

SKRIPSI

TEKNIK KLASIFIKASI INKREMENTAL BIG DATA PADA SISTEM TERSEBAR SPARK



Cindia Winarta

NPM: 2016730015

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»

UNDERGRADUATE THESIS

«JUDUL INGGRIIS»



Cindia Winarta

NPM: 2016730015

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY

«tahun»

LEMBAR PENGESAHAN

TEKNIK KLASIFIKASI INKREMENTAL BIG DATA PADA SISTEM TERSEBAR SPARK

Cindia Winarta

NPM: 2016730015

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

TEKNIK KLASIFIKASI INKREMENTAL BIG DATA PADA SISTEM TERSEBAR SPARK

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»

Meterai Rp. 6000

Cindia Winarta
NPM: 2016730015

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Apache Spark	5
2.1.1 Fitur-fitur Apache Spark	5
2.1.2 Komponen-komponen Spark	5
2.1.3 Resilient Distributed Dataset	6
2.1.4 Arsitektur Spark	7
2.2 Scala	7
2.2.1 Pemrograman Dasar Scala	8
2.2.2 Operasi Map Reduce dengan Scala	8
2.3 Klasifikasi	8
2.3.1 Teknik Klasifikasi di Spark MLlib	8
A KODE PROGRAM	9
B HASIL EKSPERIMEN	11

DAFTAR GAMBAR

B.1	Hasil 1	11
B.2	Hasil 2	11
B.3	Hasil 3	11
B.4	Hasil 4	11

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Bagian ini akan diisi dengan apa yang melatarbelakangi pembuatan template skripsi ini. Termasuk juga masalah-masalah yang akan dihadapi untuk membuatnya, termasuk kurangnya kemampuan penguasaan \LaTeX sehingga template ini dibuat dengan mengandalkan berbagai contoh yang tersebar di dunia maya, yang digabung-gabung menjadi satu jua. Bagian lain juga akan dilengkapi, untuk sementara diisi dengan lorem ipsum versi bahasa inggris.

Big data selalu datang dan terkumpul secara terus menerus. Mengklasifikasikan data merupakan salah satu proses yang diperlukan untuk mendapatkan informasi dan prediksi dari data-data yang ada. Model klasifikasi data yang dilakukan harus diperbarui seiring dengan bertambahnya data agar tetap mendapatkan informasi dan prediksi yang diinginkan.

Untuk mendapatkan hasil prediksi dengan akurasi yang tinggi, digunakan teknik yang namanya Ensemble Method. *Ensemble method* merupakan teknik untuk meningkatkan akurasi pada teknik klasifikasi. Pada teknik ini, *dataset training* akan dipecah secara acak, tiap bagian digunakan untuk membuat model klasifikasi. *Dataset training* dipecah menggunakan teknik *bagging*. *Bagging* (*bootstrap aggregating*) merupakan teknik untuk membagi-bagi dataset dengan menerapkan ‘*sampling with replacement*’ yang dapat dimanfaatkan pada *ensemble method*. Pada tahap prediksi, kasus baru akan diumpankan ke semua model. Hasil prediksi kelas akan ditentukan dari kelas dengan jumlah voting terbanyak.

Perangkat lunak akan dibuat dengan bantuan *Apache Spark*. *Apache Spark* adalah teknologi komputasi *cluster* yang dirancang untuk komputasi cepat yang berdasar pada *Hadoop MapReduce* dan model *MapReduce*. Library klasifikasi paralel untuk klasifikasi *big data* pada sistem tersebar *Spark* sudah tersedia pada Spark Machine Learning Library (Spark MLlib). Perlu dikembangkan agar menangani pembuatan model klasifikasi secara inkremental.

Pada skripsi ini, akan dibuat sebuah program yang mengklasifikasikan big data secara inkremental dengan menggunakan *ensemble method classifier* dengan mengadopsi *bagging*. Inkremental maksudnya adalah model klasifikasi data harus diperbarui seiring dengan bertambahnya data agar model tersebut dapat tetap valid, merepresentasikan *big data* baru dan memiliki akurasi yang diharapkan.

1.2 Rumusan Masalah

Bagian ini akan diisi dengan penajaman dari masalah-masalah yang sudah diidentifikasi di bagian sebelumnya. Masalah-masalah yang ingin diselesaikan dalam skripsi ini adalah sebagai berikut.

1. Bagaimana cara mengimplementasikan *ensemble method classifier* dengan *bagging* dan voting ke dalam algoritma klasifikasi?
2. Apa saja algoritma klasifikasi yang ada pada *Spark*?
3. Bagaimana cara algoritma klasifikasi yang ada pada Spark bekerja?

4. Bagaimana *ensemble method* bekerja?
5. Bagaimana teknik *bagging* bekerja?
6. Bagaimana melakukan algoritma klasifikasi pada spark dengan bahasa scala?

1.3 Tujuan

Akan dipaparkan secara lebih terperinci dan terstruktur apa yang menjadi tujuan pembuatan template skripsi ini Tujuan dari pembuatan perangkat lunak adalah sebagai berikut.

1. Mengembangkan algoritma klasifikasi pada *Spark* menjadi *ensemble method classifier* dengan *bagging*.
2. Mempelajari jenis-jenis algoritma klasifikasi pada *Spark*.
3. Mempelajari bagaimana Spark bekerja.
4. Mempelajari *ensemble method*.
5. Mempelajari teknik *bagging*.
6. Mempelajari bahasa Scala.

1.4 Batasan Masalah

Untuk mempermudah pembuatan template ini, tentu ada hal-hal yang harus dibatasi, misalnya saja bahwa template ini bukan berupa style L^AT_EX pada umumnya (dengan alasannya karena belum mampu jika diminta membuat seperti itu)

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

1.5 Metodologi

Tentunya akan diisi dengan metodologi yang serius sehingga templatennya terkesan lebih serius.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

1.6 Sistematika Pembahasan

Rencananya Bab 2 akan berisi petunjuk penggunaan template dan dasar-dasar \LaTeX . Mungkin bab 3,4,5 dapt diisi oleh ketiga jurusan, misalnya peraturan dasar skripsi atau pedoman penulisan, tentu jika berkenan. Bab 6 akan diisi dengan kesimpulan, bahwa membuat template ini ternyata sungguh menghabiskan banyak waktu.

Bab 2 akan menjelaskan tentang Big Data, Spark, algoritma klasifikasi yang ada pada Spark MLLib, bahasa pemrograman Scala. Bab 3 akan berisi hasil dari eksplorasi di bab 2. Bab 4 akan berisi perancangan model algoritma klasifikasi yang menggunakan ensemble method.

BAB 2

LANDASAN TEORI

2.1 Apache Spark

Rencananya akan diisi dengan penjelasan umum mengenai buku skripsi.

Apache spark adalah komputasi cluster cepat yang dirancang untuk komputasi cepat. Apache spark dibangun di atas Hadoop MapReduce dan meningkatkan model MapReduce agar dapat menggunakan lebih banyak jenis perhitungan secara efisien. Fitur utama spark adalah komputasi cluster di dalam memori yang meningkatkan kecepatan pemrosesan aplikasi. Spark dirancang untuk mencakup berbagai data yang besar dan berat seperti aplikasi batch, algoritma iteratif, query yang interaktif dan streaming.

2.1.1 Fitur-fitur Apache Spark

Apache Spark digunakan dikarenakan banyaknya kelebihan-kelebihan yang dimilikinya. Kelebihan tersebut antara lain.

1. Kecepatan
Spark membantu menjalankan aplikasi di cluster Hadoop, hingga 100 kali lebih cepat dalam memori, dan 10 kali lebih cepat saat dijalankan pada disk. Hal tersebut dapat terjadi dengan cara mengurangi jumlah operasi baca / tulis ke disk dan cara mengurangnya adalah dengan menyimpan pemrosesan data di dalam memori.
2. Kemampuan cache yang kuat.
3. Dapat dipasang di berbagai platform misal Mesos, Hadoop via Yarn atau cluster manager milik spark sendiri.
4. Menyediakan komputasi real-time.
5. Mendukung banyak bahasa (Java, Scala, dan Python).
6. Spark dapat menangani penambahan beban misalnya penambahan volume data atau jumlah pengguna tanpa penurunan kinerja yang berarti.

2.1.2 Komponen-komponen Spark

Komponen-komponen yang ada pada spark antara lain.

1. Apache Spark Core
Spark Core adalah komponen yang mendasari semua platform spark dengan semua fungsinya. Menyediakan komputasi dalam memory dan mereferensikan dataset-dataset di dalam sistem penyimpanan eksternal.

2. Spark SQL

Spark SQL adalah komponen diatas SparkCore yang menyediakan abstraksi data yang disebut SchemaRDD. SchemaRDD menyediakan bantuan untuk kueri-kueri data terstruktur dan data semi-terstruktur.

3. Spark Streaming

Spark Streaming memanfaatkan kemampuan penjadwalan cepat dari Spark Core untuk melakukan analisis streaming. Analisis streaming mengambil data dalam batch-batch kecil dan menjalankan transformasi RDD (Resilient Distributed Datasets) pada data di dalam batch-batch kecil tersebut.

4. MLlib(Machine Learning Library)

MLlib adalah framework machine learning terdistribusi diatas Spark dikarenakan arsitektur Spark yang berbasis memori terdistribusi. Spark MLlib menyediakan algoritma-algoritma yang dapat digunakan untuk analisis data seperti regresi, klasifikasi, dan lain-lain.

5. GraphX

GraphX adalah framework untuk pemrosesan grafik terdistribusi di atas spark. GraphX menyediakan API untuk membuat perhitungan grafik yang dapat memodelkan grafik yang ditentukan user.

6. SparkR

SparkR adalah package milik R yang menyediakan implementasi data frame terdistribusi. SparkR juga mendukung operasi seperti seleksi, filter, agregasi untuk dataset dataset yang ukurannya besar.

2.1.3 Resilient Distributed Dataset

RDD adalah bagian inti dari semua aplikasi Spark. Secara umum, RDD bersifat immutable. Immutable artinya objek tersebut tidak dapat diubah setelah dibuat, akan tetapi dapat ditransformasi. RDD artinya adalah

- Resilient : Toleran terhadap kesalahan dan mampu mengembalikan data yang gagal
- Distributed : Data terdistribusi ke semua node-node dalam sebuah cluster
- Dataset : Kumpulan partisi data dengan value

Jadi, dapat disimpulkan bahwa RDD adalah dataset yang tersebar ke semua node di dalam sebuah cluster dan data tersebut tahan/toleran terhadap kesalahan serta dapat mengembalikan data yang gagal.

Cara kerja RDD adalah data di dalam RDD dibagi menjadi beberapa bagian berdasarkan suatu key. RDD sangat resilient (highly resilient) karena RDD dapat pulih dengan cepat dari kesalahan-kesalahan. Hal tersebut dapat terjadi karena potongan data yang sama direplikasi di beberapa node pelaksana (executor nodes). Jadi, jika satu node pelaksana gagal maka yang lain masih akan memproses data. Oleh karena itu, RDD memungkinkan untuk melakukan perhitungan fungsi terhadap dataset dengan sangat cepat dengan memanfaatkan beberapa node.

Setiap dataset di RDD dibagi menjadi partisi yang logis, yang artinya dapat di komputasi/ dijalankan di node yang berbeda dalam sebuah cluster. Karena sifatnya ini, kita dapat melakukan transformasi atau aksi ke seluruh data secara paralel. Distribusinya dilakukan secara otomatis oleh Spark.

Ada 2 cara membuat RDD, cara pertama adalah dengan memparalelkan koleksi data yang ada di dalam driver program dan cara yang kedua adalah dengan mereferensi dataset di luar storage system seperti shared file system, HDFS, HBase, dan lain-lain.

RDD bisa melakukan 2 operasi sebagai berikut:

- Transformations (transformasi) : operasi yang dipakai untuk membuat RDD baru.
- Actions (aksi) : aksi dipakai di dalam RDD untuk memberitahukan Apache Spark untuk menggunakan komputasi dan memberikan hasilnya kembali ke driver.

Transformasi RDD

RDD dapat melakukan berbagai macam operasi transformasi, berikut adalah operasi-operasi transformasi yang dapat dilakukan oleh RDD.

Aksi RDD

RDD juga dapat melakukan berbagai operasi aksi. Berikut adalah operasi-operasi aksi yang dapat dilakukan oleh RDD.

2.1.4 Arsitektur Spark

Dalam spark, terdapat yang namanya master node dan juga worker node. Master node bertugas untuk manajemen pembagian RDD dan juga pembagian tugas-tugas ke semua worker node yang ada. Master node memiliki sebuah driver program. Tugas dari driver program ini adalah untuk menjalankan aplikasi yang telah dibuat. Kode yang dibuat bertindak sebagai driver program atau jika menggunakan interactive shell, shell yang bertindak sebagai driver program.

Di dalam driver program, hal pertama yang harus dilakukan adalah membuat Spark Context. Spark Context merupakan gerbang ke semua fungsi-fungsi milik spark. Semua fungsi yang akan dilakukan pada driver program akan melewati Spark Context. Driver program dan spark context bertugas menangani eksekusi pekerjaan di dalam sebuah cluster. Sebuah pekerjaan dibagi-bagi menjadi beberapa tugas-tugas kecil yang nantinya didistribusikan ke worker node. RDD juga dibuat di dalam spark context dan juga didistribusikan ke berbagai worker node. Kemudian RDD akan dicache di dalam worker node tersebut.

Worker node adalah node pekerja yang pekerjaannya adalah pada dasarnya mengerjakan tugas-tugas yang diberikan kepadanya. Tugas-tugas tersebut dieksekusi di dalam node tersebut. Tugas tersebut adalah tugas di dalam RDD yang sudah dipartisi. Kemudian, worker node akan mengembalikan hasilnya kembali ke Spark Context.

Jadi secara garis besar, Spark Context mengambil pekerjaan, memecah pekerjaan dalam tugas-tugas/tasks dan mendistribusikannya ke worker node. Tugas-tugas tersebut bekerja pada RDD yang dipartisi, melakukan operasi, mengumpulkan hasil dan mengembalikannya ke Spark Context. Jika node worker ditambah, maka pekerjaan akan menjadi lebih cepat selesai. Hal ini terjadi karena pekerjaan dapat dipecah ke lebih banyak partisi dan dapat mengeksekusi pekerjaan tersebut secara paralel dalam banyak sistem yang berbeda. Besar memori juga akan bertambah yang berefek pada meningkatnya kekuatan men-cache pekerjaan. Oleh karena itu, pekerjaan dapat dieksekusi dengan lebih cepat.

2.2 Scala

Mengapa menggunakan \LaTeX untuk buku skripsi dan apa keunggulan/kerugiannya bagi mahasiswa dan pembuat template.

Scala adalah bahasa pemrograman yang berbasis Java Development Kit. Bahasa Scala menyediakan optimasi untuk kompleksitas kode dan concise notation (notasi singkat). Concise Notation artinya memiliki banyak informasi dalam simbol atau tulisan yang sangat sedikit. Bahasa Scala juga kompatibel dengan bahasa Java. Oleh karena itu, Scala juga mendukung pemrograman berbasis objek (object-oriented) yang hasilnya adalah kode yang pendek namun berbobot. Selain itu, kompatibel dengan Java juga memungkinkan Scala untuk menggunakan keuntungan dari Java Virtual Machine (JVM) dan juga menggunakan library dari Java.

2.2.1 Pemrograman Dasar Scala

2.2.2 Operasi Map Reduce dengan Scala

2.3 Klasifikasi

Di dalam KBBI, klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Klasifikasi dilakukan untuk memprediksi suatu objek dengan atribut-atribut tertentu berada di dalam kelompok kelas yang mana. Cara klasifikasi ada bermacam-macam dan dengan akurasi yang bermacam-macam pula. Salah satu metode klasifikasi yang ada di machine learning adalah ensemble method.

Ensemble Method atau metode ensemble adalah algoritma dalam machine learning dimana algoritma ini adalah algoritma pencarian solusi prediksi terbaik dibandingkan dengan algoritma yang lain karena metode ensemble ini menggunakan beberapa algoritma pembelajaran untuk pencapaian solusi prediksi yang lebih baik daripada algoritma yang bisa diperoleh dari salah satu pembelajaran algoritma saja. Cara menerapkan algoritma ini ada bermacam-macam dan salah satu yang paling sederhana adalah dengan cara bootstrap aggregating atau disingkat bagging.

Bootstrap Aggregating atau bagging merupakan metode yang dapat memperbaiki hasil dari algoritma klasifikasi machine learning dengan menggabungkan klasifikasi prediksi dari beberapa model. Hal ini digunakan untuk mengatasi ketidakstabilan pada model yang kompleks dengan kumpulan data yang relatif kecil. Bagging adalah salah satu algoritma berbasis ensemble yang paling awal dan paling sederhana, namun efektif. Bagging paling cocok untuk masalah dengan dataset pelatihan yang relatif kecil. Bagging mempunyai variasi yang disebut Pasting Small Votes. cara ini dirancang untuk masalah dengan dataset pelatihan yang besar, mengikuti pendekatan yang serupa, tetapi membagi dataset besar menjadi segmen yang lebih kecil.

2.3.1 Teknik Klasifikasi di Spark MLlib

Akan dipaparkan bagaimana menggunakan template ini, termasuk petunjuk singkat membuat referensi, gambar dan tabel. Juga hal-hal lain yang belum terpikir sampai saat ini.

Logistic Regression

Decision Tree

Naive Bayes

Random Forest

Gradient Boosted Tree

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Listing A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```


LAMPIRAN B

HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4