

ISSN : 2301-7201

ITSMART

Volume 4. Nomor 1. Juni 2015



JURNAL ITSMART

Teknik Informatika FMIPA UNS

Penanggung Jawab

Kepala Program Studi Informatika FMIPA UNS

Pemimpin Redaksi

Rini Anggrainingsih, M.T

Sekretaris

Afrizal Doewes, S.Kom, M.eng

Bendahara

Esti Suryani, S.Si, M.Kom

Mitra Bestari

Dr. Ir Abdul Kadir, M.M, M.T (UTM Malaysia)

Dr Lukito Edi Nugroho (UGM)

Dr. Anto Satriyo Nugroho (BPPT)

Dewan Redaksi

Dr. Wiranto

Drs. Bambang Harjito, Ph.D

Heri Prastyawan, Ph.D

Umi Salamah, S.Si, M.Kom

Wiharto, S.T, M.Kom

Redaksi Pelaksana

Ristu Saptono, S.Si, M.T

Abdul Aziz, S.Kom, M.Cs

Sari Widya Sihwi, S.Kom, M.TI

Hasan Dwi Cahyono, S.Kom, M.Kom

Alamat Redaksi :

Program Studi Informatika FMIPA UNS

Jl. Ir. Sutami 36A Kertingan Surakarta

Telp. (0271) 663451, Fax. (0271) 663451

Email : itsmart_uns@yahoo.co.id

ISSN : 2301–7201

DAFTAR ISI

- Pengembangan Sistem Analisis Akademis Menggunakan OLAP dan *Data Clustering* Studi Kasus : Akademik Universitas Sebelas Maret Surakarta 1 – 11
Bakharudin Yusuf B, Antonius Bima Murti W, Hasan Dwi Cahyono
- Digital Watermarking Image dengan Menggunakan Discrete Wavelet Transform dan Singular Value Decomposition (DWT-SVD) untuk Copyright Labeling 12 – 19
Nur Anisah, Bambang Harjito, Esti Suryani
- Pembuatan Peta Data Prediksi Penderita Penyakit DBD Di Kota Surakarta Dengan Metode *Susceptible-Infected-Removed(SIR)* 20 – 27
Diannita Kartikasari, Sarngadi Palgunadi, Bambang Harjito
- Penentuan Model Terbaik pada Metode Naive Bayes Classifier dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter 28 – 35
Apriliya Fitri C, Ristu Saptono, Sari Widya Sihwi
- Sistem Klasifikasi *Feedback* Pelanggan Dan Rekomendasi Solusi Atas Keluhan Di UPT Puskom UNS Dengan Algoritma *Naive Bayes Classifier* Dan *Cosine Similarity* 36 – 42
Aisha Alfiani Mahardhika, Ristu Saptono, Rini Anggrainingsih
- Sistem Pakar Untuk Mendiagnosis Penyakit Infeksi Menggunakan Forward Chaining 43 – 47
Ahmad Aniq Noor Mutsaqof, Wiharto, Esti Suryani

INDEX JUDUL

Pengembangan Sistem Analisis Akademis Menggunakan OLAP dan Data Clustering Studi Kasus : Akademik Universitas Sebelas Maret Surakarta, 1 – 11

Digital Watermarking Image dengan Menggunakan Discrete Wavelet Transform dan Singular Value Decomposition (DWT-SVD) untuk Copyright Labeling, 12 - 19

Pembuatan Peta Data Prediksi Penderita Penyakit DBD Di Kota Surakarta Dengan Metode Susceptible-Infected-Removed(SIR), 20 – 27

Penentuan Model Terbaik pada Metode Naive Bayes Classifier dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter, 28 – 35

Sistem Klasifikasi Feedback Pelanggan Dan Rekomendasi Solusi Atas Keluhan Di UPT Puskom UNS Dengan Algoritma Naive Bayes Classifier Dan Cosine Similiarity, 36 – 42

Sistem Pakar Untuk Mendiagnosis Penyakit Infeksi Menggunakan Forward Chaining, 43 – 47

INDEX PENULIS

Ahmad Aniq Noor Mutsaqof, 43 – 47

Aisha Alfiani Mahardhika, 36 – 42

Antonius Bima Murti W, 1 – 11

Apriliya Fitri C, 28 – 35

Bakharudin Yusuf B, 1 – 11

Bambang Harjito, 12 – 19, 20 - 27

Diannita Kartikasari, 20 – 27

Esti Suryani, 12 – 19, 43 – 47

Hasan Dwi Cahyono, 1 – 11

Nur Anisah, 12 – 19

Rini Anggrainingsih, 36 – 42

Ristu Saptono, 28 – 35, 36 – 42

Sari Widya Sihwi, 28 – 35

Sarngadi Palgunadi, 20 – 27

Wiharto, 43 – 47

PENGEMBANGAN SISTEM ANALISIS AKADEMIS MENGGUNAKAN OLAP DAN DATA CLUSTERING STUDI KASUS : AKADEMIK UNIVERSITAS SEBELAS MARET SURAKARTA

Bakharudin Yusuf Bakhtiar

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta
bakharudin@student.uns.ac.id

Antonius Bima Murti Wijaya

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta
bimamurti@uns.ac.id

Hasan Dwi Cahyono

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta
hasandc@staff.uns.ac.id

Abstrak— Tingkat keberhasilan universitas dapat dilihat dari kualitas lulusannya. Salah satu cara meningkatkan keberhasilan tersebut adalah dengan melengkapi dan memanfaatkan sistem informasi. Universitas Sebelas Maret merupakan universitas yang memanfaatkan sistem informasi secara *departemental* karena beberapa dibangun dengan menggunakan *framework* dan *database* yang berbeda. Dimana hal ini akan menyulitkan civitas akademik dalam mengolah data dan menganalisis data-data tersebut. Aplikasi Pengembangan Sistem Analisis Akademis Menggunakan OLAP dan Data *Clustering* ini menjadi solusi untuk mengatasi masalah tersebut sekaligus membantu menganalisis data-data. Proses pembuatan aplikasi ini meliputi perancangan *data warehouse*, pengambilan data akademik dari sumber data, proses *extraction, transformation, loading* (ETL), pembuatan *cube*, pembuatan laporan, dan pengelompokan data menggunakan K-means *clustering*.

Penelitian ini berhasil mengembangkan sistem OLAP dan *clustering* untuk memenuhi fungsionalitas sistem yaitu sistem dapat melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah mahasiswa, jumlah calon mahasiswa baru, jumlah lulusan, mencari nilai maksimum, minimum, rata-rata dan presentase profil ipk lulusan, mencari nilai maksimum, minimum, rata-rata dan presentase profil lama studi lulusan, beban sks dosen, presentase pemakaian ruang dari berbagai dimensi.

Kata Kunci : OLAP, K-means, Clustering, ETL, Data warehouse.

1. PENDAHULUAN

Menurut Prof. Dr. Ravik Karsidi, M.S sebagai rektor dari Universitas Sebelas Maret (UNS) periode 2011-2015 menyatakan bahwa keberhasilan UNS yang lebih utama terlihat dari semakin meningkatnya kualitas lulusan yang ditandai dengan semakin meningkatnya rata-rata Indeks Prestasi Kumulatif (IPK), Angka Efisiensi Edukasi (AEE), lulusan berpredikat *cumlaude* serta semakin singkatnya masa studi. Salah satu cara yang digunakan UNS untuk meningkatkan keberhasilan tersebut adalah dengan melengkapi dan memanfaatkan sistem informasi.

Namun sistem informasi yang ada di UNS sampai saat ini masih bersifat *departemental* karena beberapa dibangun dengan menggunakan *framework* dan *database* yang berbeda. Dengan sistem yang masih bersifat *departemental* ini akan menyulitkan civitas akademik dalam mengolah data dan

menganalisis data-data tersebut karena harus dihimpun dari berbagai unit yang sulit untuk didapatkan secara cepat dan terkadang terdapat data yang tidak valid. Seperti kesulitan dalam pembuatan laporan, penyusunan borang akreditasi, menentukan segmentasi pasar promosi SMA dan lain sebagainya.

Dalam penelitian yang dilakukan oleh Kavitha [1] menyebutkan bahwa *data warehouse* merupakan dasar dari OLAP yang dapat melakukan analisis interaktif data multidimensi. *Online analytical processing* (OLAP) adalah sebuah teknik yang digunakan untuk menggabungkan data sehingga memungkinkan untuk membantu *user* memecahkan masalah bisnis dengan menggali data transaksional [2]. Dengan menggunakan OLAP, data yang bersifat *departemental* seperti yang terjadi di UNS dapat akan digabungkan menjadi sebuah database terpusat dengan konsep *multidimensi*. Kemudian dalam penelitian Usman [3] Kombinasi OLAP dan data Mining dapat digunakan untuk memvisualisasikan data yang kompleks agar dapat dianalisa secara efisien, interaktif dan bermakna. Untuk membantu melakukan analisis data-data yang terdapat dalam OLAP dapat menggunakan algoritma *clustering*. *Clustering* digunakan untuk mengelompokkan data sesuai dengan kesamaan karakteristik yang dimiliki setiap data dalam kelompok tersebut [4]. Algoritma K-means adalah salah satu metode *clustering* non hirarki yang paling populer dan banyak dikembangkan karena algoritma K-Means sederhana, efisien dan selalu konvergen [5]. Seperti halnya di UNS, algoritma K-means ini dapat digunakan untuk mengelompokkan data mahasiswa dimana data-data yang akan digunakan berasal dari *data warehouse*.

Berdasarkan permasalahan tersebut maka penulis ingin membuat sistem *clustering* K-means yang terintegrasi dengan OLAP melalui *data warehouse* sebagai sumber datanya dimana hasil dari *clustering* ini dapat membantu prodi UNS dalam menentukan segmentasi pasar SMA yang menjadi prioritas promosi program studi UNS.

2. DASAR TEORI

2.1. Data warehouse

Data warehouse merupakan database relasional yang didesain lebih kepada query dan analisis daripada proses transaksi, biasanya mengandung *history* data dari proses transaksi dan bisa juga data dari sumber lainnya. *Data warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi untuk menggabungkan data dari berbagai macam sumber [6] sehingga digunakan untuk membantu mendukung keputusan-keputusan manajemen [7].

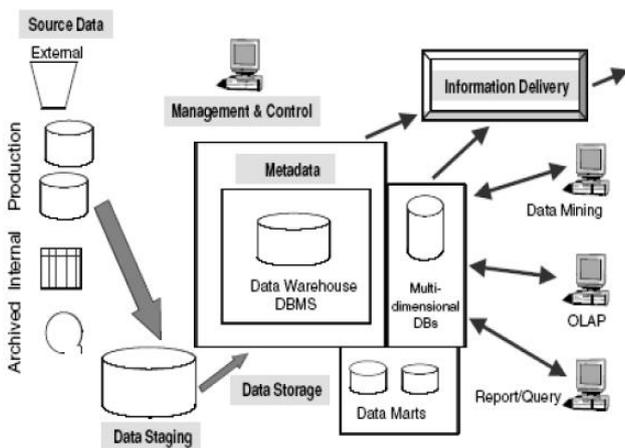
2.1.1. Karakteristik Data warehouse

Menurut Bill Inmom dalam buku Ponniah, karakteristik dari *data warehouse* [7] adalah sebagai berikut:

- a. Berorientasi subjek
Data warehouse didesain untuk menganalisis data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu.
- b. Terintegrasi
Sumber data yang ada dalam *data warehouse* tidak hanya berasal dari data operasional (internal source) tetapi juga berasal dari data di luar sistem (external source).
- c. *Time-variant*
Sistem operasional mengandung data yang bernilai sekarang sedangkan data dalam *data warehouse* mengandung data tidak hanya data terkini tetapi juga data masa lampau yang akan digunakan dalam analisis dan pengambilan keputusan.
- d. *Non-volatile*
Data dalam database operasional akan secara berkala atau periodik dipindahkan ke dalam *data warehouse* sesuai dengan jadwal yang sudah ditentukan.
- e. *Granularity*
Pada sistem operasional data dibuat secara real-time sehingga untuk mendapatkan informasi langsung dilakukan proses query. Granularitas menunjuk pada level perincian atau peringkasan yang ada pada unit-unit data dalam *data warehouse*.

2.1.2. Komponen Data warehouse

Menurut Ponniah [7] komponen-komponen *data warehouse* digambar sesuai dengan gambar 1 dimana komponen sumber data berada di sebelah kiri. kemudian komponen data *staging* sebagai blok pembangun. Komponen penyimpanan data yang mengelola *data warehouse* berada di tengah. Komponen *information delivery* yang terdiri dari semua hal menyediakan informasi dari *data warehouse* bagi pengguna berada di sebelah kanan.



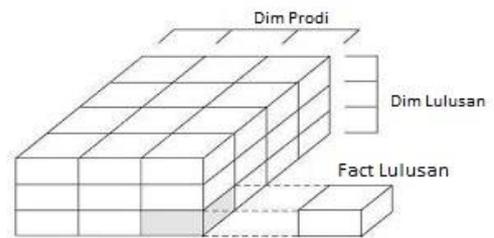
Gambar 1. Komponen *Data warehouse* [7]

Komponen sumber data yang digunakan dalam *data warehouse* dikelompokkan menjadi empat, yaitu: *Production Data, Internal Data, Archived Data, External Data*.

Didalam data staging akan dilakukan *Extraction, Transformation, dan Loading (ETL)* yang digunakan untuk membersihkan, mengubah, menggabungkan, mengkonversi, mencegah duplikasi data, dan menyiapkan data sumber untuk penyimpanan dan penggunaan dalam *data warehouse*.

2.2. Model Data Multidimensional

Pembuatan *data warehouse* didasarkan pada model data multidimensi yang berbentuk kubus. Data multidimensi adalah ketika sebuah data dapat dipandang dari berbagai sudut. Pusat dari objek pada multidimensional adalah *cube* atau kubus yang mengandung hubungan struktur dimensi, hirarki, level dan anggota.



Gambar 2. Data Multidimensional

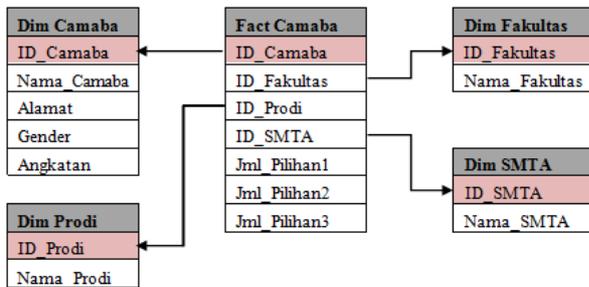
Menurut Prasetyo [8], komponen model multidimensional yang secara umum ditemukan dalam perancangan *data warehouse* adalah :

- a. Dimensi
merupakan kategori yang independen dari multidimensional database. Tipe dari dimensi ini mengandung item yang digunakan sebagai kriteria query untuk ukuran database.
- b. Tabel fakta
Tabel fakta merupakan pusat dari schema pada OLAP yang didalamnya mempunyai dua tipe kolom, yaitu kolom *measure* yang menyimpan nilai-nilai numerik dan kolom yang menyimpan *foreign key* yang mengacu ke tabel dimensi.
- c. Measure
Measure juga cerminan dari fakta dan juga mengandung data yang akan dianalisis. OLAP memerlukan informasi kolom bertipe numerik yang akan dijadikan *measure*
- d. Hirarki
Hirarki merupakan bentuk kesatuan dari dimensi. Hirarki didefinisikan bagaimana hubungan antar level.
- e. Level
Level merepresentasikan sebuah posisi pada hirarki. Level mengumpulkan data untuk agregasi dan digunakan untuk proses komputasi.
- f. Atribut
Attribute merepresentasikan informasi tambahan pada sebuah level tertentu.
- g. Cube
Cube adalah obyek OLAP yang tersusun dari *measure, dimensi dan attribute*. Sisi-sisi pada *cube* ditentukan oleh masing-masing dimensi yang terlibat dalam *cube* itu.

Menurut prasetyo [8] , model dimensional yang sering digunakan pada *data warehouse* adalah star atau snowflake skema.

a. Star schema

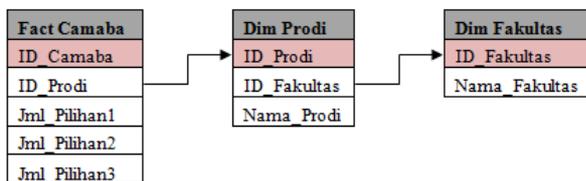
Star schema seperti yang terlihat pada gambar 3, terdiri dari satu atau lebih tabel fakta dan satu atau lebih tabel dimensi. Tabel fakta merupakan pusat dari star schema, karena fungsinya sebagai pengikat dari tabel-tabel dimensi yang terletak di sekelilingnya.



Gambar 3. Star Schema

b. Snowflake schema

Snowflake schema merupakan pengembangan dari star schema. Pada snowflake schema, untuk mengurangi nilai duplikat pada tabel tabel dimensi akan dilakukan normalisasi secara sebagian atau keseluruhan. Dengan kata lain satu atau lebih tabel dimensi tidak bergabung secara langsung kepada tabel fakta tapi pada tabel dimensi lainnya seperti pada gambar 4.



Gambar 4. Snowflake Schema

2.3. Clustering

Menurut Alfina, pengertian clustering dalam data mining adalah pengelompokan sejumlah data atau objek ke dalam cluster (group) sehingga setiap dalam cluster tersebut akan berisi data yang semirip mungkin dan berbeda dengan objek yang berada dalam cluster lainnya [10].

2.3.1. K-means Clustering

Algoritma K-means termasuk dalam partitioning clustering yang memisahkan data ke k daerah bagian yang terpisah. K-means merupakan algoritma yang sangat terkenal karena kemudahannya dan kemampuannya untuk mengklaster data besar dan data outlier dengan sangat cepat. Sesuai dengan karakteristik partitioning clustering, setiap data harus termasuk ke dalam cluster tertentu, dan memungkinkan bagi setiap data dalam satu tahapan proses tertentu sudah masuk dalam satu cluster, pada satu tahapan berikutnya data tersebut akan berpindah ke cluster lain [11].

langkah-langkah dalam pembuatan algoritma K-Means adalah sebagai berikut [12]:

1. Menentukan k sebagai jumlah cluster yang ingin dibentuk.
2. Membangkitkan nilai random untuk pusat cluster awal (centroid) sebanyak k.
3. Menghitung jarak setiap data input terhadap masing-masing centroid menggunakan rumus jarak Euclidian

(Euclidian Distance) hingga ditemukan jarak yang paling dekat dari setiap data dengan centroid. Berikut adalah persamaan Euclidian Distance:

$$d(x_i, \mu_j) = \sqrt{(x_i - \mu_j)^2} \quad (1)$$

dimana:

x_i : data kriteria

μ_j : centroid pada cluster ke-j

4. Mengklasifikasikan setiap data berdasarkan kedekatannya dengan centroid (jarak terkecil).
5. Memperbaharui nilai centroid. Nilai centroid baru diperoleh dari rata-rata cluster yang bersangkutan dengan menggunakan rumus:

$$\mu_j(t + 1) = \frac{1}{N_{sj}} \sum_{j \in S_j} x_j \quad (2)$$

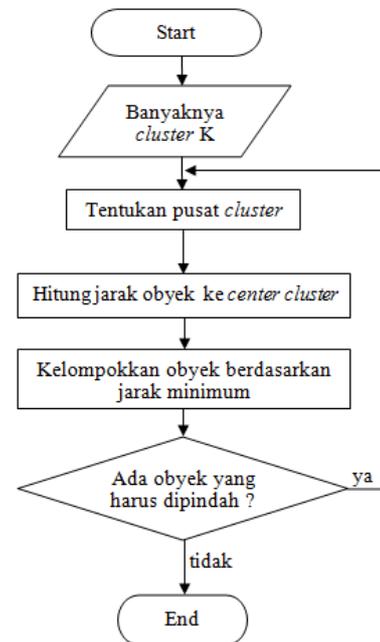
dimana:

$\mu_j(t+1)$: centroid baru pada iterasi ke (t+1),

N_{sj} : banyak data pada cluster S_j

6. Melakukan perulangan dari langkah 2 hingga 5 hingga anggota tiap cluster tidak ada yang berubah.
7. Jika langkah 6 telah terpenuhi, maka nilai pusat cluster (μ_j) pada iterasi terakhir akan digunakan sebagai parameter untuk menentukan klasifikasi data.

Berikut adalah gambar flowchart algoritma K-Means :



Gambar 5. Flowchart Algoritma K-means [13]

2.4. Sum Square Error (SSE)

SSE (Sum Square Error) adalah salah satu metode statistik yang dipergunakan untuk mengukur selisih total dari nilai sebenarnya terhadap nilai yang tercapai [14].

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} d^2(m_i, x) \quad (3)$$

Dimana

:

- d : jarak titik representatif ke pusat cluster
- x : pusat cluster dalam cluster C_i

- m : titik representatif untuk *cluster* C_i
 Hasil *clustering* yang baik adalah jika memiliki nilai SSE paling rendah.

3. METODOLOGI PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut :

3.1. Tahap pengumpulan data

3.1.1. Data primer

Data atau informasi yang diperoleh secara langsung dari pihak narasumber, dalam kasus ini narasumber adalah civitas akademik UNS. Setiap jawaban dari pertanyaan dan uraian hasil wawancara merupakan data yang nantinya dapat digunakan dalam menganalisis kebutuhan sistem yang akan dibuat.

3.1.2. Data sekunder

Pengumpulan data sekunder ini dapat dilakukan dengan menggunakan metode studi literatur dan telaah dokumen. Studi literatur dilakukan dengan mencari bahan materi yang berhubungan dengan permasalahan, perancangan, *data warehouse*, proses ETL, OLAP dan metode *K-Means Clustering* untuk mempermudah proses implementasi sistem. Pencarian materi dilakukan melalui pencarian di buku panduan, internet dan dokumen yang didapatkan dari UNS.

3.2. Tahap analisis

Dalam tahap analisis membuat Spesifikasi Kebutuhan Perangkat Lunak (SKPL) yang terdiri dari : menentukan tabel dan kolom pada data sumber yang akan diambil, merancang tabel dimensi dan tabel fakta *data warehouse* dengan menggunakan *snowflake skema*, membuat usecase untuk aplikasi *K-means clustering* dan membuat ERD untuk aplikasi *K-means clustering*.

3.3. Tahap desain

Dalam tahap desain ini terdiri dari dua tahapan yaitu membuat Deskripsi Perancangan Perangkat Lunak (DPPL) dan merancang algoritma *K-means clustering*.

3.4. Tahap implementasi

3.4.1. Implementasi pembersihan data (ETL)

Dalam tahapan ini, sumber data yang berasal dari database SIAKAD, SPMB, dan SIGEJE akan di *extract* ke dalam data *staging* kemudian data harus ditransformasikan sesuai dengan format desain *data warehouse* yang dibuat. Setelah data dibersihkan baru kemudian dilakukan proses *loading* ke *data warehouse*.

3.4.2. Implementasi Pembuatan OLAP

Setelah proses ETL selesai maka *data warehouse* siap digunakan untuk pembuatan *cube multidimensional* OLAP.

3.4.3. Implementasi algoritma K-means clustering

Tahap ini akan mengimplementasikan keseluruhan algoritma *k-means* yang ada di tahap proses *clustering* data.

3.5. Pengujian validasi hasil

Dalam tahapan ini dilakukan pengujian hasil *clustering* menggunakan *Sum of Square Error (SSE)* untuk menghitung kesalahan jarak dari cluster terdekat. Hasil *clustering* yang baik

adalah jika pada *iterasi* terakhir memiliki nilai SSE yang paling rendah dibanding *iterasi* sebelumnya.

4. HASIL DAN PEMBAHASAN

4.1. Pembangunan data warehouse dan OLAP

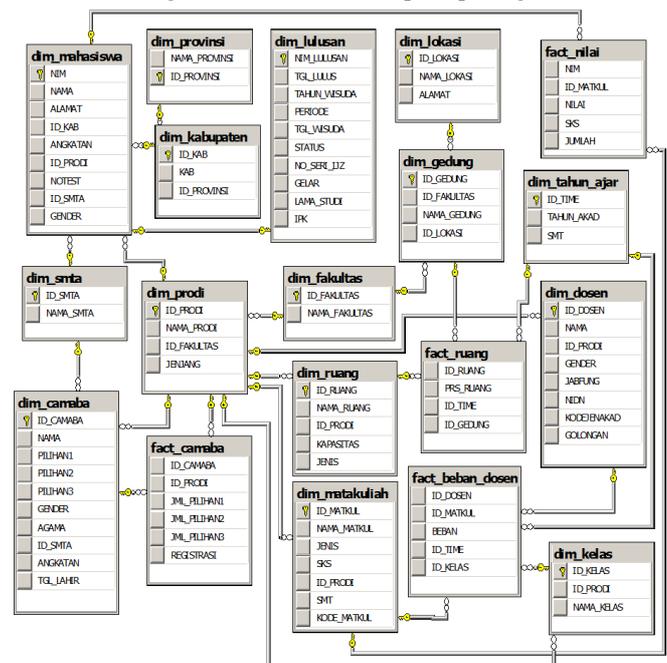
Setelah dilakukan analisis terhadap sumber data untuk mengetahui apa saja informasi yang harus disediakan oleh sistem. Fungsi-fungsi yang dapat ditangani sistem adalah :

- Melihat jumlah mahasiswa, jumlah lulusan, jumlah mahasiswa yang diwisuda, jumlah calon mahasiswa baru dari berbagai dimensi.
- Melihat jumlah calon mahasiswa baru untuk setiap prodi.
- Melihat profil IPK dan lama studi lulusan.
- Melihat profil beban sks dosen.
- Melihat jumlah mahasiswa tiap kelas.
- Melihat presentase pemakaian gedung.
- Melakukan pengelompokkan data

Untuk memenuhi kebutuhan fungsi-fungsi tersebut, sistem akan mengambil data-data yang terdapat dalam database SIAKAD, SPMB dan SIGEJE.

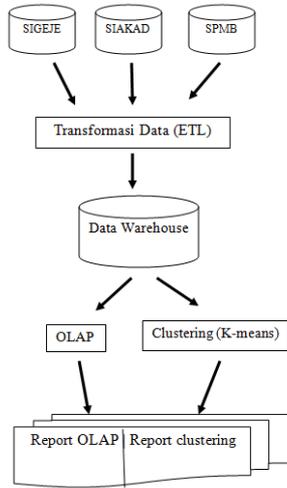
Data-data dari data sumber tersebut akan diolah sesuai dengan format yang diinginkan, seperti data yang mempunyai format "*nvarchar*" akan diubah menjadi "*varchar*" dan data-data yang akan digunakan untuk proses perhitungan, tipe datanya akan diubah sesuai dengan kebutuhan seperti dari format "*varchar*" ke format "*float*" atau dari format "*varchar*" ke format "*integer*".

Dari analisis yang dilakukan dibuat sebuah skema model data menggunakan *snowflake schema* yang akan digunakan untuk membangun *data warehouse* seperti pada gambar 6.



Gambar 6. Rancangan *snowflake schema*

Dalam proses pengambilan data dari data sumber sampai dengan pembuatan laporan akan dilakukan seperti arsitektur perangkat lunak pada gambar 7.



Gambar 7. Arsitektur Perangkat Lunak

Pada table 1 merupakan table sumber data, *staging area* dan *data warehouse*.

Tabel 1. Daftar Tabel sumber data, *staging area* dan *data warehouse*.

| Database Sumber | Tabel Sumber | Staging Area | Data warehouse (Result) | Data warehouse (Area) |
|-----------------|-----------------------------|--------------|-------------------------|---|
| SIGEJE | FAKULTAS | fakultas | dim_fakultas | - |
| SIGEJE | PRODI | prodi | dim_prodi | - |
| SIGEJE | LOKASI | lokasi | dim_lokasi | - |
| SIAKAD | replika_identitas_mahasiswa | r_mahasiswa | dim_provinsi | - |
| SIAKAD | replika_identitas_mahasiswa | r_mahasiswa | dim_kabupaten | dim_provinsi |
| SIGEJE | DOSEN | dosen | dim_dosen | - |
| SPMB | replika_camaba | r_camaba | dim_smta | - |
| SIAKAD | replika_identitas_mahasiswa | r_mahasiswa | dim_mahasiswa | dim_SMTA, dim_kabupaten, dim_provinsi, dim_smta |
| SIGEJE | MAKULPENAWARAN | mapen | dim_matakuliah | - |
| | MATAKULIAH | Matakuliah | | |
| SIGEJE | RUANG | ruang | dim_ruang | dim_prodi |
| SIGEJE | TAS | TAS | dim_tahun_ajar | - |
| SPMB | replika_camaba | r_camaba | dim_camaba | dim_smta |
| SIAKAD | replika_alumni | r_alumni | dim_lulusan | dim_mahasiswa |
| SIGEJE | KELAS | kelas | dim_kelas | - |
| | PRODI | prodi | | |
| SIGEJE | LOKASI | lokasi | dim_lokasi | - |
| SIGEJE | GEDUNG | gedung | dim_gedung | - |

| | | | | |
|--------|---------------------|-------------|------------------|-------------------------------|
| | | | fact_camaba | dim_camaba, dim_mahasiswa |
| SIGEJE | PLOTDOSEN | plot_dosen | fact_beban_dosen | dim_matakuliah, dim_kelas |
| SIGEJE | MAKULPENAWARAN | mapen | | |
| SIAKAD | replika_khs_mipa | r_khs | fact_nilai | dim_mahasiswa, dim_matakuliah |
| | replika_khs_ekonomi | | | |
| | replika_khs_fk | | | |
| | replika_khs_fisip | | | |
| | replika_khs_fkip | | | |
| | replika_khs_hukum | | | |
| | replika_khs_teknik | | | |
| | replika_khs_sastra | | | |
| SIAKAD | replika_alumni | r_alumni | fact_lulusan | dim_mahasiswa |
| SIGEJE | RUANG | ruang | fact_ruang | |
| SIGEJE | WAKTU | waktu | | |
| SIGEJE | RUANGWAKTU | ruang_waktu | | |

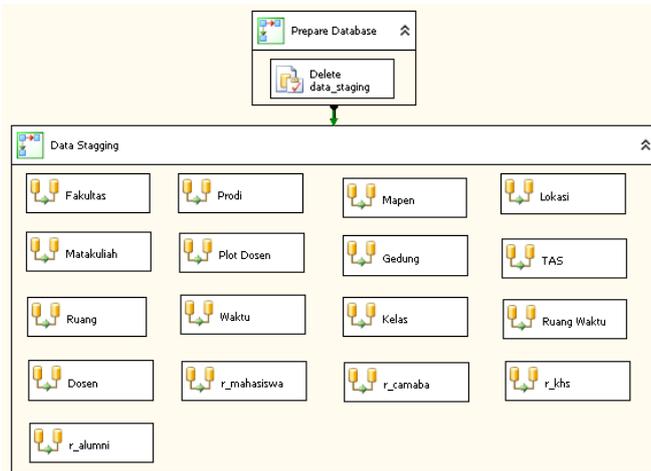
Tahapan-tahapan pembangunan yang akan dilakukan dalam membangun *data warehouse* dan pembuatan OLAP seperti tabel 2, pada tabel tersebut database yang berada pada platform berbeda (basis data *heterogen*) akan disatukan dengan platform yang sama (basis data *homogen*).

Tabel 2. Tahap-tahap pembangunan *data warehouse*

| Proses | Sumber dan Tujuan | Tahapan Dalam Proses |
|--|--|--|
| Data Sumber ke <i>Staging Area</i> | Basis data <i>heterogen</i> | Membuat <i>package</i> baru |
| | | menentukan <i>Control Flow Item</i> yang akan digunakan |
| | | menentukan sumber metadata |
| | | Menentukan <i>Staging Area</i> untuk metadata |
| | | Membersihkan, menggabungkan dan me-load tabel-tabel pada <i>staging area</i> |
| | | <i>Execute Package</i> |
| | | <i>Load data ke staging area</i> |
| <i>Staging Area</i> ke <i>Data warehouse</i> | Basis data <i>homogen</i> | Membuat <i>package</i> baru |
| | | menentukan <i>Control Flow Item</i> yang akan digunakan |
| | | Menentukan sumber metadata |
| | | Menentukan metadata untuk <i>data warehouse</i> |
| | | Me-load data ke tabel dimensi dan tabel fakta |
| | | <i>Execute Package</i> |
| | | <i>Load data ke data warehouse</i> |
| Pembuatan <i>Cube</i> | <i>Data warehouse</i> ke <i>Analysis Service</i> | Menentukan <i>data source</i> |
| | | Menentukan <i>data source views</i> |
| | | Membuat <i>cube</i> |
| | | Mengubah atribut, measure dan hirarki pada <i>cube</i> |
| | | Mendefinisikan kalkulasi pada <i>cube</i> |
| | | <i>Deploy Analysis Services database</i> |

Langkah-langkah untuk melakukan proses ETL dalam penelitian ini pada dasarnya terbagi menjadi dua bagian yaitu :

1) Control Flow ETL Bagian I

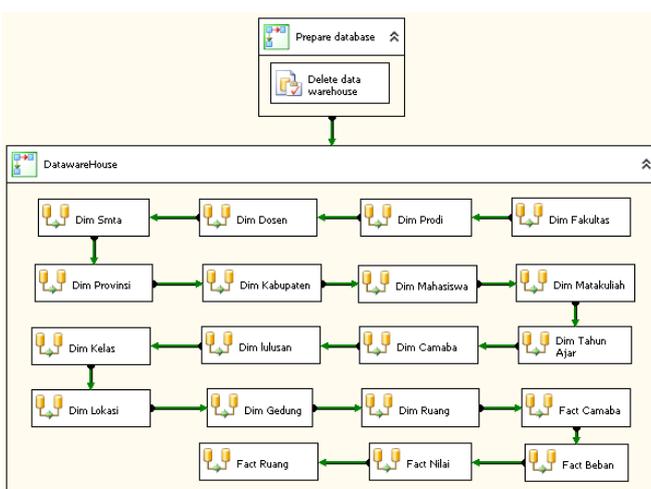


Gambar 8. Desain Control Flow ETL Bagian I

Pada gambar 8 merupakan desain control flow ETL yang pertama. Pada container “Prepare Database” merupakan proses penghapusan isi data staging agar siap diisi data baru.

Kemudian pada container “Data Staging” merupakan tempat dimana proses ETL yang pertama dilakukan. Dalam proses ETL data sumber yang berasal dari SPMB, SIAKAD dan SIGEJE akan di ekstrak kemudian dilakukan transformasi untuk mengisi nilai yang kosong dan mengganti tipe data agar sesuai dengan format rancangan data warehouse. Setelah dilakukan transformasi data, data tersebut disimpan dalam data staging sebelum diolah menjadi data warehouse.

2) Control Flow ETL Bagian II



Gambar 9. Desain Control Flow ETL Bagian II

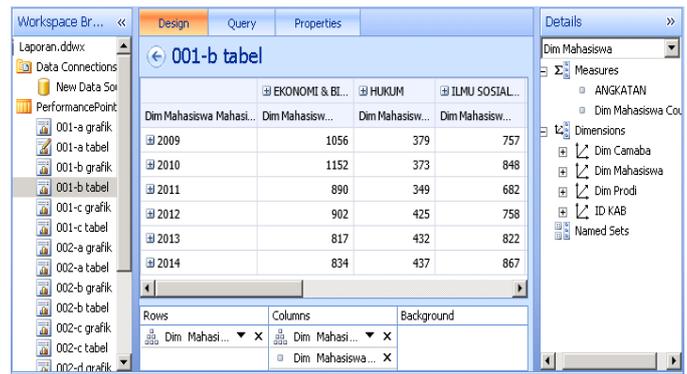
Pada gambar 9 merupakan desain control flow ETL yang kedua. Pada container “Prepare Database” merupakan proses penghapusan isi data warehouse agar siap diisi data baru.

Kemudian pada container “Data warehouse” akan dilakukan proses ETL dari data staging ke data warehouse. Data-data yang terdapat dalam data staging akan di ekstrak lalu

dilakukan transformasikan sesuai dengan kebutuhan seperti pada tabel 1, kemudian data tersebut disimpan dalam data warehouse.

4.2. Dashboard OLAP

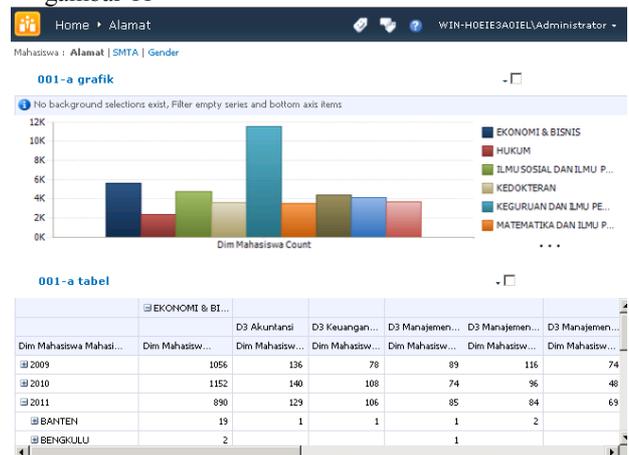
Setelah pembuatan cube OLAP selesai dilakukan, untuk menampilkan OLAP agar sesuai dengan kebutuhan yang diinginkan dapat menggunakan sharepoint. Pada gambar 10 merupakan contoh pembuatan laporan menggunakan sharepoint.



Gambar 10. Penggunaan Sharepoint

Berikut ini merupakan hasil dari dashboard laporan sesuai dengan fungsi sistem :

- a. Melakukan drill up dan drill down untuk membantu merekap data jumlah mahasiswa dari dimensi fakultas, prodi, tahun masuk, provinsi dan kabupaten seperti pada gambar 11



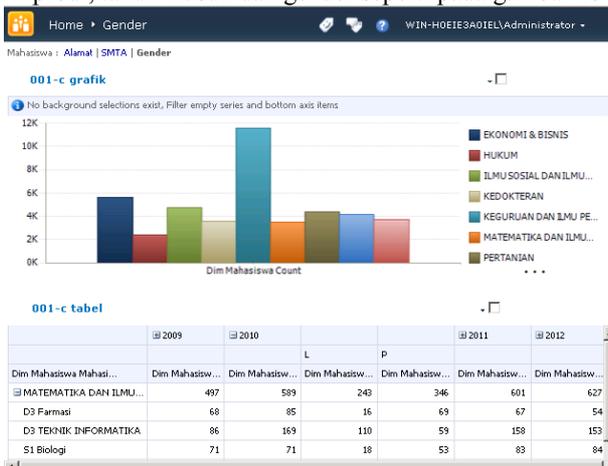
Gambar 11. Dashboard Laporan Alamat Mahasiswa

- b. Melakukan drill up dan drill down untuk membantu merekap data jumlah mahasiswa dari dimensi fakultas, prodi, tahun masuk dan asal SMTA seperti pada gambar 12



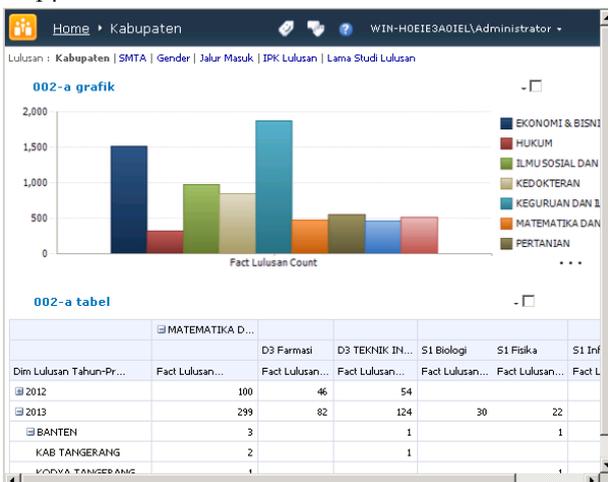
Gambar 12. Dashboard Laporan asal SMTA Mahasiswa

- c. Melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah mahasiswa dari dimensi fakultas, prodi, tahun masuk dan gender seperti pada gambar 13



Gambar 13. Dashboard Laporan Gender Mahasiswa

- d. Melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah lulusan dari dimensi fakultas, prodi, tahun wisuda, provinsi dan kabupaten seperti pada gambar 14



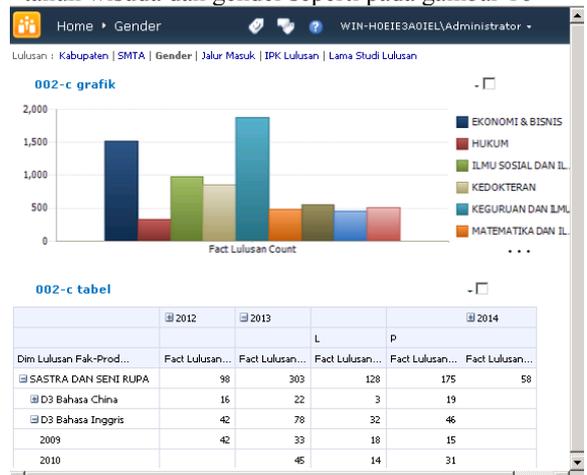
Gambar 14. Dashboard Laporan Kabupaten Lulusan

- e. Melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah lulusan dari dimensi fakultas, prodi, tahun wisuda dan asal SMTA seperti pada gambar 15



Gambar 15. Dashboard Laporan asal SMTA Lulusan

- f. Melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah lulusan dari dimensi fakultas, prodi, tahun wisuda dan gender seperti pada gambar 16



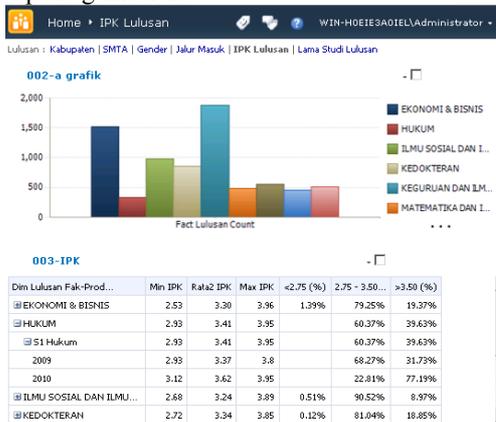
Gambar 16. Dashboard Laporan Gender Lulusan

- g. Melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah lulusan dari dimensi fakultas, prodi, tahun wisuda dan jalur masuk seperti pada gambar 17



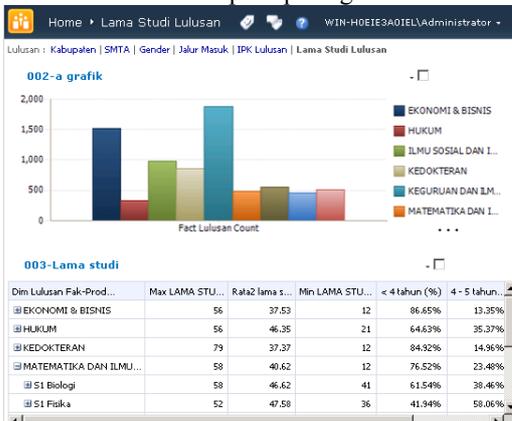
Gambar 17. Dashboard Laporan Jalur Masuk Lulusan

h. Melakukan *drill up* dan *drill down* untuk membantu merekap profil IPK lulusan yang terdiri dari nilai ipk minimal, ipk maksimal, rata-rata IPK dan persentase IPK yang dilihat dari dimensi fakultas, prodi dan tahun wisuda seperti pada gambar 18



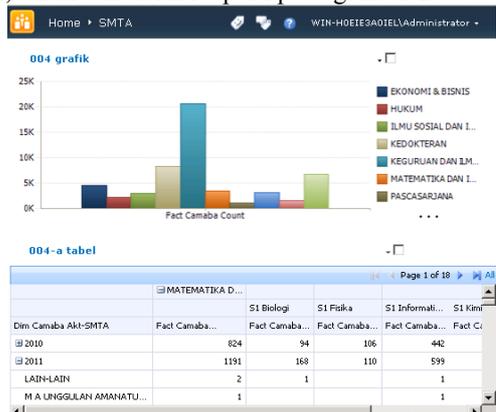
Gambar 18. Dashboard Laporan IPK Lulusan

i. Melakukan *drill up* dan *drill down* untuk membantu merekap profil lama studi lulusan yang terdiri dari lama studi terlama, lama studi tercepat, rata-rata lama studi dan persentase lama studi yang dilihat dari dimensi fakultas, prodi dan tahun wisuda seperti pada gambar 19



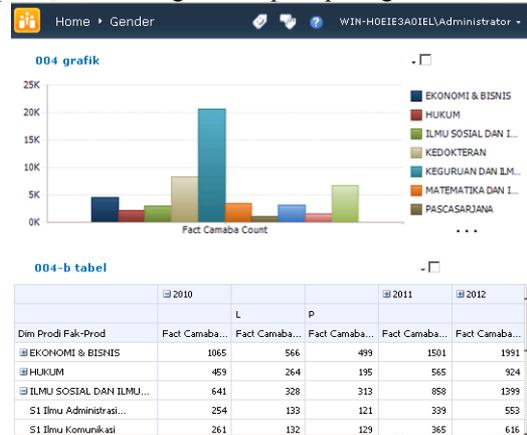
Gambar 19. Dashboard Laporan Lama Studi Lulusan

j. Melakukan *drill up* dan *drill down* untuk membantu merekap data calon mahasiswa baru dari dimensi fakultas, prodi, tahun dan SMTA seperti pada gambar 20



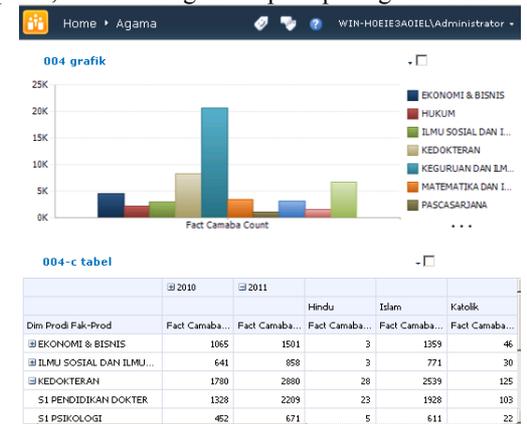
Gambar 20. Dashboard Laporan asal SMTA Camaba

k. Melakukan *drill up* dan *drill down* untuk membantu merekap data calon mahasiswa baru dari dimensi fakultas, prodi, tahun dan gender seperti pada gambar 21



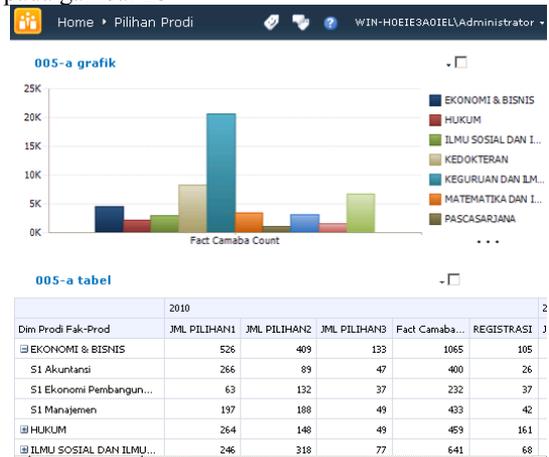
Gambar 21. Dashboard Laporan Gender Camaba

l. Melakukan *drill up* dan *drill down* untuk membantu merekap data calon mahasiswa baru dari dimensi fakultas, prodi, tahun dan agama seperti pada gambar 22



Gambar 22. Dashboard Laporan Agama Camaba

m. Melakukan *drill up* dan *drill down* untuk membantu merekap data calon mahasiswa baru dari dimensi fakultas, prodi, tahun, jumlah pilihan 1, jumlah pilihan 2, jumlah pilihan 3, total camaba yang melakukan registrasi seperti pada gambar 23



Gambar 23. Dashboard Laporan Pilihan Camaba

- n. Melakukan *drill up* dan *drill down* untuk membantu merekap data beban sks dosen dari dimensi fakultas, prodi, dosen dan tahun ajaran seperti pada gambar 24



Gambar 24. Dashboard Laporan Bebas SKS Dosen

- o. Melakukan *drill up* dan *drill down* untuk membantu merekap data pemakaian gedung dari dimensi fakultas, gedung, tahun ajaran dan semester seperti gambar 25



Gambar 11. Dashboard Laporan Pemakaian Gedung

4.3. K-means Clustering

K-means clustering digunakan untuk mengelompokkan data-data alumni UNS sesuai dengan attribute yang digunakan. Dalam penggunaan K-means ini attribute yang digunakan adalah lama studi dan nilai IPK setiap alumni UNS.

Dalam penelitian ini, data yang digunakan untuk melakukan clustering adalah data alumni jurusan S1 Informatika Fakultas MIPA dari tahun lulus 2013 sampai 2014 dengan jumlah cluster sebanyak 3 buah.

4.3.1. Perhitungan algoritma K-means

Berikut ini adalah perhitungan K-means menggunakan sample data pada titik :

Tabel 3. Sample data k-means clustering

| NIM_LULUSAN | LAMA_STUDI | IPK |
|-------------|------------|------|
| M0509006 | 4.5 | 3.22 |

1. Langkah pertama dalam proses perhitungan ini adalah menentukan banyaknya "k" sebanyak jumlah cluster yang diinginkan yaitu 3 buah.

2. Langkah kedua adalah menentukan centroid setiap cluster.
- Pusat cluster 1 (4.48 ; 3.17)
 - Pusat cluster 2 (4.29 ; 3.43)
 - Pusat cluster 3 (4.10 ; 3.69)
3. Langkah ketiga adalah menghitung jarak obyek ke pusat cluster (centroid).

$$P1(x_i, \mu_j) = \sqrt{(4.5 - 4.48)^2 + (3.22 - 3.17)^2}$$

$$= \sqrt{0.0004 + 0.0025}$$

$$= 0.053852$$

Tabel 4. Hasil perhitungan jarak obyek ke centroid

| Pusat | Lama Studi | IPK | Jarak |
|-------|------------|------|----------|
| P1 | 4.48 | 3.17 | 0.053852 |
| P2 | 4.29 | 3.43 | 0.296985 |
| P3 | 4.10 | 3.69 | 0.617171 |

4. Kemudian data dikelompokkan berdasarkan jarak minimum obyek ke pusat cluster. Sesuai dengan hasil perhitungan jarak obyek ke pusat cluster diketahui bahwa jarak terdekat data dengan pusat cluster 1 (P1) sehingga sample data masuk kedalam kelompok cluster 1. Karena jarak $P1 < P3 < P2$.
5. Selanjutnya dilakukan pengecekan kelompok data sample terhadap kelompok data sebelumnya, apabila data mengalami perubahan cluster maka nilai centroid akan diperbarui dimana nilai centroid yang baru diperoleh dari rata-rata kelompok cluster yang sama. Kemudian kembali ke langkah nomor 3.

$$P1_{Lama\ studi} = \frac{Jumlah\ lama\ studi\ di\ centroid\ 1}{Jumlah\ data\ pada\ centroid\ 1}$$

$$= \frac{4.5 + 4.58 + 4.58 + 4.58}{4}$$

$$= \frac{18.24}{4}$$

$$= 4.56$$

$$P1_{IPK} = \frac{Jumlah\ IPK\ pada\ centroid\ 1}{Jumlah\ data\ pada\ centroid\ 1}$$

$$= \frac{3.22 + 3.06 + 3.05 + 3.36}{4}$$

$$= \frac{12.69}{4}$$

$$= 3.173$$

Tabel 5. Nilai centroid lama dan centroid baru

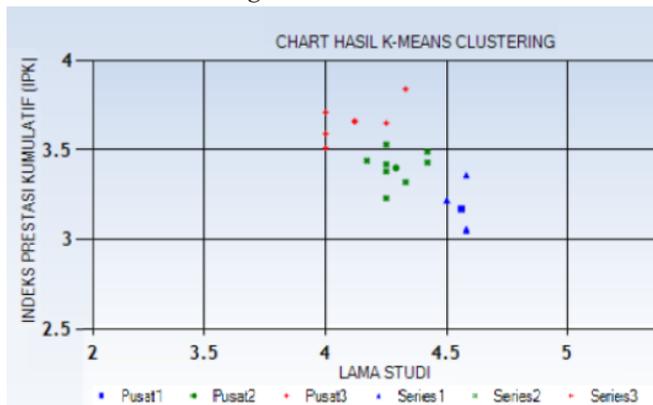
| Pusat | Lama | Baru |
|-------|---------------|----------------|
| P1 | (4.48 ; 3.17) | (4.56 ; 3.173) |
| P2 | (4.29 ; 3.43) | (4.29 ; 3.405) |
| P3 | (4.10 ; 3.69) | (4.11 ; 3.660) |

6. Jika sudah tidak ada lagi data yang berpindah kelompok pada masing-masing cluster maka proses dinyatakan selesai.

4.3.2. Hasil algoritma K-means

Hasil pengelompokan ini akan ditampilkan dalam bentuk *chart* dan table dimensi yang terdapat dalam table database *dim_lulusan* dan *dim_mahasiswa* seperti :

a. *Chart* hasil *clustering*



Gambar 25. *Chart* Hasil *Clustering*

b. Nilai IPK (<2.75, 2.75-3.50, dan >3.50).

| Dimensi IPK | | | |
|-------------|-----------|-----------|-----------|
| IPK | kelompok1 | kelompok2 | kelompok3 |
| 2.00 - 2.75 | 0 | 0 | 0 |
| 2.76 - 3.50 | 4 | 7 | 0 |
| 3.51 - 4.00 | 0 | 1 | 5 |

Gambar 26. Hasil *Clustering* Dimensi IPK

c. Lama Studi (< 4 tahun, 4-5 tahun dan, > 5 tahun)

| Dimensi Lama Studi | | | |
|--------------------|-----------|-----------|-----------|
| Masa Studi | kelompok1 | kelompok2 | kelompok3 |
| < 4 tahun | 4 | 8 | 5 |
| 4 - 5 tahun | 0 | 0 | 0 |
| > 5 tahun | 0 | 0 | 0 |

Gambar 27. Hasil *Clustering* Dimensi Lama Studi

d. Gender

| Dimensi GENDER | | | | |
|----------------|-----------|-----------|-----------|--------|
| GENDER | kelompok1 | kelompok2 | kelompok3 | Jumlah |
| P | 2 | 4 | 2 | 8 |
| L | 2 | 4 | 3 | 9 |

Gambar 28. Hasil *Clustering* Dimensi Gender

e. Asal SMTA

| Dimensi SMTA | | | | |
|---------------------------------------|-----------|-----------|-----------|--------|
| Nama SMTA | kelompok1 | kelompok2 | kelompok3 | Jumlah |
| SMAN 2 KOTA MANNA KAB BENGKULU SELATA | 0 | 1 | 0 | 1 |
| SMA NEG 2 BENGKULU | 0 | 1 | 0 | 1 |
| SMA NEG 2 CIREBON | 0 | 1 | 0 | 1 |
| MAS COKROAMINOTO KARANGKOBAR KAB BANJ | 0 | 0 | 1 | 1 |
| SMA NEG 1 PURWOREJO | 0 | 1 | 0 | 1 |
| SMAN 1 GEMOLONG KAB SRAGEN | 0 | 1 | 0 | 1 |
| SMAN 1 KOTA SURAKARTA | 0 | 0 | 1 | 1 |
| SMA NEG 1 SURAKARTA | 0 | 1 | 1 | 2 |
| SMA NEG 3 SURAKARTA | 2 | 1 | 1 | 4 |
| SMA WARGA, SURAKARTA | 1 | 0 | 0 | 1 |
| SMA NEG 1 TEGAL | 0 | 0 | 1 | 1 |
| SMA NEG 1 YOGYAKARTA | 0 | 1 | 0 | 1 |
| SMA NEG 1 PONOROGO | 1 | 0 | 0 | 1 |

Gambar 29. Hasil *Clustering* Dimensi SMTA

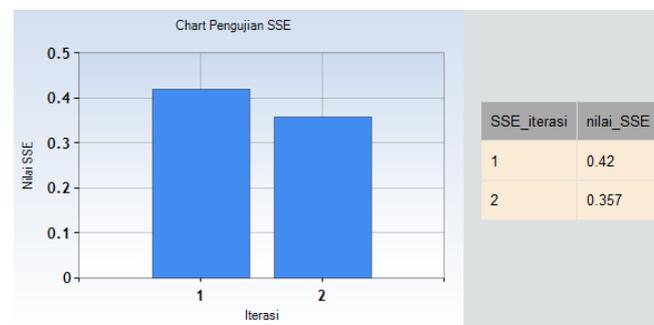
f. Provinsi

| Dimensi Provinsi | | | | |
|------------------|-----------|-----------|-----------|--------|
| Nama Provinsi | kelompok1 | kelompok2 | kelompok3 | Jumlah |
| BENGKULU | 0 | 1 | 0 | 1 |
| JAWA TENGAH | 3 | 7 | 5 | 15 |
| JAWA TIMUR | 1 | 0 | 0 | 1 |

Gambar 30. Hasil *Clustering* Dimensi Provinsi

4.4. Pengujian K-means Clustering

Pada gambar 18 merupakan *chart* dan tabel hasil pengujian K-means *clustering* menggunakan metode SSE (*Sum Square Error*).



Gambar 31. Hasil Pengujian K-means menggunakan SSE

Dari gambar 31 diketahui bahwa pengujian memiliki hasil *clustering* yang baik karena nilai SSE selalu berkurang setiap kali iterasi dilakukan.

5. KESIMPULAN DAN SARAN

Pada penelitian ini berhasil mengembangkan sistem OLAP dan *clustering* untuk memenuhi fungsionalitas sistem yaitu sistem dapat melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah mahasiswa, jumlah calon mahasiswa baru, jumlah lulusan, mencari nilai maksimum, minimum, rata-rata dan presentase profil ipk lulusan, mencari nilai maksimum, minimum, rata-rata dan presentase profil lama studi lulusan, beban sks dosen, presentase pemakaian ruang dari berbagai dimensi. Pada pengujian SSE *clustering* iterasi 1 adalah 0.398 sedangkan pada iterasi 2 nilai SSE *clustering* adalah 0.357. Penurunan nilai SSE ini menunjukkan bahwa jumlah selisih jarak setiap data ke pusat *cluster* semakin sedikit yang berarti semakin bagus. Hasil dari pengelompokkan ini, setiap data lulusan akan dikelompokkan berdasarkan kualitas lulusan yang dinilai berdasarkan IPK dan lama studi.

Adapun saran yang dipertimbangkan untuk pengembangan penelitian ini adalah mengembangkan proses ETL sehingga data dapat diperbarui secara otomatis jika terdapat perubahan pada data sumber. Kemudian menggabungkan algoritma k-means dengan algoritma pengambil keputusan seperti algoritma genetika agar sistem dapat melakukan pengambilan keputusan.

6. DAFTAR PUSTAKA

1. Kavitha, P., 2013. A Survey of *Data warehouse* and OLAP Technology. *International Journal of Latest Trends in Engineering and Technology*, 3(1), pp.387-90.
2. Janus, P. & Fouché, G., 2010. *Pro SQL Server 2008 Analysis Services*. United States, United States of America: Paul Manning.
3. Usman, M., Asghar, S. & Fong, S., 2009. A Conceptual Model for Combining Enhanced OLAP and Data Mining Systems. In *INC, IMS and IDC. Fifth International Joint Conference on*. Seoul, 2009.
4. Vipin Kumar, H.C.D.P., 2013. K-Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(4), pp.1-4.
5. Kumar, Y. & G., S., 2014. A New Initialization Method to Originate Initial Cluster Centers for K-Means Algorithm. *International Journal of Advanced Science and Technology*, 62, pp.43-54.
6. Lane, P., 2012. *Oracle OLAP Application Developer's Guide 10g Release 2 (10.2)*. California: Oracle Corporation.
7. Ponniah, P., 2010. *Data Warehousing Fundamentals For IT Professionals*. 2nd ed. New Jersey, Canada: John Wiley & Sons, Inc.
8. Prasetyo, M.A..S.A.&S., 2010. *Pembuatan Aplikasi OLAP Untuk Pelaporan pada PT. Aneka Tuna Indonesia Menggunakan SQL Server 2005*. [Online] Available at: <http://digilib.its.ac.id/public/ITS-Undergraduate-9803-Paper.pdf> [Accessed 22 February 2014].
9. Adithama, S.P., Wisnubhadra, I. & Sinaga, B.L., 2013. Analisis Dan Desain Real-Time Business Intelligence Untuk Subjek Kegiatan Akademik Pada Universitas Menggunakan Change Data Capture. *Seminar Nasional Teknologi Informasi dan Komunikasi*, pp.87-95.
10. Alfina, T., Santosa, B. & Barakbah, A.R., 2012. Analisis Perbandingan Metode Hierarchical Clustering, K-means dan Gabungan Keduanya dalam Cluster Data (Studi kasus : Problem Kerja Praktek Jurusan Teknik Industri ITS). *JURNAL TEKNIK ITS*, 1, pp.521-25.
11. Noor, M.H. & Hariadi, M., 2009. Image Cluster Berdasarkan Warna untuk Identifikasi Kematangan Buah Tomat dengan Metode Valley Tracing. *Seminar Nasional Informatika*, pp.15-24.
12. Sarwono, Y.T., 2010. Aplikasi Model Jaringan Syaraf Tiruan Dengan Radial Basis Function Untuk Mendeteksi Kelainan Otak (Stroke Infark). *Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya*.
13. Andayani, S., 2007. Pembentukan cluster dalam Knowledge Discovery in Database dengan Algoritma K-Means. *SEMNAS Matematika dan Pendidikan Matematika*.
14. Sari, I.P., 2013. Perancangan Sistem Pengenalan Wajah Manusia Menggunakan Web Camera dengan Metode Summary Squared Error (SSE). *Repository Universitas Andalas*.

Digital Watermarking Image dengan Menggunakan Discrete Wavelet Transform dan Singular Value Decomposition (DWT-SVD) untuk Copyright Labeling

Nur Anisah

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta 57126
nnur.anisah@gmail.com

Bambang Harjito

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta 57126
bambang_harjito@staff.uns.ac.id

Esti Suryani

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta 57126
suryapalapa@yahoo.com

ABSTRAK

Penyebaran konten digital yang begitu mudah membuat pemberian identitas pada konten digital menjadi sangat penting. Konten digital yang tidak beridentitas membuatnya mudah diklaim oleh pihak yang tidak bertanggung jawab. Watermarking merupakan salah satu cara pemberian watermark yang mudah dan tanpa merusak konten yang disisipi.

Teknik watermarking pada image digital dilakukan dengan mentransformasikan image dengan transformasi domain frekuensi Discrete Wavelet Transform (DWT). Pengolahan image input seperti cover medium dan watermark sebelumnya diperlukan agar cover medium dapat disisipi. Pada makalah ini, DWT dipadukan dengan faktorisasi matrik Singular Value Decomposition (SVD) untuk Copyright Labeling yang bertujuan agar hasil watermarking lebih robust.

Hasil penelitian menunjukkan bahwa teknik watermarking dengan DWT-SVD mampu menghasilkan watermark yang tahan terhadap beberapa serangan seperti Gaussian Noise, Salt & Pepper Noise, dan Poisson Noise tetapi tidak tahan terhadap serangan bluring dan geometrical attacks (rotation dan cropping).

Kata Kunci : Image Digital, Discrete Wavelet Transform, Singular Value Decomposition, Watermarking

1. Pendahuluan

Kemudahan penyebaran konten digital di dunia maya membuat konten rawan terhadap pembajakan. Konten digital seperti digital image memerlukan suatu identitas yang melekat pada konten agar tidak mudah diklaim oleh pihak lain yang tidak bertanggung jawab. Identitas yang melekat pada kontendigital image disebut dengan watermark image

Copyright labeling adalah salah satu aplikasi dari watermarking digital [1]. Watermarking dilakukan dengan cara menyisipkan watermark atau informasi ke dalam cover medium yang menghasilkan media yang terwatermark. Watermark dapat berfungsi sebagai bukti otentikasi kepemilikan konten. Watermarking dapat dilakukan berbagai media seperti media digital.

Digital Watermarking adalah proses penyisipan informasi ke media digital yang tak terlihat oleh mata manusia tetapi dapat terdeteksi oleh komputer [2]. Cover medium dan watermark dapat berupa teks, image digital, audio, dan video. Sedangkan watermarking pada image digital disebut Digital Watermarking Image.

Digital Watermarking Image bekerja pada dua domain pengolahan image digital, yaitu domain spasial dan domain frekuensi [3]. Transformasi domain spasial bekerja dengan

mengubah nilai piksel pada image digital sedangkan transformasi domain frekuensi menyisipkan watermark pada koefisien frekuensi hasil transformasi image asli. Teknik ini lebih rumit dari teknik yang berkerja pada domain spasial, tetapi memiliki ketahanan yang lebih baik. Beberapa teknik yang memanfaatkan domain frekuensi adalah Fast Fourier Transform (FFT), Discrete Fourier Transform (DFT), Discrete Cosinus Transform (DCT), dan Discrete Wavelet Transform (DWT).

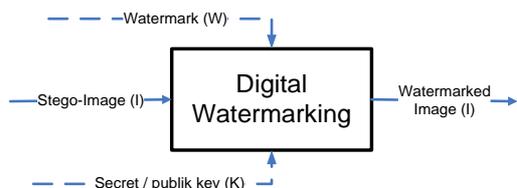
Beberapa jurnal terkait watermarking dengan DWT-SVD antara lain penelitian yang dilakukan oleh Dean Fatony Alfatawa [4]. Dean mencoba menerapkan Discrete Wavelet Transform pada image hitam putih (binary image). Sayangnya, image terwatermark hanya tahan terhadap beberapa serangan seperti perubahan format file image dari PNG ke BMP. Anumol dan Karthigaikumar [5] menerapkan Discrete Wavelet Transform pada image RGB yang dikonversi ke HSV (Hue, Saturation, and Value) terlebih dahulu. Hasil penelitian menunjukkan bahwa watermark yang disisipkan bisa diekstrak kembali. Terakhir penelitian yang dilakukan oleh Praful Saxena dan rekan-rekan [6]. Praful menggabungkan teknik transformasi DWT dan Singular Value Decomposition untuk penyisipan watermark pada image grayscale. Pada penelitian ini watermark disisipkan pada subband HH. Image terwatermark dapat diekstraksi dengan baik setelah diberi serangan-serangan tertentu.

Mengacu pada beberapa jurnal di atas, penelitian ini menggunakan Image Digital dengan model warna RGB yang dikonversi ke grayscale kemudian baru ditransformasikan dengan Discrete Wavelet Transform (DWT) dan faktorisasi matrik pada subband yang akan disisipi watermark dengan Singular Value Decomposition (SVD).

2. Teknik Digital Watermarking Image

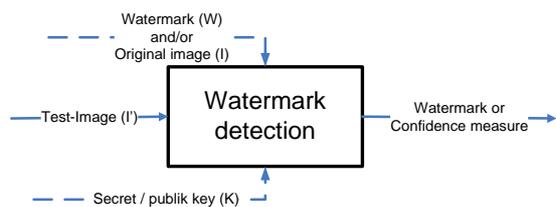
Watermarking merupakan proses pemberian tanda air yang berupa data ke dalam suatu media [1]. Watermark bisa disisipkan di berbagai media seperti media digital. Penyisipan informasi yang dilakukan di media image digital disebut Image Digital Watermarking [2].

Teknik watermarking secara umum dibagi menjadi dua tahap, yaitu: penyisipan watermark (embedding atau encoding) dan recovery watermark (ekstraksi atau decoding). Pemberian kunci bertujuan agar watermark hanya dapat diekstraksi oleh pihak yang sah [2]. Berikut proses encoding watermark dan pemberian kunci pada image digital:



Gambar 1 Proses *encoding image* digital (M. Kutter dan F. A. P. Petitcolas, 2000)

Encoding merupakan proses penyisipan informasi ke dalam *image digital* sedangkan *decoding* bertujuan untuk mengungkap kembali *watermark* yang ditanam dalam *image digital*. Proses *decoding* perlu adanya verifikasi *watermark* untuk membuktikan status kepemilikan *image digital*. Verifikasi *watermark* dilakukan dengan cara mencocokkan kunci yang diinputkan dengan kunci yang ada pada *image*. Berikut proses ekstraksi *image watermark*:



Gambar 2 Proses *decoding Image* Digital (M. Kutter dan F. A. P. Petitcolas, 2000)

Kedua proses tersebut saling berhubungan, untuk selanjutnya proses *encoding* disebut sebagai proses penyisipan dan proses *decoding* disebut sebagai proses ekstraksi.

3. Discrete Wavelet Transform dalam Image Digital

Dalam pengolahan *image digital*, diperlukan proses transformasi *image* agar informasi yang tertentu yang terkandung dalam *image* dapat diekstrak [7]. Transformasi *image* dibagi menjadi dua domain, yaitu domain spasial dan domain frekuensi. Domain spasial berkerja dengan melibatkan piksel-piksel pada *image digital*, sedangkan domain frekuensi dilakukan transformasi untuk mengetahui koefisien pada *image digital* tersebut sehingga proses penyisipan dapat dilakukan. Penelitian ini menggunakan transformasi dalam domain frekuensi yaitu Wavelet yang diterapkan pada *image* 2 dimensi sebagai metode transformasi *image* digital.

Transformasi Wavelet digunakan untuk menganalisa suatu bentuk sinyal (gelombang) yang merupakan kombinasi dari waktu (skala) dan frekuensi. Wavelet memiliki dua fungsi utama yaitu penskalaan dan *filtering*.

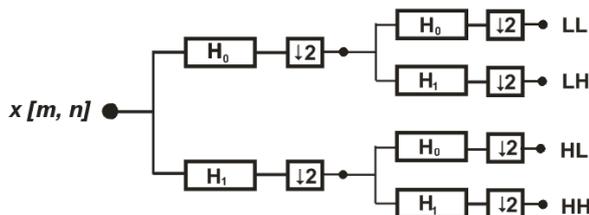
Proses *filtering* pada *image digital* bertujuan untuk mendapatkan informasi tertentu dari sinyal *image*. *Filter bank* mentransformasikan suatu sinyal *image* yang menghasilkan data berupa koefisien yang biasa disebut tapis *low pass* dan *high pass* yang kemudian dapat menentukan subimage dari *image* tersebut. Sedangkan fungsi penskalaan menghasilkan *Mother Wavelet*, Berikut persamaan dari fungsi penskalaan ϕ :

Wavelet paling tua yang lahir dari hasil penskalaan, dilasi, dan transformasi *Mother Wavelet* [7]. Dari *mother wavelet*, lahir Haar Wavelet dan kemudian disusul Wavelet lain seperti Daubechies. Beberapa transformasi Wavelet yang sering digunakan saat ini adalah *Discrete Wavelet Transform* dan *Continous Wavelet Transform*. DWT digunakan pada transformasi *image* digital, sedangkan CWT digunakan pada *image* yang bergerak, seperti video.

Konsep transformasi Wavelet pada *image* 2 dimensi sangat sederhana. *Image* ditransformasi bagi (dekomposisi)

terlebih dahulu menjadi 4 subimage yaitu LL, LH, HL, dan HH. Setiap *subimage* berukuran seperempat kali *image* asli. *Subimage* LL yang terletak di kiri atas merupakan *subimage* yang paling halus (*smooth*) dari *image* asli. Sedangkan *subimage* lain, seperti HL, LH, dan HH mengandung frekuensi tinggi dari *image* sehingga dapat disebut versi kasar dari *image* asli.

Proses transformasi dimulai dengan proses dekomposisi. Dekomposisi adalah proses perubahan menjadi bentuk yang lebih sederhana atau penguraian. Dalam transformasi domain frekuensi, dekomposisi berarti proses perubahan cira digital menjadi koefisien frekuensi. Karena *image* yang didekomposisikan 2 dimensi, proses dekomposisi dibagi menjadi dua tahap, yaitu dekomposisi baris dan dekomposisi kolom.



Gambar 3 Transformasi Wavelet 2D 1 level (Darma Putra, 2010)

Hasil dekomposisi *image* 1 level menghasilkan 4 koefisien, yaitu:

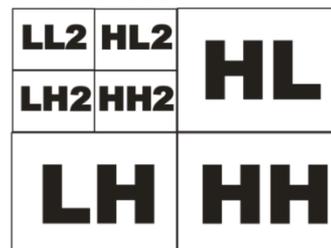
- LL diperoleh melalui proses tapis *Low pass* dilanjutkan dengan *Low pass*. *Image* dibagian ini mirip dengan *image* aslinya dan lebih halus.
- LH merupakan koefisien yang didapat dari proses tapis *Low pass* yang kemudian dilanjutkan dengan *High pass*.
- HL merupakan hasil dari proses tapis *High pass* kemudian dilanjutkan *Low pass*.
- HH adalah hasil dari proses tapis *High pass* kemudian *High pass* lagi.

Hasil transformasi wavelet digambarkan dalam skema berikut ini:



Gambar 4 Skema Transformasi wavelet 2D 1 level (Darma Putra, 2010)

Skema diatas menggambarkan transformasi wavelet 2 dimensi level 1. Sedangkan untuk level 2, koefisien *image* pada bagian LL akan didekomposisikan lagi menjadi empat bagian, yaitu LL2, LH2, HL2, dan HH2, begitu seterusnya pada level-level berikutnya. Berikut skema transformasi wavelet 2 level:



Gambar 5 Skema Transformasi wavelet 2D 2 level (Darma Putra, 2010)

4. Singular Value Decomposition (SVD)

SVD merupakan suatu pemfaktoran matrik dengan menguraikan suatu matrik ke dalam dua matrik P dan Q. Jika diketahui matrik A berukuran $m \times n$ dengan rank $r > 0$, maka dekomposisi dari matrik A dirumuskan sebagai berikut:

$$A = P\Delta Q^T$$

Keterangan:

P = matriks ortogonal berukuran $m \times r$

Q = matriks orthogonal berukuran $n \times r$

Δ = matriks diagonal berukuran $r \times r$

Terbentuknya matrik Δ tergantung kondisi matrik A, yaitu:

- a. Δ bila $r = m = n$
- b. $\begin{bmatrix} \Delta \\ (0) \end{bmatrix}$ bila $r = n$ dan $r < m$
- c. $\begin{bmatrix} \Delta & (0) \end{bmatrix}$ bila $r = m$ dan $r < n$
- d. $\begin{bmatrix} \Delta & (0) \\ (0) & (0) \end{bmatrix}$ bila $r < m$ dan $r < n$

Matrik P diperoleh melalui perkalian A, Q, dan Δ^{-1} dapat dinyatakan sebagai berikut:

$$P = A\Delta Q^{-1}$$

5. PSNR dan NC

Kualitas *image* yang disisipi *watermark* biasanya tidak berubah banyak. Perbedaan antara *image* ber*watermark* dengan *image* asli dan *watermark* asli dengan *watermark* hasil ekstraksi tidak dapat dilihat dengan mata telanjang. Untuk itu, diperlukan pengujian secara subjektif yaitu dengan menghitung nilai PSNR dan NC pada kedua *image* yang dibandingkan tersebut.

5.1 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya *noise* yang berpengaruh pada *image* tersebut [7]. PSNR digunakan untuk mengetahui perbandingan antara *image* asli sebelum dan sesudah disisipi *watermark*. Untuk mendapatkan nilai PSNR, ditentukan nilai MSE (*Mean Square Error*) terlebih dahulu. MSE adalah nilai error kuadrat rata-rata antara *image* asli dengan *image* ber*watermark*. Berikut cara menghitung nilai MSE:

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2$$

Keterangan:

MSE = Nilai *Mean Square Error*

M = panjang *image watermark* (dalam piksel)

N = lebar *image watermark* (dalam piksel)

$I(x,y)$ = nilai piksel dari *image* asli

$I'(x,y)$ = nilai piksel pada *image watermark*

Setelah diperoleh nilai MSE, nilai PSNR dapat dihitung dengan rumus sebagai berikut:

$$PSNR = 10 \cdot \log \left(\frac{MAX_i^2}{MSE} \right)$$

Keterangan:

MSE = nilai *Mean Square Error*

MAX_i = nilai maksimum dari piksel *image* yang digunakan.

Nilai MSE menunjukkan banyaknya error dari *image*. Semakin rendah nilai MSE maka akan semakin baik. Sedangkan nilai PSNR yang didapat dari nilai maksimum piksel dibagi nilai MSE, sehingga semakin besar nilai PSNR

yang dihasilkan, menunjukkan kualitas *image* semakin baik. *Image* 8-bit dikatakan berkualitas baik jika rentan nilai PSNR antara 30-50 dB.

5.2 Normalized Correlation Coefficient (NC)

NC atau nilai kolerasi dapat digunakan juga untuk mendeteksi kehadiran *watermark*. Rumus nilai kolerasi adalah sebagai berikut:

$$NC(W, W') = \frac{\sum W W'}{\sqrt{\sum W_i^2} \sqrt{\sum W'_i^2}}$$

Keterangan:

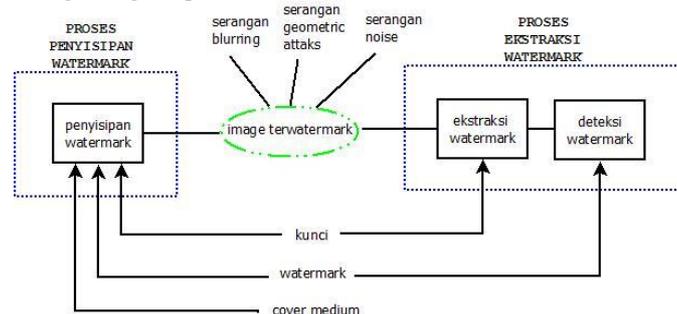
W = *watermark* asli

W' = *watermark* hasil ekstraksi

Nilai kolerasi menunjukkan perbandingan antara *watermark* asli dengan *watermark* hasil ekstraksi. Nilai kolerasi bernilai 0 (nol) berarti kedua *image* yang dibandingkan tidak memiliki kemiripan sama sekali, sedangkan kolerasi yang bernilai 1 (satu) menunjukkan kedua *image* yang dibandingkan sama persis.

6. Implementasi Sistem

Implementasi sistem merupakan inti proses dalam penelitian. Langkah - langkah implementasi sistem ditampilkan pada gambar berikut ini:



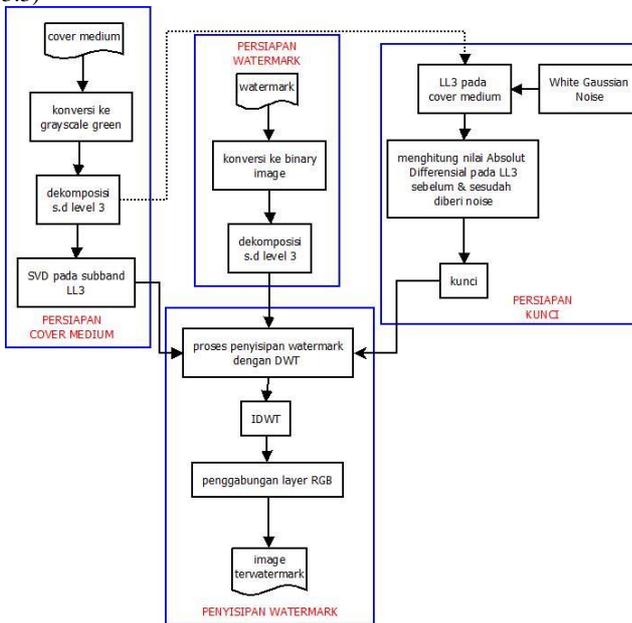
Gambar 6 Diagram alir implementasi

Proses implementasi sistem terdiri dari dua proses dasar, yaitu proses penyisipan *watermark* dan ekstraksi *watermark*. Penyisipan *watermark* merupakan proses penyisipan *image watermark* ke *image* asli sedangkan ekstraksi *watermark* adalah proses mengeluarkan *watermark* yang ada didalam *image* asli. Antara kedua proses tersebut terdapat proses lain seperti serangan *image terwatermark* dan deteksi *watermark*.

Serangan *image* dilakukan dengan memberikan serangan pada *image terwatermark* yang berupa *noise*, *geometrical attacks*, *blurring*, dan sebagainya. Serangan ini bertujuan untuk menguji ketahanan *watermark* yang ada di dalam *image*. Selain serangan *image terwatermark*, proses deteksi *watermark* juga diperlukan untuk menentukan apakah *watermark* berhasil diekstraksi dengan baik dan layak untuk ditampilkan. Penentuan keberhasilan ekstraksi dilakukan dengan membandingkan nilai kolerasi dan *threshold*. *Watermark* dengan nilai kolerasi dibawah nilai *threshold* bisa dikatakan tidak berhasil diekstraksi dengan baik.

6.1 Proses Penyisipan Watermark

Proses penyisipan watermark terdiri dari persiapan image asli dan watermark yang akan menjadi input perangkat lunak, dan proses penyisipan watermark. Proses penyisipan watermark digambarkan pada skema berikut ini: (lihat gambar 3.3)



Gambar 7 Diagram Proses Penyisipan Watermark

Image asli dengan model warna RGB perlu diolah terlebih dahulu agar bisa disisipi watermark. Proses pengolahan terdiri dari konversi dari image berwarna menjadi grayscale dan dekomposisi image. Konversi ke grayscale dilakukan dengan memisahkan image asli ke dalam tiga layer warna yaitu red, green, dan blue sehingga terbentuk tiga image grayscale dari masing-masing warna red, green, dan blue. Tetapi yang digunakan dalam penyisipan watermark hanya image grayscale layer green saja.

Proses selanjutnya adalah proses transformasi. Image asli yang ditransformasikan dengan Discrete Wavelet Transform dan Singular Value Decomposition kemudian disisipi watermark dan terbentuklah image terwatermark. Algoritma penyisipan watermark adalah sebagai berikut:

Algoritma 1

Input : cover medium, watermark, kunci

Output : image terwatermark

Proses :

1. Pertama konversi cover medium ke grayscale green dengan perintah: `image=image(:, :2);`
2. Dekomposisi image layer green dengan DWT sampai level 3. Dekomposisi level 1 menghasilkan 4 subband, yaitu: LL, HL, LH, HH dengan perintah: `[LL,LH,HL,HH] = dwt2(image,'haar');`
Dekomposisikan dengan DWT level 2, subband LL didekomposisikan menjadi subband LL2, HL2, LH2, HH2, HL, LH, HH dengan perintah `[LL2,LH2,HL2,HH2] = dwt2(LL,'haar');`
Kemudian pada level 3 subband LL2 didekomposisikan lagi hingga menghasilkan subband LL3 dengan perintah: `[LL3,LH3,HL3,HH3] = dwt2(LL2,'haar');`
3. Faktorisasi subband LL3 yang berupa matrik image dengan menggunakan SVD. Rumus SVD adalah $A = UDV^T$ dengan A = subband LL3

Keterangan:

A = matrik ukuran (m x n)

U = singular vektor dari matrik A (vektor ini orthonormal)

D = diagonal vektor yang menyimpan singular value dari koresponding singular vektornya

V^T = singular vector dari matrik A yang juga orthonormal (T berarti transpose)

Sehingga menjadi 3 matriks yaitu matriks U, D, dan V $svd(LL3) = [U, D, V];$

4. Pembuatan kunci dilakukan dengan mengambil subband LL3 kemudian dilakukan Add White Gaussian Noise dengan perintah:

`n = awgn(LL3,1,0,'linear');`

`n = LL3` setelah diberi White Gaussian Noise

5. Matrik yang telah diberi White Gaussian Noise kemudian dilakukan perhitungan perbandingan nilai absolute pada LL3 sebelum dan sesudah diberi White Gaussian Noise, dengan perintah:

`N = imabsdiff(n,LL3);`

N = nilai absolute differential

6. Watermark yang akan disisipi dikonversi ke binary image terlebih dahulu dengan perintah `watermark=rgb2gray(wmark);`
7. Watermark yang telah menjadi binary image kemudian didekomposisikan 3 level dengan DWT dengan perintah:

Dekomposisi level 1:

`[L_L,L_H,H_L,H_H]= dwt2(watermark,'haar');`

Dekomposisi level 2:

`[L_L2,L_H2,H_L2,H_H2]= dwt2(L_L,'haar');`

Dekomposisi level 3:

`[L_L3,L_H3,H_L3,H_H3]= dwt2(L_L2,'haar');`

8. Melakukan SVD pada matriks L_L3 watermark dengan cara yang sama dengan SVD pada matriks $LL3$ sebelumnya sehingga menghasilkan $svd(L_L3)=[U_wmark, D_wmark, V_wmark];$
9. Membuat matriks $LL3$ yang baru (setelah proses SVD) dengan cara mengganti matriks D dengan matriks D gabungan dari matriks D cover medium dan watermark seperti berikut:

`D_watermark = D+(0.01*D_wmark);`

`LL3 = U*D_watermark*transpose(V);`

10. Watermark yang telah berbentuk L_L3 disisipi di subband $LL2$ yang telah diolah dengan menggunakan rumus

$$f' = f + \alpha \cdot w(k) + K(k), k = 1, \dots, L$$

f' = image terwatermark

f = cover medium

w = watermark

α = nilai ambang yang menunjukkan tingkat kekuatan penyisipan watermark

K = kunci

$k = (m,n)$ posisi pada koefisien subband $LL2$

11. Proses IDWT 3 level sehingga menghasilkan image terwatermark dengan perintah

Untuk IDWT level 1:

`imagerwatermark_level1=idwt2(imagerwatermark, LH3, HL3, HH32, 'haar');`

Untuk IDWT level 2:

`Imagerwatermark_level2=idwt2(imagerwatermark_level1, LH2, HL2, HH2, 'haar');`

Untuk IDWT level 3:

`Imagerwatermark_final=idwt2(imagerwatermark_level2, LH, HL, HH, 'haar');`

12. Penggabungan image layer green dengan layer red dan blue dengan perintah:

`newColorImage(:, : ,1) =wimg;`

`newColorImage(:, : ,2) =wimg2;`

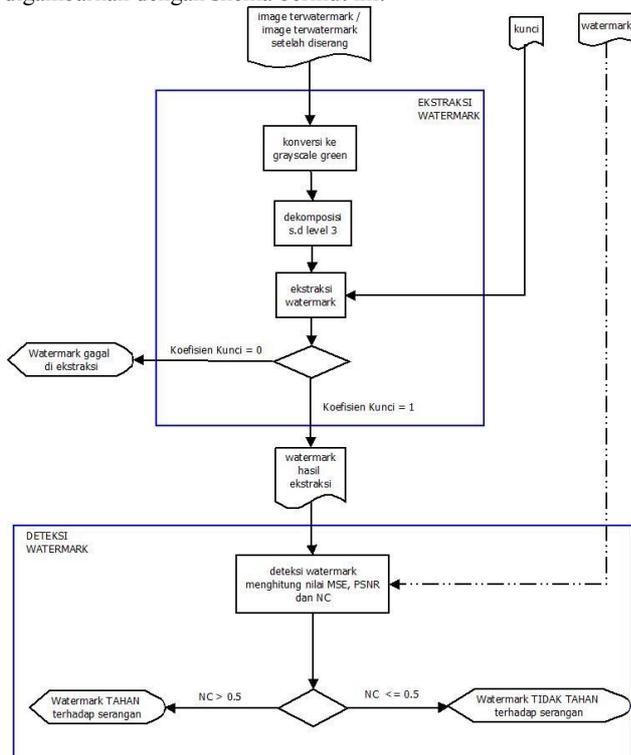
`newColorImage(:, : ,3) =wimg3;`

`newColorImage = image RGB hasil penggabungan`

13. Setelah proses penggabungan menghasilkan output image terwatermark

6.2 Proses Ekstraksi Watermark

Proses ekstraksi watermark terdiri dari pengolahan image terwatermark agar siap diekstraksi, proses ekstraksi itu sendiri dan deteksi watermark. Proses ekstraksi watermark digambarkan dengan skema berikut ini:



Gambar 8 Diagram Proses Ekstraksi Watermark

Image terwatermark tidak dapat langsung diekstraksi, diperlukan proses konversi image dari image berwarna (true color) ke grayscale green dan dekomposisi dengan Haar Wavelet terlebih dahulu. Algoritma ekstraksi watermark adalah sebagai berikut:

Algoritma 2

Input : image terwatermark (image terwatermark yang telah diserang), kunci

Output : watermark hasil ekstraksi

Proses :

- Konvensi image terwatermark ke grayscale green dengan perintah:
image=image(:, :2);
 - Dekomposisi image layer green dengan DWT sampai level 3. Dekomposisi level 1 menghasilkan subband LL, HL, LH, HH dengan perintah sebagai berikut:
[LL, LH, HL, HH] = dwt2(image, 'haar');
 - Dekomposisi level 2 dari subband LL menghasilkan subband LL2, HL2, LH2, HH2, HL, LH, HH dengan perintah
[LL2, LH2, HL2, HH2] = dwt2(LL, 'haar');
 - Kemudian pada level 3 subband LL2 didekomposisikan yang menghasilkan subband LL3 dengan perintah:
[LL3, LH3, HL3, HH3] = dwt2(LL2, 'haar');
 - Sebelum proses ekstraksi, dilakukan pencocokan kunci dengan mencocokkan nilai besaran scalar pada kunci yang dimasukkan dengan nilai besaran scalar pada cover medium.
 - Proses ekstraksi dilakukan dengan mengeluarkan watermark dari cover medium. Ekstraksi dilakukan dengan perintah:
recovered_image = ((LL2' - LL2) - k) / 0.001;
- Keterangan:
LL2' : image terwatermark

LL2 : cover medium

k : matriks kunci

- Proses deteksi watermark dengan melakukan perhitungan PSNR, MSE, dan NC untuk menentukan apakah watermark berhasil diekstraksi atau gagal diekstraksi. Jika watermark memiliki nilai NC ≥ 0.5 maka dikatakan watermark berhasil diekstraksi, sedangkan jika nilai NC < 0.5, watermark gagal diekstraksi.

$$NC(W, W') = \frac{\sum W(n)W'(n)}{\sqrt{\sum W^2(n)}\sqrt{\sum W'^2(n)}}$$

Keterangan:

w = watermark

w' = watermark hasil ekstraksi

- Membandingkan nilai kolerasi image watermark (NC) dengan nilai threshold, nilai threshold = 0.5.
- Jika NC < T, berarti watermark berhasil diekstraksi dengan baik, sedangkan jika nilai NC ≥ T, watermark tidak dapat diekstraksi dengan baik dan dianggap tidak layak untuk ditampilkan di monitor.

7. HASIL DAN PEMBAHASAN

Pengujian yang dilakukan dalam tiga macam. Pertama cover medium disisipi watermark dengan berbagai ukuran dan format file, pengujian ini dijadikan acuan untuk analisis pada pengujian yang lain. Pengujian kedua adalah menyisipkan watermark ke cover medium dengan berbagai level dekomposisi, sedangkan pengujian ketiga dilakukan dengan memberi berbagai serangan pada image terwatermark.

Penelitian ini dilakukan dengan cover medium lena.jpg yang berukuran (512 x 512 piksel) dengan beberapa watermark dalam berbagai ukuran dan format file, yaitu uns.png yang berukuran (400 x 400 piksel), dogs.jpg yang berukuran (640 x 480 piksel), dan pepper.bmp yang berukuran (512 x 512 piksel).

Pengujian dilakukan untuk mendapatkan nilai PSNR, MSE dan NC (Normalized Coefficient Correlation) untuk mengetahui kualitas image watermark hasil ekstraksi. Analisa dari ketiga pengujian dideskripsikan pada subbab-subbab berikut ini:

7.1 Analisis Hasil Ekstraksi Image dengan Berbagai Ukuran dan Format File Watermark

Pengujian dilakukan menggunakan cover medium lena.jpg dengan 3 macam watermark, yaitu uns400.png, dogs.jpg, dan pepper.bmp. Hasil pengujian ditampilkan pada tabel 1.

Tabel 1 Hasil Ekstraksi Image terwatermark

| Watermark Asli | Watermark setelah dikonversi ke Binary Image | Watermark Hasil Ekstraksi | PSNR (dB), MSE, dan NC |
|---|--|---|--|
|  uns400.png (400 x 400 piksel) |  |  | PSNR: 48.9434 MSE: 0.829346 NC: 0.889784 |
|  dogs.jpg (640x480 piksel) |  |  | PSNR: 54.152 MSE: 0.249967 NC: 0.959068 |

| | | | |
|---|---|---|---|
|  |  |  | PSNR: 51.4941 MSE: 0.460964 NC: 0.97296 |
| pepper.bmp (512x512 piksel) | | | |

Nilai-nilai yang ditunjukkan pada tabel 1 merupakan nilai dasar sebagai acuan pada pengujian-pengujian berikutnya. Nilai NC tertinggi diperoleh saat watermark pepper.bmp dengan format file .bmp. Watermark dengan tingkat kemiripan tertinggi dengan watermark asli dikarenakan ukuran file watermark yang sama dengan ukuran file cover medium, yaitu (512 x 512 piksel).

7.2 Analisis Hasil Ekstraksi Image dengan Berbagai Level Dekomposisi

Pengujian dilakukan pada file cover medium dengan disisipi 3 macam file watermark secara bergantian yang diuji pada beberapa level dekomposisi, yaitu dekomposisi level 1, level 2, dan level 3. Hasil Pengujian ditampilkan pada Tabel 2.

Tabel 2 Hasil Pengujian Watermark dengan Berbagai Level Dekomposisi

| Dekomposisi | File Watermark | Image Hasil Ekstraksi | PSNR (dB), MSE, dan NC |
|-------------|------------------------------|---|--|
| Level 1 | uns400.png (400 x400 piksel) |  | PSNR: 48.9385 MSE: 0.830297 NC: 0.977922 |
| Level 2 | uns400.png (400 x400 piksel) |  | PSNR: 48.9402 MSE: 0.829974 NC: 0.947448 |
| Level 3 | uns400.png (400 x400 piksel) |  | PSNR: 48.9434 MSE: 0.829346 NC: 0.889784 |
| Level 1 | dogs.jpg (640x480 piksel) |  | PSNR: 54.1472 MSE: 0.250244 NC: 0.991549 |
| Level 2 | dogs.jpg (640x480 piksel) |  | PSNR: 54.1488 MSE: 0.25015 NC: 0.975812 |
| Level 3 | dogs.jpg (640x480 piksel) |  | PSNR: 54.152 MSE: 0.249967 NC: 0.959068 |
| Level 1 | pepper.bmp (512x512 piksel) |  | PSNR: 51.4891 MSE: 0.461497 NC: 1 |
| Level 2 | pepper.bmp (512x512 piksel) |  | PSNR: 51.4909 MSE: 0.461312 NC: 1 |
| Level 3 | pepper.bmp (512x512 piksel) |  | PSNR: 51.4941 MSE: 0.460964 NC: 0.97296 |

Tabel 2 menunjukkan hasil pengujian watermark hasil ekstraksi pada beberapa level dekomposisi. Dari ketiga image pola perubahan nilai NC, MSE dan PSNR hampir sama. Nilai NC tertinggi didapat pada dekomposisi level 1, diikuti dengan nilai-nilai pada dekomposisi level 2 dan nilai NC terendah pada dekomposisi level 3. Image dengan level dekomposisi terendah memiliki tingkat kemiripan tertinggi dengan watermark asli.

Nilai PSNR tertinggi diperoleh pada level 3, semakin sedikit level dekomposisi yang dijalankan, semakin kecil nilai PSNR yang didapat. Sedangkan nilai MSE tertinggi didapat pada image level dekomposisi terendah, yaitu level 1. Nilai MSE menurun seiring dengan meningkatnya level dekomposisi. Hal ini menunjukkan image dengan level

dekomposisi tertinggi memiliki tingkat error terendah.

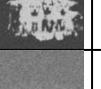
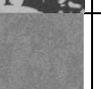
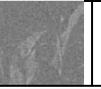
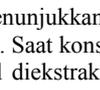
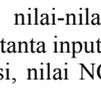
7.3 Analisis Hasil Ekstraksi Image dengan Berbagai Serangan

Pengujian selanjutnya dilakukan dengan memberi beberapa serangan terhadap image terwatermark secara bergantian dan menguji kualitas image watermark hasil ekstraksinya. Serangan yang dilakukan antara lain pemberian noise, geometrical attacks dan blurring.

7.3.1 Gaussian Noise

Gaussian Noise merupakan noise yang terdiri dari titik-titik putih dengan dua macam nilai input yaitu nilai rata-rata dan variasi. Semakin besar nilai rata-rata dan variasinya maka image yang terlihat semakin kabur begitu pula sebaliknya. Pengujian dilakukan dengan berbagai variasi nilai rata-rata (M) dengan nilai variasi (V) tetap, yaitu (1 x 10⁻⁷). Hasil pengujian ditampilkan pada tabel 3.

Tabel 3 Hasil Pengujian Serangan Gaussian Noise

| Watermark | Gaussian Noise V: (1 x 10 ⁻⁷) | Image terwatermark | Watermark Hasil Ekstraksi | PSNR (dB), MSE dan NC |
|------------------------------|---|---|---|---|
| uns400.png (400 x400 piksel) | M: (1 x 10 ⁻⁸) |  |  | PSNR: 48.9504 MSE: 0.828019 NC: 0.832774 |
| uns400.png (400 x400 piksel) | M: (1 x 10 ⁻⁵) |  |  | PSNR: 48.9504 MSE: 0.828025 NC: 0.0764321 |
| uns400.png (400 x400 piksel) | M: 0.1 |  |  | PSNR: 49.0274 MSE: 0.81347 NC: 0.0218132 |
| dogs.jpg (640x480 piksel) | M: (1 x 10 ⁻⁸) |  |  | PSNR: 54.1589 MSE: 0.249567 NC: 0.918677 |
| dogs.jpg (640x480 piksel) | M: (1 x 10 ⁻⁵) |  |  | PSNR: 54.1587 MSE: 0.249582 NC: 0.0993994 |
| dogs.jpg (640x480 piksel) | M: 0.01 |  |  | PSNR: 54.1651 MSE: 0.249214 NC: 0.0458374 |
| pepper.bmp (512x512 piksel) | M: (1 x 10 ⁻⁸) |  |  | PSNR: 51.5011 MSE: 0.460227 NC: 0.942206 |
| pepper.bmp (512x512 piksel) | M: (1 x 10 ⁻⁵) |  |  | PSNR: 51.501 MSE: 0.46024 NC: 0.117945 |
| pepper.bmp (512x512 piksel) | M: 0.1 |  |  | PSNR: 51.5662 MSE: 0.45338 NC: 0.0462066 |

Hasil pengujian menunjukkan nilai-nilai yang didapat memiliki pola yang sama. Saat konstanta input noise kecil (1 x 10⁻⁸) watermark berhasil diekstraksi, nilai NC yang didapat tinggi. Tetapi ketika konstanta noise yang diinputkan lebih besar (1 x 10⁻⁵), watermark gagal diekstraksi dengan nilai NC rendah, kemiripan antara watermark hasil ekstraksi dengan watermark asli rendah..

Nilai MSE yang didapat bervariasi, menurun seiring dengan konstanta noise yang diinputkan semakin kecil. Tetapi secara keseluruhan, MSE yang didapat bernilai rendah, sehingga image yang dikenai serangan memiliki tingkat error yang rendah.

Nilai PSNR yang didapat menunjukkan penambahan yang signifikan walau sangat kecil seiring dengan semakin kecil konstanta input noise yang dimasukkan. Semakin kecil nilai konstanta Gaussian Noise yang diinputkan, semakin besar nilai PSNR yang didapat.

7.3.2 Salt & Pepper Noise

Salt & Pepper merupakan noise yang terdiri titik-titik hitam dan titik-titik putih. Pada Matlab, salt & pepper diinputkan dengan nilai konstanta antara 0 – 1, semakin besar nilai konstanta yang diinputkan maka image akan semakin kabur. Pengujian dengan Salt & Pepper ini dilakukan menggunakan file cover medium lena.jpg dan 3 file watermark, yaitu uns400.png, dogs.jpg, dan pepper.bmp. Hasil pengujian ditampilkan pada tabel 4.

Tabel 4 Hasil Pengujian dengan Serangan Salt & Pepper Noise

| Watermark | Salt & Pepper Noise | Image terwater mark | Watermark Hasil Ekstraksi | PSNR (dB), MSE, dan NC |
|------------------------------|-------------------------|---------------------|---------------------------|---|
| uns400.png (400 x400 piksel) | (1 x 10 ⁻⁸) | | | PSNR: 48.9434 MSE: 0.829346 NC: 0.889784 |
| uns400.png (400 x400 piksel) | (1 x 10 ⁻⁵) | | | PSNR: 48.9435 MSE: 0.829333 NC: 0.12513 |
| uns400.png (400 x400 piksel) | (1 x 10 ⁻²) | | | PSNR: 49.0012 MSE: 0.818399 NC: 0.0098972 |
| dogs.jpg (640x480 piksel) | (1 x 10 ⁻⁸) | | | PSNR: 54.152 MSE: 0.249967 NC: 0.959068 |
| dogs.jpg (640x480 piksel) | (1 x 10 ⁻⁵) | | | PSNR: 54.1519 MSE: 0.249969 NC: 0.186604 |
| dogs.jpg (640x480 piksel) | 0.01 | | | PSNR: 54.1929 MSE: 0.247624 NC: 0.0302736 |
| pepper.bmp (512x512 piksel) | (1 x 10 ⁻⁸) | | | PSNR: 51.4941 MSE: 0.460964 NC: 0.97296 |
| pepper.bmp (512x512 piksel) | (1 x 10 ⁻⁵) | | | PSNR: 51.4942 MSE: 0.460957 NC: 0.242827 |
| pepper.bmp (512x512 piksel) | 0.01 | | | PSNR: 51.5622 MSE: 0.453799 NC: 0.0521021 |

Hasil pengujian yang ditunjukkan pada Tabel 4 memiliki pola yang sama dengan hasil pengujian pada serangan Gaussian Noise. Watermark dapat diekstraksi dengan baik saat input konstanta kecil (1 x 10⁻⁸) dengan nilai NC yang tinggi, tingkat kemiripan watermark hasil ekstraksi dengan watermark asli tinggi. Tetapi saat input konstanta noise lebih besar, watermark gagal diekstraksi, nilai NC yang didapat rendah menunjukkan tingkat kemiripan watermark asli dengan watermark hasil ekstraksi rendah.

Nilai MSE menunjukkan bahwa semakin rendah konstanta noise yang diinputkan, nilai MSE yang diperoleh semakin tinggi. Hal ini berarti seiring dengan tingkat kemiripan yang naik tingkat error yang didapat semakin rendah. Sedangkan nilai PSNR yang didapat berkisar 48 – 50 dB menunjukan kualitas image baik.

7.3.3 Poisson Noise

Poisson noise merupakan noise yang dapat ditambahkan langsung tanpa memasukkan parameter apapun. Pengujian dilakukan menggunakan file cover medium lena.jpg dan 3 file watermark, yaitu uns400.png, dogs.jpg, dan pepper.bmp. Hasil pengujian ditampilkan pada tabel 5.

Tabel 5 Hasil Pengujian dengan Serangan Poisson Noise

| Watermark | Image | Watermark | PSNR (dB), |
|-----------|-------|-----------|------------|
|-----------|-------|-----------|------------|

| | terwater mark | k Hasil Ekstraksi | MSE dan NC |
|------------------------------|---------------|-------------------|--|
| uns400.png (400 x400 piksel) | | | PSNR: 48.9434 MSE: 0.829346 NC: 0.889782 |
| dogs.jpg (640x480 piksel) | | | PSNR: 54.152 MSE: 0.249967 NC: 0.959066 |
| pepper.bmp (512x512 piksel) | | | PSNR: 51.4941 MSE: 0.460964 NC: 0.972959 |

Hasil pengujian menunjukkan image yang diberi serangan poisson noise dapat diekstraksi dengan baik dengan nilai NC yang didapat tinggi sehingga dapat dikatakan tingkat kemiripan antara watermark hasil ekstraksi dengan watermark asli tinggi. Nilai PSNR yang didapat sekitar 48 – 54 dB menunjukkan kualitas watermark hasil ekstraksi baik. Sedangkan error pada image yang ditunjukkan dengan nilai MSE rendah, yaitu sekitar 0.4 – 0.8.

7.3.5 Geometrical Attacks

Geometrical attacks merupakan serangan yang sangat merubah image, karena dapat membuat seluruh piksel berpindah tempat atau hilang. Serangan geometrical attacks dapat berupa cropping dan rotasi. Pengujian dilakukan menggunakan file cover medium lena.jpg dan 3 file watermark, yaitu uns400.png, dogs.jpg, dan pepper.bmp. Hasil pengujian serangan rotation ditampilkan pada tabel 6 sedangkan pengujian pada serangan cropping ditunjukkan pada Tabel 7.

Tabel 6 Hasil Pengujian dengan Serangan Rotation

| Watermark | Rotasi | Image terwater mark | Watermark Hasil Ekstraksi | PSNR (dB), MSE dan NC |
|------------------------------|--------|---------------------|---------------------------|--|
| uns400.png (400 x400 piksel) | 1° | | | PSNR: 44.9739 MSE: 2.06866 NC: -0.00996111 |
| uns400.png (400 x400 piksel) | 30° | | | PSNR: 37.5684 MSE: 11.3825 NC: -0.032968 |
| uns400.png (400 x400 piksel) | 90° | | | PSNR: 39.8552 MSE: 6.72292 NC: -0.00857926 |
| dogs.jpg (640x480 piksel) | 1° | | | PSNR: 47.0091 MSE: 1.29471 NC: 0.0668491 |
| dogs.jpg (640x480 piksel) | 30° | | | PSNR: 38.7895 MSE: 8.59262 NC: 0.275835 |
| dogs.jpg (640x480 piksel) | 90° | | | PSNR: 40.2667 MSE: 6.11517 NC: 0.00574302 |
| pepper.bmp (512x512 piksel) | 1° | | | PSNR: 46.1264 MSE: 1.5865 NC: 0.0246164 |
| pepper.bmp (512x512 piksel) | 30° | | | PSNR: 38.226 MSE: 9.78311 NC: 0.0908375 |
| pepper.bmp (512x512 piksel) | 90° | | | PSNR: 40.4286 MSE: 5.8914 NC: 0.185294 |

Hasil pengujian image yang dikenai serangan rotasi dengan persentasi rotasi yang bervariasi seluruhnya gagal diekstraksi. Seluruh image yang diujikan memiliki nilai NC rendah sekitar 0.01 – 0.2 yang menunjukkan tingkat kemiripan antara watermark hasil ekstraksi dengan watermark asli sangat

rendah.

Nilai MSE yang didapat pada seluruh *image* tinggi, yaitu antara angka 1 – 9.7 sehingga tingkat *error* yang didapat tinggi. Sedangkan untuk nilai PSNR pada keseluruhan *image* sekitar 36 – 42 dB yang menunjukkan kualitas *image* baik.

Tabel 7 Hasil Pengujian dengan Serangan Cropping

| Watermark | Image terwater mark | Watermark Hasil Ekstraksi | PSNR (dB), MSE dan NC |
|------------------------------|---|---|--|
| uns400.png (400 x400 piksel) |  |  | PSNR: 40.472 MSE: 5.83287 NC: 0.0154622 |
| dogs.jpg (640x480 piksel) |  |  | PSNR: 40.5899 MSE: 5.67657 NC: 0.0788692 |
| pepper.bmp (512x512 piksel) |  |  | PSNR: 40.8074 MSE: 5.39934 NC: 0.189743 |

Tabel 7 menunjukan seluruh *image* yang dikenai serangan *cropping* gagal diekstraksi. Nilai NC yang diperoleh sangat kecil antara 0.01 – 0.1 yang menunjukkan tingkat kemiripan antara *watermark* asli dengan *watermark* hasil ekstraksi sangat rendah. Sedangkan nilai MSE yang didapat pada ketiga *image* tinggi, sekitar 5 – 6, sehingga dapat dikatakan *error* pada *image* tinggi. Kualitas *watermark* hasil ekstraksi baik karena Nilai PSNR yang didapat berkisar 40 dB.

7.3.6 Blurring

Blurring adalah operasi pengaburan *image*. Pengaburan *image* adalah suatu filter *low-pass* yang melenyapkan detail halus dari suatu *image*. Pengujian dilakukan menggunakan file *cover medium lena.jpg* dan 3 file *watermark*, yaitu *uns400.png*, *dogs.jpg*, dan *pepper.bmp*. Hasil pengujian ditampilkan pada tabel 8.

Tabel 8 Hasil Pengujian dengan Serangan Blurring

| Watermark | Image terwater mark | Watermark Hasil Ekstraksi | PSNR (dB), MSE dan NC |
|------------------------------|---|---|--|
| uns400.png (400 x400 piksel) |  |  | PSNR: 47.4451 MSE: 1.17103 NC: -0.0040617 |
| dogs.jpg (640x480 piksel) |  |  | PSNR: 50.4313 MSE: 0.588778 NC: 0.00212683 |
| pepper.bmp (512x512 piksel) |  |  | PSNR: 49.1225 MSE: 0.795855 NC: 0.00891258 |

Tabel 8 menunjukkan hasil pengujian pada ketiga *image* gagal diekstraksi. Nilai NC yang didapat pada ketiga *image*

rendah, yaitu berkisar (-0.004) – 0.008 yang berarti tingkat kemiripan *watermark* hasil ekstraksi dengan *watermark* asli sangat rendah. Sedangkan nilai PSNR menunjukkan ketiga *image* berkualitas baik walaupun dengan tingkat *error* yang tinggi.

8. Kesimpulan

Berdasarkan penelitian, hasil dan pembahasan yang telah dipaparkan di atas, maka dapat ditarik kesimpulan bahwa *Watermarking Image Digital* dengan *Discrete Wavelet Transform* dan *Singular Value Decomposition (DWT-SVD)* dapat diekstraksi kembali dengan baik. Selain itu, *watermark* tahan terhadap beberapa serangan seperti *Gaussian Noise*, *Speckle Noise*, *Salt & Pepper Noise*, dan *Poisson Noise* dengan rata-rata nilai NC 0.8, *watermark* tidak tahan terhadap serangan *rotation*, *cropping*, dan *blurring*. Untuk penelitian selanjutnya variasi serangan pada pengujian dapat ditambah, atau dapat menggunakan filter *Wavelet* lain seperti *Daubechies* dan atau menggunakan model warna *image* selain RGB, seperti HSI (*Hue, Saturation, Intensity*) dan CMY (*Cyan, Magenta, Yellow*).

9. Daftar Pustaka

- [1]. Cox, Ingemar J., dkk. 2008. *Digital Watermarking and Steganography – Second Edition*. Burlintong, USA: Morgan Kaufmann Publishers.
- [2]. Katzenbeisser, S., F. A. P. Petitcolas. 2000. *Information Hiding: Techniques for Steganography and Digital Watermarking*. Computer Security Series: Artech House Books.
- [3]. Gonzales, R.C., Woods, R.E. 2010. *Digital Image Processing – Third Edition*. Prentice Hall. Upper Saddle River, New Jersey, 07458.
- [4]. Alfatwa, Dean Fathony. 2011. *Watermarking pada Image Digital menggunakan Discrete Wavelet Transform*. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [5]. T. J., Anumol., Karthigaikumar, P. 2011. *DWT based Invisible Image Watermarking Algorithm for Color Images*. IJCA Special Issue on “Computational Science – New Dimensions & Perspectives” NCCSE 2011 pp. 76-79.
- [6]. Praful Saxena, dkk. 2012. *DWT-SVD Semi-Blind Image Watermarking Using High Frequency Band*. 2nd International Conference on Computer Science and Information Technology (ICCSIT’2012) Singapore April 28-29, 2012 pp.138-142.
- [7]. Putra, Dharma. 2010. *Pengolahan Image Digital*. Yogyakarta: Penerbit Andi.

PEMBUATAN PETA DATA PREDIKSI PENDERITA PENYAKIT DBD DI KOTA SURAKARTA DENGAN METODE *SUSCEPTIBLE-INFECTED-REMOVED* (SIR)

Diannita Kartikasari

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
21diannita@gmail.com

Sarngadi Palgunadi

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
palgunadi@uns.ac.id

Bambang Harjito

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
harjitob2011@gmail.com

Abstrak— Penyakit Demam Berdarah *Dengue* (DBD) adalah penyakit menular yang disebabkan oleh virus *dengue* dan ditularkan oleh nyamuk *Aedes aegypti*. Di Kota Surakarta, angka kematian akibat penyakit DBD menunjukkan angka lebih dari 1% dalam jangka waktu tahun 2008-2012. Berdasarkan angka kematian tersebut, Pemerintah Kota Surakarta perlu mengambil tindakan pencegahan sekaligus melakukan pengobatan yang efektif guna mengurangi kematian akibat DBD, salah satunya membuat peta data simulasi untuk memprediksi jumlah penderita DBD. Penelitian ini memperkenalkan model simulasi jumlah penderita dan prediksi penyakit DBD di Kota Surakarta dengan metode SIR (*Susceptible-Infected-Removed*) dan menampilkan hasilnya dalam peta Kota Surakarta dengan warna pada setiap kecamatan yang mewakili jumlah kasus DBD yang terjadi pada setiap kecamatan. Data yang digunakan dalam penelitian ini adalah data jumlah penduduk, jumlah kasus DBD, dan jumlah penduduk meninggal akibat DBD di Kota Surakarta tahun 2009 - 2013 yang diperoleh dari Dinas Kesehatan Kota Surakarta. Penelitian ini menunjukkan bahwa peta pemodelan data prediksi jumlah kasus DBD di Kota Surakarta dapat dibuat menggunakan metode SIR dengan nilai laju penularan (β) antara 0,019 – 0,020.

Katakunci— DBD, SIR dan Peta Data

1. LATAR BELAKANG

Penyakit Demam Berdarah *Dengue* (DBD) atau *Dengue Hemorrhagic Fever* (DHF) sampai saat ini merupakan salah satu masalah kesehatan masyarakat di Indonesia yang cenderung meningkat jumlah pasien serta semakin luas penyebarannya. Penyakit DBD ini ditemukan hampir di seluruh belahan dunia terutama di negara-negara tropik dan subtropik, baik sebagai penyakit endemik maupun epidemik [1]. Menurut Departemen Kesehatan RI, penyakit Demam Berdarah *Dengue* (DBD) adalah penyakit menular yang disebabkan oleh virus *dengue* dan ditularkan oleh nyamuk

Aedes aegypti, yang ditandai dengan demam mendadak 2 sampai 7 hari tanpa penyebab yang jelas, lemah/lesu, gelisah, nyeri ulu hati, disertai tanda pendarahan di kulit berupa bintik perdarahan (*petechiae*), lebam (*echymosis*) atau ruam (*purpura*). Kadang-kadang mimisan, berak darah, muntah darah, kesadaran menurun atau renjatan (*shock*). Sebagian besar DBD menyerang anak-anak dengan usia dibawah 15 tahun. Penyakit ini merupakan salah satu penyakit yang persebarannya melalui vektor, maka dari itu perlu diwaspadai karena penularan penyakit ini akan semakin meningkat dengan perubahan iklim [2].

Data dari Dinas Kesehatan Provinsi Jawa Tengah dalam Buku Profil Kesehatan Provinsi Jawa Tengah Tahun 2012 [3], angka kesakitan/*Incidence Rate* (IR) DBD di Jawa Tengah menunjukkan angka 19,29/100.000 penduduk dan angka kematian/*Case Fatality Rate* (CFR) sebesar 1,52%. Angka ini menunjukkan peningkatan apabila dibandingkan dengan IR pada tahun 2011 yang menunjukkan angka 15,27/100.000 penduduk dan CFR sebesar 0,93%. Sementara angka kematian/CFR akibat DBD di Kota Surakarta menunjukkan angka lebih dari 1% dalam jangka waktu tahun 2008-2012.

Berdasarkan data tersebut, maka pihak pengelola kesehatan Kota Surakarta harus segera mengambil tindakan untuk menekan angka kematian akibat DBD. Tindakan pengobatan dan pencegahan yang efektif diperlukan supaya penanganan kasus DBD lebih cepat dan tepat sasaran. Dalam menangani kasus DBD, pihak pengelola kesehatan Kota Surakarta belum mempunyai peta yang menggambarkan jumlah penderita penyakit DBD. Peta tersebut diperlukan untuk mengetahui status penyebaran jumlah penderita DBD di setiap wilayah kecamatan di Kota Surakarta. Selain itu, prediksi jumlah kasus DBD di waktu yang akan datang juga diperlukan untuk pertimbangan dalam pengambilan tindakan pencegahan yang efektif. Namun saat ini pihak pengelola kesehatan Kota Surakarta belum memiliki peta untuk simulasi dan prediksi penyebaran penyakit DBD di Kota Surakarta. Simulasi penyebaran dan prediksi penyakit DBD di Kota Surakarta diperlukan supaya memudahkan pihak

pengelola kesehatan di Kota Surakarta dalam membuat kebijakan pencegahan penyebaran penyakit DBD.

Simulasi pemodelan untuk penyebaran penyakit menular telah banyak dilakukan. Salah satu pemodelan penyebaran penyakit yang digunakan adalah dengan *Susceptible-Infected-Recovered (SIR)*. SIR pada awalnya dikembangkan untuk mengetahui laju penyebaran dan kepunahan suatu wabah penyakit dalam populasi tertutup dan bersifat epidemis [4]. Dalam penelitian ini, SIR model akan digunakan untuk memodelkan data jumlah penderita penyakit DBD di Kota Surakarta. Hasil pemodelan data tersebut akan dikelompokkan sesuai warna yang mewakili jumlah kasus DBD yang terjadi di setiap kecamatan, kemudian warna-warna tersebut akan divisualisasikan ke dalam peta Kota Surakarta.

Metode SIR digunakan dalam pembuatan peta data prediksi penderita DBD di Kota Surakarta karena metode ini adalah metode yang paling umum digunakan untuk

pemodelan data penyakit menular, seperti pada penelitian yang dilakukan oleh Side dan Noorani (2013). Penelitian tersebut menggunakan SIR untuk memodelkan data penyebaran penyakit DBD berdasarkan kasus DBD yang terjadi di Sulawesi Selatan (Indonesia) dan Selangor (Malaysia). Hasil penelitian tersebut berupa grafik yang menunjukkan jumlah individu rentan terinfeksi, jumlah kasus, dan individu yang sembuh dari DBD dengan waktu, dimana grafik tersebut hanya dapat dipahami oleh orang tertentu. Sedangkan apabila hasil pemodelan data tersebut ditampilkan peta, maka hasil pemodelan data akan lebih mudah dipahami dan dapat memudahkan dalam analisa untuk mengambil tindakan penanganan kasus DBD.

2. PENELITIAN TERKAIT

Pemodelan penyebaran penyakit menular telah dilakukan pada beberapa penelitian yang dijelaskan pada Tabel 1 berikut ini

Tabel 1. Beberapa metode SIR, Tujuan, Kelebihan dan Kelemahan

| No | Penulis | Tujuan | Metode | Kelebihan | Kelemahan |
|----|---|---|---|--|---|
| 1 | Side dan Noorani (2013) [5] | Mengetahui kecocokan data empiris dengan hasil simulasi peyebaran penyakit DBD di Sulawesi Selatan dan Selangor dengan SIR model. | SIR model dengan melibatkan 2 populasi, yaitu populasi manusia sebagai <i>host</i> , dan populasi nyamuk sebagai populasi <i>vector</i> . | Dapat mengetahui hubungan populasi manusia yang terinfeksi DBD dengan populasi nyamuk. | <i>Parameter value</i> dalam penelitian tersebut menggunakan nilai dari penelitian yang pernah dilakukan dan hanya untuk wilayah Malaysia dan Indonesia. |
| 2 | Syaripuddin (2009) [6] | Melakukan simulasi pemodelan penyakit epidemis jenis SIR. | SIR | Dapat mengetahui pengaruh faktor kelahiran dan kematian dalam pemodelan SIR. | - Hasil simulasi pada penelitian ini berupa grafik yang maksud dari grafik tersebut hanya dapat dimengerti oleh orang tertentu. - Tidak menggunakan data sesuai kenyataan. |
| 3 | Tjolleng, Komalig, dan Prang (2013) [7] | Mengetahui dinamika perkembangan HIV/AIDS di Sulawesi Utara menggunakan model persamaan diferensial nonlinear SIR | SIR | Dapat mengetahui satu titik tetap bebas penyakit | Hasil simulasi pada penelitian ini berupa grafik yang maksud dari grafik tersebut hanya dapat dimengerti oleh orang tertentu. |
| 4 | Fredlina, Oka, & Dwipayana (2012) [8] | Membuat pemodelan data penyebaran penyakit tuberkulosis dan mengetahui parameter yang paling berpengaruh dalam model penyebaran penyakit tuberkulosis | SIR dan Runge Kutta Orde 4 | Dapat mengetahui pengaruh paling signifikan dari parameter yang digunakan pada simulasi. | Hasil simulasi pada penelitian ini berupa grafik hanya dapat dimengerti oleh orang tertentu. |

Tabel 1. Beberapa metode SIR, Tujuan, Kelebihan dan Kelemahan (lanjutan)

| No | Penulis | Tujuan | Metode | Kelebihan | Kelemahan |
|----|--------------------------------------|---|---|--|---|
| 5 | Widi, Nataliani, & Hendry (2011) [9] | Mengetahui daerah-daerah penyebaran penyakit DBD dan memvisualisasikan pemetaan daerah endemis DBD dengan pemodelan SIR di Kabupaten Semarang | SIR dengan menggunakan variabel jumlah kelahiran dan kematian akibat DBD dan non DBD. | Dalam penelitian tersebut juga menggunakan variabel jumlah kelahiran dan kematian akibat DBD dan non DBD, sehingga lebih mendekati keadaan sebenarnya. | Tidak dijelaskan bagaimana cara memperoleh persamaan SIR yang ditambahkan dengan variabel jumlah kelahiran dan kematian akibat DBD dan non DBD. |
| 6 | Picollo & Billings, (2005) [10] | Membuat model untuk mengetahui pengaruh vaksinasi dan penduduk imigran pada penyakit anak-anak di New York | Expanded SIR model. (SIR dengan melibatkan variabel jumlah penduduk asli dan imigran) | Dapat membuat dua sub populasi dalam satu pemodelan SIR. | Model SIR yang telah dimodifikasi dalam penelitian tersebut tidak sesuai untuk pemodelan di Indonesia. |
| 7 | Palgunadi & Herlambang (2014) [11] | Melakukan analisis simulasi penyebaran penyakit ISPA pada balita di Kota Surakarta | SIR, Game of Life | Dapat mengetahui perpindahan dan penyebaran penyakit pada setiap sel. | Perubahan pada sel belum dinamis. |

3. Susceptible-Infected-Recovered (SIR)

Suatu fenomena/peristiwa alam dapat dimodelkan ke dalam bentuk lain sehingga fenomena tersebut dapat dipelajari. Pengertian model adalah suatu usaha untuk menciptakan suatu replika/tiruan dari suatu fenomena/peristiwa alam. Ada tiga jenis model yaitu model fisik, model analogi, dan model matematik [12]. Pada model matematik, suatu fenomena/peristiwa alam dengan suatu set persamaan. Kemudian nilai kecocokan model matematika tergantung dari ketepatan formulasi matematis dalam mendeskripsikan fenomena alam tersebut.

Contoh fenomena/peristiwa yang dapat dimodelkan supaya dapat dipelajari lebih lanjut adalah penyebaran penyakit menular. Penyebaran penyakit menular dalam suatu populasi dapat dimodelkan ke dalam bentuk matematika [13]. Salah satu pemodelan penyakit menular adalah *Susceptible-Infected-Recovered* (SIR).

SIR model pertama kali diperkenalkan oleh Kermack dan McKendrick pada tahun 1927 [13]. Pada SIR model yang umum digunakan, jumlah kelahiran dan kematian akibat penyakit lain diabaikan supaya SIR model tetap sederhana [14]. Dalam SIR model, suatu populasi dibagi menjadi 3 class yaitu *susceptible* (*S*) yang terdiri dari individu yang rentan terserang penyakit, *infected* (*I*) yang terdiri dari individu yang sedang terserang penyakit dan dapat menularkannya, dan *removed* (*R*) yang terdiri dari individu yang sudah tidak menderita penyakit, baik yang sembuh ataupun yang meninggal akibat penyakit tersebut [15].

Kemudian menurut Dimitrov & Meyers [16], setiap individu dalam model tersebut bertindak sebagai berikut :

1. Setiap individu yang ada dalam kelas *susceptible* menggambarkan seseorang yang acak dari suatu populasi.
2. Jika seorang dalam kelas *susceptible* yang dipilih secara acak terinfeksi penyakit, maka individu tersebut berubah statusnya menjadi pada kelas *infected* dengan probabilitas β .
3. Setiap individu dalam fase *infected* berubah ke kelas *removed* dengan probabilitas γ .
4. Setiap individu yang berada dalam kelas *removed* tetap *removed*.

Setiap individu yang baru dilahirkan termasuk ke kelas *susceptible*. Setelah terjadi kontak dengan individu yang terinfeksi penyakit, individu *susceptible* akan berpindah menjadi kelas *infected*. Setelah berada dalam kelas *infected*, individu akan berpindah ke kelas *removed*. SIR model dapat dilihat dalam Gambar 1.

**Gambar 1. Diagram SIR model [17]**

Suatu populasi dibagi menjadi 3 kelas, yaitu kelas *susceptible* (*S*), *infected* (*I*), dan *removed* (*R*). Individu akan berubah dari kelas *S* ke *I*, lalu ke *R* sesuai dengan arah tanda panah. Total individu dalam populasi dinyatakan dengan $N = S + I + R$.

R. Parameter β adalah kekuatan dari infeksi, yaitu probabilitas individu dalam kelas S terinfeksi penyakit menular, dan parameter γ adalah *recovery rate*. Parameter γ berkaitan dengan panjang periode dimana seseorang dapat menularkan penyakit, disebut *infectious period*. Secara khusus, total waktu yang dihabiskan pada kelas I oleh individu adalah suatu variabel acak geometris dengan probabilitas keberhasilan γ , hal ini membuat panjang periode masa penularan yang diharapkan sebesar $\frac{1}{\gamma}$ [16]. Nilai γ diperoleh dari pembagian 1 dengan jumlah hari dimana penyakit dapat menular [18].

Persamaan umum dari SIR model adalah [16]:

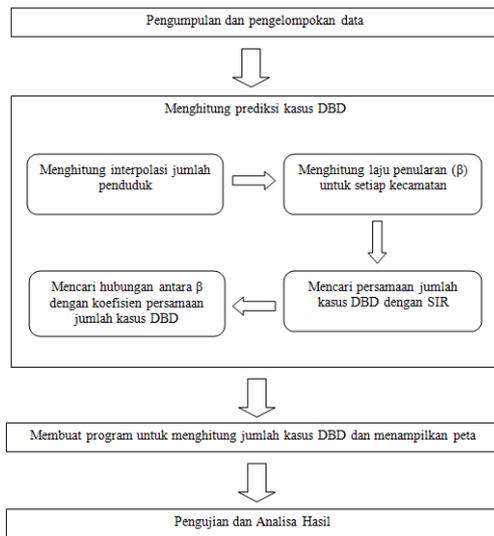
$$\frac{dS_t}{dt} = -\beta \cdot S_t \cdot I_t \tag{2.1}$$

$$\frac{dI_t}{dt} = \beta \cdot S_t \cdot I_t - \gamma \cdot I_t \tag{2.2}$$

$$\frac{dR_t}{dt} = \gamma \cdot I_t \tag{2.3}$$

3. ALUR PENELITIAN

Langkah-langkah dalam penelitian ini dapat dilihat pada Gambar 2 berikut ini



Gambar 2. Alur Pembuatan Peta Data Prediksi Penderita Penyakit DBD di Kota Surakarta dengan Metode SIR

Data yang digunakan dalam penelitian ini adalah data jumlah penduduk, jumlah kasus DBD, dan jumlah penduduk meninggal akibat DBD di Kota Surakarta dari tahun 2009 sampai 2013 yang diperoleh dari Dinas Kesehatan Kota Surakarta. Selanjutnya data tersebut dikelompokkan sesuai dengan kelas pada SIR meliputi :

- Kelas S adalah data jumlah penduduk per kecamatan dikurangi dengan data jumlah kasus DBD per bulan di setiap kecamatan.
- Kelas I adalah data jumlah kasus DBD per bulan.
- Kelas R adalah data jumlah penduduk meninggal akibat DBD per bulan.

Secara garis besar, penelitian ini dibagi dalam 2 tahap yaitu menghitung prediksi jumlah kasus DBD dan membuat program untuk menampilkan peta.

4. PREDIKSI KASUS DBD

Untuk menghitung prediksi kasus DBD per kecamatan di Kota Surakarta, langkah-langkah yang dilakukan adalah sebagai berikut :

4.1. Menghitung Interpolasi Jumlah Penduduk

Jumlah penduduk per bulan pada setiap kecamatan dihitung dengan melakukan interpolasi pada data jumlah penduduk per tahun yang telah diperoleh dari Dinas Kesehatan Kota Surakarta. Interpolasi jumlah penduduk dilakukan dengan rumus interpolasi linier [19] :

$$\frac{f(x) - f(x_1)}{x - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \tag{4.1}$$

Karena 1 tahun ada 12 bulan, maka dari persamaan 4.1. di atas kemudian diperoleh persamaan untuk menghitung jumlah penduduk sebagai berikut :

$$f(x_j) = \frac{x_j - 1}{12} \times (f(x_{i+1}) - f(x_i)) + f(x_i) \tag{4.2}$$

$$i = 1,2,3,4$$

$$j = 1,2,3, \dots, 12$$

- $f(x_j)$ = Jumlah penduduk per bulan
- $f(x_i)$ = Jumlah penduduk pada tahun ke- i
- $f(x_{i+1})$ = Jumlah penduduk pada tahun $i+1$
- x_j = Bulan ke- j

4.2. Menghitung Laju Penularan (β) Setiap Kecamatan

Dalam penelitian ini, nilai β setiap kecamatan dihitung berdasarkan persamaan yang dijabarkan oleh Dimitrov dan Meyers [11]. β dalam penelitian diperoleh dengan rumus sebagai berikut :

$$\beta = \frac{S_{t+1} - S_t}{S_t \cdot \frac{1}{N}} \tag{4.3}$$

$$t = 1,2,3, \dots, 11$$

- S_t = jumlah individu yang rentan tertular pada waktu t
- Y_t = jumlah individu yang menderita penyakit pada waktu t
- N = jumlah individu dalam populasi

Nilai β dicari pada setiap bulan pada tahun 2009 sampai 2013, kemudian dicari median dari β setiap kecamatan yang akan digunakan untuk mendapatkan persamaan jumlah kasus DBD dari persamaan SIR.

4.3. Mencari Persamaan Jumlah Kasus DBD dengan Sir

Nilai β dan γ dimasukkan ke persamaan SIR (2.1) dan (2.2) dengan menggunakan *software* Maple 11 yang akan menghasilkan grafik hubungan antara jumlah kasus DBD dengan waktu. Nilai γ untuk seluruh wilayah kelurahan dan untuk semua waktu adalah sama yaitu $\frac{1}{14}$ atau 0.07. Angka tersebut diperoleh dari lamanya masa infeksi penyakit DBD yaitu 14 hari. Kemudian menurut Nicho [18], nilai γ diperoleh dari pembagian 1 dengan jumlah hari dimana penyakit dapat menular, sehingga diperoleh angka $\frac{1}{14}$ atau 0.07.

Langkah selanjutnya adalah mencari persamaan dari grafik tersebut dengan *software* Curve Expert. Caranya adalah dengan memasukkan nilai x dan y pada grafik ke Curve Expert, dengan x adalah waktu dan y adalah jumlah kasus DBD, kemudian dicari model persamaan yang bentuk grafiknya mendekati grafik yang telah dihasilkan dari Maple 11. Hasilnya adalah persamaan (4.4)

$$y = \frac{a + bx}{1 + cx + dx^2} \tag{4.4}$$

dengan a, b, c, dan d merupakan koefisien yang nilainya dapat berubah tergantung pada nilai β .

Hasil yang diperoleh pada persamaan (4.4) tersebut merupakan rasio jumlah kasus DBD dengan jumlah penduduk, sehingga untuk memperoleh angka jumlah kasus DBD per kecamatan, rasio tersebut harus dikalikan jumlah penduduk per kecamatan seperti pada persamaan (4.5) berikut

$$\text{jumlah kasus} = \frac{a + bx}{1 + cx + dx^2} \times \text{jumlah penduduk} \tag{4.5}$$

4.4. Mencari Hubungan Antara β Dengan Koefisien Persamaan Jumlah Kasus DBD

Persamaan menghitung jumlah kasus DBD memiliki koefisien a, b, c, dan d yang nilainya berbeda pada setiap nilai β . Untuk mempermudah dalam menentukan nilai koefisien persamaan jumlah kasus DBD, maka dicari hubungan antara nilai β dengan koefisien a, b, c, dan d pada persamaan jumlah kasus DBD dengan cara :

- Memasukkan nilai β dan koefisien a ke Curve Expert, yang akan menghasilkan persamaan

$$a = \frac{0,015 - 0,047 \beta}{1 - 0,709 \beta + 0,433 \beta^2} \tag{4.6}$$

- Memasukkan nilai β dan koefisien b ke Curve Expert, yang akan menghasilkan persamaan

$$b = \frac{-0,001 + 0,046 \beta}{1 - 0,249 \beta + 0,019 \beta^2} \tag{4.7}$$

- Memasukkan nilai β dan koefisien c ke Curve Expert, yang akan menghasilkan persamaan

$$c = \frac{0,037 - 0,364 \beta}{1 + 0,411 \beta + 0,451 \beta^2} \tag{4.8}$$

- Memasukkan nilai β dan koefisien d ke Curve Expert, yang akan menghasilkan persamaan

$$d = \frac{0,001 + 0,010 \beta}{1 - 0,215 \beta + 0,016 \beta^2} \tag{4.9}$$

5. PEMBUATAN PETA DATA PREDIKSI PENDERITA DBD DENGAN SIR

Program dibuat dalam 3 *class* yaitu class untuk tampilan utama dan memasukkan input tahun, bulan, dan β . Yang kedua adalah *class* untuk menghitung jumlah kasus dengan SIR model yang terdapat *method* untuk menghitung koefisien a, b, c, dan d, dan *method* untuk jumlah kasus DBD per kecamatan. Algoritma program untuk menghitung jumlah kasus DBD per kecamatan ditunjukkan pada Gambar 3.

```

public double hitungA(double beta){
    a=(0.015-0.047*beta)/(1-0.709*beta
    +0.433*(beta*beta));
    return a;
}

public double hitungB(double beta){
    b=(-0.001+0.046*beta)/(1-
    0.249*beta+0.019*(beta*beta));
    return b;
}

public double hitungC(double beta){
    c=(0.037-0.364*beta)/(1+
    0.411*beta+0.451*(beta*beta));
    return c;
}

public double hitungD(double beta){
    d=(0.001+0.01*beta)/(1-
    (0.215*beta)+(0.016*(beta*beta)));
    return d;
}

public int hitungKasus(int x, int kec, double beta){
    kasus=(hitungA(beta)+(hitungB(beta)*x))/(1+(hitungC(b
    eta)*x)+ (hitungD(beta)*(x*x))* jmlPend[kec][x];
    kasusKec=(int) kasus;

    return kasusKec;
}
    
```

Gambar 3. Algoritma Program Menghitung Jumlah Kasus DBD

Program akan menghitung nilai koefisien a, b, c, dan d yang akan digunakan sebagai koefisien untuk persamaan jumlah kasus DBD. Persamaan untuk menghitung nilai koefisien tersebut diperoleh dari langkah 4.4. Setelah mendapatkan nilai koefisien kemudian menghitung jumlah kasus DBD dengan persamaan jumlah penderita DBD dengan nilai koefisien seperti yang telah diperoleh.

Class yang ketiga adalah untuk menampilkan hasil perhitungan jumlah kasus per kecamatan ke dalam peta sesuai dengan warna klasifikasi jumlah kasus DBD tersebut. Warna untuk jumlah kasus DBD diklasifikasikan berdasarkan standar deviasi dari data kasus DBD yang telah diperoleh. Jumlah kasus DBD diklasifikasikan dalam 3 warna yaitu hijau, kuning, dan merah. hijau untuk jumlah kasus

antara 0-14, kuning untuk 15-21 kasus, dan merah untuk jumlah kasus lebih dari 21 kasus.

Algoritma program untuk menampilkan peta ditunjukkan pada Gambar 4.

```

bk = 15;
bm = 21;

for (j=0; j<height; j++){
for (i = 0; i < width; i++){
for(int kec=0; kec<5; n++){
if(label[j][i]==kec){

    if(jml[kec][0]< bk) img.setRGB(i, j, g.getRGB());

    else if (jml[kec][0]>=bk && jml[n][0]<=bm)
img.setRGB(i, j, y.getRGB());

    else if (jml[kec][0] > bm) img.setRGB(i, j,
r.getRGB());
}
    if(label[j][i]==5) img.setRGB(i, j, w.getRGB());
}
}
}
    
```

Gambar 4. Algoritma Program Menampilkan Peta

Untuk menampilkan warna pada setiap kecamatan, piksel setiap kecamatan diberi label yang berbeda, yaitu label 0 untuk Kecamatan Laweyan, 1 untuk Kecamatan Jebres, 2 untuk Kecamatan Banjarsari, 3 untuk Kecamatan Pasar Kliwon, dan 4 untuk Kecamatan Serengan. Kemudian setiap piksel diperiksa apabila jumlah kasus DBD pada kecamatan yang ditunjukkan oleh label yang sedang diperiksa adalah kurang dari 15 kasus maka pada piksel tersebut akan diberi warna hijau. Apabila jumlah kasus adalah 15-21 kasus maka piksel akan diberi warna kuning, dan apabila jumlah kasus lebih dari 21 kasus maka piksel akan diberi warna merah.

Pada saat dijalankan, program akan menerima input tahun dan bulan yang akan diprediksi dan nilai β untuk setiap kecamatan. Nilai β dibatasi antara 0,019 – 0,020. Setelah nilai β dimasukkan, program akan menghitung koefisien a, b, c, dan d untuk persamaan penyelesaian yang kemudian akan digunakan untuk menghitung jumlah kasus per kecamatan. Hasil perhitungan jumlah kasus per kecamatan yang diperoleh kemudian diklasifikasikan sesuai pembagian warna yang telah ditentukan. Kemudian warna – warna tersebut ditampilkan ke dalam peta Kota Surakarta dengan warna pada setiap kecamatan yang merepresentasikan jumlah kasus DBD.

6. HASIL DAN PEMBAHASAN

6.1. Hasil Perhitungan Nilai β

Nilai β setiap kecamatan dihitung pada setiap bulan dari tahun 2009 – 2013 dengan persamaan 4.3. Data hasil perhitungan tersebut kemudian dicari nilai median data setiap kecamatan yang hasilnya dapat dilihat pada Tabel 2 berikut

Tabel 2. Hasil Perhitungan Nilai β

| Kecamatan | Nilai β |
|-----------|---------------|
|-----------|---------------|

| | |
|--------------|-------|
| Laweyan | 0.200 |
| Jebres | 0.500 |
| Banjarsari | 0.111 |
| Pasar Kliwon | 0.225 |
| Serengan | 0.286 |

Kemudian nilai β tersebut digunakan untuk menghitung data jumlah kasus DBD pada tahun 2013 bulan ke-2. Hasilnya ditunjukkan pada tabel 3

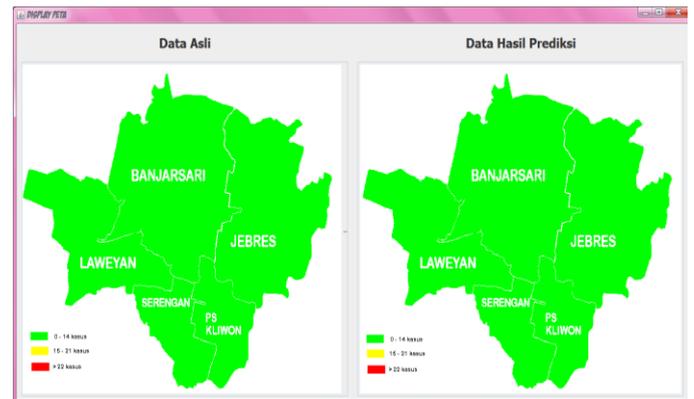
Tabel 3. Hasil Simulasi Kasus DBD Tahun 2013 Bulan Ke-2

| Kecamatan | Data asli | Data Hasil Prediksi |
|--------------|-----------|---------------------|
| Laweyan | 5 | 3043 |
| Jebres | 4 | 5770 |
| Banjarsari | 4 | 3175 |
| Pasar Kliwon | 0 | 2600 |
| Serengan | 1 | 159 |

Berdasarkan tabel 3 diketahui bahwa data hasil prediksi sangat tidak sesuai dengan keadaan pada data asli, sehingga nilai β harus diganti dengan nilai sekitar $\frac{1}{10} \beta$ akibat adanya fenomena gunung es. Maka dari itu, nilai β pada penelitian ini dibatasi antara 0,019 sampai 0,020.

6.2. Hasil Simulasi

Simulasi diawali dengan memilih tahun dan bulan yang akan dimodelkan atau diprediksi. Untuk nilai β default telah ditentukan, tetapi masih dapat diganti dengan nilai antara 0,019-0,020. Untuk tahun 2009-2013, karena telah ada data asli maka data asli kasus DBD tersebut dapat dibandingkan dengan data hasil prediksi, data asli dan data prediksi tersebut juga dapat ditampilkan pada peta. Hasil simulasi untuk tahun 2009 bulan ke-1 (Januari) dengan nilai β untuk setiap kecamatan adalah 0,019 ditunjukkan pada gambar 5 berikut



Gambar 5. Peta Hasil Simulasi Tahun 2009 Bulan Ke-1 $\beta=0,019$

Peta hasil simulasi untuk tahun 2009 pada bulan ke-1 menunjukkan bahwa pada data asli dan data hasil prediksi

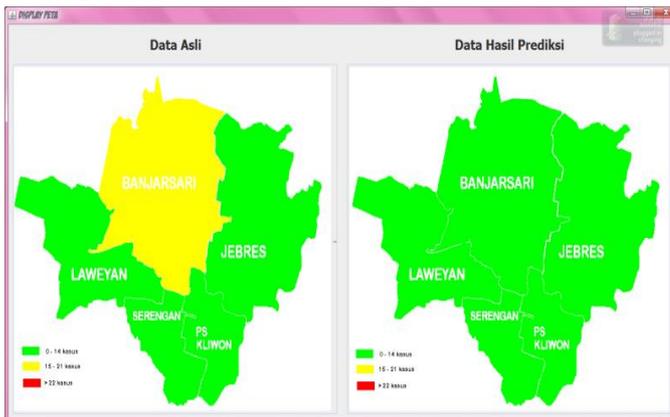
jumlah kasus DBD di semua kecamatan adalah kurang dari 15 kasus. Selain itu, peta tersebut juga menunjukkan bahwa perhitungan jumlah kasus dengan menggunakan SIR untuk tahun 2009 bulan ke-1 yang ditampilkan pada peta hasilnya sama dengan peta data asli tahun 2009 bulan ke-1, meskipun jumlah kasusnya tidak sama persis. Hasil perhitungan jumlah kasus DBD ditunjukkan dalam Tabel 4

Tabel 4. Jumlah Kasus DBD Tahun 2009 Bulan Ke-1 $\beta=0,019$

| Kecamatan | Data asli | Data Hasil Prediksi |
|--------------|-----------|---------------------|
| Laweyan | 2 | 3 |
| Jebres | 6 | 4 |
| Banjarsari | 8 | 6 |
| Pasar Kliwon | 2 | 3 |
| Serengan | 4 | 2 |

Selisih antara data asli dan data hasil prediksi tidak terlalu jauh, sehingga pemodelan dengan SIR yang tepat untuk memodelkan data jumlah kasus DBD di Kota Surakarta pada tahun 2009 bulan ke-1 adalah dengan nilai $\beta = 0,019$ untuk setiap kecamatan.

Simulasi lainnya kemudian dilakukan untuk tahun 2013 bulan ke-6 dengan nilai $\beta 0,019$ pada setiap kecamatan. Peta hasil perhitungan dapat dilihat pada Gambar 6 berikut



Gambar 6. Peta Hasil Simulasi Tahun 2013 Bulan Ke-6 dengan $\beta=0,019$

Pada tampilan peta tersebut terlihat perbedaan warna antara peta tampilan data asli dan data hasil perhitungan di Kecamatan Banjarsari. Pada peta data asli, Kecamatan Banjarsari berwarna kuning yang menunjukkan bahwa jumlah kasus DBD di Banjarsari adalah antara 15 – 21 kasus. Sedangkan di peta hasil perhitungan, Kecamatan Banjarsari berwarna hijau yang menunjukkan bahwa jumlah kasus hasil perhitungan dengan SIR di Banjarsari berjumlah kurang dari 15 kasus. Data asli dan hasil perhitungan ditampilkan pada Tabel 5

Tabel 5. Jumlah Kasus DBD Tahun 2013 Bulan Ke-6 $\beta=0,019$

| Kecamatan | Data asli | Data Hasil Prediksi |
|--------------|-----------|---------------------|
| Laweyan | 8 | 0 |
| Jebres | 4 | 0 |
| Banjarsari | 18 | 0 |
| Pasar Kliwon | 5 | 0 |
| Serengan | 7 | 0 |

| Kecamatan | Data asli | Data Hasil Prediksi |
|--------------|-----------|---------------------|
| Laweyan | 8 | 8 |
| Jebres | 4 | 7 |
| Banjarsari | 18 | 18 |
| Pasar Kliwon | 5 | 7 |
| Serengan | 7 | 7 |

Hasil tersebut menunjukkan bahwa model SIR dengan $\beta 0,019$ untuk setiap kecamatan kurang tepat untuk simulasi tahun 2013 bulan ke-6, karena jumlah kasus pada data hasil prediksi jauh di bawah jumlah kasus pada data asli. Apabila terjadi hasil prediksi semacam itu, dikhawatirkan pihak pengelola kesehatan Kota Surakarta mengambil tindakan pencegahan yang kurang tepat untuk mengatasi kasus DBD. Oleh karena itu nilai β diganti dengan nilai sebagai berikut

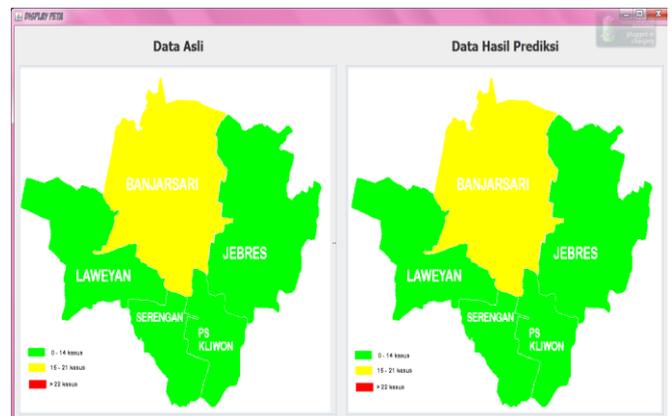
- β Kecamatan Laweyan = 0,0193
- β Kecamatan Jebres = 0,0192
- β Kecamatan Banjarsari = 0,0194
- β Kecamatan Pasar Kliwon = 0,0193
- β Kecamatan Serengan = 0,0195

Dengan mengganti nilai β tersebut maka jumlah data hasil perhitungan akan berubah seperti pada Tabel 6 berikut ini

Tabel 6. Jumlah Kasus DBD Tahun 2013 Bulan Ke-6

| Kecamatan | Data asli | Data Hasil Prediksi |
|--------------|-----------|---------------------|
| Laweyan | 8 | 8 |
| Jebres | 4 | 7 |
| Banjarsari | 18 | 18 |
| Pasar Kliwon | 5 | 7 |
| Serengan | 7 | 7 |

Kemudian hasil tampilan peta untuk data tersebut ditunjukkan pada Gambar 7.



Gambar 7. Peta Hasil Simulasi Tahun 2013 Bulan Ke-6

Dengan mengganti nilai ternyata lebih sesuai untuk memodelkan data tahun 2013 bulan ke-6. Jumlah kasus pada data hasil prediksi tidak jauh di bawah jumlah kasus DBD pada data asli dan peta data asli dan data hasil prediksi menunjukkan kesamaan warna, sehingga model SIR dengan

nilai β tersebut dapat digunakan untuk memodelkan data jumlah kasus DBD di Kota Surakarta pada tahun 2013 bulan ke-6.

7. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa peta untuk simulasi dan prediksi jumlah penderita DBD di Kota Surakarta dapat dibuat dengan metode SIR menggunakan nilai β antara 0,019 – 0,020.

Untuk penelitian selanjutnya, peta data prediksi penderita penyakit DBD dapat dikembangkan untuk cakupan wilayah provinsi dan untuk penyakit menular lainnya yang menyebar di masyarakat supaya mempermudah pihak pengelola kesehatan untuk membuat kebijakan pencegahan dan pengobatan penyakit menular.

8. DAFTAR PUSTAKA

- [1] Wati, W. E., Astuti., D., & Darnoto, S. (2009). Beberapa Faktor yang Berhubungan dengan Kejadian Demam Berdarah Dengue (DBD) di Kelurahan Ploso Kecamatan Pacitan Tahun 2009. *Jurnal Vektora Vol. III No 1*.
- [2] Dini, A. M., Fitriany, R. N., & Wulandari, R. A. (2010). Faktor Iklim dan Angka Insiden Demam Berdarah Dengue di Kabupaten Serang. *Makara, Kesehatan, Vol. 14, NO. 1*, 31-38.
- [3] Dinas Kesehatan Jawa Tengah. (2013). *Buku Profil Kesehatan Provinsi Jawa Tengah*. Semarang.
- [4] Tamrin, H., Riyanto, M. Z., & Akhid, A. A. (2007). *Model SIR Penyakit Tidak Fatal*. Yogyakarta: Universitas Gadjah Mada.
- [5] Side, S., & Noorani, S. M. (2013). A SIR Model for Spread of Dengue Fever Disease (Simulation for South Sulawesi, Indonesia and Selangor, Malaysia). *World Journal of Modelling and Simulation Vol. 9 (2013) No. 2, ISSN 1 746-7233, England, UK*, 96-105.
- [6] Syarippudin. (2009). Model Simulasi Penyebaran Penyakit Epidemis Jenis SIR. *Bioprospek, Volume 6, Nomor II, September 2009, ISSN: 1829-7226*.
- [7] Tjolleng, A., Komalig, H. A., & Prang, J. D. (2013). Dinamika Perkembangan HIV/AIDS di Sulawesi Utara Menggunakan Model Persamaan Diferensial Nonlinear SIR. *Jurnal Ilmiah Sains Vol. 13 N.1*, 9-14.
- [8] Fredlina, K. Q., Oka, T. B., & Dwipayana, M. E. (2012). Model SIR (Susceptible-Infectious-Recovered) untuk Penyebaran Penyakit Tuberkulosis. *e-Jurnal Matematika Vol. 1 No. 1*, 52-58.
- [9] Widi, C. A., Nataliani, Y., & Hendry. (2011). Deteksi dan Prediksi Daerah Endemis Demam Berdarah Dengue (DBD) dengan Pemodelan Matematis Susceptible, Infected, Recovered (SIR) (Studi Kasus: Kabupaten Semarang). *Jurnal Teknologi Informasi-Aiti, Vol. 8. No.2, Agustus 2011*, 177-189.
- [10] Piccolo, C III., & Billings, L. (2005). The Effect of Vaccinations in an Immigrant Model. *Mathematical and Computer Modelling 42*, 291-299.
- [11] Palgunadi, S., & Herlambang, T. (2014). Simulasi Penyebaran Penyakit ISPA (Infeksi Saluran Pernafasan Akut) pada Balita di Kota Surakarta Menggunakan Game Of Life. *Prosiding SNST ke-5 Tahun 2014*, 17-22.
- [12] Luknanto, D. (2003). *Model Matematika Numerik*. Yogyakarta: Universitas Gadjah Mada.
- [13] Murray, J. D. (2002). *Mathematical Biology : An Introduction. Third Edition*. New York, Berlin: Springer-Verlag, Heidelberg.
- [14] Handel, A., Longini, I. M., & Antia, J. a. (2006). What is the best control strategy for multiple infectious disease outbreaks? *Proceedings of The Royal Society. Proc. R. Soc. B (2007) 274*, 833-837.
- [15] Abramson, G. (2001). *A series of lectures given at PANDA, UNM*.
- [16] Dimitrov, N. B., & Meyers, L. A. (2010). Mathematical Approaches to Infectious Disease. *Tutorials in Operation Research INFORMS 2010 ISBN 978-0-9843378-0-4*.
- [17] Choisy, M., Guégan, J.-F., & Rohani, P. (2007). Mathematical Modeling of Infectious Diseases Dynamics. In M.Tibayrenc, *Encyclopedia of Infectious Diseases: Modern Methodologies* (pp. 379-404). New Jersey: John Wiley & Sons, Inc.
- [18] Nicho, J. (2010). The SIR Epidemiology Model in Predicting Herd Immunity. *Undergraduate Journal of Mathematical Modeling: One + Two: Vol. 2: Iss. 2, Article 8*, 833-837.
- [19] Sasongko, S. B. (2010). *Metode Numerik dengan Scilab*. Yogyakarta: ANDI.

Penentuan Model Terbaik pada Metode *Naive Bayes Classifier* dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter

Apriliya Fitri Cahyanti
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta

apriilya.fc@student.uns.ac.id

Ristu Saptono
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta

ristu.saptono@staff.uns.ac.id

Sari Widya Sihwi
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan
Surakarta

sari.widya.sihwi@gmail.com

ABSTRAK

Untuk proses klasifikasi dalam studi kasus penilaian status gizi balita menggunakan metode *Naive Bayes Classifier*, asumsi independensi antar parameter perlu diperhitungkan. Independensi antar parameter dilihat dari korelasi antar parameter yang digunakan. Artikel ini membahas mengenai uji korelasi antar parameter dalam studi kasus penilaian status gizi menggunakan metode *Cosine Similarity*. Kemudian hasil uji korelasi tersebut dijadikan prosedur penentuan model dalam metode *Naive Bayes Classifier*. Sehingga dapat diketahui model yang paling baik dalam penilaian status gizi menggunakan metode *Naive Bayes Classifier*. Penentuan model terbaik dilihat dari akurasi, kesederhanaan, waktu, dan akuisisi data pada model. Pada skenario data 60%:40%, model terbaik ditunjukkan oleh model yang terdiri dari parameter berat, bmi, dan umur, dengan akurasi sebesar 94,4%. Sedangkan pada skenario data 80%:20% model terbaik ditunjukkan pada model yang terdiri dari parameter berat, bmi, tinggi, umur, dan jenis kelamin, dengan akurasi 94,8%. Penelitian ini menunjukkan bahwa korelasi parameter mempengaruhi hasil klasifikasi. Penggunaan parameter independen belum tentu menghasilkan akurasi yang maksimal. Bahkan, model terbaik yang dipilih terdiri dari parameter dependen.

Kata Kunci — *Cosine Similarity, Naive Bayes Classifier, Penilaian Status Gizi.*

1. PENDAHULUAN

Naive Bayes Classifier merupakan metode *classifier* yang berdasarkan probabilitas dan teorema Bayes dengan asumsi keindependenan atribut [1]. Asumsi independensi atribut akan menghilangkan kebutuhan banyaknya jumlah data latih dari seluruh atribut yang dibutuhkan untuk mengklasifikasi suatu data. Data latih untuk Teorema Bayes membutuhkan paling tidak perkalian kartesius dari seluruh kelompok atribut yang mungkin, sehingga semakin sedikit atribut yang digunakan, akan mengurangi data latih yang dibutuhkan. Padahal, pada kenyataannya asumsi atribut independen pada *Naive Bayes Classifier* sering dilanggar [2]. Hal ini disebabkan karena asumsi keindependenan atribut dalam dunia nyata hampir tidak pernah terjadi [3].

Independensi parameter sendiri ditentukan berdasarkan nilai korelasi atau kemiripan parameter yang digunakan pada klasifikasi. Dengan menggunakan metode *Cosine Similarity* dapat diketahui similaritas antar parameter, dan korelasi antar parameter tersebut, sehingga dapat menentukan independensi antar parameter. Pada penelitian yang dilakukan oleh Domingos dan Pazzani, ternyata menghasilkan *Naive Bayes* memiliki performa yang baik meskipun di dalamnya terdiri dari atribut yang dependen. [2]

Dalam penelitian ini penulis menggunakan studi kasus mengenai penilaian status gizi. Penilaian status gizi sendiri menggunakan parameter berupa ukuran tubuh seperti berat badan, tinggi badan, lingkaran lengan atas dan tebal lemak di bawah kulit [4]. Penilaian status gizi menggunakan parameter yang dibuat menjadi kategorikal, sehingga masing – masing parameter saling berkaitan. Penelitian ini membahas mengenai korelasi antara parameter dan hasil klasifikasi, dan uji independensi antar parameter menggunakan metode *Cosine Similarity*. Penulis ingin mengeksplor lebih lanjut bagaimana keoptimalan *Naive Bayes Classifier* pada studi kasus penilaian status gizi dengan mempertimbangkan independensi parameter, dengan pemodelan data menggunakan *Backward Feature Selection*. Pemodelan data *Backward Feature Selection* dilakukan dengan menghapus fitur, yang apabila fitur tersebut dihapus dapat meningkatkan akurasi sistem. Selanjutnya, dipilih model yang paling baik dengan mempertimbangkan akurasi, kesederhanaan model, waktu eksekusi, dan akuisisi data model.

2. LANDASAN TEORI

2.1. *Naive Bayes Classifier*

Naive Bayesian Classifier merupakan klasifikasi dengan model statistik untuk menghitung peluang dari suatu kelas yang memiliki masing – masing kelompok atribut yang ada, dan menentukan kelas mana yang paling optimal. Pada metode ini semua atribut akan memberikan kontribusinya dalam pengambilan keputusan, dengan bobot atribut yang sama penting dan setiap atribut saling bebas satu sama lain [5].

Dasar dari teorema *Naive Bayes Classifier* yang dipakai dalam pemrograman adalah rumus Bayes sebagai berikut [8].

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \tag{1}$$

Dimana

- P(H|X) = probabilitas posterior H di dalam X
- P(X|H) = Probabilitas posterior X di dalam H
- P(H) = Probabilitas prior dari H
- P(X) = Probabilitas prior dari X

Dalam sebuah dataset yang besar, pemilihan data *training* secara *random* akan menyebabkan kemungkinan adanya nilai nol dalam model probabilitas. Nilai nol ini akan menyebabkan *Naive Bayes Classifier* tidak dapat mengklasifikasi sebuah data inputan. Oleh karena itu diperlukan suatu metode *smoothing* yang dapat menghindari adanya nilai nol dalam model probabilitas. *Laplacian Smoothing* merupakan metode *smoothing* yang biasa digunakan dalam *Naive Bayes Classifier*. *Laplacian Smoothing* biasa dikenal dengan nama *add one smoothing*, karena dalam perhitungannya, setiap variabel pada masing – masing parameter ditambahkan 1. Persamaan *Laplace Smoothing* dituliskan pada persamaan (2).

$$P(x_k|C) = \frac{P(x_k|C)+1}{P(C)+|V|} \tag{2}$$

Dimana

- $P(x_k|C)$ = probabilitas tiap atribut dari x_k
- $P(C)$ = total jumlah probabilitas dalam x_k
- $|V|$ = jumlah kemungkinan nilai dari x_k .

2.2. Backward Feature Selection

Seleksi fitur atau *feature selection* digunakan untuk memilih fitur sesuai kriteria yang telah ditentukan. Pemodelan data Backward Feature Selection dilakukan dengan menghapus fitur, yang apabila fitur tersebut dihapus dapat meningkatkan akurasi sistem. Selain itu, jika dengan menghapus fitur tersebut berdampak pada penurunan performa, maka fitur tersebut dianggap sangat diperlukan dalam proses klasifikasi [6]. Pada Backward Feature Selection, dimulai dengan menggunakan semua parameter yang ada. Selanjutnya menghapus satu per satu parameter, yang pada setiap penghapusan dapat menurunkan tingkat kesalahan sistem [7].

| Iterasi | Feature Set | Score | Hapus Fitur |
|-----------|-------------|-------|-------------|
| Iterasi 0 | [A B C D] | 0.41 | |
| Iterasi 1 | [A B C] | 0.24 | C |
| | [A B D] | 0.43 | |
| | [A C D] | 0.36 | |
| | [B C D] | 0.19 | |
| Iterasi 2 | [A B] | 0.45 | B |
| | [A D] | 0.58 | |
| | [B D] | 0.29 | |
| Iterasi 3 | [A] | 0.68 | D |
| | [D] | 0.47 | |

Gambar 1. Simulasi *Backward Feature Selection*

Berdasarkan simulasi diatas, pembentukan fitur set dimulai dengan fitur set yang lengkap. Selanjutnya dilakukan penghapusan satu persatu fitur, hingga ditentukan fitur set yang paling baik digunakan adalah A pada iterasi 3 dengan nilai 0.68. Penghapusan fitur B, C, dan D pada iterasi 1 dan iterasi 2 dapat meningkatkan nilai performa.

2.3. Cosine Similarity

Metode *Cosine* digunakan untuk menghitung nilai cosinus sudut antara dua *vector* dan mengukur kemiripan antar dua dokumen [8]. Untuk membandingkan dua parameter A dan B, maka perhitungan *similarity* dapat dihitung dengan persamaan (3).

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \tag{3}$$

Dimana

- A = bobot data yang dibandingkan
- B = bobot data pembanding
- $\|A\|$ = panjang data yang dibandingkan
- $\|B\|$ = panjang data pembanding

Cosine Similarity berpusat x dan y, dan dibatasi antara -1 dan 1. Namun, nilai cosinus dibatasi 0 dan 1 jika x dan y bernilai positif. Menurut O’Connor, *Cosine Similarity* dapat menunjukkan korelasi. Pada umumnya *Cosine Similarity* dibahas dalam hal sudut vektor, tetapi dapat dianggap sebagai korelasi, jika vektor yang dibandingkan merupakan suatu data yang berpasangan [9]. Dua parameter dengan nilai similaritas besar menunjukkan parameter yang saling berkorelasi.

2.4. Korelasi

Informasi korelasi diketahui berdasarkan pada nilai kemiripan [10]. Semakin besar nilai kemiripan dari kedua objek, maka menunjukkan hubungan atau korelasi parameter yang kuat. Sedangkan parameter yang memiliki korelasi tinggi menunjukkan parameter yang dependen. Kekuatan korelasi antar parameter ditunjukkan oleh koefisien korelasi.

2.5. Stratified Random Sampling atau Holdout

Metode *Holdout* merupakan metode yang menyediakan sejumlah data untuk digunakan sebagai data *testing* dan sisanya sebagai data *training* [11]. Dalam penelitian ini data dibagi menjadi dua kelompok, yaitu data *training* dan data pengujian yang dibagi secara acak. Data pelatihan digunakan untuk memperoleh model, sedangkan data pengujian digunakan untuk mengestimasi akurasi.

Dalam pengacakan data *training* dan *testing* ini ada kemungkinan akan menghasilkan data yang tidak proporsional dalam tiap klasifikasi. Misalnya satu klasifikasi data lebih dominan jika dibandingkan klasifikasi yang lain. Oleh karena itu digunakan metode stratified random sampling untuk pengacakan data untuk menghasilkan data *training* dan *testing* yang proporsional.

2.6. Penilaian Status Gizi

Antropometri sebagai indikator status gizi dapat dilakukan dengan mengukur beberapa parameter. Parameter adalah ukuran tunggal dari tubuh manusia yaitu umur, berat badan, tinggi badan, lingkaran lengan atas, lingkaran dada, lingkaran panggul, dan tebal lemak di bawah kulit [4]. Standar antropometri pada balita berbeda untuk tiap jenis kelamin, baik laki – laki maupun perempuan. Berdasarkan berat dan tinggi anak, dapat diketahui *Body Mass Index* (BMI) yang juga dapat menentukan nilai status gizi anak tersebut.

Penggunaan *Body Mass Index* (BMI) pada umumnya hanya berlaku untuk orang dewasa. Pada anak – anak pengukuran BMI sangat terkait dengan umurnya, karena dengan perubahan umur terjadi perubahan komposisi tubuh. Karena itu, pada anak – anak digunakan indeks BMI menurut umur (BMI/U). BMI dapat dihitung dengan rumus:

$$BMI = \frac{\text{berat badan (kg)}}{\text{tinggi badan}^2(m)} \quad (4)$$

Selain itu, menurut [4], faktor yang mempengaruhi keadaan gizi yaitu konsumsi makanan dan tingkat kesehatan. Sedangkan konsumsi makanan sendiri dipengaruhi oleh pendapatan orang tua, makanan, dan tersedianya bahan makanan. Terdapat hubungan antara tingkat pendidikan dan pendapatan keluarga terhadap status gizi balita [13]. Pendapatan keluarga sendiri menyebabkan keluarga masuk ke kategori keluarga miskin (gakin) atau tidak).

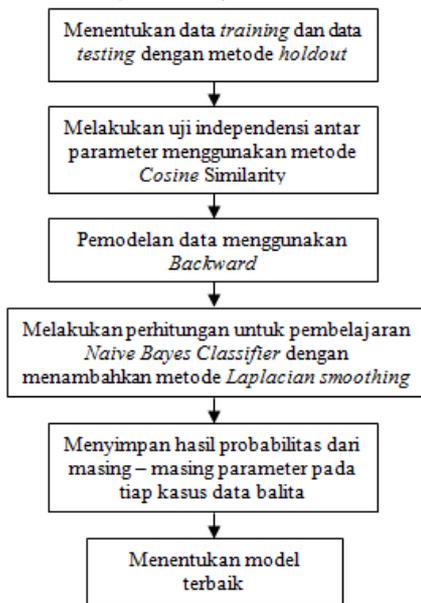
3. METODOLOGI PENELITIAN

3.1. Pengumpulan Data

Dataset yang digunakan merupakan data posyandu pada bulan September 2013 yang diambil dari puskesmas kelurahan Cangkreng, Kecamatan Purworejo, Kabupaten Purworejo. Jumlah seluruh dataset yaitu 250 data.

3.2. Pemodelan Data

Pemodelan data dilakukan untuk mengetahui alur bagaimana proses – proses dan metode berjalan sebelum diimplementasikan ke dalam sebuah aplikasi dengan data yang ada. Berikut tahap pemodelan data untuk penentuan status gizi menggunakan *Naïve Bayes Classifier*.



Gambar 2. Alur Pemodelan Data

Dalam proses pengkategorian parameter menurut Kementerian Kesehatan, beberapa kategori parameter dipengaruhi oleh parameter yang lain [12]. Kategori berat badan ditentukan oleh umur dalam bulan, jenis kelamin, dan berat dalam kg. Pada kategori tinggi badan, ditentukan oleh umur dalam bulan, jenis kelamin, dan tinggi dalam cm. Sedangkan untuk kategori BMI (*Body Mass Index*) ditentukan oleh umur dalam bulan, jenis kelamin, berat dalam kg, dan tinggi dalam cm. Untuk label pengkategorian dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 Diskritisasi Parameter

| No | Parameter | Kategori | Label |
|----|-------------|-------------|-------|
| 1 | Status gizi | Gizi buruk | 0 |
| | | Gizi kurang | 1 |

| | | | |
|---|---|---------------|---|
| | | Gizi baik | 2 |
| | | Gizi Lebih | 3 |
| 2 | Jenis Kelamin | Perempuan | 0 |
| | | Laki-Laki | 1 |
| 3 | Status ekonomi keluarga (Keluarga Miskin/Gakin) | Ya | 0 |
| | | Tidak | 1 |
| 4 | Berat Badan | Sangat Kurang | 0 |
| | | Kurang | 1 |
| | | Baik | 2 |
| | | Lebih | 3 |
| 5 | Tinggi Badan | Sangat Pendek | 0 |
| | | Pendek | 1 |
| | | Normal | 2 |
| | | Tinggi | 3 |
| 6 | <i>Body Mass Index</i> (BMI) | Sangat Kurus | 0 |
| | | Kurus | 1 |
| | | Normal | 2 |
| | | Gemuk | 3 |
| 7 | Umur (bulan) | 0 sd. 60 | |

Penelitian ini mengasumsikan data yang digunakan adalah bersifat kategorikal, sehingga untuk data yang bertipe kontinu harus melalui *preprocessing* diskritisasi untuk menghasilkan data yang bersifat kategorikal. Angka binary yang digunakan untuk label kategorikal ini dimulai dari 0. Untuk parameter umur sendiri, tidak dilakukan diskritisasi, karena nilai umur yaitu 0 sampai 60 sudah dianggap bersifat kategorikal. Untuk parameter yang lain, yaitu jenis kelamin, status ekonomi, berat badan, tinggi badan, dan BMI, pengkategorian berdasarkan pada standar antropometri yang ditetapkan oleh Kementerian Kesehatan RI.

3.2.1 Pengacakan Data menggunakan Metode Holdout

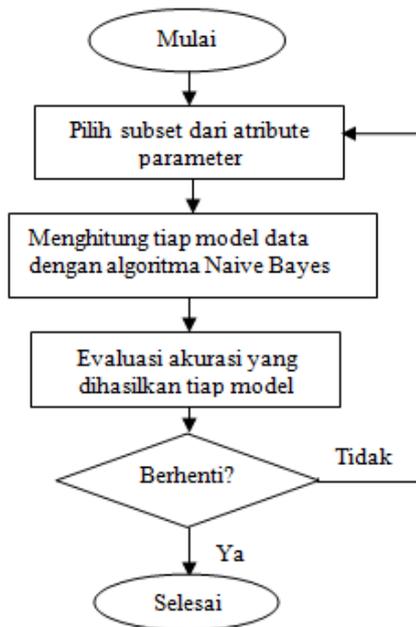
Untuk pembagian data *training* dan data *testing* dilakukan secara acak dengan persentase 60% untuk data *training* dan 40% untuk data *testing*, serta 80% untuk data *training* dan 20% untuk data *testing*. Untuk masing – masing skenario, dilakukan 5 kali percobaan.

3.2.2 Uji Korelasi Parameter dengan Cosine Similarity

Uji korelasi dilakukan pada tiap parameter yang digunakan (jenis kelamin, umur, berat, tinggi, dan gakin) terhadap parameter status gizi. Dari hasil uji korelasi tersebut, parameter *independent* yang memiliki nilai *similarity* terbesar berarti memiliki pengaruh paling tinggi terhadap parameter *dependent* (status gizi). Selanjutnya, melakukan uji korelasi antar parameter menggunakan *Cosine* untuk menentukan independensi antar parameter.

3.2.3 Pembentukan Model Data

Uji independensi parameter menentukan model data yang akan diujikan menggunakan metode *Naïve Bayes Classifier*. Model data ini dibentuk berdasarkan urutan similaritas parameter yang tersedia. Prosedur penentuan model dilakukan secara *Backward Feature Selection* dengan menghilangkan satu persatu parameter mulai dari parameter yang memiliki similaritas terkecil.



Gambar 3. Alur Pembentukan Model Data

3.3. Pengembangan Sistem Aplikasi

Pada penelitian ini dilakukan pengembangan suatu aplikasi untuk menentukan model yang paling baik dalam penilaian status gizi balita dengan menggunakan bahasa pemrograman PHP dan database MySql.

3.4. Pengujian dan Analisis Hasil

Untuk mengetahui kinerja dari sistem yang dibuat, dilakukan pengujian dengan membandingkan hasil klasifikasi menggunakan software WEKA dengan hasil klasifikasi pada sistem. Hasil yang dibandingkan adalah hasil akurasi dari kedua sistem. Jika hasil model terbaik dari WEKA dan sistem sama, maka dapat dikatakan hasil sistem benar.

Dalam percobaan, dataset yang ada dibagi menjadi data *training* dan data *testing*. Terdapat dua prosedur dalam pembagian data *training* dan *testing* dalam persentase data yang berbeda. Dengan adanya prosedur yang berbeda tersebut dapat dilihat performa metode yang diuji berdasarkan masing – masing prosedur.

Tabel 3.3 Skenario Pengujian Data

| Percobaan | Persentase Data | |
|-------------|-----------------|----------------|
| | <i>Training</i> | <i>Testing</i> |
| Percobaan 1 | 60% | 40% |
| Percobaan 2 | 80% | 20% |

Untuk mengukur tingkat akurasi dari metode *Naive Bayes Classifier* dihitung menggunakan persamaan (5).

$$Akurasi = \frac{\text{jumlah prediksi benar}}{\text{jumlah Data}} \times 100\% \quad (5)$$

Selanjutnya parameter lain yang diukur adalah waktu eksekusi dari sistem dalam satuan detik. Selain itu, untuk menentukan model terbaik dilihat akuisisi parameter yang terdapat pada model. Akuisisi parameter yaitu penentuan parameter lapangan yang cocok dengan daerah survey.

4. HASIL DAN PEMBAHASAN

4.1. Deskripsi Data

Data yang digunakan dalam penelitian ini sebanyak 250 data balita. Data tersebut diambil dari data posyandu Puskesmas Kecamatan Purworejo. Data balita yang disimpan terdiri dari parameter umur, jenis kelamin, berat, tinggi, bmi, dan gakin.

4.2. Pemodelan Data

4.1.1. Uji Korelasi Parameter dengan *Cosine Similarity*

Dari beberapa parameter yang digunakan, terlebih dahulu dilakukan uji korelasi dengan menghitung nilai *similarity* dari tiap parameter terhadap status gizi. Nilai *similarity* yang besar berarti nilai independensinya kecil. Uji korelasi ini dilakukan menggunakan metode *Cosine Similarity*, dan diurutkan berdasarkan nilai *similarity* paling besar. Karena *Cosine Similarity* tidak memiliki batas untuk koefisien korelasi, maka penulis memberikan batasan sebesar 0,4 sebagai batas untuk parameter dependen.

Tabel 4.1 Uji Korelasi Status Gizi Dengan Semua Parameter pada Skenario 60%:40%

| Parameter dependent | Parameter independent | Similarity |
|---------------------|-----------------------|------------|
| Status gizi | berat | 0.99333 |
| Status gizi | bmi | 0.97072 |
| Status gizi | tinggi | 0.88979 |
| Status gizi | umur | 0.80056 |
| Status gizi | jenis kelamin | 0.64215 |
| Status gizi | gakin | 0.29101 |

Berdasarkan hasil uji korelasi tersebut, terlihat bahwa berat memiliki angka *similarity* yang paling tinggi yaitu 0.99333. Hal ini menunjukkan parameter berat memiliki keterkaitan yang tinggi dengan status gizi. Sedangkan parameter gakin memiliki angka similaritas paling kecil dengan nilai 0,29101. Parameter gakin dapat dikatakan sebagai parameter yang independen terhadap status gizi. Selanjutnya dilakukan uji korelasi antar parameter untuk mengetahui hubungan antar parameter.

Tabel 4.2 Uji Korelasi Antar Parameter pada Skenario 60%:40%

| Parameter | berat | bmi | tinggi | umur | jenis kelamin | gakin |
|---------------|-------|-------|--------|-------|---------------|-------|
| berat | 1 | 0.970 | 0.891 | 0.790 | 0.638 | 0.281 |
| bmi | 0.970 | 1 | 0.862 | 0.848 | 0.671 | 0.329 |
| tinggi | 0.891 | 0.862 | 1 | 0.755 | 0.580 | 0.329 |
| umur | 0.790 | 0.848 | 0.755 | 1 | 0.619 | 0.370 |
| jenis kelamin | 0.638 | 0.671 | 0.580 | 0.619 | 1 | 0.314 |
| gakin | 0.281 | 0.329 | 0.329 | 0.370 | 0.314 | 1 |

Hasil uji korelasi antar parameter dapat dilihat pada tabel 4.2. Urutan parameter yang memiliki similaritas paling tinggi dengan parameter berat adalah BMI, tinggi, umur, jenis kelamin, kemudian gakin. Nilai similaritas bmi adalah 0.970, dan nilai similaritas gakin adalah 0.281.

Tabel 4.3 Uji Korelasi Status Gizi Dengan Semua Parameter pada Skenario 80%:20%

| Parameter dependent | Parameter independent | Similarity |
|---------------------|-----------------------|------------|
| Status gizi | berat | 0.99321 |
| Status gizi | bmi | 0.97226 |
| Status gizi | tinggi | 0.88111 |
| Status gizi | umur | 0.79924 |
| Status gizi | jenis kelamin | 0.65397 |
| Status gizi | gakin | 0.27398 |

Tabel 4.3 diatas menunjukkan hasil uji korelasi status gizi dengan semua parameter yang dilakukan pada skenario 80%:20%. Sama halnya dengan skenario 60%:40%, pada skenario ini juga menunjukkan parameter berat memiliki nilai similaritas tertinggi dengan status gizi, dengan nilai 0.99321. Urutan parameter dari nilai similaritas tertinggi dari kedua skenario juga sama, yaitu berat, bmi, tinggi, umur, jenis kelamin, kemudian gakin.

Tabel 4.4 Uji Korelasi Antar Parameter pada Skenario 80%:20%

| Parameter | berat | bmi | tinggi | umur | jenis kelamin | gakin |
|---------------|-------|-------|--------|-------|---------------|-------|
| berat | 1 | 0.971 | 0.884 | 0.789 | 0.651 | 0.272 |
| bmi | 0.971 | 1 | 0.861 | 0.852 | 0.681 | 0.298 |
| tinggi | 0.884 | 0.861 | 1 | 0.764 | 0.572 | 0.296 |
| umur | 0.789 | 0.852 | 0.764 | 1 | 0.607 | 0.348 |
| jenis kelamin | 0.651 | 0.681 | 0.572 | 0.607 | 1 | 0.251 |
| gakin | 0.272 | 0.298 | 0.296 | 0.348 | 0.251 | 1 |

Pada uji korelasi antar parameter pada skenario 80%:20% juga menghasilkan urutan yang sama dengan uji korelasi pada skenario 60%:40%. Urutan parameter berdasarkan nilai similaritas tertinggi terdapat pada parameter bmi, mempunyai nilai sebesar 0.971, kemudian tinggi, umur, jenis kelamin, dan yang terakhir adalah gakin dengan nilai similaritas sebesar 0.272.

4.1.2. Pembentukan Model Data Backward

Uji independensi antar parameter akan membentuk prosedur model data yang akan diujikan menggunakan metode *Naïve Bayes Classifier*. Model ini didapatkan menggunakan metode *backward* dengan mengurangi satu persatu parameter, sehingga hanya tersisa satu parameter. Model data yang

dihasilkan berdasarkan proses uji independensi adalah sebanyak 21 model.

Pemodelan data dibagi menjadi enam bagian berdasarkan jumlah parameter. Pertama terdiri dari enam parameter atau parameter lengkap yaitu model 1. Yang kedua terdiri dari lima parameter yaitu model 2, 3, 4, 5, 6, dan 7. Bagian ketiga terdiri dari empat parameter yaitu model 8, 9, 10, 11, dan 12. Bagian keempat terdiri dari tiga parameter yaitu model 13, 14, 15, dan 16. bagian kelima terdiri dari dua parameter yaitu model 17, 18, dan 19. Sedangkan bagian terakhir terdiri dari satu parameter yaitu model 20 dan 21.

Tabel 4.5 Model Data pada Skenario 60%:40%

| Model | Parameter yang digunakan | Akurasi |
|-------|--|---------|
| 1 | berat, bmi, tinggi, umur, jenis kelamin, gakin | 96.4 |
| 2 | berat, bmi, tinggi, umur, jenis kelamin | 96.533 |
| 3 | berat, bmi, tinggi, umur, gakin | 96.4 |
| 4 | berat, bmi, tinggi, jenis kelamin, gakin | 95.733 |
| 5 | berat, bmi, umur, jenis kelamin, gakin | 96.266 |
| 6 | berat, tinggi, umur, jenis kelamin, gakin | 95.6 |
| 7 | bmi, tinggi, umur, jenis kelamin, gakin | 86.133 |
| 8 | berat, bmi, tinggi, umur | 96.666 |
| 9 | berat, bmi, tinggi, jenis kelamin | 95.866 |
| 10 | berat, bmi, umur, jenis kelamin | 96.266 |
| 11 | berat, tinggi, umur, jenis kelamin | 95.734 |
| 12 | bmi, tinggi, umur, jenis kelamin | 85.6 |
| 13 | berat, bmi, tinggi | 95.733 |
| 14 | berat, bmi, umur | 96.4 |
| 15 | berat, tinggi, umur | 95.866 |
| 16 | bmi, tinggi, umur | 86.133 |
| 17 | berat, bmi | 95.2 |
| 18 | berat, umur | 96 |
| 19 | bmi, umur | 81.866 |
| 20 | berat | 95.066 |
| 21 | umur | 75.6 |

Berdasarkan tabel 4.5, model yang dipilih sebagai model terbaik dari masing-masing bagian adalah model 1, 2, 8, 14, 17, dan 20. Dari keenam model tersebut dapat dilihat bagaimana pengaruh keberadaan parameter tidak independen (bmi, tinggi, jenis kelamin, umur) dan parameter independen (gakin). Pengaruh parameter tidak independen dan parameter independen ini dapat dilihat dari nilai akurasi masing-masing model.

1. Model 1 yang menggunakan semua parameter ketika dibandingkan dengan model 2 yang menghilangkan parameter jenis kelamin menunjukkan bahwa gakin memiliki pengaruh sebesar 0,13334% terhadap status gizi.
2. Model 8 dengan menghilangkan parameter jenis kelamin mengalami peningkatan 0,13334% dari model 2. Hal ini menunjukkan parameter jenis kelamin memiliki pengaruh 0,13334% terhadap status gizi.
3. Model 14 dengan menghilangkan parameter tinggi mengalami penurunan 0,26668% dari model 8. Hal ini menunjukkan bahwa parameter tinggi memiliki pengaruh 0,26668% terhadap status gizi.
4. Model 18 dengan menghilangkan parameter bmi mengalami penurunan akurasi 0,4% dari model 14. Hal

ini menunjukkan bahwa parameter bmi memiliki pengaruh 0,4% terhadap status gizi.

- Model 20 dengan menghilangkan parameter umur mengalami peningkatan akurasi 0,93334% dari model 18. Hal ini menunjukkan bahwa parameter umur memiliki pengaruh 0,93334% terhadap status gizi.

Tabel 4.6 Model Data pada Skenario 80%:20%

| Model | Parameter yang digunakan | Akurasi |
|-------|--|---------|
| 1 | berat, bmi, tinggi, umur, jenis kelamin, gakin | 95,7 |
| 2 | berat, bmi, tinggi, umur, jenis kelamin | 95,8 |
| 3 | berat, bmi, tinggi, umur, gakin | 95,8 |
| 4 | berat, bmi, tinggi, jenis kelamin, gakin | 95,4 |
| 5 | berat, bmi, umur, jenis kelamin, gakin | 95,7 |
| 6 | berat, tinggi, umur, jenis kelamin, gakin | 94,9 |
| 7 | bmi, tinggi, umur, jenis kelamin, gakin | 87,3 |
| 8 | berat, bmi, tinggi, umur | 95,9 |
| 9 | berat, bmi, tinggi, jenis kelamin | 95,4 |
| 10 | berat, bmi, umur, jenis kelamin | 95,7 |
| 11 | berat, tinggi, umur, jenis kelamin | 95 |
| 12 | bmi, tinggi, umur, jenis kelamin | 86,4 |
| 13 | berat, bmi, tinggi | 95,5 |
| 14 | berat, bmi, umur | 95,9 |
| 15 | berat, tinggi, umur | 95,1 |
| 16 | bmi, tinggi, umur | 85,9 |
| 17 | berat, tinggi | 95,4 |
| 18 | berat, umur | 95,2 |
| 19 | tinggi, umur | 82,7 |
| 20 | berat | 94,5 |
| 21 | tinggi | 73,7 |

Berdasarkan tabel 4.6, model 1, 2, 8, 15, 17, dan 20 merupakan model yang dipilih sebagai model terbaik dari masing-masing bagian. Seperti halnya pada skenario 60%:40%, pada skenario 80%:20% ini juga terlihat bagaimana pengaruh parameter tidak independen dan parameter independen terhadap akurasi model. Pengaruh parameter tersebut dapat dijelaskan sebagai berikut.

- Model 1 yang terdiri dari semua parameter jika dibandingkan dengan model 2 dengan menghilangkan parameter gakin mengalami peningkatan 0,1%. Hal ini menunjukkan bahwa gakin memiliki pengaruh 0,1% terhadap status gizi.
- Model 8 dengan menghilangkan parameter jenis kelamin mengalami peningkatan akurasi dari model 2 sebesar 0,1%. Hal ini menunjukkan bahwa parameter jenis kelamin memiliki pengaruh 0,1% terhadap status gizi.
- Model 14 dengan menghilangkan parameter tinggi tidak mengalami perubahan akurasi dari model 8. Hal ini menunjukkan parameter tinggi tidak memiliki pengaruh terhadap status gizi.
- Model 18 dengan menghilangkan parameter bmi mengalami peningkatan 0,7% dari model 14. Hal ini menunjukkan parameter bmi memiliki pengaruh 0,7% terhadap status gizi.
- Model 20 dengan menghilangkan parameter umur mengalami penurunan akurasi 0,7% dari model 18. Hal ini menunjukkan parameter umur memiliki pengaruh 0,7% terhadap status gizi.

4.3. Hasil dan Evaluasi

4.3.1. Penentuan Model Terbaik

Dalam menentukan model terbaik, dianalisa berdasarkan akurasi, waktu, dan akuisisi dari model. Model yang dipilih dilihat dari nilai akurasi, kesederhanaan model, waktu eksekusi, dan akuisisi data. Akuisisi dari model dilihat dari komponen parameter yang ada di dalam model tersebut. Akuisisi parameter sendiri merupakan penentuan parameter lapangan yang cocok dengan daerah survey. Dalam kasus ini, parameter gakin kurang cocok digunakan, karena keadaan gakin pada balita sulit untuk ditentukan. Sedangkan untuk parameter berat, tinggi, umur, jenis kelamin, dan bmi, parameter tersebut saling terkait satu dengan yang lain, sehingga akuisisi data parameter tersebut tetap diperlukan sebagai data awal balita.

Tabel 4.7 Model Data pada Skenario 60%:40%

| Model | Parameter | Training | | Testing | |
|-------|--|----------|-------|---------|-------|
| | | Akurasi | Waktu | Akurasi | Waktu |
| 1 | berat,BMI,tinggi,umur,jenis kelamin, gakin | 96,4 | 0,41 | 94 | 0,157 |
| 2 | berat,BMI,tinggi,umur,jenis kelamin | 96,53334 | 0,281 | 94 | 0,147 |
| 3 | berat,BMI,tinggi,umur,gakin | 96,4 | 0,282 | 94,2 | 0,147 |
| 4 | berat,BMI,tinggi,jenis kelamin,gakin | 95,73332 | 0,293 | 93,4 | 0,122 |
| 5 | berat,BMI,umur,jenis kelamin,gakin | 96,26666 | 0,323 | 94,2 | 0,141 |
| 6 | berat,tinggi,umur,jenis kelamin,gakin | 95,6 | 0,315 | 92 | 0,143 |
| 7 | BMI,tinggi,umur,jenis kelamin,gakin | 86,13334 | 0,319 | 76,6 | 0,138 |
| 8 | berat,BMI,tinggi,umur | 96,66668 | 0,276 | 94,4 | 0,128 |
| 9 | berat,BMI,tinggi,jenis kelamin | 95,86666 | 0,225 | 93,6 | 0,125 |
| 10 | berat,BMI,umur,jenis kelamin | 96,26666 | 0,247 | 94,2 | 0,137 |
| 11 | berat,tinggi,umur,jenis kelamin | 95,73334 | 0,283 | 92 | 0,138 |
| 12 | BMI,tinggi,umur,jenis kelamin | 85,6 | 0,29 | 76,8 | 0,129 |
| 13 | berat,BMI,tinggi | 95,73332 | 0,206 | 93,4 | 0,096 |
| 14 | berat,BMI,umur | 96,4 | 0,237 | 94,4 | 0,121 |
| 15 | berat,tinggi,umur | 95,86666 | 0,237 | 92 | 0,121 |
| 16 | BMI,tinggi,umur | 86,13332 | 0,25 | 76,6 | 0,116 |
| 17 | berat,BMI | 95,2 | 0,182 | 94,2 | 0,089 |
| 18 | berat,umur | 96 | 0,211 | 93 | 0,108 |
| 19 | BMI,umur | 81,86668 | 0,214 | 81 | 0,109 |
| 20 | berat | 95,06666 | 0,14 | 93,4 | 0,082 |
| 21 | umur | 76,6667 | 0,185 | 66 | 0,096 |

Berdasarkan tabel 4.7 di atas, pada data training 60%, model 8 memiliki akurasi paling tinggi, yaitu 96,66668%. Namun, model 14 memiliki nilai akurasi yang tidak signifikan dibandingkan model 8 yaitu dengan akurasi 96,4%. Dari segi waktu, model 14 juga memiliki waktu yang lebih singkat dibandingkan model 8. Akuisisi data dari model 14 cocok untuk digunakan dalam klasifikasi status gizi. Pada data *training* dengan presentase 60% ini, model 14 merupakan model yang paling baik.

Pada data testing, model 8 dan model 14 memiliki nilai akurasi yang paling tinggi, yaitu 94,4%. Model 17 memiliki nilai akurasi tinggi yang tidak berbeda signifikan dengan model 8 dan 14 yaitu sebesar 94,2%. Namun, model 17 ketika dihitung menggunakan data training memiliki akurasi yang tidak maksimal. Dari segi akuisisi data, model 14 memiliki parameter yang cocok untuk klasifikasi status gizi. Meskipun dalam menentukan BMI dibutuhkan jenis kelamin dan tinggi, namun dalam penentuan status gizi hanya digunakan parameter berat,

BMI, dan umur saja. Oleh karena itu, model 14 dijadikan model terbaik untuk kasus skenario data 60%:40%.

Tabel 4.8 Model Data pada Skenario 80%:20%

| Model | Parameter | Training | | Testing | |
|-------|---|----------|-------|---------|-------|
| | | Akurasi | Waktu | Akurasi | Waktu |
| 1 | berat,bmi,tinggi,umur,jenis kelamin,gakin | 95,7 | 0,473 | 94,8 | 0,051 |
| 2 | berat,bmi,tinggi,umur,jenis kelamin | 95,8 | 0,465 | 94,8 | 0,057 |
| 3 | berat,bmi,tinggi,umur,gakin | 95,8 | 0,546 | 94 | 0,053 |
| 4 | berat,bmi,tinggi,jenis kelamin,gakin | 95,4 | 0,445 | 94 | 0,048 |
| 5 | berat,bmi,umur,jenis kelamin,gakin | 95,7 | 0,54 | 93,6 | 0,055 |
| 6 | berat,tinggi,umur,jenis kelamin,gakin | 94,9 | 0,528 | 92,8 | 0,055 |
| 7 | bmi,tinggi,umur,jenis kelamin,gakin | 87,3 | 0,534 | 75,6 | 0,057 |
| 8 | berat,bmi,tinggi,umur | 95,9 | 0,432 | 94,4 | 0,056 |
| 9 | berat,bmi,tinggi,jenis kelamin | 95,4 | 0,383 | 94 | 0,056 |
| 10 | berat,bmi,umur,jenis kelamin | 95,7 | 0,434 | 93,6 | 0,058 |
| 11 | berat,tinggi,umur,jenis kelamin | 95 | 0,43 | 93,2 | 0,057 |
| 12 | bmi,tinggi,umur,jenis kelamin | 86,4 | 0,466 | 75,2 | 0,06 |
| 13 | berat,bmi,tinggi | 95,5 | 0,314 | 93,6 | 0,045 |
| 14 | berat,bmi,umur | 95,9 | 0,366 | 93,6 | 0,058 |
| 15 | berat,tinggi,umur | 95,1 | 0,377 | 94 | 0,06 |
| 16 | bmi,tinggi,umur | 85,9 | 0,363 | 76,4 | 0,052 |
| 17 | berat,bmi | 95,4 | 0,247 | 93,6 | 0,041 |
| 18 | berat,umur | 95,2 | 0,3 | 92,8 | 0,048 |
| 19 | bmi,umur | 82,7 | 0,298 | 81,2 | 0,054 |
| 20 | berat | 94,5 | 0,196 | 94 | 0,043 |
| 21 | umur | 73,7 | 0,204 | 68 | 0,046 |

Berdasarkan tabel 4.8, pada data training, model 8 dan 14 memiliki akurasi yang paling tinggi yaitu 95,9%. Jika dilihat waktu eksekusinya, model 14 memiliki waktu yang relatif lebih singkat daripada model 8. Dari akuisisi data, model 14 terdiri dari parameter yang cocok untuk melakukan klasifikasi. Oleh karena itu, pada skenario data training 80% ini model yang terbaik ditunjukkan oleh model 14.

Pada data testing 20%, model yang memiliki akurasi paling tinggi merupakan model 1 dan model 2 dengan nilai akurasi sebesar 94,8%. Meskipun model 8 memiliki akurasi yang tidak signifikan, namun waktu eksekusi model 8 tidak berbeda jauh dengan model 2. Oleh karena itu, pada skenario data 80%:20% model yang paling baik ditunjukkan pada model 2 dengan parameter berat, BMI, tinggi, umur, dan jenis kelamin. Dari segi akuisisi data, model 2 cocok untuk pengklasifikasian status gizi.

4.3.2. Pengujian WEKA

Pengujian data menggunakan software WEKA hanya dilakukan pada pengujian dengan skenario data testing 40%. Dalam software WEKA juga telah dilakukan lima kali percobaan untuk setiap modelnya, sehingga dihasilkan perbandingan nilai akurasi pada sistem dan WEKA. Jika pada kedua sistem menunjukkan model terbaik yang sama, maka sistem dapat dikatakan berhasil.

Tabel 4.9 Perbandingan Hasil Akurasi *Testing*

| MODEL | Hasil Akurasi Sistem (%) | Hasil Akurasi WEKA |
|-------|--------------------------|--------------------|
| 1 | 94 | 94 |
| 2 | 94 | 94 |
| 3 | 94,2 | 94,4 |
| 4 | 93,4 | 93,6 |
| 5 | 94,2 | 94,4 |
| 6 | 92 | 92 |
| 7 | 76,6 | 76,6 |
| 8 | 94,4 | 94,4 |
| 9 | 93,6 | 94 |
| 10 | 94,2 | 94,2 |
| 11 | 92 | 92,4 |
| 12 | 76,8 | 76,4 |
| 13 | 93,4 | 93,8 |
| 14 | 94,4 | 94,4 |
| 15 | 92 | 92,2 |
| 16 | 76,6 | 77,2 |
| 17 | 94,2 | 94,2 |
| 18 | 93 | 90,4 |
| 19 | 81 | 81,8 |
| 20 | 93,4 | 93,4 |
| 21 | 66 | 68,8 |

Perbandingan hasil akurasi antara WEKA dan system yang dibuat dapat dilihat pada tabel 4.9. Jika dilihat dari perbandingan akurasinya, bisa dikatakan hasil akurasi system sama dengan hasil pada software WEKA. Terdapat perbedaan hasil akurasi pada beberapa model, namun perbedaan nilai yang dihasilkan tersebut tidak berbeda jauh. Untuk hasil model terbaik pada masing-masing bagian antara dua system tersebut ditunjukkan oleh model yang sama, yaitu model 1, 3, 8, 14, 17, dan 20. Model terbaik yang ditunjukkan oleh WEKA dan sistem yang dibuat sama, yaitu model 14 dengan akurasi 94,4%.

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa *Backward Feature Selection* dapat digunakan untuk pembentukan model berdasarkan nilai korelasi menggunakan metode *Cosine Similarity*. Berdasarkan akurasi, waktu eksekusi, dan akuisisi parameter yang digunakan, ditemukan model yang paling baik pada dua skenario percobaan. Pada skenario 60%:40%, model terbaik ditunjukkan oleh model yang terdiri dari parameter berat, bmi dan umur dengan nilai akurasi sebesar 94,4%. Sedangkan pada skenario 80%:20% model terbaik ditunjukkan pada model yang terdiri dari parameter berat, bmi, tinggi, umur, dan jenis kelamin dengan nilai akurasi 94,8%.

Penelitian ini menunjukkan bahwa korelasi parameter mempengaruhi hasil akurasi *Naïve Bayes Classifier*. Oleh karena itu, sebelum melakukan klasifikasi, perlu dilakukan perhitungan uji korelasi antar parameter terlebih dahulu. Selanjutnya dapat diketahui parameter apa yang bisa dihilangkan, sehingga mendapatkan model yang paling sederhana. Penelitian ini memunculkan asumsi baru, bahwa dalam *Naive Bayes Classifier*, sebelum melakukan uji korelasi antar parameter, akan lebih baik dilakukan uji korelasi parameter dengan atribut kelas hasil klasifikasi. Hal ini dikarenakan hubungan parameter dan kelas hasil klasifikasi mempengaruhi pembentukan model. Model terbaik pada skenario 80%:20% diperoleh dengan penghilangan parameter gakin yang memiliki korelasi paling rendah.

Seperti halnya pada penelitian yang dilakukan oleh Domingos dan Pazzani, penelitian ini juga menunjukkan performa yang baik meskipun terdiri dari parameter dependen. Bahkan, dalam penelitian ini model yang terbaik ditunjukkan oleh model yang terdiri dari parameter yang dependen.

Cosine Similarity tidak memiliki batasan untuk korelasinya, sehingga untuk penelitian selanjutnya penulis memberikan saran dalam uji korelasi *Spearman*. Fungsi *Spearman* tersebut memiliki batas untuk koefisien korelasi sehingga lebih jelas dalam menentukan korelasi antar parameter.

6. DAFTAR PUSTAKA

- [1] Berson, A., and Smith S. J. (2001). *Data Warehousing, Data Mining, & OLAP*. New York, NY : McGraw-Hill.
- [2] Domingos, P., and Pazzani, M. (1997). On the optimality of the Simple Bayesian *Classifier* under Zero-One Loss. *Machine Learning*, 29, 103-130
- [3] Shadiq, M. A. (2009). Keoptimalan *Naïve Bayes* dalam Klasifikasi. Bandung: Universitas Pendidikan Indonesia.
- [4] Supariasa, D. N., Bakri, B., & Fajar, I. (2002). *Penilaian Status Gizi*. Jakarta: Penerbit Buku Kedokteran EGC.
- [5] Kusumadewi, S. (2009). Klasifikasi Status Gizi Menggunakan *Naive Bayesian Classification*. *CommIT*, Vol. 3 No. 1, 6-11.
- [6] Abe, S. (2005). Modified Backward Feature Selection. *European Symposium on Artificial Neural Networks*, 163-168.
- [7] Ladha, L., & Deepa, T. (2011). Feature Selection Methods and Algorithms. *International Journal on Computer Science and Engineering (IJCSSE)*, Vol. 3 No.5, 1787-1797.
- [8] Susanto, S., & Sensuse, D. I. (2008, Vol.1 No.2). Pengklasifikasian Artikel Berita Berbahasa Indonesia secara Otomatis Menggunakan Naive Bayes Classifier. *Jurnal Ilmu Komputer dan Informasi*
- [9] O'Connor, B. T. (2012, Maret 13). *Cosine similarity, Pearson correlation, and OLS coefficients*. Retrieved Januari 7, 2015, from AI and Social Science - Brendan O'Connor: <http://brenocon.com/blog/2012/03/cosine-similarity-pearson-correlation-and-ols-coefficients/>
- [10] Marmanis H, Babenko D. 2009. *Algorithms of the Intelligent Web*. Greenwich(UK): Manning Publ
- [11] Nugroho, Bhuono Agung. 2005. *Strategi Jitu Memilih Metode Statistik Penelitian dengan SPSS*. Yogyakarta : Andi
- [12] Witten, I. H., Frank, E., & Mark, A. H. (2011). *Data Mining : Practical Machine Learning Tools and Techniques 3rd Edition*. Burlington: Elsevier.
- [13] Kementerian Kesehatan RI. (2011). *Standar Antropometri Penilaian Status Gizi Anak*. Jakarta: Menteri Kesehatan.

SISTEM KLASIFIKASI *FEEDBACK* PELANGGAN DAN REKOMENDASI SOLUSI ATAS KELUHAN DI UPT PUSKOM UNS DENGAN ALGORITMA NAÏVE BAYES *CLASSIFIER* DAN *COSINE SIMILARITY*

Aisha Alfiani Mahardhika
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A
Surakarta
aish.alfiani@gmail.com

Ristu Saptono
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A
Surakarta
r_saptono@uns.ac.id

Rini Angrainingsih
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
rinianggra@gmail.com

Abstrak—Saat ini, konsumen dapat menyampaikan keluhan terhadap UPT Puskom UNS melalui *mentions* terhadap akun Twitter. *Mentions* yang diberikan oleh konsumen kemudian diklasifikasikan apakah *mentions* tersebut termasuk keluhan, berita atau *spam*. Klasifikasi *mentions* dilakukan menggunakan algoritma Naïve Bayes *Classifier* berdasarkan *supervised learning*. Peningkatan akurasi untuk algoritma Naïve Bayes *Classifier* dilakukan dengan menggunakan teknik *Laplacian Smoothing*. Algoritma *Cosine Similarity* digunakan untuk mengelompokkan *mentions* keluhan yang memiliki *term* yang sama. Dari kelompok *mentions* tersebut, administrator akan memberikan solusi yang relevan terhadap keluhan. Hasil penelitian menunjukkan bahwa proses klasifikasi dengan algoritma Naïve Bayes *Classifier* untuk proses pelatihan memiliki tingkat akurasi terendah 86.67% dengan data pelatihan sebanyak 30 *mentions* dan tingkat akurasi tertinggi 100% dengan data pelatihan sebanyak 20 *mentions*. Proses pengujian dilakukan secara bertahap dengan tingkat akurasi terendah adalah 60% yang dicapai pada pengujian pertama dan kedua, sedangkan tingkat akurasi tertinggi dicapai pada pengujian kelima dan keenam yakni 90%. *Mentions* keluhan tidak dapat dikelompokkan dengan algoritma *Cosine Similarity* karena jumlah data yang sangat terbatas yakni 29 data dan tidak ada *mentions* yang memiliki *term* sama. Namun setelah dilakukan *self-test*, *mentions* keluhan yang memiliki *term* sama dapat dikelompokkan dengan baik.

Kata kunci—Klasifikasi, Naïve Bayes *Classifier*, *Cosine Similarity*, rekomendasi solusi.

1. PENDAHULUAN

Keluhan merupakan salah satu sinyal ketidakpuasan yang diberikan konsumen terhadap sebuah perusahaan. Perbedaan antara harapan dan kemampuan sesungguhnya dari produk atau jasa yang diterima oleh konsumen akan menyebabkan ketidakpuasan, dimana hal ini dapat menimbulkan negative effect yang diyakini akan berpengaruh terhadap loyalitas konsumen [1]. Oleh sebab itu, penanganan terhadap keluhan konsumen pun menjadi hal yang mutlak dilaksanakan oleh perusahaan.

Beragam cara digunakan perusahaan agar konsumen dapat menyampaikan keluhannya dengan baik. Saat ini,

cara lain yang mulai dilirik oleh perusahaan agar konsumen dapat menyampaikan keluhannya adalah melalui *social media*. Data menunjukkan bahwa dari 63 juta masyarakat pengguna internet di Indonesia, sebanyak 95% diantaranya menggunakan internet untuk mengakses *social media* yakni Facebook dan Twitter [2].

Salah satu *social media* yang dapat digunakan untuk menyampaikan keluhan konsumen adalah Twitter, yang beralamat di <http://www.twitter.com>. Twitter dapat dilihat sebagai alat komunikasi interaktif dari mulut ke mulut melalui media elektronik [3]. Oleh sebab itu, Twitter merupakan *social media* yang tepat untuk menjalin hubungan dengan pelanggan [3].

Sebagai sebuah institusi pelayanan, UPT. Puskom UNS memiliki cara untuk menghimpun keluhan dari konsumen, salah satunya adalah melalui Twitter yang dapat diakses di <http://www.twitter.com/UPTPuskomUNS>. Konsumen dapat menyampaikan keluhan terhadap UPT. Puskom UNS dengan melakukan *mentions* terhadap akun Twitter tersebut.

Salah satu kelemahan penyampaian keluhan melalui Twitter adalah *mentions* berbentuk teks digital tidak terstruktur. Selain itu, tidak semua *mentions* yang diberikan kepada UPT. Puskom UNS adalah berupa keluhan. Hal tersebut menyulitkan administrator Twitter, karena administrator harus memilih terlebih dahulu *mentions* mana yang dianggap sebagai keluhan, dan selanjutnya menjawab keluhan dari konsumen secara manual. Selain itu, *mentions* yang tidak terbaca ataupun adanya *mentions* berisi keluhan yang sama dari dua atau lebih *user* berbeda menyebabkan administrator tidak dapat menjawab keluhan secara maksimal.

Oleh sebab itu dibutuhkan sebuah sistem yang dapat memudahkan administrator untuk mengelola keluhan pelanggan dengan memilah *tweet* berupa keluhan dan bukan keluhan, dan selanjutnya mengelompokkan *tweet* keluhan yang dianggap memiliki makna sama. Salah satu cara yang dapat digunakan adalah dengan metode analisis *text mining*. Analisis *text mining* diperlukan dalam menangani masalah teks digital tidak terstruktur. Salah satu kegiatan penting dalam *text mining* adalah klasifikasi atau kategorisasi teks. Kategorisasi teks sendiri saat ini memiliki berbagai cara pendekatan antara lain pendekatan *probabilistic*, *support vector machine* (SVM), dan *artificial neural network*, atau *decision tree classification*.

Penelitian untuk melihat performa Naïve Bayes dalam pengklasifikasian dokumen telah dilakukan oleh Ting [4]. Dalam penelitian tersebut, didapatkan hasil bahwa metode Naïve Bayes merupakan metode klasifikasi paling baik jika dibandingkan dengan metode lain seperti *decision tree*, *neural network* dan *support vector machines* dalam hal akurasi dan efisiensi komputasi. Penelitian yang dilakukan oleh Hamzah [5] yang mengkaji kinerja metode Naïve Bayes *Classifier* untuk kategorisasi teks berita dan teks akademik memberikan hasil bahwa algoritma Naïve Bayes *Classifier* memiliki kinerja yang baik untuk klasifikasi dokumen teks, baik dokumen berita maupun dokumen akademik.

Penelitian lain dilakukan oleh Isa dan Abidin [6] untuk mengukur tingkat kesamaan antar dokumen menggunakan algoritma *Vector Space Model* untuk mendeteksi plagiarisme. Hasilnya, algoritma *Vector Space Model* dapat mendeteksi dengan baik kesamaan dokumen melalui kesamaan paragraf dalam dokumen.

Penelitian ini akan mengklasifikasikan *feedback* pelanggan UPT Puskom UNS yang disampaikan melalui *mentions* Twitter dengan pendekatan *probabilistic* menggunakan algoritma Naïve Bayes *Classifier*. *Tweet* akan diklasifikasikan menjadi keluhan, berita dan spam. Algoritma Naïve Bayes *Classifier* telah digunakan secara luas karena kemudahannya, baik dalam proses pelatihan maupun klasifikasi [7]. Algoritma Naïve Bayes *Classifier* menjadi sangat efisien jika dilakukan berdasarkan *supervised learning* [8].

Dalam perhitungan menggunakan algoritma Naïve Bayes *Classifier*, sering ditemukan adanya perhitungan yang mengandung nilai peluang sama dengan 0, menyebabkan hasil perhitungan menjadi kurang akurat. Untuk menghindari munculnya peluang bernilai 0 pada algoritma ini, digunakan teknik Laplacian Smoothing.

Algoritma Cosine Similarity digunakan untuk mengelompokkan *mentions* keluhan yang memiliki term sama. Dari kelompok *mentions* tersebut, administrator akan memberikan solusi yang relevan terhadap keluhan. Selanjutnya ketika ada *mentions* baru berupa keluhan, sistem akan menampilkan rekomendasi solusi untuk kemudian dimoderasi oleh administrator. Dengan cara ini administrator tidak perlu memberikan jawaban yang sama secara berulang-ulang kepada akun yang berbeda. Hal ini akan mempercepat kinerja administrator dalam memberikan tanggapan terhadap keluhan konsumen.

2. LANDASAN TEORI

2.1. TWITTER API

Twitter API terdiri dari dua komponen yang berbeda, REST dan SEARCH API. REST API memungkinkan *developer* Twitter untuk mengakses data *core* Twitter (*tweet*, *timeline*, *user data*), sedangkan SEARCH API digunakan untuk membuat *query tweet* [9].

2.2. TEXT MINING

Text mining merupakan variasi dari *data mining* yang digunakan untuk menemukan pola tertentu dari sekumpulan besar data tekstual [10].

Langkah yang dilakukan dalam *text mining* adalah sebagai proses *text preprocessing*. Tindakan yang dilakukan pada tahap *text preprocessing* adalah *toLowerCase*, yaitu mengubah semua karakter huruf menjadi huruf kecil serta *tokenizing*, yaitu proses pemecahan kalimat menjadi token berupa kata atau *term*, dimana setiap *term* dipisahkan oleh delimiter. Tanda titik (.), koma (,), spasi () dan karakter angka yang ada pada kalimat dapat dianggap sebagai delimiter [11].

2.3. JARO-WINKLER DISTANCE

Salah satu metode similaritas yang digunakan untuk mendeteksi kesamaan dua dokumen adalah Jaro *metric*. Dalam penelitian persamaan dokumen, didapatkan hasil yang baik dengan menggunakan metode Jaro, yang didasarkan pada jumlah dan urutan karakter yang sama antara dua dokumen [12].

Algoritma Jaro mendefinisikan ‘karakter yang sama’ sebagai karakter pada kedua *string* yang sama dan memenuhi ketentuan jarak teoritis [12]. Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi nilai persamaan di bawah ini.

$$\left[\frac{\max(|s_1|, |s_2|)}{2} \right] - 1$$

Persamaan di bawah ini menunjukkan rumus untuk menghitung jarak (d_j) antara dua *string* yaitu s_1 dan s_2 pada algoritma Jaro.

$$d_j = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

dimana:

m = jumlah karakter yang sama dan memenuhi kriteria

$|s_1|$ = panjang *string* 1

$|s_2|$ = panjang *string* 2

t = jumlah transposisi

Pengembangan dari algoritma Jaro berdasarkan Winkler menggunakan nilai panjang *prefix* yang sama di awal *string* dengan nilai maksimal adalah 4 (l) [13].

Persamaan di bawah ini menunjukkan nilai Jaro-Winkler *distance* (d_w) bila *string* s_1 dan s_2 yang diperbandingkan.

$$d_w = d_j + (lp(1-d_j))$$

dimana:

d_j = Jaro *distance* untuk *string* s_1 dan s_2

l = panjang *prefix* umum di awal *string* (panjang karakter yang sama sebelum ditemukan ketidaksamaan, maksimal 4)

p = konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah $p = 0.1$.

Semakin tinggi Jaro-Winkler *distance* untuk dua *string* berarti semakin mirip kedua *string* tersebut. Nilai terendah Jaro-Winkler *distance* adalah 0 yang menandakan tidak ada kesamaan antara kedua *string*. Nilai tertingginya adalah 1 yang menunjukkan kedua *string* sama persis [14].

2.4. NAÏVE BAYES CLASSIFIER

Naive Bayes Classifier merupakan klasifikasi yang berdasarkan pada teorema Bayes. *Naive Bayesian Classifier* mengasumsikan bahwa setiap atribut dalam

sebuah kelas merupakan atribut independen yang tidak terkait dengan atribut lain. Asumsi ini disebut *class conditional independence*. Naive Bayesian classifier memiliki tingkat akurasi dan kecepatan yang tinggi ketika diaplikasikan kepada data berjumlah besar [15].

Persamaan di bawah ini merupakan persamaan Naive Bayes untuk klasifikasi dokumen.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

dimana:

$P(c|d)$ = *posterior*, yakni probabilitas dokumen d berada di kelas c ,

$P(c)$ = *prior*, yaitu probabilitas kelas c sebelum masuknya dokumen d ,

$P(t_k|c)$ = *likelihood*, yaitu probabilitas kemunculan token t_k dalam kelas c ,

n_d = jumlah token dalam dokumen d [15].

Dalam Naive Bayes Classifier, dokumen d akan masuk ke dalam kelas c yang memiliki *maximum a posteriori* (MAP) atau kelas c_{map} , dihitung dengan persamaan sebagai berikut:

$$c_{map} = \arg \max_{c \in C} P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

dimana c adalah variabel kelas yang tergabung dalam himpunan kelas C [15].

2.5. LAPLACIAN SMOOTHING

Teknik *Laplacian Smoothing* digunakan untuk mengatasi nilai probabilitas kondisional pada Naive Bayes Classifier yang dapat bernilai 0. Cara yang digunakan pada teknik ini adalah dengan menambahkan angka 1 pada perhitungan *Likelihood* [16].

Persamaan di bawah ini menunjukkan perhitungan nilai *Likelihood* untuk algoritma Naive Bayes Classifier.

$$P(F_i|C) = \frac{1+n(F_i,C)}{|W|+n(C)}$$

dimana:

$n(F_i,C)$ = jumlah term F_i yang ditemukan di seluruh data pelatihan dengan kategori C

$n(C)$ = jumlah term di seluruh data pelatihan dengan kategori C

$|W|$ = jumlah seluruh term dari seluruh data pelatihan [16].

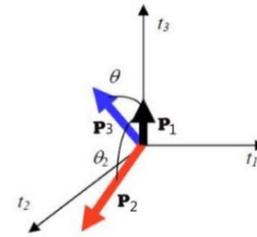
2.6. VECTOR SPACE MODEL

Representasi satu set dokumen sebagai vector dalam ruang vektor dikenal sebagai *Vector Space Model* (VSM) dan merupakan dasar untuk sejumlah operasi pengambilan informasi seperti penilaian dokumen dalam *query*, klasifikasi dan *clustering* dokumen [17].

Vector Space Model digunakan untuk mengukur kemiripan antara dua buah dokumen. Dokumen merupakan *vector* berdimensi n , sedangkan t adalah seluruh *term* yang ditemukan dalam *library* tanpa duplikasi [6].

Gambar 1 memperlihatkan tiga buah vector pada ruang dimensi 3. Nilai *cosinus* digunakan untuk mengukur kesamaan antara dua *vector*. Pada Gambar 1,

P_1 adalah vector dari dokumen pembanding, sementara P_2 dan P_3 adalah vector dari dokumen yang dibandingkan.



Gambar 1. Vector Space Model [6]

2.7. PEMBOBOTAN TF x IDF

Term Frequency (TF) adalah jumlah kemunculan *term t* pada dokumen d , yang dirumuskan sebagai $freq(d, t)$. Matriks bobot *term frequency* atau $TF(d, t)$ menunjukkan hubungan antara *term t* dengan dokumen d , dimana jika dokumen d tidak mengandung *term t* maka bobotnya bernilai 0, dan sebaliknya. Fungsi di bawah ini menunjukkan perhitungan nilai TF [15].

$$TF(d, t) = freq(d, t)$$

Document Frequency (DF) merupakan jumlah dokumen yang mengandung *term t*. *Inverse Document Frequency* (IDF) menunjukkan pembobotan dari *term t*. *Term* yang jarang muncul dalam dokumen memiliki nilai IDF yang tinggi, sementara *term* yang sering muncul dalam dokumen memiliki nilai IDF yang lebih rendah. Fungsi di bawah ini menunjukkan perhitungan nilai IDF [17].

$$IDF(t) = \log \frac{N}{df(t)}$$

Nilai TF-IDF dalam *Vector Space Model* dihitung dengan fungsi sebagai berikut [15]:

$$TF-IDF(d, t) = TF(d, t) \times IDF(t)$$

2.8. COSINE SIMILARITY

Untuk menghitung kesamaan antara kedua dokumen dalam *vector space*, maka akan dihitung nilai *cosine similarity* dari representasi vector kedua dokumen [17].

$$Sim(P_1, P_2) = \cos \theta = \frac{P_1 \cdot P_2}{|P_1| |P_2|}$$

Pada fungsi diatas, pembilang merepresentasikan nilai *dot product* dari kedua vector, sedangkan penyebut merepresentasikan nilai perkalian dari Euclidean *length* kedua vector. Nilai *dot product* dari kedua vector dapat dicari dengan fungsi sebagai berikut [17]:

$$P_1 \cdot P_2 = \sum_{i=1}^M P_{1i} P_{2i}$$

Sedangkan nilai Euclidean *length* dari vector P dapat dicari dengan fungsi di bawah ini [17]:

$$|P| = \sqrt{\sum_{i=1}^M P_i^2}$$

Jika nilai *cosine similarity* dari kedua vector adalah 1 maka kedua dokumen adalah sama persis. Jika nilai *cosine similarity* adalah 0 maka dapat dikatakan bahwa kedua dokumen tidak sama.

2.9. CONFUSION MATRIX

Confusion matrix merupakan matriks yang menampilkan prediksi klasifikasi dan klasifikasi yang aktual. *Confusion matrix* berukuran LxL, dimana L adalah jumlah label klasifikasi yang berbeda. Tabel 1 menunjukkan *confusion matrix* untuk L=2 [18].

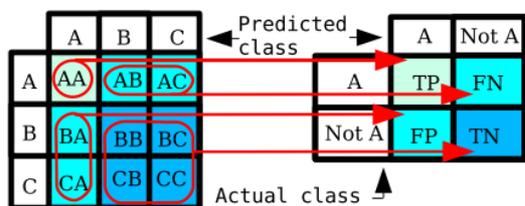
Tabel 1. Confusion Matrix untuk L = 2 [18]

| | | | |
|---------|----------|---------|---------|
| | Prediksi | Negatif | Positif |
| Aktual | | | |
| Negatif | | a | b |
| Positif | | c | d |

Nilai akurasi didapatkan dari rumus di bawah ini:

$$\text{Akurasi} = \frac{a+d}{a+b+c+d} [18]$$

Gambar 2 menunjukkan perubahan dari *extended confusion matrix* berukuran 3x3 menjadi berukuran 2x2, dengan kelas ‘A’ sebagai kelas positif dan kelas ‘Not A’ sebagai kelas negatif.



Gambar 2. Extended confusion matrix 3x3 [19]

3. METODOLOGI

Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.1. STUDI LITERATUR

Pada tahap ini, dilakukan studi literatur untuk mempelajari *text mining*, metode *Jaro-Winkler distance*, algoritma *Naïve Bayes Classifier* dan *Cosine Similarity*.

3.2. PENGUMPULAN DATA

Data diambil dari *mentions* yang dilakukan pihak luar terhadap Twitter milik UPT Puskom UNS sejak tanggal 17 Februari 2014 hingga 4 September 2014. Jumlah data yang diambil sebanyak 90 data.

3.3. TEXT MINING

Tahap *text mining* yang dilakukan pada penelitian ini adalah *text preprocessing*. Pada tahap ini, *mentions* diolah dengan cara mengubah seluruh abjad dalam *mentions* menjadi huruf kecil, menghapus kalimat *retweet* (RT), menghapus *username* dan *hashtag*, serta menghapus seluruh karakter selain abjad pada *mentions*. Selanjutnya *mentions* yang berupa *string* diubah menjadi bentuk token/*term*, yang dipisahkan oleh delimiter berupa spasi ().

3.4. KLASIFIKASI DENGAN NAÏVE BAYES CLASSIFIER

Data yang ada diklasifikasikan sebagai keluhan, berita dan spam menggunakan algoritma *Naïve Bayes*

Classifier. Pertanyaan, keluhan dan sarkasme akan dimasukkan dalam kategori keluhan. Sementara itu, *mentions* kosong atau yang memiliki konten hanya berupa *username* akan diklasifikasikan sebagai spam.

Algoritma *Naïve Bayes Classifier* dilakukan berdasarkan proses *supervised learning*. Proses *supervised learning* pertama dilakukan terhadap 20 *mentions* awal. Selanjutnya ditambahkan 10 *mentions* untuk digunakan dalam proses *supervised learning* berikutnya hingga 80 *mentions* digunakan sebagai dokumen pelatihan.

Langkah-langkah yang dilakukan dalam *supervised learning* menggunakan algoritma *Naïve Bayes Classifier* adalah sebagai berikut:

1. Menghitung nilai *prior* setiap kategori
2. Menghitung frekuensi setiap *term* pada *mentions* untuk setiap kategori
3. Menghitung nilai *likelihood* setiap *term* pada *mentions* untuk setiap kategori
4. Menghitung nilai *posterior* setiap *mentions* untuk setiap kategori
5. Menentukan klasifikasi *mentions* berdasarkan nilai *posterior* tertinggi

3.5. PEMBAHARUAN LIBRARY

Setelah proses klasifikasi selesai dilakukan, *administrator* akan memberikan tinjauan terhadap hasil klasifikasi, dan mengubah hasil klasifikasi jika dianggap tidak sesuai.

Term dari seluruh *mentions* akan dibandingkan dengan seluruh *stop word* dengan menggunakan algoritma *Jaro-Winkler*. *Stop word* yang digunakan berasal dari penelitian yang telah dilakukan oleh Fadilla Z. Tala [20]. Jika *term* dari *mentions* memiliki kesamaan minimal 95% dengan *term* dari *stop word*, maka *term* tersebut dianggap sebagai *stop word*. Selanjutnya, sistem secara otomatis akan memasukkan seluruh *term* dari *mentions* selain *stop word* ke dalam *library* sesuai dengan kategori *mentions*.

3.6. PENGELOMPOKAN MENTIONS KELUHAN DENGAN COSINE SIMILARITY

Berikut ini merupakan langkah-langkah pengelompokan *mentions* keluhan yang bermakna sama dengan algoritma *Cosine Similarity*.

1. Menghitung nilai *term frequency* (TF) untuk setiap *term*
2. Menghitung nilai *index document frequency* (IDF) untuk setiap *term*
3. Menghitung bobot setiap *term*
4. Melakukan normalisasi
5. Menghitung panjang vektor
6. Menghitung nilai similaritas setiap *mentions*

3.7. PENENTUAN SOLUSI TERHADAP MENTIONS KELUHAN

Pada tahap ini, *administrator* akan memberikan solusi terhadap setiap kelompok *mentions* keluhan.

3.8. ANALISIS HASIL

Pada tahap ini dilakukan pengujian terhadap proses klasifikasi oleh algoritma Naïve Bayes Classifier. Setelah dilakukan supervised learning terhadap 20 dokumen, 10 dokumen selanjutnya digunakan sebagai proses pengujian. Selanjutnya setelah dilakukan supervised learning terhadap 30 dokumen, 10 dokumen selanjutnya digunakan sebagai proses pengujian. Proses pengujian dilakukan hingga 10 dokumen terakhir. Pengujian dilakukan dengan cara membandingkan hasil klasifikasi dokumen menggunakan algoritma Naïve Bayes Classifier dengan klasifikasi dokumen secara manual. Pengujian dilakukan untuk mengetahui tingkat akurasi hasil klasifikasi.

Pengujian hasil klasifikasi dalam penelitian ini menggunakan extended confusion matrix. Dokumen yang termasuk False Positive atau error tipe I adalah dokumen berita dan spam yang diklasifikasikan oleh sistem sebagai dokumen keluhan serta dokumen keluhan dan spam yang diklasifikasikan oleh sistem sebagai berita. Klasifikasi ini tidak menyebabkan kesalahan yang fatal karena dokumen keluhan dan berita akan cenderung dibaca oleh administrator.

Sedangkan dokumen keluhan dan berita yang diklasifikasikan oleh sistem sebagai spam dikategorikan sebagai False Negative atau error tipe II. Error tipe II menyebabkan kesalahan yang fatal karena dokumen keluhan dan berita tidak terbaca oleh administrator dikarenakan dokumen dengan klasifikasi spam cenderung diabaikan.

Gambar 3 menunjukkan pengaplikasian extended confusion matrix dalam penelitian ini.

Gambar 3. Aplikasi extended confusion matrix dalam penelitian

| | | | | |
|---------|---------|-------------------------------|----------------|--------------------------------|
| | | Sistem | | |
| | | Keluhan | Berita | Spam |
| Realita | Keluhan | True Positive | False Positive | False Negative (Type II Error) |
| | Berita | | True Positive | |
| | Spam | False Positive (Type I Error) | | True Negative |

4. HASIL DAN PEMBAHASAN

4.1. IMPLEMENTASI

Pada penelitian ini, berhasil dikumpulkan sebanyak 90 data yang berasal dari mentions terhadap Twitter milik UPT Puskom UNS. Selanjutnya, dilakukan proses text preprocessing terhadap data tersebut. Setelah proses text preprocessing selesai, dilakukan proses klasifikasi terhadap dokumen dengan algoritma Naïve Bayes Classifier.

Selanjutnya, administrator melakukan peninjauan terhadap hasil klasifikasi. Jika ada kategori yang tidak sesuai, administrator berhak mengubah kategori mentions.

Pada saat proses pengubahan kategori dilakukan, sistem akan mencocokkan term dari seluruh mentions dengan term yang termasuk stop word menggunakan algoritma Jaro-Winkler distance. Jika term dari mentions mirip dengan term dari stop word, dengan batas nilai minimal adalah 0.95, maka term tersebut akan dianggap sebagai stop word. Selanjutnya, sistem akan memasukkan term yang bukan merupakan stop word ke dalam library sesuai dengan kategori yang telah ditinjau oleh administrator.

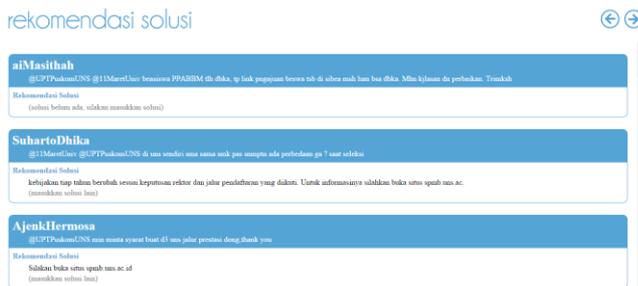
Setelah proses klasifikasi dengan algoritma Naïve Bayes Classifier selesai, maka dilakukan pengelompokan dokumen keluhan yang dianggap memiliki makna sama dengan algoritma Cosine Similarity. Hasil pengelompokan dokumen dengan algoritma Cosine Similarity menunjukkan bahwa dokumen keluhan tidak dapat terkelompokkan karena diantara data yang sangat terbatas yakni 29 data, tidak ada mentions yang memiliki term sama sehingga tidak ada pasangan mentions yang memiliki batas minimal nilai cosinus adalah 0.8. Karena alasan tersebut, maka dilakukan proses self-test untuk menguji algoritma Cosine Similarity, yaitu menguji dokumen dengan dokumen itu sendiri. Dokumen yang digunakan berjumlah 10 dokumen yang sama persis (dipilih secara acak) dan 5 dokumen yang mirip (dipilih secara acak). Dokumen yang mirip memiliki makna yang serupa namun sebagian kata telah diubah.

Tabel 3 menunjukkan hasil proses Cosine Similarity untuk self-test. ID_Dokumen dengan nilai awalan '0_' menunjukkan 10 dokumen yang sama persis, sedangkan ID_Dokumen dengan nilai awalan '1_' menunjukkan 5 dokumen yang mirip.

Tabel 3. Hasil proses Cosine Similarity untuk self-test

| No | ID_ Dokumen 1 | ID_ Dokumen 2 | Nilai Cosinus | Ket. |
|----|--------------------------|------------------------|---------------|-------|
| 1 | 0010 | 0_0010 | 1 | Mirip |
| 2 | 0017 | 0_0017 | 1 | Mirip |
| 3 | 0019 | 0_0019 | 1 | Mirip |
| 4 | 0025 | 0_0025 | 1 | Mirip |
| 5 | 0028 | 0_0028 | 1 | Mirip |
| 6 | 0030 | 0_0030 | 1 | Mirip |
| 7 | 0033 | 0_0033 | 1 | Mirip |
| 8 | 0034 | 0_0034 | 1 | Mirip |
| 9 | 0035 | 0_0034 | 1 | Mirip |
| 10 | 0036 | 0_0036 | 1 | Mirip |
| 11 | 1_45199758 9743734784 | 451997589 743734784 | 0.84 | Mirip |
| 12 | 1_49753811 4123206658 | 497538114 123206658 | 0.80 | Mirip |

Dari Tabel 2 dapat dilihat bahwa untuk dokumen yang sama persis, seluruh dokumen dapat dideteksi dengan nilai cosinus adalah 1. Sedangkan untuk dokumen yang mirip, terdapat 2 pasang dokumen yang terdeteksi karena batas minimal nilai cosinus dua dokumen yang dianggap mirip adalah 0.8. Sehingga untuk selanjutnya, rekomendasi solusi dapat diberikan pada dokumen yang mirip. Gambar 3 menunjukkan halaman pemberian rekomendasi solusi.



Gambar 3. Halaman pemberian rekomendasi solusi

Karena sifatnya yang berupa rekomendasi, maka administrator dapat melakukan perubahan atau update terhadap solusi yang diberikan untuk selanjutnya dikirimkan kepada user. Rekomendasi solusi yang diberikan dapat berjumlah lebih dari satu.

4.2. ANALISIS HASIL

Pada data klasifikasi dokumen secara manual, terdapat 29 dokumen dengan kategori keluhan, 57 dokumen dengan kategori berita dan 4 dokumen dengan kategori spam. Sedangkan setelah dilakukan proses

klasifikasi dengan algoritma Naïve Bayes Classifier, didapatkan data klasifikasi dari setiap proses supervised learning. Dalam penelitian ini diasumsikan bahwa administrator memusatkan perhatian pada dokumen yang diklasifikasikan sebagai keluhan serta membaca dokumen yang diklasifikasikan sebagai berita. Sedangkan dokumen dengan klasifikasi spam cenderung untuk diabaikan, dimana dokumen ini termasuk dalam False Negative. False Negative Rate menunjukkan tingkat kesalahan fatal oleh sistem, dihitung dari jumlah dokumen yang termasuk False Negative dibagi dengan jumlah total seluruh dokumen.

Selanjutnya dilakukan perhitungan jumlah dokumen yang hasil klasifikasinya akurat, yakni dokumen yang termasuk True Positive dan True Negative. Nilai persentase akurasi dari setiap proses didapatkan dari persamaan di bawah ini.

$$Akurasi = \frac{TP + TN}{N} \times 100\% [18]$$

Tingkat akurasi kategori dari setiap proses training dan testing ditunjukkan oleh Tabel 4.

Tabel 4. Tingkat akurasi dokumen pelatihan

| Jumlah Data Training | Fase Training | | Jumlah Data Testing | Fase Testing | | Akurasi (%) | |
|----------------------|---------------|---------|---------------------|--------------|---------|-------------|---------|
| | TP + TN | FP + FN | | TP + TN | FP + FN | Training | Testing |
| 20 data | 20 | 0 | 10 data | 6 | 4 | 100% | 60% |
| 30 data | 26 | 4 | 10 data | 6 | 4 | 86.67% | 60% |
| 40 data | 36 | 4 | 10 data | 7 | 3 | 90% | 70% |
| 50 data | 46 | 4 | 10 data | 8 | 2 | 92% | 80% |
| 60 data | 58 | 2 | 10 data | 9 | 1 | 96.67% | 90% |
| 70 data | 69 | 1 | 10 data | 9 | 1 | 98.57% | 90% |
| 80 data | 79 | 1 | 10 data | 8 | 2 | 98.75% | 80% |

10 data pertama yang digunakan untuk proses testing merupakan data yang muncul setelah dilakukan supervised learning terhadap 20 dokumen. 10 data kedua yang digunakan untuk proses pengujian merupakan data yang muncul setelah dilakukan supervised learning terhadap 30 dokumen. Sementara 10 data terakhir pada proses pengujian merupakan data yang muncul setelah dilakukan supervised learning terhadap 80 dokumen.

Pada proses awal pelatihan, tingkat akurasi cenderung tinggi, kemudian menurun. Selanjutnya, tingkat akurasi naik secara bertahap dan cenderung stabil. Hal ini mengindikasikan bahwa sistem dapat menerima proses pembelajaran yang diberikan.

Sedangkan pada proses pengujian, tingkat akurasi naik secara bertahap dan cenderung stabil. Namun demikian, pada proses pengujian setelah pelatihan sebanyak 80 data, tingkat akurasi mengalami penurunan sebesar 10%. Hal ini disebabkan term pada mentions belum muncul pada mentions sebelumnya yang digunakan untuk proses pembelajaran, sehingga sistem tidak dapat mengkategorikan mentions dengan baik. Sementara untuk pengujian dengan tingkat akurasi yang tinggi menunjukkan bahwa term pada mentions sudah terdapat pada mentions sebelumnya yang digunakan untuk proses pembelajaran, sehingga sistem dapat mengkategorikan mentions tersebut dengan baik.

5. KESIMPULAN DAN SARAN

Hasil penelitian menunjukkan bahwa dapat dilakukan proses klasifikasi pada mentions Twitter dengan algoritma Naïve Bayes Classifier. Pada proses pengujian, tingkat akurasi tertinggi yang berhasil dicapai adalah 90%, sedangkan tingkat akurasi terendah adalah 60%.

Mentions keluhan tidak dapat dikelompokkan dengan algoritma Cosine Similarity dikarenakan jumlah data yang sangat terbatas yakni 29 data dan tidak ada mentions yang memiliki term yang sama. Namun setelah dilakukan proses self-test dengan jumlah total 15 mentions, didapatkan hasil bahwa mentions yang mirip dapat dikelompokkan dengan algoritma Cosine Similarity.

Saran yang dapat dipertimbangkan untuk penelitian lebih lanjut antara lain:

1. Perlu dilakukan pengelompokan mentions dengan Cosine Similarity jika jumlah mentions cukup banyak dan term dalam mentions terbatas pada bidang tertentu saja.
2. Dapat mempertimbangkan semantik atau makna kata agar hasil klasifikasi yang didapatkan lebih akurat.

6. DAFTAR PUSTAKA

- [1] Wijaya, T. 2008. "Pengaruh Kepuasan Pada Penanganan Keluhan dan Citra Perusahaan Terhadap Loyalitas Konsumen Natasha Skin Care". , vol. XIV, no. 1.
- [2] Pitakasari, A. R. (2013, Oktober) Republika Online. [Online].
<http://www.republika.co.id/berita/trendtek/internet/13/10/30/mvh7rm-penggunaan-internet-di-indonesia-95-persen-untuk-sosmed>
- [3] Devi, M. F.S. and Sanaji, 2013. "Penerapan Bauran Pemasaran Dalam Jaringan Melalui Media Sosial Untuk Membangun Hubungan Pelanggan". *Jurnal Ilmu Manajemen Volume 1 Nomor 5*, pp. 1314-1326.
- [4] Ting, S. L. , Ip, W. H. , and Tsang, H. C. July 2011. "Is Naïve Bayes Classifier a Good Classifier for Document Classification?" *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37-46.
- [5] Hamzah, A. 2012. *Klasifikasi Teks Dengan Naive Bayes Classifier (NBC) Untuk Pengelompokan Teks Berita dan Abstract Akademis in Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III*, Yogyakarta, pp. 269-277.
- [6] Isa, T. M. and Abidin, T. F. 2013. "Mengukur Tingkat Kesamaan Paragraf Menggunakan Vector Space Model untuk Mendeteksi Plagiarisme". *Seminar Nasional dan Expo Teknik Elektro*, pp. 229-234.
- [7] Chakrabarti, S. , Roy, S. , and Soundalgekar, M.V. 2003. "Fast and Accurate Text Classification Via Multiple Linear Discriminant Projection". *The International Journal on Very Large Data Bases*, pp. 170-185.
- [8] Aribowo, T. 2010. "Aplikasi Inferensi Bayes pada Data Mining terutama Pattern Recognition". Bandung,.
- [9] Wardhani, Y. 2012. "Deteksi Spammer di Twitter dengan Mempelajari Tweet-Based Features". Surakarta,.
- [10] Feldman, R. and Sanger, J. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data..* New York: Cambridge University Press.
- [11] Weiss, S. M. et al. 2005. *Text Mining: Predictive Methods for Analyzing Unstructured Information..* New York: Springer.
- [12] Jaro, M. A. 1989. "Advances In Record-Linkage Methodology As Applied To Matching The 1985 Censuf of Tampa, Florida". *Journal of The American Statistical Association*, pp. 414-420.
- [13] Winkler, W. E. 1999. "The State of Record Linkage and Current Research Problems"..
- [14] Kurniawati, A. , Puspitodjati, S. , and Rahman, S. 2010. "Implementasi Algoritma Jaro-Winkler Distance Untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia"..
- [15] Han, J. and Kamber, M. 2006. *Data Mining: Concepts and Techniques, Second Edition..* San Fransisco, United States of America: Morgan Kaufmann Publishers.
- [16] Dai, W. et al. 2007. "Transferring Naive Bayes Classifiers for Text Classification"..
- [17] Manning, C. D. , Raghavan, P. , and Schutze, H. 2009. *An Introduction to Information Retrieval..* Cambridge, England: Cambridge University Press.
- [18] Kohavi, R. and Provost, F. 1998. "Glossary of Terms". *Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, vol. 30, no. 2/3, pp. 271-274.
- [19] Felkin, M. 2007. "Comparing Classification Results between N-ary and Binary Problems" in *Quality Measures in Data Mining*, GuilletFabrice J. and HamiltonHoward J. , Eds.: Springer Berlin Heidelberg, pp. 277-301.
- [20] Tala, F. Z. 2003. "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia". Netherlands,.

SISTEM PAKAR UNTUK MENDIAGNOSIS PENYAKIT INFEKSI MENGGUNAKAN FORWARD CHAINING

Ahmad Aniq Noor Mutsaqof

Jurusan Informatika, Fakultas MIPA,
Universitas Sebelas Maret
Jl. Ir. Sutami 36 A Surakarta 57126
kupersemar@gmail.com

Wiharto S.T M.Kom

Jurusan Informatika, Fakultas MIPA,
Universitas Sebelas Maret
Jl. Ir. Sutami 36 A Surakarta 57126
wi_harto@yahoo.com

Esti Suryani S.Si M.Kom

Jurusan Informatika, Fakultas MIPA,
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Surakarta 57126
suryapalapa@yahoo.com

ABSTRAK

Penyakit infeksi merupakan salah satu masalah kesehatan yang paling utama di negara-negara berkembang termasuk Indonesia. Penyakit infeksi yang sering di derita adalah penyakit diare, demam *tifoid*, demam berdarah, radang paru-paru. Hal ini menunjukkan bahwa penyakit infeksi harus cepat didiagnosis agar tidak semakin parah. Penyakit infeksi merupakan penyakit menular yang mudah menyerang anak, karena anak belum mempunyai sistem imun yang baik. Namun, jumlah dokter umum di Indonesia belum mencukupi dan pendistribusian dokter tidak merata. Oleh karena itu, diperlukan adanya sistem yang dapat membantu tenaga medis non dokter untuk mendiagnosis penyakit infeksi agar penanganan penyakit infeksi lebih cepat teratasi.

Metode yang digunakan sebagai mesin inferensi adalah *Forward chaining*. Metode ini mempunyai konsep logika yang sama dengan dokter pada saat mendiagnosis pasien karena berbasis *if-then*. *Forward chaining* merupakan metode yang mempunyai strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data tersebut menghasilkan suatu kesimpulan.

Pengujian dilakukan oleh dokter muda (Co-Ass) di Rumah Sakit Umum Daerah Sunan Kalijaga Demak. Pengujian dilakukan terhadap 50 pasien yang menghasilkan 6 kegagalan dalam mendeteksi penyakit infeksi. Nilai akurasi yang diperoleh adalah 88%. Hasil dari pengujian dapat disimpulkan bahwa metode *forward chaining* dapat diimplementasikan untuk mendiagnosis penyakit infeksi dengan melakukan tahapan akuisisi pengetahuan dan representasi pengetahuan.

Kata Kunci :

Forward Chaining, Penyakit Infeksi, Sistem Pakar.

1. PENDAHULUAN

Penyakit infeksi merupakan satu kumpulan jenis-jenis penyakit yang mudah menyerang anak-anak yang disebabkan oleh infeksi virus, infeksi bakteri, dan infeksi parasit [1]. Penyakit infeksi merupakan salah satu masalah kesehatan yang paling utama di negara-negara berkembang termasuk Indonesia. Berdasarkan Survei Kesehatan Rumah Tangga Tahun 2007, penyebab utama kematian antara lain 28,1 % disebabkan oleh penyakit infeksi dan parasit, 18,9 % disebabkan oleh penyakit vaskuler, dan 15,7 % disebabkan oleh penyakit pernapasan [2]. Hal lain yang juga menjadi pertimbangan adalah banyaknya angka kematian pada anak Indonesia yaitu lebih dari 200 anak meninggal per 100.000 angka kelahiran. Angka ini masih jauh dari MDGs (*Millennium Development Goals*) Negara Indonesia di tahun 2015 yaitu 102 per 100.000 kelahiran [3].

Penyakit diare, demam *tifoid*, demam berdarah, infeksi saluran pernapasan atas (influenza, radang amandel, radang tenggorokan), radang paru-paru, dan demam yang belum diketahui penyebabnya (observasi *febris*) merupakan penyakit infeksi yang termasuk ke dalam 10 penyakit terbanyak rumah sakit di Indonesia. Sedangkan penyakit lainnya antara lain : *dispepsi* (gangguan tidak nyaman

pada perut), hipertensi (darah tinggi), penyakit apendiks (usus buntu) dan gastritis (nyeri lambung) [4]. Hal ini menunjukkan bahwa penyakit infeksi harus cepat didiagnosis dan ditangani agar tidak semakin parah. Namun demikian jumlah dokter spesialis di Indonesia belum mencukupi. Terutama spesialis anak, kandungan dan kebidanan, bedah, penyakit dalam, anestesi dan rehabilitasi medik [5]. Kebutuhan dokter spesialis di tahun 2014 berjumlah 29.452, padahal ketersediaan pada tahun 2013 hanya 20.602 dokter spesialis [6].

Permasalahan lain terjadi pada pendistribusian dokter yang tidak merata. Pusat-pusat distribusi dokter adalah di Pulau Jawa dan Bali serta di Provinsi Sumatera Utara dan Sulawesi Selatan. Menurut kepala Badan Pengembangan dan Pemberdayaan SDM Kesehatan Kementerian Kesehatan tahun 2010, Bambang Giatno, hingga tahun 2010 diperkirakan ada sekitar 1600 puskesmas yang terutama berada di daerah terpencil tidak memiliki dokter jaga akibat dari kurang pemerataan distribusi [7]. Melihat fakta tersebut diperlukan adanya sistem yang dapat membantu tenaga medis non dokter untuk mendiagnosis penyakit infeksi.

Sistem pakar akan membantu tenaga medis non dokter untuk mendeteksi penyakit serta mendapat informasi solusi pengobatannya sejak dini sehingga dapat dilakukan penanganan segera [8]. Sistem pakar juga diharapkan dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli [9].

Komponen-komponen yang terdapat dalam sistem pakar antara lain adalah antarmuka pengguna, basis pengetahuan, akuisisi pengetahuan, mesin inferensi, fasilitas penjelasan, *workplace*, dan perbaikan pengetahuan [10]. Mesin inferensi dalam sistem pakar merupakan hal yang pokok karena merupakan otak dari sistem pakar. Salah satu metode yang digunakan dalam mesin inferensi dari sistem pakar adalah metode *forward chaining*. Metode ini mempunyai konsep logika yang mudah dimengerti oleh dokter karena berbasis *if-then*. *Forward chaining* merupakan metode yang mempunyai strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dan dari data-data tersebut dicari suatu kesimpulan [11]. *Forward Chaining* sudah banyak di implementasikan di bidang kedokteran, misalnya pada penyakit THT (telinga, hidung dan tenggorokan), penyakit asma dan gangguan pernapasan, dan penyakit mata [12] [13] [14].

Metode *forward chaining* cocok untuk diagnosa penyakit, karena mampu mengenali jenis penyakit berdasarkan gejala yang dipenuhi [15]. *Forward chaining* merupakan mesin inferensi yang melakukan pencarian dari suatu masalah kepada solusinya, di mana *forward chaining* merupakan inferensi *data-drive*, inferensi dimulai dari informasi yang tersedia dan baru konklusi [16].

Berdasarkan latar belakang di atas penulis ingin melakukan penelitian dengan judul "Sistem Pakar untuk Mendiagnosis Penyakit Infeksi Menggunakan Metode *Forward Chaining*".

Tujuan yang ingin dicapai dari penelitian ini adalah dapat menghasilkan sistem pakar menggunakan metode *forward chaining* dalam mendiagnosis penyakit infeksi.

Manfaat dari penelitian ini adalah ini diharapkan dapat membantu tenaga medis non dokter untuk memudahkan mendiagnosis penyakit infeksi sedini mungkin.

2. DASAR TEORI

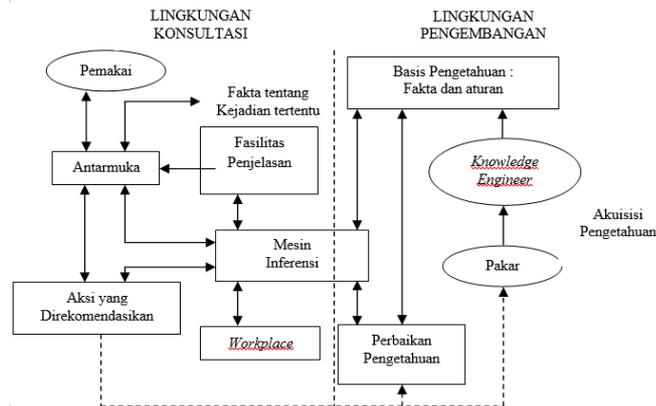
Terdapat beberapa teori yang mendukung dalam penelitian ini yaitu sebagai berikut.

2.1 Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah, orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam [17].

2.2 Arsitektur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan Pengembangan digunakan untuk memasukkan pengembangan pakar ke dalam lingkungan sistem pakar, lingkungan konsultasi digunakan oleh non pakar untuk memperoleh pengetahuan dan nasihat pakar [10]. Arsitektur sistem pakar dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Sistem Pakar [10]

2.3 Akuisisi Pengetahuan

Akuisisi pengetahuan dilakukan untuk memperoleh pengetahuan dari pakar atau sumber lain (sumber terdokumentasi, buku, sensor, file komputer, dan lain-lain) [10]. Dalam proses akuisisi pengetahuan, seseorang perakayasa pengetahuan menjembatani antara pakar dengan basis pengetahuan (*knowledge base*). Perakayasa pengetahuan mendapatkan pengetahuan dari pakar, mengolahnya bersama pakar tersebut dan menaruhnya dalam basis pengetahuan dengan format tertentu.

2.4 Tabel Keputusan

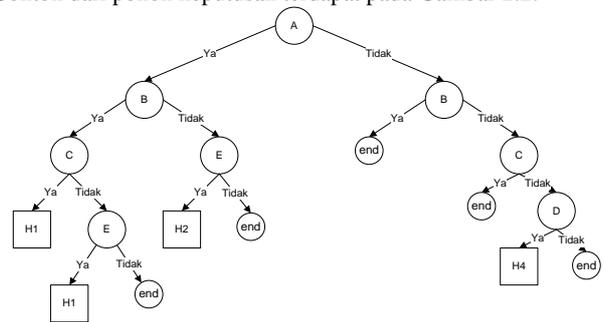
Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matriks kondisi yang dipertimbangkan dalam pendeskripsian kaidah [18]. Contoh tabel keputusan ada pada Tabel 2.1.

Tabel 2.1 Tabel Keputusan [18]

| | Hipotesa 1 | Hipotesa 2 | Hipotesa 3 | Hipotesa 4 |
|---------|------------|------------|------------|------------|
| Fakta A | v | v | v | |
| Fakta B | v | | v | v |
| Fakta C | v | | | v |
| Fakta D | | | | v |
| Fakta E | | v | v | |

2.5 Pohon Keputusan (Decision Tree)

Pohon keputusan adalah salah satu bentuk representasi pengetahuan yang digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi [19]. Pohon keputusan dibuat dengan mengacu pada tabel keputusan. Contoh dari pohon keputusan terdapat pada Gambar 2.2.



Gambar 2.2 Pohon Keputusan [18]

2.6 Runut Maju (Forward Chaining)

Forward chaining merupakan strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data tersebut dicari suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang premisnya sesuai dengan data-data tersebut, kemudian dari kaidah-kaidah tersebut diperoleh suatu kesimpulan. *Forward chaining* memulai proses pencarian dengan data sehingga strategi ini disebut juga *data-driven* [10].

Contoh proses *forward chaining* berbentuk kaidah produksi [18]:

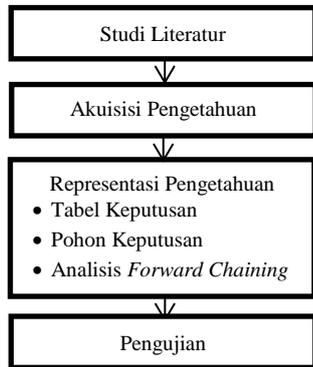
IF kondisi 1
 AND kondisi 2
 AND kondisi 3
 THEN kesimpulan.

2.7 Penyakit Infeksi

Penyakit infeksi merupakan satu kumpulan jenis-jenis penyakit yang mudah menyerang anak-anak yang disebabkan oleh infeksi virus, infeksi bakteri, dan infeksi parasit. [1]. Ada bermacam – macam penyakit infeksi di antaranya diare, demam *tifoid*, demam berdarah, infeksi saluran pernapasan atas (influenza, amandel, radang tenggorokan akut), radang paru-paru, dan *observasi febris* [6].

3. METODE PENELITIAN

Metode penelitian yang dilakukan penulis untuk menyelesaikan penelitian ini digambarkan dengan langkah-langkah sebagai berikut:



Gambar 3.1 Alur Penelitian

3.1 Studi Literatur

Studi literatur dilaksanakan dengan melakukan studi kepustakaan untuk memahami teori yang berasal dari buku maupun artikel dan jurnal yang bersumber dari internet. Studi kepustakaan dilakukan untuk mempelajari hal-hal yang berhubungan dengan penelitian secara umum, sistem pakar, *forward chaining*, macam-macam penyakit infeksi dan juga gejala-gejala apa saja yang dirasakan akibat penyakit infeksi.

3.2 Akuisisi Pengetahuan

Tahap akuisisi pengetahuan disebut juga tahap pengumpulan *knowledge*. Tahap ini dilakukan *input* data gejala-gejala, penyakit dan saran pengobatan yang diperoleh dari Dokter Spesialis Anak RSUD Sunan Kalijaga Demak yaitu dr. Budi Nur Cahyani, Sp.A.

3.3 Representasi Pengetahuan

Tahap representasi pengetahuan merupakan tahap yang digunakan untuk pengodean pengetahuan sistem pakar. Representasi pengetahuan terdiri atas pembuatan tabel keputusan, perancangan pohon keputusan dan analisis inferensi *forward chaining*

3.4 Pengujian

Tahap ini dilakukan pengujian terhadap aplikasi dengan tujuan memastikan fungsionalitas aplikasi dapat berjalan sama seperti apa yang diharapkan. Dan juga nantinya akan dilakukan pengujian dengan cara membandingkan hasil penyakit pasien yang sudah didiagnosis oleh dokter dengan hasil diagnosis sistem. Selanjutnya dilakukan perhitungan akurasi sebagai berikut

$$Akurasi = \frac{Jumlah\ data\ terdeteksi\ positif}{Jumlah\ keseluruhan\ data} \times 100\%$$

4. HASIL DAN PEMBAHASAN

Pembahasan sistem pakar untuk mendeteksi penyakit infeksi ini melalui beberapa tahapan. Dimulai dari akuisisi pengetahuan, representasi pengetahuan, implementasi dan pengujian sistem. Pembahasan sistem pakar untuk mendeteksi penyakit infeksi ini melalui beberapa tahapan. Dimulai dari akuisisi pengetahuan, representasi pengetahuan, implementasi dan pengujian sistem.

4.1 Akuisisi Pengetahuan

Berdasarkan hasil wawancara yang penulis lakukan di RSUD Sunan Kalijaga Demak dengan dr. Budi Nur Cahyani, Sp.A, beliau menyarankan supaya studi kasus yang digunakan untuk penelitian ini adalah 6 penyakit infeksi yang termasuk dalam 10 penyakit terbesar yang ada di rumah sakit Indonesia. Penyakit infeksi tersebut antara lain penyakit diare, demam *tifoid*, demam berdarah, pneumonia, observasi *febris*, dan ISPA (infeksi saluran pernapasan atas). ISPA sendiri dapat dibagi menjadi 3 penyakit yaitu influenza, amandel, dan radang tenggorokan akut. Studi kasus yang digunakan dalam penelitian terdapat 8 jenis penyakit dengan 10 gejala, 1 di antaranya adalah gejala umum. Demam merupakan gejala umum, sedangkan gejala lainnya antara lain batuk, pilek, nyeri perut, sesak napas, BAB (buang air besar) cair lebih dari 4x sehari, pendarahan (bintik merah di kulit, mimisan, BAB berdarah), lidah kotor, sakit saat menelan, dan mendengkur saat tidur.

4.2 Perancangan Tabel Keputusan

Perancangan tabel keputusan dilakukan dengan cara mengidentifikasi *rule* dari *knowledge* pakar kemudian memasukkan *rule* ke dalam alur penelusuran untuk ditelusuri dan untuk dijabarkan ke dalam Tabel 4.1.

Tabel 4.1 Tabel Keputusan Penyakit Infeksi

| | P00 1 | P00 2 | P00 3 | P00 4 | P00 5 | P00 6 | P00 7 | P00 8 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| G00 1 | V | V | V | V | V | V | V | V |
| G00 2 | | | | V | V | | V | V |
| G00 3 | | | | V | V | | V | V |
| G00 4 | V | V | V | | | | | |
| G00 5 | | | | | V | | | |
| G00 6 | V | | | | | | | |
| G00 7 | | | V | | | | | |
| G00 8 | | V | | | | | | |
| G00 9 | | | | | | | V | V |
| G01 0 | | | | | | | | V |

Tabel 4.1 terdiri dari kolom gejala (G) dan kolom penyakit (P). Terdapat 10 gejala-gejala penyakit infeksi, dengan kode G001 sampai G010. Detail gejala – gejala penyakit infeksi terdapat pada Tabel 4.2. Terdapat 8 jenis penyakit pada kolom penyakit yang dikodekan dengan huruf P001 sampai P008. Detail penyakit infeksi terdapat pada Tabel 4.3.

Tabel 4.2 Gejala Penyakit Infeksi

| Kode | Nama Gejala |
|------|---|
| G001 | Demam |
| G002 | Batuk |
| G003 | Pilek |
| G004 | Nyeri perut |
| G005 | Sesak napas |
| G006 | BAB cair lebih dari 4 kali sehari |
| G007 | Pendarahan (mimisan, bintik merah di kulit atau BAB berdarah) |

| | |
|------|--|
| G008 | lidah kotor (tertutup selaput putih atau kuning yang menempel di permukaan), tepi <i>hipertermi</i> (tepi lidah kemerahan), dan tremor |
| G009 | Sakit saat menelan |
| G010 | Tidur mendengkur |

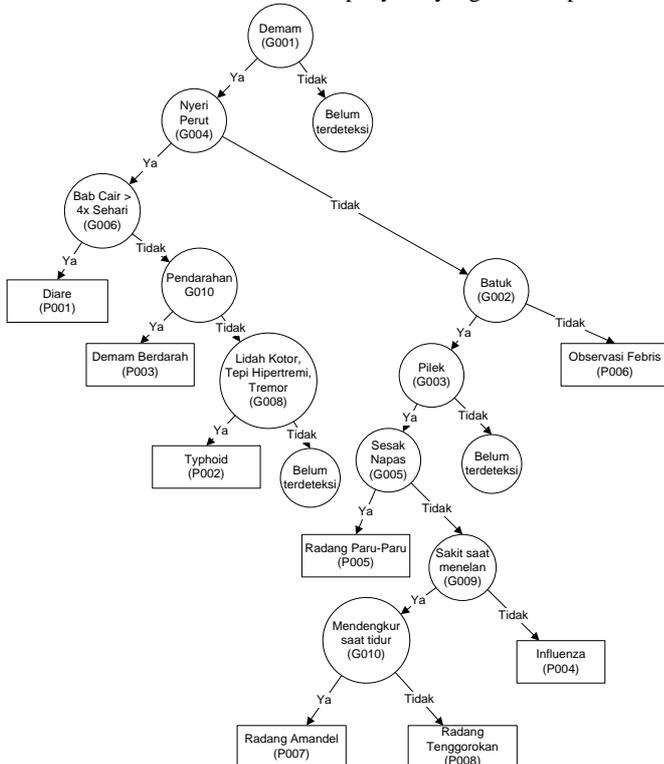
Tabel 4.3 Penyakit Infeksi

| Kode | Nama Penyakit |
|------|---|
| P001 | Diare |
| P002 | Demam <i>Tifoid</i> |
| P003 | Demam Berdarah |
| P004 | Influenza |
| P005 | <i>Pneumonia</i> (radang paru-paru) |
| P006 | <i>Observasi Febris</i> |
| P007 | <i>Tonsillitis</i> (Radang Amandel) |
| P008 | <i>Tonsillopharyngitis</i> (Radang tenggorokan) |

4.3 Perancangan Pohon Keputusan

Gambar 4.1 berdasarkan Tabel 4.1. G001 merupakan gejala umum bagi penderita penyakit infeksi yaitu demam. Gejala selanjutnya adalah nyeri perut. Berdasarkan gejala nyeri perut dapat dikelompokkan penyakit yang diderita oleh pasien apakah memiliki gejala nyeri perut ataukah tidak. Apabila pertanyaan gejala mempunyai jawaban “Ya” maka pada pohon keputusan akan dilanjutkan pertanyaan dengan jawaban “Ya” dan penelusuran akan dilanjutkan sehingga akan terjadi keputusan gejala penyakit infeksi yang diderita pasien.

Contoh untuk gejala pertama pasien mengalami gejala demam maka selanjutnya sistem akan menampilkan pertanyaan “Apakah pasien mengalami nyeri perut?” apabila pasien tidak mengalaminya maka dilanjutkan dengan pertanyaan “Apakah pasien sedang batuk?”. Apabila pasien juga tidak mengalaminya sistem akan mendiagnosis pasien tersebut menderita observasi febris. Sistem akan berhenti menampilkan pertanyaan apabila sudah mencapai hasil, yaitu pasien didiagnosis penyakit infeksi atau sistem belum bisa mendeteksi penyakit yang diderita pasien.



Gambar 4.1 Pohon Keputusan Penyakit Infeksi

4.4 Analisis Inferensi Forward Chaining

Analisis inferensi *forward chaining* dapat berbentuk sebuah *rule*. *Rule* tersebut menjelaskan penyakit berdasarkan gejalanya. *Rule* terdiri atas informasi masukan yang berupa gejala (*if*), lanjutan gejala-gejala (*and*) dan konklusi yang berupa diagnosis penyakit infeksi (*then*). *Rule* 1 berisi *IF* G001 *AND* G004 *AND* G006 *THEN* P001. Hal ini berarti jika gejala 1 (demam), gejala 4 (nyeri perut), dan gejala 6 (bab cair > 4x sehari) diderita oleh pasien maka sistem mendiagnosis pasien tersebut menderita penyakit P001 (diare).

- | | |
|--|---|
| <p>Rule 1 <i>IF</i> G001 <i>AND</i> G004 <i>AND</i> G006 <i>THEN</i> P001</p> | <p>Rule 5 <i>IF</i> G001 <i>AND</i> G002 <i>AND</i> G003 <i>AND</i> G005 <i>THEN</i> P005</p> |
|--|---|
- | | |
|--|--|
| <p>Rule 2 <i>IF</i> G001 <i>AND</i> G004 <i>AND</i> G008 <i>THEN</i> P002</p> | <p>Rule 6 <i>IF</i> G001 <i>THEN</i> P006</p> |
|--|--|
- | | |
|--|---|
| <p>Rule 3 <i>IF</i> G001 <i>AND</i> G004 <i>AND</i> G010 <i>THEN</i> P003</p> | <p>Rule 7 <i>IF</i> G001 <i>AND</i> G002 <i>AND</i> G003 <i>AND</i> G009 <i>THEN</i> P008</p> |
|--|---|
- | | |
|--|--|
| <p>Rule 4 <i>IF</i> G001 <i>AND</i> G002 <i>AND</i> G003 <i>THEN</i> P004</p> | <p>Rule 8 <i>IF</i> G001 <i>AND</i> G002 <i>AND</i> G003 <i>AND</i> G009 <i>AND</i> G010 <i>THEN</i> P007</p> |
|--|--|

4.5 Pengujian

Pengujian yang dilakukan terhadap pasien diare yang berjumlah 12 orang sistem hanya mampu mendeteksi sejumlah 11 orang. Dikarenakan sistem belum bisa mendeteksi penyakit yang diderita pasien. Misalnya dalam melakukan diagnosis penyakit diare, sistem dapat menyimpulkan bahwa pasien sedang menderita penyakit diare bilamana memenuhi *rule* sebagai berikut:

IF Demam
AND Nyeri perut
AND BAB cair > 4x sehari
THEN Diare

akan tetapi terkadang pasien tidak mengalami atau merasakan salah satu gejala yang terdapat dalam *rule* penyakit infeksi. Misalnya pasien hanya mengalami gejala demam dan BAB cair > 4x sehari saja, sehingga sistem tidak dapat mendeteksi bahwa pasien menderita penyakit diare. Sama halnya pada penyakit demam *tifoid*, influenza, radang amandel dan radang tenggorokan.

Tabel 4.4 Tabel Pengujian

| No. | Nama Penyakit | Jumlah Diagnosis Dokter | Jumlah Diagnosis Sistem |
|-----|---------------------|-------------------------|-------------------------|
| 1 | Diare | 12 | 11 |
| 2 | Demam <i>Tifoid</i> | 6 | 5 |
| 3 | Demam Berdarah | 3 | 3 |
| 4 | Influenza | 10 | 9 |

| | | | |
|--------|-------------------------|----|----|
| 5 | Radang Paru-Paru | 5 | 5 |
| 6 | <i>Observasi Febris</i> | 3 | 3 |
| 7 | Amandel | 5 | 4 |
| 8 | Radang Tenggorokan Akut | 6 | 4 |
| Jumlah | | 50 | 44 |

Tabel 4.4 menunjukkan bahwa pengujian terhadap 50 pasien menghasilkan 6 kegagalan dalam mendeteksi penyakit infeksi. Nilai akurasi pengujian tersebut adalah $(\frac{44}{50}) \times 100 \% = 88\%$. Pengujian dilakukan oleh dokter muda (Co-Ass) yang sedang melakukan praktek di Poliklinik Anak RSUD Sunan Kalijaga Demak selama 6 hari dari tanggal 3-8 November 2014.

Jumlah diagnosis antara sistem dan penyakit berbeda dikarenakan dalam pengujianya sistem tidak bisa mendeteksi penyakit yang diderita pasien. Hal ini disebabkan karena pasien tidak merasakan atau mengalami gejala yang ditanyakan oleh sistem.

4.6 Kesimpulan dan Saran

4.6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan bahwa dengan melakukan tahap akuisisi pengetahuan dan representasi pengetahuan sistem pakar ini berhasil diimplementasikan. Metode *forward chaining* digunakan sebagai mesin inferensi di mana mesin inferensi berfungsi sebagai otak dari sistem pakar.

Hasil yang didapat dari akuisisi pengetahuan adalah dengan menggunakan penyakit infeksi yang termasuk dalam 10 penyakit terbesar yang ada di rumah sakit di Indonesia. Setelah akuisisi pengetahuan dilakukan, lalu direpresentasikan kemudian diimplementasikan ke sebuah sistem pakar pengujian dilakukan terhadap 50 pasien. Terdapat 6 kegagalan pendeteksian dari 50 pengujian yang dilakukan. Nilai akurasi dihitung dengan membandingkan nilai keberhasilan dalam pendeteksian sistem dengan jumlah pasien pengujian. Nilai akurasi yang didapat sebesar 88%.

4.6.2 Saran

Saran yang dapat digunakan untuk penelitian selanjutnya yaitu :

1. Perlu adanya *update knowledge base* sehingga lebih banyak penyakit infeksi yang dapat didiagnosis
2. Menambahkan metode lain untuk menghindari penyakit yang belum bisa didiagnosis dikarenakan pasien tidak merasakan atau mengalami gejala. Misal dengan menggunakan *certainty factor*.

5. DAFTAR PUSTAKA

- [1] T. Rampengan dan I. Laurentz, Penyakit Infeksi Tropik Pada Anak, Jakarta: Buku Kedokteran EGC, 1997.
- [2] S. K. Nasution, "Isolasi Bakteri Penghasil Antimikroba Dari Sampel Tanah Taman Wisata Alam Sibolangit Dan Kemampuannya Menghambat Pertumbuhan Mikroba Patogen," Universitas Sumatra Utara, Medan, 2012.
- [3] UNICEF, Ringkasan Kajian Kesehatan Ibu dan Anak, Jakarta: UNICEF Indonesia, 2012.
- [4] Kemenkes, "Profil Kesehatan Indonesia," Ditjen Bina Upaya Kesehatan, Jakarta, 2009.
- [5] Republika, "IDI Sebut Jumlah Dokter Spesialis Masih Minim," 1 10 2014. [Online]. Available: <http://www.republika.co.id/berita/nasional/jabodetabek-nasional/14/10/01/ncr7ix-idi-sebut-jumlah-dokter-spesialis-masih-minim>.
- [6] Depkes, "Ketersediaan dan Kesiapan SDM Kesehatan dalam Pencapaian Target RPJMN," Pusat Perencanaan dan Pendayagunaan SDM Kesehatan, Surabaya, 2012.
- [7] Dikti, Potret Ketersediaan dan Kebutuhan Tenaga Dokter, Jakarta: Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan, 2010.
- [8] A. S. Broto, "Perancangan dan Implementasi Sistem Pakar untuk Analisa Penyakit Dalam," Universitas Diponegoro, Semarang, 2010.
- [9] L. Handayani dan T. Sutikno, "Sistem Pakar untuk Diagnosis Penyakit THT Berbasis Web dengan "e2gLite Expert System Shell"," *Jurnal Teknologi Industri*, vol. Vol. XII, no. No.1, pp. 19 -26, 2008.
- [10] E. Turban, R. Sharda dan D. Delen, Decision Support System and Business Intelligent Systems Ninth Edition, New Jersey: Pearson Education, 2011.
- [11] F. Sari, "Expert System untuk Mengetahui Aspek Kepribadian Seseorang Berdasarkan Tes EQ (Emotional Quotien) dengan Metode Forward Chaining," Universitas Putra Indonesia YPTK, Padang, 2011.
- [12] Suraya, "Sistem Pakar untuk Mendiagnosa Penyakit THT Berdasarkan Gejalanya untuk Menentukan Alternatif Pengobatan menggunakan Tanaman Obat," *Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III*, pp. 337-346, 2012.
- [13] A. R. Tohir, "Rancang Bangun Aplikasi Sistem Pakar untuk Diagnosis Penyakit Asma dan Gangguan Pernafasan (Studi Kasus Rumah Skait Dokter Soetomo)," Sekolah Tinggi Ilmu Komputer, Surabaya, 2012.
- [14] R. Reisa, Jusak dan P. Sudarmaningtyas, "Sistem Pakar untuk Diagnosis Penyakit Mata," *Jurnal Sistem Informasi dan Komputerisasi Akuntansi (JSIKA) Vol.2 No.2*, pp. 30-40, 2013.
- [15] D. R. Fitriana, "Merancang Sistem Pakar Diagnosa Penyakit Jantung dengan Metode Forward Chaining Berbasis Web," Universitas Muhammadiyah Ponorogo, Ponorogo, 2003.
- [16] R. Sofa, D. Destiani dan A. Susanto, "Pembangunan Aplikasi Sistem Pakar untuk Diagnosis Penyakit Tanaman Padi," *Algoritma Sekolah Tinggi Teknologi Garut Vol. 09 No. 03*, pp. 1-8, 2012.
- [17] Kusriani, Aplikasi Sistem Pakar, Yogyakarta: Andi, 2008.
- [18] S. Hartati dan S. Iswanti, Sistem Pakar & Pengembangannya, Edisi Pertama, Yogyakarta: Graha Ilmu, 2008.
- [19] J. Giarrattano dan G. Riley, 1994. Expert System Principles and Programming, Boston: Publishing Company, 1994.

Volume 4. Nomor 1. Juni 2015

- Pengembangan Sistem Analisis Akademis Menggunakan OLAP dan Data Clustering Studi Kasus : Akademik Universitas Sebelas Maret Surakarta
Bakharudin Yusuf B, Antonius Bima M, Hasan Dwi Cahyono 1 – 11
- Digital Watermarking Image dengan Menggunakan Discrete Wavelet Transform dan Singular Value Decomposition (DWT-SVD) untuk Copyright Labeling
Nur Anisah, Bambang Harjito, Esti Suryani 12 – 19
- Pembuatan Peta Data Prediksi Penderita Penyakit DBD Di Kota Surakarta Dengan Metode Susceptible-Infected-Removed(SIR)
Diannita Kartikasari, Sarngadi Palgunadi, Bambang Harjito 20 – 27
- Penentuan Model Terbaik pada Metode Naive Bayes Classifier dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter
Apriliya Fitri C, Ristu Saptono, Sari Widya Sihwi 28 – 35
- Sistem Klasifikasi Feedback Pelanggan Dan Rekomendasi Solusi Atas Keluhan Di UPT Puskom UNS Dengan Algoritma Naive Bayes Classifier Dan Cosine Similarity
Aisha Alfiani Mahardhika, Ristu Saptono, Rini Anggrainingsih 36 – 42
- Sistem Pakar Untuk Mendiagnosis Penyakit Infeksi Menggunakan Chaining
Ahmad Aniq Noor Mutsaqof, Wiharto, Esti Suryani 43 – 47

Diterbitkan oleh :

Program Studi Informatika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Ketingan Surakarta 57126 Indonesia
Telp. 62-271-663451, Fax. 62-271-663451
email : itsmart_uns@yahoo.co.id



ISSN : 2301-7201