**MAKALAH**

**PENGENALAN TEKNOLOGI KOMPUTER DAN INFORMASI  
“BASH LINUX”**

**Logo

Description automatically generated**

**Disusun oleh Kelompok IV:**

**Dafa Fikri Kanadua (50422368)**

**Fadiah Ahmad (50422485)**

**Hendy Andrean Pasaribu (50422665)**

**Kautsar Hasby Dastien Fredila (50422770)**

**Muhammad Raka Pratama (50422956)**

**Muhammad Faridsyah Ziqri (51422040)**

**Niko Dwicahyo Widiyanto (51422245)**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS GUNADARMA**

**2022/2023**

# KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT yang telah memberikan segala nikmat, berkat dan rahmat-Nya sehingga kami dapat menyelesaikan tugas kelompok pembuatan makalah Pengenalan Teknologi Komputer dan Informasi tentang “Bash Linux”

Era komputer sudah dimulai sejak lama, segala perintah perintah atau syntax yang dapat di gunakan untuk menjalankan isi sebuah komputer pun telah dibuat, yang memudahkan kita untuk menerjemahkan bahasa kita kedalam Bahasa komputer

Tujuan dari penulisan ini secara umum untuk mengetahui bagian dari komputer yaitu Bash linux, semoga dengan makalah ini kita semua dapat mempelajari bash linux untuk mempermudah pekerjaan kita kelak. Jika terdapat kekurangan atau ketidaktepatan dalam makalah ini, maka kami memohon maaf dan siap untuk menerima masukan dari pembaca.

Penulis menyadari bahwa masih banyak kekurangan pada penulisan makalah ini. maka dari itu, saran dan kritik yang membangun sangat penulis harapkan dari pembaca sekalian. Penulis berharap semoga makalah ini dapat bermanfaat bagi siapa saja yang membacanya.

Depok. 08 Januari 2023

Penulis

# **DAFTAR ISI**

[KATA PENGANTAR 2](#_Toc124147913)

[DAFTAR ISI 3](#_Toc124147914)

[BAB 1 4](#_Toc124147915)

[PENDAHULUAN 4](#_Toc124147916)

[1.1 Latar Belakang 4](#_Toc124147917)

[1.2 Rumusan Masalah 5](#_Toc124147918)

[1.3 Tujuan 5](#_Toc124147919)

[BAB 2 6](#_Toc124147920)

[PEMBAHASAN 6](#_Toc124147921)

[2.1 Konsep Bash Linux 6](#_Toc124147922)

[2.2 Sejarah Bash Linux 6](#_Toc124147923)

[2.3 Fitur Bash Linux 7](#_Toc124147924)

[2.3.1 Ekspansi penjepit 7](#_Toc124147925)

[2.3.2 Manajemen Proses 8](#_Toc124147926)

[2.3.3 Eksekusi Bersyarat 10](#_Toc124147927)

[2.3.4 Pelaporan Bug 10](#_Toc124147928)

[2.3.5 Penyelesaian yang dapat diprogram 10](#_Toc124147929)

[2.4 Contoh Program Bash Linux 10](#_Toc124147930)

[2.5 Perbandingan Bash Scripting dengan Bahasa script lain 13](#_Toc124147931)

[2.6 Keuntungan Bash Linux 13](#_Toc124147932)

[2.7 Kekurangan Bash Linux 14](#_Toc124147933)

[BAB 3 16](#_Toc124147934)

[PENUTUP 16](#_Toc124147935)

[3.1 Kesimpulan 16](#_Toc124147936)

[3.2 Saran 16](#_Toc124147937)

[DAFTAR ISI 17](#_Toc124147938)

# BAB 1

# PENDAHULUAN

## Latar Belakang

Linux merupakan nama yang diberikan untuk komputer berbasis sistem operasi unix, Linux juga adalah salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama atau disebut juga dengan open source yang dimana kode sumber Linux dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja. Linux pada awalnya dibuat oleh seorang mahasiswa finlandia yang bernama Linus Torvalds. Dulunya Linux merupakan proyek hobi yang diinspirasikan dari minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanen-baum. Beberapa versi linux di buat dengan update secara terus menerus , Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991 dan 2 bulan setelahnya, tepatnya pada tanggal 5 Oktober 1991, Linus mengumumkan tentang pembuatan versi resmi Linux yaitu versi 0.02 , yang dimana hanya dapat menjalankan Shell Bash (GNU Bourne Again Shell) dan GCC (GNU C Compiler)

Shell merupakan program atau penerjemah perintah yang bertindak sebagai antarmuka atau interface antara user dengan sistem operasi, yang mana kernel yang menjadi inti dari sistem operasi, pada umumnya shell menyediakan prompt sebagai user interface, yaitu tempat untuk mengetikkan perintah-perintah yang diinginkan oleh user baik berupa perintah internal shell (internal command), ataupun perintah eksekusi suatu file progam (eksternal command).

Shell dapat digunakan oleh user untuk menyusun sekumpulan perintah pada sebuah file atau beberapa file untuk dieksekusi sebagai suatu program. Berbeda dengan sistem operasi lain yang hanya menyediakan satu atau 2 shell saja, sistem operasi dari keluarga unix ini, dalam hal ini misalnya linux, saat ini dilengkapi oleh berbagai jenis shell dengan sekumpulan perintah yang sangat banyak, sehingga memungkinkan user untuk memilih shell mana yang paling baik untuk membantu menyelesaikan pekerjaannya, atau dapat pula berpindah-pindah dari shell yang satu ke shell yang lain dengan mudah, beberapa shell yang ada di linux antara lain seperti Bourne Shell (sh), C Shell (csh), Korn Shell (ksh), dan Bourne Again Shell (bash)

## Rumusan Masalah

Dari uraian latar belakang di atas, beberapa permasalahan yang akan dibahas pada makalah ini adalah:

* Bagaimana konsep Bash Linux ?
* Bagaimana Bash Linux dapat dibuat ?
* Apa saja Fitur yang terdapat pada Bash ?
* Apa Keuntungan dan kekurangan apa yang terdapat pada Bash Linux ?

## Tujuan

* Mengetahui Konsep Bash Linux
* Mengetahui Cara Menggunakan Bash Linux
* Mengetahui Fitur apa saja yang ada pada Bash Linux
* Mengetahui Kelebihan Bash Linux
* Mengetahui Kekurangan Bash Linux

# BAB 2

# PEMBAHASAN

## Konsep Bash Linux

Bash Linux adalah Unix Shell dan Bahasa  Perintah yang ditulis oleh Brian Fox untuk proyek GNU sebagai pengganti perangkat lunak bebas untuk Bourne Shell. Shell atau bahasa perintah ini pertama kali dirilis pada tahun 1989 dan telah didistribusikan secara luas karena merupakan shell default pada distribusi Linux dan MacOS. Kemudian pada tahun 2016 shell ini juga disediakan oleh Microsoft pada produknya yaitu Windows 10 Anniversary Update, Meskipun tidak di install secara default.

Bash adalah prosessor perintah yang biasanya berjalan di jendela teks, dimana pengguna jenis perintah yang menyebabkan tindakan. Bash juga dapat membaca perintah dari file yang disebut script. Soal penamaan, Bash adalah singkatan dari Bourne Again Shell yang merupakan versi baru dari Bourne Shell yang dikembangkan oleh Steve Bourne.

Pemrograman bash shell itu sendiri adalah pemrograman kumpulan perintah menggunakan script yang ditulis kedalam bahasa shell, sehingga nantinya dapat dieksekusi oleh sistem operasi. Pada linux sendiri, penggunaan bash ini dijalankan pada terminal.

Konsep kerja dari pemrograman bash shell hampir mirip dengan bahasa pemrograman lainnya. Pemrograman bash shell juga menggabungkan perintah-perintah untuk memilih suatu kondisi, memproses suatu I/O, looping, dan membuat fungsi-fungsi yang dapat dijalankan user. Konsep pemrograman bash shell ini akan mudah dipelajari apabila kita sudah mengetahui perintah-perintah sederhana yang ada di bash shell seperti whoami, cd, cat, dan lainnya. Dengan penguasaan perintah-perintah sederhana ini, pemrograman bash shell akan membuat pekerjaan user menjadi lebih efektif. Hal penting yang harus diketahui sebelum menggunakan pemrograman bash shell adalah konsep mengenai variabel, format syntax dan struktur di dalam pemrograman bash shell itu sendiri.

## Sejarah Bash Linux

Bash adalah shell Unix dan bahasa perintah yang ditulis oleh Brian Fox untuk Proyek GNU sebagai pengganti perangkat lunak gratis untuk Bourne shell. Pertama kali dirilis pada tahun 1989, telah digunakan sebagai shell login default untuk sebagian besar distribusi Linux. Bash adalah salah satu program pertama Linus Torvalds porting ke Linux, bersama GCC . Sebuah versi juga tersedia untuk Windows 10 dan Windows 11 melaluiSubsistem Windows untuk Linux. Ini juga merupakan shell pengguna default di Solaris 11. Bash juga merupakan shell default dalam versi Apple macOS dari 10.3 (awalnya, shell default adalah tcsh ) hingga rilis 2019 macOS Catalina , yang mengubah shell default menjadi zsh , meskipun Bash tetap tersedia sebagai shell alternatif.

Bash adalah prosesor perintah yang biasanya berjalan di jendela teks tempat pengguna mengetik perintah yang menyebabkan tindakan. Bash juga dapat membaca dan menjalankan perintah dari file, yang disebut skrip shell . Seperti kebanyakan shell Unix, ini mendukung globbing nama file (pencocokan karakter pengganti), piping , here documents , command substitution , variables , dan struktur kontrol untuk pengujian kondisi dan iterasi . Kata kunci , sintaks , variabel cakupan dinamis dan fitur dasar lainnya daribahasa semua disalin dari sh . Fitur lain, misalnya history , disalin dari csh dan ksh . Bash adalah shell yang sesuai dengan POSIX , tetapi dengan sejumlah ekstensi.

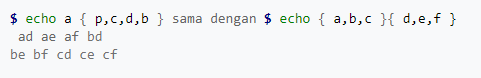
Nama shell adalah akronim untuk Bourne Again Shell , plesetan dari nama shell Bourne yang digantikannya dan gagasan " dilahirkan kembali ".

Sebuah lubang keamanan di Bash berasal dari versi 1.03 (Agustus 1989), dijuluki Shellshock , ditemukan pada awal September 2014 dan dengan cepat menyebabkan berbagai serangan di Internet. Tambalan untuk memperbaiki bug tersedia segera setelah bug teridentifikasi.

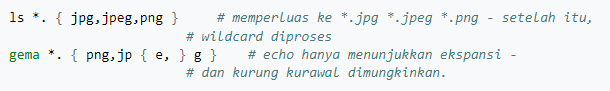
## Fitur Bash Linux

### Ekspansi penjepit

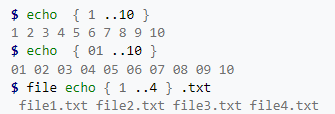
Ekspansi brace ( brace expansion ), juga disebut pergantian, adalah fitur yang disalin dari shell C. Ini menghasilkan satu set kombinasi alternatif. Hasil yang dihasilkan tidak perlu ada sebagai file. Hasil dari setiap string yang diperluas tidak diurutkan dan urutan dari kiri ke kanan dipertahankan:



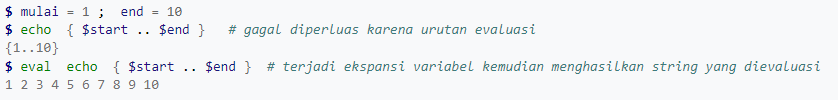
Pengguna tidak boleh menggunakan ekspansi penjepit dalam skrip shell portabel, karena shell Bourne tidak menghasilkan output yang sama.



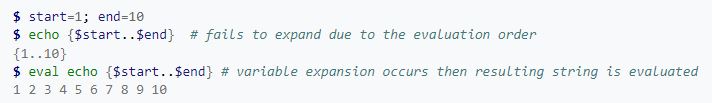
Saat ekspansi brace digabungkan dengan wildcard, brace diperluas terlebih dahulu, lalu wildcard yang dihasilkan diganti secara normal. Karenanya, daftar gambar JPEG dan PNG di direktori saat ini dapat diperoleh dengan menggunakan:



Selain pergantian, ekspansi penjepit dapat digunakan untuk rentang berurutan antara dua bilangan bulat atau karakter yang dipisahkan oleh titik ganda. Versi Bash yang lebih baru memungkinkan bilangan bulat ketiga untuk menentukan kenaikan.



Ketika ekspansi brace digabungkan dengan ekspansi variabel (AKA ekspansi parameter dan substitusi parameter ) ekspansi variabel dilakukan setelah ekspansi brace, yang dalam beberapa kasus mungkin memerlukan penggunaan ***eval*** built-in



### Manajemen Proses

Shell Bash memiliki dua mode eksekusi untuk perintah: batch , dan mode bersamaan. Untuk menjalankan perintah dalam batch (yaitu, secara berurutan) mereka harus dipisahkan dengan karakter ";", atau pada baris terpisah:



dalam contoh ini, ketika perintah1 selesai, perintah2 dijalankan.

Eksekusi latar belakang dari perintah1 dapat terjadi menggunakan (simbol &) di akhir perintah eksekusi, dan proses akan dieksekusi di latar belakang dengan segera mengembalikan kontrol ke shell dan mengizinkan eksekusi lanjutan dari perintah.

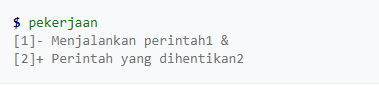


Atau untuk melakukan eksekusi bersamaan dari dua perintah1 dan perintah2, keduanya harus dijalankan di Bash Shell dengan cara berikut:



Dalam hal ini perintah1 dijalankan di latar belakang & simbol, segera mengembalikan kontrol ke shell yang menjalankan perintah2 di latar depan.

Suatu proses dapat dihentikan dan kontrol dikembalikan ke bash dengan mengetikkan Ctrl+ z saat proses sedang berjalan di latar depan. Daftar semua proses, baik di latar belakang maupun berhenti, dapat dicapai dengan menjalankan ***jobs***:



Di output, angka dalam tanda kurung mengacu pada id pekerjaan. Tanda tambah menandakan proses default untuk bgdan fg. Teks "Menjalankan" dan "Berhenti" mengacu pada status Proses . String terakhir adalah perintah yang memulai proses.

Keadaan suatu proses dapat diubah menggunakan berbagai perintah. Perintah fgmembawa proses ke latar depan, sementara bgmenyetel proses yang berhenti berjalan di latar belakang. bgdan fgdapat mengambil id pekerjaan sebagai argumen pertama mereka, untuk menentukan proses yang akan ditindaklanjuti. Tanpa satu, mereka menggunakan proses default, yang diidentifikasi dengan tanda plus di output dari jobs. Perintah killdapat digunakan untuk mengakhiri proses sebelum waktunya, dengan mengirimkannya sinyal . ID pekerjaan harus ditentukan setelah tanda persen:



### Eksekusi Bersyarat

Bash memasok pemisah perintah "eksekusi bersyarat" yang membuat eksekusi perintah bergantung pada kode keluar yang ditetapkan oleh perintah preseden. Sebagai contoh:



Di mana ./do\_something hanya dijalankan jika perintah cd (ubah direktori) "berhasil" (mengembalikan status keluar nol) dan perintah gema hanya akan dijalankan jika perintah cd atau ./do\_something mengembalikan "kesalahan" (status keluar bukan nol).

Untuk semua perintah, status keluar disimpan dalam variabel khusus $?. Bash juga mendukung dan membentuk evaluasi perintah bersyarat : **if** ...;**then** ...;**else** ...;**ficase** $VARIABLE **in** $pattern)...;;$other\_pattern)...;; **esac**

### Pelaporan Bug

bashbug" dialihkan ke sini. Untuk bug September 2014 yang dilaporkan secara luas ditemukan di Bash.

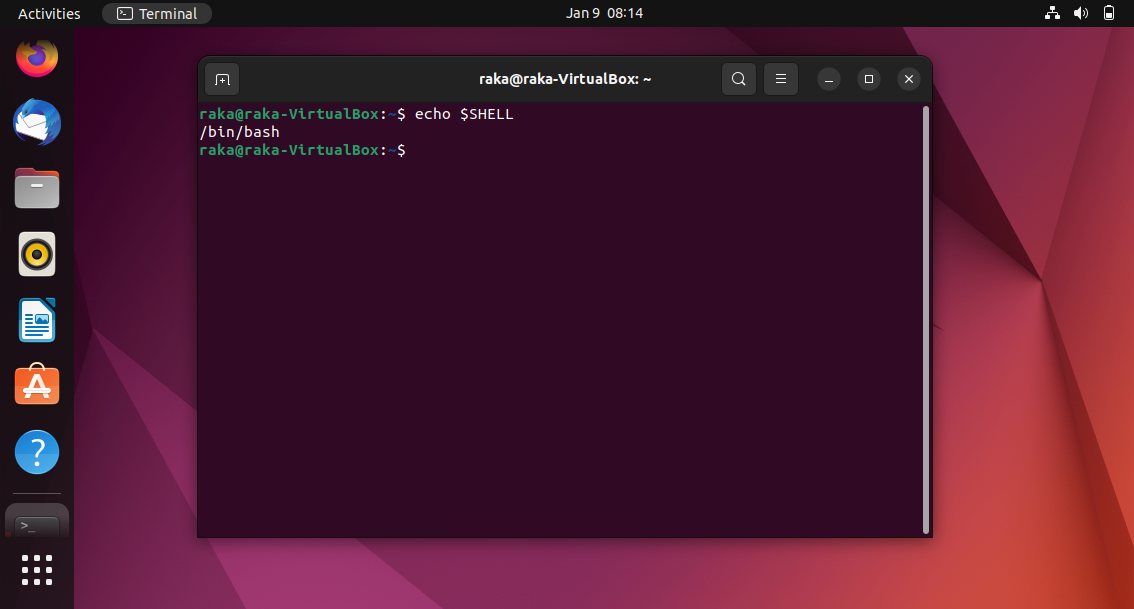
Perintah eksternal yang disebut bashbug melaporkan bug shell Bash. Ketika perintah dipanggil, itu akan menampilkan editor default pengguna dengan formulir untuk diisi. Formulir dikirimkan ke pengelola Bash (atau secara opsional ke alamat email lain).

### Penyelesaian yang dapat diprogram

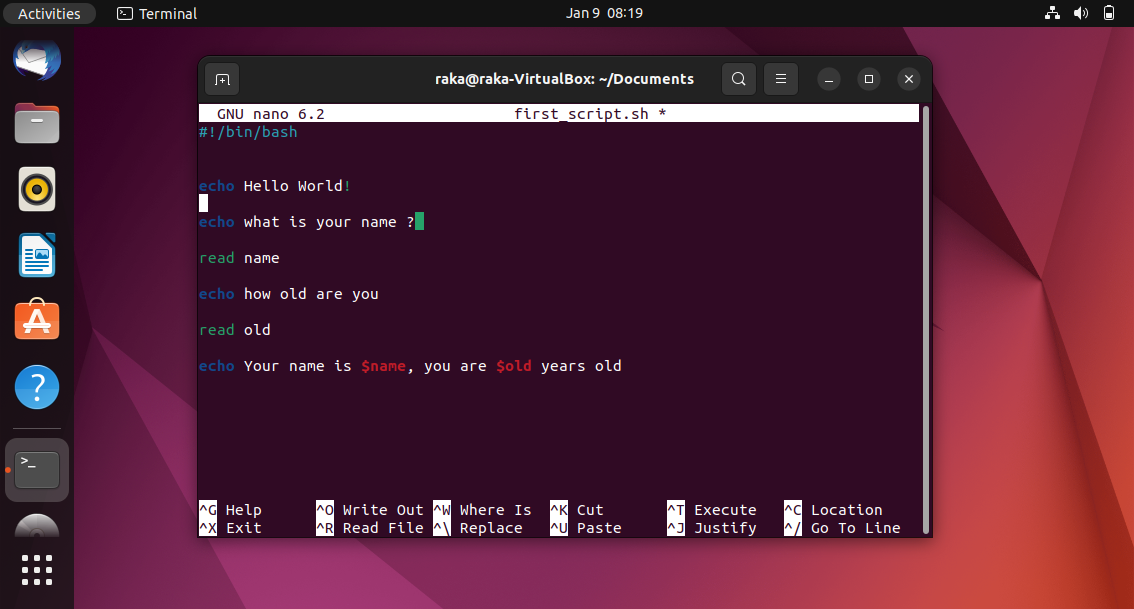
Perintah complete dan compoptdua menentukan bagaimana argumen dari beberapa perintah atau opsi yang tersedia akan dicantumkan dalam input. Pada versi 5.1 penyelesaian perintah atau opsi biasanya diaktifkan dengan menekan tombol . Tab ↹ setelah mengetik namanya

## Contoh Program Bash Linux

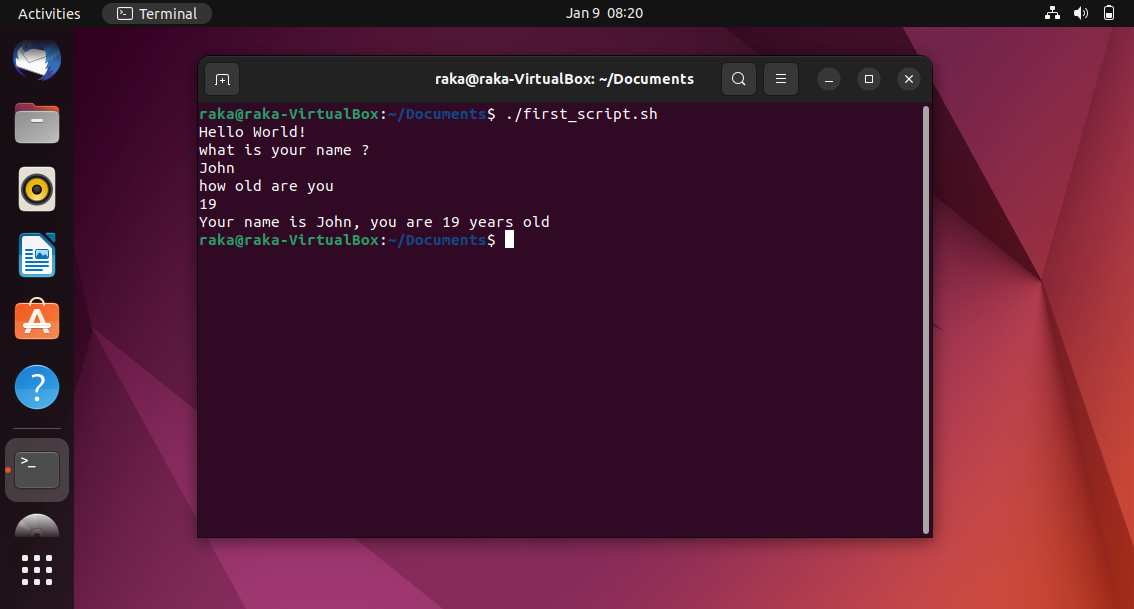
Pada contoh program Bash Linux berikut menggunakan salah satu distribusi linux yaitu ubuntu. Shell yang digunakan pada terminal ubuntu merupakah Bash, sehingga kita bisa menggunakan terminal ubuntu untuk membuat program bash linux. Untuk mengecek apakah shell pada distribusi linux menggunakan bash, kita dapat mengetik “echo $SHELL” untuk bisa melihat jenis shell yang digunakan



Berikut merupakan contoh program sederhana Bash linux :



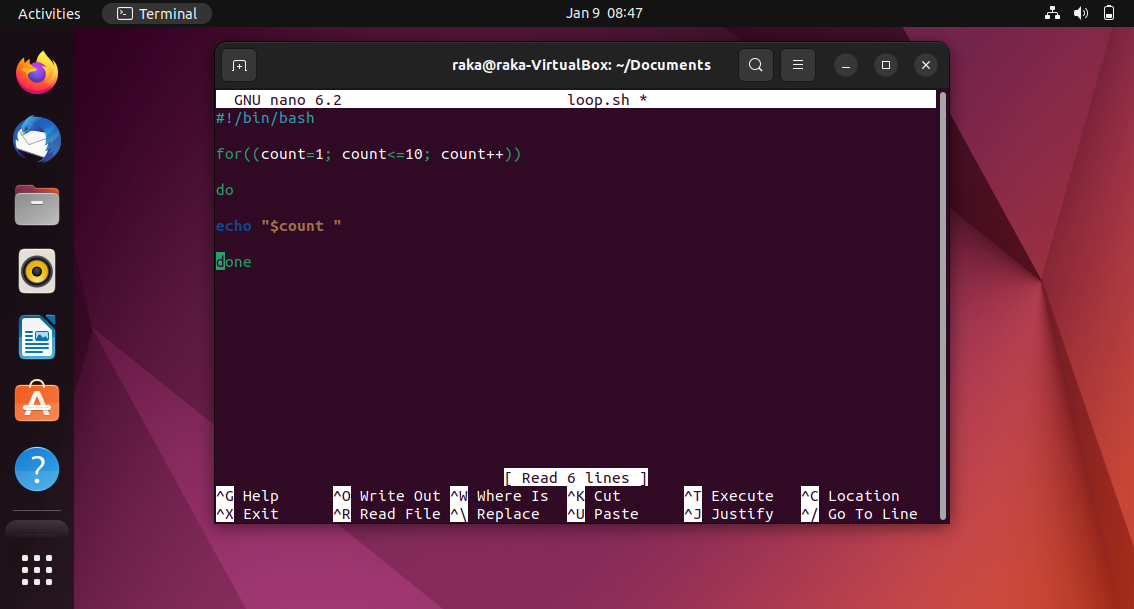
Saat file ini di akses :



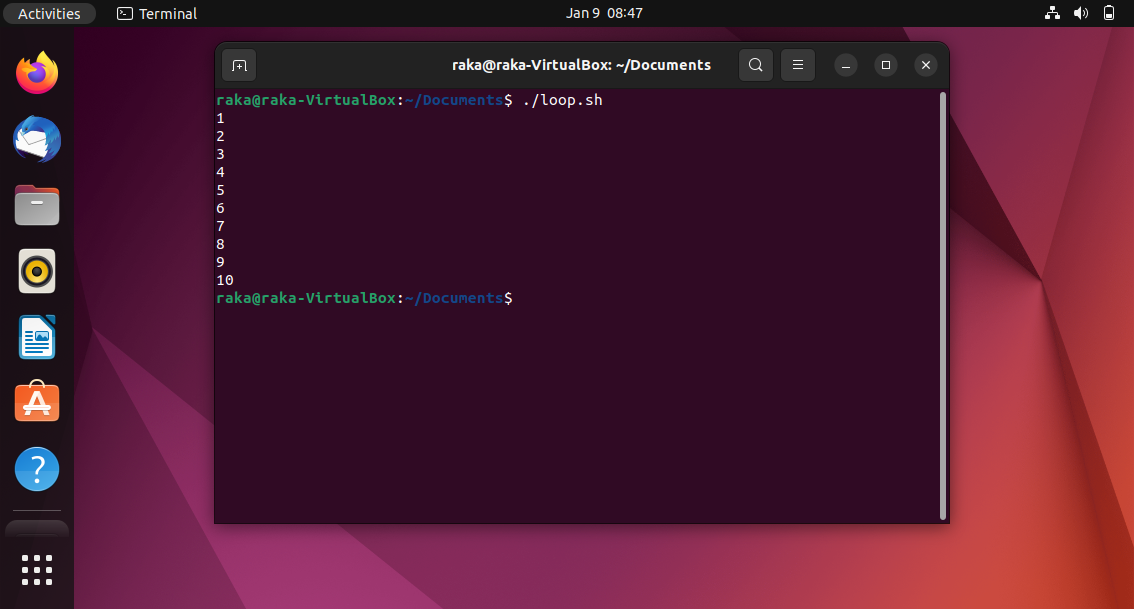
Bash script juga mendukung :

* While loop
* For loop
* If statement
* And logic
* Or logic
* Else if statement
* Case statement

Di bawah ini merupakan salah satu contoh fitur Bash Script, For loop :



Bash script ini akan menghasilkan output sebagai berikut :



## Perbandingan Bash Scripting dengan Bahasa script lain

Skrip Bash adalah jenis skrip yang ditulis dalam bahasa pemrograman Bash, yang merupakan shell Unix dan penerjemah baris perintah. Skrip Bash biasanya digunakan di Linux, MacOS, dan sistem mirip Unix lainnya untuk mengotomatiskan tugas dan menyederhanakan proses yang rumit.

Salah satu keuntungan utama skrip Bash adalah ketersediaan dan kompatibilitasnya yang luas dengan sistem mirip Unix. Karena Bash adalah komponen standar dari sistem ini, skrip Bash dapat dijalankan di mesin apa pun yang memiliki juru bahasa Bash yang terpasang.

Namun, Bash memiliki sejumlah keterbatasan dibandingkan dengan bahasa scripting lainnya. Itu tidak memiliki dukungan untuk pemrograman berorientasi objek atau tipe data yang kuat, dan sintaksnya bisa kikuk dan sulit dibaca.

Sebagai perbandingan, bahasa seperti Python dan Ruby umumnya lebih kuat dan fleksibel daripada Bash, dengan sintaks yang lebih modern dan fitur yang lebih beragam. Mereka juga sering lebih mudah dipelajari dan digunakan, dan memiliki komunitas pengembang dan pengguna yang lebih besar.

Secara keseluruhan, pilihan antara Bash dan bahasa skrip lain akan bergantung pada persyaratan dan tujuan khusus dari tugas yang ada. Bash mungkin merupakan pilihan yang baik untuk tugas-tugas sederhana pada sistem mirip Unix, sementara proyek yang lebih kompleks atau lintas platform mungkin lebih cocok untuk bahasa yang berbeda.

## Keuntungan Bash Linux

Bash Linux memiliki beberapa keuntungan dibandingkan shell lain yaitu :

1. Bash shell merupakan shell yang menggabungkan fitur-fitur yang ada di Korn Shell dan C Shell dengan peningkatan fitur-fitur yang cocok untuk programming ataupun penggunaan oleh user secara interaktif. Bash shell menggunakan standard POSIX yang paling umum digunakan dalam pemrograman shell.
2. Bash shell bisa menjalankan hampir semua file .sh (script) tanpa modifikasi terlebih dahulu
3. Syntax dari bash shell programming sederhana dan tidak bertele-tele, sama seperti dengan menggabungkan function-function yang sudah ada di dalam statu file script.
4. Hanya ada sedikit ‘rule’ yang perlu dipelajari untuk menulis program menggunakan bash shell
5. Pemrograman menggunakan shell merupakan salah satu metode untuk prototipe aplikasi yang lebih kompleks lagi.
6. Pemrograman menggunakan shell merupakan pendekatan yang paling mudah dalam hal problem solving, karena masalah yang kompleks dipecah menjadi sub-sub masalah yang lebih mudah dipecahkan.
7. Manfaat yang paling penting untuk system administrator adalah memelihara system dengan command-command yang bisa dijalankan secara otomatis dan berlaku default untuk semua user tanpa user sendiri yang harus mensetting environment dari system.

Kelebihan shell di linux dibanding sistem operasi lain memang ada berbagai macam, tapi yang paling utama adalah bahwa shell di linux memungkinkan kita untuk menyusun serangkaian perintah seperti halnya bahasa pemrograman (interpreter language), melakukan proses I/O, menyeleksi kondisi, looping, membuat fungsi, dsb. adalah proses - proses yang umumnya dilakukan oleh suatu bahasa pemrograman, jadi dengan shell di linux kita dapat membuat program seperti halnya bahasa pemrograman, untuk pemrograman shell pemakai unix atau linux menyebutnya sebagai script shell.

## Kekurangan Bash Linux

Tiap-tiap jenis Shell memiliki kekurangan dan kelebihan masing-masing. Disesuaikan pada kebutuhan saja. Pada Linux, jenis Shell yang sering digunakan adalah Shell bash. Karena perintah-perintahnya kuat dan juga digunakan oleh banyak orang sehingga dukungan untuk shell bash ini banyak tersedia dan shell bash ini selalu di perbaharui.

Berikut ini beberapa kelemahan dari menggunakan skrip Bash:

* Fitur yang terbatas : Bash tidak mendukung banyak fitur canggih yang terdapat pada bahasa pemrograman lain, seperti pemrograman berorientasi objek, tipe data kuat, dan penanganan pengecualian. Hal ini dapat membuat sulit untuk menulis skrip Bash yang kompleks atau berkualitas tinggi.
* Bacaan yang buruk : Skrip Bash dapat sulit dibaca dan dipahami, terutama bagi mereka yang tidak familiar dengan sintaks Bash. Hal ini dapat membuat lebih sulit untuk memelihara dan memodifikasi skrip yang ada, serta untuk memudahkan bagi pengguna baru untuk mempelajarinya dan menggunakannya.
* Kinerja yang buruk : Skrip Bash dapat lebih lambat dieksekusi dibandingkan dengan program yang dikompilasi atau skrip yang ditulis dalam bahasa lain. Hal ini dapat menjadi masalah bagi tugas-tugas yang harus segera dilakukan atau secara real-time.
* Tergantung platform : Skrip Bash terutama digunakan pada sistem sejenis Unix, dan mungkin tidak bekerja pada platform lain seperti Windows tanpa perangkat lunak tambahan. Hal ini dapat membatasi portabilitas dan fleksibilitas skrip Bash.
* Kelemahan keamanan : Skrip Bash dapat rentan terhadap berbagai masalah keamanan, seperti serangan injeksi dan eksekusi perintah yang tidak sengaja. Perlu dipertimbangkan dengan hati-hati untuk memperbaiki input dan memperkuat keamanan skrip Bash agar dapat mencegah risiko-risiko tersebut.

# BAB 3

# PENUTUP

## Kesimpulan

Kesimpulannya, skrip Bash adalah alat yang ampuh untuk mengotomatiskan tugas dan menyederhanakan proses yang rumit. Mereka memungkinkan pengguna untuk menulis dan mengeksekusi kode yang dapat melakukan berbagai fungsi, dari eksekusi perintah sederhana hingga logika yang lebih kompleks dan kontrol aliran. Skrip Bash dapat digunakan pada berbagai platform, termasuk Linux, MacOS, dan sistem mirip Unix lainnya.

Salah satu manfaat utama menggunakan skrip Bash adalah fleksibilitas dan keserbagunaannya. Mereka dapat digunakan untuk mengotomatiskan tugas-tugas sederhana, seperti menjalankan serangkaian perintah atau menyiapkan lingkungan pengembangan, atau proses yang lebih kompleks, seperti menerapkan aplikasi perangkat lunak atau mengelola jaringan.

Secara keseluruhan, skrip Bash adalah alat penting bagi siapa pun yang bekerja dengan sistem mirip Unix dan layak dipelajari bagi siapa pun yang ingin meningkatkan efisiensi dan produktivitasnya.

## Saran

## Mempelajari lebih dalam tentang Bash Linux, fitur fitur yang diberikan oleh bash sangatlah berguna dan efektif

## Mempelajari Bahasa script lebih lanjut, selain Bash

## Menggunakan program bash untuk menyelesaikan masalah di dunia nyata

# DAFTAR ISI

<https://rumahradhen.wordpress.com/2018/04/01/makalah-sistem-operasi-linux/#:~:text=Linux%20pada%20awalnya%20dibuat%20oleh,dikerjakan%20sekitar%20bulan%20Agustus%201991>

<https://www.awonapa.com/2021/12/makalah-sistem-operasi-working-with-bash.html?m=1>

<https://g.co/kgs/bJYK7W>

<https://id.m.wikipedia.org/wiki/Linux>

<https://en.wikipedia.org/wiki/Bash_(Unix_shell>

<https://mahasiswa.ung.ac.id/531413015/home/2013/11/7/bash-shell.html#:~:text=Kelebihan%20shell%20di%20linux%20dibanding,proses%20yang%20umumnya%20dilakukan%20oleh>

<https://iks32753.wordpress.com/2012/04/03/pemrograman-bash-shell-2/>

<https://mahasiswa.ung.ac.id/531413017/home/2013/10/17/bash-shell.html>

<https://chat.openai.com/chat>