

OPTIMASI KERJA WEB DENGAN MENGGUNAKAN MODEL ADAPTIF WAP PROXY PADA MOBILE SERVER

Agung Kridoyono

Program Teknik Informatika, Fakultas Teknik
Universitas 17 Agustus 1945 Surabaya
Jl. Semolowaru No.45 Surabaya
E-mail : akridoyono@gmail.com

ABSTRAK

Terbatasnya bandwidth dan kurangnya pemanfaatan program dinamik web berbasis wireless mobile untuk meningkatkan kerja di layer transport diperlukan untuk menjalankan berbagai layanan yang minim infrastruktur jaringan datanya seperti daerah non-urban . Pada paper ini akan dibahas pembuatan suatu server proxy yang bersifat adaptif dimana server ini selalu menyesuaikan kondisi kemampuan perangkat client dan koneksinya dengan memanfaatkan parsing yang berbasis mobile programming (WAP). Dengan penggunaan model wap mobile ini framework dapat dioptimasi kecepatannya dan konten yang diakses dapat ditampilkan kualitasnya menurut dengan kemampuan jaringan atau mobile profile client access (perangkat akses request data server).

Kata kunci: *Mobile Adapted Proxy*, Pemrograman Dinamik Server Perangkat Bergerak, *Wireless Application Protocol*

ABSTRACT

Limited bandwidth and lack of the use of the dynamic web program based mobile wireless for optimization in transport layer Is required to run the range of public services with minimal data network infrastructure such as the non urban area. This paper will be explained an adaptive server proxy which the proxy server always adjust automatically the ability of the client and the Link by using parsing based mobile programming (WAP). With a mobile WAP model in this framework proxy, It can be optimized the speed and the content that being accessed can be shown the quality According with the ability of network or mobile profile client access.

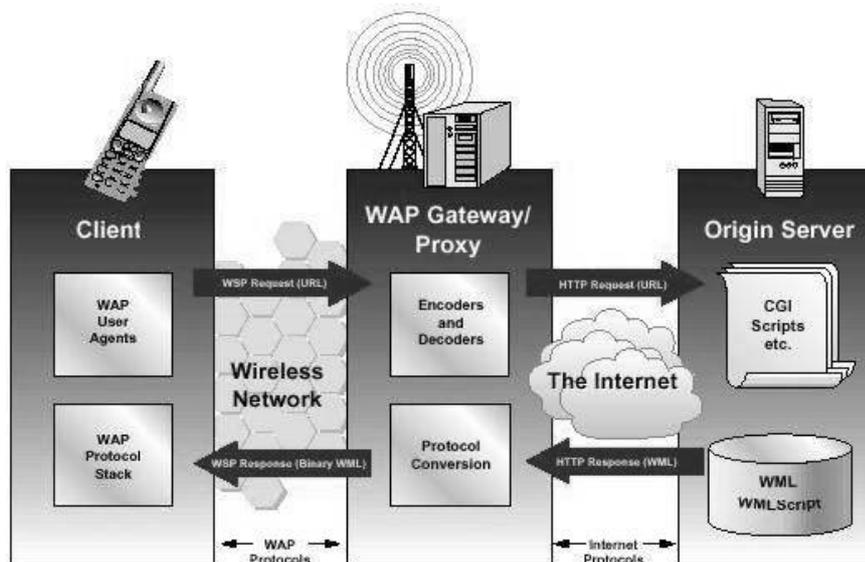
Keywords: *mobile adapted proxy, mobile server programming, WAP access protocol*

PENDAHULUAN

Pengembangan teknologi mobile bergerak bertahap dan sangat cepat dengan diikuti berbagai model yang berurutan seperti midp1.2, wap 1.0 wap 2.0 dimana wap 2.x memiliki fasilitas yang mendukung protokol seperti IP, TCP, HTTP [1], serta bahasa program pendukung arsitektur yang selalu menyesuaikan dan bertransformasi, ini dikarenakan operasional perangkat bergerak berbasis wireless sehingga koneksi data selalu menjadi prioritas

dimana kecepatan akses ini yang dijadikan ukuran.

Ada banyak teknik optimasi lain untuk komunikasi, seperti yang berhubungan dengan transmisi multimedia dan *layer* presentasi atau model format data yang disediakan. Konten multimedia untuk teknik beberapa coding, scalable dan berlapis coding, telah diciptakan untuk menangani kondisi jaringan heterogen dan variabel, seperti membuat Mobeware [2] dan RAPIDware [3].



Gambar 1. Arsitektur jaringan pada koneksi wap akses proxy.

Klasifikasi pendekatan berbasis proxy, seperti Dikaiakos [4] menulis survey yang sangat menarik tentang infrastruktur berbasis proxy yang dikhususkan untuk web. Penulis mengajukan klasifikasi pendekatan proxy dalam tiga dimensi yaitu arsitektur sistem, fungsi, dan interaksi. Mengenai arsitektur sistemnya penulis membedakan antara arsitektur sentral dan distribusi, pilihan untuk penempatan proxy dan konfigurasi proxy atau programming proxy. Arsitektur jaringan pada koneksi wap akses proxy dapat dilihat pada Gambar 1.

Sejak proxy digunakan untuk memegang penyesuaian dan adaptasi pada tingkat berbagai perangkat lunak, kita yakin akan kriteria yang sesuai dengan proxy ini. Dalam pandangan

kami pada umumnya proxy dapat digunakan pada tiga level:

- a. level dalam komunikasi, pada level ini proxy mengendalikan semua jenis persoalan yang berhubungan dalam protokol komunikasi dan pemecahan. Tujuan utamanya adalah membuat perangkat mobile transparansi wireless link ke lapisan software yang lebih tinggi. Tipe penyesuaian pada level ini adalah protokol untuk kabel dan nirkabel dalam hal translasi atau optimasi, buffering, manajemen pengambilan keputusan dsb, seperti yang ada pada proposal TCP jaringan wireless [2] dan wireless CORBA [5].
- b. Level middleware, pada level ini melakukan tugas umum maupun menyesuaikan pada tipe aplikasi yang

ditentukan tidak juga berhubungan dengan protokol komunikasi. Contoh pada beberapa bentuk penyesuaian konten [6,3] pengaturan yang konsisten pada data *cache* [7,8] layanan atau menemukan *resources*[9,10].

- c. Level aplikasi, beberapa arsitektur proxy ini difokuskan pada aplikasi tipe tertentu seperti web browsing [11,12,13], akses database [14], *P2P sharing* [15] dan yang lain. pada kasus ini, proxy mengeksekusi tugas yang disesuaikan pada kebutuhan tertentu dan fungsi pada aplikasinya sebagai contoh ketika dibandingkan *cache* pada aplikasi web dan database bentuk pemegang obyek yang heterogen pada dasarnya bertujuan mengurangi waktu respon, sementara itu diperlukan manajemen konsistensi *cache*.

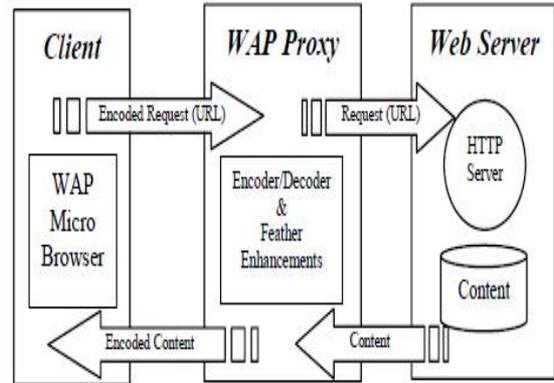
METODE

Arsitektur yang digunakan memiliki model tier 3 (*client, WAP proxy, web server tier*) [1], dengan penggunaan gateway serta proxy sehingga request akan dilakukan oleh client dan *dihandle* oleh server proxy adaptif untuk dioptimasi model perangkat aksesnya dan jaringannya dengan memanfaatkan *mobile information device profile* (midp).

Tree tier architecture model dapat dilihat pada Gambar 2. Proses pada Gambar 2 memiliki penjelasan komponen antara lain:

- a. Server merupakan pusat konten yang diminta oleh klien dimana server tersebut ada yang memiliki fasilitas kompresi ataupun kelengkapan format data parser yang dimiliki oleh berbagai klien.
- b. WAP proxy Gateway
Menerima request dari client : disini merupakan gateway operator yang memiliki fasilitas berbeda-beda dimana kebijakan yang diambil sesuai dengan layanan yang diberikan menangani response, response ini dari

klien berupa inputan yangmana dapat berupa sms, telepon, atau data. Pada data ini maka dimasukkan jalur internet atau data, berbagi dengan jalur multimedia sms saat berada di gateway response ini.



Gambar 2. Tree tier architecture model

HTML Parser terjadi pada lapisan middle tier atau WAP proxy server, memecah dokumen html menjadi XHTML, XML, WBXML. Mengenali suatu kode HTML, seperti: memperbaiki tag HTML, mengecek link maupun ekstrak informasi misal dokumen html seperti ditunjukkan pada Gambar 3.

```
<head>
  <title>' 'Selamat
tes' ' HTML</title>

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;title&gt;#039;#039;
;#039;Selamat
```

Gambar 3. Cek doctype format konten

Penggunaan Document Analyzer Menganalisa hasil pecahan dokumen. Deteksi suatu halaman informasi. Bentuk HTML Parser import merupakan *filter* dokumen dari format request data dari server yang dituju ke server proxy untuk dilakukan *matching* terhadap client profile access. Proses parser pada server dari dokumen server ke proxy dapat dilihat pada Gambar 4.

Client merupakan komponen profile akses ke web server yang dituju melalui wap proxy dan client ini merupakan titik akhir hasil penyesuaian resource terhadap konten yang dituju sampai dimana dan sampai beberapa maksimal fasilitas resource yang dimiliki client disitu wap proxy akan melakukan *tag suitable component programming*.

Disisi perangkat lunak sendiri merupakan pendukung arsitektur server memiliki kualitas berbeda-beda dalam menjalankan operasinya dimana bahasa yang berbasis mobile memang lebih memiliki karakteristik berbeda dibanding dengan bahasa program berbasis desktop, dan pada paper ini akan sedikit diulas uji perhitungan perangkat lunak penyusun

server antara bahasa python dengan bahasa PHP.

Fast CGI protokol CGI sebagai antarmuka pada aplikasi lain ke web server yang merupakan fasilitas alternatif untuk meningkatkan performansi CGI untuk proses aplikasi web server dimana disini adalah web server apache terhadap bahasa dinamis lain. Proses penanganan dari CGI ke fastCGI dapat dilihat pada Gambar 5. Untuk itu penggunaan fasilitas ini mendukung kerja:

- python sebagai bahasa dinamis yang digunakan
- module fastCGI untuk server yang akan digunakan pada caching proxy RFC2109

| | | |
|---|---|---|
| <pre>class MyHTMLParser(HTMLParser): def handle_starttag(self, tag, attrs): print "Encountered a start tag:", tag def handle_endtag(self, tag): print "Encountered an end tag :", tag def handle_data(self, data): print "Encountered some data :", data # instantiate the parser and fed it some HTML parser = MyHTMLParser() parser.feed('<html><head><title>Test</ti tle></head>' '<body><h1>Parse me!</h1></body></html>')</pre> |  | <pre>Encountered a start tag: html Encountered a start tag: head Encountered a start tag: title Encountered some data : Test Encountered an end tag : title Encountered an end tag : head Encountered a start tag: body Encountered a start tag: h1 Encountered some data : Parse me! Encountered an end tag : h1 Encountered an end tag : body Encountered an end tag : html</pre> |
|---|---|---|

Gambar 4. Proses parser pada server dari dokumen server ke proxy

```
mobile<VirtualHost *:80>
  ServerName test
  DocumentRoot /var/www/test/htdocs/
  <Directory />
    Options FollowSymLinks +ExecCGI
    AllowOverride All
  </Directory>
  <Files app.py>
    SetHandler fastcgi-script
  </Files>
  FastCgiServer /var/www/test/htdocs/app.py -processes 4 -socket
/tmp/fcgi.sock
</VirtualHost>
```

Gambar 5. Contoh Proses penanganan dari CGI ke fastCGI

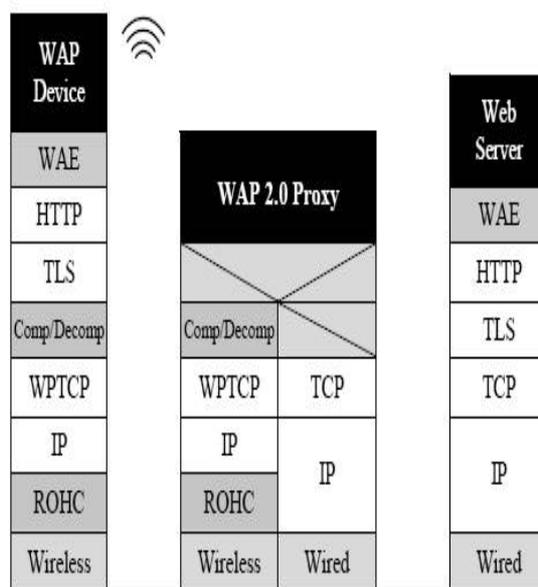
WAP proxy based redirect dapat dilakukan. Untuk HTTP request bisa diproses oleh mod_fastcgi, request's handler harus fastcgi-script atau request MIME type harus application/x-httpd-fcgi. Konfigurasi virtual host agar yang dilakukan dapat berinteraksi dengan dinamik web yaitu python yang digunakan sebagai bahasa programnya.

Alur komunikasi dari WAP proxy ini tetap teracu pada protocol mobile / protocol perangkat bergerak dalam melakukan komunikasi datanya. Dimana pada proxy ini dilakukan komunikasi antar proxy antara proxy server dengan proxy operator atau disebut juga gateway operator, gateway ini juga termasuk operator selular atau penyedia layanan internet yang lain seperti *indiehome* Telkom. Jadi yang paling terlengkap MIME tipe nya adalah teradapat pada proxy ini, karena merupakan penjemabatan antara server yang dituju dengan client pengakses data ke server. Server yang memiliki format presentasi yang lengkap maka proxy tinggal meneruskan ke client dan menyesuaikan *source profile client* saja.

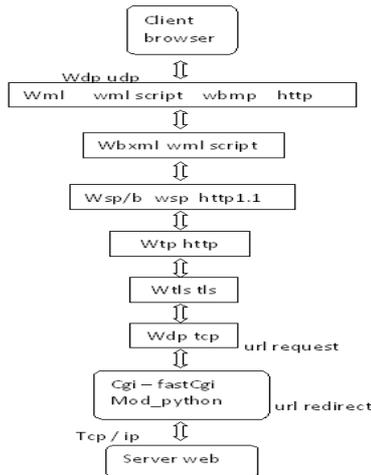
Client diberikan default markup language berbasis wireless dengan default ini maka otomatis diarahkan menjadi format binary XML sehingga pengaktifan protocol WBXML dilakukan setelah didapat format binary ini maka protokol kirim teks atau HTTP dapat terdekati dengan menggunakan port 80 misalnya pada HTML maka komunikasi kirim teks berbasis WEB dapat dilakukan. Gambar 6. Penyesuaian antara HTML WML ini yang terjadi proses didalam proxy ini dan arsitektur redirect ini merupakan arsitektur yang dipilih pada arsitektur menurut *open mobile alliance* berbasis komunikasi WAP proxy agar kerja alur komunikasi lebih ringan dan proses dua tipe komunikasi WEB WAP dapat dijalankan beriringan melalui perantara CGI server yang dibentuk menggunakan fastCGI untuk kerja khusus mobile Wireless data akses.

Prokolnya seperti pada Gambar 7 tetap menggunakan protocol TCP/IP dengan sifat connection oriented maka alur pertukaran data dapat selalu terjadi beriringan dan koreksi antar dua modul dapat dilakukan CGI – fastCGI pada server dan data pada MIME type dilakukan untuk database informasi tipe format data yang melalui server yang dilakukan transaksi. Proses ini seperti layakny translasi dari HTML- XHTML- XML-WBXML WML yang terjadi pada layer protocol network proxy dengan dibentuk bahasa penyusunnya berbasis mobile python yang kecepatannya melebihi program web desktop saat melakukan perhitungan atau pada bahasa lain perhitungan counter pada PC saat diregister instruksi (IR).

Konfigurasi wap 2.X ini merupakan alur tes komunikasi yang dilakukan dengan menggunakan server dan komunikasi secara PPP menggunakan modul wvdial untuk melihat apakah proxy di server sudah aktif proxynya, dengan menggunakan PPP yang dilakukan menggunakan operator selular dan testing menggunakan *dectkit* emulator untuk komunikasi WAP seperti yang ada pada gambar 9 maka komunikasi wireless mobile dapat dijalankan.



Gambar 6. Konfigurasi proxy WAP 2.x



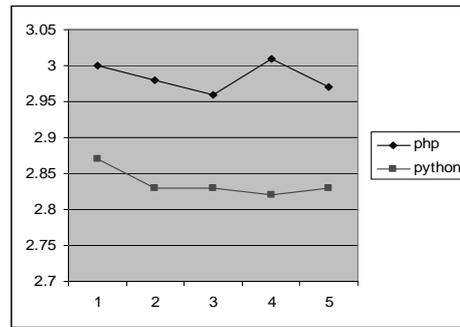
Gambar 7. Model protocol komunikasi wap adaptif proxy

HASIL DAN PEMBAHASAN

Pengujian dalam melakukan perhitungan dasar pada web dinamis programming yaitu php dan python pengujian dilakukan terlebih dahulu pada PHP sebanyak lima kali percobaan. Setelah itu, barulah percobaan dilakukan pada python, juga sebanyak lima kali. Percobaan dilakukan dalam kondisi penggunaan CPU oleh sistem berada di bawah 1%.

Berikut ini adalah hasil percobaan dalam satuan detik. Hasil pada Gambar 8 merupakan hasil uji suatu skrip yang dilakukan oleh dinamik

programming dimana pengujiannya melakukan perhitungan 1-1000 secara mode counter dan hasilnya dapat diperoleh python memiliki keunggulan kecepatan dalam perhitungan / counter. Jenis source ini yang digunakan untuk pembentukan antarmuka yang diletakkan pada proses modul CGI berjenis fastCGI.



Gambar 8. Hasil perbandingan perhitungan hit dinamik program

Hasil pada proses tersebut memiliki tampilan yang bermacam-macam hasilnya, ini dikarenakan profile yang mengakses pada server berbeda-beda tingkat sourcena dan spesifikasinya sehingga proxy menyesuaikan apa yang didapat. Berikut ini sampel hasil yang dilakukan pada situs yang berbeda-beda. User perceived latency (UPL): seberapa cepat interval waktu saat client request data sampai menerima data.



Gambar 9. Model sampel tampilan pada proxy dan hasil paket proxy adaptif.

Tabel 1. Data akses proxy mobile

| No. | Jam 24:00 pada menit | Lama waktu akses dalam detik | |
|-----|----------------------|------------------------------|--------------------------|
| | | Server domestic (s) | Server internasional (s) |
| 1 | 0-4 | 20 | 35 |
| 2 | 5-8 | 24 | 40 |
| 3 | 9-13 | 19 | 43 |
| 4 | 14-18 | 14 | 42 |
| 5 | 15-19 | 23 | 39 |
| 6 | 20-24 | 17 | 37 |

Tabel 2. Data akses web

| No. | Jam 24:00 pada menit | Lama waktu akses dalam detik | |
|-----|----------------------|------------------------------|--------------------------|
| | | Server domestic (s) | Server internasional (s) |
| 1 | 0-4 | 35 | 49 |
| 2 | 5-8 | 30 | 47 |
| 3 | 9-13 | 26 | 48 |
| 4 | 14-18 | 34 | 50 |
| 5 | 15-19 | 33 | 37 |
| 6 | 20-24 | 37 | 41 |

Tabel 3. Hasil perhitungan perbandingan dua sistem akses domestik site

| NO | wap | Web | selisih akses |
|----|---|-------------|---------------|
| 1 | 20 | 35 | -15 |
| 2 | 24 | 30 | -6 |
| 3 | 19 | 26 | -7 |
| 4 | 14 | 34 | -20 |
| 5 | 23 | 33 | -10 |
| 6 | 17 | 37 | -20 |
| | sample mean | | -13 |
| | sample variance | | 39.2 |
| | sample standard dev | | 6.26099034 |
| | Confidence interval 90%,5 degree of freedom | | 2.01504837 |
| | | lower bound | -18.1705731 |
| | | upper bound | -7.82942689 |

Tabel 4. Hasil perhitungan Perbandingan dua sistem akses internasional site

| NO | wap | web | selisih akses |
|----|---|-------------|---------------|
| 1 | 35 | 49 | -14 |
| 2 | 40 | 47 | -7 |
| 3 | 43 | 48 | -5 |
| 4 | 42 | 50 | -8 |
| 5 | 39 | 37 | 2 |
| 6 | 37 | 41 | -4 |
| | sample mean | | -6 |
| | sample variance | | 27.6 |
| | sample standard dev | | 5.25357021 |
| | Confidence interval 90%,5 degree of freedom | | 2.01504837 |
| | | lower bound | -10.3386058 |
| | | upper bound | -1.66139422 |

SIMPULAN

Dari pembahasan penelitian yang sudah diuraikan sebelumnya, maka dapat ditarik kesimpulan bahwa:

1. Pada penelitian ini menghasilkan Perancangan Sistem Informasi Inventori BBM yang dapat menampilkan stock volume secara akurat.
2. Staf Divisi Sales / Marketing tidak lagi harus menunggu konfirmasi laporan dari divisi Stock / Cargo untuk menyusun draft Sales Budget.
3. Pencatatan proses bisnis yang dilakukan perusahaan dapat ditelusuri dengan mudah sehingga memudahkan staff dalam melaksanakan tugas-tugasnya.

DAFTAR PUSTAKA

- [1] Yin, Z., Leung, C.M., Victor, A *Proxy Architecture to Enhance the Performance of WAP 2.0 by Data Compression*, Department of Electrical and Computer Engineering The University of British Columbia Vancouver, 2005.
- [2] Angin, O., Campbell, A.T., Kounavis, M.E., Liao, R.R.-F. "The Mobiware Toolkit: Programmable Support for Adaptive Mobile Networking", *IEEE Personal Communications Magazine, Special Issue on Adapting to Network and Client Variability*, 1998.
- [3] McKinley, P.K., Udiyan, I., Padmanabhan, Design of composable proxy filters for wireless collaborative computing, Technical Report MSU-CSE-00-26, Department of Computer Science, Michigan State University, East Lansing, Michigan, 2000.
- [4] Fox, S., Gribble, Y., Chawathe., Brewer, E., "Adapting to network and client variation using active proxies: Lessons and perspectives", *IEEE Personal Communications*, 1998.
- [5] Song, J., Zhang Y., *Architecture of a Web Accelerator for Wireless Networks*, the Thirtieth Australasian Computer Science Conference (ACSC), Ballarat, Australia, 2007
- [6] Harini, B., Anupam, J., Sansanee, A., "An active transcoding proxy to support mobile web access", In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems*, 1998.
- [7] Spyrou, C., Samaras, G., Pitiura. E., Evripidou, P., "Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies", *Mobile Network and Applications*, 9(5):517–528, 2004.
- [8] Satyanarayanan, M., Kistler, J., Kumar, P., Okasaki, M.E., Siegel, E.H., Steere, D.C., Coda: A highly available file system for a distributed workstation environment, *IEEE Transactions on Computers*, 39:447–459, 90.
- [9] Barron C. Housel dan David B. Lindquist. Webexpress: a system for optimizing web browsing in a wireless environment. In *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 108–116. ACM Press, 1996.
- [10] Anurag Kahol, Sumit Khurana, Sandeep K.S. Gupta, dan Pradip K. Srimani. A strategy to manage cache consistency in a disconnected distributed environment. *IEEE Trans. Parallel Distrib. Syst.*, 12(7):686–700, 2001.
- [11] Herman Chung-Hwa Rao, Di-Fa Chang, Yih-Farn Chen, dan Ming-Feng Chen. iMobile: a proxy-based platform for mobile services. In *ACM Workshop on Wireless Mobile Internet (WMI01), Rome*, pages 3–10, Juli 2001.
- [12] E. Brewer dan et al. A network architecture for heterogeneous mobile computing. *IEEE Personal Communications Magazine*, oktober 1998.
- [13] Robertson, A.,. Wireless Application Architectures for Business: Key Implementation Consideration. *Cutter IT Journal*, March 2001, Vol.14, No. 3, 2001.
- [14] Issa, A. *Supporting Mobile Databases By Translating Traditional Web Pages Into Wireless Markup Language (Wml)Pages*. MSc Thesis. University of Jordan, 2003.
- [15] M. Liljeberg, T. Alanko, M. Kojo, H. Laamanen, dan K Raatikainen. Optimizing world-wide web for weakly connected mobile workstations: An indirect approach. In *Proc. 2nd International Workshop on Services in Distributed and Networked Environments (SDNE'95), Whistler, Canada*, June 1996