IMPLEMENTASI DAN ANALISIS PENGUATAN PADA SISI *AUTHENTICATION* PADA HADOOP MENGGUNAKAN *OPEN SOURCES TOOLS* KERBEROS

IMPLEMENTATION AND HARDENING ANALYSIS ON THE AUTHENTICATION SIDE OF HADOOP USING KERBEROS OPEN SOURCES TOOLS

Muhammad Valen[1], Adityas Widjajarto[2], Ahmad Almaarif[3]

[1,2,3] Universitas Telkom. Bandung
[1]**muhammadvalen@student.telkomuniversity.ac.id,** [2] **adtwjrt@telkomuniveristy.co.id,** [3] **ahmadalmaarif@telkomuniversity.ac.id**

Abstrak

Penggunaan Hadoop sebagai framework pengolahan big data udah menjadi kebutuhan untuk dilindungi dari serangan. Penerapan Kerberos pada Hadoop untuk mengatasi serangan yang dapat menggagu sistem Hadoop. Penelitian ini bertujuan untuk melakukan mekanisme pengamanan dari sisi Authentication dengan melakukan implementasi Kerberos sebagai alat untuk mengamankan Hadoop. Dari hasil pengujian yang telah dilakukan, mekanisme Kerberos dalam mengamanankan Hadoop dapat berjalan tanpa ada nya kendala yang artinya server Kerberos dapat berlajan sesuai mekanisme yang telah dilakukan untuk mengamankan hadoop pada sisi *Authenticaiton*. Semua bukti terkait informasi pengamanan dari kinerja sistem dicatatan dan tersimpan dalam log.

Kata kunci : ***Hadoop, Kerberos, Security, Kemanan pada Hadoop***

Abstract

The use of Hadoop as a big data processing framework has become a necessity to be protected from attacks. Implemented Kerberos on Hadoop to address attacks that could disrupt Hadoop systems. This study aims to carry out a security mechanism from the Authentication side by implementing Kerberos as a tool to secure Hadoop. From the results of the tests that have been carried out, the Kerberos mechanism in securing hadoop can run without any obstacles, which means the Kerberos server can run according to the mechanism that has been carried out to secure hadoop on the Authenticaiton side. All evidence related to security information of system performance is recorded and stored in the logs.

Keywords: ***Hadoop, Kerberos, Security, Security on Hadoop***

1. Pendahuluan

Big Data memerlukan sebuah infrastruktur untuk mengolahnya. Salah satu framework pengolahan *Big Data* yaitu Hadoop. Hadoop adalah sebuah *software framework* untuk komputasi yang *reliable, scalable, parallel,* dan Komputasi terdistribusi[1]. Hadoop berisi berbagai modul, yang diperbarui dari waktu ke waktu untuk menambahkan fungsi yang berbeda ke fungsi inti dan Hadoop berfungsi untuk memproses, mengatur dan menganalisis dari berbagai macam tipe data misalnya *structured*, *unstructured* dan *semi-structured*.[2]. Terdapat berbagai macam platform dan software *Big Data* pada saat ini, yang digunakan untuk memproses dan menganalisa suatu *data sets* yang besar, contohnya *IBM Big Data analytics, HP Big Data, SAP Big Data analytics*, dan *Microsoft Big Data.* Namun saat ini platform Hadoop banyak digunakan[1].

Keamanan adalah masalah utama di Hadoop karena perlu adanya pengamanan. Adapun berbagai tantangan keamanan di Hadoop yaitu mulai dari *Authentication* yang tepat oleh pengguna, Enkripsi data tingkat perusahaan saat data istirahat dan dalam transit, *history* dalam mengakases data untuk semua pengguna dicatat untuk tujuan penting dan peraturan kepatuhan dan pengguna memilik hak hanya dapat mengakses data yang seharusnya dapat diakses. Ancaman adalah elemen yang sangat berbahaya bagi sistem apa pun. Ancaman dapat berupa orang atau program yang menemukan kerentanan pada individu atau perusahaan untuk menyerang sistem mereka. CIA adalah prinsip keamanan yang mengacu pada integritas, ketersediaan, dan kerahasiaan. Kerahasiaan berarti bahwa pengguna yang diautentikasi dapat mengakses sistem Hadoop mengasumsikan lingkungan tepercaya. Keamanan Hadoop dibagi menjadi dua tingkat. Di level 1, sistem dimulai di lingkungan tepercaya dengan komputasi yang kurang aman. Di level 2, sistem Hadoop menambahkan Kerberos sebagai parameter keamanan. Selama fase keamanan Hadoop, berbagai proyek ditambahkan, seperti Rhino, Apache Sentry, dan Apache Ranger. Dalam proyek untuk menemukan solusi keamanan khusus sistem Hadoop digunakan pada level 4. *Authetication* dan Authirization adalah akar dari keamanan. Jika kami tidak melakukan ini, kami tidak akan dapat memisahkan pengguna yang diautentikasi yang dapat mengaksesnya dari pengguna yang tidak dapat

menggunakannya. Hadoop *stacks* memiliki berbagai jenis komponen, tanpa keamanan dan keamanan default. Saat menjalankan konfigurasi yang berbeda, manajemen tambalan diperlukan. Tidak ada alat khusus untuk mendeteksi operasi dan penyalahgunaan yang berbahaya. Ketika ada banyak data, gunakan alat audit dan pemantauan[3].

Kemudian Informasi bersumber dari beberapa portal berita seperti eweek.com dan infoworld.com menunjukkan bahwa penggunaan DemonBot untuk menyerang server Hadoop telah dicatat, yang dapat memicu penyebaran serangan DDOS (*Distrubuted Denial of Service*) pada server Hadoop dan klasternya. Dan situasi ini sudah terjadi. Serangan tersebut adalah kerusakan berbahaya pada sistem Hadoop. Penyerang mengubah konfigurasi sistem Hadoop dan memasukkan kata-kata yang tidak menguntungkan ke konfigurasi tersebut. Dari dampaknya pada sistem, mungkin kata-kata ini tidak akan mempengaruhi sistem, tetapi jika Anda melihat potensinya Serangannya mungkin ketika server dimatikan atau data besar yang terdapat dalam Hadoop mungkin dicuri, mengubah, menghapus dan menghancurkan file konfigurasi sistem Hadoop akan sangat merugikan perusahaan. Dalam pengembangan *Big Data*, pengembang tidak mempertimbangkan isu yang muncul dari segi keamanan dan privasi. Masalah *Big Data* yang paling menantang yaitu keamanan dan privasi itu sendiri[2]. Maka dari itu perlu adanya langkah pengamanan yang dilakukan untuk mengamankan Hadoop. Adapun langkah pengamanan yang akan digunakan pada penelitian kali ini berfokus pada mekanisme pengaman yang dapat di lakukan pada Hadoop dari sisi *Authentication*.

Selaran dengan prinsip pada CIA yang mengacu pada integritas, ketersediaan, dan kerahasiaan. Kerahasiaan berarti bahwa pengguna yang diautentikasi dapat mengakses sistem Hadoop. Dengan mengaplikasikan satu dari konsep diatas dapat menjadikan Hadoop lebih aman.

Tujuan penelitian ini untuk mengimplementasi langkah pengamanan untuk memperkuat aspek kemanan dari sisi *Authentication* untuk mengamankan Hadoop dengan menggunakan Kerberos.

.

2. Dasar Teori /Material dan Metodologi/perancangan

2.1 *Big Data*

Big Data pada masa kini telah banyak diperbincangkan dan menjadi sebuah isu yang hangat, hal ini terkait dengan semakin berkembangnya teknologi informasi yang memperkenalkan Big Data untuk mendukung industri kedepannya. Big Data adalah kumpulan dari sejumlah besar data yang penting dan memiliki sejarah, yang merupakan aset paling berharga dari setiap organisasi dan individu yang kemudian dimanfaatkan secara cerdas untuk keperluan bisnis sehingga dapat mendukung dalam pengambilan keputusan berdasarkan fakta daripada persepsi(Bhathal & Singh, 2019).

Big Data memiliki karakteristik umum yang dikenal dengan istilah 3 V's of data. yaitu volume, velocity, dan variety. Tetapi hal-hal tersebut kurang lengkap untuk menjelaskan mengenai Big Data pada saat ini. Sekarang Big Data tidak mencakup itu saja melainkan terdapat beberapa faktor lain yang dapat menjelaskan Big Data secara lebih jelas seperti veracity, validity, value, variability, venue, vocabulary, dan vagueness(Yadav et al., 2019).

Karakteristik dari 3 V’s dari Big Data yaitu:

1. *Volume* (Besaran) Jumlah nya sangat besar (*Volume*). Biasanya ukuran total data dalam terabytes keatas.
2. *Velocity* (Kecepatan) Pertumbuhan data sangat cepat (*Velocity*) sehingga data bertambah dalam jumlah yang sangat banyak dalam kurun waktu relatif singkat.
3. *Variety* (Variasi) Bentuk atau format datanya beraneka ragam (*Variety*). Format disini bisa berupa data dalam tabel-tabel relasional database seperti MySQL, file text biasa, File *Excel* atau bentuk apapun.

2.2 Hadoop

Hadoop adalah *framework open source* berbasis Java di bawah lisensi Apache untuk mensupport aplikasi yang berjalan pada *Big Data*. Hadoop berjalan pada lingkungan yang menyediakan *storage* dan komputasi secara terdistribusi ke kluster-kluster dari *computer/node*. Hadoop sendiri berfungsi untuk memproses, mengatur dan menganalisis dari berbagai macam tipe data misalnya *structured, unstructured* dan *semi-structure*[2]. Kumudian Hadoop sendiri memiliki dua komponen utama yaitu *Map Reduce* dan Hadoop *Distributed File System* (HDFS).

2.3 HDFS

HDFS adalah sistem yang terdistribusi mandiri yang menyediakan penyemipanan data yang dapat diskalakan serta *fault-tolerant* terhadap kesalahan pada perangkat keras yang dirancang agar dapat diaplikasikan pada kluster dan dapat di jalan dengan menggunakan *proprietary* atau *commodity Server* yang dikembangkan oleh Apache *Software Foundation*[1].

Keuntungan utama dari HDFS adalah *fault tolerance*. Dengan menyediakan transfer data yang cepat antara node dan menggunakan Hadoop untuk memberikan layanan bahkan ketika terjadi kekgagalan node akan mengurangi resiko yang terjadi yang disebabkan oleh kegagalan. Maka dari itu HDFS juga mampu memberikan solusi penyimanan *skala-out* untuk Hadoop[1].

2.4 Kerberos

MIT (*Massachusetts Institute of Technology*) membangun protokol ini untuk *authentication* identitas. Kerberos digunakan untuk memverifikasi akses pengguna ke cluster Hadoop. Ini memberikan pengidentifikasi unik sehingga komunikasi yang aman dapat dibuat dengan bantuan enkripsi kunci pada jaringan yang tidak aman. Itu bergantung pada KDC (*Key Distribution Centre*) dan merupakan sistem tiket berdasarkan SSO (*Single Sign-On*).

Hadoop menggunakan Kerberos sebagai dasar untuk autentikasi yang kuat dan penyebaran identitas untuk pengguna dan layanan. Kerberos adalah mekanisme autentikasi pihak ketiga, di mana pengguna dan layanan bergantung pada pihak ketiga - server Kerberos - untuk mengautentikasi satu sama lain. Server Kerberos sendiri dikenal sebagai Key Distribution Center, atau KDC. Pada tingkat tinggi, ia memiliki tiga bagian:

1. Basis Data Pengguna dan Layanan yang Dikenal (dikenal sebagai Prinsipal) dan Kata Sandi Kerberos
2. *Autentication Server* (AS) Melakukan autentikasi awal dan menerbitkan Ticket Award Ticket (TGT)
3. *Ticket Granting Server* (TGS) menerbitkan tiket untuk layanan selanjutnya berdasarkan TGT yang awal.

Prinsipal pengguna meminta autentikasi dari AS mengembalikan TGT terenkripsi dengan kata sandi prinsip Kerberos pengguna yang hanya diketahui oleh prinsipal pengguna dan AS Saat tiket kedaluwarsa, prinsipal pengguna dapat menggunakan TGT, untuk mendapatkan tiket layanan dari TGS yang memungkinkan prinsipal untuk mengakses berbagai layanan.

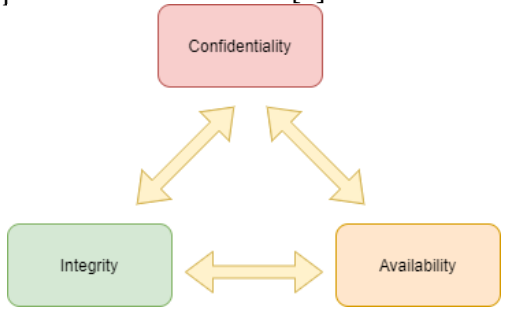
Karena sumber daya *cluster* (*host* atau layanan) tidak dapat memberikan kata sandi untuk mendekripsi TGT setiap kali, mereka menggunakan *file* khusus yang disebut **keytab** yang berisi informasi autentikasi utama sumber daya. Kelompok *host*, pengguna, dan layanan di mana *server* Kerberos berada dan memiliki kontrol dikenal **realm**.

Tabel 1. Terminologi Kerberos

Term	Deskripsi
Key Distribution Center atau KDC	Pusat Distribusi Kunci, atau KDC Sumber tepercaya untuk autentikasi di lingkungan yang mendukung Kerberos.
Kerberos KDC Server	Mesin, atau server, yang berfungsi sebagai Key Distribution Center (KDC).
Kerberos Client	Setiap mesin di *cluster* yang mengautentikasi terhadap KDC.
Principal	Nama unik pengguna atau layanan yang diautentikasi terhadap KDC.
Keytab	File yang menyertakan satu atau lebih prinsipal dan kuncinya.
Realm	Jaringan Kerberos yang mencakup KDC dan sejumlah *client*.
KDC *Admin Account*	Akun administratif yang digunakan oleh Ambari untuk membuat prinsipal dan membuat *keytab* di KDC.

2.5 CIA

Ada tiga aspek utama dalam keamanan keamanan informasi, dan biasa disingkat CIA. Di bawah ini adalah penjelasan dari aspek keamanan informasi[4]:



Gambar 1. Aspek Kemanan Informasi

Ada tiga aspek utama dalam keamanan keamanan informasi, dan biasa disingkat CIA. Di bawah ini adalah

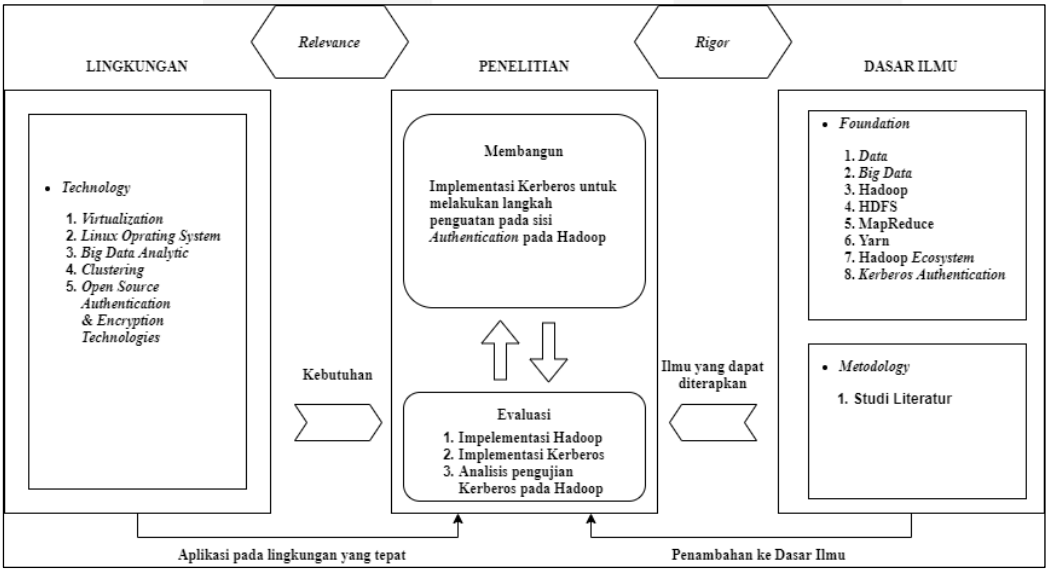
penjelasan dari aspek keamanan informasi[4]:

1. *Confidentiality*
   Confidentiality ketika informasi yang dilindungi bersifat rahasia terjaga keamananntya. *Confidentiality* data dan informasi dapat akses atau dilihat oleh siapapun yang berhak. Aspek ini mudah dipahami oleh orang. Misalnya, untuk identitas atau *Authenticaiton*, aspek ini disebut dengan istilah *pricacy* (Kerashasiaan). Menyerang aspek ini dalam bentuk penyadapan dan pencurian *drive* yang digunakan untuk menyimpan data. Memberikan perlindungan terhadap aspek *confidentiality* ini dengan menggunakan kriptografi dan akses terbatas.
2. *Integrity*
   Integrity tercapai ketika informasi dilindungi dari modifikasi oleh orang yang tidak berwenang, hilangnya integritas memiliki dampak terhadap bisinis besar yang terwujud dalam penipuan dan pencurian layanan. *Integrity* menyatakan, tanpa persetujuan maka data tidak boleh ada yang berubah. Serangan integritas dilakukan oleh lingkungan, yang menangkap data saat mengirim dan menghancurkan atau memodifikasi data, dan kemudian meneruskannya ke tujuan yang diinginkan. Perlindungan integritas dapat dicapai melalui kode *authentication*.
3. Availability
   Availability tercapai bila layanan yang diberikan oleh perusahaan dilindungi dari akses yang tidak sah. *availability* artinya sistem harus tersedia pada saat dibutuhkan. karena merupakan bagian dari keamanan.serangan aksesibilitas adalah serangan DOS karena membuat layanan tidak dapat digunakan atau sangat lambat. dapat dilindungi dengan menyediakan redundansi.

3. Metode Penelitian

3.1 Model Konseptual

Model koseptual merukapan sebuah gambaran mengenai sebuah konsep secara logis dari suatu permasalahan yang akan dilakukan pada penelitian yang dapat membantu dan mengevaluasi factor-faktor yang relevan dari permasalahan yang ada. Pada Gambar III-1 dibawah ini merupakan gambaran model konseptual yang akan di gunakanan pada penelitian ini.



Gambar 2. Model Konseptual

3.2 Sistematika Penyelesaian Masalah

Berdasarkan kerangka penyelesaian masalah atau model konseptual di atas. Berikut merupakan sistematika penyelesaian masalah yang mana merupakan langkah atau alur-alur yang akan dilaksanakan pada penilitian ini. Langkah yang digunakan dimulai dari tahapan awal (tahap perumusan masalah), tahap perancangan, tahap eksperimen, tahap analisis dan tahap akhir (kesimpulan). Penjelasan dari penyelesaian permasalahan yang digunakan dapat dilihat pada Gambar 2 sebagai berikut.

3.3 Sistematika Penyelesaian Masalah

Berdasarkan kerangka penyelesaian masalah atau model konseptual di atas. Berikut merupakan sistematika penyelesaian masalah yang mana merupakan langkah atau alur-alur yang akan dilaksanakan pada penilitian ini. Langkah yang digunakan dimulai dari tahapan awal (tahap perumusan masalah), tahap perancangan, tahap eksperimen, tahap analisis dan tahap akhir (kesimpulan).

Penjelasan dari penyelesaian permasalahan yang digunakan dapat dilihat pada Gambar III-2 sebagai berikut.

3.3.1 Tahap Awal (Perumusan Masalah)

Pada tahap pertama yaitu perumusahan masalah. Pada tahap ini melakukan identifikasi berdasarkan latar belakang permasalahan dalam penelitian. Kemudian dilanjutkan dengan studi literatur dengan melakukan ovservasi terkait sumber-sumber dari buku, thesis maupun paper terkait dengan penelitian ini. Selanjutnya yaitu membuat rumusan masalah dan batasan masalah yang menjadi acuan dalam penelitian ini sehingga dapat memfokuskan ruang lingkup pembahasan berdasarkan penelitian yang dilakukan..

3.3.2 Tahap Perancangan

Pada tahap perancangan dilakukan rancangan dari instalasi sistem operasi serta spesifikasi Hardware dan Software yang dibutuhkan untuk menjunjang penelitian dan dilanjutkan pada tahap perancangan sistem dengan melakukan implementasi penguatan kemanan pada Hadoop berdasarkan skenario yang telah dibuat dengan menggunakan open sources security tools Kerberos untuk melakukan pengamanan dari sisi Authentication.

3.3.3 Tahap Eksperimen

Pada tahap ketiga yaitu tahap eksperimen dan analisis. Pada tahap ini mekakukan eksperimen yang dillalukan mulai dari perancangan strategi pengujian untuk keamanan pada Hadoop. Kemudian dilanjutkan dengan proses instalasi *tools* yang dibutuhkan untuk penelitian yang akan dilakukan. Kemudian dilanjutkan dengan tahap melakukan pengujian yang mengacu terhadap skenario yang telah dibuat. penelitian yang akan dilakukan. Pengujian dilakukan dengan menggunakan Kerberos. Hasil dari eksperimen yang diperoleh selama pengujian akan diolah pada tahap selanjutnya.

3.3.4 Tahap Analisis

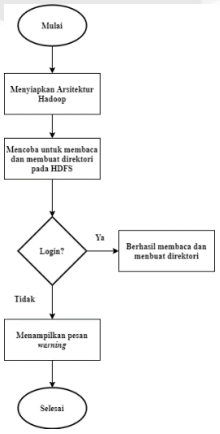
Pada tahap ini juga melakukan analisis terhadap penelitian yang mana tahap analisis dilakukannya analisa terhadap hasil dari pengujian yang sudah dilakukan dengan menggunakan Kerberos yang digunakan pada tahap eksperimen. Sehingga setelah dilakukan proses analisa maka akan di dapatkan analisis dari pengujian yang telah dilakukan.

3.3.5 Tahap Akhir Kesimpulan

Pada tahap akhir akan dilakukan pembuatan kesimpulan dan saran dari yang berasal dari hasil analisis yang dilakukan sebelumnya. Dimana hasil dari penelitian ini berupa laporan hasil dari penelitian. Kesimpulan dan saran yang telah didapatkan dapat menjadi acuan atau penunjang untuk pengembangan topik atau pembahasan pada penelitian selanjutnya.

4. Analisis dan Implementasi

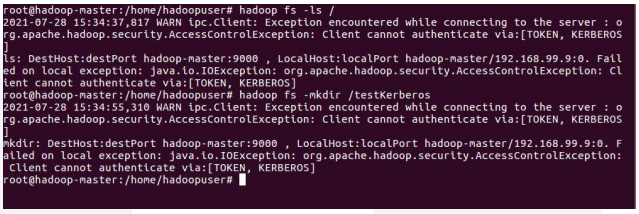
4.1 Hasil Pengujian Analisis pada HDFS sebelum Proses Autentikasi



Gambar 3. Diagram Pengujian pada HDFS

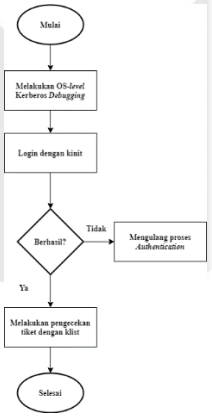
Pada penelitian ini dilakukan percobaan dengan melakukan pada direktori HDFS dengan melakukan pengujian dengan dua *command* yaitu:

1. Tahap pertama, melakukan pengujian dengan melakukan pengecekan pada direktori HDFS proses ini dilakukan tanpa login terlebih dalu. Dengan menggunakan *command* hadoop fs -ls / berfungsi untuk mendapatkan semua informasi yang terdapat pada list direktori yang ada jika semuanya baik baik saja.
2. Kemudian dilanjutkan dengan mencoba membuat direktori pada direktori HDFS dengan menggunakan command hadoop fs -ls /.
3. Setelah mengeksekusi dua command diatas, didapatkan hasil nya yaitu dengan menampilkan pesan WARN ipc.Client: Exception encountered cannot while connecting to the server dan Client cannot authenticate via:[TOKEN, KERBEROS]. Yang mana jika berhasil direktori akan langsung dapat dibaca dan dibuat.
4. WARN yang artinya *warning* dengan keterangan yang tertera pada Gambar V-3 menunjukan bahwa *client/user* dan tidak dapat melakukan koneksi ke server tanpa adanya *authentication* terlebih dahulu melalui *token* dan *ticket* Kerberos.



Gambar 4. Hasil Pengujian pada HDFS

4.2 Hasil Pengujian Analisis pada *debugguging* dan *login* menggunakan kinit



Gambar 5. Diagram Pengujian debugging dan login menggunakan kinit

1. Tahap pertama, menambahkan *variable environment* untuk melakukan proses *debugging* pada direoktori /tmp.kinit.log sebagai tempat dimana log hasil debugging akan disimpan dengan menggunakan *command* export KRB_TRACE=/tmp/kinit.log.
2. Kemudian dilanjutkan dengan proses pembuatan tiket dimana kinit sendiri mengasumsikan jika ingin mendapatkat tiket harus menggunakan username sendiri dan menggunakan default realm, dengan menggunakan *command* kinit -kt hadoopuser.keytab hadoopuser/hadoop-master@HDREALM.COM. Ketikan *command* ini dipakai jika benar akan langsung secara otomatis login dan jika salah maka akan diminta *authentication* kembali dengan meminta mekasukan ulang kata sandi.

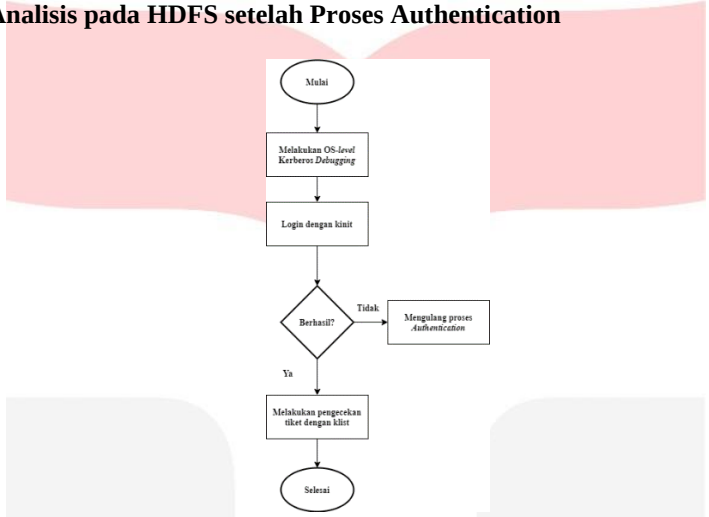
3. Melakukan pengecekan klist yang memiliki arti *list cached* Kerberos *ticket*, yang memiliki informasi dari sebuah tiket yaitu meliputi *Valid starting, Expires, Service principal*.

```
root@hadoop-master:/home/hadoopuser# export KRB5_TRACE=/tmp/kinit.log
root@hadoop-master:/home/hadoopuser# kinit -kt hadoopuser.keytab hadoopuser/hadoop-master@HDREALM
.COM
root@hadoop-master:/home/hadoopuser# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: hadoopuser/hadoop-master@HDREALM.COM

Valid starting       Expires              Service principal
07/28/2021 15:37:30  07/29/2021 01:37:30  krbtgt/HDREALM.COM@HDREALM.COM
        renew until 07/29/2021 15:37:30
```

Gambar 6. Hasil pengujian debugging dan *login* dengan kinit

4.3 Hasil Pengujian Analisis pada HDFS setelah Proses Authentication



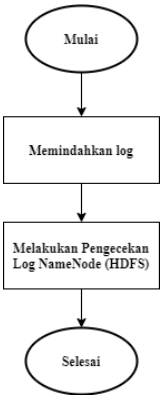
Gambar 7. Diagram Pengujian HDFS setelah Proses Authentication

1. Tahap pertama, pengujian pada HDFS kembali dilakukan lagi setelah berhasil melakukan proses *authentication* pada tahapan sebelumnya dimana pada kondisi kali ini akan baik baik saja karena telah melakukan proses yang telah dilakukan pada tahapan sebelumnya untuk mendapatkan tiket pada suatu layanan pada kali ini yaitu layanan HDFS. Dapat dilihat sekarang direktori HDFS telah dapat dibaca dengan menggunakan *command* hadoop fs -ls /.
2. Selanjutnya pengujian dilanjutkan dengan melakukan percobaan kembali dengan menambahkan direktori baru pada direktori HDFS proses itu gagal, dengan keterangan tidak dapat membuat direktori karena direktori HDFS dalam kondisi *safemode.* Ini terjadi karena pada proses implementasi terdapat proses yang menambahkan variable baru terkait *security* dan kemudian sistem dari Hadoop merespon dengan mengaktifkan *safemode*. Maka dari itu untuk dapat menambahkan direktori baru harus mematikan *safemode* terlebih dahulu dengan menggunakan *command hdfs dfsadmin -safemode leave* yang artinya keluar dari *safemode*. Ketika ingin mengaktifkan kembali dapat dilakukan dengan hdfs dfsadmin -safemode enter dan jika ingin mengecek status dari *safemode* dapat menggunakan *command* hdfs dfsadmin -safemode get.

```
root@hadoop-master:/home/hadoopuser# hadoop fs -ls /
Found 1 items
drwxr-xr-x   - hadoopuser supergroup          0 2020-08-11 01:19 /WordCount
root@hadoop-master:/home/hadoopuser# hadoop fs -mkdir /testKerberos
mkdir: Cannot create directory /testKerberos. Name node is in safe mode.
root@hadoop-master:/home/hadoopuser# hdfs dfsadmin -safemode get
Safe mode is ON
root@hadoop-master:/home/hadoopuser# hdfs dfsadmin -safemode leave
Safe mode is OFF
root@hadoop-master:/home/hadoopuser# hadoop fs -mkdir /testKerberos
root@hadoop-master:/home/hadoopuser# hadoop fs -ls /
Found 2 items
drwxr-xr-x   - hadoopuser supergroup          0 2020-08-11 01:19 /WordCount
drwxr-xr-x   - hadoopuser supergroup          0 2021-07-28 15:41 /testKerberos
root@hadoop-master:/home/hadoopuser#
```

Gambar.8 Hasil Pengujian HDFS setelah *Proses Authentication*

4.4 Hasil Pengujian Analisis Pengecekan *Log*



Gambar 9. Diagram Pengecekan *Log*

1. Memindahkan *log* ke direktori baru agar lebih mudah untuk diakses, atau dapat juga langsung membaca pada direktori yang ada dengan menggunakan *text editor*.
2. Proses terakhir yaitu pengecekan Log pada NameNode (HDFS). Untuk mendapatkan log hanya perlu membuka dimana direktori log berada. Pada penilitian ini semua log hadoop tersimpan dalam direktori /usr/local/hadoop/logs. Lebih lengkap nya terkait log akan dilampirkan pada halaman lampiran.

```
hadoopuser@hadoop-master:~$ cp /usr/local/hadoop/logs/hadoop-hadoopuser-namenode-hadoop-master.lo
g /home/hadoopuser/logupdate/
hadoopuser@hadoop-master:~$ cp /usr/local/hadoop/logs/hadoop-hadoopuser-resourcemanager-hadoop-ma
ster.log /home/hadoopuser/logupdate/
hadoopuser@hadoop-master:~$ cp /usr/local/hadoop/logs/hadoop-hadoopuser-secondarynamenode-hadoop-
master.log /home/hadoopuser/logupdate/
hadoopuser@hadoop-master:~$
```

Gambar 10. Memindahkan *log*

Table 2. Sebagian dari *log* yang didapat

STARTUP_MSG: build = https://gitbox.apache.org/repos/asf/hadoop.git -r b3cbbb467e22ea829b3808f4b7b01d07e0bf3842; compiled by 'rohithsharmaks' on 2019-09-10T15:56Z STARTUP_MSG: java = 1.8.0_265 **/ 2021-07-28 14:04:48,739 INFO org.apache.hadoop.hdfs.server.namenode.NameNodeUtils: fs.defaultFS is hdfs://hadoop-master:9000 2021-07-28 14:04:48,740 INFO org.apache.hadoop.hdfs.server.namenode.NameNode: Clients should use hadoop-master:9000 to access this namenode/service. 2021-07-28 14:04:49,549 INFO org.apache.hadoop.security.UserGroupInformation: Login successful for user hadoopuser/hadoop-master@HDREALM.COM using keytab file /usr/local/hadoop/conf/hadoopuser.keytab 2021-07-28 14:04:51,355 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: fsOwner = hadoopuser/hadoop-master@HDREALM.COM (auth:KERBEROS) 2021-07-28 14:04:51,355 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: supergroup = supergroup 2021-07-28 14:04:51,355 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: isPermissionEnabled = true 2021-07-28 14:15:35,623 INFO org.apache.hadoop.hdfs.StateChange: STATE* Safe mode is OFF 2021-07-28 14:15:35,623 INFO org.apache.hadoop.hdfs.StateChange: STATE* Leaving safe mode after 642 secs 2021-07-28 14:05:56,946 INFO SecurityLogger.org.apache.hadoop.ipc.Server: Auth successful for hadoopuser/hadoop-master@HDREALM.COM (auth:KERBEROS) 2021-07-28 14:05:57,022 INFO SecurityLogger.org.apache.hadoop.security.authorize.ServiceAuthorizationManager: Authorization successful for hadoopuser/hadoop-master@HDREALM.COM (auth:KERBEROS) for protocol=interface org.apache.hadoop.hdfs.server.protocol.NamenodeProtocol

5. Kesimpulan

Berdasarkan analisis yang telah di lakukan tanpa ada nya autentikasi dari Kerberos *cluster* Hadoop menjadi sangat tidak aman apabila hanya dengan menggunakan *default configuration*. Untuk itu maka perlu mengimplentasi Kerberos karena Kerberos telah menyediakan lapisan autentikasi yang aman dan terverifikasi.

Mekanisme dalam pemengamanan hadoop pada sisi *authentication* dapat dilakukan oleh Kerberos. Kerberos memiliki mekanisme authentikasi dari pihak ketiga dimana pengguna dan layanan bergantung pada pihak ketiga yaitu *server* Kerberos untuk dapat mengauthentikasi satu sama lain yang dikenal sebagai *Key Distributed Center* (KDC). Semua bukti terkait informasi pengamanan dari kinerja sistem dicatat dan disimpan dalam *log*.

Referensi:

[1] M. R. Ghazi and D. Gangodkar, “Hadoop, mapreduce and HDFS: A developers perspective,” *Procedia Comput. Sci.*, vol. 48, no. C, pp. 45–50, 2015, doi: 10.1016/j.procs.2015.04.108.

[2] D. Yadav, D. H. Maheshwari, and D. U. Chandra, “Big Data Hadoop: Security and Privacy,” *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3350308.

[3] S. Sinha, S. Gupta, and A. Kumar, “Emerging Data Security Solutions in HADOOP based Systems: Vulnerabilities and Their Countermeasures,” *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, vol. 2019-Janua, pp. 235–240, 2019, doi: 10.1109/ICCCIS48478.2019.8974535.

[4] B. Rahardjo, *Keamanan Sistem Informasi Berbasis Internet*, vol. 0, no. April. 2005.