

# Representasi Pohon dalam Hierarki Linux Filesystem dan Manajemen Direktori/File

Bervianto Leo P - 13514047  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13514047@std.stei.itb.ac.id

**Abstrak**—Linux filesystem memiliki hierarki direktori yang dapat direpresentasikan menjadi pohon secara khusus yaitu pohon n-ary (salah satu contoh pohon berakar). Hal ini dapat memudahkan pengguna memahami suatu filesystem terutama Linux filesystem. Dengan menggunakan representasi pohon akan mudah memahami path atau jalur untuk menuju lokasi direktori yang diinginkan serta paham dengan prinsip manajemen direktori/file dalam Linux. Dari hal ini pengguna akan dengan mudah menguasai sistem operasi yang dimilikinya.

**Kata kunci**—filesystem, Linux, representasi, pohon.

## I. PENDAHULUAN

Berkembangnya teknologi menyebabkan sistem operasi yang akan digunakan ikut berkembang. Sebagai contoh sistem operasi Linux. Sistem operasi Linux terus dikenal oleh banyak orang dikarenakan open source-nya dan kebebasan untuk menggunakannya. Bahkan sistem operasi Linux saat ini sudah banyak digunakan oleh end-user atau pengguna rumahan bukan hanya digunakan oleh perusahaan-perusahaan.

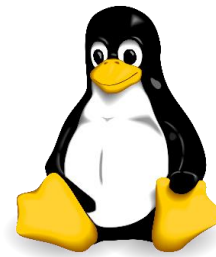


Gambar 1 Pencipta Sistem Operasi Linux

Sumber : <http://i1-news.softpedia-static.com/images/news2/linus-torvalds-says-people-who-believe-in-an-ai-singularity-are-on-drugs-486373-2.jpg>, diakses pada tanggal 8 Desember 2015

Mayoritas pengguna komputer (dalam hal ini dan untuk selanjutnya yang dimaksud yaitu end-user) hanya mengetahui direktori bagian pengguna itu sendiri. Sehingga memiliki keterbatasan pengetahuan mengenai direktori lain yang mendukung adanya suatu sistem operasi. Sehingga saat terjadi kerusakan, pengguna cenderung pasrah atau tidak mengerti cara mengatasinya. Namun dengan Linux diharapkan untuk paham mengenai direktori penyusunnya serta berisi apakah dalam direktori

tersebut agar tidak sembarangan mengubah direktori beserta isinya.



Gambar 2 Logo Linux

Sumber :

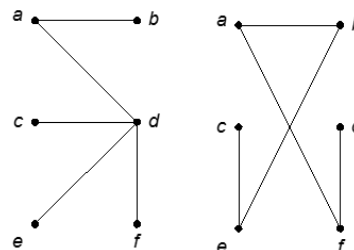
<https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/2000px-Tux.svg.png>, diakses pada tanggal 8 Desember 2015

Pada dasarnya suatu sistem operasi tersusun dari banyak direktori dengan memiliki satu buah akar yang sebagai inti dari suatu sistem operasi. Begitu juga dengan Linux yang memiliki suatu direktori bernama akar atau dalam simbol sebagai '/'.  
Dikarenakan uniknya filesystem dalam Linux. Filesystem tersebut direpresentasikan atau divisualisasikan agar mempermudah dalam pemahaman susunan dari filesystem dalam Linux.

## II. DASAR TEORI

### A. Definisi Pohon

Pohon adalah graf tidak berarah yang terhubung dan tidak mengandung sirkuit. Pohon yang digunakan mengacu pada teori graf. Misalkan  $G = (V, E)$  yaitu pohon, dengan  $V$  sebagai himpunan simpul yang tidak boleh kosong dan  $E$  sebagai himpunan sisi yang boleh kosong.

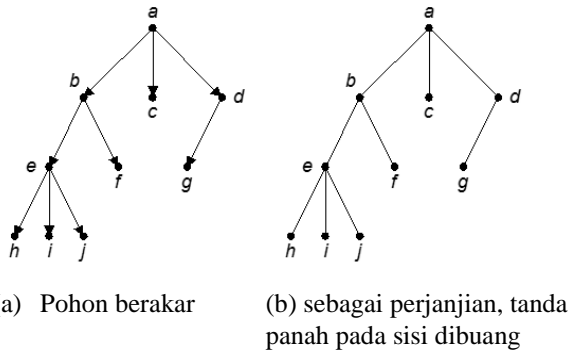


Gambar 3 Contoh Pohon

Sumber : referensi[1]

## B. Pohon Berakar

Pohon berakar yaitu pohon yang memperlakukan sebuah simpulnya sebagai akar serta sisi-sisinya diberi arah sehingga menjadi suatu graf berarah. Dalam pohon berakar setiap simpul dapat dikunjungi dengan sebuah lintasan yang unik. Arah sisi di dalam pohon dapat dibuang, namun tidak merubah arti dari pohon berakar. Simpul di pohon harus dicapai dari akar, sehingga lintasan di dalam pohon berakar akan selalu dari “atas” ke “bawah”. Berikut ini gambar yang mengilustrasikan pohon berakar tersebut.



Gambar 4 Ilustrasi Pohon Berakar  
Sumber : referensi [1]

Adapun terminologi pada pohon berakar yaitu sebagai berikut.

### i. Anak (*child* atau *children*) dan Orangtua (*parent*)

Dari gambar 4 (b) didapatkan seperti berikut, misalkan a sebagai **orangtua**, maka b, c dan d merupakan **anak** dari a. Sedangkan simpul c, f, g, h, i dan j tidak memiliki anak. Dari contoh tersebut dapat disimpulkan, misalkan ada simpul y dan x, simpul y akan dikatakan sebagai anak dari simpul x jika ada sisi dari simpul x ke y dengan simpul x sebagai orangtua.

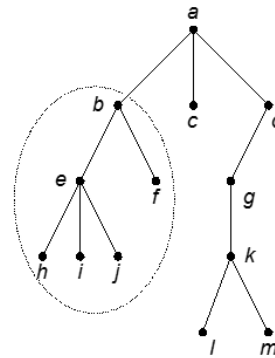
### ii. Lintasan (*path*)

Sebagai contoh pada gambar 4 (b) lintasan dari a ke g yaitu a, d, g dengan panjang lintasan 2. Lintasan didefinisikan runtutan simpul dari simpul  $v_1$  ke simpul  $v_n$  sehingga membentuk  $v_1, v_2, v_3, \dots, v_n$  sedemikian sehingga  $v_i$  merupakan orangtua dari  $v_{i+1}$  dengan  $1 \leq i < n$ . Serta panjang lintasan yaitu jumlah sisi yang dilewati dalam suatu lintasan, yaitu  $n-1$  dengan  $n$  sebagai simpul ke  $n$  pada runtutan simpul.

### iii. Saudara kandung (*sibling*)

Saudara kandung yaitu simpul yang memiliki orangtua yang sama. Sebagai contoh dalam gambar 4 (b) h merupakan saudara kandung i karena orangtua h dan i sama yaitu e.

### iv. Upapohon/Subpohon (*subtree*)



Gambar 5 Subpohon dengan b sebagai akarnya  
Sumber : referensi [1]

Upapohon atau subpohon merupakan bagian yang dilingkari pada gambar 5.

### v. Derajat (*degree*)

Derajat yang digunakan dalam pohon berakar berbeda dengan derajat dalam graf. Derajat yang dimaksud yaitu jumlah anak atau jumlah upapohon dalam suatu simpul. Sebagai contoh pada gambar 4 (b) simpul e berderajat 3 karena memiliki 3 anak.

### vi. Daun (*leaf*)

Daun merupakan simpul yang berderajat nol atau tidak memiliki anak. Sebagai contoh dalam gambar 4, simpul c, f, h, i, j, l dan m merupakan daun.

## C. Pohon n-ary

Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak disebut pohon n-ary (baca: ener). Pohon *n-ary* banyak digunakan di berbagai bidang ilmu maupun dalam kehidupan sehari-hari. Dalam terapannya, pohon *n-ary* digunakan sebagai model yang merepresentasikan suatu struktur. [1]

## III. HIERARKI LINUX FILESYSTEM

### A. Definisi Filesystem

Filesystem merupakan suatu metode dan struktur data yang digunakan oleh suatu sistem operasi untuk mengatur jalur dari suatu file dalam *disk* (tempat penyimpanan) atau suatu partisi dalam *disk*. Secara singkat merupakan suatu cara mengatur atau mengorganisasikan file-file dalam *disk* (tempat penyimpanan). [2]

### B. Definisi Hierarki

Menurut KBBI, hierarki merupakan urutan tingkatan atau jenjang jabatan. [3] Dari definisi tersebut jelas bahwa hierarki akan menunjukkan suatu tingkatan tertentu.

### C. Hierarki Filesystem dalam Linux

Linux memiliki hierarki tunggal dalam struktur direktori. [2] Artinya memiliki satu jalan utama yaitu dari suatu direktori yang dinamakan akar atau direktori akar.

Secara umum dalam hierarki *Linux Filesystem* terdapat subdirektori sebagai berikut, */bin*, */boot*, */dev*, */etc*, */lib*, */media*, */mnt*, */opt*, */sbin*, */tmp*, */usr*, dan */var*. Secara singkat mengenai subdirektori dalam */* sebagai berikut.

<i>/bin</i>	Berisi perintah-perintah penting dalam bentuk <i>file executable binary</i> .
<i>/boot</i>	<i>File</i> statis berhubungan dengan <i>boot loader</i> .
<i>/dev</i>	<i>File-file</i> berhubungan dengan perangkat seperti <i>disk</i> , <i>mouse</i> , dan lain-lain.
<i>/etc</i>	Konfigurasi sistem secara khusus dalam perangkat.
<i>/lib</i>	Berisi <i>library</i> yang dapat digunakan oleh sistem dan berisi modul kernel.
<i>/media</i>	Berisi data-data perangkat yang dapat digunakan sementara, seperti perangkat penyimpanan melalui USB.
<i>/mnt</i>	Direktori penyimpan data sementara dari suatu <i>filesystem</i> lain.
<i>/opt</i>	Berisi data-data dari paket aplikasi dari pihak ketiga.
<i>/sbin</i>	<i>Binary</i> penting dalam sistem.
<i>/tmp</i>	Berisi file-file sementara
<i>/usr</i>	Hierarki sekunder
<i>/var</i>	Data variabel

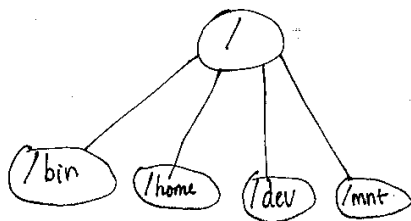
Sumber : referensi[2]

Sebagai tambahan, biasanya sistem *Linux* akan menambahkan subdirektori */home* yang berguna untuk menyimpan data-data dari pengguna.

Sebagai informasi lanjutan, */bin* merupakan direktori yang memiliki berbagai perintah-perintah penting, seperti *cp*, *mv*, *rm*, *cat*, dan *ls*, perintah tersebut sangat penting untuk mengatur direktori yang ada dalam *Linux filesystem*.

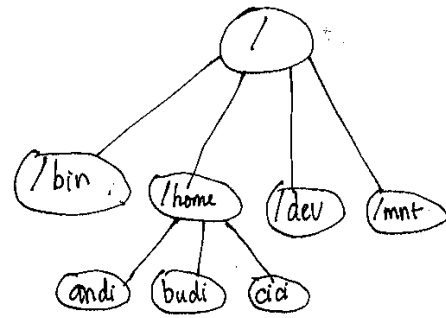
#### IV. REPRESENTASI POHON DALAM HIERARKI LINUX FILESYSTEM

Sebelumnya sudah mengenal mengenai *Linux Filesystem*. Secara umum hierarki dalam *Linux Filesystem* tersebut dapat dibentuk menjadi suatu pohon tepatnya menjadi pohon berakar. Berikut ini representasi pohon yang didapatkan berdasarkan hierarki dalam *Linux Filesystem* dengan hanya menampilkan beberapa subdirektori karena untuk menunjukkan representasinya dalam pohon.



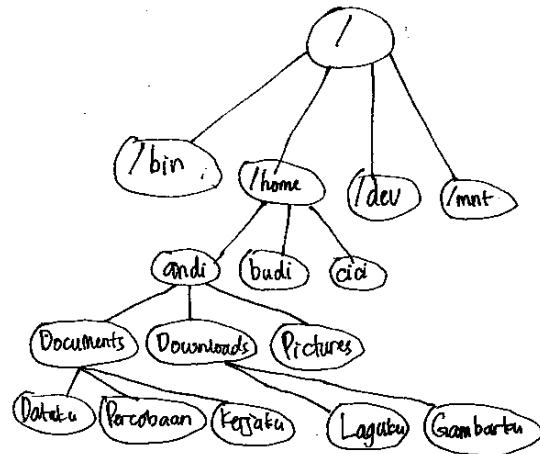
Gambar 6 Representasi Pohon pada Linux Filesystem secara Sederhana

Lalu jika dalam sistem *Linux* memiliki 3 *user* dengan *username* masing-masing yaitu *andi*, *budi*, dan *cici*. Maka representasi pohon akan sebagai berikut.



Gambar 7 Representasi Pohon setelah memiliki 3 user

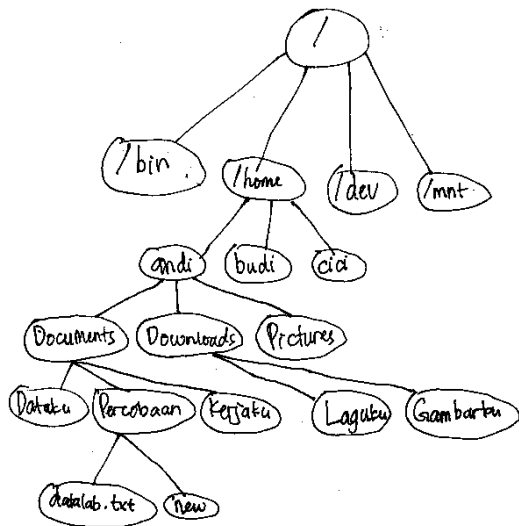
Di dalam direktori *user* bernama *andi* misalkan memiliki subdirektori *Documents*, *Downloads*, dan *Pictures*. Serta *Documents* memiliki subdirektori *Dataku*, *Percobaan*, dan *Kerjaku*. Sedangkan *Downloads* memiliki subdirektori *Laguku* dan *Gambarku*. Dengan *Pictures* tidak memiliki subdirektori. Maka representasi pohon akan sebagai berikut.



Gambar 8 Representasi Pohon pada pengguna andi

Dari representasi tersebut dapat dilihat bahwa *Dataku*, *Percobaan* dan *Kerjaku* sebagai daun sama dengan *Laguku* serta *Gambarku*. Sampai di sini dapat terlihat bahwa penyusunan direktori dalam *Linux* menyerupai pohon. Untuk mengakses direktori *Dataku* perlu melewati */*, */home*, *andi*, *Documents*, *Dataku*. Di dalam sistem operasi *Linux* jalur ini dikenal sebagai *path* yang biasanya dituliskan sebagai berikut */home/andi/Documents/Dataku*. Sampai di sini yang dikatakan daun ketika suatu direktori atau akar tidak memiliki subdirektori. Namun pada kenyataannya akan memiliki suatu file-file yang bukan suatu direktori. Setelah ini akan diberikan beberapa cara untuk memasukan file-file ke dalam representasi pohon tersebut.

Alternatif pertama dengan menganggap sama suatu *file* dengan direktori hanya berbeda memiliki suatu ekstensi *file* khusus yang menandakan bahwa data tersebut sebuah *file*. Berikut ini representasi pohon tersebut.



Gambar 9 Representasi Pohon dengan file dan direktori

Secara umum representasi ini sudah baik. *File* ditandai dengan ekstensi tertentu seperti *datalab.txt* yang menandakan *file* tersebut berekstensi *txt*. Namun jika dalam penggunaan struktur data atau dalam sistem *Linux*. Memiliki suatu program atau dapat disebutkan sebagai suatu *file* tidak memiliki ekstensi sehingga jika menggunakan representasi tersebut akan menyebabkan ambigu.

Dalam mengatasi hal tersebut dapat diterapkan struktur data elemen yang dibagikan dua jenis. Berikan informasi mengenai data tersebut adalah direktori atau *file*. Namun representasi dari pohon tidak akan berubah. Hanya isi informasi mengenai elemennya akan berubah.

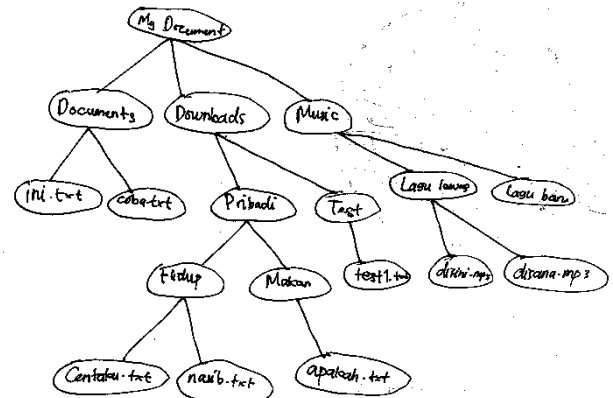
Sehingga yang dikatakan daun yaitu suatu *file* atau direktori yang tidak memiliki subdirektori. Setelah selesai merepresentasikan pohon tersebut. Representasi tersebut akan dihubungkan dengan terminologi pohon berakar.

Misalkan */home* sebagai anak maka orangtuanya yaitu */*, dalam *filesystem* dikenal */home* sebagai *directory* dan */* sebagai *parent directory*. Lalu lintasan dari */* ke *andi* yaitu */, /home, andi*, dalam *filesystem* juga dikenal sebagai *path* dengan penulisan sebagai berikut */home/andi*. Begitu juga dengan */bin, /home, dan /dev* sebagai *sibling* dari */mnt* dengan orangtua */*, dalam *filesystem* dikenal sebagai beberapa subdirektori dari */*. Selanjutnya upapohon, misalkan *andi* sebagai upapohon, dalam *filesystem* dikenal sebagai subdirektori. Derajat pada *Documents* yaitu 3, dalam *filesystem* dikenal sebagai isi dari *Documents* yaitu 3 *items*. Terakhir *datalab.txt* dan *new* sebagai daun, dalam *filesystem* dikenal sebagai *file* jika merupakan data *file*, dikenal sebagai direktori kosong jika sebagai data direktori. Hal tersebut menyatakan bahwa hierarki *Linux filesystem* dapat direpresentasikan dengan baik dalam bentuk pohon berakar yaitu pohon *n-ary*.

## V. MANAJEMEN DIREKTORI/FILE

Setelah mengetahui representasi dari hirarki *Linux Filesystem* sehingga pengguna juga perlu mengetahui

beberapa metode yang digunakan untuk dalam *filesystem* seperti menghapus direktori, menghapus *file*, menambahkan *file*, menambahkan direktori, pencarian direktori atau *file*, dan lain-lain. Dalam hal ini penulis akan menggunakan suatu pohon berakar (*n-ary*) yang akan digunakan untuk memvisualisasikan langkah-langkah manajemen direktori/*file*.



Gambar 10 Representasi Pohon dari Direktori My Document sebagai contoh

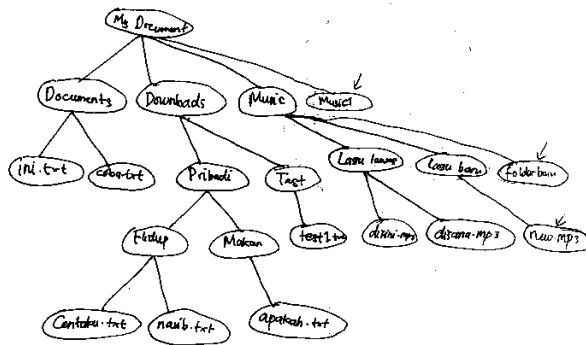
### A. Pencarian Direktori/File

Dalam pencarian direktori/*file* lebih baik menggunakan suatu kata kunci dan memulai pencarian dari direktori yang diduga tidak jauh dari direktori/*file* tersebut. Misalkan ingin mencari *file ini.txt* dengan menggunakan kata kunci ini, kemungkinan akan keluar nama lain yang memiliki sub-kata dari ini seperti *disini.mp3* jika mencarinya mulai dari *My Documents*. Jika mencari berawal dari *Documents*, akan sangat cepat ditemukan dan lebih tepat. Untuk isi direktori yang sedikit akan tidak terasa untuk mencarinya, namun jika sudah sangat banyak akan lama untuk mencari satu *file* yang diinginkan bahkan beresiko tidak tercarinya *file* atau direktori yang dimaksud. Sangat disarankan jika sudah terorganisir dengan baik, pencarian *file* dengan kategori khusus cari di direktori dengan kategori tersebut. Pada prinsipnya komputer atau dalam *Linux* akan mencari hingga semua anak/subdirektori yang ada di dalamnya sudah mencapai daun seluruhnya.

### B. Menambahkan Direktori/File

Untuk menambahkan prinsipnya cukup mudah. Hanya menambahkan suatu subpohon atau daun menjadi anaknya. Direktori yang akan ditambahkan yaitu direktori dengan nama folder baru pada direktori *Music* dan menambahkan *file new.mp3* pada direktori *Lagu Baru*. Hal tersebut akan langsung ditambahkan jika tidak ada nama yang sama. Namun berbeda halnya dengan ingin menambahkan direktori *Music* pada *My Document*, karena direktori *Music* sudah ada. Sistem akan memberi peringatan bahwa direktori tersebut sudah ada dan akan direkomendasikan dengan nama lain seperti pada gambar di bawah atau dapat ditimpa dengan yang baru namun pada contoh saat ini akan diganti dengan nama lain. Pada dasarnya setiap direktori atau *file* yang menjadi *sibling* harus memiliki nama yang

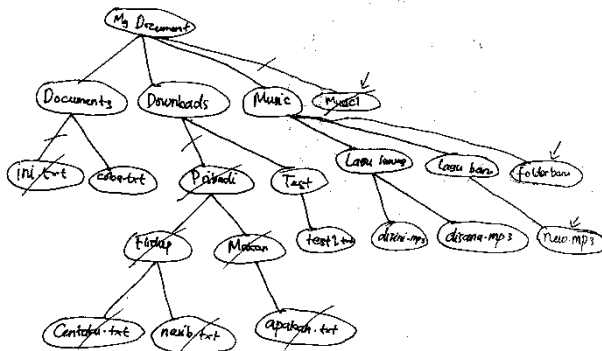
unik. Berikut ini ilustrasi dari menambahkan direktori/file. Direktori/file yang baru ditunjukkan oleh tanda panah.



Gambar 11 Penambahan File/Direktori baru

### C. Menghapus Direktori/File

Untuk menghapus direktori kosong atau file sangat mudah karena menghapus daun. Namun untuk menghapus direktori yang tidak kosong akan dilakukan penghapusan subdirektori yang ada di dalamnya. Misalkan direktori pribadi akan dihapus, maka subdirektori dan file yang dibawahnya juga akan dihapus. Berbeda dengan menghapus Music1 atau *ini.txt*, cukup menghapus direktori atau file yang dimaksud. Berikut ini ilustrasi dari penghapusan tersebut. Direktori/file yang dihapus, ditandai dengan simpul tersebut dicoret.



Gambar 12 Penghapusan Direktori/File

### D. Tambahan

Sedikit yang unik dari Linux Filesystem yaitu jika mengalokasikan beberapa direktori khusus misalnya direktori /home dan / sebagai partisi yang terpisah. Jika ingin menghapus partisi /, partisi /home tidak akan ikut terhapus, karena hal ini /home dalam / bukan yang sebenarnya atau /home dalam / hanya berisi informasi penunjuk /home berada. Yang artinya / dan /home sebagai pohon yang berdiri sendiri.

## VI. SIMPULAN

Dapat disimpulkan bahwa Linux Filesystem dapat direpresentasikan menggunakan pohon terutama pohon berakar yaitu pohon *n-ary*. Pohon *n-ary* dengan baik memodelkan atau merepresentasikan hierarki Linux Filesystem. Sehingga akan berhubungan dengan

terminologi atau sifat dari pohon berakar. Seperti, setiap mengakses direktori tertentu akan memiliki *path* yang unik. Selain itu, dalam penentuan struktur data pohon yang akan digunakan dengan pengenalan perbedaan file dan direktori dimasukan data khusus menyimpan informasi identitas suatu data merupakan direktori atau file. Serta dalam manajemen direktori atau file dapat direpresentasikan melalui pohon juga.

## VII. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan YME untuk segala karunia-Nya sehingga penulis dapat menyelesaikan makalah ini. Terimakasih kepada dosen pengampu IF2120 Matematika Diskrit, Ibu Harlili dan Bapak Rinaldi Munir, untuk pengajaran yang telah diberikan dalam kuliah Matematika Diskrit secara khusus mengenai pohon yang menjadi dasar dalam penulisan makalah ini.

## REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF2120 Matematika Diskrit", Bandung: Teknik Informatika ITB, 2003, halaman IX-1 – IX-16.
- [2] <http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/Linux-Filesystem-Hierarchy.pdf>, diakses pada tanggal 8 Desember 2015.
- [3] <http://kbbi.web.id/hierarki>, diakses pada tanggal 8 Desember 2015.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2015

Bervianto Leo Pratama - 13514047