Aplikasi Rekomendasi Metode Analisis Sesuai dengan Karakter Data

Andre Gunawan, Henry Novianus Palit, Andreas Handojo
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: agun111295@gmail.com, hnpalit@petra.ac.id, andreas@petra.ac.id

ABSTRAK

Data telah menjadi asset yang sangat berharga dalam perusahaan, namun yang lebih berhaga dari data adalah informasi yang dapat kita olah dari data tersebut. Informasi yang kita dapatkan dari data memiliki banyak manfaat. Dalam pengambilan keputusan bisnis data memiliki peran yang besar untuk menentukan apakah tindakan bisnis yang diambil dapat menimbulkan dapak baik pada perusahaan.Perusahaan – perusahaan yang mendasari keputusan yang dibuat berdasarkan data yang ada (*Data-Driven Decision Making*) disebut sebagai *Data-Driven Company*, perusahaan ini pada uumumnya memiliki sebuah divisi yang bergerak dalam bidang *Business Intelligence*.

Business Intelligence adalah sebuah sistem terintegrasi yang memberikan fakta / informasi dari data yang telah diolah untuk kepentingan pengambilan keputusan. Sistem ini bertugas untuk memberikan dukungan keputusan (Decision Suport) untuk tujuan yang spesifik pada sebuah proses bisnis. Fondasi dari Business Intelligence adalah data yang telah diolah menjadi informasi yang berguna untuk proses pendukungan keputusan. *Business Intelligence* menggunakan bergabai metode untuk melakukan ekstraksi informasi dari data yang ada. *Business Intelligenc*e memberikan informasi pada waktu yang tepat, kepada orang yang tepat dalam bentuk yang mudah dipahami pula.

Kata Kunci: *Business Intelligece, Classification, Regression*

ABSTRACT

Data has become a very valuable asset in the company, but the more powerful than the data is information that we can though from the data. The information we get from the data has many benefits. In business decision making the data has a big role to determine whether the business action taken can lead to a good foot in the company. Companies that underlie data-driven decision making (Data-Driven Decision Making) are referred to as Data-Driven Company this in general has a division that is engaged in Business Intelligence.

Business Intelligence is an integrated system that provides facts / information from data that has been processed for the benefit of decision making. This system is in charge of providing decision support for the specific purpose of a business process. The foundation of Business Intelligence is data that has been processed into useful information for the decision support process. Business Intelligence uses a multitude of methods to extract information from existing data. Business Intelligence provides information at the right time, to the right person in easy-to-understand form.

Keywords: *Business Intelligece, Classification, Regression*

1. PENDAHULUAN

Data telah menjadi asset yang sangat berharga dalam perusahaan, namun yang lebih berhaga dari data adalah informasi yang dapat kita olah dari data tersebut. Informasi yang kita dapatkan dari data memiliki banyak manfaat. Dalam pengambilan keputusan bisnis data memiliki peran yang besar untuk menentukan apakah tindakan bisnis yang diambil dapat menimbulkan dampak baik pada perusahaan.Perusahaan – perusahaan yang mendasari keputusan yang dibuat berdasarkan data yang ada (*Data-Driven Decision Making*) disebut sebagai *Data-Driven Company*, perusahaan ini pada uumumnya memiliki sebuah divisi yang bergerak dalam bidang *Business Intelligence*. Karena banyak perusahaan yang melakukan *business intelligence,* maka kebutuhan untuk melakukan analisa semakin tinggi. *Analytics tools* yang interaktif dan dapat memberikan rekomendasi metode analisis akan sangat membantu proses analisa.

Sistem yang dibuat akan melakukan proses *data pre-processing* yang memungkikan pengguna untuk melakukan:

1. Data Cleaning
   Mengisi data yang hilang dengan data lainya yang dapat menggmbarkan data yang hilang tersebut dalam bentuk yang general [12].
2. Data Transformation
   Mengubah data kedalam bentuk yang lain untuk mempermudah proses analisa. Transformasi data juga dapat memberikan variable baru untuk dianalisa atau mengurangi dimensi dari data (*dimention reduction*) [12].
3. Data Integration
   Menggambungkan data dari berbagai sumber dan dari berbagai format menjadi format baru yang memudahkan untuk melakukan analisa

Setelah proses *data-preprocessing* selesai maka sistem dapat memberikan rekomendasi metode analisis. Rekomendasi ini dapat diterima atau di tolak oleh pengguna.

2. DASAR TEORI

2.1 Business Intelligence Task

Untuk mencapai tujuan dalam melakukan analisa ada beberapa tugas / *task* yang harus dilalui. Beberapa tugas / *task* itu adalah:

1. Data *Task*
   Tugas ini adalah bagaimana mendapatkan data dan apa pengertian / informasi mengenai data tersebut. Informasi yang didapatkan saat melakukan *Business Intelligence* semua berasal dari data, maka tugas ini perlu dilakukan untuk mengetahui sumber data.

2. *Business and Data Understanding*
   Tugas ini dilakukan untuk memahami dan mengerti apa yang dimaksut dengan data yang dimiliki. Selain itu pengertian mengenai proses bisnis juga diperlukan untuk memahami bagaimana data tergenerasi.

3. *Modeling Task*
   Tugas ini dilakukan untuk mengetahui model analisa yang ingin dilakukan. Model analisa ini digunakan untuk menghasilkan informasi yang dibutuhkan.

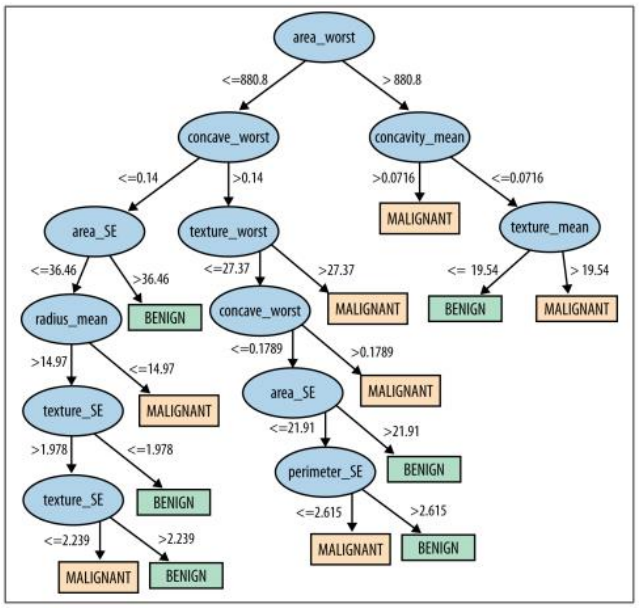
4. *Analysis Task*
   Tugas ini adalah implementasi dari *Modeling Task* dimana algoritma digunakan untuk menghasilkan informasi. Tugas ini adalah tugas utama dari *Business Intelligence* untuk menghasilkan fakta / pengetahuan mengenai bisnis.

5. Evaluation & Reporting Task
   Tugas ini adalah tugas terakhir dalam *Business Intelligence* tugas ini dilakukan untuk melakukan evaluasi mengenai performa yang dihasilkan model yang dibuat. Pada tahap ini visualisasi dari data juga diperlukan untuk proses *reporting.* Visualisasi yang baik dapat menghasilkan informasi yang mudah untuk dipahami [9].

2.2 Metode Prediksi

2.2.1 Classification

Classification merupakan metode yang mencoba untuk melakukan prediksi dengan melakukan klasifikasi data. Klasifikasi mencari data tersebut milik *class* yang mana. Salah satu contoh pertanyaan *classification* adalah "Dari semua *customer* siapa yang akan merespon penawaran yang akan diberikan?". Dalam kasus ini terdapat 2 *class* yaitu respon dan tidak respon. Salah satu contoh metode yang dapat melakukan *classification* adalah *decision tree. Decision tree* merupakan metode yang memodelkan data menjadi bentuk sebuah *tree* dimana data baru yang yang masuk akan berjalan dan menghasilkan sebuah alur yang menyatakan pada *class* mana data tersebut terklasifikasi [9].

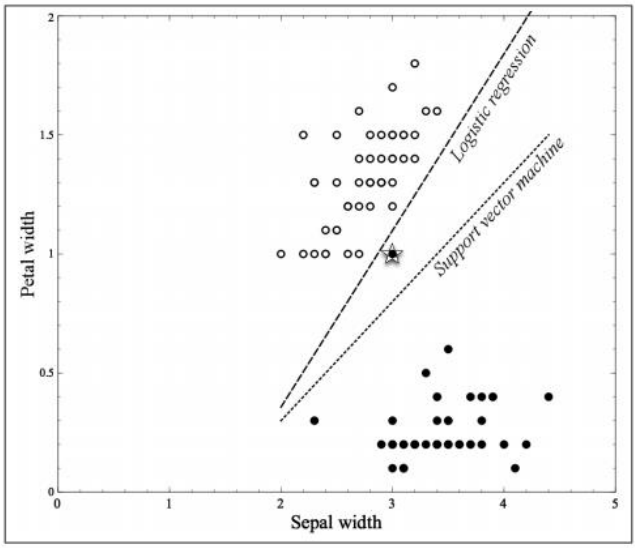


Gambar 1. Contoh Metode Classification [9]

2.2.2 Metode Regression

Regression atau value estimation mencoba untuk melakukan estimasi atau meprediksi sebuah nilai numeric variabel pada sekumpulan data. Salah satu contoh pertanyaan *regression* adalah "Berapa banyak customer yang akan menggunakan layanan kita". Properti variabel yang ingin diprediksi adalah jumlah penggunaan layanan dan model analisa dapat terbuat dengan melihat variable lain yang sejenis.

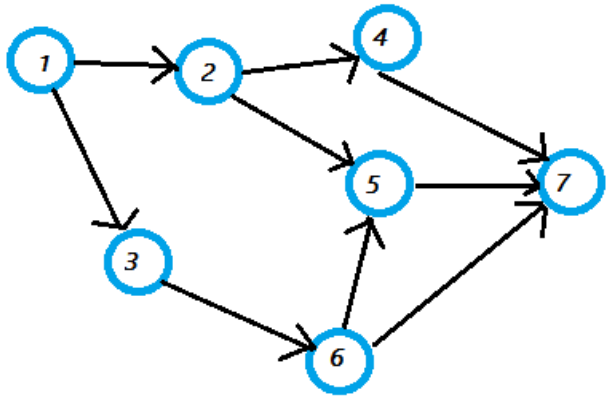
Regression berhubungan dengan Classification tapi bukan merupakan sesuatu hal yang sama. Classification mencoba untuk memprediksi apakah sesuatu akan terjadi atau tidak, sedangkan Regression mencoba untuk memprediksi berapa banyak yang akan dihasilkan [9].



Gambar 2. Contoh Metode Regresi [9]

2.3 Directed Acyclic Graph (DAG)

Directed Acyclic Graph merupakan *graph* yang tidak memiliki *cycle* atau kondisi dimana sebuah *node* kembali ke *node* sebelumnya [8].



Gambar 3. *Directed Acyclic Graph [10]*

DAG digunakan untuk membuat tampilan *interface data flow* yang interaktif. Tampilan ini merupakan bentuk antar muka yang antara *user* dengan aplikasi, dimana *user* melakukan *pre-processing* data. DAG tidak memiliki siklus sehinggan membuat tampilan lebih mudah dipahami.

2.4 Flask

Flask merupakan *microframework* yang dibangun dengan menggunakan bahasa pemrograman *Python*. *Flask* digunakan untuk me-*develop* sebuah aplikasi web. *Flask* merupakan *microframework* yang artinya *flask* membuat sebuah pengerjaan aplikasi web menjadi mudah dan *simple* karena dapat menjalankan sebuah web hanya dengan menggunakan 1 file *Python*. *Flask* membuat susunan kerja yang ringan, dan mudah tetapi juga dapat dikembangkan dengan mudah [3].

2.5 Data Kaggle

Dataset yang digunakan dalam penelitian ini adalah data – data yang berasal dari website kaggle. Data – data tersebut antara lain:

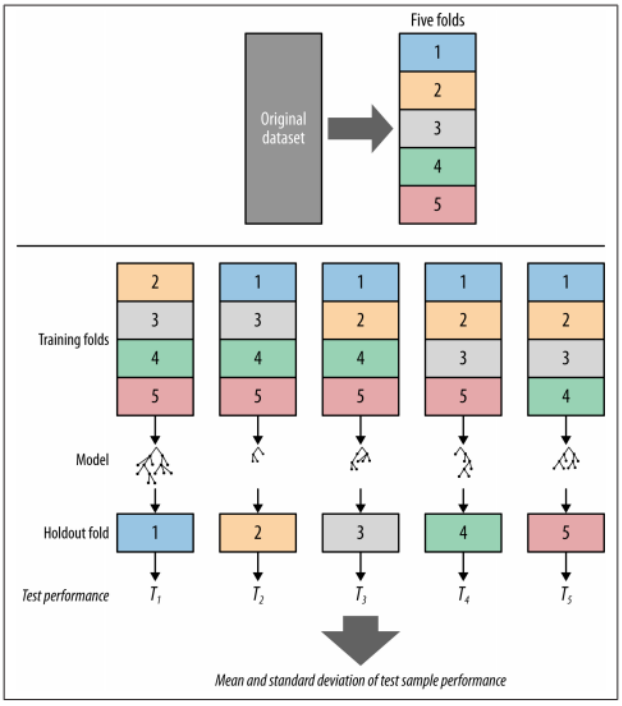
a. *Human Resource* [5]
b. *House Price [6]*
c. *West Nile Virus [7]*
d. *Bakery [11]*

2.6 Collaborative Analysis

Colaborative Analytics adalah sebuah metode yang digunakan untuk meningkatakan hasil akurasi berdasarkan banyak predictive model. Ketika data training siap diproses, data di proses dengan lebih dari 1 predictive model. Sebagai contoh, jika ada sebuah data yang ingin di proses dengan metode classification, maka data tersebut diproses dengan banyak classifier. Classifier terbaik akan dipilih sebagai classifier yang akan digunakan untuk mem-prediksi data training [2].

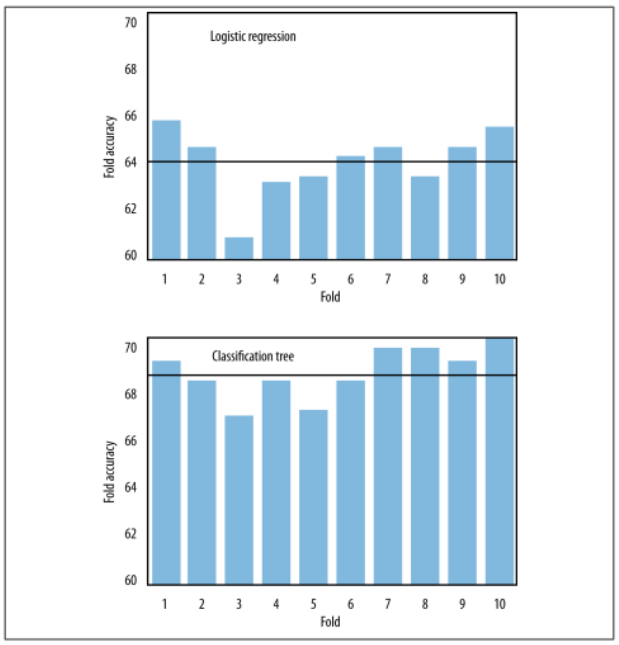
2.7 Cross Validation

Merupakan metode yang memecah data utuh menjadi k bagian data, dimana k merupakan nilai yang ditentukan oleh pengguna. bagian dari data ini akan dimodelkan sedangkan sisanya akan dijadikan data untuk melakukan testing.



Gambar 4. Contoh Cross Validation [9]

Pada gambar diatas, data dibagi menjadi 5 bagian. Untuk setiap bagian akan dijadikan sebagai *data testing*. Kemudian setiap bagian dari data tersebut akan menghasilkan nilai performa / akurasi [9].



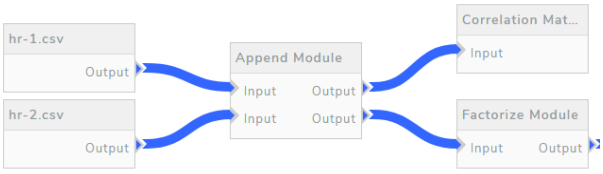
Gambar 5. Contoh Performa Cross Validation [9]

Hasil dari performa yang dilakukan terdiri dari 2 nilai, yaitu rata – rata dan *standard deviation* semua nilai akurasi pada *fold* data / bagian data. Nilai rata – rata adalah nilai akurasi yang baru yang menentukan seberapa baik performa metode analisis setelah dilakukan pembagian data. *Standard deviation* adalah nilai yang menentukan rata – rata jarak antar data pada nilai rata – rata. Nilai tersebut digunakan untuk menentukan presisi dari metode analisis.

3. DESAIN SISTEM

3.1 Data Flow

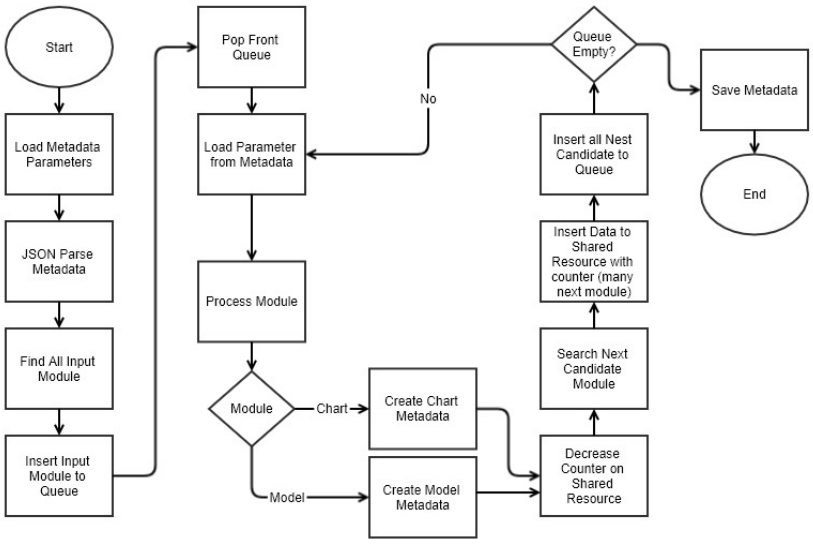
Data Flow merupakan tampilan antar muka yang memudahkan *user* untuk melakukan proses *pre-processing*. Tampilan ini dibuat berdasarkan DAG, yang mana masing – masing *node* memiliki parameter / data yang diinputkan *user [1]*.



Gambar 6. Tampilan Data Flow

3.2 Flow Process

Proses ini dilakukan untuk mengubah metadata dari aplikasi menjadi urutan proses kerja. Proses ini akan dijalankan ketika ada *request* untuk menjalankan proses dari *client*. *Service* kemudian menjalankkan algoritma pada Gambar 7. Setelah algoritma dijalankan maka data yang dihasilkan akan ditampilkan. Selain itu data seperti gambar grafik dan evaluasi model juga disimpan dan ditampilkan sebagai hasil dari menjalankan alur proses.



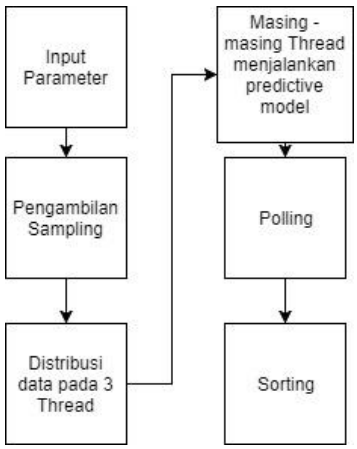
Gambar 7. Flowchart process pengubahan

Pertama sistem akan mencari semua modul yang memiliki tipe *input*. Semua modul ini akan dianggap sebagai *root node* dari DAG. Setelah itu data *modul input* tersebut dimasukan kedalam *queue*. Selanjutnya *queue* akan dijalankan sampai tidak ada data di dalam *queue* yang tersisa. Setiap kali data dikeluarkan dari *queue* maka data tersebuat akan diproses dan hasil dari proses tersebut akan dimasukan kedalam penyimpanan data sementara yang disebut sebagai *shared resource. Share resource* merupakan array yang menyimpan data hasil proses sebuah modul. Jika proses selesai dijalankan dan data berhasil masuk kedalam *shared resource* maka *queue* akan ditambahkan dengan data proses yang selanjutnya dijalankan setelah sebuah modul selesai dijalankan. Data proses yang selanjutnya dikerjakan merupakan data dari *node* yang memiliki *input port* yang berasal dari *output por*t modul yang sedang diproses.

DAG juga memiliki modul output dimana modul ini tidak memiliki *output port.* Modul ini tidak akan menambahkan kandidat kedalam *queue.* Modul ini terdiri dari modul *chart* dan *model.* Setiap data yang dihasilkan dari modul ini akan disimpan dalam array selain *shared resource.*

3.3 Proses Rekomendasi

Proses ini dilakukan secara multithreading dengan membagi thread sebanyak metode prediksi yang ingin dijalankan. Untuk sistem yang diuji akan menggunakan 3 metode analisis sehingga jumlah *thread* adalah 3. Proses dimulai dengan pemisahan data menjadi n bagian sesuai dengan banyak metode prediksi yang ingin dijalankan bersama. Setelah itu data pada masing – masing thread akan di olah sesuai dengan metode prediksi tiap thread. Setelah itu terjadi proses polling dimana aplikasi akan menunggu hasil dari semua thread, jika semua hasil tiap thread sudah didapatkan, maka selanjutnya adalah proses sorting. Proses sorting memiliki 2 mode, yaitu *accuracy* atau *time.* Proses sorting akan melakukan sort berdasarkan mode yang dipilih pengguna. Sistem rekomendasi akan selalu memilih model yang memiliki nilai akurasi dari nilai tertinggi dimana untuk mode *time* akan memilih model yang memiliki waktu proses *training* dari rendah. Proses rekomendasi secara singkat dapat digambarkan seperti Gambar 8.



Gambar 8. Proses Rekomendasi Secara Umum

4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada computer dengan spesifikasi:

- RAM: 8GB, DDR3
- HDD: 500 GB
- CPU: Core i5
- OS Ubuntu Server 16.06 LTS

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 2.7.13. *Framework* yang digunakan untuk sistem ini adalah Flask *Micro-Framework.* Adapun beberapa *library* yang mendukung sistem ini adalah:

- Pandas
- Matplotlib
- Scikit-learn
- Flask-CORS
- Jupyter Notebook
- Pymysql
- Scipy

Selain itu aplikasi juga dijalankan dalam server dan diatur oleh Apache Server dengan memanfaatkan mod_wsgi. Database yang digunakan pada aplikasi ini adalah MySQL.

Aplikasi web ini dibuat dengan menggunakan arsitektur *Service Oriented*. Setiap tabel pada database akan dibuat sebuah API (*Application Programming Interface*) yang akan membantu interaksi aplikasi dengan database. Setiap data memiliki akses dengan berbagai *HTTP Method*, antara lain:

- GET (Menerima data dalam bentuk *array* atau *object*)
- POST (Memasukan data, INSERT)
- PUT (Mengubah data, UPDATE)
- DELETE (Menghapus data, DELETE)

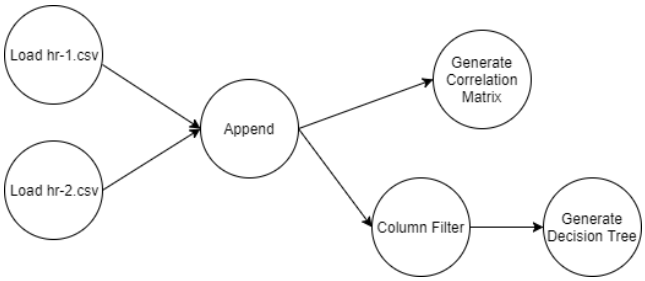
5. ANALISA DAN PENGUJIAN

5.1 *Data Flow*

Data flow diuji dengan hasil berjalanya proses, apakah hasilnya sesuai dengan *module - module* yang di inginkan.

5.1.1 Bentuk DAG

Bentuk DAG yang diuji adalah sebagai berikut



Gambar 9. DAG yang diuji

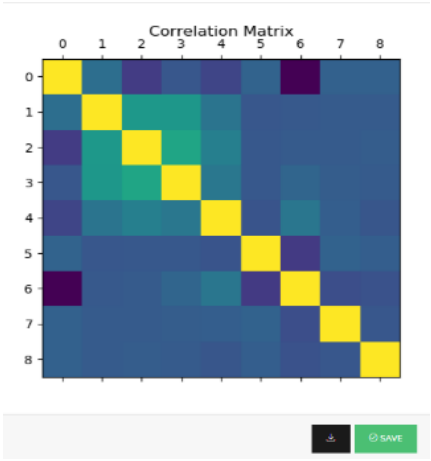
Data Flow diatas akan menghasilkan sebuah model dan sebuah *correlation matrix* dalam bentuk *chart*. Setelah di jalankan aplikasi menghasilkan tampilan data seperti pada Gambar 10.

DATA

	SATISFACTION_LEVEL	LAST_EVALUATION	NUMBER_PROJECT
0	0.38	0.53	2
1	0.80	0.86	5
2	0.11	0.88	7
3	0.72	0.87	5
4	0.37	0.52	2

Gambar 10. Data Hasil DAG

Data yang dihasilkan sesuai dengan modul dalam DAG Gambar 9. Data yang dihasilkan akan ditamplkan dalam bentuk tabel dengan jumlah baris 10. Selain itu statistik singkat data seperti *count, average, std.deviation, max, min,* 25% (kuartil atas), 75%(kuartil bawah). Statistik singkat data dapat membantu pengguna dalam memahami kondisi data yang sedang diproses.



Gambar 11. Chart Hasil DAG

Hasil kedua yang tampil adalah *chart* / grafik. Chart yang diinginkan sesuai dengan *module* yang digunakan. Grafik yang dihasilkan dapat didownload atau disimpan dalam sistem.

Gambar 12. Model Hasil DAG

Hasil ketiga adalah overview dari model yang digunakan.. Statistik sederhana ini memberikan gambaran akurasi dari metode yang digunakan. Nilai yang ditampikan pada tampilan ini adalah rata – rata dan *standard deviation* dari akurasi yang sudah divalidasi.

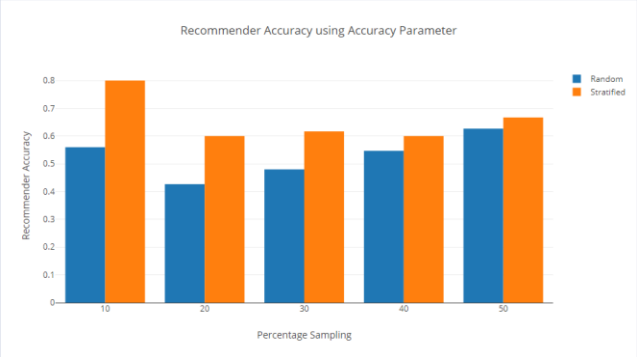
5.2 Pengujian Sistem Rekomendasi

Pengujian sistem rekomendasi dilakukan dengan 4 data, dari setiap data proses akan berjalan 5 kali dimana setiap iterasi aplikasi akan melakukan sampling dari 10% sampai 50% dengan rentan 10%. Sampling yang dilakukan ada 2 yaitu random sampling dan stratified sampling. Perhitungan akurasi rekomendasi dilakukan dengan membandingkan 2 urutan kemudian melakukan preseentasi berapa banyak yang memiliki urutan yang sama Hasil rata – rata dari pengujian ini dapat dilihat pada Tabel 1:

Tabel 1. Hasil Sistem Rekomendasi, Parameter Accuracy

Persentase	Random Sampling	Stratified Sampling
10%	0.56	0.8
20%	0.426666667	0.6
30%	0.48	0.616666667
40%	0.546666667	0.6
50%	0.626666667	0.666666667
AVG	0.528	0.656666667

Terlihat bahwa rata – rata *accuracy recommender* lebih baik ketika *sampling* yang digunakan adalah *Stratified Sampling*.



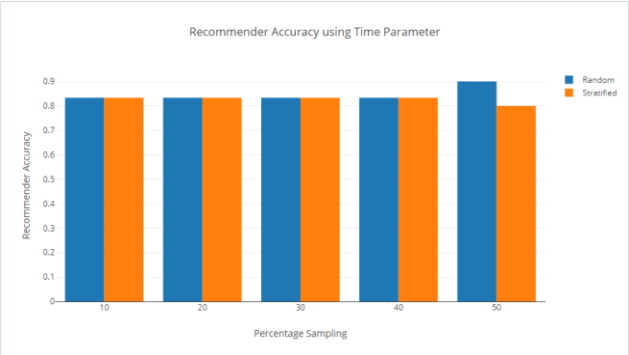
Gambar 13. Chart Hasil Sistem Rekomendasi Parameter Accuracy

Selanjutnya pengujian akan dilakukan terhadap parameter *time*. Pengujian ini dilakukan dengan cara yang sama dengan sebelumnya namun ketika proses sorting dilakukan, aplikasi akan melakukan sort berdasarkan kecepatan waktu dalam eksekusi metode analisis. Pengujian ini juga dilakukan dengan membandingkan metode *random sampling* dengan *stratified sampling*.

Tabel 2. Hasil Sistem Rekomendasi, Parameter Time

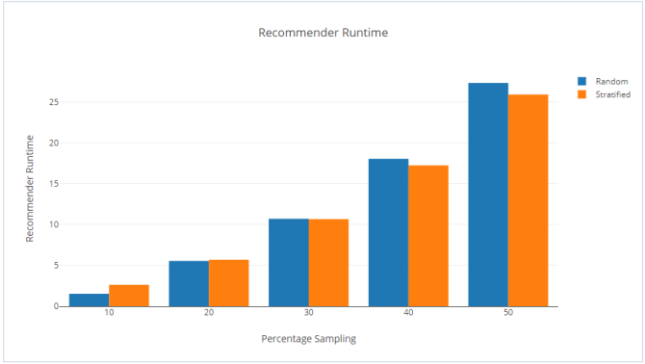
Persentase	Random Sampling	Stratified Sampling
10	0.833333333	0.833333333
20	0.833333333	0.833333333
30	0.833333333	0.833333333
40	0.833333333	0.833333333
50	0.9	0.8
AVG	0.846666667	0.826666667

Terlihat bahwa rata – rata *accuracy recommender* lebih baik ketika *sampling* yang digunakan adalah *Random Sampling*.



Gambar 14. Chart Hasil Sistem Rekomendasi Parameter Time

Pengujian yang terakhir adalah dengan melihat waktu berjalanya sistem rekomendasi. Pengujian ini akan dilakukan dengan membandingkan rata – rata lamanya sistem rekomendasi memberikan hasil rekomendasi jika menggunakan metode *random sampling* dan *stratified sampling*. Pengujian ini dilakukan dengan melihat persentase *sampling* yang dilakukan.



Gambar 15. Chart Lama Runtime Sistem Rekomendasi

Hasil yang dicapai dalam penelitian ini tidak terlalu baik dimana sistem rekomendasi tidak dapat memberikan rekomendasi yang relevan. Seringnya perubahan nilai akurasi ketika metode analisis diberikan data dengan ukuran *sampling* yang berbeda membuat sistem rekomendasi tidak dapat memberikan rekomendasi yang *general*. Kondisi ini terjadi karena tidak konsistenya metode analisis dalam data yang diberikan. Metode analisis juga dipengaruhi oleh kondisi data yang diproses. Kondisi data dapat membuat metode analisa tidak dapat memodelkan data sesuai dengan seharusnya. Banyak faktor pada data yang dapat menyebabkan kondisi demikian.

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Sistem rekomendasi dipengaruhi oleh metode analisis dan metode analisi dipengaruhi oleh data.
- Metode analisis cenderung berubah – ubah akurasinya dan dipengaruhi oleh data.
- Sistem rekomendasi tidak dapat sepenuhnya bisa menebak atau hampir tidak mungkin memberikan kesipulan metode analisis mana yang lebih baik.
- Standar deviasi adalah salah satu faktor yang harus diperhatikan dalam sistem rekomendasi.
- Data sangat mempengaruhi tetapi data memiliki deviasi yang tinggi, selain itu data juga banyak memiliki *missing value*.
- *Data Flow* dapat diartikan kedalam urutan kerja dan menghasilkan data yang sesuai.

Saran untuk pengembangan kedepannnya adalah:

- Penambahan module pada halaman *workspace*.
- Pengembangan *error log* yang lebih interaktif untuk pengguna terutama ketika *error* yang terjadi merupakan *runtime error*.
- Pengembangan fitur untuk memberikan *hyper parameter tuning* seperti *GridSearch* atau *GridSearchCV* untuk meningkatkan sistem rekomendasi metode analisa beserta parameternya.
- Integrasi dengan infrastruktur *Big Data* seperti Hadoop, Spark, Hive, Kafka.
- Peningkatan infrastruktur *cloud* yang terdistribusi sesusai dengan point sebelumnya, dengan tujuan untuk meningkatkan kekuatan *computing* dan infrastruktur memiliki sifat skalabilitas

7. DAFTAR PUSTAKA

[1] Akidau, T., et.al, S. 2015. .*Processing of VLDB Endowment.*

[2] Chong, C.S., et.al. 2012. Collaborative Analytics for Predicting Expressway-Traffic Congestion. *ICEC'12, Singapore.*

[3] Flask. Flask Overview. URI = http://flask.pocoo.org

[4] Guo, T.,Xu, J., et.al. 2016. Ease the Process of Machine Learning with Dataflow. CIKM'16.

[5] Kaggle. 2015. 11 Januari 2018; Human Resources Analytics. URI = https://www.kaggle.com/ludobenistant/hr-analytics/data

[6] Kaggle. 2015. 11 Januari 2018; House Prices: Advanced Regression Techniques. URI = https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[7] Kaggle. 2015. 11 Januari 2018; West Nile Virus Prediction. URI = https://www.kaggle.com/c/predict-west-nile-virus

[8] Monash. 1999. 11 Januari 2018. Directed Acyclic Graphs. URI = http://users.monash.edu/~lloyd/tildeAlgDS/Graph/DAG

[9] Provost, F dan Fawcett, T. 2013. 11 Januari 2018. Data Science for Business. O'Reilly

[10] Stephanie. 2016. 11 Januari 2018. Statistics How to. Acyclic Graph & Directed Acyclic Graph: Definition, Examples. URI = http://www.statisticshowto.com/directed-acyclic-graph/

[11] Trac. 2015. 11 Januari 2018; Extended BAKERY dataset. URI = https://wiki.csc.calpoly.edu/datasets/wiki/ExtendedBakery

[12] Tomar, D.,Agarwal, S. A Survey on Pre-processing and Post-processing Techniques in Data Mining. 2014. *International Journal of Database Theory and Application*.