Analisis dan Implementasi Algoritma gSpan dan LPBoost pada Klasifikasi Struktur Molekul Kimia

Analysis and Implementation Algorithm gSpan and LPBoost on Classification Structure Chemical Compound

Ahmad Ridwan Rezani[1], Kemas Rahmat Saleh W, S.T., M.Eng.[2], Nungki Selviandro S.Kom., M.Kom.[3]
[1, 2, 3] Program Studi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
[1]ridwanrezani@gmail.com, [2]bagindokemas@telkomuniversity.ac.id, [3]nselviandro@telkomuniversity.ac.id

Abstrak - Perkembangan teknologi dan informasi tentang pemodelan data elektronik yang semakin meningkat seperti xml text, html,graph, senyawa kimia dan lain lain menyebabkan jumlah aplikasi untuk memodelkan data tersebut semakin pesat. Salah satu yang popular dan banyak dikembangkan yaitu Graph. Graph sangat powerful karena bisa memodelkan struktur yang kompleks. Salah satu penerapannya yaitu risk assessment, toxic prediction dan regulatory decision. Studi mengenai graph berbasis classification masih kurang dan untuk penerapannya masih jarang sehingga perlu penelitian lebih lanjut guna mendapatkan pemodelan data yang baik. Berbagai penelitian telah dilakukan dengan menggunakan teknik dalam klasifikasi graph salah satunya Graph Classification yang bisa diterapkan untuk chemical compound.

Dalam penelitian Tugas Akhir ini akan membahas tentang metode Graph Classification dengan menggunakan algoritma gSpan dan Boosting dalam melakukan klasifikasi molekul kimia dan menghitung akurasi yang diperoleh. Tujuannya untuk menentukan dan mengidentifikasi apakah suatu molekul kimia mengandung mutagen atau tidak berdasarkan model klasifikasi yang dibuat. Model klasifikasi ini akan membuat prediction rule dengan beberapa iterasi untuk mendapatkan pola. Pola ini didapatkan dengan cara mengenumerasi secara frequent kemunculan pola subgraph yang bisa digunakan sebagai feature dalam klasifikasi. Pemilihan teknik yang tepat dan rancangan sistem yang benar akan menghasilkan performansi sistem yang maksimal. molekul kimia dipilih karena keunggulan dan keunikannya yaitu memiliki ciri vertex berlabel dan edge yang tidak berarah sehingga molekul kimia cocok jika direpresentasikan dengan graph. Metode Graph Classification akan mengklasifikasi graph yang mempunyai karakteristik struktural information serta menggunakan semua subgrah yang terseleksi sebagai set fitur. Hasil dari penelitian ini menunjukan efisiensi dari algoritma gSpan dan Boosting untuk molekul kimia dengan akurasi tertinggi yaitu 78,18 %.

Kata kunci: *Graph Classification*, *Frequent Subgraph Mining*, Klasifikasi, *Cheminformatics*

Abstract – *The developments of information technology and data electronic modelling quite increasing such as text xml, html, graph , chemical compound and others led to the number of applications to model data as a graph growing rapidly. One of the popular and widely developed is Graph. Graph is very poverful because it can modeled complex structure. The example of application graph are risk assessment, toxic prediction and regulatory decision. Studies on graph-based classification are still lacking and for its application is still rare used that more needs to be done using the technique of classification. Graph classification are the one that can be applied to chemical compound.*

At this final project study will discussed about method Graph Classification with algorithm gSpan and boosting for the chemical molecular classification and counting accuracy are obtained. The goal is to determine and identify whether a chemical compound containing mutagen or not based on classification models are made. This classification model will make a prediction rule with a fewer iterations to get the patterns. This pattern is obtained b enumerating the frequent occurance of subgraph patterns which can be used as a feature in classification. the selection of feature extraction techniques appropriate for build model classification and the design of the right system in order to generate maximum system performance. Chemical molecules chosen for their excelent and uniqueness that has the characteristics of labeled verticle and edges are not directed so that the chemical molecules suitable if represent by a graph. Graph classification method will classify graphs which has the characteristics of structural information as well as using all subgraph are selected as feature set. Results from this study show the efficiency from algoritma gSpan and Boosting for maximum accuracy chemical compound is 78,18 %.

Keywords: *Graph Classification, Frequent Subgraph Mining*, Classification, *Cheminformatics*

I. PENDAHULUAN

Beberapa tahun terakhir peningkatan aplikasi pemodelan data menggunakan graph dalam berbagai bidang semakin meningkat karena graph bisa merepresentasikan model struktur yang kompleks. sebuah graph model secara general dapat diimplementasikan dalam berbagai bidang seperti cheminformatics, information management system, bioinformatics, computer network dan lain lain. untuk pemodelan struktur data banyak menggunakan representasi dengan vector numeric walaupun dalam keadaan realnya data tidak hanya di representasikan dengan vector numeric saja tetapi bisa direpresentasikan dengan bentuk graph. Struktur graph data bisa di konversi ke berbagai macam object akan tetapi semakin banyaknya data yang kompleks penggunaan basis data relasional sudah tidak efisien lagi jika data tersebut memiliki struktur yang rumit misalnya molekul kimia. molekul kimia memiliki struktur kompleks dan untuk melakukan klasifikasi terlalu rumit karena pattern dan polanya beragam. Sehingga jika menggunakan data relasional tidak cocok dalam untuk kasus molekul kimia yang mempunyai ciri vertex berlabel dan edge tidak berarah. Salah satu alternatifnya bisa menggunakan graph database dalam menentukan. graph database adalah sebuah sistem manajemen database online dengan menggunakan metode Create, Read, Update dan Delete(CRUD) yang menunjukan sebuah model data graph [1].

Graph di deskripsikan dengan bentuk node dan edge. Node digambarkan sebah titik atau obect tertentu sedangkan edge di gambarkan sebagai garis penghubung antar titik titik tersebut. Implementasi graph pada bidang kimia melahirkan pendekatan baru yaitu QSAR analisis. Quantitative Structure-Activity Relationship (QSAR) analisis menggunakan model classification dan regression untuk memanipulasi dan memprediksi data kimia untuk dibuat model structur entity relationship. QSAR sudah diterapkan di berbagain bidang contohnya risk assessment, toxic prediction dan regulatory decision [2].

Salah satu permasalahan dalam graph data processing adalah klasifikasi. Klasifikasi menjadi topik penting karena berpengaruh dalam pengambilan keputusan. Secara umum klasifikasi melakukan pembentukan model dari data training. Sehingga fokusnya adalah bagaimana mempelajari data dan membentuk model yang dapat mengenali karakteristik data. Hubungan antara graph dan klasifikasi menjadi topic baru yaitu graph classification. Graph classification bisa dijelaskan dengan dua pendekatan pertama membuat model untuk memprediksi label class dari keseluruhan graph yang kedua memprediksi label class node dalam graph atau label propagation. Untuk melakukan graph classification bisa menggunakan dua metode yaitu graph kernel dan graph boosting (frequent subgraph mining).

Kernel methods dalam graph kernel membangun prediksi rule berdasarkan kesamaan fungsi antara dua object label graph. metode ini berdasarkan random walk setiap graph dienumerasi pathnya dan menentukan probabilitas antara dua graph. Sedangkan graph boosting membangun prediksi rule berdasarkan perulangan. setiap perulangan hanya mempunyai sedikit subgraph informative yang ditemukan. Secara umum graph boosting merupakan gabungan antara graph mining dan mesin learning. Dalam beberapa literature algoritma untuk mengenumerasi graph mining pada frequent subgraph mining bisa menggunakan AGM,Gaston, Mofa dan Gspan. Dari beberapa algoritma enumerasi tersebut Gspan lebih cepat proses enumerasi frequent subgraph pada data graph yang besar [3].

Pada penelitian ini, penulis melakukan percobaan metode graph classification untuk mengidentifikasi dan klasifikasi molekul kimia apakah suatu molekul kimia mengandung mutagen atau tidak berdasarkan model klasifikasi yang dibuat. Model klasifikasi ini akan membuat prediction rule dengan beberapa iterasi untuk mendapatkan pola. Pola ini didapatkan dengan cara mengenumerasi secara frequent kemunculan pola subgraph yang bisa digunakan sebagai feature dalam klasifikasi. meskipun size kandidat fitur menjadi besar tapi akan secara otomatis menseleksi fitur set yang compact dan relevan berdasarkan algoritma boosting yang digunakan.

II. LANDASAN TEORI

A. Graph Boosting

Untuk melakukan graph classification bisa menggunakan dua metode yaitu graph kernel dan graph boosting (frequent subgraph mining). Kernel methods dalam graph kernel membangun prediksi rule berdasarkan kesamaan fungsi antara dua object label graph. metode ini berdasarkan random walk setiap graph dienumerasi pathnya dan menentukan probabilitas antara dua graph. Sedangkan graph boosting membangun prediksi rule berdasarkan perulangan. setiap perulangan hanya mempunyai sedikit subgraph informative yang ditemukan Secara umum graph boosting merupakan gabungan antara graph mining dan mesin learning. Dalam beberapa literature algoritma untuk mengenumerasi graph mining pada frequent subgraph mining bisa menggunakan AGM,Gaston, Mofa dan Gspan. Ide dari graph boosting ini yaitu membangun beberapa set feature classifier dari data training dan membuat beberapa model secara iterative mengubah data training samplenya. lalu mengkombinasikan model untuk diprediksi serta menerapkan graph classification berdasarkan struktur graph yang relevan dengan membuat binary feature classification untuk mengetahui adanya atau tidak substruktur subgraph. graph boosting cocok jika data tersebut besar karena hanya mempunyai O(n) time di banding kernel method yang mempunyai $O(n^2)$ time dalam pencarian search space dan pencocokan graph[4].

B. Algoritma gSpan

Algoritma gSpan Graph based Substructure Pattern Mining adalah algoritma yang menggunakan representasi sparse adjacency list untk menyimpan graph. gSpan merupakan algoritma untuk mengexplore depth-first search (DFS) dalam frequent subgraph mining dan bertujuan untuk mencari frequent substructure tanpa kandidat generasi dan false prunning. Algoritma ini di kemukakan oleh Xifeng Yan et al. gSpan menggunakan representasi kanonik untuk struktur graphnya yaitu DFS-Code. Langkah-langkah dalam proses algoritma gspan adalah DFS Subscripting, Rightmost path Extension, Canonical DFS Code, Lexicographic Order DFS Code[5].

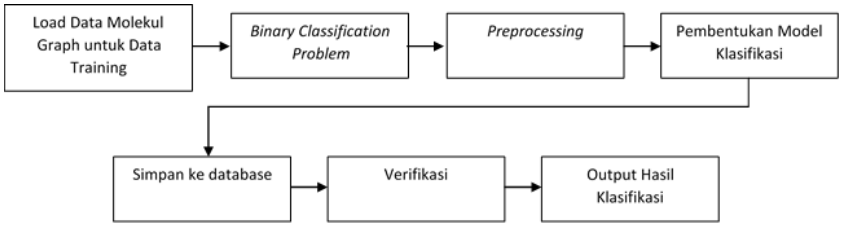
C. Algoritma LPBoost

Algoritma boosting merupakan meta-algoritma dalam machine learning untuk melakukan supervised learning. Boosting merupakan metode yang efektif untuk meningkatkan akurasi dari beberapa algoritma learning dengan mengkombinasikan rule yang digunakan. Idenya yaitu dengan membangkitkan beberapa model setiap classifier dan memberi pilian kelas yang di prediksi dan kelas yang dipilih oleh seluruh classifier. Teori boosting secara umum terjadi dalam iterasi, secara increment menambahkan weak learner ke dalam suatu strong learner pada setiap iterasi, satu weak learner belajar dari suatu data latihan. Kemudian weak learner itu ditambahkan ke dalam strong learner. Setelah weak learner ditambahkan data data kemudian diubah masing masing bobotnya. Data yang mengalami kesalahan klasifikasi akan mengalami penambahan bobot dan data data yang terkasifikasi dengan benar akan mengurangi pengurangan bobot. Oleh karena itu weak lerner pada iterasi selanjutnya akan lebih terfokus pada pada yang mengalami klasifikasi oleh weak learner yang sebelumnya. LPBoost adalah algoritma yang dikemukakan oleh (Demiriz et al,) dan termasuk algoritma boosting yang popular. Algoritma ini di desain untuk menyelesaikan masalah optimisasi soft margin. Column generation digunakan dalam large-scale integer untuk performansi column generation secara efisien. Setiap iterasi hanya subset kolom yang di gunakan untuk menjelaskan current situation (basic feasible solution). Ide dari column generation adalah untuk membatasi primal problem dengan mempertimbangkan hanya subset dari semua kemungkinan label berdasarkan weak hypothesis yang di generate. Tujuan dari LPBoost ini adalah untuk mencari optimal bobot koefisien linear dengan memaksimalkan nilai minimum margin dari semua data training sampel[7].

III. PERANCANGAN SISTEM

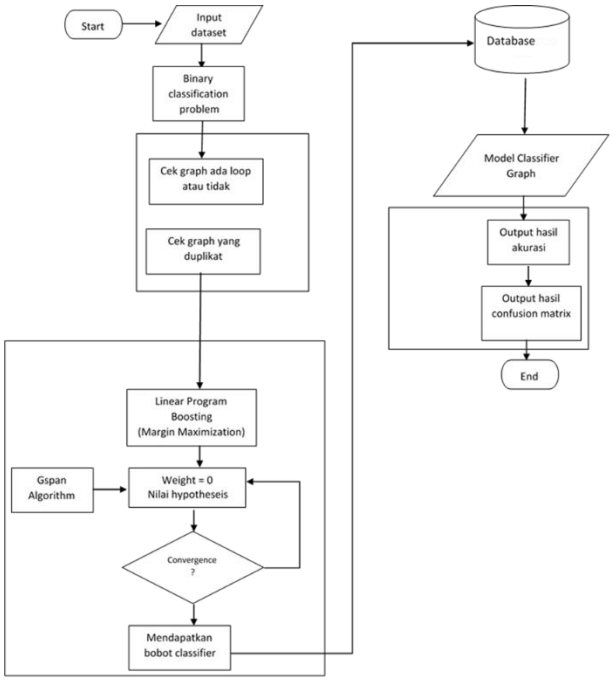
1. Gambaran Umum Sistem

Berikut ini adalah gambaran umum sistem yang akan dibangun pada Tugas Akhir ini.



2. Flowchart Sistem

Untuk penjelasan dari gambaran umum sistem lebih detilnya bisa dilihat dari flowchart dibawah ini :



Mengacu pada flowchat proses sistem di atas, detail rancangan sistemnya dijelaskan pada beberapa tahap sebagai berikut : Binary Classification Problem of Graph, Preprocessing, Pembentukan Model Klasifikasi menggunakan algoritma gSpan dan LPboost, Verifikasi, Analisis Hasil Klasifikasi.

- Binary Classification Problem merupakan proses pengambilan pola molekul graph dari dara training yang berasosiasi dengan label class yang nantinya akan digunakan dalam pembuatan prediction rule.
- Preprocessing graph molekul merupakan proses perubahan graph yang berkualitas dan bisa di proses. Yang dimaksud graph berkualitas adalah graph yang menadi data training bersih dari noise, tidak ada yang duplicate dan tidak ada self loop.
- Pada tahapan pembentukan model, metode graph boosting diterapkan untuk proses pembuatan model klasifikasi. Prosesnya yaitu menggunakan algoritma gSpan untuk enumerasi frequent pattern subgraph.
- Setelah subgraph didapatkan lalu subgraph tersebut dipakai sebagai set feature untuk mendapatkan pola dan menyimpannya dalam database. selanutnya dengan menggunakan algoritma LPBoost subgraph yang didapatkan akan dilakukan proses learning dengan algoritma LPBoost yang nantinya akan didapatkan nlai bobot masing masing subgraph tersebut. Setelah model dibuat lalu akan di test dengan data testing untuk mendapatkan hasil akurasi. Dengan menggunakan confusion matrix dan nilai bobot gain akan dibandingkan hasil dari sistem dengan label yang sebenarnya. Semakin besar nilai kemiripan graph semakin mirip dengan pola yang aslinya.
- Analisis Hasil Klasifikasi yang dihasilkan oleh sistem yaitu “non-mutagen” atau data tersebut tidak dikenali oleh sistem, sebaliknya jika nilai yang dihasilkan lebih dari threshold maka keputusan yang diambil yaitu “mutagen” atau data molekul tersebut dikenali oleh sistem.

IV. PENGUJIAN DAN SKENARIO

Untuk pengujian data menggunakan confusion matrix. Pengujian perlu dilakukan untuk mengukur performansi sistem yang telah dibangun. ada beberapa skenario yang dilakukan yaitu:.

A. Skenario 1

Skenario 1 menggunakan data keseluruhan sampel sebagai data training. Pada pengujian ini skenarionya yaitu melakukan perbandingan antar data training dan data testing untuk melihat apakah ada pengaruh dari model yang di buat oleh system dalam melakukan klasifikasi. Pengujian ini dilakukan untuk mengetahui hasil akurasi dengan cara mengubah komposisi data training dan data testing kemudian membandingkan dan menganalisa hasil akurasi yang di peroleh. Tujuan dari skenario 1 adalah untuk mencari komposisi data training dan data testing yang paling optimal

B. Skenario 2

Skenario 2 menggunakan beberapa parameter yang digunakan. Pada pengujian ini scenario yang dilakukan adalah dengan mengubah parameter data yang di pakai dengan perbandingan rasio. Analisis yang digunakan yaitu membandingkan hasil akurasi yang di dapat dari model klasifikasi dengan mengubah parameter tertentu. Disini parameter yang akan diubah yaitu parameter regularization parameter (v), convergence , dan jumlah graph. Tujuan dari skenario 2 adalah untuk mencari konfigurasi parameter yang berpengaruh yang paling optimal.

C. Skenario 3

Skenario 3 menggunakan hasil akurasi yang diperoleh dari sistem. Pada scenario ini semua data akan dilakukan pengujian performansi berdasarkan runtime proses yang di peroleh secara keseluruhan dari skenario 1 dan skenario 2 dengan pemilihan label data untuk data uji dan data latih secara random. Pemilihan label secara random adalah pemilihan data pada setiap subjek dipilih untuk data latih dan data uji secara random. Tujuan dari skenario 3 adalah untuk mengetahui performansi runtime proses dalam melakukan klasifikasi

D. Skenario 4

Skenario terakhir adalah mencari performansi hasil klasifikasi yang diperoleh yaitu dengan melakukan validasi dari hasil yang diperoleh dari system. Untuk melakukan validasi bisa menggunakan confusion matrix dan dilihat berdasarkan nilai precision recal dan f-measure. Tujuan dari skenario 4 adalah mengukur performansi tingkat akurasi sistem dihitung dari nilai precision, recall, f-measure dalam hal validasi dan identifikasi.

V. HASIL DAN ANALISIS

Pada penelitian dilakukan beberapa skenario pengujian untuk mendapatkan parameter optimal dan mengukur performansi dari sistem yang dibangun. Analisis performansi sistem dilakukan dengan menggunakan pengukuran confusion matrix. Performansi didapatkan dari hasil nilai akurasi, precision, recall dan f-measure. Semakin besar nilai Akurasi, semakin bagus performansi yang dihasilkan.

A. Spesifikasi Perangkat Keras dan Perangkat Lunak

Pengujian dijalankan di atas komputer dengan prosesor Intel(R) *Core* (TM) i5 2,4 GHz dan RAM 4 GB menggunakan sistem operasi Linux Ubuntu 12.04 64-bit dan *tools* simulasi MATLAB R2012a.

B. Dataset

Dataset yang digunakan dalam penelitian ini adalah MUTAG dataset yang terdiri dari 188 molekul kimia. Dari 188 molekul tersebut 125 bersifat mutagen dan 63 bersifat negative mutagen.

C. Pengukuran Performansi

Pengukuran performansi dilakukan dengan cara yaitu menggunakan confusion matrix menghitung akurasi, precision, recall dan f-measure, selain itu juga mengukur parameter yang berpengaruh yaitu regularization, convergence, dan jumlah data graph. Akurasi dihitung menggunakan rumus:

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

sedangkan Precision, Recall dan F-measure dihitung menggunakan rumus:

$$\text{Precission} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F-Measure} = \frac{2\ x\ precission\ x\ Recall}{Precission + Recall} \quad (4)$$

D. Parameter Regularization

Pengujian ini dilakukan untuk mencari nilai konfigurasi parameter yang paling optimal dari parameter regularization. Pengujian dilakukan dengan menggunakan data keseluruhan sampel sebagai data training secara *random*. Sehingga didapat data training dan data testing yang diuji sebanyak jumlah datasetnya dengan cross validation.. Nilai parameter berkisar dari rentang {0.05, 0.1 , 0.15 , 0.2 , 0.25} berdasarkan [7]. Hasil pengujian bisa dilihat dibawah ini

Tabel 1 – Hasil pengujian paremater regularization

Jumah Dataset	regularization parameter				
	0.05	0.1	0.2	0.3	0.4
96 graph	71,0 %	72,4 %	69,5 %	68,1 %	66,6 %
150 graph	69,3 %	68,74 %	68,35 %	67,15 %	66,87 %
188 graph	63,3 %	60,6 %	59,3 %	64,66 %	64,6 %
220 graph	63,62 %	63,10 %	62,89 %	62,63 %	61,92 %
400 graph	59,18 %	58,7 %	57,60 %	56,8 %	55,5 %
880 graph	54,3%	53,6%	53,2%	51,7%	50,9%

Dari hasil scenario tersebut dapat dilihat bahwa tinggi rendahnya akurasi dipengaruhi oleh parameter yaitu regularization parameter dan jumlah dataset yang di gunakan. Dari hasil pengujian dapat dilihat semakin banyak dataset yang digunakan semakin tidak akurat hasil klasifikasi yang di hasilkan oleh sistem hal ini dikarenakan parameter regularization ini berpengaruh terhadap pembentukan model klasifikasi pada saat perhitungan soft margin untuk mendapatkan nilai gain. Parameter ini berpengaruh dalam mengontrol misclassification error pada saat melakukan klasifikasi. Sehingga nilai parameter regularization sebaiknya diberi nilai yang rendah agar hasil klasifikasi bisa tinggi. Parameter ini baik digunakan dalam graph optimisasi proses LPBoost untuk mencari frequent subgraph. Semakin tinggi nilai regularization menyebabkan penurunan performansi dalam training set.

E. Parameter Convergence

Selanjutnya parameter yang berpengaruh adalah parameter convergence. Dengan menggunakan nilai konfigurasi parameter convergence yang didapatkan dari skenario, didapatkan bahwa nilai convergence berperan penting untuk menentukan batas nilai bobot model klasifikasi untuk menentukan pola graph yang membuat akurasi tinggi. Berikut ini adalah akurasi hasil dari pengaruh parameter convergence:

Tabel 2 – Akurasi untuk parameter convergence

Dataset	Percobaan ke -				
	1 (conv: 0,05)	2 (conv: 0,10)	3 (conv: 0,15)	4 (conv: 0,20)	5 (conv: 0,25)
96 graph	72,4 %	72,4 %	72,46 %	79,71 %	79,71 %
150 graph	69,8 %	69,8 %	71,24 %	71,24 %	71,24 %
188 graph	60,6 %	60,6 %	60,6 %	62,0 %	62,0 %
220 graph	57,74%	57,74%	57,74%	59,43 %	59,43 %
400 graph	58,1 %	58,1 %	58,73%	59, 73%	59 ,73 %
880 graph	61,36 %	61,36 %	61,36 %	63,52 %	63,52 %

Nilai convergence threshold adalah nilai ambang batas konvergensi yang digunakan dalam algoritma LPBoost untuk membatasi nilai gain. Convergace biasanya diperoleh setelah beberapa iterasi. Dalam molekul kimia nilai convergence threshold yang sering dipakai adalah adalah 0.05. dalam pengujian ini nilai convergence threshold berkisar antara 0.05 sampai 0.25. dari hasil pengujian didapatkan bahwa akurasi yang didapatkan cenderung stabil dan hanya berubah sedikit dan kenaikannya tidak terlalu signifikan. semakin besar nilai convergence maka akurasi akan semakin besar. Ini dikarenakan pada saat sistem membuat model klasifikasi menggunakan algoritma LPBoost nilai convergence akan butuh banyak iterasi untuk menyelesaikan learning proses. Semakin tinggi nilainya proses learning akan lebih cepat tetapi mungkin dapat menambah atau mengurangi akurasi tergantung pada batasan yang diambil.

F. Performansi runtime proses

Pengujian ini dilakukan untuk mengetahui pengaruh terhadap akurasi dan performansi runtime proses pada data label yang tersebar acak dan merata. Dari hasil pengujian untuk runtime dapat dilihat dari tabel 3.

Tabel 3 Akurasi dan performansi runtime

Data Training	Data Testing	Runtime (sec)	Akurasi
96 graph	69 graph	65.59	72,4 %
150 graph	69 graph	128.32	73,28 %
188 graph	150 graph	265.2	74,31%
220 graph	150 graph	382.42	77,62 %
400 graph	220 graph	612.97	76,05%
880 graph	220 graph	997.06	78,18%

Berdasarkan hasil uji skenario yang diperoleh dari pengujian runtime, dapat disimpulkan bahwa banyaknya pola graph pada dataset mempengaruhi hasil klasifikasi yang diperoleh ini dikarenakan proses runtime akan semakin lama dalam enumerasi pencarian frequent subgraph sehingga proses runtime berbanding lurus dengan jumlah pola graph. Terbukti bahwa pengujian runtime ini kompleksitas waktunya linear O(n) sehingga dapat disimpulkan semakin banyak data maka proses akan semakin lama. Nilai akurasi untuk pemilihan data label random mendapatkan nilai akurasi yang bervariasi berdasarkan beberapa kali uji coba dan mendapatkan nilai akurasi terbaik 78,18%.

G. Pengaruh Perbandingan Jumlah Data

Pada skenario terakhir ini, bertujuan untuk untuk mencari konfigurasi orde dan radius yang paling optimal. Pengujian ini dilakukan dengan cara melakukan perbandingan pengaruh data training dan data testing dengan jumlah interval yang berbeda. Pada pengujian ini yang terhadap pengaruh data training dan data testing terhadap hasil akurasi yaitu:

a. Pengujian pengaruh data terhadap scenario jumlah data testingnya sama dengan data trainingnya berbeda beda.
b. Pengujian pengaruh data terhadap scenario jumlah data testingnya berbeda beda dengan data trainingnya sama..

Disini scenario yang dilakukan adalah dengan melakukan percobaan dimana untuk data testingnya dan trainingnya berbeda beda. Tabel di bawah ini menunjukan akurasi dari masing-masing kombinasi:

Tabel 4 - Pengujian Pengaruh Data Training

Dataset		
Data Training (random)	Data Testing	Hasil
69 graph	188 graph	51,06 %
96 graph	188 graph	55 %
150 graph	188 graph	59,04 %
220 graph	188 graph	60,10 %
400 graph	188 graph	58,15 %
880 graph	188 graph	**61,36 %**

Dari hasil scenario yang telah dilakukan diatas dalam melakukan percobaan scenario pengujian pengaruh data train ini dapat dilihat bahwa semakin banyak data yang di training maka hasil akurasi akan semakin besar ini dikarenakan sistem bisa menangani pola graph yang semakin beragam dalam data training sehingga nilai akurasi yang diperoleh cenderung naik.

Tabel 5 - Pengujian Pengaruh Data Testing

Dataset		
Data Training	Data Testing (random)	Hasil
188 graph	69 graph	**70,3 %**
188 graph	96 graph	68,08 %
188 graph	150 graph	66,7 %
188 graph	220 graph	58,9 %
188 graph	400 graph	51,5 %
188 graph	880 graph	49,6 %

Dari hasil scenario yang telah dilakukan diatas untuk data testing dapat dilihat bahwa semakin banyak data yang di testing maka hasil akurasi akan semakin kecil ini dikarenakan system tidak bisa menangani pola graph yang semakin beragam dalam data training. Sehingga kemungkinan adanya misclassification error tinggi sehingga karakteristik data sangat berpengaruh terhadap hasil akurasi .

Untuk analisa jumlah pola graph terhadap nilai akurasi hasil pengujian pada tabel 4 dan tabel 5 , dapat dilihat semakin banyak jumlah data pada data training maka semakin banyak nilai akurasi hal ini dikarenakan semakin banyak data yang digunakan hasil klasifikasi dari true positive false positive dan false negative akan semakin besar juga dari ketiga parameter itulah yang nantinya akan didapatkan nilai akurasi. False positive dan false negative digunakan sebagai pembanding untuk mencari nilai precision dan recall. Apabila nilai dari false positif dan false negative sebagai pembanding semakin besar maka akan didapatkan nilai precision dan recall yang semakin kecil dan nilai ini berpengaruh terhadap hasil akurasi. model klasifikasi yang dibuat oleh sistem dapat menangani keberagaman data dan dapat mewakili dataset tersebut dengan benar. Meskipun hasil akurasi tergolong rendah ini dikarenakan pattern substruktur graph yang diklasifikasikan beragam

Hasil pengujian skenario Pada skenario terakhir ini, bertujuan untuk mengukur performansi tingkat akurasi sistem dihitung dari nilai precision, recall, f-measure dalam hal validasi dan identifikasi. Skenario ini menggunakan rasio data latih dan data uji. Hasil pengujian ini dilihat berdasarkan hasil rata rata jumlah dataset terhadap akurasi, precision, recall, dan f-measure:

Tabel 6 - Performansi Sistem

Dataset	Percobaan ke -				
	1 (conv: 0,05)	2 (conv: 0,10)	3 (conv: 0,15)	4 (conv: 0,20)	5 (conv: 0,25)
96 graph	72,4 %	72,4 %	72,46 %	79,71 %	79,71 %
150 graph	69,8 %	69,8 %	71,24 %	71,24 %	71,24 %
188 graph	60,6 %	60,6 %	60,6 %	62,0 %	62,0 %
220 graph	57,74%	57,74%	57,74%	59,43 %	59,43 %
400 graph	58,1 %	58,1 %	58,73%	59, 73%	59 ,73 %
880 graph	61,36 %	61,36 %	61,36 %	63,52 %	63,52 %

Berdasarkan hasil dari pengujian scenario di atas dapat dilihat bahwa semakin banyak data training nilai f-measure akan semakin turun ini dipengaruhi dengan nilai precision dan recall yang nilainya beragam. Selain itu pada data graph

yang berjumlah 400 ada penurunan hasil baik itu akurasi precision recall dan f-measure secara signifikan ini disebabkan terjadi misclassification error yang tinggi. Terjadi misclassification ini ada beberapa penyebabnya yaitu model klasifikasi klasifikasi menghasilkan nilai yang rendah karena karakteristik data yang beragam sehingga untuk pencocokan subgraph dengan graph molekul berbeda, selain itu bentuk pattern dan nilai bobotnya berpengaruh terhadap nilai akurasi.

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang telah dilakukan terhadap sistem dan hasil analisis dapat disimpulkan bahwa:

1. Metode graph classification bisa diimplementasikan pada struktur graph molekul kimia dalam melakukan klasifikasi dengan cara membuat binary feature vector berdasarkan ada atau tidaknya frequent pattern.
2. Semakin banyak data yang digunakan hasil akurasi dari system semakin tinggi. hasil rata rata dari pengujian menunjukan hasil tertinggi akurasi pada dataset 880 data graph dengan akurasi sebesar 78,18 %.
3. Banyaknya data training yang dipakai untuk pembuatan model klasifikasi akan semakin bagus dan tinggi hasil klasifikasi yang diperoleh karena karakteristik data semakin beragam.
4. Dari kombinasi algoritma gspan dan lpboost ada parameter yang berpengaruh besar dalam meningkatkan performansi sistem yaitu regularization, convergence dan jumlah data

Adapun saran yang penulis ajukan dalam penelitian selanjutnya yaitu:

1. Dapat ditambahkan atau dikombinasikan dengan metode lain untuk mencari bentuk graph multi-label atau unlabel.
2. Bisa menggunakan dataset graph molekul kimia yang lebih besar struktur molekulnya.
3. Enumerasi frequent pattern bisa menggunakan algoritma selain gspan untuk lebih efisien.

REFERENSI

[1] J. W. d. E. E. I. Robinson, Graph Databases, O'Reailly Media, 2013..

[2] H. H. X. Q. S. L. F. H. P. R. Tong W, "Assessing QSAR Limitations – A Regulatory Perspective," Current Computer-Aided Drug Design, p. 195–205, April 2005.

[3] S. N. K. K. T. Hiroto Saigo, "gBoost: a mathematical programming approach to graph classification and regression," Machine Learning, vol. 75, no. 1, pp. 69-89, 2009.

[4] H. S. Joji Tsuda, "Graph Classification," Managing and Mining Graph Data, pp. 337-363, 2010.

[5] J. H. Xifeng Yan, "gSpan: Graph-Based Substructure Pattern Mining," In Proceedings of the 2002 IEEE International Conference on IEEE Society, p. pp. 721–724, December 09 - 12, 2002

[6] W. M. J. Mohammed J. Zaki, "Data Mining and Analysis: Fundamental Concepts and Algorithms," Cambridge University Press, 2014.

[7] A. B. K. S.-t. J. Demiriz, "Linear programming Boosting via Column Generation," Machie Learning, vol. 46, pp. 225-254, 2002.

[8] A. Inokuchi, "Mining generalized substructures fom a set labeled graph," In proceedings of the 4th IEEE International Conference on Data Mining, pp. 415-418, 2005.

[9] A. L. d. C. R. D. G. S. A. a. H. C. Debnath, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity," J. Medicinal Chemistry, no. 34, p. 786–797, 1991.

[10] T. K. a. E. M. a. Y. Matsumoto, "An Application of Boosting to Graph Classification," 2004.

[11] H. F. a. J. Huan, "Boosting with structure information in the functional space: An application to graph classification," Proc. 16th ACM SIGKDD, p. 643–652, 2010.