Pembelajaran Big Data

Presentation · March 2018

DOI: 10.13140/RG.2.2.35146.62405

1 author:

Budi Susanto
Universitas Kristen Duta Wacana
36 PUBLICATIONS **31** CITATIONS

SEE PROFILE

Pembelajaran Big Data

Budi Susanto

***Abstraksi*—Pengenalan terhadap Big Data masih terus perlu dilakukan dan bahkan harus masuk ke pembelajaran di pendidikan tinggi. Deskripsi yang diuraikan lebih mengarah pada sedikit pengenalan terhadap terminologi *Big Data*, arsitektur umumnya, dan gambaran sebaran teknologi yang ada di pasaran untuk mendukung pengelolaan dan analisis data dalam ukuran (*volume*), percepatan (*veolocity*), dan keragaman (*variety*). Untuk memberikan gambaran tentang bagaimana arsitektur Hadoop dasar dapat digunakan pada satu mesin untuk mensimulasikan *name node*, YARN, dan *data node* menjadi salah satu contoh pembelajaran tentang *Big Data*. Berkaitan dengan pembelajaran *Big Data* disampaikan usulan sederhana tentang laboratorium dan materi matakuliah yang dapat dibangun untuk mahasiswa.**

***Index Terms*—Big Data applications**

I. Pendahuluan

Pertumbuhan data dengan berbagai macam tipenya, baik terstruktur, semi terstruktur, maupun tidak terstruktur terus menunjukkan model eksponensial. IDC [1] memperkirakan pada tahun 2025 pertumbuhan data secara global mencapai 163 zettabyte (triliun gigabyte). Marr, B. [2] pernah melaporkan untuk Forbes tentang beberapa fakta terkait kondisi percepatan pertumbuhan data yang sangat cepat dan menghasilkan volume data yang meledak. Laporan IDC tahun 2014 [3] memperkirakan pertumbuhan data meningkat dua kali setiap dua tahunnya, di mana pada tahun 2020, Internet akan menghubungkan 7.6 miliar pengguna dan 32 miliar '*things*'. Senada dengan berbagai studi tersebut, McKinsey [4] mencatat pertumbuhan trafik IP di Indonesia per bulan sebesar 60% dari 227 petabyte (tahun 2014) menjadi 448 petabyte (tahun 2015).

Volume dan percepatan pertumbuhan data yang demikian besar harus dikelola dengan baik untuk dapat digunakan dan memberikan manfaat. Pengelolaan terhadap volume data yang sangat besar memerlukan dukungan infrastruktur penyimpanan dan komputasi. Di sinilah kemudian muncul istilah *Big Data*. Artikel ini bertujuan dapat memberikan gambaran terkait definisi dan parameter *Big Data*, infrastruktur yang telah banyak diterapkan, dan contoh aplikasi analisis yang dapat dilakukan untuk volume, percepatan, dan juga keragaman tipe data yang tersimpan.

Perkembangan teknologi Big Data dapat dikatakan telah dapat diteirma dan digunakan oleh banyak pihak. Hal ini dikuatkan dari rujukan *Gartner Hype Cycle for Emerging Technologies* tahun 2015 [5] yang tidak lagi memasukan Big Data dalam siklus *Emerging Technology*, semenjak dimunculkan (pada fase *Technology Trigger*) pada tahun 2012. Dalam periode 3 tahun, Big Data dapat dikatakan telah dapat diterima dan diterapkan untuk mendukung industri, oleh karena Big Data telah menjadi bagian dari banyak *hype cycle* lainnya [6]. Berdasar pertimbangan tersebut, maka sudah saatnya pembelajaran terhadap Big Data di level pendidikan tinggi mulai diberi perhatian. Artikel ini mendeskripsikan sebuah contoh simulasi yang dapat dilakukan oleh sivitas akademika untuk mulai belajar terkait Big Data. Pada bagian akhir disampaikan juga kebutuhan analisis data yang juga menjadi trend teknologi seperti yang disebutkan oleh *Gartner Hype Cycle* tahun 2017 [7].

Artikel ini diseminarkan pada acara Seminar BIG DATA FIT Competition UKSW 2018 tanggal 15 Maret 2018 di Universitas Kristen Satya Wacana, Salatiga.

Budi Susanto, Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta (budsus@ti.ukdw.ac.id).

II. Pengertian *Big Data*

Definisi terminologi *Big Data* secara konsep dapat dikatakan belum memiliki acuan ilmiah yang baku. Diebold [8] menegaskan bahwa apa yang disebut Big Data sekarang berbeda dengan definisi *Big Data* pada 15 tahun yang lalu. Beberapa rujukan tentang *Big data* pada awal atau sebelum tahun 2000 sangatlah menarik namun belum menyakinkan. Kesadaran akan kebutuhan pengelolaan dan analisis data berukuran besar telah merebak di lingkungan Silicon Graphics (SGI) pada pertengahan tahun 1990. Press [9] bahkan menyebutkan jauh sebelum itu, kesadaraan tersebut sudah ada. Gandomi dan Haider [10] yang juga merujuk ke [8], menambahkan bahwa terminologi ini mulai banyak diperbincangan pada naskah publikasi ilmiah pada tahun 2012 dan 2013 sebanyak 2-3 kali lipat dari jumlah publikasi tahun 2011 (berdasar dokumen yang mengandung *big data* dalam *ProQuest Research Library*).

Pada umumnya, penggunaan parameter ukuran data lebih umum digunakan untuk menjawab apakah *big data* atau tidak. Parameter ukuran (*volume*) data bukanlah satu-satunya acuan. [10] merujuk [11] pada penggunaan parameter: *Three-V*, yaitu *Volume*, *Variety*, dan *Velocity* (Gartner menyebutnya sebagai bagian – *Three-Parts* [12]). Dari sekian banyak definisi terkait *Big Data* [13], beberapa di antaranya menggunakan *Three-V* sebagai penekannya.

Volume merujuk pada ukuran besarnya data yang dikelola (dalam satuan MB, GB, TB, PB, ZB). Batasan ukuran yang dapat disebut *Big Data* masih dapat dikatakan beragam. [14] dan [15] menyatakan ukuran minimum sebuah sistem Big Data adalah dari Terabytes (TB) sampai Petabyte (PB). *Variety* merujuk pada tingkat keragaman struktur dalam

dataset, dapat berupa terstruktur (contoh tabel), semi terstruktur (dokumen XML), dan tidak terstruktur (dokumen, email, text message, audio, video, gambar, graphic, dan lainnya). *Velocity* merujuk pada cara dan tingkat kecepatan penerimaan data, termasuk apakah melalui proses *batch*, *near time*, *real time*, dan *stream*. Terhadap ketiga dimensi tersebut, mengadopsi dari [14], pada Tabel I memberikan ilustrasi perbedaan antara data tradisional dengan *big data*. Gambaran fakta terkait ketiga dimensi tersebut, antara lain: pada tahun 2012 Facebook dilaporkan mengelola: 2.5 Miliar item konten yang dibagikan, 2.7 Miliar *Likes*, 300 juta foto di upload, 500+ TB data diproses, 70 ribu eksekusi *query*, 100+ PB ruang penyimpanan dalam sebuah HDFS *cluster*, dan 105 TB data tertelusuri melalui Hive [16]. Data serupa juga ditunjukkan oleh Twitter tahun 2010 [17]: 1+ miliar tweet dalam seminggu, rata-rata 50 juta tweet terkirim per hari.

TABLE I
PERBANDINGAN TRADISIONAL DAN *BIG DATA*

	Tradisional Data	*Big Data*
Volume	GB	TB dan PB
Laju Data	Per jam, per hari	lebih cepat
Struktur Data	Terstruktur	semi atau tidak terstruktur
Sumber Data	Terpusat	Tersebar
Intergrasi Data	Mudah	Sulit
Penyimpan Data	RDBMS	HDFS, NoSQL
Akses Data	Interaktif	Batch atau near real time

Sumber *big data* dapat dikelompokkan dalam: 1) bermacam transaksi, 2) data enterprise, 3) data publik, 4) media sosia, dan 5) data sensor.

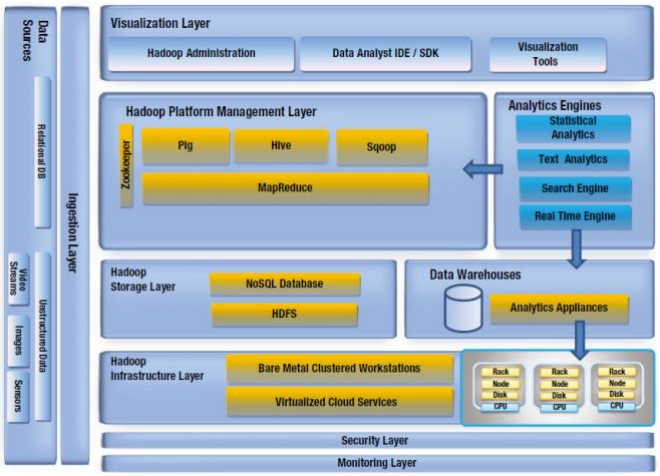
Selain ketiga dimensi tersebut, beberapa pemain industri menyebutkan dimensi lainnya, yaitu: *Veracity* (ketelitian) disebutkan oleh IBM dan lebih merujuk terhadap kualitas data [18]; *Variability* (dan *complexity*) dinyatakan oleh SAS [19] untuk menunjukkan variasi kecepatan aliran data yang berasal dari beberapa sumber; dan *Value* dinyatakan oleh Oracle [20] untuk menekankan bahwa data mentah dari sumber lebih cenderung tidak berkualitas (*low value density*) sehingga perlu dianalisis untuk mendapatkan *high value*.

III. PLATFORM *BIG DATA*

Untuk lebih memudahkan dalam melihat arsitektur sebuah sistem *big data*, ada baiknya mengenal 3 lapisan umum seperti yang ditunjukkan oleh Bigdatalandscape.com, yaitu 1) Aplikasi, 2) Infrastruktur, dan 3) Teknologi. Pada setiap lapis dapat terlibat berbagai produk sistem *big data* yang ada di pasaran saat ini. Lapisan Aplikasi dan Infrastruktur didasari dengan teknologi utama, yaitu Hadoop.

Dengan adanya teknologi Hadoop dan komponen lainnya, telah mendorong berbagai organisasi untuk dapat melakukan analisis terhadap data mereka secara optimal. Hadoop telah menggeser tidak lagi terlalu fokus pada teknologi (tahun 1990-sebelum 2010), namun fokus untuk menghasilkan dan memanfaatkan informasi, bahkan sebagai sebuah komoditas [21, p. 8].

Arsitektur manajemen *big data* harus mampu menerima beragam sumber data dengan cepat dan tidak mahal. Gambar 1 mengilustrasikan outline arsitektur komponen yang seharusnya menjadi bagian dari *big data tech stack*. Dari arsitektur tersebut, kita dapat memilih menggunakan produk *open source* atau produk berbayar untuk mendapatkan keuntungan fungsional secara optimal [22].

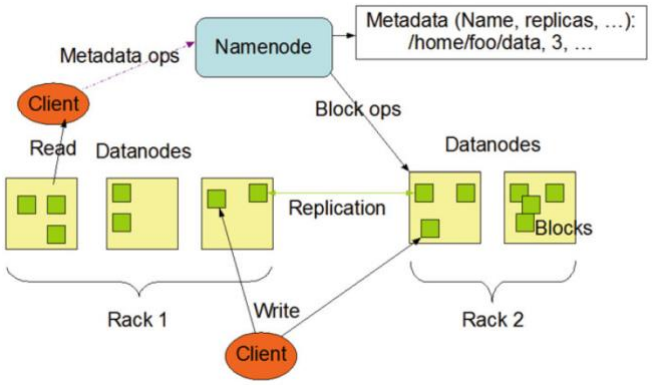


Gambar 1. *Big data Tech Stack* (adopsi dari [22, p. 10])

Arsitektur Hadoop mendukung skalabilitas sistem dalam dua tipe [23]: 1) *Horizontal scaling (scale out)* yang mampu mendistribusikan beban kerja pada beberapa server; dan 2) *Vertical scaling (scale up)* yang meliputi penambahan prosesor, memori, piranti keras yang lebih cepat lainnya pada satu server tunggal. Pemanfaatan GPU (*Graphical Processing Unit*) dapat menjadi salah satu alternatif untuk penambahan prosesor ataupun memori.

A. *Ekosistem Hadoop*

Pada saat ini Hadoop dapat dikatakan sebagai teknologi potensial yang didukung oleh berbagai *framework* (komponen) yang dikembangkan di bawah *Apache Software Foundation*. Hadoop dikembangkan pertama kali sebagai bagian dalam proyek search engine Apache Nutch yang terinspirasi oleh Google MapReduce dan Google File System. Pada tahun 2015, dibentuklah organisasi nirlaba bernama Open Data Platform inisiative (ODPi - https://www.odpi.org) yang menaungi standarisasi keseluruhan ekosistem dari Hadoop.



Gambar 2. Ilustrasi Arsitektur HDFS (adopsi dari [24, p. 54])

Beberapa komponen inti dari ekonsisi Hadoop, antara lain [24]:

1) Komponen Distributed File System (DFS)

Komponen DFS memberikan layanan penting untuk ketersediaan data dan perlindungan terhadap kehilangan data jika beberapa nodes cluster *down*. Dalam ekosistem Hadoop digunakan HDFS yang merupakan modul dari proyek Apache Hadoop. Gambar 2 memberi ilustrasi terkait arsitektur HDFS.

Proses *Name Node* (yang dipasang di *Management Node*) memelihara meta data yang disimpan dalam setiap *Data Node* (termasuk dalam lapisan *Namespace HDFS*). *Data Node* (sebagai bagian dari lapisan *Block* Storage) bertanggung jawab untuk melayani setiap permintaan *read* dan *write* dari aplikasi *client*. Proses *Data Node* melaporkan semua blok datanya (ukuran per data blok 128 MB) ke proses *Name Node* selama *boot*. *Name Node* menggunakan version number sebuah blok data untuk mengetahui apakah blok data tidak digunakan. Selama sedang menuliskan sebuah berkas, *client* API menghubungi proses *Name Node* untuk mendapatkan daftar Data Node yang berisi replikas primer dan sekunder. Saat *client* melakukan *push* perubahan ke semua proses *Data Node*, perubahan tersebut disimpan dalam buffer di setiap proses *Data Node*. Pada saat *client* mengirimkan commit ke *Name Node* primer, data terkomit di primer dan replika lainnya. Semua perubahan ditulis ke berkas log operasi yang tersimpan di *Management Node* yang memelihara daftar urutan operasi.

HDFS mendukung partisi vertikal dari data dengan bantuan layanan *HDFS Federation* [25]. HDFS juga mendukung sistem berkas POSIX. Konfigurasi untuk Name Node harus memiliki RAM besar, karena Name Node lebih banyak menggunakan RAM untuk menyimpan peta data blok dari semua *Data Node*.

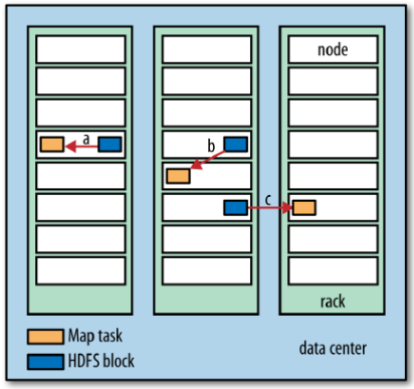
Selain HDFS, terdapat komponen DFS lain yang tersedia di pasar industri *Big Data*, antara lain: MapR-FS, IBM Spectrum Scale FPO, Intel Lusture, EMCS Isilon, Tachyon, Cassandra DataStack, Ceph, NetApp Open Solution [26].

2) Komponen Distributed Processing

Komponen ini digunakan oleh komponen aplikasi. Komponen pertama yang dikembangkan adalah *Hadoop MapReduce*. *MapReduce* adalah sebuah model pemrograman untuk pemrosesan data. Dalam *MapReduce*, task dibagi ke dalam dua fase, yaitu: *Map* dan *Reduce*. Fase *Map* membagi *task* yang besar ke dalam beberapa *sub task* dan menjalankan setiap *sub task* ke *node* yang aktif dalam *cluster*. Fase *Reduce* mengumpulkan hasil dari fase *Map* dan memproses hasil untuk mendapatkan hasil akhirnya. Sebuah *Hadoop job* pada umumnya melibatkan beberapa *mapper* dan *reducers* yang berjalan di beberapa *node* yang berbeda dalam *cluster*. *Task-task* dijadwalkan menggunakan YARN (*Yet Another Resource Negotiator*). Dalam membagi *task* yang dijalankan oleh *node* dalam *cluster*, Hadoop melakukan pembagian dengan memperhatikan lokasi data (disebut *data locality optimization*). Ada tiga tipe *data locality optimization* [27, p. 32]: Data-local, rack-local, dan off-rack (Lihat Gambar 3).

Berdasar *framework MapReduce* banyak dikembangkan *wrapper* yang dapat menyediakan kontrol lebih baik terhadap *MapReduce*. Contohnya: Apache Pig, Hive, DryadLINQ, Mahout [23].

[24] menyebutkan beberapa tipe komponen *Distributed Processing*, yaitu: 1) komponen hanya dapat bekerja di ekosistem Hadoop, contohnya: MapReduce dan Tez; 2) seperti tipe pertama tapi juga digunakan olehkomponen aplikasi tunggal; 3) komponen dikembangkan untuk dapat menjalankan beberapa teknologi *Big Data*, komponen aplikasi, dan dipaket dengan komponen Hadoop lain (contoh: Spark dan Flink); dan 4) komponen mendukung multi teknologi *Big Data*, mendukung hanya satu tipe komponen aplikasi dan umumnya dipaket degan komponen lain dari ekosistem Hadoop, contohnya: Apache Drill, Apache Storm, Apache Solr, dan lainnya.



Gambar 3. Ilustrasi Arsitektur HDFS (adopsi dari [27, p. 32])

Komponen distributed processing lainnya adalah Apache Tez, yang fokus pada penerapan tipe data *directed-acyclic-graph* (DAG). Apache Tez dapat digunakan komponen aplikasi yang memproses query dalam *near real time*. Komponen aplikasi seperti Hive, Pig, dan Cascading menggunakan Tez.

3) Komponen Aplikasi

Komponen aplikasi dapat dikategorikan ke dalam beberapa kelompok [24, p. 57]: SQL, Data Flow, Pemroses Graph, Machine Learning, Streaming, NoSQL, dan Search. Sebuah komponen aplikasi dapat menggunakan satu atau beberapa komponen *distributed processing* (contoh: Hive menggunakan MapReduce, Tez, dan Spark).

a) Komponen SQL

Komponen ini mendukung ANSI SQL untuk query dan proses data. Aplikasi open source yang mendukung SQL, antara lain: Apache Hive, dan Apache Drill.

b) Komponen Data Flow

Data flow memungkinkan pengembang untuk menuliskan sebuah aliran pemroses data yang akan ditransformasikan dan ektraksi. Data flow menggunakan DAG. Contoh: Apache Pig, Cascading.

c) Komponen Pemrosesan Graf

Komponen ini memberikan fungsi penerapan algoritma yang dapat digunakan untuk memroses data Graf. Contoh: Apache Giraph.

d) Komponen Machine Learning

Komponen ini menyediakan berbagai algoritma yang dapat digunakan untuk menghasilkan model prediktif. Contoh: Apache Mahout, RHadoop.

4) Komponen Security

Komponen keamanan harus dapat mencakup kontrol otentikasi dan otorisasi, enkripsi data, audit. Hadoop mendukung Kerberos (berbasis LDAP). Data terenkripsi dapat digunakan secara transparan oleh berbagai aplikasi yang mengakses HDFS melalui protokol: Hadoop File System Java API, Hadoop libhdfs C library, atau WebHDFS REST API. Contoh: Apache Knox (berbasis REST API), Apache Sentry (column level acces control, granular privileges untuk perintah Database), Apache Ranger.

5) Komponen Service Management

Komponen ini bertanggung jawab untuk mengelola dan memelihara layanan-layanan yang dijalankan dalam *cluster Big Data*. Contoh: YARN (seperti sistem operasi pada umumnya yang mendukung berbagai tipe proses *concurrent* dengan mengalokasikan berbagai sesumber komputasi), Apache Mesos, IBM Platform Symphony, Apache ZooKeeper (memastikan *high availability* melalui koordinasi distribusi proses dari layanan-layanan yang berjalan di *Hadoop Cluster*), Apache Slider, Apache Ambari (menyediakan fasilitas *monitoring cluster* berbasis web).

6) Komponen Integrasi Data dan Governance

Komponen ini memberikan fungsi terkait *Meta data management*, data import/export dari dan ke ekosistem Hadoop, *Workflow*, dan Taksonomi. Contoh: Apache HCatalog (metadata), Apache Sqoop (ekspor/impor), Apache Oozie (workflow), Apache Falcon (*feed processing* dan *feed management*), Apache Atlas (*governance* untuk pemrosesan data).

B. Database NoSQL

Salah satu latar belakang pengembangan dan penggunaan teknologi NoSQL adalah mengembangkan database *build for purpose* daripada penggunaan RDBMS untuk segala kebutuhan *enterprise*. Tingginya kebutuhan operasi read/write untuk aplikasi web membutuhkan sebuah teknologi database yang tidak mampu diberikan oleh RDBMS.

[24] menyebutkan 4 karakteristik utama dari database NoSQL: 1) *Logical data Model Layer* mendukung perluasan tipe skema data yang *loosely*; 2) *Data Distribution Layer* memastikan *horizontal scaling*; 3) *Persistence Layer* dengan fleksibilitas penyimpanan data, baik di disk atau memori; dan 4) *Interface Layer* mendukung berbagai antarmuka non SQL, seperti REST, Thrift, API bahasa pemrograman tertentu, untuk akses data tanpa dukungan transaksi.

Produk yang menyediakan teknologi NoSQL, antara lain: MongoDB, Apache Cassandra, Apache HBase, CockroachDB, Foundation DB. Selain itu juga tersedia NoSQL di Cloud, contohnya: Amazon Dynamo DB, IBM Cloudant, ObjectRocket, Google Cloud BigTable.

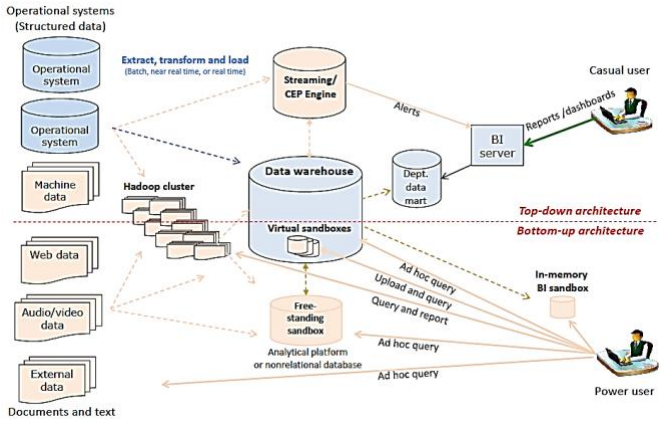
C. Database In-Memory

Sistem *Database In-Memory* sepenuhnya menggunakan *volatile memory* (RAM) utama untuk menyimpan dan memproses data. In-Memory membuat *snapshot savepoint* dari *database in-memory* yang dituliskan ke media *non-volatile* [28]. Dengan *snapshot* ini, maka dapat dilakukan *rollback* atau *rolling forward* untuk data jika terjadi *crash*.

Terdapat beberapa produk *Database In-Memory open source*, yaitu: Aerospike, Hazelcast, Terracota BigMemory, Gem Fire, Apache Ignite. Produk yang berbayar, antara lain: Oracle Coherence, Oracle Timesten, IBM Db2 Blu, SAP Hana, Exasol, dan lain.

D. Sistem Data Warehouse

Sebelum adanya pemanfaatan *Big Data*, *Data Warehouse* lebih banyak menyimpan histori transaksi terstruktur yang dimiliki oleh organisasi. Proses *Extraction, Transformation,* dan *Loading* (ETL) menjadi satu *gateway* yang penting untuk dilakukan agar data transaksi dapat disimpan di *Data Warehouse* sesuai dengan model dimensi yang ditentukan. Setiap dimensi dan fakta dapat diberikan *metadata* yang menyangkut pembangunan *cube*. Dari setiap *cube* yang didefinisikan dapat dianalisis menjadi suatu *reporting* menggunakan bahasa MDX (*MultiDimensional eXpression*).



Gambar 4. Ilustrasi Arsitektur Data Warehouse dan Big Data (adopsi dari [40, p. 16])

Dengan adanya *Big Data*, maka sekarang sumber data yang berasal dari berbagai sumber yang terdefinisikan (termasuk format datanya), tersimpan dalam sebuah *data lake* [29] [30].

Salah satu produk yang dapat digunakan untuk membangun *Data Warehouse* dengan ekosistem *big data*, adalah Apache Hive, Greenplum Database. Produk yang berbayar antara lain Oracle Datawarehouse, Pentaho Business Intelligent (Mondrian, tersedia versi *Community*), atau Apache Spark. Untuk membangun infrastruktur dengan ekosistem *big data*, dapat digunakan Apache BigTop yang telah menyediakan komponen-komponen utama, antara lain [31]: Apache Crunch, Apache Flume, Apache Giraph, Apache HBase, Apache HCatalog, Apache Hive, Apache Mahout, Apache Oozie, Apache Pig, Apache Solr, Apache Sqoop, Apache Whirr, Apache Zookeeper, Cloudera Hue, dan LinkedIn DataFu. Gambar 4 memberikan ilustrasi arsitektur *Data Warehouse* dan *Big Data* dalam suatu ekosistem.

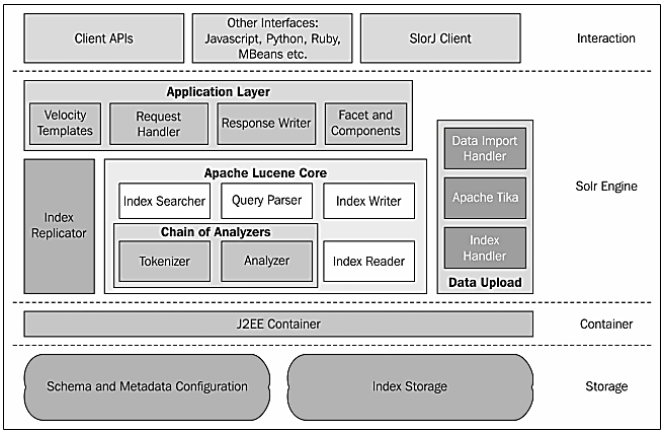
E. Pemrosesan Streaming Event

Teknologi Pengolahan *Streaming Event* adalah *platform* yang dapat mendukung pemrosesan volume yang sangat besar dari *streaming event* (atau data) yang diterima secara terus menerus dalam satuan sub-detik sampai beberapa detik. Teknologi pemrosesan ini biasanya menggunakan konsep *Producer-Consumer Agent Network*. Setiap agen *producer/consumer* menangkap data atau *stream event*, menerapkan logika pemrosesan tertentu, dan mengirimkan ke agen berikutnya.

Agen-agen biasanya proses yang berjalan dalam waktu yang lama dan dapat dikelola menggunakan manajemen *resource*, seperti YARN dan Mesos. *High Availability* dikelola menggunakan ZooKeeper. Teknologi seperti Spark atau Flink menyediakan mekanisme pemrosesan *Stream Event*. Beberapa aplikasi *open source* untuk pemrosesan *stream event*, antara lain: Flume, Storm, Kafka.

F. Search Engine

Untuk menjawab kebutuhan pengelolahan data, khususnya dokumen, yang dapat dilakukan pencarian berbasis kata kunci (*keywords*), penggunaan aplikasi seperti Apache Solr, Elasticsearch, Sphinx Search. Gambar 5 memberikan ilustrasi arsitektur Apache Solr.



Gambar 5. Ilustrasi Arsitektur Apache Solr (adopsi dari [41, p. 29])

G. Data Analytics Stack

Berkeley Data Analytics Stack (BDAS) terdiri dari sehimpunan *Big data framework*. BDAS dapat mengakses dan menyimpan data dari dan ke lokal, *networked*, atau *cloud* (S3, NFS, dan lain), dan dari database lain (Cassdandra, HBase, RDBMS, dan lain). Beberapa produk *open source* yang dapat digunakan untuk menganalisis: Apache Spark (lebih detil [32]), Velox, Apache Flinx, Tachyon, Succinet, Apache Mesos

IV. Analisis Big Data

Big Data tidak lagi menjadi domain dari aktuaris atau ilmuwan. Ekosistem *Big Data* telah memungkinkan pengguna lebih luas, seperti humaniora dan ilmu sosial, pemasaran, organisasi pemerintah, institusi pendidikan, personal yang termotivasi, untuk dapat menghasilkan, berinteraksi, mengorganisasikan data. Namun demikian, untuk dapat menganalisis data secara khusus, tetap dibutuhkan profesi *data scientist*, yang harus memiliki ketrampilan pengolahan data, statistik, dan matematika.

Berkaitan dengan dimensi *velocity*, [33] menguraikan tiga tantangan berkaitan dengan: 1) analisis *digital shadow* yang berhubungan dengan *velocity*, *volume*, dan *value*; 2) *smart city* dan *urban computing*, dan industri 4.0 yang menghubungkan *velocity*, *volume*, dan *veracity*.

Berkaitan dengan *digital shadow*, salah satu pendorong penghasil data adalah pengguna Internet, khususnya media sosial. Tantangannya adalah bagaimana dapat menangkan *stream event* secara lebih *real time* untuk dapat melihat keberlangsungan dan akurasi terhadap evolusi *digital shadow*. Data yang terus mengalir terkait *digital shadow* juga dapat digunakan untuk mengetahui perilaku pengguna media sosial, sekaligus dapat dimanfaatkan oleh pengiklan agar dapat lebih tepat sasaran.

Setiap kota adalah sebuah sistem komplek dengan banyak subsistem di dalamnya untuk membuatnya berjalan, seperti pekerjaan, makanan, pakaian, pendudukan, kantor-kantor, hiburan, transportasi, air, energi, dan lainnya. Pengembangan teknologi digital akan mengubah harapan antara hal tersebut. *Smart City* terkait dengan penginderaan status kota dan bertindak dengan cara baru yang pintar pada tiap level: penduduk, pemerintah, mobil, transportasi, komunikasi, energi, lingkungan, penyimpanan sumber daya, dan lainnya. Dengan *Big Data*, bagaimana dapat mengumpulkan data terkait ekspresi / persepsi penduduk dari setiap area daerah, sehingga dapat dianalisis kaitan antara tempat, pendudukan, dan pemanfaatannya. Hal lain yang dapat dianalisis adalah bagaimana ruang kota, layanan, dan kegiatan/kejadian mempengaruhi emosi penduduk.

Arsitektur *Big Data Analityc* dapat memanfaatkan data terstruktur, tidak terstruktur dan *fast-moving* yang besar. Industri *Simultaneous Localization and Mapping* (SLAM) yang banyak menghasilkan robot, menjadi tantangan untuk dapat memanfaatkan *Big Data* agar dapat diterapkan di robot dapat lebih cerdas dan memberikan layanan yang lebih komplek.

Analisis *Big data* lain yang sudah ada dan masih terus dikembangkan, antara lain [10]: analisis teks (*information extraction, text summarization, question answering (QA), sentimen analysis*); analisis audio (*speech analytics, interactive voice response*); analisis video (*video indexing* dan *retrieval*); analisis media sosial (*content-based analytics, structure-based analytics, community detection, social influence analysis, link prediction*); analisis prediksi.

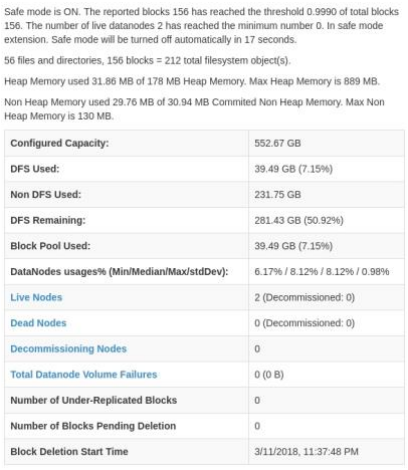
Dari semua komponen machine learning ataupun analytics, komponen visualisasi data juga penting untuk dikembangkan. Ketrampilan untuk memvisualisasi data juga harus dimiliki oleh seorang *data scientist*. Berdasar beberapa kurikulum Massive Open Online Courses (MOOC), beberapa subject berikut yang hampir selalu dibahas [34]: kalkulus, aljabar linier, statistik, algoritma, machine learning dan algoritma big data. Tambahan pembahasan lain yang juga muncul adalah: teori graf, *game theory*, dan *information theory*.

V. Contoh Simulasi Hadoop dengan Docker

Sebagai uji coba secara sederhana untuk dapat menjalankan komponen *Name Node*, *YARN*, *Data Node*, dan *Map Reduce*,

kita dapat menggunakan Docker container. Adapun komputer yang digunakan untuk uji coba adalah prosesor i3 4160, RAM 12 GB, dengan sistem operasi *host* adalah Fedora 27. Adapun docker container yang digunakan semuanya dibuat oleh Garry Knox [35]. Berikut adalah urutan langkah yang dapat dicoba:

1. Download dan pasang Docker CE
2. dari terminal, masuk sebagai root
   ```
   $ sudo su -
   ```
3. jalankan service dengan:
   ```
   # systemctl start docker
   ```
4. pasang hadoop namenode
   ```
   # docker pull portworx/hadoop-namenode
   ```
5. pasang hadoop yarn
   ```
   # docker pull portworx/hadoop-yarn
   ```
6. pasang hadoop data node
   ```
   # docker pull portworx/hadoop-datanode
   ```
7. jika diperlukan, hapus container yang pernah ada
   ```
   # docker system prune
   ```
8. setup network untuk docker
   ```
   # docker network create hadoopnet
   ```
9. jalankan name node
   ```
   # docker run -itd -p 50070:50070 --
   net=hadoopnet --name namenode
   portworx/hadoop-namenode
   ```
10. jalankan yarn
    ```
    # docker run -itd -p 8088:8088 -p
    19888:19888 --net=hadoopnet --name
    hadoop-yarn -e
    HADOOP_HOST_NAMENODE=e871aefa2484
    portworx/hadoop-yarn
    ```
    Catatan untuk nilai `e871aefa2484` didapat dari perintah `docker ps`
11. dari web browser, kunjungi:
    http://localhost:50070/dfshealth.html#tab-overview
    untuk melihat name node (Gambar 6).



Safe mode is ON. The reported blocks 156 has reached the threshold 0.9990 of total blocks 156. The number of live datanodes 2 has reached the minimum number 0. In safe mode extension. Safe mode will be turned off automatically in 17 seconds.

56 files and directories, 156 blocks = 212 total filesystem object(s).

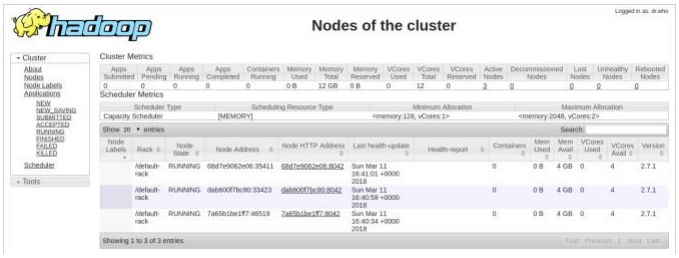
Heap Memory used 31.86 MB of 178 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 29.76 MB of 30.94 MB Commited Non Heap Memory. Max Non Heap Memory is 130 MB.

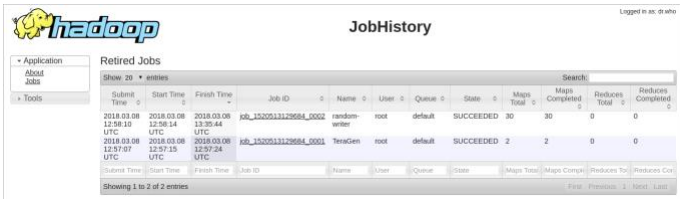
Configured Capacity:	552.67 GB
DFS Used:	39.49 GB (7.15%)
Non DFS Used:	231.75 GB
DFS Remaining:	281.43 GB (50.92%)
Block Pool Used:	39.49 GB (7.15%)
DataNodes usages% (Min/Median/Max/stdDev):	6.17% / 8.12% / 8.12% / 0.98%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	3/11/2018, 11:37:48 PM

Gambar 6. Ilustrasi Ringkasan sebuah *name node*

12. dari web browser, kunjungi (Gambar 7):
    http://localhost:8088/cluster/nodes
13. perhatikan belum ada data node yang aktif di sana.
14. jalankan data node #1
    ```
    # docker run -itd --net=hadoopnet --
    name hadoop-node1 -e
    HADOOP_HOST_NAMENODE=e871aefa2484 -e
    HADOOP_HOST_YARN=7db7630cc77a
    portworx/hadoop-datanode
    ```
15. jalankan data node #2
    ```
    # docker run -itd --net=hadoopnet --
    name hadoop-node2 -e
    HADOOP_HOST_NAMENODE=e871aefa2484 -e
    HADOOP_HOST_YARN=7db7630cc77a
    portworx/hadoop-datanode
    ```
16. jalankan data node #3
    ```
    # docker run -itd --net=hadoopnet --
    name hadoop-node3 -e
    HADOOP_HOST_NAMENODE=e871aefa2484 -e
    HADOOP_HOST_YARN=7db7630cc77a
    portworx/hadoop-datanode
    ```
17. dari web browser, kunjungi kembali:
    http://localhost:8088/cluster/nodes
18. untuk mencoba contoh aplikasi MapReduce, masuk ke bash shell dari yarn:
    ```
    # docker exec -i –t hadoop-yarn
    /bin/bash
    ```
19. berikan perintah:
    ```
    # export PATH=$PATH::/usr/local/hadoop/bin
    # export HADOOP_PREFIX=/usr/local/hadoop
    # cd $HADOOP_PREFIX
    hadoop jar
    $HADOOP_PREFIX/share/hadoop/mapreduce/hadoo
    p-mapreduce-examples-2.7.1.jar teragen
    10000 /teragen
    ```
20. dari web browser, kunjungi (Gambar 8):
    http://localhost:19888/jobhistory



Gambar 7. Ilustrasi Data Node di YARN



Gambar 8. Ilustrasi Job List di YARN

VI. USULAN SKEMA PEMBELAJARAN BIG DATA

Oleh karena untuk membangun infrastruktur ekosistem *Big Data* membutuhkan biaya yang tidak sedikit, pertimbangan tersebut menjadi hal utama bagi pendidikan tinggi untuk dapat menyelenggarakan pembelajaran tentang *Big data*. Mungkin perguruan tinggi yang memiliki kapital kuat saja yang akan

mampu membangunnya. Dalam hal pertimbangan tersebut, pembelajaran tentang *Big Data* dapat tetap diberikan walaupun dengan membangun sebuah laboratorium yang masih sederhana, namun mencukupi untuk dijadikan sebagai sebuah simulator untuk *small data* berbasis ekosistem Hadoop.

Secara sederhana dapat dibangun sebuah laboratorium *small data* dengan 20 komputer, di mana masing-masing terpasang prosesor i3/i5/i7, RAM minimum 16 GByte, Harddisk 1 TB. Setiap komputer tersebut akan bertindak sebagai *data node*. Disediakan juga dua buah komputer server di mana masing-masing dengan spesifikasi minimal: prosesor 8 core, RAM 128 GByte, harddisk 2 TB. Kedua komputer server dihubungkan sebagai satu *rack* dengan kecepatan koneksi 10Gbps. Sedangkan untuk ke-20 komputer terhubung ke satu jaringan yang sama dan ke server dengan bandwidth 10 Gbps.

Sistem operasi yang digunakan untuk semua komputer adalah Linux (dapat menggunakan CentOS atau Ubuntu). Setiap komputer *data node* terpasang sebagai *data node*. Jika pada akhirya dibutuhkan bahwa tiap komputer mahasiswa dipasang Apache BigTop, tidak menjadi persoalan juga dengan catatan beberapa kekurangan akan terasa.

Dengan infrastruktur sederhana tersebut, berikutnya dapat dikembangkan minimal 2 matakuliah, yaitu *Administrasi Ekosistem Big Data*, dan *Data Science*. Garis besar materi yang dapat diberikan pada matakuliah Administrasi Ekosistem Big Data, antara lain: 1) Pengantar Ekosistem Big Data; 2) Setup Hadoop; 3) Operasi berkas di HDFS; 4) Pemrograman Dasar MapReduce; 5) Framework YARN; 6) Apache Pig; 7) dan 8) Analisis statistik di Hadoop; 9) Hadoop dan data warehouse; 10) penyimpanan data dengan HBase; 11) Penyimpanan data dengan Hive; 12) Integrasi RDBMS menggunakan Sqoop; 13) dan 14) Studi kasus. Sedangkan untuk matakuliah Data Science, dapat mengadopsi dari silabus yang dibuat oleh Alcorn [36].

Terhadap contoh-contoh data yang dapat digunakan sebagai bahan percobaan, dapat diperoleh dari beberapa organisasi publik yang menyediakannya. Contoh terkait dengan data terstruktur untuk MariaDB ColumnStore dengan ukuran sekitar 1.7 GB dapat diperoleh[1]. Daftar data publik lain yang dapat digunakan dapat melihat di [37] [38] [39].

VII. PENUTUP

Berdasar studi literatur terhadap *Big Data*, ekosistem *Big Data Hadoop*, percobaan sederhanaan menggunakan Docker, dan tinjauan beberapa silabus, sebaiknya teknologi ekosistem ini dapat menjadi satu pembelajaran di lingkungan sivitas pendidikan tinggi. Ekosistem *Big Data* telah menjadi bagian dalam setiap pengelolaan dan analisis data yang dapat diterapkan di semua bidang. Melihat tantangan kebutuhan analisis data untuk menghasilkan layanan yang lebih baik bagi setiap organisasi atau pemerintah, teknologi *Big data* sudah saatnya mulai diajarkan sejak awal.

[1] dapat diunduh di https://www.dropbox.com/s/ljx19r3mgbw7umh/dbt3.tar.gz

REFERENSI

[1] D. Reinsel, J. Gantz and J. Rydning, "Data Age 2025: the evolution of data to life-critical don't focus on big data; focus on the data that's big," IDC, 2017.

[2] B. Marr, "The Digital Universe Of Opportunities: Rich Data and the Increasing Value Of the Internet Of Things," Forbes, 19 November 2015. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#7cde8dc917b1. [Accessed 10 Maret 2018].

[3] M. Zwolenski and L. Weatherill, "The Digital Universe: Rich data and the increasing value of the internet of things," *Australian Journal of Telecommunications and the Digital Economy*, vol. 2, no. 3, pp. 47.1-47.9, 2014.

[4] McKinsey Indonesia Office, "https://www.mckinsey.com/~/media/McKinsey/Locations/Asia/Indonesia/Our%20Insights/Unlocking%20Indonesias%20digital%20opportunity/Unlocking_Indonesias_digital_opportunity.ashx," Oktober 2016. [Online]. Available: https://www.mckinsey.com/~/media/McKinsey/Locations/Asia/Indonesia/Our%20Insights/Unlocking%20Indonesias%20digital%20opportunity/Unlocking_Indonesias_digital_opportunity.ashx. [Accessed 10 Maret 2018].

[5] Gartner, "Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor," 18 Agustus 2015. [Online]. Available: https://www.gartner.com/newsroom/id/3114217. [Accessed 10 Maret 2018].

[6] A. Woodie, "Why Gartner Dropped Big Data Off the Hype Curve," Datanami, 26 Agustus 2015. [Online]. Available: https://www.datanami.com/2015/08/26/why-gartner-dropped-big-data-off-the-hype-curve/. [Accessed 10 Maret 2018].

[7] K. Panetta, "Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017," Gartner, 15 Agustus 2017. [Online]. Available: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/. [Accessed 10 Maret 2018].

[8] F. X. Diebold, "A Personal Perspective on the Origin(s) and Development of "Big Data": The Phenomenon, the Term, and the Discipline," PIER Working Paper No. 13-003, 26 November 2012. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2202843.

[9] G. Press, "A Very Short History Of Big Data," Forbess, 9 Mei 2013. [Online]. Available: https://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/. [Accessed 10 Maret 2018].

[10] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, pp. 137-144, 2015.

[11] D. Laney, "3-D data management: Controlling data volume, velocityand variety. Application Delivery Strategies by META Group Inc.," 6 Februari 2001. [Online]. Available: https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf. [Accessed 10 Maret 2018].

[12] Gartner Inc., "Gartner's Big Data Definition Consists of Three Parts, Not to Be Confused with Three "V"s," Forbes, 27 Maret 2013. [Online]. Available: https://www.forbes.com/sites/gartnergroup/2013/03/27/gartners-big-data-definition-consists-of-three-parts-not-to-be-

confused-with-three-vs/#2abd0a8042f6. [Accessed 10 Maret 2018].

[13] G. Press, "12 Big Data Definitions: What's Yours?," Forbes, 3 September 2014. [Online]. Available: https://www.forbes.com/sites/gilpress/2014/09/03/12-big-data-definitions-whats-yours/#519a377313ae. [Accessed 10 Maret 2018].

[14] B. Furht and F. Villanustre, "Introduction to Big Data," in *Big Data Technologies and Application*, Switzerland, Springer International Publishing, 2016, pp. 3-11.

[15] M. E. Driscoll, "How much data is "Big Data"? Is there classification for various levels of "Big Data" by amount of data processed or other constraints, like for example throughput? What's the minimum data size which still qualifies as a "Big Data"?," Quora, 25 Desember 2010. [Online]. Available: https://www.quora.com/How-much-data-is-Big-Data-Is-there-classification-for-various-levels-of-Big-Data-by-amount-of-data-processed-or-other-constraints-like-for-example-throughput-What%E2%80%99s-the-minimum-data-size-which-still-qualifies-as-a-Big-Data%E2. [Accessed 10 Maret 2018].

[16] C. Chan, "What Facebook Deals with Everyday: 2.7 Billion Likes, 300 Million Photos Uploaded and 500 Terabytes of Data," Gizmodo, 22 Agustus 2012. [Online]. Available: https://gizmodo.com/5937143/what-facebook-deals-with-everyday-27-billion-likes-300-million-photos-uploaded-and-500-terabytes-of-data. [Accessed 10 Maret 2018].

[17] Twitter Inc., "#numbers," Twitter.com, 14 Maret 2011. [Online]. Available: https://blog.twitter.com/official/en_us/a/2011/numbers.html. [Accessed 10 Maret 2018].

[18] IBM Big Data & Analytics Hub, "The Four V's of Big Data," IBM, [Online]. Available: http://www.ibmbigdatahub.com/infographic/four-vs-big-data. [Accessed 10 Maret 2018].

[19] SAS Institute Inc., "Big Data: What it is and why it matters," SAS Institute Inc., [Online]. Available: https://www.sas.com/en_id/insights/big-data/what-is-big-data.html. [Accessed 10 Maret 2018].

[20] Oracle, "Bringing the Value of Big Data to the Enterprise," [Online]. Available: http://www.oracle.com/us/products/database/big-data-appliance/value-of-big-data-brief-2008771.pdf. [Accessed 10 Maret 2018].

[21] D. Feinleib, Big Data Bootcamp: What Managers Need to Know to Profit from the Big Data Revolution, Apress, 2014.

[22] N. Sawant and H. Shah, Big Data Application Architecture Q & A, Apress, 2013.

[23] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *Journal of Big data*, vol. 1, no. 8, pp. 1-20, 2014.

[24] S. Mazumder, "Big Data Tools and Platforms," in *Big Data Concepts, Theories, and Applications*, Switzerland, Springer International Publishing, 2016, pp. 29-128.

[25] Apache Software Foundation, "HDFS Federation," Apache Hadoop, [Online]. Available: http://hadoop.apache.org/docs/r3.0.0/hadoop-project-dist/hadoop-hdfs/Federation.html. [Accessed 11 Maret 2018].

[26] D. Harris, "Because Hadoop isn't perfect: 8 ways to replace HDFS," GigaOm, 11 Juli 2012. [Online]. Available: https://gigaom.com/2012/07/11/because-hadoop-isnt-perfect-8-ways-to-replace-hdfs/. [Accessed 11 Maret 2018].

[27] T. White, Hadoop The definitive guide: Storage and analysis at Internet scale, Gravenstein Highway North, Sebastopol, CA: O'Reilly Media Inc., 2015.

[28] McObject LLC, "In-Memory Database Systems - Questions and Answers," McObject LLC, [Online]. Available: http://www.mcobject.com/in_memory_database. [Accessed 11 Maret 2018].

[29] M. Rouse and J. Vaughan, "Hadoop data lake," TechTarget, [Online]. Available: http://searchdatamanagement.techtarget.com/definition/Hadoop-data-lake. [Accessed 11 Maret 2018].

[30] J. Caserta and E. Cordo, "Data Warehousing in the Era of Big Data," Database Trend and Applications, 19 Januari 2016. [Online]. Available: http://www.dbta.com/BigDataQuarterly/Articles/Data-Warehousing-in-the-Era-of-Big-Data-108590.aspx. [Accessed 11 Maret 2018].

[31] D. deRoos, "Apache Bigtop and Hadoop," Dummies, [Online]. Available: http://www.dummies.com/programming/big-data/hadoop/apache-bigtop-and-hadoop/. [Accessed 11 Maret 2018].

[32] L. Joseji, "6 Sparkling Features of Apache Spark!," Big Data Zone, 15 Agustus 2014. [Online]. Available: https://dzone.com/articles/6-sparkling-features-apache. [Accessed 11 Maret 2018].

[33] G. Vargas-Solar, J. A. Espinosa-Oviedo and J. L. Zechinelli-Martini, "Big Continuous Data: Dealing with Velocity by Composing Event Streams," in *Big Data Concepts, Theories, and Applications*, Switzerland, Springer International Publishing, 2016, pp. 1-27.

[34] M. A. Alcorn, "How to become a data scientist," opensource.com, 13 September 2017. [Online]. Available: https://opensource.com/article/17/9/data-scientist. [Accessed 11 Maret 2018].

[35] G. Knox, "Hadoop in Docker with Networking and Persistent Storage," Github, 23 November 2015. [Online]. Available: https://github.com/portworx/docker-hadoop. [Accessed 11 Maret 2018].

[36] M. A. Alcorn, "Michael's Data Science Curriculum," 11 September 2016. [Online]. Available: https://github.com/airalcorn2/Michael-s-Data-Science-Curriculum. [Accessed 11 Maret 2018].

[37] hadoop illuminated, "Publicly Available Big Data Sets," hadoop illuminated, [Online]. Available: http://hadoopilluminated.com/hadoop_illuminated/Public_Bigdata_Sets.html. [Accessed 11 Maret 2018].

[38] B. Marr, "Big Data: 33 Brilliant And Free Data Sources Anyone Can Use," Forbes, 12 Februari 2016. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2016/02/12/big-data-35-brilliant-and-free-data-sources-for-2016/#322f1ff0b54d. [Accessed 11 Maret 2018].

[39] dbpedia, "DBPedia Dataset," dbpedia, [Online]. Available: http://wiki.dbpedia.org/Datasets. [Accessed 11 Maret 2018].

[40] W. Eckerson, "Big Data and its Impact on Data Warehousing," 2012. [Online]. Available: http://cdn.ttgtmedia.com/BeyeNETWORK/downloads/BigDataE-Book_final.pdf. [Accessed 11 Maret 2018].

[41] H. Karambelkar, Scaling Big Data with Hadoop and Solr, Birmingham: Packt Publishing, 2013.