TESIS – KI 142502

PENCARIAN INFORMASI MENGGUNAKAN MODEL TERDEKOMPOSISI: APLIKASI PADA RISET PENEMUAN ANTIBIOTIK

NYOMAN JUNIARTA
5114201042

DOSEN PEMBIMBING
Amedeo Napoli
Chedy Raïssi

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

THESIS – KI 142502

KNOWLEDGE DISCOVERY USING DECOMPOSABLE MODELS: AN APPLICATION TO THE RESEARCH OF NEW ANTIBACTERIAL DRUGS

NYOMAN JUNIARTA
5114201042

SUPERVISORS
Amedeo Napoli
Chedy Raïssi

MASTER PROGRAMME
DEPARTMENT OF INFORMATICS ENGINEERING
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
NYOMAN JUNIARTA
Nrp. 5114201042

dengan judul:
PENCARIAN INFORMASI MENGGUNAKAN MODEL TERDEKOMPOSISI: APLIKASI PADA RISET PENEMUAN ANTIBIOTIK

Tanggal Ujian : 16-9-2016
Periode Wisuda : 2016 Gasal

Disetujui oleh:

Direktur Program Pascasarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

(halaman ini sengaja dikosongkan)

PENCARIAN INFORMASI MENGGUNAKAN MODEL TERDEKOMPOSISI: APLIKASI PADA RISET PENEMUAN ANTIBIOTIK

Nama mahasiswa	: Nyoman Juniarta
NRP	: 5114201042
Pembimbing I	: Amedeo Napoli
Pembimbing II	: Chedy Raïssi

ABSTRAK

Penemuan obat-obatan antibiotik adalah salah satu tantangan pada bidang kemoinformatika. Dibutuhkan antibiotik baru secara cepat dan efektif karena banyak bakteri menjadi kebal terhadap antibiotik lama. Molekul-molekul kimia yang tersimpan di beberapa perusahaan dan laboratorium menyediakan kandidat yang berpotensi sebagai antibiotik baru. Tetapi, terlalu banyak kandidat yang harus diteliti. Untuk mengatasinya, dibutuhkan pencarian informasi yang dapat mendeteksi kandidat-kandidat penting melalui atribut mereka. Jumlah atribut tersebut sangatlah besar.

Tujuan penelitian ini adalah mempelajari atribut-atribut tersebut dan menentukan atribut yang penting, dengan kata lain, untuk mereduksi dimensi data molekul. Fokus penelitian ini ditujukan pada molekul-molekul antibiotik yang sudah ada di pasaran, dengan sekitar 500 atribut yang diperoleh dari penelitian sebelumnya. Sebagai prosedur seleksi fitur, penelitian ini menggunakan analisis log-linear untuk menemukan asosiasi di antara atribut. Karena jumlah atribut mencapai ratusan, maka digunakan Chordalysis yang bekerja pada model log-linear yang bisa didekomposisi.

Penelitian ini menemukan bahwa atribut-atribut dari penelitian sebelumnya memiliki beberapa asosiasi. Dengan demikian, beberapa atribut yang redundan dapat dieliminasi.

Kata kunci: antibiotik, model grafis probabilistik, pencarian informasi, seleksi fitur.

(halaman ini sengaja dikosongkan)

KNOWLEDGE DISCOVERY USING DECOMPOSABLE MODELS: AN APPLICATION TO THE RESEARCH OF NEW ANTIBACTERIAL DRUGS

Student's name	: Nyoman Juniarta
Student's ID	: 5114201042
First supervisor	: Amedeo Napoli
Second supervisor	: Chedy Raïssi

ABSTRACT

Antibacterial drug discovery is one of the emerging challenges in chemoinformatics. There is an urgent need for finding new effective drugs faster because many bacteria become resistant to the old drugs. The chemical molecules stored in companies and laboratories' databases provide potential candidates for developing new drugs. However, there are far too many candidates to investigate. This is where knowledge discovery could be of help, by sifting through the known properties of the molecule to select the most promising candidates for further experiments. The number of properties, or descriptors, characterizing the molecules is rather large.

The aim of the present work is to study these descriptors and find out which ones really matter, that is, to reduce the dimension of the description space. We focus on a subset of antibacterial molecules already on the market, with around 500 descriptors obtained from the selection process in the previous work. As our feature selection procedure, we use log-linear analysis (LLA) to discover associations among descriptors. Given that the number of descriptors is high, we study Chordalysis that focuses on a specific subset of log-linear models: decomposable models.

We find that the selected descriptors from the previous work still have many associations among them. Therefore, a number of redundant descriptors can still be left out.

Keywords: antibacterial drug, feature selection, knowledge discovery, probabilistic graphical model.

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah SWT yang telah memberi kemampuan pada penulis untuk dapat menyelesaikan Tesis yang berjudul "Pencarian Informasi Menggunakan Model Terdekomposisi: Aplikasi pada Riset Penemuan Antibiotik". Tesis ini dilaksanakan dalam program *joint degree* antara Institut Teknologi Sepuluh Nopember di Indonesia dengan Universitas Lorraine di Prancis.

Pada kesempatan ini, penulis ingin menyampaikan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Ayah dan ibu tercinta serta keluarga besar penulis yang selalu memberi doa dan dukungan selama proses perkuliahan di Teknik Informatika ITS.
2. Mr. Amedeo Napoli dan Mr. Chedy Raïssi sebagai pembimbing penulis selama mengerjakan tesis di Prancis.
3. Kementerian Pendidikan dan Kebudayaan Indonesia yang telah memberikan bantuan beasiswa sehingga penulis dapat mengikuti program *joint degree* di Prancis.
4. Ibu Dr. Chastine Fatichah, S.Kom., M.Kom. sebagai dosen wali penulis yang telah membantu melancarkan kegiatan akademik penulis selama perkuliahan di ITS.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku ketua program studi magister Teknik Informatika ITS yang telah membantu penulis dalam menyelesaikan kegiatan akademik selama program *joint degree*.
6. Pegawai administrasi Teknik Informatika dan Pascasarjana ITS yang telah memberikan kemudahan bagi penulis dalam pengurusan administrasi program *joint degree*.
7. Teman-teman mahasiswa Magister Teknik Informatika ITS yang telah memberi semangat dan bantuan sewaktu kuliah.
8. Teman-teman mahasiswa *joint degree* dari ITS dan PPNS yang telah menemani penulis kuliah di Prancis.
9. Pihak lain yang tidak dapat disebutkan satu persatu yang telah membantu

terselesaikannya tesis ini.

Akhir kata, penulis berharap bahwa Tesis ini dapat bermanfaat bagi perkembangan ilmu pengetahuan di Indonesia.

Surabaya, November 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

(halaman ini sengaja dikosongkan)

DAFTAR TABEL

(halaman ini sengaja dikosongkan)

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Penyakit-penyakit yang berasal dari bakteri dan parasit merupakan salah satu penyebab utama kematian di seluruh dunia. Karena aman dan efektif dalam menangani penyakit-penyakit menular tersebut, antibiotik banyak digunakan tanpa proses diagnosis dan uji kerentanan. Dengan semakin banyaknya penggunaan antibiotik, beberapa bakteria mengalami evolusi, mutasi, dan seleksi, sehingga mereka menjadi kebal terhadap antibiotik tersebut [5]. Kemudian, seiring dengan munculnya "*superbug*" yang kebal antibiotik, seperti *Staphylococcus aureus* (MRSA), antibiotik-antibiotik tradisional menjadi tidak efektif.

Munculnya bakteri-bakteri yang kebal antibiotik menyebabkan banyak perusahaan obat-obatan mengurangi penelitian di bidang tersebut. Riset untuk menemukan antibiotik baru kini dijalankan oleh komunitas akademik, termasuk para peneliti di ilmu komputer. Kerja sama antara ilmu komputer dan kimia komputasional melahirkan bidang ilmu baru, kemoinformatika. Kemoinformatika menggabungkan bermacam-macam area penelitian dan teknologi, termasuk pembelajaran mesin, pengenalan pola, dinamika molekul, mekanika kuantum, dan statistika. Bidang ilmu ini banyak dieksplorasi seiring dengan bertumbuhnya kemampuan kalkulasi komputer dan volume data. Salah satu tujuannya adalah menghasilkan metode yang lebih efisien sebagai metode penemuan obat antibiotik baru.

Dengan berlimpahnya data tentang molekul yang tersimpan di perusahaan-perusahaan dan di laboratorium, dimensi yang sangat besar dari tiap molekul merupakan suatu keuntungan sekaligus kerugian. Volume yang besar dari basis data tersebut menjamin adanya molekul yang bisa menjadi antibiotik, tetapi juga memunculkan kesulitan dalam menemukan satu dari ribuan molekul. Untuk mengatasinya, peneliti dapat menggunakan subset dari seluruh molekul, dengan ribuan atribut untuk tiap molekul.

Struktur dan properti molekul merupakan sumber informasi yang sangat berharga untuk mendefinisikan suatu antibiotik. Selain informasi fisikokimia dan properti dari graf kimia, informasi lain bisa diperoleh dari perubahan struktur dinamis dan dari interaksi antara senyawa dan target. Agar informasi yang berlimpah tersebut dapat digunakan secara efektif, informasi kimia perlu dikonversi menjadi data yang penuh arti dan berguna. Tiap molekul dapat direpresentasikan oleh suatu himpunan nilai numerik atau kategorikal, yang disebut atribut. Atribut ini menunjukkan properti fisikokimia dan topologi. Data tersebut dapat dihitung melalui berbagai macam metode, kompleksitas informasi, dan waktu eksekusi. Beberapa atribut berkaitan dengan grafik molekul, sedangkan yang lainnya berkaitan dengan representasi tiga dimensi. Hasilnya, dari suatu himpunan molekul, didapatkan matriks dengan baris sebagai ID molekul, dan kolom sebagai atribut.

Dengan adanya variasi atribut kimia yang dapat digunakan (terdapat ribuan atribut), maka muncul permasalahan mengenai pemilihan atribut untuk analisis kemiripan kimia secara akurat. Atribut-atribut yang terpilih harus dapat digunakan untuk mendeskripsikan molekul-molekul dengan tingkah laku yang sama. Ektraksi informasi tersebut dapat membantu menentukan, di antara himpunan data molekul yang sangat besar, molekul-molekul yang membutuhkan penanganan yang sama.

Dengan demikian, kesulitan yang ditemui adalah bagaimana mereduksi dimensi dari matriks atribut (dari ribuan menjadi ratusan) tanpa mengurangi informasi penting. Cara yang umum adalah dengan mendeteksi korelasi antar atribut untuk menghindari redundansi. Tetapi, diketahui bahwa analisa statistik dasar tidak cukup efisien sehingga dibutuhkan metode baru.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang muncul adalah:

1. Bagaimana menentukan atribut-atribut yang saling berkorelasi sehingga dapat dieliminasi.
2. Bagaimana menggunakan hasil analisa log-linear untuk menemukan korelasi antar atribut.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan masalah sebagai berikut:

1. Penelitian diaplikasikan pada data yang telah diseleksi dari penelitian sebelumnya.
2. Analisa log-linear dijalankan pada atribut-atribut yang tidak memiliki *missing values*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mereduksi dimensi data molekul, sehingga proses penemuan antibiotik dapat berjalan secara lebih efisien.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat membantu para peneliti di bidang kimia dan biologi untuk menentukan metode penemuan antibiotik baru, sehingga kemunculan bakteri yang kebal antibiotik dapat diatasi dengan cepat.

1.6 Kontribusi

Penelitian ini menghasilkan kontribusi dengan cara menunjukkan bahwa analisa log-linear dengan model terdekomposisi dapat digunakan untuk mereduksi dimensi suatu data yang memiliki ribuan atribut.

1.7 Sistematika Penulisan

Laporan penelitian ini tersusun atas beberapa bab, sebagai berikut:

1. Bab I, Pendahuluan, berisi Latar Belakang, Permasalahan, Batasan Masalah, Tujuan, Manfaat, Kontribusi, dan Sistematika Penulisan.
2. Bab II, Kajian Pustaka dan Dasar Teori, yang akan membahas dasar-dasar ilmu yang mendukung penelitian ini.
3. Bab III, Metode Penelitian, yang akan membahas metode yang digunakan dalam menyelesaikan permasalahan.
4. Bab IV, Hasil dan Pembahasan, yang akan membahas hasil dari penelitian.

5. Bab V, Kesimpulan, berisi kesimpulan dari hasil penelitian, beserta saran untuk penelitian berikutnya.

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Knowledge Discovery

Penelitian ini berfokus pada permasalahan reduksi dimensi, lebih tepatnya pada seleksi fitur. Tujuan dari penelitian ini adalah mengurangi dimensi dari data kimia dengan memilih sebagian atribut untuk menghindari redundansi. Reduksi dimensi ini merupakan salah satu subproses dari *knowledge discovery in database* (KDD). KDD dapat dibedakan menjadi dua, yaitu numerik dan simbolis. Metode yang digunakan pada penelitian ini merupakan model grafis probabilistik, salah satu bagian dari metode numerik.

Akhir-akhir ini, data diperoleh dengan sangat cepat. Selain kuantitas, keragaman data juga berkembang secara signifikan. Akibatnya, dibutuhkan suatu instrumen yang dapat membantu manusia untuk memahami data dengan lebih baik. Hal tersebut merupakan fokus penelitian pada KDD, yang menerima masukan berupa data mentah dan menghasilkan informasi yang mudah dimengerti.

Seperti ditunjukkan pada Gambar 2.1. KDD terdiri dari beberapa langkah [4]. Langkah pertama dari KDD adalah proses memahami domain permasalahan dan keinginan klien. Kemudian, pada proses seleksi, peneliti memilih satu atau beberapa data set yang akan digunakan, disebut data target. Pada langkah ketiga, dilakukan praproses pada data target. Praproses tersebut dapat berupa penghilangan

Gambar 2.1 Diagram Proses KDD

derau atau penanganan *missing values*. Keempat, peneliti mengurangi kompleksitas data dengan transformasi, yaitu membentuk representasi dari data. Proses transformasi ini dapat dilakukan dengan mengeliminasi atau mentransformasi beberapa atribut.

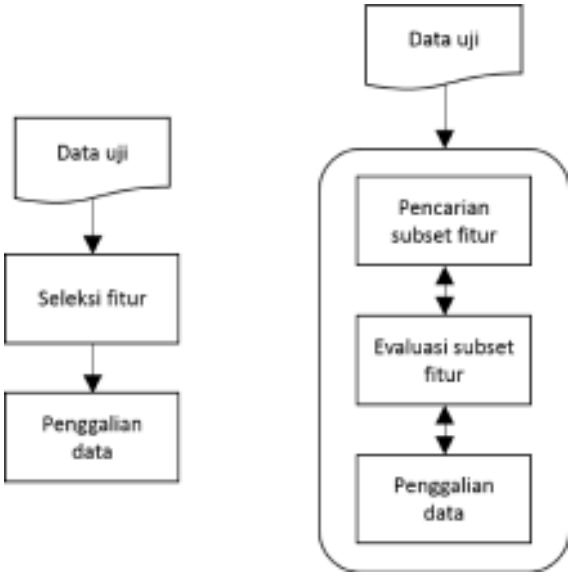
Tiga langkah berikutnya merupakan proses penggalian data dari data yang telah ditransformasi. Langkah kelima adalah mencocokkan keinginan klien terhadap metode penggalian data (klasifikasi, pengelompokan, regresi, dsb.). Setelah menentukan metode, pada langkah keenam, peneliti kemudian menentukan algoritma yang akan digunakan. Pada langkah ketujuh, peneliti menjalankan proses penggalian data. Dari langkah tersebut, peneliti dapat menemukan pola-pola yang cukup berarti dari data. Langkah berikutnya adalah interpretasi. Setelah menemukan pola, peneliti harus menginterpretasikan pola tersebut. Salah satu cara adalah dengan visualisasi. Langkah terakhir adalah mengintegrasikan pengetahuan (*knowledge*) ke proses KDD lain, melaporkannya ke pihak-pihak yang berkepentingan, atau langsung menggunakannya. Peneliti juga dapat menjalankan kembali beberapa langkah sebelumnya untuk meningkatkan kualitas pengetahuan.

2.2 Seleksi Fitur

Untuk menemukan antibiotik baru, peneliti memeriksa sejumlah molekul kimia, untuk dapat mengetahui molekul mana yang dapat berlaku sebagai antibiotik. Untuk menjalankannya, peneliti dapat melihat properti dari tiap molekul seperti jumlah atom, adanya atom-atom tertentu, adanya cincin, polaritas, eksentrisitas, dan lain-lain. Properti-properti tersebut berjumlah ribuan, yang berarti bahwa peneliti membutuhkan waktu dan ruang yang sangat besar.

Untuk meminimalkan kebutuhan ruang dan waktu, sebelum penggalian data, peneliti sebaiknya mereduksi dimensi data, sebagai proses transformasi dari KDD pada Gambar 2.1. Proses ini bisa dilakukan dengan menggabungkan beberapa atribut menjadi satu, atau bisa juga dengan mengeliminasi beberapa atribut.

Dari ribuan atribut untuk tiap molekul, dapat ditemukan beberapa redundansi. Persentase atom H berkorelasi dengan banyak atom H, banyak atom C berkorelasi dengan banyak struktur CR3X, dan beberapa korelasi tersembunyi lainnya. Dengan demikian, suatu atribut dapat dieliminasi bila atribut tersebut bisa



(a) (b)

Gambar 2.2 Dua Model Seleksi Fitur: Model *Filter* (a) dan Model *Wrapper* (b)

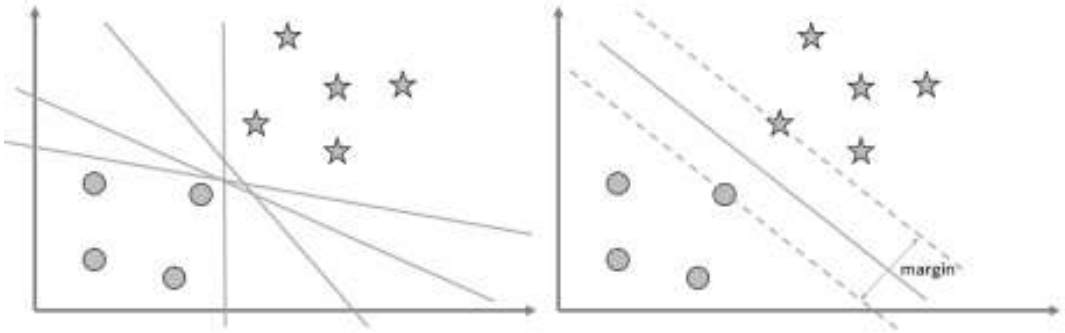
diwakilkan oleh atribut lain. Atribut-atribut yang terpilih harus dapat mendefiniskan molekul tanpa kehilangan informasi secara signifikan.

Selain redundansi, peneliti juga dapat mengeliminasi atribut yang tidak deskriptif. Sebagai contoh, jika terdapat atribut yang menyatakan banyak phospane dalam suatu molekul, tetapi semua molekul tidak memiliki phospane, maka atribut tersebut tidak deskriptif. Dia tidak dapat membedakan antara molekul antibiotik dengan molekul non-antibiotik.

Beberapa algoritma telah diajukan untuk permasalahan seleksi fitur. Seperti ditunjukkan pada Gambar 2.2, algoritma-algoritma tersebut dapat dibedakan menjadi dua kategori [10]:

1. Model *filter*. Algoritma dengan model ini menjalankan seleksi fitur secara terpisah terhadap proses penggalian data. Model ini tidak memperhatikan algoritma klasifikasi yang akan digunakan.

   Salah satu metode yang umum dipakai adalah χ^2, yang menentukan skor tiap atribut berdasarkan hubungan antara atribut dengan kelas. Fitur-fitur yang penting memiliki skor yang tinggi. Seleksi fitur kemudian dilakukan dengan



(a) (b)

Gambar 2.3 Contoh SVM pada Koordinat *Cartesian*

mengeliminasi fitur-fitur yang memiliki skor kurang dari suatu *threshold*.

2. Model *wrapper*. Model ini melakukan proses seleksi fitur sebagai bagian dari algoritma klasifikasi. Hasil dari algoritma klasifikasi akan dipertimbangkan oleh proses seleksi fitur.

   Salah satu contoh model ini dipelajari oleh Li dan Yang [15]. Mereka meneliti dua tipe model *wrapper*, yaitu tipe rekursif dan tipe non-rekursif. Pada tiap iterasi di tipe rekursif, algoritma klasifikasi dijalankan untuk mendapatkan bobot tiap fitur. Fitur dengan bobot terkecil dieliminasi. Proses iterasi berhenti bila telah tersisa fitur sebanyak *t*.

   Pada tipe non-rekursif, setelah didapatkan bobot untuk semua fitur, *t* fitur dengan skor tertinggi dipilih, sedangkan sisanya dieliminasi.

2.3 KDD Numerik

Pada subbab ini, salah satu tipe KDD akan dijelaskan, yaitu KDD numerik. Tipe ini memperhatikan properti-properti numerik dari data set, dan memberi keluaran berupa angka.

2.3.1 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu contoh KDD numerik. Pada dasarnya, SVM merupakan metode klasifikasi biner, yang berarti bahwa data diklasifikasikan menjadi dua kelas. *Gambar 2.3* menunjukkan titik-titik dan garis pada koordinat *Cartesian* dua dimensi (konsep yang sama juga berlaku untuk

(a) (b)

Gambar 2.4 Anjing Samoyed (a) dan Serigala Putih (b)

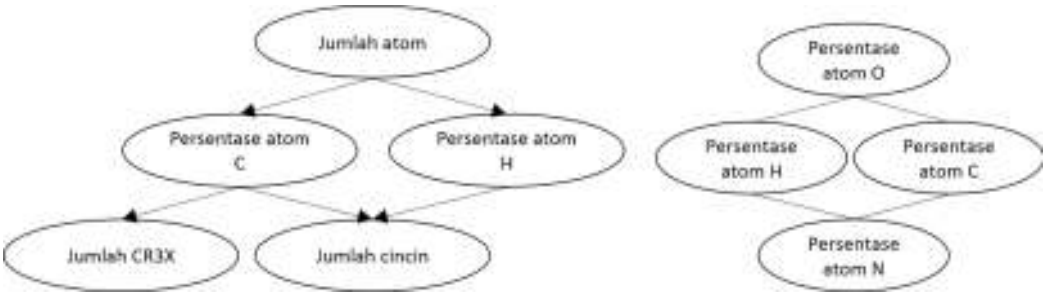
dimensi yang lebih tinggi). Untuk data set yang bisa dipisahkan secara linier, terdapat banyak garis lurus yang bertindak sebagai pemisah antara dua kelas (*Gambar 2.3*a). Tujuan SVM adalah menemukan pemisah terbaik, yaitu pemisah yang berada paling jauh dari semua titik (*Gambar 2.3*b). Untuk mengklasifikasikan titik baru, SVM membandingkan posisinya dengan permisah terbaik tersebut.

2.3.2 Deep Learning

Contoh lain dari KDD numerik adalah *deep learning* (DL) [14]. DL adalah metode pembelajaran mesin yang menciptakan beberapa level representasi untuk permasalahan klasifikasi. Dari masukan mentah, DL membangun beberapa *layer* yang semakin dalam semakin abstrak.

Sebagai contoh, pada klasifikasi citra, masukannya berupa nilai dari piksel-piksel citra. Pada *layer* pertama, DL biasanya mendeteksi tepi pada beberapa lokasi spesifik. *Layer* kedua memperhatikan susunan tepi tertentu untuk mendeteksi motif. *Layer* berikutnya akan menggabungkan beberapa motif untuk mendeteksi suatu bagian dari objek. Akhirnya, DL menemukan suatu objek berdasarkan kombinasi bagian-bagian tersebut.

Ilustrasi ditunjukkan pada *Gambar 2.4* untuk klasifikasi antara citra serigala putih dengan citra anjing Samoyed. Secara fisik, varietas anjing ini mirip dengan serigala putih. Untuk mengklasifikan citra-citra tersebut, suatu algoritma



(a) (b)

Gambar 2.5 Contoh Representasi Grafis dari Jaringan Bayesian (a) dan Jaringan Markov (b)

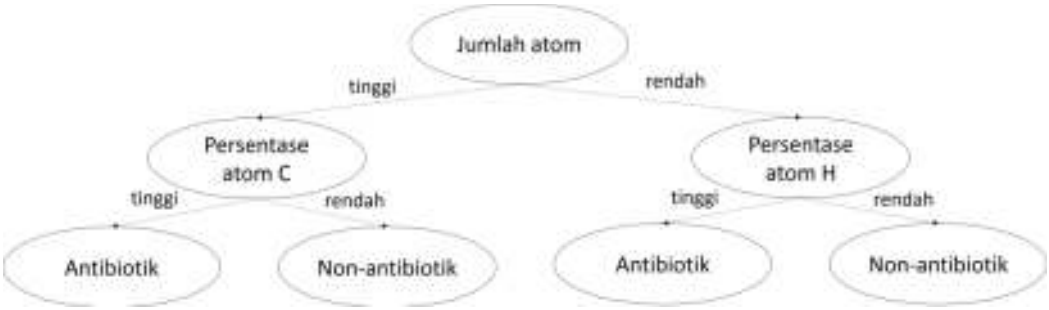
tidak boleh berfokus pada pose hewan atau warna rambutnya, karena hal-hal tersebut tidak dapat membedakan kedua hewan. Untuk mengetahui fitur apa yang bisa membedakannya, peneliti harus berkolaborasi dengan ahli hewan. Berdasarkan informasi darinya, peneliti kemudian membangun algoritma yang berfokus pada fitur-fitur tersebut.

Di sisi lain, DL mampu menentukan fitur-fitur apa saja yang penting tanpa harus diberitahu oleh programmer. Kemampuan tersebut merupakan karakteristik dari DL. DL bisa tidak sensitif terhadap fitur-fitur yang tidak relevan, tetapi sensitif terhadap fitur-fitur yang relevan.

DL telah mendapatkan hasil-hasil yang sangat bagus di berbagai area akademik, seperti citra [12] dan pengenalan suara [8], prediksi obat-obatan [16], dan identifikasi penyakit berdasarkan mutasi gen [26]. Mengenai munculnya permasalahan pencarian antibiotik baru, peneliti dapat menggunakan DL untuk mengklasifikasikan molekul ke dalam kelas antibiotik atau kelas non-antibiotik. Dengan adanya data set molekul yang memiliki ribuan atribut, DL bisa belajar untuk berfokus pada atribut-atribut yang bisa membedakan molekul dengan baik.

2.3.3 Model Grafis Probabilistik

Salah satu aspek dari KD numerik adalah model probabilistik. Sebagai contoh, untuk menemukan antibiotik baru, penting untuk mengetahui probabilitas bahwa suatu molekul dapat berperan sebagai antibiotik. Probabilitas ini bergantung



Gambar 2.6 Contoh Kecil Pohon Keputusan untuk Klasifikasi Molekul

pada atribut tiap molekul. Tiap molekul memiliki kira-kira 4000 atribut, mengenai struktur, elemen, massa, dan sebagainya. Tiap atribut dapat memiliki lebih dari dua nilai, sehingga ruang probabilitas memiliki minimal 2^{4000} nilai. Peneliti kemudian harus mengetahui jumlah kemunculan tiap nilai tersebut untuk mendapatkan probabilitas gabungan. Dari sini, peneliti mendapatkan probabilitas untuk suatu nilai spesifik, seperti:

P(antibiotik = benar | jumlah cincin = tinggi, karbon = tidak ada, massa molekul > 400,…)

Dengan banyaknya kemungkinan nilai pada data kimia, penyusunan distribusi gabungan akan menjadi sangat kompleks. Untuk mendeskripsikan distibusi yang kompleks tersebut, digunakan model grafis probabilistik, yang menggambarkan suatu distribusi sebagai graf. Graf memiliki dua macam komponen: titik dan garis.

Terdapat dua tipe model grafis probabilistik: jaringan Bayesian dan jaringan Markov [11]. Graf dari dua tipe tersebut memiliki titik sebagai atribut, dan garis sebagai interaksi probabilistik. Garis pada jaringan Bayesian merupakan garis berarah, sedangkan garis pada jaringan Markov merupakan garis tak berarah. Contoh graf dari dua tipe ini digambarkan pada Gambar 2.5.

Pada graf-graf tersebut, terdapat beberapa independensi antar atribut, disimbolkan dengan ⊥. Seperti terlihat pada *Gambar 2.5*a, banyak atom pada suatu

Tabel 2.1 Contoh Basis Data Keranjang Belanja Beberapa Pelanggan

Keranjang	Barang
1	roti, susu
2	roti, popok, bir, telur
3	susu, popok, bir, coke
4	roti, susu, popok, bir
5	roti, susu, popok, coke

molekul tidak berinteraksi langsung dengan banyak CR3X. Hal ini menunjukkan, jika diketahui persentase atom C dan H, maka distribusi banyak cincin dapat diketahui, tanpa harus mengetahui banyak atom. Independensi ini ditulis sebagai (cincin ⊥ atom|persentase C, persentase H). Selain itu, jika diketahui banyak atom, maka persentase C independen terhadap persentase H.

Sementara itu, pada *Gambar 2.5*b, terdapat dua independensi, (C ⊥ H|N, O) dan (N ⊥ O|C, H).

2.4 KD Simbolis

Subbab ini menjelaskan tentang bagian lain dari KD, yaitu KD dengan pendekatan simbolis. Pada pendekatan ini, hasil yang diperoleh berupa simbol. Contoh pada klasifikasi antar dua kelas molekul: antibiotik dan non-antibiotik. Dengan pohon keputusan, didapatkan sebuah pohon yang dapat ditelusuri untuk menentukan kelas sebuah molekul. Contoh kecil terdapat pada Gambar 2.6. Berdasarkan pohon ini, jika sebuah molekul memiliki atom cukup banyak dan persentase atom C yang cukup tinggi, maka molekul tersebut adalah antibiotik.

2.4.1 Association Rule Mining

Pendekatan simbolis lainnya adalah *association rule mining* (ARM), diajukan oleh Agrawal *et al.* [1]. Dalam sebuah basis data mengenai transaksi pelanggan di sebuah supermarket, mereka mencoba menemukan barang-barang apa saja yang dibeli secara bersamaan. Basis data ini menyimpan data tiap keranjang belanja, yaitu barang-barang yang dibeli oleh tiap pelanggan. ARM kemudian mengetahui, sebagai contoh, jika di dalam suatu keranjang terdapat roti, maka di

Tabel 2.2 Contoh Beberapa *Support* dan *Confidence* dari Beberapa Aturan

Aturan	***Support***	***Confidence***
{susu,popok} → {bir}	0,40	0,67
{susu,bir} → {popok}	0,40	1,00
{popok,bir} → {susu}	0,40	0,67
{bir} → {susu,popok}	0,40	0,67
{popok} → {susu,bir}	0,40	0,50
{susu} → {popok,bir}	0,40	0,50

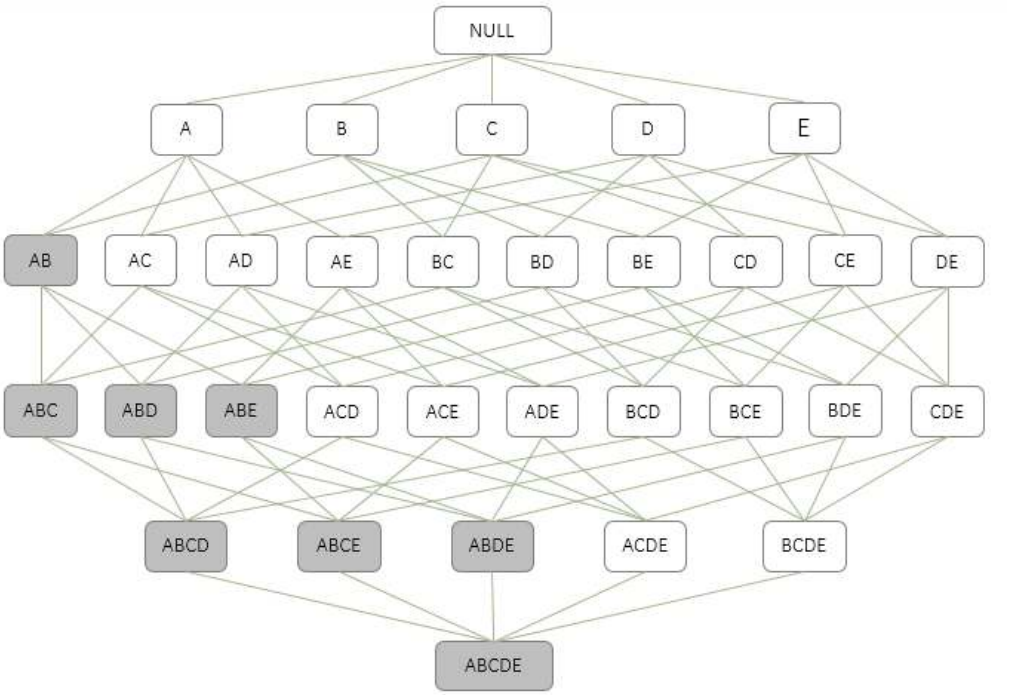
dalamnya juga terdapat susu. Supermarket bisa memaksimalkan keuntungan melalui aturan {roti} → {susu} ini dengan meletakkan kedua barang tersebut berdekatan.

Untuk mengevaluasi aturan {roti} → {susu}, digunakan metrik *support* (*sup*) dan *confidence* (*conf*). Dari semua keranjang, *sup* menunjukkan berapa banyak diantara mereka yang memiliki *itemset* {roti,susu}. Kemudian, dari semua keranjang yang memiliki roti, *conf* menunjukkan berapa banyak diantara mereka yang memiliki susu (contoh pada Tabel 2.1 dan Tabel 2.2). Berdasarkan ukuran tersebut, tugas ARM adalah mencari aturan yang memiliki *sup* dan *conf* yang cukup signifikan (dengan membandingkannya terhadap *minsup* dan *minconf*). Dengan *brute force*, semua aturan dipertimbangkan, dihitung *sup* dan *conf*-nya, lalu dieliminasi jika kurang *sup* dan *conf* kurang dari batas.

Jika terdapat *d* macam barang, maka dengan *brute force* akan terdapat 2^d *itemset*. Untuk menghindari kebutuhan ruang yang eksponensial tersebut, maka ARM dipecah menjadi dua tahap:

1. *Frequent itemset mining* (FIM). Pada tahap ini, semua *itemset* dipertimbangkan, kemudian beberapa *frequent itemset* dipilih, yaitu *itemset* dengan *sup* memenuhi *minsup*.
2. Generasi aturan. Setelah semua *frequent itemset* diperoleh, maka dilakukan proses generasi aturan dengan *conf* yang tinggi dari tiap *frequent itemset*.

Komputasi pada tahap FIM sangatlah kompleks, sedangkan komputasi pada tahap generasi aturan relatif lebih mudah, karena aturan bisa langsung



Gambar 2.7 Representasi Lattice dari Barang A,B,C,D, dan E, dengan Contoh Prinsip Apriori

didapatkan dari hasil tahap FIM. Dengan demikian, tahap terberat ARM adalah tahap FIM.

Salah satu cara untuk mendapatkan semua *itemset* secara sistematis adalah dengan representasi *lattice*, seperti ditunjukkan pada Gambar 2.7. Proses dimulai dari generasi *itemset* dengan panjang 1. Kemudian, *itemset-itemset* tersebut dikombinasikan untuk mendapatkan *itemset* dengan panjang 2, dan seterusnya. *Sup* dihitung untuk tiap *itemset*. Cara ini masih terlalu kompleks karena banyak kandidat *itemset* bersifat eksponensial terhadap banyak barang.

Salah satu teknik untuk mereduksi kompleksitas adalah dengan mengurangi banyak kandidat *itemset*. Prinsip apriori berjalan berdasarkan teknik tersebut dan mendefinisikan:

- Jika suatu *itemset* sering muncul, maka semua subsetnya juga sering muncul.
- Jika suatu *itemset* jarang muncul, maka semua supersetnya juga jarang muncul.

Tabel 2.3 Tabel Kontingensi Data PJB

Profesi	Jenis Kelamin	Bacaan		Total
		Ilmiah	Misteri	
Politikus	Pria	15	15	30
	Wanita	10	15	25
	Total	25	30	55
Administrator	Pria	10	30	40
	Wanita	5	10	15
	Total	15	40	55
Penari	Pria	5	5	10
	Wanita	10	25	35
	Total	15	30	45

Sumber: [23]

Jadi, pada saat pembangunan *lattice*, jika ditemui suatu *itemset* yang jarang muncul, *itemset* tersebut dieliminasi, bersamaan dengan seluruh supersetnya. Prinsip apriori ini diilustrasikan pada Gambar 2.7.

Setelah memperoleh semua *itemset* yang sering muncul, maka langkah selanjutnya adalah membangkitkan aturan asosiasi dari *itemset*. Sebagai contoh, dari *itemset* {A, B, C}, aturan-aturan yang dibangkitkan adalah {A} → {B, C}, {A, C} → {B}, dan seterusnya. *Conf* untuk tiap aturan dihitung, kemudian aturan-aturan dengan *conf* memenuhi *minconf* dipilih. Aturan-aturan yang terpilih tersebut merupakan hasil dari ARM.

2.5 Analisis Log-linear

Subbab ini menjelaskan analisis log-linear (ALL) dan representasi grafiknya, yang merupakan salah satu model grafis probabilistik dari KD numerik.

Misal terdapat data set dari beberapa orang mengenai profesi dan jenis kelamin. Variabel dan domain data set tersebut adalah:

1. Profesi (P), Dom(P) = {politikus, administrator, penari}
2. Jenis kelamin (J), Dom(J) = {pria, wanita}

Hubungan antara kedua variabel diskrit tersebut dapat dipelajari menggunakan tes asosiasi χ^2 dua arah. Tetapi, jika terdapat variabel ketiga:

3. Bacaan (B), Dom(B) = {ilmiah, misteri}

maka dibutuhkan analisis frekuensi multiarah untuk mempelajari asosiasi dua arah dan tiga arah antara ketiga variabel tersebut. ALL adalah suatu ekstensi dari analisis frekuensi multiarah yang mencoba menemukan relasi statistik antara tiga atau lebih variabel diskrit. ALL akan membangun sebuah model (seperti pada Persamaan 2.6) untuk menemukan logaritma dari frekuensi harapan.

Untuk selanjutnya, data set mengenai profesi, jenis kelamin, dan bacaan disebut data set PJB. Dari PJB (dengan tabel kontingensi ditunjukkan pada Tabel 2.3), ALL mencoba menjawab beberapa pertanyaan mengenai relasi antar variabel. Apakah profesi seseorang berhubungan dengan jenis kelaminnya? Apakah jenis kelamin seseorang berkaitan dengan bacaannya? Apakah ada relasi tiga arah antara profesi, jenis kelamin, dan bacaan? Dengan mengetahui tipe bacaan seseorang, apakah bisa profesi orang tersebut bisa diketahui?

Untuk melakukan analisis frekuensi multiarah dengan ALL, dibangun sebuah model linear terhadap logaritma frekuensi harapan tiap sel. Contoh model tersebut ditunjukkan pada Persamaan 2.6, dengan tiap komponen merepresentasikan satu asosiasi. Seiring dengan bertambahnya jumlah variabel, jumlah asosiasi juga bertambah. Karena terdapat tiga variabel pada PJB, maka terdapat tujuh asosiasi: satu asosiasi tiga arah, tiga asosiasi dua arah, dan tiga asosiasi satu arah. Model pada Persamaan 2.6 mengandung semua asosiasi yang mungkin. Untuk meminimalkan kompleksitas sebuah model, ALL mencoba menentukan asosiasi mana yang akan dieliminasi. Asosiasi yang akan dieliminasi adalah asosiasi yang tidak signifikan.

Dengan ribuan variabel, banyak asosiasi akan menjadi sangat besar sedemikian hingga tidak praktis untuk menguji semua asosiasi. Batasan ini dapat diselesaikan menggunakan Chordalysis, suatu model grafis probabilistik yang akan dijelaskan kemudian.

2.5.1 Goodness-of-fit

Goodness-of-fit dari suatu ALL mengukur kecocokan antara frekuensi observasi dan frekuensi harapan. Dari dua variabel, terdapat satu asosiasi dua arah. Seperti disebutkan sebelumnya, untuk uji *goodness-of-fit* asosiasi dua arah, dapat

Tabel 2.4 Derajat Kebebasan (*df*), G^2, dan Probabilitas Didapatkannya G^2 yang Lebih Besar

Asosiasi	*df*	G^2	Probabilitas
total	11	48,09	<0,05
B	1	13,25	<0,05
J	1	0,16	>0,05
P	2	1,32	>0,05
J × B	1	0,62	>0,05
P × B	2	4,42	>0,05
P × J	2	27,12	<0,05
P × J × B	2	1,85	>0,05

digunakan uji χ^2:

$$\sum_i \frac{(O_i - E_i)^2}{E_i} \tag{2.1}$$

Untuk tiap sel *i* dari tabel kontingensi, O_i adalah frekuensi observasi dan E_i adalah frekuensi harapan.

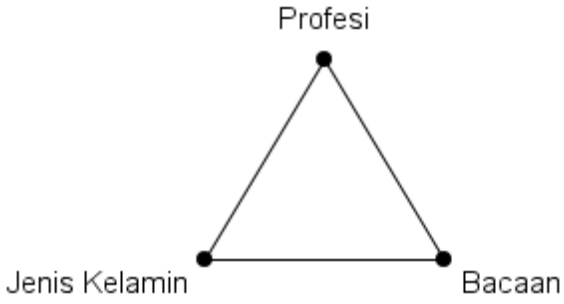
Tetapi, karena terdapat asosiasi-asosiasi multiarah di ALL, digunakan uji G^2. Uji ini dapat menggantikan uji χ^2 dalam mengukur kecocokan antara frekuensi observasi dan frekuensi harapan. Distribusi G^2 sama dengan distribusi χ^2, sehingga tabel χ^2 dapat digunakan untuk mengevaluasi level signifikan. Statistik G^2 memiliki persamaan:

$$2\sum_i O_i \ln\left(\frac{O_i}{E_i}\right) \tag{2.2}$$

Statistik G^2 juga digunakan karena memiliki properti *additivity*, yang tidak dimiliki oleh χ^2 [23]. Dengan properti ini, dalam analisis dua arah dari variabel A dan B, uji asosiasi total G_T^2 adalah jumlah dari uji orde pertama, G_A^2 dan G_B^2, dan uji asosiasi G_{AB}^2:

$$G_T^2 = G_A^2 + G_B^2 + G_{AB}^2 \tag{2.3}$$

Properti ini berguna untuk menguji frekuensi residual dari suatu model.



Gambar 2.8 Representasi Grafis untuk Model Log-linear Jenuh dari Data PJB

2.5.2 Perhitungan G^2

Untuk menentukan apakah suatu asosiasi signifikan atau tidak, G^2 dari asosiasi tersebut akan dipertimbangkan. Pada subbab ini, perhitungan beberapa G^2 dari data set PJB akan dijelaskan.

2.5.2.1 G^2 Total

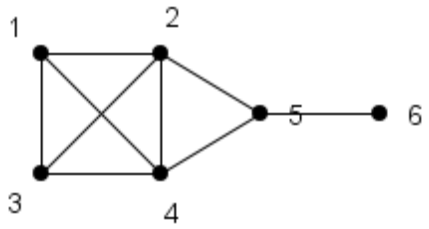
G^2 total mengukur asosiasi total. Untuk itu, diperlukan frekuensi observasi dan frekuensi harapan dari tiap sel pada tabel kontingensi. Frekuensi observasi data PJB dapat dilihat pada Tabel 2.3, sedangkan frekuensi harapan harus dihitung terlebih dahulu. Untuk asosiasi total, frekuensi harapan tiap sel bernilai sama, yaitu rata-rata, didapatkan dari pembagian frekuensi total oleh banyak sel. Pada data PJB, karena terdapat 155 orang, maka frekuensi harapan tiap sel *i* adalah:

$$E_i = \frac{155}{12} \approx 12{,}9167 \tag{2.4}$$

Dengan Tabel 2.3 sebagai frekuensi observasi, 12,9167 sebagai frekuensi harapan, dan menggunakan Persamaan 2.2, maka didapatkan G_T^2 = 48,09. Derajat kebebasan *df* diperoleh dari banyak sel dikurangi 1, sehingga *df* untuk asosiasi total adalah 11.

2.5.2.2 G^2 Orde Pertama

Pada data PJB, terdapat tiga G^2 orde pertama: G_P^2, G_J^2, dan G_B^2, satu untuk tiap variabel. *Goodness-of-fit* dari G^2 orde pertama mengukur keseimbangan antar



Gambar 2.9 Contoh Graf dengan 6 Titik yang Memiliki *Clique* Maksimal: {1,2,3,4}, {2,4,5}, dan {6}; *Clique* Maksimum: {1,2,3,4}; *Separator* Minimal: {2,4} dan {5}; *Separator* Minimum: {5}; dan *Separator*-(2,6) Minimal: {5}

nilai suatu variabel.

Untuk tiap variabel frekuensi harapan E diperoleh dari pembagian frekuensi total dengan banyak macam nilai, sedangkan *df* diperoleh dari banyak macam nilai dikurangi 1. Dengan demikian, untuk variabel B, didapatkan $E_{\text{ilmiah}} = E_{\text{misteri}} = 155/2 = 77{,}5$ dan $df = 1$. Frekuensi observasi diperoleh dari tabel kontingensi pada Tabel 2.3, sehingga diketahui $O_{\text{ilmiah}} = 55$ dan $O_{\text{misteri}} = 100$. Berdasarkan Persamaan 2.2, diperoleh G_B^2:

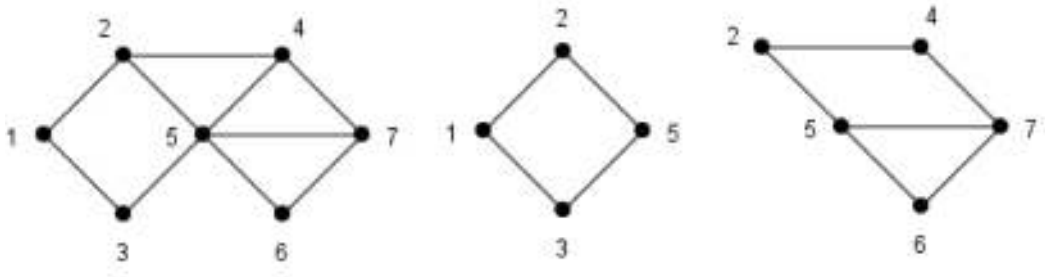
$$G_B^2 = 2\left[55 \ln\left(\frac{55}{77{,}5}\right) + 100 \ln\left(\frac{100}{77{,}5}\right)\right] = 13{,}25 \tag{2.5}$$

Perhitungan yang sama dilakukan untuk dua variabel lainnya. G^2 dari kedua variabel tersebut ditunjukkan pada Tabel 2.4.

2.5.2.3 G^2 Orde Kedua

Karena data PJB memiliki tiga variabel, maka terdapat tiga G^2 orde kedua untuk tiga asosiasi dua arah: $P \times J$, $P \times B$, dan $J \times B$.

Untuk asosiasi $J \times B$, frekuensi harapan dari semua sel pria-ilmiah didapatkan dari perkalian antara jumlah pria dengan jumlah ilmiah, kemudian membaginya dengan frekuensi total. Dengan demikian, ketiga sel pria-ilmiah memiliki frekuensi harapan $80 \times 55/155 = 28{,}387$. Derajat kebebasan mereka adalah $(2 - 1)(2 - 1) = 1$. Frekuensi harapan dari kombinasi-kombinasi lainnya kemudian dihitung, dan melalui Persamaan 2.2, diperoleh $G_{JB}^2 = 0{,}62$.



(a) (b) (c)

Gambar 2.10 Dekomposisi Graf G (a) menjadi Dua Komponen (b) dan (c) (Lauritzen, 2011)

2.5.2.4 G^2 Orde Ketiga

Data PJB memiliki tiga variabel, sehingga hanya terdapat satu asosiasi tiga arah. G^2 orde ketiga untuk asosiasi $P \times J \times B$ adalah $G^2_{PJB} = 1{,}85$ (perhitungan tidak ditampilkan, dan *df* dari asosiasi tersebut adalah (3 – 1)(2 – 1)(2 – 1) = 2. G^2 dari setiap asosiasi ditampilkan pada Tabel 2.4.

2.5.3 Model Log-linear

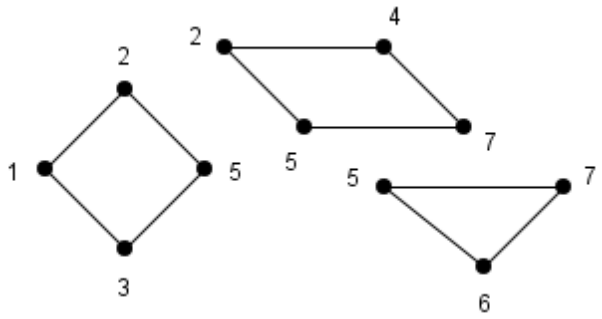
Salah satu aplikasi ALL adalah untuk menemukan model terbaik dalam memprediksi frekuensi harapan *E* untuk tiap sel.

Model log-linear direpresentasikan sebagai persamaan untuk menemukan logaritma dari *E*. Dari data PJB, dapat dibangun suatu model yang mengandung seluruh asosiasi yang dimungkinkan. Model ini disebut model jenuh, ditulis sebagai:

$$\ln E = \theta + \lambda_P + \lambda_J + \lambda_B + \lambda_{PJ} + \lambda_{PB} + \lambda_{JB} + \lambda_{PJB} \qquad (2.6)$$

dengan θ adalah konstanta, dan komponen-komponen λ mewakili suatu asosiasi. Tiap λ memiliki nilai sebanyak jumlah level, dan nilai-nilai tersebut memiliki total 0. Sebagai contoh, karena terdapat tiga level pada variabel profesi, maka λ_P memiliki tiga kemungkinan nilai: untuk politikus ($\lambda_{P\,\mathrm{pol}}$), adminstrator ($\lambda_{P\,\mathrm{adm}}$), dan penari ($\lambda_{P\,\mathrm{tar}}$), dengan

$$\lambda_{P\text{pol}} + \lambda_{P\text{adm}} + \lambda_{P\text{tar}} = 0 \tag{2.7}$$



Gambar 2.11 Graf-graf Prima Maksimal dari Graf pada Gambar 2.10

Suatu model log-linear dapat bertipe hierarkis atau non-hierarkis. Model hierarkis dapat direpresentasikan oleh asosiasi-asosiasinya yang memiliki orde tertinggi dalam kurung siku. Sebagai contoh, selain terdiri dari asosiasi λ_{PJ} dan λ_B , model [PJ][B] juga terdiri dari asosiasi λ_P dan λ_J.

Skor suatu model diperoleh dari perhitungan G^2 dan mengevaluasi signifikansinya. Berdasarkan Persamaan 2.3, G^2 suatu model diperoleh dari pengurangan G^2 tiap asosiasi dari G^2 total, sehingga menampilkan frekuensi residualnya.

Sebagai contoh, model [PJ][B] memiliki persamaan:

$$\ln E = \theta + \lambda_P + \lambda_J + \lambda_B + \lambda_{PJ} \tag{2.8}$$

Dengan G^2 pada Tabel 2.4, model tersebut memiliki skor G^2:

$$G^2_{[\text{PJ}][\text{B}]} = G^2_T - G^2_{PJ} - G^2_P - G^2_J - G^2_B$$

$$G^2_{[\text{PJ}][\text{B}]} = 48{,}09 - 27{,}12 - 0{,}16 - 1{,}32 - 13{,}25$$

$$G^2_{[\text{PJ}][\text{B}]} = 6{,}24 \tag{2.9}$$

dan $df = 11 - 2 - 1 - 2 - 1 = 5$. Residual tersebut tidak signifikan (>0,05 pada tabel χ^2), maka model tersebut bagus.

Kemudian, model hierarkis lain yang lebih kompleks akan dievaluasi, misal model [PJ][PB]. Dengan cara yang sama seperti contoh pada Persamaan 2.9,

diperoleh $G^2_{[PJ][PB]} = 2{,}48$ dan *df* = 3. Model ini juga bagus, sehingga harus ditentukan pemilihan antara model [PJ][B] dengan model [PJ][PB].

Kedua model bersifat hierarkis, dan [PJ][B] adalah submodel dari [PJ][PB] (seluruh komponen λ pada [PJ][B] dapat ditemukan pada [PJ][PB]). Dalam kondisi tersebut, perbedaan G^2 antara keduanya juga merupakan G^2 [23]:

$$G^2_{\text{diff}} = G^2_{[PJ][B]} - G^2_{[PJ][PB]} = 6{,}24 - 2{,}48 = 3{,}76 \tag{2.10}$$

dengan *df* = 5 – 3 = 2. Perbedaan tersebut tidak signifikan, sehingga model yang lebih sederhana dapat dipilih, yaitu [PJ][B].

2.5.4 Model Grafis

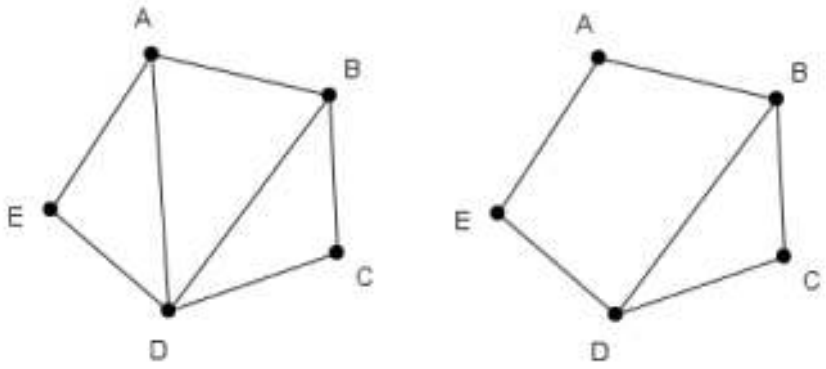
Suatu model log-linear bersifat grafis jika, kapanpun terdapat seluruh asosiasi dua arah yang berasal dari orde yang lebih tinggi, model tersebut juga mengandung orde tinggi tersebut. Karena adanya asosiasi dua arah, maka model grafis bisa digambarkan sebagai graf, dengan titik sebagai variabel dan garis sebagai asosiasi dua arah antar variabel. Model jenuh pada Persamaan 2.6 termasuk model grafis, dengan graf yang ditunjukkan pada Gambar 2.8. Sementara itu, model log-linear:

$$\ln E = \theta + \lambda_P + \lambda_J + \lambda_B + \lambda_{PJ} + \lambda_{PB} + \lambda_{JB} \tag{2.11}$$

tidak mengandung komponen λ_{PJB}, walaupun model tersebut terdiri dari λ_{PJ}, λ_{PB}, dan λ_{JB}. Akibatnya, model tersebut tidak bisa digambarkan sebagai suatu graf. Model tersebut tidak termasuk model grafis.

Pada tiap graf, terdapat properti *clique* maksimal dan *separator* minimal, diilustrasikan pada Gambar 2.9. *Clique* dari graf *G* adalah subset dari titik-titik *G* yang grafnya komplet (tiap dua titik terhubung oleh satu garis). *Clique* maksimal adalah suatu *clique* yang bukan merupakan subset dari *clique* lain. *Clique* maksimum adalah *clique* dengan jumlah titik paling banyak.

Sebuah *separator* dari *G* adalah subset dari titik-titik *G* yang jika dihapus akan menyebabkan *G* tidak terkoneksi. *Separator* minimal adalah *separator* yang tidak memiliki *separator* lain dalam subsetnya. *Separator* dengan jumlah titik paling sedikit adalah *separator* minimum.



(a) (b)

Gambar 2.12 Contoh Graf *Chordal* (a) dan Non-*Chordal* (b)

Sepasang titik juga memiliki properti *separator* minimal. *Separator*-(*a*,*b*) minimal adalah subset dari titik yang jika dihapus akan menyebabkan *a* tidak terhubung dengan *b*, dan subset tersebut tidak memiliki *separator* lain dalam subsetnya.

2.5.5 Model *Decomposable*

Model log-linear grafis disebut *decomposable* bila representasi grafnya bisa didekomposisi. Dekomposisi graf *G* adalah partisi dari titik-titik *V* menjadi tiga subset independen (*A, B, S*), dengan:

- $A \neq \emptyset$ dan $B \neq \emptyset$,
- *S* membentuk subgraf komplet,
- *A* dan *B* tidak terhubung dalam $G - S$.

Hasil dari dekomposisi *G* disebut komponen dari *G*. Komponen-komponen tersebut adalah subgraf yang terbentuk dari $A \cup S$ dan $B \cup S$. Salah satu contoh dekomposisi diilustrasikan pada *Gambar 2.10*. Titik-titik pada graf *G* pada *Gambar 2.10*a dapat dipartisi menjadi tiga subset: $A = \{1,3\}$, $B = \{4,6,7\}$, dan $S = \{2,5\}$. Partisi tersebut menghasilkan komponen dari $A \cup S$ pada *Gambar 2.10*b dan $B \cup S$ pada *Gambar 2.10*c.

Setiap graf dapat didekomposisi secara rekursif sampai pada subgrafnya

yang bersifat prima maksimal, yaitu subgraf yang tidak bisa didekomposisi lagi. Graf *G* pada *Gambar 2.10*a dapat didekomposisi menjadi tiga subgraf yang ditunjukkan pada Gambar 2.11.

Suatu graf disebut *decomposable* jika graf tersebut komplet atau jika graf tersebut bisa didekomposisi menjadi graf-graf lain yang *decomposable*. Dengan definisi tersebut, maka seluruh graf prima maksimal dari graf *decomposable* adalah *clique*. Oleh karena itu, graf pada Gambar 2.10a tidak *decomposable*, karena terdapat dua komponen (Gambar 2.11) yang tidak *decomposable*.

Selain itu, suatu graf bersifat *decomposable* jika dan hanya jika graf tersebut *chordal*. Pada graf *chordal*, setiap siklus dengan panjang lebih dari tiga memiliki *chord*, yaitu suatu garis yang bukan bagian dari siklus tetapi menghubungkan dua titik pada siklus. Graf pada Gambar 2.12a termasuk graf *chordal*, tetapi graf pada Gambar 2.12b bukan merupakan graf *chordal* karena siklus (A,B,D,E) tidak memiliki *chord*.

Model *decomposable* adalah satu-satunya model log-linear yang memiliki *maximum likelihood estimates* yang tertutup [7]. Selain itu, model *decomposable* juga memiliki keuntungan mengenai *separator* minimal dan *clique* maksimal. Properti-properti tersebut dapat diperoleh dalam waktu linear, menggunakan *Lexicographic Breadth First Search* atau menggunakan *Maximum Cardinality Search* [2]. Keuntungan lain adalah mengenai statistik G^2, yang dapat diperoleh melalui observasi struktur grafnya.

BAB 3
METODE PENELITIAN

Pada bab ini, akan dijelaskan metode yang digunakan dalam proses seleksi fitur data kimia untuk penemuan antibiotik baru.

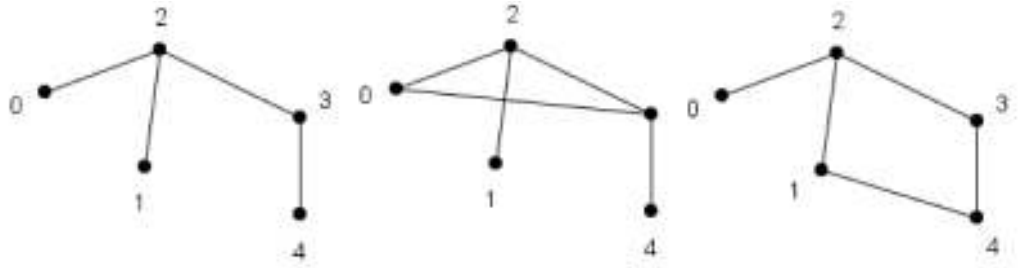
3.1 Chordalysis

Terdapat dua cara untuk menentukan asosiasi-asosiasi yang akan dipilih dalam model log-linear, eliminasi mundur dan seleksi maju. Eliminasi mundur dimulai dari model jenuh dan satu persatu mengeliminasi asosiasi-asosiasi yang tidak signifikan. Seleksi maju dimulai dari model kosong dan menambah asosiasi secara iteratif sampai penambahan tidak lagi signifikan. Metode ALL yang telah ada saat ini mempertimbangkan semua kemungkinan asosiasi untuk menentukan yang mana yang akan ditambahkan atau dieliminasi. Metode tersebut menjadi tidak praktis jika jumlah variabel bertambah besar, karena jumlah asosiasi bertambah secara eksponensial terhadap jumlah variabel.

Sebagai salah satu metode numerik grafis, Chordalysis kemudian membantu ALL dalam menentukan asosiasi mana yang akan dimasukkan ke dalam model log-linear [18]. Chordalysis berfokus pada model log-linear yang *decomposable*, karena model tersebut memiliki beberapa kelebihan, seperti telah dijelaskan pada bab sebelumnya. Metode ini merupakan pendekatan seleksi maju, karena bermula dari graf kosong, lalu secara iteratif menambahkan satu garis yang membuat graf tetap *chordal*.

3.2 Contoh Data Set

Untuk dapat memahami Chordalysis lebih baik, pada subbab ini akan dideskripsikan satu contoh data set kecil *D*, yang berjudul Congressional Voting Recors Data Set, yang tersedia di situs UCI Machine Learning Repository. Data ini berisi tentang pilihan dari 435 orang anggota kongres Amerika Serikat pada 16 topik (seperti pembagian biaya air, pembekuan upah fisikawan, bantuan kepada El Salvador, dan lain-lain). Dengan demikian, data ini memiliki $N = 435$ dan jumlah



(a) (b) (c)

Gambar 3.1 Contoh *M** (a) pada Suatu Iterasi, Kandidat yang Memenuhi Syarat (b), dan Kandidat yang Tidak Memenuhi Syarat (c)

variabel M = 17. Variabel-variabel diskrit tersebut disimbolkan dengan $\mathcal{V} = \{V_0,...,V_{16}\}$. Variabel pertama berisi partai politik dari anggota kongres, dengan domain Dom(V_0) = {dem,rep}. Enam belas variabel lainnya mendeskripsikan pilihan tiap anggota, Dom(V_1) = Dom(V_2) = … = Dom(V_{16}) = {y,n,a}, yang berarti ya (y), tidak (n), atau tidak diketahui (a). Domain dari dua atau lebih variabel adalah kombinasi dari domain tiap variabel, contohnya:

Dom({V_0, V_1}) = {(dem,y),(dem,n),(dem,a),(rep,y),(rep,n),(rep,a)}

Chordalysis kemudian diaplikasikan pada data set tersebut untuk menemukan asosiasi-asosiasi signifikan, yang digambarkan sebagai garis pada graf *chordal*. Model awal adalah graf kosong, yaitu graf tanpa titik dan garis.

3.3 Pembangkitan Model Kandidat

Langkah pertama pada tiap iterasi adalah pembangkitan model-model kandidat. Salah satu dari mereka akan menggantikan model terbaik dari iterasi sebelumnya (*M**). Suatu kandidat M^c merupakan *M** yang ditambahi satu garis. Penambahan satu garis tersebut harus membuat graf tetap *chordal*. Untuk menemukan garis tersebut, dilihat konektivitas antara dua titik di ujungnya. Suatu garis (*a*, *b*) bisa ditambahkan jika:

- *a* dan *b* tidak terhubung (mereka ada dalam komponen yang berbeda), atau
- *a* dan *b* terhubung, dengan semua jalur tanpa *chord* memiliki panjang dua.

Tabel 3.1 Perhitungan $H(\{0,4\})$ untuk Data *D*

	a	n	y	Total
dem	n = 8 p = 0,018 p ln p = -0,073	n = 245 p = 0,563 p ln p = -0,323	n = 14 p = 0,032 p ln p = -0,111	n = 267 p = 0,613 p ln p = -0,507
rep	n = 3 p = 0,007 p ln p = -0,034	n = 2 p = 0,005 p ln p = -0,024	n = 163 p = 0,375 p ln p = -0,368	n = 168 p = 0,387 p ln p = -0,426
Total	n = 11 p = 0,025 p ln p = -0,107	n = 247 p = 0,568 p ln p = -0,347	n = 177 p = 0,407 p ln p = -0,479	n = 435 p = 1 p ln p = -0,934

Karena suatu garis menggambarkan asosiasi antara dua variabel, maka penambahan satu garis relevan dengan penambahan satu asosiasi dua arah (atau orde yang lebih tinggi) pada model log-linear. Salah satu contoh ditunjukkan pada *Gambar 3.1*. Anggap bahwa pada suatu iterasi, diperoleh model [02][12][23][34] (*Gambar 3.1*a) dengan persamaan:

$$\ln E = \theta + \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_{0,2} + \lambda_{1,2} + \lambda_{2,3} + \lambda_{3,4} \quad (3.1)$$

Suatu kandidat haruslah bersifat *decomposable*. Dengan demikian, Chordalysis tidak membangkitkan seluruh model-model dengan menambahkan satu garis, tetapi hanya beberapa model yang *chordal*, contohnya [023][12][34] pada *Gambar 3.1*b dengan persamaan:

$$\ln E = \theta + \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_{0,2} + \lambda_{0,3} + \lambda_{1,2} + \lambda_{2,3} + \lambda_{3,4} + \lambda_{0,2,3} \quad (3.2)$$

Model-model yang tidak hierarkis tidak terpilih sebagai kandidat, karena grafnya tidak *chordal* (*Gambar 3.1*c). Proses seleksi ini memperkecil ruang pencarian secara signifikan.

3.4 Skor Kandidat

Seperti ALL, Chordalysis juga menggunakan statistik G^2 untuk menghitung skor suatu model. Berdasarkan Persamaan 2.2, skor G^2 dari suatu model *M* adalah:

$$G^2(M) = 2 \cdot \sum_{x \in Dom(V)} O_x \ln\left(\frac{O_x}{E_x}\right) \tag{3.3}$$

yang dapat ditulis sebagai berikut:

$$G^2(M) = 2 \cdot \sum_{x \in Dom(V)} O_x \left(\ln O_x - \ln E_x\right)$$

$$G^2(M) = 2 \cdot \left(\sum_{x \in Dom(V)} O_x \ln O_x - \sum_{x \in Dom(V)} O_x \ln E_x\right)$$

$$G^2(M) = 2 \cdot \left(\sum_{x \in Dom(V)} O_x \ln O_x - \sum_{x \in Dom(V)} O_x \ln N - \left(\sum_{x \in Dom(V)} O_x \ln E_x - \sum_{x \in Dom(V)} O_x \ln N\right)\right)$$

$$G^2(M) = 2 \cdot \left(\sum_{x \in Dom(V)} O_x \left(\ln O_x - \ln N\right) - \sum_{x \in Dom(V)} O_x \left(\ln E_x - \ln N\right)\right)$$

$$G^2(M) = 2 \cdot \left(\sum_{x \in Dom(V)} O_x \ln\left(\frac{O_x}{N}\right) - \sum_{x \in Dom(V)} O_x \ln\left(\frac{E_x}{N}\right)\right)$$

$$G^2(M) = 2 \cdot N\left(\sum_{x \in Dom(V)} \frac{O_x}{N} \ln\left(\frac{O_x}{N}\right) - \sum_{x \in Dom(V)} \frac{O_x}{N} \ln\left(\frac{E_x}{N}\right)\right) \tag{3.4}$$

dengan *N* sebagai banyak datum. Selain itu, jika didefinisikan $\hat{p}_V(x) = O_x / N$ dan $\hat{p}_\mu(x) = E_x / N$, maka Persamaan 3.4 dapat ditulis sebagai:

$$G^2(M) = 2 \cdot N\left(\sum_{x \in Dom(V)} \hat{p}_V(x) \ln \hat{p}_V(x) - \sum_{x \in Dom(V)} \hat{p}_V(x) \ln \hat{p}_\mu(x)\right) \tag{3.5}$$

Komponen pertama pada Persamaan 3.5 merupakan entropi *–H(V)*. Kemudian, menurut [17], untuk model *decomposable*, komponen kedua pada Persamaan 3.5 dapat ditulis menurut entropi *H*:

$$-\sum_{x \in Dom(V)} \hat{p}_V(x) \ln \hat{p}_\mu(x) = \sum_{C \in C'} H(C) - \sum_{S \in S'} H(S) \tag{3.6}$$

dengan *C'* adalah himpunan *clique* maksimal, dan *S'* adalah himpunan *separator* minimal. Dengan demikian, Persamaan 3.6 dapat disederhanakan menjadi:

$$G^2(M) = 2 \cdot N\left(\sum_{C \in C'} H(C) - \sum_{S \in S'} H(S) - H(V)\right) \tag{3.7}$$

Contoh perhitungan entropi *H* ditunjukkan pada Tabel 3.1, dengan *n* adalah banyak datum dan *p* adalah probabilitas. Entropi didapatkan dari $-\sum_i p_i \ln p_i$. Dengan

demikian, untuk himpunan variabel {0,4}, entropinya adalah $H(\{0,4\}) = 0{,}934$.

Pada Chordalysis, M^c dan M^* bersifat hierarkis dan hanya berbeda pada satu garis. Menurut ALL, perbedaan G^2 mereka dapat dianggap sebagai G^2 tersendiri. Karena itu, untuk mendapatkan signifikansi dari penggantian M^* dengan M^c (M^* vs. M^c), perbedaan G^2 mereka didapatkan dari pengurangan:

$$G^2(M^* \text{ vs. } M^c) = G^2(M^*) - G^2(M^c)$$

$$G^2(M^* \text{ vs. } M^c) = 2 \cdot N\left(\sum_{C \in C^*} H(C) - \sum_{S \in S^*} H(S) - \sum_{C \in C^c} H(C) + \sum_{S \in S^c} H(S) \right) \tag{3.8}$$

Jumlah komponen pada Persamaan 3.8 akan banyak tereduksi, karena banyaknya pembatalan. Untuk model kandidat pada *Gambar 3.1*b, perbedaan dengan M^* menjadi:

$$\begin{aligned} G^2(M^* \text{ vs. } M^c) = 2 \cdot N\big(&H(\{0,2\}) + H(\{1,2\}) + H(\{2,3\}) + H(\{3,4\}) \\ &- H(\{2\}) - H(\{3\}) \\ &- H(\{0,2,3\}) - H(\{1,2\}) - H(\{3,4\}) \\ &+ H(\{3\})\big) \end{aligned}$$

$$G^2(M^* \text{ vs. } M^c) = 2 \cdot N\big(H(\{0,2\}) + H(\{2,3\}) - H(\{0,2,3\}) - H(\{2\})\big) \tag{3.9}$$

Rumus perbedaan tersebut sesuai dengan teorema berikut, yang menunjukkan bahwa level signifikan suatu kandidat dapat diperoleh hanya dengan menghitung empat entropi:

Teorema 1 [18]: Jika terdapat dua model *decomposable* M^* dan M^c yang berbeda hanya pada satu garis (a, b), maka:

$$G^2(M^* \text{ vs. } M^c) = 2 \cdot N\big(H(S_{ab} \cup \{a\}) + H(S_{ab} \cup \{b\}) - H(S_{ab} \cup \{a,b\}) - H(S_{ab})\big)$$

Sebagai contoh, pada iterasi pertama Chordalysis pada data D, M^* tidak memiliki garis, dan akan terdapat kandidat M^c dengan satu garis (a, b) = (0, 4). *Separator* minimal untuk garis tersebut adalah $S_{ab} = \emptyset$. Menurut Teorema 1, statistik G^2 untuk kandidat ini adalah:

$$G^2 = 2 \cdot N\big(H(\{0\}) + H(\{4\}) - H(\{0,4\}) - H(\{\ \})\big)$$

$$G^2 = 2 \cdot 435(0{,}667 + 0{,}78 - 0{,}934 - 0)$$

$$G^2 = 446{,}2678 \tag{3.10}$$

Kemudian, dihitung skor untuk mengganti M^* dengan M^c ini. Skor tersebut adalah *p-value* yang diperoleh dari statistik G^2 dan derajat kebebasan df_r [19]:

$$df_r = par(S_{ab} \cup \{a,b\}) + par(S_{ab}) - par(S_{ab} \cup \{a\}) - par(S_{ab} \cup \{b\}) \tag{3.11}$$

dengan fungsi $par(A) = -1 + \prod_{v \in A} |Dom(v)|$. Untuk M^c dengan garis (0, 4), diperoleh $G^2 = 446{,}2678$ dan $df_r = 1$. Dengan demikian, skor untuk kandidat tersebut adalah 0.

3.5 Optimasi Komputasi Entropi Marginal

Untuk menghitung G^2 berdasarkan Teorema 1, dibutuhkan entropi $H(A)$. Beberapa entropi dibutuhkan di banyak perhitungan, sehingga Chordalysis menyimpan entropi-entropi yang telah diperoleh. Dengan demikian, jika suatu kandidat membutuhkan entropi yang telah dihitung sebelumnya, Chordalysis tidak perlu menghitungnya lagi.

Selain itu, seperti ditunjukkan pada *Tabel 3.1*, entropi untuk himpunan variabel A dirumuskan sebagai:

$$H(A) = -\sum_{i \in Dom(A)} p_i \ln p_i \tag{3.12}$$

yang dapat ditulis sebagai:

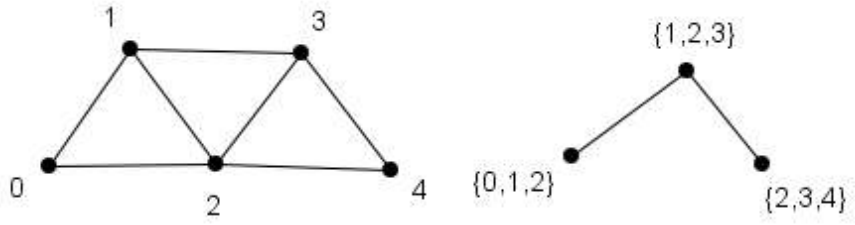
$$H(A) = -\frac{1}{N} \sum_{i \in Dom(A)} n_i \ln\left(\frac{n_i}{N}\right)$$

$$H(A) = -\frac{1}{N} \sum_{i \in Dom(A)} n_i (\ln n_i - \ln N) \tag{3.13}$$

Fungsi ln membutuhkan waktu dan ruang komputasi yang signifikan. Diketahui bahwa nilai n_i tidak akan melebihi ukuran data (pada D, $0 \leq n_i \leq 435$), sehingga sebelum iterasi pertama dimulai, terlebih dahulu dihitung semua entropi parsial:

$$\text{entropi\_parsial}(n) = n \cdot (\ln n - \ln N), 1 \leq n \leq N \tag{3.14}$$

dengan entropi_parsial(0) = 0. Dengan demikian, untuk memperoleh entropi selama iterasi berjalan, hanya dibutuhkan agregasi dari entropi parsial:



(a) (b)

Gambar 3.2 Suatu Graf *Chordal* (a) dan *Clique-graph* (b)

$$H(A) = -\frac{1}{N} \sum_{i \in Dom(A)} \text{entropi\_parsial}(n_i) \tag{3.15}$$

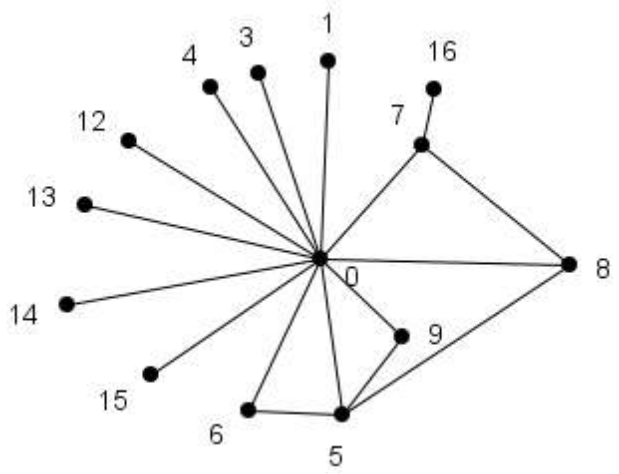
3.6 *Prioritized* Chordalysis

Pada Chordalysis, di setiap iterasi, dibangkitkan model-model *decomposable* yang berbeda pada satu garis saja dengan M^*. Dengan kata lain, satu garis ditambahkan pada M^*, yang membuat graf tetap *chordal*. Dalam mencari garis-garis tersebut, dihitung skor tiap garis. Banyaknya perhitungan skor tersebut dapat direduksi dengan mengetahui bahwa skor dari beberapa garis tidak berubah antar iterasi.

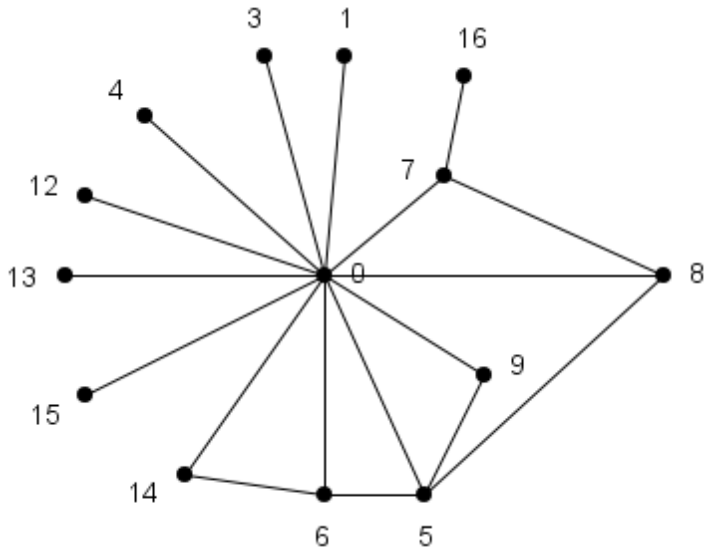
Untuk menghitung skor suatu garis (a, b), hanya dibutuhkan dua nilai: G^2 dan df_r. Berdasarkan Teorema 1 dan Persamaan 3.11, *Prioritized* Chordalysis mengetahui bahwa untuk (a, b) yang sama, skornya juga sama, kecuali *separator* minimal S_{ab} berbeda. Dengan demikian, pada suatu iterasi, suatu garis butuh dihitung kembali skornya apabila *separator* minimalnya berubah [19].

Kemudian, untuk menentukan skor yang harus dihitung ulang, seluruh garis ditinjau, kemudian dilihat apakah *separator* minimalnya berubah. Tetapi, untuk ribuan variabel, cara tersebut tidaklah efisien. Untuk mengatasinya, *Prioritized* Chordalysis menggunakan *clique-graph* yang berkaitan dengan graf *chordal*. Dari graf *chordal* G, *clique-graph* $C(G) = \{V_c, E_c\}$ didefinisikan sebagai [6]:

- V_c adalah himpunan *clique* maksimal dari G



Gambar 3.3 Graf *Chordal* untuk *M** Setelah Iterasi Ke-17



Gambar 3.4 Graf *Chordal* untuk *M** Setelah Penambahan Garis (6,14) pada Iterasi Ke-18

- (C_1, C_2) termasuk dalam E_c jika dan hanya jika $C_1 \cap C_2$ adalah *separator*-(a, b) minimal untuk tiap $a \in C_1 \backslash C_2$ dan tiap $b \in C_2 \backslash C_1$

Contoh suatu G dan $C(G)$-nya ditunjukkan pada Gambar 3.2. Untuk memperbarui $C(G)$ selama iterasi berjalan dan menemukan *separator* minimal secara efisien, digunakan fungsi kelayakan $\varepsilon_M^f(a,b)$, dengan $a < b$. Sebagai contoh,

Tabel 3.2 Nilai Fungsi Kelayakan dari D Setelah Iterasi ke-17

(a,b)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-															
2	{}	{}														
3	-	{0}	{}													
4	-	{0}	{}	{0}												
5	-	{0}	{}	{0}	{0}											
6	-	{0}	{}	{0}	{0}	-										
7	-	{0}	{}	{0}	{0}	{0,8}	-									
8	-	{0}	{}	{0}	{0}	-	{0,5}	-								
9	-	{0}	{}	{0}	{0}	-	{0,5}	-	{0,5}							
10	{}	{}	{}	{}	{}	{}	{}	{}	{}	{}						
11	{}	{}	{}	{}	{}	{}	{}	{}	{}	{}	{}					
12	-	{0}	{}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{}	{}				
13	-	{0}	{}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{}	{}	{0}			
14	-	{0}	{}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{}	{}	{0}	{0}		
15	-	{0}	{}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{}	{}	{0}	{0}	{0}	
16	{7}	-	{}	-	-	-	-	-		-	{}	{}	-	-	-	-

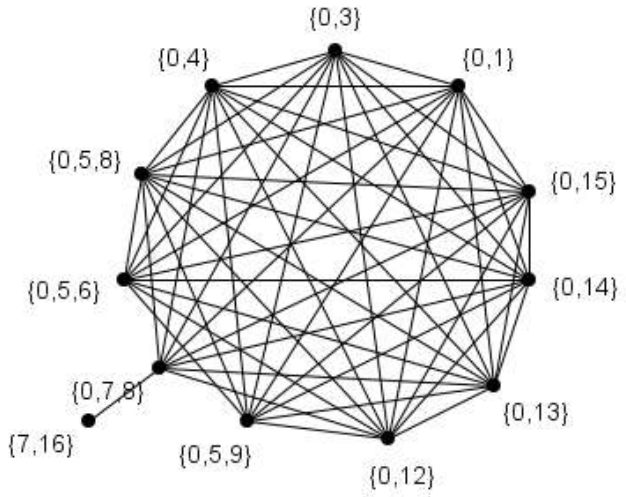
M^* dari data D pada iterasi ke-17 (Gambar 3.3) memiliki nilai $\varepsilon_M^f(a,b)$ yang ditunjukkan pada Tabel 3.2. Terdapat beberapa garis yang jika ditambahkan akan menyebabkan graf menjadi tidak *chordal*, seperti garis (4,16). Garis (8,9) dapat ditambahkan, dengan *separator* minimal $\{0, 5\}$.

Fungsi kelayakan diperbarui tiap iterasi. Pada G, penambahan satu garis (a, b) dengan *separator* minimal S_{ab} akan memicu dua tipe aksi:

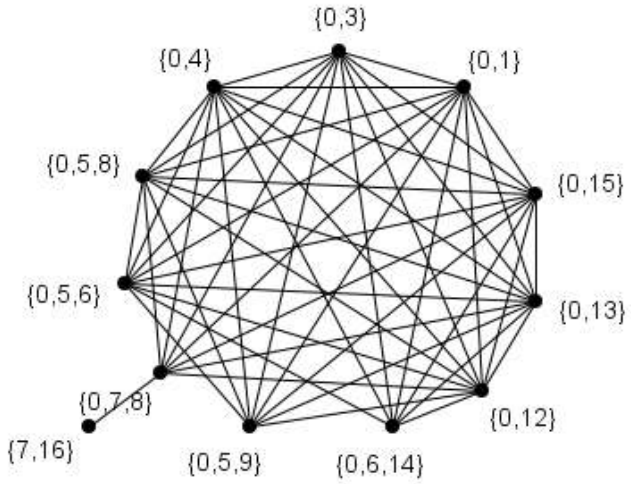
- Penambahan garis (C', C_{ab}) pada $C(G)$, dengan $C_{ab} = S_{ab} \cup \{a\} \cup \{b\}$. Untuk semua $x \in C' \backslash C_{ab}$, $a \in C_{ab} \backslash C'$ dan garis (x, a) belum ada pada G: fungsi $\varepsilon_M^f(x,a)$ diaktifkan dengan *separator* minimal $C' \cap C_{ab}$.
- Penghapusan garis (C_1, C_2) pada $C(G)$. Fungsi $\varepsilon_M^f(x,y)$ dinonaktifkan untuk semua (x, y), x (atau y) berada pada komponen terhubung yang sama dengan a (atau b) pada $G - S_{ab}$.

Sebagai penjelasan, akan diambil contoh proses pembaruan $\varepsilon_M^f(a,b)$ pada iterasi ke-19. Garis (6,14) dengan *separator* minimal $\{0\}$ akan ditambahkan ke G seperti ditunjukkan pada Gambar 3.4. Akibatnya, pada $C(G)$, titik $\{0,14\}$ diganti dengan titik $C_{ab} = \{0,6,14\}$, seperti ditunjukkan pada Gambar 3.5 dan Gambar 3.6. Penggantian tersebut memicu:

- Pada $G - S_{ab}$, titik-titik yang berada pada komponen terhubung yang sama



Gambar 3.5 $C(G)$ Setelah Iterasi Ke-17



Gambar 3.6 $C(G)$ Setelah Penambahan Garis (6,14) pada Iterasi Ke-18

dengan a adalah $x \in \{5,6,7,8,9,16\}$.

- Pada $G - S_{ab}$, titik-titik yang berada pada komponen terhubung yang sama dengan b adalah $y \in \{14\}$.
- Pada $C(G)$, garis-garis berikut dihapus: $(\{0,1\},\{0,14\})$, $(\{0,3\},\{0,14\})$, $(\{0,4\},\{0,14\})$, $(\{0,5,6\},\{0,14\})$, $(\{0,5,8\},\{0,14\})$, $(\{0,5,9\},\{0,14\})$, $(\{0,7,8\},\{0,14\})$, $(\{0,12\},\{0,14\})$, $(\{0,13\},\{0,14\})$, dan $(\{0,14\},\{0,15\})$.

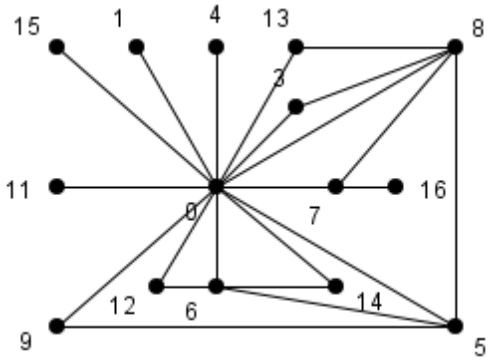
Akibatnya, $\varepsilon_M^f(5,14)$, $\varepsilon_M^f(6,14)$, $\varepsilon_M^f(7,14)$, $\varepsilon_M^f(8,14)$, $\varepsilon_M^f(9,14)$, dan $\varepsilon_M^f(14,16)$ dinonaktifkan.

- Pada $C(G)$, ditambahkan $(\{0,1\},\{0,6,14\})$. $\varepsilon_M^f(1,14)$ diaktifkan dengan *separator* minimal $\{0,1\} \cap \{0,6,14\} = \{0\}$. $\varepsilon_M^f(6,14)$ tidak diaktifkan karena garis (6,14) sudah terpilih dari iterasi sebelumnya.
- Pada $C(G)$, ditambahkan $(\{0,3\},\{0,6,14\})$. $\varepsilon_M^f(3,6)$ dan $\varepsilon_M^f(3,14)$ diaktifkan dengan *separator* minimal $\{0,3\} \cap \{0,6,14\} = \{0\}$.
- Pada $C(G)$, ditambahkan $(\{0,4\},\{0,6,14\})$. $\varepsilon_M^f(4,6)$ dan $\varepsilon_M^f(4,14)$ diaktifkan dengan *separator* minimal $\{0,4\} \cap \{0,6,14\} = \{0\}$.
- Pada $C(G)$, ditambahkan $(\{0,5,6\},\{0,6,14\})$. $\varepsilon_M^f(5,14)$ diaktifkan dengan *separator* minimal $\{0,5,6\} \cap \{0,6,14\} = \{0,6\}$.
- Pada $C(G)$, ditambahkan $(\{0,12\},\{0,6,14\})$. $\varepsilon_M^f(12,6)$ dan $\varepsilon_M^f(12,14)$ diaktifkan dengan *separator* minimal $\{0,12\} \cap \{0,6,14\} = \{0\}$.
- Pada $C(G)$, ditambahkan $(\{0,13\},\{0,6,14\})$. $\varepsilon_M^f(13,6)$ dan $\varepsilon_M^f(13,14)$ diaktifkan dengan *separator* minimal $\{0,13\} \cap \{0,6,14\} = \{0\}$.
- Pada $C(G)$, ditambahkan $(\{0,15\},\{0,6,14\})$. $\varepsilon_M^f(15,6)$ dan $\varepsilon_M^f(15,14)$ diaktifkan dengan *separator* minimal $\{0,15\} \cap \{0,6,14\} = \{0\}$.

Dengan beberapa pembaruan tersebut, didapatkan informasi mengenai garis-garis yang mengalami perubahan *separator* minimal, yang selanjutnya butuh dihitung ulang.

Selama iterasi berjalan, *Prioritized* Chordalysis membangun suatu *priority queue q* untuk mengurutkan garis-garis berdasarkan skornya. Pada iterasi pertama, semua garis ditambahkan ke dalam q. Kemudian pada iterasi-iterasi berikutnya, pembaruan pada $\varepsilon_M^f(a,b)$ mengakibatkan garis (a, b):

- ditambahkan pada q, karena garis tersebut memenuhi syarat (membuat graf tetap *chordal*),
- berpindah posisi di q, karena *separator* minimalnya berubah (sehingga skornya dihitung ulang), atau



Gambar 3.7 Model Akhir untuk Data *D*, Diperoleh Setelah 22 Iterasi

- dihapus dari *q*, karena garis tersebut tidak lagi memenuhi syarat.

3.7 Pemilihan Kandidat

Setelah perhitungan skor, Chordalysis memilih M^c terbaik. Kandidat terbaik tersebut adalah kandidat dengan *p-value* terkecil, yang dapat diketahui dari urutan garis di *q*. Skor tersebut kemudian dibandingkan dengan suatu nilai ambang. Jika skor terkecil lebih rendah dari nilai ambang, maka *M** diganti dengan M^c, lalu dijalankan iterasi berikutnya. Jika tidak, maka *M** adalah model akhir, yang tidak lagi membutuhkan penambahan asosiasi.

3.7.1 Pembaruan Nilai Ambang

Pada tiap iterasi, skor dari kandidat terbaik akan dibandingkan dengan nilai ambang α untuk menentukan apakah *M** akan diganti dengan kandidat tersebut, atau *M** menjadi model akhir dan iterasi berhenti. Chordalysis menjalankan banyak tes statistik, yang bisa menghasilkan banyak *false positive*. Akibatnya, kandidat yang tidak signifikan bisa dianggap signifikan. Untuk mengatasinya, nilai α diperbarui tiap iterasi sehingga Chordalysis tidak terlalu sering mengganti *M**. Aturan pembaruan ini mengikuti *layered critical value* [25]. Pada iterasi *i* dengan *M** memiliki garis sebanyak *L*, nilai ambang α_i adalah:

$$\alpha_i = \frac{\alpha}{2^L \cdot S_i} \tag{3.16}$$

dengan S_i adalah ruang pencarian, yaitu jumlah graf *chordal* yang bisa dibentuk dari

penambahan satu garis pada M^*. Nilai α pada Persamaan 3.16 adalah ambang *p-value*, yang pada umumnya bernilai 0,05.

Sebagai contoh, dalam iterasi ke-18 pada data D, M^* memiliki 17 garis (Gambar 3.3). Karena terdapat 107 kandidat yang memenuhi syarat untuk M^* ini, maka nilai ambang diperbarui menjadi:

$$\alpha_{18} = \frac{0{,}05}{2^{17} \cdot 107} \approx 3{,}565 \times 10^{-9} \tag{3.17}$$

Selain itu, M^c terbaik pada iterasi ini adalah kandidat dengan garis (6,14), ditunjukkan pada Gambar 3.4. Kandidat tersebut memiliki $G^2 \approx 87{,}897$ dan $df_r = 8$, sehingga skornya adalah $1{,}221 \times 10^{-15}$. Skor tersebut lebih kecil dari nilai ambang, sehingga M^* diganti dengan kandidat ini.

Pada iterasi ke-23, model M^* ditunjukkan pada Gambar 3.7. Terdapat 87 kandidat, sehingga nilai ambang $\alpha_{23} \approx 1{,}37 \times 10^{-10}$. Diantara kandidat-kandidat tersebut, skor terbaik adalah $8{,}736 \times 10^{-10}$, dengan menambahkan garis (4,5). Karena skor ini lebih besar dari nilai ambang, maka M^* tidak diganti, dan menjadi model akhir.

Dalam model akhir yang ditunjukkan pada Gambar 3.7, terdapat banyak titik yang terhubung dengan titik 0. Hal ini tidak mengherankan, karena V_0 menunjukkan partai politik, dan pilihan dari tiap anggota bergantung pada partai politik mereka.

(*halaman ini sengaja dikosongkan*)

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini, akan dijelaskan mengenai aplikasi Chordalysis pada proses penemuan antibiotik. Fokus aplikasi ini adalah reduksi dimensi, dengan memilih beberapa fitur dari ribuan. Chordalysis akan menemukan beberapa asosiasi antar variabel, dan dari asosiasi tersebut akan dihapus beberapa variabel yang bisa diwakilkan oleh variabel lain. Penelitian ini didasarkan pada penelitian sebelumnya yang telah dilakukan pada tim yang sama.

4.1 Penelitian Sebelumnya

Tujuan dari penelitian sebelumnya adalah menguji hasil dari enam algoritma dalam mengklasifikasikan molekul antibiotik dan non-antibiotik [9]. Keenam algoritma tersebut adalah: *Support Vector Machine* (SVM) dengan kernel linear, *random forest*, regresi logistik, *gradient boosted trees*, *naïve* Bayes, dan pohon keputusan.

Data mengenai molekul-molekul antibiotik dan non-antibiotik diperoleh dari antibiotik yang ada di pasaran, dan juga dari MDDR dan Life Chemical Inc. Dari pasar, didapatkan 150 antibiotik. Dari MDDR, didapatkan 2854 antibiotik dan 57179 non-antibiotik. Dari Life Chemical Inc, diperoleh 38907 antibiotik dan 52604 non-antibiotik. Perangkat lunak Dragon [24] digunakan untuk mendefinisikan 4885 atribut tiap molekul. Nilai tiap atribut dihitung menggunakan Corina [22].

Tabel 4.1 Filter Atribut

Filter	Standar deviasi lebih besar dari	*Pair correlation* lebih kecil dari	Jumlah atribut terpilih
1	0,010	0,4	87
2	0,010	0,8	576
3	0,001	0,8	588
4	0,100	0,8	524

Tabel 4.2 Rekapitulasi Hasil Chordalysis pada Tiga Data Uji

Data Uji	Jumlah Atribut	Representasi Grafis dari Hasil		
		Jumlah Garis	Jumlah Komponen Terkoneksi	Jumlah Atribut yang Tidak Ditampilkan
D2	576	88	21	467
D3	588	89	21	478
D4	524	85	20	419

Pertama-tama, jumlah atribut direduksi melalui beberapa prosedur. Atribut yang memiliki *missing value* dihapus. Selain itu, jika terdapat grup atribut yang berkorelasi sempurna, hanya satu dari mereka yang diambil. Kedua prosedur penghapusan atribut ini menghasilkan 4532 atribut.

Dari data yang sudah tereduksi ini, empat filter digunakan secara independen untuk mendapatkan jumlah atribut yang lebih sedikit, dengan memperhatikan standar deviasi dan *pair correlation*. Parameter dan hasil dari filter-filter tersebut (disebut Filter 1, Filter 2, Filter 3, dan Filter 4) ditunjukkan pada Tabel 4.1. Dengan demikian, terdapat 5 data set: satu data yang tidak disaring dengan 4532 atribut, dan empat data tersaring.

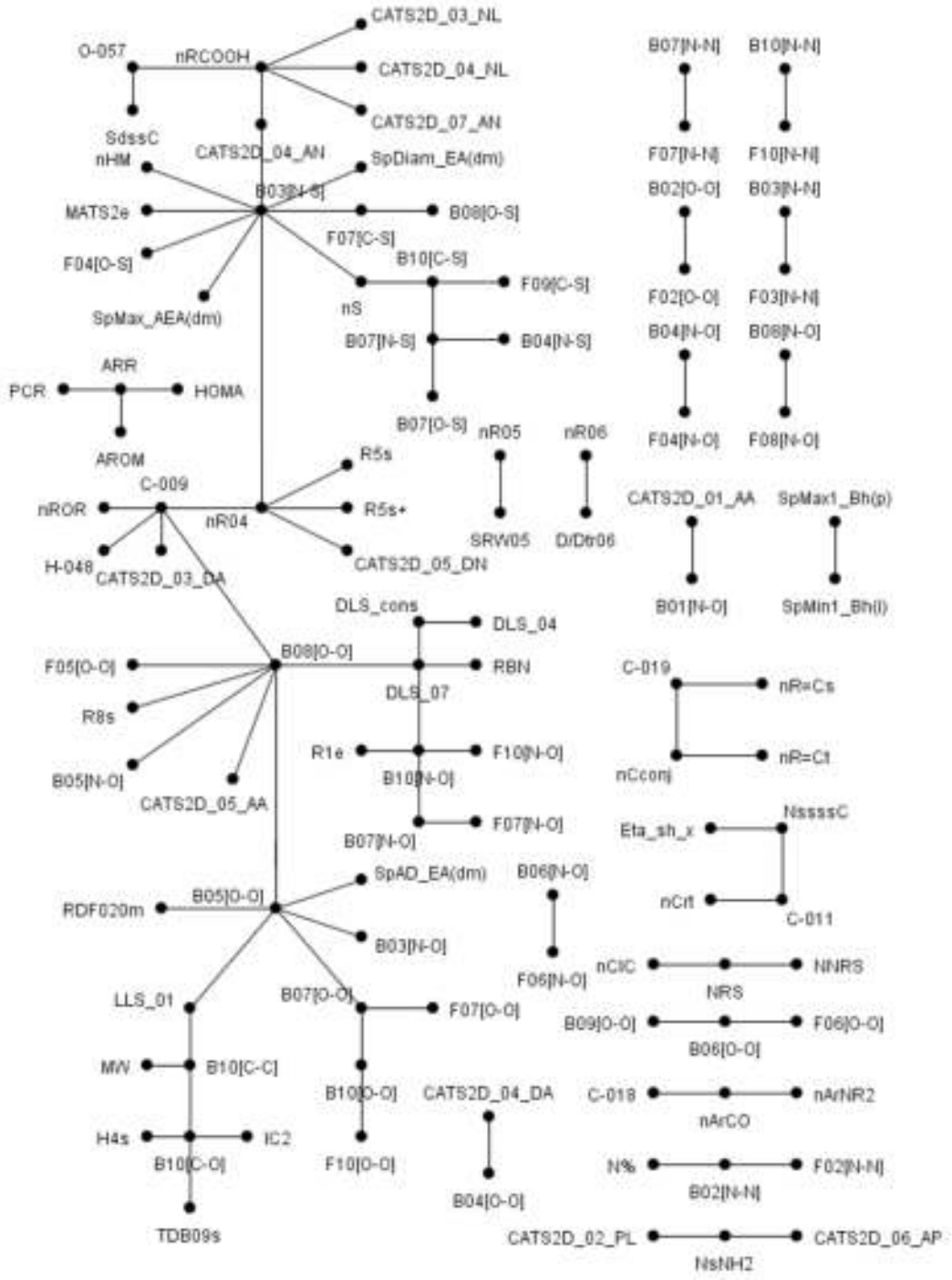
Penelitian tersebut menunjukkan bahwa SVM memiliki presisi terbaik, tetapi *recall*-nya sangat rendah. *Recall* dari *random forest* jauh lebih baik dari SVM, sehingga *random forest* dapat digunakan untuk klasifikasi molekul, dengan mengorbankan presisi SVM.

Pada subbab-subbab berikutnya, akan dijelaskan penelitian kami dalam mengaplikasikan Chordalysis untuk menyeleksi fitur diantara ratusan atribut.

4.2 Data Uji

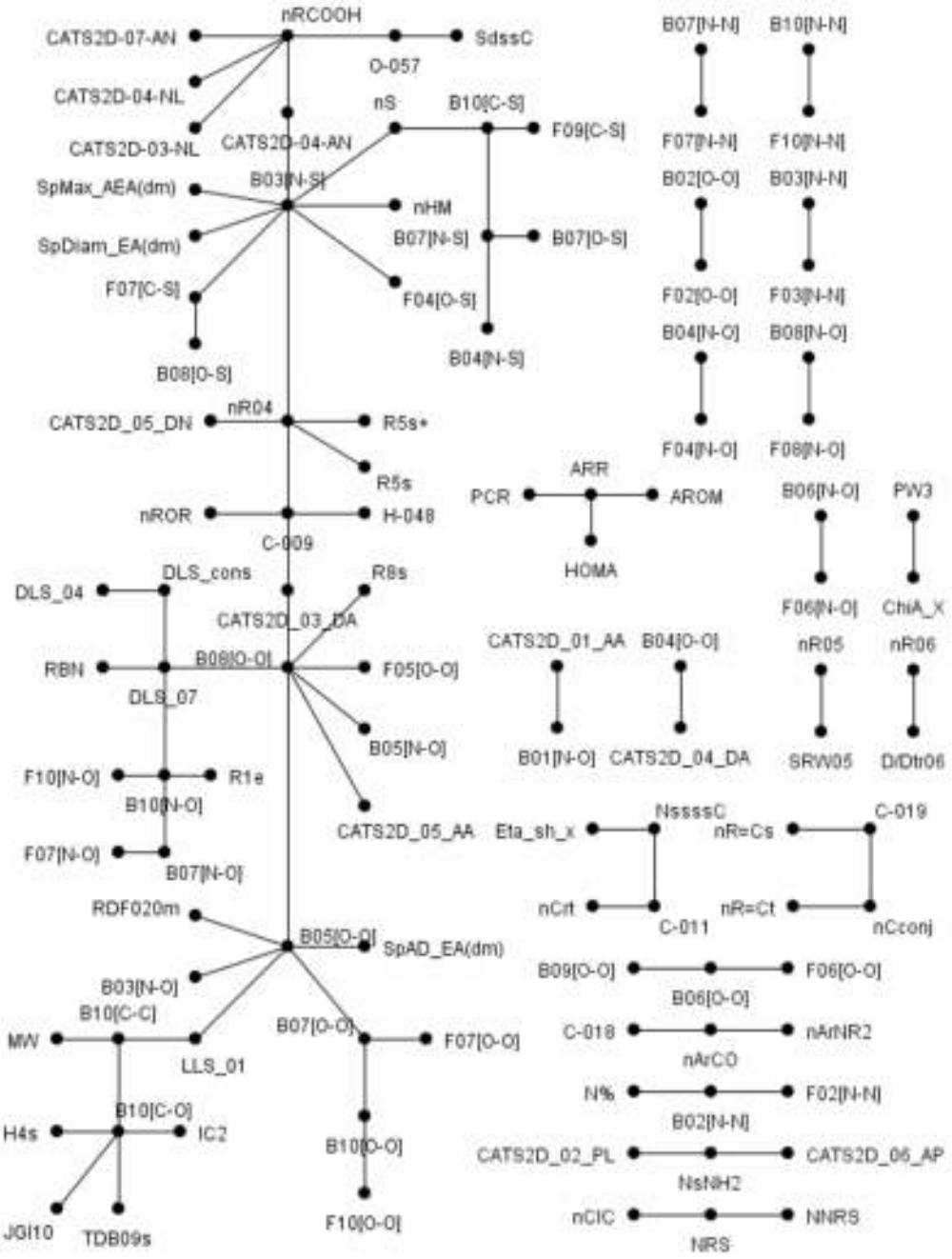
Penelitian ini berfokus pada 150 molekul antibiotik yang ada di pasaran. Dari molekul-molekul tersebut, digunakan hasil dari Filter 2, 3, dan 4 dari penelitian sebelumnya, karena hasil filter tersebut memiliki jumlah atribut yang besar, lebih dari 500 (ditunjukkan pada Tabel 4.1). Oleh karena itu, pada penelitian ini, terdapat 3 data uji:

- D2, dari Filter 2, dengan 576 atribut,



Gambar 4.1 Graf Hasil Chordalysis pada D2

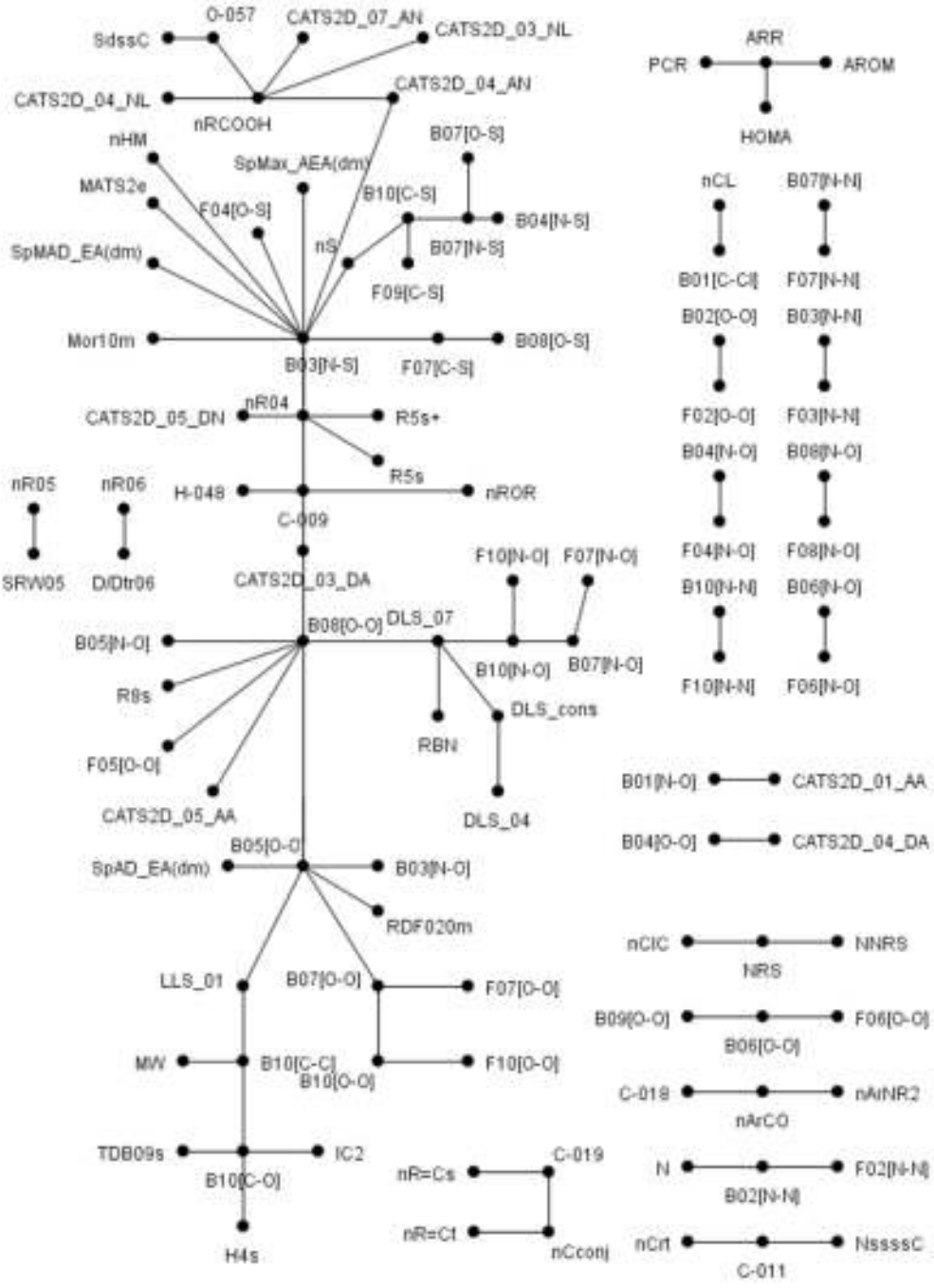
- D3, dari Filter 3, dengan 588 atribut, dan
- D4, dari Filter 4, dengan 524 atribut.



Gambar 4.2 Graf Hasil Chordalysis pada D3

4.3 Diskritisasi

Beberapa atribut bersifat numerik. Karena ALL dan Chordalysis bekerja pada variabel diskrit, maka atribut-atribut numerik tersebut harus melalui praproses setelah proses filter sehingga mereka menjadi diskrit. Praproses ini dijalankan pada



Gambar 4.3 Graf Hasil Chordalysis pada D4

atribut-atribut yang memiliki lebih dari 10 macam nilai. Kemudian, digunakan *equal-width discretization* dari Weka, dengan 10 *bin* sebagai hasil diskritisasi.

Tabel 4.3 Hasil Seleksi Fitur pada Tiga Data Uji

Data Uji	Jumlah Atribut	Jumlah Atribut Terpilih		
		Independen	Komponen Terkoneksi	Total
D2	576	467	43	510
D3	588	478	44	522
D4	524	419	42	461

4.4 Hasil dari Chordalysis

Chordalysis diaplikasikan pada ketiga data uji. Hasil grafis dari data-data tersebut ditunjukkan pada Gambar 4.1, Gambar 4.2, dan Gambar 4.3, serta direkapitulasi pada Tabel 4.2. Jumlah garis menunjukkan jumlah asosiasi yang ditemukan.

Terdapat atribut-atribut yang tidak muncul dalam graf. Sebagai contoh, untuk D2, dari 576 atribut, graf yang dihasilkan memiliki 109 atribut. Hal ini berarti bahwa 467 atribut lainnya tidak memiliki asosiasi.

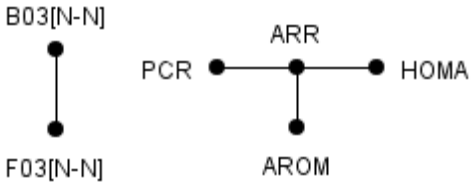
Komponen-komponen terkoneksi memiliki jumlah titik yang berbeda-beda. Komponen terkoneksi terbesar dari D2, D3, dan D4 secara berturut-turut memiliki 58, 59, dan 58 titik, sementara komponen-komponen terkoneksi lainnya memiliki paling banyak 4 titik.

4.5 Seleksi Fitur

Setelah memperoleh tiga graf yang menggambarkan asosiasi antar atribut, seleksi fitur dijalankan dengan melihat graf-graf tersebut. Hasil dari seleksi ini ditunjukkan pada Tabel 4.3.

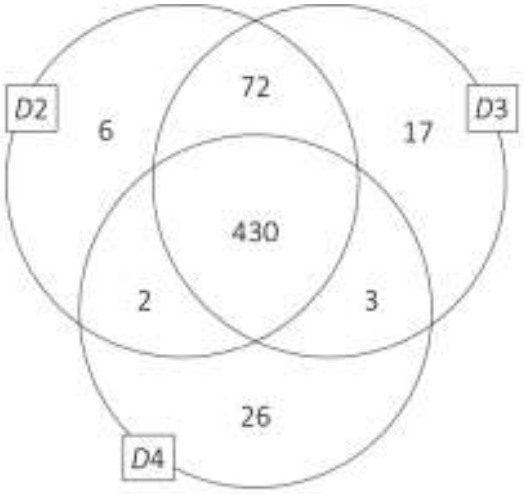
Atribut-atribut yang independen (tidak memiliki asosiasi) diambil, karena mereka tidak bisa diwakilkan oleh atribut lain. Jumlah atribut-atribut ini ditunjukkan pada kolom “independen” pada Tabel 4.3.

Seleksi fitur dijalankan pada tiap komponen terkoneksi. Atribut-atribut yang dihapus adalah atribut yang memiliki hanya satu asosiasi. Dengan demikian, jika ditemui komponen terkoneksi seperti Gambar 4.4b, atribut yang terpilih adalah ARR. Hal ini berarti bahwa ARR dapat merepresentasikan atribut PCR, HOMA,



(a) (b)

Gambar 4.4 Contoh Komponen Terkoneksi dengan 2 Titik (a) dan Lebih dari 2 Titik (b)



Gambar 4.5 Diagram Venn Atribut-atribut yang Terpilih dari D2, D3, dan D4

dan AROM. Di sisi lain, jika suatu komponen terkoneksi hanya memiliki 2 titik, seperti Gambar 4.4a, satu atribut dipilih dan lainnya dihapus. Jumlah atribut yang terpilih dari proses ini ditunjukkan pada kolom “komponen terkoneksi” pada Tabel 4.3.

Di antara hasil-hasil seleksi fitur dari D2, D3, dan D4 (Gambar 4.5), terdapat 430 atribut yang terpilih dari ketiga data uji.

Selain itu, ditunjukkan bahwa proses seleksi fitur pada penelitian sebelumnya dapat ditingkatkan. Pada tahap awal dari penelitian tersebut, atribut-atribut yang berkorelasi sempurna dihapus. Setelah mengaplikasikan Chordalysis, diketahui bahwa masih ada beberapa asosiasi (lebih dari 80). Oleh sebab itu, dari D2, D3, dan D4, secara berturut-turut dihapus 66, 66, dan 63 atribut.

(*halaman ini sengaja dikosongkan*)

BAB 5
KESIMPULAN

5.1 Kesimpulan

Pada penelitian ini, ditunjukkan bahwa hasil dari Chordalysis dapat digunakan untuk seleksi fitur. Gagasan awalnya adalah bahwa seleksi dapat dijalankan dengan menghilangkan atribut yang bisa diwakilkan oleh atribut lain. Dengan demikian, dibutuhkan suatu metode yang dapat menemukan asosiasi antar atribut. Chordalysis dapat melakukan hal tersebut dalam waktu yang layak, dan dari hasil Chordalysis, dapat ditentukan atribut-atribut yang bisa dieliminasi.

5.2 Saran

Penelitian ini berfokus pada molekul-molekul antibiotik yang terdapat di pasaran. Untuk penelitian selanjutnya, dapat dilakukan seleksi fitur berbasis Chordalysis untuk data uji yang lebih besar, yaitu molekul-molekul antibiotik dan non-antibiotik, termasuk dari MDDR dan Life Chemical Inc.

Chordalysis bekerja pada variabel diskrit dan mengeliminasi variabel-variabel yang memiliki *missing values*. Untuk itu, diperlukan suatu metode yang tepat untuk proses diskritisasi. Selain itu, dibutuhkan juga pengolahan *missing values* agar variabel yang akan dipertimbangkan menjadi lebih beragam.

(*halaman ini sengaja dikosongkan*)

DAFTAR PUSTAKA

[1] Agrawal, R., Imielinski, T., dan Swami, A. (1993), "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Eds: Buneman, P., Jajodia, S., University of Pennsylvania, Washington D.C., hal. 207-216.

[2] Berry, A. dan Pogorelnick, R. (2011), "A simple algorithm to generate the minimal separators and the maximal cliques of a chordal graph", *Information Processing Letters*, Vol. 111, No. 11, hal. 508-511.

[3] Durrant, J.D. dan Amaro, R.E. (2015), "Machine-learning techniques applied to antibacterial drug discovery", *Chemical Biology & Drug Design*, Vol. 85, hal. 14-21.

[4] Fayyad, U., Piatetsky-Shapiro, G., dan Smyth, P. (1996), "From Data Mining to Knowledge Discovery in Databases", *AI Magazine*, Vol. 17, No. 3, hal. 37-54.

[5] Fernandes, P. (2015), "The global challenge of new classes of antibacterial agents: an industry perspective", *Current Opinion in Pharmacology*, Vol. 24, hal. 7-11.

[6] Galinier, P., Habib, M., dan Paul, C. (1995), "Chordal graphs and their clique graphs", *Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science*, hal. 358-371.

[7] Haberman, S.J. (1977), *The analysis of frequency data*, Vol. 4, University of Chicago Press, Chicago.

[8] Hinton, G. et al. (2012), "Deep Neural Networks for Acoustic Modeling in Speech Recognition", *Signal Processing Magazine*, Vol. 29, hal. 82-97.

[9] Hung, J. (2015), *An experiment about the classification of antibacterial molecules*, Internal Technical Report, University of Lorraine, Nancy.

[10] John, G.H., Kohavi, R., dan Pfleger, K. (1994), "Irrelevant Features and the Subset Selection Problem", *Machine Learning: Proceedings of the Eleventh International Conference*, Eds: Kaufmann, M., Rutgers University, New

Brunswick, hal. 121-129.

[11] Koller, D. dan Friedman, N. (2009), *Probabilistic Graphical Models*, The MIT Press, Cambridge.

[12] Krizhevsky, A., Sutskever, I., dan Hinton, G.E. (2012), "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 25, hal. 1090-1098.

[13] Lauritzen, S. (2011), Decomposition and decomposable models, University of Oxford, Oxford.

[14] LeCun, Y., Bengio, Y., dan Hinton, G. (2015), "Deep Learning", *Nature*, Vol. 521, hal. 436-444.

[15] Li, F. dan Yang, Y. (2005), "Using recursive classification to discover predictive features", *Proceedings of the 2005 ACM symposium on Applied computing*, Ed: Liebrock, L.M., New Mexico Institute of Mining and Technology, Santa Fe, hal. 1054-1058.

[16] Ma, J. et al. (2015), "Deep Neural Nets as a Method for Quantitative Structure Activity Relationships", *J. Chem. Inf. Model.*, Vol. 55, hal. 263-274.

[17] Malvestuto, F. (1991), "Approximating discrete probability distributions with decomposable models", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 5, hal. 1287-1294.

[18] Petitjean, F., Webb, G.I., dan Nicholson, A.E. (2013), "Scaling log-linear analysis to high-dimensional data", *IEEE 13th International Conference on Data Mining*, Eds: Xiong, H. et al., IEEE Computer Society, Dallas, hal. 597-606.

[19] Petitjean, F. dan Webb, G.I. (2015), "Scaling log-linear analysis to datasets with thousands of variables", *Proceedings of the 2015 SIAM International Conference on Data Mining*, Eds: Venkatasubramanian, S. dan Ye, J., Society of Industrial and Applied Mathematics, Vancouver, hal. 469-477.

[20] Poon, H. dan Domingos, P. (2011), "Sum-Product Networks: A New Deep Architecture", *Uncertainty in Artificial Intelligence*, Vol. 27.

[21] Rau, A. et al. (2015), "Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models",

Bioinformatics, Vol. 31, hal. 1420-1427.

[22] Sadowski, J., Gasteiger, J., dan Klebe, G. (1994), "Comparison of Automatic Three-Dimensional Model Builders Using 639 X-ray Structures", *J. Chem. Inf. Comput. Sci.*, Vol. 34, hal. 1000-1008.

[23] Tabachnik, B.G dan Fidell, L.S. (2007), *Using Multivariate Statistics*, 5th edition, Pearson Education, Inc., Upper Saddle River.

[24] Todeschini, R. dan Consonni, V. (2009), *Molecular Description for Chemoinformatics*, 2nd edition, Wiley-VCH, Weinheim.

[25] Webb, G.I. (2008), "Layered critical values: A powerful direct-adjustment approach to discovering significant patterns", *Machine Learning*, Vol. 71, hal. 307-323.

[26] Xiong, H.Y. et al. (2015), "The human splicing code reveals new insights into the genetic determinants of disease", *Science*, Vol. 347, Issue 6128.

(*halaman ini sengaja dikosongkan*)

BIOGRAFI PENULIS

Nyoman Juniarta lahir di Surabaya pada tanggal 19 Juni 1991 sebagai anak ketiga dari tiga bersaudara. Penulis menempuh pendidikan formal di SDN Rungkut Menanggal I Surabaya pada tahun 1997-2003, kemudian di SMP Negeri 12 Surabaya pada tahun 2003-2006, lalu di SMA Negeri 16 Surabaya pada tahun 2006-2009. Pada tahun 2009-2014, penulis melaksanakan studi tingkat sarjana di Institut Teknologi Sepuluh Nopember (ITS) Jurusan Teknik Informatika, dengan mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV).

Pada tahun 2014, penulis melanjutkan studi program magister di ITS pada program studi yang sama. Kemudian untuk tahun ajaran 2015-2016, penulis mengikuti program *joint degree* dengan Universitas Lorraine di kota Metz, Prancis.

Penulis dapat dihubungi melalui surat elektronik nyoman.juniarta@yahoo.com.