

## BAB II

### LANDASAN TEORI

Pada landasan teori ini akan dibahas mengenai dasar atau landasan di dalam pembuatan aplikasi ini.

#### 2.1. Registry

Definisi *Registry* adalah Sebuah *database* hierarkis utama yang digunakan pada *Microsoft Windows 9.x*, *Windows CE*, *Windows NT* dan *Windows 2000* untuk menyimpan informasi yang berguna bagi konfigurasi sistem untuk satu atau lebih pengguna, aplikasi, dan alat-alat *hardware*.

Registry berisi informasi yang menjadi referensi bagi *Windows* selama operasinya, seperti profile *user*, aplikasi-aplikasi yang terinstall di komputer dan tipe dokumen yang bisa dibuat aplikasi-aplikasi tersebut, daftar *setting* untuk *folder-folder* dan *icon-icon* aplikasi, hardware apa saja yang ada di sistem, dan *port-port* mana saja yang sedang digunakan.

Registry menggantikan sebagian besar *file-file* berbasis teks yang mengatur konfigurasi *Windows 3.x* dan *MS-DOS*, seperti *Autoexec.bat* dan *Config.sys*. Meskipun sebagian besar *platform* *Windows* memiliki Registry, terdapat perbedaan untuk tiap-tiap versi *Windows*. Data Registry disimpan dalam file-file *biner*, sehingga untuk menyuntingnya memerlukan editor khusus, misalnya *Registry Editor (Regedit)* yang terdapat dalam paket *Windows* dalam bentuk file "*Regedit.exe*" atau "*Regedit32.exe*".

### 2.1.1. Sejarah Registry

Pada awal kemunculan sistem operasi Windows memakai dua jenis file sistem konfigurasi yaitu `config.sys` dan `autoexec.bat`. Kemudian pada perkembangannya ditambahkan dengan dua file untuk menambah kemampuannya yaitu `win.ini` dan `system.ini`. File dengan tipe (*extension*) `.INI` inilah yang mengatur sistem operasi Windows dan aplikasi di dalamnya. `System.ini` digunakan untuk mengendalikan perangkat keras, sedangkan `win.ini` digunakan untuk mengendalikan *software*, *desktop* dan aplikasinya. Seluruh perubahan pada *driver*, *font*, *setting*, dan preferensi akan disimpan ke dalam file `.INI`. Semuanya berjalan dengan lancar saat user hanya memakai beberapa aplikasi saja.

Seiring dengan penambahan dan makin kompleksnya aplikasi yang dipakai, maka akan bertambah pula entri pada file `.INI`. jika user menginstall suatu aplikasi baru, maka akan bertambah masukan pada file `.INI`. Namun, pada saat user menghapus aplikasi yang telah terinstall, maka masukan pada file `.INI` tersebut tidak pernah dihapus yang berakibat file `system.ini` dan `win.ini` akan semakin membesar. Dengan bertambah besarnya ukuran dapat berarti akan memperlambat kinerja sistem. Jika suatu aplikasi diupdate, aplikasi tersebut hanya sekedar menambahkan data baru, tidak menghapus masukan (*entry*) yang sudah ada. Karena aplikasi yang digunakan untuk mengupdate akan menganggap bahwa data yang ada masih terpakai atau dibutuhkan oleh aplikasi yang lain.

Oleh karena ukuran maksimal file `.INI` adalah *64KB*, maka akan timbul masalah jika file `.INI` semakin bertambah besar ukurannya. Jika ukuran sebuah file `.INI` melebihi ukuran *64KB* maka akan timbul masalah pada sistem. Para pengembang software pun berusaha mencari pemecahan untuk masalah ukuran

ini, yaitu dengan membuat file .INI milik mereka sendiri. Dimana para pengembang software hanya akan menyisipkan suatu *pointer* ke dalam file win.ini dan system.ini yang merujuk pada file .INI miliknya. Setiap aplikasi yang diinstall akan menambahkan suatu pointer ke file .INI. Pointer inilah yang kemudian direferensikan ke dalam aplikasi. File .INI tambahan yang berfungsi untuk pengendalian program aplikasi ditambahkan oleh pengembang software yang membutuhkan tambahan kontrol untuk aplikasinya dikarenakan keterbatasan file win.ini dan system.ini. Sebagai contoh misalnya program pengolah kata *Microsoft Word* memiliki file *WORD.INI*, yang digunakan untuk menyimpan pilihan, setting, *defaults*, dan beberapa informasi lainnya sehingga *Word* dapat berjalan dengan baik. Pointer yang dibutuhkan oleh WIN.INI adalah WORD.INI berikut *path* dan nama *filenya*. Kelemahan dari sistem penyisipan pointer ini adalah tidak terkendalinya file-file .INI yang dibuat tersebut sehingga akan semakin sulit untuk ditangani dan didefinisikan dengan cepat.

Seiring dengan perkembangan teknologi, maka lahirnya suatu konsep untuk mengatasi permasalahan mengenai file .INI yaitu dengan dibentuknya sebuah database yang menangani permasalahan tersebut. Inilah titik awal dari registry. Registry dibuat secara eksklusif dengan aplikasi *32-bit*, dan ukuran file yang membentuknya dapat lebih dari *40 MegaByte*. Ukuran yang sangat besar bila dibandingkan dengan ukuran maksimum file .INI yang hanya sebesar *64KB*.

Registry secara prinsip akan menyimpan dua jenis informasi yang mendasar, yaitu:

1. *Global Information*.

Informasi ini sangat penting untuk operasi dasar hardware dan software dasar pada komputer.

## 2. *User Specific Information.*

Bagian ini berisi tentang konfigurasi user, seperti layout dari *desktop* Windows, dan pilihan-pilihan yang telah diatur oleh user untuk Windows maupun pada aplikasi Windows.

### 2.1.2. Struktur Registry

Registry ialah sebuah database kompleks yang dibuat dengan gaya hierarki oleh perancangannya, yang digunakan Windows untuk mengontrol hardware, software, user environment, dan *interface*. Secara umum Registry mengandung informasi mengenai hardware yang terpasang pada sistem, software yang ada pada saat Windows dijalankan, serta pembagian *memory*. Registry disusun dari komponen-komponen kecil sehingga membentuk suatu kumpulan komponen dasar. Registry merupakan misteri, karena Microsoft tidak pernah menjelaskan secara terperinci isi yang ada di dalam registry serta cara mengeditnya.

Registry secara global terdiri dari 6 (enam) kategori informasi, bagian ini sering disebut dengan nama kunci induk (*root keys*) atau juga biasa disebut sebagai *Handle Keys*.

Pada setiap versi Windows jumlah Handle Keys tersebut sering kali berbeda, maupun berbeda dalam hal penamaan. Handle Keys ini seperti sebuah kontainer data, setiap informasi terhubung dengan aspek tertentu pada sistem komputer. Handle Keys atau yang biasa disebut dengan singkatan *HKEY* dapat

berisi data nilai atau *value*, dapat juga berisi kunci atau *keys* itu sendiri ataupun *subkey*. Dengan cara ini maka data Registry dapat dikelola secara terstruktur.

*Database* Registry dapat diakses dan diedit dengan menggunakan software yang telah terdapat di Windows, yaitu Regedit. Untuk menjalankannya, klik [Start]>[Run]. Setelah itu, ketik “regedit” (tanpa tanda kutip) diikuti penekanan tombol [Enter].

Pada umumnya jika melalui Registry Editor (Regedit) maka pada *window* Regedit akan tampil 6 bagian utama dari Registry, yang dapat disebut pula sebagai

*Handle Keys*, yaitu:

#### **A. HKEY\_CLASSES\_ROOT**

Pada bagian ini berisi apa saja yang dibutuhkan oleh sebuah aplikasi untuk dapat berjalan pada Windows. Yang dimaksud antara lain adalah tipe (*extension*) *file* yang digunakan serta asosiasinya dengan aplikasi lain, nama *driver*, ikon yang digunakan, informasi *DDE* dan *OLE*, nomor *Class ID* (nomor yang digunakan untuk mengakses *item*), dan *string* yang digunakan sebagai *pointer* ke arah teks asli.

#### **B. HKEY\_LOCAL\_MACHINE**

Bagian ini digunakan untuk menyimpan *setting* untuk membuat *software*, *hardware*, serta Windows bekerja pada komputer. Keamanan, hak *user*, dan *sharing* data juga tersimpan pada *handle keys* ini.

#### **C. HKEY\_CURRENT\_CONFIG**

HKEY\_CURRENT\_CONFIG merupakan bagian yang berisi konfigurasi *hardware* yang ada pada HKEY\_LOCAL\_MACHINE. Jika pengguna PC

menggunakan hanya satu konfigurasi, yaitu konfigurasi *default*, maka data yang terdapat pada HKEY\_CURRENT\_CONFIG dan yang terdapat pada HKEY\_LOCAL\_MACHINE akan sama. Dengan menciptakan konfigurasi tambahan, maka informasi konfigurasi tersebut akan tersimpan di HKEY\_LOCAL\_MACHINE.

#### **D. HKEY\_USER**

Bagian ini mengandung informasi mengenai *setting* yang telah diatur sedemikian rupa oleh *default user* dan *logged-in user*. *Setting* ini akan menginformasikan icon, *start menu*, warna dan *font*, serta pengaturan *Control Panel* yang digunakan oleh *user*.

#### **E. HKEY\_CURRENT\_USER**

Tidak seperti HKEY\_USER yang menyimpan informasi mengenai seluruh *user*, HKEY\_CURRENT\_USER hanya menyimpan informasi mengenai *user* yang sekarang sedang *logged-in*.

#### **F. HKEY\_DYN\_DATA**

Berbeda dengan *handle keys* yang lain karena *handle keys* ini tidak tersimpan di dalam *harddisk* namun tersimpan pada *RAM* dan digunakan untuk mengontrol *hardware*. HKEY\_DYN\_DATA menangani informasi tentang *plug and play*. Karena HKEY\_DYN\_DATA tersimpan di *RAM*, maka informasi yang terdapat didalamnya akan hilang dan berganti apabila *Windows restart*.

### 2.1.3. File Fisik Registry

Registry terbentuk dari beberapa file fisik yang tersimpan pada sistem, file-file tersebut berbeda-beda bergantung pada versi dari Windows. Pada Windows 95 dan Windows 98 registry terbentuk atas 2 buah file yang tersembunyi pada *directory* Windows yaitu **USER.DAT** dan **SYSTEM.DAT**, sedangkan pada Windows ME terdapat sebuah file tambahan yaitu **CLASSES.DAT**. Sementara pada Windows NT, Windows 2000, dan Windows XP file-file pembentuk registry tersimpan secara terpisah pada beberapa folder, yaitu pada folder **%Systemroot%\System32\Config** dan pada folder **%Systemroot%\Profiles\UserName**.

Pada folder **%Systemroot%\System32\Config** terdapat file-file yang membentuk beberapa handle keys pada registry, file-file tersebut ialah:

1. *Sam.log*, *Sam.sav* membentuk HKEY\_LOCAL\_MACHINE\SAM
2. *Security.log*, *Security.sav* membentuk HKEY\_LOCAL\_MACHINE\Security
3. *Software.log*, *Software.sav* membentuk HKEY\_LOCAL\_MACHINE\Software
4. *System.alt*, *System.log*, *System.sav* membentuk HKEY\_LOCAL\_MACHINE\System
5. *System.alt*, *System.log*, *System.sav*, *Ntuser.dat*, *Ntuser.dat.log* membentuk HKEY\_CURRENT\_CONFIG
6. *Default.log*, *Default.sav* membentuk HKEY\_USERS\DEFAULT

Sedangkan file yang terdapat pada folder *%Systemroot%\Profiles\UserName* membentuk handle keys HKEY\_CURRENT\_USER.

#### 2.1.4. Nilai (Value) pada Registry

Fungsi utama dari registry adalah menyimpan data. Pada saat data diorganisasi di bawah kunci-kunci registry, data itu sendiri berisi nilai. Nilai inilah yang diakses oleh Windows dan program aplikasi. Nilai-nilai registry ini sudah terbentuk pada saat instalasi Windows. Selain itu penambahan nilai juga akan terjadi jika terdapat penambahan hardware atau menginstall aplikasi baru. Nilai-nilai registry ini berbeda antara komputer satu dengan komputer lainnya bergantung pada peralatan (*device*) yang terpasang pada komputer dan konfigurasi sistem. Setiap nilai yang ada pada registry terdiri dari tiga bagian, yaitu:

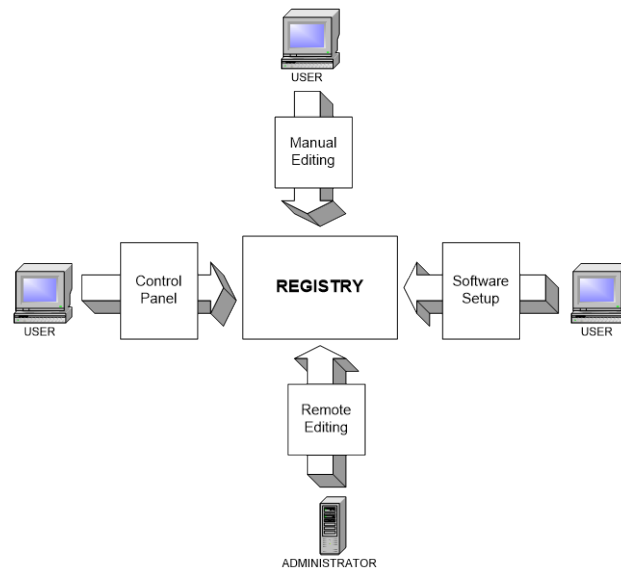
- A. Nama Nilai (*Values Name*). Sesuai dengan sebutannya, bagian ini digunakan sebagai identifikasi atau nama dari nilai.
- B. Tipe Data (*Data Type*). Bagian ini menerangkan tipe data dari nilai tersebut. Adapun tipe data pada registry ialah:
  1. ***String Value*** atau ***REG\_SZ*** merupakan tipe data berbentuk teks sederhana, dapat berisi angka dan juga huruf.
  2. ***Binary Value*** atau ***REG\_BINARY*** merupakan tipe data berbentuk binary (0 dan 1). Tipe data ini banyak digunakan untuk menyimpan nilai mengenai informasi hardware.
  3. ***DWORD Value*** atau ***REG\_DWORD*** merupakan tipe data numerik 32-bit dan dapat berbentuk integer atau heksadesimal.



4. **Multi-String Value** atau **REG\_MULTI\_SZ** merupakan tipe data berbentuk teks, tipe ini dapat berisi lebih panjang dari pada tipe String (REG\_SZ).
5. **Expandable String Value** atau **REG\_EXPAND\_SZ** merupakan tipe data yang hampir sama dengan tipe data String (REG\_SZ), perbedaannya terletak pada penggunaan variabel. Variable yang tertulis pada REG\_EXPAND\_SZ akan digunakan oleh sistem, contoh variable *%SYSTEMROOT%* yang jika diaktifkan oleh sistem akan merujuk pada direktori dimana Windows disimpan.

C. **Data (Data Values)**. Bagian ini digunakan untuk menyimpan informasi atau data yang terdapat dalam nilai, bentuk informasi atau data ini sesuai dengan tipe data masing-masing nilai.

Pembuatan nilai, penghapusan nilai, dan perubahan nilai pada registry selain dapat dilakukan secara manual melalui registry editor (regedit) juga dapat dilakukan dengan tiga cara lainnya, yaitu melalui control panel atau *utility internal* Windows, instalasi software dan *driver* untuk hardware, dan *system policies* dari server oleh administrator melalui *Remote Registry Editing*.

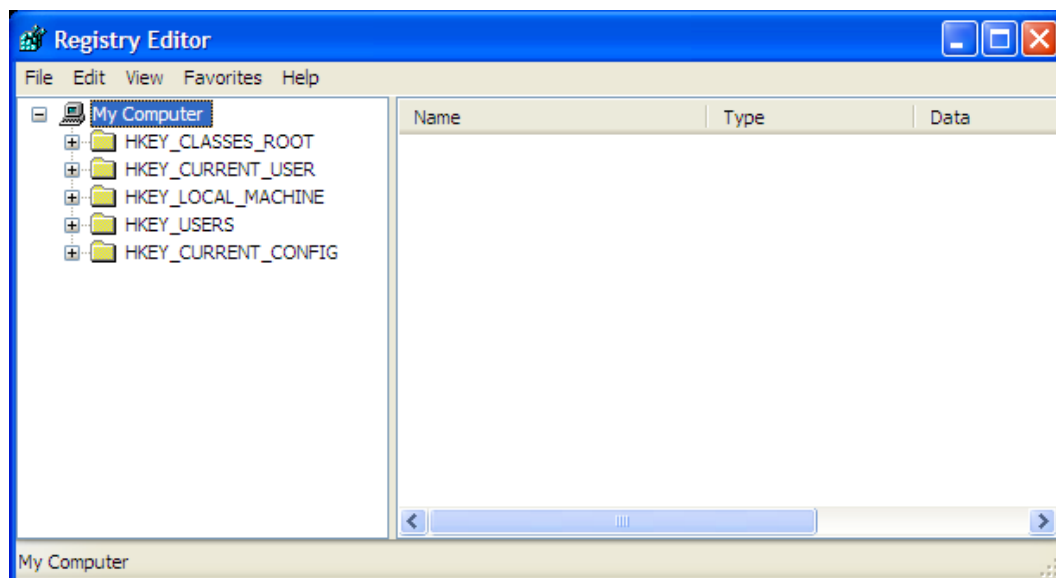


Gambar 2.1. Cara-cara Perubahan pada Registry

## 2.2. Registry Editor (Regedit)

Registry Editor (Regedit) adalah sebuah tool yang digunakan untuk menampilkan dan mengubah setting atau nilai pada registry, yang berisi informasi-informasi yang diperlukan sistem untuk beroperasi. Windows menyimpan informasi konfigurasi di dalam sebuah database (Registry) yang diorganisasikan menyerupai format pohon pada Windows Explorer.

Cara untuk menjalankan Regedit, yaitu klik [Start]>[Run]. Setelah itu, ketik “regedit” (tanpa tanda kutip) diikuti penekanan tombol [Enter]. Setelah itu maka akan tampak tampilan dari regedit, seperti gambar berikut:



Gambar 2.2. Tampilan Registry Editor pada Windows XP

Pada registry editor, sebuah folder mewakili sebuah *keys* dan ditunjukkan didalam area navigasi pada bagian kiri registry editor. Pada topik area disebelah kanan, ditampilkan nilai-nilai yang terdapat di dalam sebuah *keys*.

### 2.3. Windows Application Program Interface (Windows API)

*Windows API* merupakan sekumpulan fungsi-fungsi *eksternal* yang terdapat dalam file-file perpustakaan Windows (*Windows Library*) atau *file library* lainnya yang dapat digunakan oleh program yang berbasis Windows. Fungsi ini dapat menangani semua yang berhubungan dengan Windows, seperti pengaksesan disk, interface printer, grafik Windows, kotak dialog, *Windows shell*, setting sistem operasi, penanganan file, mengakses sistem registry, memainkan musik, dan sebagainya. Fungsi ini menyediakan banyak fitur-fitur standar untuk semua program yang berbasis Windows.

Semua fungsi Windows API hampir terdapat dalam direktori sistem milik Windows, dan paling banyak berekstensi *.DLL* yang digunakan oleh sistem

operasi Windows. Selain itu fungsi ini juga memastikan secara konsisten penggunaan semua sumber yang terdapat dalam Windows. File-file itulah yang disebut dengan Windows API.

Fungsi Windows API yang terdapat di dalam file-file library Windows, biasanya tiap library terdiri atas sejumlah fungsi untuk menangani tugas tertentu. Contoh penggunaan *library GDI32.DLL* untuk mengatur fungsi-fungsi grafik, dan *USER32.DLL* untuk pengaturan *User Interface* Windows.

Karena fungsi Windows API merupakan fungsi *eksternal*, maka untuk menggunakan fungsi tersebut programmer harus terlebih dahulu mendeklarasikan fungsi Windows API tersebut ke dalam bahasa pemrogramannya. Sintaks untuk mendeklarasikan fungsi atau sub procedure adalah:

a. Deklarasi untuk Fungsi

**Declare Function** NamaFungsi **Lib** "NamaLibrary"  
 [Alias "AliasFungsi"] [( [[ByVal/ByRef] variabel [As type]  
 [, [ByVal/ByRef] variabel [As type]]...)] **As Type**

b. Deklarasi untuk Sub Procedure

**Declare Sub** NamaFungsi **Lib** "NamaLibrary"  
 [Alias "AliasFungsi"] [( [[ByVal/ByRef] variabel [As type]  
 [, [ByVal/ByRef] variabel [As type]]...)]

keterangan:

1. **Declare Function/Sub**: Sintaks deklarasi fungsi atau prosedur.
2. **NamaFungsi**: Nama fungsi yang ada di dalam library.
3. **Lib**: Sintaks deklarasi library.
4. **NamaLibrary**: Nama library yang terdapat nama fungsi.

5. **Alias**: Sintaks alias.
6. **AliasFungsi**: Penamaan fungsi, untuk membedakan dengan yang sama.
7. **ByVal/ByRef**: Tipe dari parameter fungsi.
8. **Variable**: Nama variabel yang digunakan fungsi.
9. **As type**: Tipe dari variabel.
10. **As Type**: Nilai yang dikembalikan oleh fungsi.

Contoh dari pendeklarasian fungsi Windows API, sebagai berikut:

```
Public Declare Function SetWindowText Lib "user32" Alias
"SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String) As Long.
```

Setelah fungsi tersebut dideklarasikan, maka fungsi tersebut dapat digunakan di dalam program Visual Basic. Berikut ini contoh pemanggilan fungsi Windows API yang bernama *SetWindowText* dalam *form* pada bagian *event load* Visual Basic.

```
Private Sub Form1_Load()
Dim x as long
x = SetWindowText(Form1.hWnd, "Test Fungsi API")
End Sub
```

Contoh fungsi diatas berfungsi untuk merubah judul *window (caption)* pada sebuah *form* Visual Basic menjadi "Test Fungsi API".

Untuk mencari fungsi Windows API, tipe data, dan konstanta, digunakan aplikasi API Viewer yang telah disertakan di dalam Visual Basic.

### 2.3.1. API Viewer

*Visual Basic* secara *default* menyertakan program pembantu yang disebut *API Viewer*. Cara kerja program ini cukup sederhana yaitu dengan

menyalin dan memindahkan nama fungsi, konstanta, dan tipe data ke dalam kode program Visual Basic.

Aplikasi API Viewer menyediakan hampir seluruh deklarasi fungsi, konstanta, dan tipe data yang terdapat dalam file teks (.TXT) maupun dalam file database (.MDB). Setelah programmer menemukan fungsi yang diinginkan, programmer dapat mengkopinya ke dalam *clipboard* yang nantinya dapat digabungkan dengan kode program Visual Basic.

#### 2.4. Visual Basic

Rancangan aplikasi ini akan dibuat dengan menggunakan salah satu editor bahasa pemrograman yang banyak digunakan oleh para programmer yaitu Visual Basic 6.0

Visual Basic merupakan *Development Software (Microsoft Visual Studio)* yang mendukung *OLE, COM, dan ActiveX (OCX)*. Terdapat di dalamnya fitur yang dapat digunakan untuk membuat aplikasi yang dapat mengakses sistem operasi Windows. Dengan kemudahan dalam pembuatan program yang disertai banyaknya file *ocx* yang beredar di internet baik secara *freeware* maupun *shareware* membuat Visual Basic sebagai bahasa pemrograman yang tepat untuk membuat suatu aplikasi secara cepat.

Hal ini dikarenakan Visual Basic memiliki kemampuan untuk menggunakan Windows API agar dapat berhubungan langsung dengan sistem operasi Windows sehingga akan sangat membantu dalam pemrograman untuk berinteraksi dengan registry Windows.

## 2.5. Microsoft HTML Help Workshop

*Microsoft HTML Help* merupakan generasi penerus untuk perancangan dan pembuatan sistem *help online* yang dibuat oleh Microsoft. HTML Help menggunakan komponen yang telah tersedia di dalam *Microsoft Internet Explorer* untuk menampilkan isi dari file help. HTML Help juga mendukung *Hyper Text Markup Language (HTML)*, *ActiveX*, *Java*, *Javascript*, *Microsoft Visual Basic Script*, dan gambar HTML yang berformat *.JPEG*, *.GIF*, *.PNG*.

HTML Help ditujukan bagi programmer atau pengembang untuk membuat online help untuk software program, *multimedia title*, *intranets*, *extranets*, atau untuk *internet*. Komponen-komponen yang terdapat pada Microsoft HTML Help Workshop ini adalah :

### a. HTML Help Workshop

merupakan help authoring tool (hhw.exe) dengan *Graphical Interface* yang cukup mudah digunakan dalam pembuatan file help, topik, isi, indeks, dan segala sesuatunya yang dibutuhkan dalam pembuatan *online help system*.

### b. HTML Help Image Editor

merupakan *online graphic tool* untuk membuat *screenshot*, yaitu: *converting*, *editing*, *viewing image file*, dan *browsing thumbnail image*.

## 2.6. Windows XP

Windows XP diluncurkan pertama kali di Amerika Serikat pada tanggal 25 Oktober 2001 dan di Indonesia pada tanggal 1 November 2001. Windows XP merupakan kelanjutan dari Microsoft Windows 2000 dan Microsoft Windows Millenium Edition (Win ME), dan merupakan sistem operasi pertama yang secara

khusus dirancang oleh Microsoft untuk memenuhi kebutuhan pengguna bisnis dan pribadi.

Windows XP dihadirkan dalam 3 edisi, yaitu:

- a. Windows XP Professional yang dirancang untuk berbagai ukuran bisnis, dan dalam banyak hal juga cocok untuk pengguna di rumah (*home user*).
- b. Windows XP Home Edition dirancang untuk komputer yang tidak bekerja dalam sebuah domain jaringan, meskipun Windows XP versi Home Edition dapat dihubungkan ke dalam sebuah jaringan namun bukan sebagai pengontrol domain. Windows XP Home Edition dirancang sebagai upgrade dari Microsoft Windows 98 atau Windows ME. Windows XP Home Edition mencakup sejumlah komponen baru yang hebat yang mempermudah *user* bekerja dan bermain dengan multimedia dan kemampuan internet.
- c. Windows XP 64-bit Edition. Windows XP versi ini hanya bisa bekerja pada tipe komputer tertentu yang telah mendukung komputasi 64-bit. Komputer-komputer 64-bit ini dirancang untuk pekerjaan khusus yang membutuhkan memori *bandwidth* yang lebar, seperti komputer server dan komputer design grafis.