BAB 2
TINJAUAN PUSTAKA

2.1 Profil Blastech Digital

Perusahaan PT. Blastech Digital merupakan perusahaan yang bergerak di bidang *cloud computing* di Indonesia. PT Blastech Digital selaku *Cloud Provider* hadir untuk memberikan solusi komputasi awan bagi para *developer* dan bisnis-bisnis yang bergerak dibidang IT. PT Blastech Digital memfokuskan pemasaran layanannya hanya di Indonesia dan telah berdiri sejak Maret 2013.

Perusahaan ini menawarkan berbagai layanan *cloud hosting* diantaranya *web* hosting, *iserver*, *dedicated server*, dan domain gTLD serta ccTLD. PT Blastech Digital memasarkan layanannya melalui berbagai *platform* seperti *web*site, media sosial, dan iklan di Google serta Instagram. PT Blastech selalu berfokus dalam meningkatkan kepuasaan pelanggan dengan memberikan dukungan selama 24 jam dan meningkatkan kualitas layanannya.

2.1.1 Visi dan Misi PT Blastech Digital

Menurut Wibisono, Visi merupakan rangkaian kalimat yang menyatakan cita-cita atau impian sebuah organisasi atau perusahaan yang ingin dicapai di masa depan. Atau dapat dikatakan bahwa visi merupakan pernyataan *want to be* dari organisasi atau perusahaan. Visi juga merupakan hal yang sangat krusial bagi perusahaan untuk menjamin kelestarian dan kesuksesan jangka panjang [14].

Dalam visi suatu organisasi terdapat juga nilai-nilai, aspirasi serta kebutuhan organisasi di masa depan seperti yang diungkapkan oleh Kotler yang dikutip oleh Nawawi, Visi adalah pernyataan tentang tujuan organisasi yang diekspresikan dalam produk dan pelayanan yang ditawarkan, kebutuhan yang dapat ditanggulangi, kelompok masyarakat yang dilayani, nilai-nilai yang diperoleh serta aspirasi dan cita-cita masa depan [15].

Berdasarkan penjelasan Visi menurut Wibisono, dan Nawai dapat disimpulkan bahwa Visi adalah pernyataan tentang tujuan organisasi yang

diekspresikan dari layanan yang ditawarkan dan sangat krusial untuk menjamin kesuksesan jangka panjang.

Menurut Ismail Solihin Misi merupakan pernyataan tujuan atau alasan eksistensi organisasi atau perusahaan, misi juga merupakan suatu atau beberapa tindakan yang harus dilakukan untuk bisa mencapai visi yang telah ditetapkan sebelumnya [14] . Sedangkan menurut Hery Misi merupakan alat untuk membuat segenap anggota perusahaan bergerak dan menuju ke arah yang sama, sebagaimana telah ditetapkan [15] . Sedangkan menurut Eddy Yunus Misi merupakan rangkaian kegiatan utama yang harus dilakukan organisasi untuk mencapai visinya. Misi akan melakuka arah sekalipun batasan proses pencapaian tujuan [15] .

Berdasarkan penjelasan Misi menurut Ismail Solihin, Hery, dan Eddy Yunus dapat disimpulkan bahwa Misi adalah rangkaian kegiatan yang harus dilakukan sebuah organisasi menuju ke arah yang sama sebagaimana yang sudah ditetapkan untuk mencapai visinya.

Visi dari PT. Blatech Digital adalah “Menjadi Cloud Provider yang maju dan terpercaya, mandiri, mempunyai daya saing nasional serta mampu memberikan kepuasan dan kemudahan kepada klien dalam memberikan layanan *cloud* yang berkualitas dan terjangkau”. Sedangkan Misi dari PT. Blastech Digital adalah sebagai berikut :

1. Menciptakan ekosistem sehat dan membangun infrastuktur cloud kelas dunia bagi para developer dan pelaku bisnis di Indonesia.
2. Menghasilkan layanan *cloud* yang berkualitas tinggi sehingga dapat memenuhi kebutuhan komputasi awan para *developer* dan pelaku bisnis melalui:
    a. Layanan *cloud* yang sesuai dengan aturan dan ketentuan Kementrian Riset, Teknologi, dan Pendidikan Tinggi Indonesia dan hukum yang berlaku di Indonesia
    b. Proteksi kerahasiaan data klien dengan menerapkan Kebijakan Privasi dan Sistem Keamanan tingkat tinggi pada halaman klien.

c. Penerapan Infrastuktur *Cloud* kelas *Enterprise* pada setiap komponen cloud seperti *Processor*, Media Penyimpanan, *Memory*, Jaringan dan didesain dengan redudansi tinggi.
d. Keamanan tingkat tinggi dengan menggunakan *Cloud Firewall*, *Web Application Firewall*, dan *Antivirus Premium*.
e. Pengelolaan layanan dengan Control Panel yang ramah pengguna langsung dari halaman klien.
f. Support 24 jam melalui Tiket Dukungan, Chat, dan Telpon.

3. Membangun *Brand Awareness* sebagai strategi *marketing* untuk membangun kesadaran *brand* bagi para pelaku *developer* dan pelaku bisnis agar Blast Compute mampu bersaing secara nasional.
4. Meningkatkan *User Experience* dari klien dengan selalu memberikan kemudahan dalam pengelolaan layanan dan mengoptimasi kecepatan *web*site agar ramah pengguna.

2.1.2 Logo

Menurut Hery Logo merupakan sebuah lambang yang dimiliki oleh setiap perusahaan atau instansi. Pembuatan logo dimaksudkan untuk merepresentasikan identitas suatu perusahaan yang mencerminkan jiwa, visi dan misi suatu perusahaan/instansi [15].

Logo PT. Blastech Digital merepresentasikan lingkungan *cloud* dengan pemilihan warna dan gambar awan, logo perusahaan dapat dilihat pada gambar 2.1 Logo PT. Blastech Digital.

Gambar 2.1 Logo Blastech Digital

2.1.3 Badan Hukum

Menurut Prof. Subekti Badan Hukum adalah suatu badan atau perkumpulan yang dapat memiliki hak-hak dan melakukan perbuatan seperti menerima serta memiliku kekayaan sendir, dan dapat digugat dan menggugat di muka hukum [17]. Sedangkan menurut Molengraff Badan Hukum adalah hak dan kewajiban dari para

anggotanya secara bersama-sama, dan di dalamnya terdapat harta kekayaan bersama yang tidak dapat dibagi-bagi [18] .

Berdasarkan penjelasan Badan Hukum menurut Prof.Subekti dan Molengraaff Badan Hukum adalah suatu badan atau perkumpulan yang dapat memiliki hak-hak dan kewajiban dari para anggotanya secara bersama-saa dan di dalamnya terdapat harta kekayaan bersama yang tidak dapat dibagi- bagi.

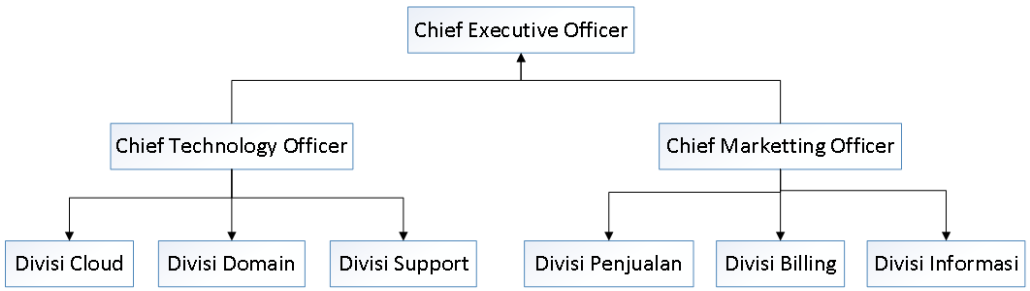
PT Blastech Digital adalah perusahaan perseroan terbatas yang telah berbadan hokum dan terdaftar di menkunham secara legal.

2.1.4 Struktur Organisasi Perusahaan

Menurut Eddy Soeryanto Struktur Organisasi adalah suatu susuna dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan yang diharapkan [19]. Sedangkan menurut Prof.Dr.Aime Heene Struktur Organisasi merupakan struktur baku yang mendasari untuk dilakukannya pemilahan, pengelompokan, dan pengkoordinasiaan tugas-tugas [20].

Berdasarkan penjelasan Struktur Organisasi menurut Prof.Dr.Aime Heene dan Eddy Soeryanto dapat disimpulkan bahwa Struktur Organisasi merupakan sebuah suatu susunan yang menjadi dasar untuk dilakukannya pengelompokkan, pengkoordinasian tugas-tugas pada sebuah perusahaan.

Berikut ini adalah struktur organisasi dari PT. Blastech Digital dapat dilihat pada gambar 2.2 Struktur Organisasi.



Gambar 2.2 Struktur Organisasi

1. *Chief Executive Officer*
   a. Memimpin perusahaan dengan menetapkan kebijakan-kebijakan yang berlaku di perusahaan.

b. Bertanggung jawab terhadap keberlangsungan manajemen perusahaan keseluruhan.
c. Memilih, menetapkan, mengawasi, memberhentikan tugas dari karyawan.
d. Merumuskan Strategi umum dan jangka panjang untuk keberlangsungan perusahaan.

2. *Chief Technology Officer*
   a. Memimpin tiga divisi yakni Divisi Cloud, Divisi Domain, dan Divisi Support.
   b. Bertanggung jawab terhadap divisi yang dibawahinya.
   c. Merumuskan strategi untuk meningkatkan kualitas layanan melalui R&D .
3. *Chief Marketting Officer*
   a. Bertanggung jawab terhadap manajemen bagian pemasaran.
   b. Mengadakan hubungan/kontrak dengan relasi.
   c. Bertanggung jawab terhadap perolehan hasil penjualan dan penggunaan dana promosi.
   d. Sebagai koordinator manajer produk dan manajer penjualan
4. *Divisi Cloud*
   a. Mengelola layanan cloud seperti *web* hosting, *cloud server*, *dedicated server*, *colocation*,dll.
   b. Memberikan dukungant teknis kepada klien mengenai *cloud*.
5. Divisi *Domain*
   a. Mengelola layanan domain gTLD dan ccTLD
   b. Memberikan dukungant teknis untuk layanan domain kepada klien.
6. Divisi *Support*
   a. Memberikan dukungan teknis secara umum kepada klien.
7. Divisi Penjualan
   a. Menjalankan strategi marketing yang telah ditetapkan oleh CMO
   b. Melakukan pemasaran terhadap semua layanan Blast Compute.
8. Divisi *Billing*
   a. Mengelolah semua *invoice* dari setiap pembelian yang masuk.

9. Divisi Informasi

a. Memberikan informasi mengenai perusahaan, gambaran umum layanan kepada klien atau calon klien.

2.2 Landasan Teori

Landasan Teori merupakan definisi dan konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan penelitian ini meliputi Optimasi *Web Server*, Caching, *Reverse Proxy*, *Reverse Proxy* Caching, *Apache, LiteSpeed Web Server*, *Google Pagespeed Insight*, *Lighthouse Engine*, Plesk Onyx *Cloud, HTML*, *CSS, PHP, JavaScript, jQuery*, *MySQL, SQL, Web Browser.*

2.2.1 Metode PPDIOO

PPDIOO merupakan metode perancangan jaringan dari Cisco atau biasa disebut sebagai siklus hidup layanan jaringan Cisco yang dirancang untuk mendukung berkembangnya jaringan. PPDIOO terdiri dari *Prepare, Plan, Design, Implement, Operate, dan Optimize*. Dengan kebutuhan layanan jaringan yang semakin kompleks, maka diperlukan suatu metodologi yang mendukung perancangan arsitektur dan disain jaringan[1].

Adapun pemahaman detail mengenai tiap-tiap fase pada metode pengembangan jaringan PPDIOO adalah sebagai berikut Cisco, Inc (2011,p13):

1. Fase *Prepare* (Persiapan)

Fase Prepare (persiapan), menetapkan kebutuhan organisasi dan bisnis, mengembangkan strategi jaringan, dan mengusulkan konsep arsitektur dengan level tingkat tinggi, untuk mendukung suatu strategi, yang didukung dengan kemampuan keuangan pada organisasi atau perusahaan tersebut[2].

2. Fase *Plan* (Perencanaan)

Fase Plan (perencanaan) mengidentifikasi persyaratan jaringan berdasarkan tujuan, fasilitas, dan kebutuhan pengguna. Fase ini mendeskripsikan karakteristik suatu jaringan, yang bertujuan untuk menilai jaringan tersebut, melakukan gap analisis pada perancangan terbaik sebuah arsitektur, dengan melihat perilaku dari lingkungan operasional. Sebuah perencanaan proyek dikembangkan untuk mengelola tugas-tugas (*tasks*), pihak-pihak yang bertanggung jawab, batu pijakan

(milestones), dan semua sumber daya untuk melakukan desain dan implementasi. Perencanaan proyek harus sejalan dengan ruang lingkup (batasan), biaya dan parameter sumber daya yang disesuaikan dengan kebutuhan bisnis . Rencana proyek ini diikuti (dan diperbarui) selama fase-fase dalam siklus[2].

3. Fase *Design* (Desain)

Desain jaringan dikembangkan berdasarkan persyaratan teknis, dan bisnis yang diperoleh dari kondisi sebelumnya. Spesifikasi desain jaringan adalah desain yang bersifat komprehensif dan terperinci, yang memenuhi persyaratan teknis dan bisnis saat ini. Jaringan tersebut haruslah menyediakan ketersediaan, kehandalan, keamanan, skalabilitas dan kinerja. Hasil desain termasuk diagram jaringan, dan daftar peralatan-peralatan. Rencana proyek harus terus diperbarui, dengan informasi yang lebih terperinci untuk diimplementasikan. Setelah tahap desain disetujui, fase implementasi dimulai[2].

4. Fase *Implement* (Implementasi)

Pada fase ini, peralatan-peralatan baru dilakukan instalasi dan di konfigurasi, sesuai spesifikasi desain. Perangkat-perangkat baru ini akan mengganti atau menambah infrastruktur yang ada. Perencanaan proyek juga harus diikuti selama fase ini, jika ada perubahan seharusnya disampaikan dalam pertemuan (*meeting*), dengan persetujuan yang diperlukan untuk dilanjutkan. Setiap langkah dalam implementasi, harus menyertakan deskripsi, rincian pedoman pelaksanaan, perkiraan waktu untuk penerapan, evaluasi (*rollback*) langkah-langkah jika terdapat kegagalan, dan informasi-informasi lainnya sebagai referensi tambahan. Seiring perubahan yang telah di implementasikan, tahapan ini juga menjadi langkah pengujian, sebelum pindah ke fase operasional (*operate phase*) [2].

5. Fase *Operate* (operasional)

Fase operasional adalah mempertahankan ketahahan kegiatan sehari-hari jaringan. Operasional meliputi pengelolaan dan memonitor komponen-komponan jaringan, pemeliharaan routing, mengelola kegiatan upgrade, mengelola kinerja, mengidentifikasi dan mengoreksi kesalahan jaringan. Tahapan ini adalah ujian akhir bagi tahapan desain. Selama operasi, manajemen jaringan harus memantau stabilitas dan kinerja jaringan, Deteksi kesalahan, koreksi konfigurasi, dan

kegiatan-kegiatan pemantauan kinerja, yang menyediakan data awal untuk fase selanjutnya, yaitu fase optimalisasi (*optimize phase*) [2].

6. Fase *Optimize* (Optimalisasi)

Fase optimalisasi, melibatkan kesadaran proaktif seorang manajemen jaringan dengan mengidentifikasi dan menyelesaikan masalah, sebelum persoalan tersebut mempengaruhi jaringan. Fase optimalisasi, memungkinkan untuk memodifikasi desain jaringan, jika terlalu banyak masalah jaringan yang timbul, kemudian juga untuk memperbaiki masalah kinerja, atau untuk menyelesaikan masalah-masalah pada aplikasi (*software*). Persyaratan-persayaratan untuk desain jaringan yang dimodifikasi mengarahkan perkembangan jaringan tersebut, kembali ke awal siklus hidup dalam model fase PPDIOO[2].

2.2.2 Web Server

Web Server merupakan perangkat lunak pada *Server* yang memiliki fungsi sebagai penerima permintaan *(request*) yang berupa halaman *web* dari *client* dan mengirimkan kembali (*respons*) hasil yang diminta dalam bentuk halaman-halaman *web* [3].

Web Server adalah perangkat lunak yang menjadi tulang belakang dari *world wide web* (www) yang pertama kali tercipta sekitar tahun 1980an. *Web Server* menunggu permintaan dari *client* yang menggunakan *browser* seperti, *Internet Explorer, Mozilla Firefox,* dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser* [3].

Dari segi arsitektur, *Web Server* menggunakan dua pendekekatan untuk menangani konkurensi yakni *Thread Based Model, Process Based Model* dan *event-driven model*.

2.2.3 *Process Based Model*

Dalam arsitektur *Process Based Model* (main *server* process membuat salinan dirinya sebagai child *process* (*preforking*). *Child process* menangani permintaan secara mandiri. Strategi ini memastikan stabilitas sistem sedemikian rupa sehinggasatu proses yang hancur tidak mempengaruhi yang lain. Namun, setiap *child process* menempati ruang memory yang terpisah. Penggunaan memori,

dalam kasus ini, berbanding lurus dengan jumlah *child process*. Apalagi sistem operasinya perlu menghabiskan waktu CPU tambahan untuk beralih antara *child process (context switching).* Komunikasi antar-proses juga sulit karena ada pemisahan antara *child process*[3].

2.2.4 *Thread Based Model*

Dalam Arsitektur *Thread Based Model*, *dispatcher thread* membuat antrian koneksi baru untuk ke sebuah *socket*. *Request* terhadap *thread handler* mengeksekusi koneksi yang diterima dari *dispatcher queue*. Karena kumpulan thread berisi maksimum jumlah thread, *handler* tertentu mungkin menganggur dan sibuk lainnya jika tidak cukup koneksi tersedia dalam antrian. *Thread* dapat berbagi memori, operasi I / O, dan soket karena mereka adalah bagian dari proses yang sama. Demikian, tidak ada context switching dalam kasus ini. Namun interaksi buruk pada data yang sama dan / atau variabel antara thread dapat menyebabkan seluruh sistem mati[3].

2.2.5 Event Based Model

Dalam arsitektur *Event Based Model*, satu proses loop peristiwa tunggal berurutan mengeksekusi banyak koneksi menggunakan mekanisme I / O *non-blocking*. *Event* dihasilkan dalam *sources* yang berbeda termasuk *socke*t, operasi I / O atau di dalam *event-handler* karena ada perubahan yang signifikan dalam data atau *states*. Event Handler tidak menarik *event resources*. Sebaliknya, event di *push* ke *event loop* sebagai notifikasi. Arsitektur *Event Based Model* dapat menghilangkan *performance bottleneck* jika terjadi beban koneksi tinggi berkat komunikasi asinkronnya. Namun, tidak dapat melakukan operasi I / O *non-blocking* dalam *single threaded event handler*[3].

2.2.6 Caching

Cache adalah perangkat keras atau perangkat lunak yang digunakan untuk menyimpan sesuatu, biasanya data, untuk sementara waktu di lingkungan komputasi[5].

Sejumlah kecil memori yang lebih cepat dan lebih mahal digunakan untuk meningkatkan kinerja data yang baru-baru ini diakses atau sering diakses yang disimpan sementara di media penyimpanan yang dapat diakses dengan cepat yang

bersifat lokal untuk klien *cache cache* dan terpisah dari penyimpanan massal. *Cache* sering digunakan oleh klien *cache*, seperti CPU, aplikasi, *web browser* atau sistem operasi (OS).

Cache digunakan karena media penyimpanan tidak bisa memenuhi permintaan klien *cache* secepat yang diinginkan. *Cache* mempersingkat waktu akses data, mengurangi latensi dan meningkatkan input / output (I / O). Karena hampir semua beban kerja aplikasi bergantung pada operasi I / O, caching meningkatkan kinerja aplikasi[5].

Ketika klien *cache* perlu mengakses data, klien akan terlebih dahulu memeriksa *cache*. Ketika data yang diminta ditemukan dalam *cache*, itu disebut *cache hit*. Persentase upaya yang menghasilkan hit *cache* dikenal sebagai rasio hit *cache* atau rasio.

Jika data yang diminta tidak ditemukan dalam *cache* - situasi yang dikenal sebagai *cache* miss - itu ditarik dari memori utama dan disalin ke dalam *cache*. Bagaimana ini dilakukan, dan data apa yang dikeluarkan dari *cache* untuk memberikan ruang bagi data baru, tergantung pada algoritma caching atau kebijakan yang digunakan sistem.

Browser web, seperti Internet Explorer, Firefox, Safari dan Chrome, menggunakan *cache* browser untuk meningkatkan kinerja halaman *web* yang sering diakses. Saat Anda mengunjungi halaman *web*, file yang diminta disimpan dalam penyimpanan komputasi Anda di *cache* browser.

Mengklik kembali dan kembali ke halaman sebelumnya memungkinkan browser Anda untuk mengambil sebagian besar file yang dibutuhkan dari *cache* alih-alih membuat mereka semua membenci dari *server web*. Pendekatan ini disebut read *cache*. Browser dapat membaca data dari *cache* browser lebih cepat daripada membaca ulang file dari halaman *web*.

Berikut adalah jenis-jens *caching*:

1. *Browser caching*

Browser *cache* membantu mempermudah dan membuat loading lebih cepat. Dengan browser caching, *web server* tidak perlu lagi melakukan request dan transmisi data untuk menampilkan *web*site yang ingin dikunjungi di *browser*.

Dengan caching, data-data yang dibutuhkan untuk menampilkan *web*site yang ingin Anda kunjungi sudah ada di computer Anda. Dengan begitu, *web*site akan memiliki waktu loading yang lebih cepat dan data dari *web*site dapat diakses secepat mungkin. Semua *web* browser dari Google Chrome sampai Firefox sudah melakukan *browser caching*[8].

2. OpCode

OpCode Cache adalah caching pada script PHP dengan menyimpan hasil kompilasi dari PHP. Sehinnga proses untuk melakukan eksekusi PHP jadi lebih singkat tanpa harus melalui tahapan parsing dan kompilasi dan disimpan didalam *memory*/[8].

3. Object Cache

Dengan object caching, data objek dapat disimpan secara lokal sehingga tidak perlu diambil secara konstan untuk permintaan tambahan. Dengan demikian, object *cache* membantu meningkatkan kecepatan dan kinerja aplikasi *web*. Objek adalah kumpulan data yang mencakup dokumen kata, video, atau gambar. Jika objek di-*cache* ketika pengguna meminta informasi, itu dapat ditransfer langsung dari *cache* lokal, daripada memintanya dari *server*.

Ini adalah manfaat utama dari object caching. Jika pengguna meminta data yang tidak berubah, mereka dapat mengaksesnya tanpa menggunakan *server*, membuat semuanya jauh lebih mudah, dan lebih cepat. Karena pengguna tersebut tidak dibiarkan menunggu konten dimuat, dan *bandwidth* tidak terbuang[8].

4. Page Cache

Page cache memiliki kemiripan dengan *cache* lain. Salah satu manfaat melakukan page *cache* adalah untuk meningkatkan kecepatan waktu loading sebuah halaman *web*site untuk memberikan user experience yang lebih baik. Page *cache* menyimpan halaman *web* lengkap untuk ditampilkan di lain waktu kepada pengunjung. Data ini disimpan dalam bagian RAM yang tidak terpakai dan dengan demikian tidak memiliki dampak nyata pada memori. Bahkan, meskipun digunakan untuk menyimpan informasi ini, komputer mungkin masih mendaftarkan bagian memori komputer ini sebagai tersedia atau bahkan kosong. Jika data dibaca lagi nanti, itu akan dibaca dari *cache* ini yang sudah ada dalam memori[8].

5. *Content Delivery Network Cache*

CDN *cache* adalah bentuk penyimpanan data yang lebih luas. Dengan *cache* CDN, konten situs *web* statis ditambahkan ke *server proxy* yang didistribusikan secara global. Ini memungkinkan pengunjung dari seluruh dunia untuk mengunduh konten situs Anda lebih cepat sehingga mempercepat waktu buka situs.

Caching CDN juga membantu pemilik situs *web* mengurangi biaya, menghilangkan tekanan dari *server* asli, dan menempatkan fokus pada *server* lokal yang lebih kecil yang berlokasi secara global di mana pengunjung dapat mengakses data secara *local*[8].

2.2.7 *Reverse Proxy*

Reverse proxy adalah sebuah *proxy* yang berada di depan dari *web server*, digunakan sebagai *cache* atau bisa juga sebagai load balancer. *Reverse proxy* menjadi perantara user-user di internet terhadap akses ke *web-web server* yang berada pada *local area network*, sehingga seolah-olah user di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya user di internet mengakses *web-web server* yang terdapat di local area network melalui *reverse proxy* tersebut[5].

Sebuah *reverse proxy server* diperlukan dalam permintaan dari *web* browser di seluruh internet. *Server proxy* meneruskan permintaan tersebut ke *server* lain. Permintaan tersebut menghasilkan respon balik ke *server proxy reverse*, yang kemudian diteruskan kembali ke *web* browser. Misalnya, permintaan untuk mendapatkan informasi produk di toko online masuk ke *server* gambar yang menyimpan gambar produk serta *server* database, yang menyimpan harga persediaan, saat ini dan deskripsi item[5].

Salah satu penggunaan umum *reverse proxy* adalah untuk menyimpan sementara (*cache*) data statis (*static data*). Ketika *proxy* ini menerima sebuah request untuk konten statis, seperti gambar, suara, atau video, *proxy* ini akan menyimpan data sebelum mengirimkannya pada client. Di lain waktu informasi ini direquest kembali, *reverse proxy* dapat mengirimkanya langsung pada client, ketimbang harus meneruskan request pada *web server* sederhana. Hal ini dapat menghasilkan beberapa beban muat pada regular *server*[5].

Sebuah *server proxy reverse* sangat meningkatkan keamanan *web*site karena tidak ada *server* perusahaan secara langsung diakses ke internet. Selama pemilik *web*site tidak menempatkan aplikasi kritis seperti email dan penggajian pada *reverse proxy server*, *server* lain perusahaan dan aplikasi yang aman. Sebagai contoh, seorang *hacker* ingin untuk menggantikan citra sebuah produk dengan porno dan perubahan harga produk untuk satu sen. Baik *server* foto atau *server* database produk dapat disentuh langsung oleh pengguna di internet. Karena *reverse proxy server* dan situs *web* tidak dirancang untuk pengguna untuk memperbarui gambar produk atau harga, hacker mungkin tidak akan berhasil dalam melakukan apa yang ia inginkan Dengan memisahkan kegiatan yang berbeda dari sebuah situs online, pemilik situs *web* mendistribusikan beban antara *server* yang berbeda seperti *server database*, *server* aplikasi dan *server* akuntansi sehingga tidak ada satu komputer melakukan segalanya. *Reverse proxy server* skala serta *web*site tumbuh besar tanpa mengubah arsitektur aslinya. Banyak situs yang lebih besar menggunakan beberapa *server proxy* yang mengarah ke kelompok mesin yang berbeda untuk mendistribusikan beban lebih[5].

2.2.8 *Reverse Proxy* Caching

Reverse proxy Caching merupakan suatu perantara antara aplikasi *web* dengan client [5]*Reverse proxy caching* seringkali disebut juga dengan *web server acceleration*. Hal ini dikarenakan tujuan dari penggunaan *reverse proxy* caching adalah untuk mengurangi beban pada *web server*, baik yang menyediakan konten statis maupun dinamis [5]

Reverse proxy caching diletakkan lebih dekat kepada *web server* daripada client [5]. Cara utama yang digunakan untuk meringankan beban *web server* yaitu dengan menggunakan *cache* antara *server* dan internet [5]

2.2.9 Apache

Apache HTTP *Server*, biasa disebut *Apache*, adalah perangkat lunak *web server* lintas-platform gratis dan open-source, dirilis di bawah persyaratan *Apache* License 2.0. *Apache* dikembangkan dan dikelola oleh komunitas pengembang terbuka di bawah naungan *Apache* Software Foundation.

Desain *Apache* adalah berbasis thread dimana proses utamanya (Multi-Processing) Modul-MPM) dipanggil saat start-up dan prefor *child process* (modul) untuk secara bersamaan menangani permintaan. *Apache* dapat bertindak sebagai multithreaded, model multi-proses atau keduanya dan ini dapat ditentukan oleh MPM. httpd adalah intinya modul di *Apache* yang mengimplementasikan HTTP pemrosesan permintaan / tanggapan.

2.2.10 *LiteSpeed*

LiteSpeed Web Server (LSWS), adalah perangkat lunak *server web* milik. Ini adalah *server web* ke-4 terpopuler, diperkirakan digunakan oleh 3,5% situs *web* pada Oktober 2018. [3] LSWS dikembangkan oleh *LiteSpeed* Technologies milik pribadi. Perangkat lunak ini merupakan pengganti drop-in *Apache*, artinya menggunakan format konfigurasi yang sama dengan *Apache*. [13]

LSWS dirilis pada tahun 2003, dan pada bulan Agustus 2008 menjadi *server web* ke-16 yang paling populer. [13] Pada November 2016, pangsa pasar *LiteSpeed* tumbuh dari 0,39% menjadi 3,29%, meningkatkan posisinya dari *server web* terpopuler ke-10 ke-4 menurut Netcraft. [7] Pada 2017, sebuah tim dari Hong Kong Polytechnic University menemukan itu menjadi salah satu dari enam *server web* paling populer, [13] dan diperkirakan oleh tim di RWTH Aachen University untuk menjalankan 9,2% dari semua situs *web* yang mendukung HTTP / 2-enabled. [13] Pada Desember 2017, LSWS digunakan oleh 97,5% situs *web* menggunakan QUIC.

LiteSpeed Web Server (LSWS) kompatibel dengan fitur *Apache* yang umum digunakan, termasuk *mod_rewrite*, *.htaccess*, dan *mod_security*. LSWS dapat memuat *file* konfigurasi *Apache* secara langsung dan berfungsi sebagai pengganti *drop-in* untuk *Apache* sementara sepenuhnya terintegrasi dengan panel kontrol yang populer. LSWS menggantikan semua fungsi *Apache*, tetapi menggunakan pendekatan *event driven* untuk menangani permintaan. [13]

Web Server LiteSpeed meningkatkan kinerja & skalabilitas platform *hosting web* melalui arsitekturnya yang digerakkan oleh peristiwa. Ini memiliki kemampuan melayani ribuan klien secara bersamaan dengan penggunaan sumber daya *server* minimal seperti Memori dan CPU. Kode unik yang dikembangkan & dioptimalkan dari *server web LiteSpeed* meningkatkan kinerja PHP dan juga

melayani situs *web* statis lebih cepat daripada *Apache*. Ia memiliki kemampuan menangani lonjakan lalu lintas yang tiba-tiba serta membantu mengelola serangan DDOS tanpa perangkat keras mitigasi DDOS. [13]

2.2.11 Google Pagespeed Insight

PageSpeed Insights (PSI) adalah *tools* untuk mengukur kinerja halaman pada perangkat seluler dan desktop, dan memberikan saran tentang bagaimana halaman tersebut dapat ditingkatkan menggunakan *Lighhouse Engine*. [11]

Di bagian atas laporan, PSI memberikan skor yang merangkum kinerja halaman. Skor ini ditentukan dengan menjalankan Mercusuar untuk mengumpulkan dan menganalisis *data lab* tentang halaman. Skor 90 atau lebih dianggap cepat, dan 50 hingga 90 dianggap rata-rata. Di bawah 50 dianggap lambat.

PSI menyajikan distribusi metrik ini sehingga pengembang dapat memahami rentang nilai FCP dan FID untuk halaman atau asal tersebut.[11]

1. Performance Metrics

Performance metrics adalah sitem pemberian skor dari *Lighhouse Engine* dari lima parameter yakni *First Contentful Paint, First Meaningful Paint, Speed Index, First CPU Idle, Time to Interactive*, dan *Estimated Input Latency*. *Lighthouse* memberikan skor performa antara 0 dan 100 yang dihitung secara *overall* dari lima parameter tersebut[11]. Skor mengindikasikan skor paling rendah. Skor 0 biasanya mengindikasikan ada *error* pada *Lighthouse*. Skor 100 adalah skor terbaik yang merepresentasikan persentil ke 98 yang mengindikasikan performa terbaik dari suatu *web*site. Skor 50 dilain sisi merepresentasikan persentil ke 75[11]. Terdapat 6 metrik yang digunakan untuk menghasilkan skor total dari *Performance Score* diantaranya:

1. First Contentful Paint

First Contentful Paint (FCP) mengukur waktu dari navigasi ke waktu ketika browser me*render* bit pertama suatu konten dari DOM (*Document Object Model*). Hal ini merupakan tahapan penting untuk pengguna karena FCP menyediakan umpan balik bagaimana halaman *web* di load secara *actual*[11].

2. First Meaningful Paint

First Meaningful Paint (FMP) mengukur berapa lama konten utama dari suatu *web* akan ditampilkan. Bagian audit ini mengidentifikasi bagaimana pengguna mempersepsi kapan konten utama benar-benar ditampilkan[11].

3. Speed Index

Speed Index menunjukan seberapa cepat suatu halaman terlihat secara jelas. Semakin kecil skornya maka semakin baik[11].

4. First CPU Idle

Metrik ini mengukur kapan suatu halaman secara minimal mulai interaktif. First CPU Idle menandai waktu awal saat *thread* utama siap menangani input. [11] .

5. Time to Interactive

Time to Interactive merupakan jumlah waktu yang dibutuhkan untuk suatu halaman untuk menjadi interaktif seluruhnya[11]. Hal ini dapat diindikasikan dari hal berikut:

1. Halaman menampilkan useful content yakni diukur dari *First Contentful Paint*.
2. Semua event handler telah tersedia untuk setiap elemen visual pada halaman.
3. Halaman merespon interaksi user minimal dalam 50ms.

6. Estimated Input Latency

Estimated Input Latency merupakan estimasi berapa lama waktu yang dibutuhkan *web*site untuk merespon input pengguna dalam *milisecond*, selama loading halaman. Jika latensi lebih tinggi dari 50ms makan pengguna akan mempersepsikan *web*site tersebut *laggy*. [11].

2. Opportunities

Opportunities merupakan peluang potensial yang didapat dari hasil audit yang dapat digunakan untuk mempercepat suatu halaman *web*site. Berikut adalah peluang *Opportunities* dan rekomendasi untuk dapat melakukan optimasi. [11].

1. Enable Text Compression

Enable Text Compression minimalisir ukuran bit dari respon jaringan yang mengikutsertakan konten teks. Ukuran yang kecil untuk di unduh akan membuat halaman *web*site menjadi lebih cepat untuk ditampilkan. [11].

2. ***Eliminate render-blocking resources***

Render blocking resources memblok *paint* pertama dari halaman *web*site, sehingga *browser* harus menunggu proses unduh dari aset sebelum bisa melakukan *rendering* seluruh berkas HTML. Hal ini akan membuat proses penampilan halaman menjadi lambat. [11].

3. ***Remove Unused CSS***

CSS yang tidak digunakan hanya akan menambah beban pada *networks trips* yang secara signifikan akan meningkatkan waktu pemuatan halaman sebelum pengguna dapat melihat konten di layer. [11].

4. ***Remove Defer Offscreen images***

CSS Offscreen *images* adalah *images* yang tampak diluar *viewport* atatau dimensi *layer* pengguna. Karena pengguna tidak dapat melihat *offscreen images* ketika mereka memuat halaman, jadi tidak ada alasan untuk mengunduh images yang tidak kelihatan tersebut diproses awal pemuatan halaman karena hanya akan memperlambat proses pemuatan halaman dan *time to interactive* [11].

5. ***Serve Images in Next-Generation Formats.***

Format images *Web*p memiliki teknologi kompresi yang lebih superior dan memiliki kualitas dibandingkan format lama seperti JPEG dan PNG. Meng*encode* image ke format terbaru berarti akan mengurangi konsumsi data seluler [11] .

6. ***Properly size Images***

Format Terapkan ukuran yang tepat untuk gambar pada layer pengguna. Ukuran *images* yang besar hanya akan mengabiskan bit dan melambatkan pemuatan halaman. [11] .

7. ***Minify JS dan CSS***

Minifikasi berkas JSS dan CSS akan sangat mengurangi waktu pemuatan halaman. [11] .

8. ***Reduce server response times (TFTB)***

Time To First Byte mengidentikasi waktu yang dibutuhkan *server* untuk memberikan respon. Respon yang lama akan membuat pengguna tidak menyukai suatu website. [11] .

9. ***Optimize Image***

Pengoptimasian image akan membuat halaman dimuat dengan cepat dan mengurangi konsumsi data seluler [11] .

10. Avoid Multiple page Redirect

Redirect 301 atau 302 yang terlalu banyak akan menambah *delay* pada halaman yang ingin dimuat. [11] .

11. Use Video formats for animated content

File GIF tidak efesien dalam mengirim *content* animasi. Gunakan MPEG4/*Web*M sebagai format animasi dan video dan PNG/*Web*P untuk gambar statis daripada GIF untuk menghemat *network bit* [11] .

3. Diagnostics

Diagnostics menyediakan informasi lebih detail tentang performa aplikasi suatu *web*site. [11] .

1. Serve Statics assets with an efficient cache policy

Penerapan kebijakan cahing menggunakan *HTTP Caching* dapat mempercepat proses pemuatan halaman untuk kunjungan berulang dari user [11] .

2. Ensure text remains visible during webfont load.

Menampilkan teks ketika menunggu proses *webfon*t di muat hanya akan memperlambat proses pemuatan [11] .

3. Avoid an execessive DOM size..

Browser enginer merekomendasikan ukuran DOM Node kurang dari 1500. Dan *tree depht* < dari 32 dan kurang dari 60 *children/parent element*. DOM yang besar akan sangat berdampak pada performa [11] .

1. Efesiensi *network* dan *load performance*.
2. *Runtime performance.*
3. *Memory performances.*
4. Minimalisir

4. Ensure text remains visible during webfont load.

Menampilkan *text* ketika menunggu proses webfont di muat hanya akan memperlambat proses pemuatan [11] .

5. Minimize main-thread work.

Kurangi waktu *parsing, compiling* dan eksekusi dari javascript [11] .

6. Reduce Javascript Execution time..

Network payload yang besar membebani *user* dengan biaya dan menciptakan pemuatan halaman yang lama karena harus mengirim jumlah total berkas yang sangat besar [11] .

7. Reduce JavaScript Execution time

Network payload yang besar membebani user dengan biaya dan menciptakan pemuatan halaman yang lama karena harus mengirim jumlah total berkas yang sangat besar [11] .

2.2.12 Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan sebuah aplikasi, bahasa pemograman atau lainnya yang membantu programmer dalam membangun sebuah sistem / program. Tanpa adanya perangkat lunak pendukung ini memungkinkan para programmer kesulitan dalam membuat sistem atau program yang akan dibuatnya. [11]

Berikut merupakan beberapa perangkat lunak pendukung yang diperlukan dalam penelitian Analisis Perbandingan : [11]

2.2.12.1 *Chrome DevTools*

Chrome DevTools adalah satu set alat untuk menulis dan men-*debug web* yang menjadi bawaan Google Chrome. Chrome DevTools sangat berguna untuk melakukan audit terhadap *web*site untuk menganalisa *networking,* keamanan, *performance,*dll. Salah satu *engine* yang digunakan oleh *Chrome DevTools* dalam mengukur performansi suatu *web* adalah *Lighthouse*[4].

2.2.12.2 *Lighthouse*

Lighthouse adalah sumber terbuka, alat otomatis untuk meningkatkan kinerja, kualitas, dan kebenaran aplikasi *web* Anda. Saat mengaudit halaman, *Lighthouse* menjalankan rentetan tes terhadap halaman, dan kemudian menghasilkan laporan tentang seberapa baik halaman itu. Dari sini Anda dapat menggunakan tes gagal sebagai indikator tentang apa yang dapat Anda lakukan untuk meningkatkan aplikasi Anda[11].

2.2.12.3 Plesk Onyx

Plesk adalah sebuah *control panel* yang berguna untuk memudahkan pengguna dalam memanage *server*. Terutama untuk kebutuhan *hosting* seperti membuat *Web Hosting, Email, DNS management,* hingga *CMS installer.*

Dibuat pertama kali di Novosibirsk, Rusia pada awal 2001 dan terus berkembang hingga saat ini. Plesk merupakan *Control Panel* yang cukup terkenal di dunia *hosting* setelah CPanel[4].

Plesk mampu berjalan pada banyak *Platform* terkenal seperti Windows, Centos, Redhat, Debian, dan CloudLinux. Tidak hanya itu, Plesk juga mampu mensupport beberapa *Virtual Server* seperti HyperV, Docker, Virtuozzo, dan OpenVZ. Dengan banyaknya Platform yang di*support*, Plesk merupakan pilihan tepat jika kita ingin membangun *server* dengan Control Panel Hosting[4].

2.2.12.4 Wordpress

WordPress adalah sebuah aplikasi *open source* yang sangat populer digunakan sebagai *blog engine*. WordPress dibangun dengan bahasa pemrograman PHP, database MySQL, PHP dan MySQL, keduanya merupakan *open source software*.[4] Selain sebagai blog, WordPress juga mulai digunakan sebagai sebuah CMS (*Content Management System*) karena kemampuannya untuk dimodifikasi dan disesuaikan dengan kebutuhan penggunanya[4].

WordPress adalah penerus resmi dari b2/cafelog yang dikembangkan oleh Michel Valdrighi.[5] Nama WordPress diusulkan oleh Christine Selleck, teman Matt Mullenweg[5]. WordPress saat ini menjadi platform content management system (CMS) bagi beberapa situs *web* ternama seperti CNN, *Reuters*, The New York Times, TechCrunch, dan lainnya [6].

2.2.12.5 HTML *(HyperText Markup Language)*

HTML kependekan dari Hypertext Markup Language adalah bahasa markup (penanda) berbasis text atau bisa juga disebut sebagai *formatting language* (bahasa untuk memformat), Jadi sudah jelas bahwa *HTML* bukanlah bahasa pemrograman, melainkan bahasa markup/formatting [21].

HyperText Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. *HTML* merupakan pengembangan dari standar pemformatan dokumen teks, yaitu *Standard Generalized Markup Language (SGML)* [21].

Berdasarkan pengertian *HTML* menurut Rian Ariona dan Taryana Suryana, maka dapat disimpulkan *HTML* adalah sebuah bahasa markup/formatting yang digunakan untuk menuliskan halaman *web*. Jadi intinya *HTML* ini adalah bukan bahasa pemograman melainkan sebuah bahasa formatting teks.

2.2.12.6 CSS *(Cascading Style Sheet)*

CSS adalah kependekan dari *Cascading Style Sheet*, berfungsi untuk mempercantik penampilan *HTML* atau menentukan bagaimana elemen *HTML* ditampilkan, seperti menentukan posisi, merubah warna teks atau *background* dan lain sebagainya [21].

CSS (Cascading Style Sheet) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *web*site, baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan. Pada umunya *CSS* digunakan untuk memformat halaman *web* yang ditulis dengan *HTML* atau *XHTML* [21].

Berdasarkan pengertian CSS menurut Rian Ariona dan Taryana Suryana, maka dapat disimpulkan CSS adalah sebuah bahasa *stylesheet* yang digunakan untuk mempercantik tampilan HTML atau suatu *web*site mulai dari tata letak, jenis huruf, warna, background dan lainnya.

2.2.12.7 PHP *(Hypertext Processor)*

PHP adalah suatu bahasa *Scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *Server side scripting*, maka untuk menjalankan *PHP* harus menggunakan *web server* [22].

PHP merupakan bahasa pemograman yang digunakan untuk membuat program *web*site dimana kode program yang telah dibuat dikompilasi dan dijalankan pada sisi *Server* untuk menghasilkan halaman *web* yang dinamis [22].

Berdasarkan pengertian *PHP* menurut Prianto Hidayatullah dan Wahana Komputer, maka dapat disimpulkan *PHP* adalah sebuah scripting bahasa pemograman yang digunakan untuk membuat *web*site yang dijalankan pada sisi *Server*.

2.2.12.8 *Javascript*

Javascript adalah bahasa pemrograman *web* yang berjalan disisi *Client*/*Browser*. *Javascript* biasa digunakan untuk memanipulasi element-element *HTML* dan menambahkan *Style* secara otomatis atau lebih sederhananya membuat dokumen *HTML* menjadi lebih Interaktif [21].

JavaScript adalah bahasa script berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada satu dokumen *HTML*. Dimana objek tersebut dapat berupa suatu *window, frame, URL, dokumen, form, button* atau item lainnya. Yang semuanya itu mempunyai property yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi warna nilai dan atribut lain [21].

Berdasarkan pengertian *Javascipt* menurut Rian Ariona, Taryana Suryana dan Prianto Hidayatullah, maka dapat disimpulkan bahwa Javascript adalah bahasa scripting pada suatu *web* yang berjalan pada sisi *client* untuk memanipulasi elemen pada *HTML* agar lebih terlihat interaktif.

2.2.12.9 *MySQL*

My*SQL* adalah salah satu aplikasi *DBMS (Database Management System)* yang sudah sangat banyak digunakan oleh para pemograman aplikasi *web.* Kelebihan dari *MySQL* ini adalah gratis, handal, selalu di-update dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi *DBMS* yang sering dibundling dengan *web Server* sehingga proses instalasinya jadi lebih mudah [23].

My*SQL* merupakan salah satu perangkat lunak untuk sistem manajemen database *SQL*. My*SQL* diciptakan oleh David Axmark, Allan Larson, dan Michael Widenius. My*SQL* juga merupakan perangkat lunak gratis dibawah lisensi GPL (*General Public License*), tetapi lisensi My*SQL* juga dijual untuk kasus-kasus tertentu karena penggunaannya tidak cocok dengan penggunaan perangkat GPL [23].

Berdasarkan pengertian *MySQL* menurut Prianto Hidayatullah dan Wahana Komputer, maka dapat disimpulkan My*SQL* merupakan salah satu aplikasi *DBMS*

yang banyak digunakan oleh programmer *web* untuk menyimpan *database* karena gratis, handal, selalu *update*.

2.2.12.10 SQL *(Structured Query Languange)*

SQL merupakan singkatan dari *Structured Query Language*. *SQL* atau juga sering disebut sebagai *query* merupakan suatu bahasa (*language*) yang digunakan untuk mengakses *database*. Standar ini tidak tergantung pada mesin yang digunakan (*IBM, Microsoft atau Oracle*). Hampir semua *software database* mengenal atau mengerti *SQL*. [23]