

BAB 5 HASIL DAN ANALISIS

Pada bab ini memuat tentang hasil dan analisis terhadap skenario pengujian yang telah dilakukan pada bab sebelumnya. Hasil pengujian ini didapat dari tiga skenario pengujian yang berbeda-beda berdasarkan variasi ukuran data, variasi jumlah *job*, dan variasi jenis *job*. Dalam hasil pengujian ini akan membandingkan kinerja dari algoritme *Fair Share Scheduling* dan *Capacity Scheduling* berdasarkan grafik nilai dari parameter pengujian *Job Fail Rate*, *Latency*, dan *Throughput*.

5.1 Hasil

Untuk mendapatkan hasil dari pengujian yang telah dilakukan, dapat dilakukan dengan melakukan perhitungan terhadap parameter pengujian yang digunakan yaitu, *Job Fail Rate*, *Latency*, dan *Throughput*.

Untuk melakukan pengukuran pada parameter *Job Fail Rate* adalah untuk mengetahui seberapa besar *job* mengalami kegagalan pada saat melakukan running program. *Job* yang mengalami kegagalan tersebut dapat diulang kembali prosesnya. Pada saat pengulangan proses tersebut terdapat jeda waktu sampai proses tersebut dikerjakan kembali yang disebut dengan nilai *timeout*. Untuk melihat tabel data hasil pengujian secara lengkap terdapat pada lampiran B. Berikut ini pada gambar 5.1 merupakan nilai *failed* pada saat melakukan running program.

Untuk menghitung nilai dari parameter *Job Fail Rate* dengan menggunakan rumus sebagai berikut (Yang XIA, 2011) :

$$Fail\ rate = \frac{\lambda_1 \times timeout + \lambda_2 \times error}{sum}$$

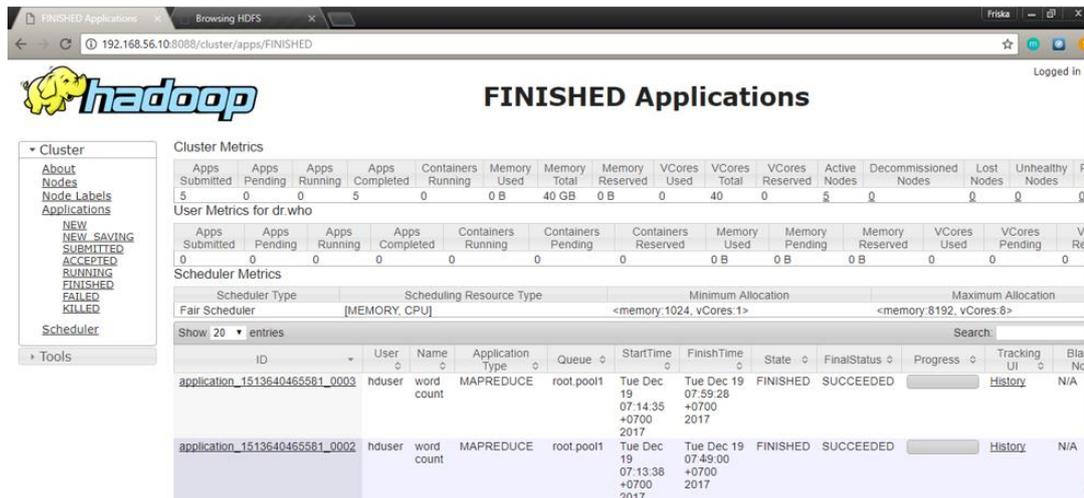
Dimana λ_1 merupakan bobot dari *timeout* dengan nilai 0,5. *Timeout* merupakan jeda waktu dimana *job* mengalami kegagalan. λ_2 merupakan bobot dari *error* dengan nilai 1. *Error* merupakan jumlah *job* yang gagal dan diulang kembali. Sedangkan *sum* merupakan jumlah keseluruhan dari *job* yang telah dijalankan.

Untuk melakukan pengukuran pada parameter *Latency* adalah untuk mengukur rata-rata waktu yang dibutuhkan suatu *job* untuk dikerjakan dari client ke server dalam suatu antrian. Nilai yang dihasilkan untuk *Latency* ini diperoleh dari jumlah keseluruhan waktu awal suatu *job* dikerjakan sampai waktu *job* selesai dikerjakan dan dibagi oleh jumlah *job* yang telah ditentukan. Satuan yang digunakan dalam *Latency* ini adalah menit. Untuk melihat tabel data hasil

pengujian secara lengkap terdapat pada lampiran A. Untuk menghitung nilai dari parameter *Latency* dengan menggunakan rumus sebagai berikut :

$$Latency = \frac{\text{Jumlah total waktu keseluruhan job}}{\text{jumlah job}}$$

Berikut ini pada gambar 5.1 hasil pengujian terdapat pada *resourcemanager* untuk mengukur parameter *Latency*.



Gambar 5.1 Pengukuran *Latency*

Tabel 5.1 Data Hasil Pengujian Untuk Pengukuran *Latency*

<i>JOB</i>	<i>DATASET</i>	STARTED	FINISHED	<i>TIME</i>
1	facebook-names-unique.txt (1,5 GB)	07:14:13	07:53:46	39mins, 32sec
2	facebook-names-unique.txt (1,5 GB)	07:13:38	07:49:00	35mins, 22sec
3	facebook-names-unique.txt (1,5 GB)	07:14:35	07:59:28	44mins, 52sec
4	facebook-names-unique.txt (1,5 GB)	07:16:51	08:30:46	1hrs, 13mins, 55sec
5	facebook-names-unique.txt (1,5 GB)	07:14:12	08:27:46	1hrs, 13mins, 33sec

Jika dilihat dari hasil pengujian yang telah dilakukan. Pada tabel tersebut terdapat 5 *job* yang telah selesai dikerjakan. Setiap *job* terdapat nilai *time* yang didapatkan dari berapa lama *job* tersebut dikerjakan. Jumlah keseluruhan untuk mengerjakan kelima *job* tersebut selama 267.2 menit. Untuk menghitung rata-rata keseluruhan kelima *job* tersebut dengan melakukan perhitungan sebagai berikut:

$$Latency = \frac{267.2 \text{ menit}}{5 \text{ job}} = 53.44 \text{ menit}$$

Untuk melakukan pengukuran pada parameter *Throughput* adalah untuk mengukur kecepatan rata-rata transfer data yang diterima oleh suatu *node* dalam selang waktu tertentu. Parameter ini akan diperoleh dari jumlah total data yang diterima dibagi dengan durasi interval waktu pengiriman data secara keseluruhan. Satuan yang digunakan untuk menghitung nilai *Throughput* adalah *Mbit/s*. Untuk melihat tabel data hasil pengujian secara lengkap terdapat pada lampiran A. Untuk menghitung nilai dari parameter *Throughput* dengan menggunakan rumus sebagai berikut :

$$Throughput = \frac{\text{Jumlah total data yang diterima (Bytes)}}{\text{durasi interval waktu (sec)}}$$

5.1.1 Variasi Ukuran Data

5.1.1.1 Job Fail Rate

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Job Fail Rate* berdasarkan skenario pengujian pertama yaitu, yang ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil Nilai *Job Fail Rate* pada Skenario Variasi Ukuran Data

Algoritme Scheduling	Jumlah Ukuran Data		
	1,5GB	286,81MB	20MB
Fair share	3,45%	2,18%	1,32%
capacity	2,95%	1,88%	0,99%

Dari tabel 5.2 tersebut dapat dilihat nilai *Job Fail Rate* pada algoritme *Fair Share Scheduling* dengan jumlah ukuran data sebesar 1,5 GB memiliki nilai *fail rate* sebesar 3,45%, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *fail rate* sebesar 2,18%, dan jumlah ukuran data sebesar 20 MB memiliki nilai *fail rate* sebesar 1,32%. Sedangkan pada *Capacity Scheduling* jumlah ukuran data sebesar 1,5 GB memiliki nilai *fail rate* sebesar 2,95%, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *fail rate* sebesar 1,88%, dan jumlah ukuran data sebesar 20 MB memiliki nilai *fail rate* sebesar 0,99%.

5.1.1.2 Latency

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Latency* berdasarkan skenario pengujian pertama yaitu, yang ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Nilai *Latency* pada Skenario Variasi Ukuran Data

	Jumlah Ukuran Data

Algoritme Scheduling	1,5GB	286,81MB	20MB
Fair share	53,44 menit	31,6 menit	19,25 menit
capacity	69,9 menit	44,48 menit	39,35 menit

Dari tabel 5.3 tersebut dapat dilihat nilai *Latency* pada algoritme *Fair Share Scheduling* dengan jumlah ukuran data sebesar 1,5 GB memiliki nilai *Latency* sebesar 53,44 menit, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *Latency* sebesar 31,6 menit, dan jumlah ukuran data sebesar 20 MB memiliki nilai *Latency* sebesar 19,25 menit. Sedangkan pada *Capacity Scheduling* jumlah ukuran data sebesar 1,5 GB memiliki nilai *Latency* sebesar 69,9 menit, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *Latency* sebesar 44,48 menit, dan jumlah ukuran data sebesar 20 MB memiliki nilai *Latency* sebesar 39,35 menit.

5.1.1.3 Throughput

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Throughput* berdasarkan skenario pengujian pertama yaitu, yang ditunjukkan pada tabel 5.4.

Tabel 5.4 Hasil Nilai *Throughput* pada Skenario Variasi Ukuran Data

Algoritme Scheduling	Jumlah Ukuran Data		
	1,5GB	286,81MB	20MB
Fair share	0,47 Mbit/s	0,45 Mbit/s	0,0072 Mbit/s
capacity	0,36 Mbit/s	0,32 Mbit/s	0,0035 Mbit/s

Dari tabel 5.4 tersebut dapat dilihat nilai *Throughput* pada algoritme *Fair Share Scheduling* dengan jumlah ukuran data sebesar 1,5 GB memiliki nilai *Latency* sebesar 0,47 Mbit/s, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *Latency* sebesar 0,45 Mbit/s, dan jumlah ukuran data sebesar 20 MB memiliki nilai *Latency* sebesar 0,0072 Mbit/s. Sedangkan pada *Capacity Scheduling* jumlah ukuran data sebesar 1,5 GB memiliki nilai *Latency* sebesar 0,36 Mbit/s, pada jumlah ukuran data sebesar 286,81 MB memiliki nilai *Latency* sebesar 0,32 Mbit/s, dan jumlah ukuran data sebesar 20 MB memiliki nilai *Latency* sebesar 0,0035 Mbit/s.

5.1.2 Variasi Jumlah Job

Pada hasil skenario variasi jumlah *job* dengan menggunakan variabel bebas berupa variasi jumlah *job*, sedangkan variabel tetap berupa jumlah ukuran data dan jenis *job*. Variasi jumlah *job* yaitu, sebanyak 5 *job*, 10 *job*, dan 15 *job*. Sedangkan jumlah ukuran data yang digunakan yaitu, Facebook-names-unique.txt dengan ukuran data sebesar 1,5 GB dan jenis *job* yang digunakan adalah *job* wordcount yang dilakukan pada masing-masing algoritme *Fair Share Scheduling* dan *Capacity Scheduling*. Dari hasil pengujian tersebut dapat diketahui nilai dari parameter pengujian *Job Fail Rate*, *Latency*, dan *Throughput* sebagai berikut.

5.1.2.1 Job Fail Rate

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Job Fail Rate* berdasarkan skenario pengujian kedua yaitu, yang ditunjukkan pada tabel 5.5.

Tabel 5.5 Hasil Nilai *Job Fail Rate* pada Variasi Jumlah *Job*

Algoritme Scheduling	Jumlah <i>Job</i>		
	5 <i>Job</i>	10 <i>Job</i>	15 <i>Job</i>
Fair share	3,45%	2,97%	2,89%
capacity	2,95%	2,64%	2,44%

Dari tabel 5.5 tersebut dapat dilihat nilai *Job Fail Rate* pada algoritme *Fair Share Scheduling* dengan jumlah *job* sebesar 5 *job* memiliki nilai *fail rate* sebesar 3,45%, pada jumlah *job* sebesar 10 *job* memiliki nilai *fail rate* sebesar 2,97%, dan jumlah *job* sebesar 15 *job* memiliki nilai *fail rate* sebesar 2,89%. Sedangkan pada *Capacity Scheduling* jumlah *job* sebesar 5 *job* memiliki nilai *fail rate* sebesar 2,95%, pada jumlah *job* sebesar 10 *job* memiliki nilai *fail rate* sebesar 2,64%, dan jumlah *job* sebesar 15 *job* memiliki nilai *fail rate* sebesar 2,44%.

5.1.2.2 Latency

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Latency* berdasarkan skenario pengujian kedua yaitu, yang ditunjukkan pada tabel 5.6.

Tabel 5.6 Hasil Nilai *Latency* pada Skenario Variasi Jumlah *Job*

Algoritme Scheduling	Jumlah <i>Job</i>		
	5 <i>Job</i>	10 <i>Job</i>	15 <i>Job</i>
Fair share	53,44 menit	68,29 menit	99,89 menit
capacity	69,9 menit	98,32 menit	118,69 menit

Dari tabel 5.6 tersebut dapat dilihat nilai *Latency* pada algoritme *Fair Share Scheduling* dengan jumlah *job* sebesar 5 *job* memiliki nilai *Latency* sebesar 53,44 menit, pada jumlah *job* sebesar 10 *job* memiliki nilai *Latency* sebesar 68,29 menit, dan jumlah *job* sebesar 15 *job* memiliki nilai *Latency* sebesar 99,89 menit. Sedangkan pada *Capacity Scheduling* jumlah *job* sebesar 5 *job* memiliki nilai *Latency* sebesar 69,9 menit, pada jumlah *job* sebesar 10 *job* memiliki nilai *Latency* sebesar 98,32 menit, dan jumlah *job* sebesar 15 *job* memiliki nilai *Latency* sebesar 118,69 menit.

5.1.2.3 Throughput

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Throughput* berdasarkan skenario pengujian kedua yaitu, yang ditunjukkan pada tabel 5.7

Tabel 5.7 Hasil Nilai *Throughput* pada Skenario Variasi Jumlah *Job*

Algoritme Scheduling	Jumlah <i>Job</i>		
	5 <i>Job</i>	10 <i>Job</i>	15 <i>Job</i>
Fair share	0,47 <i>Mbit/s</i>	0,37 <i>Mbit/s</i>	0,25 <i>Mbit/s</i>
capacity	0,36 <i>Mbit/s</i>	0,25 <i>Mbit/s</i>	0,21 <i>Mbit/s</i>

Dari tabel 5.7 tersebut dapat dilihat nilai *Throughput* pada algoritme *Fair Share Scheduling* dengan jumlah *job* 5 *job* memiliki nilai *Latency* sebesar 0,47 *Mbit/s*, pada jumlah *job* sebesar 10 *job* memiliki nilai *Latency* sebesar 0,37 *Mbit/s*, dan jumlah *job* sebesar 15 *job* memiliki nilai *Latency* sebesar 0,25 *Mbit/s*. Sedangkan pada *Capacity Scheduling* jumlah *job* sebesar 5 *job* memiliki nilai *Latency* sebesar 0,36 *Mbit/s*, pada jumlah *job* sebesar 10 *job* memiliki nilai *Latency* sebesar 0,25 *Mbit/s*, dan jumlah *job* sebesar 15 *job* memiliki nilai *Latency* sebesar 0,21 *Mbit/s*.

5.3 Variasi Jenis *Job*

Pada hasil skenario variasi jenis *job* dengan menggunakan variabel bebas berupa variasi jenis *job*, sedangkan variabel tetap berupa jumlah ukuran data dan jumlah *job*. Variasi jenis *job* yaitu, menggunakan *job* wordcount dan *job* wordmean. Sedangkan jumlah ukuran data yang digunakan yaitu, Facebook-names-unique.txt dengan ukuran data sebesar 1,5 GB dan jumlah *job* sebanyak 5 *job* yang dilakukan pada masing-masing algoritme *Fair Share Scheduling* dan *Capacity Scheduling*. Dari hasil pengujian tersebut dapat diketahui nilai dari parameter pengujian *Job Fail Rate*, *Latency*, dan *Throughput* sebagai berikut.

5.1.3.1 *Job Fail Rate*

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Job Fail Rate* berdasarkan skenario pengujian ketiga yaitu, yang ditunjukkan pada tabel 5.8.

Tabel 5.8 Hasil Nilai *Job Fail Rate* pada Skenario Variasi Jenis *Job*

Algoritme Scheduling	Jenis <i>Job</i>	
	WordCount	WordMean
Fair share	3,45%	2,92%
Capacity	2,95%	1,81%

Dari tabel 5.8 tersebut dapat dilihat nilai *Job Fail Rate* pada algoritme *Fair Share Scheduling* dengan jenis *job* wordcount memiliki nilai *fail rate* sebesar 3,45%, pada jenis *job* wordmean memiliki nilai *fail rate* sebesar 2,92%. Sedangkan pada *Capacity Scheduling* dengan jenis *job* wordcount memiliki nilai *fail rate* sebesar 2,95%, pada jenis *job* wordmean memiliki nilai *fail rate* sebesar 1,81%.

5.1.3.2 Latency

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Latency* berdasarkan skenario pengujian ketiga yaitu, yang ditunjukkan pada tabel 5.9.

Tabel 5.9 Hasil Nilai *Latency* pada Variasi Jenis *Job*

Algoritme Scheduling	Jenis <i>Job</i>	
	WordCount	WordMean
Fair share	53,44 menit	50,2 menit
capacity	69,9 menit	65,3 menit

Dari tabel 5.9 tersebut dapat dilihat nilai *Latency* pada algoritme *Fair Share Scheduling* dengan jenis *job* wordcount memiliki nilai *Latency* sebesar 53,44 menit, pada jenis *job* wordmean memiliki nilai *Latency* sebesar 50,2 menit. Sedangkan pada *Capacity Scheduling* dengan jenis *job* wordcount memiliki nilai *Latency* sebesar 69,9 menit, pada jenis *job* wordmean memiliki nilai *Latency* sebesar 65,3 menit.

5.1.3.3 Throughput

Pada sub bab ini akan diketahui hasil nilai dari parameter pengujian *Throughput* berdasarkan skenario pengujian ketiga yaitu, yang ditunjukkan pada tabel 5.10.

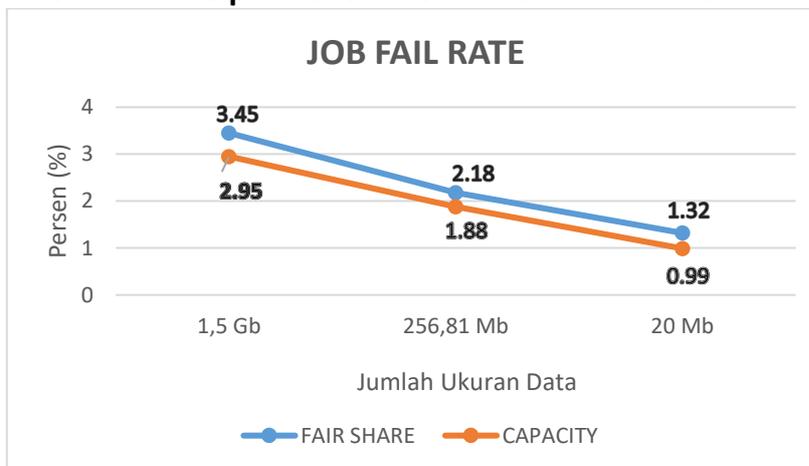
Tabel 5.10 Hasil Nilai *Throughput* pada Variasi Jenis *Job*

Algoritme Scheduling	Jenis <i>Job</i>	
	WordCount	WordMean
Fair share	0,47 <i>Mbit/s</i>	0,00011 <i>Mbit/s</i>
capacity	0,36 <i>Mbit/s</i>	0,000084 <i>Mbit/s</i>

Dari tabel 5.10 tersebut dapat dilihat nilai *Throughput* pada algoritme *Fair Share Scheduling* dengan jenis *job* wordcount memiliki nilai *Throughput* sebesar 0,47 *Mbit/s*, pada jenis *job* wordmean memiliki nilai *Throughput* sebesar 0,00011 *Mbit/s*. Sedangkan pada *Capacity Scheduling* dengan jenis *job* wordcount memiliki nilai *Throughput* sebesar 0,36 *Mbit/s*, pada jenis *job* wordmean memiliki nilai *Latency* sebesar 0,000084 *Mbit/s*.

5.2 Analisis

5.2.1 Nilai *Job Fail Rate* pada Skenario Variasi Ukuran Data

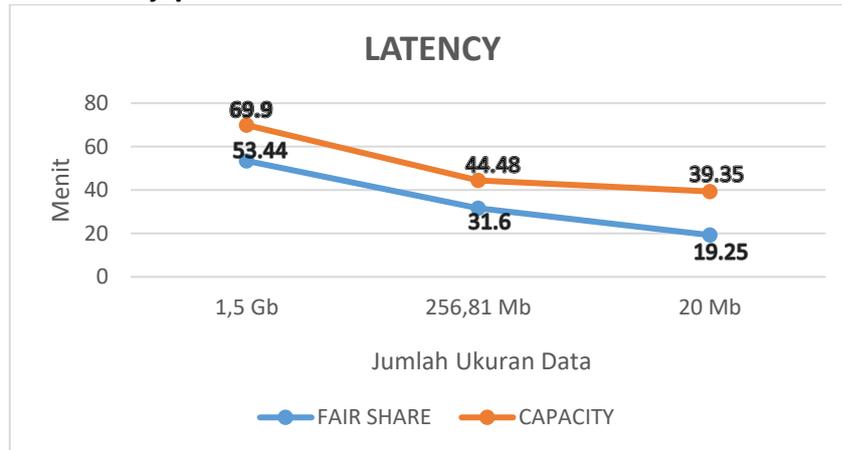


Gambar 5.2 Grafik Perbandingan *Job Fail Rate* pada Skenario Variasi Ukuran Data

Dapat dilihat grafik pada gambar 5.3 pada variasi ukuran data dengan pengiriman *job wordcount* menggunakan ukuran data yang berbeda-beda. Hasil nilai dari parameter *Job Fail Rate* terlihat bahwa pada *Fair Share Scheduling* dan *Capacity Scheduling* memiliki nilai *Fail rate* maksimum pada pengiriman *job* dengan ukuran *dataset* sebesar 1,5 GB kemudian semakin menurun seiring dengan semakin kecilnya ukuran *dataset* yang digunakan. Pada setiap melakukan proses eksekusi *job*, HDFS membaca potongan-potongan data yang disebut *block size* dengan ukuran data sebesar 128 MB. Sehingga perbedaan ukuran *dataset* yang digunakan juga sangat berpengaruh, semakin besar ukuran data set yang digunakan maka proses pembacaan *maps* akan semakin lama dan mengakibatkan nilai *error* dan nilai *timeout* semakin besar. Begitupula sebaliknya jika ukuran *dataset* yang digunakan semakin kecil maka proses pembacaan *maps* akan semakin cepat dan nilai *fail* yang diakibatkan akan semakin kecil juga. Untuk *job* dengan ukuran data sebesar 1,5 GB memiliki jumlah sebesar 12 *maps*, untuk *job* dengan ukuran data sebesar 286,81 MB memiliki jumlah sebesar 3 *maps*, untuk *job* dengan ukuran data sebesar 20MB memiliki jumlah sebesar 1 *maps*.

Untuk perbandingan besar nilai *Job Fail Rate* pada algoritme *Capacity Scheduling* memiliki nilai maksimal sebesar 2.95%, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai maksimal sebesar 3.45% pada pengiriman *job* dengan ukuran data 1,5 GB. Jadi algoritme *Capacity Scheduling* memiliki nilai *fail rate* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 0.5%

5.2.2 Nilai *Latency* pada Skenario Variasi Ukuran Data



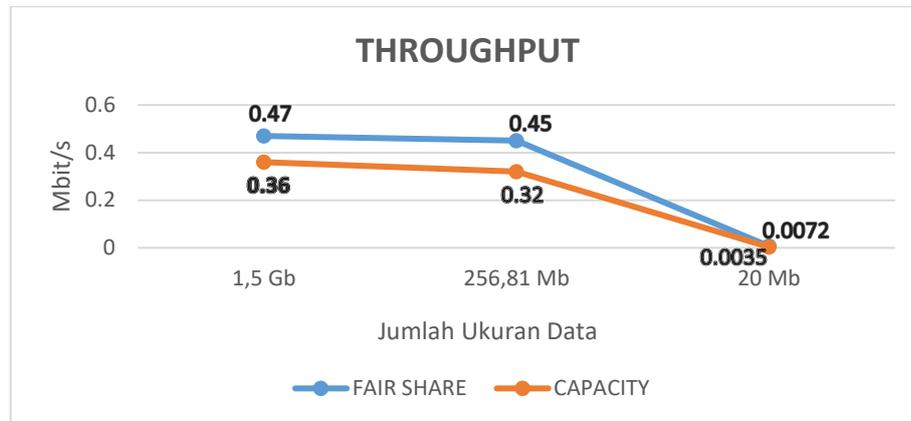
Gambar 5.3 Grafik Perbandingan *Latency* pada Skenario Variasi Ukuran Data

Dapat dilihat grafik pada gambar 5.4 pada variasi ukuran data dengan pengiriman *job wordcount* menggunakan ukuran *data* yang berbeda-beda. Hasil nilai dari parameter *Latency*, dapat dilihat bahwa rata-rata waktu yang dibutuhkan *Fair Share Scheduling* dari masing-masing jumlah *job* yang telah dieksekusi kedalam antrian memiliki waktu pengerjaan yang lebih cepat jika dibandingkan dengan *Capacity Scheduling*. Pada *Fair Share Scheduling* dapat melakukan running *job* sebanyak 3 *job* pada waktu yang bersamaan dengan membagi resource secara adil antara *job* satu dengan *job* lainnya, sehingga proses pengerjaan *job* akan lebih cepat selesai jika dibandingkan dengan *Capacity Scheduling*.

Jika dilihat dari gambar tersebut grafik akan semakin menurun setiap berkurangnya jumlah ukuran data yang digunakan. Hal ini dikarenakan adanya pengaruh penggunaan ukuran *dataset* yang berbeda-beda. Semakin besar ukuran *dataset* yang digunakan maka proses pembacaan *maps* akan semakin lama. Dengan semakin lamanya proses pembacaan *maps* tersebut, maka akan mengakibatkan munculnya kegagalan (*failed*) pada saat melakukan running program semakin besar. Pada *job* yang mengalami kegagalan tersebut akan menghambat proses pengerjaan *job* sehingga akan membuat *job* semakin lama untuk diselesaikan.

Untuk perbandingan besar nilai *Latency* pada algoritme *Capacity Scheduling* memiliki nilai maksimal sebesar 69,9 menit, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai maksimal sebesar 53,44 menit pada pengiriman *job* dengan ukuran data 1,5 GB. Jadi algoritme *Fair Share Scheduling* memiliki rata-rata nilai *Latency* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 16,46 menit.

5.2.3 Nilai *Throughput* pada Skenario Variasi Ukuran Data

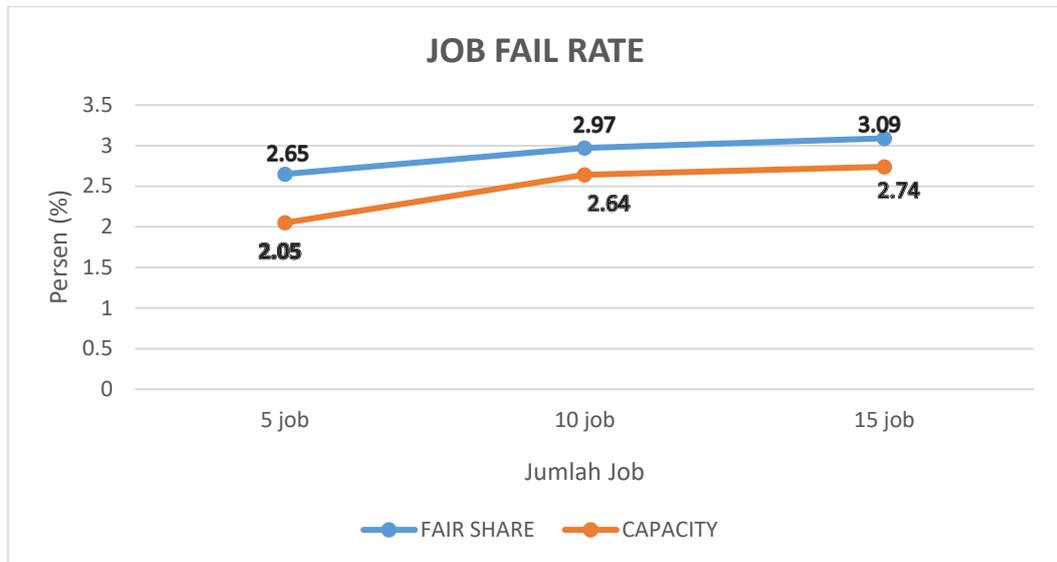


Gambar 5.4 Grafik Perbandingan *Throughput* pada Skenario Variasi Ukuran Data

Dapat dilihat grafik pada gambar 5.5 pada variasi ukuran data dengan pengiriman *job wordcount* berdasarkan pengaruh pada jumlah ukuran *dataset* yang digunakan. Jika dilihat dari gambar tersebut grafik akan semakin menurun setiap berkurangnya jumlah ukuran data yang digunakan. Hal ini dikarenakan semakin kecil ukuran data yang digunakan, maka semakin kecil juga ukuran data yang diterima. Interval waktu selama *job* dikerjakan juga memengaruhi nilai *throughput*. Semakin besar jumlah ukuran data yang digunakan maka semakin lama waktu yang dibutuhkan *job* untuk menyelesaikannya.

Untuk perbandingan kecepatan transfer data pada algoritme *Capacity Scheduling* memiliki nilai maksimal sebesar 0,36 *Mbit/s*, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai maksimal sebesar 0,47 *Mbit/s*. Jadi algoritme *Fair Share Scheduling* memiliki kecepatan transfer data yang lebih cepat jika dibandingkan dengan algoritme *Capacity Scheduling* dengan selisih nilai sebesar 0.11 *Mbit/s*. Hal ini dikareman nilai *timeout* pada *Capacity Scheduling* lebih rendah sehingga untuk melakukan transfer data akan semakin cepat jika dibandingkan dengan *Fair Share Scheduling*.

5.2.4 Nilai *Job Fail Rate* pada Skenario Variasi Jumlah *Job*

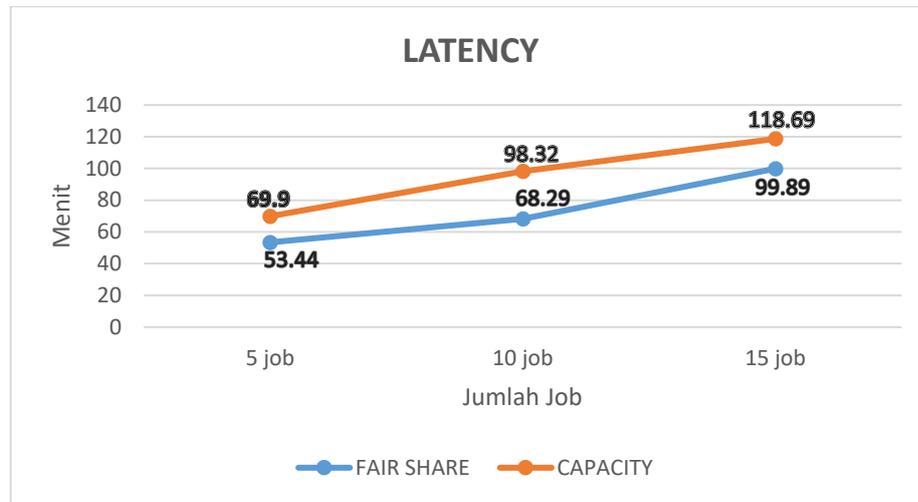


Gambar 5.5 Grafik Perbandingan *Job Fail Rate* pada Variasi Jumlah *Job*

Dapat dilihat grafik pada gambar 5.6 pada variasi jumlah *job* dengan pengiriman *job wordcount* menggunakan ukuran *dataset* yang sama yaitu, sebesar 1,5 GB. Hasil dari parameter *Job Fail Rate* terlihat bahwa pada *Fair Share Scheduling* dan *Capacity Scheduling* memiliki nilai *Fail rate* manimum pada pengiriman *job* ke 5 pertama kemudian semakin meningkat seiring dengan bertambahnya jumlah *job* yang telah dieksekusi. Semakin banyak jumlah *job* yang dikerjakan maka antrian yang terjadi akan semakin banyak. Dengan semakin banyaknya antrian tersebut, *job* akan semakin lama menunggu hingga proses eksekusi dilakukan. Dengan semakin lama proses eksekusi *job* tersebut maka akan menimbulkan nilai *failed* yang dapat menghambat waktu pengerjaan *job*.

Untuk perbandingan besar nilai *Job Fail Rate* pada algoritme *Capacity Scheduling* memiliki nilai minimal sebesar 2.05%, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai minimal sebesar 2.65%. Jadi algoritme *Capacity Scheduling* memiliki nilai *fail rate* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 0.6%. Hal ini disebabkan karena nilai *failed* yang muncul pada algoritme *Capacity Scheduling* lebih rendah jika dibanding *Fair Share Scheduling*.

5.2.5 Nilai *Latency* pada Skenario Variasi Jumlah *Job*

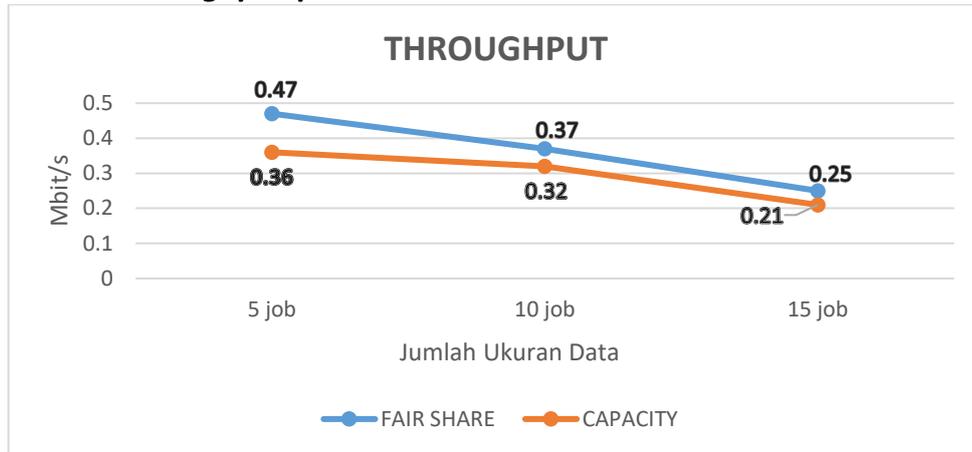


Gambar 5.6 Grafik Perbandingan *Latency* pada Variasi Jumlah *Job*

Dapat dilihat grafik pada gambar 5.7 pada variasi jumlah *job* dengan pengiriman *job wordcount* menggunakan ukuran *dataset* yang sama yaitu, sebesar 1,5 GB. Secara keseluruhan hasil dari parameter *Latency* dapat dilihat bahwa rata-rata waktu yang dibutuhkan *Fair Share Scheduling* dari masing-masing jumlah *job* yang telah dieksekusi kedalam antrian memiliki waktu pengerjaan yang lebih cepat jika dibandingkan dengan *Capacity Scheduling*. Pada *Fair Share Scheduling* dan *Capacity Scheduling* memiliki nilai *Latency* minimum pada pengiriman *job* ke 5 pertama kemudian semakin meningkat seiring dengan bertambahnya jumlah *job* yang telah dieksekusi. Hal ini disebabkan karena pengaruh nilai *failed* yang semakin banyak seiring dengan bertambahnya *job* yang dieksekusi maka akan menghambat proses pengerjaan *job* dan nilai *timeout* yang diperoleh juga semakin lama sehingga tidak dapat memaksimalkan waktu pengerjaan *job*.

Untuk perbandingan besar nilai *Latency* pada algoritme *Capacity Scheduling* memiliki nilai maksimal sebesar 118,69 menit, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai maksimal sebesar 99,89 menit pada pengiriman *job* sebanyak 15 *job*. Jadi algoritme *Fair Share Scheduling* memiliki rata-rata nilai *Latency* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 18,8 menit.

5.2.6 Nilai *Throughput* pada Skenario Variasi Jumlah *Job*

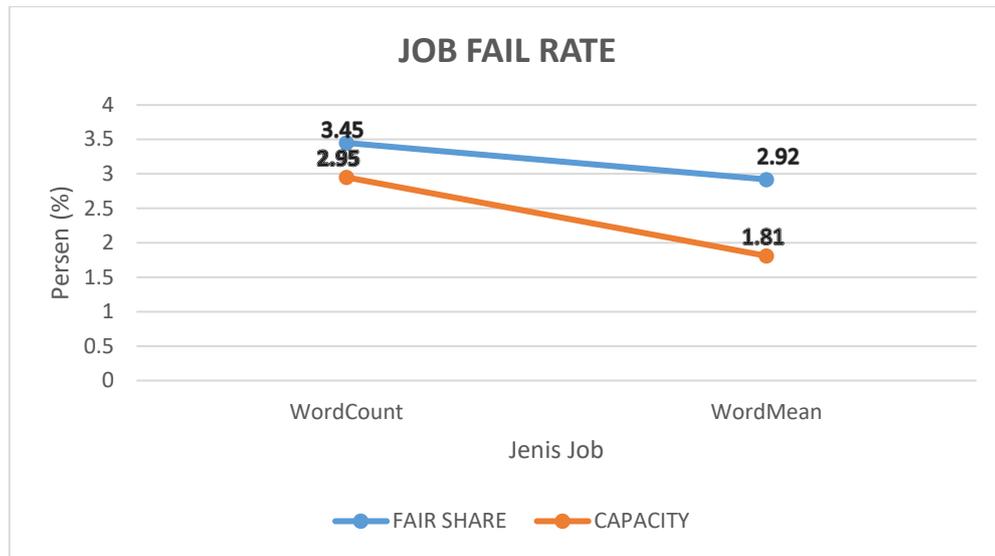


Gambar 5.7 Grafik Perbandingan *Throughput* pada Variasi Jumlah *Job*

Dapat dilihat grafik pada gambar 5.8 pada variasi jumlah *job* dengan pengiriman *job wordcount* berdasarkan pengaruh pada jumlah *job* yang digunakan yaitu, 5 *job*, 10 *job*, dan 15 *job*. Dari data tersebut dapat diketahui bahwa semakin besar jumlah *job* yang digunakan dalam skenario ini maka semakin kecil kecepatan transfer data yang di dapat. Hal ini dikarenakan semakin besar jumlah *job* yang digunakan maka akan bertambah banyak antrian suatu *job*, sehingga akan menyebabkan *timeout*. Semakin besar nilai *timeout* maka semakin lama waktupengerjaan suatu *job* sehingga akan memperkecil nilai *throughput*.

Semakin besar nilai *throughput* semakin besar kecepatan transfer data suatu jaringan, maka semakin baik jaringan tersebut. Untuk perbandingan besar nilai *Throughput* pada pengiriman *job* sebanyak 5 *job*. Algoritme *Capacity Scheduling* memiliki nilai maksimal sebesar 0,36 *Mbit/s*, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai maksimal sebesar 0,47 *Mbit/s*. Jadi algoritme *Fair Share Scheduling* memiliki kecepatan transfer data yang lebih cepat jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 0.11 *Mbit/s*.

5.2.7 Nilai *Job Fail Rate* pada Skenario Variasi Jenis *Job*

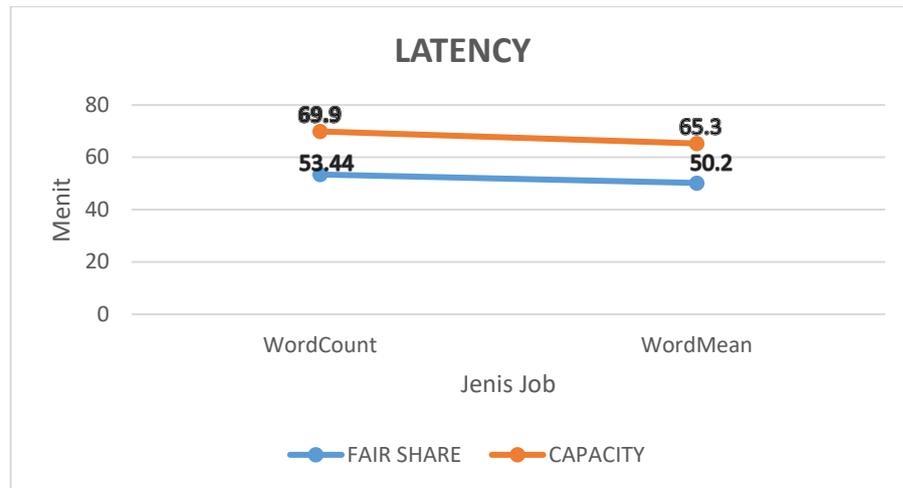


Gambar 5.8 Grafik Perbandingan *Job Fail Rate* pada Variasi Jenis *Job*

Dapat dilihat grafik pada gambar 5.9 pada variasi jenis *job* dengan pengiriman menggunakan variasi jenis *job* yang berbeda yaitu, *job wordcount* dan *job wordmean* dan menggunakan ukuran *dataset* yang sama. Hasil dari parameter *Job Fail Rate* terlihat bahwa pada *job wordmean* memiliki perbedaan nilai *Fail rate* yang cukup kecil jika dibandingkan dengan *job wordcount*. Hal ini dikarenakan karena *job wordmean* melakukan proses pengerjaan *job* yang sederhana yaitu, hanya melakukan proses perhitungan jumlah panjang kata dalam suatu *file*, sehingga pada eksekusi *maps* yang dijalankan akan semakin cepat jika dibanding dengan *job wordcount* yang akan menghitung kemiripan jumlah kata yang terdapat pada *file*. Dengan semakin cepat proses eksekusi *maps* tersebut maka akan meminimalisir munculnya *failed* pada saat dilakukannya proses pengiriman, sehingga akan berpengaruh pada nilai *fail rate*.

Untuk perbandingan besar nilai *Job Fail Rate* pada algoritme *Capacity Scheduling* memiliki nilai minimal sebesar 1.81%, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai minimal sebesar 2.92% pada pengiriman *job wordmean*. Jadi algoritme *Capacity Scheduling* memiliki nilai *fail rate* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 1.11%

5.2.8 Nilai *Latency* pada Skenario Variasi Jenis *Job*

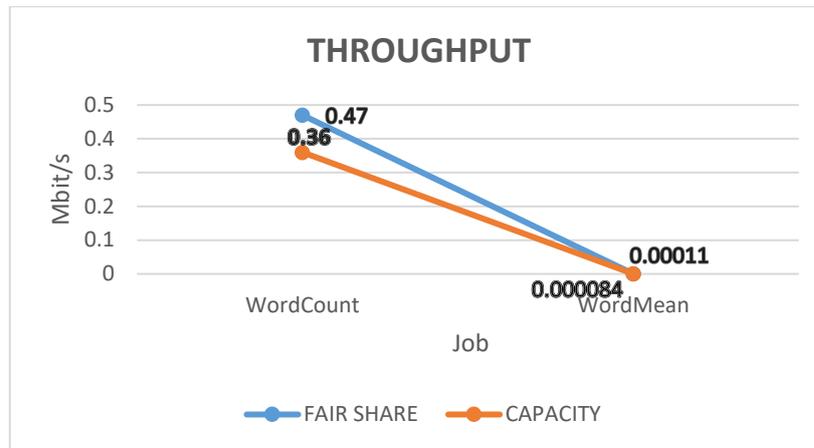


Gambar 5.9 Grafik Perbandingan *Latency* pada Variasi Jenis *Job*

Dapat dilihat grafik pada gambar 5.10 pada variasi jenis *job* dengan pengiriman menggunakan variasi jenis *job* yang berbeda yaitu, *job wordcount* dan *job wordmean* dan menggunakan ukuran *dataset* yang sama. Secara keseluruhan hasil dari parameter *Latency* dapat dilihat bahwa bahwa rata-rata waktu yang dibutuhkan *Fair Share Scheduling* dari masing-masing jenis *job* memiliki waktu pengerjaan yang lebih cepat jika dibandingkan dengan *Capacity Scheduling*. Pada *job wordmean* memiliki perbedaan nilai *Latency* lebih cepat jika dibandingkan dengan *job wordcount*. Karena pada proses pengerjaan dengan menggunakan *job wordmean* memiliki nilai *timeout* yang lebih sedikit dibandingkan dengan *job wordcount*. Pada nilai *timeout* tersebut akan berpengaruh pada cepat dan lamanya *job* untuk dieksekusi.

Untuk perbandingan besar nilai *Latency* pada algoritme *Capacity Scheduling* memiliki nilai minimal sebesar 65,3 menit, sedangkan pada algoritme *Fair Share Scheduling* memiliki nilai minimal sebesar 50,2 menit pada pengiriman *job wordmean*. Jadi algoritme *Fair Share Scheduling* memiliki rata-rata nilai *Latency* yang lebih rendah jika dibandingkan dengan algoritme *Fair Share Scheduling* dengan selisih nilai sebesar 15,1 menit.

5.2.9 Nilai *Throughput* pada Skenario Variasi Jenis *Job*



Gambar 5.10 Grafik Perbandingan *Throughput* pada Variasi Jenis *Job*

Dapat dilihat grafik pada gambar 5.11 pada variasi jenis *job* dengan pengiriman menggunakan *job wordcount* dan *job wordmean* dan menggunakan ukuran *dataset* yang sama dan jumlah *job* yang sama yaitu, 5 *job*. Secara keseluruhan hasil dari parameter *Throughput* dapat dilihat bahwa bahwa kecepatan transfer data yang dibutuhkan *pada job wordcount* dari masing-masing memiliki kecepatan transfer data yang lebih cepat jika dibandingkan dengan *job wordmean*. Dapat dibuktikan pada algoritme *Fair Share Scheduling* dengan *Capacity Scheduling* yang memiliki selisih nilai *Throughput* pada *job wordcount* sebesar 0,11 *Mbit/s* sedangkan pada *job wordmean* memiliki selisih nilai *Throughput* sebesar 0.000026 *Mbit/s*. Karena ukuran data yang diterima pada proses pengerjaan dengan menggunakan *job wordcount* lebih besar jika dibandingkan dengan *job wordmean*. Sehingga dengan menggunakan *job wordcount* maka kecepatan transfer data juga akan semakin tinggi, meskipun kedua jenis *job* tersebut menggunakan ukuran dataset yang sama.

Untuk perbandingan besar nilai throughput yang diperoleh dari pengujian menggunakan algoritme *Fair Share Scheduling* dengan *Capacity Scheduling*, nilai throughput pada *Capacity Scheduling* lebih kecil dibandingkan dengan nilai throughput yang didapat ketika menggunakan algoritme *Fair Share Scheduling*.