BAB VI
PENGUJIAN DAN ANALISIS

Pada Bab VI akan dilakukan proses pengujian dan analisis terhadap sistem kompresi terdistribusi yang telah dibangun. Dalam proses pengujian digunakan tiga (3) buah strategi, yaitu pengujian unit, pengujian integrasi dan pengujian validasi. Pada pengujian unit dan pengujian integrasi, akan digunakan teknik pengujian *White Box (White Box Testing).* Sedangkan pada pengujian validasi akan digunakan teknik pengujian *Black Box (Black Box Testing).*

Proses analisis dilakukan untuk mengetahui unjuk kerja dari sistem kompresi terdistribusi dan perbandingannya dengan unjuk kerja sistem kompresi yang mandiri *(standalone).* Analisis juga dilakukan untuk mengetahui kualitas gambar hasil kompresi terdistribusi.

6.1. Pengujian

Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian integrasi dan pengujian validasi.

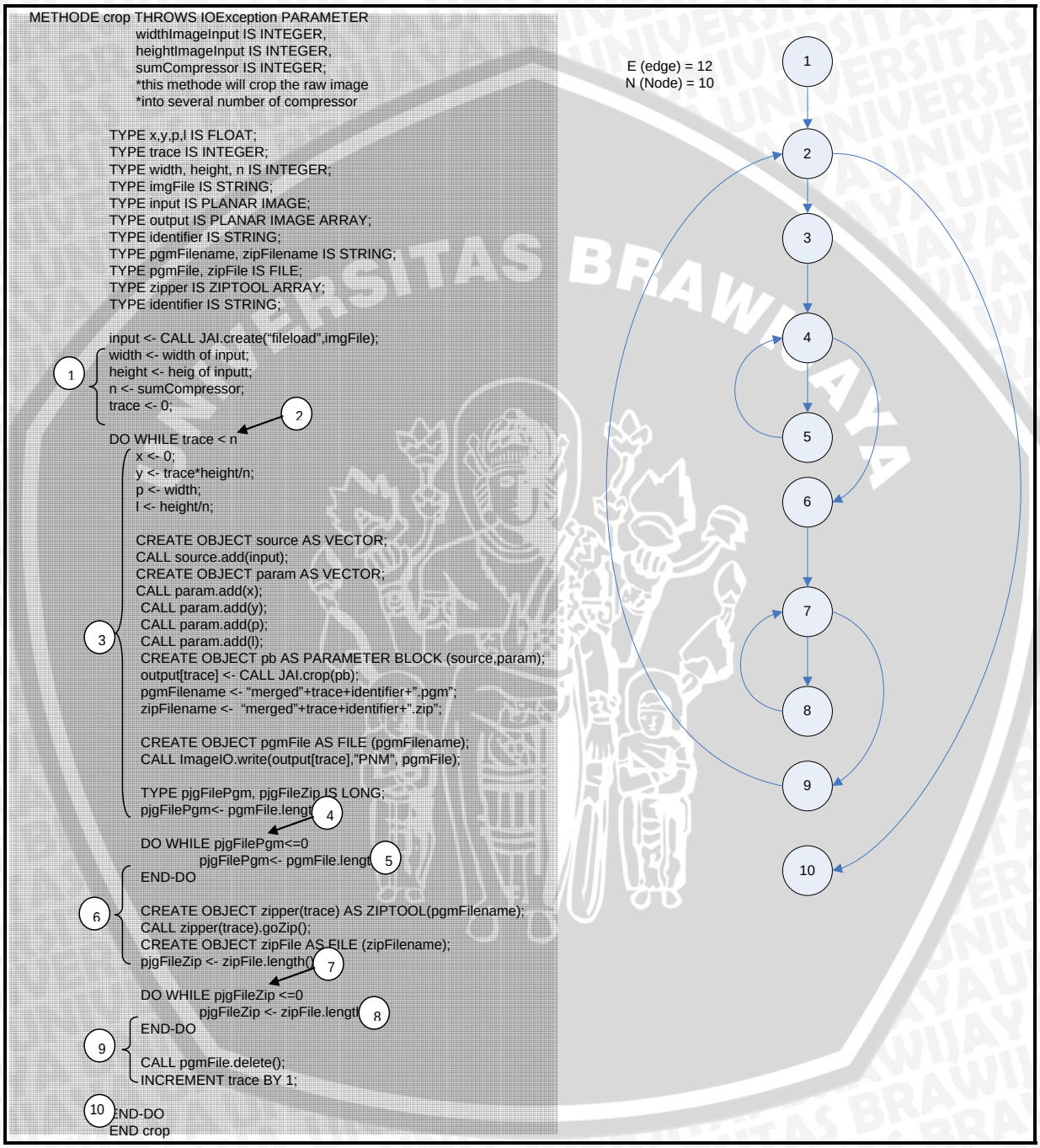
6.1.1. Pengujian Unit

Dalam sistem yang dibangun dengan paradigma pemrograman berorientasi objek, pengujian unit diterapkan untuk suatu metode (operasi) dari suatu klas. Pada pengujian unit sistem kompresi terdistribusi ini, digunakan teknik pengujian *White Box (White Box Testing)* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis *(cyclomatic complexity),* menentukan sebuah basis set dari jalur independen dan memberikan kasus uji *(test case)* pada setiap basis set yang telah ditentukan. Di dalam penulisan laporan skripsi ini, hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

a. Pengujian Unit untuk Operasi `crop()`

Operasi `crop()` merupakan implementasi dari algoritma proses pemotongan gambar mentah ke dalam beberapa bagian sesuai dengan jumlah kompresor yang akan

digunakan. Operasi (metode) `crop()` terdapat di dalam klas `ImageCropper`. Dalam Gambar 6.1 diperlihatkan proses pemodelan dalam *flow graph* pada operasi `crop()`.



Gambar 6.1. Pemodelan algoritma `crop` ( ) ke dalam *flow graph*

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `crop()` (Gambar 6.1), ditentukan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2., dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).*

V(G) = E – N + 2
= 12– 10 + 2
= 4

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 4 ditentukan empat buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-10
Jalur 2 : 1-2-3-4-6-7-9-2…
Jalur 3 : 1-2-3-4-5-4…
Jalur 4 : 1-2-3-4-6-7-8-7…

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

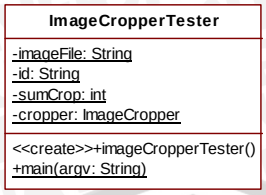
Tabel 6.1 *Test case* untuk pengujian unit operasi `crop()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	sumCompressor = 0;	Tidak akan didapatkan potongan gambar (hasil *cropping*)	Tidak ada hasil pemotongan.
2	Tidak bisa diuji secara mandiri dan merupakan satu rangkaian dengan jalur 3	Diperoleh hasil potongan gambar dengan nama ”merged0”+identifier+”.zip” dan ”merged1”+identifier+”.zip” dimana identifier di-*generate* secara random untuk identifikasi pada saat penggabungan sehingga bisa tepat digabungkan. Di dalam file *.zip yang dihasilkan terdapat file	2 file dalam format *.zip berhasil didapatkan dengan nama file yang tepat. 

		*.pgm (gambar mentah) yang akan diproses oleh kompresor	
3	sumCompressor = 2	Diperoleh hasil potongan gambar dengan nama ”merged0”+identifier+”.zip” dan ”merged1”+identifier+”.zip” dimana identifier di-*generate* secara random untuk identifikasi pada saat penggabungan sehingga bisa tepat digabungkan. Di dalam file *.zip adalah file gambar dalam format *.pgm	2 file dalam format *.zip berhasil didapatkan dengan nama file yang tepat. 
4	Tidak bisa diuji secara mandiri dan merupakan satu rangkaian dengan jalur 3	Diperoleh hasil potongan gambar dengan nama ”merged0”+identifier+”.zip” dan ”merged1”+identifier+”.zip” dimana identifier di-*generate* secara random untuk identifikasi pada saat penggabungan sehingga bisa tepat digabungkan. Di dalam file *.zip adalah file gambar dalam format *.pgm	2 file dalam format *.zip berhasil didapatkan dengan nama file yang tepat. 

Sumber: Pengujian

Untuk proses pengujian tersebut digunakan sebuah klas *dummy* yaitu klas `ImageCropperTester`. Atribut dan operasi yang ada dalam klas `ImageCropperTester` seperti ditunjukkan dalam Gambar 6.2. Sampel gambar yang digunakan dalam proses pengujian dimasukkan dengan mengeset pada atribut `imageFile` dari klas *dummy*. Demikian pula untuk *identifier* diset melalui atribut `id` dari klas *dummy*.

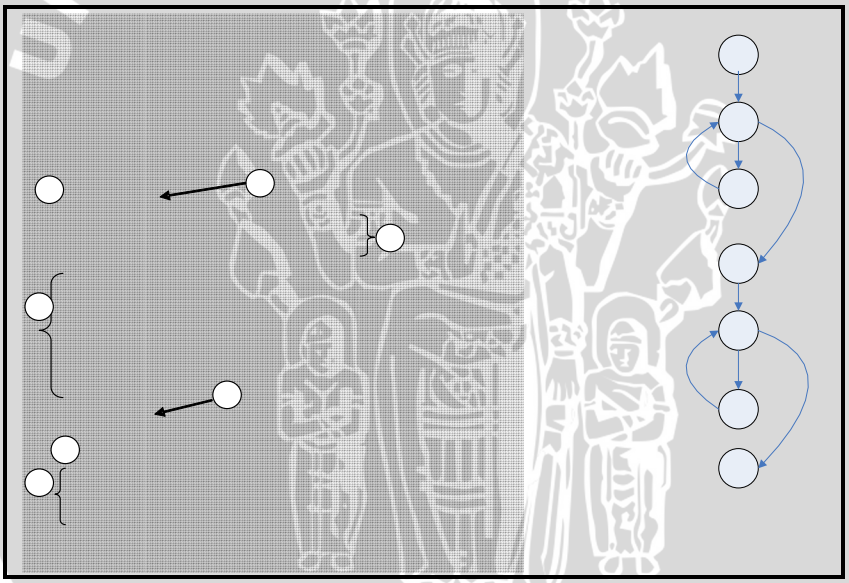


Gambar 6.2. Diagram klas *dummy* `ImageCropperTester`

Sumber: Pengujian

b. Pengujian Unit untuk Operasi `goMerge` ()

Operasi `goMerge()` terdapat pada klas `ImageUnifier`. Operasi (metode) ini merupakan sebuah implementasi dari algoritma penggabungan gambar. Pemodelan algoritma penggabungan gambar ke dalam *flow graph* ditunjukkan dalam Gambar 6.3.



Gambar 6.3. Pemodelan algoritma operasi `goMerge()` ke dalam *flow graph*

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `goMerge()` (Gambar 6.3), ditentukan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2.

V(G) = E – N + 2

= 8 – 7 + 2

= 3

:ception;
ages*

ARRAY;
:;

TYPE imgString IS STRING ARRAY;
TYPE elementSum IS INTEGER;
TYPE i IS INTEGER;
TYPE fileOut IS STRING;

5

1

i <- 0;
DO WHILE i < elementSum

2

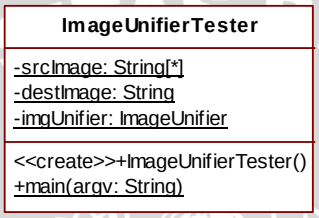
Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 3 ditentukan tiga buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-4-5-7

Jalur 2 : 1-2-3-2-…

Jalur 3 : 1-2-4-5-6-5-…

Untuk keperluan pengujian, digunakan klas bantu yaitu klas `ImageUnifierTester` seperti digambarkan melalui diagram klas dalam Gambar 6.4

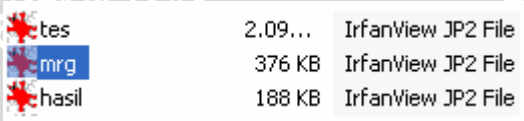
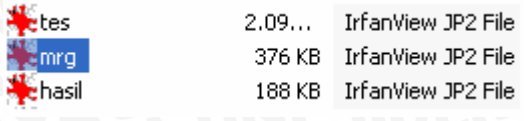


Gambar 6.4. Diagram klas *dummy* `ImageUnifierTester`

Sumber: Pengujian

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.2.

Tabel 6.2. *Test case* untuk pengujian unit operasi `goMerge()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	elementSum=0	Tidak didapatkan hasil penggabungan	Tidak ada file hasil penggabungan.
2	elementSum=2	Didapatkan sebuah file JPEG2000 hasil penggabungan dengan nama sesuai dengan atribut fileOut.	Berhasil didapatkan file dengan nama fileOut (berekstensi *.j2k).  tes 2.09… IrfanView JP2 File mrg 376 KB IrfanView JP2 File hasil 188 KB IrfanView JP2 File Pada proses pengujian, fileOut didapatkan dari nilai atribut destImage dari klas ImageUnifierTester. destImage diberi nilai “mrg.j2k”
3	elementSum=2	Didapatkan sebuah file JPEG2000 hasil penggabungan dengan nama sesuai dengan atribut fileOut.	Berhasil didapatkan file dengan nama fileOut (berekstensi *.j2k).  tes 2.09… IrfanView JP2 File mrg 376 KB IrfanView JP2 File hasil 188 KB IrfanView JP2 File

			Pada proses pengujian, fileOut didapatkan dari nilai atribut destImage dari klas

Sumber: Pengujian

c. Pengujian Unit untuk Operasi `sendMessageToDistrib ()`

Operasi `sendMessageToDistrib()` merupakan implementasi dari algoritma pengiriman pesan dari subsistem *client* kepada subsistem distributor yang ada di dalam klas `MessageToDistributorSender`. Di dalam metode `sendMessageToDistrib()` dibentuk *client socket* untuk subsistem *client* yang akan membentuk koneksi dengan mengirimkan pesan kepada *server socket* pada susistem distributor. Pemodelan algoritma pengiriman pesan ke dalam *flow graph* ditunjukkan dalam Gambar 6.5.

Gambar 6.5. pemodelan algoritma `sendMessageToDistrib()` ke dalam *flow graph*

Sumber: Pengujian

Jumlah kompleksitas siklomatis *(cyclomatic complexity)* dari *flow graph* algoritma `sendMessageToDistrib()` yang telah dimodelkan dalam Gambar 6.5 adalah 1 buah.

V(G) = E – N + 2

= 0 – 1+ 2

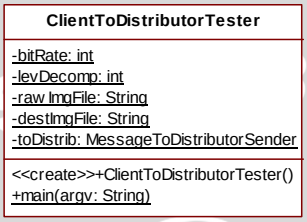
= 1

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 1 ditentukan sebuah jalur independen pengujian yaitu:

```
METHODE sendMessageToDistrib THROWS IOException;
  *this methode will send message from client to distribut
TYPE socket IS SOCKET;
TYPE os IS OUTPUT STREAM;
TYPE osw IS OUTPUT STREAM WRITER;
TYPE out IS PRINT WRITER;
TYPE hostDistrib IS STRING;
TYPE numPort IS INTEGER;
```

Jalur 1 : 1

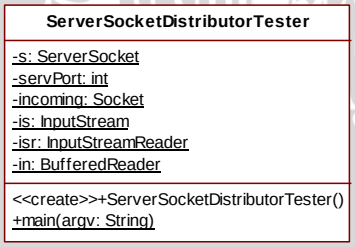
Subsistem *client* dan subsistem distributor diimplementasikan dalam sebuah komputer yang sama yaitu pada komputer dengan alamat IP 172.17.8.10 (Gambar 5.3). Dalam pengujian, digunakan klas `ClientToDistributorTester` untuk menjalankan metode `sendMessageToDistrib()`. Gambar 6.6 menunjukkan atribut dan metode dari klas `ClientToDistributorTester`.



Gambar 6.6. Diagram klas *dummy* `ClientToDistributorTester`

Sumber: Pengujian

Untuk menyediakan koneksi di sisi distributor dengan membentuk *server socket* di port 5010 pada proses pengujian, digunakan klas bantu `ServerSocketDistributorTester`. Atribut dan metode yang dimiliki oleh klas `ServerSocketDistributorTester` ditunjukkan melalui diagram klas dalam Gambar 6.7.



Gambar 6.7. Diagram klas *dummy* `ServerSocketDistributorTester`

Sumber: Pengujian

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.3.

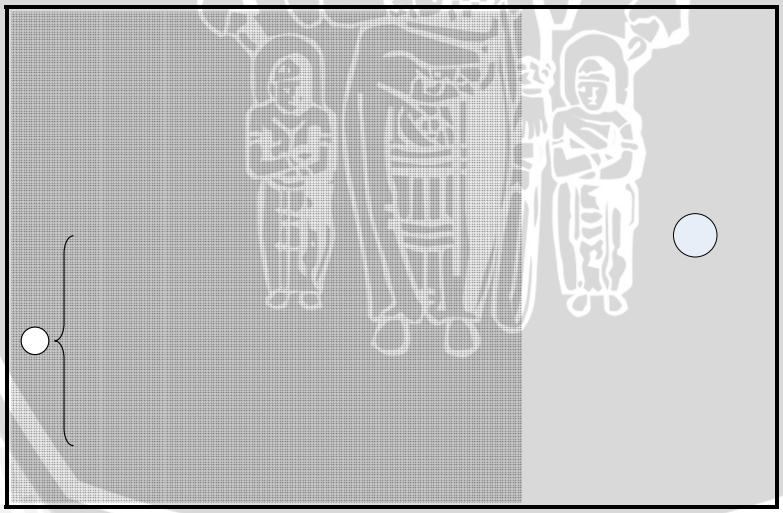
Tabel 6.3. *Test case* untuk pengujian unit operasi `sendMessageToDistrib()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	• bitRate=4 • levDecomp=5 • rawImgFile = "Sunset2.JPG", • destImgFile = "Sunset2.j2k"; • Pada distributor telah dibuat *server socket* pada port 5010 melalui klas ServerSocketDistributorTester (klas *dummy*)	Client berhasil mengirimkan pesan permintaan proses dan distributor menerima pesan tersebut. Isi pesan adalah "CLIENTORDERPROCESS|Sunset2.JPG|4|5|Sunset2.j2k"	Client berhasil mengirimkan pesan CLIENT -> CLENT READY CLIENT -> SENDING MESSAGE TO Distributor CLIENT -> ORDER A PROCESS Sukses kirim pesan BUILD SUCCESSFUL (total time: 0 seconds) Distributor menerima pesan run-single: Menunggu pesan dari client PESAN -CLIENTORDERPROCESS|Sunset2.JPG|4|5|Sunset2.j2k- TELAH DITERIMA BUILD SUCCESSFUL (total time: 10 seconds)

Sumber: Pengujian

d. Pengujian Unit untuk Operasi `receiveMessage ()` (Penerimaan Pesan dari Distributor oleh Client)

Operasi `receiveMessage()` merupakan implementasi dari algoritma penerimaan pesan dari subsistem distributor oleh subsistem *client*. Di dalam operasi `receiveMessage()` dibentuk *server socket* pada *client* di port 9031 untuk menyediakan koneksi dengan menerima pesan berakhirnya proses dari distributor. Pemodelan algoritma penerimaan pesan tersebut dalam suatu *flow graph* ditunjukkan dalam Gambar 6.8.



Gambar 6.8. pemodelan algoritma `receiveMessage ()` ke dalam *flow graph*

Sumber: Pengujian

Jumlah kompleksitas siklomatis *(cyclomatic complexity)* dari *flow graph* algoritma `receiveMessage()` yang telah dimodelkan dalam Gambar 6.8 adalah 1 buah.

V(G) = E– N + 2

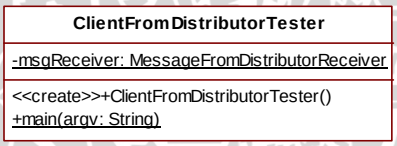
= 0 – 1 + 2

= 1

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 1 ditentukan sebuah jalur independent yaitu:

Jalur 1 : 1

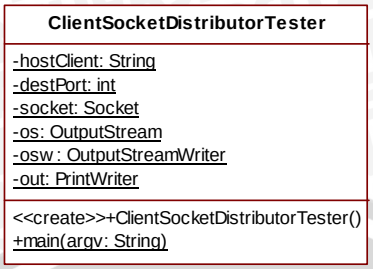
Dalam proses pengujian, untuk menjalankan metode `receiveMessage()` dari klas `MessageFromDistributorReceiver`, digunakan klas *dummy* `ClientFromDistributorTester` yang ditunjukkan dalam Gambar 6.9.



Gambar 6.9. Diagram klas *dummy* `ClientFromDistributorTester`

Sumber: Pengujian

Sedangkan untuk mengirimkan pesan kepada *client* dengan membentuk *client socket* (pada distributor) dalam proses pengujian, digunakan klas `ClientSocketDistributorTester`. Atribut dan operasi dari klas `ClientSocketDistributorTester` ditunjukkan melalui diagram klas dalam Gambar 6.10.



Gambar 6.10. Diagram klas *dummy* `ClientSocketDistributorTester`

Sumber: Pengujian

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.4.

Tabel 6.4. *Test case* untuk pengujian unit operasi `receiveMessage ()`

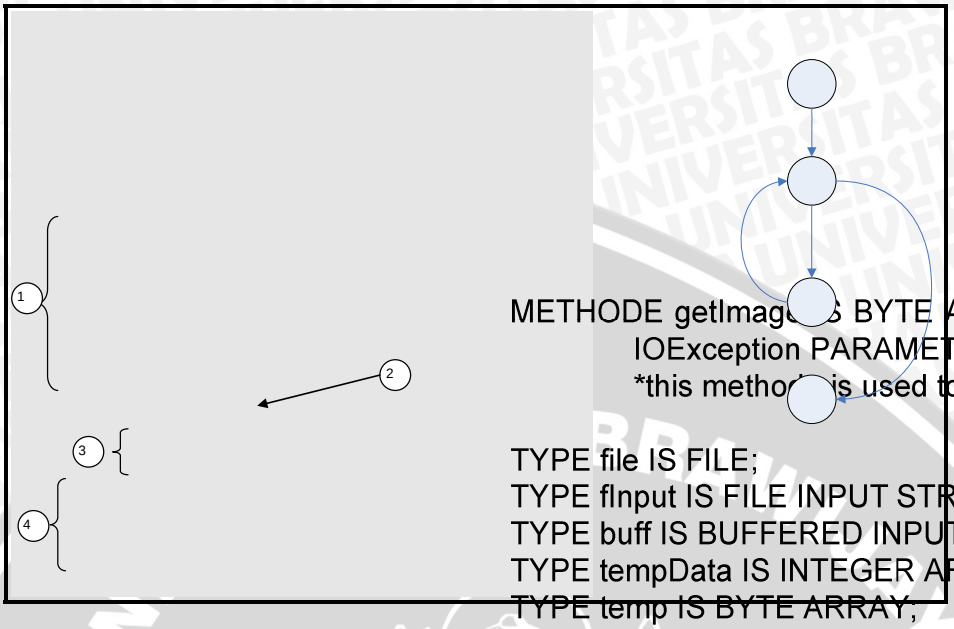
Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan

1	Pada Distributor dibuat *client socket* untuk mengirimkan pesan ke *client* . Pembentukan *client socket* dilakukan oleh klas ClientSocketDistributorTester (klas *dummy*)	Client mendapatkan pesan yang benar (PROCESSFINISHED) sebagai tanda bahwa semua rangkaian proses kompresi telah berakhir.	Pesan PROCESSFINISHED diterima
			`compile-single:`
			`run-single:`
			`CLIENT -> WAITING -PROCESSFINISHED- MESSAGE`
			`PESAN -PROCESSFINISHED- TELAH DITERIMA`
			`BUILD SUCCESSFUL (total time: 18 seconds)`

Sumber: Pengujian

e. Pengujian Unit untuk Operasi `getImage ()`

Metode `getImage()` terdapat di dalam klas ImageImplementation yang merupakan subklas dari klas `UnicastRemoteObject` dan implementasi dari *interface* `ImageInterface`. Objek dari klas `ImageImplementation` adalah *remote object* yang bisa dijalankan secara *remote* setelah di-*bound* dalam *rmiregistry*. Algoritma dari metode `getImage()` beserta pemodelannya ke dalam *flow graph* ditunjukkan dalan Gambar 6.11.



Gambar 6.11. pemodelan algoritma getImage ke dalam *flow graph*

Sumber: Pengujian

Jumlah kompleksitas siklomatis *(cyclomatic complexity)* dari *flow graph* algoritma getImage yang telah dimodelkan dalam Gambar 6.11 adalah 2 buah.

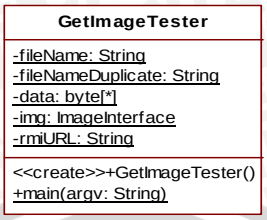
$$V(G) = E - N + 2$$
$$= 4 - 4 + 2$$
$$= 2$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 2 ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-4

Jalur 2 : 1-2-3-2-…

Untuk melakukan pengujian terhadap metode `getImage()`, digunakan klas `GetImageTester` yang berperan sebagai RMI *client* yang akan meng-*invoke* metode `getImage()` yang ada di dalam klas `ImageImplementation` secara *remote*. Atribut dan operasi dari klas `GetImageTester` seperti ditunjukkan dalam Gambar 6.12.



Gambar 6.12. Diagram klas GetImageTester
Sumber: Pengujian

Untuk pengujian, objek dari klas ImageImplementation (di dalamnya terdapat metode `getImage()`) di-*bound* pada komputer dengan alamat IP 172.17.8.9 (sebagai RMI *server*). Sedangkan klas penguji yaitu klas `GetImageTester` diletakkan pada komputer dengan alamat IP 172.17.8.10 (sebagai RMI *client*) yang akan meng-*invoke* metode `getImage()` dari objek klas `ImageImplementation` *(remote object).*

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.5.

Tabel 6.5. *Test case* untuk pengujian unit operasi `getImage ()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan

1	• fileName = "xx.txt", dimana file xx.txt berupa file kosong	Panjang temp = 0 dan akan didapatkan duplikasi file "xx.txt" yang juga berupa file kosong	Panjang temp = 0
			`compile-single:`
			`run-single:`
			`Panjang data dari temp = 0`
			`BUILD SUCCESSFUL (total time: 0 seconds)`
			Selain itu didapatkan juga file duplikasi dari xx.txt dengan ukuran 0 KB
			xx 0 KB Text
			policy 1 KB Text
			TesImageUploader 1 KB MS-D
			tesImageDownlo... 1 KB MS-D

2	• fileName = "xxx.txt", dimana file xxx.txt berupa file yang tdak kosong, yaitu berisi 12345	Panjang temp = 5, dan akan didapatkan duplikasi file "xxx.txt"	Panjang temp=5
			`compile-single:`
			`run-single:`
			`Panjang data dari temp = 5`
			`BUILD SUCCESSFUL (total time: 1 second)`
			Didapatkan file "xx.txt" hasil download
			Name Size Type
			build 4 KB XML Docum
			xxx 1 KB Text Docur

Sumber: Pengujian

f. Pengujian Unit untuk Operasi `giveImage`

Di dalam klas `ImageImplementation`, selain terdapat metode `getImage()` sebagai implementasi dari proses pengambilan *(download)* suatu data, juga terdapat metode `giveImage()` untuk mengimplementasikan proses pengiriman data. Sebagaimana metode `getImage()`, `giveImage()` juga merupakan metode *remote* yang bisa di-*invoke* oleh objek dari klas yang berada pada komputer yang lain. Dalam Gambar 6.13 ditunjukkan algoritma serta pemodelannya ke dalam *flow graph.*

Gambar 6.13. pemodelan algoritma `giveImage` ( ) ke dalam *flow graph*

Sumber: Pengujian

Jumlah kompleksitas siklomatis dari *flow graph* algoritma giveImage dalam Gambar 6.13 adalah 1 buah.

V(G) = E – N + 2

= 0 – 1 + 2

= 1

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yang di ditentukan sebuah basis set dari jalur independent yaitu:

Jalur 1 : 1

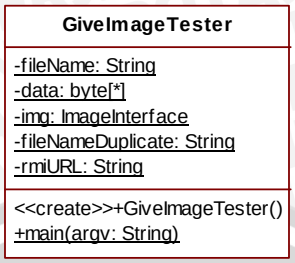
Klas `GiveImageTester` merupakan klas bantu untuk proses pengujian terhadap metode `giveImage()`. Klas `GiveImageTester` berperan sebagai RMI *client* yang akan meng-*invoke* metode `giveImage()` yang ada di dalam klas ImageImplementation secara *remote.* Atribut dan operasi dari klas GiveImageTester ditunjukkan dalam Gambar 6.14

```
img IS BYTE A
IS STRING TRHOWS FileNotFoundException,
*this methode is used to merge images*

TYPE filedata IS BYTE ARRAY;
TYPE file IS FILE;
TYPE output IS FILE OUTPUT STREAM;

filedata = img;
CREATE OBJECT file AS FILE (imgFile);
CREATE OBJECT output AS FILE OUTPUT STREAM
CALL output.WRITE (filedata);
CALL output.CLOSE;

END giveImage
```

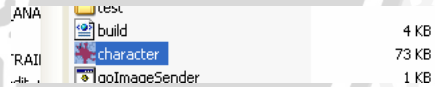


Gambar 6.14 Diagram klas `GiveImageTester`
Sumber: Pengujian

Seperti proses pengujian pada metode `getImage()`, pada pengujian `giveImage()`, objek dari klas `ImageImplementation` (di dalamnya terdapat metode `giveImage()`) di-*bound* pada komputer dengan alamat IP 172.17.8.9 (sebagai RMI *server*). Sedangkan klas penguji yaitu klas `GiveImageTester` diletakkan pada komputer dengan alamat IP 172.17.8.10 (sebagai RMI *client*) yang akan meng-*invoke* metode `giveImage()` dari objek klas `ImageImplementation` *(remote object)*.

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.6.

Tabel 6.6. *Test case* untuk pengujian unit operasi `giveImage`

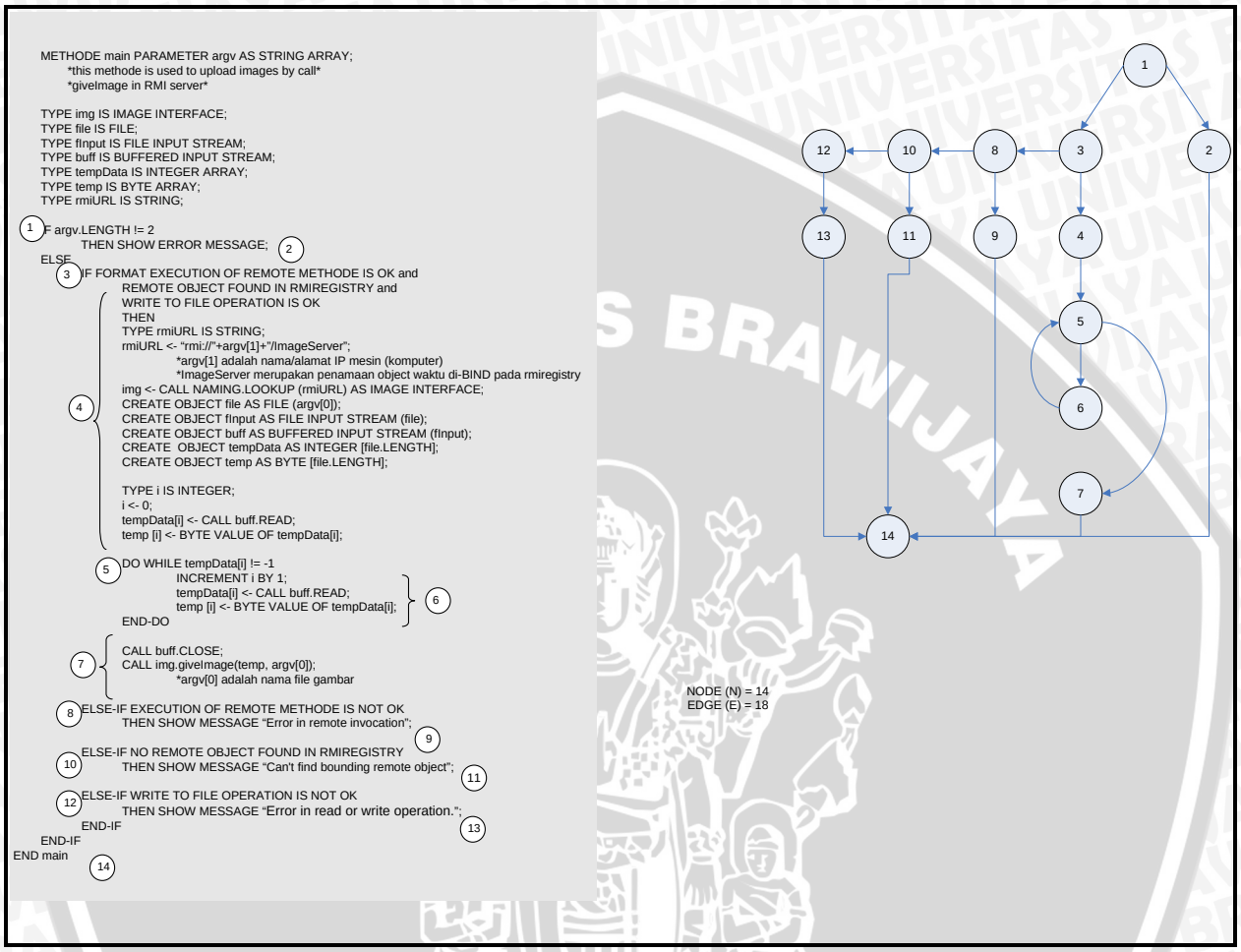
Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	• imgFile = "character.JPG" • img = data, dimana data merupakan argumen dari metode pemanggil dalam klas penguji GiveImageTester.	Didapatkan file "character.JPG" pada RMI *server* yaitu computer 172.17.8.9	Didapatkan file "character.JPG" 

Sumber: Pengujian

g. Pengujian Unit untuk Metode `main( )` pada Klas `ImageUploader`

Klas `ImageUploader` merupakan klas yang akan memanggil *remote object* yang telah di-*bound* pada *server* RMI. Oleh karena itu, sebelum proses pengujian dilakukan, *server* RMI harus dalam keadaan aktif yaitu dengan mengaktifkan RMI *registry* dan mem-*bound remote object pada* RMI *registry*.. Algoritma dari operasi

`main()` dari klas `ImageUploader` beserta pemodelan dari algoritma tersebut ke dalam *flow graph* ditunjukkan dalam Gambar 6.15.



Gambar 6.15. pemodelan algoritma metode `main()` dari klas `ImageUploader` ke dalam *flow graph*

Sumber: Pengujian

Jumlah kompleksitas siklomatis dari *flow graph* dalam Gambar 6.15 adalah 6 buah.

$$V(G) = E - N + 2$$
$$= 18 - 14 + 2$$
$$= 6$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 6 ditentukan enam buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-14

Jalur 2 : 1-3-4-5-7-14

Jalur 3 : 1-3-4-5-6-5-…

Jalur 4 : 1-3-8-9-14

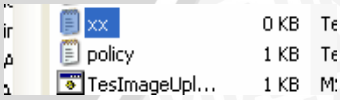
Jalur 5 : 1-3-8-10-11-14

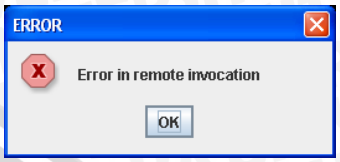
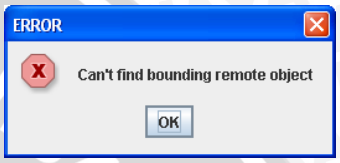
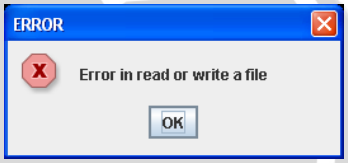
Jalur 6 : 1-3-8-10-12-13-14

Untuk menguji metode `main()` dari klas `ImageUploader` tersebut dilakukan dengan menjalankan (mengeksekusi) klas `ImageUploader` dengan dua (2) buah argumen. Argumen pertama untuk memasukkan nama file yang akan di-*upload*, sedangkan argument yang kedua digunakan untuk memasukkan alamat IP komputer kompresor yang dituju (berperan sebagai RMi *server*)digunakan sebuah klas bantu yaitu klas `ImageUploaderTester`.

Penentuan kasus uji *(test case)* untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.7.

Tabel 6.7. *Test case* untuk pengujian unit operasi `main()` pada klas `ImageUploader`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	argv.length = 0 (emngeksekusi klas ImageUploader tanpa melewatkan argumen)	Keluar peringatan untuk menjalankan klas ImageUploader dengan argumen berupa nama file yang akan di*upload* dan alamat IP tujuan *upload.*	Keluar peringatan 
2	• argv.length = 2 • argv[0] = “xx.txt” (file ada tetapi kosong) • argv[1]=”172.17.8.9” • rmiURL = “rmi://172.17.8.9/ImageServer”; • rmiregistry telah dijalankan • objek dari ImageImplementation telah do-*bound* dengan nama ImageServer pada rmiregistry	• Panjang temp (temp.length) adalah 0 • Didapatkan file “xx.txt” pada mesin 172.17.8.9	• Panjang temp = 0  • didapatkan file “xx.txt” di 172.17.8.9 dengan ukuran = 0 KB 
3	• argv.length = 2 • argv[0] = “xx.txt” (file ada dan berisi data 12345) • argv[1]=”172.17.8.9” • rmiURL = “rmi://172.17.8.9/ImageServer”;	• Panjang temp (temp.length) adalah 5 • Didapatkan file “xx.txt” pada mesin 172.17.8.9	• Panjang temp = 5  • didapatkan file “xx.txt” di 172.17.8.9 dengan ukuran != 0

	• rmiregistry telah dijalankan • objek dari ImageImplementation telah do-*bound* dengan nama ImageServer pada rmiregistry		KB 
4	• argv.length = 2 • argv[0] = "xx.txt" • argv[1] = "172.17.8.1", dimana dalam computer 172.17.8.1 tidak difungsikan sebagai RMI server	Keluar peringatan ada error pada proses pemanggilan metode dari *remote object*	Muncul peringatan error 
5	• argv.length = 2 • argv[0] = "xx.txt" • argv[1] = "172.17.8.9" • Pada 172.17.8.9 rmiregistry sudah dijalankan, tetapi tidak ada objek yang di-*bound* di rmiregistry	Keluar peringatan ada error bahwa tidak ditemukan *remote object* yang di-*bound* pada rmiregistry	Muncul peringatan error 
6	• argv.length = 2 • argv[1]="172.17.8.9" • argv[0] = "xxxxxx.txt" sedangkan file characterrrr.JPG tidak ada.	Keluar peringatan bahwa terjadi eror pada pembacaan atau penulisan file	Muncul peringatan error 

Sumber: Pengujian

6.1.2. Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Dalam melakukan pengujian validasi digunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan.

a. Kasus Uji Validasi

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem, pada setiap kebutuhan *(requirement)* dilakukan proses pengujian dengan kasus uji masing-masing.

a. Kasus Uji Permintaan Proses Kompresi

Nama kasus uji : Kasus Uji Permintaan Proses Kompresi

Obyek uji : Kebutuhan no.1

Tujuan pengujian :Untuk menguji validitas dari kinerja subsistem client dalam memenuhi kebutuhan fungsionalitas untuk memasukkan permintaan proses kompresi dari operator pengguna sistem, disertai dengan memasukkan paramtere proses kompresi (nilai bit rate dan level dekomposis), nama file gambar mentah dan file tujuan untuk menyimpan gambar hasil kompresi.

Prosedur uji :

1. Menjalankan subsistem *client*
2. Pada menu utama *client*, pilih menu untuk masuk ke *form* permintaan proses kompresi dengan menekan tombol “start” .
3. Setelah muncul *form* permintaan proses kmpresi, pilih file gambar mentah (letak file gambar mentah di D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp) dengan menekan tombol “browse”, pilih nama file tujuan (terletak di C:\Documents and Settings\ASISTEN\My Documents\psnr4-2.j2k) dengan menekan tombol “save as”, memasukkan nilai *bit rate* dengan 4 pada *field* isian *bit rate*, menuliskan nilai level dekomposisi dengan 5, dan menekan tombol “start”.
4. Setelah proses kompresi selesai (ditandai dengan diterimanya pesan proses selesai oleh *client* dari distributor), tekan tombol “calculate result variable” untuk menampilkan nilai PSNR, MSE dan waktu total proses kompresi dan tombol “show result image” untuk menampilkan gambar hasil kompresi.

Hasil yang diharapkan : Subsistem client bisa menerima parameter proses kompresi (nilai bitrate = 4 dan level dekomposisi = 5), file gambar mentah dengan lokasi

"D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp" dan file tujuan yang dimasukkan oleh pengguna dengan lokasi "C:\Documents and Settings\ASISTEN\My Documents\psnr4-2.j2k" dan kemudian mengirimkan masukan dari pengguna tersebut bersama pesan permintaan proses ke distributor.

b. Kasus Uji Penampilan Gambar

Nama kasus uji : Kasus Uji Penampilan Gambar

Obyek uji : Kebutuhan no.2

Tujuan pengujian : Untuk mengetahui validitas dari kemampuan subsistem client untuk menampilkan gambar dari suatu file yang dipilih oleh pengguna sistem.

Prosedur uji :

1. Menjalankan subsistem *client*
2. Pada menu utama *client*, pilih menu untuk masuk ke *form* untuk *display* gambar dengan menekan tombol "display" .
3. Setelah muncul *form* penampilan gambar, pilih file gambar pada lokasi "D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp" yang akan ditampilkan dengan masuk ke menu "File" dan "Open".
4. Setelah gambar ditampilkan, ubah skala tampilan (semula 100%) dengan 50 %.

Hasil yang diharapkan :Client bisa menampilkan gambar dari file "D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp" yang telah dipilih oleh pengguna dan menampilkannya dalam skala tampilan yang ditentukan oleh *user* (50 %) dalam suatu *form* tertentu.

c. Kasus Uji Permintaan Panduan Client

Nama kasus uji : Kasus Uji Permintaan Panduan Client

Obyek uji : Kebutuhan no.3

Tujuan pengujian : Untuk menguji kemampuan client untuk menampilkan item-item panduan bagi pengguna. Baik item panduan pada menu utama client maupun item panduan pada *form* utama permintaan proses kompresi *client.*

Prosedur uji :

1. Menjalankan subsistem *client*
2. Pada menu utama *client,* pilih menu untuk masuk ke *form* panduan *client* dengan menekan tombol "help" untuk menampilkan *form* panduan dari menu utama *client* .
3. Atau masuk ke dalam *form* utama permintaan proses kompresi (dengan menekan tombol "start" di menu utama *client*) dan kemudian menekan tombol "help" untuk menampilkan form panduan dari *form* permintaan proses kompresi.

Hasil yang diharapkan : Client bisa memberikan list dari item-item bantuan yang dibutuhkan oleh pengguna untuk menggunakan elemen-elemen dari subsistem *client,* baik pada *form* menu utama maupun pada *form* utama permintaan proses kompresi pada client.

d. Kasus Uji Pemilihan File

Nama kasus uji : Kasus Uji Pemilihan File

Obyek uji : Kebutuhan no.4

Tujuan pengujian :Untuk mengetahui kesesuaian kinerja dari client dengan fungsionalitas yang dimiliki oleh client untuk menyediakan fasilitas pemilihan file.

Prosedur uji :

1. Menjalankan subsistem *client*
2. Pada menu utama *client,* pilih menu untuk masuk ke *form*

permintaan proses kompresi dengan menekan tombol “start” .

3. Setelah muncul *form* permintaan proses kompresi, pilih file gambar mentah dengan menekan tombol “browse” untuk memilih file (membuka file) dan kemudian menyeleksi suatu file (“D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp”) dan menekan tombol “open”.
4. Menekan tombol “save as” untuk memiih file (menuliskan file) tujuan pada “C:\Documents and Settings\ASISTEN\My Documents\psnr4-2.j2k”.

Hasil yang diharapkan : Client bisa menampilkan sebuah *form* untuk pemilihan file dan setelah pegguna memilih suatu file (“D:\ASISTEN\imam\ALHAMDULILLAH\ALLAHU AKBAR\DATA_GAMBAR\data_4.bmp” atau “C:\Documents and Settings\ASISTEN\My Documents\psnr4-2.j2k”, subsistem client bisa mendapatkan file yang diseleksi oleh pengguna tersebut.

e. Kasus Uji Pengalokasian IP Kompresor

Nama kasus uji : Kasus Uji Pengalokasian IP Kompresor

Obyek uji : Kebutuhan no.5

Tujuan pengujian : Untuk mengetahui kemampuan subsistem distributor dalam menyediakan fasilitas bagi pengguna untuk memasukkan alamat IP dari komputer-komputer yang akan digunakan sebagai kompresor.

Prosedur uji :

1. Menjalankan subsistem distributor
2. Pada menu utama distributor, pilih menu untuk masuk ke *form* pengalokasian alamat IP untuk kompresor dengan menekan tombol “Allocate IP Compressor” .
3. Setelah muncul *form* pengalokasian alamat IP kompresor,

dituliskan alamat IP 172.17.8.3 pada *field* alamat IP dan dilanjutkan dengan menekan tombol “Add”. Tuliskan kembali IP 172.17.8.8 pada *field* alamat IP dan dilanjutkan dengan menekan tombol “Add”.

4. Setelah semua IP kompresor selesai dimasukkan, akhiri dengan menekan tombol “set” dan “close” untuk keluar dari *form* pengalokasian alamat IP kompresor.

Hasil yang diharapkan : Subsistem distributor mendapatkan list alamat IP dari komputer-komputer yang akan digunakan sebagai kompresor. Daftar alamat IP yang telah didapatkan ini yang akan dicek oleh distributor status keaktifan atau kesiapannya untuk digunakan.

f. Kasus Uji Persiapan Distributor

Nama kasus uji : Kasus Uji Persiapan Distributor

Obyek uji : Kebutuhan no.6

Tujuan pengujian : Untuk mengecek validitas fungsi dari subsistem distributor pada saat persiapan menuju status *ready* (siap) untuk menerima pesan permintaan proses dari *client* .

Prosedur uji :

1. Menjalankan subsistem distributor
2. Pada menu utama tekan tombol “start” sehungga akan muncul *form* utama distributor disertai dengan tampilan IP kompresor yang aktif (172.17.8.3 dan 172.17.8.8) serta siap digunakan.
3. Setelah muncul *form* utama distributor, alamat IP 172.17.8.3, 172.17.8.8 dari kompresor yang ditampilkan. Dan setiap pemilihan IP diakhiri dengan menekan tombol “Add”.
4. Menekan tombol “start” setelah selesai memilih IP kompreor yang digunakan.

Hasil yang diharapkan : Distributor berada dalam kondisi *ready* untuk menerima permintaan proses dari *client* dan siap untuk mengatur

proses kompresi (meneruskan permintaan kompresi kepada kompresor 172.17.8.3, 172.17.8.8) saat pesan permintaan proses dari client sudah diterima.

g. Kasus Uji Permintaan Panduan Distributor

Nama kasus uji : Kasus Uji Permintaan Panduan Distributor

Obyek uji : Kebutuhan no.7

Tujuan pengujian : Untuk menguji kemampuan distributor untuk menampilkan item-item panduan bagi pengguna. Baik item panduan pada menu utama distributor maupun item panduan pada *form* utama proses distributor.

Prosedur uji :

1. Menjalankan subsistem distributor.
2. Pada menu utama distributor, pilih menu untuk masuk ke *form* panduan distributor dengan menekan tombol "help" untuk menampilkan *form* panduan dari menu utama distributor .
3. Atau masuk ke dalam *form* utama proses distributor (dengan menekan tombol "start" di menu utama distributor) dan kemudian menekan tombol "help" untuk menampilkan form panduan dari penggunaan *form* utama proses distributor.

Hasil yang diharapkan : Distributor bisa memberikan list dari item-item bantuan yang dibutuhkan oleh pengguna untuk menggunakan elemen-elemen dari subsistem distributor, baik pada *form* menu utama maupun pada *form* utama permintaan proses kompresi pada distributor.

h. Kasus Uji Pengaturan Proses Kompresi

Nama kasus uji : Kasus Uji Pengaturan Proses Kompresi

Obyek uji : Kebutuhan no.8

Tujuan pengujian : Untuk memastikan proses pengaturan kompresi (pemotongan, pengiriman/pengambilan gambar dari dan ke

kompresor, penggabungan gambar hasil kompresi serta pengiriman pesan kepada *client* saat proses penggabungan gambar sudah selesai) yang dilakukan oleh distributor sudah valid sesuai dengan daftar kebutuhan.

Prosedur uji :

1. Menjalankan distributor dan mengkondisikannya dalam keadaan *ready* (dengan melakukan langkah sesuai dengan prosedur uji Persiapan Distributor pada poin f)
2. Setelah distributor dalam keadaan *ready* , kirimkan pesan permintaan proses dari subsistem *client* (dengan menekan tombol “start” pada *form* permintaan proses kompresi di *client*)

Hasil yang diharapkan : Didapatkan gambar hasil akhir kompresi yang terletak pada “C:\Documents and Settings\ASISTEN\My Documents\” dengan nama file “psnr4-2.j2k” dan pesan bahwa rangkaian proses telah berakhir terkirim ke *client*.

i. Kasus Uji Kompresi Gambar

Nama kasus uji : Kasus Uji Kompresi Gambar

Obyek uji : Kebutuhan no.9

Tujuan pengujian : Untuk memastikan validitas dari subsistem kompresor untuk melakukan proses kompresi terhadap potongan gambar yang telah didapatkan dari distributor.

Prosedur uji :

1. Menjalankan subsistem kompresor sehingga kompresor akan berada pada kondisi *ready* siap untuk menerima pesan dari distributor.
2. Mengirimkan pesan dari distributor (berupa pesan pengiriman gambar mentah).

Hasil yang diharapkan : Didapatkan file J2K dengan nama “merged0”+identifier+”.j2k” pada kompresor 172.17.8.3

dan “merged1”+identifier+”.j2k” pada kompresor 172.17.8.8, dan kompresor memberikan pesan kepada distributor bahwa gambar J2K telah dapat diambil. Identifier merupakan pemberian nomor acak sebagai identitas tiap proses yang unik antar proses satu dengan yang lain.

j. Kasus Uji Pengukuran Waktu Kompresi

Nama kasus uji : Kasus Uji Pengukuran Waktu Kompresi

Obyek uji : Kebutuhan no.10

Tujuan pengujian : Untuk mengetahui apakah waktu yang diperlukan untuk proses kompresi secara terdistribusi lebih singkat daripada waktu kompresi secara *standalone* untuk suatu obyek gambar yang sama.

Keterangan : Sebagai bahan analisis unjuk kerja sistem ditinjau dari segi waktu total yang dibutuhkan untuk proses kompresi, digunakan enam (6) buah data gambar mentah dengan ukuran yang bervariasi, seperti ditunjukkan pada Tabel 6.10. Untuk setiap gambar dilakukan kompresi dengan 2, 4, 6 dan 8 buah kompresor.

Tabel 6.10 Besar ukuran file dari masing-masing gambar

Data Gambar	Besar Ukuran Gambar (KB)
Gambar 1	3716
Gambar 2	5373
Gambar 3	10736
Gambar 4	23637
Gambar 5	59015
Gambar 6	71720

Sumber: Pengujian

Prosedur uji :

1. Menjalankan subsistem kompresor (sehingga kompresor berada pada kondisi *ready*)
2. Menjalankan subsistem distributor, dan menyiapkannya

sehingga *ready* untuk menerima pesan dari *client.*

3. Menjalankan *client,* dan memasukkan nama file gambar mentah (dari gambar 1, 2, 3, 4, 5 dan 6), nama file tujuan untuk setiap proses kompresi serta parameter-parameter proses kompresi (nilai *bitrate* = 4, level dekomposisi = 5).
4. Menekan tombol "start" pada *form* permintaan proses kompresi di *client* sebagai *event* untuk memulai rangkaian proses kompresi.
5. Mencatat waktu dari setiap proses kompresi.

Hasil yang diharapkan : Waktu yang diperlukan untuk melakukan kompresi oleh sistem terdistribusi lebih cepat daripada waktu yang dibutuhkan oleh sistem *standalone* untuk melakukan kompresi terhadap boyek gambar yang sama.

k. Kasus Uji Penilaian Kualitas Gambar Hasil Kompresi

Nama kasus uji : Kasus Uji Penilaian KualitasGambar Hasil Kompresi

Obyek uji : Kebutuhan no.11

Tujuan pengujian : Untuk mengetahui nilai PSNR dari gambar hasil kompresi dari sistem terdistribusi.

Keterangan : Serupa dengan pengujian waktu total proses kompresi pada poin j, untuk bahan analisis unjuk kerja sistem ditinjau dari segi nilai PSNR sebagai kualitas gambar hasil kompresi, digunakan enam buah data gambar mentah dengan ukuran yang bervariasi. Masing-masing data gambar diterapkan untuk sistem terdistribusi dengan 2, 4, 6 atau 8 buah kompresor.

Prosedur uji :

1. Menjalankan subsistem kompresor (sehingga kompresor berada pada kondisi *ready*)
2. Menjalankan subsistem distributor, dan menyiapkannya

sehingga *ready* untuk menerima pesan dari *client.*

3. Menjalankan *client,* dan memasukkan nama file gambar mentah, nama file tujuan serta parameter-parameter proses kompresi (nilai *bitrate* = 4, level dekomposisi = 5)
4. Menekan tombol "start" pada *form* permintaan proses kompresi di *client* sebagai *event* untuk memulai rangkaian proses kompresi.
5. Mencatat nilai PSNR dari masing-masing hasil kompresi.

Hasil yang diharapkan : Nilai PSNR dari gambar hasil kompresi oleh sistem terdistribusi lebih besar dari 42 dB.

l. Kasus Uji Penelusuran Kode Program

Nama kasus uji : Kasus Uji Penelusuran Kode Program

Obyek uji : Kebutuhan no.12

Tujuan pengujian : Memastikan bahwa semua kode program untuk membangun sistem adalah dalam bahasa pemrograman Java.

Hasil yang diharapkan : Semua kode program adalah dalam bahasa pemrograman Java.

Prosedur uji :

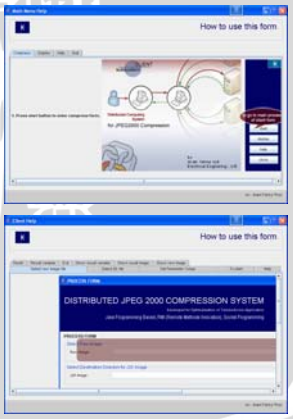
1. Melakukan penelusuran kode kode program *(source code)* terhadap semua klas yang digunakan untuk membangun sistem.

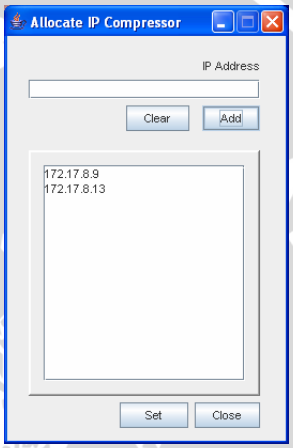
b. Hasil Pengujian Validasi

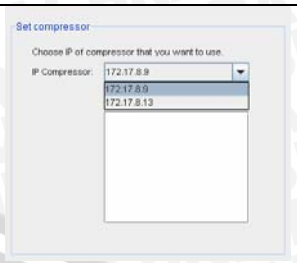
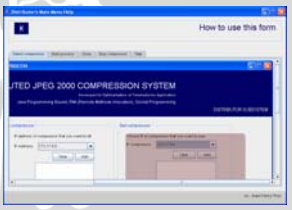
Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan VI.1.3.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.11.

Tabel 6.11 *Test case* untuk pengujian validasi

No.	*Nama Kasus uji*	Hasil yang diharapkan	Hasil yang didapatkan	Status validitas
1	Kasus Uji Permintaan Proses Kompresi	Subsistem client bisa menerima parameter proses kompresi (nilai bitrate dan level dekomposisi), file gambar	Client mampu menerima masukan dari pengguna dan kemudian mengirimkan	Valid

		mentah dan file tujuan yang dimasukkan oleh *user* dan kemudian mengirimkan masukan dari *user* tersebut bersama pesan permintaan proses ke distributor.	masukan tersebut bersama pesan ke distributor	
2	Kasus Uji Penampilan Gambar	Client bisa menampilkan gambar dari file yang telah dipilih oleh pengguna dan menampilkannya dalam skala tampilan yang ditentukan oleh *user* dalam suatu *form* tertentu.	Subsistem client telah dapat menampilkan gambar dari suatu file gambar yang diinginkan oleh pengguna pada suatu *form* dengan skala tampilan bis diubah.  Demikian juga bisa diubah skala tampilannya.	Valid
3	Kasus Uji Permintaan Panduan Client	Client bisa memberikan list dari item-item bantuan yang dibutuhkan oleh pengguna untuk menggunakan elemen-elemen dari subsistem *client.*	Subsistem client telah dapat memberikan dua buah *form* yang berisi tentang pon-poin bantuan untuk menu utama pada client maupun bantuan untuk penggunaan *form* permintaan proses kompresi pada client 	Valid
4	Kasus Uji Pemilihan File	Client bisa memberikan fasilitas untuk memilih suatu file tertentu, sehingga bisa digunakan untuk menyeleksi file untuk gambar mentah, file gambar tujuan, maupun file gambar yang ingin ditampilkan.	Untuk memilih suatu file, pada client telah ada *form* untuk memilih suatu lokasi file.	Valid

				
5	Kasus Uji Pengalokasian IP Kompresor	Muncul *form* untuk memasukkan alamat IP yang diinginkan menjadi kompresor, dimana status keaktifan dari masing-masing kompresor yang telah dimasukkan akan dicek pada saat *form* utama proses distributor di-*load*.	Subsistem distributor telah menyediakan sebuah *form* untuk memasukkan daftar alamat IP dari kompresor yang ingin digunakan.  Ketika ditekan tombol set, distributor mengeset alamat IP yang telah dimasukkan sebagai daftar kompresor yang akan dicek keaktifannya kemudian.	Valid
6	Kasus Uji Persiapan Distributor	Distributor melakukan pengecekan kompresor dengan alamat IP berapa saja yang aktif (dengan indikasi membalas pesan yang telah dikirimkan oleh distributor) dan siap digunakan sebagai kompresor. Hasil pengecekan ditampilkan dalam sebuah *combo box.* Setelah dipilih beberapa alamat IP dan ditekan tombol "start" diharapkan distributor berada pada kondisi siap menerima permintaan proses dari *client*	Distributor berhasil mengecek status keaktifan dari daftar kompresor yang telah dimasukkan oleh pengguna pada *form* pengalokasian alamat IP untuk kompresor. Proses pengecekan dilakukan pada saat *form* proses utama distributor di-*load* dan hanya IP dari kompresor yang aktif yang akan ditampilkan menjadi item kompresor yang siap digunakan.	Valid

			 Setelah dipilih beberapa kompresor yang digunakan dan ditekan tombol "start" distributor dalam kondisi *ready* untuk menerima permintaan proses.	
7	Kasus Uji Permintaan Panduan Distributor	Muncul *form* pada distributor yang berisi item-item bantuan untuk menggunakan distributor.	Distributor telah dapat memberikan dua buah *form* yang berisi tentang pon-poin bantuan untuk menu utama distributor maupun bantuan untuk penggunaan *form* proses utama distributor  	Valid
8	Kasus Uji Pengaturan Proses Kompresi	Didapatkan hasil akhir proses kompresi dengan menggabungkan file-file gambar J2K dari masing-masing kompresor.	Berhasil didapatkan hasil akhir proses kompresi dalam file J2K dari hasil penggabungan beberapa file hasil kompresi dari tiap kompresor.	Valid
9	Kasus Uji Kompresi Gambar	Kompresor bisa menerima kiriman gambar dari distributor dan mengkompresnya ke dalam format J2K, dan setelah selesai kemudian mengirimkan pesan ke distributor untuk mengambil hasil kompresi (J2K)	Berhasil mengkompres gambar mentah yang diterimanya ke dalam format J2K dan kemudian mengirimkan pesan ke distributor	Valid

10	Kasus Uji Pengukuran Waktu Kompresi	Waktu total proses kompresi yang dilakukan secara terdistribusi lebih cepat daripada waktu kompresi secara standalone	Untuk data Gambar 1, 2, 3, 4	Tidak valid
			Untuk data Gambar 5, 6	Valid
11	Kasus Uji Penilaian KualitasGambar Hasil Kompresi	Nilai PSNR dari hasil kompresi lebih besar dari 42 dB	Nilai PSNR dari semua gambar hasil kompresi adalah di atas 42 dB	Valid
12	Kasus Uji Penelusuran Kode Program	Semua kode program ditulis dalam bahasa Java	Semua kode program yang digunakan untuk membangun elemen-elemen sistem ditulis dalam bahasa Java	Valid

Sumber: Pengujian

6.1.3. Pengujian Unjuk Kerja Sistem

Pengujian unjuk kerja sistem kompresi terdistribusi ini ditujukan untuk mengambil data yang akan digunakan sebagai acuan analisis terhadap kinerja sistem dan berbagai faktor yang mempengaruhinya. Data-data tersebut diantaranya adalah data waktu pemotongan dan penggabungan gambar, waktu pengiriman gambar, ukuran dan kualitas gambar hasil kompresi, serta waktu total proses kompresi sebagai perbandingan dengan waktu kompresi oleh sebuah mesin *standalone*.

Dalam proses pengujian unjuk kerja sistem ini digunakan 6 buah gambar yang didapatkan dari hasil *scan* foto *rontgen* dengan sinar-x dengan ukuran yang bervariasi. Ukuran dari masing-masing data gambar telah ditunjukkan pada Tabel 6.10.

Spesifikasi perangkat keras dari komputer yang digunakan dalam proses pengujian, baik difungsikan sebagai subsistem *client*, distributor maupun kompresor ditunjukkan pada Tabel 6.12.

Tabel 6.12 Spesifikasi *hardware* komputer yang digunakan dalam pengujian

Subsistem	**Simbol**	**Spesifikasi *Hardware***
Client	C	Processor Pentium IV, 2,66 GHz
Distributor	D	Memori 1 GB
		Hard Disk 80 GB
Kompresor	K2	Processor Pentium IV, 2,66 GHz
(8 buah)	K1	Memori 256 MB
	K4	Hard Disk 40 GB

	K5	
	K3	Processor Pentium IV, 2,66 GHz
	K6	Memori 256 MB
	K7	Hard Disk 80 GB
	K8	

Sumber: Pengujian

6.1.4.1 Kajian Secara Teori *(Big O Notation)*

Kajian secara teori dilakukan dengan menghitung nilai dari O Besar *(Big O),* dengan mengambil acuan dari algoritma yang telah disusun.

a. Tujuan

Mengukur waktu yang diperlukan oleh sebuah algoritma dengan menghitung banyaknya operasi/instruksi yang dieksekusi.

b. Prosedur Pengujian

- Menentukan kompleksitas (*complexity*) waktu T(n), diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n.
- Menentukan notasi O Besar, yang merupakan notasi kompleksitas waktu.
- Berikan suatu harga *n* tertentu pada nilai kompleksitas, untuk menghitung waktu *T(n)* yang diperlukan untuk melakukan operasi algoritma dengan masukan *n*.
- Analisis hasil dari pengujian diatas.

c. Kasus Uji

- **Algoritma Crop**
  - **Notasi O Besar**

    Proses pembagian gambar menjadi beberapa bagian sesuai dengan jumlah kompresor yang kan digunakan (lihat pada Algoritma 5.1):

    T(n) =1+1+1+1+1+n(1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+n

(1)+1+1+1+1+n(1)+1+1)

= 5+n(18+n+4+n+2)

= 5+n(2n+24)

= $2n^2$+24n+5

Tmax(n) = O($2n^2$+24n+5)

= O(n^2)

- **Hasil Pengujian**

Tabel 6.13 Nilai dari *T(n)*

N	4	16	256	1024	10^4	10^5
n^2	16	16^2	256^2	1024^2	10^8	10^{10}

Sumber: Pengujian

- **Analisis**

Dari Tabel 6.13 didapatkan kompleksitas berupa linear, sehingga jika nilai *n* dijadikan dua kali semula, maka waktu pelaksanaan algoritma *T(n)* juga dua kali semula.

- **Algoritma Merge**

- **Notasi O Besar**

Algoritma proses penyatuan kembali gambar hasil proses kompresi dari masing-masing kompresor menjadi satu kesatuan seperti sedia kala (Algoritma 5.2):

T(n) = 1+n(1+1+1)+1+1+1+1+n(1+1)+1+1

= 1+3n+4+2n+2

= 5n+7

Tmax(n) = O(5n+7)

= O(n)

- **Hasil Pengujian**

Tabel 6.14 Nilai dari *T(n)*

N	**4**	**16**	**256**	**1024**	**10^4**	**10^5**
N	4	16	256	1024	10^4	10^5

Sumber: Pengujian

- **Analisis**

  Dari Tabel 6.14 didapatkan kompleksitas berupa linear, sehingga jika nilai *n* dijadikan dua kali semula, maka waktu pelaksanaan algoritma *T(n)* juga dua kali semula.

- **Algoritma Sending Image**

- **Notasi O Besar**

  Algoritma Get Image (Algoritma 5.3):

  T1(n) = 1+1+1+1+1+1+1+n(1+1+1)+1+1
  = 7+3n+2
  = 3n+9

  Algoritma Give Image (Algoritma 5.4):

  T2(n) = 1+1+1+1+1
  = 5

  T(n) = T1(n)+T2(n)
  = 3n+9+5
  = 3n+14

  Tmax(n) = O(3n+14)
  = O(n)

- **Hasil Pengujian**

Tabel 6.15 Nilai dari *T(n)*

N	**4**	**16**	**256**	**1024**	**10^4**	**10^5**

n	4	16	256	1024	10^4	10^5

Sumber: Pengujian

- **Analisis**

  Dari Tabel 6.15 didapatkan kompleksitas linear untuk Sending Image, sehingga jika nilai *n* dijadikan dua kali semula, maka waktu pelaksanaan algoritma *T(n)* juga dua kali semula. Tetap jika dianalisis lebih detail, dapat diketahui bahwa proses Get Image akan lebih lama daripada Give Image, hal ini disebabkan karena pada algoritma Get Image, dilakukan pembacaan tiap bagian *stream* dari gambar. Setelah selesai baru dituliskan ke dalam suatu file.

- **Algoritma Sending Message**

- **Notasi O Besar**

  Algoritma Send Message (Algoritma 5.7):

  T1(n) = 1+1+1+1+1+1+1
  = 7

  Algoritma Receive Message (Algoritma 5.8):

  T2(n) = 1+1+1+1+1+1+1+1+1
  = 9

  T(n) = T1(n)+T2(n)
  = 7+9
  = 14

  Tmax(n) = O(14)
  = $O(n^0)$

- **Hasil Pengujian**

Tabel 6.16 Nilai dari *T(n)*

N	**4**	**16**	**256**	**1024**	**10^4**	**10^5**
n^0	1	1	1	1	1	1

Sumber: Pengujian

- **Analisis**

  Dari Tabel 6.16, terlihat bahwa pada algoritma serah terima pesan antara subsistem yang satu dengan yang lain tidak dipengaruhi oleh variabel n. Sehingga jalannya waktu dari segi algoritma akan konstan

6.1.4.2 Kajian Secara Aktual

a. Pengujian Waktu Pemotongan Gambar

Data hasil pengujian waktu yang dibutuhkan untuk melakukan proses pemotongan gambar seperti ditunjukkan pada Tabel 6.17. Pengujian untuk mendapatkan waktu pemotongan gambar dilakukan terhadap objek gambar dengan ukuran yang bervariasi dan setiap gambar diterapkan terhadap sistem dengan 2, 4, 6 dan 8 buah kompresor. Diasumsikan dari proses pemotongan gambar akan membutuhkan waktu yang semakin bertambah besar seiring dengan semakin besarnya ukuran objek gambar yang dipotong. Tetapi jumlah elemen pemotongan (sesuai dengan jumlah kompresor), tidak akan berpengaruh besar terhadap perbedaan waktu pemotongan (relatif sama), karena jumlah piksel dari objek gambar yang dipotong adalah sama antara jumlah pemotongan 2, 4, 6 atau 8 buah. Dari data yang didapatkan akan dilakukan analisis pengaruh ukuran gambar dan jumlah kompresor terhadap waktu yang dibutuhkan untuk pemotongan gambar. Pengujian pemotongan gambar dilakukan pada subsistem distributor.

b. Pengujian Waktu Penggabungan Gambar

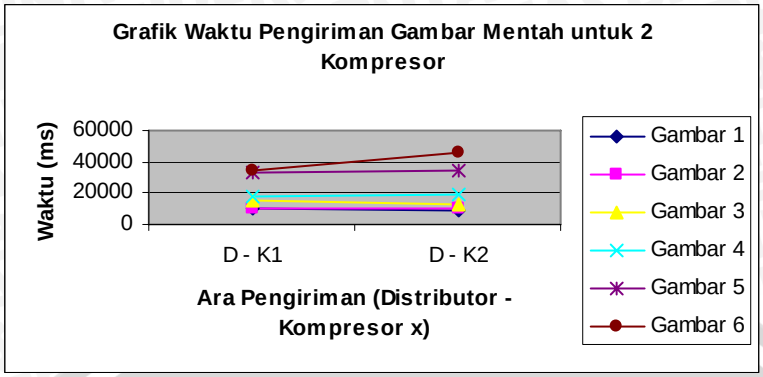
Serupa dengan pengujian terhadap pemotongan gambar, pada subsistem distributor juga dilakukan pengujian kinerja proses penggabungan gambar. Asumsi yang

digunakan dalam pengujian ini adalah bahwa setiap terjadi penaikan ukuran gambar yang akan digabungkan, maka akan berpengaruh pada semakin lamanya waktu penggabungan. Sedangkan untuk jumlah gambar yang digabungkan baik 2, 4, 6 atau 8 akan membutuhkan waktu penggabungan yang hampir sama, karena ukuran total dari piksel gambar juga akan sama. Waktu yang diperlukan untuk menggabungkan setiap gambar obyek uji pada sistem ditunjukkan pada Tabel 6.18.

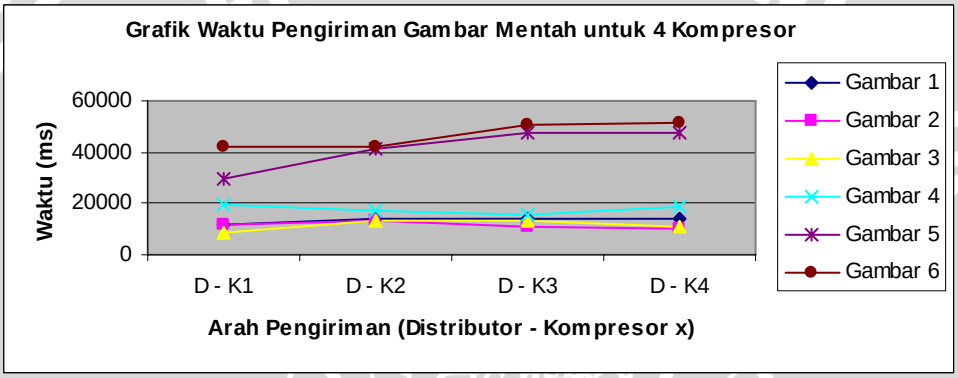
c. Pengujian Waktu Pengiriman Gambar Mentah dari Distributor ke Kompresor

Proses pengiriman gambar mentah (setelah melalui proses pemotongan) dilakukan oleh distributor dengan tujuan pengiriman adalah beberapa komputer yang difungsikan sebagai kompresor. Gambar yang dikirimkan merupakan gambar hasil pemotongan dalam format PGM yang diperkecil dalam paket ZIP. Diasumsikan semakin kecil ukuran file yang dikirimkan, maka akan semakin singkat waktu yang dibutuhkan dalam proses pengiriman. Sehingga untuk sebuah gambar mentah yang sama ukuran filenya (sebelum dipotong), pengiriman hasil pemotongan dari gambar tersebut akan membutuhkan waktu yang semakin sedikit jika jumlah hasil pemotongan semakin besar. Hal ini karena dengan semakin besarnya jumlah pemotongan, maka akan semakin kecil ukuran file dari masing-masing hasil pemotongan.

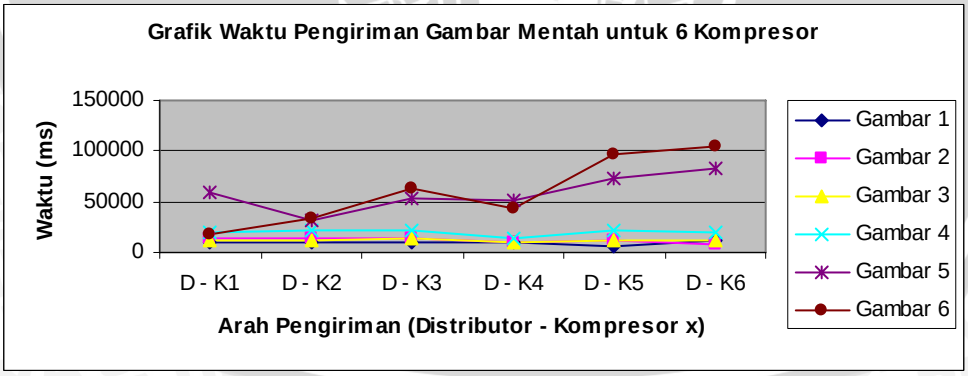
Pada setiap sistem dengan jumlah kompresor yang bervariasi dilakukan pengujian untuk mengetahui waktu proses pengiriman gambar mentah ke masing-masing kompresor dalam sistem. Waktu yang dibutuhkan oleh distributor untuk mengirimkan masing-masing gambar hasil pemotongan ke masing-masing kompresor pada sistem kompresi terdistribusi dengan 2, 4, 6 dan 8 buah ditunjukkan pada Tabel 6.19, 6.20, 6.21 dan Tabel 6.22. Dari tabel tersebut jika digambarkan dalam grafik akan tampak seperti dalam Gambar 6.19, 6.20, 6.21, 6.22.



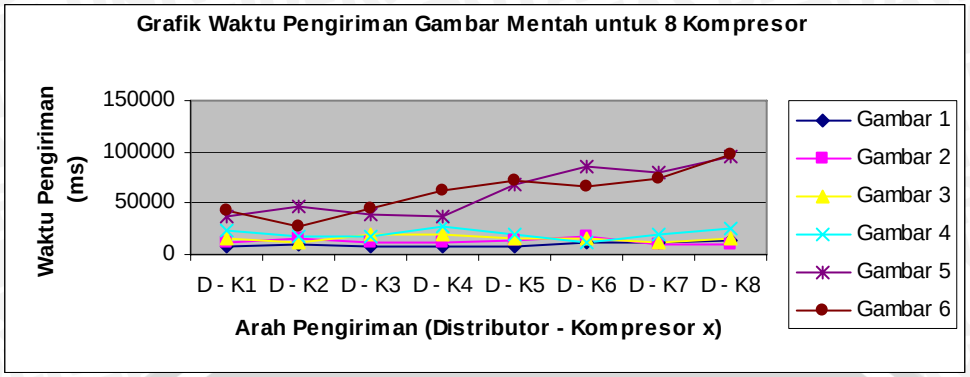
Gambar 6.19. Grafik waktu pengiriman gambar mentah kepada 2 kompresor
Sumber: Pengujian



Gambar 6.20. Grafik waktu pengiriman gambar mentah kepada 4 kompresor
Sumber: Pengujian



Gambar 6.21. Grafik waktu pengiriman gambar mentah kepada 6 kompresor
Sumber: Pengujian



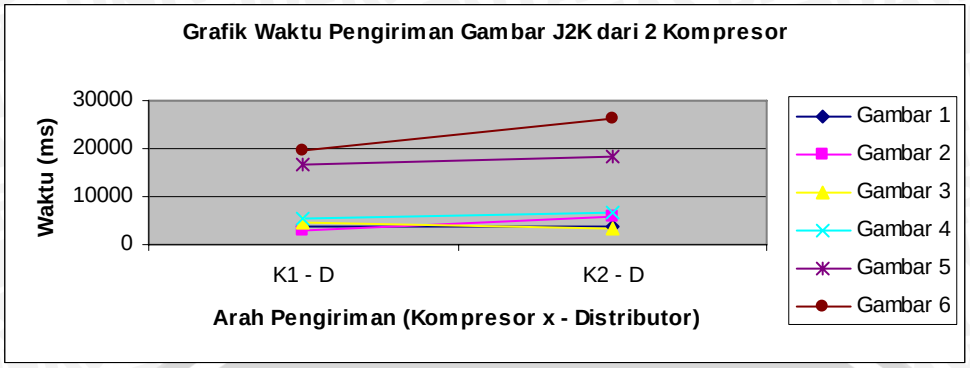
Gambar 6.22. Grafik waktu pengiriman gambar mentah kepada 8 kompresor
Sumber: Pengujian

Jika diambil rata-rata dari waktu yang dibutuhkan distributor untuk mengirimkan setiap gambar mentah kepada kompresor untuk masing-masing sistem dengan 2, 4, 6 dan 8 buah kompresor, didapatkan data seperti pada Tabel 6.23.

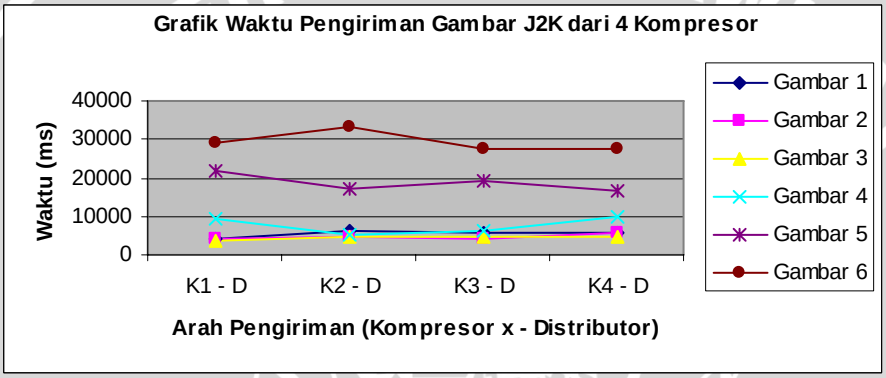
d. Pengujian Waktu Pengiriman Gambar J2K dari Kompresor ke Distributor

Pengiriman gambar hasil kompresi JPEG2000 dilakukan oleh masing-masing kompresor kepada distributor. Subsistem distributor yang selanjutnya akan menggabungkan gambar-gambar tersebut menjadi satu kesatuan kembali. Pengujian waktu pengiriman gambar hasil kompresi ke distributor digunakan untuk mendapatkan data waktu yang dibutuhkan oleh masing-masing kompresor untuk mengirimkan gambar J2K yang telah dihasilkan kepada distributor. Diasumsikan bahwa waktu yang diperlukan untuk mengirimkan gambar J2K dari kompresor ke distributor akan lebih singkat dibandingkan waktu pengiriman gambar mentah dari distributor ke kompresor, karena ukuran file J2K yang lebih kecil dari pada ukuran file gambar mentah. Sedangkan penambahan elemen kompresor akan berpengaruh pada penurunan waktu kirim gambar J2K, karena dengan semakin banyaknya jumlah kompresor berarti sebuah gambr dibagi dalam ukuran yang semakin kecil.

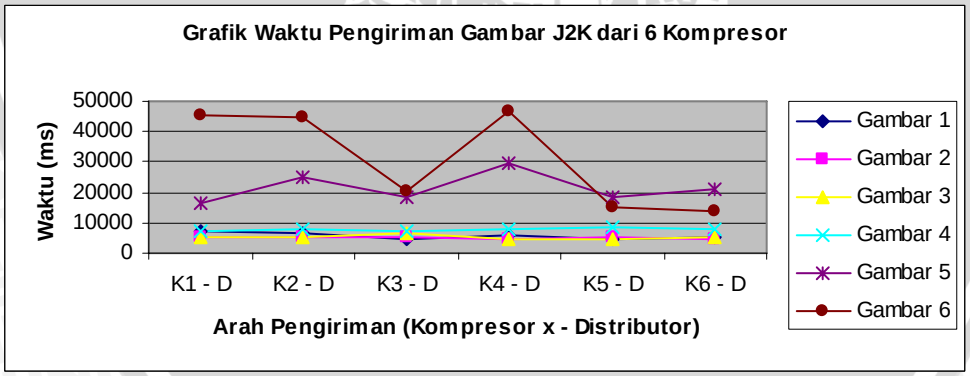
Waktu pengiriman gambar J2K dari masing-masing kompresor ke distributor pada sistem dengan 2, 4, 6 dan 8 buah komputer sebagai kompresor ditunjukkan pada Tabel 6.24, 6.25, 6.26 dan Tabel 6.27. Dari tabel tersebut jika digambar kan dalam grafik akan tampak seperti dalam Gambar 6.23, 6.24, 6.25, 6.26.



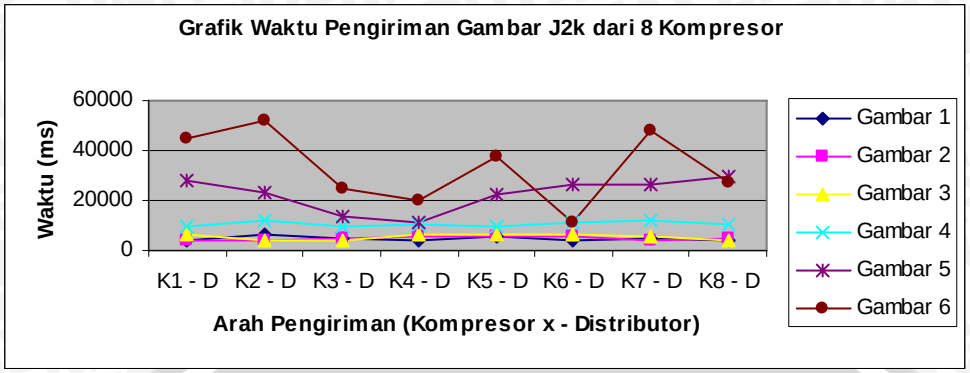
Gambar 6.23. Grafik waktu pengiriman gambar J2K dari 2 kompresor
Sumber: Pengujian



Gambar 6.24. Grafik waktu pengiriman gambar J2K dari 4 kompresor
Sumber: Pengujian



Gambar 6.25. Grafik waktu pengiriman gambar J2K dari 6 kompresor
Sumber: Pengujian



Gambar 6.26. Grafik waktu pengiriman gambar J2K dari 8 kompresor
Sumber: Pengujian

Jika diambil rata-rata waktu yang dibutuhkan kompresor untuk mengirimkan setiap gambar mentah kepada distributor untuk masing-masing sistem dengan 2, 4, 6 dan 8 buah kompresor, didapatkan data seperti pada Tabel 6.28.

e. Pengujian Kualitas Gambar Hasil Kompresi

Untuk mendapatkan data sebagai bahan analisis pengaruh sistem terdistribusi terhadap kualitas gambar hasil kompresi (untuk mengetahui apakah hasil kompresi dengan sistem terdistribusi bisa memenuhi standar analisis kedokteran), dilakukan pengujian dengan menentukan nilai bit rate dengan 4 dan nilai level dekomposisi adalah 5 (dibuat konstanta). Besarnya nilai PSNR dari gambar hasil kompresi belum bisa diasumsikan, sehingga harus dilakukan pengujian terlebih dahulu. Tetapi nilai PSNR antar gambar hasil kompresi dengan 2, 4, 6, atau 8 diperkirakan akan relatif sama, karena meskipun dari sebuah gambar dibagi ke dalam beberapa bagian dan masing-masing bagian dikompres oleh sebuah kompresor yang *standalone*, tetapi proses yang dikenakan pada setiap kompresor yang satu dengna yang lain adalah sama. Sehingga ketika gambar hasil kompresi tersebut pada akhirnya digabungkan kembali juga akan sama, baik untuk sistem dengan 2, 4, 6 atau 8 kompresor. Data hasil pengujian ditunjukkan pada Tabel 6.29

Tabel 6.29 Data nilai PSNR dari gambar hasil kompresi (dalam dB)

Data Gambar	Jumlah Kompresor			
	2 Kompresor	4 Kompresor	6 Kompresor	8 Kompresor
Gambar 1	51,52	51,52	51,52	51,52

Gambar 2	49,88	49,88	49,88	49,88
Gambar 3	50,49	50,49	50,49	50,49
Gambar 4	48,58	48,58	48,58	48,58
Gambar 5	48,66	48,66	48,66	48,66
Gambar 6	48,98	48,98	48,98	48,98

Sumber: Pengujian

f. Pengujian Waktu Total Proses Kompresi

Pengujian waktu total proses kompresi dilakukan untuk mengetahui perbandingan waktu proses kompresi secara terdistribusi dengan waktu yang dibutuhkan untuk proses kompresi secara *standalone.* Diperkirakan dengan diaplikaskannya sistem kompresi terdistribusi ini akan mampu untuk mempersingkat waktu dari proses kompresi terhadap suatu objek gambar jika dibandingkan dengan kompresi *standalone* yang dilakukan pada sebuah mesin saja. Sedangkan dengan semakin banyaknya komputer yang difungsikan sebagai kompresor, maka diasumsikan proses kompresi membutuhkan waktu yang semakin kecil. Dari hasil pengujian didapatkan data waktu yang dibutuhkan oleh masing-masing sistem untuk melakukan proses kompresi terhadap masing-masing objek gambar, seperti ditunjukkan pada Tabel 6.30.

Tabel 6.17. Data waktu pemotongan gambar hasil pengujian (dalam milisecond)

Data Gambar	2 Kompresor				4 Kompresor				6 Kompresor				8 Kompresor			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	781	2390	1234	1468,333333	1453	1722	1029	1401,333333	1734	1484	1078	1432	656	1542	2035	1411
Gambar 2	2282	2625	2171	2359,333333	2288	2337	2140	2255	2203	2574	2125	2300,666667	2625	2300	2342	2422,333333
Gambar 3	2235	4610	4187	3677,333333	2574	4835	4172	3860,333333	2546	4547	4140	3744,333333	2641	3946	4486	3691
Gambar 4	11031	10391	9406	10276	12250	10765	9359	10791,33333	12015	10375	9375	10588,33333	11938	10555	11232	11241,66667
Gambar 5	20813	28156	28891	25953,33333	23703	27203	23938	24948	19109	24469	31422	25000	18422	23235	35444	25700,33333
Gambar 6	33359	37031	48266	39552	34578	34985	51234	40265,66667	39617	40407	43782	41268,66667	34125	40123	41120	38456

Sumber: Pengujian

Tabel 6.18. Data waktu proses penggabungan gambar (dalam milisecond)

Data Gambar	2 Kompresor				4 Kompresor				6 Kompresor				8 Kompresor			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	781	4472	2969	2740,666667	1453	3009	3844	2768,666667	734	4641	3297	2890,666667	656	4032	3323	2670,333333
Gambar 2	2282	5453	5375	4370	2688	5562	5465	4571,666667	2203	5641	5704	4516	2625	4355	6354	4444,666667
Gambar 3	2235	10875	10765	7958,333333	2574	10890	10781	8081,666667	2546	10984	10875	8135	2641	9888	13034	8521
Gambar 4	11031	42922	45000	32984,33333	26250	29109	43032	32797	12015	44250	43875	33380	11938	43564	50003	35168,33333
Gambar 5	20813	109860	105657	78776,66667	19703	80938	130078	76906,33333	19109	107094	108032	78078,33333	18422	90876	108453	72583,66667
Gambar 6	33359	162078	177797	124411,3333	34578	165078	161000	120218,6667	29617	170609	162375	120867	34125	205666	145640	128477

Sumber: Pengujian

Tabel 6.19 Data waktu pengiriman gambar mentah ke 2 buah kompresor (dalam milisecond)

Data Gambar	D - K1				D - K2				Rata-Rata Waktu Pengiriman
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
Gambar 1	14047	7062	11344	10817,66667	8359	9953	9078	9130	9973,8
Gambar 2	10125	10343	11110	10526	10953	12266	8937	10718,66667	10622
Gambar 3	15235	14984	14328	14849	9297	14156	13984	12479	13664
Gambar 4	15953	21297	18203	18484,33333	18093	20391	20546	19676,66667	19081
Gambar 5	56859	20906	22234	33333	51250	22094	28546	33963,33333	33648
Gambar 6	41438	37297	24875	34536,66667	48125	42422	46703	45750	40143

Sumber: Pengujian

Tabel 6.20 Data waktu pengiriman gambar mentah ke 4 buah kompresor (dalam milisecond)

Data Gambar	D - K1				D - K2				D - K3				D - K4				Rata-Rata Waktu Pengiriman
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
Gambar 1	8391	19032	7406	11609,66667	14031	13734	13922	13895,66667	14265	13875	14000	14046,66667	14141	13562	13938	13880,33333	13358,0833
Gambar 2	11141	12359	11875	11791,66667	15531	12172	10937	12880	9594	11937	10078	10536,33333	10906	9438	10110	10151,33333	11339,8333
Gambar 3	12844	7203	5719	8588,666667	12750	14203	12343	13098,66667	13187	14578	11984	13249,66667	6469	14078	11500	10682,33333	11404,8333
Gambar 4	18922	19969	19500	19463,66667	11687	20250	18937	16958	19015	20078	6875	15322,66667	18984	18500	19485	18989,66667	17683,5
Gambar 5	18201	33188	36453	29280,66667	47125	35843	41484	41484	58172	37906	46272	47450	58156	38766	45109	47343,66667	41389,5833
Gambar 6	34672	32844	58719	42078,33333	45390	27359	54062	42270,33333	43813	41828	66032	50557,66667	44297	41750	68031	51359,33333	46566,4167

Sumber: Pengujian

Tabel 6.21 Data waktu pengiriman gambar mentah ke 6 buah kompresor (dalam milisecond)

Data Gambar	D - K1				D - K2				D - K3				D - K4			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	7016	10766	11047	9609,666667	8244	11953	11062	10419,66667	6437	9641	11125	9067,666667	8656	10954	12500	10703,33333
Gambar 2	16062	12563	12468	13697,66667	15343	12172	12453	13322,66667	15468	12250	11235	12984,33333	5094	12203	11156	9484,333333
Gambar 3	14844	10468	9032	11448	9235	14015	13141	12130,33333	15266	14438	12516	14073,33333	5047	10313	12953	9437,666667
Gambar 4	20703	19203	20625	20177	21796	18188	22406	20796,66667	20843	21109	22312	21421,33333	9625	19938	10828	13463,66667
Gambar 5	28750	114546	33515	58937	41359	21234	31781	31458	40062	89875	27734	52557	16891	94844	44343	52026
Gambar 6	3100	23953	24656	17236,33333	41812	31781	28171	33921,33333	16890	28079	145016	63328,33333	18188	85937	24390	42838,33333

D - K5				D - K6				Rata-Rata Waktu Pengiriman
Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
8234	6407	4750	6463,666667	13156	11938	11562	12218,667	9747,111111
14796	9516	12125	12145,66667	14579	4078	4282	7646,3333	11546,83333
14953	14141	8344	12479,33333	9031	13687	11547	11421,667	11831,72222
21688	20688	22813	21729,66667	20422	19406	21906	20578	19694,38889
43078	141000	32203	72093,66667	47812	1E+05	73765	82619,333	58281,83333
46719	67718	173265	95900,66667	57844	83203	2E+05	105114,67	59723,27778

Sumber: Pengujian

Tabel 6.22 Data waktu pengiriman gambar mentah ke 8 buah kompresor (dalam milisecond)

Data Gambar	D - K1				D - K2				D - K3				D - K4				D - K5			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	8891	12012	3502	8135	9250	12054	9241	10181,66667	5813	8546	7001	7120	9469	10012	3002	7494,333333	9110	5145	8100	7451,666667
Gambar 2	15687	10235	12031	12651	15594	20152	10276	15340,66667	10375	12564	9872	10937	11234	11235	12091	11520	15672	12452	14114	14079,33333
Gambar 3	15547	20124	12546	16072,33333	10812	12035	9546	10797,66667	18656	21025	19888	19856,33333	15469	30124	12846	19479,66667	15782	12452	19542	15925,33333
Gambar 4	13484	9456	45662	22867,33333	13078	20215	21468	18253,66667	11329	21658	21548	18178,33333	18375	19456	45662	27831	18703	19954	22001	20219,33333
Gambar 5	41562	26540	45731	37944,33333	42110	42659	54693	46487,33333	41016	35689	39887	38864	39140	26540	45731	37137	43015	92122	71546	68894,33333
Gambar 6	29156	45680	56124	43653,33333	45734	25486	12548	27922,66667	44016	50124	39458	44532,66667	43765	45680	95124	61523	46672	114877	55125	72224,66667

D - K6				D – K7				D - K8				Rata-Rata Waktu Pengiriman
Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
14453	12548	9425	12142	13234	13054	9841	12043	14375	15420	12002	13932,33333	9812,5
18437	12892	21544	17624,333	9203	12152	10276	10543,66667	8312	12548	8021	9627	12790,375
15187	21558	9854	15533	15312	12035	9546	12297,66667	15812	13254	18562	15876	15729,75
2000	19584	15481	12355	19219	20215	21468	20300,66667	18188	12022	45842	25350,66667	20669,5
45860	95642	1E+05	85208,333	43656	109659	84693	79336	24906	161023	99145	95024,66667	61112
41422	1E+05	58124	66519	49203	95486	79548	74745,66667	49016	62220	180000	97078,66667	61024,95833

Sumber: Pengujian

Tabel 6.23 Data waktu rata-rata transfer gambar mentah dari distributor ke kompresor (dalam milisecond)

Data Gambar	Jumlah Kompresor			
	2 Kompresor	4 Kompresor	6 Kompresor	8 Kompresor
Gambar 1	9973,833333	13358,08333	9747,111111	9812,5
Gambar 2	10622,33333	11339,83333	11546,83333	12790,375
Gambar 3	13664	11404,83333	11831,72222	15729,75
Gambar 4	19080,5	17683,5	19694,38889	20669,5
Gambar 5	33648,16667	41389,58333	58281,83333	61112
Gambar 6	40143,33333	46566,41667	59723,27778	61024,95833

Sumber: Pengujian

Tabel 6.24 Data waktu pengiriman gambar J2K dari 2 buah kompresor (dalam milisecond)

Data Gambar	K1 – D				K2 – D				Rata-Rata Waktu Pengiriman
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
Gambar 1	3594	4407	3562	3854,333333	1563	6391	3719	3891	3872,7
Gambar 2	1813	2074	4875	2920,666667	6484	5719	4878	5693,666667	4307,2
Gambar 3	6922	4657	1570	4383	1266	5360	2953	3193	3788
Gambar 4	6937	5375	4000	5437,333333	6031	7594	6469	6698	6067,7
Gambar 5	18078	13781	18750	16869,66667	19797	17859	17219	18291,66667	17581
Gambar 6	16937	16625	24719	19427	20422	17328	41500	26416,66667	22922

Sumber: Pengujian

Tabel 6.25 Data waktu pengiriman gambar J2K dari 4 buah kompresor (dalam milisecond)

Data Gambar	K1 – D				K2 – D				K3 – D				K4 - D				Rata-Rata Waktu Pengiriman
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
Gambar 1	6016	3875	3078	4323	6407	7265	4953	6208,333333	5578	6985	5312	5958,333333	4047	7485	5234	5588,666667	5520
Gambar 2	4703	3985	4406	4364,666667	3515	4844	5922	4760,333333	3688	4125	5026	4279,666667	4844	6672	5313	5609,666667	4754
Gambar 3	4078	3562	3891	3843,666667	3735	5344	4172	4417	4828	5078	4344	4750	3172	4891	6594	4885,666667	4474
Gambar 4	5390	11672	11328	9463,333333	4125	6234	5563	5307,333333	4812	7281	7156	6416,333333	6593	10453	12125	9723,666667	7728
Gambar 5	21391	11984	32563	21979,33333	10781	11234	29734	17249,66667	17031	13406	27766	19401	11890	13047	25609	16848,66667	18870
Gambar 6	37344	11766	38031	29047	27391	24485	47485	33120,33333	29812	18422	34485	27573	29812	19329	33359	27500	29310

Sumber: Pengujian

Tabel 6.26 Data waktu pengiriman gambar J2K dari 6 buah kompresor (dalam milisecond)

Data Gambar	K1 - D				K2 - D				K3 - D				K4 - D				K5 - D			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	8545	7219	5875	7213	8562	6250	5625	6812,333333	4391	3625	6078	4698	6000	5781	5250	5677	5829	3797	3625	4417
Gambar 2	5328	4531	5156	5005	5171	5906	4672	5249,666667	5453	5860	4860	5391	5094	4687	4610	4797	5656	4016	6312	5328
Gambar 3	5031	4485	5407	4974,333333	4641	5250	5921	5270,666667	6297	6609	6078	6328	5047	3000	5094	4380,333333	4422	6063	3718	4734,333333
Gambar 4	5328	8672	8562	7520,666667	11922	3797	8291	8003,333333	9219	8031	4969	7406,333333	9625	6844	7828	8099	11938	7359	6891	8729,333333
Gambar 5	13078	10625	24812	16171,66667	11278	41688	21454	24806,66667	11390	17500	25438	18109,33333	16891	17093	55281	29755	15953	11156	27688	18265,66667
Gambar 6	11094	22484	103484	45687,33333	14047	12750	107516	44771	19938	14266	27063	20422,33333	18188	15891	105875	46651,33333	13141	12562	19828	15177

K6 - D				Rata-Rata Waktu Pengiriman
Data 1	Data 2	Data 3	Rata-Rata	
3969	4656	6500	5041,6667	5643,166667
7078	3438	3625	4713,6667	5080,722222
4297	5187	5656	5046,6667	5122,388889
11156	5688	6672	7838,6667	7932,888889
11953	37469	13219	20880,333	21331,44444
12078	12700	16250	13676	31064,16667

Sumber: Pengujian

Tabel 6.27 Data waktu pengiriman gambar J2K dari 8 buah kompresor (dalam milisecond)

Data Gambar	K1 - D				K2 - D				K3 - D				K4 - D				K5 - D			
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata
Gambar 1	4750	3954	2999	3901	6422	5987	6420	6276,333333	4969	4550	4957	4825,333333	3843	3900	3541	3761,333333	5671	5487	5846	5668
Gambar 2	3375	3564	4578	3839	4438	4379	3981	4266	4640	4990	4641	4757	5781	5249	5214	5414,666667	5313	5158	5468	5313
Gambar 3	6719	5554	7895	6722,666667	3688	3945	4012	3881,666667	4328	3900	4521	4249,666667	6344	6823	6816	6661	6906	6417	7000	6774,333333
Gambar 4	9516	8945	9456	9305,666667	12016	12091	12999	12368,66667	9671	9191	10024	9628,666667	11969	10284	9984	10745,66667	9735	9754	9845	9778
Gambar 5	18625	17984	48564	28391	11578	45943	12458	23326,33333	10516	20675	9985	13725,33333	11188	11895	11204	11429	16421	15447	35484	22450,66667
Gambar 6	11672	10885	111542	44699,66667	24062	25001	106014	51692,33333	24187	25124	24540	24617	24188	15265	20154	19869	21516	20548	69984	37349,33333

K6 - D				K7 – D				K8 - D				Rata-Rata Waktu Pengiriman
Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
3656	3857	3965	3826	5203	5346	4521	5023,333333	5312	3575	2365	3750,666667	4629
5078	5235	5621	5311,3333	4671	4652	3015	4112,666667	4609	6214	4658	5160,333333	4771,75
6437	5948	5987	6124	5234	5684	5612	5510	4640	3974	4221	4278,333333	5525,208333
11016	11200	10987	11067,667	12391	12467	11987	12281,66667	11906	10256	9897	10686,33333	10732,79167
13547	53021	13254	26607,333	16281	46495	16510	26428,66667	16750	55942	16142	29611,33333	22746,20833
12250	10258	11256	11254,667	21047	20988	1E+05	47674	21203	34458	26458	27373	33066,125

Sumber: Pengujian

Tabel 6.28 Data waktu rata-rata transfer gambar J2K dari kompresor ke disributor (dalam milisecond)

Data Gambar	Jumlah Kompresor			
	2 Kompresor	4 Kompresor	6 Kompresor	8 Kompresor
Gambar 1	3872,666667	5519,583333	5643,166667	4629
Gambar 2	4307,166667	4753,583333	5080,722222	4771,75
Gambar 3	3788	4474,083333	5122,388889	5525,208333
Gambar 4	6067,666667	7727,666667	7932,888889	10732,79167
Gambar 5	17580,66667	18869,66667	21331,44444	22746,20833
Gambar 6	22921,83333	29310,08333	31064,16667	33066,125

Sumber: Pengujian

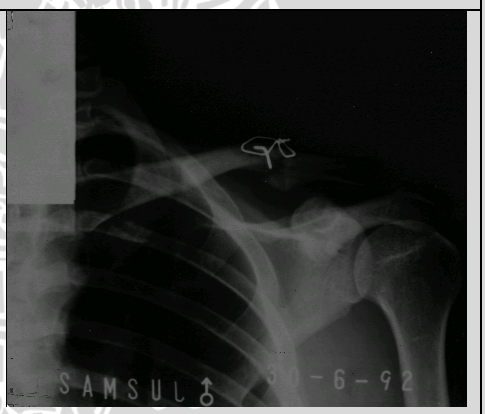
Tabel 6.30 Data waktu total proses kompresi (dalam milisecond)

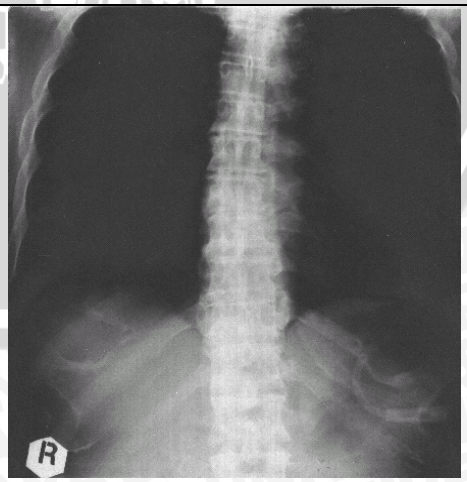
Data Gambar	2 Kompresor				4 Kompresor				6 Kompresor				8 Kompresor				Standalone
	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	Data 1	Data 2	Data 3	Rata-Rata	
Gambar 1	29375	28948	30023	29448,66667	28625	35922	25094	29880,33333	23297	27172	25234	25234,33333	24859	19543	21458	21953,33333	4625
Gambar 2	34672	28703	27625	30333,33333	29344	28000	25859	27734,33333	26640	27094	27516	27083,33333	30391	23658	24152	26067	6094
Gambar 3	34313	46313	43453	41359,66667	28797	35469	33469	32578,33333	31828	35937	33438	33734,33333	33047	36524	38412	35994,33333	9609
Gambar 4	74453	88281	87063	83265,66667	66797	81797	78781	75791,66667	73406	77016	78203	76208,33333	41329	61256	51236	51273,66667	29829
Gambar 5	2E+05	188078	308203	245166,3333	194328	155266	212406	187333,3333	174250	159437	164531	166072,6667	176344	186254	124952	162516,6667	4E+05
Gambar 6	3E+05	271250	301422	291015,6667	237922	229703	300391	256005,3333	200234	262094	252219	238182,3333	227141	245987	221456	231528	5E+05

Sumber: Pengujian

Untuk perbandingan ukuran file dari gambar mentah *(raw image)* dalam format BMP dengan file gambar hasil kompresi terdistribusi dalam format J2K ditunjukkan pada Tabel 6.31.

Tabel 6.31 Data waktu total proses kompresi (dalam milisecond)

No	Nama File Gambar Mentah	Ukuran File Gambar Mentah	Nama File Gambar J2K	Ukuran File J2K	Tampilan Gambar J2K
1	lutut.bmp	3,6 MB	lutut.j2k	495 KB	
2	bahu.bmp	5,19 MB	bahu.j2k	858 KB	

3	lengan.bmp	10,7 MB	lengan.j2k	1,86 MB	
4	tangan.bmp	22,7 MB	tangan.j2k	8,9 MB	
5	dada.bmp	57,4 MB	dada.j2k	17,96 MB	

6	p_paha.bmp	70 MB	p_paha.j2k	29,7 MB	

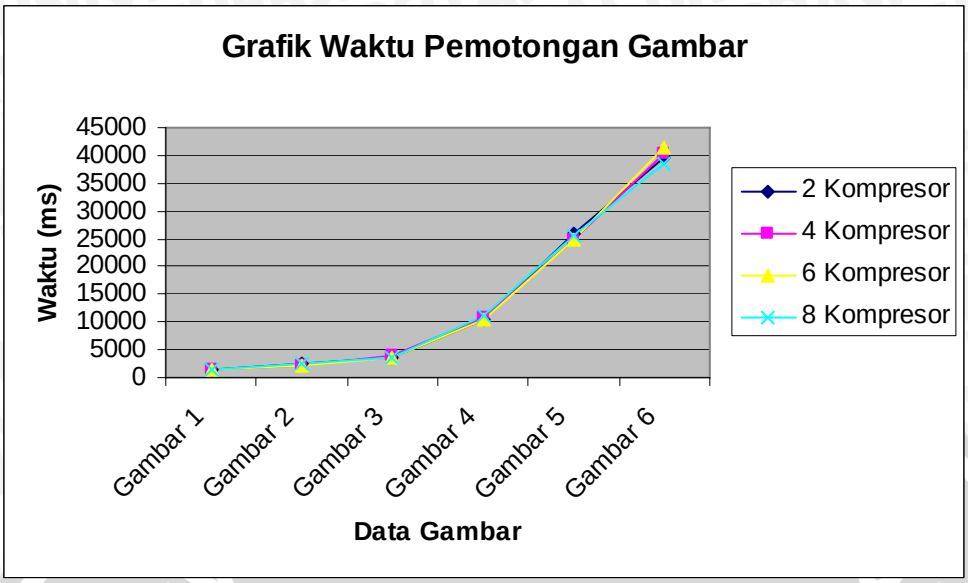
Sumber: Pengujian

6.2. Analisis

Analisis dilakukan kepada setiap data yang didapatkan dari proses pengujian. Analisis pada unjuk kerja sistem ditujukan untuk mengetahui unjuk kerja sistem kompresi terdistribusi (berkenaan dengan waktu total proses serta kualitas gambar hasil kompresi) dan faktor-faktor yang berpengaruh terhadapnya.

6.2.1. Analisis Waktu Pemotongan Gambar

Data hasil pengujian yang dicantumkan pada Tabel 6.13 merupakan data waktu yang dibutuhkan oleh subsistem distributor untuk melakukan pemotongan gambar mentah, termasuk di dalamnya adalah proses konversi dari file BMP ke file PGM dan mengepaknya dalam format ZIP. Dari data yang dicantumkan pada Tabel 6.13 jika ditampilkan dalam grafik seperti tampak dalam Gambar 6.27.



Gambar 6.27. Grafik waktu pemotongan gambar
Sumber: Pengujian

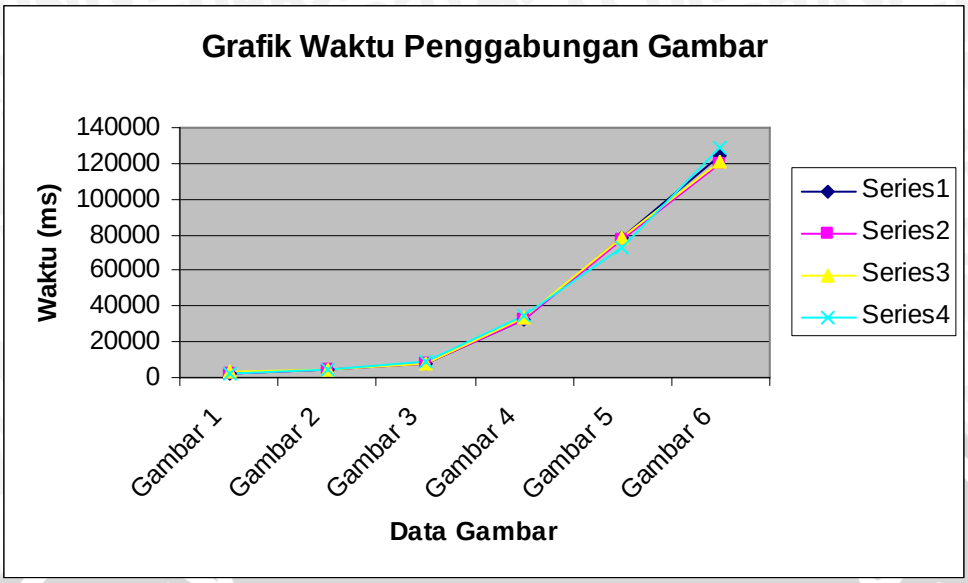
Dari grafik dalam Gambar 6.27 dapat dilihat sebuah pola peningkatan waktu yang dibutuhkan untuk melakukan proses pemotongan gambar seiring dengan semakin bertambah besarnya ukuran gambar yang dipotong. Hal ini disebabkan karena ukuran file dari suatu gambar akan sebanding dengan jumlah piksel yang menyusun gambar tersebut. Dari persamaan:

```
ukuran file gambar = (jumlah piksel x kedalaman warna) / (8 x 1024) KB
```

dapat dilihat bahwa ukuran file suatu gambar sebanding dengan jumlah piksel di dalamnya. Sedangkan perbedaan jumlah kompresor yang berarti juga perbedaan jumlah potongan tidak memberikan pengaruh yang signifikan untuk perbedaan waktu proses pemotongan. Waktu untuk memotong suatu gambar dengan jumlah potongan 2, 4, 6 atau 8 relatif sama, meskipun terdapat perbedaan hanya beberapa millisecond saja.

6.2.2. Analisis Waktu Penggabungan Gambar

Data waktu penggabungan gambar yang dilakukan oleh subsistem distributor (Tabel 6.14) jika digambarkan pada suatu diagram akan nampak seperti dalam Gambar 6.28.



Gambar 6.28. Grafik waktu penggabungan gambar
Sumber: Pengujian

Serupa dengan analisis terhadap waktu yang dibutuhkan untuk proses pemotongan gambar, dengan semakin bertambahnya ukuran gambar yang akan digabungkan, bertambah pula jumlah piksel penyusun gambar tersebut, maka akan semakin besar waktu yang harus dikeluarkan untuk menggabungkan gambar. Sedangkan perbedaan jumlah elemen gambar yang akan digabungkan (sesuai dengan jumlah kompresor yang digunakan) kurang berpengaruh pada selisih waktu proses penggabungan tersebut. Waktu yang diperlukan untuk menggabungkan 2, 4, 6 atau 8 buah gambar relatif sama untuk setiap ukuran gambar yang dijadikan obyek uji. Faktor alokasi sumberdaya komputer untuk melakukan proses ini juga akan sangat berpengaruh.

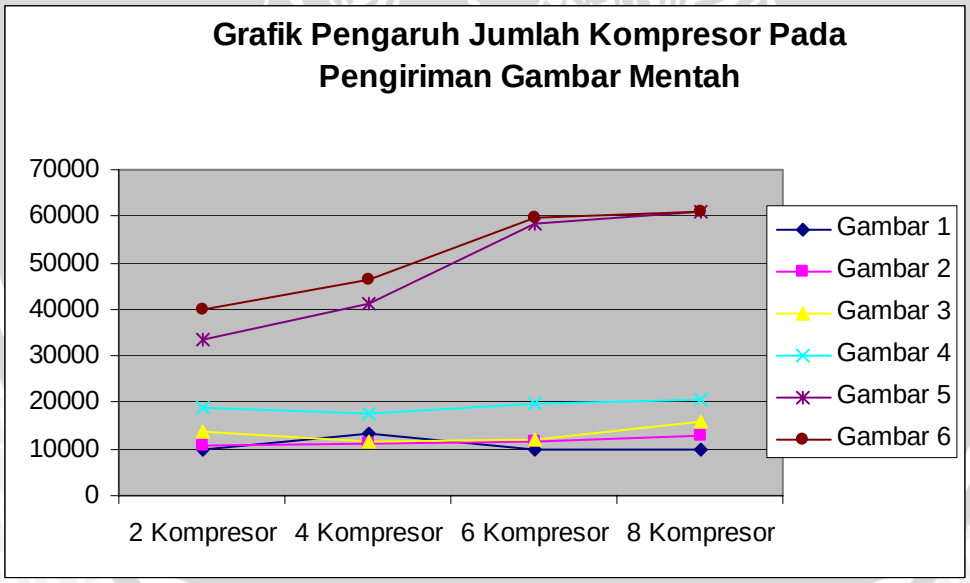
6.2.3. Analisis Waktu Pengiriman Gambar Mentah dari Distributor ke Kompresor

Gambar mentah yang dikirimkan oleh distributor kepada kompresor berada dalam format ZIP dengan tujuan untuk menpersingkat waktu kirim. Dari grafik yang terdapat dalam Gambar 6.20, 6.21, 6.22, 6.23, terlihat bahwa unjuk kerja proses pengiriman gambar dari distributor menuju ke beberapa kompresor tidak membentuk suatu pola tertentu dan cenderung kurang optimal. Meskipun ukuran hasil pemotongan gambar berukuran relatif sama (dengan perbedaan ukuran file yang sangat kecil) tetapi waktu yang dibutuhkan untuk mengirimkan hasil pemotongan tersebut ke masing-masing

kompresor tidak merata. Ada bagian dari hasil potongan yang dikirimkan ke kompresor dengan waktu yang lebih singkat dari yang lain, meskipun ukurannya relatif hampir sama. Selain itu, ada kalanya waktu yang dibutuhkan untuk mengirimkan gambar dengan ukuran file yang lebih besar ke suatu kompresor, lebih singkat daripada waktu yang diutuhkan untuk mengirimkan gambar dengan ukuran yang lebih kecil.

Kurang optimalnya kinerja proses pengiriman gambar mentah ini disebabkan oleh faktor komponen *switching* yang digunakan pada jaringan dimana sistem diimplementasikan adalah hub. Model *switching* yang dilakukan oleh hub adalah dengan meneruskan paket data yang dikomunikasikan tidak hanya dengan alamat yang dituju, tetapi deteruskan *(broadcast)* ke semua *host* yang terhubung dengan hub tersebut. Permasalahan ini bisa diperbaiki dengan menggunakan *switch* sebagai komponen *switching* menggantikan hub.

Dari Tabel 6.19 yang memuat data waktu rata-rata proses transfer gambar mentah dari distributor ke kompresor jika digambarkan dalam grafik akan nampak seperti Gambar 6.29.



Gambar 6.29. Grafik waktu rata-rata pengiriiman gambar mentah
Sumber: Pengujian

Dari pola grafik yang terdapat dalam Gambar 6.29 bisa disimpulkan bahwa semakin banyak kompresor yang digunakan, maka waktu yang dibutuhkan untuk mengirimkan gambar mentah dari distributor menuju ke kompresor menjadi semakin

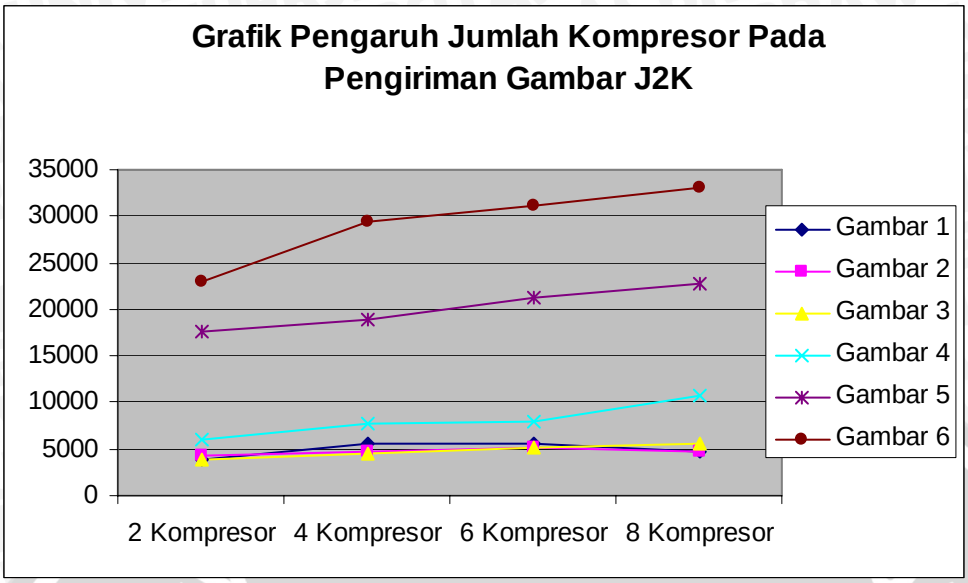
besar. Meskipun dengan semakin banyak kompresor yang digunakan berdampak pada semakin kecilnya ukuran gambar mentah yang dikirimkan oleh distributor ke kompresor, tetapi perfomansi jaringan komputer (dengan menggunakan komponen *switching* berupa hub) yang kurang optimal (kurang bagus) dalam pengiriman data berakibat pada semakin besarnya waktu yang untuk mengirimkan gambar mentah ke kompresor.

6.2.4. Analisis Waktu Pengiriman Gambar J2K dari Kompresor ke Distributor

Sebagaimana proses pengiriman gambar mentah yang dilakukan dari distributor menuju ke kompresor, proses pengiriman gambar hasil kompresi (dalam format J2K) dari masing-masing kompresor menuju ke distributor mempunyai performansi yang kurang optimal. Dari Gambar 6.23, 6.24, 6.25, 6.26 nampak bahwa tidak terdapat suatu pola yang baku. Waktu yang dibutuhkan untuk proses pengiriman antara suatu kompresor dengan distributor dan kompresor yang lain dengan distributor yang sama relatif tidak merata. Ini menunjukkan bahwa unjuk kerja jaringan komputer di mana sistem diimplementasikan kurang optimal. Komponen hub yang digunakan untuk saling mengkoneksikan komputer-konputer dalam sistem menyebabkan ketidakoptimalan proses transfer data antara kompresor dengan distributor.

Sedangkan jika dibandingkan dengan waktu pengiriman gambar mentah, maka waktu yang dibutuhkan untuk mengirimkan gambar J2K (hasil kompresi) dari kompresor ke distributor akan lebih singkat. Hal ini disebabkan karena gambar J2K merupakan gambar hasil kompresi dengan ukuran file yang lebih kecil.

Dari Tabel 6.24 yang memuat data waktu rata-rata proses transfer gambar J2K dari kompresor ke distributor jika digambarkan dalam grafik akan nampak seperti Gambar 6.30.



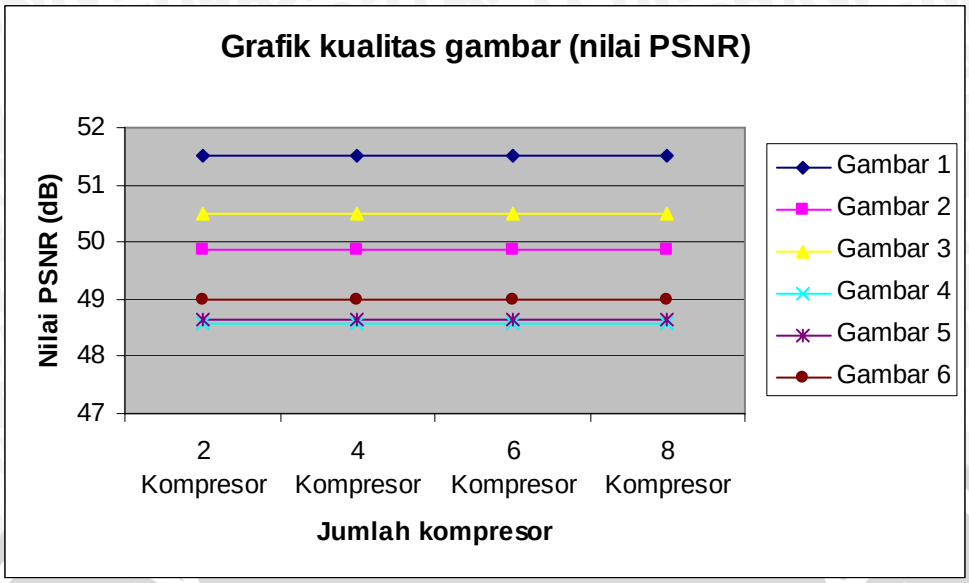
Gambar 6.30. Grafik waktu rata-rata pengiriiman gambar hasil kompresi (J2K) dari kompresor
Sumber: Pengujian

Jumlah kompresor yang digunakan juga berpengaruh pada lamanya waktu yang digunakan untuk proses pengiriman gambar J2K dari kompresor menuju ke distributor. Dari grafik yang terdapat dalam Gambar 6.30 terlihat bahwa semakin besar jumlah kompresor yang digunakan maka semakin besar jumlah waktu yang dibutuhkan untuk mengirimkan gambar J2K dari kompresor-kompresor menuju ke distributor. Serupa dengan proses pengiriman gambar mentah dari distributor menuju ke kompresor, performansi jaringan komputer yang kurang optimal menjadi penyebab semakin banyaknya waktu transfer yang dibutuhkan.

Faktor pengiriman data gambar, baik dari distributor menuju kompresor ataupun sebaliknya, merupakan faktor yang sangat mempengaruhi kinerja sistem kompresi terdistribusi. Semakin optimal proses pengiriman gambar, maka akan semakin optimal pula kinerja sistem yang ditandai dengan peningkatan kecepatan proses total kompresi.

6.2.5. Analisis Kualitas Gambar Hasil Kompresi Terdistribusi

Gambar 6.31 merupakan grafik nilai PSNR (parameter kualitas gambar) dari hasil proses kompresi.

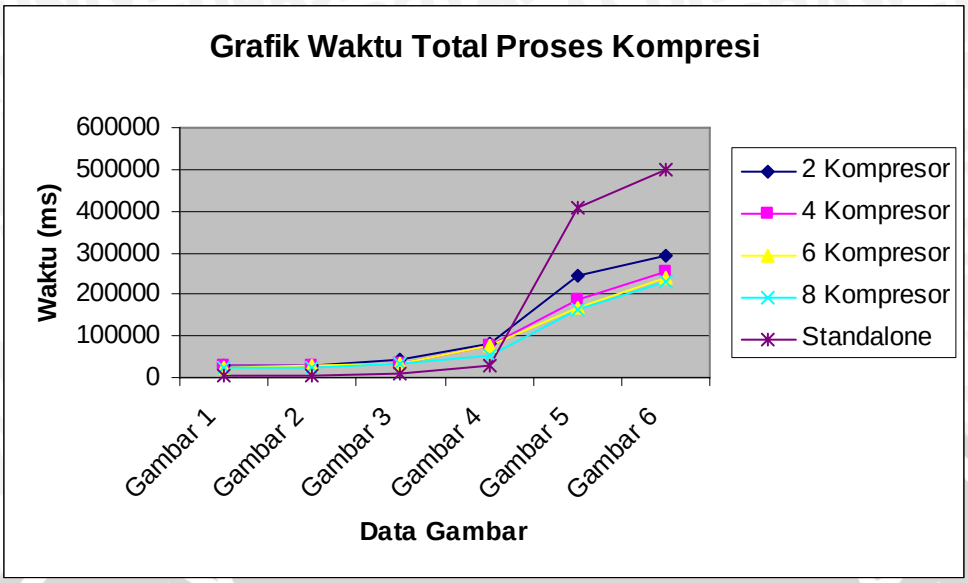


Gambar 6.31. Grafik nilai PSNR
Sumber: Pengujian

Dari Gambar 6.31 terlihat bahwa kualitas gambar hasil kompresi baik yang dilakukan oleh sistem dengan 2, 4, 6 atau 8 buah kompresor adalah sama. Hal ini ditunjukkan dengan nilai PSNR yang sama dari masing-masing gambar untuk setiap jumlah kompresor. Dari hasil percobaan didapatkan nilai PSNR untuk setiap gambar objek rata-rata di atas 48 dB. Sehingga bisa disimpulkan bahwa sistem kompresi terdistribusi mampu menghasilkan gambar hasil kompresi dengan kualitas yang memenuhi syarat untuk dilakukan analisis kedokteran tanpa harus membutuhkan penilaian dokter ahli terhadap gambar tersebut.

6.2.6. Analisis Waktu Total Proses Kompresi Terdistribusi

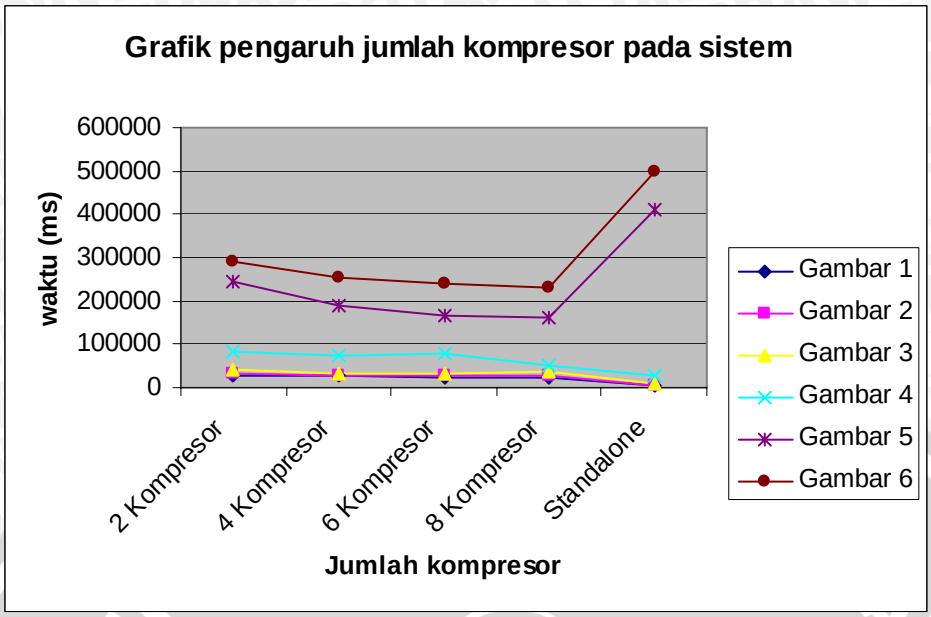
Dari Tabel 6.26 yang berisi tentang data waktu total yang diperlukan untuk melakukan rangkaian proses kompresi, dibuat grafik yang ditunjukkan dalam Gambar 6.32.



Gambar 6.32. Grafik waktu total proses kompresi
Sumber: Pengujian

Dari Gambar 6.32 bisa diketahui bahwa sistem kompresi terdisribusi akan optimal digunakan pada gambar dengan ukuran file yang besar. Untuk gambar 1, 2, 3, dan gambar 4, sistem kompresi terdistribusi kurang optimal digunakan, karena waktu yang dibutuhkan untuk melakukan rangkaian proses kompresi dengan sistem terdistribusi, lebih lama daripada kompresi yang dilakukan secara *standalone*. Sedangkan untuk gambar 5 dan 6, waktu untuk melakukan kompresi terdistribusi jauh lebih cepat daripada waktu kompresi secara *standalone*. Hal ini disebabkan karena pada gambar yang kecil, jika harus dilakukan kompresi secara terdistribusi akan memerlukan waktu yang lebih banyak untuk proses transfer gambar, pemotongan dan penggabungannya daripada proses kompresinya sendiri. Sedangkan untuk gambar dengan ukuran yang besar, meskipun diperlukan waktu untuk pengiriman, pemotongan dan penggabungan gambar, tetapi dengan melakukan kompresi pada beberapa mesin kompresor, waktu untuk proses kompresi akan jauh lebih kecil daripada waktu kompresi gambar yang besar secara *standalone*. Oleh karena itu, waktu total proses kompresi oleh sistem terdistribusi pada gambar dengan ukuran yang besar lebih singkat dari waktu kompresi *standalone*.

Pengaruh jumlah kompresor terhadap waktu total yang dibutuhkan untuk melakukan suatu rangkaian proses kompresi pada sistem kompresi terdistribusi terlihat dalam Gambar 6.33.



Gambar 6.33. Grafik waktu total proses kompresi terhadap jumlah kompresor
Sumber: Pengujian

Dengan semakin banyaknya jumlah kompresor yang digunakan, maka waktu total yang diperlukan untuk suatu proses kompresi menjadi semakin menurun. Meskipun semakin banyak jumlah kompresor yang digunakan akan membawa dampak pada peningkatan waktu yang dibutuhkan untuk transfer gambar antara distributor dan kompresor, tetapi proses yang diperlukan oleh masing-masing kompresor untuk mengkompres gambar yang telah diterima dari disitributor menjadi semakin singkat. Hal ini disebabkan karena dengan semakin banyak jumlah kompresor yang digunakan akan memperkecil ukuran file gambar yang harus dikompres oleh masing-masing kompresor.