BAB II

LANDASAN TEORI

2.1 Big Data

2.1.1 Pengertian Data, Informasi, dan Pengetahuan

Data adalah elemen yang paling dasar, bersifat diskrit, dan belum diproses, sehingga belum memiliki makna. Contoh: angka, kata, kode, tabel, dan basis data (Halim, 2018).

Berdasarkan pengertian di atas, data merupakan suatu hal yang paling mendasar yang dibutuhkan berbagai perusahaan untuk diperoleh dari proses-proses operasional sehari-hari maupun sumber-sumber luar yang akan diolah menurut keinginan perusahaan tersebut.

Informasi adalah elemen yang saling terhubung dan merupakan hasil pemrosesan terhadap data, sehingga memiliki suatu makna. Contoh: kalimat, paragraf, persamaan, konsep, ide, pertanyaan, dan cerita sederhana (Halim, 2018).

Berdasarkan pengertian di atas, informasi adalah hasil dari penyusunan data-data yang tersusun dan bertransformasi yang menjadikan data tersebut dapat memberikan makna baru kepada penerima.

Pengetahuan adalah kumpulan informasi yang terorganisir mengenai suatu bidang yang sudah dipahami. Contoh: teori, aksiom, kerangka kerja konseptual, cerita rumit, dan fakta (Halim, 2018).

Berdasarkan pengertian di atas, *knowledge* menjadi sarana bagi para penerima yang terdiri dari informasi-informasi untuk pemahaman, untuk

media pembelajaran, sebagaimana yang diperlukan oleh seorang manajer perusahaan untuk membuat keputusan-keputusan yang krusial dan berdampak besar bagi perusahaan, dimana kesalahan atau kecacatan dalam *knowledge* dapat memberikan dampak buruk bagi perusahaan.

2.1.2 Pengertian Data Warehouse

Data Warehouse adalah sebuah Sistem Informasi yang menyimpan data sekarang dan historical dari satu atau lebih sumber data yang digunakan untuk tujuan reporting dan analysis dari sebuah organisasi (Febrianto, 2019).

Sedangkan menurut Firman Noor Hasan (2021) *Data Warehouse* adalah basis data relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, dan biasanya mengandung *history* data dari proses transaksi dan bisa juga data dari sumber lainnya.

Berdasarkan pengertian di atas, *Data warehouse* salah satu bagian yang penting dari struktur/arsitektur suatu BI karena posisinya sebagai penyimpanan data- data yang telah tersusun dan terorganisasi yang telah memiliki makna, jadi maka dari itu harus memiliki struktur data serta desain yang baik untuk mendukung pengambilan data-data dan informasi secara akurat, tepat dan cepat dari dalam data warehouse itu sendiri.

2.1.3 Pengertian *Big Data*

Menurut Zikopoulos et al (2012) *Big Data* merupakan istilah yang berlaku untuk informasi yang tidak dapat diproses atau dianalisis menggunakan alat tradisional.

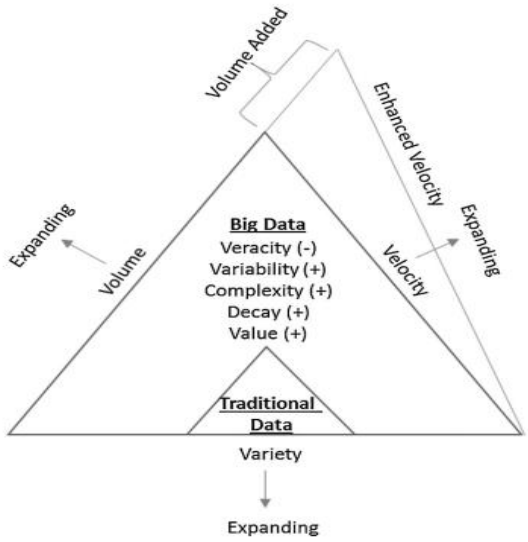
Big Data adalah data yang memiliki volume besar sehingga tidak dapat diproses menggunakan alat tradisional biasa dan harus menggunakan cara dan alat baru untuk mendapatkan nilai dari data ini (Widy, 2017).

Sedangkan *Big Data* menurut Gartner (2013) didefinisikan sebagai data yang memiliki ukuran(volume), kecepatan(velocity), dan/atau ragam (variety) yang ekstrim, yang menuntut pemrosesan informasi yang cepat dan inovatif untuk mendukung pengambilan keputusan dan otomatisasi proses.

Menurut pengertian para ahli di atas, dapat disimpulkan bahwa *Big Data* adalah data yang memiliki volume besar dan sangat *kompleks* sehingga tidak bisa diproses atau dianalisis menggunakan alat tradisional biasa dan harus menggunakan cara dan alat baru untuk mendapatkan isi dan nilai dari *Big Data* ini.

2.1.4 Dimensi-dimensi *Big Data*

Ada 3 dimensi awal dalam Big Data yaitu 3V: *Volume, Velocity,* dan *Variety*.



Gambar 2. 1 Dimensi *Big Data* (Lee, 2017)

i. Volume

Volume mengacu pada jumlah data yang dikumpulkan atau dihasilkan oleh organisasi atau individu. Sedangkan saat ini minimal 1 *terabyte* adalah ambang batas data besar, ukuran minimum untuk memenuhi syarat karena *big data* merupakan fungsi dari perkembangan teknologi. Saat ini, 1 *terabyte* menyimpan data sebanyak mungkin muat di 1.500 *CD* atau 220 *DVD*, cukup untuk menyimpan sekitar 16 juta foto *Facebook* Gandomi & Haider (2015). *E-commerce*, media sosial, dan sensor menghasilkan volume data tidak terstruktur yang tinggi seperti audio, gambar, dan video. Data baru memiliki telah ditambahkan pada tingkat yang meningkat karena lebih banyak komputasi perangkat terhubung ke internet.

ii. Velocity

Velocity mengacu pada kecepatan di mana data berada dihasilkan dan diproses. Kecepatan data bertambah seiring waktu. Awalnya, perusahaan menganalisis data menggunakan sistem pemrosesan *batch* karena sifat pemrosesan data yang lambat dan mahal. Seperti kecepatan pembuatan dan pemrosesan data yang meningkat, pemrosesan *real time* menjadi norma untuk aplikasi komputasi. Gartner Inc (2015) sebelumnya menyatakan bahwa 6,4 miliar perangkat yang terhubung akan melakukannya digunakan di seluruh dunia pada tahun 2016 dan jumlahnya akan mencapai 20,8 miliar pada tahun 2020. Pada 2016, 5,5 juta perangkat baru diperkirakan akan terhubung setiap hari untuk mengumpulkan, menganalisis, dan berbagi data. Kemampuan *streaming* data yang ditingkatkan dari perangkat yang terhubung akan terus mempercepat kecepatan.

iii. Variety

Variety mengacu pada jumlah tipe data. Kemajuan teknologi memungkinkan organisasi untuk menghasilkan berbagai jenis struktur, semi-terstruktur, dan data tidak terstruktur. *Teks*, foto, audio, video, data *clickstream*, dan data sensor adalah contohnya data tidak terstruktur, yang kurang terstandarisasi struktur yang diperlukan untuk komputasi yang efisien. Data semi terstruktur tidak sesuai dengan spesifikasi dari *database* relasional, tetapi dapat ditentukan untuk memenuhi kebutuhan struktural aplikasi tertentu. Sebuah Contoh data semi terstruktur adalah *Extensible Business Reporting Language* (XBRL), dikembangkan

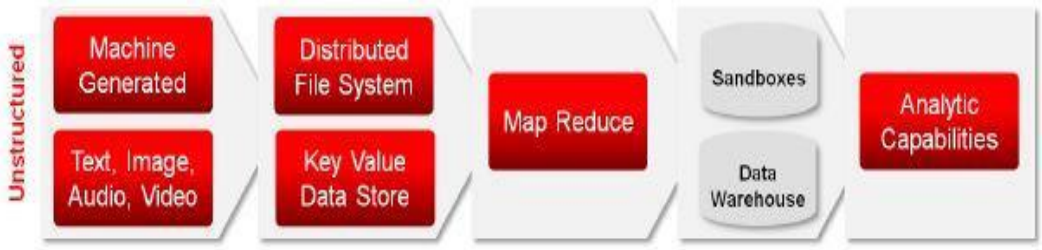
untuk mengubah data keuangan antar organisasi dan agensi pemerintahan. Data terstruktur sudah ditentukan sebelumnya dan dapat ditemukan berbagai jenis basis data tradisional. Saat teknik analitik baru dikembangkan, data tidak terstruktur dibuat dengan lebih cepat menilai dari data terstruktur dan tipe data menjadi lebih sedikit dari hambatan untuk analisis.

2.1.5 Arsitektur *Big Data*

Untuk memahami level aspek arsitektur yang tinggi dari *Big Data*, sebelumnya harus memahami arsitektur informasi logis untuk data yang terstruktur. Pada penjelasan di bawah ini menunjukkan dua sumber data yang menggunakan teknik integrasi (*ETL*/*Change Data Capture*) untuk mentransfer data ke dalam DBMS *data warehouse* atau operational *data store*, lalu menyediakan bermacam-macam variasi dari kemampuan analisis untuk menampilkan data. Beberapa kemampuan analisis ini termasuk ; *dashboards*, laporan, *EPM/BI Applications*, ringkasan dan *query statistic*, interpretasi *semantic* untuk data tekstual, dan alat visualisasi untuk data yang padat. Informasi utama dalam prinsip arsitektur ini termasuk cara memperlakukan data sebagai aset melalui nilai, biaya, resiko, waktu, kualitas dan akurasi data.

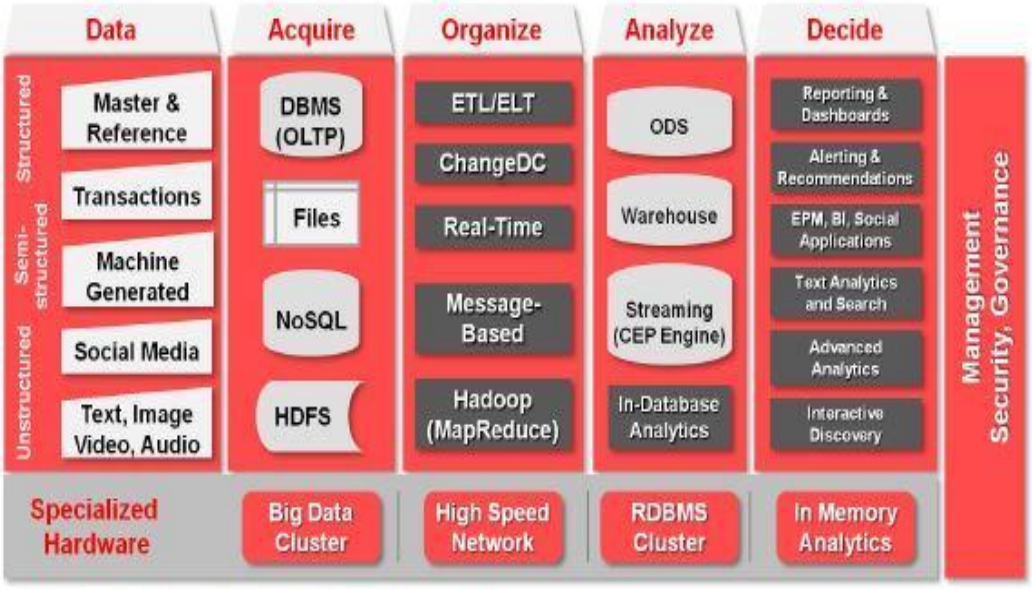
Mendefinisikan kemampuan memproses untuk *big data architecture*, diperlukan beberapa hal yang perlu dilengkapi; volume, percepatan, variasi, dan nilai yang menjadi tuntutan. Ada strategi teknologi yang berbeda untuk *real-time* dan keperluan *batch processing*. Untuk *real-time*, menyimpan data

nilai kunci, seperti NoSQL, memungkinkan untuk performa tinggi, dan pengambilan data berdasarkan indeks. Untuk *batch processing*, digunakan teknik yang dikenal sebagai *Map Reduce*, memfilter data berdasarkan pada data yang spesifik pada strategi penemuan. Setelah data yang difilter ditemukan, maka akan dianalisis secara langsung, dimasukkan ke dalam *unstructured database* yang lain, dikirimkan ke dalam perangkat *mobile* atau digabungkan ke dalam lingkungan *data warehouse* tradisional dan berkorelasi pada data terstruktur.



Gambar 2. 2 *Big Data Information Architecture Capabilities* (Heller & Piziak, 2015)

Kekuatan informasi ada dalam kemampuan untuk asosiasi dan korelasi. Maka yang dibutuhkan adalah kemampuan untuk membawa sumber data yang berbeda-beda, memproses kebutuhan bersama-sama secara tepat waktu dan analisis yang berharga.



Gambar 2. 3 *Oracle Integrated Information Architecture Capabilities* (Heller & Piziak, 2015)

Ketika bermacam-macam data telah didapatkan, data tersebut dapat disimpan dan diproses ke dalam DBMS tradisional, *simple files*, atau sistem *cluster* terdistribusi seperti NoSQL dan Hadoop Distributed File System (HDFS).

Secara arsitektur, komponen kritikal yang memecah bagian tersebut adalah layer integrasi yang ada di tengah. Layer integrasi ini perlu untuk diperluas ke seluruh tipe data dan domain, dan menjadi jembatan antara data penerimaan yang baru dan tradisional, dan pengolahan kerangka. Kapabilitas integrasi data perlu untuk menutupi keseluruhan spektrum dari kecepatan dan frekuensi. Hal tersebut diperlukan untuk menangani kebutuhan ekstrim dan *volume* yang terus bertambah banyak. Oleh karena itu diperlukan teknologi

yang memungkinkan untuk mengintegrasikan Hadoop / Mapreduce dengan data warehouse dan data transaksi.

Layer berikutnya digunakan untuk *load* hasil reduksi dari big data ke dalam *data warehouse* untuk analisis lebih lanjut. Diperlukan juga kemampuan untuk mengakses data terstruktur seperti informasi profil pelanggan ketika memproses dalam big data untuk mendapatkan pola seperti mendeteksi aktivitas yang mencurigakan.

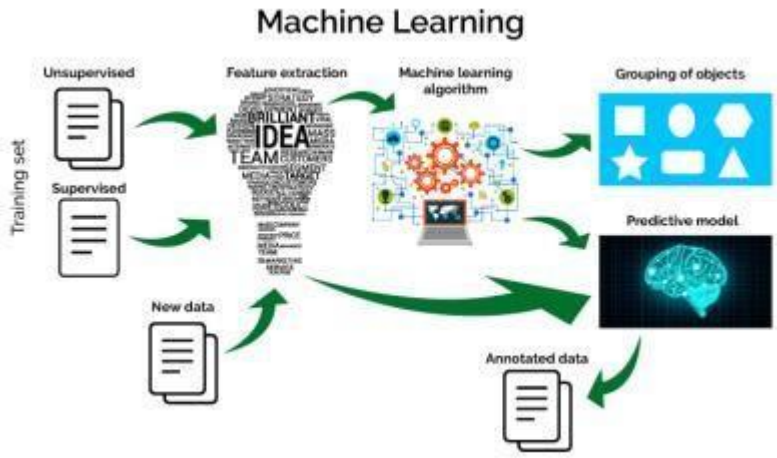
Hasil pemrosesan data akan dimasukkan ke dalam ODS tradisional, *data warehouse*, dan *data marts* untuk analisis lebih lanjut seperti data transaksi. Komponen tambahan dalam layer ini adalah *Complex Event Processing* untuk menganalisa arus data secara *real-time*. Layer *business intelligence* akan dilengkapi dengan analisis lanjutan, dalam analisis *database* statistik, dan visualisasi lanjutan, diterapkan dalam komponen tradisional seperti laporan, *dashboards*, dan *query*. Pemerintahan, keamanan, dan pengelolaan operasional juga mencakup seluruh spektrum data dan lanskap informasi pada tingkat *enterprise*.

Dengan arsitektur ini, pengguna bisnis tidak melihat suatu pemisah, bahkan tidak sadar akan perbedaan antara data transaksi tradisional dan *big data*. Data dan arus analisis akan terasa mulus tanpa halangan ketika dihadapkan pada bermacam-macam data dan set informasi, hipotesis, pola analisis, dan membuat keputusan.

2.2 Machine Learning

Machine learning adalah cabang dari ilmu kecerdasan buatan yang berfokus pada pembangunan dan studi sebuah sistem agar mampu belajar dari data-data yang diperolehnya (Fikriya et al., 2017).

Machine learning merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran (Danakusumo, 2017).



Gambar 2. 4 *Machine Learning* (Pantech, 2018)

Machine learning merupakan cabang dari Artificial Intelligence dengan kemampuan mesin untuk mengakses data yang ada dengan perintah mereka sendiri (Learners, 2019).

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel di tahun 1959. *Machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas (Samuel, 1959).

Machine Learning (ML) atau pembelajaran mesin merupakan pendekatan dalam AI yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi (Hania, 2017).

Dalam pembelajaran *machine learning*, terdapat beberapa skenario-skenario. seperti:

1. *Supervised Learning*

   Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

   Penggunaan skenario *Unsupervised Learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui.

3. *Reinforcement Learning*

   Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar. Saat ini telah banyak pendekatan *machine learning* yang digunakan untuk deteksi *spam*, *Optical character recognition (OCR)*, pengenalan wajah, deteksi penipuan *online*, *NER (Named Entity Recognition)*, dan *Part-of-Speech Tagger*.

2.3 Support Vector Machine

Menurut Mohammed, M., Khan, M. B., & Bashier (2017) SVM dalam *machine learning* dikenal juga dengan *support vector network* yang merupakan metode *supervised* terkait dengan *learning algorithm* untuk analisa pola data yang digunakan untuk klasifikasi dan regresi.

Sementara itu, Islami (2019) SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*. Tulisan ini membahas teori dasar SVM dan aplikasinya dalam bioinformatika, khususnya pada analisa ekspresi gen yang diperoleh dari analisa *microarray*.

SVM merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linier maupun non linear (Samsudiney, 2019).

SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antar kelas dalam 3D disebut *plane similarly*, pasangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi disebut *hyperplane*.

Savan Patel (2017) RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara linear. RBF kernel memiliki dua parameter yaitu Gamma dan Cost. Parameter Cost atau biasa disebut sebagai C merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi di setiap sampel dalam training dataset. Parameter Gamma menentukan seberapa jauh pengaruh dari satu sampel training dataset dengan nilai rendah berarti “jauh”, dan nilai tinggi berarti “dekat”. Dengan gamma yang rendah, titik yang berada jauh dari garis pemisah yang masuk akal dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika gamma tinggi berarti titik – titik berada di sekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan

2.4 Google Colab

Google colab adalah lingkungan *Jupyter notebook* gratis yang berjalan sepenuhnya dalam berbasis *cloud* (Tutorials Point, 2019).

Sedangkan menurut Naik (2021) *Colaboratory*, atau singkatnya *'Colab'*, adalah produk dari *Google Research. Colab* memungkinkan siapa saja untuk menulis dan mengeksekusi kode *python arbitrer* melalui *browser*, dan sangat cocok untuk pembelajaran mesin, analisis data, dan pendidikan.

Berdasarkan penelitian di atas, *Google colab* adalah suatu *tools* untuk menulis kode *python* yang digunakan untuk proses data analisis berbasis *cloud*.

Adapun beberapa *library* yang yang mendukung *Google colab* untuk proses analisis data diantaranya:

a) Numpy

NumPy (*Numerical Python*) adalah *library python* yang digunakan untuk bekerja dengan *array* dan juga memiliki fungsi yang bekerja dalam domain aljabar linier, transformasi fourier, dan matriks.

b) Scipy

SciPy (*Scientific Python*) adalah perpustakaan *open-source* yang digunakan untuk perhitungan ilmiah tingkat tinggi yang dibangun di atas ekstensi *NumPy* dan bekerja bersama untuk menangani komputasi yang kompleks.

c) Pandas

Library untuk *machine learning* yang bersifat *open source* ini menyediakan struktur data tingkat tinggi yang fleksibel serta berbagai alat analisis. Penggunaannya memudahkan analisis data, manipulasi data, dan pembersihan data. Pandas mendukung berbagai jenis operasi seperti penyortiran, pengindeksan ulang, iterasi, penggabungan, konversi data, visualisasi, agregasi, dan lain sebagainya.

d) Matplotlib

Jenis *library* ini bertanggung jawab untuk merencanakan data numerik. Itulah alasan *Matplotlib* digunakan dalam analisis data. *Library python*

yang bersifat open source ini dapat memplot angka-angka berdefinisi tinggi seperti diagram lingkaran, histogram, scatterplot, grafik, dan lain-lain.

e) Scikit-learn

Library open source ini mendukung *machine learning* dengan mendukung berbagai algoritma yang diawasi dan tidak diawasi seperti regresi linier, klasifikasi, pengelompokan, dan lain sebagainya. *Library* ini bekerja sama dengan *Numpy* dan *SciPy*.

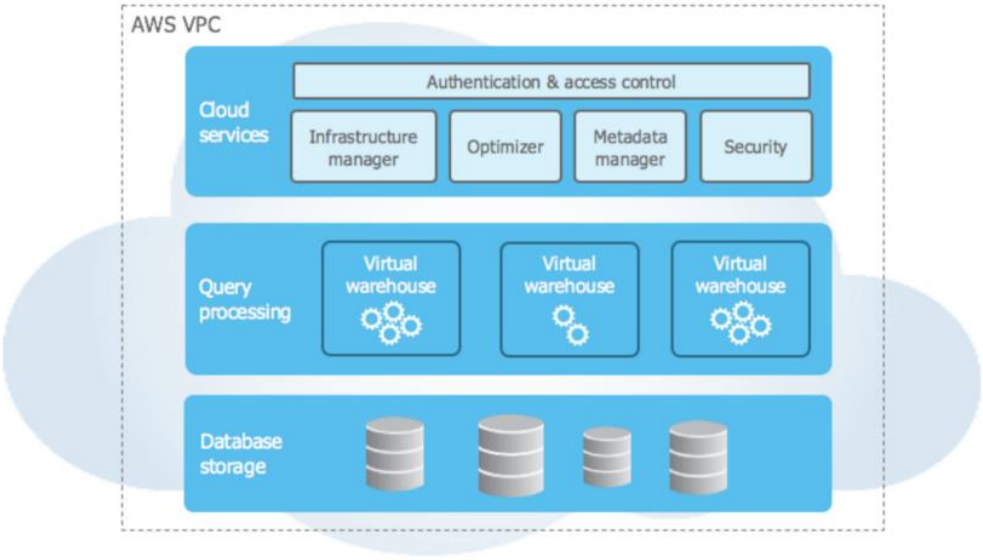
2.5 Snowflake

Pada penelitian yang dilakukan Melissaris et al (2022) Menjelaskan bahwa Snowflake *Data Cloud* itu adalah *platform SaaS* yang mendukung berbagai beban kerja data seperti penyimpanan data, data *lake*, data *science*, rekayasa data, dan lain-lain.

Dageville et al (2016) memaparkan bahwa Snowflake adalah sebuah sistem *Multi-tenancy*, transaksional, aman, berskala tinggi dan elastis, dengan dukungan SQL penuh, dan ekstensi bawaan untuk data semi-terstruktur dan *schema-less* data.

Sedangkan menurut Hashmap (2018) Snowflake adalah *cloud data warehouse* yang dibangun di atas infrastruktur cloud Amazon Web Services (AWS) dimana tidak ada perangkat keras baik virtual ataupun fisik serta perangkat lunak yang perlu dibangun, konfigurasikan, atau dikelola, yang mana semua perawatan, manajemen, dan penyetelan yang sedang berlangsung ditangani oleh Snowflake.

Secara arsitektur ada 3 komponen utama yang membentuk Snowflake *cloud data warehouse*



Gambar 2. 5 Arsitektur Snowflake

1. *Database Storage*, adalah sistem file mendasar yang sebenarnya di Snowflake didukung oleh S3 di akun Snowflake, semua data dienkripsi, dikompresi, dan didistribusikan untuk mengoptimalkan kinerja. Di Amazon S3, data bersifat geo-redundant dan memberikan ketahanan dan ketersediaan data yang sangat baik.
2. *Query Processing*, dimana Snowflake menyediakan kemampuan untuk membuat "*Virtual Warehouse*" yang pada dasarnya menghitung *cluster* di EC2 yang disediakan di belakang layar. *Virtual Warehouse* dapat digunakan untuk memuat data atau menjalankan *query* dan mampu melakukan kedua tugas ini secara bersamaan. *Virtual Warehouse* ini dapat ditingkatkan atau diturunkan

sesuai permintaan dan dapat dihentikan sementara saat tidak digunakan untuk mengurangi pengeluaran pada penghitungan.

3. *Cloud Services*, yang mana mengkoordinasikan dan menangani semua layanan lain di Snowflake termasuk sesi, otentikasi, kompilasi SQL, dan enkripsi.

Dengan desain arsitektur tersebut, masing-masing dari 3 lapisan ini dapat diskalakan secara independen dan melimpah.

2.6 Penelitian Terkait

Penelitian terdahulu sebagai kajian penelitian yang akan dilakukan sangat penting untuk mengetahui hubungan antara penelitian yang dilakukan sebelumnya dengan penelitian yang dilakukan saat ini serta dapat menghindari adanya duplikasi. Hal ini bermanfaat untuk menunjukkan bahwa penelitian yang dilakukan mempunyai arti penting sehingga dapat diketahui kontribusi penelitian terhadap ilmu pengetahuan.

Tabel 2. 1 Penelitian Terkait

No	Peneliti/Tahun	Judul	Problem	Metode / Algoritma / Teknik / Model / Sensor / Platform	State Of The Art / Keterbaruan
1.	(Demidova et al., 2016)	*Big Data Classification Using the SVM Classifiers with the Modified Particle Swarm Optimization and the SVM Ensembles*	Permasalahan dengan pengembangan dukungan *SVM* menggunakan *Particle Swarm Optimization* (PSO).	Algoritma: *Support Vector Machine* Teknik: *Particle Swarm Optimization,* dan *SVM Ensembles*	Hasil penelitian menghasilkan efisiensi dari pendekatan, *SVM ensembles* dengan akurasi 85,75%-91,5%. Sementara keakuratan *SVM two level classifier* sebesar 97,26%.

Lanjutan Tabel 2.1

2.	(Karya & Moertini, 2017)	Eksplorasi Teknologi Big Data Hadoop Untuk Sistem Aplikasi Berbasis Komunitas Studi Kasus: Aplikasi Pembukuan UMK	Pengembangkan aplikasi pembukuan untuk usaha mikro dan kecil (UMK) berbasis mobile cloud.	Teknik: *big data processing*	Hasil studi dan penerapannya menunjukkan bahwa Hadoop dapat diadopsi pada aplikasi pembukuan UMK khususnya HBase.
3.	(L. B. D. Cahyo, 2018)	Implementasi Metode *Support Vector Machine* Untuk Melakukan	Penderita *medulloblastoma* antara 18%-20% dari semua tumor otak pada anak-	Algoritma: *Support Vector Machine* Teknik: *Microarray*	Berdasarkan hasil analisis SVM mampu memprediksi kelas

Lanjutan Tabel 2.1

		Klasifikasi Pada Data Bioinformatika	anak serta 70% dari penderita *medulloblastoma* terdeteksi pada usia 10 tahun ke bawah.		penderita dengan akurasi 95% dengan nilai AUC 98%
4.	(Oliviandi et al., 2018)	Implementasi Apache Spark Pada Big Data Berbasis Hadoop *Distributed File System*	Big data merupakan kumpulan data dalam skala besar, yang mempunyai karakteristik data yang variatif, sangat cepat pertumbuhannya dan kompleks datanya.	Algoritma: *MapReduce*	Skenario yang digunakan adalah memproses *wordcount* suatu data dengan besar data yang berbeda yang bertujuan

Lanjutan Tabel 2.1

					untuk menganalisis *response time* dan penggunaan hardware dari kedua platform tersebut.
5.	(Dageville et al., 2016)	*The Snowflake Elastic Data Warehouse*	Pada saat ini Model Software-as-a-Service (SaaS) menghadirkan kelas perusahaan		Snowflake adalah sistem multi-tenant, transaksional, aman, sangat scalable dan elastis dengan dukungan

Lanjutan Tabel 2.1

			sistem untuk pengguna yang sebelumnya tidak mampu membayar sistem seperti itu karena biaya dan kompleksitasnya		SQL penuh dan ekstensi bawaan untuk data semi-terstruktur dan kurang berskema.
6.	(Febtriani, 2018)	*Studi dan Perbandingan Apache Spark SQL dan Hive Dalam Konteks Analisis Big Data*	Agar pengguna bisa mendapatkan informasi dari Big Data, Big Data harus diolah dan dianalisis dengan cepat, yaitu harus diolah dan dianalisis	Algoritma: *Word Count* dengan *MapReduce*	Waktu eksekusi kueri Spark SQL dan Hive adalah Spark SQL lebih cepat dalam

Lanjutan Tabel 2.1

			dengan teknologi Apache Hadoop dan Apache Spark.		mengerjakan kueri dibandingkan Hive.
7.	(K. D. Cahyo, 2018)	Studi dan Implementasi Apache Spark MLLIB Untuk Analisis Big Data	Dibutuhkan komputer dengan kekuatan komputasi yang sangat tinggi untuk menganalisis data dengan ukuran yang sangat besar.	Metode: *Machine Learning*	Kinerja dari fungsi-fungsi MLlib sangat baik untuk komputasi pada ukuran data yang besar.
8.	(Melissaris et al., 2022)	Elastic Cloud Services	Pelanggan Snowflake	Metode: ECS Cluster Management	Mengevaluasi kemampuan ECS dalam produksi dan

Lanjutan Tabel 2.1

			mengharapkan tersedia untuk menjalankan beban kerja mereka dengan kinerja tinggi. Berada di background, perangkat lunak yang menjalankan beban kerja pelanggan perlu dilayani dan dikelola. Selain itu, kegagalan dalam komponen individu seperti kebutuhan Mesin Virtual (VM).		menyajikan hasil pada skala Cloud Data Snowflake.

1.6 Penelitian Terdekat

Berikut adalah beberapa penelitian terdekat dengan penelitian ini:

Tabel 2. 2 Penelitian terdekat

No	Peneliti/Tahun	Judul	Problem	Metode / Algoritma / Teknik / Model / Sensor / Platform	State Of The Art / Keterbaruan
1.	(Oliviandi et al., 2018)	Implementasi Apache Spark Pada Big Data Berbasis Hadoop *Distributed File System*	Big data merupakan kumpulan data dalam skala besar, yang mempunyai karakteristik data yang variatif, sangat cepat pertumbuhannya dan kompleks datanya.	Algoritma: *MapReduce*	Skenario yang digunakan adalah memproses *wordcount* suatu data dengan besar data yang berbeda yang bertujuan

Lanjutan tabel 2.2

2.	(K. D. Cahyo, 2018)	Studi dan Implementasi Apache Spark MLLIB Untuk Analisis Big Data	Dibutuhkan komputer dengan kekuatan komputasi yang sangat tinggi untuk menganalisis data dengan ukuran yang sangat besar.	Metode: *Machine Learning*	Kinerja dari fungsi-fungsi MLlib sangat baik untuk komputasi pada ukuran data yang besar.

Tabel 2.2 menjelaskan beberapa penelitian terkait yang dijadikan acuan untuk penelitian dalam bidang *Big Data Analysist*. Penelitian berjudul "Studi dan Implementasi *Apache Spark* MLLIB Untuk Analisis Big Data" yang dilakukan K. D. Cahyo (2018) dengan memanfaatkan *Apache Spark* MLLIB dalam proses *Big Data Analysist*.

Penelitian berjudul "Implementasi *Apache Spark* Pada Big Data Berbasis Hadoop *Distributed File System*" yang dilakukan Oliviandi et al (2018) menghasilkan sebuah skenario yang digunakan dalam memproses *wordcount* suatu data dengan besar data yang berbeda yang bertujuan untuk menganalisis *response time* dan penggunaan hardware.