**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Radite Bayu Prakoso**

**NRP : 5110 100 040**

**DOSEN WALI : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.**

**DOSEN PEMBIMBING : 1. Royyana Muslim Ijtihadie, S.Kom., M.Kom. Ph.D.**

**2. Ary Mazharuddin Shiddiqi S.Kom., M.Comp.Sc.**

# JUDUL TUGAS AKHIR

**“Implementasi *Progressive Caching* pada *Content-Centric Networking* untuk *Streaming* Video”**

# LATAR BELAKANG

Pada saat ini kebutuhan akan *streaming* merupakan salah satu kebutuhan vital dari masyarakat. Kebutuhan akan informasi *realtime* yang berupa berita menjadi kebutuhan yang dirasa semakin penting dalam masyarakat. Namun, dengan paradigma jaringan konvensional pada saat ini, *live streaming* mengalami masalah yang cukup pelik. Performa dari *streaming server* akan berkurang secara drastis seiring dengan banyaknya pengguna yang mengakses *server* pada saat yang bersamaan. Hal ini menjadi masalah yang cukup serius mengingat banyaknya jumlah pengguna internetpada saat ini.

*Content-Centric Networking* (CCN) merupakan sebuah paradigma baru dalam arsitektur jaringan komputer. Pengguna tidak perlu tahu lokasi fisik dari sebuah konten*,* tapi pengguna hanya perlu tahu kontenapa yang diinginkan.

*Progressive caching*  merupakan sebuah metode *caching* dalam CCN. Pada jaringan CCN akan dilakukan *caching* dengan algoritma tertentu. *Progressive caching* memiliki performa yang lebih baik dari algoritma CCN pada awalnya [1].

VLC Media Player adalah perangkat lunak *open source* yang digunakan untuk memutar media, baik video maupun audio dan juga sebagai *streaming server* [2].

Pada Tugas Akhir ini, akan dibuat sebuah VLC *streamer* sebagai *server* yang akan dikoneksikan dengan *host* melalui jaringan CCN.

# RUMUSAN MASALAH

Berikut beberapa hal yang menjadi rumusan masalah dalam Tugas Akhir ini:

1. Bagaimana menerapkan CCNpada jaringan yang sudah ada?
2. Bagaimana menerapkan metode *progressive caching* pada jaringan CCN?
3. Bagaimana *host* dapat berkomunikasi dengan jaringan CCN dan menjalankan aplikasi *streaming*?

# BATASAN MASALAH

Batasan masalah pada Tugas Akhir ini adalah sebagai berikut:

* 1. CCNdan metode *progressive caching* hanya dijalankan pada lapisan *application.*
  2. Video yang digunakan memiliki ekstensi .avi.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari Tugas Akhir ini adalah menghasilkan sebuah aplikasi *live streaming* beserta sebuah jaringan dengan arsitektur CCN*.*

# MANFAAT TUGAS AKHIR

Dengan dibuatnya Tugas Akhir ini maka masyarakat dapat mengakses aplikasi *streaming* dengan performa yang terjamin.

# TINJAUAN PUSTAKA

* 1. ***Content-Centric Networking***

*Content-Centric Networking* (CCN) adalah sebuah paradigma arsitektur jaringan. Pengguna tidak perlu tahu lokasi fisik dari sebuah konten*,* tapi pengguna hanya perlu tahu kontenapa yang diinginkan [1]. Salah satu keunggulan utama dari CCN adalah efisiensi distribusi data. Pada CCN, data akan disebar pada *node-node* yang terdapat dalam jaringan dalam bentuk *cache* (*in-network storage*) sebagai potongan-potongan data dengan panjang data yang sama (*chunk*). Pada metode *progressive caching*, *request* pada suatu *node* untuk suatu *content* yang berasal dari *host* yang berbeda (*request aggregation*) akan dikumpulkan pada *Pending Interest Table* (PIT). Kemudian, *request* akan diteruskan pada *node* selanjutnya hingga kontenyang diinginkan dapat ditemukan. Kontenyang telah ditemukan akan diteruskan hingga ke *node* di mana *request aggregation* terjadi. Pada *node* tersebut, akan dilakukan *caching* dan *content* akan diperbanyak sejumlah *request* dari masing-masing *host* yang tadinya dikumpulkan dalam *request aggregation* [1].

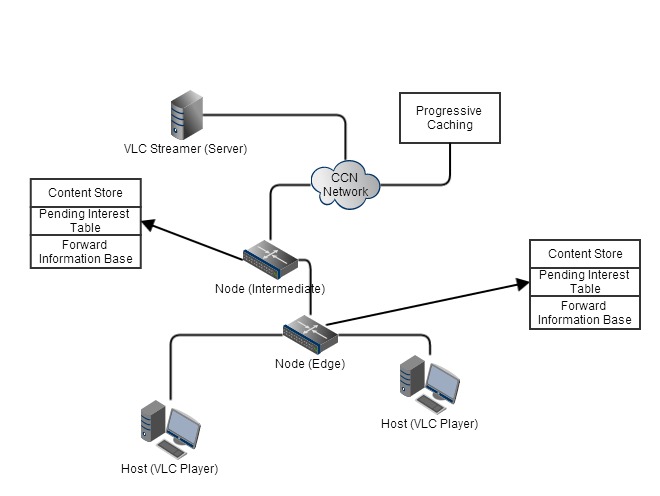
* 1. **VLC Media Player**

VLC Media Player adalah perangkat lunak *open source* yang digunakan untuk memutar media, baik video maupun audio dan juga sebagai *streaming server*. VLC dikembangkan oleh VideoLAN. VLC dapat memutar berkas audio dan video dengan berbagai macam metode kompresi maupun sebagai protokol yang digunakan untuk *streaming* video [2].

# RINGKASAN ISI TUGAS AKHIR

Tugas Akhir ini akan membuat sebuah klien pada perangkat untuk mengakses *streaming* menggunakan arsitektur jaringan CCN. *Host* akan meminta video dengan *name* tertentu. Kemudian, perangkat akan mengirimkan paket *interest* ke dalam jaringan CCN [3].

Pada jaringan CCN, *node* yang menerima paket *interest* akan mengecek apakah *node* tersebut memiliki konten yang sesuai dengan yang diinginkan oleh paket *interest* pada *content store*. Jika *node* tersebut memiliki konten yang diinginkan, maka secara langsung *node* tersebut akan mengirimkan konten menuju asal dari paket *interest*. Jika tidak, maka *node* tersebut akan mengecek *Pending Interest Table* (PIT) jika ada permintaan untuk konten yang sama. Jika ada, maka PIT akan mengumpulkan semua permintaan yang serupa dan melanjutkan permintaan ke *node* selanjutnya. Jika *node* menerima konten yang diinginkan, maka *node* akan menyebarkan konten berdasarkan semua *host* yang meminta konten tersebut berdasarkan data pada PIT [3]. Model *forwarding* beserta arsitektur jaringan dari CCN secara garis besar dijelaskan pada Gambar 1.



Gambar 1. Model *forwarding* CCN

# METODOLOGI

## Penyusunan proposal Tugas Akhir

Proposal Tugas Akhir ini berisikan mengenai apa saja yang dibutuhkan, serta rumusan masalah yang ada dalam perancangan arsitektur CCN beserta implementasi klien untuk *streaming*.

## Studi literatur

Tugas Akhir ini menggunakan literatur *paper* beserta artikel dari internet. *Paper* yang digunakan adalah *“Live Streaming with Content Centric Networking”* dan *“Progressive Caching in CCN”* yang merupakan acuan utama dan dasar dalam pengerjaan Tugas Akhir ini.

## Analisis dan desain perangkat

Sebelum membahas lebih jauh mengenai arsitektur jaringan untuk CCN, ada dua istilah utama yang perlu dipahami.

* *ccnd*

*ccnd* adalah sebuah CCN *routing daemon*. Masing masing *node* dari CCN harus menjalankan *ccnd.*

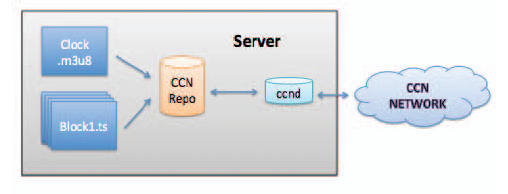
* *ccn-repo*

Sebuah aplikasi berbasis CCN yang digunakan sebagai sarana penyimpanan data.

Pada penerapannya, arsitektur CCN untuk *streaming* memiliki dua elemen utama dari CCN yakni pada *client-end* dan ­*server-end.*

1. Arsitektur *server-end*

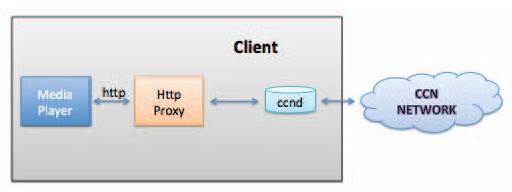
Sama seperti HTTP *streaming* pada umumnya, masukan video dipecah menjadi beberapa bagian dan akan dibangkitkan berkas indeks sebagai daftar putar yang digunakan untuk menyusun kembali masukan video yang telah dipecah. Arsitektur *server-end* dari jaringan CCN untuk *streaming* dijelaskan pada Gambar 2.



Gambar 2. *Server-end architecture*

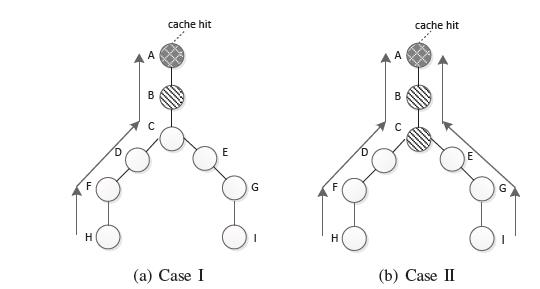
1. Arsitektur *client-end*

Implementasi pada sisi klien dijalankan pada perangkat Android yang berbasis perangkat lunak. Pada perangkat lunak disertakan juga *media player* yang mendukung HTTP *streaming.* HTTP *proxy* akan menerjemahkan permintaan HTTP menjadi permintaan CCN. Lalu, *ccnd* akan meneruskan permintaan yang berbentuk paket *interest*. Kemudian, *proxy* akan menerjemahkan respon CCN menjadi respon HTTP yang akan dikirim kembali ke *media player*. Arsitektur *client-end* dari jaringan CCN untuk *streaming* dijelaskan pada Gambar 3.



Gambar 3. *Client-end architecture*

Setelah itu akan diterapkan metode *progressive caching* pada masing-masing *node* pada jaringan CCN. Pada saat sebuah paket *interest* mengalami *hit* (konten yang diinginkan berhasil ditemukan pada *node* yang dilewati paket *interest*) maka akan dilakukan proses *caching* pada *node* di bawahnya. Adapun mekanisme *caching* dijelaskan pada Gambar 4.



Gambar 4. Mekanisme *caching*

## Pada *progressive caching*, terdapat perbedaan algoritma *caching* pada *edge node* dan *intermediate node*. *Edge node* merupakan *node* yang berbatasan langsung dengan *host.* Sedangkan *intermediate node* merupakan *node* perantara dari *edge node* dengan *node* yang mengalami *hit*. Adapun perbedaan antara *edge node* dan *intermediate node* akan dijelaskan sebagai berikut:

* *Edge node*

Pada *edge node*, konten yang sering diakses akan mengalami kenaikan nilai prioritas. Sehingga kemungkinan konten tersebut akan dihapus akan semakin kecil. Algoritma untuk *edge node* dijelaskan pada Gambar 5.

Gambar 5: Algoritma *caching* pada *edge node*

Caches are organized into *multiple* queues: Q[K]

Initially *L 🡨* 0

The current request matches chunk *p;*

**IF** *p* is already in cache space

Remove it from current queue Q[*Hp*];

Append it to tail of Q[*H’p*] where *H’p = L + cp;*

**IF** *p* is nor in the cache space

**IF** eviction is needed:

Let *L 🡨* arg minQ[*k*]≠ Ø;

Evict data chunk *q* from the head of Q[L]

*i.e.,* the queue with *Hq = L;*

append *p* to the tail of Q[*Hp*], where *Hp = L + cp*

* *Intermediate node*

Pada *intermediate node,* konten yang sering diakses oleh *host* yang sama akan mengalami pengurangan nilai prioritas karena diasumsikan bahwa konten tersebut telah mengalami proses *caching* pada *node* di bawahnya. Jika konten sering diakses oleh banyak *host,* maka hal ini menandakan bahwa konten tersebut populer dan untuk menjamin ketersediaan konten, maka nilai prioritasnya akan dinaikkan. Hal ini akan memperkecil peluang konten tersebut untuk dihapus. Algoritma akan dijelaskan pada Gambar 6.

Gambar 6. Algoritma *caching* pada *intermediate node*

The current request matches chunk data *p*:

**IF** *p* is already in the cache space,

*f:* the request incoming interface;

**IF** *f*  Sp1,

Set *Hp 🡨 Hp - cp ,* if *f* Sp2:Sp2=Sp2;

**ELSE IF** *f*  Sp1

*Hp 🡨 Hp + cp;* Sp1 = Sp1

**IF** *p* is not in the cache space,

**IF** eviction is needed:

Evict the chunk arg min*Hv;*

Select an item *q* randomly: *Hq 🡨 [Hq-cq]+;*

Promote *p* from the buffer space to the cache space;

Set *Hp 🡨 dp x cp;* initialize Sp1; Sp2 = Ø

## Implementasi perangkat

Implementasi merupakan tahap untuk membangun metode tersebut. Untuk mengimplementasikan metode tersebut digunakan bahasa pemrograman Java.

## Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian dan pencatatan hasil kinerja perangkat, serta mencatat kendala-kendala apa saja yang dapat mempengaruhi kinerja perangkat ini.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi perangkat yang telah dibuat. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

Berikut ini adalah jadwal pengerjaan Tugas Akhir yang dijelaskan pada Tabel 1.

Tabel 1. Jadwal kegiatan pembuatan Tugas Akhir

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2013/2014 | | | | | | | | | | | | | | | | | | | | |
| Maret | | | | April | | | | Mei | | | | Juni | | | | | Juli | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | B. B. Jason Min Wang, "Progressive Caching in CCN," in *Global Communication Conference (GLOBECOM) 2012 - Next Generation Networking and Internet Symposium*, 2012. |
| [2] | VideoLAN, "http://videolan.org," [Online]. Available: https://videolan.org/vlc/features.php. [Accessed 15 3 2014]. |
| [3] | H. Xu, Z. Chen, R. Chen and C. Junwei, "Live Streaming with Content Centric Networking," in *Third International Conference on Networking and Distributed Computing*, 2012. |