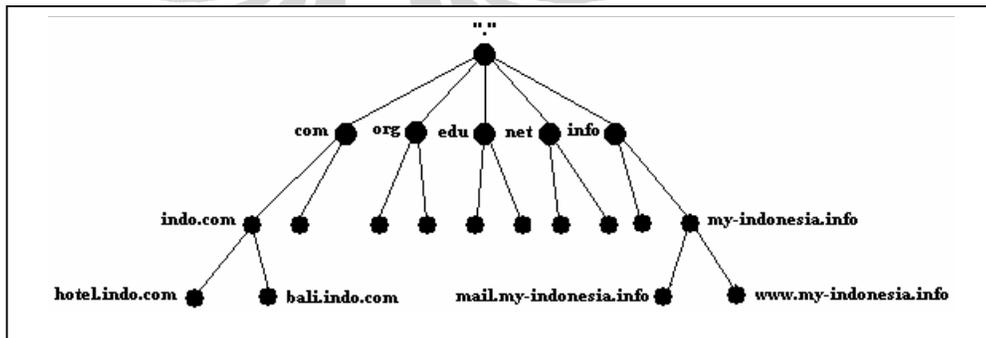


## BAB II LANDASAN TEORI

### 2.1 Domain Name System (DNS)

Pengalamatan di protokol Internet yang menggunakan kombinasi angka cukup sulit untuk diingat. Penggunaan alamat dengan nama akan lebih mudah diingat. Untuk itu diperlukan suatu pemetaan alamat IP Address ke nama host yang ada dan sebaliknya. Pada awalnya digunakan teknik yang dinamakan *host table*. Masing-masing host/komputer menyimpan daftar kombinasi nama komputer dan alamat IP Address, pada file yang dinamakan HOSTS.TXT. File ini berisi nama dan alamat IP seluruh komputer yang terkoneksi ke internet. Tahun 1984, Paul Mockapetris mengusulkan sistem basis data terdistribusi yang dinamakan DNS (*Domain Name System*). spesifikasi asli muncul di RFC 882 dan 883. Tahun 1987, penerbitan RFC 1034 dan RFC 1035 membuat update terhadap spesifikasi DNS [3].

Format penamaan host di Internet dibuat memiliki hirarki. Skema hirarki tersebut digambarkan berbentuk *tree* (pohon). Satu node/titik membentuk *tree*, memiliki beberapa *subnode*. *Subnode* ini membentuk *tree* yang memiliki beberapa *subnode* lagi, dan seterusnya. Pada masing-masing node ini terdapat label. Node berlabel ini disebut *domain*. *Domain* ini bisa berupa nama *host*, *subdomain* atau *top level domain*.



Gambar. 2.1 Model Tree Domain [3]

*Domain* teratas ialah *Root Domain*. Domain ini dituliskan dalam bentuk titik (“.”). *Top Level Domain* terdiri atas semua node yang tepat berada di bawah

root. Pada gambar di atas hal ini ditunjukkan dengan node com, org, edu dan seterusnya. *Subdomain* merupakan kumpulan keturunan *Top Level Domain*. Node yang berada tepat di bawah *Top Level Domain* disebut *Second Level Domain*. Node di bawah *Second Level Domain* disebut *Third Level Domain* dan seterusnya. Sesuai dengan konvensi, label yang menunjukkan domain ditulis dari kiri ke kanan, dipisahkan dengan tanda titik, dengan domain yang paling jauh dari root ditulis terlebih dahulu. Penulisan secara lengkap seperti di atas mulai dari nama host hingga tanda titik yang melambangkan root disebut sebagai *Fully Qualified Domain Name* (FQDN).

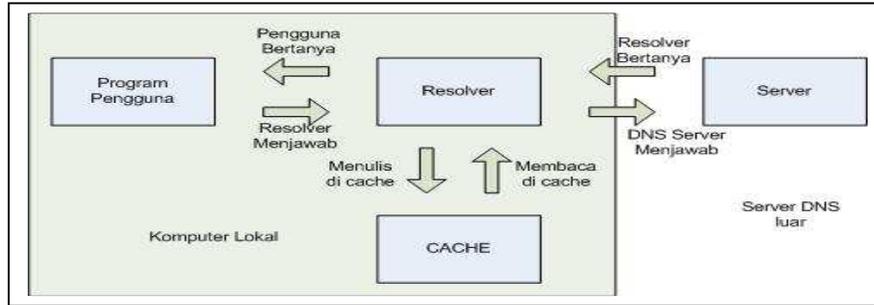
Top level domain digunakan untuk menunjukkan jenis perusahaan, instansi, lembaga atau negara tempat komputer ini berada. *Top Level Domain* (TLD) ini dapat dibagi menjadi 3 jenis yaitu :

- TLD generik (*generic domain*)
- TLD negara (*country domain*)
- TLD *arpa*

Pada mulanya TLD yang dipakai ialah TLD *generic*. TLD *generic* ini terdiri atas tujuh jenis domain yang terdiri atas tiga huruf. Domain com digunakan oleh organisasi bersifat komersial (ibm.com, microsoft.com). Domain edu (Berkeley.edu, purdue.edu, mit.edu) digunakan untuk lembaga pendidikan (universitas). Domain gov, digunakan untuk lembaga pemerintahan (whitehouse.gov, odci.gov). domain int digunakan oleh organisasi internasional (nato.int). domain mil digunakan oleh badan kemiliteran Amerika Serikat (army.mil, navy.mil, af.mil). Domain net digunakan oleh penyedia jaringan Internet (ibm.net, mci.net). Sedangkan domain org digunakan oleh organisasi nonkomersial (greenpeace.org). hingga saat ini sistem pembagian organisasional ini masih berlaku di Amerika. Dengan semakin banyaknya negara-negara yang terhubung ke internet, kemudian diputuskan untuk menggunakan standar pembagian geografis yang ditetapkan sesuai standar ISO 3166. Inilah yang disebut TLD Negara. Berdasarkan konvensi tersebut dialokasikan TLD yang merupakan pengenal geografis (negara) dan terdiri atas dua huruf yang unik Sebagai contoh negara Indonesia memiliki TLD .id, negara Inggris memiliki TLD .uk (united kingdom), negara Malaysia memiliki TLD .my dan sebagainya.

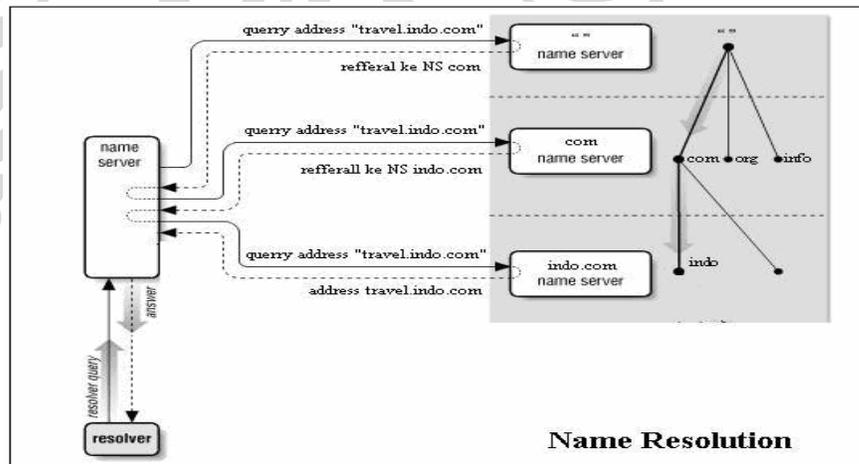
## 2.2 Prinsip Kerja DNS

Untuk memahami prinsip kerja DNS, maka terlebih dahulu kita perlu untuk memahami beberapa komponen yang terdapat dalam DNS seperti di tunjukkan dalam gambar dibawah berikut:



Gambar 2.2 Model Kerja Server DNS [6]

*Resolver* (gambar 2.2) merupakan bagian dari program aplikasi yang berfungsi menjawab pertanyaan tentang domain. *Resolver* akan menjawab pertanyaan dengan dua cara yaitu melihat isi dari *cache* nya (dengan asumsi bahwa pertanyaan ini pernah ditanyakan dan jawabanya tersimpan pada *cache*) dan bertanya kepada Server DNS luar [6].



Gambar 2.3 Name Resolution [3]

Gambaran umum proses kerja dari DNS dapat di ilustrasikan seperti gambar 2.3, apabila ada seseorang yang ingin menghubungi sebuah alamat internet, misalnya alamat website travel.indo.com. Maka komputer akan menghubungi *name server* lokal dan *name server* akan memeriksa terlebih dahulu

apakah ada *record host* tersebut di *cache name server* lokal. Jika tidak ada, *name server* lokal akan melakukan *query* kepada *root server* dan mereferensikan *name server* untuk TLD (*Top Level Domain*) .com, kemudian *root server* akan mendelegasikan ke salah satu *name server* yang memiliki otoritas terhadap TLD .com, *name server* lokal kembali melakukan *query* kepada *name server* .com dengan jenis *query* yang sama dan mereferensikan indo.com, kemudian *name server* .com akan mendelegasikan ke salah satu *name server* yang memiliki otoritas terhadap domain indo.com. *Name server* lokal kembali melakukan *query* ke *name server* indo.com dan mereferensikan *query* travel.indo.com, kemudian *name server* yang memiliki otoritas terhadap domain indo.com akan memberikan jawaban IP Address dari domain travel.indo.com

## 2.3 Berkeley Internet Name Domain ( BIND )

Ada beberapa buah program *name server* yang dapat diaplikasikan di sistem operasi unix dan variannya, diantaranya software *Berkeley Internet Name Domain* (BIND), DJBNS dan DENTS. Saat ini perangkat lunak *name server* yang lazim dipakai adalah BIND, aplikasi ini bersifat client server. Dalam prakteknya BIND dapat di fungsikan sebagai *Authoritative Name Servers* dan *Caching Name Servers*.

### 2.3.1 Authoritative Name Servers

Pada konfigurasi ini BIND dapat difungsikan sebagai *primary server* atau *slave server* yang sama-sama memiliki otoritas untuk menjawab permintaan yang datang dari pengguna terhadap domain yang di tangannya.

- **Primary Server**

*Primary Server* adalah *name server* utama dari sebuah *domain*. *Name server* ini memuat informasi domain langsung dari media penyimpanan di server yang langsung diatur oleh administrator. Misalkan, ns1.indo.com merupakan *primary server* dari *domain* indo.com maka setiap ada penambahan dan perubahan host di *domain* indo.com harus langsung di update pada zona file db.indo.com

- **Secondary Server**

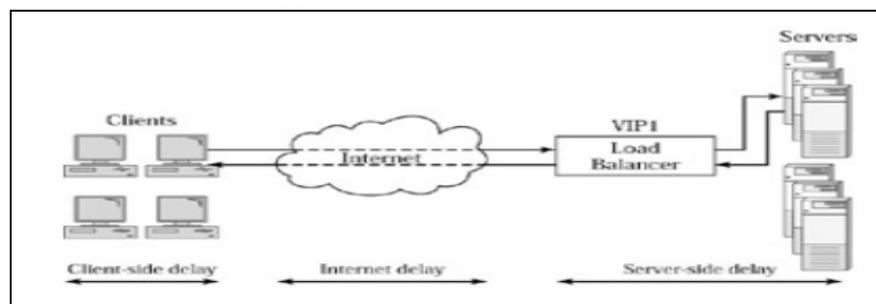
*Secondary Server* adalah DNS server yang tidak memelihara langsung zona file dari suatu domain. *Secondary server* akan mengupdate zona file dari *primary server*. Proses updating *secondary server* dari *primary server* disebut *zona transfer*. *Secondary server* digunakan untuk mengurangi *traffic query* permintaan ke *primary server*. *Client* cukup mengirim *query* ke *secondary server*. Setiap ada perubahan pada *primary server* akan diupdate secara otomatis oleh *secondary server*. Sebuah DNS server dapat merupakan *primary server* dari beberapa zona sekaligus dan dapat juga merupakan *secondary server* dari beberapa domain sekaligus.

### 2.3.2 Caching Name Server

*Caching Name Server* yaitu server tidak mempunyai kewenangan terhadap domain tertentu, dia akan menjawab permintaan terhadap nama domain dari nameserver lain. Dengan menjalankan *Caching-only DNS Server* maka setiap permintaan dari pengguna akan diteruskan ke DNS server root domain (.) yang dapat diakses oleh DNS server tersebut. Daftar server untuk root domain ini disimpan pada suatu file *hint.conf*. Untuk lebih detailnya yang berkaitan dengan konfigurasi DNS server akan diuraikan pada halaman lampiran.

### 2.4 Beberapa teknik untuk mengurangi internet delay

Kecepatan akses pengguna dalam menikmati layanan yang tersedia di internet di pengaruhi oleh beberapa faktor yang dapat dikelompokkan menjadi 3 kategori, yaitu *Client-side delay*, *Internet delay* dan *Server-side delay* [9].

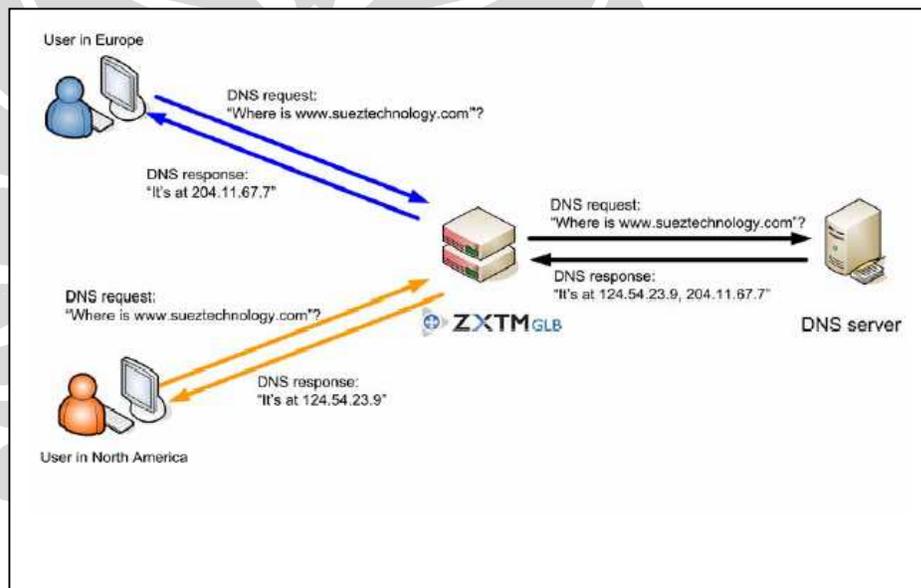


**Gambar 2.4 Internet delay bagian dari response time [9]**

Dalam penelitian ini akan berfokus pada masalah yang berkaitan dengan *internet delay* khususnya yang disebabkan karena jarak tempuh. Ada beberapa teknik yang dapat diterapkan untuk mengurangi *internet delay* sehingga kecepatan akses pengunjung terhadap layanan dapat di tingkatkan. Dibawah ini akan dijelaskan beberapa teknik yang dapat digunakan untuk meminimalisasi internet delay.

#### 2.4.1 ZXTM Global Load Balancer (ZXTM GLB)

Teknologi ini menawarkan *multiple, redundant datacenter* dan mengarahkan pengguna ke server terdekat. ZXTM GLB bekerja dengan cara memodifikasi atau mengembangkan *DNS server*, produk ini mengacu pada *DNS proxy* [1].



Gambar 2.5 Prinsip Kerja ZXTM GLB [1]

Gambar 2.5 memperlihatkan prinsip kerja dari ZXTM GLB, secara detailnya adalah sebagai berikut :

- ZXTM GLB menerima permintaan untuk sebuah domain, kemudian ZXTM GLB akan meneruskan permintaan ke DNS server.
- DNS server akan memberikan jawaban IP Address dengan sistem *round robin* ke ZXTM GLB dan ZXTM GLB akan menerima respon yang diberikan oleh DNS server.

- ZXTM GLB akan melakukan pengecekan apakah *domain* yang di minta oleh pengguna di konfigurasi untuk *loadbaancing*. Apabila *domain* tidak di konfigurasi untuk *loadbalancing*, maka IP Address yang diberikan oleh DNS server langsung di teruskan ke pengguna. Namun apabila *domain* yang di minta oleh pengguna di konfigurasi untuk *loadbalancing*, maka ZXTM GLB akan melakukan pengecekan terlebih dahulu dan akan memberikan IP Address server yang lokasi geografisnya dekat dengan pengguna yang melakukan permintaan terhadap *domain*.

Dengan memanfaatkan teknologi ini, pengguna dapat diarahkan ke lokasi server terdekat yang dapat mempercepat akses pengguna terhadap layanan yang diberikan. Selain itu teknik ini menawarkan *redudancy server* yang dapat menjaga keberlangsungan layanan. Namun dari sisi harga yang ditawarkan, teknik ini masih terlalu mahal bila di implementasikan pada sebuah website yang jumlah pengunjungnya belum begitu besar dan belum begitu mendunia.

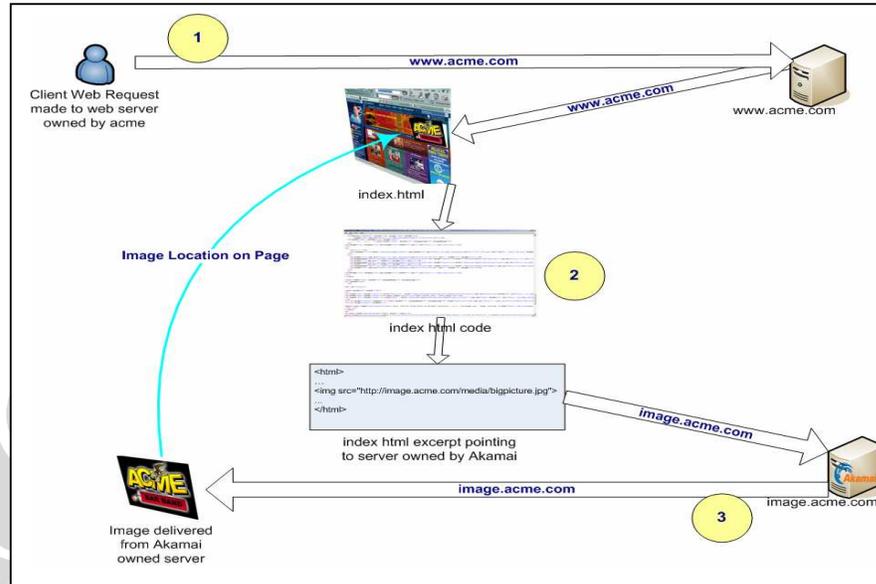
#### 2.4.2 Teknologi Akamai

Akamai Technologies Inc. adalah *global service provider* yang menyediakan layanan akselerasi "*global internet content*" untuk membantu mempercepat performa situs dengan cara melakukan *copy cache* dari website di banyak server yang tersebar di berbagai lokasi. Saat ini Akamai Technologies Inc. mempunyai lebih dari 15.000 server yang tersebar di 70 negara.

Ada beberapa komponen infrastruktur CDN (*Content Delivery Network*) yang dimiliki akamai, tiga komponen utamanya yaitu: *Edge Server*, *Content Delivery* dan *Mapping Center*. [4]

- *Edge Server*
  - Melakukan kerjasama dengan beberapa ISP untuk *install edge caches*.
  - Server tersebar di beberapa lokasi negara (2- 18 komputer tiap lokasi).
- *Content Delivery*
  - Berhubungan dengan pengiriman konten (*audio, graphics, animation, video*) ke *edge caches*.

- *Mapping Center*
  - Berkaitan dengan informasi *round-trip time edge server dan lokasi terdekat edge server* dengan pengguna yang melakukan *request*.



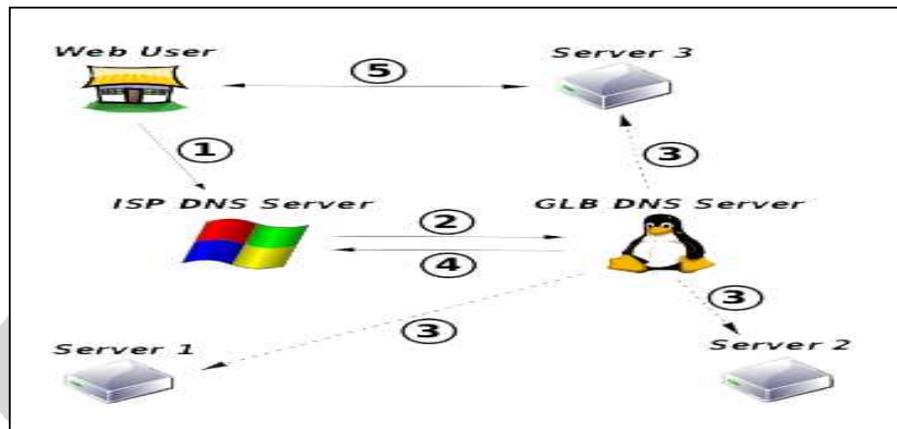
**Gambar 2.6 Content delivery to a user [2]**

Gambar 2.6 mengilustrasikan bagaimana Akamai dalam melakukan pengiriman *content* ke pengguna. Akamai secara transparan melakukan *mirror content* (*audio, graphics, animation, video*) yang tersimpan pada server klien Akamai ke beberapa *edge cache* server yang tersebar di beberapa negara. Ketika ada request user ke *website* www.acme.com, maka file – file *image, animasi, video* yang terdapat pada halaman www.acme.com akan di ambil dari *edge cache server* yang lokasinya dekat dengan pengguna. Namun ada juga *website* yang keseluruhan kontennya diletakkan pada *edge cache server akamai* yang tersebar di beberapa negara.

Teknologi Akamai merupakan salah satu teknik yang terbaik saat ini untuk memperkecil *internet delay* yang diakibatkan adanya jarak antara server dengan pengguna, namun dari sisi ekonomi akamai masih sangat terlalu mahal bila di terapkan pada sebuah *website* yang jumlah pengunjunnya belum begitu besar.

### 2.4.3 Teknologi GeoDNS

Teknik ini memanfaatkan kemampuan DNS yang telah diintegrasikan dengan GeoIP database. Dengan adanya integrasi ini, DNS server mampu melakukan pengarahan terhadap *query* pengguna ke lokasi server terdekat, prinsip kerjanya adalah sebagai berikut:



Gambar. 2.5 Prinsip Kerja GeoDNS Server [12]

#### Keterangan:

1. Ketika pertama kali komputer pengguna melakukan permintaan terhadap sebuah domain, maka komputer pengguna akan melakukan *resolve* terhadap ISP DNS Server.
2. Kemudian ISP DNS server akan menghubungi *Global Load Balancing* (GLB) DNS Server.
3. GLB DNS Server akan melihat IP Address komputer pengguna yang melakukan *query* kemudian akan memilih IP Address dari server yang paling dekat dengan lokasi geografis komputer pengguna.
4. GLB DNS Server memberitahukan IP Address dari server yang akan melayani komputer pengguna ke ISP DNS server.
5. Setelah komputer pengguna mengetahui IP Address dari server, maka komputer pengguna akan melakukan komunikasi dengan komputer server yang lokasi geografisnya dekat dengan pengguna.

Dari tiga teknik yang ditawarkan diatas, penulis berfokus pada pembahasan Teknologi GeoDNS. Teknik ini memanfaatkan DNS server dan GeoIP database yang berfungsi memetakan IP Address ke kode negara. Selain dapat mengarahkan *query* pengguna ke lokasi server terdekat, teknik ini juga dapat digunakan untuk *redundancy server*. Teknik ini cocok diterapkan pada sebuah website yang pengunjungnya belum begitu besar dan mendunia. Dari sisi biaya teknik ini sangat terjangkau karena memanfaatkan produk *open source*. Saat ini aplikasi *open source* DNS server yang telah bisa di implementasikan untuk teknik ini antara lain, OpenDNS, DJBNS dan BIND.

## 2.5 Maxmind's GeoIP Database

*GeoIp database* merupakan database yang berisikan informasi pengelompokan IP Address berdasarkan lokasi geografis atau lokasi negara [5]. Dewasa ini GeoIp sangat sering kita lihat penggunaannya dalam beberapa website untuk kebutuhan dalam mengidentifikasi lokasi geografis pengunjung website. Ada beberapa alasan lain penggunaan GeoIP dalam teknologi web, diantaranya yaitu:

- Mengetahui Statistik lokasi geografis pengunjung dari sebuah website
- Untuk mengetahui pemetaan pelayanan bahasa yang disediakan dalam website
- Untuk keperluan kegiatan marketing
- Melakukan black list terhadap IP Address berdasarkan negara
- Dll.

*GeoIP Maxmind's Database* merupakan salah satu produk dari MaxMind LLC. yang berisi database tentang pengelompokan IP Address berdasarkan lokasi geografis. MaxMind LLC, membagi produknya menjadi 2, yaitu *GeoLite Country* (Free) dan *GeoIP Country* (Komersial). Untuk mempermudah dalam penggunaannya, MaxMind LLC menyediakan API untuk mendukung beberapa bahasa pemrograman agar dapat mengakses GeoIP Maxmind's Database, selain itu MaxMind LLC. menyediakan *GeoLite Country* dalam bentuk format file binary yang telah dioptimalisasi [7] dan dalam situsnya <http://maxmind.com> disebutkan

bahwa file dalam format binary ini mampu menangani 1000 qps (*query per second*).

## 2.6 Integrasi GeoIP Maxmind's Database dengan Bind 9.x.x

Bind 9.x.x memiliki beberapa fitur tambahan salah satunya yaitu fitur *view*, dengan adanya fitur ini memungkinkan Bind 9.x.x memiliki lebih dari satu informasi domain pada satu zone tergantung dari user yang melakukan *query* dns. Namun Bind 9.x.x hanya bisa membedakan informasi zone berdasarkan IP Address atau *IP prefix*, hal inilah yang menjadikan permasalahan karena Bind 9.x.x tidak bisa membedakan informasi zone berdasarkan kode negara [11].

Dengan melakukan integrasi GeoIP terhadap BIND 9.x.x, hal ini memungkinkan untuk dilakukan pemfilteran berdasarkan kode negara menggunakan fitur *view* [2] yang ada pada BIND 9.x.x tanpa menggunakan *IP prefix*. Untuk melakukan integrasi ini, ada beberapa file pada BIND 9.x.x yang dimodifikasi. File-file tersebut adalah sebagai berikut:

- `lib/dns/include/dns/acl.h`, pada file ini akan dilakukan penambahan jenis baru dari *access control list (acl)* yaitu *dns\_aclelementtype\_ipcountry*
- `lib/dns/acl.c`, `dns_aclelement_match` untuk menambahkan fungsi pencocokan terhadap jenis *ipcountry* dari komponen *access control list (acl)*
- `lib/isccfg/parser.c`, untuk meakukan parsing jenis baru dari *access control list (acl)* terhadap file konfigurasi.

Setelah dilakukan modifikasi terhadap beberapa file diatas, maka memungkinkan BIND 9.x.x untuk melakukan pemfilteran terhadap informasi zone dengan menggunakan kode negara. Contoh sederhana konfigurasi BIND 9.x.x setelah di patch dengan `bind-9.x-geodns-patch` adalah sebagai berikut:

```
view "america" {
  match-clients { country_US; country_CA; country_MX; };
  recursion no;
  zone "travel.indo.com" {
    type master;
    file "pri/travel-america.db";
  };
};
view "uerupe" {
```

```

match-clients { country_UK; country_DE; country_FR; };
recursion no;
zone "travel.indo.com" {
    type master;
    file "pri/travel-uerupe.db";
};
};
view "other" {
    match-clients { any; };
    recursion no;
    zone "travel.indo.com" {
        type master;
        file "pri/travel-other.db";
    };
};
};

```

Konfigurasi diatas memperlihatkan bahwa zone travel.indo.com mempunyai 3 konfigurasi informasi pemetaan IP Address, tergantung dari lokasi negara pengguna yang melakukan permintaan. Untuk lebih detailnya sebagai berikut:

- Bila permintaan terhadap domain travel.indo.com berasal dari IP Address Negara US, CA, MX maka informasi zona file berada pada file travel-america.db
- Bila permintaan terhadap domain travel.indo.com berasal dari IP Address Negara UK, DE, FR maka informasi zona file berada pada file travel-uerupe.db
- Bila permintaan terhadap domain travel.indo.com berasal dari IP Address negara yang tidak ada dalam daftar negara di fitur view maka informasi zona file berada pada file travel-other.db