G20$	0.909 1	$-G20 \rightarrow -G21$	0.909 1	1.00
5	$G22 \rightarrow G9$	0.727 3	$-G9 \rightarrow -G22$	0.727 3	1.00

Dalam pencarian *rules* kontraposisi ini terdapat beberapa *rules* yang tidak memiliki bentuk implikasinya, sehingga *rules* ini digunakan sebagai tambahan dari *rules* implikasi yang telah diperoleh. *Rules* kontraposisi dan implikasi ini akan menjadi *rules* gabungan yang akan menjadi pembanding untuk mencari hasil terbaik dalam pengujian yang akan dilakukan pada tahap berikutnya.

**Tabel 3. *Rules* hasil kontraposisi yang tidak memiliki bentuk implikasi**

No	<i>Rules</i> Kontraposisi		
	<i>Antecedent</i> $\rightarrow$ <i>consequent</i>	<i>Support</i>	<i>Confidence</i>

1	$\sim G42 \rightarrow \sim G100$	0.91	0.97
2	$\sim G59 \rightarrow \sim G55$	0.88	0.94
3	$\sim G38 \rightarrow \sim G86$	0.85	0.93
4	$\sim G3 \rightarrow \sim G44$	0.82	0.96

### 3.2.2 Pengelompokan Ulang Gejala

Dalam penelitian ini, 128 gejala dibagi menjadi 2 kelompok yaitu kelompok gejala yang muncul di beberapa penyakit sebanyak 32 gejala dan kelompok gejala yang hanya muncul hanya di satu penyakit (gejala khusus) sebanyak 96 gejala. Pada umumnya, gejala-gejala yang telah terbagi menjadi dua kelompok sebelumnya dikelompokkan ulang menjadi kelompok gejala yang merupakan *antecedent rules*, kelompok gejala *consequent*, kelompok gejala yang tidak memiliki *rules*.

Berikut contoh pengelompokan ulang gejala. *Rules* yang digunakan adalah contoh *rules* pada tabel 1 di atas. (diambil 10 contoh dari 128 gejala)

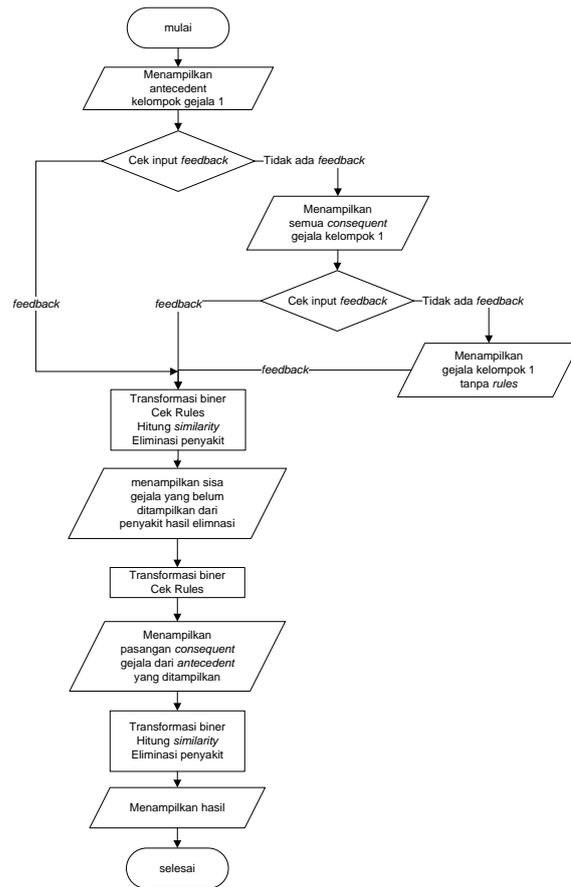
**Tabel 4. Contoh pengelompokan ulang gejala hasil penerapan *rules*.**

Kelompok gejala awal	G3, G6, G8, G9, G11, G12, G14, G38, G42, G44
Kelompok gejala setelah penerapan <i>rules</i>	kelompok gejala <i>antecedent</i> : G6, G9, G11 kelompok gejala <i>consequent</i> : G8, G12, G14 kelompok gejala tanpa <i>rule</i> : G3, G38, G42, G44

### 3.2.3 Eliminasi Penyakit dan Penemuan Penyakit yang mungkin dialami

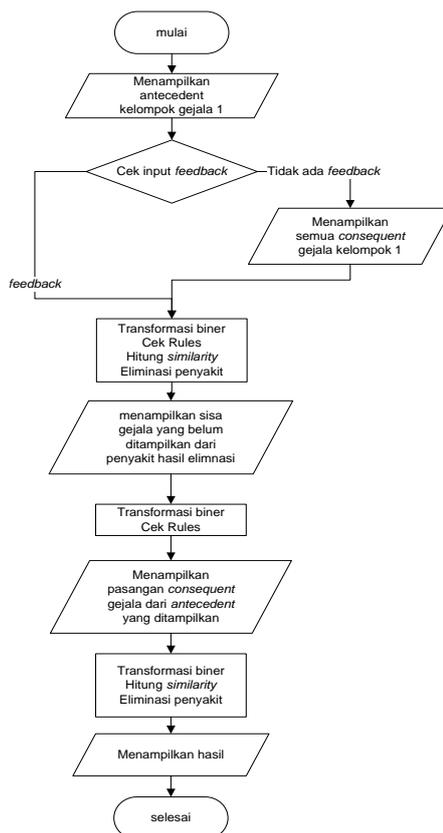
Pengeliminasian penyakit bertujuan untuk menghilangkan penyakit yang sudah tidak mungkin lagi muncul. Dampak positifnya, jumlah gejala yang akan ditanyakan menjadi semakin sedikit pula. Pertanyaan yang diajukan terlebih dahulu gejala-gejala yang termasuk dalam kelompok pertama karena merupakan jenis gejala yang umum dirasakan. Dua bentuk *rules* yang berbeda menghasilkan pengelompokan gejala yang berberda, di mana hal tersebut mempengaruhi dalam proses pengajuan pertanyaan.

Misal *rule*  $G8 \rightarrow G6$ , memiliki makna jika G8 diderita oleh suatu penyakit maka G6 juga diderita oleh penyakit tersebut, tetapi tidak berlaku kebalikannya, sehingga jika G8 mendapat jawaban 'tidak' maka G6 harus ditanyakan kepada user. Secara umum gejala yang diajukan sebagai pertanyaan adalah kelompok gejala *antecedent*, dari sini jika ada *feedback* 'ya' / '1' diperoleh maka proses perhitungan *similarity* dan eliminasi penyakit dapat dilakukan, tetapi jika tidak maka pasangan *consequent* dari *antecedent* tadi harus ditanyakan.



**Gambar 2. Alur proses pengajuan pertanyaan menggunakan *rules* implikasi**

Alur pengajuan pertanyaan menggunakan *rules* gabungan di bawah ini hampir sama dengan yang menggunakan *rules* implikasi, tetapi disini terdapat *rules* kontraposisi hasil penegasian gejala-gejala penyakit. Semisal *rules*  $\sim G42 \rightarrow \sim G100$  bermakna jika G42 tidak diderita oleh suatu penyakit maka G100 juga tidak, dan tidak berlaku sebaliknya, sehingga ketika G42 memperoleh *feedback* 'ya' maka G100 harus ditanyakan kepada *user* sebagai konfirmasi. *Antecedent* pada *rules* kontraposisi ini ditanyakan bersamaan dengan kelompok *antecedent rules* implikasi pada tahap pertama.



**Gambar 3. Alur proses pengajuan pertanyaan menggunakan *rules* gabungan**

Berikut ini adalah contoh proses pengeliminasian dan penentuan penyakit penyakit. Dalam contoh ini diambil kasus seseorang yang akan dideteksi penyakitnya. *Rules* yang digunakan dalam contoh kasus ini adalah *rules* implikasi.

1. Transformasi biner untuk *feedback*

**Tabel 5. Daftar Pertanyaan Tahap Pertama**

Kode	Gejala yang ditanyakan	<i>Feedback</i>	
		asli	Bin er
G8	Batuk	Ya	1
G12	Kedinginan	Ya	1
G14	Penglihatan agak kabur	Ya	1
G21	Nyeri pada dada saat bernafas	Tidak	0
G22	Mengantuk	Tidak	0
G36	Kehilangan keseimbangan	Tidak	0
G37	Dada sering berdebar	Tidak	0
G41	Mual	Tidak	0
G58	Kehilangan gairah seks	Tidak	0

G71	Gemetar	Tidak	0
G73	Sulit untuk konsentrasi	Tidak	0
G94	Kulit, mata, air seni berwarna kuning	Tidak	0
G97	Perut kembung	Tidak	0
G107	Rasa ingin kencing terus	Tidak	0

Pada tabel di atas adalah pertanyaan yang diajukan tahap pertama. Pada tahap pertama ini diajukan 14 pertanyaan yang merupakan *antecedent rules* dari kelompok gejala 1. *Feedback* gejala penyakit yang dirasakan dari pengguna aplikasi diubah ke dalam bentuk himpunan biner.

2. Menerapkan *association rules* pada *feedback* yang telah diperoleh  
Setelah memperoleh *feedback* dari pertanyaan yang diajukan pada tahap pertama, maka selanjutnya adalah mencocokkannya dengan daftar *rules* yang ada. Jika *feedback* merupakan *antecedent* dari sebuah *rule* dan *feedback* tersebut memiliki nilai '1' maka gejala yang merupakan pasangannya juga bernilai '1'.
3. Menghitung nilai *similarity* penyakit dengan *feedback* menggunakan *cosine similarity*.

*Feedback* yang diperoleh dibandingkan dengan biner seluruh penyakit yang ada menggunakan rumus *cosine similarity*. Setelah diperoleh nilai *similarity*-nya dilakukan pengeliminasian penyakit dengan nilai *similarity* rendah menggunakan algoritma berikut:

```
if(sim(A)>=0,5) then A lolos;
```

```
else
```

```
if ((simN(A)-sim(A))<-0,2) then A lolos;
```

```
else A tereliminasi;
```

Beberapa penyakit yang tidak memenuhi persyaratan nilai *similarity* minimalnya  $\geq 0,5$  atau selisih antara *similarity* negatif dan positifnya  $< -0,2$  akan tereliminasi. Setelah itu pada tahapan ini diperoleh beberapa penyakit yang lolos dari proses eliminasi.

4. Mempersiapkan pertanyaan tahap selanjutnya.

Tahap selanjutnya adalah mengajukan pertanyaan berupa sisa gejala yang masih terdapat pada penyakit-penyakit hasil eliminasi.

**Tabel 6. Daftar Pertanyaan Tahap Kedua**

Kode	Gejala yang ditanyakan	<i>Feedback</i>	
		asli	Biner
G13	Mata kemerahan	Ya	1
G15	Hidung terasa gatal dan berair	Ya	1
G17	Tenggorokan sakit saat menelan	Tidak	0
G18	Nafsu makan berkurang/hilang	Tidak	0
G19	Bengkak pada kedua sisi leher	Tidak	0

G109	Nyeri pada perut bagian bawah	Tidak	0
G110	Nyeri saat kencing	Tidak	0
G112	Kencing berdarah	Tidak	0
G113	Bengkak pada pinggang	Tidak	0

Pada tahap ini sama seperti tahap sebelumnya, yaitu *feedback* yang diperoleh ditransformasikan ke dalam bentuk biner, setelah itu jika *feedback* yang bernilai '1' merupakan *antecedent* dari sebuah *rule* maka pasangan gejala *consequent*-nya juga bernilai '1'. Selain itu jika ada *antecedent* yang telah ditampilkan memiliki nilai '0' maka tahap berikutnya adalah menampilkan semua *consequent* dari *antecedent* tersebut.

**Tabel 7. Daftar pertanyaan gejala yang merupakan *consequent* dari pertanyaan sebelumnya**

Kode	Gejala yang ditanyakan	<i>Feedback</i>	
		asli	Biner
G111	Nyeri pada pinggang	Tidak	0

### 5. Penentuan penyakit

Setelah diperoleh *feedback* pada tahap kedua di atas, maka dilakukan penggabungan biner hasil proses tahap pertama dan tahap kedua untuk kemudian dihitung nilai *similarity*-nya dengan penyakit hasil eliminasi pada tahap pertama. Setelah itu penyakit dengan nilai *similarity* kurang dari 0.5 akan dieliminasi. Penyakit yang menjadi hasil perhitungan adalah Influenza dengan nilai *similarity* sebesar 1.

### 3.3 Hasil Pengujian

Output dari pengujian aplikasi ini adalah nilai akurasi yang menyatakan prosentase percobaan yang benar dan jumlah pertanyaan untuk menyatakan efektifitas pengurangan jumlah pertanyaan yang dilakukan oleh aplikasi.

#### 3.3.1 Pengujian dengan *Feedback* Lengkap

Pengujian pertama dengan *feedback* lengkap menggunakan data 33 penyakit dan inputan gejala yang sesuai dengan setiap penyakit yang diuji.

**Tabel 8. Rangkuman hasil pengujian *feedback* lengkap dengan *Rules* implikasi dan *Rules* gabungan**

Pengujian	<i>Rules</i> implikasi			<i>Rules</i> gabungan		
	Percobaan benar	Gagal ditemukan	Akurasi	Percobaan benar	Gagal ditemukan	Akurasi
1	33 (100%)	0 (0%)	100%	33 (100%)	0 (0%)	100%

#### 3.3.2 Pengujian dengan *feedback* random 20%

Pengujian *feedback* random 20% ini menggunakan data 33 penyakit dengan mengosongkan 20% gejala secara acak dalam proses penginputan setiap penyakit yang diuji.

**Tabel 9. Rangkuman hasil pengujian *feedback random* 20% dengan *Rules* implikasi & *Rules* gabungan**

Penguji n	<i>Rules</i> implikasi			<i>Rules</i> gabungan		
	Percobaan benar	Gagal ditemukan	Akurasi	Percobaan benar	Gagal ditemukan	Akurasi
1	33 (100%)	0 (0%)	100%	30 (90,91%)	3 (9,09%)	90,91%
2	31 (93,94%)	2 (6,1%)	93,94%	33 (100%)	0 (0%)	100%
3	33 (100%)	0 (0%)	100%	33 (100%)	0 (0%)	100%
Rata-rata			97,97%	Rata-rata		96,97%

### 3.3.3 Pengujian dengan *feedback random* 40%

Pengujian *feedback random* 40% ini menggunakan data 33 penyakit dengan mengosongkan 40% gejala secara acak dalam proses penginputan setiap penyakit yang diuji.

**Tabel 10. Rangkuman hasil pengujian *feedback random* 40% dengan *Rules* implikasi & *Rules* gabungan**

Penguji n	<i>Rules</i> implikasi			<i>Rules</i> gabungan		
	Percobaan benar	Gagal ditemukan	Akurasi	Percobaan benar	Gagal ditemukan	Akurasi
1	32 (96,97%)	1 (3,03%)	96,97%	30 (90,91%)	3 (9,09%)	90,91%
2	31 (93,94%)	2 (6,1%)	93,94%	31 (93,94%)	2 (6,1%)	93,94%
3	33 (100%)	0 (0%)	100%	33 (100%)	0 (0%)	100%
Rata-rata			96,97%	Rata-rata		94,95%

### 3.3.4 Pengujian dengan *feedback data real*

Pada pengujian keempat, digunakan 30 data *real* dari hasil wawancara terprogram terhadap 30 orang responden. Hasil wawancara tersebut diperoleh 16 jenis penyakit yang pernah diderita oleh responden dalam kurun waktu tidak lebih dari tiga bulan.

**Tabel 11. Rangkuman hasil pengujian *feedback data real* dengan *Rules* implikasi.**

Penguji n	<i>Rules</i> implikasi			<i>Rules</i> gabungan		
	Percobaan benar	Gagal ditemukan	Akurasi	Percobaan benar	Gagal ditemukan	Akurasi
1	29 (96,97%)	1 (3,33%)	96,97%	29 (96,97%)	1 (3,33%)	96,97%

**Tabel 12. Jumlah pertanyaan yang harus dijawab di setiap pengujian dengan *Rules* implikasi**

Pengujian	Percobaan benar	Jumlah Pertanyaan yang harus dijawab		
		Minimal	Maksimal	Rata-rata
<i>Feedback lengkap</i>	33	23 (19,17%)	61 (47,66%)	36,33 (28,38%)
<i>Random 20% I</i>	33	23 (19,17%)	61 (47,66%)	40,55 (31,68%)
<i>Random 20% II</i>	31	21 (16,41%)	61 (47,66%)	38,32 (29,94%)
<i>Random 20% III</i>	33	23 (19,17%)	61 (47,66%)	35,76 (27,94%)
<i>Random 40% I</i>	32	19 (14,84%)	61 (47,66%)	39,56 (30,91%)
<i>Random 40% II</i>	31	20 (15,63%)	61 (47,66%)	38,65 (30,20%)
<i>Random 40% III</i>	33	23 (19,17%)	61 (47,66%)	38,64 (30,19%)
<i>Feedback data real</i>	29	17 (13,28%)	61 (47,66%)	33,76 (26,78%)
Rata-rata keseluruhan				37,70 (29,45%)

**Tabel 13. Jumlah pertanyaan yang harus dijawab di setiap pengujian dengan Rules gabungan**

Pengujian	Percobaan benar	Jumlah Pertanyaan yang harus dijawab		
		Minimal	Maksimal	Rata-rata
<i>Feedback lengkap</i>	33	21 (16,41%)	57 (44,53%)	36,45 (28,48%)
<i>Random 20% I</i>	30	21 (16,41%)	57 (44,53%)	40,13 (31,35%)
<i>Random 20% II</i>	33	23 (19,17%)	57 (44,53%)	40,42 (31,58%)
<i>Random 20% III</i>	33	21 (16,41%)	57 (44,53%)	38,27 (29,90%)
<i>Random 40% I</i>	30	21 (16,41%)	57 (44,53%)	40,07 (31,30%)
<i>Random 40% II</i>	31	21 (16,41%)	57 (44,53%)	40,35 (31,52%)
<i>Random 40% III</i>	33	21 (16,41%)	57 (44,53%)	39,94 (31,20%)
<i>Feedback data real</i>	29	21 (16,41%)	57 (44,53%)	36,83 (28,77%)
Rata-rata keseluruhan				39,06 (30,51%)

Pada Tabel 12 dan 13 di atas menunjukkan jumlah pertanyaan yang harus dijawab di setiap pengujian yang dilakukan (hanya percobaan yang benar, yang diperhitungkan di dalam tabel ini) pada penelitian ini.

### 3.4 Pembahasan

Pada pengujian *feedback* lengkap yang diujikan pada setiap penyakit menunjukkan bahwa setiap penyakit yang diuji keluar sebagai *output* aplikasi pada urutan pertama dan menghasilkan nilai akurasi 100%. Hal ini menunjukkan bahwa akurasi aplikasi dari kedua bentuk *rules* yang digunakan dalam kasus pengujian ini sempurna.

Pengujian kedua, yaitu *feedback random* 20%, *rules* implikasi menghasilkan nilai akurasi sebesar 97,97% dengan total 2 kegagalan dari 99 percobaan. Sedangkan *rules* gabungan menghasilkan nilai akurasi sebesar 96,97% dengan 3 kegagalan dari 99 percobaan. Hal ini menunjukkan bahwa secara keseluruhan aplikasi yang menerapkan *rules* implikasi sebagai acuannya memiliki akurasi yang lebih baik.

Pengujian kedua, yaitu *feedback random* 40%, *rules* implikasi menghasilkan nilai akurasi sebesar 96,97% dengan total 3 kegagalan dari 99 percobaan. Sedangkan *rules* gabungan menghasilkan nilai akurasi sebesar 94,95% dengan 5 kegagalan dari 99 percobaan. Hal ini menunjukkan bahwa secara keseluruhan aplikasi yang menerapkan *rules* implikasi sebagai acuannya memiliki akurasi yang lebih baik. Penyebab kegagalan selain karena gejala yang seharusnya menjadi inputan, tidak dimunculkan pada tahap pertama pengajuan pertanyaan, juga jumlah inputan yang semakin berkurang yang mempengaruhi perhitungan *similarity* sehingga eliminasi penyakit menjadi tidak akurat. Meskipun demikian, *rules* implikasi menghasilkan nilai akurasi yang lebih baik.

Pengujian *feedback* data *real* yang merupakan pengujian keempat, menunjukkan hasil akurasi yang sama baiknya dari penerapan masing-masing *rules*. Pada penerapan *rules* implikasi maupun *rules* gabungan, kegagalan hanya terjadi 1 kali dalam 30 percobaan, hal ini menunjukkan bahwa dalam kasus ini penerapan *rules* kontraposisi lebih baik karena menghasilkan nilai akurasi yang lebih tinggi yaitu sebesar 96,67%. Kegagalan pada pengujian ini disebabkan karena kurangnya gejala yang dirasakan oleh responden yang seharusnya menjadi input aplikasi.

Meskipun secara akurasi, penggunaan *rules* implikasi dan *rules* gabungan memiliki hasil yang sama baiknya akan tetapi dalam beberapa kasus percobaan, nilai *similarity* yang dihasilkan oleh penerapan *rules* implikasi lebih besar dibandingkan ketika menerapkan *rules* gabungan.

Pada penerapan *rules* implikasi, rata-rata jumlah pertanyaan yang harus dijawab disetiap percobaan adalah 38 pertanyaan gejala (Tabel 12 dengan rata-rata 37,70) dari 128 gejala penyakit yang ada atau sebesar 29,45% sehingga dapat dikatakan dengan penerapan *rules* implikasi ini menghasilkan efektifitas pengurangan jumlah pertanyaan yang diajukan sebesar 70,55%. Sedangkan pada penerapan *rules* gabungan, rata-rata jumlah pertanyaan yang harus dijawab disetiap percobaan adalah 39 pertanyaan gejala (Tabel 13 dengan rata-rata 39,06) dari 128 gejala penyakit atau sebesar 30,51% sehingga penerapan *rules* gabungan ini menghasilkan efektifitas pengurangan jumlah pertanyaan yang diajukan sebesar 69,49%.

Tujuan lain penelitian untuk meminimalisir jumlah pertanyaan sudah tercapai karena jumlah pertanyaan yang harus dijawab jauh lebih sedikit jika dibandingkan dengan pertanyaan yang harus dijawab tanpa pengelompokan gejala dan pengelompokan penyakit yang jumlahnya mencapai 128 gejala atau pengelompokan gejala tanpa menggunakan metode *association rules*.

## 5. PENUTUP

Pengujian *feedback* lengkap yang bertujuan untuk menguji kebenaran aplikasi untuk setiap penyakitnya menghasilkan nilai akurasi 100%. Artinya, aplikasi ini telah berhasil dalam pengujian terhadap setiap penyakit yang ada dan menghasilkan *output* penyakit sesuai harapan. Sedangkan pengujian menggunakan *feedback random* 20%, *random* 40%, dan *feedback data real* menunjukkan bahwa akurasi aplikasi ini sangat baik untuk kedua bentuk *rules*. Meskipun demikian penerapan *rules* implikasi menghasilkan nilai akurasi dan nilai *similarity* lebih tinggi.

Penerapan metode *Association Rules* untuk pengelompokan gejala dan *Cosine Similarity* untuk pengelompokan penyakit dalam aplikasi deteksi dini penyakit dapat mengurangi jumlah pertanyaan yang diberikan secara signifikan. Efektifitas pengurangan jumlah pertanyaan yang diajukan menggunakan *rules* implikasi sebesar 70.55% atau 29.45% (37.70 dari 128 ) pertanyaan yang harus dijawab, sedangkan efektifitas pengurangan jumlah pertanyaan yang diajukan menggunakan *rules* gabungan sebesar 69.49% atau 30.51% (39.06 dari 128 ) pertanyaan yang harus dijawab. Hal ini menunjukkan bahwa penggunaan *rules* implikasi memiliki hasil yang lebih baik dari sisi kemampuannya dalam mengurangi jumlah pertanyaan yang diajukan.

Kegagalan pada beberapa percobaan disebabkan karena tidak ada atau kurangnya gejala yang dirasakan sebagai *feedback* aplikasi pada seleksi awal penyakit tersebut. Sehingga penyakit yang kemungkinan diderita menjadi tereliminasi dan tidak diikuti dalam proses selanjutnya karena nilai *similarity*-nya berada di bawah batas minimal.

## DAFTAR PUSTAKA

- [1] Karhendana, A., “**Pemanfaatan Document Clustering pada Agregator Berita**”, Program Studi Teknik Informatika, Institut Teknologi Bandung, Bandung, 2008.
- [2] Sebayu, R.P.E., Saptono, R., Wiharto, “**Pengembangan Fitur Deteksi Dini Penyakit pada Toko Online Obat Herbal dengan Metode Cosine Similarity**”, Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret, Surakarta, Januari 2013.
- [3] Tan, Pang-Ning, Michael Steinbach, Vipin Kumar, “**Introduction to Data Mining**”, Pearson Education Inc, London, 2006.

- [4] Katno, dan Pramono, S. **“Tingkat Manfaat dan Keamanan Tanaman Obat dan Obat Tradisional”**, Balai Penelitian Tanaman Obat Tawangmangu, 2003.
- [5] Kantardzic, M. **“Data Mining: Concepts, Models, Methods, and Algorithms”**, John Wiley & Sons. New Jersey, 2003.
- [6] Utomo, P., **“Apresiasi Penyakit, Pengobatan Secara Tradisional dan Modern”**, PT Rineka Cipta, Jakarta, 2005.