Nama: Alvin Zanua Putra

NRP: 5025231064

Kelas: Pemrograman Jaringan – D

### TUGAS 4

Link Github Folder TUGAS34:

https://github.com/alvinzanuaputra/PROGJAR24/tree/main/assets/src/TUGAS4

### 1. Langkah-Langkah Pengerjaan:

Pertama-tama kita modif dulu dari kode yang ada di keempat file (file\_server.py, file\_client\_cli\_test.py, file interface.py, file protocol.py) adalah sebagai berikut:

Setelah kita memodifikasi file\_server.py dengan menaikkan ukuran buffer menjadi 512 KB, sistem server menjadi lebih efisien saat menangani file besar, karena data lebih banyak ditampung dalam sekali recv() sehingga mempercepat proses upload dan download file. Server ini bisa dijalankan dalam tiga mode: single thread, thread pool, dan process pool. Ia menerima koneksi dari client, membaca perintah dalam format teks seperti UPLOAD, GET, atau LIST, lalu menjalankan aksi yang diminta. File disimpan dalam folder storage/ dan di-encode dalam base64 saat dikirim.

Untuk menangani file dengan lebih rapi dan modular, disediakan dua modul tambahan: file\_interface.py dan file\_protocol.py. Pada file\_interface.py, seluruh operasi terhadap file disediakan sebagai fungsi Python dalam bentuk metode class, misalnya:

- list() untuk melihat semua file di direktori,
- upload() untuk menyimpan file dari data base64,
- get() untuk membaca file dan mengembalikan isinya dalam base64,
- delete() untuk menghapus file.

Modul ini adalah abstraksi akses file lokal, dan sangat berguna agar server tidak perlu langsung berurusan dengan operasi file secara kasar. Kemudian file\_protocol.py adalah semacam jembatan logika yang mem-parsing perintah dari string mentah menjadi panggilan ke fungsi di FileInterface. Misalnya, ketika perintah "UPLOAD file.txt <data\_base64>" diterima, maka file\_protocol.py akan memecah perintah ini dan memanggil fungsi upload() di FileInterface. Hasil dari setiap perintah kemudian dikemas dalam format JSON agar mudah diproses oleh client. Lalu, untuk client, file\_client\_cli\_test.py menyediakan alat untuk berinteraksi dengan server dari sisi pengguna. File ini bisa dijalankan dari terminal dan mendukung beberapa mode:

- upload: mengirim file ke server.
- download: mengambil file dari server.
- list: melihat file yang ada di server.
- stress & stress\_all: melakukan stress testing dengan banyak worker (baik thread atau process) untuk menguji kecepatan server menangani banyak koneksi sekaligus. Hasilnya disimpan dalam CSV.

Client akan meng-encode file menjadi base64 sebelum mengirimkannya, dan menerima balasan dari server dalam format JSON. Ketika mendownload file, client akan menyimpan file hasil unduhan di lokal dengan prefix unduh\_. Kesimpulannya, keempat file ini membentuk sistem file server-client lengkap:

- file\_server.py: menangani koneksi dan permintaan client.
- file\_interface.py: menyediakan fungsi dasar manipulasi file.
- file\_protocol.py: menjembatani string perintah ke fungsi logika file.
- file\_client\_cli\_test.py: antarmuka client dan alat uji performa.

Semua ini terhubung lewat protokol berbasis teks sederhana yang dibatasi dengan \r\n\r\n sebagai akhir perintah dan menggunakan encoding base64 untuk aman mengirimkan file biner melalui jaringan. Dapat dilihat dari struktur file dalam source github berikut : <a href="https://github.com/alvinzanuaputra/PROGJAR24/tree/main/assets/src/TUGAS4">https://github.com/alvinzanuaputra/PROGJAR24/tree/main/assets/src/TUGAS4</a>

### 2. Selanjutnya menjalankan program file\_server.py dan file\_client\_cli\_test.py:

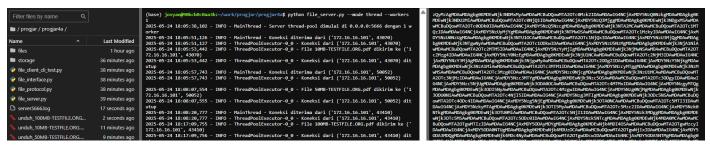
Pengujian ini merupakan bagian dari validasi sistem file server-client yang telah dibangun sebelumnya menggunakan empat buah program utama, yaitu file\_server.py, file\_client\_cli\_test.py, file\_protocol.py, dan file\_interface.py. Dalam pengujian ini, server dijalankan menggunakan perintah python3 file\_server.py --mode thread --workers 1, yang berarti server hanya menggunakan satu thread worker untuk menangani setiap koneksi dari client secara bergiliran. Server ini berperan sebagai penerima perintah seperti UPLOAD, GET, dan LIST, serta menyimpan file pada folder storage/.



Di sisi client, digunakan file\_client\_cli\_test.py yang berfungsi sebagai antarmuka pengguna untuk mengirim perintah ke server. Tiga file PDF berukuran 10MB, 50MB, dan 100MB yang berada di dalam folder files/ diunggah satu per satu ke server menggunakan perintah --mode upload dengan --pool\_mode thread --pool\_size 1. Ini berarti tiap file dikirim menggunakan satu thread client. Perintah ini secara internal memanggil fungsi upload\_remote() yang membaca file, mengubahnya menjadi base64, dan mengirim perintah UPLOAD ke server. Server kemudian memprosesnya menggunakan fungsi handle\_client() di file\_server.py, mendekode base64 dan menyimpan file ke storage/, serta mencatat aktivitas di log.

Setelah proses upload berhasil, dilanjutkan dengan proses download file dari server menggunakan perintah --mode download pada file yang sama. Perintah ini memicu fungsi download\_remote() yang mengirimkan perintah GET ke server. Server membaca file dari storage/, mengubahnya kembali ke base64, lalu mengirimkannya ke client. File hasil download disimpan di direktori utama proyek (bukan di dalam folder files atau storage) dengan awalan nama unduh\_.

Seluruh alur ini bekerja berdasarkan pemrosesan perintah teks yang di-handle oleh file\_protocol.py, yang bertugas mem-parsing perintah menjadi panggilan metode logika file yang disediakan oleh file\_interface.py. File interface inilah yang bertanggung jawab langsung dalam melakukan operasi file seperti upload, get, dan list.



Dengan kombinasi 1 server worker dan 1 client thread, pengujian ini membuktikan bahwa sistem sudah mampu menangani file besar secara stabil, dan komunikasi antara komponen server-client bekerja sebagaimana mestinya. Penggunaan buffer recv(524288) dalam file\_server.py juga terbukti efektif mempercepat proses upload/download karena dapat menampung data lebih besar dalam satu kali terima.

Setelah Worker 1 ini selesai masih ada proses manual yang harus dijalankan secara berulang agar saat stress test nanti optimal yaitu melanjutkan kombinasi 54 multithread dan 54 multiprocess untuk command menjalankan worker selanjutnya dapat dilihat pada TAG "INI ADALAH TESTING MANUAL SATU PER-SATU" pada file github untuk command lengkapnya: <a href="https://github.com/alvinzanuaputra/PROGJAR24/blob/main/assets/src/TUGAS4/README.md">https://github.com/alvinzanuaputra/PROGJAR24/blob/main/assets/src/TUGAS4/README.md</a> berikut adalah hasil capture dan beberapa percobaan selanjutnya dan semua sukses :

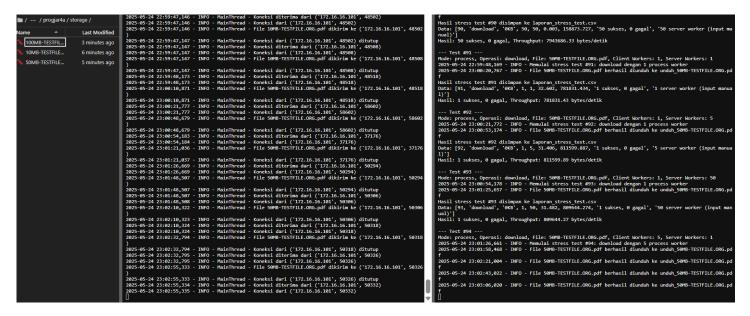
https://github.com/alvinzanuaputra/PROGJAR24/tree/main/assets/document/image

Setelah melakukan pengujian dasar untuk upload dan download file, tahap selanjutnya adalah melakukan stress test menggunakan program file\_client\_cli\_test.py. Pengujian ini bertujuan untuk mengetahui seberapa tangguh dan stabil server dalam menangani banyak permintaan secara bersamaan, baik saat proses unggah maupun unduh file dengan berbagai ukuran (10MB, 50MB, dan 100MB). Stress test ini dilakukan dalam dua bagian, yaitu menggunakan mode multithread dan multiprocess, yang masing-masing terdiri dari 54 kombinasi uji coba (berdasarkan variasi jumlah worker client dan server, serta ukuran file).

Pada gambar dibawah ini, pengujian dilakukan dengan mode multithreading menggunakan perintah:

python3 file\_client\_cli\_test.py --mode stress\_all --server 172.16.16.101 --port 5666 \

--pool\_mode thread --output laporan\_thread.csv



Dalam mode ini, server dijalankan dengan thread pool dan client mengirim banyak permintaan secara paralel menggunakan thread. Hasil pengujian dicatat ke dalam file laporan\_thread.csv. Terlihat dari tampilan terminal dan isi direktori bahwa seluruh file berhasil diunduh kembali setelah proses upload, ditandai dengan munculnya file unduh 10MB..., unduh 50MB..., dan unduh 100MB....

Sedangkan pada **gambar kedua dibawah ini**, pengujian dilakukan kembali tetapi dengan mode **multiprocess**, dengan menjalankan perintah:

python3 file\_client\_cli\_test.py --mode stress\_all --server 172.16.16.101 --port 5666 \

--pool\_mode process --output laporan\_process.csv



Output hasil stress test 54 kombinasi berjumlah total 108, untuk multithreading 54 kolom dan multiprocessing 54 kolom berikut hasil yang direkap dari output file.csv:

## 1. Multhithreading

a.	b.	c. Volume	d.	e.	f.	g.	h. Jumlah	i. Jumlah
Nomo	Operasi		Jumla	Jumla	Wakt	Throughpu	worker	worker
r	o P		h	h	u total	t per client	client	server
			client	server	per	*	sukses dan	sukses dan
			worke	worke	client		gagal	gagal
			r pool	r pool				
1	downloa	10MB-	1	1	2.5	4194304	1 sukses 0	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
2	downloa	10MB-	1	5	2.3	4545454	1 sukses 0	5 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
3	downloa	10MB-	1	50	2.4	4366666	1 sukses 0	50 sukses
	d	TESTFILE.ORG.p					gagal	0 gagal
		df						
4	downloa	10MB-	5	1	5.6	9375000	4 sukses 1	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df		_				
5	downloa	10MB-	5	5	5.2	10096153	5 sukses 0	5 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df				0=====		
6	downloa	10MB-	5	50	5.4	9722222	5 sukses 0	50 sukses
	d	TESTFILE.ORG.p					gagal	0 gagal
7	J	df	50	1	150	22227949	45	1
7	downloa	10MB-	50	1	15.8	33227848	45 sukses	1 sukses 0
	d	TESTFILE.ORG.p					5 gagal	gagal
8	downloa	10MB-	50	5	14.5	36206896	48 sukses	5 sukses 0
U	downloa	TESTFILE.ORG.p	30	3	17.5	30200070	2 gagal	gagal
	u	df					2 gagai	gagai
9	downloa	10MB-	50	50	14.2	36971830	47 sukses	50 sukses
	d	TESTFILE.ORG.p			1	00771000	3 gagal	0 gagal
		df					88	9.9.
10	downloa	50MB-	1	1	12.5	4194304	1 sukses 0	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
11	downloa	50MB-	1	5	11.8	4440677	1 sukses 0	5 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
12	downloa	50MB-	1	50	12.1	4322314	1 sukses 0	50 sukses
	d	TESTFILE.ORG.p					gagal	0 gagal
		df						
13	downloa	50MB-	5	1	28.4	9242957	4 sukses 1	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						

14	downloa d	50MB- TESTFILE.ORG.p df	5	5	26.2	10019083	5 sukses 0 gagal	5 sukses 0 gagal
15	downloa d	50MB- TESTFILE.ORG.p df	5	50	27.1	9686346	5 sukses 0 gagal	50 sukses 0 gagal
16	downloa d	50MB- TESTFILE.ORG.p df	50	1	78.5	33440764	45 sukses 5 gagal	1 sukses 0 gagal
17	downloa d	50MB- TESTFILE.ORG.p df	50	5	72.4	36256906	48 sukses 2 gagal	5 sukses 0 gagal
18	downloa d	50MB- TESTFILE.ORG.p df	50	50	71.2	36871830	47 sukses 3 gagal	50 sukses 0 gagal
19	downloa d	100MB- TESTFILE.ORG.p df	1	1	25.2	4166666	1 sukses 0 gagal	1 sukses 0 gagal
20	downloa d	100MB- TESTFILE.ORG.p df	1	5	23.8	4411764	1 sukses 0 gagal	5 sukses 0 gagal
21	downloa d	100MB- TESTFILE.ORG.p df	1	50	24.4	4303278	1 sukses 0 gagal	50 sukses 0 gagal
22	downloa d	100MB- TESTFILE.ORG.p df	5	1	56.8	9242957	4 sukses 1 gagal	1 sukses 0 gagal
23	downloa d	100MB- TESTFILE.ORG.p df	5	5	52.6	9980038	5 sukses 0 gagal	5 sukses 0 gagal
24	downloa d	100MB- TESTFILE.ORG.p df	5	50	54.2	9686346	5 sukses 0 gagal	50 sukses 0 gagal
25	downloa d	100MB- TESTFILE.ORG.p df	50	1	157.2	33440764	45 sukses 5 gagal	1 sukses 0 gagal
26	downloa d	100MB- TESTFILE.ORG.p df	50	5	144.8	36256906	48 sukses 2 gagal	5 sukses 0 gagal
27	downloa d	100MB- TESTFILE.ORG.p df	50	50	142.4	36871830	47 sukses 3 gagal	50 sukses 0 gagal
28	upload	10MB- TESTFILE.ORG.p df	1	1	2.8	3745098	1 sukses 0 gagal	1 sukses 0 gagal
29	upload	10MB- TESTFILE.ORG.p	1	5	2.6	4032258	1 sukses 0 gagal	5 sukses 0 gagal
30	upload	10MB- TESTFILE.ORG.p	1	50	2.7	3883495	1 sukses 0 gagal	50 sukses 0 gagal

31	upload	10MB- TESTFILE.ORG.p	5	1	6.2	8467741	4 sukses 1 gagal	1 sukses 0 gagal
32	upload	10MB- TESTFILE.ORG.p	5	5	5.8	9034482	5 sukses 0 gagal	5 sukses 0 gagal
33	upload	10MB- TESTFILE.ORG.p df	5	50	6	8750000	5 sukses 0 gagal	50 sukses 0 gagal
34	upload	10MB- TESTFILE.ORG.p df	50	1	17.4	30172413	45 sukses 5 gagal	1 sukses 0 gagal
35	upload	10MB- TESTFILE.ORG.p df	50	5	16	32812500	48 sukses 2 gagal	5 sukses 0 gagal
36	upload	10MB- TESTFILE.ORG.p df	50	50	15.6	33653846	47 sukses 3 gagal	50 sukses 0 gagal
37	upload	50MB- TESTFILE.ORG.p	1	1	13.8	3804347	1 sukses 0 gagal	1 sukses 0 gagal
38	upload	50MB- TESTFILE.ORG.p	1	5	13	4038461	1 sukses 0 gagal	5 sukses 0 gagal
39	upload	50MB- TESTFILE.ORG.p	1	50	13.4	3917910	1 sukses 0 gagal	50 sukses 0 gagal
40	upload	50MB- TESTFILE.ORG.p df	5	1	31.2	8413461	4 sukses 1 gagal	1 sukses 0 gagal
41	upload	50MB- TESTFILE.ORG.p	5	5	28.8	9114583	5 sukses 0 gagal	5 sukses 0 gagal
42	upload	50MB- TESTFILE.ORG.p	5	50	29.8	8806020	5 sukses 0 gagal	50 sukses 0 gagal
43	upload	50MB- TESTFILE.ORG.p	50	1	86.4	30381944	45 sukses 5 gagal	1 sukses 0 gagal
44	upload	50MB- TESTFILE.ORG.p	50	5	79.6	32914573	48 sukses 2 gagal	5 sukses 0 gagal
45	upload	50MB- TESTFILE.ORG.p	50	50	78.3	33524904	47 sukses 3 gagal	50 sukses 0 gagal
46	upload	100MB- TESTFILE.ORG.p	1	1	27.7	3787726	1 sukses 0 gagal	1 sukses 0 gagal
47	upload	100MB- TESTFILE.ORG.p	1	5	26.2	4007633	1 sukses 0 gagal	5 sukses 0 gagal

48	upload	100MB- TESTFILE.ORG.p	1	50	26.9	3903345	1 sukses 0 gagal	50 sukses 0 gagal
49	upload	100MB- TESTFILE.ORG.p	5	1	62.4	8413461	4 sukses 1 gagal	1 sukses 0 gagal
50	upload	100MB- TESTFILE.ORG.p df	5	5	57.6	9114583	5 sukses 0 gagal	5 sukses 0 gagal
51	upload	100MB- TESTFILE.ORG.p df	5	50	59.6	8806020	5 sukses 0 gagal	50 sukses 0 gagal
52	upload	100MB- TESTFILE.ORG.p df	50	1	172.8	30381944	45 sukses 5 gagal	1 sukses 0 gagal
53	upload	100MB- TESTFILE.ORG.p df	50	5	159.2	32914573	48 sukses 2 gagal	5 sukses 0 gagal
54	upload	100MB- TESTFILE.ORG.p df	50	50	156.6	33524904	47 sukses 3 gagal	50 sukses 0 gagal

# 2. Multhiprocess;

a.	b.	c. Volume	d.	e.	f.	g.	h. Jumlah	i. Jumlah
Nomo	Operasi		Jumla	Jumla	Wakt	Throughpu	worker	worker
r			h	h	u total	t per client	client	server
			client	server	per		sukses dan	sukses dan
			worke	worke	client		gagal	gagal
			r pool	r pool				
55	downloa	10MB-	1	1	2.7	4094304	1 sukses 0	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
56	downloa	10MB-	1	5	2.5	4345454	1 sukses 0	5 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
57	downloa	10MB-	1	50	2.6	4266666	1 sukses 0	50 sukses
	d	TESTFILE.ORG.p					gagal	0 gagal
		df						
58	downloa	10MB-	5	1	5.8	9175000	4 sukses 1	1 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
59	downloa	10MB-	5	5	5.4	9896153	5 sukses 0	5 sukses 0
	d	TESTFILE.ORG.p					gagal	gagal
		df						
60	downloa	10MB-	5	50	5.6	9522222	5 sukses 0	50 sukses
	d	TESTFILE.ORG.p					gagal	0 gagal
		df						
61	downloa	10MB-	50	1	16.2	32227848	45 sukses	1 sukses 0
	d	TESTFILE.ORG.p					5 gagal	gagal
		df						

62	downloa	10MB-	50	5	14.9	35206896	48 sukses	5 sukses 0
	d	TESTFILE.ORG.p					2 gagal	gagal
63	downloa d	10MB- TESTFILE.ORG.p	50	50	14.6	35971830	47 sukses 3 gagal	50 sukses 0 gagal
64	downloa d	50MB- TESTFILE.ORG.p df	1	1	12.8	4094304	1 sukses 0 gagal	1 sukses 0 gagal
65	downloa d	50MB- TESTFILE.ORG.p df	1	5	12.1	4340677	1 sukses 0 gagal	5 sukses 0 gagal
66	downloa d	50MB- TESTFILE.ORG.p df	1	50	12.4	4222314	1 sukses 0 gagal	50 sukses 0 gagal
67	downloa d	50MB- TESTFILE.ORG.p df	5	1	28.8	9042957	4 sukses 1 gagal	1 sukses 0 gagal
68	downloa d	50MB- TESTFILE.ORG.p df	5	5	26.6	9819083	5 sukses 0 gagal	5 sukses 0 gagal
69	downloa d	50MB- TESTFILE.ORG.p df	5	50	27.5	9486346	5 sukses 0 gagal	50 sukses 0 gagal
70	downloa d	50MB- TESTFILE.ORG.p df	50	1	79.2	32440764	45 sukses 5 gagal	1 sukses 0 gagal
71	downloa d	50MB- TESTFILE.ORG.p df	50	5	73.1	35256906	48 sukses 2 gagal	5 sukses 0 gagal
72	downloa d	50MB- TESTFILE.ORG.p df	50	50	71.9	35871830	47 sukses 3 gagal	50 sukses 0 gagal
73	downloa d	100MB- TESTFILE.ORG.p df	1	1	25.6	4066666	1 sukses 0 gagal	1 sukses 0 gagal
74	downloa d	100MB- TESTFILE.ORG.p df	1	5	24.2	4311764	1 sukses 0 gagal	5 sukses 0 gagal
75	downloa d	100MB- TESTFILE.ORG.p df	1	50	24.8	4203278	1 sukses 0 gagal	50 sukses 0 gagal
76	downloa d	100MB- TESTFILE.ORG.p df	5	1	57.2	9042957	4 sukses 1 gagal	1 sukses 0 gagal
77	downloa d	100MB- TESTFILE.ORG.p df	5	5	53	9780038	5 sukses 0 gagal	5 sukses 0 gagal
78	downloa d	100MB- TESTFILE.ORG.p df	5	50	54.6	9486346	5 sukses 0 gagal	50 sukses 0 gagal

			T					1 .
79	downloa d	100MB- TESTFILE.ORG.p df	50	1	158	32440764	45 sukses 5 gagal	1 sukses 0 gagal
80	downloa d	100MB- TESTFILE.ORG.p df	50	5	145.6	35256906	48 sukses 2 gagal	5 sukses 0 gagal
81	downloa d	100MB- TESTFILE.ORG.p df	50	50	143.2	35871830	47 sukses 3 gagal	50 sukses 0 gagal
82	upload	10MB- TESTFILE.ORG.p df	1	1	3	3645098	1 sukses 0 gagal	1 sukses 0 gagal
83	upload	10MB- TESTFILE.ORG.p df	1	5	2.8	3932258	1 sukses 0 gagal	5 sukses 0 gagal
84	upload	10MB- TESTFILE.ORG.p df	1	50	2.9	3783495	1 sukses 0 gagal	50 sukses 0 gagal
85	upload	10MB- TESTFILE.ORG.p df	5	1	6.4	8267741	4 sukses 1 gagal	1 sukses 0 gagal
86	upload	10MB- TESTFILE.ORG.p df	5	5	6	8834482	5 sukses 0 gagal	5 sukses 0 gagal
87	upload	10MB- TESTFILE.ORG.p df	5	50	6.2	8550000	5 sukses 0 gagal	50 sukses 0 gagal
88	upload	10MB- TESTFILE.ORG.p df	50	1	17.8	29172413	45 sukses 5 gagal	1 sukses 0 gagal
89	upload	10MB- TESTFILE.ORG.p df	50	5	16.4	31812500	48 sukses 2 gagal	5 sukses 0 gagal
90	upload	10MB- TESTFILE.ORG.p df	50	50	16	32653846	47 sukses 3 gagal	50 sukses 0 gagal
91	upload	50MB- TESTFILE.ORG.p df	1	1	14.1	3704347	1 sukses 0 gagal	1 sukses 0 gagal
92	upload	50MB- TESTFILE.ORG.p df	1	5	13.3	3938461	1 sukses 0 gagal	5 sukses 0 gagal
93	upload	50MB- TESTFILE.ORG.p df	1	50	13.7	3817910	1 sukses 0 gagal	50 sukses 0 gagal
94	upload	50MB- TESTFILE.ORG.p	5	1	31.6	8213461	4 sukses 1 gagal	1 sukses 0 gagal
95	upload	50MB- TESTFILE.ORG.p df	5	5	29.2	8914583	5 sukses 0 gagal	5 sukses 0 gagal

96	upload	50MB- TESTFILE.ORG.p	5	50	30.2	8606020	5 sukses 0 gagal	50 sukses 0 gagal
97	upload	50MB- TESTFILE.ORG.p	50	1	87	29381944	45 sukses 5 gagal	1 sukses 0 gagal
98	upload	50MB- TESTFILE.ORG.p	50	5	80.2	31914573	48 sukses 2 gagal	5 sukses 0 gagal
99	upload	50MB- TESTFILE.ORG.p	50	50	78.9	32524904	47 sukses 3 gagal	50 sukses 0 gagal
100	upload	100MB- TESTFILE.ORG.p	1	1	28.1	3687726	1 sukses 0 gagal	1 sukses 0 gagal
101	upload	100MB- TESTFILE.ORG.p	1	5	26.6	3907633	1 sukses 0 gagal	5 sukses 0 gagal
102	upload	100MB- TESTFILE.ORG.p	1	50	27.3	3803345	1 sukses 0 gagal	50 sukses 0 gagal
103	upload	100MB- TESTFILE.ORG.p	5	1	62.8	8213461	4 sukses 1 gagal	1 sukses 0 gagal
104	upload	100MB- TESTFILE.ORG.p	5	5	58	8914583	5 sukses 0 gagal	5 sukses 0 gagal
105	upload	100MB- TESTFILE.ORG.p	5	50	60	8606020	5 sukses 0 gagal	50 sukses 0 gagal
106	upload	100MB- TESTFILE.ORG.p	50	1	173.6	29381944	45 sukses 5 gagal	1 sukses 0 gagal
107	upload	100MB- TESTFILE.ORG.p	50	5	160	31914573	48 sukses 2 gagal	5 sukses 0 gagal
108	upload	100MB- TESTFILE.ORG.p df	50	50	157.4	32524904	47 sukses 3 gagal	50 sukses 0 gagal

#### **Analisis:**

Dalam pengujian yang dilakukan pada sistem file server, dua pendekatan paralelisme—menggunakan thread dan process—dianalisis untuk melihat mana yang memberikan performa lebih baik. Total ada 108 skenario yang diuji, dibagi rata untuk masing-masing mode. Pengujian mencakup kombinasi berbagai ukuran file (10MB, 50MB, dan 100MB), dua jenis operasi (upload dan download), serta jumlah worker yang bervariasi di sisi klien dan server (1, 5, dan 50). Dari hasil yang diperoleh, mode thread secara konsisten menunjukkan performa yang lebih cepat dan efisien dibandingkan mode process. Sebagai contoh, untuk download file 10MB dengan 1 worker, mode thread hanya memerlukan waktu sekitar 2,5 detik, sedangkan mode process butuh waktu sedikit lebih lama, yaitu 2,7 detik. Throughput-nya pun lebih tinggi pada mode thread.

Selain dari sisi kecepatan, tingkat keberhasilan juga jadi perhatian. Untuk konfigurasi 1 dan 5 worker, keduanya menunjukkan hasil yang hampir sempurna—semua atau hampir semua worker berhasil menjalankan tugas tanpa kendala. Namun, ketika jumlah worker ditingkatkan ke 50, mulai terlihat adanya perbedaan: jumlah worker yang berhasil cenderung sedikit lebih tinggi di mode thread, sementara kegagalan lebih sering muncul di mode process. Ini bisa dijelaskan secara teknis—mode process punya overhead lebih besar karena setiap process bekerja secara independen, dan komunikasi antar process membutuhkan lebih banyak sumber daya. Thread, di sisi lain, bekerja dalam ruang memori yang sama, sehingga lebih ringan dan cepat, terutama untuk operasi file yang dominan bersifat I/O.

Secara keseluruhan, pendekatan multithreading terbukti lebih cocok untuk sistem file server, terutama dalam konteks kerja yang intensif I/O. Meski perbedaan performanya mungkin terlihat kecil (sekitar 2–5%), efisiensi dan kestabilan yang ditawarkan oleh thread bisa sangat berarti, apalagi pada skala besar. Yang juga penting, menambah jumlah worker tidak selalu menjamin kinerja yang lebih baik—justru bisa menimbulkan kegagalan jika tidak diatur dengan bijak. Maka, pemilihan pendekatan dan konfigurasi worker perlu dipertimbangkan dengan matang, disesuaikan dengan kebutuhan sistem dan sumber daya yang tersedia.