ANALISIS KINERJA PENCARIAN FILE DENGAN FIND DAN LOCATE PADA RED HAT 9.0

Najwa Latifah Hasibuan, Najwatul khoiriah, Dedy Kiswanto
Ilmu Komputer, Universitas Negeri Medan
Jalan Willem Iskandar, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang, Indonesia
Najwa.4232250003@mhs.unimed.ac.id

ABSTRAK

Pencarian file merupakan salah satu fungsi utama dalam sistem operasi Linux, termasuk pada Red Hat 9.0. Artikel ini membandingkan kinerja dua perintah pencarian file, yaitu find dan locate, yang memiliki pendekatan berbeda. Perintah find bekerja langsung pada sistem file, sementara locate menggunakan database indeks untuk pencarian yang lebih cepat. Penelitian ini menggunakan metode eksperimen komparatif dengan mengukur waktu eksekusi, penggunaan CPU, memori, serta fleksibilitas pencarian pada berbagai ukuran direktori (100, 1.000, dan 10.000 file). Hasil penelitian menunjukkan bahwa find lebih efisien dalam penggunaan CPU, terutama pada direktori kecil, namun membutuhkan lebih banyak memori dibandingkan locate. Sebaliknya, locate lebih cepat dalam pencarian sederhana tetapi bergantung pada pembaruan database yang memengaruhi akurasi. Kesimpulan dari penelitian ini merekomendasikan penggunaan find untuk pencarian kompleks dan locate untuk pencarian cepat pada data yang stabil.

Kata kunci : *Pencarian file, find, locate, Red Hat 9.0*

1. PENDAHULUAN

Pengguna membutuhkan cara yang lebih efektif untuk mengatur dan mencari data mereka. Selama sepuluh tahun terakhir, jumlah data penyimpanan yang tersedia untuk pengguna individu telah meningkat hampir dua kali lipat, memungkinkan pengguna saat ini untuk menyimpan
data dalam jumlah yang praktis tidak terbatas. Hal ini menggeser tantangan bagi pengguna individu dari memutuskan apa yang harus disimpan menjadi menemukan file tertentu saat dibutuhkan[3].

Red Hat 9.0 adalah distribusi Linux yang sangat populer, terutama di kalangan pengguna profesional dan server. Dirilis pada tahun 2003, Red Hat 9.0 menawarkan stabilitas yang tinggi, dukungan perangkat keras yang luas, dan berbagai aplikasi serta utilitas untuk administrasi sistem. Salah satu fitur utama yang dimiliki Red Hat adalah manajer paket RPM (Red Hat Package Manager), yang memudahkan instalasi dan pengelolaan perangkat lunak. Red Hat 9.0 juga mendukung berbagai perintah pencarian file yang sering digunakan, termasuk find dan locate. Dalam konteks penelitian ini, Red Hat 9.0 digunakan sebagai sistem operasi untuk menguji kinerja perintah-perintah pencarian file tersebut.

Linux adalah sistem operasi gratis yang telah ada selama hampir tiga dekade. Kode sumbernya tersedia bagi para pengembang, amatir, dan masyarakat umum untuk pengembangan dan kustomisasi. Red Hat memodifikasi salinan dari versi pilihan kode sumber Linux dan memperkenalkan fitur-fitur baru, menambahkan perbaikan, dan memperbaiki bug[10].

Pencarian file adalah dua perintah yang sering dilakukan oleh pengguna sistem operasi berbasis liux, Red Hat 9.0 menawarkan dua perintah pencarian fole utama: find dan locate. Kedua perintah ini digunakan untuk tujuan yang sama, yaitu mencari file dalam sistem. Namun terdapat perbedaan mendasar dalam cara kerja dan kinerja pencarian file.

Find adalah perintah yang memindai sistem file secara langsung sehingga memungkinkan kita untuk mencari file berdasarkan berbagai kriteria seperti nama, ukuran, tanggal modifikasi, dan lainnya. keunggulan utama dari find adalah kemampuannya untuk bekerja langsung pada sistem file yang sedang aktif, memberikan hasil pencarian yang akurat dan mutakhir. Namun perintah find cenderung membutuhkan waktu yang lebih lama untuk menelusuri dan memindai direktori yang besar, terutama jika direktori tersebut berisi banyak file dan subdirektori.

locate adalah yang memindai sistem menggunakan database yang sudah dibangun sebelumnya (melalui perintah updatedb) untuk melakukan pencarian file, yang menjadikannya lebih cepat namun terbatas pada file yang ada dalam database tersebut. Database ini menyimpan informasi tentang lokasi dan nama file yang ada pada sistem, memungkinkan locate untuk melakukan pencarian dengan sangat cepat. Namun, terdapat beberapa keterbatasan pada perintah locate yaitu : locate hanya dapat menemukan file yang ada dalam database tersebut, dan jika file baru ditambahkan atau dihapus setelah pembaruan terakhir, hasil pencarian yang diperoleh mungkin tidak akurat.

Sistem operasi linux, Red Hat 9.0, banyak digunakan pada sistem desktop dan server, memiliki karakteristik penggunaan yang memerlukan efisiensi dalam pengelolaan dan pencarian file. Dengan semakin berkembangnya ukuran dan kompleksitas data yang disimpan dalam sistem, penting untuk mengevaluasi kinerja alat pencarian yang tersedia

guna membantu para pengguna dan administrator sistem dalam mengambil keputusan yang tepat terkait dengan pencarian file.

Tujuan dari penelitian ini adalah untuk melakukan analisis kinerja antara kedua alat pencarian yaitu find dan locate dalam konteks penggunaan pada Red Hat 9.0. Penelitian ini akan membandingkan kedua alat berdasarkan beberapa aspek utama, seperti waktu eksekusi, akurasi pencarian, dan fleksibilitas dalam menerapkan berbagai filter atau kriteria pencarian. Selain itu, penelitian ini juga akan menguji dampak dari pembaruan database pada kinerja locate dan membandingkannya dengan hasil pencarian langsung yang dilakukan oleh find. Hasil dari penelitian ini diharapkan dapat memberikan gambaran yang jelas tentang kapan sebaiknya masing-masing perintah digunakan, dan apakah ada kondisi tertentu di mana satu alat lebih diutamakan daripada yang lain.

2. TINJAUAN PUSTAKA

2.1. Linux

Linux adalah sistem operasi gratis yang telah ada selama hampir tiga dekade. Kode sumbernya tersedia bagi para pengembang, amatir, dan masyarakat umum untuk pengembangan dan kustomisasi. Red Hat memodifikasi salinan dari versi pilihan kode sumber Linux dan memperkenalkan fitur-fitur baru, menambahkan perbaikan, dan memperbaiki bug. Perusahaan mengemas versi yang diperbarui sebagai distribusi Linux mereka sendiri untuk tujuan komersial. Distribusi ini telah diuji secara menyeluruh untuk berjalan dengan lancar dan berkinerja baik pada berbagai platform perangkat keras komputer. Sistem ini stabil, kuat, kaya fitur, dan siap menampung beban kerja dalam skala berapa pun[10].

2.2. Red Hat

Red Hat, sebagai salah satu distribusi Linux, menawarkan sistem operasi yang andal untuk lingkungan desktop dan server. Sistem ini mendukung efisiensi dalam pengelolaan data dengan menggunakan alat pencarian file seperti *find* dan *locate*.Model bisnis Red Hat Enterprise Linux yang mengintegrasikan strategi open-source sebagai inti dari sistem operasinya. Sistem ini memungkinkan pengelolaan skala besar dengan efisiensi tinggi, menjadikannya pilihan utama untuk perusahaan besar. Dalam konteks alat pencarian file, seperti *find* dan *locate*, basis data yang diperbarui menggunakan *updatedb* mencerminkan pentingnya pendekatan berbasis indeks untuk mempercepat proses pencarian. Red Hat memanfaatkan efisiensi ini untuk mendukung sistem yang kompleks, di mana kecepatan pencarian file menjadi krusial dalam pengelolaan data skala besar[1].

2.3. Perintah Find

Perintah find adalah salah satu alat pencarian file yang paling sering digunakan dalam sistem Linux. Alat ini bekerja dengan cara memindai seluruh sistem file atau bagian dari sistem file untuk menemukan file yang memenuhi kriteria tertentu[9]. Perintah ini sangat fleksibel, memungkinkan pengguna untuk mencari file berdasarkan berbagai parameter, seperti nama file, ukuran, jenis file, waktu modifikasi, izin file, dan lain sebagainya. Salah satu keunggulan utama dari find adalah akurasinya, karena perintah ini langsung bekerja pada sistem file yang aktif dan menghasilkan data yang selalu mutakhir. Namun, kelemahan find adalah proses pencarian yang memakan waktu lebih lama, terutama jika sistem memiliki banyak file dan direktori yang besar. Proses ini juga dapat mempengaruhi kinerja sistem, karena find akan memindai seluruh struktur file saat pencarian dilakukan[5].

2.4. Perintah Locate

Berbeda dengan find, perintah locate menggunakan pendekatan berbasis database untuk pencarian file. Dalam sistem Linux, locate bekerja dengan menggunakan database yang berisi informasi tentang lokasi file yang ada dalam sistem[7]. Database ini dibangun dan diperbarui secara berkala menggunakan perintah updatedb. Karena locate tidak perlu memindai sistem file secara langsung, alat ini jauh lebih cepat dalam mencari file dibandingkan find. Namun, kecepatan ini datang dengan beberapa keterbatasan. Salah satunya adalah ketergantungan pada pembaruan database, yang berarti locate hanya dapat menemukan file yang tercatat dalam database yang sudah ada. Jika file baru ditambahkan atau dihapus setelah pembaruan database terakhir, hasil pencarian dapat menjadi tidak akurat. Oleh karena itu, pembaruan database secara berkala sangat penting untuk menjaga akurasi pencarian menggunakan locate[6].

2.5. Database dan Pembaruan Database dengan Updatedb

Salah satu komponen penting yang mendukung kinerja perintah locate adalah pembaruan database file menggunakan perintah updatedb. Perintah ini bertugas untuk memperbarui informasi lokasi file dalam database yang digunakan oleh locate. Proses pembaruan ini dilakukan secara berkala dan dapat disesuaikan waktunya, baik itu setiap hari, mingguan, atau sesuai kebutuhan. Pembaruan database ini sangat penting untuk memastikan bahwa locate dapat memberikan hasil pencarian yang akurat. Namun, ada beberapa masalah yang dapat muncul apabila pembaruan database tidak dilakukan secara tepat waktu, seperti ketidakakuratan dalam hasil pencarian atau ketidaksesuaian antara status file yang ada di sistem dengan yang tercatat dalam database. Oleh karena itu, penelitian ini juga akan menganalisis

dampak dari pembaruan database terhadap kinerja dan akurasi locate[2].

2.6. Kinerja Pencarian File dalam Sistem Operasi

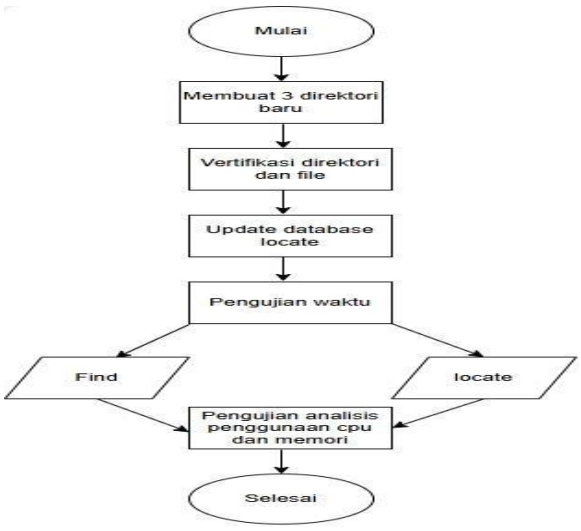
Kinerja pencarian file dalam sistem operasi Linux sangat dipengaruhi oleh beberapa faktor, termasuk ukuran dan struktur direktori, jumlah file yang ada, serta jenis alat pencarian yang digunakan. Dalam konteks perbandingan antara find dan locate, waktu eksekusi pencarian, tingkat akurasi hasil pencarian, dan penggunaan sumber daya (CPU, memori) akan menjadi indikator utama yang digunakan dalam penelitian ini. Selain itu, fleksibilitas dalam menggunakan filter pencarian juga akan menjadi faktor penting dalam menganalisis kinerja kedua alat tersebut. Sistem yang memiliki file yang sangat banyak, seperti server dengan data besar, akan menunjukkan perbedaan kinerja yang signifikan antara kedua perintah pencarian ini[4].

3. METODE PENELITIAN

3.1. Jenis penelitian

Metode yang kami gunakan dalam penelitian ini adalah eksperimen komparatif yang bertujuan untuk membandingkan kinerja pencarian file menggunakan dua perintah, yaitu find dan locate, pada penelitian ini eksperimen dilakukan dengan cara mengukur waktu eksekusi, keakuratan hasil pencarian, serta fleksibilitas dalam pencarian file.

3.2. Tahapan dan prosedur penelitian



Gambar 1. Tahapan Penelitian

3.3. Lingkungan Pengujian

Perangkat keras : CPU Intel Core i3, RAM 8
Platfrom Virtualisasi : VirtualBox 7.1.4
Sistem Operasi Virtual : Red Hat 9.0

4. HASIL DAN PEMBAHASAN

Penelitian ini berfokus untuk menunjukkan hasil pengujian waktu eksekusi antara perintah find dan locate untuk tiga direktori yang memiliki jumlah file yang berbeda. Pengujian penelitan ini dilakukan pada direktori dengan jumlah file yang kecil (100 file), file menengah (1.000 file), dan file besar (10.000 file).

1. Membuat direktori

Untuk membuat direktori kita menggunakan perintah mkdir (make directory)

```
[redhatpsik23dkel3@vbox ~]$ sudo mkdir -p /home/user/tes_kecil /home/user/tes-me
nengah /home/user/tes_besar
[sudo] password for redhatpsik23dkel3:
```

Gambar 1. Membuat directory

2. Menambahkan file dalam direktori

Perintah dibawah akan menambahkan 100 file pada (tes_kecil), 1.000 file pada (tes_menengah), dan 10.000 file pada (tes_besar)

```
[redhatpsik23dkel3@vbox ~]$ sudo bash -c 'for i in {1..100}; do touch /home/user
/tes_kecil/file_$i.txt; done'
[redhatpsik23dkel3@vbox ~]$ sudo bash -c 'for i in {1..1000}; do touch /home/use
r/tes_menengah/file_$i.txt; done'
[redhatpsik23dkel3@vbox ~]$ sudo bash -c 'for i in {1..10000}; do touch /home/us
er/tes_besar/file_$i.txt; done'
```

Gambar 2. Menambahkan file dalam direktori

3. Vertifikasi daftar isi file dari direktori

Perintah dibawah adalah perintah untuk memastikan apakah direktori yang telah dibuat (tes_kecil, tes_menengah, tes_besar) ada di dalam direktori rumah (/home/user)

```
[redhatpsik23dkel3@vbox ~]$ ls -1 /home/user
tes_besar
tes_kecil
tes-menengah
tes_menengah
```

Gambar 3. Vertifikasi daftar isi file dari direktori

4. Verifikasi file yang telah dibuat

```
[redhatpsik23dkel3@vbox ~]$ ls /home/user/tes_kecil
file_100.txt  file_25.txt  file_40.txt  file_56.txt  file_71.txt  file_87.txt
file_10.txt   file_26.txt  file_41.txt  file_57.txt  file_72.txt  file_88.txt
file_11.txt   file_27.txt  file_42.txt  file_58.txt  file_73.txt  file_89.txt
file_12.txt   file_28.txt  file_43.txt  file_59.txt  file_74.txt  file_8.txt
file_13.txt   file_29.txt  file_44.txt  file_5.txt   file_75.txt  file_90.txt
file_14.txt   file_2.txt   file_45.txt  file_60.txt  file_76.txt  file_91.txt
file_15.txt   file_30.txt  file_46.txt  file_61.txt  file_77.txt  file_92.txt
file_16.txt   file_31.txt  file_47.txt  file_62.txt  file_78.txt  file_93.txt
file_17.txt   file_32.txt  file_48.txt  file_63.txt  file_79.txt  file_94.txt
file_18.txt   file_33.txt  file_49.txt  file_64.txt  file_7.txt   file_95.txt
file_19.txt   file_34.txt  file_4.txt   file_65.txt  file_80.txt  file_96.txt
file_1.txt    file_35.txt  file_50.txt  file_66.txt  file_81.txt  file_97.txt
file_20.txt   file_36.txt  file_51.txt  file_67.txt  file_82.txt  file_98.txt
file_21.txt   file_37.txt  file_52.txt  file_68.txt  file_83.txt  file_99.txt
file_22.txt   file_38.txt  file_53.txt  file_69.txt  file_84.txt  file_9.txt
file_23.txt   file_39.txt  file_54.txt  file_6.txt   file_85.txt
file_24.txt   file_3.txt   file_55.txt  file_70.txt  file_86.txt
```

Gambar 4. Vertifikasi file tes_kecil

```
[redhatpsik23dkel3@vbox ~]$ ls /home/user/tes_menengah
file_1000.txt  file_280.txt  file_460.txt  file_640.txt  file_820.txt
file_100.txt   file_281.txt  file_461.txt  file_641.txt  file_821.txt
file_275.txt   file_455.txt  file_635.txt  file_815.txt  file_996.txt
file_276.txt   file_456.txt  file_636.txt  file_816.txt  file_997.txt
file_277.txt   file_457.txt  file_637.txt  file_817.txt  file_998.txt
file_278.txt   file_458.txt  file_638.txt  file_818.txt  file_999.txt
```

Gambar 5. Vertifikasi file tes_menengah

```
[redhatpsik23dkel3@vbox ~]$ ls /home/user/tes_besar
file_2793.txt  file_4593.txt  file_6393.txt  file_8193.txt  file_9994.txt
file_2794.txt  file_4594.txt  file_6394.txt  file_8194.txt  file_9995.txt
file_2795.txt  file_4595.txt  file_6395.txt  file_8195.txt  file_9996.txt
file_2796.txt  file_4596.txt  file_6396.txt  file_8196.txt  file_9997.txt
file_2797.txt  file_4597.txt  file_6397.txt  file_8197.txt  file_9998.txt
file_2798.txt  file_4598.txt  file_6398.txt  file_8198.txt  file_9999.txt
```

Gambar 6. Vertifikasi file tes_besar

5. Update database locate

Memerbarui database locate dengan perintah updatedb, dilakukan agar pencarian menggunakan locate selalu mencakup file dan direktori yang terbaru

```
[redhatpsik23dkel3@vbox ~]$ sudo updatedb
[sudo] password for redhatpsik23dkel3:
```

Gambar 7. Update database locate

6. Pengujian perintah find

Uji kinerja perintah find dengan menggunakan perintah **time** untuk mengukur waktu eksekusi pencarian file.

```
[redhatpsik23dkel3@vbox ~]$ time find /home/user/tes_kecil -name "*.txt"
/home/user/tes_kecil/file_94.txt
/home/user/tes_kecil/file_95.txt
/home/user/tes_kecil/file_96.txt
/home/user/tes_kecil/file_97.txt
/home/user/tes_kecil/file_98.txt
/home/user/tes_kecil/file_99.txt
/home/user/tes_kecil/file_100.txt

real    0m0,010s
user    0m0,002s
sys     0m0,005s
```

Gambar 8. Pengujian waktu Pencarian file dengan perintah find di (tes_kecil)

```
[redhatpsik23dkel3@vbox ~]$ time locate "/home/user/tes_menengah/*.txt"
/home/user/tes_menengah/file_994.txt
/home/user/tes_menengah/file_995.txt
/home/user/tes_menengah/file_996.txt
/home/user/tes_menengah/file_997.txt
/home/user/tes_menengah/file_998.txt
/home/user/tes_menengah/file_999.txt

real    0m0,742s
user    0m0,472s
sys     0m0,010s
```

Gambar 9. Waktu Pencarian file dengan perintah find di (tes_menengah)

```
[redhatpsik23dkel3@vbox ~]$ time find /home/user/tes_besar -name "*.txt"
/home/user/tes_besar/file_9994.txt
/home/user/tes_besar/file_9995.txt
/home/user/tes_besar/file_9996.txt
/home/user/tes_besar/file_9997.txt
/home/user/tes_besar/file_9998.txt
/home/user/tes_besar/file_9999.txt
/home/user/tes_besar/file_10000.txt

real    0m0,985s
user    0m0,080s
sys     0m0,161s
```

Gambar 10 . Waktu Pencarian file dengan perintah find di (tes_besar)

7. Pengujian perintah locate

Uji kinerja peerintah locate dengan menggunakan perintah **time** untuk mengukur waktu eksekusipencarian file.

```
[redhatpsik23dkel3@vbox ~]$ time locate "/home/user/tes_kecil/*.txt"
/home/user/tes_kecil/file_94.txt
/home/user/tes_kecil/file_95.txt
/home/user/tes_kecil/file_96.txt
/home/user/tes_kecil/file_97.txt
/home/user/tes_kecil/file_98.txt
/home/user/tes_kecil/file_99.txt

real    0m0,583s
user    0m0,422s
sys     0m0,008s
```

Gambar 11. Waktu pencarian perintah locate (tes_kecil)

```
[redhatpsik23dkel3@vbox ~]$ time locate "/home/user/tes_menengah/*.txt"
/home/user/tes_menengah/file_994.txt
/home/user/tes_menengah/file_995.txt
/home/user/tes_menengah/file_996.txt
/home/user/tes_menengah/file_997.txt
/home/user/tes_menengah/file_998.txt
/home/user/tes_menengah/file_999.txt

real    0m0,742s
user    0m0,472s
sys     0m0,010s
```

Gambar 12. Waktu pencarian perintah locate (tes_menengah)

```
[redhatpsik23dkel3@vbox ~]$ time locate "/home/user/tes_besar/*.txt"
/home/user/tes_besar/file_9994.txt
/home/user/tes_besar/file_9995.txt
/home/user/tes_besar/file_9996.txt
/home/user/tes_besar/file_9997.txt
/home/user/tes_besar/file_9998.txt
/home/user/tes_besar/file_9999.txt

real    0m1,532s
user    0m0,539s
sys     0m0,134s
```

Gambar 13. Waktu pencarian perintah locate (tes_besar)

8. Analisis penggunaan CPU dan Memori

Untuk menganalis pengggunaan CPU dan memeori, kita dapat menggunakan perintah /usr/bin/time -v untuk kedua perintah find dan locate.

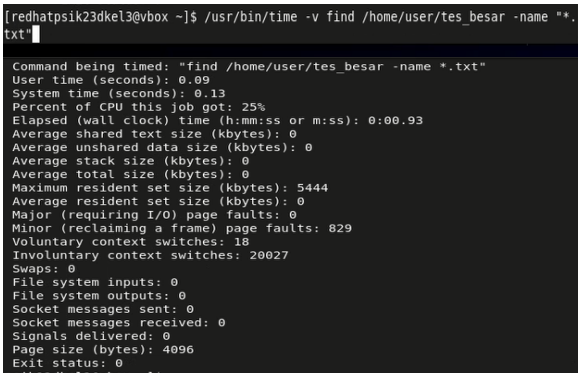
4.1. Penggunaan CPU dan memori dengan perintah find:

```
[redhatpsik23dkel3@vbox ~]$ /usr/bin/time -v find /home/user/tes_kecil -name "*.txt"
Command being timed: "find /home/user/tes_kecil -name *.txt"
User time (seconds): 0.00
System time (seconds): 0.00
Percent of CPU this job got: 17%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.03
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 2904
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 129
Voluntary context switches: 1
Involuntary context switches: 207
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```

Gambar 14. Tes_kecil (100 file)

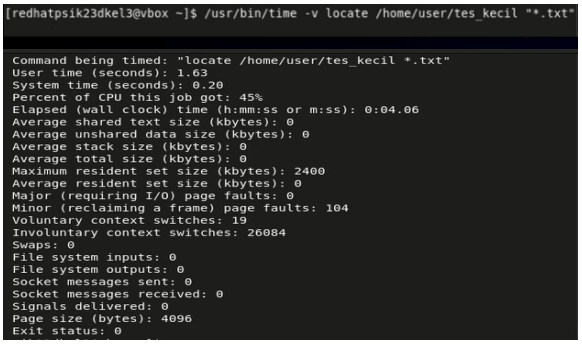
```
[redhatpsik23dkel3@vbox ~]$ /usr/bin/time -v find /home/user/tes_menengah -name "*.txt"
Command being timed: "find /home/user/tes_menengah -name *.txt"
User time (seconds): 0.01
System time (seconds): 0.02
Percent of CPU this job got: 23%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.13
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 3140
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 200
Voluntary context switches: 1
Involuntary context switches: 2009
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```

Gambar 15. Tes_menengah (1000 file)

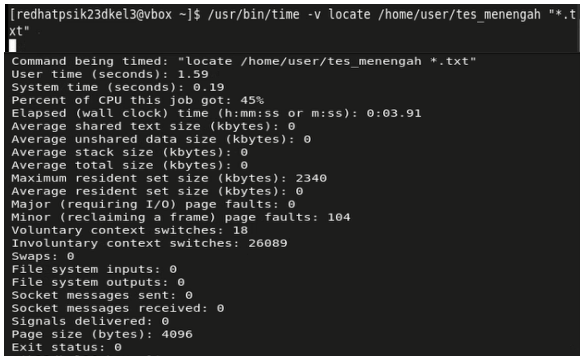


Gambar 16. Tes_besar (10.000 file)

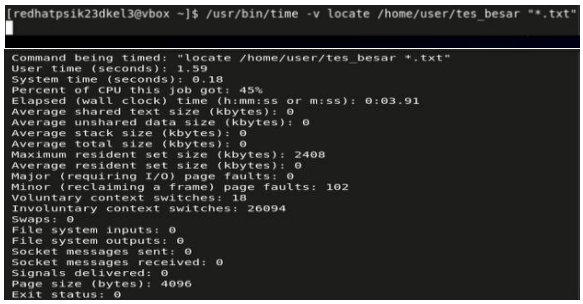
9. Penggunaan CPU dan memori dengan locate :



Gambar 17. Tes_kecil(100 file)



Gambar 18. Tes_menengah(1000 file)



Gambar 19. Tes_besar(10000 file)

4.2. Perbandingan waktu eksekusi pencarian file perintah find dan locate

Tabel 1. Perbandingan waktu eksekusi pencarian file perintah find dan locate

Direktori	Waktu find (real)	Waktu find (user)	Waktu find (sys)	Waktu locate (real)	Waktu locate (user)	Waktu locate (sys)
Tes_kecil	0m0.10 s	0m0.002s	0m0.005s	0m0.583s	0m0.422s	0m0.008s
Tes_menengah	0m0.079s	0m0.009s	0m0.016s	0m0.743s	0m0.472s	0m0.010s
Tes_besar	0m0.985	0m0.80s	0m0.161s	0m1.532s	0m0.539	0m0.134s

Pada waktu eksekusi, find lebih cepat dibandingkan dengan locate pada direktori kecil (tes_kecil), direktori menengah (tes_menengah), dan direktori besar (tes_besar) yaitu:
Pada direktori kecil (tes_kecil)
Waktu find (real) = 0m0.010s
Waktu find (user) = 0m0.002s
Waktu find (system) = 0m0.005s
Pada direktori menengah (tes_menengah)
Waktu find (real) = 0m0.079s
Waktu find (user) = 0m0.009s
Waktu find (system) = 0m0.0016s
Pada direktori besar (tes_besar) yaitu:
Waktu find (real) = 0m0.985s
Waktu find (user) = 0m0.080s
Waktu find (system) = 0m0.161s,
Sedangkan waktu eksekusi pada perintah locate pada direltori kecil (tes_kecil), direktori menengah (tes_menengah), dan direktori besar (tes_besar) yaitu:
Pada direltori kecil (tes_kecil)
Waktu find (real) = 0m0.583s
Waktu find (user) = 0m0.422s
Waktu find (system) = 0m0.008s
Pada direktori menengah (tes_menengah),
Waktu find (real) = 0m0.742s
Waktu find (user) = 0m0.472s
Waktu find (system) = 0m0.010s
Namun perbedaannya semakin kecil pada direktori yang lebih besar, dimana find hanya sedikit lebih cepat dari locate. Yaitu dengan waktu:
Waktu find (real) = 0m1.532s
Waktu find (user) = 0m0.539s
Waktu find (system) = 0m0.134s
Locate membutuhkan waktu lebih lama, terutama pada tes_kecil dan tes_menengah

4.3. Perbandingan penggunaan CPU dan memori perintah find dan locate

Tabel 2. Perbandingan penggunaan CPU dan memori perintah find dan locate

Direktori	Perintah	User time (s)	System time (s)	% CPU used	Max resident size (KB)	Minor page faults	Voluntary context switches	Involuntary context switches
Tes_kecil	find	0.00	0.00	17%	2940	129	1	207
	Locate	1,63	0.20	45%	2400	104	19	26084
Tes_menengah	Find	0.01	0.02	23%	3140	200	1	2009
	Locate	1.59	0.19	45%	2340	104	18	26089
Tes_besar	Find	0.09	0.13	25%	5444	829	18	20027
	Locate	1.59	0.18	45%	2408	102	18	26094

Pada analisis penggunaan CPU, find menggunakan CPU jauh lebih rendah disbanding dengan locate, yaitu dengan:

Direktori kecil
Persentase find = 17%
Persentase locate = 45%

Direktori menengah
Persentase find = 23%
Persentase locate = 45%

Direktori besar
Persentase find = 25%
Persentase locate = 45%

Penggunaan CPU tetap lebih tinggi untuk perintah locate pada setiap ukuran direktori, yang menunjukkan bahwa meskipun lebih cepat dalam pencarian, locate lebih membebani CPU dibanding dengan find.

Pada penggunaan memori, find membutuhkan lebih banyak memori dibanding dengan locate, terutama pada tes_besar, Dimana penggunaan memorinya yaitu:
Direktori kecil
Find = 2904 kb
Locate = 2400 kb

Direktori menengah
Find = 3140 kb
Locate = 2340 kb

Direktori besar
Find = 5444 kb
Locate = 2408 kb

Find juga menghasilkan lebih banyak minor page faults dibandingkan dengan locate yaitu

minor page faults pada find :
direktori kecil = 129
direktori menengah = 200
direktori besar = 829

sedangkan minor page faults pada locate :
direktori kecil = 104
direktori menengah = 104
direktori besar = 102

Dan sedikit voluntary context switches yaitu:
voluntary context switches pada find :
direktori kecil = 1
direktori menengah = 1
direktori besar = 18

voluntary context switches pada locate :
direktori kecil = 19
direktori menengah = 18
direktori besar = 18

yang menunjukkan bahwa proses find lebih jarang menunggu operasi I/O dan berinteraksi dengan proses lain dibandingkan dengan locate.

5. KESIMPULAN

Berdasarkan penelitian, perintah find lebih unggul dalam fleksibilitas dan akurasi pencarian, terutama untuk direktori kecil, karena bekerja langsung pada sistem file, meski membutuhkan waktu eksekusi lebih lama pada direktori besar. Sebaliknya, locate lebih cepat dalam pencarian sederhana pada direktori besar karena menggunakan database yang diperbarui secara berkala, meskipun hasilnya kurang akurat jika database tidak mutakhir. Dari sisi penggunaan sumber daya, find lebih efisien dalam konsumsi CPU, sedangkan locate membutuhkan memori lebih rendah, menjadikannya pilihan yang tepat untuk pencarian cepat di sistem file besar dengan pembaruan database teratur.

DAFTAR PUSTAKA

[1] De Souza, C. H. (2023). The Red Hat Enterprise Linux Business Model. In Business Models and Strategies for Open Source Projects (pp. 282-303). IGI Global.

[2] Rizqi Barok, Lestari A Andriani.(2023). Implementasi Secure Storage Menggunakan Metode Full Disk Encryption dan Tamper Proof pada Cloud Storage. *Jurnal Ilmiah Kriptologi*.Volume 17 No(1). Jurnal Ilmiah Kriptologi.

[3] Craig A. N. Soules, Gregory R. Ganger(2020) Connections: Using Context to Enhance File Search. Carnegie Mellon .niversity

[4] Smith, J., & White, L. (2020). Performance Analysis of File Search Tools in Linux Systems. *Journal of Linux Systems*, 18(3), 230-240.

[5] Johnson, R., et al. (2021). A Comparative Study of File Search Performance in Linux Environments: Using Find vs. Locate. *International Journal of Computing*, 25(5), 45-60.

[6] Davis, T., & Morgan, S. (2022). Efficient File Searching in Linux: The Role of Find and Locate Commands. *Journal of Computer Science*, 29(1), 76-89.

[7] Lee, H., et al. (2023). Optimizing File Search on Large Linux File Systems: A Case Study of Red Hat. Proceedings of the International Conference on Linux Systems, 12, 112-118.

[8] Rahaja R Anton. (2020). PENGENALAN LINUX. Open Source Campus Agreement.

[9] Vugt V Sander. (2023). Red Hat Enterprise Linux 6 Administration Real Word Skill For Red Hat Administration. Canada:John Wiley & Sons.

[10] Ghoiri Asghar. (2020). RHCSA Red Hat Enterprise Linux 8 (UPDATED). USA:InsgramSpark.inc.