DAFTAR PUSTAKA

Apache Kafka, 2016. *Apache Kafka.* [Online] Available at: https://kafka.apache.org [Diakses 29 Juni 2017].

Apache Spark, 2012. *Streaming Programming Guide.* [Online] Available at: http://spark.apache.org/docs/2.0.1/streaming-programming-guide.html#setting-the-right-batch-interval [Diakses 09 08 2017].

Baskoro, D. O., 2015. *Big Data Benchmark Pada Hadoop 2, Spark dan Presto,* Yogyakarta: Universitas Gadjah Mada.

Carina, J., 2017. *Penumpang Bus Transjakarta Naik 20 Persen Tahun 2016.* [Online] Available at: http://megapolitan.kompas.com/read/2017/01/25/14165351/penumpang.bus.transjakarta.naik.20.persen.tahun.2016 [Diakses 08 07 2017].

Cui, A., 2006. *Bus Passenger Origin-Destination Matrix Estimation Using Automated Data Collection Systems,* USA: Massachusetts Institute of Technology.

Goa, L.-X., Hu, J.-H., Li, G.-Y. & Liang, J.-X., 2015. *A calculation method of OD matrix in multi-modal transit network based on traffic big data.* Wuhan, 2015 International Conference on Transportation Information and Safety (ICTIS).

He, L., Nassir, N., Trepanier, M. & Hickman, M., 2015. *Validating and calibrating a destination estimation algorithm for public transport smart card fare collection systems,* Montreal, QC, Canada: Centre Interuniversitaire de Recherche sur les Reseaux d'Entreprise, la Logistique et le Transport (CIRRELT).

James, M., 2011. *Big data: the next frontier for innovation, competition, and productivity,* s.l.: The McKinsey Global Institute.

Kreps, J., Narkhede, N. & Rao, J., 2011. *Kafka: A distributed messaging system for log processing.* s.l., s.n., pp. 1-7.

Lakshman, A. & Malik, P., 2014. *Apache cassandra.* [Online] Available at: http://planetcassandra.org/what-is-apache-cassandra [Diakses 23 06 2017].

MadhaviLatha, A. & Kumar, G. V., 2016. Streaming Data Analysis using Apache Cassandra and Zeppelin. *International Journal of Innovative Science, Engineering & Technology,* 3(10).

MapRCommunity, 2017. *Real-Time Streaming Data Pipelines With Apache Apis: Kafka, Spark Streaming, And Hbase.* [Online] Available at: https://community.mapr.com/community/exchange/blog/2017/04/26/real-time-streaming-data-pipelines-with-apache-apis-kafka-spark-streaming-and-hbase [Diakses 18 07 2017].

Mzee, Kazimil, P. & Chen, Y., 2010. *Implementation of Bus Rapid Transit System as an Alternative for Public Transportation in Developing Countries Case of Dart System in Dar Es Salaam.* Changsha, China, Intelligent Computation Technology and Automation (ICICTA).

Nasiboglu, E., Kuvvetli, U., Ozkilcik, M. & Eliiyi, U., 2012. *Origin-Destination Matrix Generation Using Smart Card Data: Case Study for Izmir.* Baku, 2012 IV International Conference "Problems of Cybernetics and Informatics.

Okuhara, K. et al., 2008. Evaluation and Assignment of Traffic Volume for Urban Planning Basen on Planner and User Stances. *International Journal of Innovative Computing Information and Control,* Volume 4, pp. 1151-1160.

Riasetiawan, M., 2016. *Pusat Data untuk Pemerintahan.* Yogyakarta: Departemen Ilmu Komputer dan Elektronika FMIPA UGM.

Tang, Y. W., 2004. *Application of Conditional Demand Analysis in Origin-Destination (OD) Matrix Estimation Using Traffic Counts and Zonal Characteristics*, Hong Kong: City University of Hong Kong.

Tutorials Point (I) Pvt. Ltd, 2015. *Apache Spark Tutorial.* [Online] Available at: https://www.tutorialspoint.com/apache_spark/apache_spark_tutorial.pdf [Diakses 3 7 2017].

Tutorials Point (I) Pvt. Ltd, 2015. *Tutorials Point - Apache Cassandra.* [Online] Available at: https://www.tutorialspoint.com/cassandra/cassandra_tutorial.pdf [Diakses 29 06 2017].

Tutorials Point (I) Pvt. Ltd, 2016. *Apache Kafka.* [Online] Available at: https://www.tutorialspoint.com/apache_kafka/apache_kafka_tutorial.pdf [Diakses 29 06 2017].

Tutorials Point (I) Pvt. Ltd, 2017. *Java - Regular Expressions.* [Online] Available at: https://www.tutorialspoint.com/java/java_pdf_version.htm [Diakses 23 Juni 2017].

Wang, W., Attanucci, J. P. & Wilson, N. H., 2011. Bus passenger origin-destination estimation and related analyses using automated data collection systems. *Journal of Public Transportation,* 14(4), p. 7.

Wan, T. Y., 2004. *Application of Conditional Demand Analysis in Origin-Destination (OD) Matrix Estimation Using Traffic Counts and Zonal Characteristics,* Hong Kong: City University of Hong Kong.

Wright, L. & Hook, W., 2007. *Wright, L., Hook, W., 2007, Bus Rapid Transit Planning Guide, Institute for Transportation & Development Policy, New York, USA.*. [Online] Available at: https://www.itdp.org/wp-content/uploads/2014/07/52.-Bus-Rapid-Transit-Guide-PartIntro-2007-09.pdf

[Diakses 3 7 2017].

Zaharia, M. et al., 2012. *Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing.* s.l., USENIX Association, p. 2.

Zhang, Y., 2010. *Dynamic Public Transit Origin-Destination Estimation Based on Data Mining*, Beijing, China: Beijing Jiaotong University.

LAMPIRAN A – *Source code* Apache Spark

LAMPIRAN A – Source code Spark

```
package net.bagasp.sparkproject

import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.spark.sql._
import org.apache.log4j._
import org.apache.spark.sql.functions._
import com.datastax.spark.connector._

import need._

object joinTable {
  def main(args: Array[String]) {
      // Use new SparkSession interface in Spark 2.0
      val spark = SparkSession
        .builder
        .appName("joinTable500k")
        .master("local[*]")
        .config("spark.sql.warehouse.dir", "file:///C:/temp") // Necessary to work around a
Windows bug in Spark 2.0.0; omit if you're not on Windows.
        .config("spark.sql.streaming.checkpointLocation", "file:///C:/checkpoint")
        .config("spark.cassandra.connection.host", "127.0.0.1")
        .getOrCreate()


  val gatehalte_id = spark.
                read.
                format("org.apache.spark.sql.cassandra").
                options(Map("table" -> "gatehalte_id", "keyspace" -> "testing")).
                load()

  val brt500k = spark.
          read.
          format("org.apache.spark.sql.cassandra").
          options(Map("table" -> "brt500k", "keyspace" -> "testing")).
          load()
  val brt1000k = spark.
          read.
          format("org.apache.spark.sql.cassandra").
          options(Map("table" -> "brt1000k", "keyspace" -> "testing")).
          load()
  val brt1500k = spark.
          read.
          format("org.apache.spark.sql.cassandra").
          options(Map("table" -> "brt1500k", "keyspace" -> "testing")).
```