**Laporan Latihan Praktikum ke-5  
Sistem Operasi**

**Disusun oleh:**

**Arya Yudhistira | 121140115**

**Kelas RD | Cluster RD**



**Program Studi Teknik Informatika**

**Jurusan Teknologi Produksi dan Industri**

**Institut Teknologi Sumatera**

**Lampung Selatan**

**2023**

**BAB I  
TEORI DASAR**

**1.1 Proses Input / Output**

Proses input/output (I/O) merupakan cara komunikasi antara program dan perangkat input/output. Proses input melibatkan pengambilan data atau informasi dari sumber eksternal, seperti keyboard, mouse, atau file. Proses output melibatkan penampilan data atau informasi ke perangkat eksternal, seperti monitor, printer, atau file.

Proses input dilakukan melalui berbagai perangkat input, seperti keyboard atau mouse. Dalam pemrograman, proses input juga dapat dilakukan melalui membaca data dari file, membaca input dari argumen baris perintah, atau melalui interaksi dengan pengguna melalui antarmuka grafis.

Proses output melibatkan penampilan data kepada pengguna atau menyimpan data ke perangkat penyimpanan, seperti file. Output dapat berupa teks, grafik, atau suara tergantung pada jenis perangkat yang digunakan.

**1.2 File Descriptor**

File descriptor adalah mekanisme yang digunakan oleh sistem operasi untuk mengidentifikasi dan mengelola file yang sedang dibuka oleh suatu proses. Setiap file yang dibuka oleh proses memiliki file descriptor yang unik. File descriptor biasanya berupa bilangan bulat non-negatif.

Sistem operasi menyediakan file descriptor sebagai abstraksi untuk merepresentasikan file yang dibuka oleh suatu proses. Dalam pemrograman, file descriptor digunakan untuk melakukan operasi I/O pada file, seperti membuka, membaca, menulis, atau menutup file. Dalam lingkungan Unix, tiga file descriptor standar secara otomatis disediakan untuk setiap proses:

1. File descriptor 0 (stdin): Merupakan file descriptor standar untuk input. Biasanya terhubung ke keyboard atau sumber input lainnya.
2. File descriptor 1 (stdout): Merupakan file descriptor standar untuk output. Biasanya terhubung ke layar atau perangkat output lainnya.
3. File descriptor 2 (stderr): Merupakan file descriptor standar untuk output error. Biasanya terhubung ke layar atau perangkat output error lainnya.

**1.3 Redirection (Pembelokan)**

Redirection adalah proses mengalihkan input atau output dari atau ke sumber atau tujuan yang berbeda. Dalam sistem operasi Unix dan sebagian besar shell, Anda dapat menggunakan operator redirection untuk mengarahkan input atau output. Operator ">" digunakan untuk mengalihkan output ke file. Misalnya, perintah ls > file.txt akan mengalihkan output dari perintah ls ke file bernama "file.txt" daripada menampilkannya di layar. Operator "<" digunakan untuk mengalihkan input dari file. Misalnya, perintah sort < input.txt akan mengambil input dari file "input.txt" daripada dari keyboard. Operator ">>" digunakan untuk menambahkan output ke file yang sudah ada. Misalnya, perintah echo "baris baru" >> file.txt akan menambahkan "baris baru" ke dalam file "file.txt" tanpa menghapus kontennya yang sudah ada.

**1.4 Pipeline**

Pipeline adalah teknik yang digunakan untuk mengalirkan output suatu perintah atau program ke input perintah atau program lainnya dalam urutan yang terhubung. Dalam pipeline, output dari satu proses menjadi input untuk proses berikutnya, dan demikian seterusnya.

Operator "|" (pipe) digunakan untuk menghubungkan perintah atau program dalam pipeline. Misalnya, perintah ls | grep "file" akan mengalirkan output dari perintah ls ke input dari perintah grep "file". Dengan demikian, hanya file yang cocok dengan pola "file" yang akan ditampilkan sebagai hasil akhir.

Penggunaan pipeline memungkinkan pengguna untuk melakukan pengolahan data yang kompleks dengan menggabungkan beberapa perintah atau program ke dalam satu baris. Ini memberikan fleksibilitas dan kekuatan dalam memanipulasi, menyaring, dan menganalisis data.

**1.5 Filter**

Filter adalah program yang melakukan transformasi pada input yang diberikan dan menghasilkan output yang dimodifikasi. Filter biasanya digunakan dalam kombinasi dengan pipeline untuk melakukan manipulasi data secara berurutan.

Filter dapat digunakan untuk mencocokkan pola teks, mengganti teks, mengurutkan data, menghitung statistik, dan melakukan berbagai operasi pemrosesan data lainnya. Beberapa contoh filter yang umum digunakan adalah "grep" untuk mencocokkan pola teks, "sort" untuk mengurutkan data, "sed" untuk melakukan penggantian teks, dan "awk" untuk melakukan pemrosesan teks berbasis pola.

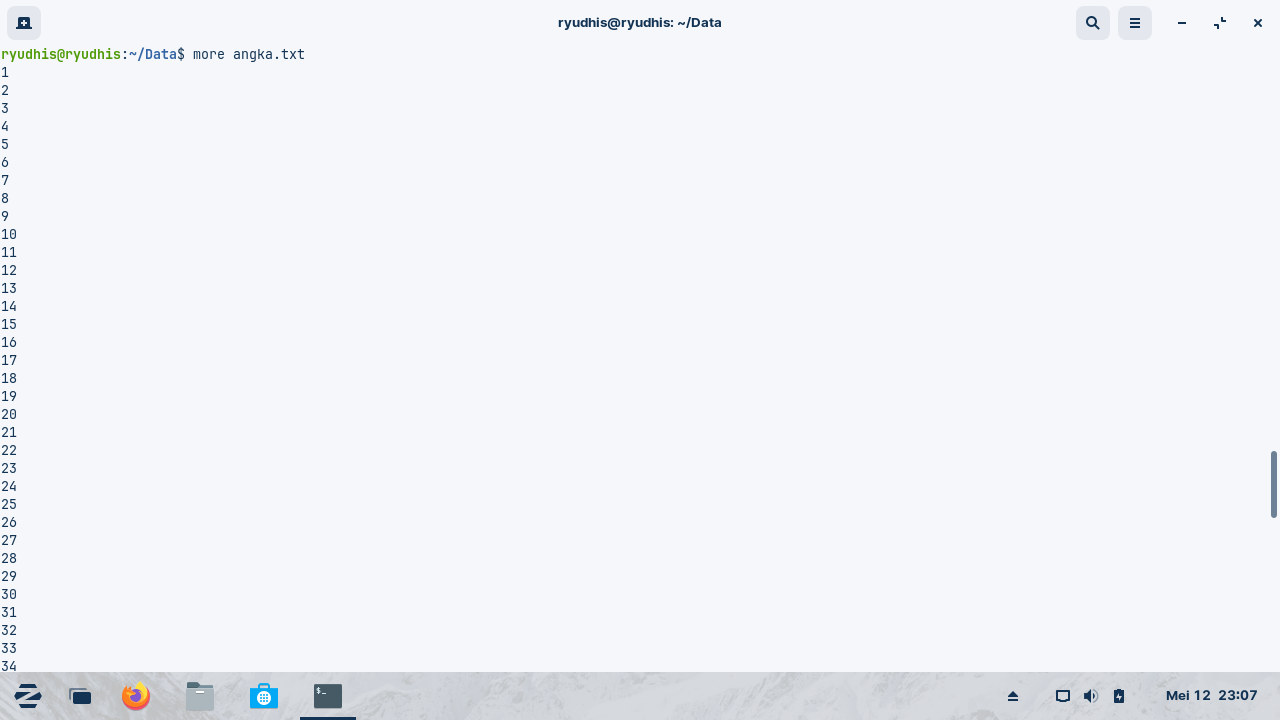
Dalam penggunaan filter, output dari satu filter digunakan sebagai input untuk filter berikutnya dalam pipeline. Dengan memadukan beberapa filter, pengguna dapat melakukan manipulasi dan analisis data secara efisien, memisahkan tugas ke dalam unit yang lebih kecil, dan memperoleh hasil yang diinginkan dengan langkah-langkah yang terurut.

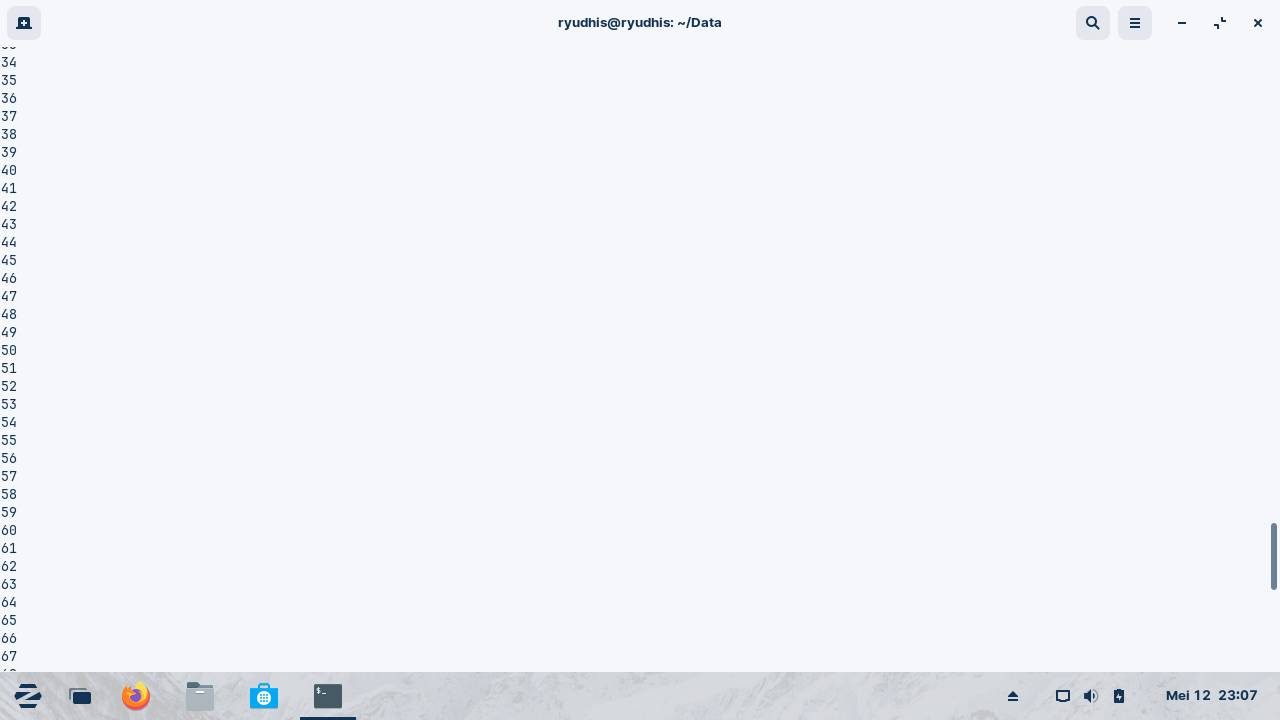
**BAB II  
PEMBAHASAN DAN ANALISIS**

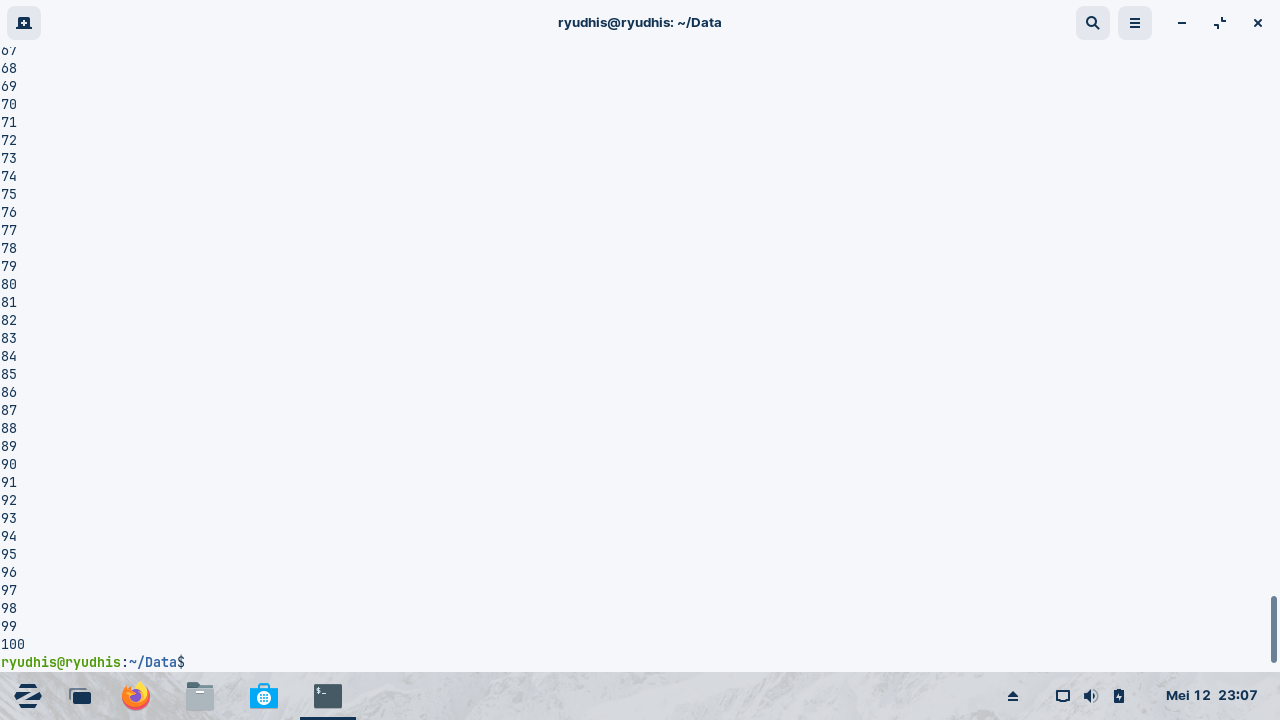
**2.1 Bagaimana perbedaan antara command more dan less, dan implementasikan pada terminal**

**2.1.1 More**

Sebelum melakukan percobaan, sudah dibuat file angka.txt yang berisi angka 1-100. Kemudian dilakukan percobaan **more** pada file angka.txt dengan sintaks **more angka.tx**t, dapat dilihat bahwa command **more** dapat mengoutputkan isi dari suatu file secara menyeluruh dalam terminal.

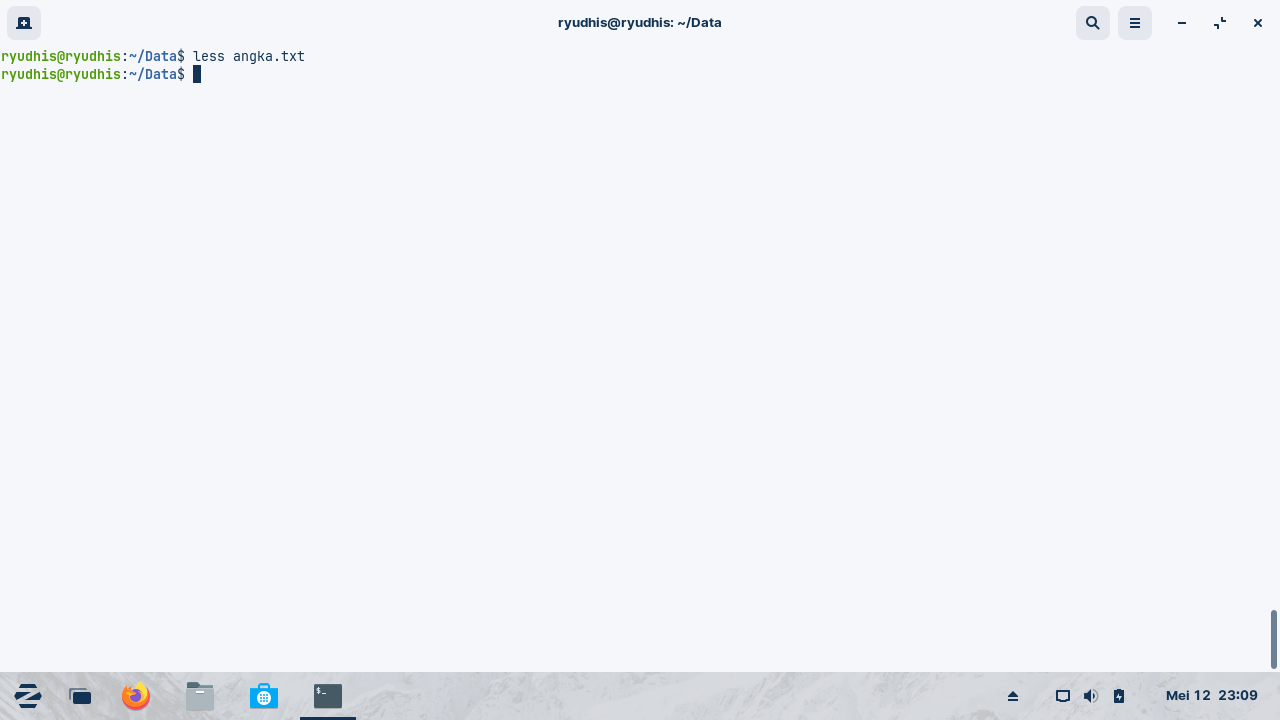


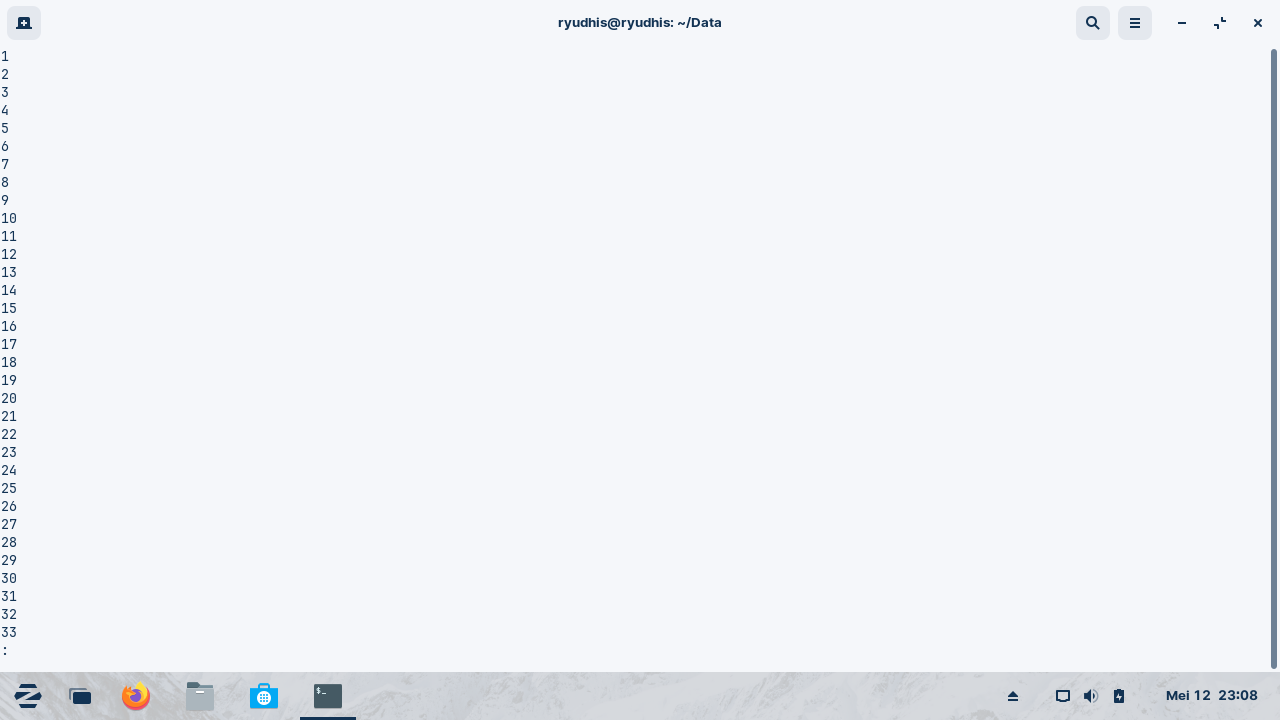




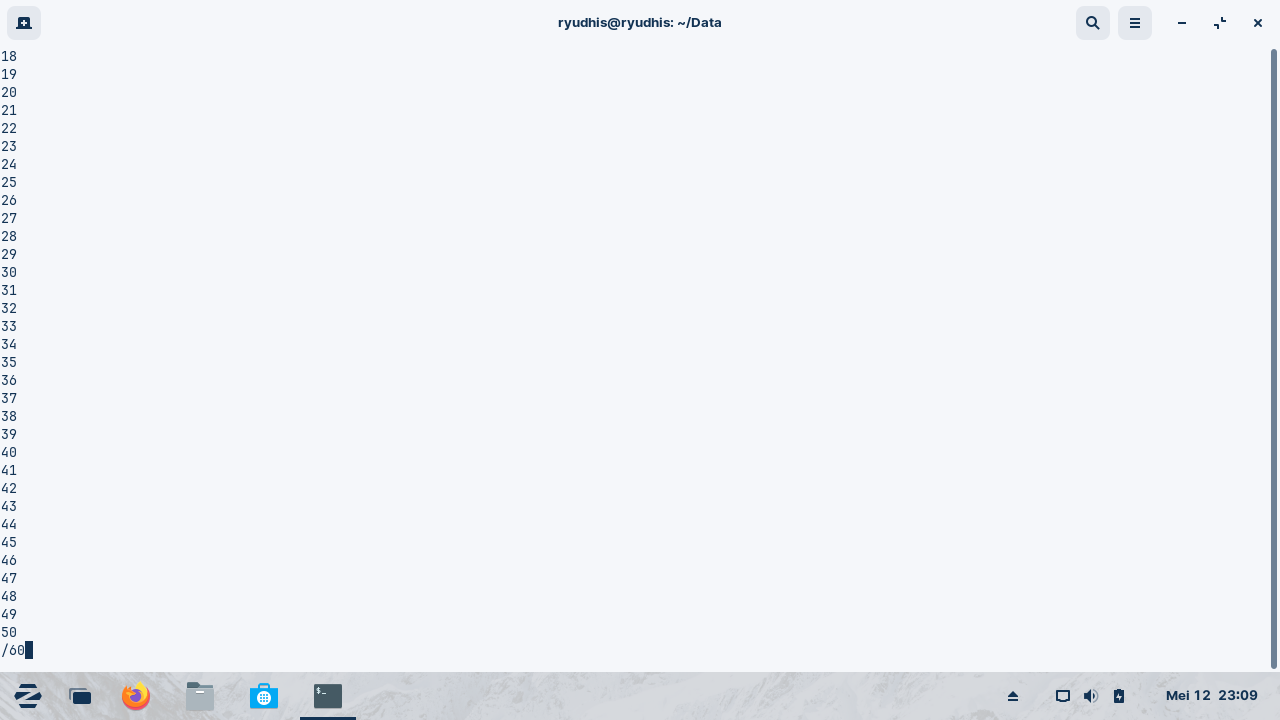
**2.1.2 Less**

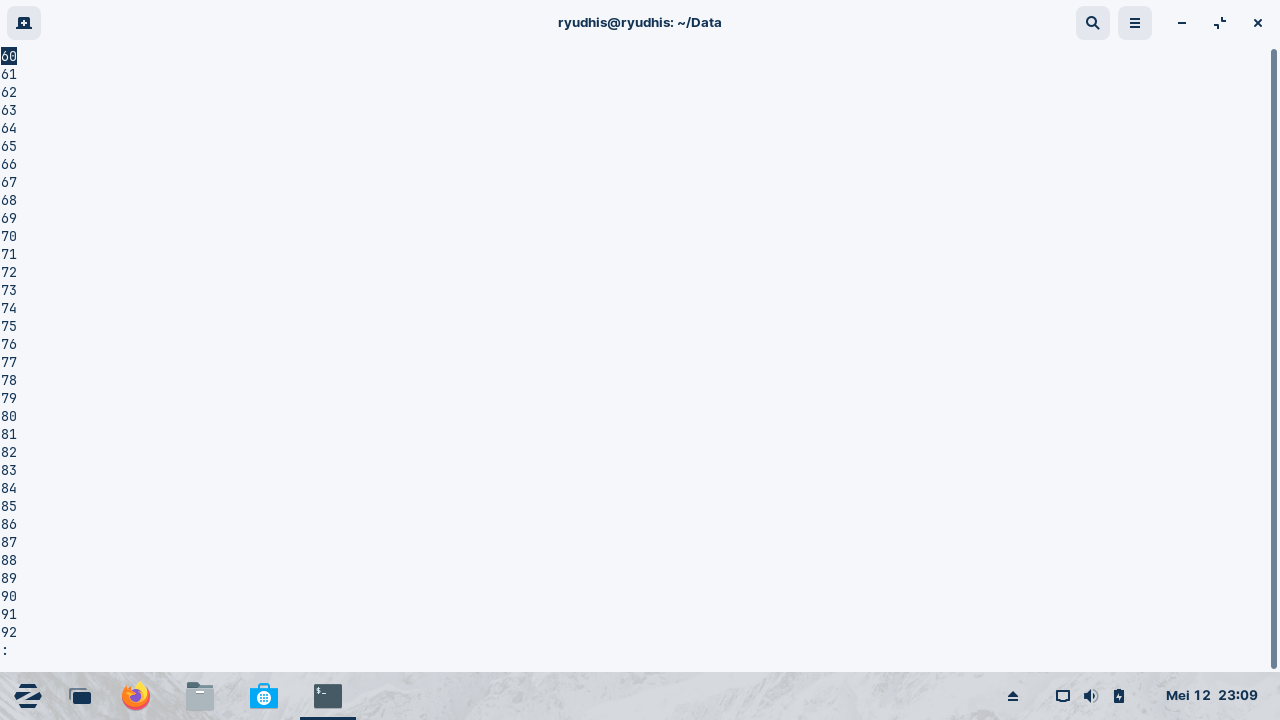
Melakukan percobaan **less** pada file angka.txt dengan sintaks **less angka.tx**t, dapat dilihat bahwa command **less** dapat mengoutputkan isi dari suatu file secara menyeluruh namun bukan dalam terminal yang sama ketika menjalankan commandnya, melainkan terbuka ui terminal lain khusus command less. Untuk melakukan scrolling baris data pada command **less**, digunakan key down dan key up pada keyboard berbeda dengan command more yang menggunakan scroll pada mouse.



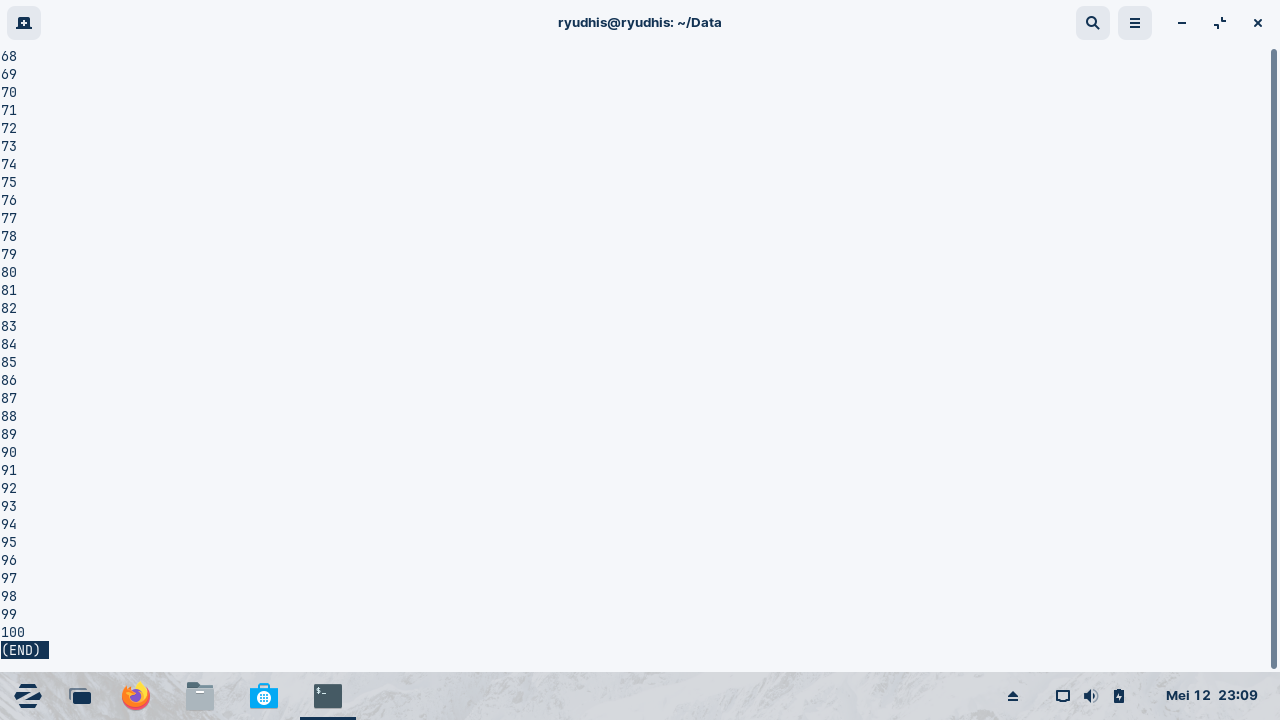


Kemudian command less memiliki fitur search yang dapat digunakan untuk mencari suatu string yang terdapat pada file tersebut, contoh penggunaannya adalah **/60** yang berarti mencari string “60”, maka setelah ditekan enter semua string yang mengandung nilai “60” akan terhighlight.





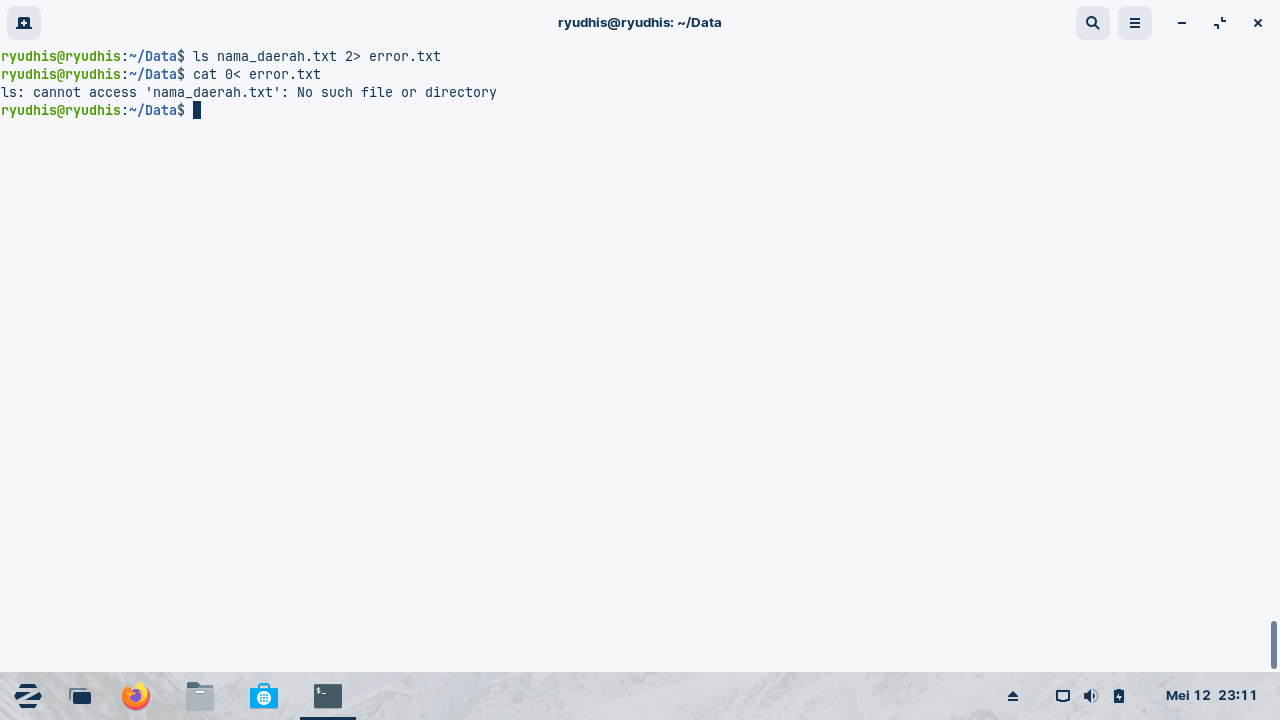
Kemudian pada baris paling akhir akan ada text (END) yang menandakan akhir dari baris string yang dimiliki file tersebut.



**2.2 Buatlah sebuah file yang berisikan daftar nama daerah dengan menggunakan file descriptor dan redirection dengan mengimplementasikan penggunaan standar input, standar output dan standar error.**

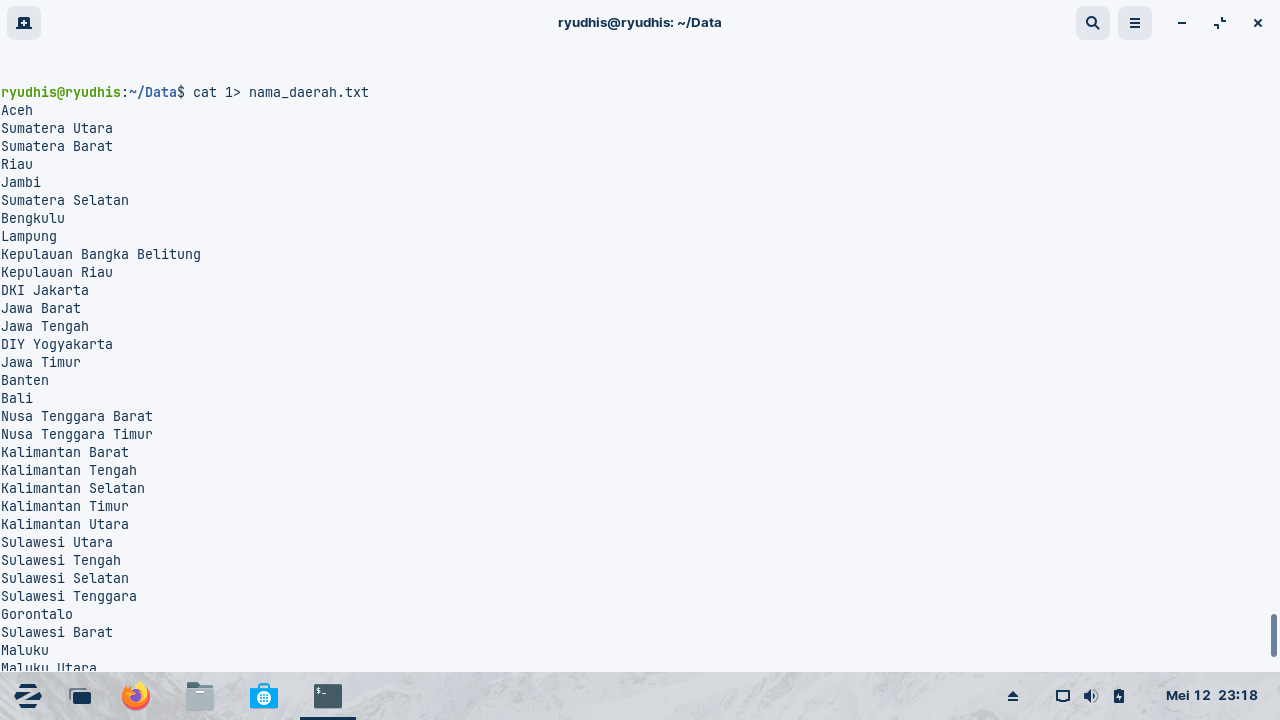
**2.2.1 Redirection standar error**

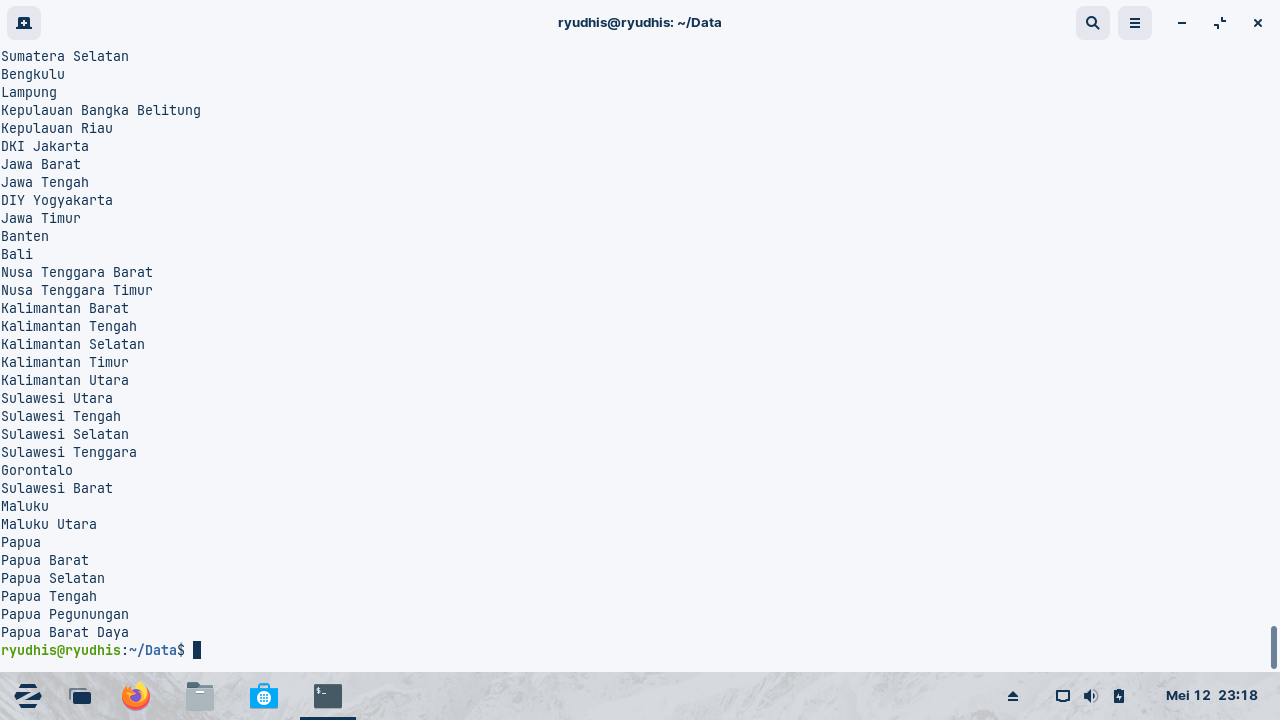
Melakukan penerapan redirection standar error pada list file **ls** menggunakan sintaks **ls nama\_daerah.txt 2> error.txt**, maka file error.txt akan terisi hasil error yaitu No such file or directory dikarenakan file nama\_daerah.txt tidak ada atau belum dibuat.



**2.2.2 Redirection standar output**

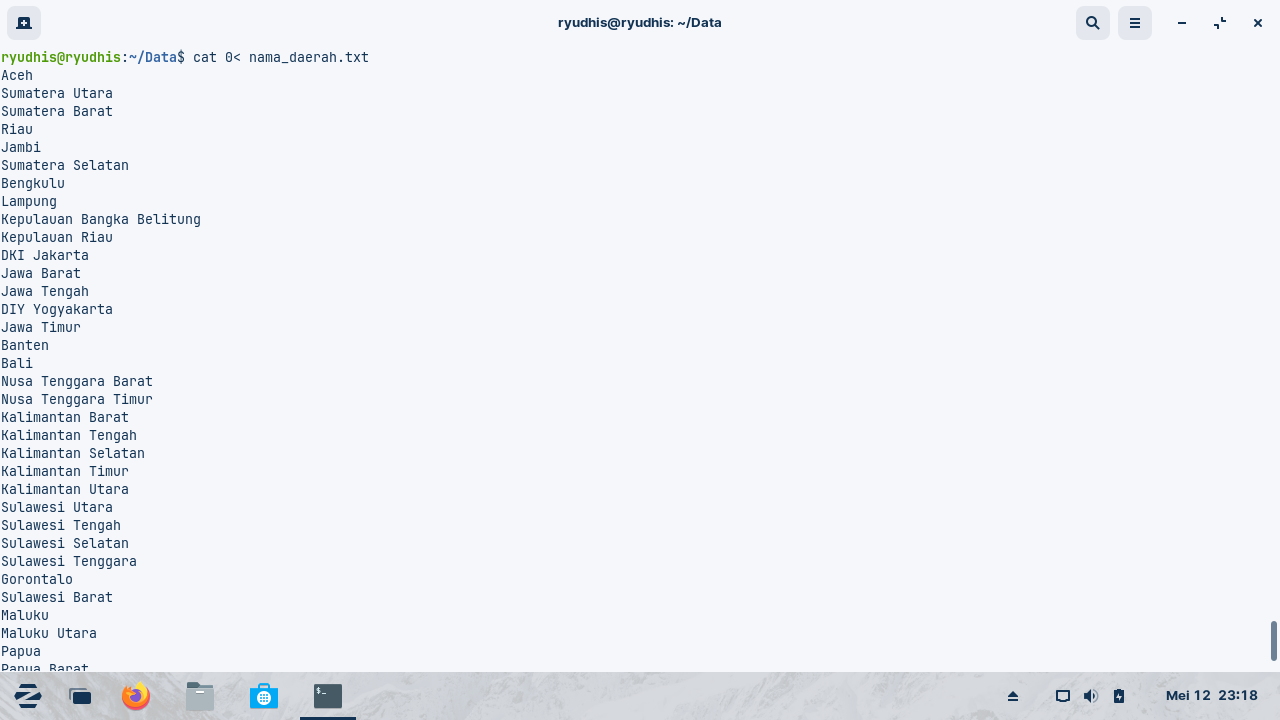
Melakukan penerapan redirection standar output ke dalam file baru nama\_daerah.txt menggunakan sintaks **cat 1> nama\_daerah.txt**, kemudian menginputkan 38 nama provinsi di Indonesia, setelah selesai maka tekan ctrl+d untuk mengakhiri inputan dan file otomatis tersimpan.

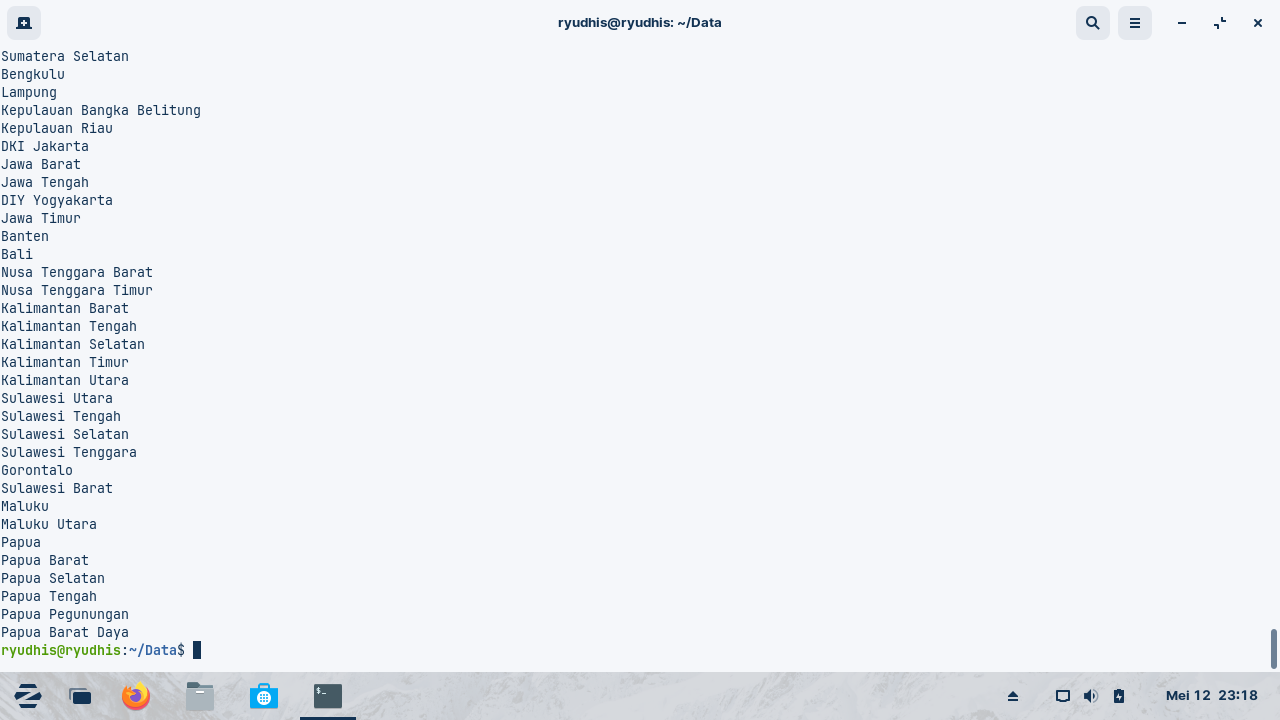


****

**2.2.2 Redirection standar input**

Melakukan penerapan redirection standar input pada file nama\_daerah.txt menggunakan sintaks **cat 0< nama\_daerah.txt**, kemudian akan keluar output isi file nama\_daerah.txt yang sebelumnya kita buat.

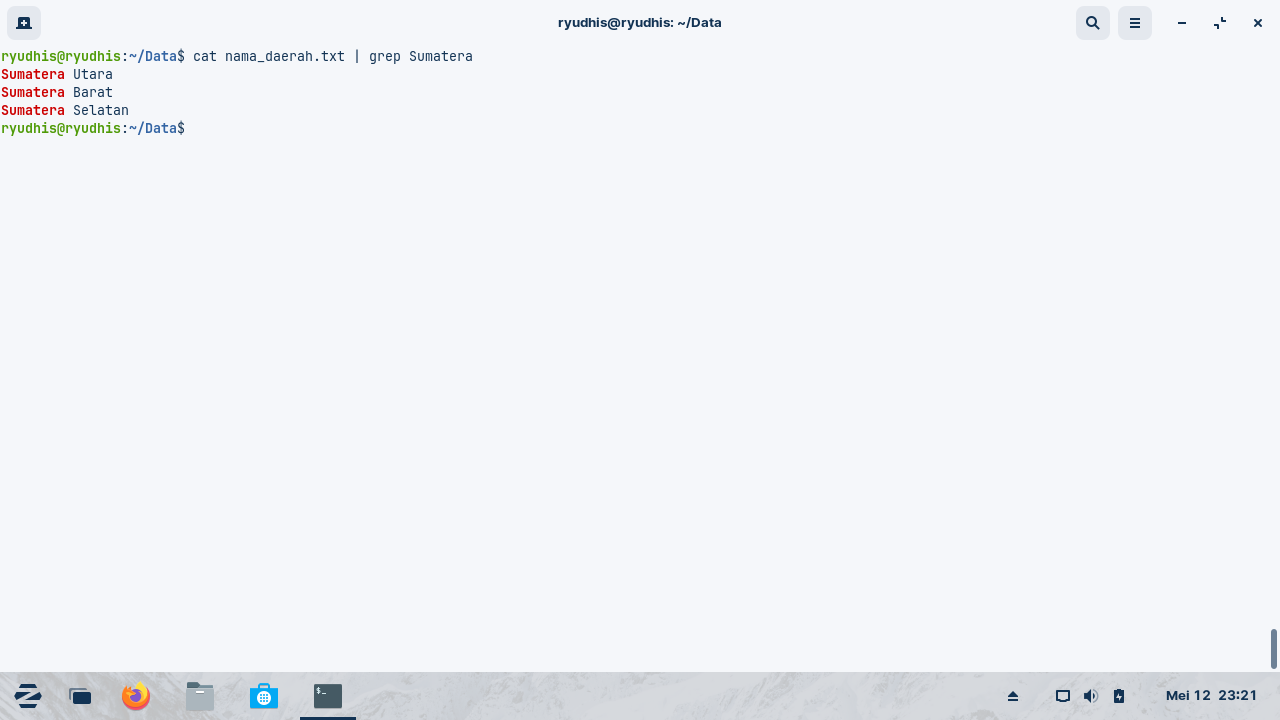


****

**2.3 Tampilkan isi file tersebut dengan menggunakan implementasi dari pipeline dan filter yaitu perintah Grep, Wc, Uniq, Sort, Head, Tail, Cut, Paste, Awk, Sed**

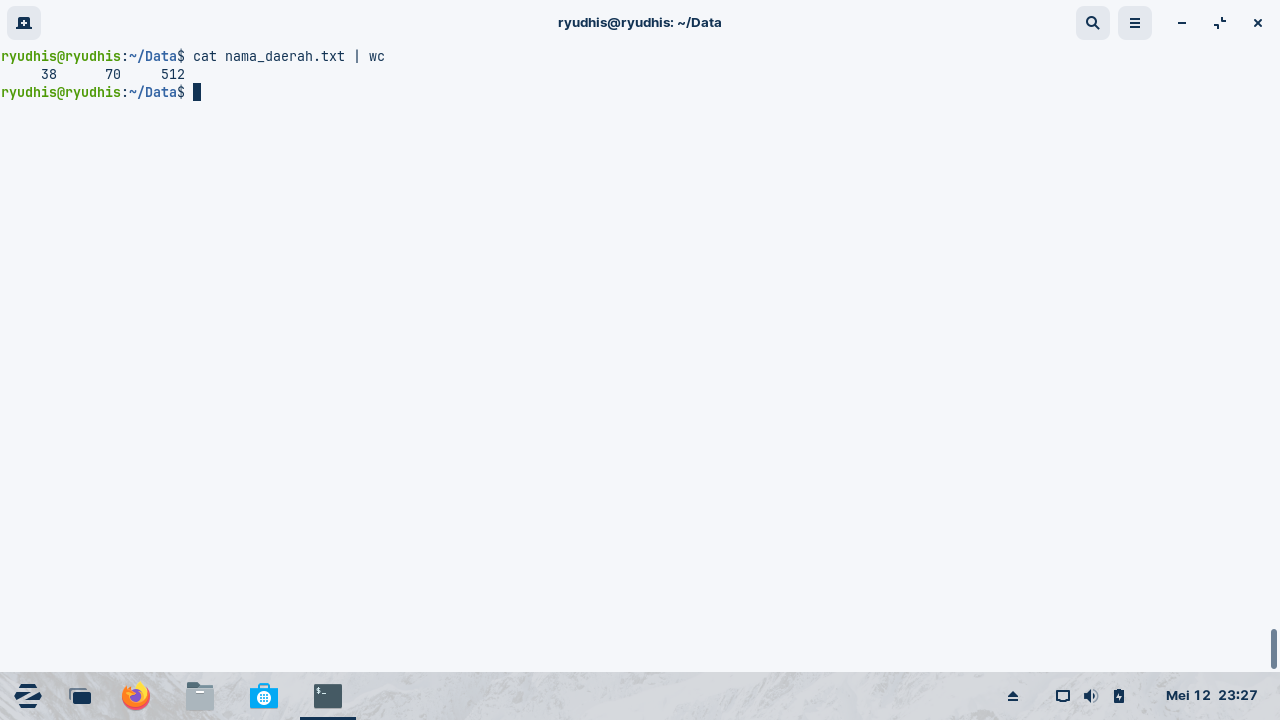
**2.3.1 Grep**

Melakukan pipeline dan filter **Grep** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | grep Sumatera** untuk hanya menampilkan baris string yang mengandung string “Sumatera”.



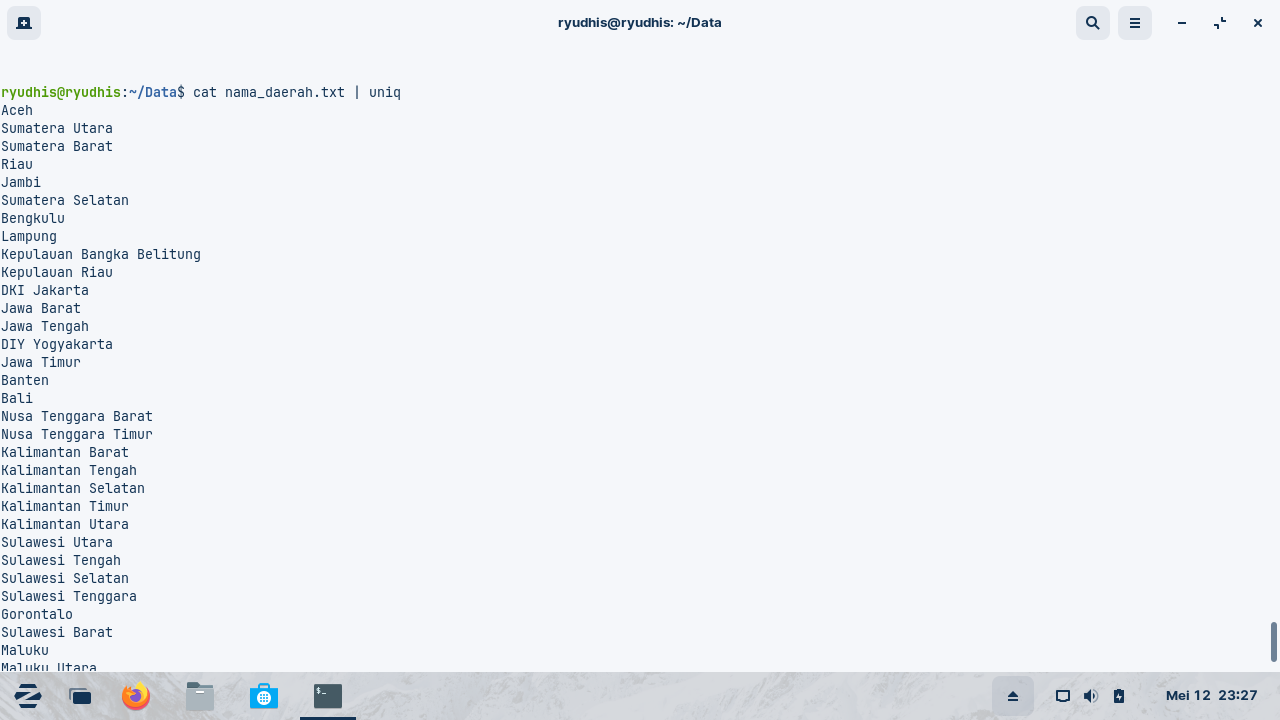
**2.3.2 Wc**

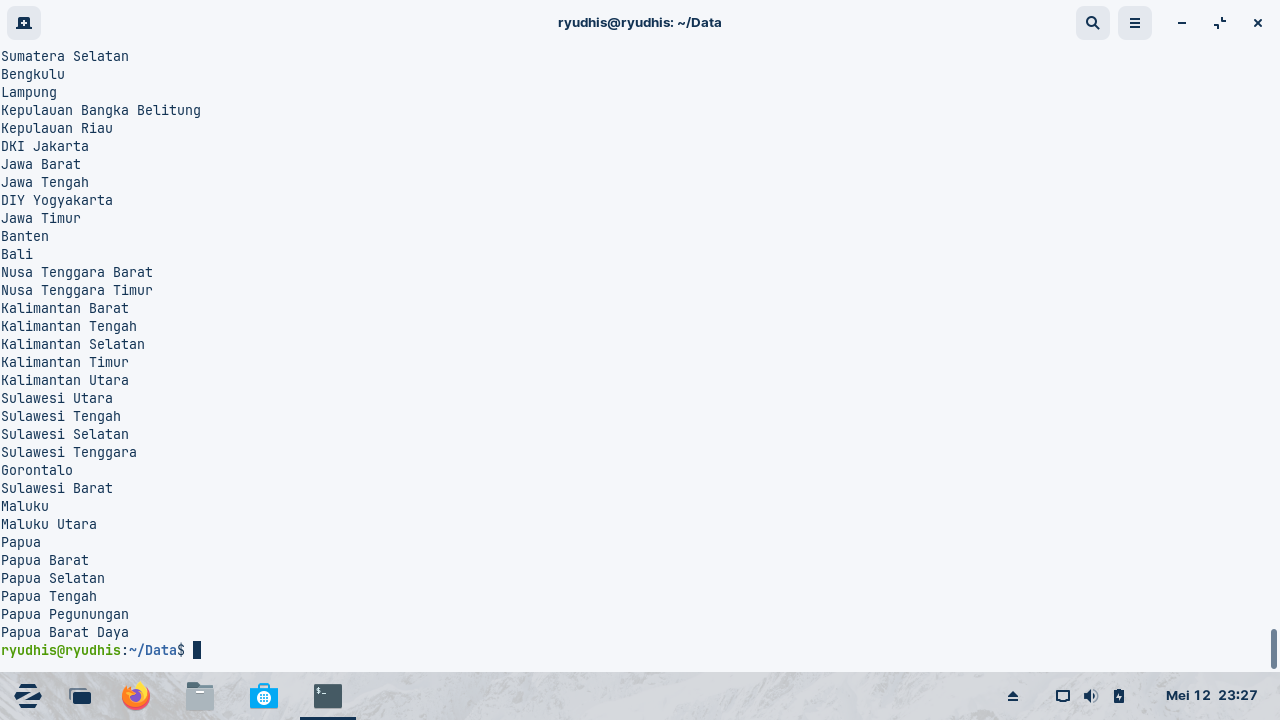
Melakukan pipeline dan filter **Wc** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | wc** untuk menampilkan jumlah baris, jumlah kata, dan jumlah karakter yang terdapat pada file.



**2.3.3 Uniq**

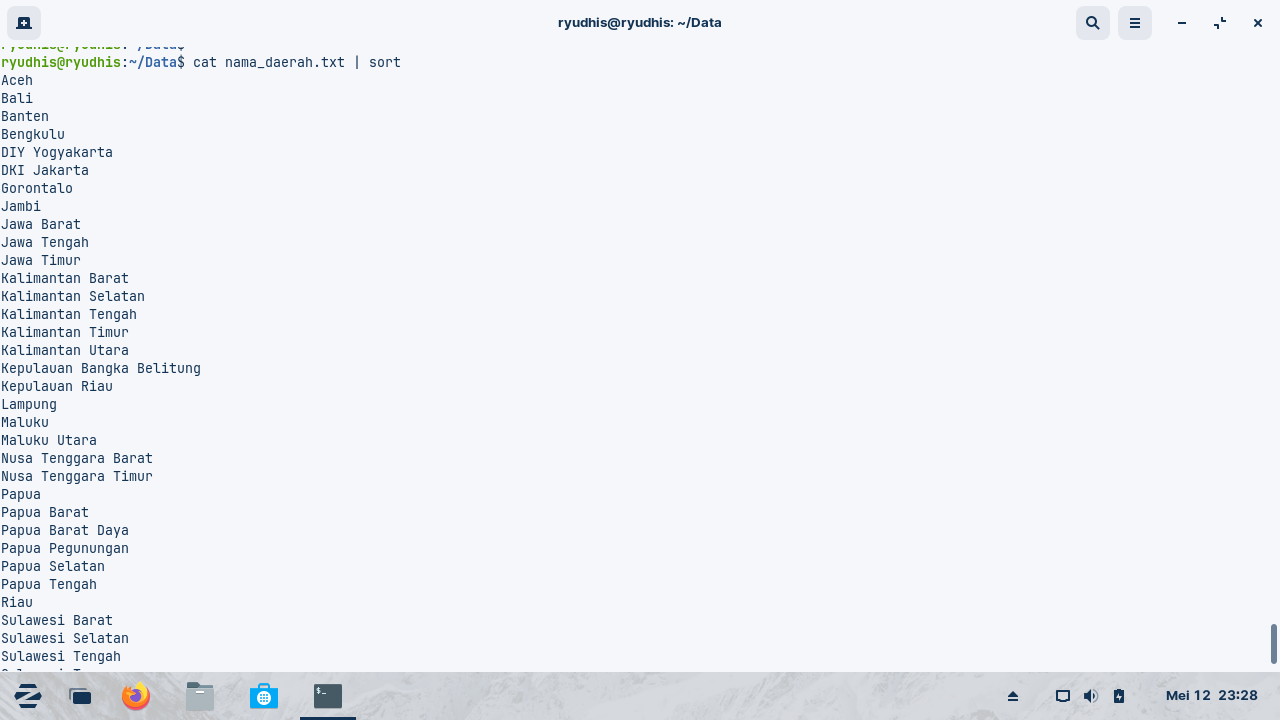
Melakukan pipeline dan filter **Uniq** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | uniq** untuk tidak menampilkan baris duplikasi, namun dalam file nama\_daerah.txt tidak ada baris duplikasi sehingga output filenya sama seperti biasa.

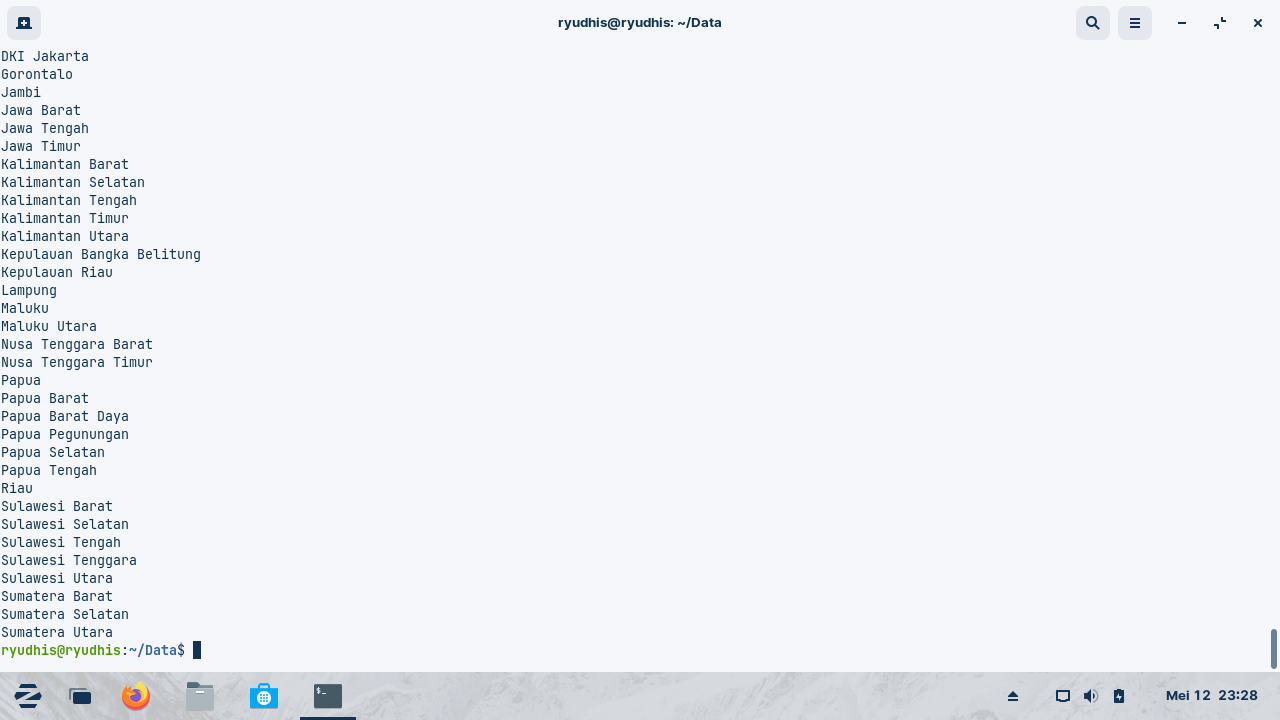




**2.3.4 Sort**

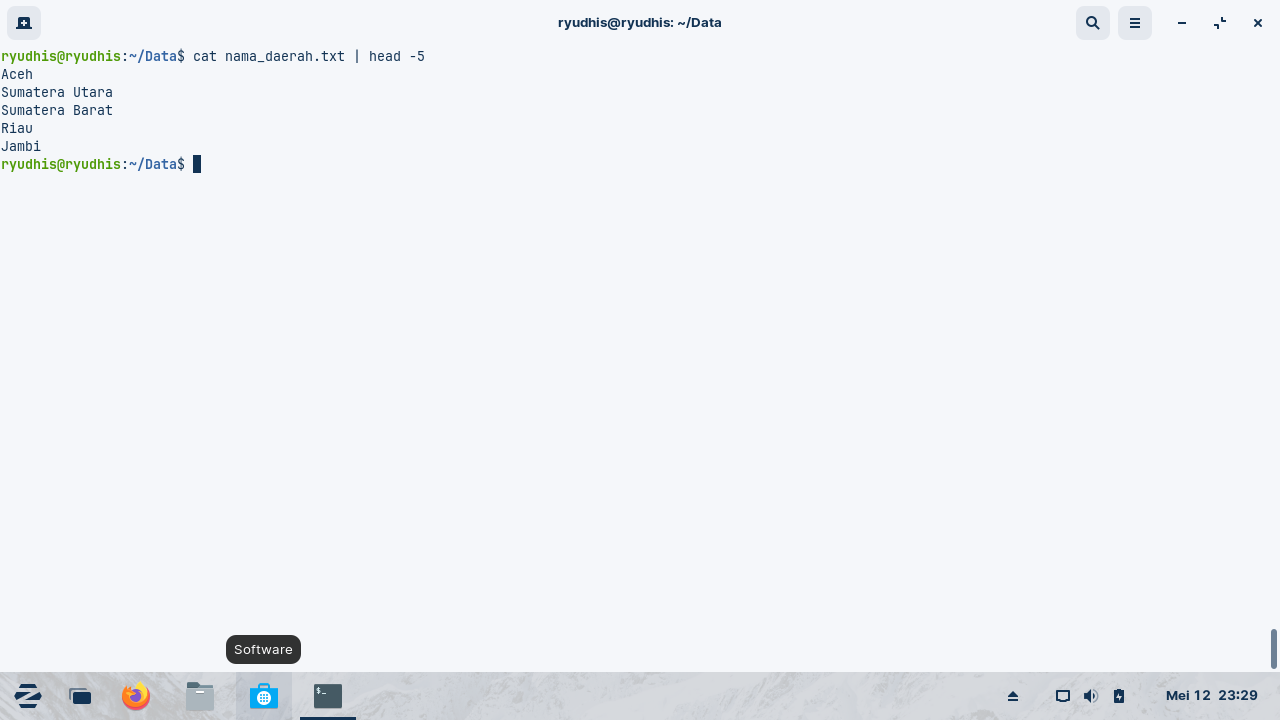
Melakukan pipeline dan filter **Sort** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | sort** untuk mengurutkan baris sesuai abjad menurun (a ke z).





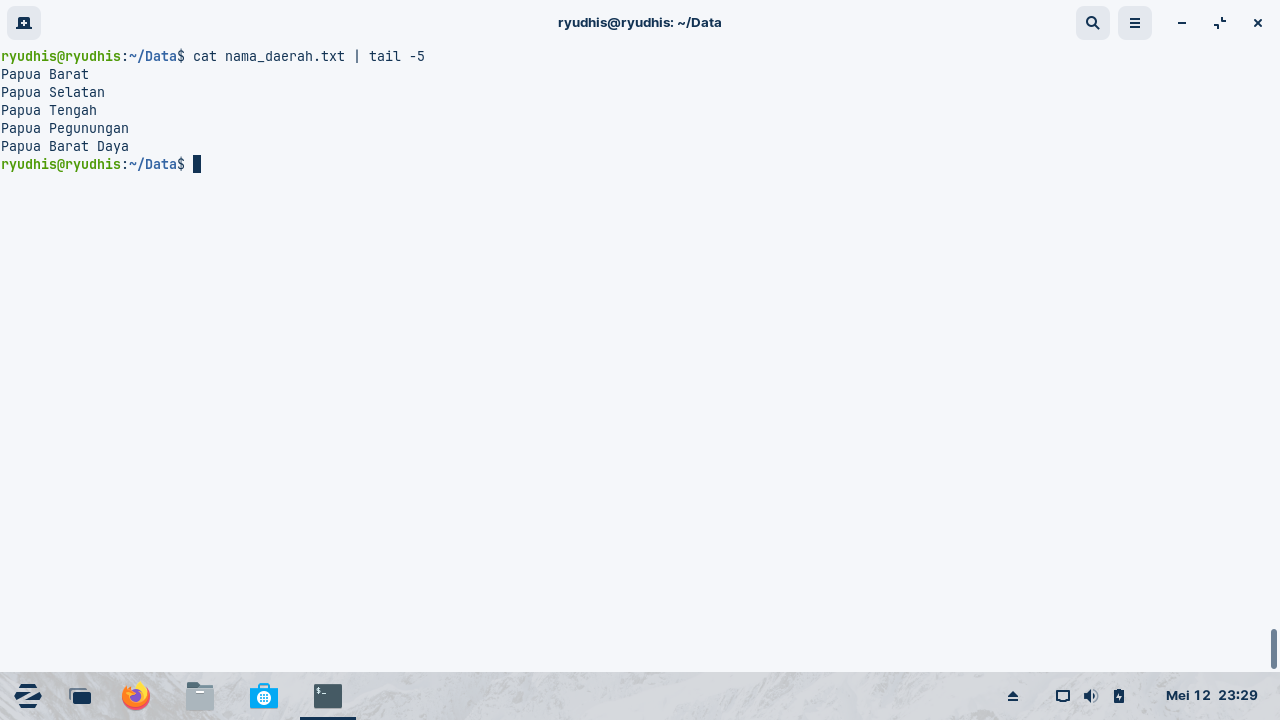
**2.3.5 Head**

Melakukan pipeline dan filter **Head** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | head -5** untuk hanya menampilkan 5 baris awal pada file.



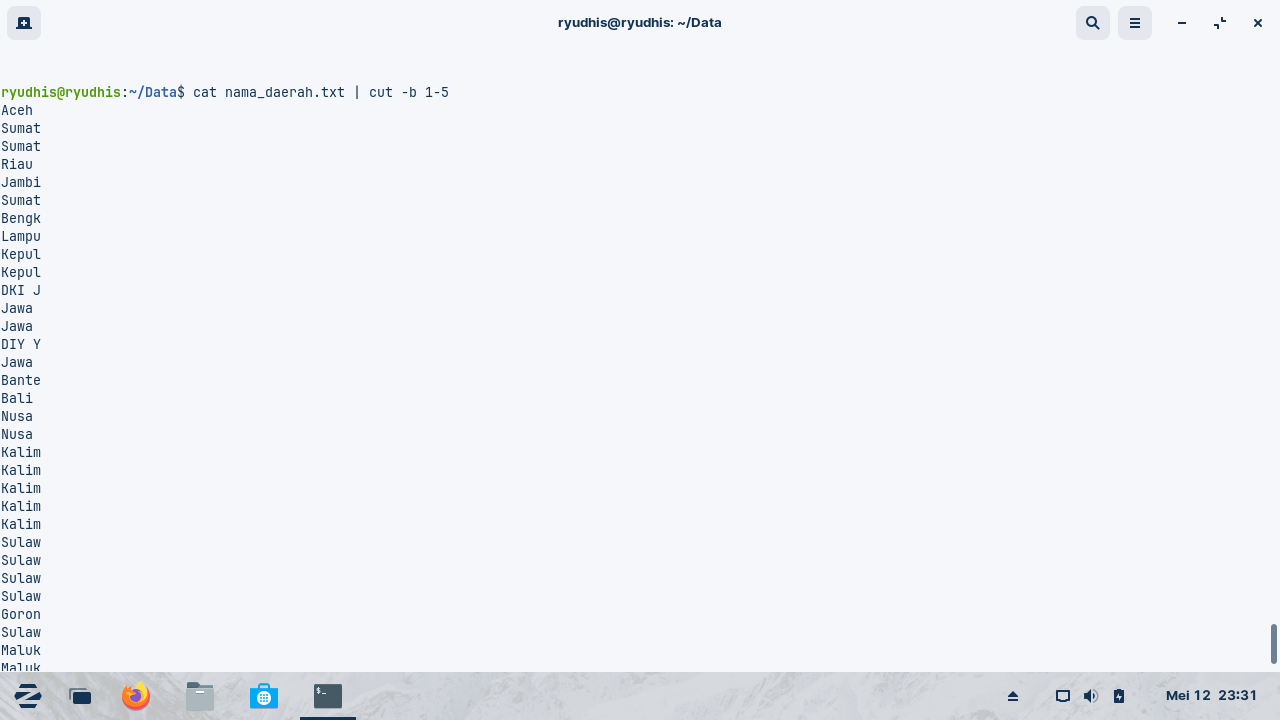
**2.3.6 Tail**

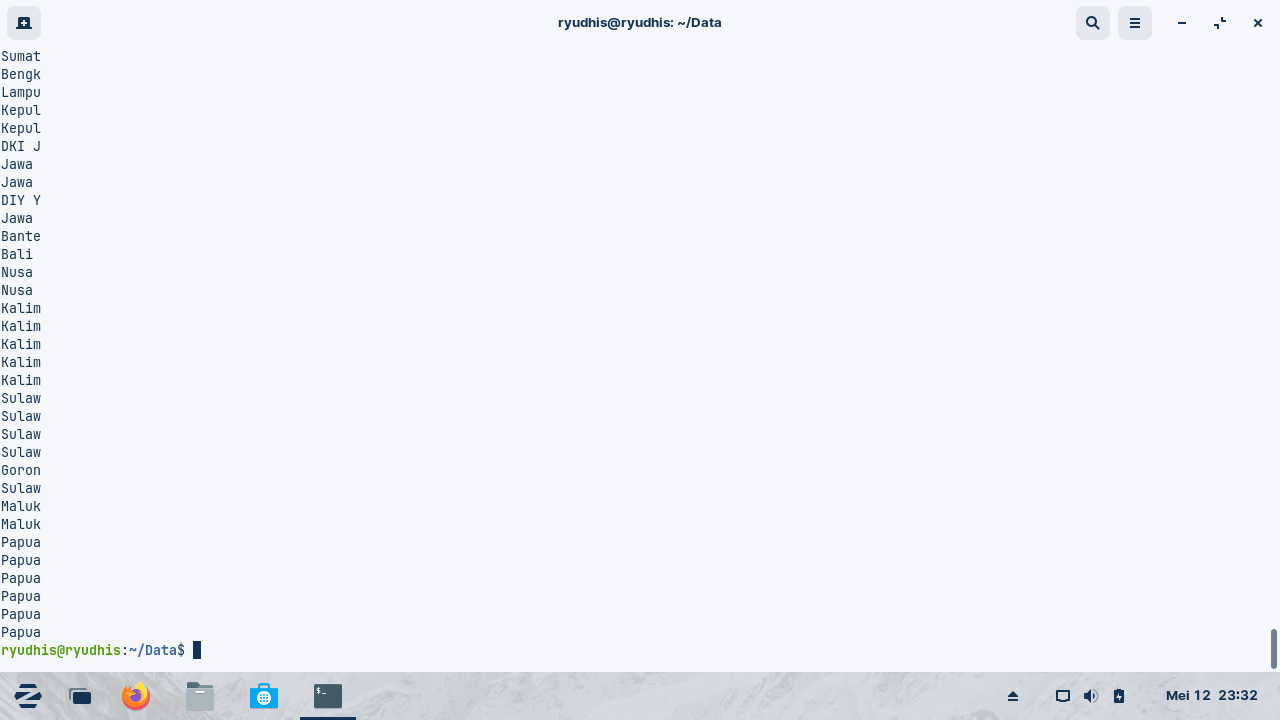
Melakukan pipeline dan filter **Tail** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | tail -5** untuk hanya menampilkan 5 baris akhir pada file.



**2.3.7 Cut**

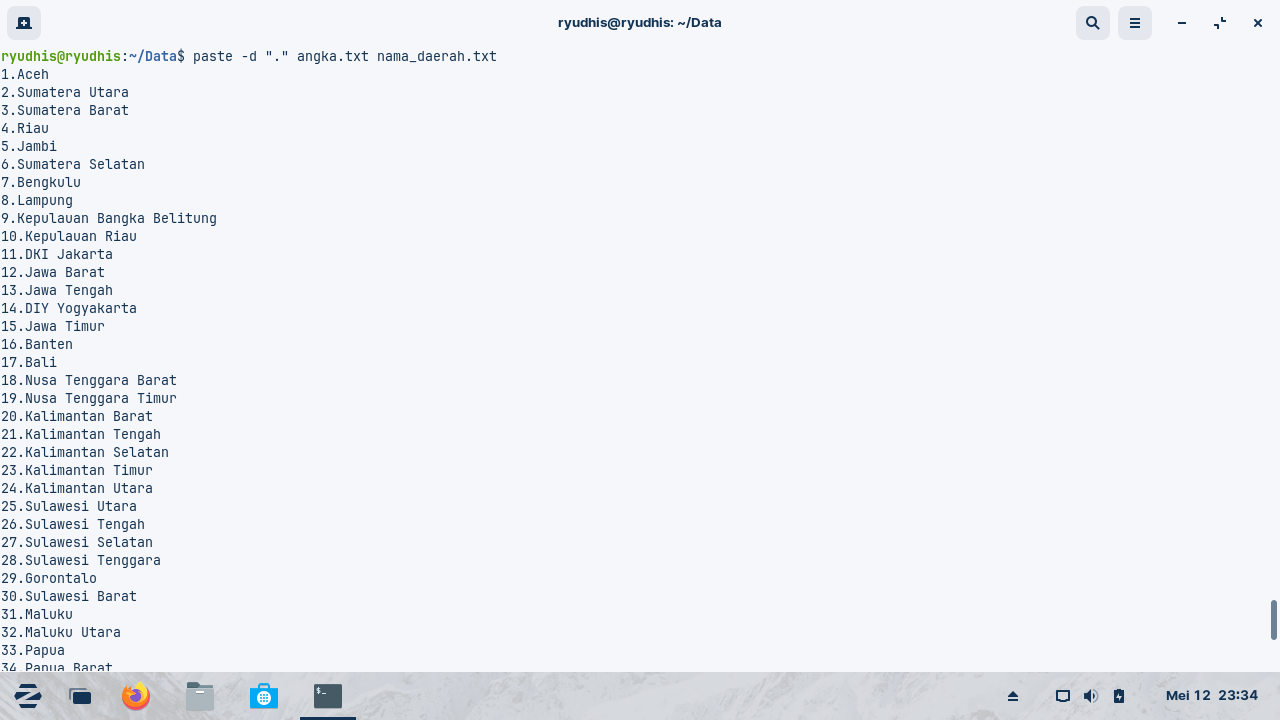
Melakukan pipeline dan filter **Cut** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | cut -b 1-5** untuk hanya menampilkan 5 bit/string pada setiap baris dalam file.

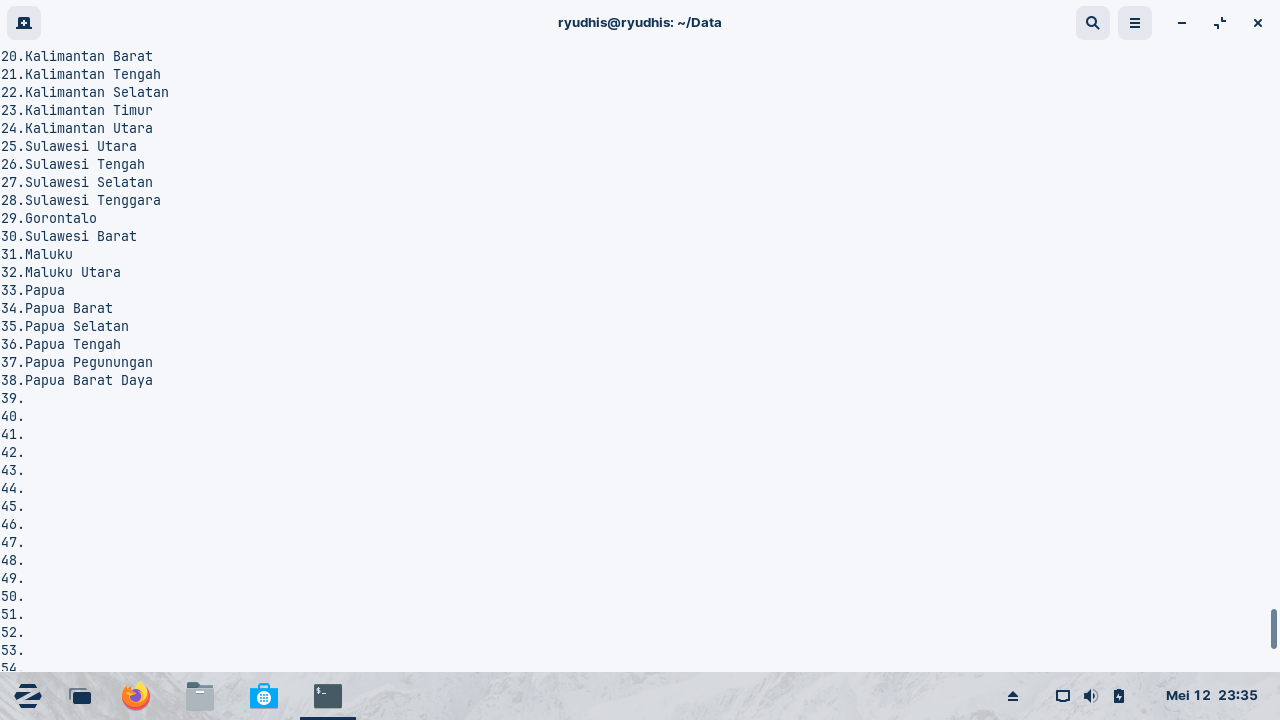




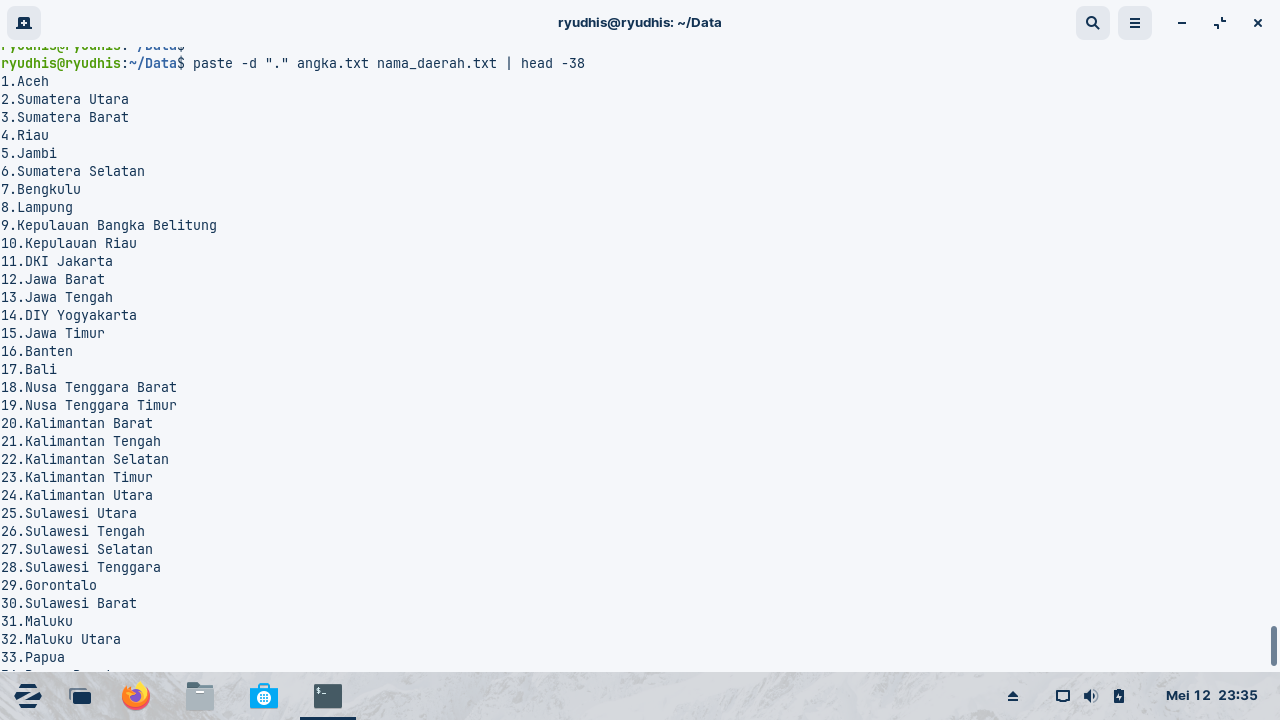
**2.3.8 Paste**

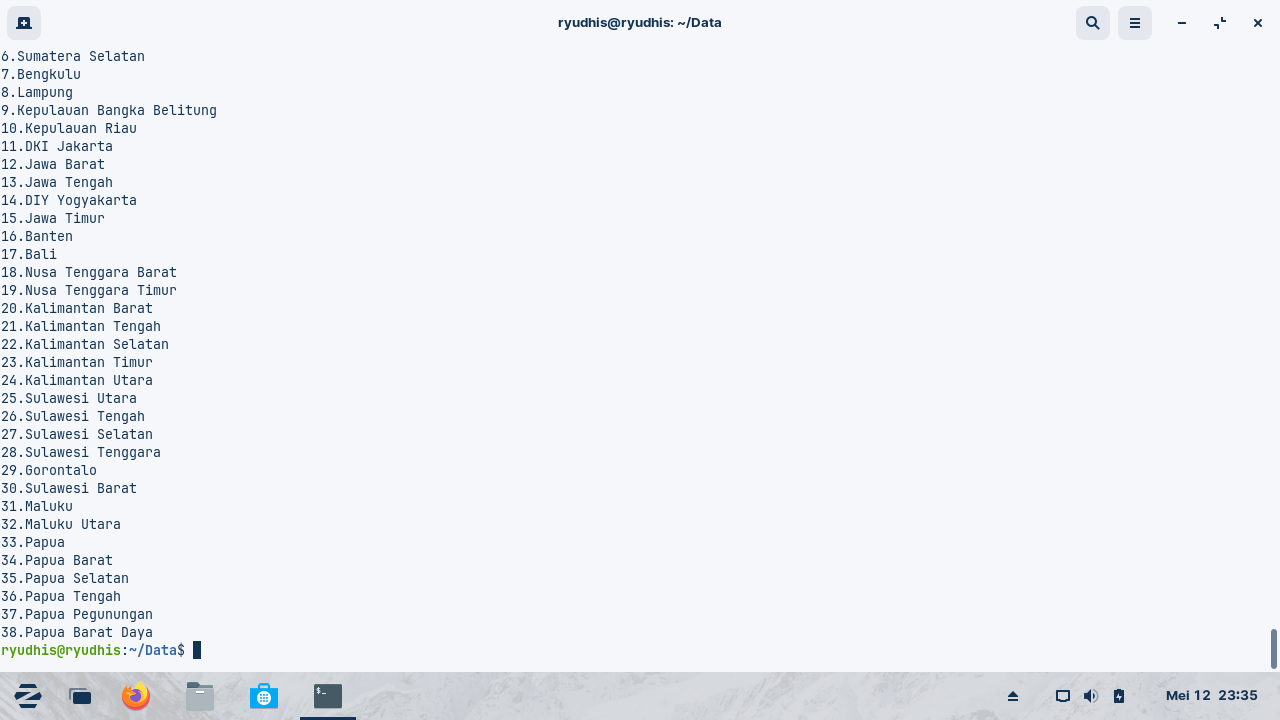
Melakukan pipeline dan filter **Paste** file nama\_daerah.txt pada file angka.txt menggunakan sintaks **paste -d “.” angka.txt nama\_daerah.txt** untuk menempel/menggabungkan isi dari file angka.txt dengan nama\_daerah.txt dengan *delimiter* “.” sehingga setelah file angka.txt teroutputkan akan ada string “.” sebelum file nama\_daerah teroutputkan.





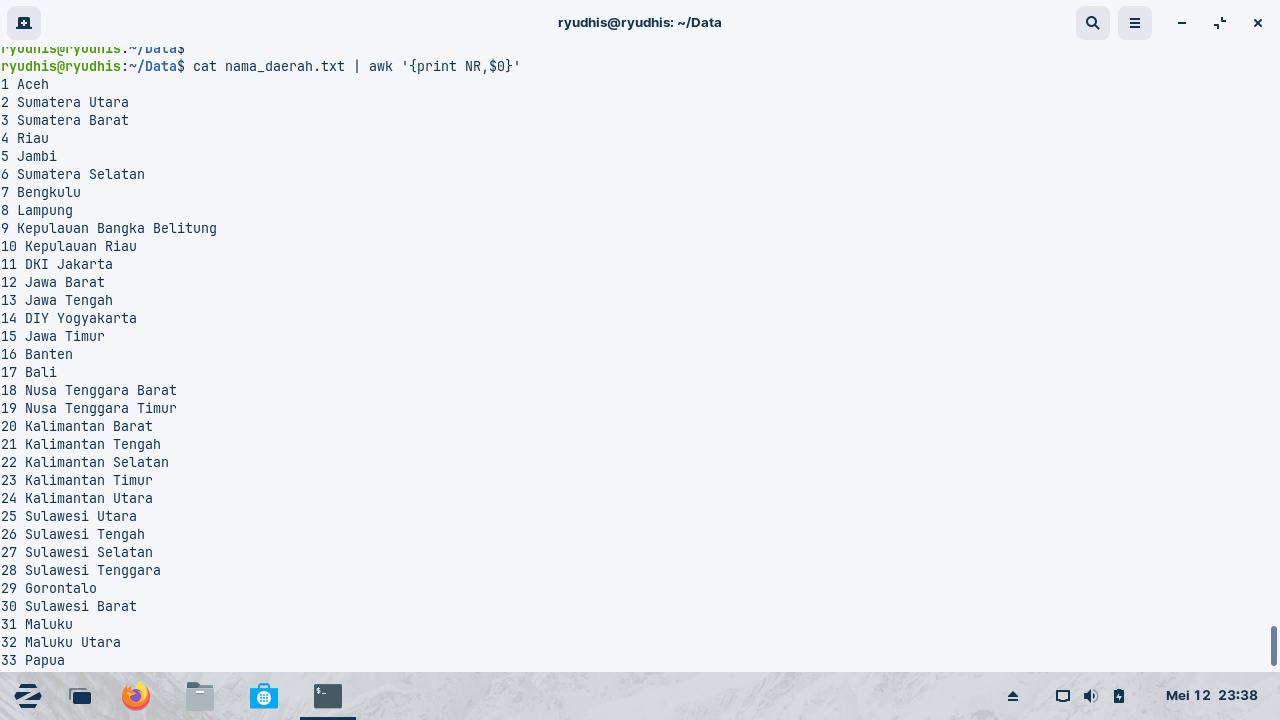
Jika dikombinasikan dengan filter **Head** untuk membatasi outputnya hanya 38 baris awal saja menggunakan sintaks **paste -d “.” angka.txt nama\_daerah.txt** **| head -38**, maka output file akan lebih rapih karena tidak ada angka dari angka.txt yang melebihi baris pada file nama\_daerah.txt.

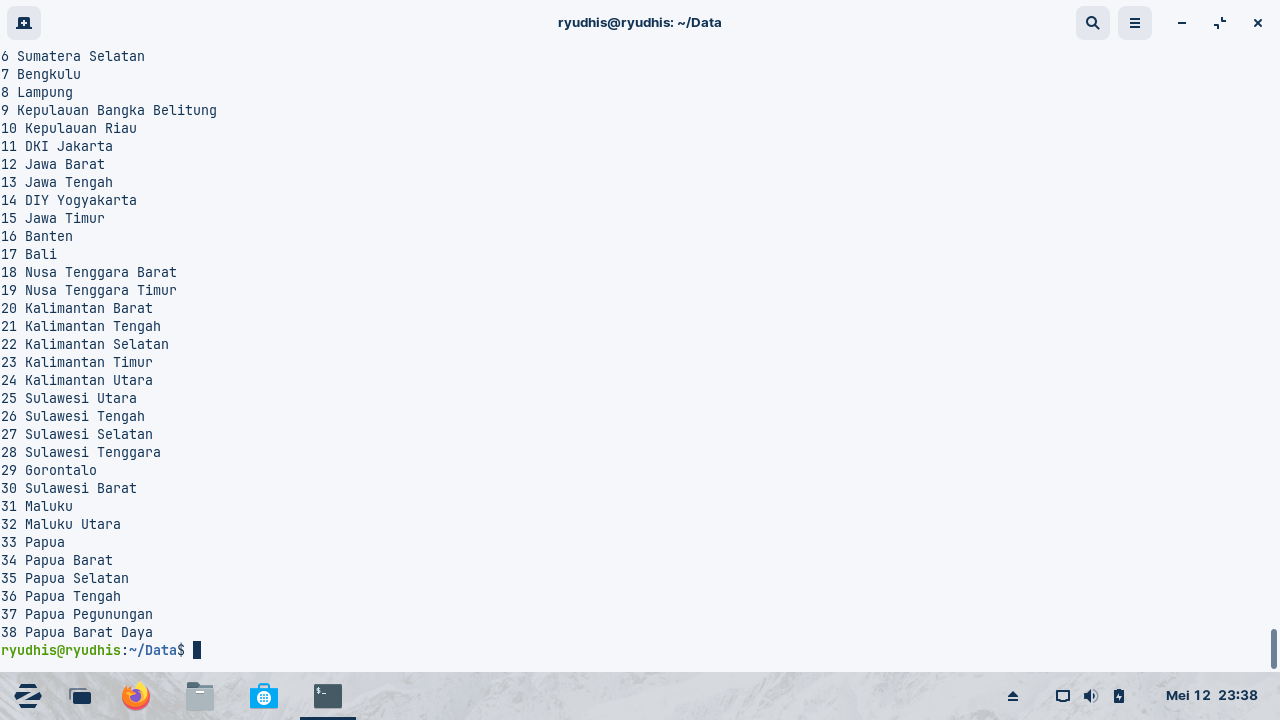




**2.3.9 Awk**

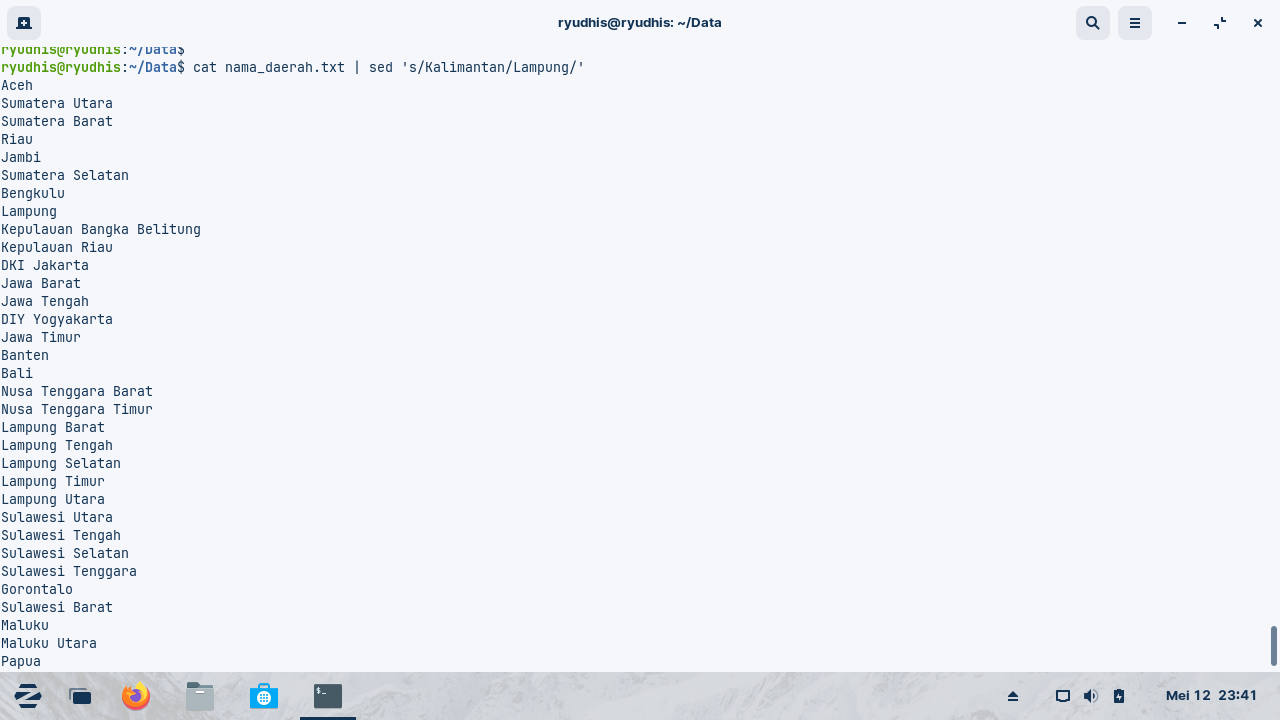
Melakukan pipeline dan filter **Awk** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | awk ‘{print NR,$0}’** untuk menampilkan angka pada tiap baris file nama\_daerah.txt. Sintaks **NR** digunakan untuk menghitung dan mengoutputkan angka pada setiap baris secara berurutan, dan sintaks **$0** untuk mengoutputkan semua baris secara utuh.

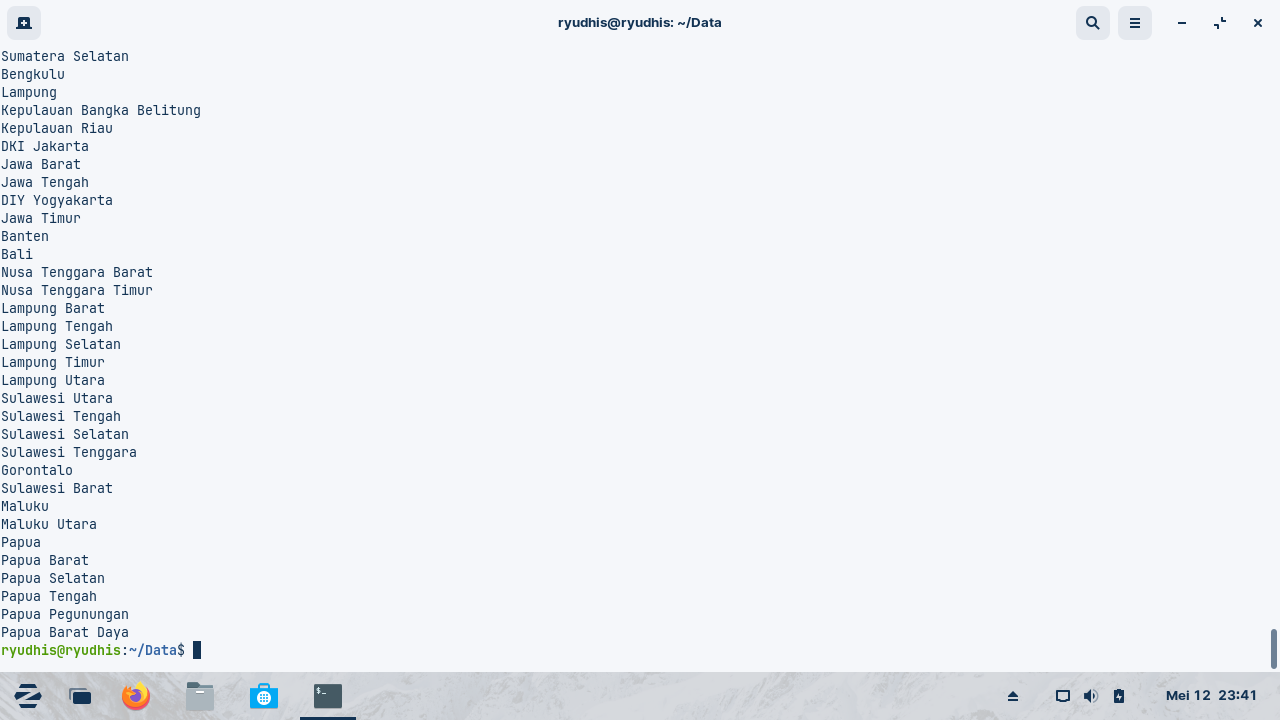




**2.3.10 Sed**

Melakukan pipeline dan filter **Sed** pada file nama\_daerah.txt menggunakan sintaks **cat nama\_daerah.txt | sed ‘s/Kalimantan/Lampung/’** untuk mengganti semua string yang mengandung “Kalimantan” menjadi string “Lampung”.





**BAB III  
KESIMPULAN**

Pada bab ini dijabarkan beberapa kesimpulan terkait kegunaan perintah yang digunakan pada praktikum:

1. Proses input berarti pengambilan data dari sumber eksternal, seperti keyboard atau file. Proses output berarti menampilkan data ke perangkat eksternal, seperti monitor atau file.
2. File descriptor adalah nomor yang digunakan oleh sistem operasi untuk mengidentifikasi dan mengelola file yang dibuka oleh suatu proses. Setiap file yang dibuka oleh proses memiliki file descriptor yang unik. File descriptor 0 (stdin): Merupakan file descriptor standar untuk input. Biasanya terhubung ke keyboard atau sumber input lainnya. File descriptor 1 (stdout): Merupakan file descriptor standar untuk output. Biasanya terhubung ke layar atau perangkat output lainnya. File descriptor 2 (stderr): Merupakan file descriptor standar untuk output error. Biasanya terhubung ke layar atau perangkat output error lainnya.
3. Redirection adalah proses mengalihkan input atau output ke sumber atau tujuan yang berbeda. Operator ">" digunakan untuk mengalihkan output ke file, Operator "<" digunakan untuk mengalihkan input dari file, Operator ">>" digunakan untuk menambahkan output ke file yang sudah ada.
4. Pipeline adalah teknik mengalirkan output dari satu perintah atau program ke input perintah atau program berikutnya secara berurutan. Operator "|" (pipe) digunakan untuk menghubungkan perintah atau program dalam pipeline. Menggunakan pipeline memungkinkan pengguna melakukan pengolahan data yang kompleks dengan menggabungkan beberapa perintah atau program.
5. Filter adalah program yang melakukan transformasi pada input dan menghasilkan output yang dimodifikasi. Filter digunakan dalam kombinasi dengan pipeline untuk melakukan manipulasi data secara berurutan.
6. Filter Grep untuk mencocokkan pola teks, filter Wc untuk menghitung jumlah baris, kata, dan karakter dalam file atau input, filter Uniq untuk menghilangkan baris duplikat berturut-turut dari input, filter sort untuk mengurutkan baris dalam input, filter head untuk mengambil sejumlah baris terawal dari input, filter tail untuk mengambil sejumlah baris terakhir dari input, filter cut untuk memotong kolom tertentu dari input berdasarkan delimiter, filter paste untuk menggabungkan baris-baris dari beberapa file secara horizontal, filter awk untuk melakukan pemrosesan teks yang kompleks berdasarkan pola dan kolom, filter sed untuk melakukan penggantian teks atau transformasi pada input.