

# KOMPONEN WEB DATA ANALYZER PADA IE STUDI KASUS: AKSES WEB TERFAVORIT LABORATORIUM IBS TEKNIK INFORMATIKA - ITS

**Darlis Heru Murti - Airlangga Eka W. – Fajar Baskoro**

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember,

Email: darlis@its-sby.edu, angga00@inf.its-sby.edu, fajarb@its-sby.edu

## ABSTRAK

Perkembangan dunia internet baik dari segi teknologi maupun jumlah pengguna yang mengakses kian lama semakin bertambah. Hal ini juga berdampak dengan munculnya berbagai situs internet yang menawarkan berbagai hal serta tujuan yang berbeda. Untuk mengakses situs-situs tersebut diperlukan suatu alat yang disebut browser. Salah satu browser yang paling terkenal dan banyak digunakan saat ini adalah Microsoft Internet Explorer (IE). Kalau diperhatikan sebenarnya browser Internet Explorer telah menyediakan fitur khusus bernama Favorites (dapat diakses melalui menu atau toolbar) untuk mencatat dan mengorganisasi web-web yang menjadi favorit user. Tetapi fitur Favorites ini kurang fleksibel karena pemasukan data maupun pengorganisasiannya masih bersifat manual oleh user. Untuk itu perlu diimplementasikan perangkat lunak dengan fitur Favorites yang bersifat otomatis dan dinamis sehingga untuk periode waktu maupun lokasi akses workstation tertentu bisa memiliki informasi favorit website yang berbeda-beda. Sebagai solusi yang dapat digunakan adalah dengan memanfaatkan teknologi rekayasa perangkat lunak yaitu pembuatan komponen yang melekat pada browser Internet Explorer atau lebih dikenal dengan istilah plugin Internet Explorer. Untuk itu di dalam pelaksanaan penelitian ini, akan dilakukan perancangan dan pembuatan sebuah perangkat lunak komponen Web Data Analyzer yang melekat pada browser Internet Explorer untuk pencarian akses web terfavorit pengguna. Uji coba dan evaluasi pada penelitian ini dilakukan dengan melakukan instalasi komponen Web Data Analyzer pada sejumlah workstation di Laboratorium IBS Teknik Informatika ITS. Hasil uji coba menunjukkan bahwa komponen Web Data Analyzer mampu memonitor dan menganalisa data aktivitas browsing pengguna serta melakukan otomatisasi terhadap fitur Favorites Internet Explorer dari data aktivitas browsing pengguna yang berhasil tersimpan ke database server.

**Kata kunci:** band object, explorer bar, browser helper object (bho), http analyzer.

## 1. PENDAHULUAN

Perkembangan dunia internet baik dari segi teknologi maupun jumlah user yang mengakses kian lama semakin bertambah. Hal ini juga berdampak dengan munculnya berbagai situs internet yang menawarkan berbagai hal serta tujuan yang berbeda.

Untuk mengakses situs-situs tersebut diperlukan suatu alat yang disebut browser. Salah satu browser yang paling terkenal dan banyak digunakan saat ini adalah Microsoft Internet Explorer (IE).

Kalau diperhatikan sebenarnya browser Internet Explorer telah menyediakan fitur khusus bernama Favorites (dapat diakses melalui menu atau toolbar) untuk mencatat / mengorganisasi web-web yang menjadi favorit pengguna. Tetapi fitur Favorites ini kurang fleksibel karena pemasukan data maupun pengorganisasiannya masih bersifat manual oleh pengguna. Untuk itu perlu diimplementasikan perangkat lunak dengan fitur Favorites yang bersifat otomatis dan dinamis sehingga untuk periode waktu maupun lokasi akses workstation tertentu bisa memiliki informasi web favorit yang berbeda-beda.

Tujuan dari perancangan dan pembuatan perangkat lunak ini adalah:

- Melekat dan terintegrasi pada browser Internet Explorer dalam bentuk komponen dan aktif memonitor akses URL pengguna.
- Menyimpan dan mengambil informasi URL, pengguna yang mengakses, waktu akses, serta lokasi akses ke dalam database server.
- Menganalisa dan menampilkan laporan baik berupa grafik/diagram maupun teks kepada user untuk akses web terfavorit pengguna tersebut.

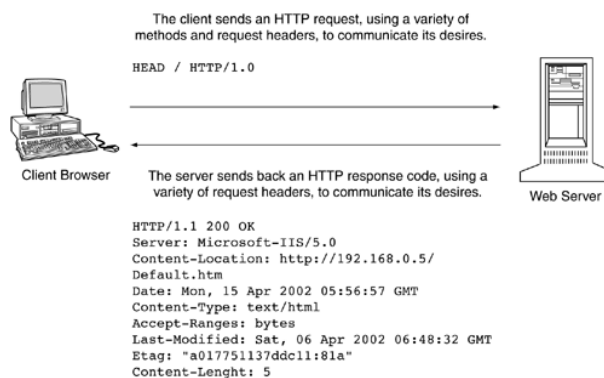
Permasalahan yang diangkat dalam pembuatan perangkat lunak ini adalah:

- Bagaimana mengimplementasikan pembuatan komponen Web Data Analyzer yang berintegrasi dengan browser Internet Explorer.
- Bagaimana mendapatkan informasi URL, login, waktu dan lokasi akses user yang menggunakan browser Internet Explorer.
- Bagaimana membuat laporan statistik berupa teks maupun diagram/grafik untuk mendapatkan akses web terfavorit berdasarkan kategori login, waktu dan lokasi akses user.

## 2. HYPERTEXT TRANSFER PROTOCOL (HTTP)

HTTP adalah protokol transfer *Web* standard. HTTP merupakan protokol ASCII dan setiap interaksi terdiri dari sebuah ASCII *request* mematuhi *response* MIME RFC 822.

Protokol HTTP terdiri dari dua hal, yaitu permintaan (*request*) dari *browser* ke *server* dan *respon* yang kembali dari *server* ke *browser*. Setiap situs *Web* memiliki proses *server* yang mengamati koneksi yang masuk dari *client* (biasanya *browser*) pada TCP port 80. Setelah koneksi ditetapkan, *client* mengirim *request* dan *server* mengirim jawabannya, dan kemudian koneksi dilepaskan.



Gambar 1. Protokol HTTP

## 3. UNIFORM RESOURCE LOCATOR (URL)

URL merupakan sebuah mekanisme untuk mengenali sumber-sumber pada *web*, SSL, atau *server* ftp, termasuk protokol layer aplikasi yang membuat *request* ke server web. Struktur umum URL adalah:

Protokol://server/path/to/resource?parameters

Tabel 1. Komponen-komponen URL

Komponen	Keterangan
Protokol	Protokol lapisan aplikasi. Kegunaan URL yang paling umum adalah meminta sumber-sumber dari Web server (server HTTP). Karena itu protokol yang paling umum adalah https:, ftp:, ldap:, telnet:, pop3:, dst, tergantung dukungan server dan browser apa yang dipakai.
Server	Nama DNS, nama Netbios, atau alamat IP dari sebuah host atau jaringan yang menjadi host sumber
Path/to/resource	Direktori path, termasuk nama sumber yang diminta. Sumber bisa dalam bentuk file statis atau sebuah aplikasi yang secara dinamis

Parameter	Secara optional, parameter dapat dilewatkan ke sumber bila ia adalah sebuah aplikasi atau sebuah program yang secara dinamis membangkitkan output. Kadang kala bagian URL yang mengkhususkan parameter disebut juga Query String.
-----------	---

## 4. EXPLORER BAR

Explorer Bar diperkenalkan pada Microsoft® Internet Explorer 4.0 untuk menyediakan area tampilan yang berdekatan dengan *browser pane*. Pada dasarnya Explorer Bar merupakan *child window* dalam *window* Internet Explorer, dan dapat dipergunakan untuk menampilkan informasi dan berinteraksi dengan pengguna. Explorer Bar biasanya ditampilkan sebagai *pane* vertikal di sebelah kiri *browser pane* atau ditampilkan secara horizontal, di bawah *browser pane*.



Gambar 2. Contoh Explorer Bar

Ada banyak kemungkinan penggunaan untuk Explorer Bar. Pengguna dapat memilih pilihan mana yang mereka ingin lihat dengan beberapa cara yang berbeda, termasuk memilihnya dari submenu Explorer Bar pada menu View, atau meng-klik tombol toolbar. Internet Explorer menyediakan beberapa Explorer Bar standar, termasuk Favorites dan Search.

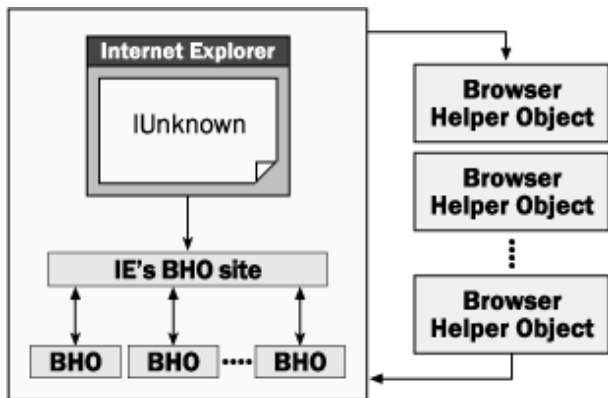
Untuk menciptakan Explorer Bar, pengguna harus mengimplementasikan dan mendaftarkan *band object*. *Band objects* diperkenalkan dengan shell versi 4.7 dan menyediakan kemampuan yang serupa dengan *windows* normal. Akan tetapi, karena *band object* adalah obyek COM dan dimuat (*contained*) baik oleh Internet Explorer atau Shell, mereka diimplementasikan sedikit berbeda. *Band objects* sederhana digunakan untuk membuat contoh Explorer Bar yang ditampilkan pada Gambar.

## 5. BROWSER HELPER OBJECT

Internet Explorer sama seperti program berbasis Win32 lainnya mempunyai tempat penyimpanan memori sendiri. Dengan Browser

Helper Object programmer dapat menciptakan komponen, khususnya komponen in-process Component Object Model (COM), yang akan di-load setiap kali Internet Explorer dijalankan. Obyek ini berjalan pada konteks memori yang sama seperti browser dan dapat melakukan action apapun pada window dan modul yang tersedia. Misalnya, BHO dapat mendeteksi event browser tipikal, seperti GoBack, GoForward, dan DocumentComplete; mengakses menu dan toolbar browser serta membuat perubahan; menciptakan window untuk menampilkan informasi tambahan pada halaman yang sedang dilihat sekarang; dan memasang hook untuk memonitor message dan action. Secara singkat, BHO bekerja sebagai mata-mata (spy) yang dikirimkan untuk menyusup (infiltrate) ke wilayah browser.

Ada beberapa hal yang harus diperjelas sebelum membahas BHO lebih jauh. Pertama, BHO terhubung ke window utama browser. Dalam kenyataannya, hal ini berarti bahwa instance baru obyek BHO akan diciptakan segera setelah window browser baru tercipta. Instance BHO manapun hidup dan mati bersama instance browser. Kedua, BHO hanya ditemukan pada Internet Explorer, versi 4.0 dan selanjutnya.



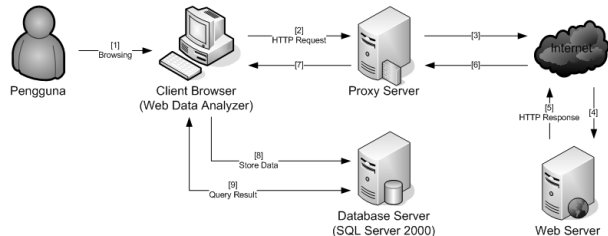
**Gambar 3. Proses inialisasi helper object pada Internet Explorer**

Secara sederhana, BHO adalah COM in-process server diregistrasikan pada registry key tertentu. Pada saat startup, Internet Explorer mencari key tersebut dan memuat semua obyek yang CLSID-nya disimpan disana. Browser menginisialisasi obyek dan meminta interface tertentu dari obyek tersebut. Jika interface tersebut ditemukan, Internet Explorer menggunakan metode yang disediakan untuk melewati pointer IUnknown-nya ke helper object. Proses ini diilustrasikan gambar diatas.

## 6. PERANCANGAN PERANGKAT LUNAK

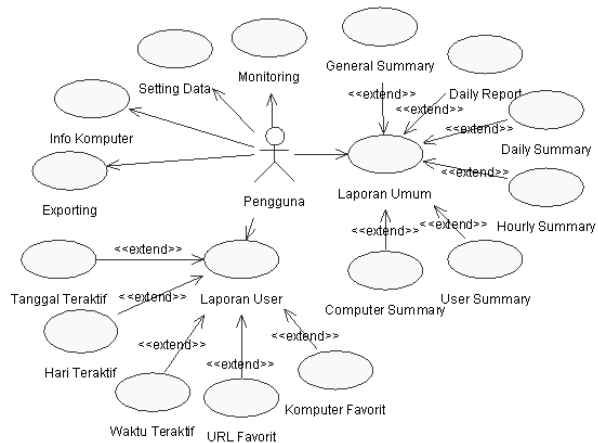
Aplikasi yang dibuat pada Penelitian ini digunakan untuk memonitor aktivitas *browsing* pengguna Laboratorium IBS Teknik Informatika - ITS. *Database server* yang digunakan sebagai

tempat penyimpanan sekaligus pengolahan data adalah Microsoft SQL Server 2000. Untuk pembahasan selanjutnya, aplikasi yang diimplementasikan dalam penelitian ini akan disebut Web Data Analyzer.



**Gambar 5. Gambaran Umum Sistem**

Secara umum proses dalam sistem Web Data Analyzer dijelaskan dengan *use case diagram* berikut:



**Gambar 4. Use Case Diagram**

Beberapa proses yang bisa dilakukan oleh pengguna dalam aplikasi Web Data Analyzer ini diantaranya adalah:

1. Proses *Monitoring* (memonitor aktivitas *browsing*)
2. Proses *Setting Data* (mengkonfigurasi setting aplikasi)
3. Proses *Info Komputer* (menampilkan informasi login dan *workstation*)
4. Proses *Exporting* (mengekspor hasil ke Favorites IE)
5. Proses *Reporting* (menampilkan hasil analisa *query* statistik) terdiri dari 2 kategori besar, yaitu Laporan User dan Laporan Umum.

Perancangan struktur data aplikasi Web Data Analyzer diimplementasikan dalam bentuk modul-modul *class* dengan masing-masing *class* memiliki properti dan metode untuk menjalankan fungsi tertentu dan saling berkaitan:

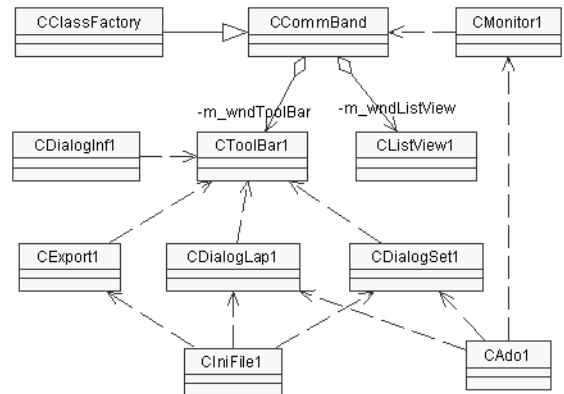
- CClassFactory, merupakan implementasi dari salah satu konsep utama pemrograman COM, yaitu ClassFactory. Class factory mempunyai tujuan utama menciptakan komponen lain

dengan mengimplementasikan *interface* standar *IClassFactory*.

- *CCommBand*, merupakan *class* induk dari semua *class* yang dibahas pada proses perancangan data ini. Tujuan utama *class* ini adalah untuk menciptakan komponen *band object* yaitu Explorer Bar serta menyediakan fungsi-fungsi umum untuk interaksi antara pengguna dengan aplikasi *Web Data Analyzer*.
- *CToolBar1*, digunakan untuk menciptakan kontrol toolbar. Kontrol toolbar berisi sejumlah ikon berteks yang merupakan daftar menu untuk menjalankan beberapa proses pada aplikasi *Web Data Analyzer*, misalnya *exporting* dan *reporting*.
- *CListView1*, digunakan untuk menciptakan kontrol listview. Kontrol listview berfungsi untuk menampilkan hasil proses *monitoring* *Web Data Analyzer* kepada pengguna selama aktivitas *browsing* berjalan.
- *CExport1*, merupakan representasi proses *Exporting* pada aplikasi *Web Data Analyzer*. *Class* ini menyediakan dua metode untuk mengatur proses ekspor URL terfavorit dari database ke menu Favorites browser IE
- *CMonitor1*, merupakan representasi proses Monitoring dari aplikasi *Web Data Analyzer*. *Class* ini menyediakan sejumlah properti dan metode yang dipergunakan untuk melakukan pemantauan (*observation*) dan penangkapan (*interception*) data-data aktivitas *browsing* yang dilakukan pengguna.
- *CDialogLap1*, digunakan untuk menciptakan dan mengatur kontrol dialog box yang berfungsi sebagai representasi tampilan proses *reporting* terhadap data aktivitas *browsing* pengguna yang berhasil tersimpan dalam *database*.
- *CDialogSet1*, merupakan representasi dari proses Setting Data pada aplikasi *Web Data Analyzer*. *Class* ini memiliki sejumlah metode untuk membaca, menampilkan dan mengubah konfigurasi aplikasi.
- *CDialogInf1*, menyediakan fungsi-fungsi pendukung yang digunakan untuk mencari informasi nama login pengguna, alamat IP dan nama komputer, serta nama domain dimana aplikasi *Web Data Analyzer* ini dijalankan.
- *CIniFile1*, menyediakan fungsi-fungsi pendukung yang digunakan untuk membaca, menampilkan, serta mengubah variabel-variabel pada file *DataWeb.ini* yang berisi konfigurasi dari aplikasi *Web Data Analyzer*.
- *CFavoritDB1*, merupakan *class* pendukung yang berhubungan langsung dengan *database* tempat penyimpanan proses *monitoring*. *Class* ini menyediakan metode-metode untuk berinteraksi dengan *database* Favorit,

misalnya: penambahan *database* dan penyaringan *database* berdasarkan kriteria.

Secara keseluruhan, class-class di atas saling berhubungan satu dengan yang lain. Visualisasi struktur dan relasinya dapat direpresentasikan menggunakan class diagram berikut.



Gambar 6. Class Diagram

Untuk keperluan penyimpanan data secara persistent, *Web Data Analyzer* menggunakan database Microsoft SQL Server. Model data konseptual *Web Data Analyzer* dapat dilihat pada gambar berikut.

Data	
DtDate	DATETIME
DtTime	Text(12)
WksUserName	Text(50)
WksIPAddress	Text(15)
WksCompName	Text(50)
WksDomainName	Text(50)
HttpMethod	Text(5)
HttpResult	Text(50)
URL	Memo
RedirectURL	Memo

Gambar 7. Model Data Konseptual

Urutan proses-proses yang terjadi pada use case diagram diatas, dijelaskan oleh sequence diagram berikut ini.

## 7. IMPLEMENTASI PERANGKAT LUNAK

### a. Implementasi *Band Object*

Prosedur dasar untuk menciptakan *band object* dengan kategori Explorer Bar adalah:

1. Implementasikan fungsi yg dibutuhkan oleh DLL.
2. Implementasikan *interface* COM yang dibutuhkan.
3. Implementasikan beberapa *interface* COM opsional yang diinginkan.
4. Registrasikan CLSID obyek.
5. Registrasikan obyek untuk kategori komponen yang sesuai.

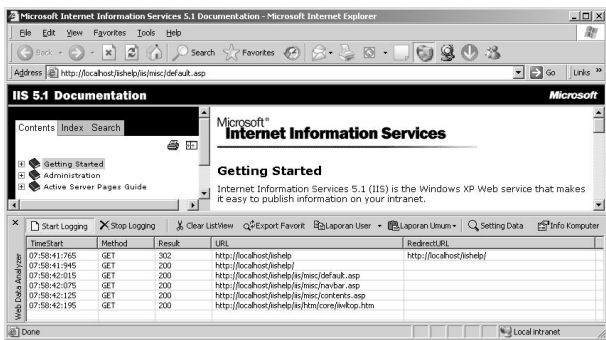
6. Buat *child window* Internet Explorer, berilah ukuran yang sesuai dengan daerah tampilan explorer bar.
7. Gunakan *child window* untuk menampilkan informasi dan berinteraksi dengan pengguna.

```

.....
//Register the comm band object.
if(!RegisterServer(CLSID SampleCommBand,
TEXT("&Web Data Analyzer")))
return SELFREG E CLASS;

//Register the component categories for the
//comm band object.
if(!RegisterComCat(CLSID SampleCommBand,
CATID CommBand))
return SELFREG E CLASS;
.....

```



Gambar 8. Proses Band Object

**b. Implementasi Proses Monitoring**

Pada proses *monitoring* ini, sistem melakukan pemantauan aktivitas *browsing* melalui komponen BHO (Browser Helper Objects). Pertama-tama BHO harus terhubung ke *browser* via *interface* *IConnectionPoint* dan dilewatkan ke tabel *IDispatch* dari fungsi yang selanjutnya akan menangani berbagai *event* tersebut.

```

.....
IoleCommandTarget* pCmdTarget;
hr = punkSite-
>QueryInterface (IID_IoleCommandTarget,
(LPVOID*) &pCmdTarget);

if (SUCCEEDED(hr))
{
IServiceProvider* pSP;
hr = pCmdTarget-
>QueryInterface (IID_IServiceProvider,
(LPVOID*) &pSP);
pCmdTarget->Release ();

if (SUCCEEDED(hr))
{
if (m_pFrameWB)
{
m_pFrameWB->Release ();
m_pFrameWB = NULL;
}

hr = pSP->QueryService (SID_SWebBrowserApp,
IID_IWebBrowser2,
(LPVOID*) &m_pFrameWB);

IConnectionPointContainer* pCPCContainer;

```

```

hr = m_pFrameWB->
QueryInterface (IID_IConnectionPointContainer,
(void*) &pCPCContainer);
if (SUCCEEDED(hr))
{
IConnectionPoint* m_pConnectionPoint;
hr = pCPCContainer->FindConnectionPoint (
DIID_DWebBrowserEvents2,
&m_pConnectionPoint);
if (SUCCEEDED(hr))
{
hr = m_pConnectionPoint->
Advise (static_cast<IDispatch*> (this),
&m_dwCookie);
if (FAILED(hr))
{
MessageBox (NULL, "Failed to Advise",
"C++ Event Sink", MB_OK);
}
}
}
pSP->Release ();
}
.....

```

TimeStart	Method	Result	URL	RedirectURL
07:58:41:765	GET	302	http://localhost/ishelp/	http://localhost/ishelp/
07:58:41:945	GET	200	http://localhost/ishelp/	http://localhost/ishelp/
07:58:42:015	GET	200	http://localhost/ishelp/ismsc/default.asp	
07:58:42:075	GET	200	http://localhost/ishelp/ismsc/navbar.asp	
07:58:42:125	GET	200	http://localhost/ishelp/ismsc/contents.asp	
07:58:42:195	GET	200	http://localhost/ishelp/ismsc/core/iewtop.htm	

Gambar 9. Proses Monitoring

**c. Implementasi Proses Setting Data**

Proses *Setting Data* sangat berpengaruh pada hasil proses analisa karena dua parameter yang diatur pada proses ini digunakan sebagai kriteria pencarian atau kondisi penyaringan untuk menyaring data aktivitas *browsing* yang telah tersimpan dalam *database*. Kedua kriteria tersebut adalah interval waktu dan batasan kuantitas (jumlah/persentase).

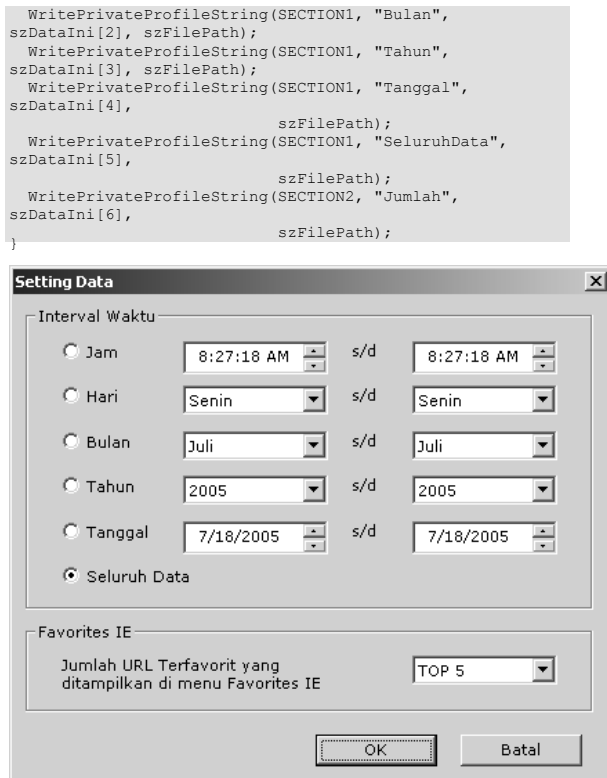
Dua metode penting yang digunakan dalam proses *setting* data adalah *ReadFileIni()* dan *WriteFileIni()* yang dipanggil ketika mengakses *class* *CIniFile1*. Metode *ReadFileIni()* berfungsi mendapatkan variabel-variabel konfigurasi dari *file* *DataWeb.ini* sedangkan *WriteFileIni()* digunakan untuk mengubah *setting* konfigurasi tersebut

```

void CIniFile1::ReadFileIni ()
{
GetPrivateProfileString (SECTION1, "Jam", "",
szDataIni [0],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION1, "Hari", "",
szDataIni [1],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION1, "Bulan", "",
szDataIni [2],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION1, "Tahun", "",
szDataIni [3],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION1, "Tanggal", "",
szDataIni [4],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION1, "SeluruhData", "",
szDataIni [5],
sizeof (szDataIni), szFilePath);
GetPrivateProfileString (SECTION2, "Jumlah", "",
szDataIni [6],
sizeof (szDataIni), szFilePath);
}

void CIniFile1::WriteFileIni ()
{
WritePrivateProfileString (SECTION1, "Jam",
szDataIni [0], szFilePath);
WritePrivateProfileString (SECTION1, "Hari",
szDataIni [1], szFilePath);
}

```



Gambar 10. Proses Setting Data

**d. Implementasi Proses *Exporting***

Proses *exporting* berfungsi untuk mengekspor hasil analisa query dari *database* ke menu Favorites *browser* Internet Explorer. Kasus analisa yang diekspor adalah daftar *web site* yang paling banyak diakses oleh pengguna dalam interval waktu tertentu.

```

BOOL CExport1::CreateFavoritFolder()
{
    .....
    // mencari nama login pengguna
    TCHAR szLogin[50];
    DWORD dwLogin = sizeof(szLogin)/sizeof(TCHAR);
    GetUserName(szLogin, &dwLogin);

    // Mengambil konfigurasi jumlah dari file DataWeb.ini
    CIniFile m_IniFile;
    m_IniFile.ReadFileIni();
    strcpy(szJumlah, m_IniFile.GetData(6));

    // Today
    wsprintf(szQuery[0], "SELECT %s URL, Count(URL) AS
    JumlahAkses \
    FROM Data \
    WHERE \
    Format(DtDate, 'dd/mm/yyyy')=Format(Now(), 'dd/mm/yyyy') \
    AND WksUserName='%s' \
    GROUP BY URL \
    ORDER BY Count(URL) DESC;", szJumlah, szLogin);
    .....

    // Mencari path folder Favorites IE
    HRESULT hr = SHGetMalloc(&pMalloc);
    if (SHGetSpecialFolderPath(NULL, szSpecialFolderPath,
    CSIDL_FAVORITES,
    TRUE) &&
    (SUCCEEDED(SHGetSpecialFolderLocation(NULL,
    CSIDL_FAVORITES, &pidlFavorites))))
    {
        pMalloc->Free(pidlFavorites);

        for(int i=0; i<5; i++)
        {
            wsprintf(szFolderPath, "%s\\%s",
            szSpecialFolderPath, szFolderName[i]);

            // melakukan checking Favorite IE
    
```

```

if (PathIsDirectory(szFolderPath))
{
    // membersihkan data Favorites lama
    DeleteAllFiles(szFolderPath);
}
else
{
    if (!CreateDirectory(szFolderPath, NULL))
    {
        return FALSE;
    }
}

_bstr_t strQuery(szQuery[i]);

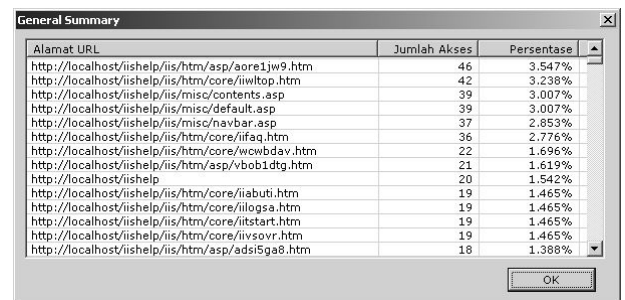
// mengakses database SQL Server
CFavoritDB1 m_FavoritDB;
m_FavoritDB.CreateConnection();
m_FavoritDB.CreateRecordSet();
m_FavoritDB.ExportToFavorites(strQuery,
szFolderPath);
m_FavoritDB.CloseDatabase();
}
else
{
    MessageBox(NULL, "Error: Favorites Folder Not Found",
    "Web Data Analyzer", MB_ICONEXCLAMATION |
    MB_OK);
    pMalloc->Free(pidlFavorites);
    return FALSE;
}
return TRUE;
}
    
```

**e. Implementasi Proses *Reporting***

Proses *reporting* ini berfungsi untuk menampilkan hasil analisa *query* kepada pengguna menggunakan tampilan windows standar melalui DialogBox sebagai media

Tahapan prosesnya adalah sebagai berikut:

1. Mendapatkan variabel-variabel konfigurasi
2. Mendefinisikan sintaks SQL sebagai parameter analisa *query* sesuai dengan jenis laporan yang diminta
3. Mengambil record hasil query dengan memanggil *class* CFavoritDB1.
4. Mengolah *query* dari sintaks SQL yang dikirimkan kemudian menampilkan hasilnya pada kontrol listview



Gambar 11. Proses Reporting

**8. UJI COBA & EVALUASI**

Untuk memastikan bahwa aplikasi Web Data Analyzer ini berjalan lancar, ada empat pertanyaan penting yang harus dijawab, yaitu:

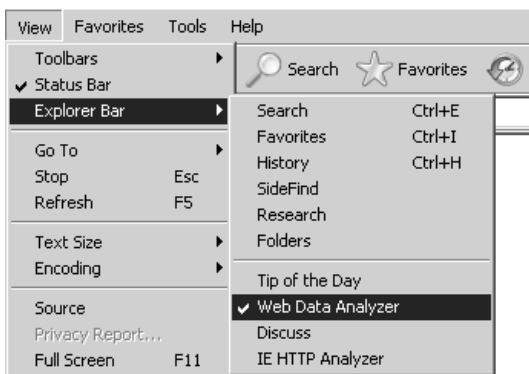
1. Apakah *band object* (Explorer Bar) sebagai komponen dasar dari aplikasi Web Data Analyzer mampu tercipta (*construct*), musnah (*destruct*) dan berintegrasi (*integrate*) dengan *browser* Internet Explorer?

2. Apakah aplikasi Web Data Analyzer dapat memonitor informasi aktivitas *browsing* pengguna lalu merekam informasi tersebut ke dalam *database*?
3. Apakah proses analisa *query* baik dengan atau tanpa penggunaan kriteria penyaringan memperoleh laporan hasil sesuai dengan data aktivitas *browsing* yang berhasil terekam di *database* dan memenuhi persyaratan yang diajukan?
4. Apakah pencatatan/pengorganisasian Internet Explorer Favorites untuk studi kasus akses *web site* terfavorit pengguna berfungsi secara otomatis dan menampilkan data favorit terbaru (*up-to-date*)?

Karena empat pertanyaan inilah disusunlah empat buah tahapan uji coba berdasarkan proses-proses yang berlangsung pada aplikasi Web Data Analyzer. Masing-masing tahapan akan berusaha mencari jawaban dan menilai seberapa jauh fungsionalitas setiap tahapan telah berjalan.

**Tabel 2. Hasil Uji Coba Instalasi Web Data Analyzer**

Workstation	Construct	Integrate	Destruct
10.126.14.40 (IBS00)	√	√	√
10.126.14.47 (IBS07)	√	√	√
10.126.14.50 (IBS10)	√	√	√
10.126.14.51 (IBS11)	√	√	√
10.126.14.52 (IBS12)	√	√	√



**Gambar 12. Menu Web Data Analyzer terintegrasi dengan IE**

Hasil uji coba tahapan pertama menunjukkan bahwa proses instalasi aplikasi Web Data Analyzer pada *workstation-workstation client* 100% berhasil dilakukan. Terlihat dari tabel bahwa proses penciptaan (*construct*), pengintegrasian (*integrate*) dan pemusnahan (*destruct*) aplikasi Web Data Analyzer pada *browser* Internet Explorer berjalan dengan benar tanpa adanya pesan kesalahan baik dari *browser* IE sendiri maupun dari sistem operasi

Windows. Hal ini berarti bahwa *band object* sebagai komponen dasar pembentuk aplikasi Web Data Analyzer berhasil diimplementasikan dan terintegrasi sebagai satu kesatuan dengan *browser* Internet Explorer.

**Tabel 3. Hasil Uji Coba Proses Monitoring**

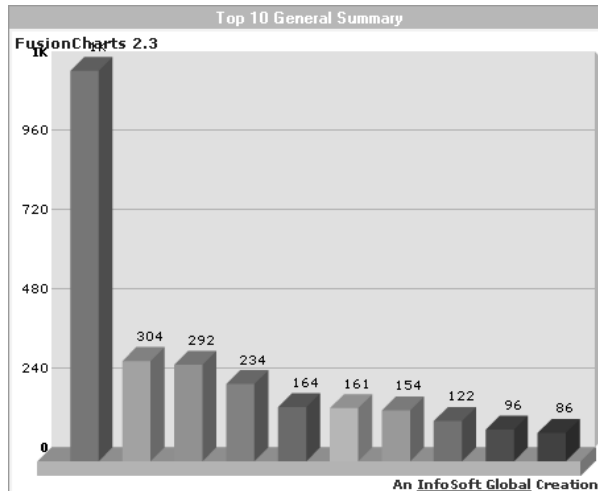
TimeStart	URL
11:17:23:586	http://www.google.com/
11:17:28:273	http://www.google.co.id/
11:18:29:020	http://www.yahoo.com/
11:19:15:017	http://10.126.10.50
11:19:19:123	http://10.126.10.50/monta

Uji coba tahapan kedua memperlihatkan bahwa proses monitoring terhadap aktivitas *browsing* pengguna pada *browser* IE berfungsi. Hal ini dapat dilihat pada tampilannya sejumlah daftar URL pada kontrol *ListView* yang merupakan hasil penangkapan (*intercept*) lalu-lintas *request* URL di jaringan internet. Dapat disimpulkan pula bahwa pengumpulan informasi yang meliputi waktu, metode *request* HTTP, kode-status *response* HTTP dan URL telah berhasil dijalankan. Sedangkan untuk proses *database* bisa terlihat dari kesesuaian data antara informasi yang tampil di kontrol *ListView* pada aplikasi Web Data Analyzer dengan record-record pada tabel Log *database* SQL Server. Hal ini berarti bahwa proses koneksi *database* dan proses penambahan *record* baru ke tabel Log berjalan sebagaimana mestinya.

**Tabel 4. Hasil Uji Coba Laporan Umum**

URL	Persen
http://202.155.84.178	2.129%
http://www.microsoft.com	0.436%
http://10.126.10.50	0.333%
http://localhost	0.316%
http://login.yahoo.com	0.265%
http://akademik.its.ac.id	0.171%

Selain dimunculkan dalam bentuk table, laporan-laporan tersebut juga dimunculkan dalam bentuk grafik seperti terlihat pada gambar 13



Gambar 13. Laporan dalam bentuk grafik

Tabel 5. Hasil Uji Coba Otomatisasi Fungsi Fitur Favorites IE

Internet Shortcut File
Nama File = [0] mail.yahoo.com (12 hits) URL = http://mail.yahoo.com/
Nama File = [1] 202.155.84.178 (10 hits) URL = http://202.155.84.178/newmail/src/login.php
Nama File = [2] 202.155.84.178 (7 hits) URL = http://202.155.84.178/
Nama File = [3] 10.126.10.50 (6 hits) URL = http://10.126.10.50/kelasditawarkan_r.asp
Nama File = [4] www.psbsby-online.net (6 hits) URL = http://www.psbsby-online.net/

Dari hasil uji coba tahapan ketiga menunjukkan bahwa aplikasi Web Data Analyzer telah berhasil mengolah dan menganalisa data-data aktivitas *browsing* pengguna yang telah terekam dalam *database server* menggunakan sintaks-sintaks SQL sederhana serta menampilkan hasilnya sebagai laporan statistik berbentuk tabel dan diagram yang deskriptif dan informatif sebagai bahan referensi bagi pengguna.

Hasil uji coba tahapan keempat menunjukkan bahwa fungsi *exporting* pada aplikasi Web Data Analyzer berhasil menjalankan fungsionalitasnya pada fitur Favorites IE dengan memperbaharui sub-menu pada fitur ini secara otomatis sehingga menampilkan daftar akses *website* terfavorit yang *up-to-date*. Hal ini juga berarti bahwa aplikasi Web Data Analyzer berhasil membuktikan kemampuannya untuk mengimplementasikan studi kasus yang diberikan dalam mencari akses web site terfavorit.

## 9. KESIMPULAN DAN SARAN

### Kesimpulan

Setelah dilakukan uji coba maka dapat diambil kesimpulan bahwa:

- Pada aplikasi ini telah berhasil dibuat suatu perangkat lunak berbentuk komponen band object yang melekat dan terintegrasi dengan *browser* Internet Explorer dengan kategori Explorer Bar Horizontal.
- Aplikasi Web Data Analyzer telah berhasil melakukan fungsi pemantauan (*monitoring*) dan penangkapan (*interception*) terhadap *request-request* URL yang diakses pengguna selama aktivitas *browsing* pada *browser* Internet Explorer berjalan.
- Proses *reporting* telah berhasil menampilkan hasil analisa *query* berdasarkan kriteria-kriteria penyaringan, yang meliputi waktu akses (tanggal, hari dan jam), alamat URL, login pengguna, serta lokasi *workstation* (alamat IP, nama komputer, nama domain) baik untuk data secara keseluruhan maupun berdasarkan login pengguna tertentu.
- Simulasi otomatisasi fitur Favorites IE untuk studi kasus akses web terfavorit berhasil dijalankan dari aplikasi Web Data Analyzer.

### Saran

- Sebagai saran untuk melakukan pengembangan untuk aplikasi Web Data Analyzer ini antara lain:
- Melakukan penambahan data aktivitas *browsing* pengguna yang bisa dimonitor dan direkam oleh Web Data Analyzer, misalnya: data lalu lintas (*traffic*) HTTP/HTTPS dengan memonitor *request* dan *response* HTTP mulai dari *Header*, *Content*, *Cookies*, *Query String*, *Post form Values*, dan sebagainya.
- Proses analisa untuk pengolahan data berdasarkan kriteria yang lebih spesifik dan menggunakan konsep-konsep *data mining* untuk studi kasus yang lebih khusus, misalnya: mengelompokkan URL berdasarkan kategori tema seperti website hiburan, website sosial, website berita, dan sebagainya.

## 10. DAFTAR PUSTAKA

1. Kadir, Abdul; "Panduan Pemrograman Visual C++"; Penerbit ANDI Yogyakarta; 2004
2. Kurniawan, Agus; "Pemrograman Jaringan Internet dengan Visual C++"; Penerbit PT Elex Media Komputindo; 2002
3. MSDN Juli 2001
4. <http://www.google.com>
5. <http://www.codeproject.com>
6. <http://www.codeguru.com/forum>
7. <http://www.functionx.com>



