

BAB 1

USULAN GAGASAN

1.1 Latar Belakang Masalah

Dengan pesatnya perkembangan lalu lintas data pada *internet* maka volume dan juga variasi konten terus bertambah, dan *Cisco* (2020) mencatat bahwa lalu lintas *IP* meningkat dengan sangat pesat menjelang tahun 2023, dan diprediksi bahwa akan lebih banyak lalu lintas *IP* pada tahun 2023, yang mana 66 persen populasi global akan menjadi pengguna *internet*, dan konten video akan menghasilkan 82 persen dari semua lalu lintas. (*IP* Permintaan request yang sama pada satu konten secara berulang juga akan menjadikan kurangnya efisiensi pada jaringan tersebut, karena memberikan beban lalu lintas pada tautan *backhaul*).

Dengan meningkatnya penggunaan *internet*, jumlah penayangan konten video mengalami pertumbuhan secara pesat. Namun, jumlah untuk mengakses halaman situs menurun seiring dengan waktu responnya karena penundaan 1 detik menyebabkan pengurangan 7% [1]. Perkiraan indeks jaringan *visual cisco* menjelaskan bahwa lalu lintas konten video saja akan mencapai 82% dari penggunaan *internet* global pada tahun 2023 [1].

Lalu lintas *internet* juga mengalami pertumbuhan secara signifikan dalam dua dekade terakhir karena meningkatnya jumlah pengguna *internet* yang menuntut layanan berkualitas tinggi, meluasnya penggunaan media sosial dengan konten video, dan perkembangan website yang canggih dan interaktif. *Website* telah menjadi salah satu elemen utama dalam ekosistem *internet*, menyediakan platform bagi individu dan bisnis untuk berbagi informasi, berinteraksi, dan mengakses berbagai layanan secara *online*. Kemajuan dalam desain dan fungsionalitas *website*, termasuk penggunaan teknologi seperti HTML5, CSS3, dan JavaScript, telah memungkinkan pengembangan *website* yang lebih menarik, responsif, dan memungkinkan pengguna untuk berpartisipasi secara aktif. Dengan demikian, *website* telah menjadi pusat komunikasi dan transaksi *online* yang semakin penting dalam kehidupan sehari-hari pengguna internet. Karena pertumbuhan jaringan yang sangat cepat, pengguna memiliki permintaan yang lebih tinggi untuk layanan baru dengan *delay* yang rendah, konsumsi daya sangat kecil, dan keandalan yang tinggi, sehingga membuat arsitektur jaringan komunikasi yang ada sangat terpengaruh dan dibatasi serta pekerjaan pemeliharaan bidang jaringan komunikasi menghadapi persyaratan yang lebih ketat.

Selama beberapa dekade terakhir, *Internet* telah mengubah masyarakat dan ekonomi modern. Hal ini telah mengubah cara orang berkomunikasi satu sama lain dan melakukan

bisnis. *Internet* menciptakan lingkungan global yang menghubungkan orang-orang dari seluruh dunia. Namun, pertumbuhan pesat *World Wide Web* telah menyebabkan kemacetan jaringan dan peningkatan waktu respons bagi pengguna akhir. Dengan kata lain, pengguna seringkali harus menunggu lebih lama untuk mendapatkan informasi yang mereka butuhkan. Saat ini, *Website* membutuhkan lebih banyak *bandwidth*, yang membuat Penyedia Layanan Internet (ISP) harus menyediakan *bandwidth* yang memadai untuk memenuhi kebutuhan pengguna akhir. Orang-orang membaca berita, mengirim/menerima email, mencari informasi, berbelanja *online*, dan menonton video melalui *web*, sehingga memiliki akses *web* yang baik menjadi sangat penting. Menambahkan lebih banyak *bandwidth*, meningkatkan kekuatan pemrosesan, dan menerapkan mekanisme lain untuk meningkatkan kualitas infrastruktur *internet* adalah solusi potensial untuk masalah kinerja yang dihadapi.

Halaman dan gambar *cache* yang diakses pengguna melalui *browser Google Chrome* disimpan dalam folder yang bernama "*Cache*" yang berada di dalam jalur profil pengguna. *Google Chrome* menyimpan data *browsing* di berbagai file seperti *Bookmark*, *Cookies*, *Current Tabs*, *History*, *Last Session*, *Last Tabs*, *Preferences*, *Visited Links*, dan *Top Sites* [2]. Informasi ini tersimpan dalam file terpisah yang dibuat di profil pengguna. Berlandaskan laporan *cisco* mengenai lalu lintas jaringan *IP* yang semakin padat, maka *google* menyediakan teknologi yang bernama *Google Global Cache (GGC)* yang mana teknologi ini memungkinkan *ISP* menyajikan konten *google* dari dalam jaringan mereka sendiri sehingga memungkinkan 70-90% lalu lintas yang dapat di *cache* untuk memastikan bahwa pengguna dilayani dari lokasi data terdekat ("google",n.d.).

1.2 Informasi Pendukung Masalah

Dalam 1 dekade terakhir, beberapa platform *online* telah menjadi hal yang populer. Berbagai platform media sosial seperti *youTube*, *facebook*, dan *twitter* telah memungkinkan pengguna untuk memposting konten mereka ke berbagai komunitas. *Netflix* dan *youtube* adalah situs pengunggahan video terkemuka dengan lebih dari satu miliar konsumen yang menggunakan *youtube* [1]. Dengan meningkatnya penggunaan ponsel cerdas, konten buatan pengguna telah berkembang pesat.

Jumlah video yang diunggah ke *youtube* terus meningkat, sehingga hampir tidak mungkin untuk menyimpan semua kontennya ke dalam *cache*. Itulah mengapa penting untuk memutuskan video mana yang akan di-*cache*. Selain itu, sejumlah besar pengguna hanya

menonton beberapa video populer yang menyebabkan peningkatan lalu lintas, sementara sebagian besar konten jarang dilihat. Oleh karena itu, tren yang memprediksi popularitas video dapat membantu mengidentifikasi konten yang harus disimpan dalam *cache*. Memprediksi penayangan video dapat berguna untuk berbagai alasan bisnis seperti pemasaran dan periklanan produk.

1.3 Analisis Umum

Caching konten terdistribusi dapat mengurangi beban pada backhaul *server* 1/3 hingga 2/3 volume data pada jaringan [3]. Dengan pengimplementasian menggunakan metode *Caching Content*, mengakses *internet* dapat dilakukan dengan lebih efisien dan efektif untuk mengurangi beban *server* agar performa *internet* lebih cepat. Maka dari itu, pengimplementasian *IP Edge Caching* bermanfaat dalam aspek - aspek sebagai berikut:

1.3.1 Aspek Teknologi

IP address adalah metode pengalamatan pada jaringan komputer agar dapat terhubung dan mengakses *internet*. Pengimplementasian *IP Edge Caching* pada *internet* skala lab dilakukan dengan tujuan mengurangi beban *server* dan untuk meningkatkan kualitas internet agar dapat digunakan secara optimal.

Penempatan konten yang optimal di *node* jaringan diperlukan untuk meningkatkan kinerja *cache* jaringan. Strategi penempatan konten berbasis popularitas dapat mengatasi faktor *caching* seperti duplikat konten, kategori konten, dan popularitas konten. Keuntungan menyimpan konten adalah konten yang diambil lebih cepat dan efisien, sehingga tidak perlu meminta konten terus-menerus.

1.3.2 Aspek Ekonomi

IP Edge Caching dapat mengoptimalkan kebutuhan *load traffic* jaringan, maka *IP Edge Caching* dapat bermanfaat dalam aspek ekonomi untuk menghemat biaya produksi penyimpanan konten *caching*. Keuntungan utama dan kesederhanaan penerapan *cache* terpusat mampu mengurangi *bandwidth* dan biaya koneksi *internet*.

1.4 Kebutuhan yang Harus Dipenuhi

Berdasarkan analisis yang telah dilakukan, *caching* konten sebaiknya memenuhi beberapa aspek berikut ini:

1.4.1 Biaya

Dengan banyaknya kebutuhan dalam membuat sistem popularitas konten *caching*, maka biaya yang dikeluarkan ditargetkan berbanding lurus dengan performa sistem yang dibuat.

1.4.2 Load Traffic

Untuk meningkatkan kualitas trafik jaringan perlu memperhatikan beberapa aspek seperti *delay* yang rendah dan *bandwidth* yang besar sehingga beban lalu lintas dapat terdistribusi secara optimal.

1.5 Solusi Sistem yang Diusulkan

A. Content Centric Network (CCN)

Content Centric Networking (CCN) adalah arsitektur yang diusulkan untuk memecahkan masalah distribusi data saat ini di internet. CCN ini sudah berdasarkan content-centric atau pencarian berdasarkan nama konten.

CCN memiliki dua jenis paket pengiriman, yaitu paket interest atau permintaan dan paket berisi data atau konten. CCN juga memiliki 3 struktur data yaitu *Content Store* (CS), *Pending Interest Table* (PIT) dan *Forwarding Information Base* (FIB) [4].

Pengoperasian CCN sendiri dalam mencari informasi konten adalah ketika pengguna meminta informasi, paket *interest* mengirimkan paket yang diinginkan ke *router*. *Router* melakukan pencarian di CS apakah ada data yang di *cache* atau tidak. Jika ada, data dikirim ke pengguna yang meminta tanpa harus melalui ke server.

B. Information Centric Network (ICN)

ICN adalah model berdasarkan penamaan konten untuk menghilangkan batasan ruang alamat, akses ke konten melalui perutean berbasis nama dan *caching* konten di *node* perantara untuk memastikan transfer data yang andal dan efisien, dan kontrol mandiri konten untuk memastikan keamanan yang lebih baik [5].

Menurut skema perutean, arsitektur ini dapat dibagi menjadi dua kategori. Salah satunya adalah pendekatan perutean berbasis nama, yang diwakili oleh ICN. Dalam pendekatan ini, permintaan dialihkan berdasarkan nama konten. Yang lainnya adalah

pendekatan resolusi nama, di mana nama konten diselesaikan menjadi satu atau satu set alamat jaringan (misalnya IP), dan kemudian permintaan dialihkan ke salah satu alamat jaringan ini. Arsitektur representatif dari kategori ini termasuk *DONA*.

C. *CAIP*

CAIP adalah arsitektur baru yang memungkinkan caching di jaringan berbasis *IP* dengan menambahkan pengidentifikasian konten ke *header* ekstensi *IPv6* yang baru ditentukan. *CAIP* melewati tabel perutean berbasis nama yang kompleks di *ICN*, tetapi mengintegrasikan pencarian rute *IP* ke dalam *cache* pencarian indeks yang kompatibel dengan jaringan *IP*.

D. *Proxy Server*

Proxy Server merupakan sebuah komputer atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan request terhadap konten dari *internet*. Dalam jaringan *proxy server* dapat dilakukan mekanisme *caching*, sehingga beban pada *server* bisa berkurang.

Mekanisme *caching proxy* merupakan teknologi *caching web* utama yang mampu melayani permintaan *web* pengguna dari jaringan *proxy* yang terletak di antara pengguna akhir dan *server web* yang menampung salinan asli dari objek yang diminta.

E. *IP Caching Network (IPCN)*

IPCN-X diusulkan dengan latar belakang pengurangan tekanan beban *Origin Server* (*OS*) dan dipelajari pada skenario di mana semua node jaringan tidak termasuk *Origin Server* (*OS*) yang dibagi menjadi beberapa *Autonomous System* (*AS*) yang tidak tumpang tindih. Perpotongan antara dua *Autonomous System* (*AS*) adalah kumpulan perangkat kosong dan informasi terkait *Autonomous System* (*AS*) dikelola oleh *Information Management Center* (*IMC*) [6].

F. *Content Delivery Network (CDN)*

CDN adalah sistem yang saling terhubung di internet yang menyediakan konten aplikasi *internet* ke banyak pengguna dengan menduplikasi konten di beberapa *server* dan mengarahkan konten ke pengguna. *CDN* digunakan oleh penyedia layanan *internet* (*ISP*) untuk mengirimkan halaman web statis atau dinamis [7].

1.5.1 Karakteristik Produk

1.5.1.1 *Content Centric Networking (CCN)*

A. Kelebihan

- Memiliki model penghapusan (*cache replacement*) dan model penyimpanan cache (*cache strategy*) atau bisa disebut *in-network caching* [4].
- *Cache* konten bisa langsung di simpan pada *router* CCN.
- Fleksibel, karena menggunakan nama untuk berkomunikasi.
- Memiliki fitur seperti *content distribution, multicast, mobility, dan delay-tolerant networking*.

B. Kekurangan

- *Security*: Karena *caching* konten yang ada di mana-mana dan tidak adanya pipa komunikasi yang aman dengan *host* akhir maka perlindungan konten terhadap akses tidak *secure*.

1.5.1.2 *Information Centric Network (ICN)*

A. Kelebihan

- ICN meningkatkan efisiensi dan throughput jaringan secara keseluruhan.
- Meringankan beban jaringan.

B. Kekurangan

- Tidak Fleksibel: ICN memiliki dua jenis paket yaitu *interest* dan data, paket *interest* digunakan untuk meminta suatu konten dengan cara mengirimkan *broadcast* pada jaringan. Sedangkan, paket data mengirimkan balasan konten yang diminta. Dengan hal itu, *bandwidth* yang dibutuhkan semakin besar [8].
- Biaya Mahal: Jumlah objek nama dan objek data konten diharapkan lebih tinggi dari jumlah *node* jaringan. Dengan demikian, melakukan perutean berbasis nama di setiap *node* di jalur antar *node* dapat menuntut biaya, yang mengarah pada peningkatan *bandwidth* dan *delay* yang tinggi. terutama untuk *router* ICN yang dibatasi sumber daya yang terbatas
- Susah diimplementasikan.

1.5.1.3 CAIP

A. Kelebihan

- Dapat mengimplementasikan *caching* dalam jaringan berbasis *IP*.
- Dapat mengurangi beban lalu lintas jaringan.

B. Kekurangan

- Hanya bisa bekerja pada *IPv6*.
- Membutuhkan banyak router karena pada arsitektur *CAIP* ini masih menggunakan router tradisional dan harus disisihkan beberapa router *CAIP* ke dalam arsitektur jaringan.

1.5.1.4 Proxy Server

A. Kelebihan

- Dapat melakukan *caching* pada web.
- Dapat mengurangi beban lalu lintas jaringan.
- Biaya lebih murah karena tidak membutuhkan tempat penyimpanan yang besar.
- Mempercepat akses ke situs lain.
- Dapat melakukan pemblokiran akses ke situs *web*.

B. Kekurangan

- Memiliki kapasitas penyimpanan *cache* terbatas.
- Rentan terhadap serangan.
- Jika tidak di *update*, maka pengguna mungkin menerima data lama.

1.5.1.5 IP Caching Network (IPCN)

A. Kelebihan

- Lebih efektif dan efisien.
- Memberikan keamanan dan kenyamanan.

B. Kekurangan

- Memberikan keamanan dan kenyamanan.
- Memerlukan sumber daya yang banyak.

1.5.1.6 Content Delivery Network (CDN)

A. Kelebihan

- Dapat melayani permintaan dan volume konten dalam jumlah besar [9].

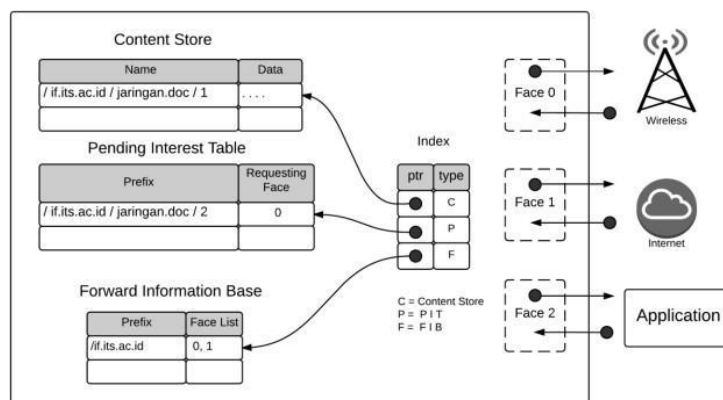
B. Kekurangan

- Biaya mahal [9].
- Tidak optimal karena *server* akan berisi konten secara berlebihan [9].
- Susah untuk diimplementasikan.

1.5.2 Skenario Penggunaan

1.5.2.1 Content Centric Network (CCN)

Pada model jaringan CCN, *node* yang menghasilkan konten disebut dengan *publisher*, dan *node* yang meminta konten disebut dengan konsumen. Pada Gambar 1.1 menjelaskan tentang Skema *forwarding*. Mesin *forwarding* yang ada pada setiap *node* jaringan CCN memiliki 3 komponen utama: *Forwarding Interest Base (FIB)*, *Content Store (CS)* dan *Pending Interest Table (PIT)* [10].



Gambar 1.1 Skenario CCN

- FIB digunakan untuk melakukan proses *forwarding* paket *interest* menuju ke lokasi yang berpotensi memiliki sumber data yang diminta.
- *Content Store* merupakan memori *buffer* untuk menyimpan data seperti pada *router* jaringan *IP*, tetapi memiliki metode *replacement* yang berbeda.
- PIT adalah tabel yang terdapat pada *node* CCN, yang mencatat setiap *face* tempat datangnya paket *interest*.

1.5.2.2 Informatic Centric Network (ICN)

A. Menerima Interest

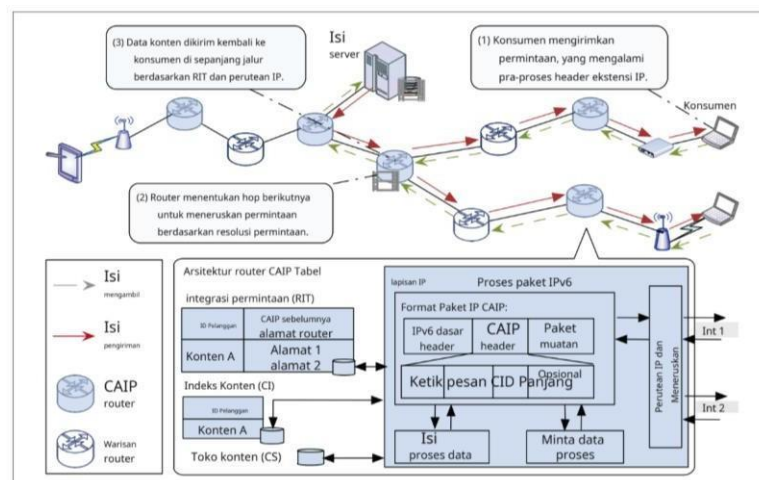
- Mencari *interest* pada *Content Store* (CS) jika konten ditemukan pada CS, *node* akan mengembalikan objek ke *hop* sebelumnya, lalu membuang *interest*.
- Mencari *interest* pada PIT, jika tidak ada kecocokan, *node* akan menambahkan bagian awal dari *interest* yang datang ke PIT. Jika masih tidak ada kecocokan maka ditambahkan ke PIT [11].
- Jika *interest* tidak ditemukan pada PIT, kemudian cari nama *interest* pada FIB. Jika nama *interest* ditemukan, teruskan *interest* ke hasil set dari bagian pertama. Jika tidak, bagian pertama akan hilang pada *interest*, pindah kembali ke antrian PIT [11].

B. Menerima Konten

- Pencarian pada konten pada PIT. Jika kecocokan ditemukan *nodenya* akan meneruskan konten ke semua pengguna yang cocok pada antrian PIT, kemudian buang antrian PIT yang sesuai. Jika tidak konten harus diturunkan [11].
- Simpan konten pada *Content Store*.

1.5.2.3 CAIP

Arsitektur CAIP memanfaatkan identifikasi konten dalam header ekstensi IPv6 baru dapat membantu jaringan IP saat ini menjadi sadar konten. Setiap *router CAIP* berisi tiga lebih banyak komponen terkait manajemen konten daripada yang lama: tabel integrasi permintaan (RIT), indeks konten (CI), dan penyimpanan konten (CS). Model penerusan *router CAIP* sama dengan *router IP* lama, yang didasarkan pada tabel alamat IP dan tabel penerusan.



Gambar 1.2 Skenario CAIP

- Pra-Proses *Header Ekstensi IPv6*

Menentukan apakah paket *IP* memiliki *header* ekstensi *IPv6* dengan pengidentifikasian konten atau tidak.

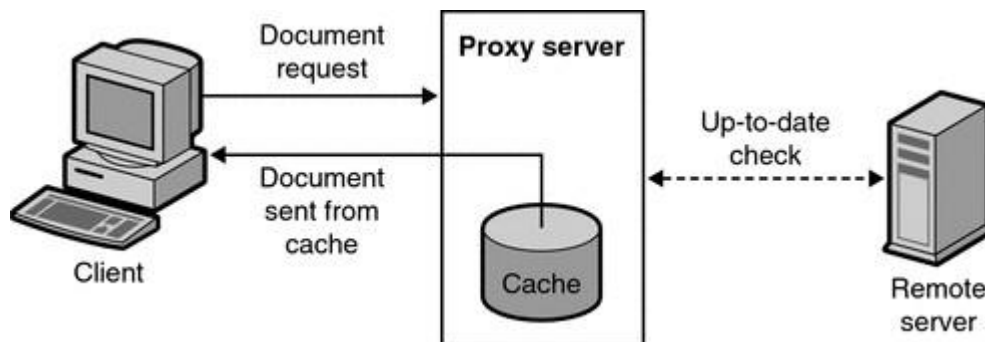
- Resolusi Permintaan

Untuk paket dengan dukungan *header* ekstensi *IP*, *router CAIP* akan memeriksa apakah paket tersebut memiliki konten yang diminta terkait dengan pengidentifikasian pada *header*. Jika tidak maka akan ditentukan *hop* selanjutnya menuju *server* tujuan berdasarkan *IP routing* pada Gambar 1.2 [12].

Pengiriman konten merupakan kebalikan dari jalur permintaan berdasarkan status penerusan yang dibuat selama fase permintaan terkait. Untuk mewujudkan transmisi *hop* demi *hop*, paket-paket tersebut harus dirakit kembali terlebih dahulu seperti pada ICN [12].

1.5.2.4 *Proxy Server*

Mekanisme *caching* akan menyimpan obyek-obyek yang merupakan hasil permintaan dari para pengguna, yang didapat dari *internet*. Karena *proxy server* bertindak sebagai perantara, maka *proxy server* mendapatkan obyek-obyek tersebut lebih dahulu dari sumbernya untuk kemudian diteruskan kepada peminta yang sesungguhnya seperti yang ditunjukkan pada Gambar 1.3 yang dimana pada proses tersebut, *proxy server* juga sekaligus menyimpan objek-objek untuk dirinya sendiri dalam ruang *disk* yang disediakan (*cache*) [13].

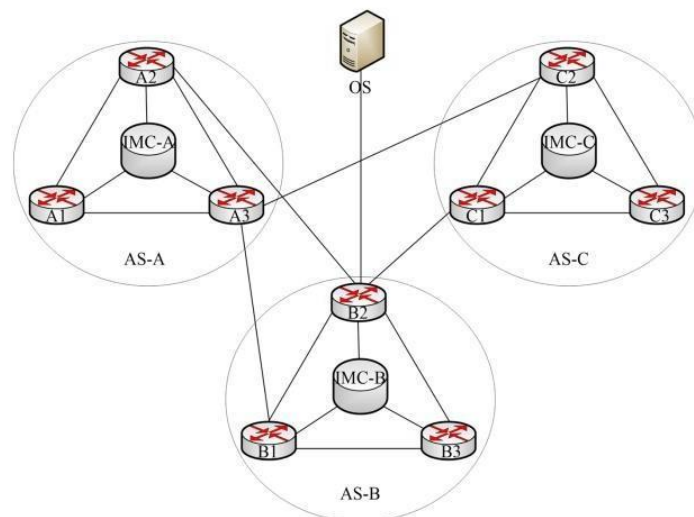


Gambar 1.3 Skenario *Proxy Server*

- Saat pengguna meminta objek *web*, *cache proxy* memproses permintaan tersebut terlebih dahulu.

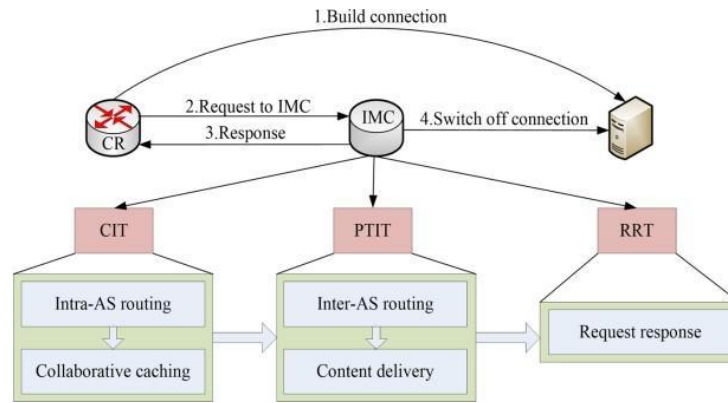
- Jika objek memiliki salinan asli yang disimpan di dalam *cache*, salinan ini segera dikembalikan ke pengguna dan permintaan tidak diteruskan ke server *web* yang *hosting* obyek yang diminta.
- Jika salinan yang di-*cache* tidak terletak di *cache*, maka dapat dikatakan bahwa terjadi *cache miss*.
- Dalam konfigurasi *cache proxy* tunggal, ketika terjadi kesalahan, *server web* asli dihubungi dan *objek web* yang dikembalikan untuk diteruskan ke program klien.
- Dalam konfigurasi yang terdiri dari jaringan *cache proxy*, sebelum tersambung ke *server web* asli, *cache proxy* mengikuti protokol tertentu yang dibuat untuk mengatur permintaan ke *cache* lain yang berpartisipasi dalam upaya menemukan salinan *cache* [13].

1.5.2.5 IP Caching Network (IPCN)



Gambar 1.4 Skenario IPCN

Gambar 1.4 merupakan kerangka sistem IPCN dimana IMC memegang peranan penting dan terdiri dari tiga tabel yaitu CIT, PTIT dan RRT. Selain itu, IPCN terdiri dari lima modul utama, yaitu rute *intra-AS*, *caching* kolaboratif, rute antar-*AS*, pengiriman konten, dan respons permintaan. Diantaranya, dua modul pertama, dua modul berikutnya dan modul terakhir masing-masing dilakukan oleh CIT, PTIT dan RRT [6].



Gambar 1.5 Modul IPCN

- Rute *intra-AS*.3

Pada Gambar 1.5 *Content Router* membangun koneksi dan meneruskan permintaan pengguna ke IMC-nya daripada OS, meskipun koneksi komunikasi ujung-ke-ujung antara pengguna dan OS (misalnya, *HTTP GET*) dibangun. Kemudian, IMC memilih CR untuk menyediakan konten.

- *Cache* kolaboratif

Tidak ada salinan konten yang berlebihan di setiap *Autonomous System (AS)*, sehingga diperlukan *caching* kolaboratif secara eksplisit. Dalam upaya untuk mencapai pengambilan konten dengan latensi rendah, IMC secara berkala mendorong konten ke *Content Router (CR)* yang akan meneruskan jumlah maksimum permintaan yang sama.

- Perutean antar-*AS*

Ketika IMC tidak menemukan CR yang sesuai, itu berarti bahwa konten tidak dapat ditemukan dengan perutean *intra-AS*. Selama proses rute antar-*AS*, IMC bertanggung jawab untuk memilih beberapa CR untuk meneruskan permintaan pengguna ke OS atau IMC lain, yang meliputi (i) pemilihan OS atau IMC dan (ii) penentuan jalur penerusan lokal antara IMC atau antara IMC dan OS.

- Pengiriman konten

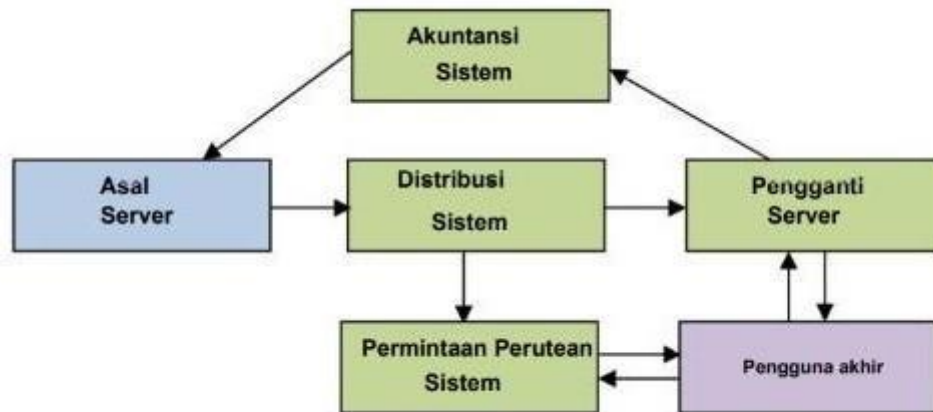
Proses ini terbalik dengan *routing*. IMC atau OS bertanggung jawab untuk memilih beberapa CR untuk mengirimkan konten ke IMC terakhir, di mana jalur pengiriman lokal antara IMC atau antara IMC dan OS ditentukan sementara *caching on-path* atau *caching off-path* dipilih secara adaptif.

- Respon permintaan

IMC dapat menanggapi permintaan pengguna secara langsung, dan secara

berkala mengirimkan informasi yang sesuai ke OS sehingga koneksi komunikasi ujung ke ujung antara pengguna dan OS dimatikan.

1.5.2.6 Content Delivery Network (CDN)



Gambar 1.6 Skenario CDN

- Pada Gambar 1.6 *Server* CDN akan menduplikasikan konten dari *server* utama dan *server* CDN diletakkan lebih dekat ke lokasi pengguna [14].
- Permintaan isi dari *end-user* diarahkan ke *Request Routing System*.
- Sistem *Perutean* Permintaan mengalihkan permintaan ke *server* CDN yang sesuai. *Server* CDN akan mengirimkan data yang sudah tersimpan sehingga proses *loading* akan lebih cepat [14].