

# Membuat Common Gateway Interface (CGI) dengan Pascal

**Youppie Arliansyah**  
y\_arl@yahoo.com

## ***Lisensi Dokumen:***

*Copyright © 2004 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Ada yang mengatakan bahwa para programmer Pascal mungkin tidak pernah bermimpi dapat membuat aplikasi berbasis web. Memang untuk saat ini tidak ada tool/library khusus yang dapat digunakan oleh programmer Pascal untuk membuat aplikasi berbasis web, kecuali mungkin framework .NET-nya Microsoft. Tapi dengan sedikit mengutak-atik, para programmer Pascal sebetulnya bisa ambil bagian dalam dunia yang satu ini.

Artikel ini saya tulis untuk berbagi pengalaman membuat program CGI dengan Pascal. Awalnya saya penasaran dengan pernyataan bahwa CGI dapat dibuat dengan sembarang bahasa pemrograman<sup>1</sup>. Jika demikian, tentunya CGI juga dapat dibuat dengan menggunakan Pascal. Maka iseng-iseng saya coba kutak-katik webserver Apache di komputer saya, dan jadilah tulisan ini.

Spesifikasi sistem yang saya gunakan adalah Pascal untuk platform DOS dan webserver Apache 1.3.22. Sebagai catatan, saya belum mencoba cara ini pada web server selain Apache, dan tidak berhasil pada web server IIS.

## **Sekilas tentang CGI**

CGI (Common Gateway Interface) adalah protokol yang memungkinkan web server untuk berkomunikasi dengan sistem operasi. Web server pada dasarnya tidak dapat melakukan proses pengolahan data terhadap input yang diterimanya. Output content dinamik sebuah situs web ditangani oleh program-program lain (content generator) yang dipanggil oleh web server. Pemrograman dalam lingkungan web ini biasa disebut pemrograman CGI.

Program CGI, terutama pada platform \*NIX, umumnya dibuat dengan bahasa PERL atau C. Web server berkomunikasi dengan program-program tersebut melalui standard input, standard output dan environment variable. Artinya, bahasa program yang sanggup membaca dari standard input, menulis ke standard output dan dapat mengakses *environment variable* bisa digunakan untuk membuat aplikasi CGI. Turbo Pascal termasuk bahasa yang memiliki kemampuan tersebut.

Pada platform DOS, setiap file program harus berekstensi \*.EXE. Berbeda dengan lingkungan UNIX yang mengenali sebuah file program melalui atributnya, sehingga memberikan fleksibilitas dalam pemberian ekstensi file. Biasanya program-program CGI pada lingkungan UNIX diberi

---

<sup>1</sup>Kim, Eugene Erick, "CGI Developer's Guide", Sams.net Publishing, Indianapolis, 1996

ekstensi \*.cgi untuk program yang dibuat dengan C atau \*.pl untuk program yang dibuat dengan PERL. Pada lingkungan DOS, karena wajib berekstensi exe, mau-tak mau program CGI yang dibuat pada platform ini juga harus berekstensi exe.

## Konfigurasi Web Server

Agar dapat berkomunikasi dengan program lain, web server Apache harus dikonfigurasi agar dapat menjalankan CGI. Silahkan lihat manual/dokumentasi Apache untuk mendapatkan informasi cara mengkonfigurasi yang lengkap. Pada artikel ini saya menggunakan salah satu cara yang dijelaskan oleh manual Apache.

Konfigurasi Apache berada dalam file httpd.conf pada direktory conf. Berikut adalah konfigurasi yang saya tambahkan agar CGI yang dibuat dengan Turbo Pascal dapat dijalankan.

```
#
# This should be changed to whatever you set
# DocumentRoot to.
#
<Directory "C:/wwwroot">
# DST, DST, DST, BAGIAN KONFIGURASI LAINNYA
# tambahkan option ExecCGI agar web server dapat
# mengeksekusi file CGI dalam direktory root web
Options Indexes FollowSymLinks ExecCGI MultiViews
</Directory>
#
# Document types.
#
<IfModule mod_mime.c>
#####
##
## Bagian ini berisi konfigurasi lainnya
##
#####
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
AddType application/x-tar .tgz
##### Bagian konfigurasi men-set MIME Header exe.
##### Mungkin tidak terlalu perlu
AddType application/x-exe .exe
##### End of bagian
##### INI BAGIAN UNTUK NGESET CGI-SCRIPT DENGAN EXE.
##### EXTENSI EXE AKAN DI PERLAKUKAN SEBAGAI CGI EUY!
AddHandler cgi-script .exe
##### End of bagian
#####
##
## Bagian ini berisi konfigurasi lainnya
##
#####
</IfModule>
```

*Listing 1. Beberapa Konfigurasi Httpd.Conf*

Dari konfigurasi diatas bisa dilihat bahwa saya menggunakan cara AddHandler yang memberitahukan Apache bahwa file dengan ekstensi EXE dapat dijalankan sebagai CGI. Cara ini tidak terlalu bagus karena ada kemungkinan file lain yang bukan CGI dapat dieksekusi oleh web server. Maka berhati-hatilah....

Jika konfigurasi dalam httpd.conf telah dirubah, jalankan Apache atau restart jika telah berjalan. Seharusnya Apache telah mengenali file EXE sebagai program CGI. Untuk mencobanya, ketikkan

program berikut, kompilasi, lalu panggil dari browser.

```
(*****)  
(* Program: TEST1.PAS *)  
(* Author : Youppie Arliansyah *)  
(* Date : October, 6, 2003 *)  
(* Description *)  
(* Program sederhana untuk contoh *)  
(* menampilkan halaman HTML. *)  
(* *)  
(* Hak cipta oleh Youppie Arlianysah. *)  
(* Program ini bebas untuk digunakan dalam *)  
(* lingkungan pendidikan. *)  
(* Apabila digunakan untuk kepentingan bisnis, *)  
(* mohon pencipta program diberitahu untuk *)  
(* mencatat track record penggunaan. *)  
(* *)  
(*****)  
program TEST1;  
begin  
  writeln('Content-type: text/html');  
  writeln;  
  writeln('<h1>Test CGI dengan Pascal</h1>');  
end.
```

*Listing 2. Program CGI test1.pas*



*Gambar 1. Tampilan CGI test1.pas*

Wholaa..., para programmer Turbo Pascal yang tidak pernah bermimpi sekarang sudah harus memulai mimpinya.

## **Tentang Pemrograman CGI**

Pemrograman CGI sedikit berbeda dengan pemrograman lainnya. Terutama dalam masalah output program, yang terkait dengan bagaimana web bekerja.

Output program CGI harus terdiri dari dua bagian: header dan isi. Header adalah bagian yang memberitahu web server tipe data yang akan diterimanya dan bagian isi adalah apa yang biasanya kita lihat dalam source file HTML. Harap bedakan antara header HTML dengan header CGI. Yang dimaksud dengan bagian header dari CGI adalah header HTTP yang umumnya berisi protokol

komunikasi, sementara header yang terdapat dalam tag HTML adalah header HTML yang mengatur isi dokumen HTML yang bersangkutan. Header HTTP dan isinya diakhiri dengan satu baris kosong. Bagian isi yang akan tampil di browser adalah apa yang di-outputkan oleh statement `writeln ke-3`, yang biasanya berupa HTML. Contoh diatas adalah dengan mengirimkan teks dalam tag heading 1.

## Standar Input, Standar Output Dan Pengolahan Form

Program CGI yang kita buat pada dasarnya harus bisa mengolah data yang diinputkan oleh user melalui browser. Input yang diterima dari browser terdiri atas dua jenis:

- Informasi tentang browser, server, dan CGI program itu sendiri. Informasi ini ditempatkan oleh web server pada environment variable.
- Informasi yang diisi oleh user. Pengambilan informasi ini tergantung pada method yang digunakan untuk mengirimkan informasi tersebut. Jika informasi dikirim dengan method GET, maka dia tersedia pada environment variable `QUERY_STRING`, sementara jika dikirimkan dengan method POST maka informasi tadi akan tersedia pada `STANDARD INPUT`.

```
(*****  
(* Program: TESTENV.PAS *)  
(* Author : Youppie Arliansyah *)  
(* Date : October, 6, 2003 *)  
(* Description *)  
(* Program contoh untuk menampilkan *)  
(* environment variable. *)  
(* *)  
(* Hak cipta oleh Youppie Arlianysah. *)  
(* Program ini bebas untuk digunakan dalam *)  
(* lingkungan pendidikan. *)  
(* Apabila digunakan untuk kepentingan bisnis, *)  
(* mohon pencipta program diberitahu untuk *)  
(* mencatat track record penggunaan. *)  
(* *)  
(***)  
program TESTENV;  
uses Dos;  
var I: Integer;  
begin  
  writeln('Content-type: text/html');  
  writeln;  
  for I := 1 to EnvCount do  
    Writeln(EnvStr(I), '<br>');  
end.
```

*Listing 3. Program untuk menampilkan environment variable*

Sebelum dikirimkan ke web server, data yang berada dalam tag `<input>` dikodekan (encode) oleh browser ke dalam format URL encoding. Konsep umum dari URL encoding kurang lebih sebagai berikut:

- Semua field dipisahkan dengan ampersand (&)
- Nama dan nilai dipisahkan dengan tanda sama dengan (=), dengan nama di sebelah kiri dan nilai disebelah kanan.
- Spasi diganti dengan tanda tambah (+)
- Semua karakter yang aneh-aneh diganti dengan tanda persen(%) diikuti oleh dua digit kode karakter hexadesimal ascii.

Informasi yang diterima webserver berupa string yang telah di-encode. Sehingga untuk mendapatkan

nilai dari data/variable, program CGI yang kita buat harus dapat men-decode string tersebut dan mengambil masing-masing keping informasi yang ada.

```
<h1>Form dengan method GET</h1><br>
<form action="testhalo.exe" method="get">
  <table>
    <tr>
      <td>nama</td><td>:</td>
      <td><input name="nama"></td>
    </tr>
    <tr>
      <td>alamat</td><td>:</td>
      <td><input name="alamat"></td>
    </tr>
    <tr>
      <td>umur</td><td>:</td>
      <td><input name="umur"></td></tr>
    </table>
    <input type="submit" value="kirim">
  </form>
```

*Listing 4. Contoh HTML sederhana*

Jika nama diisi dengan Youppie Arliansyah; alamat dengan Padang dan umur dengan 29, isi form tersebut akan di-encode menjadi string:

nama=Youppie+Arliansyah&alamat=Padang&umur=29

Program CGI harus dapat men-decode string tersebut untuk mendapatkan variable-variable:

```
nama:='Youppie Arliansyah';
alamat:='Padang';
umur:=29;
```

Berikut adalah contoh unit pascal yang berisi prosedur dan function untuk mengambil nilai dari browser.

```
(*****)  
(* Program: CGIPAS.PAS *)  
(* Author : Youppie Arliansyah *)  
(* Date : October, 6, 2003 *)  
(* Description *)  
(* Unit CGIPAS.TPU untuk menyimpan pustaka *)  
(* fungsi/prosedur yang berguna dalam *)  
(* pemrograman CGI dengan Pascal *)  
(* *)  
(* Hak cipta oleh Youppie Arliansyah. *)  
(* Program ini bebas untuk digunakan dalam *)  
(* lingkungan pendidikan. *)  
(* Apabila digunakan untuk kepentingan bisnis, *)  
(* mohon pencipta program diberitahu untuk *)  
(* mencatat track record penggunaan. *)  
(* *)  
(***)  
unit cgipas;  
interface  
uses dos;  
procedure hello;  
function cgipas_nilai(inputnya:string;  
namavar:string):string;  
procedure cgipas_ambil_input(var datanya:string);  
procedure headerhtml;  
function metodequery:string;  
implementation  
procedure headerhtml;  
begin  
writeln('Content-type: text/html');  
writeln;  
end;  
procedure hello;  
begin  
headerhtml;  
write('<h1>Hello semua, halaman ini ');  
write('halaman ini dibuat dengan Pascal</h1>');  
end;  
function cgipas_nilai  
(inputnya:string;namavar:string):string;  
var posisi:integer;  
dataambil:string;  
begin  
posisi:=pos(namavar,inputnya);  
{Kalau posisi=0 artinya tidak ketemu }  
if posisi>0 then  
begin  
dataambil:=copy(inputnya,  
posisi,length(inputnya));  
delete(dataambil,pos('&',dataambil),  
length(dataambil));  
  
{cek jika ada tanda tambah, maka ganti }  
{dengan spasi }  
posisi:=pos('+',dataambil);  
if posisi>0 then dataambil[posisi]:=' ';  
cgipas_nilai:=copy(dataambil,  
pos('=' ,dataambil)+1,  
length(dataambil));  
end  
else cgipas_nilai:='';  
end;  
function metodequery:string;  
begin
```

```
        readln(input,datanya);
        {$I+}
        if ioresult<>0 then;
        { proses read diatas pasti menimbulkan error }
        { jadi biarin aja dia nggak ngelakuin apapa }
        end;
end;
begin
end.
```

*Listing 5. Unit CGIPAS.PAS*

Keterangan singkat program:

- procedure hello digunakan untuk menampilkan text biasa.
- function cgipas\_nilai(inputnya:string;namavar:string) akan mengembalikan nilai dari variable namavar. Nilai yang dikembalikan adalah bertipe string, sehingga perlu di-convert dengan procedure val jika yang akan diambil adalah angka.
- procedure cgipas\_ambil\_input(var datanya:string) adalah untuk mengambil data yang dikirimkan oleh browser dan menempatkannya pada variable datanya. Procedure ini akan memeriksa metode query yang digunakan oleh user. Jika menggunakan GET maka procedure akan mengambil data dari Environment Variable QUERY\_STRING, sementara jika menggunakan POST maka input akan diambil dari stdin (statement readln (input,datanya); input adalah keyword Pascal untuk standard input).
- procedure headerhtml; digunakan untuk mengirimkan header http yang memberitahu browser bahwa data yang dikirim adalah berbetuk text/html. Jika prosedur ini tidak sesuai dengan fungsinya (seharusnya bernama headerhttp) maka itu adalah kesalahan yang saya lakukan sejak setahun lalu.
- function metodequery:string; akan mengembalikan metode yang digunakan oleh user/browser dalam mengirimkan datanya.





```
(*****)  
(* Program: Q_POST.PAS *)  
(* Author : Youppie Arliansyah *)  
(* Date : October, 6, 2003 *)  
(* Description *)  
(* Program contoh untuk menampilkan *)  
(* halaman HTML yang berisi form. Form *)  
(* dikirimkan dengan method POST ke *)  
(* program CGI TESTHALO.EXE *)  
(* *)  
(* Hak cipta oleh Youppie Arliansyah. *)  
(* Program ini bebas untuk digunakan dalam *)  
(* lingkungan pendidikan. *)  
(* Apabila digunakan untuk kepentingan bisnis, *)  
(* mohon pencipta program diberitahu untuk *)  
(* mencatat track record penggunaan. *)  
(* *)  
(*****)  
program Q_POST;  
begin  
  writeln('Content-type: text/html');writeln;  
  writeln('<h1>Form dengan method POST</h1><br>');  
  writeln('<form action="testhalo.exe" ');  
  writeln('method="post">');  
  writeln('<table>');  
  writeln('<tr><td>nama</td><td>:</td>');  
  writeln('<td><input name="nama"></td></tr>');  
  writeln('<tr><td>alamat</td><td>:</td>');  
  writeln('<td><input name="alamat"></td></tr>');  
  writeln('<tr><td>umur</td><td>:</td>');  
  writeln('<td><input name="umur"></td></tr>');  
  writeln('</table>');  
  writeln('<input type="submit" value="kirim">');  
  writeln('</form>');  
end.
```

Listing 7. Q\_POST.PAS, menampilkan form dengan method POST



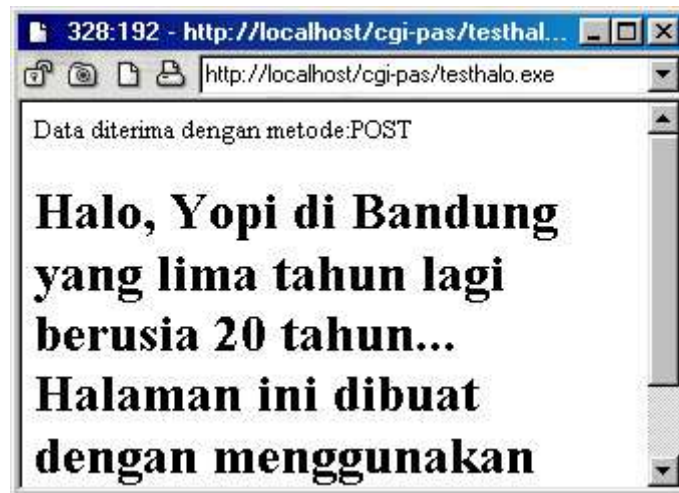
Gambar 3. Tampilan Q\_POST.PAS

```
(*****)  
(* Program: TESTHALO.PAS *)  
(* Author : Youppie Arliansyah *)  
(* Date : October, 6, 2003 *)  
(* Description *)  
(* Program contoh untuk menerima dan *)  
(* mengolah data dari HTML. Menggunakan *)  
(* unit CGIPAS.TPU untuk mengambil data *)  
(* dari browser *)  
(* *)  
(* Hak cipta oleh Youppie Arliansyah. *)  
(* Program ini bebas untuk digunakan dalam *)  
(* lingkungan pendidikan. *)  
(* Apabila digunakan untuk kepentingan bisnis, *)  
(* mohon pencipta program diberitahu untuk *)  
(* mencatat track record penggunaan. *)  
(* *)  
(*****)  
program TESTHALO;  
uses cgipas;  
var i:string;  
u,code_u:integer;  
begin  
cgipas_ambil_input(i);  
headerhtml;  
writeln('Data diterima dengan metode:',  
metodequery,'<br>');  
writeln('<h1>Halo, ',cgipas_nilai(i,'nama'),  
' di ',cgipas_nilai(i,'alamat'),  
' yang lima tahun lagi berusia ');  
val(cgipas_nilai(i,'umur'),u,code_u);  
writeln(u+5,' tahun...<br>');  
writeln('Halaman ini dibuat dengan ');  
writeln('menggunakan Pascal</h1>');  
end.
```

Listing 8. TESTHALO.PAS, program CGI yang memproses input dari user



Gambar 4. Tampilan TESTHALO yang menerima method GET



Gambar 5. Tampilan TESTHALO yang menerima method POST

## Penutup

Nah, para programmer Pascal..., Anda sekarang dapat berpartisipasi dalam dunia pemrograman berbasis web. Mulailah bermimpi karena sekarang Anda telah dapat bermimpi. Karena semua ide berasal dari mimpi. Tapi hati-hati menggunakan statement `readln` kosong, yang biasanya digunakan untuk menghentikan proses program, karena dapat menjadi mimpi buruk bagi Anda Selamat bermimpi !

## BIOGRAFI PENULIS



**Youppie Arliansyah.** Lahir di Bandung namun besar di Padang. Menamatkan SMA tahun 1993 di SMA 1 Padang, dan Diploma 3 tahun 2000. Tengah menyelesaikan S1 di STMIK Jayanusa Padang. Tertarik pada bidang pemrograman walau tidak pernah mendalami terlalu jauh. Saat ini tengah memusatkan perhatian pada pemrograman web. Juga bekerja sebagai instruktur programmer di Binus Center Padang.

Penulis dapat dihubungi melalui :  
Email : [y\\_arl@yahoo.com](mailto:y_arl@yahoo.com)