

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis Kebutuhan

Pada tahapan Analisis kebutuhan ini akan menjelaskan tentang analisa kebutuhan-kebutuhan apa saja yang akan digunakan dalam penelitian ini. Analisa dilakukan untuk mengetahui apa saja *software* yang akan digunakan dan apa saja *hardware* yang akan digunakan.

4.1.1 Analisis Kebutuhan Software

Tahapan analisis kebutuhan *software* dilakukan untuk menentukan *software* apa saja yang akan digunakan. Setelah melakukan Analisa maka dapat ditentukan *software* yang sesuai dengan kebutuhan umum dalam perancangan dashboard monitoring Hadoop menggunakan Grafana. Kebutuhan *software* yang akan digunakan adalah:

1. Sistem Operasi Ubuntu
2. Docker
3. Prometheus
4. Python
5. Hadoop
6. Grafana
7. Whimsical

4.1.2 Analisis Kebutuhan Hardware

Tahapan analisis kebutuhan *hardware* dilakukan untuk menentukan *hardware* apa saja yang akan digunakan. Setelah melakukan Analisa maka dapat menentukan apa saja *hardware* yang sesuai dengan dengan standarisasi minimum spesifikasi *software* yang digunakan yaitu Hadoop dan Grafana. Rekomendasi minimum *hardware* untuk menjalankan Hadoop yaitu:

1. Processor @2GHz
2. RAM 4 GB
3. Storage 25GB

Kemudian rekomendasi minimum *hardware* untuk menjalankan Grafana yaitu:

1. Processor @ 2 GHz
2. RAM 2GB
3. Storage 1GB

Standar minimum diatas harus dapat terpenuhi agar dapat menjalankan *software* Hadoop dan Grafana secara lancar. Jika spesifikasi yang digunakan lebih tinggi maka akan semakin nyaman untuk mengoprasikannya. Oleh sebab itu peneliti harus mengikuti spesifikasi yang sama dengan standar atau lebih tinggi agar *software* dapat digunakan dengan baik. Spesifikasi komputer yang digunakan yaitu:

1. Processor Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
2. Installed RAM 12,0 GB (11,9 GB usable)
3. System type 64-bit operating system, x64-based processor
4. Storage 475 GB (381 GB usable)

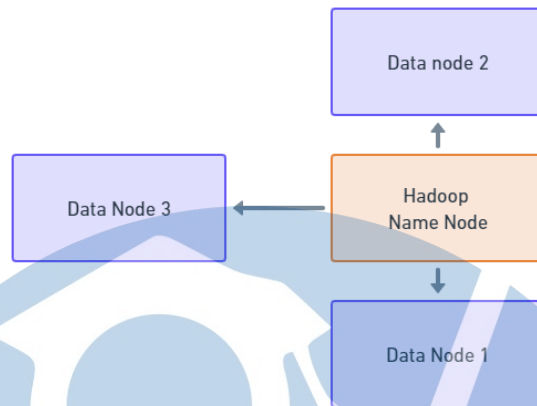
4.2 Rancangan Sistem

Setelah melakukan analisis kebutuhan dilanjutkan dengan rancangan *system* pada perancangan dashboard monitoring Hadoop menggunakan Grafana. Pada tahapan ini akan menjelaskan tentang rancangan arsitektur Hadoop, Prometheus, dan juga Grafana.

4.2.1 Arsitektur Hadoop

Pada rancangan Arsitektur Hadoop ini akan dijelaskan bagaimana rancangan Hadoop akan digunakan. Pada penelitian ini penulis akan membuat rancangan Arsitektur Hadoop yang terdiri dari satu Namenode dan juga 3 Datanode. Dimana Namenode akan berfungsi sebagai tempat penyimpanan, dan juga mengatur data yang tersimpan pada datanode-datanode yang tersebar dalam cluster. Sedangkan fungsi dari datanode yaitu menyimpan data-data yang dikirimkan kepadanya seperti configure capacity, present capacity, DFS

used, DFS remains, dan non DFS used, serta melaporkannya secara berkala kepada Namenode.

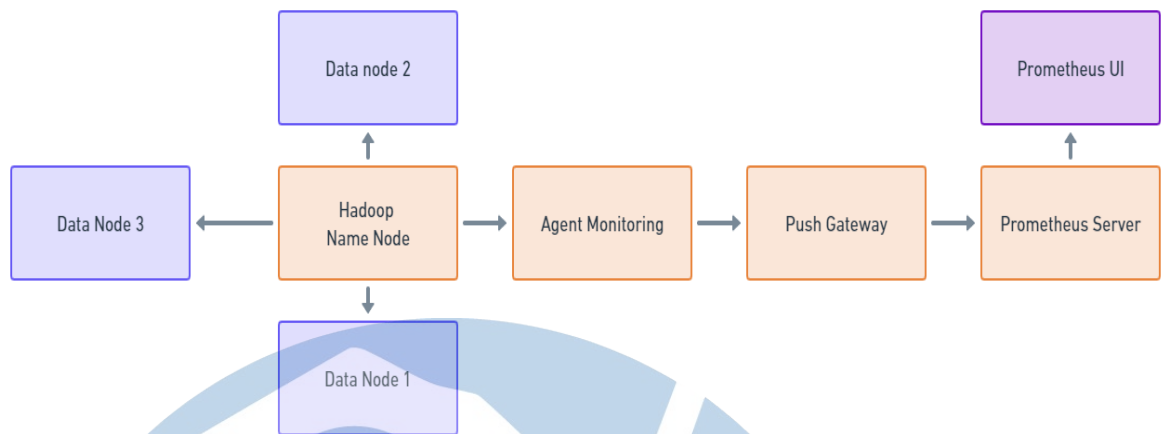


Gambar 4. 1 Arsitektur Hadoop

Pada gambar diatas adalah rancangan arsitektur untuk sistem Hadoop yang akan di monitoring. untuk membuat arsitektur tersebut penulis menggunakan satu komputer fisik (*Localhost*) yang bertugas sebagai Namenode dan juga membuat beberapa Datanode menggunakan Docker Container sebagai data bawaan.

4.2.2 Arsitektur Prometheus

Pada rancangan arsitektur Prometheus ini akan menjelaskan bagaimana rancangan Prometheus yang akan digunakan. Prometheus sendiri berfungsi untuk alat pemantauan data metrik yang disimpan dalam bentuk *time series*, Prometheus memberikan kebebasan untuk menerima kostumisasi metrik, sehingga memungkinkan untuk menerima metrik yang dimiliki Hadoop. untuk menerima metrik-metrik yang dimiliki oleh Hadoop HDFS maka akan dibuat arsitektur sebagai berikut :

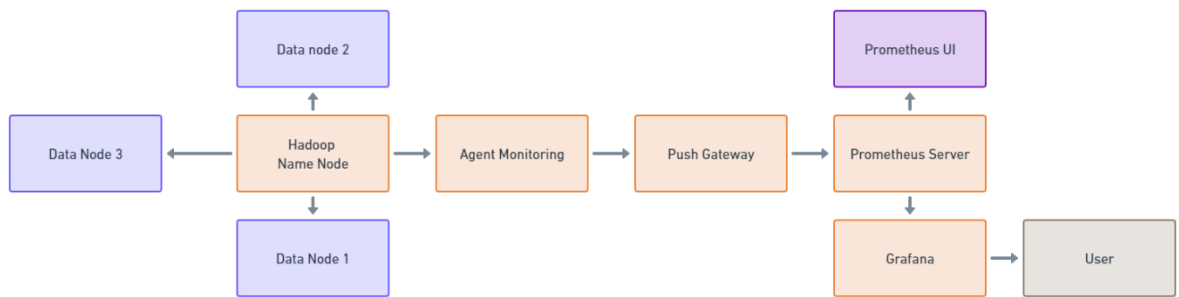


Gambar 4. 2 Arsitektur Prometheus

Pada gambar diatas dapat diketahui bahwa untuk mendapatkan metrik yang dimiliki oleh Hadoop memerlukan sebuah agent monitoring (*web scrping*). didalam agent monitoring (*web scrping*) sendiri harus membuat sebuah script yang akan mengirimkan marik dari Hadoop kepada push gateway yang nantinya akan dikirimkan (*Push*) kepada Prometheus, sehingga metrik dapat terbaca oleh Prometheus.

4.2.3 Arsitektur Grafana

Pada rancangan arsitektur Grafana ini akan menjelaskan bagaimana rancangan Grafana yang akan digunakan. Grafana itu sendiri merupakan sebuah *software* yang digunakan untuk memonitoring dan menganalisa metrik. Matrik-metrik yang akan dimonitorig didapatkan dari sebuah sumber data. Sumber data pada Grafana sangat beragam namum pada penelitian ini yang akan digunakan adalah Prometheus sebagai sumber data. Untuk dapat menampilkan metrik dari Prometheus pada Grafana maka akan dibuat arsitektur sebagai berikut:



Gambar 4. 3 Arsitektur Grafana

Berdasarkan gambar diatas dapat diketahui bahwa untuk mendapatkan matirk yang dimiliki oleh Prometheus, harus menjadikan Prometheus sebagai sumber data. Matrik-metriik yang akan dimonitoring berasal dari metriik-metriik Prometheus itu sendiri dan juga metriik-metriik yang sudah di kirim ke Prometheus dari Hadoop menggunakan Prometheus Pushgateway.

Setelah mengumpulkan metriik-metriik yang akan ditampilkan pada Grafana maka selanjutnya akan dilakukan pembuatan dashboard yang berisikan metriik-metriik Hadoop. Untuk memperbaharui metriik yang di tampilkan maka harus dilakukan pengiriman (*Push*) secara berkala.

4.3 Perancangan Pengujian

Rancangan pengujian pada perancangan dashboard monitoring Hadoop menggunakan Grafana yaitu dengan menggunakan metode *black box testing*, dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, penguji tidak harus memiliki kemampuan menulis kode program. Pengujian dilakukan untuk menemukan masalah atau bug pada aplikasi. Hal ini sangat penting guna memastikan aplikasi yang dibuat benar-benar sudah siap dipublikasikan.

4.3.1 Pengujian Arsitektur

Pada tahap ini akan menjelaskan pengujian arsitektur perancangan dashboard monitoring Hadoop menggunakan Grafana. Berikut tabel yang akan menjelaskan tentang pengujian arsitektur:

Table 4. 1 Rancangan Pengujian Arsitektur

No	Pengujian	Deskripsi pengujian	Hasil yang diharapkan	Hasil Uji
1	Membuat Datanode	Membuat tiga datanode aktif dengan menggunakan docker	Berhasil membuat tiga datanode aktif	Berhasil/Gagal
2	Mengatur target pada Prometheus	Merubah target pada Prometheus agar dapat mengakses Prometheus pushgateway	Berhasil mengatur target Prometheus dengan pushgateway	Berhasil/Gagal
3	Membuat koneksi	Membuat docker container Prometheus, Prometheus Pusgateway, dan grafana	Berhasil menjalankan Prometheus, Prometheus push gateway, dan Grafana	Berhasil/Gagal
4	Membuat script	Membuat script python untuk menarik data Hadoop	Berhasil menarik data Hadoop	Berhasil/Gagal
5	Mengumpulkan data	Mengumpulkan data Hadoop yang akan di gunakan	Berhasil mengumpulkan data Hadoop	Berhasil/Gagal
6	Mengirim data ke Prometheus	Mengirim data Hadoop dengan	Berhasil mengirim data Hadoop	Berhasil/Gagal

		Prometheus push Gateway		
7	Menampilkan data pada Prometheus	Menampilkan metrik Hadoop pada Prometheus	Berhasil menampilkan data Hadoop pada Grafana	Berhasil/Gagal
8	Menghubungkan ke Grafana	Membuat sumber data Prometheus pada Grafana	Berhasil menghubungkan Prometheus dengan Grafana	Berhasil/Gagal
9	Menerima data dari Prometheus	Menampilkan data yang terdapat pada Prometheus	Berhasil menampilkan data Prometheus	Berhasil/Gagal
10	Menampilkan data	Menampilkan data metrik Hadoop pada Grafana	Berhasil membuat dashboard monitoring Hadoop dengan Grafana	Berhasil/Gagal

4.3.2 Pengujian monitoring

Pada tahap ini akan menjelaskan pengujian perancangan dashboard monitoring

Hadoop menggunakan Grafana. Berikut tabel yang akan menjelaskan tentang pengujian monitoring:

STT - NF

Table 4. 2 Rencana Pengujian Monitoring

NO	Pengujian	Deskripsi pengujian	Hasil yang diharapkan	Hasil uji
1	Live Namenode	Menampilkan metrik Live Namenode pada Grafana	Berhasil menampilkan Live Namenode	Berhasil/Gagal
2	Live Datanode	Menampilkan metrik Live Datanode pada Grafana	Berhasil menampilkan Live Datanode	Berhasil/Gagal
			Berhasil menampilkan Dead Datanode	
3	Configure capacity Total	Menampilkan metrik Configure Capacity total pada Grafana	Berhasil menampilkan Configure Capacity Total	Berhasil/Gagal
4	Configure capacity per Node	Menampilkan metrik Configure Capacity per Node pada Grafana	Berhasil menampilkan Configure Capacity per Node	Berhasil/Gagal
5	Present Capacity	Menampilkan metrik Present Capacity pada Grafana	Berhasil menampilkan Present Capacity	Berhasil/Gagal

6	DFS Used Total	Menampilkan Matrik DFS Used Total pada Grafana	Berhasil menampilkan DFS Used Total	Berhasil/Gagal
7	DSF Used per Node	Meanampilkan metrik DFS Used per Node pada Grafana	Berhasil menampilkan DFS Used per Node	Berhasil/Gagal
8	DFS Remains Total	Menampilkan metrik DFS Remains Total pada Grafana	Berhasil menampilkan DFS Remains Total	Berhasil/Gagal
9	DFS Remains per Node	Menampilkan Matrik DFS Remains per Node pada Grafana	Berhasil menampilkan DFS Remains per Node	Berhasil/Gagal
10	Non DFS Used	Menampilkan metrik NON DFS Used pada Grafana	Berhasil menampilkan NON DFS Used	Berhasil/Gagal

STT - NF