

KOMPUTASI PARALEL UNTUK PERHITUNGAN RELASI DOMINASI MENGGUNAKAN SKYLINE QUERY PADA LOKASI WISATA DI PULAU LOMBOK

(Parallel Computing For Calculation Of Domination Relations Using Skyline Query On Tourism Locations in Lombok Island)

I Made Sathya Wirya Putra, Heri Wijayanto, Ahmad Zafrullah M.

Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: adewpsathya@gmail.com, [heri, zaf]@unram.ac.id (Calibri 10, normal)

Abstract

Pulau Lombok saat ini memiliki berbagai macam daerah wisata, untuk mengelola daerah wisata tersebut maka dibutuhkan metode untuk menentukan aspek yang harus dikembangkan di objek wisata tersebut. Dalam pengembangannya dapat menggunakan metode Skyline Query untuk mencari data skyline, yang dimana Skyline Query merupakan metode umum untuk menemukan hubungan dominasi antar objek. Setelah didapatkan data skyline tersebut maka selanjutnya akan dibentuk menjadi Dominant Graph Intersection (DGI). Dalam implementasi menggunakan metode Skyline Query untuk membentuk DGI digunakan Hadoop Cluster yang disediakan oleh Apache untuk melakukan tugas MapReduce dikarenakan implementasinya membagi tugasnya ke setiap node sehingga implementasinya menjadi lebih efisien. Dalam pengujiannya dilakukan 3 macam pengujian yakni pengujian dengan membedakan pada jumlah data, pengujian dengan membedakan jumlah dimensi dan pengujian dengan membedakan jumlah node. Pengujian dengan membedakan jumlah data dengan 13 dimensi dan menggunakan 4 node waktu pengujian berbanding lurus dengan jumlah intersection yang dihasilkan yakni bertambah dengan cepat ketika data mencapai angka jutaan. Untuk pengujian jumlah dimensi tidak terlalu berpengaruh dalam waktu pengujian, dimana dengan menggunakan 17 data dan 4 node mengalami perubahan secara signifikan. Dan pada pengujian jumlah node dengan 17 data yang dimana memiliki 13 dimensi memiliki waktu pengujian optimum pada pengujian menggunakan 4 node, yang dimana ketika pengujian dengan 5 node waktu pengujian mengalami kenaikan yakni dikarenakan adanya overhead communication.

Keywords: Hadoop, MapReduce, Elastic Map Reduce, Sistem Terdistribusi

1. PENDAHULUAN

Pulau Lombok terletak di Nusa Tenggara Barat Indonesia yang merupakan daerah tujuan wisata tingkat dunia. Banyak potensi wisata yang terdapat di Lombok yang menarik diantaranya adalah wisata pantai, air terjun, budaya, kuliner dan masih banyak lagi yang lain. Beberapa potensi wisata tersebut telah dikelola oleh pemerintah daerah dan ada juga yang memerlukan perhatian lebih untuk mengoptimalkan potensinya.

Disamping kepariwisataan di Lombok juga didukung oleh adanya sirkuit Internasional Pertamina Mandalika yang berlokasi di Lombok Tengah. Pada November 2021 dan Mei 2022 telah dilaksanakan perlombaan internasional Superbike dan MotoGP secara berturut-turut. Oleh karena itu banyak pendatang dari dalam dan luar negeri datang ke

Lombok untuk menonton acara tersebut dan sekaligus berlibur. Tentu saja hal ini berdampak positif pada masyarakat dan pemerintah karena terbukanya lapangan pekerjaan pada sektor pariwisata. Sedangkan pemerintah mendapat penghasilan berupa pajak dari wisatawan.

Oleh sebab itu perlu pengembangan daerah wisata di Lombok agar wisatawan lebih tertarik dan berkesan. Pengembangan objek wisata harus mempertimbangkan beberapa aspek yang penting supaya pengambilan keputusannya bisa efektif dan efisien. Maka, dibutuhkan sebuah alat bantu pengambilan keputusan untuk menentukan aspek apa yang harus dikembangkan di objek wisata tersebut. Untuk menentukan aspek apa yang harus dikembangkan pada sebuah objek wisata maka ada beberapa cara, diantaranya adalah Top-k Query [1] dan Skyline Query [2].

Top-k Query merupakan sebuah metode umum untuk mencari sejumlah k data terbaik berdasar atribut yang dimiliki dan sebuah fungsi skor tertentu [1]. Sebagai contoh untuk mencari obyek wisata terbaik berdasar 2 atribut (aspek) yaitu a1 dan a2. Maka untuk menghitung skor (s) digunakan persamaan seperti di Persamaan 1, dimana w1 dan w2 merupakan bobot untuk tiap atribut. Sebagai contoh, untuk objek wisata pantai Senggigi, pantai Kuta dan pantai Pink, atribut pertama adalah jarak lokasi wisata dengan kota (a1) dengan satuan km bernilai 3 untuk pantai Senggigi, 8 untuk pantai Kuta dan 9 untuk pantai Pink. Sedangkan atribut kedua adalah biaya penginapan di sekitar pantai (a2) dengan satuan ratus ribu rupiah dimana diberikan nilai 7 untuk pantai Senggigi, 3 untuk pantai Kuta, dan 4 untuk pantai Pink. Dalam contoh ini diasumsikan nilai untuk tiap atribut semakin kecil semakin baik sehingga nilai bobot dibuat bernilai negatif untuk menjaga konsistensi nilai skor semakin tinggi semakin baik. Dan bobot untuk masing-masing atribut yang didefinisikan sebagai (w1 dan w2) dimana untuk atribut a1 diberikan nilai bobot -5 dan a2 diberikan nilai bobot -2. Hasil perhitungan dengan Persamaan 1 untuk ketiga lokasi wisata tersebut dapat dilihat pada Tabel I. Dari Tabel I terlihat bahwa pantai Senggigi mendapat rangking tertinggi atau memiliki skor terbesar dan disusul oleh pantai Kuta dan pantai Pink.

$$s = w_1 \cdot a_1 + w_2 \cdot a_2 \quad (1)$$

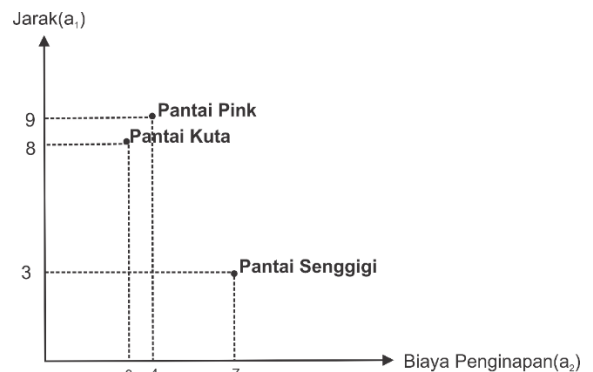
TABEL I. CONTOH PERHITUNGAN METODE TOP-K QUERY

Destinasi Wisata	Jarak (a ₁)	Biaya Penginapan (a ₂)	w ₁ *a ₁	w ₂ *a ₂	Nilai total (w ₁ *a ₁ +w ₂ *a ₂)	Rangking
Pantai Senggigi	3	7	-15	-14	-29	1
Pantai Kuta	8	3	-40	-6	-46	2
Pantai Pink	9	4	-45	-8	-53	3

Kelemahan metode top-k query terletak pada penentuan bobot yang perlu dicari melalui sebuah observasi dan setiap daerah memungkinkan untuk memiliki nilai bobot yang berbeda. Fungsi penentuan skor untuk perbandingan juga bisa berbeda beda sesuai dan tidak ada bentuk yang baku. Untuk mengatasi masalah ini digunakan Skyline Query yang diperkenalkan oleh Borsonyi [3] pada tahun 2001. Skyline Query merupakan metode umum untuk menemukan hubungan dominasi antar objek. Algoritma ini menentukan beberapa objek yang

dominan dari kumpulan data[2]. Misalnya obyek A mendominasi obyek B maka semua atribut A lebih baik atau sama dengan obyek B dan setidaknya ada satu atribut objek A lebih baik daripada obyek B. Sebagaimana data pada contoh perhitungannya di Tabel 1, 3 obyek tempat wisata dapat digambarkan dalam bidang kartesian seperti terlihat pada Gambar 1.

Dari gambar tersebut terlihat bahwa obyek wisata Pantai Kuta mendominasi tempat wisata Pantai Pink karena setiap atribut Pantai Kuta lebih baik daripada setiap atribut Pantai Pink. Di sisi lain, Pantai Senggigi tidak didominasi atau mendominasi Pantai Kuta dan Pantai Pink. Sehingga, tidak bisa dikatakan bahwa Pantai Senggigi lebih baik atau lebih buruk dibanding kedua pantai lainnya. Salah satu contoh rekomendasi yang memungkinkan untuk diberikan dari Gambar 1 adalah perlunya untuk meng-upgrade Pantai Pink sehingga tidak didominasi oleh Pantai Kuta, sehingga Pantai Pink memiliki suatu keunggulan dibandingkan Pantai Kuta.



Gambar 1. Contoh Skyline Data

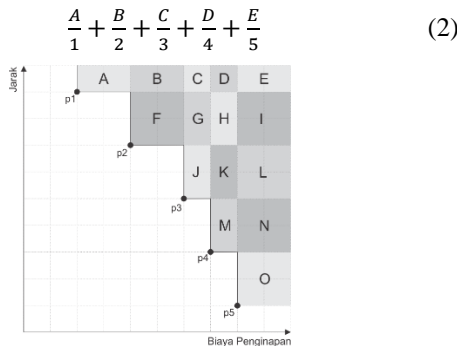
Untuk melakukan perhitungan Skyline Query dengan data yang banyak dan atribut yang banyak dibutuhkan mesin komputasi yang besar[4]. Maka diperlukan suatu komputasi paralel untuk melakukan perhitungan data tersebut. Komputasi paralel merupakan komputasi yang dilakukan oleh banyak komputer sehingga menghemat waktu yang dibutuhkan.

Hadoop Cluster adalah salah satu alternatif yang paling banyak dikembangkan untuk untuk komputasi paralel [5]. Dimana Hadoop Cluster terdiri dari banyak mesin komputer yang disebut Node. Dalam sebuah Hadoop Cluster terdapat beberapa Master Node yang mengkoordinir banyak Slave Node. Pekerjaan yang dilakukan adalah memproses data yang berukuran besar dengan cara memecahnya menjadi tugas yang lebih kecil sehingga dapat dilakukan secara paralel dengan menggunakan framework MapReduce.

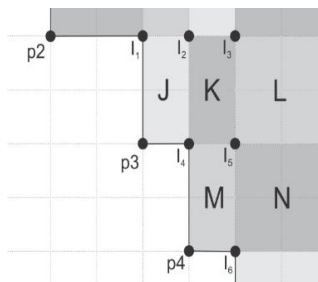
MapReduce adalah sebuah framework yang berjalan diatas computer cluster[6]. Terdapat dua fase dalam framework MapReduce yaitu fase Map dan fase Reduce. Pada fase Map, data dipecah ke dalam beberapa bagian dan setiap bagian proses di sebuah mesin Mapper. Output dari setiap Mapper dikirim ke mesin Reducer untuk disatukan menjadi output final.

2. TINJAUAN PUSTAKA

Pada penelitian yang dilakukan oleh Heri Wijayanto dkk. Pada tahun 2021 yang berjudul "Upgrading Product Based on Existing Dominant Competitor", dalam pengembangan sebuah produk yang dimana memiliki pesaing yang mendominasi peneliti menggunakan metode Top-Down Recursive Depth First Search Algorithm (TDRDFS) yang akan membangun titik –titik dari Dominant Graph of Insersection (DGI) untuk model daerah dominasi[18]. Pada Gambar 2.5 menunjukan daerah yang di dominasi oleh masing- masing produk seperti pada daerah B yang didominasi oleh produk p1 dan p2 sedangkan daerah C di dominasi oleh produk p1, p2, dan p3 begitu juga untuk daerah yang lain. Maka pengguna dalam daerah B akan memilih antara produk p1 dan p2, jadi daerah tersebut akan dibagi sebanyak produk yang didominasi. Jadi untuk potensi produk p1 dipilih adalah:

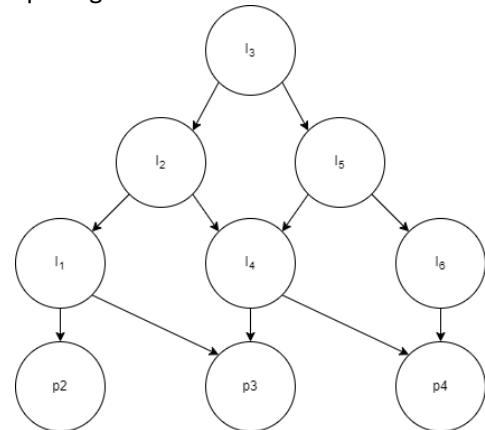


Gambar 2. Contoh wilayah yang di dominasi pada produk



Gambar 3. Contoh intersection pada suatu wilayah dominasi

Dalam graph diatas ada juga yang dinamakan intersection, dimana intersection ini adalah nilai maksimal dari titik yang berpotongan seperti pada Gambar 3 yang menampilkan I1, I2, I3, I4, intersection antar titik. Jadi untuk I1 merupakan perpotongan produk p2 dan p3 maka titik berpotongnya di dapatkan dari titik maksimal p2 dan p3, begitu juga untuk I2 yang titik berpotongnya di dapatkan dari titik maksimal p2, p3, dan p4. Maka pada Gambar 3 didapatkan poin DGI seperti pada gambar 4



Gambar 4. Dominant Graph of Intersection (DGI) poin dari gambar 2.6

Jadi untuk komputasi dengan metode TDRDFS ini sangatlah berat maka dibutuhkan MapReduce untuk membagi pengerjaan ketiap komputer, sehingga dalam pengerjaan menjadi lebih efisien.

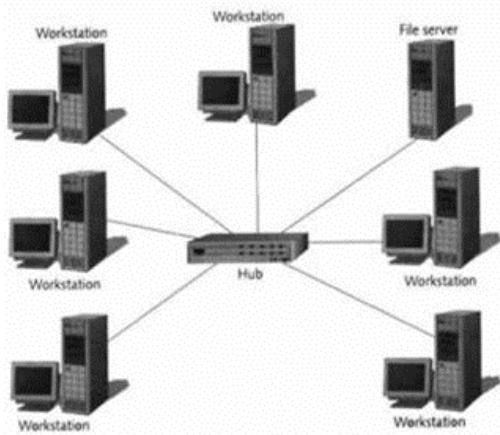
Selanjutnya pada penelitian yang dilakukan oleh Pandu Akas Tachli Taqwin dkk. yang berjudul "Implementasi Metode Mapreduce Pada Big Data Berbasis Hadoop Distributed File System". Dalam pengolahan data yang memiliki volume besar secara realtime maka diperlukan sebuah metode yakni mapreduce, mapreduce tersebut akan digunakan untuk melakukan komputasi data pada Hadoop Distributed File System (HDFS). Disini peneliti membandingkan sebuah komputasi mapreduce yang dijalankan menggunakan Flink dan mapreduce menggunakan Hadoop, peneliti mendapatkan hasil komputasi Hadoop lebih cepat sekitar 37.18% dibandingkan mapreduce menggunakan Flink

2.1. Jaringan Komputer

Jaringan komputer adalah jaringan telekomunikasi yang memungkinkan antar komputer untuk saling berkomunikasi dengan bertukar data. Jaringan komputer dibuat untuk berbagai (sharing) layanan (service). Pihak yang meminta/menerima layanan disebut klien (client) dan yang memberikan/mengirim layanan disebut peladen (server). Desain ini disebut

dengan sistem Client-Server, dan digunakan pada hampir seluruh aplikasi jaringan komputer [7].

Topologi jaringan komputer adalah cara untuk menghubungkan komputer yang satu dengan komputer lainnya dalam sebuah jaringan. Topologi yang saat ini banyak digunakan adalah star. Dalam suatu jaringan komputer, jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu dicermati kelebihan/keuntungan dan kekurangan / kerugian dari masing - masing topologi berdasarkan karakteristiknya[8].

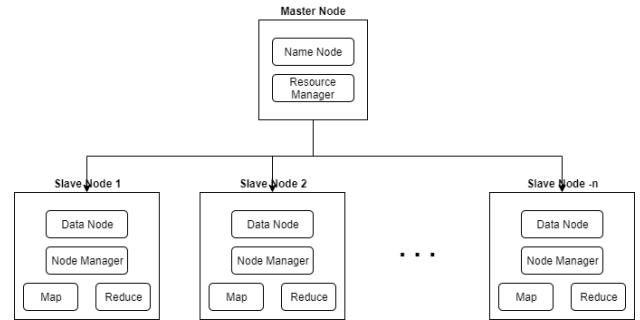


Gambar 5. Topologi Star

2.2. Hadoop Cluster

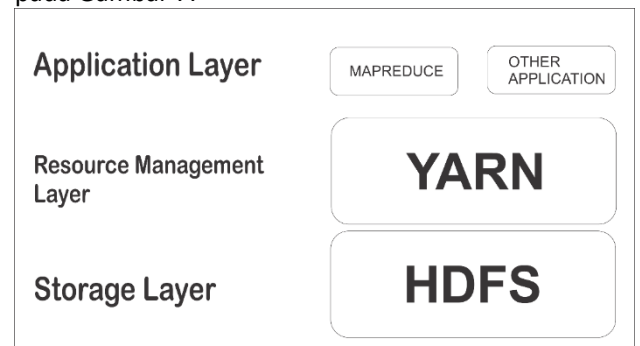
Computer Cluster adalah satu set komputer dalam jaringan yang bekerja bersama-sama untuk memproses suatu data. Salah satu Computer Cluster yang paling populer adalah Hadoop Cluster yang dikembangkan oleh Doug Cutting dan Mike Cafarella dari Apache pada tahun 2005 [10]. Hadoop Cluster dikembangkan berbasis Java dan berfungsi untuk mengolah data skala besar (Big Data). Hadoop memiliki kecepatan pengolahan data berkali-kali lipat dibandingkan dengan metode konvensional dikarenakan Hadoop bekerja dengan prinsip membagi-bagi data berukuran besar menjadi beberapa bagian dan kemudian memproses potongan kecil data tersebut secara paralel.

Walaupun Hadoop Cluster dirancang untuk diimplementasikan dengan banyak node komputer, tetapi untuk keperluan pengembangan Hadoop dapat berjalan secara standalone atau single node. Hadoop Cluster multinode minimal memiliki dua buah node komputer, Komputer pertama digunakan sebagai Master Node dan yang kedua digunakan sebagai slave [6].[11]



Gambar 6. Arsitektur Node pada Hadoop Cluster

Master Node terhubung dengan beberapa Slave Node dan memiliki beberapa komponen seperti Name Node dan Resource manager, sedangkan untuk Slave Node memiliki komponen Data Node, Node manager dan untuk pemrosesannya Slave Node memiliki fungsi Map dan Reduce. Hadoop Cluster memiliki sebuah arsitektur seperti sebuah komputer yakni sebuah penyimpanan dan pemrosesan, pada Hadoop Cluster sistem penyimpanan disebut Hadoop Distributed File System (HDFS). Pada master-node mempunyai Name Node dan untuk slave-node mempunyai Data Node. Untuk sistem pemrosesan pada Hadoop Cluster adalah Yet Another Resource Negotiator (YARN) yang dimana terdiri dari Resource Manager dan Node Manager. Adapun gambaran arsitektur layer dari Hadoop Cluster pada Gambar 7.

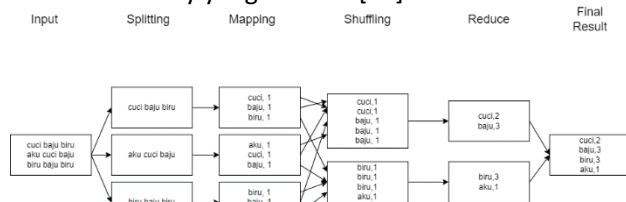


Gambar 7. Arsitektur Layer Hadoop

2.3. MapReduce

MapReduce adalah framework yang digunakan pada aplikasi dan program yang dikenalkan oleh google untuk menjalankan pekerjaan komputasi yang terdistribusi dan dijalankan pada cluster. Konsep yang digunakan dalam Mapreduce adalah fungsi Map dan Reduce untuk functional programming. Operasi map adalah dengan membagi input file secara paralel menjadi beberapa bagian yang dinamakan filesplits. Jika sebuah file tunggal terlalu besar maka akan mempengaruhi waktu membaca sehingga file tersebut dibagi menjadi beberapa bagian. Proses pembagian file tidak mengerti apapun mengenai

struktur logika dari input file, sebagai contoh file berbasis teks bergaris dibagi kedalam beberapa bagian dan dipecah secara byte. Lalu map task akan dibuat per filesplit yang ada. Ketika sebuah reduce task dijalankan, proses reduce tersebut mengambil input yang terbagi kedalam beberapa file dalam semua nodes yang digunakan saat menjalankan map tasks. Setelah semua data yang tersedia secara lokal ditambahkan kedalam satu file pada fase penambahan. File tersebut kemudian digabung dan diurutkan sehingga pasangan key value untuk key tertentu akan bersebelahan. Hal ini membuat operasi reduce yang sebenarnya menjadi sederhana. Kemudian file dibaca secara berurutan dan value dikirim ke proses reduce secara berulang-ulang berdasarkan key yang ditemui [12].



Gambar 8. Contoh Alur MapReduce Untuk Kasus WordCount

Berdasarkan Gambar 8 merupakan sebuah contoh WordCount yang memiliki data inputan teks berupa beberapa kata yang kemudian data tersebut akan di bagi menjadi ukuran yang sama dan menjadi sebuah file yang bernama filesplits. Setelah data tersebut dibagi maka akan dilanjutkan ke mapping yang mengubah filesplits tersebut menjadi data key-value pairs dimana key pada contoh tersebut merupakan setiap kata pada teks tersebut dan value nya merupakan kata –kata seperti “cuci baju biru” maka menghasilkan key-value pairs nya adalah cuci:1, baju:1, dan biru:1. Dalam proses ini terdapat satu atau lebih mesin untuk melakukan hal tersebut untuk filesplits yang lain. Setelah dilakukannya mapping selanjutnya dilakukan proses sorting atau pengurutan agar dapat melakukan agregasi dari value yang memiliki key yang sama dengan mudah pada fase reducer, yang dikarenakan ketika setelah melakukan mapping data dari key-value pairs akan tersebar di setiap mesin. Setelah semua data sudah diurutkan ke mesin yang sama maka fase reducing akan melakukan menjumlahkan nilai value dari key yang sama seperti pada Gambar contoh di atas pada salah satu mesin terdapat data cuci:1 dan cuci:1 maka data tersebut akan di agregasi menjadi cuci:2

2.4. Java

Implementasi Program MapReduce ini dibuat menggunakan bahasa pemrograman java, yang dimana Java adalah nama untuk sekumpulan teknologi untuk membuat atau menjalankan perangkat lunak pada computer standalone ataupun pada lingkungan jaringan. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystem saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaks C dan C++ namun dengan sintaks model objek yang sederhana. Java merupakan bahasa pemrograman yang bersifat umum, dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda[14].

Java berdiri diatas mesin interpreter yang diberi nama Java Virtual Machine (JVM). Platform Java terdiri dari kumpulan library JVM, kelas-kelas loader dipaket dalam sebuah compiler, debugger, dan perangkat lain yang dipaket dalam Java Development Kit (JDK). Untuk menjalankan file bytecode dibutuhkan JRE (Java Runtime Environment) yang memungkinkan pemakai untuk menjalankan program Java, hanya untuk menjalankan, tidak untuk membuat kode baru lagi [15].

3. METODE PENELITIAN

3.1. Alat dan Bahan

Dalam penelitian dibutuhkan beberapa alat dan bahan seperti Hadoop Cluster dan data tentang pariwisata di daerah Lombok yang digunakan untuk mempermudah melakukan penelitian.

1. Hadoop Cluster

Penelitian akan menggunakan Hadoop Cluster yang diinstal secara lokal di Laboratorium Komputer 2 PSTI FT Unram.. Komputer yang akan digunakan secara lokal untuk Hadoop Cluster berjumlah 6 buah komputer dengan spesifikasi:

- a. Processor : Core i3
- b. Ram : 4 Gigabyte
- c. Memori : 500 Gigabyte

Dimana tiap komputer pada spesifikasi diatas akan diinstallkan sistem operasi Linux yang secara virtual, Hadoop Cluster berjalan di sistem operasi Linux. Salah satu komputer akan berperan sebagai Master Node dan sisanya sebagai Slave Node.

2. Data

Secara umum data yang akan diolah adalah data tentang obyek wisata di pulau Lombok. Untuk analisa relasi dominasi antar obyek wisata, atribut yang digunakan dari setiap obyek wisata dapat disusun seperti dalam Tabel II:

TABEL II. TABEL DATA PARIWISATA

Nama Destinasi Wisata	Rating (Google/5)	Jumlah Pengunjung (Bulan 2020)	Jarak dari Bandara (km)	Penghajian disektor				Minimarket disektor		Jarak internet	Jarak Telekom	Pemerintah Kabupaten Terpadat	Kantor Pusat Terpadat	Jumlah Wisata Terpadat
				Jumlah	Rata-Rata Harga(DRp)	Rata-Rata Rating	Rata-Rata Jarak(km)	Jumlah	Rata-rata Jarak(km)					
Pantai Kuta	4.5	18.1	10	590341.4	4.21	0.285	3	0.993333	0.13	1.8	1.7	0.7	0.7	
Pantai Senggigi	4.5	50.7	10	682648.6	4.39	0.601	4	0.5875	0.6	6.4	1.3	0.7	0.7	
Taman Narmada	4.5	28.4	2	109846	4.35	4.75	5	0.234	0.23	3.45	0.75	0.3	0.3	
Taman Puncak Sembel	4.7	75.6	0	338453.8	4.5	4.9	4	4.425	9.4	9.1	10.1	4.3	4.3	
Pantai Tanjung Ane	4.6	21.6	5	418954.4	4.82	1.7	2	6	3.2	9.8	9.7	5.7	5.7	

Untuk data pada penelitian ini hanya menggunakan 21 objek wisata dengan masing- masing memiliki 13 atribut seperti Jumlah pengunjung, Jarak dari Bandara, Rating, Sinyal Komunikasi, Jaminan Keamanan dll., data tersebut merupakan data sekunder yang di dapatkan melalui website <https://data.ntbprov.go.id/>, dan dari <https://google.com/>. Data diatas akan diseleksi terlebih dahulu untuk membedakan daerah wisata NTB dan hanya mengambil beberapa data pariwisata yang ada di Lombok.

3.2. Alur Penelitian

Dalam Subbab ini berisi alur penelitian yang dilakukan peneliti untuk menemukan solusi dari permasalahan yang diangkat dalam penelitian ini. Langkah pertama adalah studi literatur untuk mengumpulkan informasi dari penelitian yang terkait. Kemudian dilanjutkan dengan perancangan sistem dimana peneliti merancang sebuah sistem yang baik untuk kelancaran penelitian berikut.

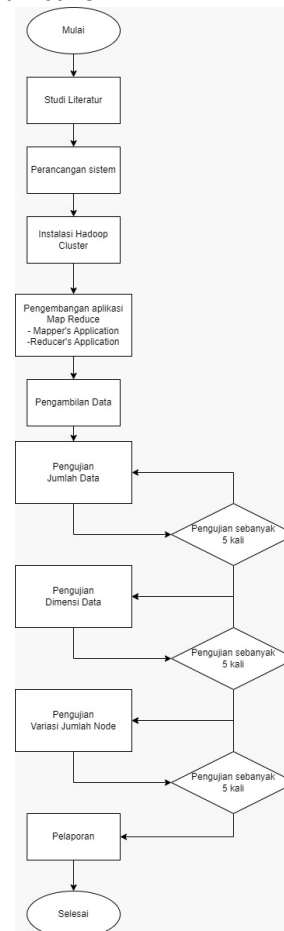
Untuk menghitung relasi dominasi pada lokasi wisata di Pulau Lombok, dibutuhkan langkah-langkah perencanaan sebelum Hadoop dapat benar-benar diimplementasikan. Langkah-langkah tersebut dilakukan untuk mempersiapkan program pengolah data dengan Hadoop dapat dijalankan, dapat menghasilkan suatu keluaran serta dapat dipahami cara kerjanya. Langkah-langkah tersebut adalah sebagai berikut:

1. Melakukan instalasi dan konfigurasi Hadoop pada namenode dan datanode.
2. Mengolah data dari tiap lokasi wisata menjadi bentuk yang lebih mudah untuk diolah lebih lanjut dalam program Hadoop.

3. Membuat program untuk mengolah data mencari suatu rekomendasi pengembangan daerah wisata berdasarkan paradigma mapreduce.

Setelah itu dilakukan instalasi Hadoop pada sistem yang sudah dirancang peneliti sebelumnya. Ketika sudah berhasil melakukan instalasi Hadoop Cluster pada sistem yang dirancang sebelumnya maka selanjutnya dilakukan pengembangan aplikasi Mapreduce yang dimana berisi Mapper's Application dan Reducer's Application.

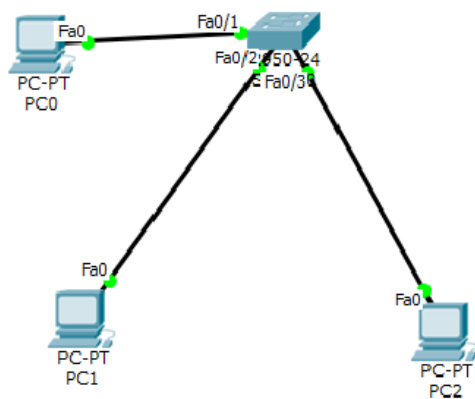
Setelah berhasil mengembangkan sebuah aplikasi MapReduce maka dilakukan pengambilan data atau pengumpulan data pariwisata di daerah Lombok oleh peneliti. Jika data- data tersebut sudah terkumpul dan cukup maka peneliti selanjutnya melakukan pengujian berdasarkan data yang sudah dikumpulkan sebelumnya. Dalam tiap pengujian yang dilakukan peneliti melakukan beberapa perbedaan pada Variasi jumlah node, dimensi data dan jumlah data. Setelah melakukan pengujian selesai maka peneliti akan melaporkan hasil yang didapatkan dari pengujian tersebut. Flowchart dari Alur penelitian tersebut dapat dilihat pada Gambar 9.



Gambar 9. Alur Penelitian

3.3. Perancangan Sistem

Sistem Hadoop Cluster akan diimplementasikan pada Laboratorium 2, Program Studi Teknik Informatika, Universitas Mataram untuk mengembangkan aplikasi MapReduce sebelum diimplementasikan di cloud AWS-EMR. Pada sistem Hadoop Cluster terdapat Resource Manager yang merupakan proses pemilik otoritas paling tinggi yang bertanggung jawab atas tiap Node Manager. Tiap node manager akan melaporkan hasil dari penggunaan sumber daya seperti penggunaan CPU, memori dan jaringan kepada resource manager. Dan juga pada sistem tersebut memiliki namenode dan datanode yang dimana datanode akan mengirimkan informasi ke namenode seperti ID registration, informasi tiap blok dan melaksanakan perintah dari namenode seperti mereplikasi blok atau menghapus blok. Desain fisik sistem hadoop dalam dilihat pada Gambar 10

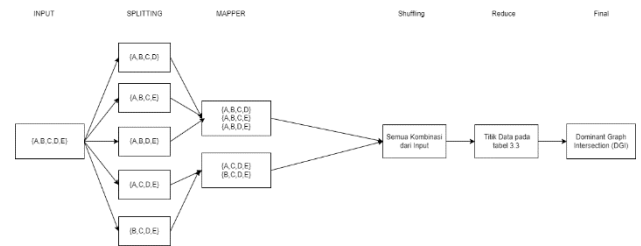


Gambar 10. Desain jaringan implementasi *Hadoop*

3.4. Perancangan Sistem

Untuk Rekomendasi Pengembangan Kepariwisata di Lombok akan menggunakan Mapreduce Framework. Program Mapreduce akan dijalankan pada dua proses secara terpisah. Pada Hadoop proses program Mapreduce dijalankan pada HDFS. Yang pertama client harus lakukan yaitu mendistribusikan data pada HDFS untuk memulai program Mapreduce. Program Mapreduce menggunakan bahasa java yang berisikan implementasi dari fungsi Mapper dan Reducer. Dalam perancangan mapReduce, hal yang dilakukan pertama yakni mengatur Hadoop API dimana berbentuk file XML seperti pada core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml.

Berikut gambaran umum mapreduce untuk mencari DGI :



Gambar 11. Gambaran secara umum mapreduce membentuk DGI

Jadi terdapat sebuah input berupa himpunan yang kemudian di split sebanyak kombinasi $nCr = \frac{n!}{r!(n-r)!}$ dengan r atau objek yang ingin dicari sebanyak $n-1$, jadi untuk rumus kombinasinya didapatkan nC_{n-1} . Kemudian data yang di split tersebut dimasukan ke mapper dimana mapper ini berisi algoritma untuk mencari titik intersection, sampai akhirnya medapatkan semua intersection dari kombinasi subset . Selanjutnya semua intersection subset tersebut di shuffling sehingga semua data tersebut terkumpul, yang kemudian intersection subset tersebut dikirim ke-reducer dimana reducer berisi fungsi untuk menentukan dominasi dari semua intersection subset sehingga menghasilkan Dominant Graph Intersection (DGI).

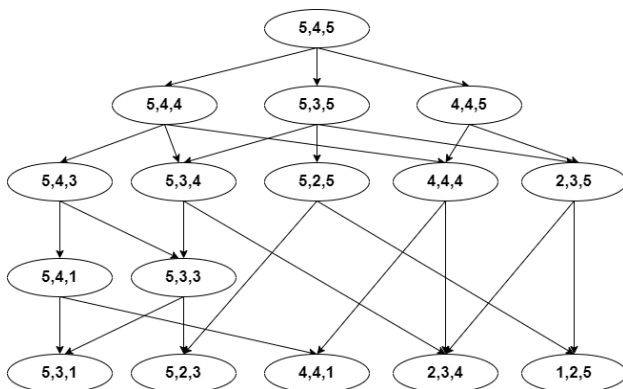
MapReduce nantinya digunakan untuk pencarian DGI yang dimana sebelum membentuk DGI, dilakukan pencarian semua titik intersection dengan algoritma sebagai berikut:

1. Membuat index menggunakan bilangan biner dari 0 hingga $(2^n - 1)$, dimana n adalah jumlah himpunan data
2. Membuat subset sesuai index dari bilangan biner yang sudah dibuat
3. Memasukan subset yang dibuat pada langkah 2 yang jumlah data dari subset tersebut lebih besar atau sama dengan 2 ke subset baru (SN_j).
4. Mencari intersection (Ik) dari SN_j dengan cara mencari nilai maksimum dari tiap SN_j
5. Melakukan penggabungan himpunan Ik dengan himpunan yang dicari diawal

Setelah mendapatkan subset dari bilangan biner, selanjutnya akan dilakukan pembuatan DGI dengan algoritma sebagai berikut :

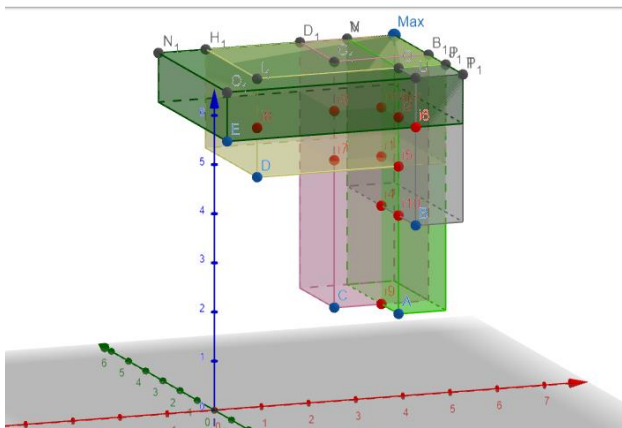
1. Menginisiasi sebuah data (DGI) dengan isian null
2. Membaca tiap subset yang sudah didapatkan sebelumnya dengan bilangan biner (s)

3. Mengecek apakah nilai DGI sama dengan null, jika iya maka lanjut langkah ke-4 dan jika tidak maka lanjut langkah ke-5
4. Menginisiasi nilai DGI = s
5. Melakukan perbandingan data s dengan DGI, jika s mendominasi DGI maka ke langkah 6, jika tidak maka lanjut langkah ke-7
6. Melakukan inisiasi s merupakan children dari DGI
7. Melakukan inisiasi DGI children dari s



Gambar 12. Hasil DGI dari sebuah himpunan titik skyline

Untuk contoh tersebut merupakan sebuah data skyline dengan dimensi 3 yang dimana dapat digambarkan, tapi ketika sudah memasuki 4 dimensi dan selebihnya maka sulit untuk digambarkan. Berikut contoh gambar data skyline dimensi 3



Gambar 13. Hasil 3D Calculator dari himpunan titik skyline

3.5. Pengujian

Setelah langkah-langkah diatas dilakukan, maka program mapreduce dengan Hadoop baru dapat dijalankan sehingga memberikan suatu hasil analisa dari sekumpulan data berukuran besar. Setelah

program Hadoop dapat berjalan dengan lancar, maka dapat dilakukan beberapa perbandingan untuk mengetahui sejauh mana performa Hadoop dalam mengolah data berukuran besar yang diukur berdasarkan kecepatan pemrosesan data serta mendapatkan komposisi Hadoop dengan performa yang baik. Perbandingan-perbandingan tersebut meliputi beberapa aspek yaitu kecepatan, CPU load dan memory usage. Perbandingan yang akan dilakukan adalah sebagai berikut:

1. Membandingkan program Hadoop mapreduce dengan adanya penambahan jumlah node.
2. Membandingkan program Hadoop mapreduce dengan dimensi dan jumlah data.

4. HASIL DAN PEMBAHASAN

4.1. Pengujian

Pada tahap ini pengujian Skyline query akan dilakukan dengan 3 cara yakni pertama dengan melakukan variasi pada jumlah data, yang kedua melakukan variasi pada jumlah dimensi, dan yang ketiga dengan melakukan variasi pada jumlah node. Dalam pengujian ini data, dimensi dan node tertinggi yang akan digunakan secara berturut-turut yakni 21 data, 13 dimensi dan 5 Node.

Jumlah data yang akan di variasikan dalam pengujian pertama yakni 5, 10, 15, 17, 20, dan 21 yang dimana datanya memiliki 13 dimensi sedangkan untuk jumlah node menggunakan 4 buah node.

4.2. Analisis Pengujian dengan Variasi Jumlah Data

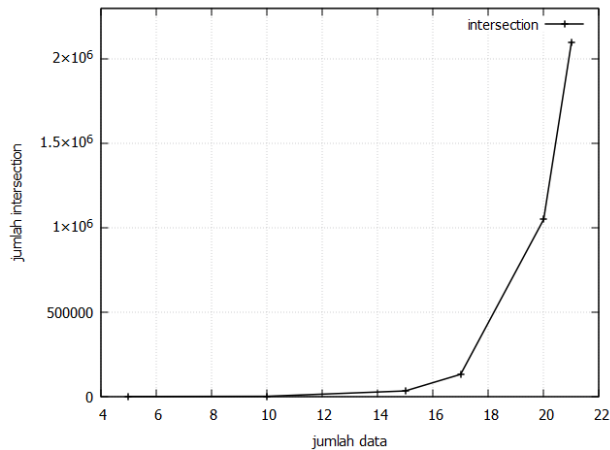
Pada pengujian ini variasi jumlah data yang digunakan yakni 5, 10, 15, 17, 20, dan 21 yang dimana datanya memiliki 13 dimensi sedangkan untuk jumlah node menggunakan 4 buah node. Berikut tabel dari hasil pengujian dengan melakukan variasi jumlah data:

TABEL III. Hasil Pengujian dengan Variasi Jumlah Data

Pengujian Jumlah Dimensi dengan 4 Node 17 Data							
Jumlah Data	Intersection yang dihasilkan				Waktu pengujian (menit)	Rata-Rata waktu pengujian (detik)	
	1	2	3	4			
5	32	0,61	0,76	0,63	0,66	0,6	39,12
10	1024	0,76	0,61	0,61	0,73	0,85	42,72
15	32768	0,66	0,88	0,76	0,78	0,85	47,16
17	131072	1,26	1,15	1	0,95	0,88	62,88
20	1048576	7,97	10,03	11,4	8,28	7,93	547,32
21	2097152	24,7	24,83	24,8	18,9	18,2	1337,16

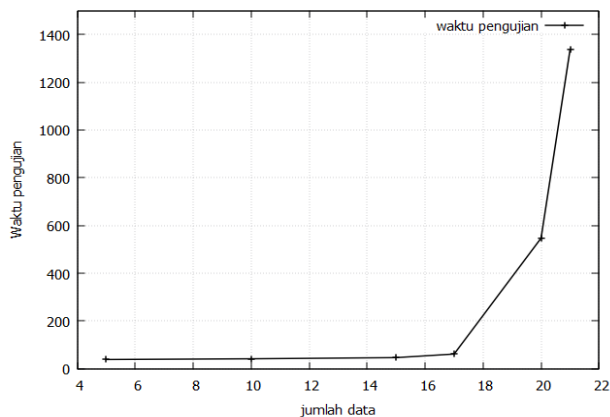
Tabel III merupakan hasil dari pengujian dengan melakukan variasi jumlah data yang dimana akan

menghasilkan jumlah intersection untuk tiap data dan dilakukan pengujian sebanyak 5 kali untuk menghindari kesalahan sistem dalam pengujian, sehingga mendapatkan rata-rata dalam tiap pengujiannya. Hasil terendah dalam pengujian ini adalah 39,12 detik atau 0,652 menit yang dimana jumlah data yang digunakan sebanyak 5 data, kemudian untuk hasil tertinggi dalam pengujian ini adalah 1337,16 detik atau 22,286 menit yang dimana jumlah data yang digunakan sebanyak 21 data.



Gambar 14. Grafik intersection yang dihasilkan dari sejumlah data

Gambar 14 menunjukkan grafik intersection yang dihasilkan dari sejumlah data yang mengalami kenaikan, dimana kenaikan tersebut terlihat tinggi atau sudah menyentuh angka jutaan saat data yang diuji sebanyak 20 data yang terlihat pada grafik berada pada garis 1000000 dan 1500000, yang kemudian untuk data pada saat menguji 21 data intersection berada pada garis 2000000 dan 2500000. Oleh karena itu data tersebut terlihat naik secara menanjak atau sangat naik dikarenakan intersection yang didapatkan adalah 2 kali lipat dari data sebelumnya, atau 2 pangkat dari data yang diuji.



Gambar 15. Grafik hasil pengujian dengan variasi jumlah data

Pada gambar 4.2 grafik menunjukkan peningkatan jumlah waktu yang diperlukan berbanding lurus untuk tiap kenaikan jumlah data yang dimana untuk 5 hingga 17 data terlihat signifikan dalam peningkatannya, sedangkan untuk jumlah data 20 dan 21 terlihat waktu yang diperlukan begitu tinggi hingga mencapai 1337,16 detik atau 22,286 menit.

Kenaikan hasil pada grafik 4.2 dikarenakan kenaikan pada jumlah intersection yang ada pada grafik 4.1, kenaikan pada jumlah intersection terlihat jelas ketika pengujian dengan 17 data hingga 21 data karena untuk mencari jumlah intersection dilakukan dengan mencari 2 pangkat dari jumlah data tersebut. Sehingga grafik dari data tersebut tidak mengalami kenaikan secara linear tetapi kenaikan yang tinggi. Hal tersebut menunjukkan pada grafik 4.1 dan 4.2 memiliki pengaruh yang sama dimana semakin naik datanya maka intersection semakin tinggi dan juga waktu yang diperlukan semakin banyak.

4.3. Analisis Pengujian dengan Variasi Jumlah Dimensi

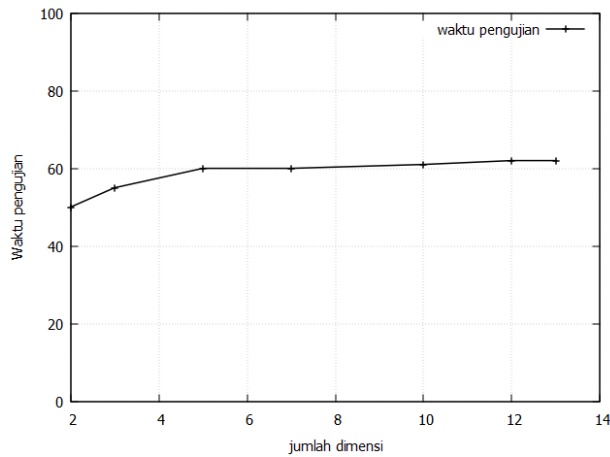
Pada pengujian ini variasi jumlah dimensi yang digunakan yakni 2, 3, 5, 7, 10, 12, dan 13 yang dimana menggunakan 17 buah data sedangkan untuk jumlah node yang digunakan sebanyak 4 buah node. Berikut tabel dari hasil pengujian dengan melakukan variasi jumlah data:

TABEL IV. Hasil Pengujian dengan Variasi Jumlah Dimensi

Pengujian Jumlah Dimensi dengan 4 Node 17 Data							
Jumlah Dimensi	Waktu pengujian (menit)					Rata-Rata waktu pengujian	Rata-Rata waktu pengujian (detik)
	1	2	3	4	5		
2	0,84	0,8	0,86	0,83	0,91	0,848	50,88
3	0,91	0,95	0,89	0,85	1	0,92	55,2
5	0,96	1	1,03	1,05	0,96	1	60
7	0,91	1,06	0,94	1,11	1,06	1,016	60,96
10	0,94	1,01	0,98	1,19	0,98	1,02	61,2
12	1,03	0,98	1,01	1,15	1,03	1,04	62,4
13	1,26	1,15	1	0,95	0,88	1,048	62,88

Tabel IV merupakan hasil dari pengujian dengan melakukan variasi jumlah dimensi yang dimana untuk tiap data dan dilakukan pengujian sebanyak 5 kali untuk menghindari kesalahan sistem dalam pengujian, sehingga mendapatkan rata-rata dalam tiap pengujiannya. Hasil terendah dalam pengujian ini adalah 50,88 detik atau 0,848 menit yang dimana jumlah dimensi yang digunakan sebanyak 2 dimensi, kemudian untuk hasil tertinggi dalam pengujian ini

adalah 62,88 detik atau 1,048 menit yang dimana jumlah dimensi yang digunakan sebanyak 13 dimensi.



Gambar 16. Grafik hasil pengujian dengan variasi jumlah dimensi

Pada gambar 16 grafik menunjukkan peningkatan jumlah waktu yang diperlukan berbanding lurus untuk tiap kenaikan jumlah dimensi yang dimana untuk peningkatannya tidak terlalu signifikan. Untuk rata-rata peningkatan waktu yang didapatkan yakni 4,21%, dimana waktu peningkatan tertinggi yakni dari 3 dimensi ke 5 dimensi yakni sebesar 8,69% sedangkan untuk peningkatan terendah yakni pada 7 dimensi ke 10 dimensi yang dimana mengalami peningkatan sebesar 0,39%.

Grafik di atas mengalami kenaikan secara linear atau tidak terlalu mengalami kenaikan, dikarenakan untuk pengolahan data dilakukan pada masing-masing slave sehingga pada tiap slave mengolah puluhan ribu intersection untuk tiap slavenya dengan ukuran data yang sama. Dan tiap slave tersebut melakukan proses yang sama yaitu melakukan skyline query.

4.4. Analisis Pengujian dengan Variasi Jumlah Node

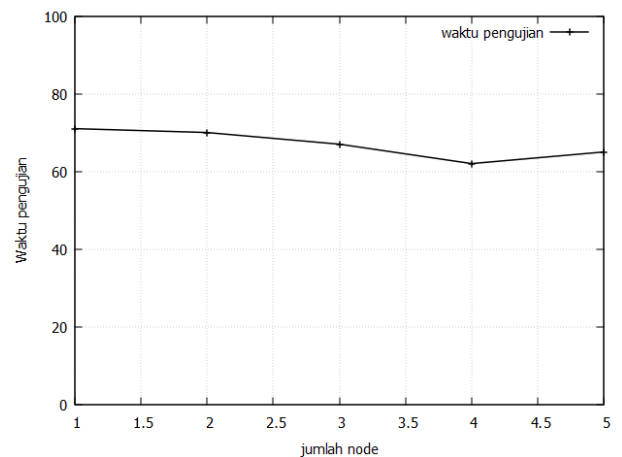
Pada pengujian ini variasi jumlah Node yang digunakan yakni 1 sampai 5 Node yang dimana menggunakan 17 buah data sedangkan untuk jumlah dimensi yang digunakan sebanyak 13 buah dimensi. Berikut tabel dari hasil pengujian dengan melakukan variasi jumlah data:

TABEL V. Hasil Pengujian dengan Variasi Jumlah Node

Pengujian Jumlah Dimensi dengan 17 Data 13 Dimensi							
Jumlah Node	Waktu pengujian (menit)					Rata-Rata waktu pengujian	Rata-Rata waktu pengujian (detik)
	1	2	3	4	5		
1	1,4	1,18	1,03	1,15	1,18	1,188	71,28

2	1,6	1,11	1,05	1,03	1,08	1,174	70,44
3	1,48	1,08	1,05	0,95	1,05	1,122	67,32
4	1,26	1,15	1	0,95	0,88	1,048	62,88
5	1,31	1,08	0,98	0,95	1,1	1,084	65,04

Tabel V merupakan hasil dari pengujian dengan melakukan variasi jumlah node yang dimana untuk tiap data dan dilakukan pengujian sebanyak 5 kali untuk menghindari kesalahan sistem dalam pengujian, sehingga mendapatkan rata-rata dalam tiap pengujiannya. Hasil terendah dalam pengujian ini adalah 62,88 detik atau 1,048 menit yang dimana jumlah node yang digunakan sebanyak 4 node, kemudian untuk hasil tertinggi dalam pengujian ini adalah 65,04 detik atau 1,084 menit yang dimana jumlah node yang digunakan sebanyak 5 node. Jadi untuk pengujian dengan penambahan jumlah node tidak memungkinkan waktu yang dibutuhkan semakin sedikit, oleh sebab itu untuk pengujian kali ini node yang paling membutuhkan waktu lebih sedikit yakni saat menggunakan 4 buah node.



Gambar 17. Grafik hasil pengujian dengan variasi jumlah node

Pada gambar 17 grafik menunjukkan penurunan waktu yang diperlukan saat menggunakan 1 node hingga 4 node, dan untuk selanjutnya node 5 mengalami kenaikan waktu yang diperlukan sehingga dapat disimpulkan untuk pengujian ini jumlah node yang optimum digunakan yakni 4 buah node. Alasan didapatkan nilai optimum pada 4 buah node dikarenakan tidak adanya overhead communication (OC) atau komunikasi yang berlebih antar node, sedangkan saat menggunakan 5 node terjadi OC sehingga ketika menggunakan 5 node waktu yang diperlukan lebih lama dari saat menggunakan 4 node. Bagian ini berisi hasil dan pembahasan penelitian. Hasil penelitian disajikan bentuk tabel atau grafik yang selanjutnya diberikan deskripsi dan pembahasan atas fakta yang diperoleh dikaitkan teori pendukung

penelitian dan atau dibandingkan dengan hasil penelitian yang sangat terkait lainnya.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian komputasi paralel untuk perhitungan relasi dominasi menggunakan skyline query pada lokasi wisata di Pulau Lombok dapat disimpulkan sebagai berikut:

1. Hasil dari pengujian dengan melakukan variasi jumlah data yakni semakin banyak data yang digunakan maka semakin banyak juga waktu yang diperlukan, ketika data yang digunakan 20 maka akan dihasilkan intersection sebanyak 1048576 sedangkan ketika naik 1 data menjadi 21 data maka dihasilkan 2097152 dimana merupakan 2 kali lipat dari jumlah intersection sebelumnya dan dibutuhkan waktu pengujian yang lebih lama lagi dengan peningkatan waktu sebanyak 144,31% dari 9,12 menit waktu pengujian dengan 20 data.
2. Hasil dari pengujian dengan melakukan variasi jumlah dimensi yakni menunjukkan kenaikan waktu pengujian yang tidak terlalu signifikan sehingga perbedaan waktu rata-rata mengalami kenaikan 6,26%.
3. Hasil dari pengujian dengan melakukan variasi jumlah Node yakni hasil dari pengujian yang terbaik yakni ketika menggunakan 4 buah Node dengan waktu 1,04 menit, yang menunjukkan jumlah node yang baik tidak selalu dengan jumlah node yang paling banyak.

5.2. Saran

Berdasarkan hasil penelitian komputasi paralel untuk perhitungan relasi dominasi menggunakan skyline query pada lokasi wisata di Pulau Lombok diperlukan beberapa saran untuk menyempurnakan penelitian ini yakni sebagai berikut:

1. Untuk mendapatkan hasil yang lebih baik sebaiknya menggunakan komputer host dengan sistem operasi Linux yang dikarenakan untuk tiap Node yang digunakan dalam penelitian ini menggunakan virtual mesin yaitu VirtualBox yang dimana dijalankan dalam host sistem Windows.
2. Dalam penelitian ini spesifikasi tiap virtual mesin yang digunakan yakni core yang digunakan sebanyak 2 sedangkan untuk RAM yang digunakan

sebanyak 2 GB untuk tiap node nya, jadi sebaiknya meningkatkan jumlah core dan RAM agar hasil pengujiannya menjadi lebih baik lagi.

3. Dalam pengujian ini masih menggunakan komputer fisik yang dimana harus diakses melalui Laboratorium Teknik Informatika Universitas Mataram, yang sebaiknya dapat dilakukan dengan menggunakan Private Server yang sudah disediakan oleh beberapa website seperti Amazon dan Google.
4. Pengujian ini menggunakan Apache Hadoop yang perlu dilakukan instalasi untuk menjalankannya, dan pengaturan dalam hadoop dilakukan secara manual dengan melakukan edit untuk tiap konfigurasinya, jadi agar memudahkan pengujian selanjutnya dapat menggunakan sistem yang sudah disediakan seperti Ambari dan Cloudera yang dimana masing-masing dari sistem tersebut akan mempermudah dalam pengujian, tetapi dibutuhkan komputer dengan spesifikasi tinggi untuk menjalankannya.

UCAPAN TERIMA KASIH

Ucapan terima kasih dapat diberikan kepada penyandang dana penelitian dan orang yang memberikan kontribusi ilmiah pada penelitian namun bukan merupakan penulis artikel ini.

DAFTAR PUSTAKA

- [1] G. Das, D. Gunopulos, N. Koudas, and D. Tsirogiannis, "Answering Top-k Queries Using Views," 2006.
- [2] M. R. Diansyah, Annisa, and W. A. Kusuma, "Analysis using top-k skyline query of protein-protein interaction reveals alpha-synuclein as the most important protein in Parkinson s disease," *Indonesian Journal of Biotechnology*, vol. 26, no. 4, pp. 197–205, Dec. 2021, doi: 10.22146/ijbiotech.63023.
- [3] S. Borzsonyi, D. Kossmann², and K. Stocker', "The Skyline Operator," 2001.
- [4] I. P. A. P. Wibawa, I. D. Giriantari, and M. Sudarma, "Komputasi Paralel Menggunakan Model Message Passing Pada SIM RS (Sistem Informasi Manajemen Rumah Sakit)," *Majalah Ilmiah Teknologi Elektro*, vol. 17, no. 3, p. 439, Dec. 2018, doi: 10.24843/mite.2018.v17i03.p20.
- [5] S. Prabowo and M. Abdurrohman, "Implementasi Algoritma Penjadwalan untuk pengelolaan Big

- Data dengan Hadoop,” vol. 2, no. 2, 2017, doi: 10.21108/indojc.2017.22.164.
- [6] Fakultas Ilmu Komputer (FILKOM), *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 3. 2016.
- [7] INDAH KUSUMA ASTUTI, “JARINGAN KOMPUTER.”
- [8] A. Supriyadi, D. Gartina, F. Komputer, and S. Badan Litbang, “MEMILIH TOPOLOGI JARINGAN DAN HARDWARE DALAM DESAIN SEBUAH JARINGAN KOMPUTER Recognizing Topology And Network Hardware, Computer Network,” 2007.
- [9] I. Nur Aziz, K. W. Rahmat Saleh, and K. Kunci, “Analisis Pengolahan Text File pada Hadoop Cluster dengan Memperhatikan Kapasitas Random Access Memory (RAM),” *Agustus*, vol. 2, no. 2, p. 6824, 2015.
- [10] W. Tow, *Hadoop: The Definitive Guide*. 2012.
- [11] K. Basuki, H. Novianus Palit, and L. P. Dewi, “Implementasi Hadoop: Studi Kasus Pengolahan Data Peminjaman Perpustakaan Universitas Kristen Petra.”
- [12] D. Kotrakona, “Amazon Web Services,” 2011.
- [13] K. Aryasa and Y. T. Paulus, “Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java,” *Creative Information Technology Journal*, vol. 1, no. 1, p. 57, 2015, doi: 10.24076/citec.2013v1i1.10.
- [14] K. Wijaya *et al.*, “Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Java (Netbeans 7.3),” *Jurnal SISFOKOM*, vol. 08, 2019.
- [15] D. A. Diartono, “Sistem Pendukung Keputusan sebagai Alat Bantu Manager,” *Jurnal Teknologi Informasi DINAMIK*, vol. Volume XI, no. No. 1, 2006.
- [16] T. Limbong *et al.*, *Sistem Pendukung Keputusan: Metode & Implementasi*. 2020. [Online]. Available: [https://books.google.co.id/books?id=6FnYDwAAQBAJ&lpg=PR5&ots=XeM-hCRLIO&dq=sistem pendukung keputusan&hl=id&pg=PR5#v=onepage&q=sistem pendukung keputusan&f=false](https://books.google.co.id/books?id=6FnYDwAAQBAJ&lpg=PR5&ots=XeM-hCRLIO&dq=sistem%20pendukung%20keputusan&hl=id&pg=PR5#v=onepage&q=sistem%20pendukung%20keputusan&f=false)
- [17] H. Wijayanto, Syauki A. Thamrin, and Arbee L.P. Chen, *53rd Hawaii International Conference on System Sciences (HICCS)*, online, January 5-8, 2021. University of Hawai’i at Manoa, 2021.
- [18] H. Wijayanto, W. Wang, W.-S. Ku, and A. L. P. Chen, “LShape Partitioning: Parallel Skyline Query Processing using MapReduce Journal: Transactions on Knowledge and Data Engineering Transactions on Knowledge and Data Engineering LShape Partitioning: Parallel Skyline Query Processing using MapReduce,” 2019.