

BAB II

TINJAUAN PUSTAKA

2.1 PT. Indomarco prismatama

Indomaret atau PT Indomarco Prismatama adalah jaringan retail waralaba di Indonesia. Indomaret merupakan salah satu anak perusahaan Salim Group. Indomaret merupakan jaringan minimarket yang menyediakan kebutuhan pokok dan kebutuhan sehari-hari dengan luas area penjualan kurang dari 200 m². Toko pertama dengan nama Indomart dibuka di Ancol, Jakarta Utara, pada tanggal 20 Juni 1988, dikelola oleh PT. Indomarco Prismatama.

Tahun 1997 perusahaan mengembangkan bisnis gerai waralaba pertama di Indonesia, setelah memiliki lebih dari 230 gerai. Jumlah gerai hingga tahun 2015 adalah 11.400 gerai dengan rincian 60% gerai adalah milik sendiri dan sisanya waralabamilik masyarakat.

Sampai dengan awal tahun 2017, jumlah gerai sebanyak 13.000 toko. Mitra usaha waralaba ini meliputi: koperasi, badan usaha dan perorangan. Indomaret tersebar merata dari Sumatra, Jawa, Madura, Bali, Lombok, Kalimantan dan Sulawesi.

Pada tahun 2016 PT. Indomarco prismatama didirikan di Kota Gresik dengan mencakup area toko pengiriman dalam tiap harinya dilakukan pengiriman dari Distribution center ke toko-toko indomaret di area Gresik.

Proses ekspedisi barang dari Pusat Distribution center ke toko Indomaret tujuan masih memakai metode konvensional manual menggunakan surat jalan kertas yang nantinya akan di serahkan ke driver dan driver tersebut melakukan ekspedisi ke tujuan berdasarkan jalan yang sering dilalui tanpa memperhatikan optimasi waktu serta jarak terdekat.

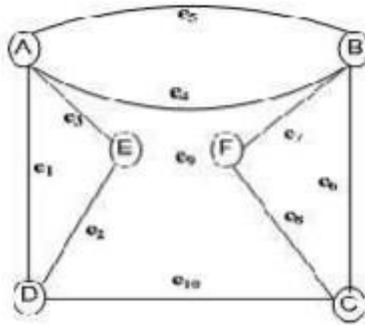
Dalam proses ekspedisi barang ke tujuan terdapat banyak jalan dan persimpangan, masing-masing persimpangan dapat dipresentasikan sebagai *vertex*, dan masing-masing jalan bisa dipresentasikan sebagai *edge*, sedangkan jarak antar persimpangan dapat di definisikan sebagai bobot dari *edge* tersebut

Sehingga data tersebut bisa digunakan untuk menentukan jarak terpendek dari titik awal pengiriman barang Distribution center ke toko indomaret yang dituju maupun dari toko indomaret ke toko indomaret lainnya dengan menggunakan Algoritma Dijkstra.

2.2 Graph

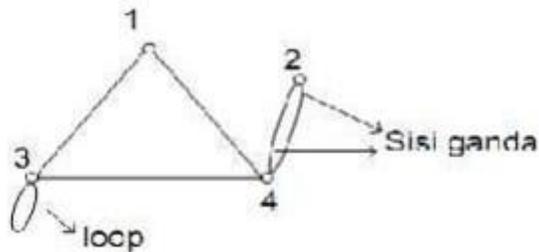
Menurut Wilson (1990) Suatu graph G terdiri atas himpunan yang tidak kosong dari elemen –elemen yang disebut titik (vertek), dan suatu daftar pasangan vertek yang tidak terurut disebut sisi (edge).

Himpunan vertek dari suatu graph G dinotasikan dengan V , dan daftar himpunan edge dari graph tersebut dinotasikan dengan E . Untuk selanjutnya suatu graph G dapat dinotasikan dengan $G = (V, E)$ [5].



Gambar 2.1 Contoh graph

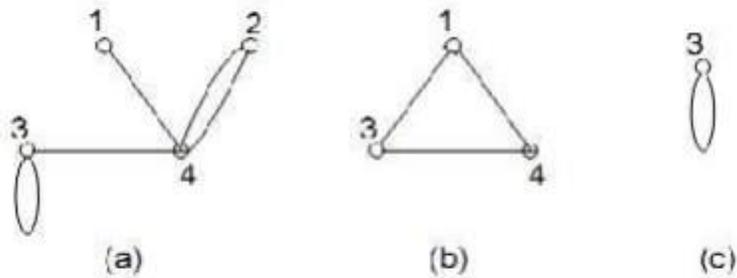
Gambar 2.1, menunjukkan graph G dengan $V = \{A, B, C, D, E, F\}$ dan $E = \{e1, e2, e3, \dots, e10\}$. Dua edge atau lebih yang menghubungkan pasangan vertek yang sama disebut sisi ganda, dan sebuah edge yang menghubungkan sebuah vertek kedirinya sendiri disebut loop



Gambar 2.2 Graph Sisi ganda dan Loop

Misal G suatu graph dengan himpunan vertek V dan himpunan edge E .

Suatu subgraph G' adalah suatu himpunan pasangan berurutan (V', E') dimana V' merupakan himpunan bagian dari V dan E' adalah himpunan bagian dari E . Dengan kata lain, subgraph dari G adalah suatu graph yang semua vertek-nya anggota V dan semua edge-nya anggota E . Jika G suatu graph terhubung seperti pada gambar 2.2, dengan $V = \{1, 2, 3, 4\}$ dan $E = \{(1,3), (1,4), (2,4), (3,3), (3,4), (4,2)\}$, maka berikutnya adalah contoh dari subgraph G' yang ditunjukkan pada gambar 2.3.



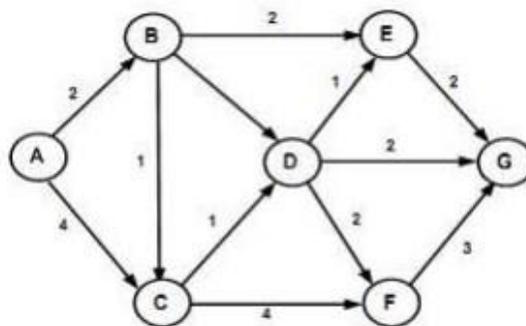
Gambar 2.3 Contoh graph G dan subgraph G'

Pada gambar 2.3 (a) merupakan subgraph G' dari graph G , dengan himpunan vertek $V' = \{1, 2, 3, 4\}$ yang merupakan himpunan bagian dari V dan himpunan edge $E' = \{(1,3), (1,4), (2,4), (3,3), (3,4), (4,2)\}$ yang merupakan himpunan bagian dari E . Gambar 2.3 (b) juga merupakan subgraph G' dari graph G dengan himpunan vertek $V' = \{1, 3, 4\}$ dan himpunan edge $E' = \{(1,3), (1,4), (3,4)\}$ yang masing – masing merupakan himpunan bagian dari V dan E . Gambar 2.3 (c) juga merupakan subgraph G' dari graph G dengan himpunan vertek $V' = \{3\}$ dan himpunan edge $E' = \{(3,3)\}$ yang masing – masing merupakan himpunan bagian dari V dan E .

2.2.1 Macam-macam graph menurut arah dan bobotnya

Menurut arah dan bobotnya, graph dibagi menjadi empat bagian, yaitu:

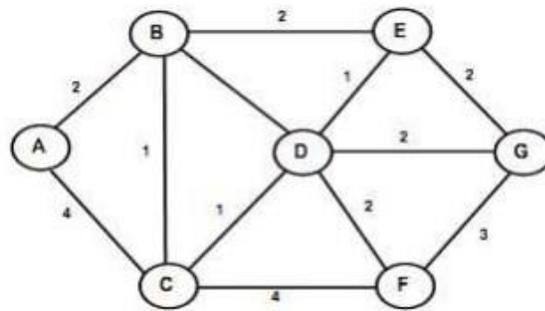
1. Graph berarah dan berbobot : setiap edge mempunyai arah (yang ditunjukkan dengan anak panah) dan bobot. Gambar 2.4 adalah contoh graph berarah dan berbobot yang terdiri dari tujuh vertek yaitu vertek A, B, C, D, E, F, G.



Gambar2.4Graph berarah dan berbobot

Vertek A mempunyai dua edge yang masing – masing menuju ke vertek B dan vertek C, vertek B mempunyai tiga edge yang masing – masing menuju ke vertek C, vertek D dan vertek E. Bobot antara vertek A dan vertek B pun telah di ketahui.

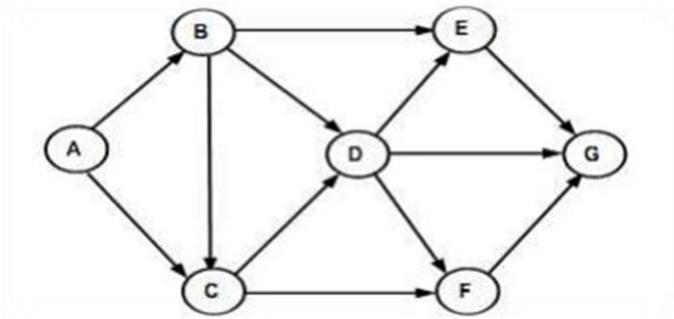
2. Graph tidak berarah dan berbobot : setiap edge tidak mempunyaiarah tetapi mempunyai bobot. Gambar 2.5 adalah contoh graph tidak berarah dan berbobot.



Gambar 2.5 graph tidak berarah dan berbobot

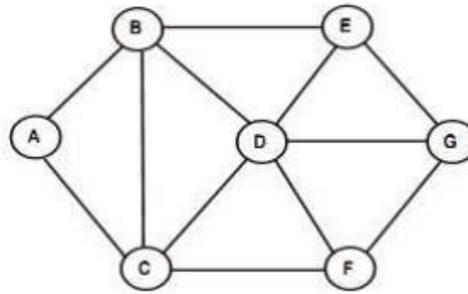
Graph terdiri dari tujuh vertek yaitu vertek A, B, C, D, E, F, G. Vertek A mempunyai dua edge yang masing – masing berhubungan dengan vertek B dan vertek C, tetapi dari masing – masing edge tersebut tidak mempunyai arah. Edge yang menghubungkan vertek A dan vertek B mempunyai bobot yang telah diketahui begitu pula dengan edge – edge yang lain.

3. Graph berarah dan tidak berbobot : setiap edge mempunyai arah tetapi tidak mempunyai bobot. Gambar 2.6 adalah contoh graph berarah dan tidak berbobot.



Gambar 2.6 graph berarah dan tidak berbobot

4. Graph tidak berarah dan tidak berbobot : setiap edge tidak mempunyai arah dan tidak terbobot. Berikut adalah contoh graph tidak berarah dan tidak berbobot.



Gambar 2.7 graph tidak berarah dan tidak berbobot

2.3 Optimasi

2.3.1 Definisi masalah optimasi

Optimisasi adalah suatu proses untuk mencapai hasil yang optimal (nilai efektif yang dapat dicapai). Dalam disiplin matematika optimisasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi riil. Untuk dapat mencapai nilai optimal baik minimal atau maksimal tersebut,

secara sistematis dilakukan pemilihan nilai variabel *integer* atau riil yang akan memberikan solusi optimal [6].

2.3.2 Definisi nilai optimal

Nilai optimal adalah nilai yang didapat melalui suatu proses dan dianggap menjadi solusi jawaban yang paling baik dari semua solusi yang ada [7].

2.3.3 Macam-macam Permasalahan Optimasi

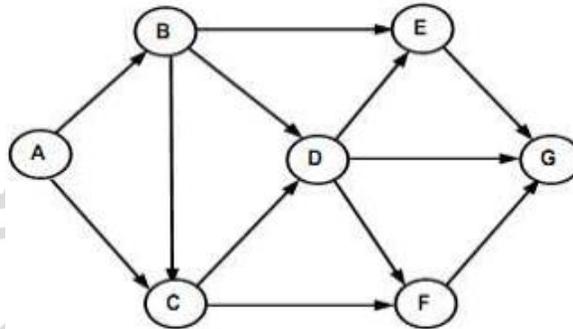
Permasalahan yang berkaitan dengan optimisasi sangat kompleks dalam kehidupan sehari-hari. Nilai optimal yang didapat dalam optimisasi dapat berupa besaran panjang, waktu, jarak, dan lain-lain. Berikut ini adalah termasuk beberapa persoalan optimisasi :

1. Menentukan lintasan terpendek dari suatu tempat ke tempat yang lain.
2. Menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal.
3. Mengatur rute kendaraan umum agar semua lokasi dapat dijangkau.
4. Mengatur routing jaringan kabel telepon agar biaya pemasangan kabel tidak terlalu besar dan penggunaannya tidak boros dan permasalahan-permasalahan lainnya.

2.3.4 Permasalahan rute terpendek

Masalah rute terpendek merupakan masalah yang berkaitan dengan penentuan edge dalam sebuah jaringan yang membentuk rute terdekat antara sumber

dan tujuan. Tujuan dari permasalahan rute terpendek adalah mencari rute yang memiliki jarak terdekat antara titik asal dan titik tujuan. Gambar 2.9 merupakan contoh *graph* yang berarah dan tidak berbobot.



Gambar 2.8 *graph* ABCDEFG

Berdasarkan *graph* diatas, misalkan kita dari kota A ingin menuju Kota G. Untuk menuju kota G, dapat dipilih beberapa rute yang tersedia :

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow G$

$A \rightarrow B \rightarrow E \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow G$

$A \rightarrow C \rightarrow F \rightarrow G$

Berdasarkan data diatas, dapat dihitung rute terpendek dengan mencari jarak antara rute-rute tersebut. Apabila jarak antar rute belum diketahui, jarak dapat dihitung berdasarkan koordinat kota kota tersebut, kemudian menghitung jarak terpendek yang dapat dilalui.

2.3.5 Penyelesaian masalah optimasi

Secara umum, penyelesaian masalah pencarian rute terpendek dapat dilakukan dengan menggunakan dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional dihitung dengan perhitungan matematis biasa, sedangkan metode heuristik dihitung dengan menggunakan sistem pendekatan.

1. Metode Konvensional

Metode konvensional adalah metode yang menggunakan perhitungan matematika eksak. Ada beberapa metode konvensional yang biasa digunakan untuk melakukan pencarian rute terpendek, diantaranya: algoritma *Dijkstra*, algoritma *Floyd-Warshall*, dan algoritma *BellmanFord* (Mutakhirah, I., Saptono, F., Hasanah, N., dan Wiryadinata, R.,2007).

2. Metode Heuristik

Metode Heuristik adalah suatu metode yang menggunakan sistem pendekatan dalam melakukan pencarian dalam optimasi. Ada beberapa algoritma pada metode heuristik yang biasa digunakan dalam II-8

permasalahan optimasi, diantaranya *Algoritma Genetika, Ant Colony Optimization, logika Fuzzy, jaringan syaraf tiruan, Tabu Search, Simulated Annealing*, dan lain-lain [8]

2. 4 Definisi algoritma

Kata algoritma (algorithm) diambil dari kata algorism yang berasal dari nama seorang ilmuwan dari Arab yang terkenal yaitu Abu Jafar Muhammad Ibnu Musa Al Khuwarizmi dan oleh orang barat dilafalkan menjadi Algorism. Pengertian algorism itu sendiri adalah sekumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis untuk penyelesaian suatu masalah. Berikut adalah bagian-bagian dari algoritma:

1. Komponen algoritma

Algoritma memiliki 5 komponen urutan yaitu finiteness (terbatas), definiteness (Kepastian), input, output dan efektivitas Dalam merancang algoritma ada tiga komponen yang harus dimiliki yaitu :

- a. Komponen Masukkan (Input)
- b. Komponen Keluaran (Output)
- c. Komponen Proses (Processing)

2. Notasi algoritma

Penulisan algoritma terkadang sulit untuk dipahami dan maksud dari algoritma tersebut. Selain itu juga sulit untuk menuliskan algoritmanya. Agar mempermudah bisa dilakukan notasi – notasi algoritma. Notasi algoritma

merupakan rancangan penyelesaian masalah (algoritma) yang dituliskan dalam notasi (tata cara penulisan). Notasi algoritma yang sering dijumpai ada 3 macam yaitu :

1. Diagram alir (flowchart)
2. Deskriptif
3. *Pseudo-code*

3. Aturan penulisan teks algoritma

Algoritma merupakan deskripsi langkah – langkah penyelesaian dari sebuah permasalahan. Deskripsi tersebut bisa dituliskan ke dalam bentuk notasi apapun, yang penting mudah dibaca dan dipahami. Tidak ada notasi yang baku dalam penulisan teks algoritma. Tiap orang dapat membuat aturan penulisan dan algoritma sendiri. Namun, agar notasi algoritma dapat dengan mudah ditranslasi ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritma itu berkoresponden dengan notasi bahasa pemrograman secara umum. Teks algoritma dinotasikan ke dalam tiga bagian (blok) : judul (header) algoritma, deklarasi, dan deskripsi.

2. 5 Macam-macam algoritma

2.5.1 Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu varian dari algoritma greedy, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang

terkait dengan masalah optimasi. Sifatnya sederhana dan lempang (straight forward). Sesuai dengan artinya yang secara harafiah berarti tamak atau rakus – namun tidak dalam konteks negatif –, algoritma greedy ini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan. Prinsipnya, ambillah apa yang bisa Anda dapatkan saat ini dan keputusan yang telah diambil pada setiap langkah tidak akan bisa diubah kembali. Intinya algoritma greedy ini berupaya membuat pilihan nilai optimum lokal pada setiap langkah dan berharap agar nilai optimum lokal ini mengarah kepada nilai optimum global.

Elemen-elemen penyusun algoritma djikstra adalah:

1. Himpunan kandidat, C

Himpunan ini berisi elemen-elemen yang memiliki peluang untuk membentuk solusi. Pada persoalan lintasan terpendek dalam graf, himpunan kandidat ini adalah himpunan simpul pada graf tersebut.

2. Himpunan solusi, S

Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat

3. Fungsi seleksi

Fungsi seleksi adalah fungsi yang akan memilih setiap kandidat yang memungkinkan untuk menghasilkan solusi optimal pada setiap langkahnya

4. Fungsi kelayakan

Fungsi kelayakan akan memeriksa apakah suatu kandidat yang telah terpilih (terseleksi) melanggar constraint atau tidak. Apabila kandidat melanggar constraint maka kandidat tidak akan dimasukkan ke dalam himpunan solusi.

5. Fungsi objektif

Fungsi objektif akan memaksimalkan atau meminimalkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.

2. 5. 2 Algoritma *Floyd-Warshall*

Algoritma Floyd-Warshall adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.

Hal yang membedakan pencarian solusi menggunakan pemrograman dinamis dengan algoritma greedy adalah bahwa keputusan yang diambil pada tiap tahap pada algoritma greedy hanya berdasarkan pada informasi yang

terbatas sehingga nilai optimum yang diperoleh pada saat itu. Jadi pada algoritma greedy, kita tidak memikirkan konsekuensi yang akan terjadi seandainya kita memilih suatu keputusan pada suatu tahap. Dalam beberapa kasus, algoritma greedy gagal memberikan solusi terbaik karena kelemahan yang dimilikinya tadi. Di sinilah peran pemrograman dinamis yang mencoba untuk memberikan solusi yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap.

Pemrograman dinamis mampu mengurangi pengenumerasian keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh pemrograman dinamis adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke- i) juga optimal.

Karakteristik algoritma floyd warshall adalah:

1. Persoalan dibagi atas beberapa tahap, yang setiap tahapnya hanya akan diambil satu keputusan.
2. Masing-masing tahap terdiri atas sejumlah status yang saling berhubungan dengan status tersebut. Status yang dimaksud di sini adalah berbagai kemungkinan masukan yang ada pada tahap tersebut
3. Ketika masuk ke suatu tahap, hasil keputusan akan transformasi
4. Ongkos (beban) pada suatu tahap akan meningkat secara teratur seiring bertambahnya jumlah tahapan

5. Ongkos yang ada pada suatu tahap tergantung dari ongkos tahapan yang telah berjalan dan ongkos pada tahap itu sendiri
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan pada tahap sebelumnya
7. Terdapat hubungan rekursif yang menyatakan bahwa keputusan terbaik dalam setiap status pada tahap k akan memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$
8. Prinsip optimalitas berlaku pada persoalan yang dimaksud.

Dalam proses penyelesaian menggunakan program dinamis, pendekatan yang dilakukan bisa jadi ada dua macam, yaitu pendekatan maju (forward) dan mundur (backward), dan perlu untuk diketahui pula bahwa solusi yang dihasilkan dari kedua pendekatan itu adalah sama. Solusi dari program dinamis bisa jadi lebih dari satu macam.

2. 5. 3 Algoritma *Belman ford*

Algoritma Bellman-Ford adalah algoritma untuk menghitung jarak terpendek (dari satu sumber) pada sebuah digraf berbobot. Maksudnya dari satu sumber ialah bahwa ia menghitung semua jarak terpendek yang berawal dari satu titik node. Algoritma Dijkstra dapat lebih cepat mencari hal yang sama dengan syarat tidak ada sisi(edge) yang berbobot negatif . Maka Algoritma Bellman Ford hanya digunakan jika ada sisi berbobot negative Dalam algoritma Bellman-Ford, apabila bilangan dicari lintasan dengan bobot

paling sedikit dari satu ke dua, maka lintasannya adalah 1-4-3-2, sehingga bobot yang didapat adalah 7-3 -2 - 2 .

2.6 Website

Website atau biasa dibilang situs merupakan kumpulan dari berbagai macam data yang bisa berupa gambar, suara, ataupun video yang terhubung dan saling terkait satu sama yang lainnya. Website awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan user atau pengguna dalam meperoleh informasi.

Berdasarkan sifatnya website dibedakan menjadi dua jenis, yaitu website statis dan website dinamis. Website statis merupakan website yang isi informasinya bersifat tetap, jaran update dan isi informasinya searah dari pemiliknya. Contoh dari website statis adalah website company profile dan website profil organisasi.

Sedangkan website dinamis isi webnya selalu berubah - berubah, uptodate dan interaksinya bersifat interaktif dari pemilik website maupun pengguna atau user website. Contoh dari website dinamis antara lain adalah facebook, olx, detik, wikipedia. Faktor yang menjadikan sebuah website menjadi dinamis adalah Content Management System atau CMS yang memiliki akses ke administrator secara langsung sehingga dapat memutakhirkan isinya dengan sangat mudah.

2.7 Web Browser

Web Browser adalah perangkat lunak yang berguna untuk mengakses informasi web ataupun untuk melakukan transaksi via web. Awalnya web browser berorientasi pada teks dan belum dapat menampilkan gambar. Akan tetapi sekarang ini dengan dikembangkannya web browser tidak hanya mampu menampilkan gambar dan teks saja, tetapi juga dapat memutar file multimedia berupa foto, video,12 maupun audio. Beberapa contoh web browser yang ada saat ini seperti : Google Chrome, Internet Explorer (IE), Mozilla Firefox, Opera, netscep, dan Safari.

Web Browser pertama bernama Word Wide Web yang diciptakan Tim Berners-Lee, kemudian diubah menjadi nexus. Pada tahun 1993, Marc Andreessen melakukan inovasi web browser dengan membuat Mosaic yang sekarang di kenal dengan Netspace. Hal tersebut membuat sistem internet mejadi mudah untukdigunakan dan mudah untuk di akses oleh banyak orang. Pada tahun 1994 Netspace ini menjadi browser terpopuler di dunia karena menguasai 90% dari seluruh pengguna web browser dunia.

Pada tahun 1995, Microsoft menciptakan Internet Exproler dan pada tahun 2002 mendominasi di pasar web browser dunia dengan lebih dari 95% pengguna. Kemudian pada tahun 1998 Netspace meluncurkan kembali yang menjadi Mozilla Firefox. Lalu pada tahun 2003 Appel mengeluarkan web browser yaitu Safari, dan Google Chrome menjadi pendatang baru di pasar web browser yang dirilis pada September 2008.