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

**Disusun oleh:**

**Pannes Diba Sabila | 121140117**

**Kelas RD | Cluster RD**



**Program Studi Teknik Informatika**

**Jurusan Teknologi Produksi dan Industri**

**Institut Teknologi Sumatera**

**Lampung Selatan**

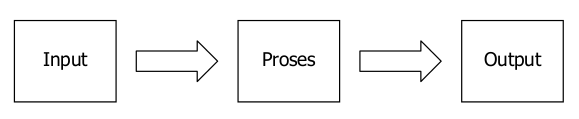
**2023**

**BAB I  
TEORI DASAR**

**1.1 Teori Dasar 1**

**Proses Input dan Output**

Sebuah proses memerlukan Input dan Output



Instruksi(Command) yang diberikan pada linux melalui Shell disebut sebagai eksekusi

program yang selanjutnya disebut proses.

Setiap kali instruksi diberikan. Maka linux kernel akan menciptakan sebuah proses

dengan memberikan nomor PID (proses identity).

Proses dalam linux selalu membutuhkan input dan menghasilkan suatu output

Dalam konteks Linux Input/output adalah:

1. Keyboard (input)

2. Layar (Output)

3. Files

4. Struktur data kernel

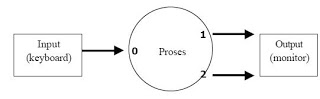
5. Peralatan I/O lainnya (misalnya Network)

1. **File descriptor**

Linux berkomunikasi dengan file melalui \*file descriptor\* yang dipresentasikan melalui angka yang dimulai dari 0,1,2 dan seterusnya. tiga buah file descriptor standar yang lalu diciptakan oleh proses adalah:

1. 0 = keyboard (standar input)

2. 1 = layer (standar output)

3. 2 = layer (standar error)

1. **Pembelokan (redirection)**

Pembelokan dilakukan untuk standard input, output dan error, yaitu untuk mengalihkan file descriptor dari 0,1 dan 2. Simbol untuk pembelokan adalah:

0< atau < pengganti standard input

1< atau > pengganti standard output

1. **Pipa (pipeline)**

Mekanisme pipa digunakan sebagai alat komunikasi antar proses.

Input --> Proses1 --> Output=Input --> Proses2 --> Output

Proses1 menghasilkan output yang selanjutnya digunakan sebagai input oleh

Proses2. Hubungkan output input ini dinamakan pipa, yang menghubungkan

proses1 dengan proses2 dan dinyatakan dengan symbol"|".

Proses1 | Proses2

1. **filter**

Filter adalah utilitas linux yang dapat memproses standard input(dari Keyboard) dan menampilkan hasilnya pada standard output (layer). Contoh filter adalah \*cat, sort, grep, pr, head, tail, paste\* dan lainya.

Pada sebuah rangkaian pipa:

P1 | P2 | P3 ... | $$Pn-1$$ | $$Pn$$

Maka P2 sampai dengan $$Pn-1$$ mutlak harus utilitas linux yang berfungsi sebagai filter. P1 (awal) dan Pn (terakhir) boleh tidak filter. Utilitas yang bukan filter misalnya who, ls, ps, lp, lpr, mail dan lainnya. Beberapa perintah linux yang digunakan untuk proses penyaringan antara lain:

1. Perintah \*Grep\*

Digunakan untuk menyaring masukannya dan menampilkan baris-baris yang hanya mengandung pola yang ditentukan. Pola ini disebut \*regular expression\*.

1. Perintah \*Wc\*

Digunakan untuk menghitung jumlah balis, kata dan karakter dari baris-baris masukan yang diberikan kepadanya. Untuk mengetahui berapa Luris gunakan option—I, untuk mengetahui berapa kata, gunakan option w dan untuk mengetahui berapa karakter, gunakan option —c. Jika salah satu ption tidak digunakan, maka tampilannya adalah jumlah baris, jumlah kata dan jumlah karakter.

1. Perintah \*sort\*

Digunakan untuk mengurutkan masukannya berdasarkan urutan nomor ASCII dari karakter.

1. Perintah \*cut\*

Digunakan untuk mengambil kolom tertentu dari baris-baris masukannya, yang ditentukan pada option —c.

1. Perintah \*uniq\*

Digunakan untuk menghilangkan baris-baris batuan yang mengalami duplikasi, biasanya digabungkan dalam pipeline dengan sort.

**1.2 Teori Dasar 2**

**Pengalihan masukan dan keluaran**

Pengalihan input dan output pada Ubuntu dapat dilakukan dengan menggunakan karakter khusus pada perintah di terminal. Beberapa karakter khusus yang umum digunakan adalah:

A. > : Digunakan untuk mengalihkan output ke file. Contoh: ls > file.txt

B. >> : Digunakan untuk menambahkan output ke file yang sudah ada. Contoh: ls >> file.txt

C. < : Digunakan untuk mengambil input dari file. Contoh: sort <file.txt

D. | : Digunakan untuk mengalihkan keluaran dari satu perintah ke perintah lain. Contoh: ls -l | grep ".txt"

**BAB II  
PEMBAHASAN DAN ANALISIS**

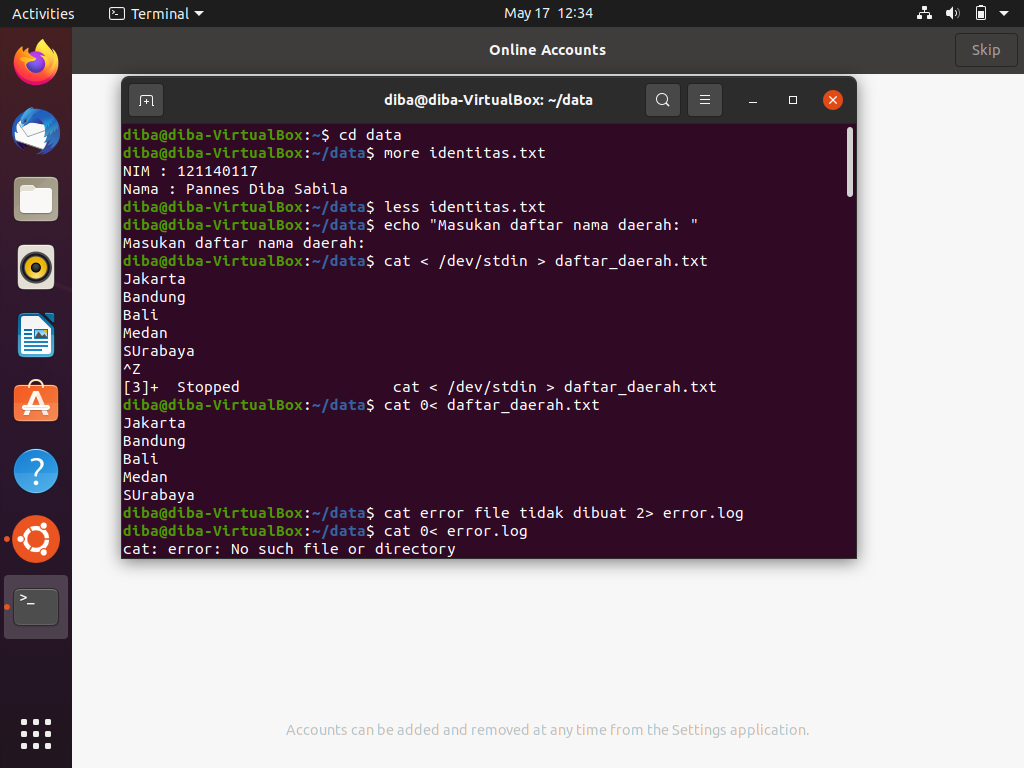
**2.1 Latihan Pertama**

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

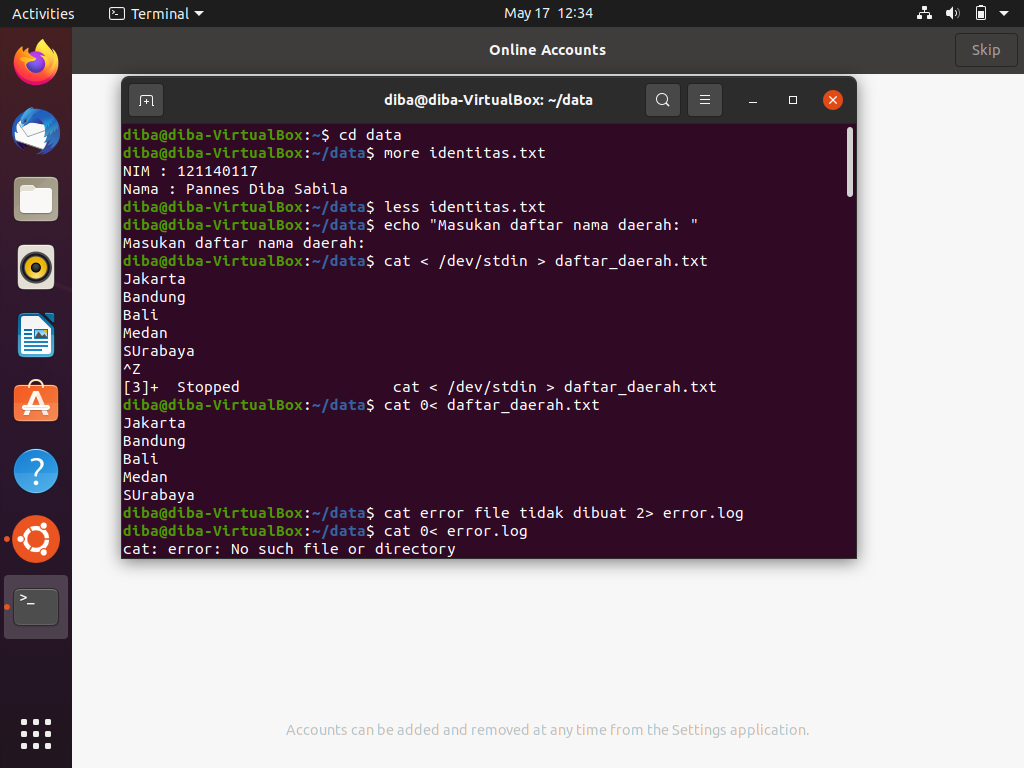
**2.1.1 Langkah Pertama**

Perbedaan utama antara perintah more dan less terletak pada cara mereka menampilkan teks pada layar. Ketika menggunakan lebih banyak perintah, teks menampilkan satu halaman pada satu waktu dan pengguna harus menekan tombol spasi untuk melihat halaman berikutnya. Di sisi lain, perintah kurang memberikan kemampuan kepada pengguna untuk menampilkan teks secara interaktif dan lebih fleksibel, seperti melakukan scroll atau pencarian.

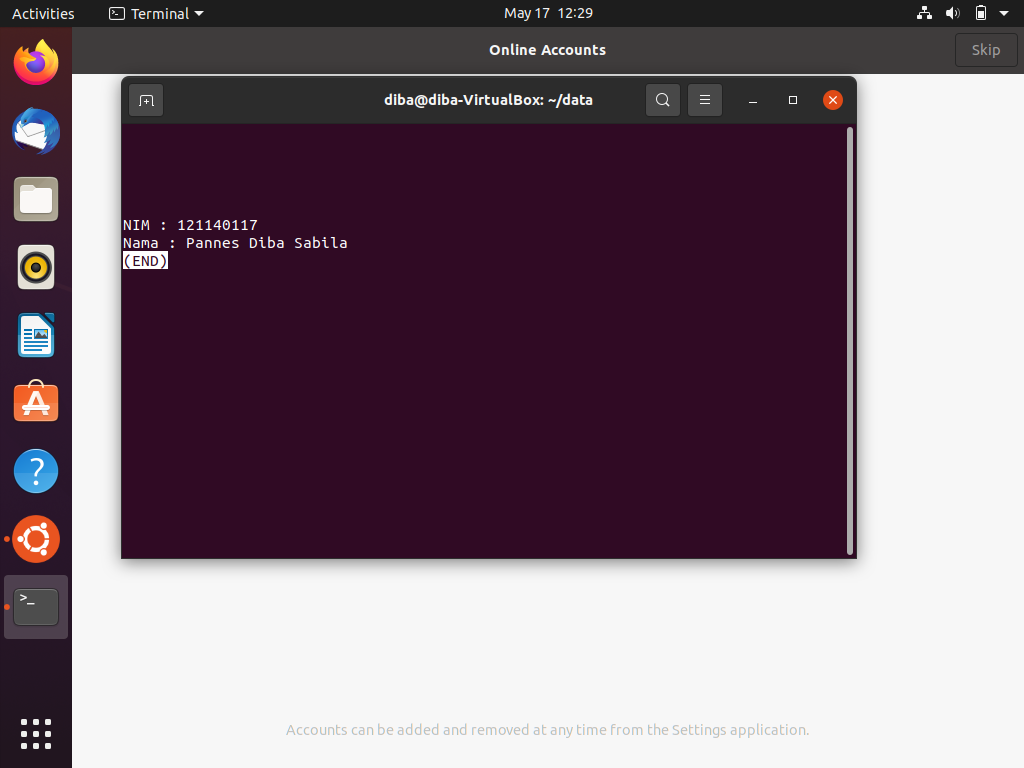
Untuk menggunakan perintah more di terminal, kita dapat mengetikkan perintah "more" diikuti dengan nama file yang ingin ditampilkan. Contoh penggunaan perintah ini dapat dilihat di bawah ini:



disini saya menggunakan perintah less di terminal untuk menampilkan isi file "identitas.txt" satu halaman dalam satu waktu. Program akan menunggu hingga kita menekan tombol spasi untuk melihat halaman berikutnya. Jika file "identitas.txt" hanya memiliki satu halaman, program akan selesai setelah itu. Untuk menggunakan perintah less, ketikkan "less" diikuti dengan nama file yang ingin ditampilkan, seperti contoh di bawah ini:



Sebelum menghentikan proses dengan menggunakan perintah stop/ctrl+c, tampilan akan menampilkan isi file "nama\_file.txt" dan memberikan kemampuan untuk berinteraksi dengan teks menggunakan fitur-fitur seperti scroll, pencarian, dan sebagainya. Kita dapat menekan tombol spasi untuk melihat halaman berikutnya atau menggunakan tombol / untuk mencari kata tertentu.



**2.2 Latihan Kedua**

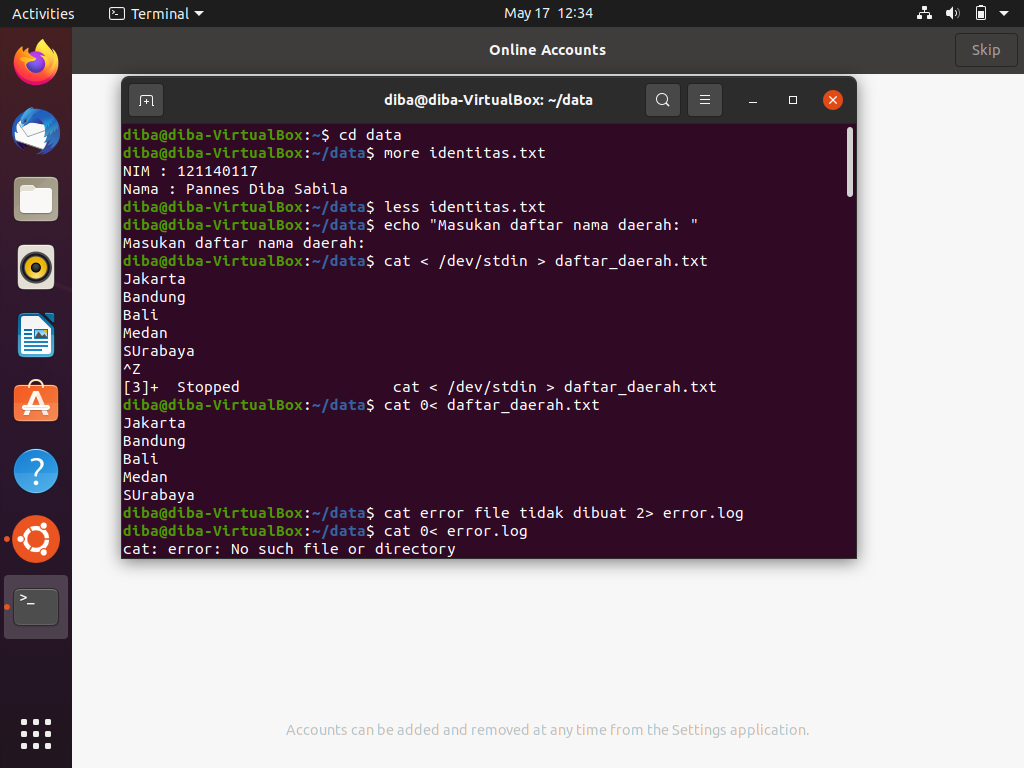
Buatlah sebuah file yang berisikan daftar nama daerah dengan menggunakan file

descriptor dan redirection dengan mengimplementasikan penggunaan standar

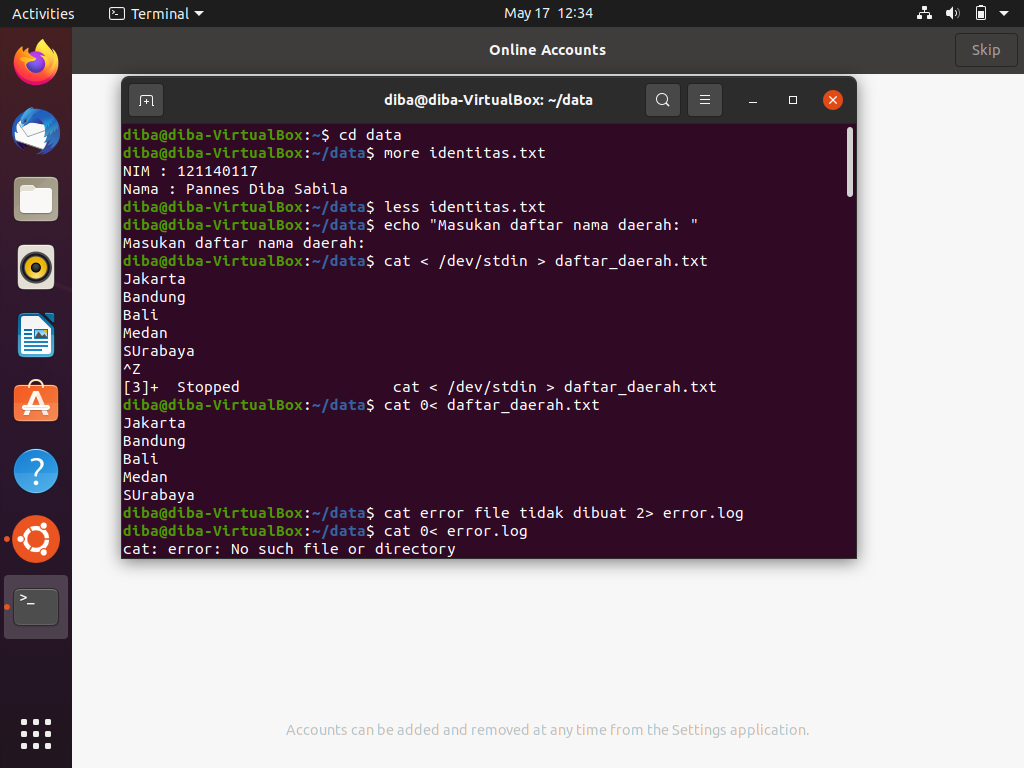
input, standar output dan standar error.

**2.2.1 Langkah Pertama**

Membuat file nama daerah

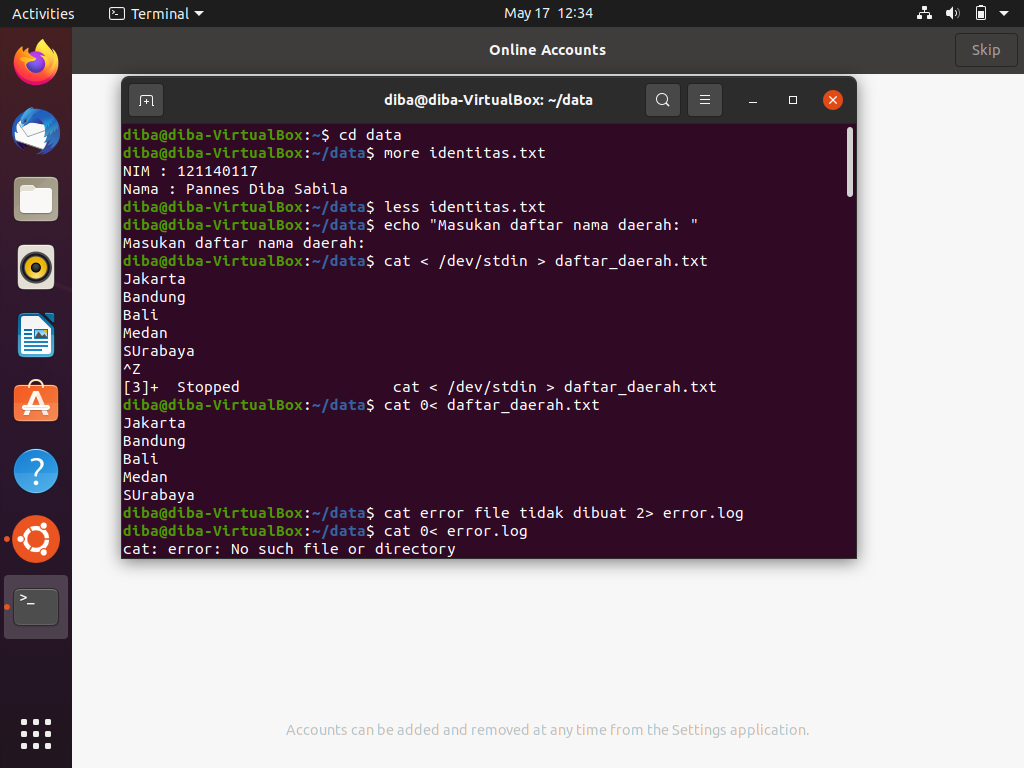


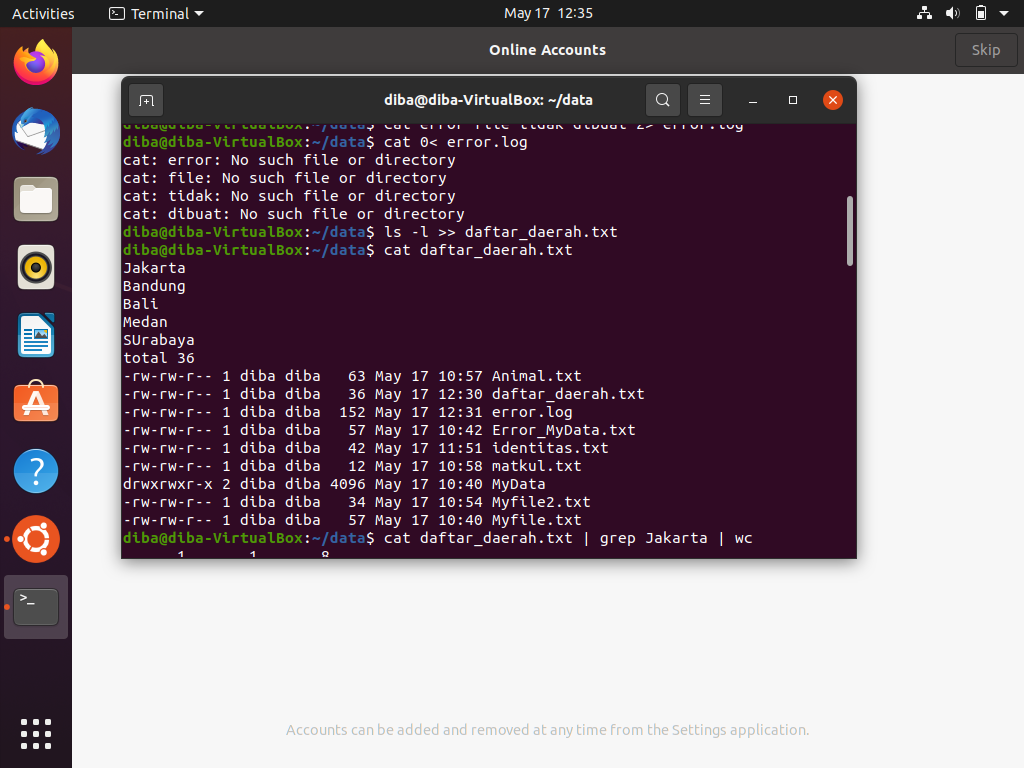
**2.2.2 Langkah Kedua**  
 Melakukan pembelokan input pada file daftar\_daerah.txt



**2.2.3 Langkah Ketiga**

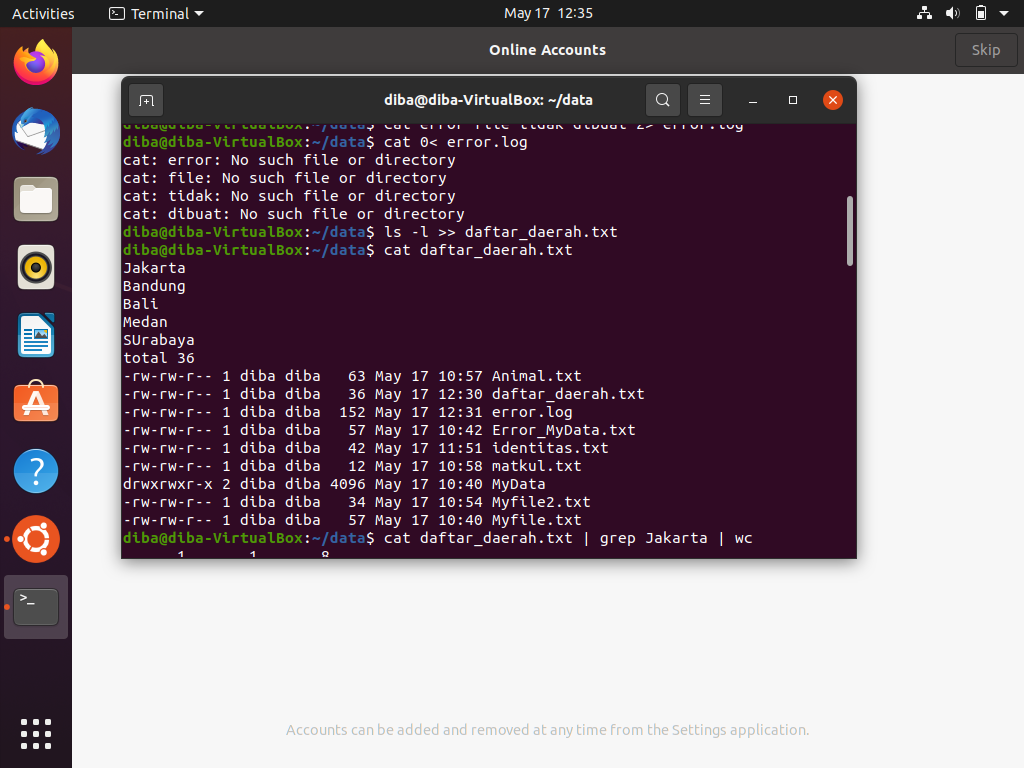
Merubah masukan pada error.log.





**2.2.4 Langkah Keempat**

Melakukan pembelokan, yaitu menambahkan keluaran -l ke dalam file daftar\_daerah.txt.



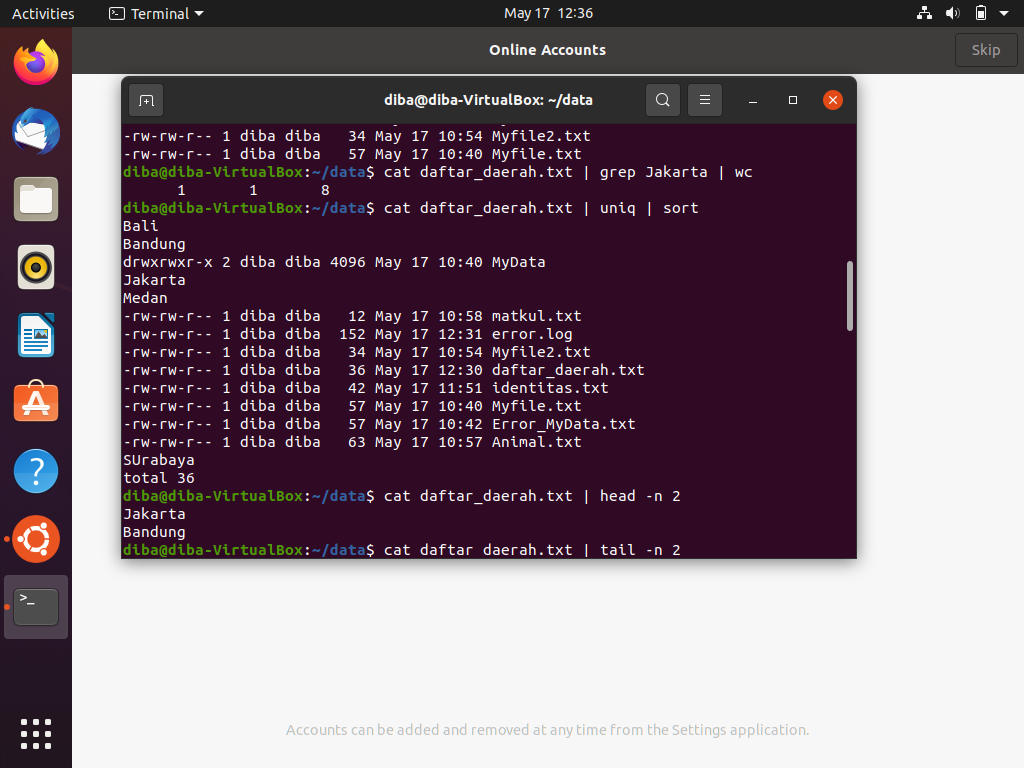
**3.3 Latihan Ketiga**

Tampilkan isi file tersebut dengan menggunakan implementasi dari pipeline dan

filter yaitu perintah

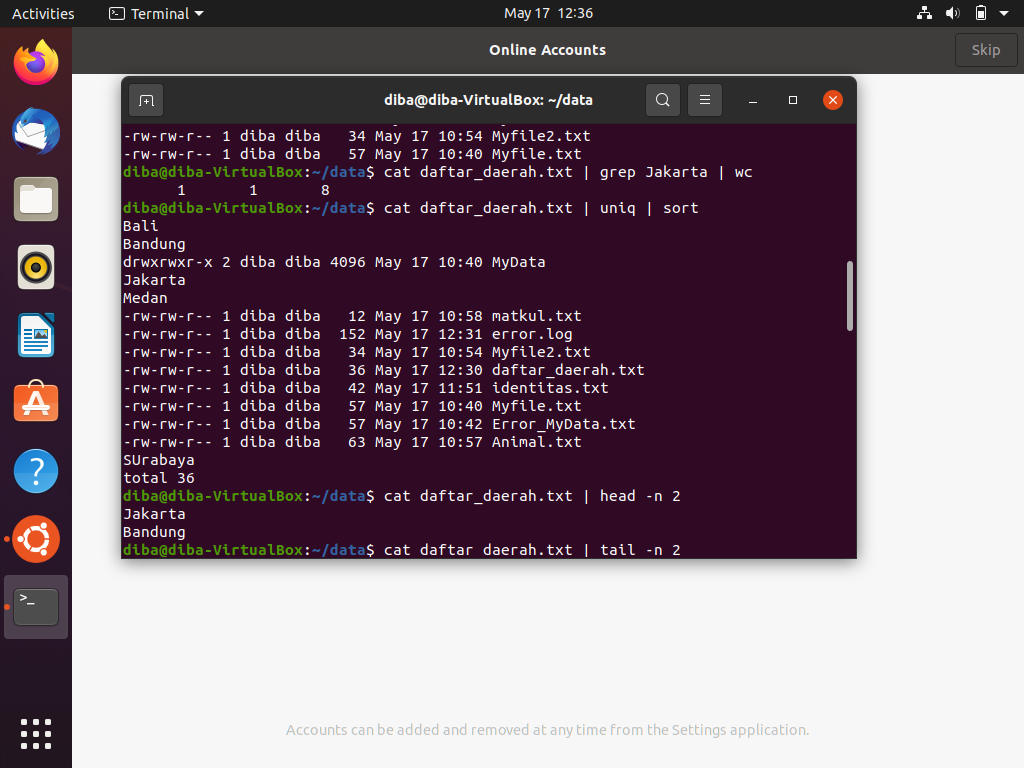
**3.3.1 Langkah Pertama (Grep)**

**3.3.2 Langkah Kedua (Wc)**

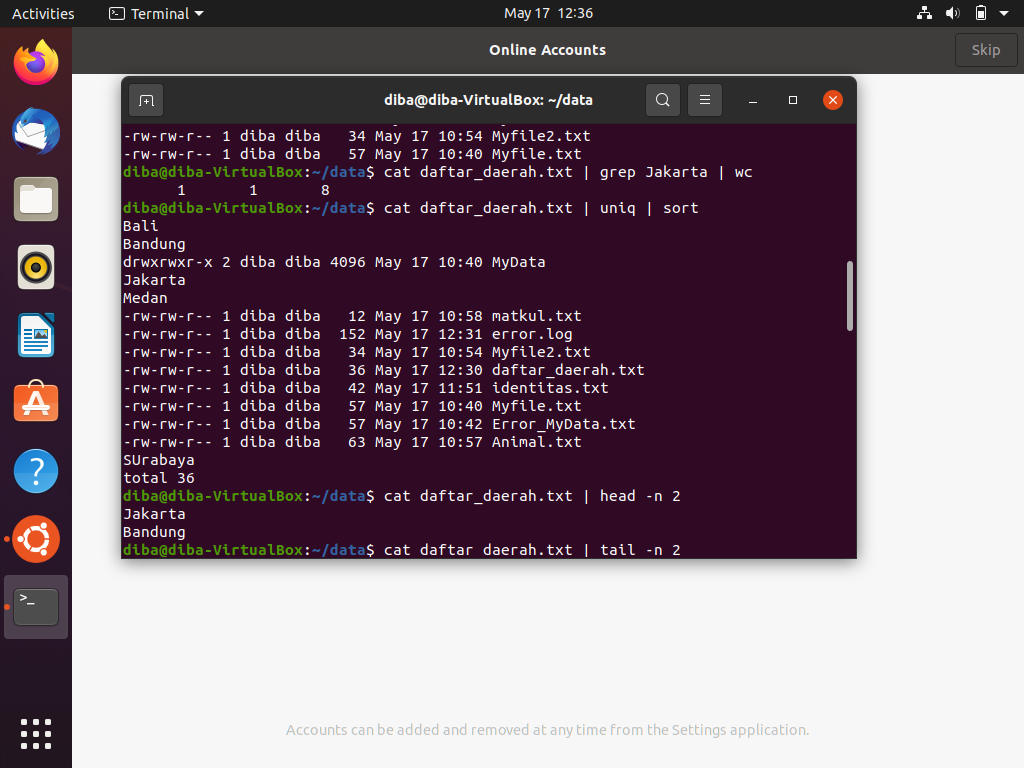


**3.3.3 Langkah Ketiga (Uniq)**

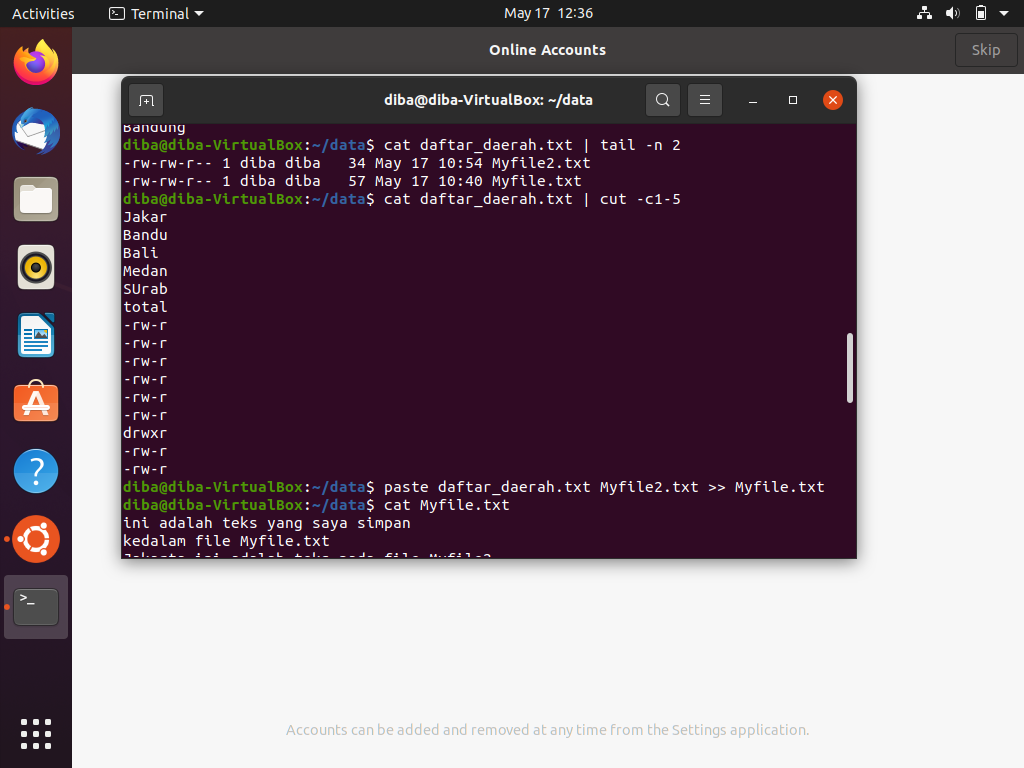
**3.3.4 Langkah Keempat (Sort)**



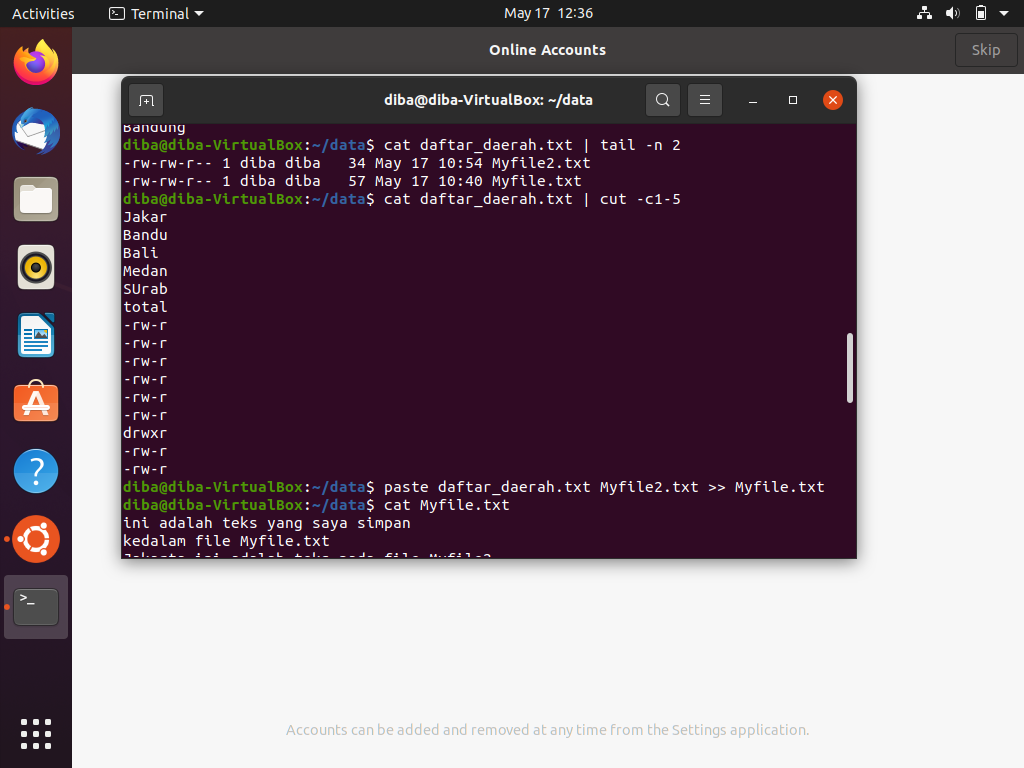
**3.3.5 Langkah Kelima (Head)**



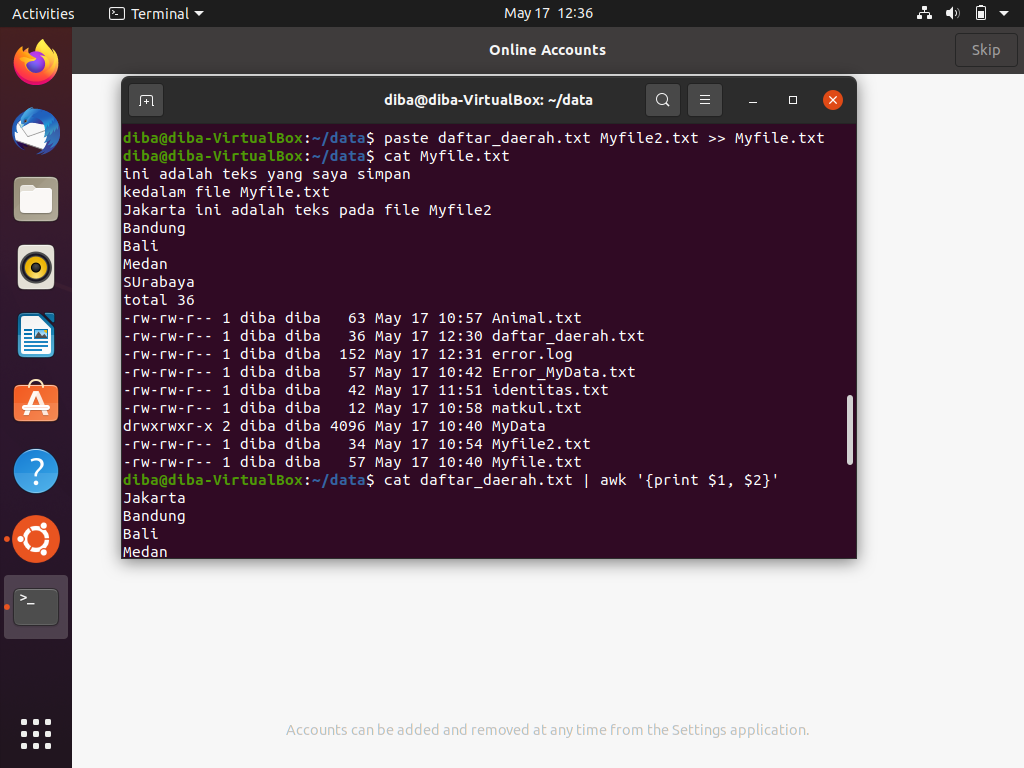
**3.3.6 Langkah Keenam (Tail)**



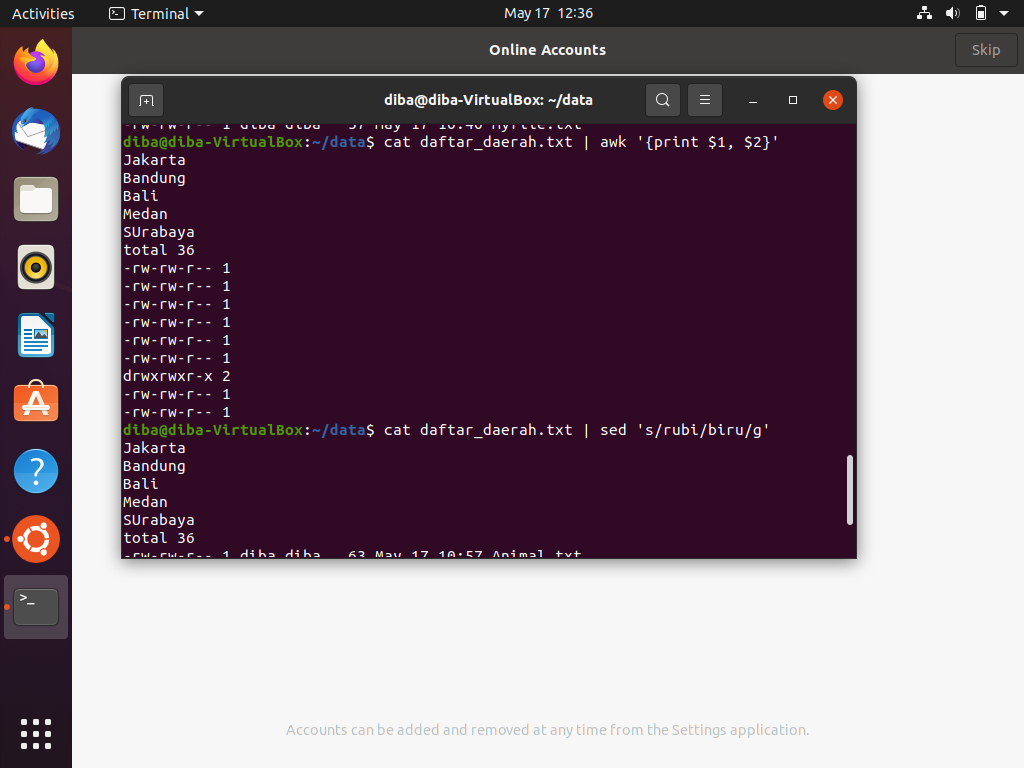
**3.3.7 Langkah Ketujuh (Cut)**



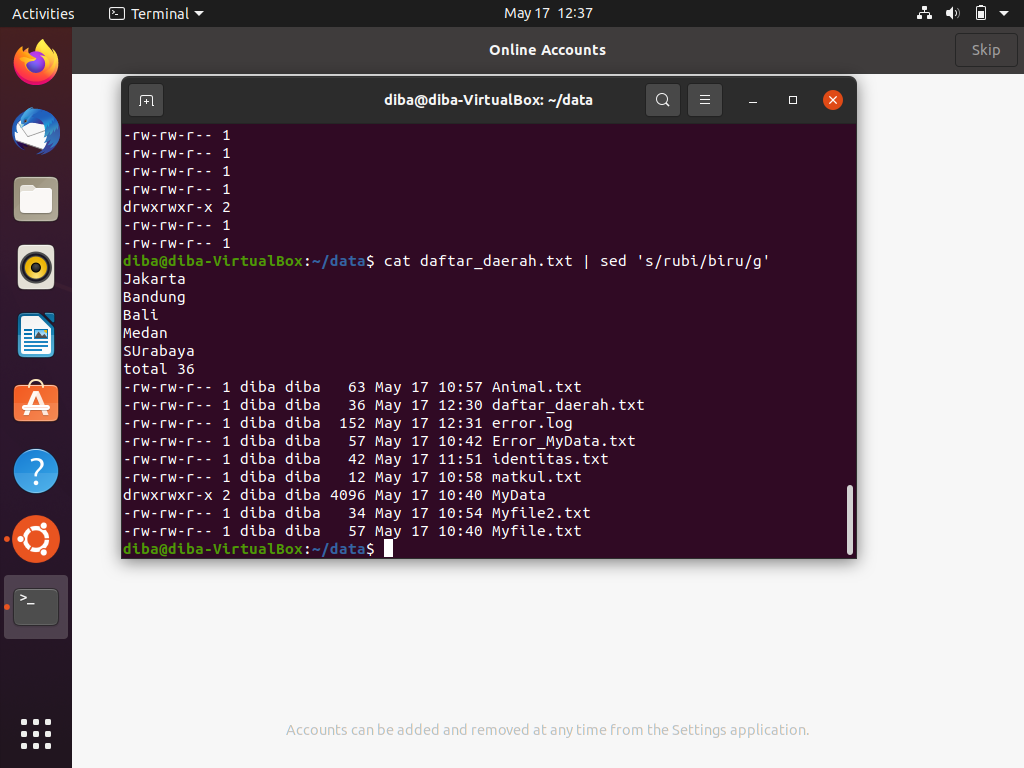
**3.3.8 Langkah Kedelapan (Paste)**



**3.3.9 Langkah Kesembilan (Awk)**



**3.3.10 Langkah Kesepuluh (Sed)**



**BAB III  
KESIMPULAN**

Proses dalam Linux membutuhkan input dan menghasilkan output. Input/output dalam konteks Linux dapat berasal dari keyboard, layar, file, struktur data kernel, dan peralatan I/O lainnya seperti jaringan.

Deskriptor file digunakan pada Linux untuk berkomunikasi dengan file. File descriptor ini direpresentasikan sebagai angka, di mana angka 0, 1, dan 2 digunakan untuk standar deskriptor file yang terkait dengan keyboard (standar input), layar (standar output), dan layer (standar error). Pembelokan (pengalihan) yang digunakan untuk mengalihkan standar deskriptor file. Simbol-simbol seperti "<" dan ">" digunakan untuk pembelokan input dan output standar. Pipa (pipeline), yang digunakan untuk menghubungkan output dari satu proses ke proses input berikutnya. Contoh penggunaan pipa ditunjukkan dengan simbol "|" yang menghubungkan proses-proses tersebut.

Utilitas Linux yang dapat memproses input dari keyboard dan menampilkannya pada layar. Beberapa contoh filter yang disebutkan adalah "cat, sort, grep, pr, head, tail, paste" dan sebagainya. Beberapa perintah Linux yang digunakan untuk proses penyaringan seperti "grep, wc, sort, cut, uniq" dan menjelaskan penggunaan karakter khusus untuk menyalakan masukan dan keluaran dalam terminal Ubuntu seperti ">", ">>", "<" , dan "|".