

MODUL - WEB PROGRAMMING 1

Oleh :
Priyanto Tamami, S.Kom.

Daftar Isi

1	Dasar-Dasar Pemrograman PHP	1
1.1	Tujuan	1
1.2	Pengantar	1
1.3	Praktek	2
1.3.1	Instalasi XAMPP	2
1.3.2	Instalasi Git dan Akun Github	9
1.3.3	Instalasi Visual Studio Code	24
1.3.4	Registrasi 000webhost.com	31
1.3.5	Aplikasi Selamat Datang	34
1.3.6	Unggah ke Github	40
1.4	Kesimpulan	42
1.5	Tugas	42
2	Operator, Variabel, dan Tipe Data	43
2.1	Tujuan	43
2.2	Pengantar	43
2.3	Praktek	43
2.3.1	Variabel	43
2.3.2	Tipe Data	48
2.3.3	Operator	50
2.4	Kesimpulan	61
2.5	Tugas	61

Bab 1

Dasar-Dasar Pemrograman PHP

1.1 Tujuan

Pada Bab ini diharapkan mahasiswa mengenal bentuk *syntax* dasar dari bahasa pemrograman PHP dan lingkungan kerja yang mendukungnya.

1.2 Pengantar

PHP adalah bahasa pemrograman yang dijalankan di sisi *server* dalam bentuk *scripting*, artinya bahasa pemrograman ini tidak perlu di *compile* terlebih dahulu untuk dapat dijalankan, kita cukup menyiapkan *interpreter*-nya saja.

PHP biasanya digunakan untuk membangun sebuah aplikasi Web yang dinamis, dimana halaman dapat melakukan respon terhadap *request* yang dilakukan oleh pengguna.

PHP pun telah digunakan secara luas dan menjadi alternatif gratis dibandingkan menggunakan bahasa sejenis seperti ASP milik Microsoft.

Untuk memulai melakukan praktek bahasa pemrograman menggunakan PHP, maka kita perlu mempersiapkan perangkat pendukung. Cara yang mungkin paling mudah adalah kita menggunakan aplikasi paket yang di dalamnya sudah terdapat *web server* yang mendukung PHP serta basis data yang akan digunakan, aplikasi yang mungkin dapat kita gunakan adalah :

1. LAMP, yang sebetulnya adalah singkatan dari Linux, Apache, MySQL, dan PHP. Tentunya aplikasi ini ditujukan untuk sistem operasi Linux, yang menggunakan Apache sebagai *web server* yang tentunya *plugin* untuk mendukung bahasa PHP sudah ada di dalamnya, dan MySQL sebagai basis datanya.
2. WAMP, adalah singkatan dari Windows, Apache, MySQL, dan PHP. Mirip seperti LAMP, hanya ini ditujukan bagi sistem operasi Windows.
3. MAMP, adalah singkatan dari Mac, Apache, MySQL, dan PHP. Untuk aplikasi ini dikhususkan bagi sistem operasi Mac.

4. XAMPP, yang ini mendukung ketiga sistem operasi di atas dengan kelebihan mampu untuk mengolah bahasa pemrograman Perl.

Maka pilihan untuk praktek Web Programming 1 kita akan menggunakan XAMPP agar adaptasi antar sistem operasi lebih mudah. XAMPP dapat diunduh pada alamat <https://www.apachefriends.org>.

Aplikasi pendukung lain untuk melakukan kegiatan praktikum kita adalah sebagai berikut :

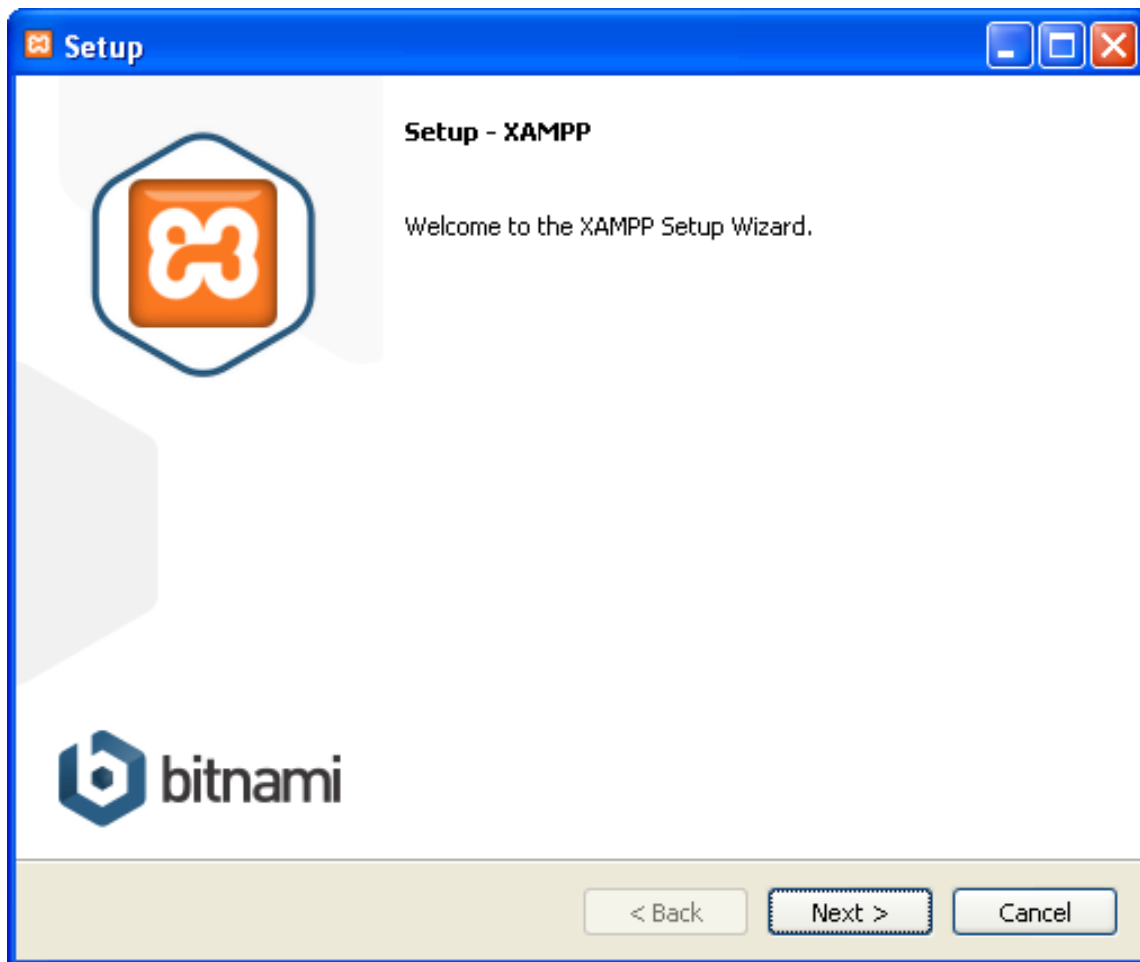
1. Git. Aplikasi ini digunakan untuk melakukan *versioning* sehingga kita lebih mudah dalam melakukan kontrol perubahan yang terjadi pada kode program yang kita bangun. Server yang kita gunakan untuk menyimpan repositori hasil *versioning* kita ada di alamat <https://github.com>. Github ini gratis. Untuk aplikasi Git dapat kita unduh di alamat <https://git-scm.com/>
2. Visual Studio Code. Aplikasi ini adalah *editor* yang akan digunakan dalam kegiatan praktikum pada mata kuliah Web Programming 1. Aplikasi ini gratis dan dapat diunduh pada alamat <https://code.visualstudio.com/> dengan dukungan instalasi untuk 3 (tiga) sistem operasi yang banyak digunakan, yaitu Linux, Windows, dan MacOS.
3. www.000webhost.com. Ini adalah layanan *hosting* gratis yang mampu menjalankan *script* PHP dengan fasilitas sistem basis data MySQL. Yang akan kita gunakan sebagai tempat aplikasi yang telah kita bangun sampai dengan akhir tatap muka mata kuliah ini.

1.3 Praktek

1.3.1 Instalasi XAMPP

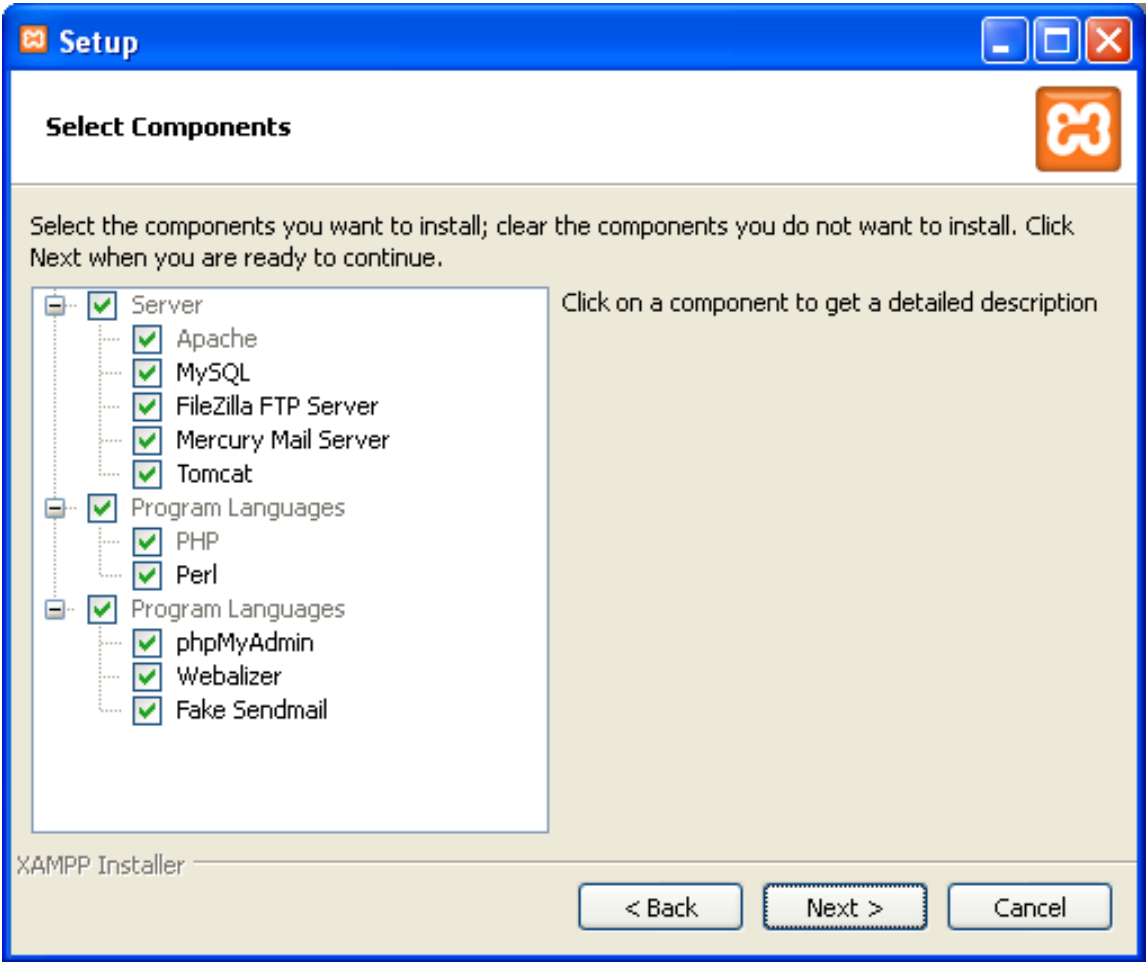
Langkah-langkah untuk instalasi XAMPP adalah sebagai berikut :

1. Tentunya mengunduh *installer* aplikasi ini dari alamat <https://www.apachefriends.org>, pilihlah yang model *installer* agar lebih mudah proses instalasinya.
2. Awal proses instalasi, kita akan bertemu dengan tampilan seperti pada gambar 1.1 berikut ini :



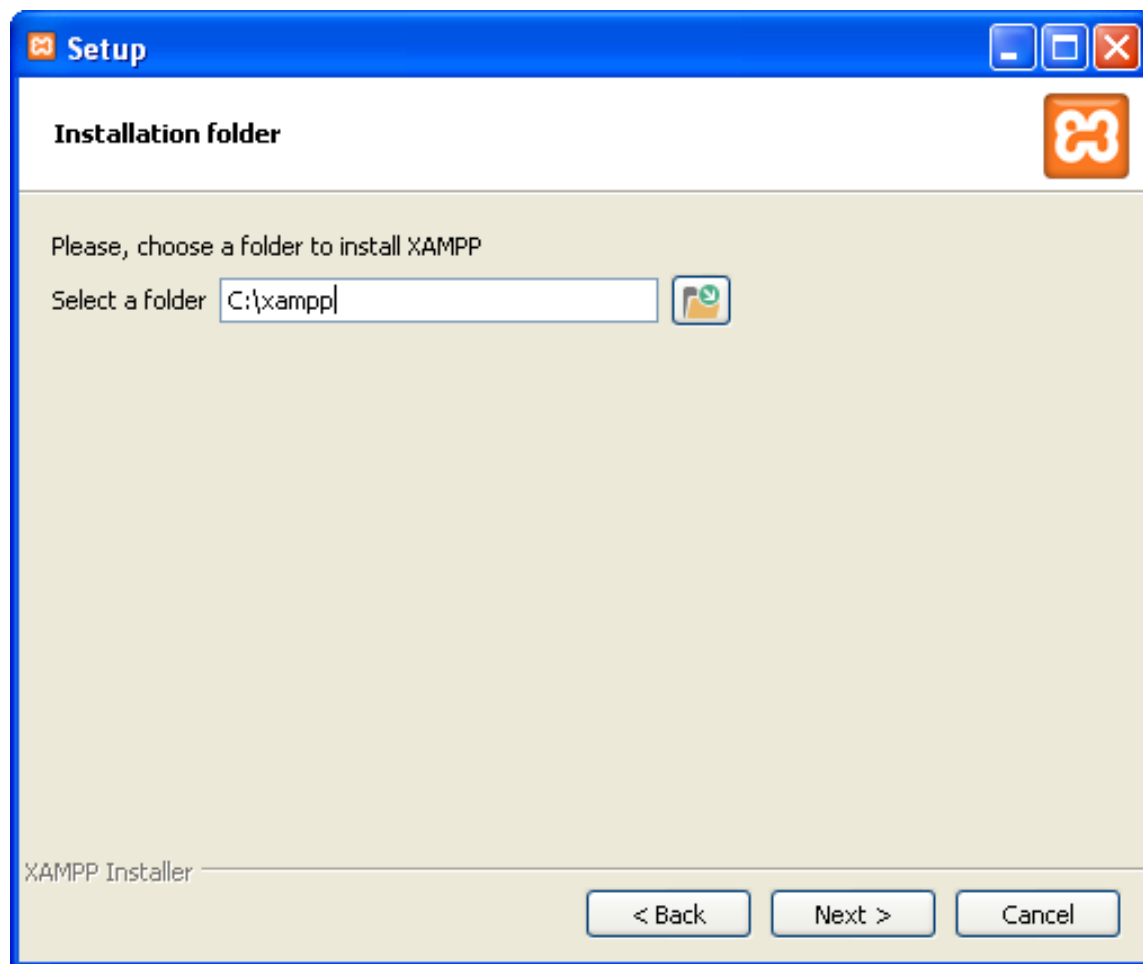
Gambar 1.1: Awal Instalasi XAMPP

3. Setelah itu kita diminta untuk memilih komponen yang akan ikut diinstall / dipasangkan seperti pada gambar 1.2 berikut :

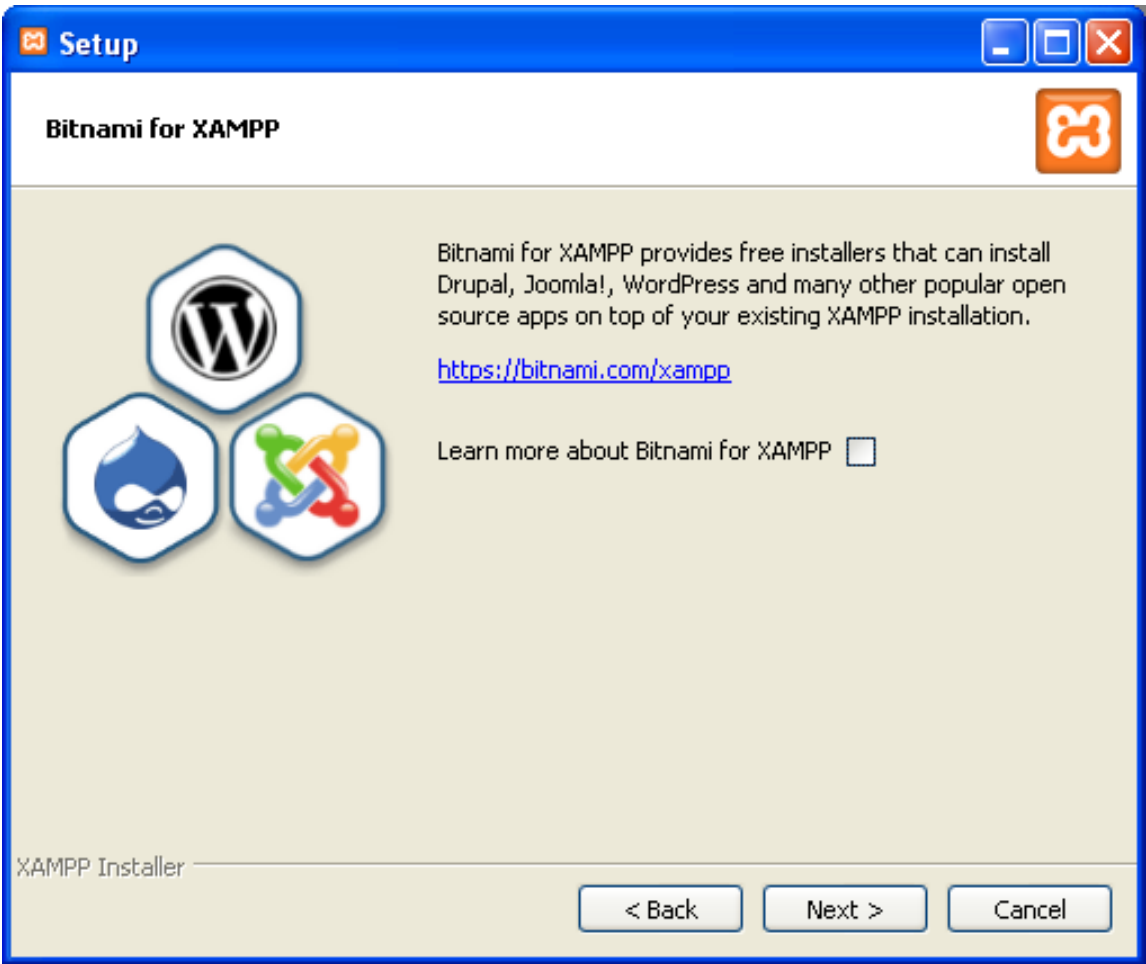


Gambar 1.2: Pemilihan Komponen

4. Selanjutnya adalah penetapan *folder* instalasi, akan ditaruh dimana hasil instalasi dari XAMPP ini. Tampilannya seperti pada gambar 1.26 berikut ini :

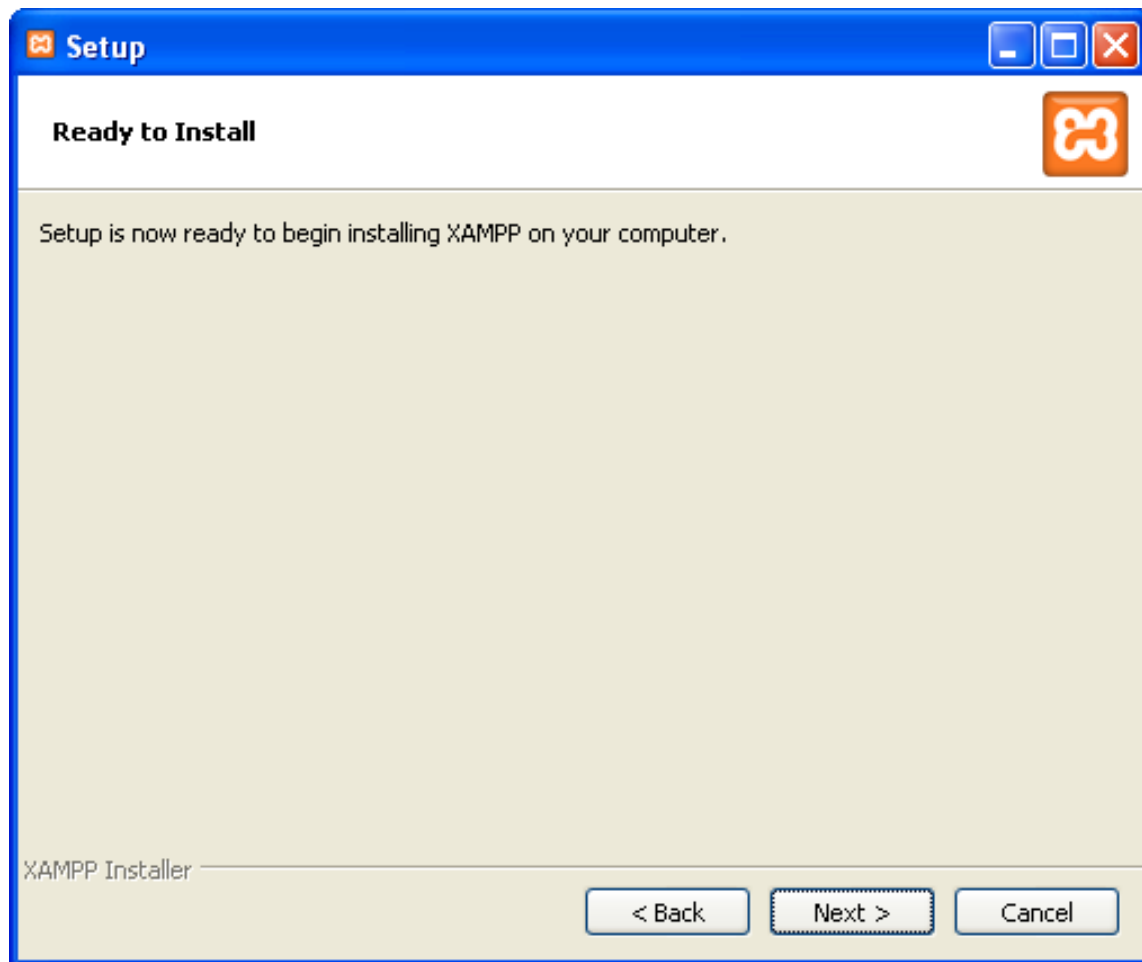
Gambar 1.3: Pemilihan *Folder* Instalasi

5. Langkah berikutnya adalah informasi mengenai Bitnami, Bitnami ini adalah aplikasi gratis yang mampu melakukan *instalasi* aplikasi lain semacam Drupal atau Joomla di atas XAMPP, tampilannya seperti pada gambar 1.4 berikut :



Gambar 1.4: Informasi Bitnami

- 6. Setelah menekan tombol **Next**, maka akan muncul informasi bahwa aplikasi siap diinstall seperti pada gambar 1.5 berikut :



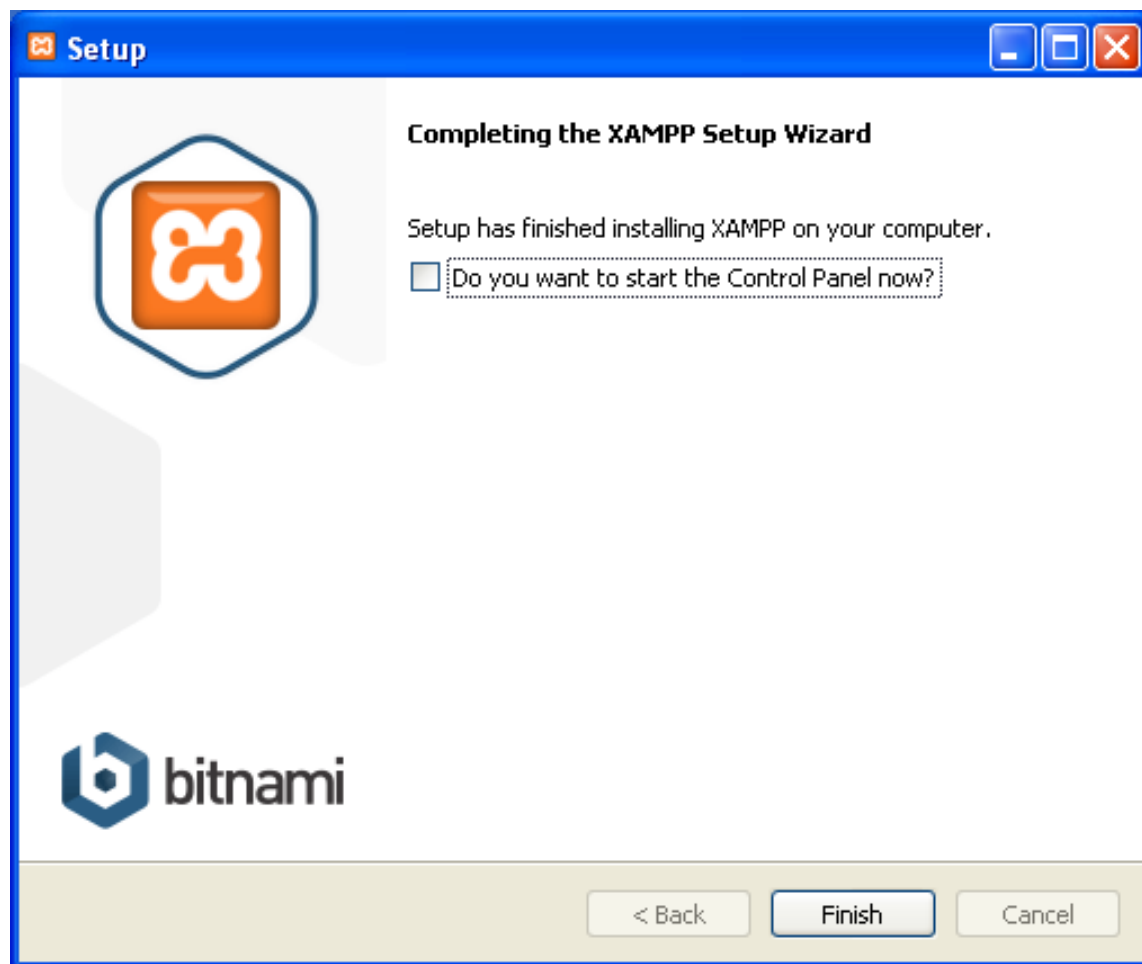
Gambar 1.5: Instalasi Sudah Siap

7. Setelah itu proses instalasi akan berjalan seperti pada gambar 1.6 berikut :



Gambar 1.6: Instalasi berjalan

8. Setelah proses instalasi selesai, maka akan muncul informasi seperti pada gambar 1.7 berikut :



Gambar 1.7: Instalasi Selesai

Sampai sini, proses instalasi XAMPP telah berhasil dan siap untuk digunakan.

1.3.2 Instalasi Git dan Akun Github

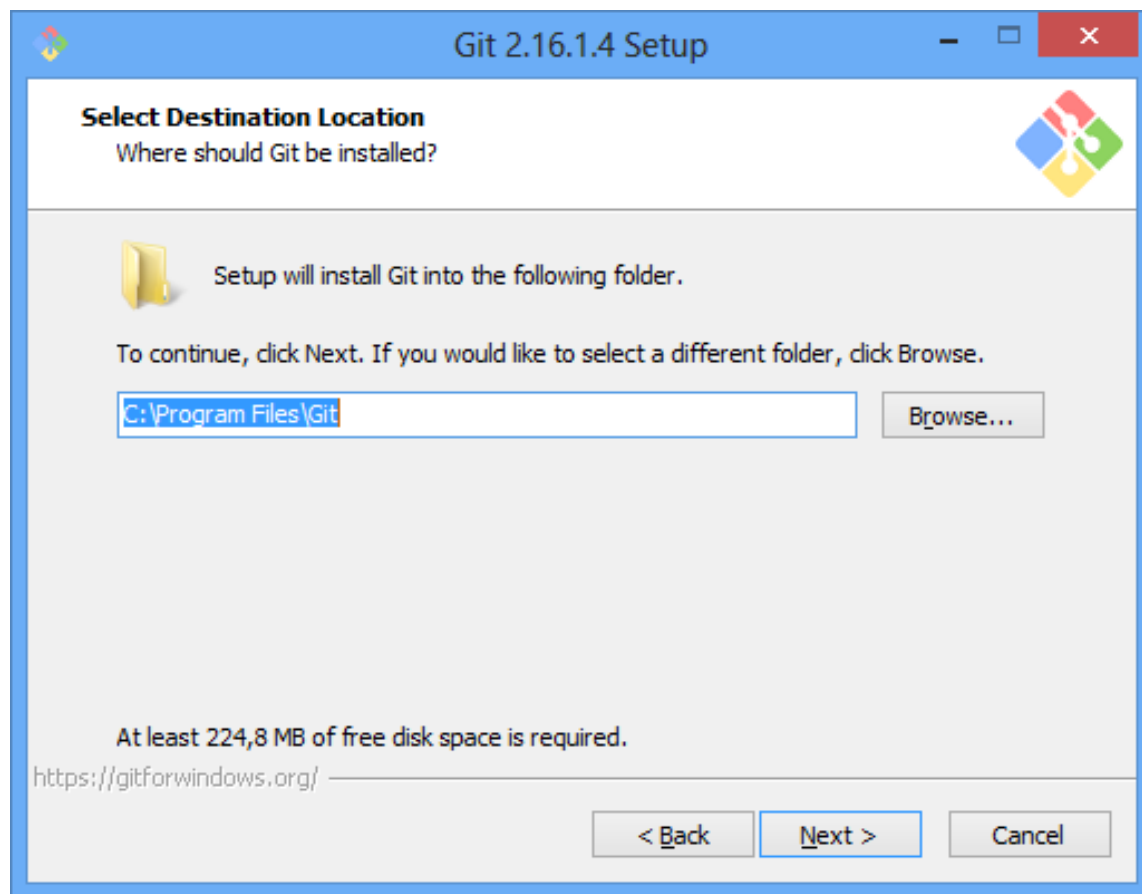
Proses instalasi dari Git adalah sebagai berikut :

1. Awalnya akan disajikan informasi mengenai lisensi seperti pada gambar 1.8 berikut :



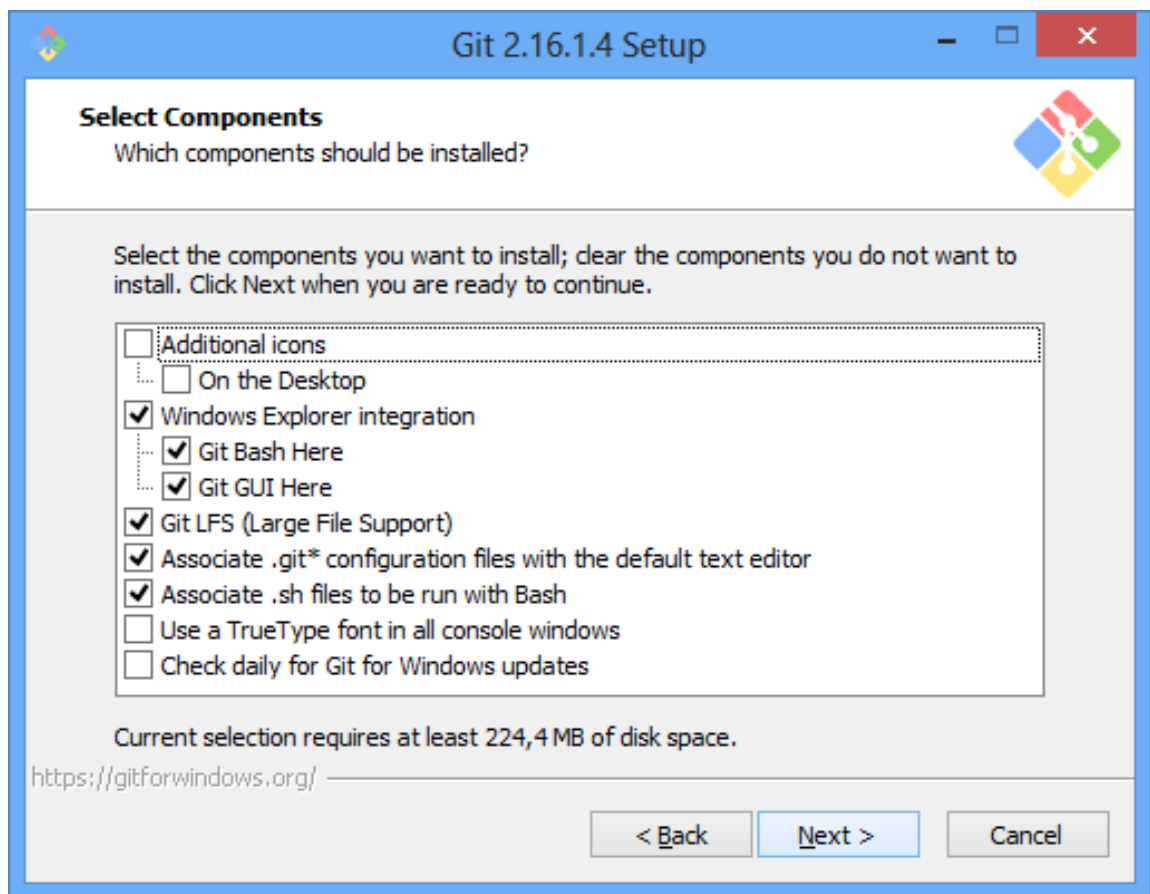
Gambar 1.8: Informasi Lisensi

- 2. Berikutnya akan ditampilkan lokasi dimana aplikasi Git akan diinstall, tampilannya adalah seperti pada gambar 1.9 berikut :



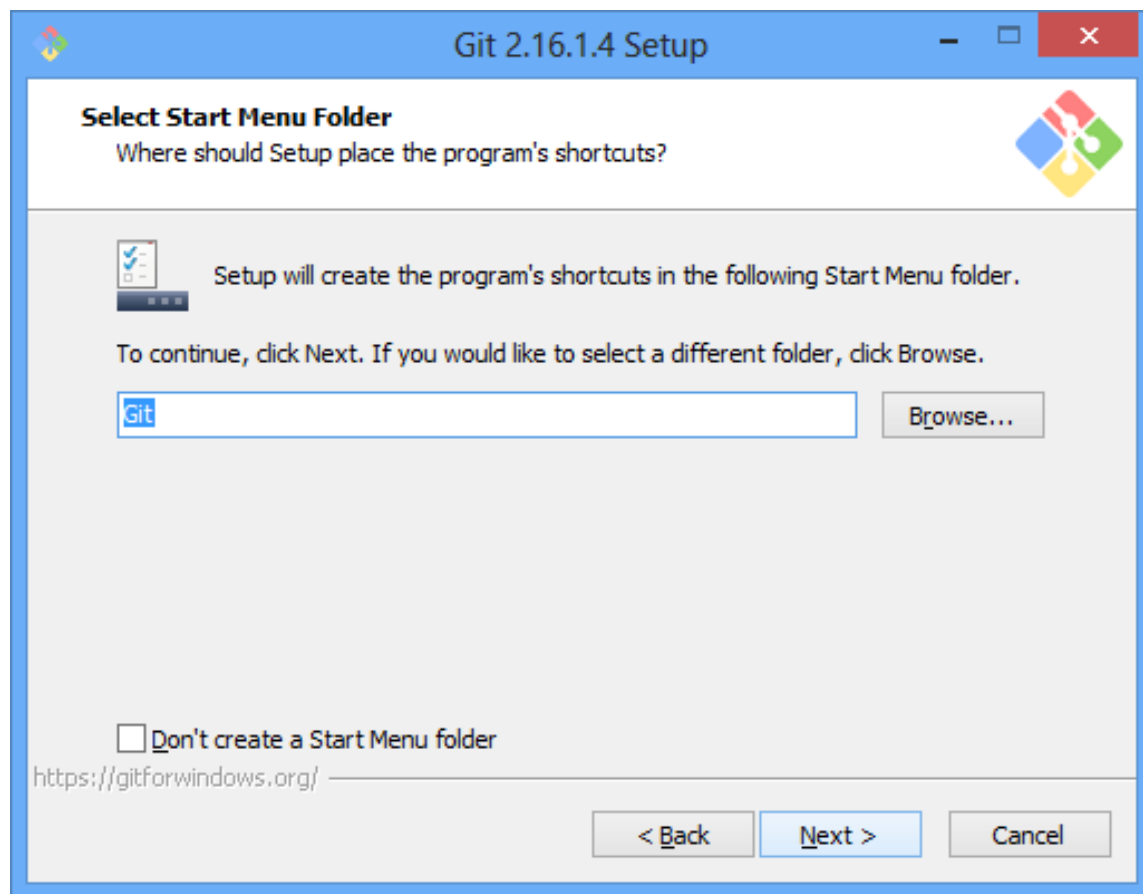
Gambar 1.9: Lokasi Tempat Instalasi Git

3. Berikutnya adalah pemilihan komponen yang akan diinstall di komputer, tampilannya adalah seperti pada gambar 1.10 berikut :

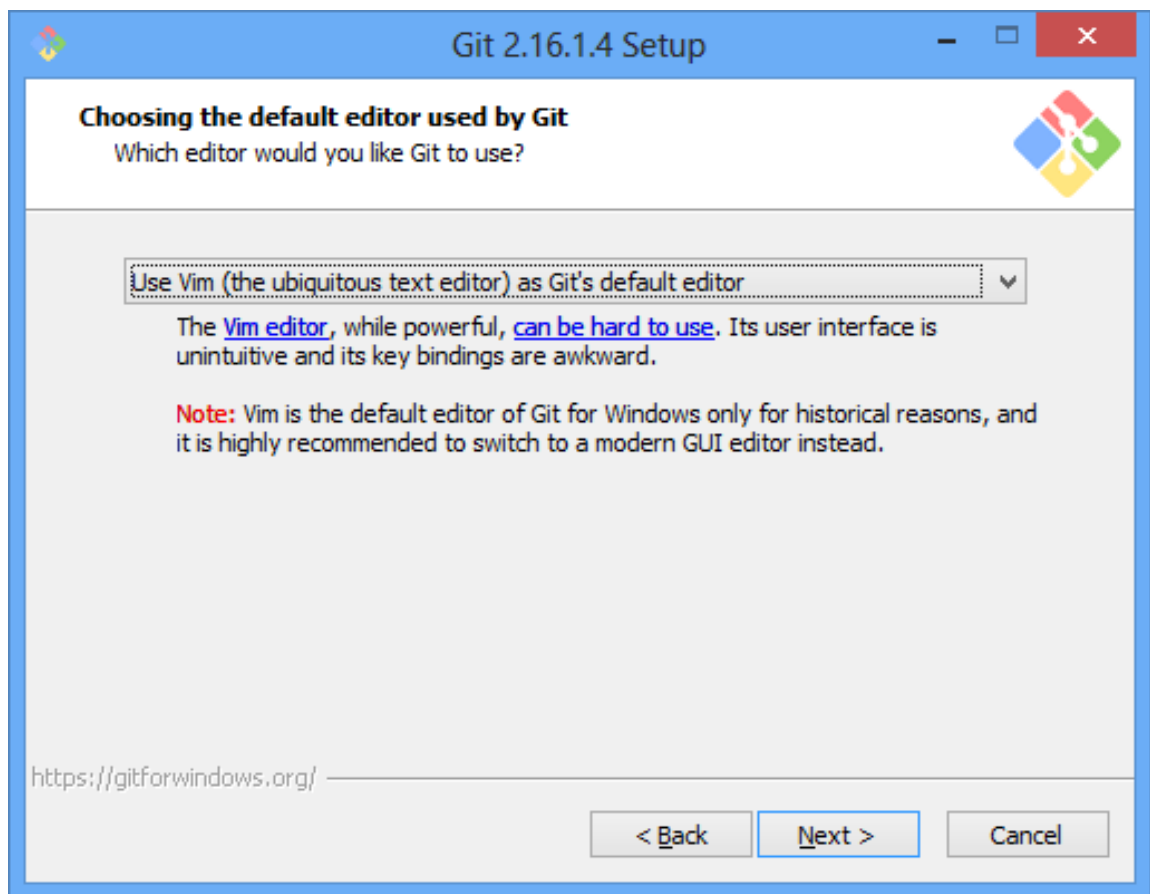


Gambar 1.10: Pemilihan Komponen

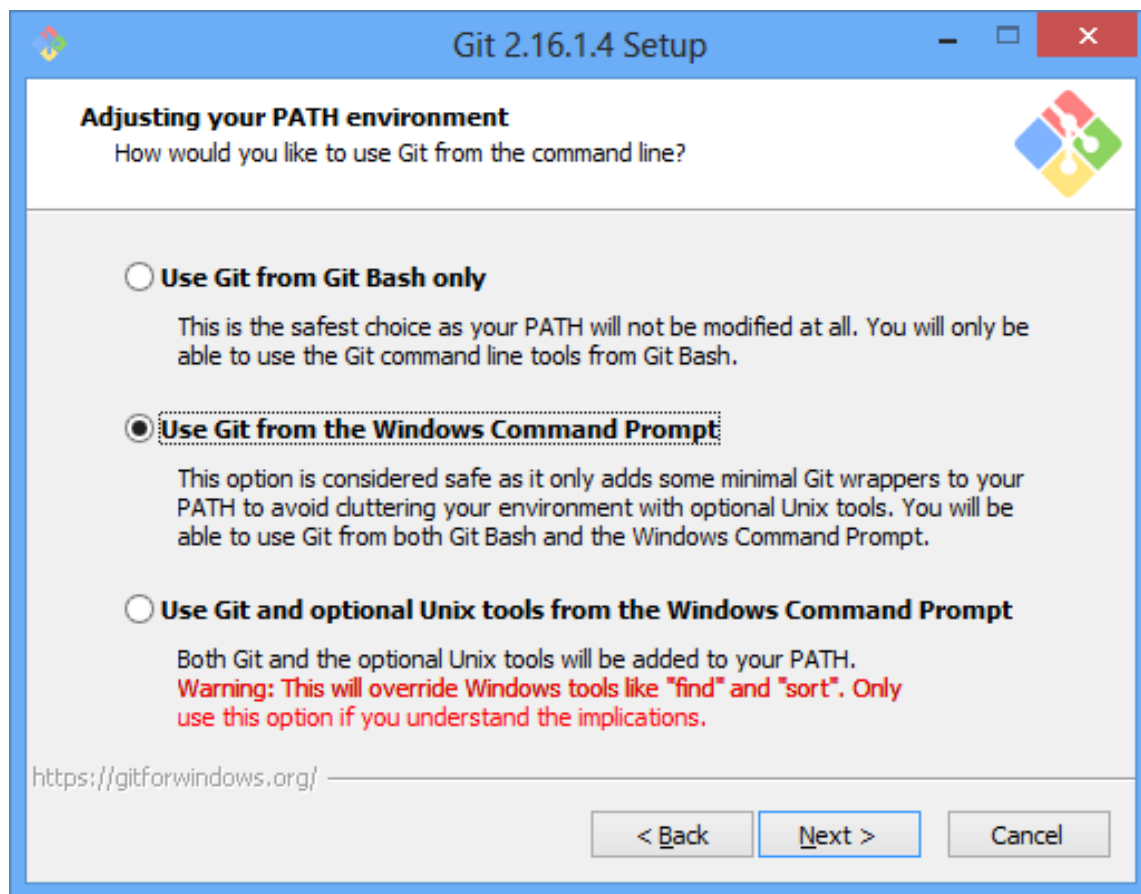
4. Selanjutnya adalah lokasi pada **start menu** dimana Git akan ditempatkan, tampilannya adalah seperti pada gambar 1.11 berikut :

Gambar 1.11: Lokasi di **Start Menu**

5. Setelah itu kita diminta untuk memilih *editor* yang digunakan oleh Git seperti pada gambar 1.12 berikut :

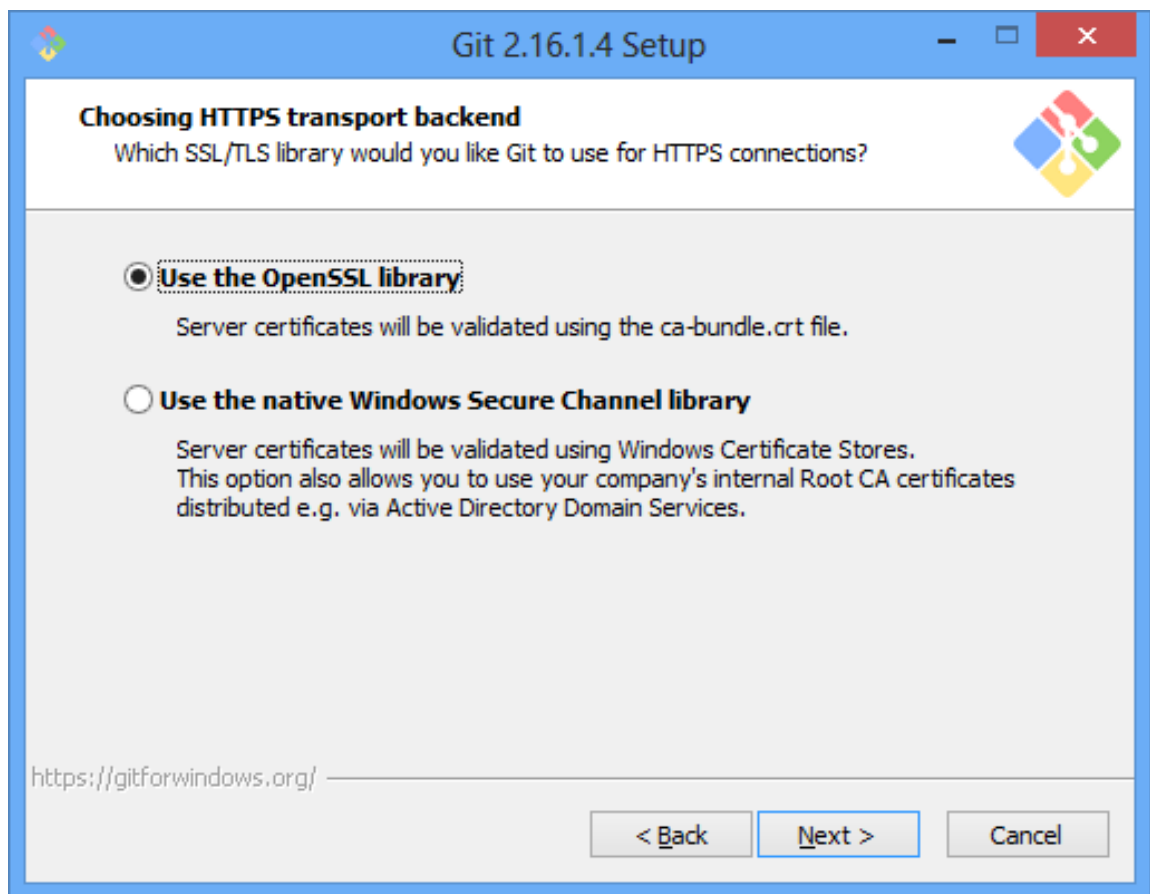
Gambar 1.12: Pemilihan *Editor*

- Langkah berikutnya adalah mengatur *environment variable* PATH untuk penggunaan Git melalui *console* atau *command prompt*, tampilannya adalah seperti pada gambar 1.13 berikut :



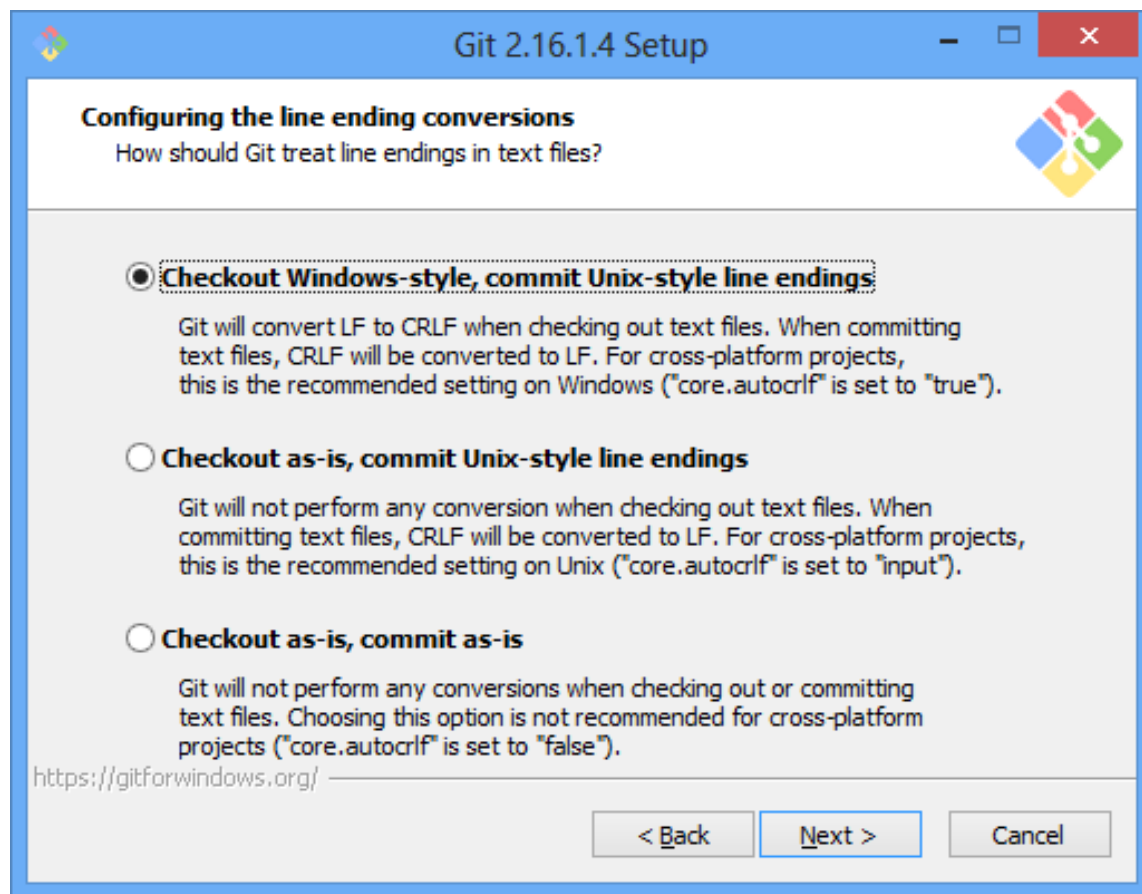
Gambar 1.13: Pengaturah PATH

7. Selanjutnya adalah memilih pustaka SSL yang akan digunakan untuk koneksi HTTPS, tampilannya adalah seperti pada gambar 1.14 berikut :

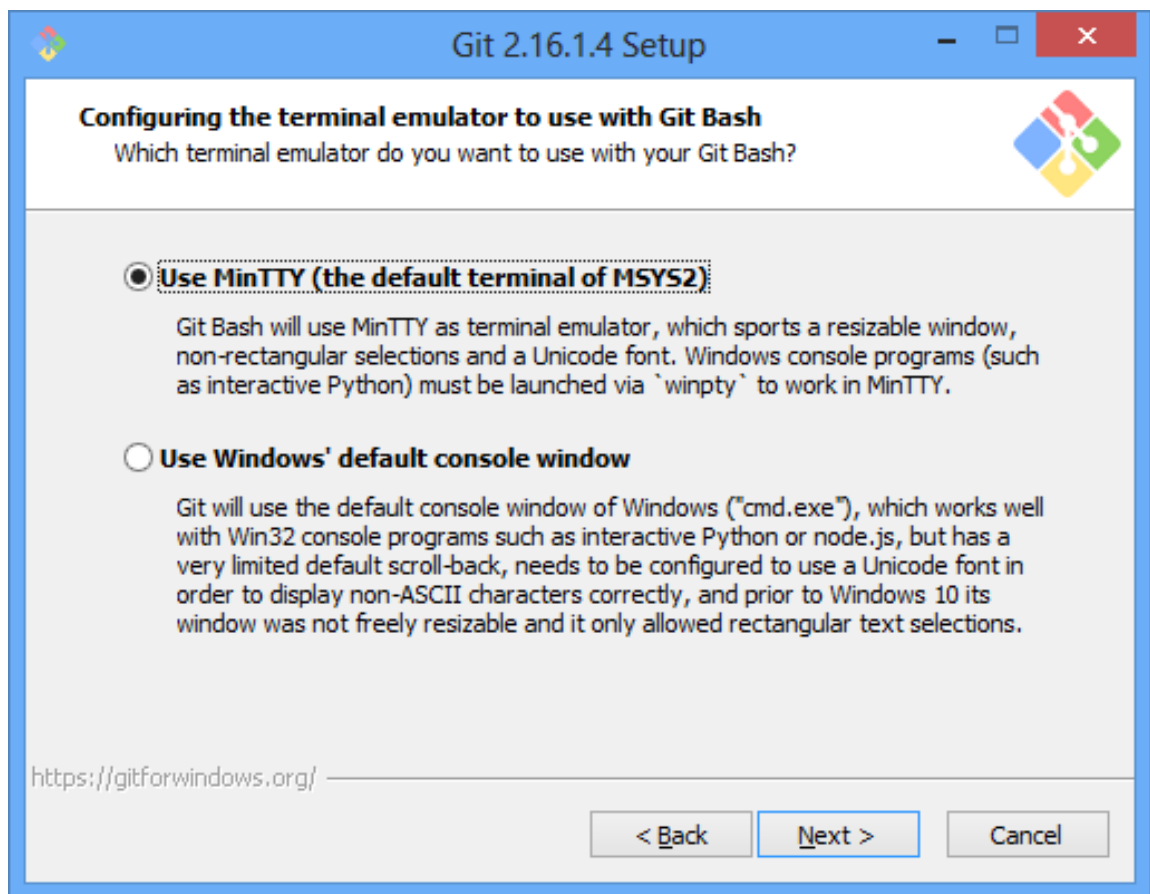


Gambar 1.14: Pemilihan Pustaka SSL

8. Berikutnya adalah memilih bagaimana cara Git memperlakukan *line endings*, tampilannya adalah seperti pada gambar 1.15 berikut :

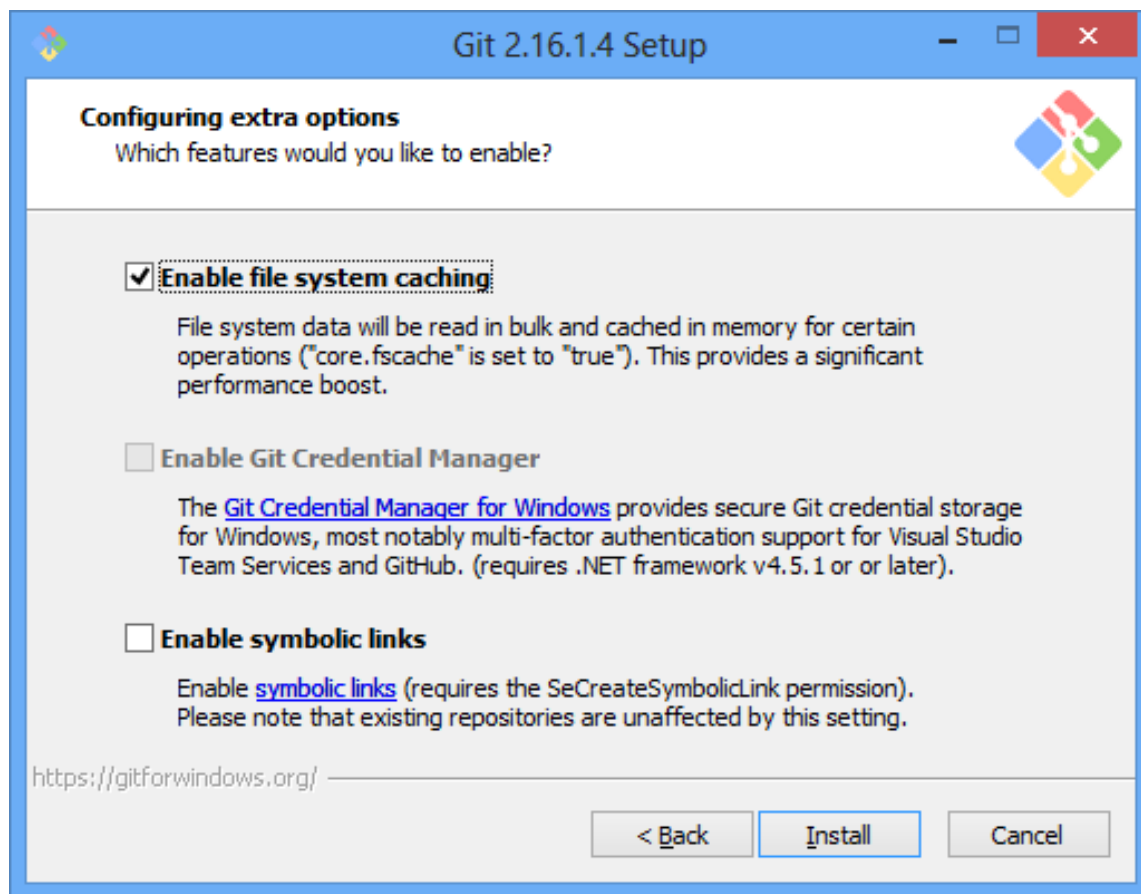
Gambar 1.15: Pemilihan Konversi *Line Ending*

9. Selanjutnya adalah pemilihan emulator terminal yang akan digunakan oleh Git, tampilannya seperti pada gambar 1.16 berikut :



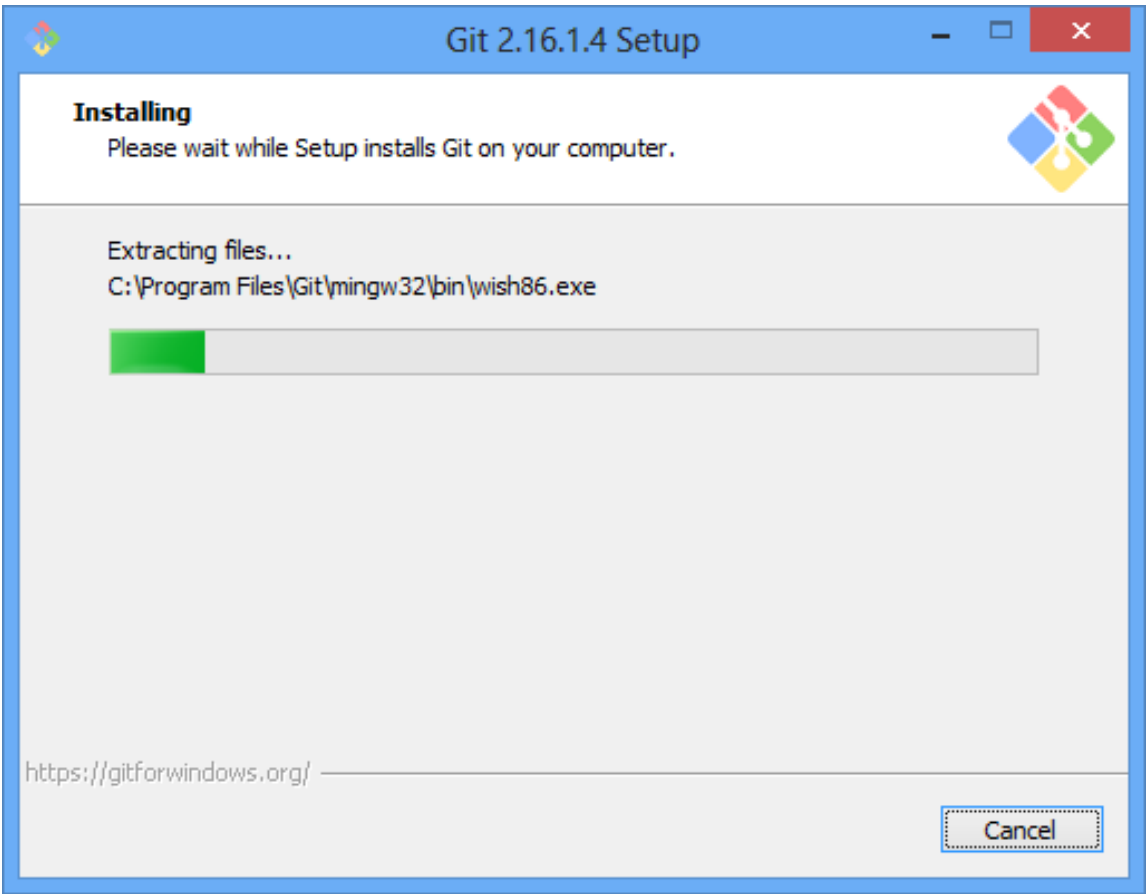
Gambar 1.16: Pemilihan Emulator Terminal

- Langkah berikutnya adalah memilih fasilitas yang akan diaktifkan, tampilannya akan terlihat seperti pada gambar 1.17 berikut :



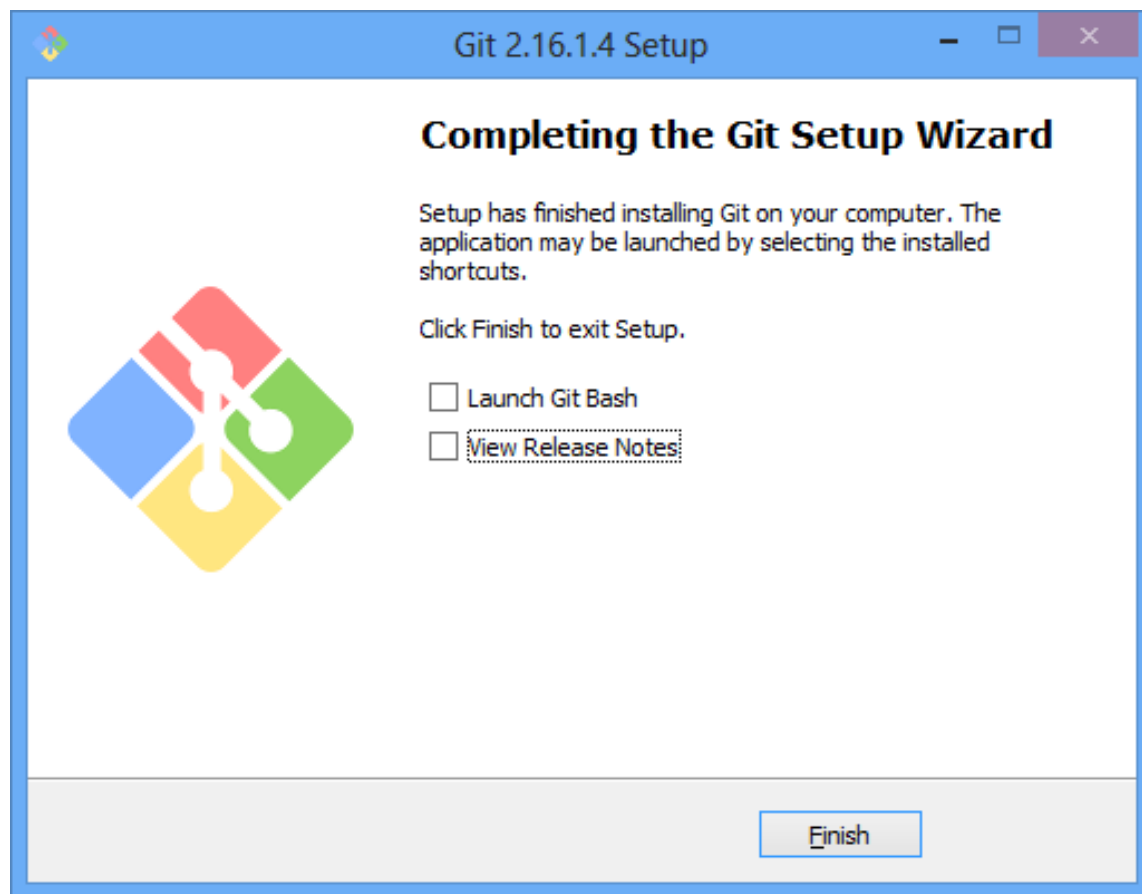
Gambar 1.17: Pemilihan Fasilitas Tambahan

11. Setelah menekan tombol **Install**, maka proses instalasi akan berjalan seperti pada gambar 1.18 berikut :



Gambar 1.18: Proses Instalasi

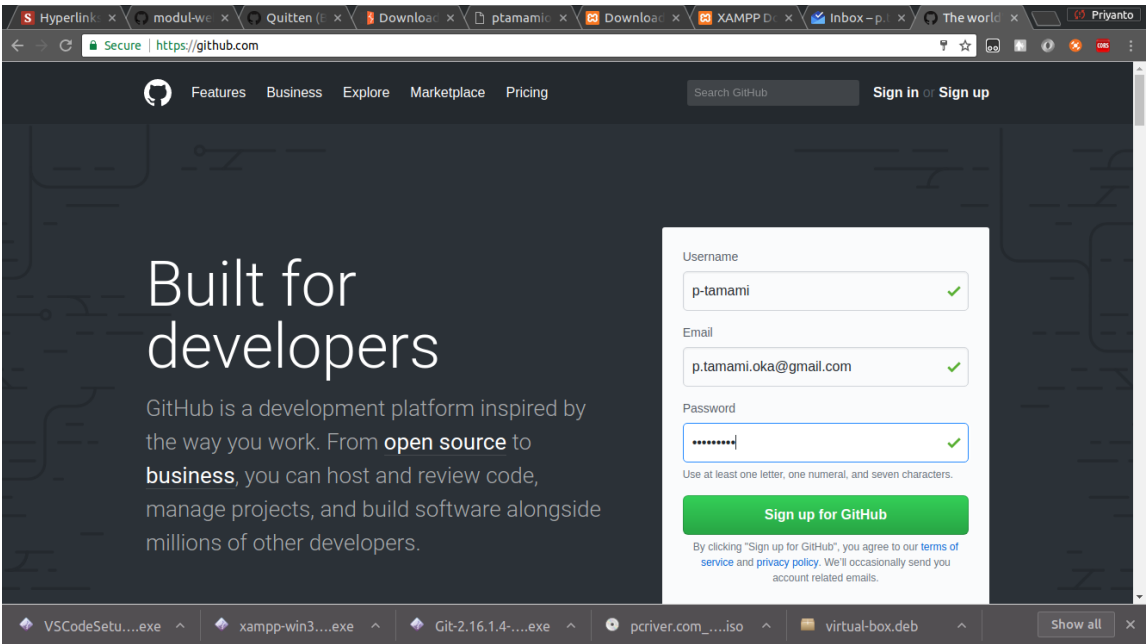
- 12. Setelah proses instalasi selesai, maka akan muncul jendela seperti pada gambar 1.19 berikut :



Gambar 1.19: Instalasi Selesai

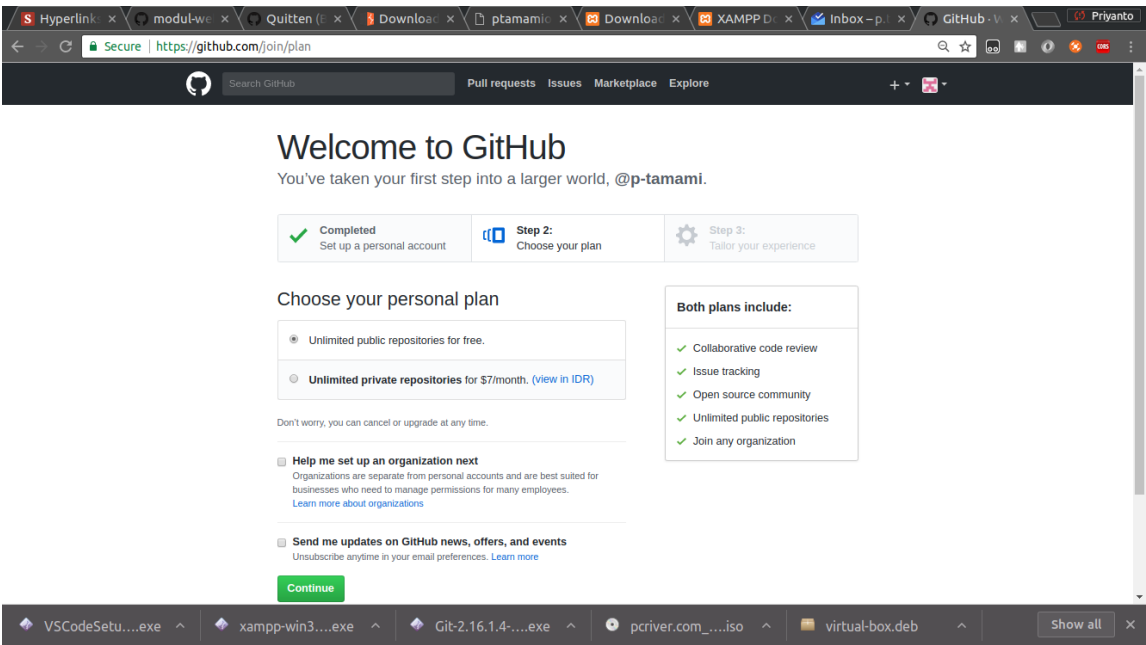
Berikutnya kita akan bahas bagaimana caranya membuat akun di Github agar kita dapat menyimpan kode yang telah kita bangun di repositori Github. Langkah-langkahnya adalah sebagai berikut :

1. Bukalah *browser* dan isikan alamatnya dengan **github.com**, sehingga nanti akan muncul tampilan seperti pada gambar 1.20 berikut :



Gambar 1.20: Halaman Depan Github

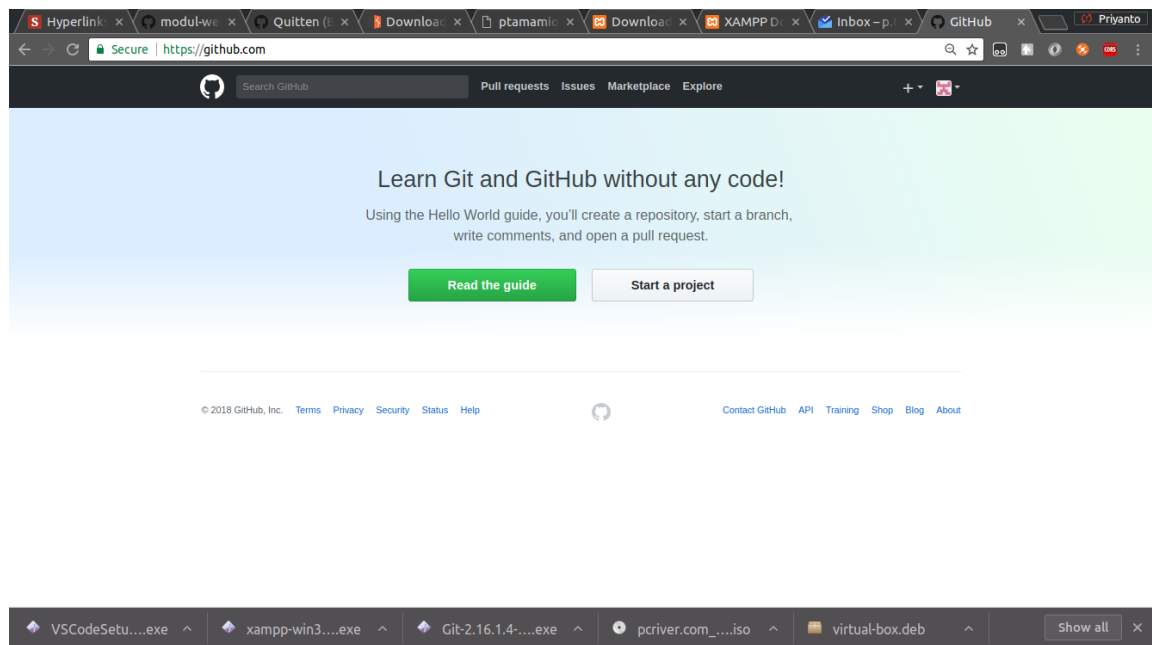
2. Setelah mengisikan *username*, alamat *email*, dan *password*, kemudian menekan tombol **Sign up for Github**, maka akan tampilan halaman berikutnya seperti pada gambar 1.21 berikut :



Gambar 1.21: Halaman Selamat Datang Github

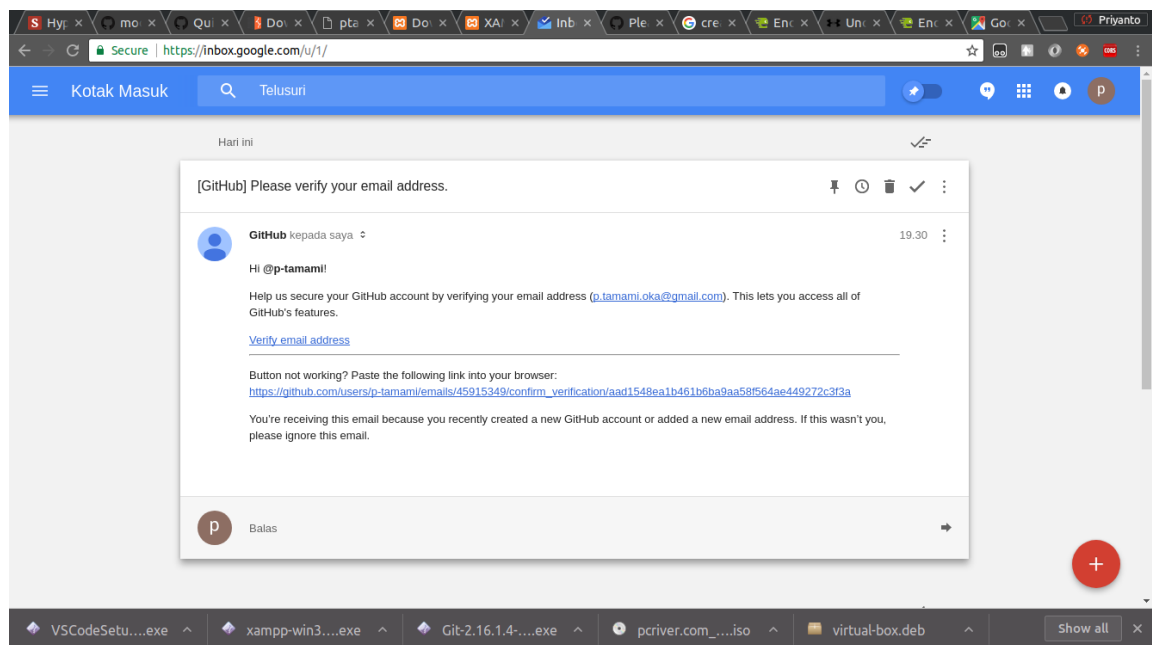
3. Karena kita menggunakan versi gratis, jadi lanjutkan saja dengan menekan tombol **Continue** sehingga muncul tampilan berikutnya seperti pada gambar

1.22 :



Gambar 1.22: Halaman Pembuka Github

4. Selanjutnya, untuk membuat repositori, kita diwajibkan melakukan verifikasi alamat *email*, bukanlah *email* yang kita gunakan untuk mendaftar Github, lalu klik *link* yang bertuliskan *Verify email address* seperti pada gambar 1.23 berikut :



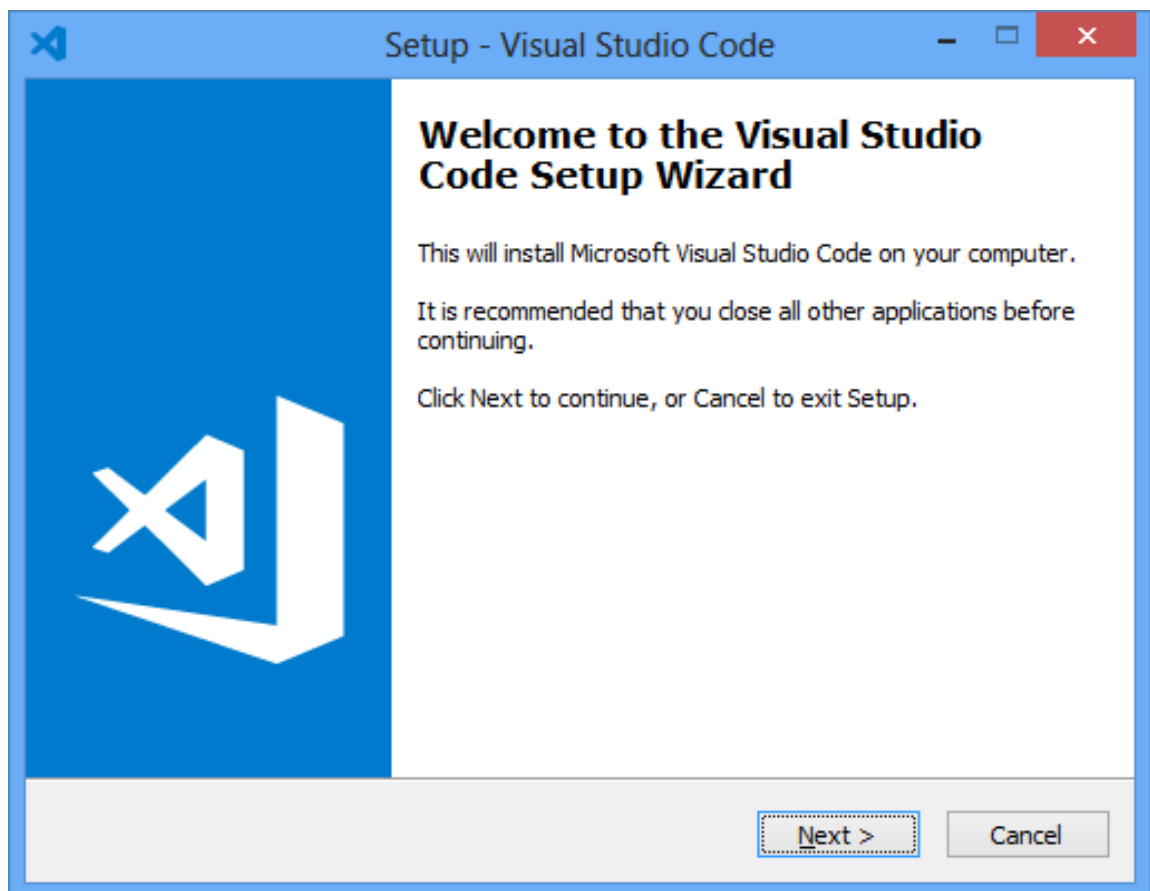
Gambar 1.23: Verifikasi Alamat *Email*

Sampai langkah ini Github sudah dapat kita gunakan untuk menyimpan kode program yang kita bangun.

1.3.3 Instalasi Visual Studio Code

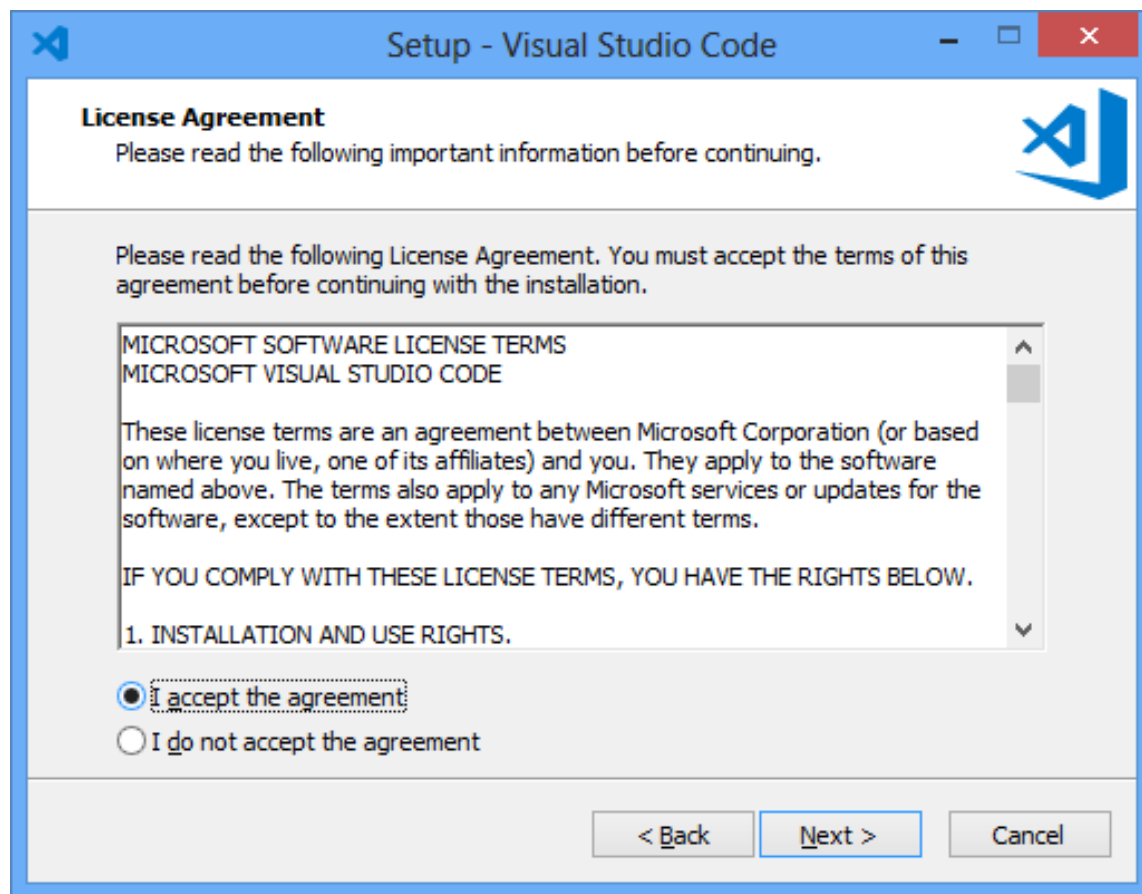
Proses instalasi Visual Studio Code adalah sebagai berikut :

1. Pertama kita akan disajikan jendela selamat datang seperti biasa, ditunjukkan oleh gambar 1.24 berikut :



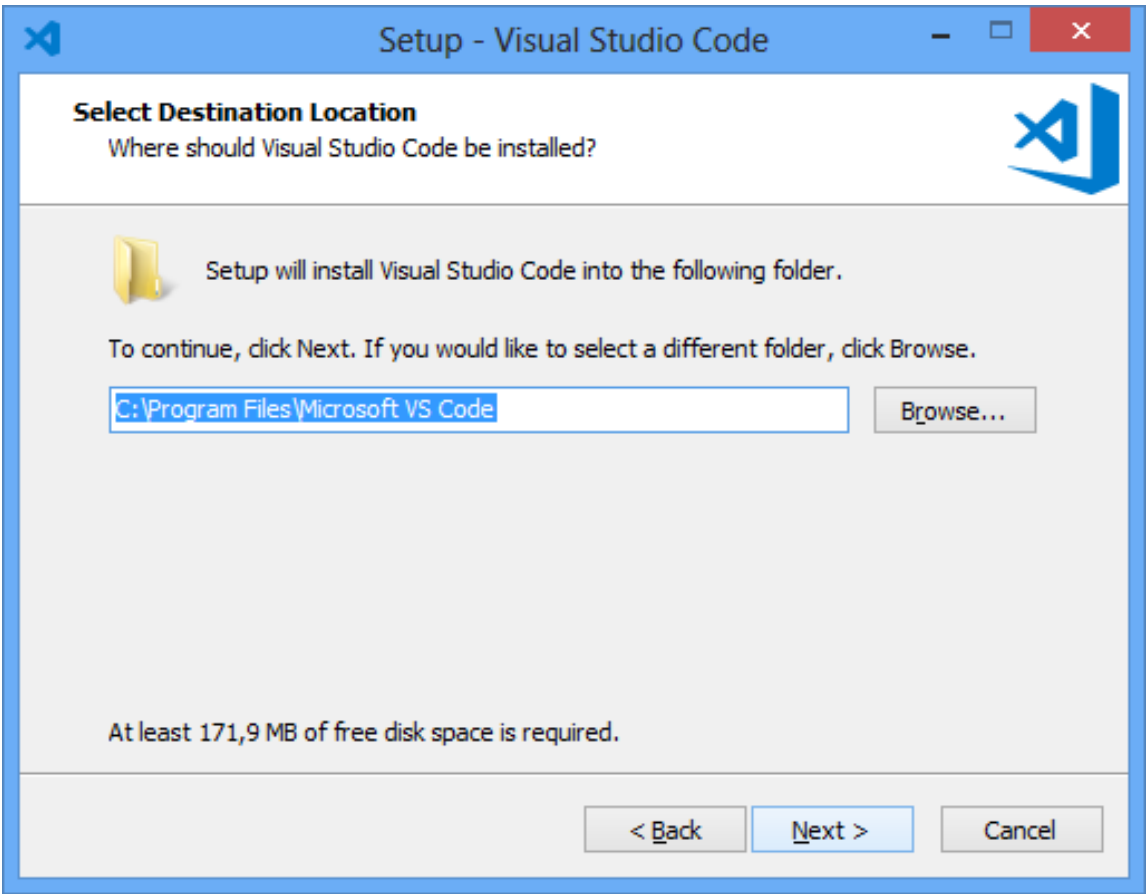
Gambar 1.24: Halaman Selamat Datang dari Visual Studio Code

2. Jendela berikutnya akan menampilkan informasi mengenai lisensi yang digunakan, tampilannya akan terlihat seperti gambar 1.25 berikut :



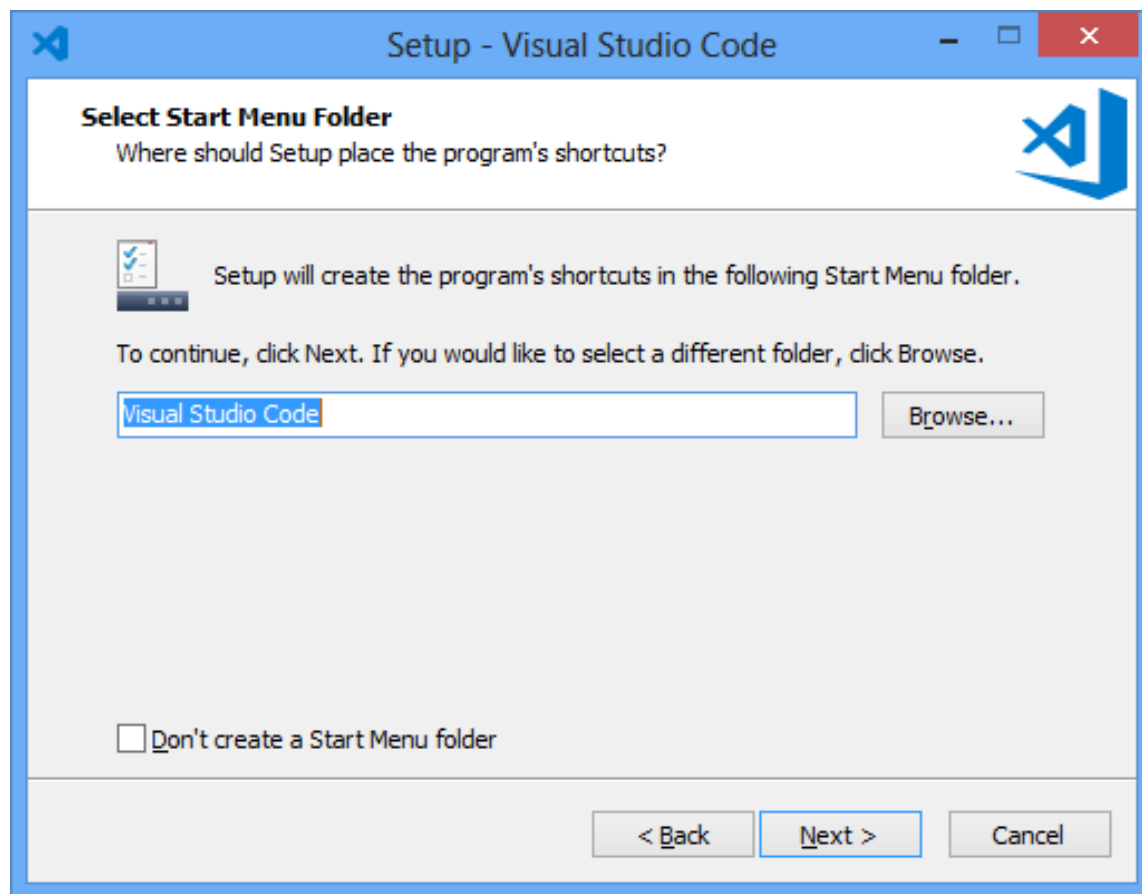
Gambar 1.25: Informasi Lisensi

3. Selanjutnya adalah menentukan lokasi direktori dimana *Visual Studio Code* akan ditempatkan, tampilannya adalah seperti pada gambar 1.26 berikut ini :

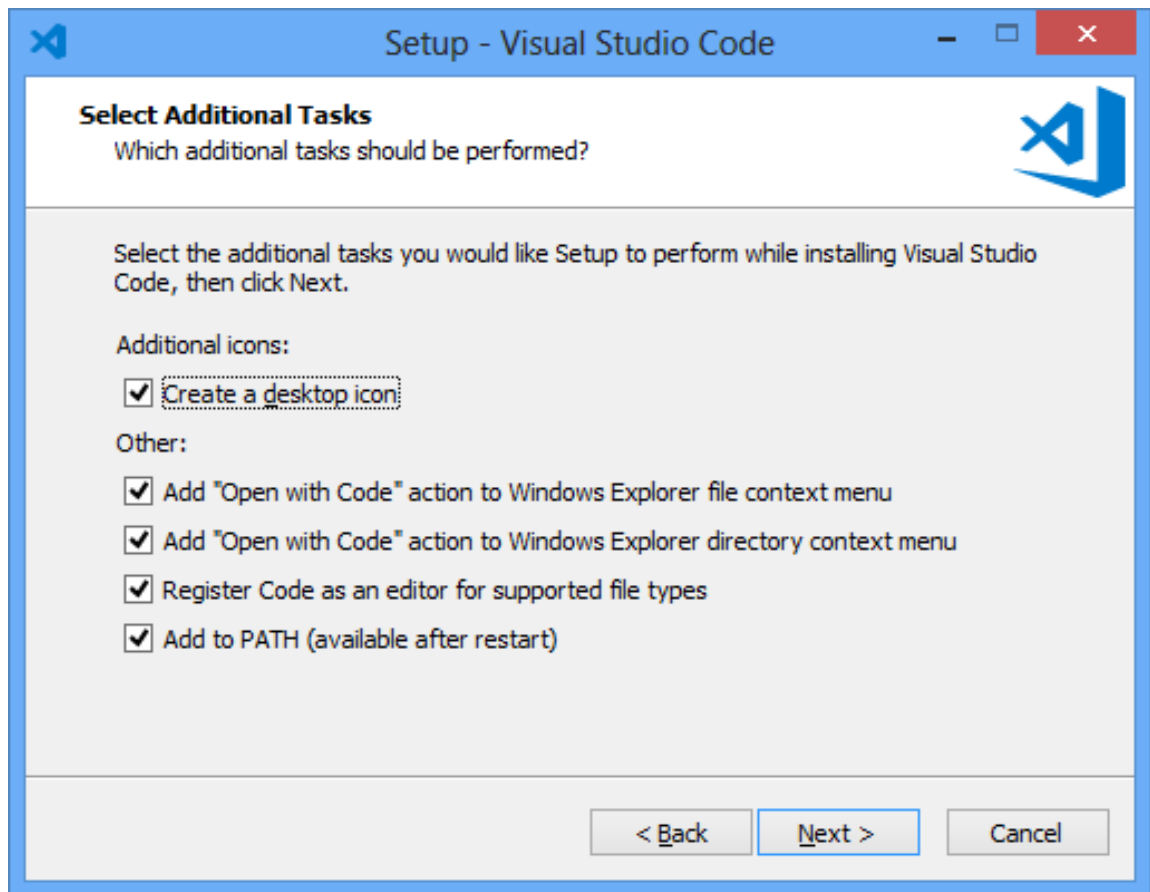


Gambar 1.26: Lokasi Direktori Instalasi

- 4. Berikutnya adalah menentukan lokasi direktori pada *start menu*, tampilannya akan terlihat seperti pada gambar 1.27 berikut :

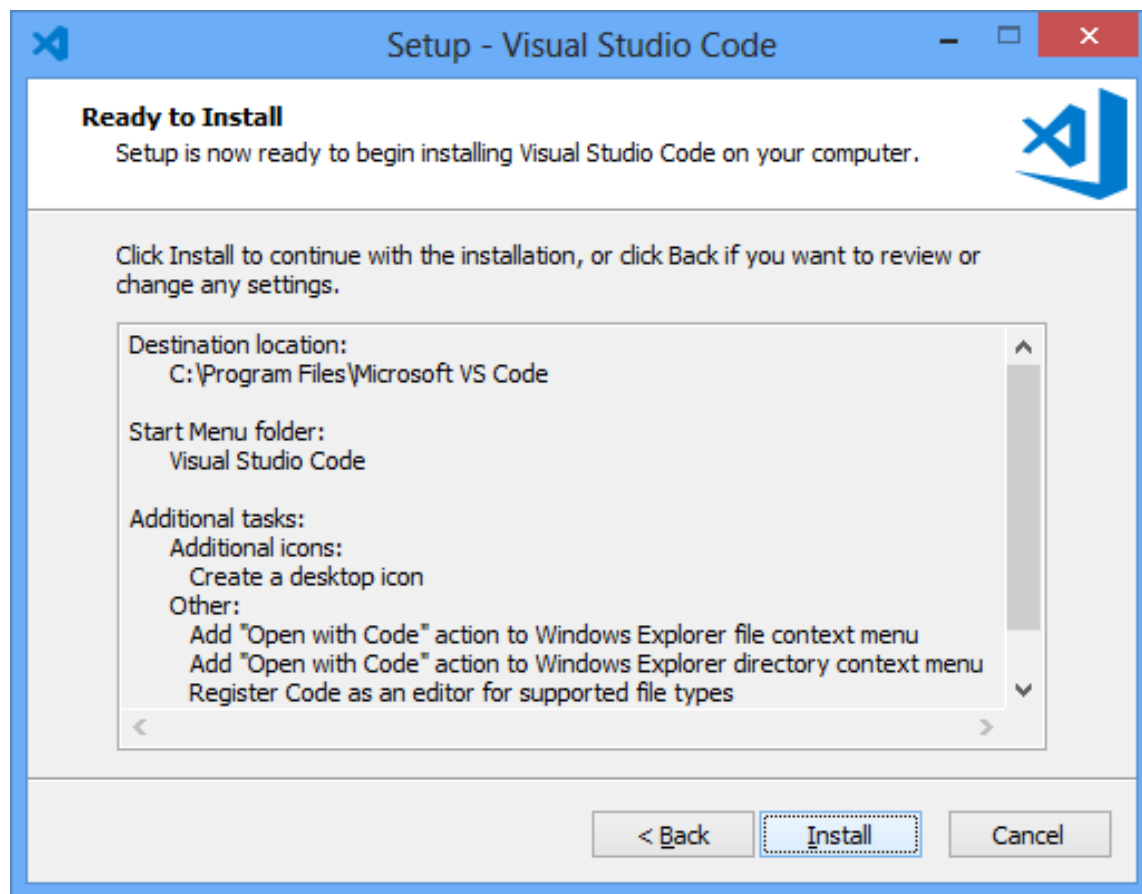
Gambar 1.27: Lokasi Pada *Start Menu*

5. Jendela berikutnya menampilkan pilihan agar pada saat instalasi dilakukan beberapa konfigurasi, tampilannya akan terlihat seperti pada gambar 1.28 berikut :



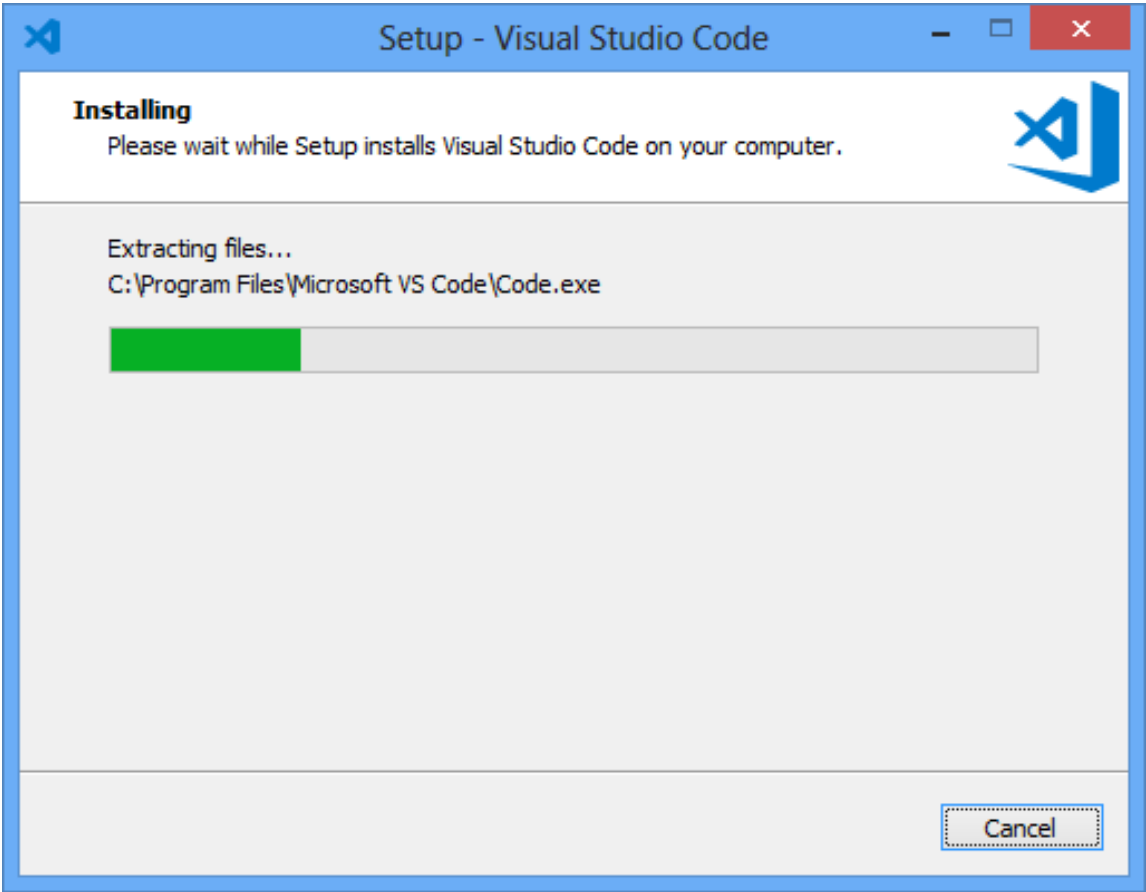
Gambar 1.28: Konfigurasi Tambahan

6. Langkah berikutnya, jendela instalasi akan menampilkan ringkasan dan proses instalasi siap dijalankan, tampilannya akan terlihat seperti gambar [1.29](#) berikut :



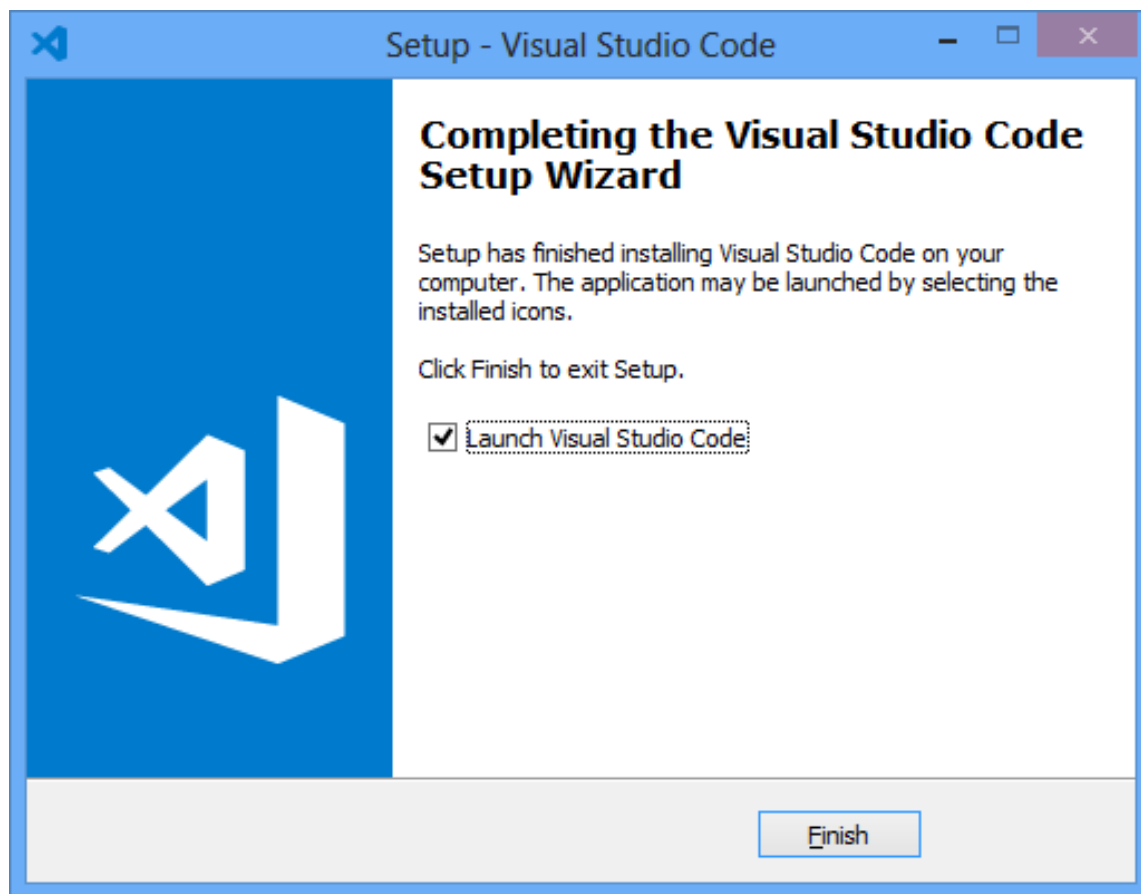
Gambar 1.29: Proses Instalasi Telah Siap

7. Jendela berikutnya akan menampilkan status proses instalasi, tampilannya akan terlihat seperti pada gambar [1.30](#) berikut :



Gambar 1.30: Proses Instalasi Berjalan

8. Terakhir adalah informasi bahwa proses instalasi telah selesai dan berhasil, tampilannya akan terlihat seperti pada gambar 1.31 berikut :

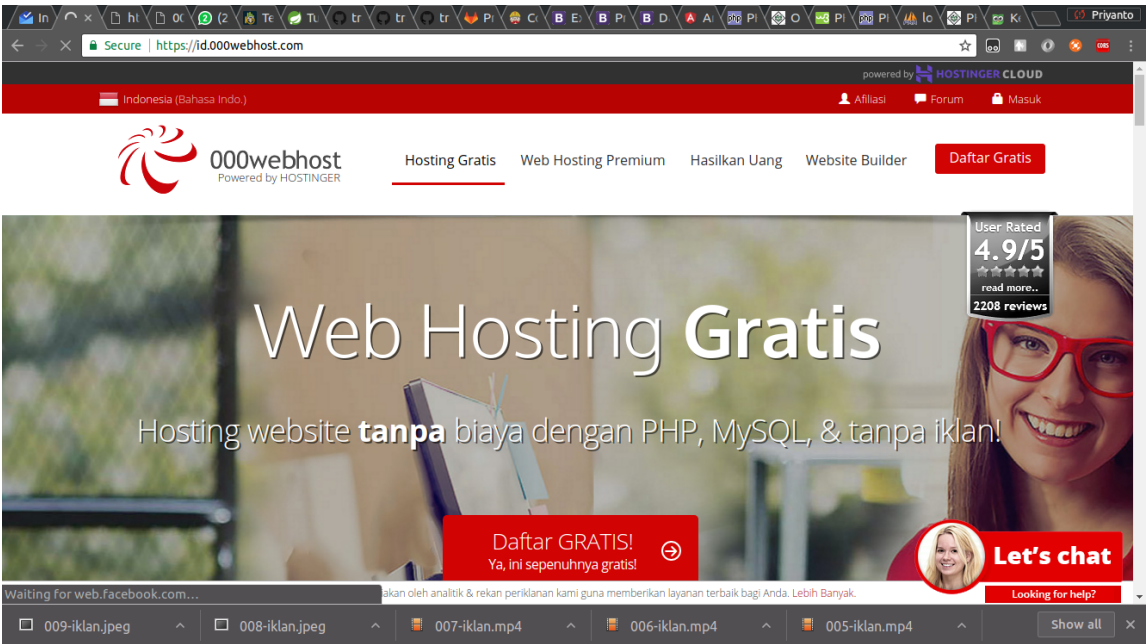


Gambar 1.31: Proses Instalasi Selesai

1.3.4 Registrasi 000webhost.com

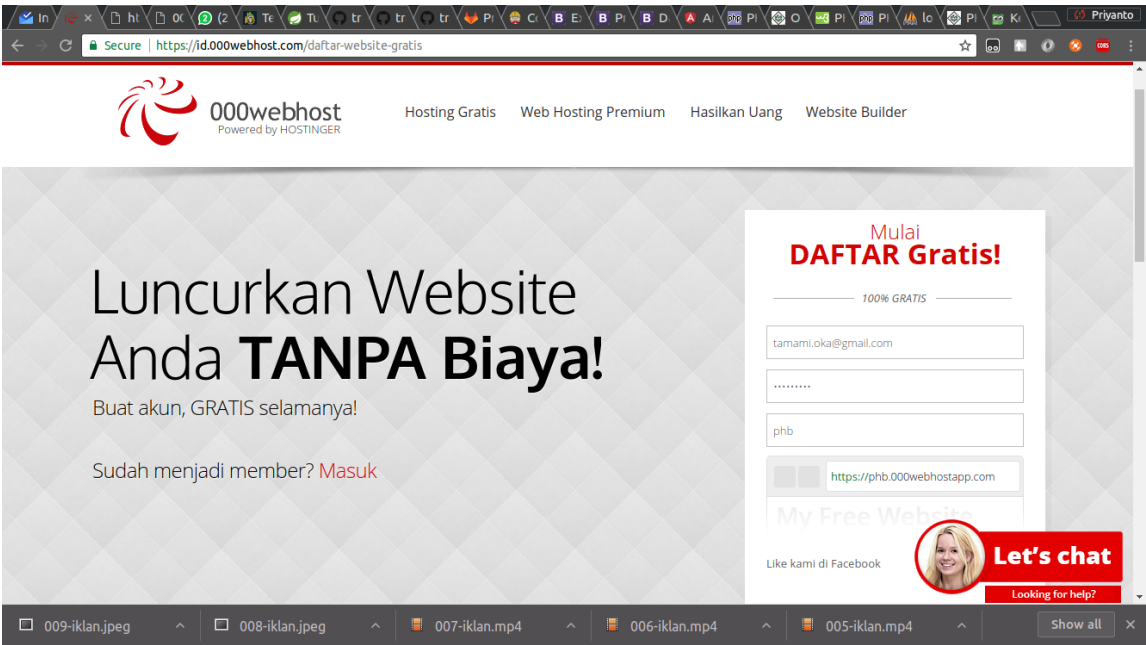
Registrasi pada *website* www.000webhost.com cukup mudah, langkahnya adalah sebagai berikut :

1. Mengunjungi *website* www.000webhost.com, apabila ingin pilihan bahasa Indonesia, kita dapat mengubahnya di bagian kiri atas, atau mengunjungi alamat id.000webhost.com, tampilan awal dari *website* ini seperti terlihat pada gambar 1.32 berikut :



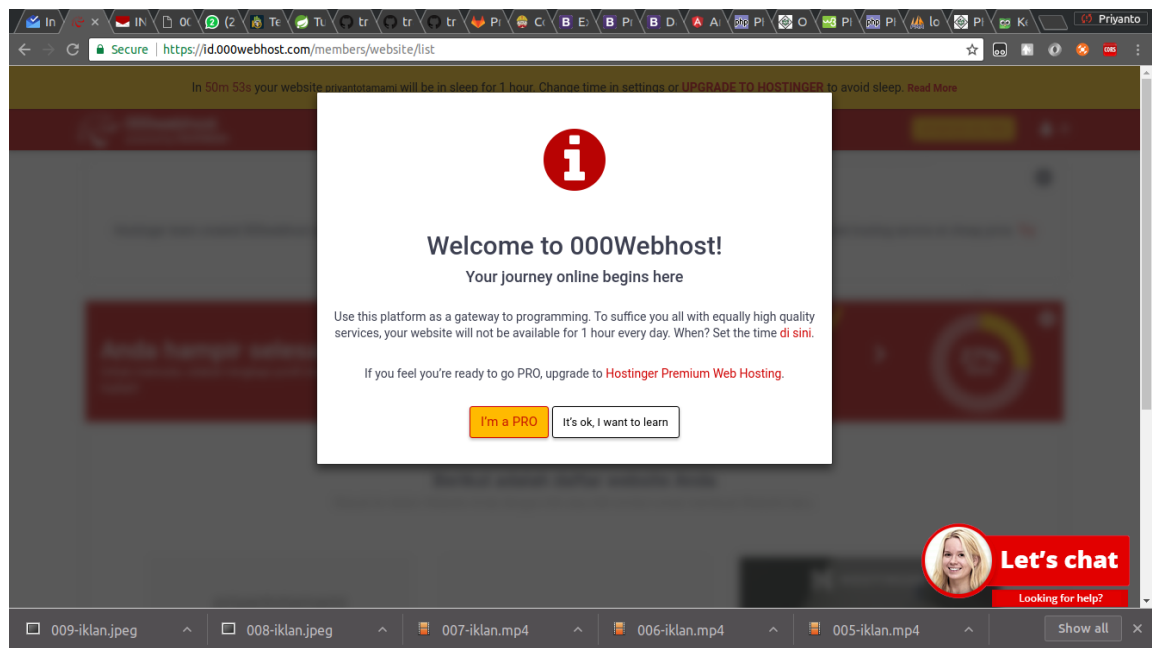
Gambar 1.32: Tampilan Awal Website id.000webhost.com

2. Setelah menekan tombol "Daftar Gratis", maka kemudian akan disajikan halaman seperti pada gambar 1.33, kita perlu mengisi alamat email, *password* untuk masuk ke halaman manajemen *website* yang kita bangun, serta alamat dari *website* yang kita inginkan. Karena sifatnya gratis, *url* yang disediakan pun mengikuti aturan dari penyedia *hosting*.



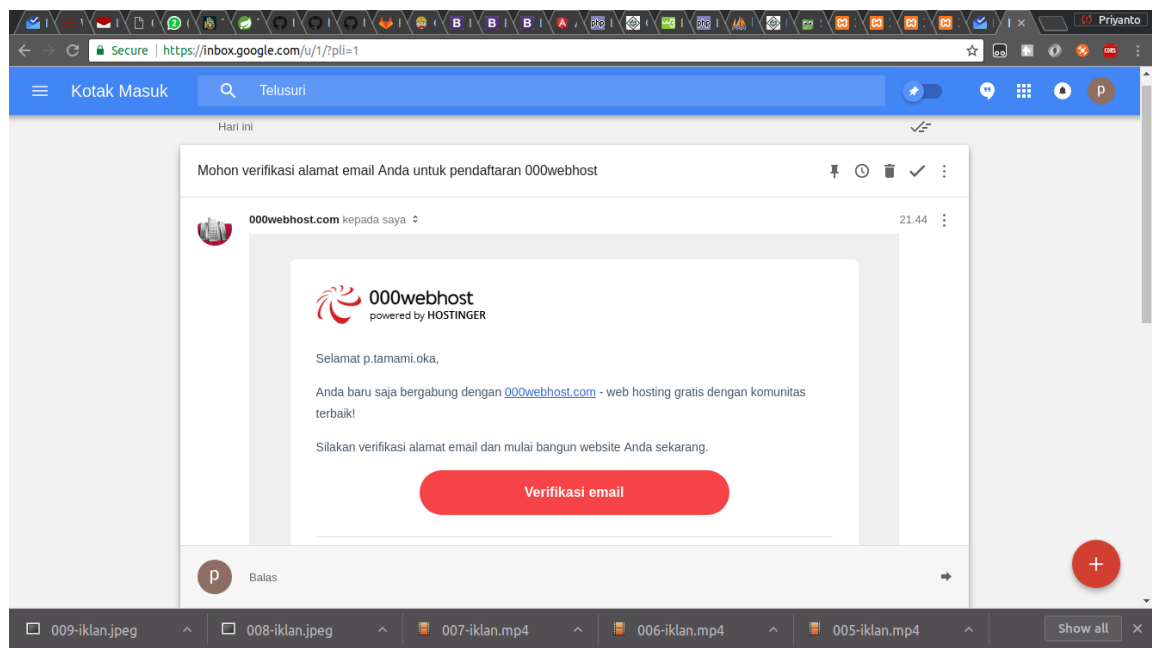
Gambar 1.33: Halaman Registrasi

3. Setelah menekan tombol pendaftaran, maka akan ditampilkan halaman selamat datang seperti pada gambar 1.34.



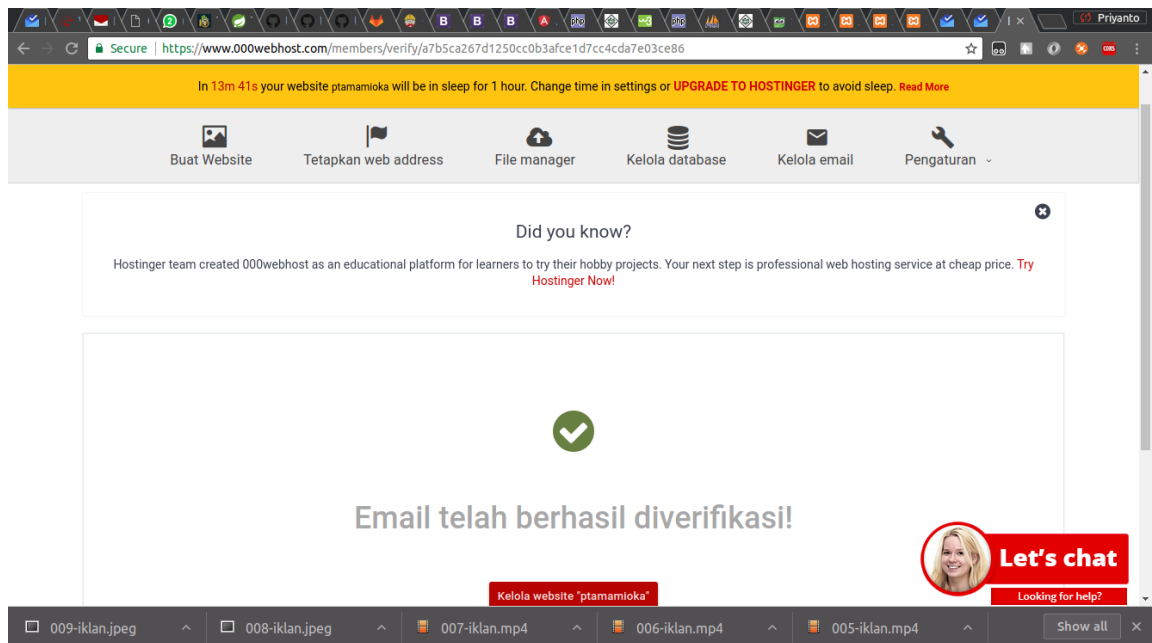
Gambar 1.34: Halaman Selamat Datang

4. Langkah berikutnya adalah melakukan verifikasi surel (*e-mail*) seperti pada gambar 1.35.



Gambar 1.35: Verifikasi Surel

5. Apabila verifikasi surel berhasil, maka akan tampil halaman seperti pada gambar 1.36.



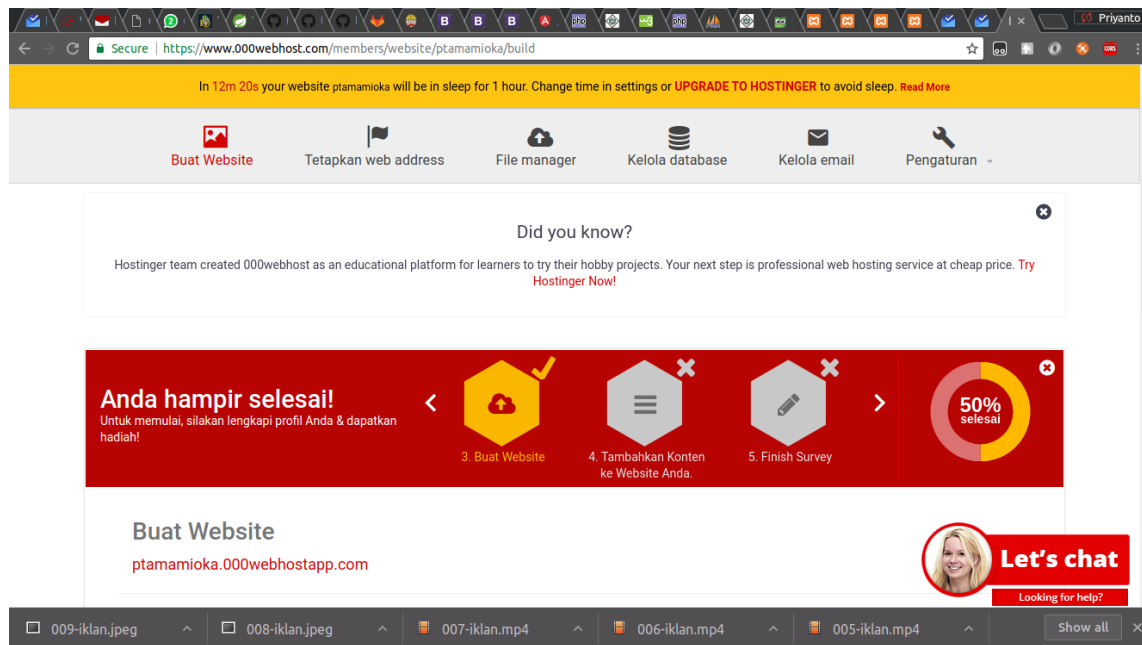
Gambar 1.36: Hasil verifikasi surel

Sampai sini, tahapan pendaftaran anggota telah berhasil kita lakukan.

1.3.5 Aplikasi Selamat Datang

Kita akan mencoba melakukan *publish* terhadap sebuah *file html* untuk membuktikan bahwa halaman *website* yang kita unggah ke id.000webhost.com dapat berhasil. Berikut adalah langkah-langkahnya :

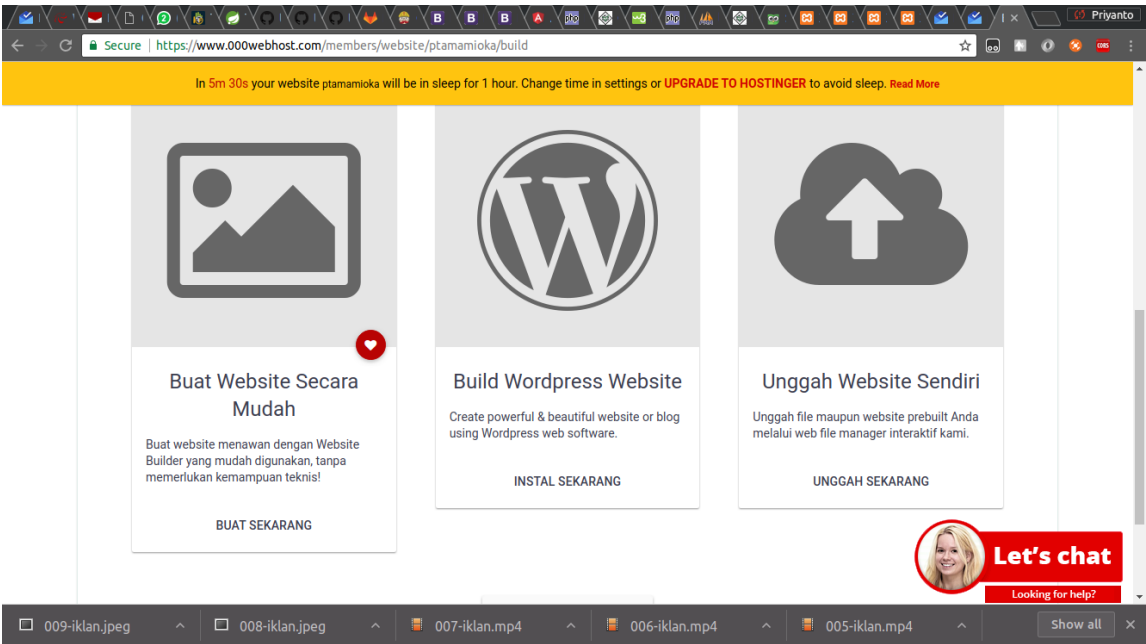
1. Langkah pertama adalah membuat sebuah halaman *website* dengan menekan tombol kiri atas sehingga tampil halaman seperti pada gambar 1.37.

Gambar 1.37: Membuat *website*

2. Selanjutnya kita membuat *file* dengan nama `index.html` terlebih dahulu, pembuatan *file* ini dapat kita lakukan dengan notepad, vim, atau Visual Studio Code yang telah kita *install*. Isi dari *file* ini adalah sebagai berikut :

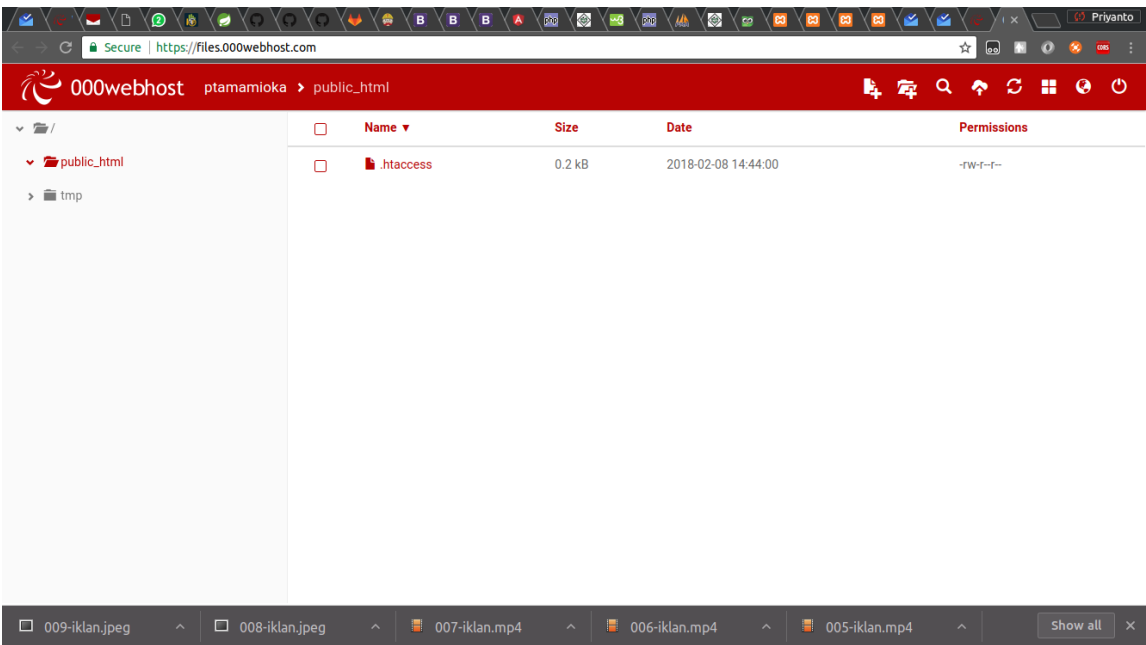
```
1 Hai , selamat datang .
```

3. Kemudian kita *scroll* ke bawah halaman id.000webhost.com, maka akan muncul tampilan seperti pada gambar 1.38. Karena kita telah membuat sebuah *file* `index.html`, maka kita memilih "Unggah Website Sendiri".



Gambar 1.38: Pilihan cara membuat *website*

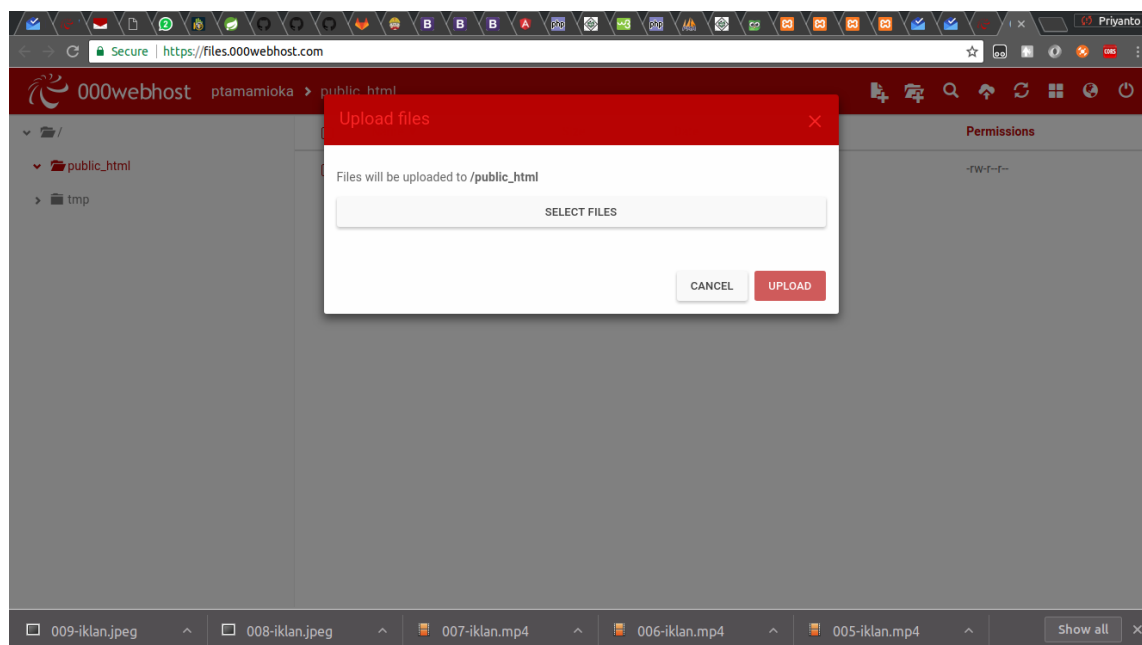
4. Selanjutnya kita akan ditunjukkan halaman *file manager* dimana nantinya di dalam *folder public_html* ini *project* kita ditempatkan. Tampilannya seperti pada gambar 1.39.



Gambar 1.39: Halaman *File Manager*

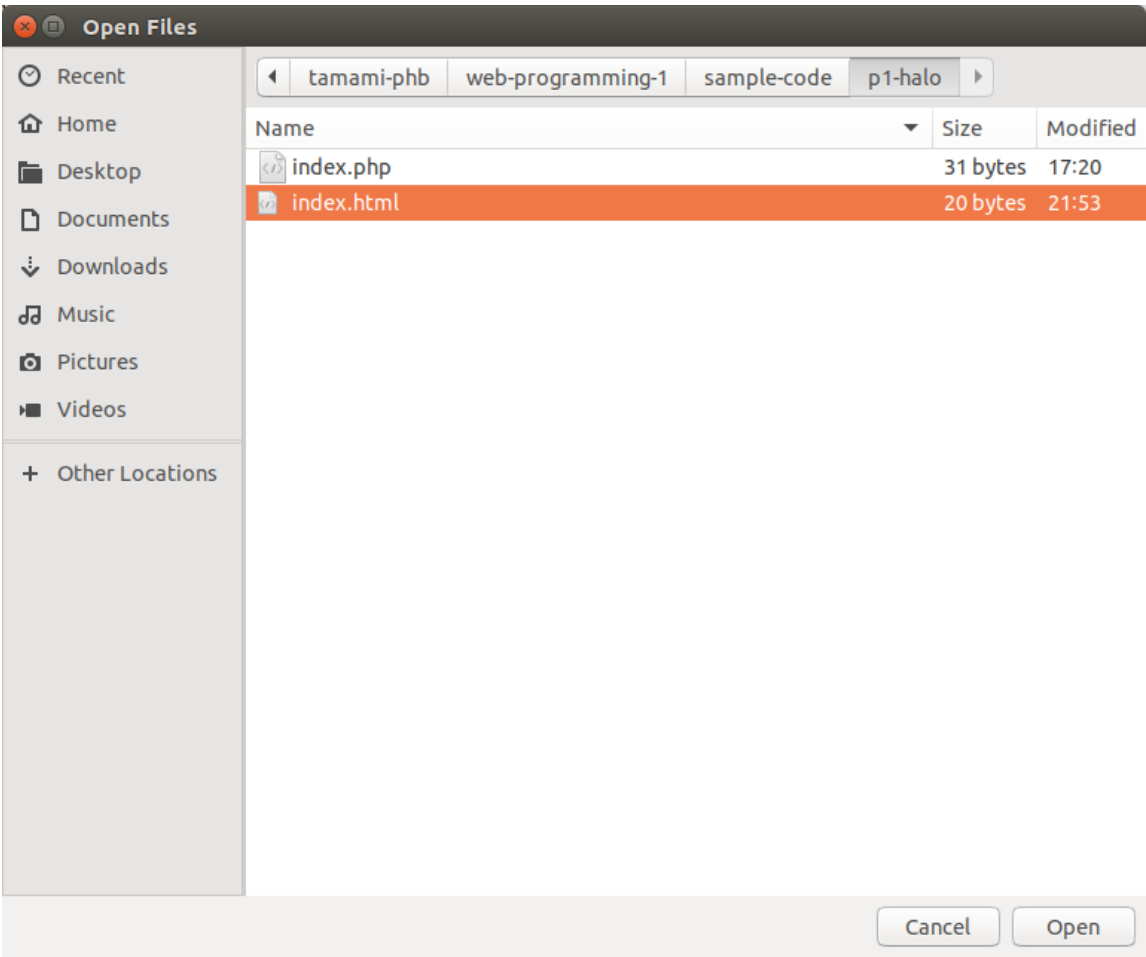
5. Pilihlah ikon dengan gambar awan dan tanda panah atas pada bagian atas kanan jendela *file manager* untuk mengunggah *file index.html* yang telah

kita buat sebelumnya. Nantinya akan muncul tampilan seperti pada gambar 1.40.



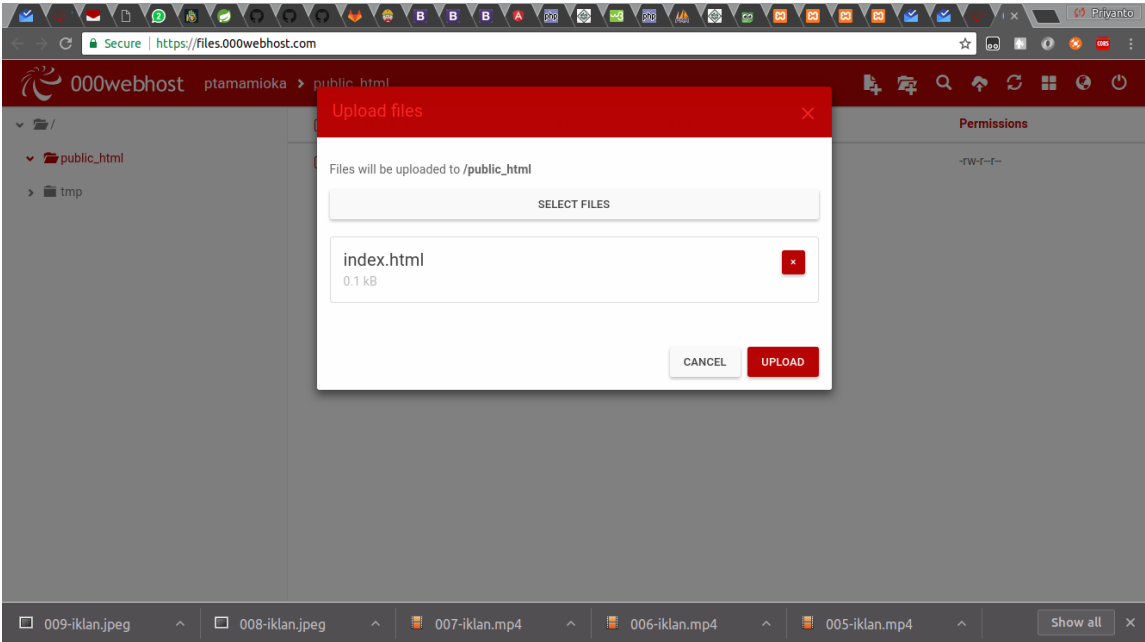
Gambar 1.40: Jendela Unggah *File*

6. Tekanlah tombol **SELECT FILES** yang berada di tengah sehingga muncul jendela pemilihan *file* seperti pada gambar 1.41, lalu memilih *file* dengan nama `index.html` yang akan kita unggah ke *server*.



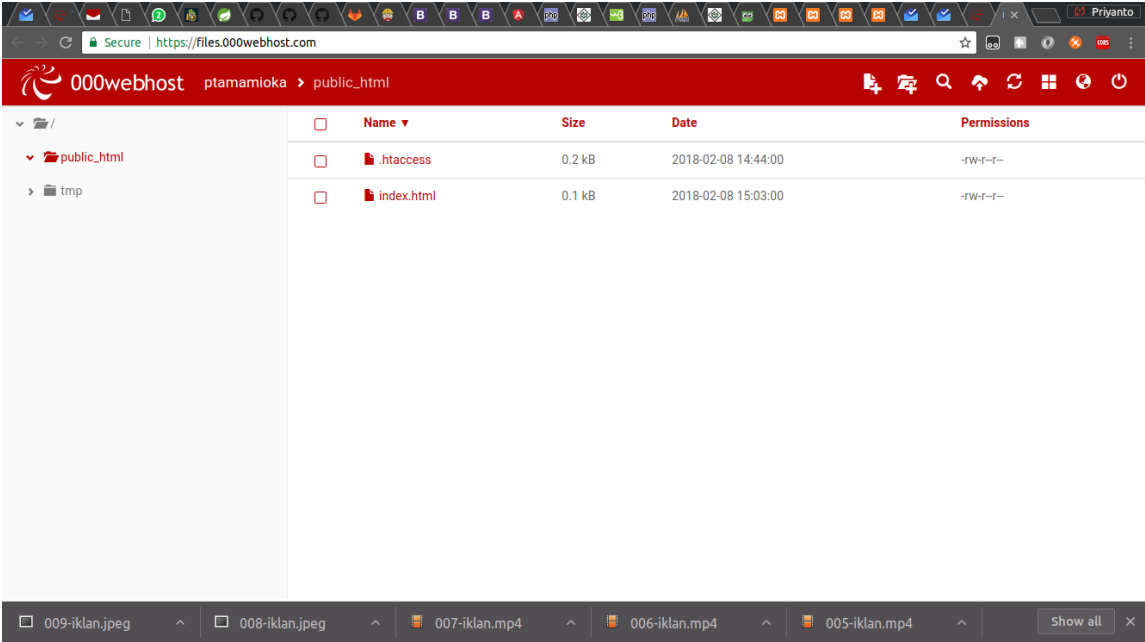
Gambar 1.41: Jendela Pemilihan *File*

- 7. Setelah *file* dipilih, maka jendela *Upload Files* akan menampilkan *file* yang terpilih untuk selanjutnya siap diunggah seperti pada gambar 1.42.



Gambar 1.42: Jendela *File* Terpilih

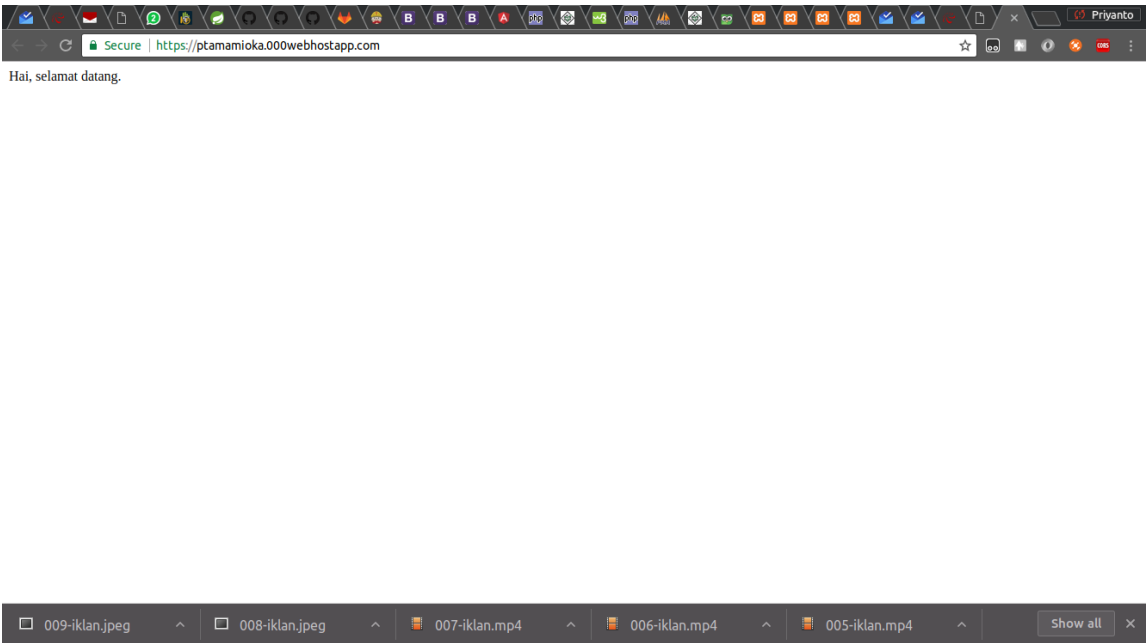
8. Setelah selesai terunggah, maka akan tampil *file* yang telah diunggah seperti pada gambar 1.43.



Gambar 1.43: Jendela *File Manager* Setelah File Terunggah

9. Lakukan akses ke halaman yang telah disediakan oleh www.000webhost.com yang hasilnya seperti terlihat pada gambar 1.44. Karena ini layanan gratis, jadi jeda antara waktu *file* telah terunggah dengan hasil *website* memakan

waktu yang bervariasi. Apabila protokol `https` tidak berhasil menampilkan halaman yang telah kita buat, cobalah menggunakan protokol `http`.

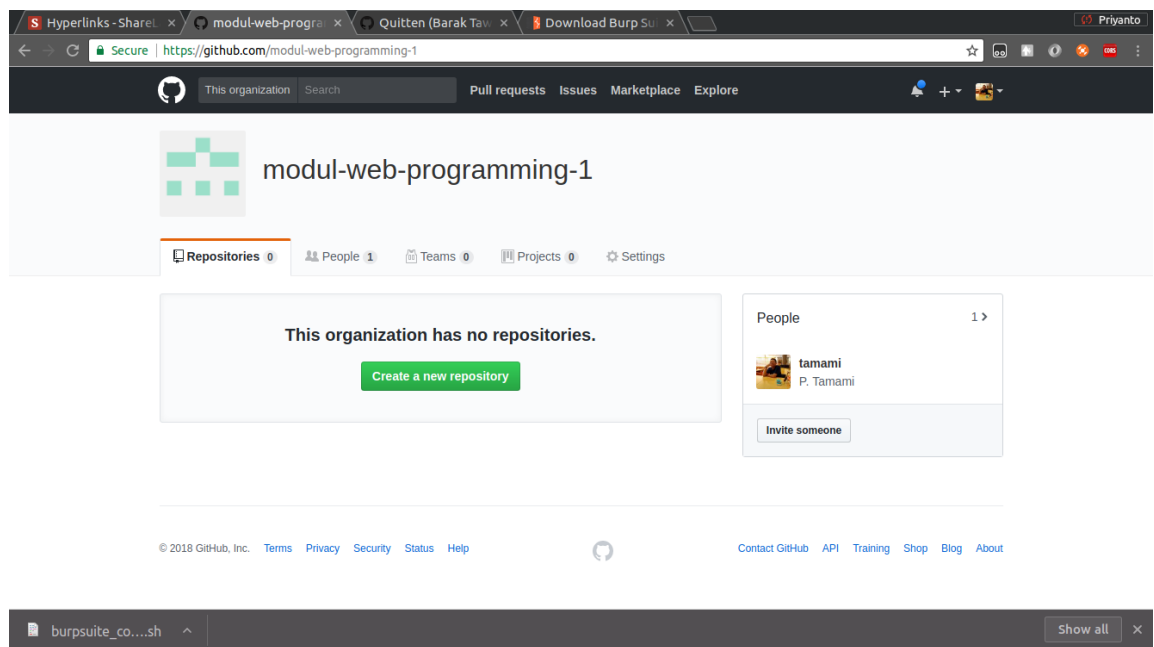


Gambar 1.44: Hasil Website

1.3.6 Unggah ke Github

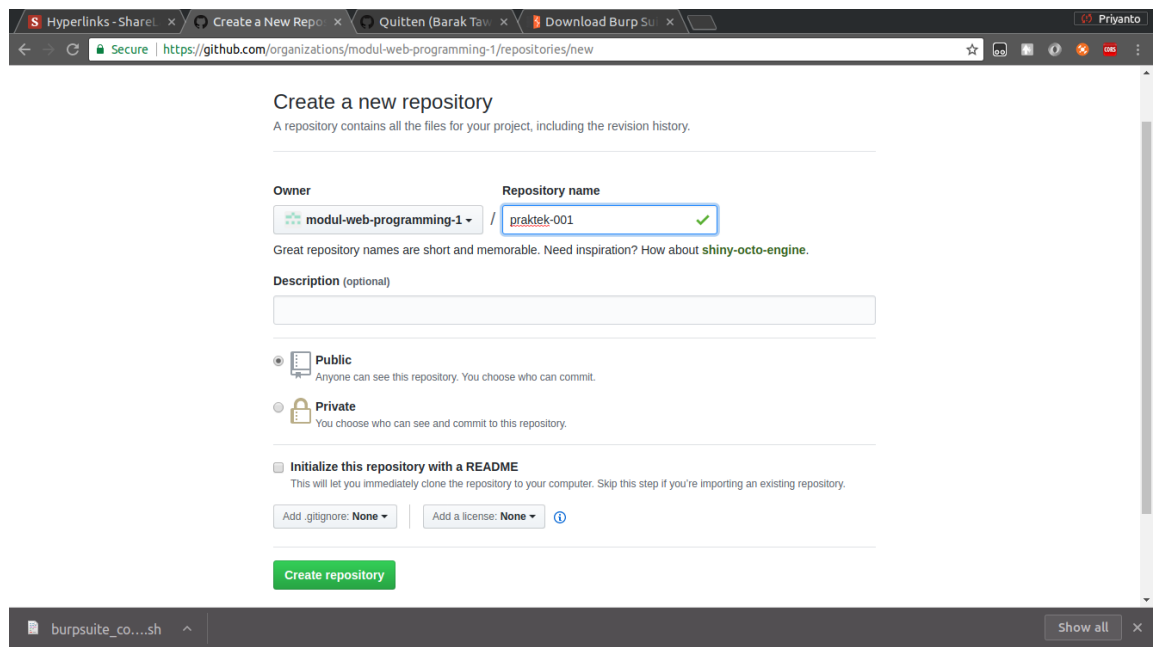
Setelah kode awal berhasil kita unggah ke id.000webhost.com, kita akan unggah pula kode yang kita buat ke [Github](https://github.com) untuk keperluan *versioning* kode. Langkahnya cukup mudah, yaitu :

1. Membuat *repository* di Github dengan menekan tombol hijau bertuliskan "Create a new repository" seperti pada gambar [1.45](#).



Gambar 1.45: Halaman Awal Github

2. Kemudian mengisikan nama repositorinya pada kolom yang tersedia, lalu tekan tombol **Enter** atau klik tombol "Create Repository" seperti pada gambar 1.46 yang nantinya.



Gambar 1.46: Membuat *Repository*

3. Membuka **terminal** atau **command prompt** atau **console** untuk melakukan *init git* dengan lokasi direktorinya adalah tempat *file index.html* berada dengan

kode berikut :

```
1 > git init
```

4. Melakukan *staging file* dengan perintah berikut :

```
1 > git add .
```

5. Melakukan *commit* terhadap *staging file* dengan perintah berikut :

```
1 > git commit -m "init commit"
```

Opsi perintah `-m "init commit"` sebetulnya wajib, jadi setiap kita melakukan *commit*, kita diminta untuk memberikan keterangan / komentar di tiap *commit* untuk memudahkan kita mencari tahap perubahan tertentu.

6. Mendaftarkan alamat penyedia layanan repositori `git` dengan perintah berikut :

```
1 > git remote add github https://github.com/modul-web-programming-1/praktek-001.git
```

`github` di atas adalah nama alias, jadi boleh diganti apapun, sedangkan *url* setelahnya adalah alamat yang diberikan oleh Github setelah kita membuat sebuah *repository*.

7. Melakukan unggah kode ke Github dengan perintah berikut :

```
1 > git push -u github master
```

`github` adalah alias yang sebelumnya kita buat, sedangkan `master` adalah *branch* utama dari `git`.

1.4 Kesimpulan

Bahwa membangun sebuah aplikasi *web* menggunakan PHP yang bersifat dinamis diperlukan beberapa perangkat / aplikasi pendukung.

Dengan fasilitas gratis pun kita masih dapat melakukan *publish project* aplikasi web yang dibangun dengan PHP kita ke internet sehingga dapat diakses oleh semua orang.

1.5 Tugas

Membuat halaman selamat datang dalam bentuk HTML dan di-*publish* ke internet.

Bab 2

Operator, Variabel, dan Tipe Data

2.1 Tujuan

Mahasiswa mampu mengimplementasikan alur sederhana untuk menggunakan operator, variabel, dan tipe data pada PHP.

2.2 Pengantar

Variabel sebetulnya bisa diibaratkan sebagai wadah yang akan diisi oleh data. Sedangkan *tipe data* bisa dianalogikan seperti jenis data yang berada pada suatu variabel. *Operator* sendiri adalah perintah untuk melakukan operasi tertentu pada satu atau lebih variabel / data.

Untuk variabel sebenarnya akan memiliki *scope* seperti berikut :

- **local**, variabel akan bersifat lokal, tidak dapat digunakan di luar deklarasinya.
- **global**, variabel akan bersifat global, dan dapat digunakan di luar deklarasinya.
- **static**, variabel jenis ini tidak akan dihapus dari memori ketika sudah dieksekusi.

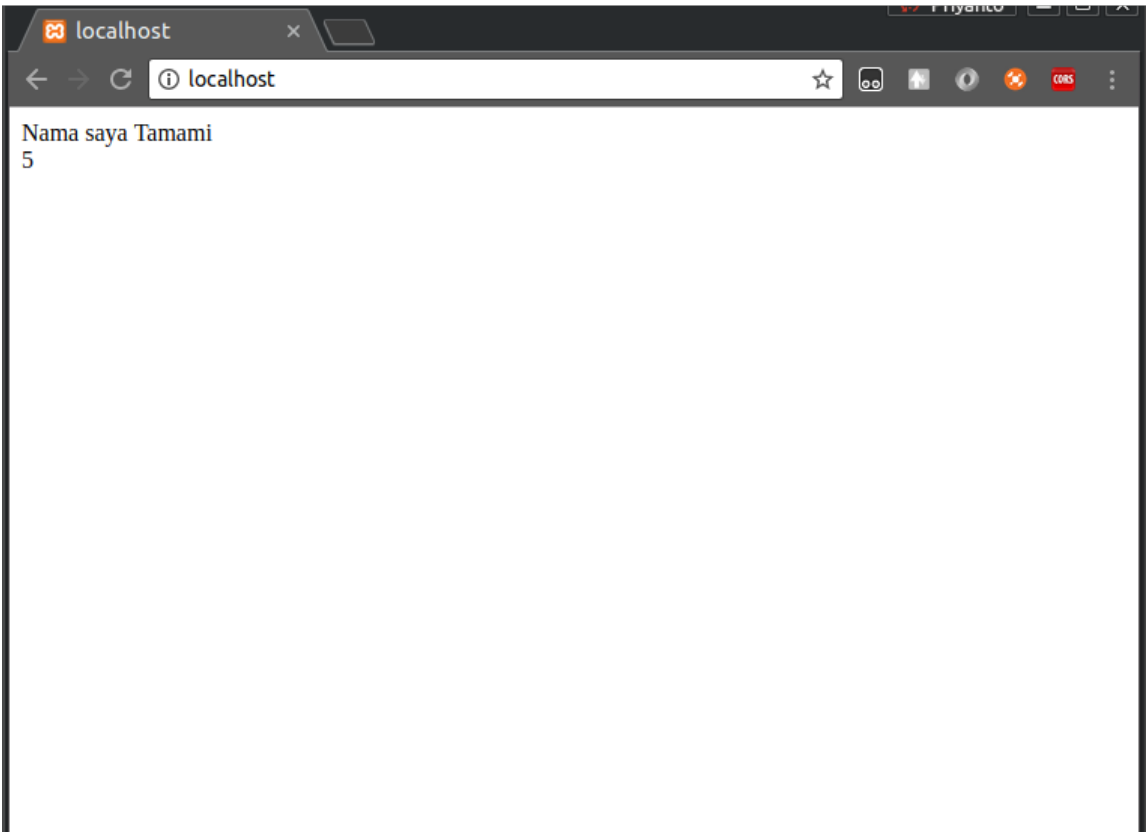
2.3 Praktek

2.3.1 Variabel

Untuk menggunakan variabel, kita cukup memberikan tanda dollar (\$) di awal nama variabel. Contoh kodenya adalah sebagai berikut :

```
1 <?php
2 $teks = "Nama saya Tamami";
3 $angka = 5;
4
5 echo $teks;
6 echo "<br>";
7 echo $angka;
8 ?>
```

Hasil keluaran dari kode di atas adalah seperti pada gambar 2.1 berikut :



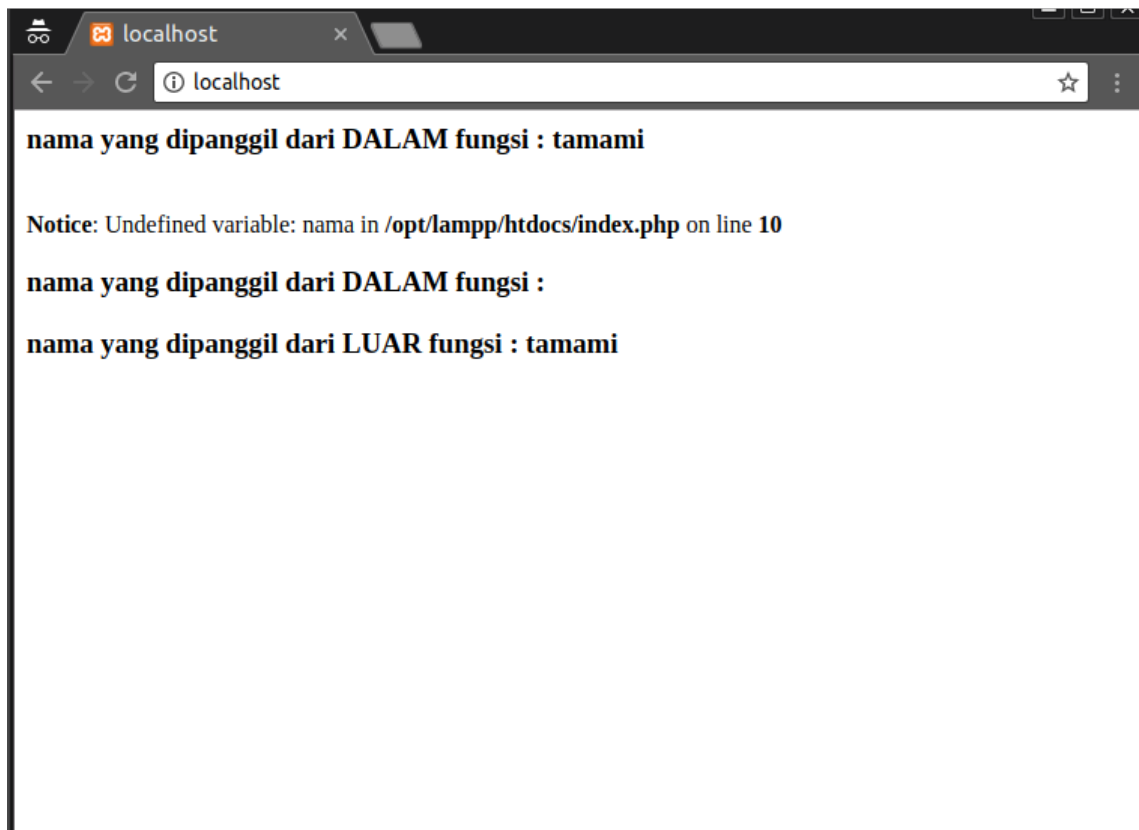
Gambar 2.1: Hasil Keluaran Variabel

Variabel *Global*

Contoh kode untuk menjelaskan bagaimana sifat variabel *global* adalah sebagai berikut :

```
1 <?php
2 $nama = "tamami";
3
4 function cetakNama1() {
5     global $nama;
6     echo "<h3>nama yang dipanggil dari DALAM fungsi : $nama</h3>";
7 }
8
9 function cetakNama2() {
10     echo "<h3>nama yang dipanggil dari DALAM fungsi : $nama</h3>";
11 }
12
13 cetakNama1();
14 cetakNama2();
15
16 echo "<h3>nama yang dipanggil dari LUAR fungsi : $nama</h3>";
17 ?>
```


Hasil dari kode di atas adalah seperti tampilan pada gambar ?? berikut :



Gambar 2.2: Hasil Keluaran Variabel Global

Artinya, variabel *global* dapat digunakan di dalam fungsi maupun di luar fungsi, apabila penggunaannya di dalam sebuah fungsi, maka perlu dideklarasikan terlebih dahulu di dalam fungsi tersebut dengan kata kunci `global`.

Variabel *Local*

Untuk penggunaan variabel *local*, contoh kodenya adalah sebagai berikut :

```
1 <?php
2
3 function cetakNama() {
4     $nama = "tamami";
5     echo "<h3>nama yang dipanggil dari DALAM fungsi : $nama</h3>";
6 }
7
8 cetakNama();
9
10 echo "<h3>nama yang dipanggil dari LUAR fungsi : $nama</h3>";
11 ?>
```

Hasil keluaran dari kode program tersebut adalah seperti pada gambar 2.3 berikut :



Gambar 2.3: Hasil Keluaran Variabel *Local*

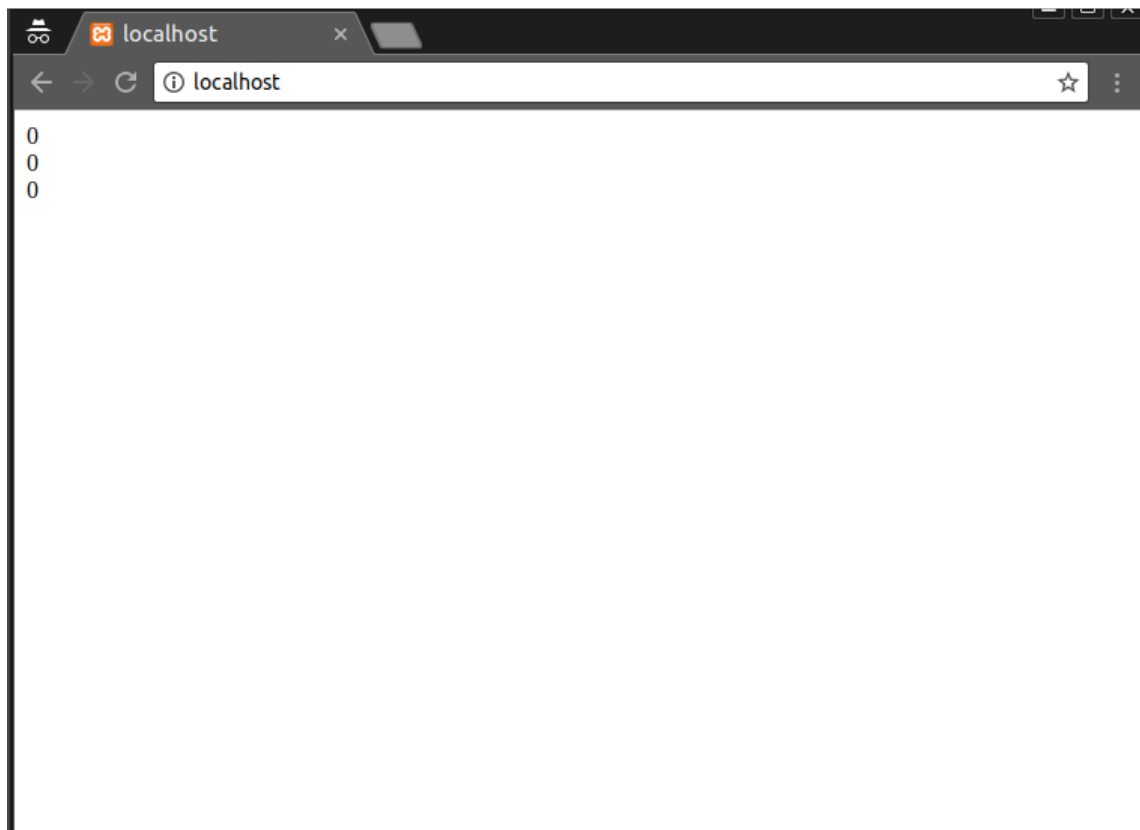
Artinya, variabel `$nama` yang dideklarasikan di dalam sebuah fungsi hanya dapat digunakan di dalam fungsi yang bersangkutan.

Variabel *Static*

Karena sifat dari variabel di bahasa pemrograman PHP ini akan dihapus dari memori ketika telah dieksekusi, maka bila kita ingin menggunakan variabel ini lebih lama, maka kita harus menjadikannya `static`, berikut adalah contoh kode dimana jenis variabel tidak `static` :

```
1 <?php
2 function hitung() {
3     $nilai = 0;
4     echo $nilai;
5     $nilai += 5;
6 }
7
8 hitung();
9 echo "<br>";
10 hitung();
11 echo "<br>";
12 hiutung();
13 ?>
```

Hasil dari kode di atas adalah seperti gambar 2.4 berikut :



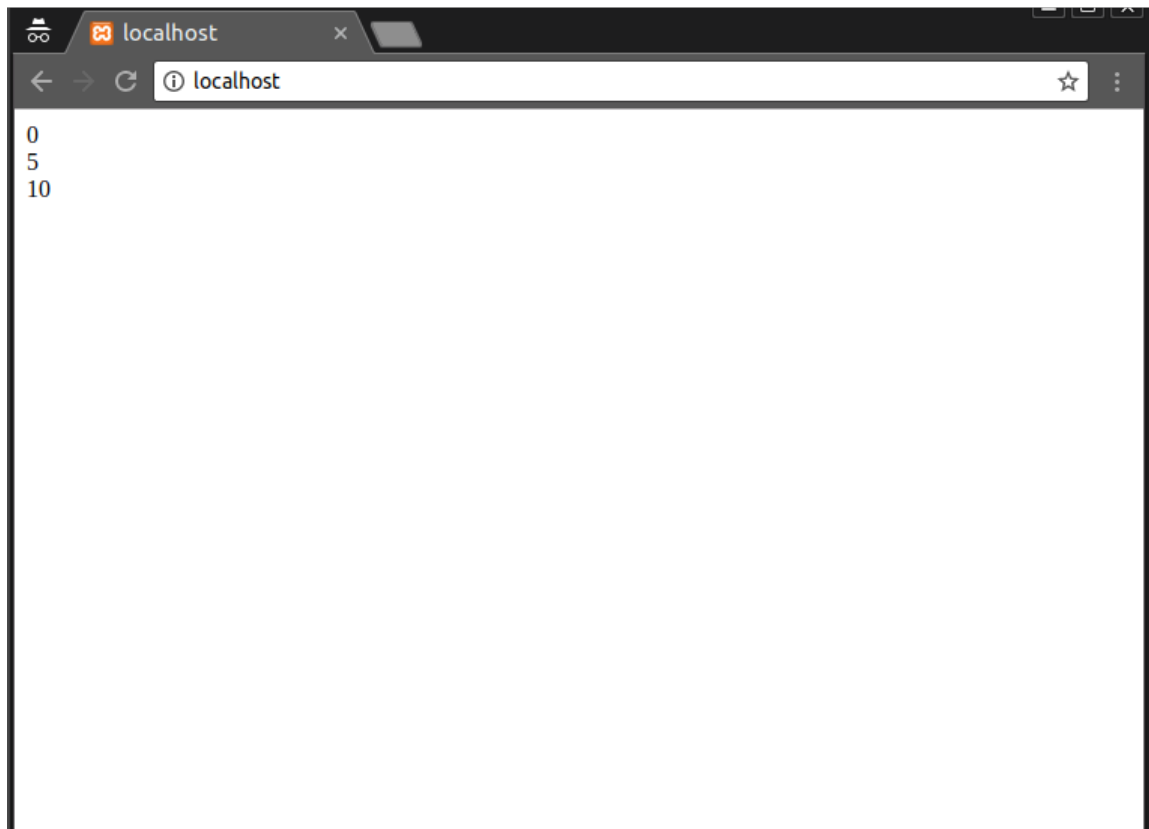
Gambar 2.4: Hasil Keluaran Variabel Tanpa `static`

Tampak bahwa variabel `$nilai` akan selalu bernilai 0 (nol) karena setiap kali eksekusi `$nilai += 5` maka begitu keluar dari fungsi `hitung()`, isi variabel akan kembali menjadi 0 (nol).

Apabila kita jadikan variabel `$nilai`, maka kodenya menjadi seperti ini :

```
1 <?php
2 function hitung() {
3     static $nilai = 0;
4     echo $nilai;
5     $nilai += 5;
6 }
7
8 hitung();
9 echo "<br>";
10 hitung();
11 echo "<br>";
12 hitung();
13 ?>
```

Hasil keluaran dari kode tersebut adalah seperti pada gambar 2.5 berikut :



Gambar 2.5: Hasil Keluaran Variabel Dengan `static`

Hasil dari keluaran pada gambar 2.5 tampak bahwa variabel `$nilai` akan melakukan eksekusi dan menyimpan hasil tanpa menghapusnya dari memori.

2.3.2 Tipe Data

Variabel pada dasarnya menyimpan sebuah data dengan tipe data tertentu, masing-masing tipe data dapat melakukan hal yang berbeda-beda. Tipe data yang didukung oleh PHP ini adalah sebagai berikut :

- String, tipe data ini untuk menyimpan teks
- Integer, tipe data ini untuk menyimpan angka bulat, tanpa desimal
- Float / Double, tipe data ini untuk menyimpan angka yang memiliki bilangan desimal
- Boolean, tipe data ini hanya menampung nilai `true` atau `false`
- Array, tipe data ini akan menampung baris data baik dengan tipe data yang sama, maupun tipe data yang berbeda.
- Object, tipe data ini akan menampung objek yang kita buat sendiri

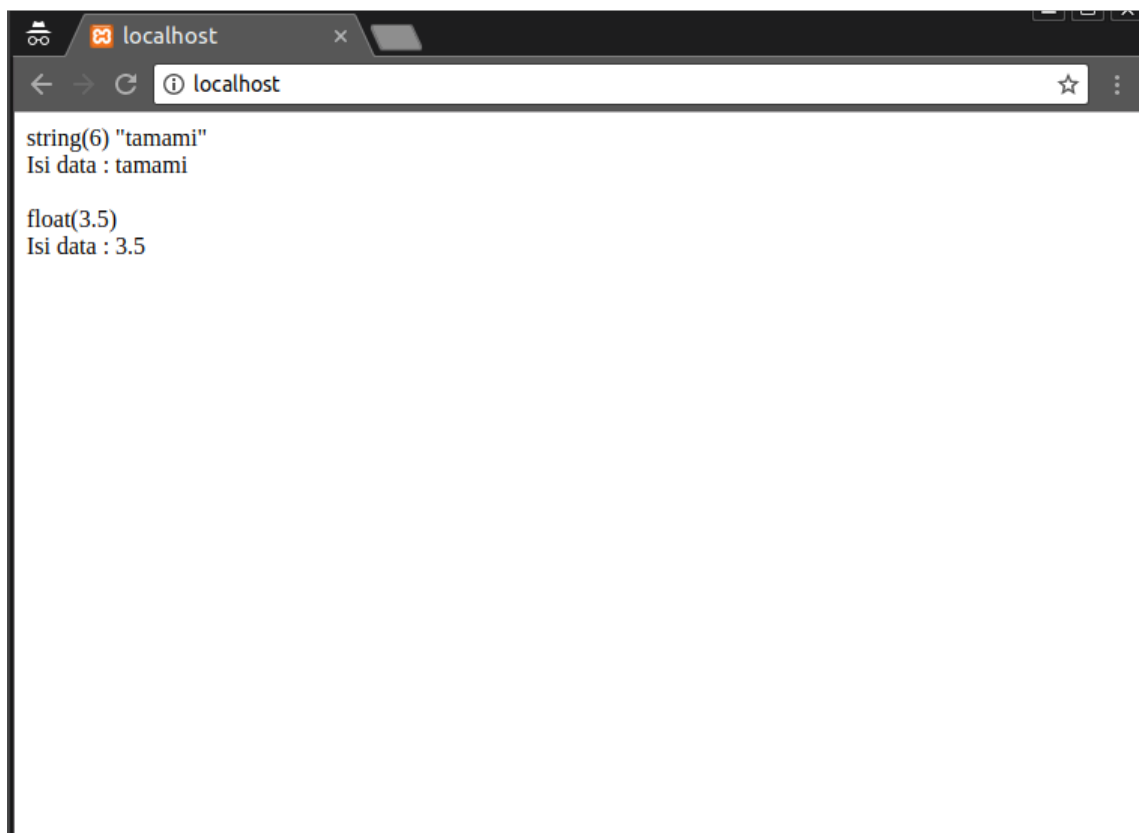
- NULL, tipe data ini artinya tidak memiliki data sama sekali.

Tipe data di PHP tidak dideklarasikan secara langsung, melainkan pada saat pengisian data ke sebuah variabel, pada saat itu juga tipe data akan ditentukan terhadap variabel tersebut.

Perhatikan kode berikut ini :

```
1 <?php
2 $data = "tamami";
3 var_dump($data);
4 echo "<br>";
5 echo "Isi data : $data";
6 echo "<br>";
7 echo "<br>";
8 $data = 3.5;
9 var_dump($data);
10 echo "<br>";
11 echo "Isi data : $data";
12 ?>
```

Kode tersebut akan menghasilkan tampilan seperti pada gambar 2.6 berikut :



Gambar 2.6: Melihat Tipe Data

Terlihat pada saat kita mengisi variabel `$data` dengan teks, maka tipe data

langsung terisi menjadi `string`, kemudian saat kita mengganti isinya menjadi 3.5 maka tipe data ikut berubah menjadi `float`.

2.3.3 Operator

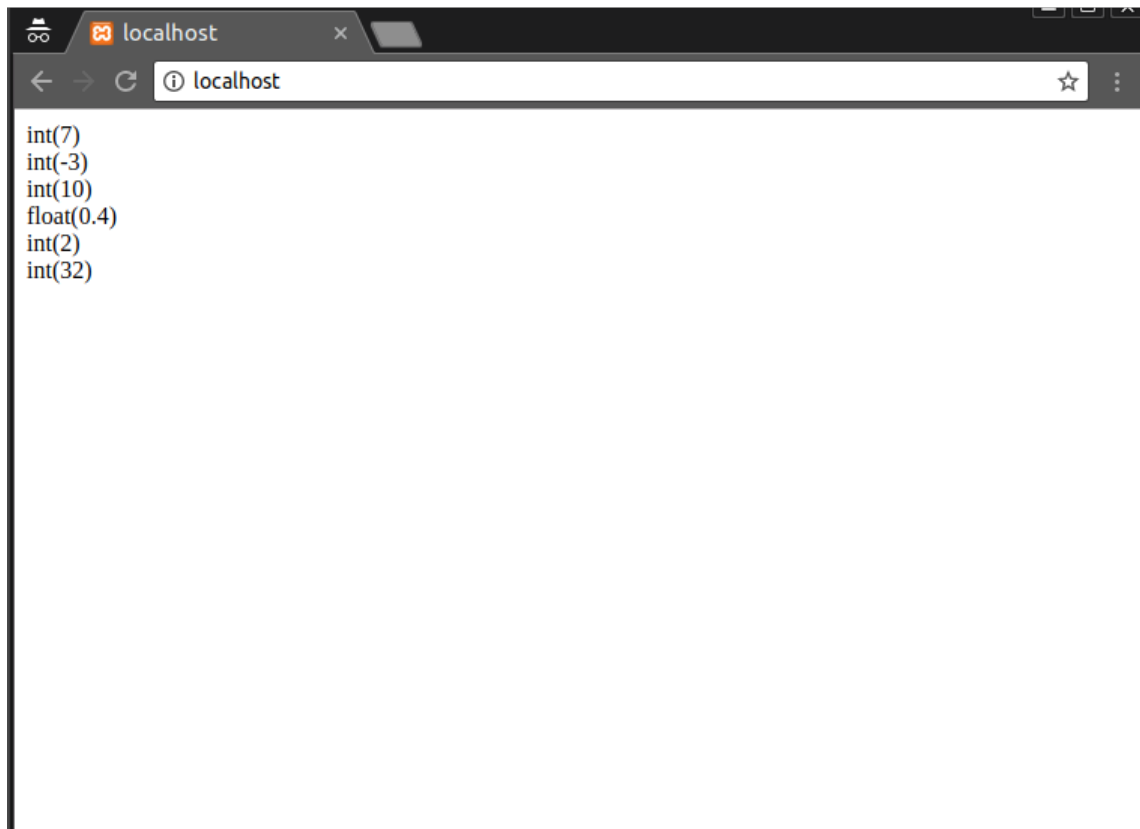
Operator pada PHP akan dibagi menjadi beberapa bagian seperti berikut :

Arithmetic Operators

Operator ini berlaku untuk melakukan operasi aritmatika dasar di PHP, contoh kodenya adalah sebagai berikut :

```
1 <?php
2 $x = 2;
3 $y = 5;
4
5 $result = $x + $y; // operasi penjumlahan
6 var_dump($result);
7 echo "<br>";
8
9 $result = $x - $y; // operasi pengurangan
10 var_dump($result);
11 echo "<br>";
12
13 $result = $x * $y; // operasi perkalian
14 var_dump($result);
15 echo "<br>";
16
17 $result = $x / $y; // operasi pembagian
18 var_dump($result);
19 echo "<br>";
20
21 $result = $x % $y; // operasi sisa bagi
22 var_dump($result);
23 echo "<br>";
24
25 $result = $x ** $y; // operasi pangkat
26 var_dump($result);
27 echo "<br>";
28 ?>
```

Hasil dari kode di atas adalah seperti pada gambar [2.7](#) berikut :



Gambar 2.7: Hasil Keluaran Operasi Aritmatika

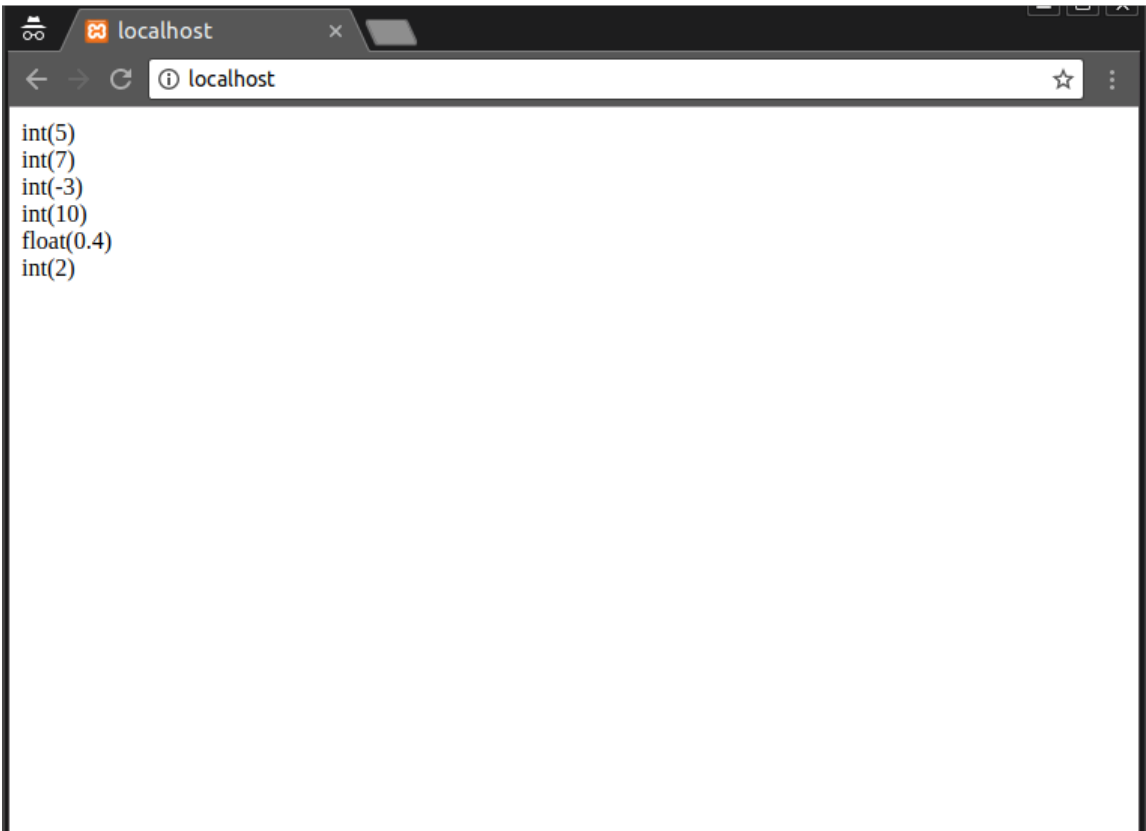
Assignment Operators

Untuk operator pengisian ditandai dengan tanda = (sama dengan). Nantinya variabel yang ada di sebelah kiri tanda = (sama dengan) akan terisi dengan ekspresi yang ada di sebelah kanan tanda = (sama dengan). Contoh kodenya adalah sebagai berikut :

```
1 <?php
2 $x = 2;
3 $y = 5;
4
5 $x = $y;          // operasi pengisian , dimana variabel x diisikan nilai
                    // dari variabel y
6 var_dump($x);
7 echo "<br>";
8 $x = 2;
9
10 $x += $y;         // operasi penambahan , sama seperti x = x + y;
11 var_dump($x);
12 echo "<br>";
13 $x = 2;
14
15 $x -= $y;         // operasi pengurangan , sama seperti x = x - y;
16 var_dump($x);
17 echo "<br>";
```

```
18 $x = 2;
19
20 $x *= $y;           // operasi perkalian , sama seperti x = x * y;
21 var_dump($x);
22 echo "<br>";
23 $x = 2;
24
25 $x /= $y;           // operasi pembagian , sama seperti x = x / y;
26 var_dump($x);
27 echo "<br>";
28 $x = 2;
29
30 $x %= $y;           // operasi sisa bagi , sama seperti x = x % y;
31 var_dump($x);
32 echo "<br>";
33 ?>
```

Hasil keluaran dari kode di atas adalah seperti pada gambar 2.8 berikut :



Gambar 2.8: Hasil Keluaran Dari Operator Penugasan

Comparison Operators

Operator ini digunakan untuk membandingkan dua data baik angka maupun teks, berikut adalah operator yang dapat digunakan disertai dengan contohnya :

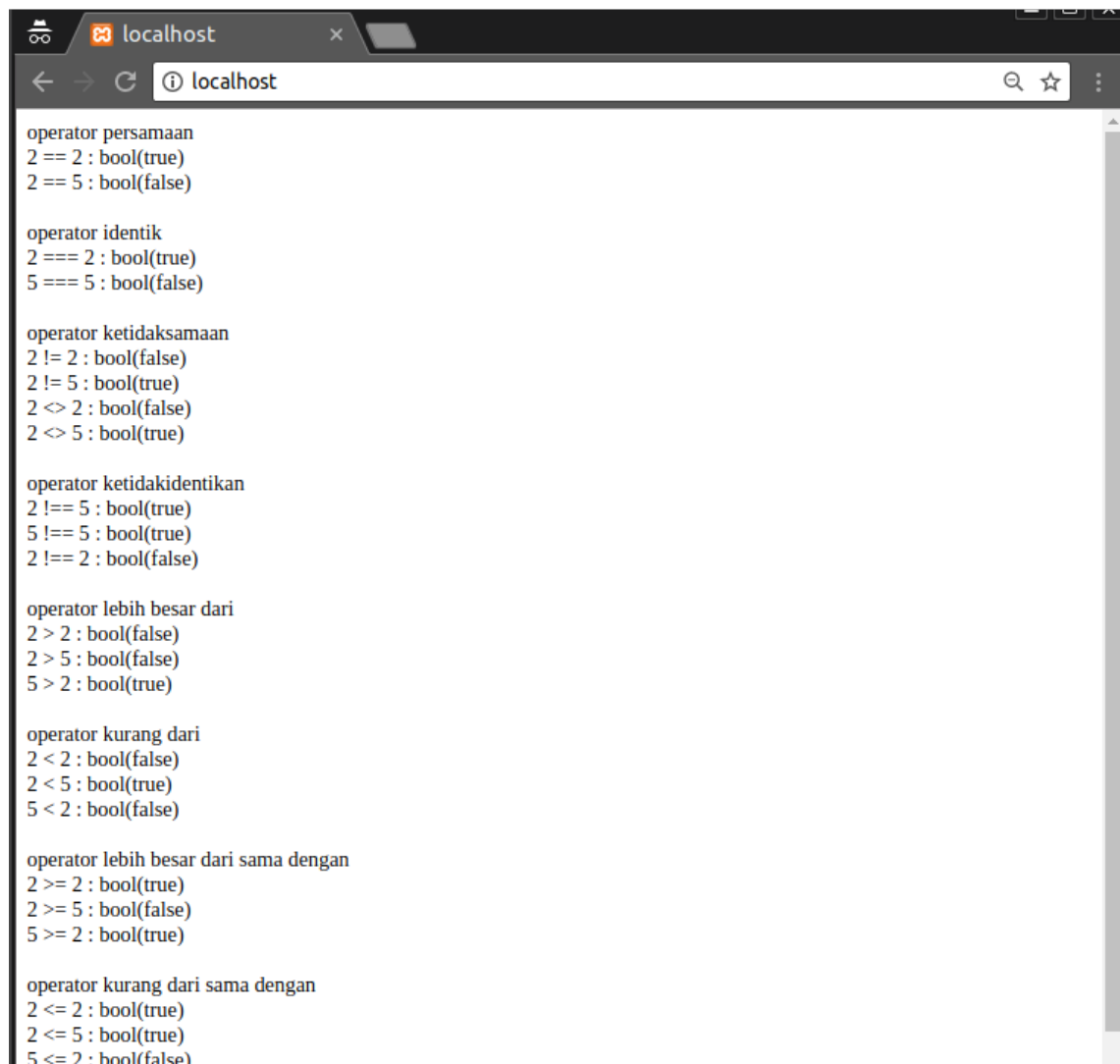

```
1 <?php
2 $w = 2;
3 $x = 2;
4 $y = 5;
5 $z = "5";
6
7 echo "operator persamaan<br>";
8 echo "$w == $x : ";
9 var_dump($w == $x);
10 echo "<br>";
11 echo "$w == $y : ";
12 var_dump($w == $y);
13 echo "<br><br>";
14
15 echo "operator identik<br>";
16 echo "$w === $x : ";
17 var_dump($w === $x);
18 echo "<br>";
19 echo "$y === $z : ";
20 var_dump($y === $z);
21 echo "<br><br>";
22
23 echo "operator ketidaksamaan<br>";
24 echo "$w != $x : ";
25 var_dump($w != $x);
26 echo "<br>";
27 echo "$w != $y : ";
28 var_dump($w != $y);
29 echo "<br>";
30 echo "$w < $x : ";
31 var_dump($w < $x);
32 echo "<br>";
33 echo "$w < $y : ";
34 var_dump($w < $y);
35 echo "<br><br>";
36
37 echo "operator ketidakidentikan<br>";
38 echo "$w !== $y : ";
39 var_dump($w !== $y);
40 echo "<br>";
41 echo "$y !== $z : ";
42 var_dump($y !== $z);
43 echo "<br>";
44 echo "$w !== $x : ";
45 var_dump($w !== $x);
46 echo "<br><br>";
47
48 echo "operator lebih besar dari<br>";
49 echo "$w > $x : ";
50 var_dump($w > $x);
51 echo "<br>";
52 echo "$w > $y : ";
53 var_dump($w > $y);
54 echo "<br>";
```

```

55 echo "$y > $w : ";
56 var_dump($y > $w);
57 echo "<br><br>";
58
59 echo "operator kurang dari<br>";
60 echo "$w < $x : ";
61 var_dump($w < $x);
62 echo "<br>";
63 echo "$w < $y : ";
64 var_dump($w < $y);
65 echo "<br>";
66 echo "$y < $w : ";
67 var_dump($y < $w);
68 echo "<br><br>";
69
70 echo "operator lebih besar dari sama dengan<br>";
71 echo "$w >= $x : ";
72 var_dump($w >= $x);
73 echo "<br>";
74 echo "$w >= $y : ";
75 var_dump($w >= $y);
76 echo "<br>";
77 echo "$y >= $w : ";
78 var_dump($y >= $w);
79 echo "<br><br>";
80
81 echo "operator kurang dari sama dengan<br>";
82 echo "$w <= $x : ";
83 var_dump($w <= $x);
84 echo "<br>";
85 echo "$w <= $y : ";
86 var_dump($w <= $y);
87 echo "<br>";
88 echo "$y <= $w : ";
89 var_dump($y <= $w);
90 echo "<br><br>";
91 ?>

```

Hasil keluaran dari kode di atas adalah seperti pada gambar [2.9](#) berikut ini :



Gambar 2.9: Hasil Kode Operator Perbandingan

Terlihat bahwa operator ini akan membandingkan dua nilai yang ada, yang perlu diperhatikan adalah bahwa operator persamaan dengan operator identik adalah bahwa operator identik akan memeriksa persamaannya sampai ke jenis tipe data. Apabila tipe datanya berbeda, maka akan dianggap tidak identik.

Sedangkan untuk operator ketidakidentikan, maka operator akan memeriksa, apabila datanya memang berbeda, maka akan memberikan nilai **true**, kemudian apabila datanya sama, namun tipe datanya berbeda, maka akan memberikan nilai **true** pula.

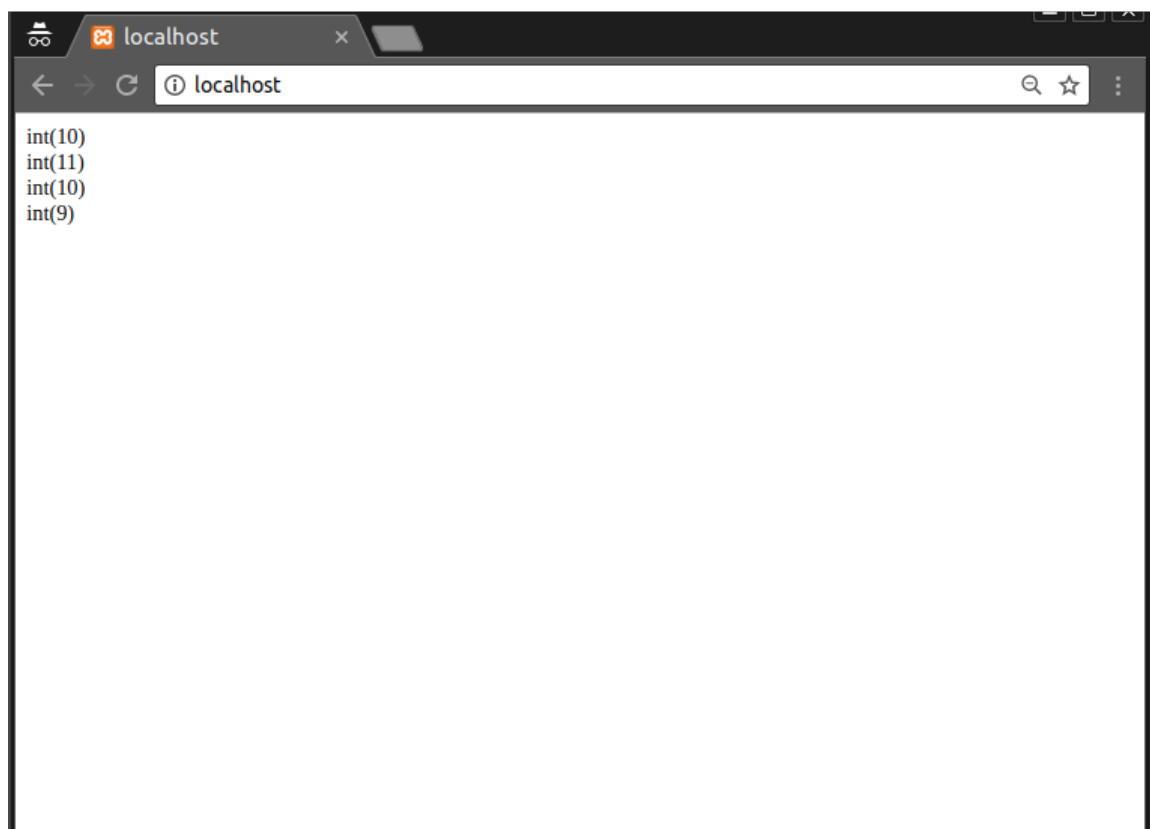
Increment / Decrement Operators

Operator ini digunakan untuk menambahkan nilai atau mengurangi nilai 1 ke dalam variabel. Berikut contoh kodenya :

```
1 <?php
2 $nilai = 10;
```

```
3 var_dump($nilai++);  
4 echo "<br>";  
5  
6 $nilai = 10;  
7 var_dump(++$nilai);  
8 echo "<br>";  
9  
10 $nilai = 10;  
11 var_dump($nilai--);  
12 echo "<br>";  
13  
14 $nilai = 10;  
15 var_dump(--$nilai);  
16 echo "<br>";  
17 ?>
```

Hasil keluaran dari dari kode di atas adalah seperti pada gambar [2.10](#) berikut :



Gambar 2.10: Hasil Operator Penambahan dan Pengurangan

Terlihat bahwa pada saat operator di tambahkan di belakang variabel, maka proses penambahan atau pengurangan dilakukan setelah ekspresi, sedangkan saat operator di tambahkan di depan variabel, maka proses penambahan atau pengurangan dilakukan sebelum ekspresi dieksekusi.

Logical Operators

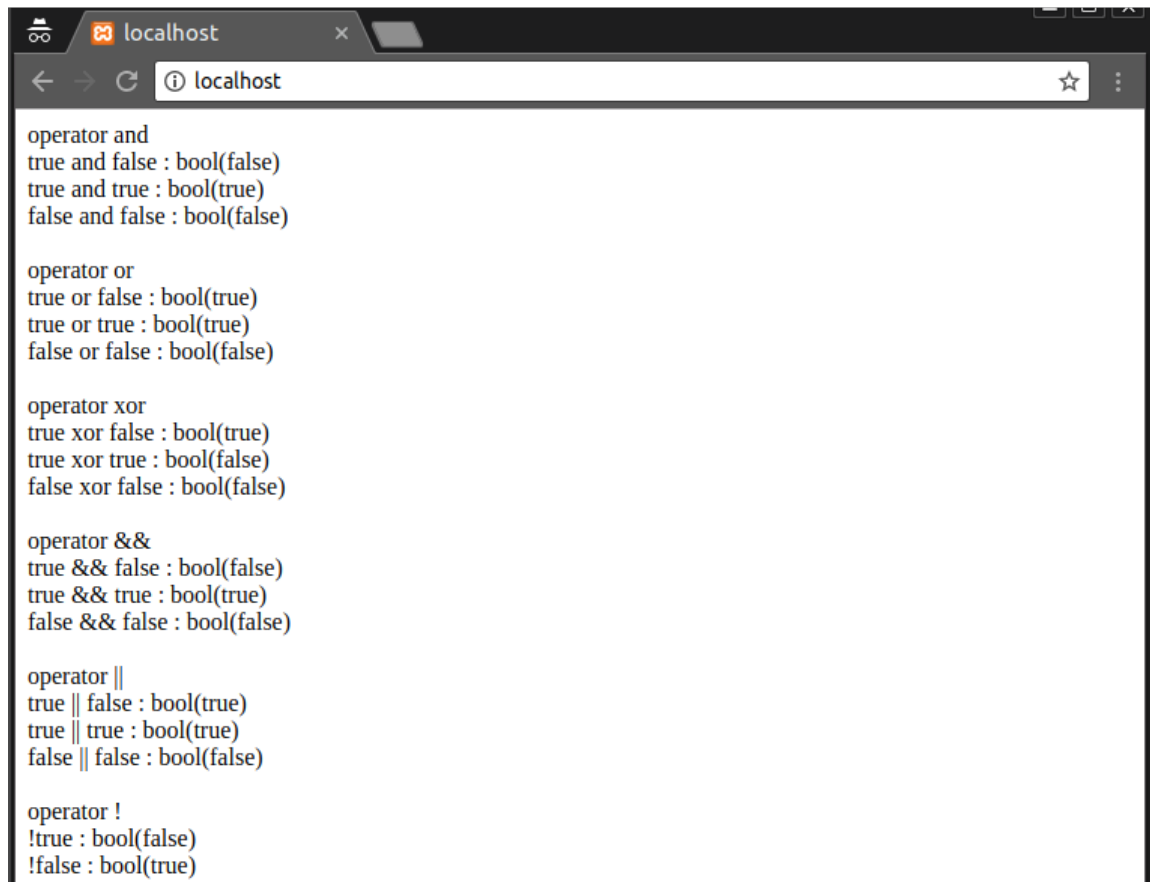
Operator logika digunakan untuk melakukan pemeriksaan benar dan salah dari dua buah data. Contoh kode untuk penggunaan operator logika ini adalah sebagai berikut :

```

1 <?php
2 $a = true;
3 $b = false;
4 $c = true;
5 $d = false;
6
7 echo "operator and<br>";
8 echo "true and false : "; var_dump($a and $b); echo "<br>";
9 echo "true and true : "; var_dump($a and $c); echo "<br>";
10 echo "false and false : "; var_dump($b and $d); echo "<br><br>";
11
12 echo "operator or<br>";
13 echo "true or false : "; var_dump($a or $b); echo "<br>";
14 echo "true or true : "; var_dump($a or $c); echo "<br>";
15 echo "false or false : "; var_dump($b or $d); echo "<br><br>";
16
17 echo "operator xor<br>";
18 echo "true xor false : "; var_dump($a xor $b); echo "<br>";
19 echo "true xor true : "; var_dump($a xor $c); echo "<br>";
20 echo "false xor false : "; var_dump($b xor $d); echo "<br><br>";
21
22 echo "operator &&<br>";
23 echo "true && false : "; var_dump($a && $b); echo "<br>";
24 echo "true && true : "; var_dump($a && $c); echo "<br>";
25 echo "false && false : "; var_dump($b && $d); echo "<br><br>";
26
27 echo "operator ||<br>";
28 echo "true || false : "; var_dump($a || $b); echo "<br>";
29 echo "true || true : "; var_dump($a || $c); echo "<br>";
30 echo "false || false : "; var_dump($b || $d); echo "<br><br>";
31
32 echo "operator !<br>";
33 echo "!true : "; var_dump(!$a); echo "<br>";
34 echo "!false : "; var_dump(!$b); echo "<br>";
35 ?>

```

Hasil keluaran dari kode di atas adalah seperti pada gambar [2.11](#) berikut :



Gambar 2.11: Hasil Keluaran Dari Operator Logika

Terlihat bahwa operator **and** dan **&&** akan menghasilkan nilai yang sama, sedangkan operator **or** and **||** akan menghasilkan nilai yang sama pula.

Untuk operator **xor**, maka apabila ada salah satu data yang bernilai **true** maka akan mengembalikan nilai **true**, namun apabila nilai kedua data adalah **false** atau **true**, maka hasilnya akan bernilai **false**.

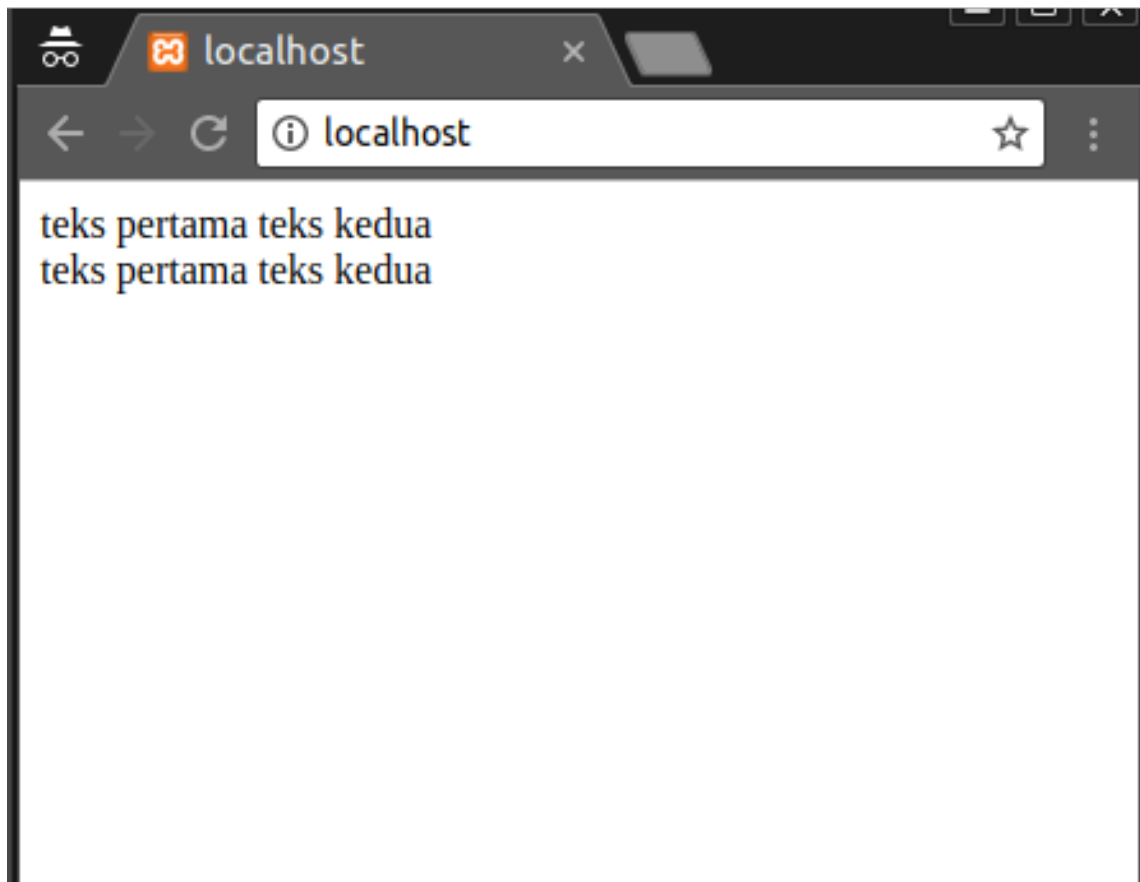
Operator **!** (tanda seru) akan menegasikan hasil datanya.

String Operators

Operator teks sederhana yang disediakan oleh PHP adalah operator untuk menggabungkan 2 (dua) teks menjadi 1 rangkaian. Contoh kodenya adalah sebagai berikut :

```
1 <?php
2 $teks1 = "teks pertama ";
3 $teks2 = "teks kedua";
4
5 echo $teks1.$teks2; echo "<br>";
6 $teks1 .= $teks2;
7 echo $teks1;
8 ?>
```

Hasil keluaran dari kode di atas adalah seperti pada gambar 2.12 berikut :



Gambar 2.12: Hasil Kode Penggabungan Teks

Operatornya hanya sebuah . (titik) untuk menggabungkan 2 (dua) teks menjadi satu, apabila hasilnya akan dimasukkan ke variabel pertama, maka dapat menggunakan operator .= (titik sama dengan).

Array Operators

Beberapa operator yang dapat digunakan untuk *array* seperti terlihat pada contoh kode berikut :

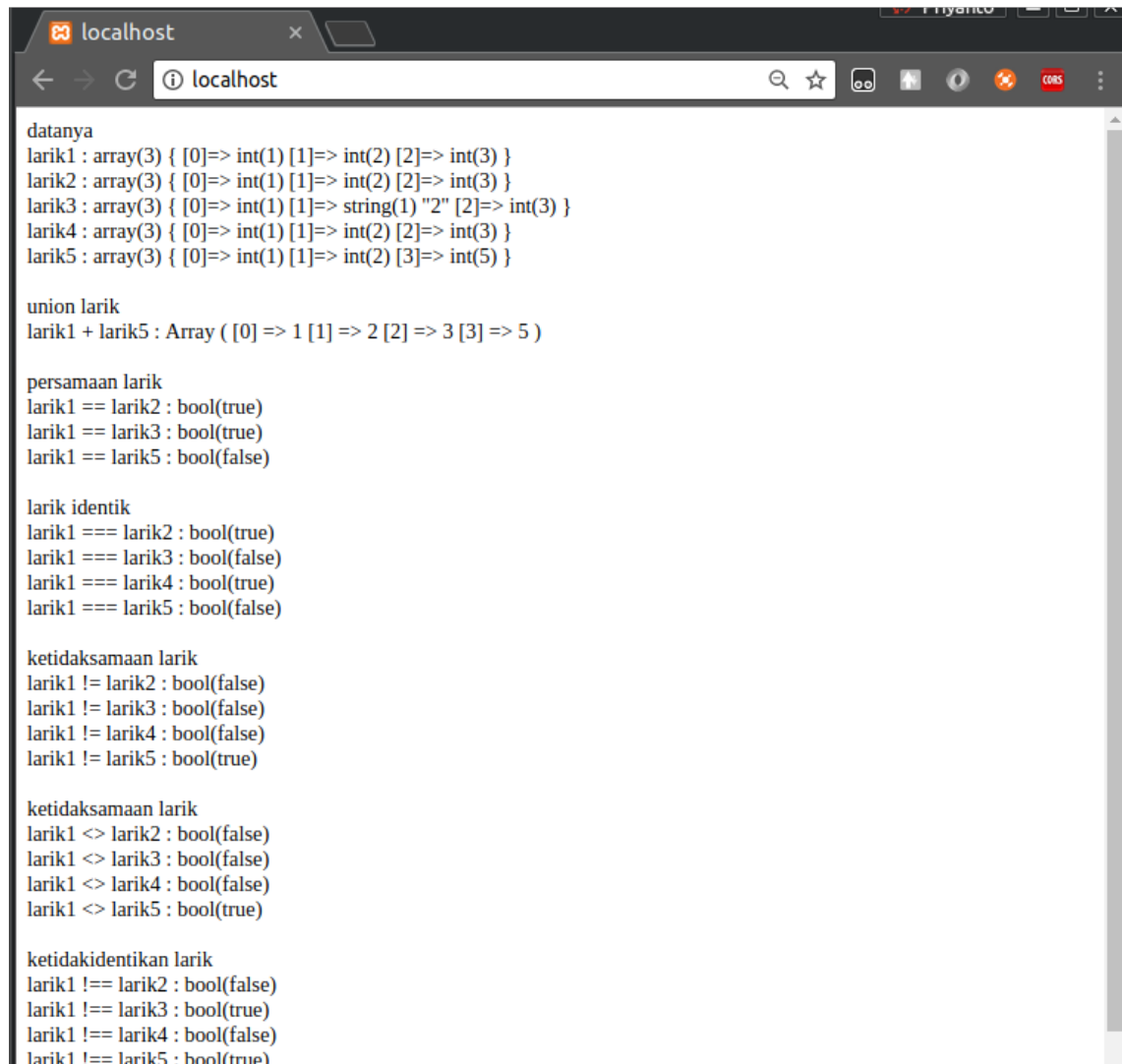
```
1 <?php
2 $larik1 = array(1, 2, 3);
3 $larik2 = array(1, 2, 3);
4 $larik3 = array(1, "2", 3);
5 $larik4 = array("0" => 1, 2, 3);
6 $larik5 = array(0 => 1, 1 => 2, 3 => 5);
7
8 echo "datanya<br>";
9 echo "larik1 : "; var_dump($larik1); echo "<br>";
```

```

10 echo "larik2 : "; var_dump($larik2); echo "<br>";
11 echo "larik3 : "; var_dump($larik3); echo "<br>";
12 echo "larik4 : "; var_dump($larik4); echo "<br>";
13 echo "larik5 : "; var_dump($larik5); echo "<br><br>";
14
15 echo "union larik<br>";
16 echo "larik1 + larik5 : "; print_r($larik1 + $larik5); echo "<br><br>";
17
18 echo "persamaan larik<br>";
19 echo "larik1 == larik2 : "; var_dump($larik1 == $larik2); echo "<br>";
20 echo "larik1 == larik3 : "; var_dump($larik1 == $larik3); echo "<br>";
21 echo "larik1 == larik5 : "; var_dump($larik1 == $larik5); echo "<br><br>";
22
23 echo "larik identik<br>";
24 echo "larik1 === larik2 : "; var_dump($larik1 === $larik2); echo "<br>";
25 ;
26 echo "larik1 === larik3 : "; var_dump($larik1 === $larik3); echo "<br>";
27 ;
28 echo "larik1 === larik4 : "; var_dump($larik1 === $larik4); echo "<br>";
29 ;
30 echo "larik1 === larik5 : "; var_dump($larik1 === $larik5); echo "<br><br>";
31
32 echo "ketidaksamaan larik<br>";
33 echo "larik1 != larik2 : "; var_dump($larik1 != $larik2); echo "<br>";
34 echo "larik1 != larik3 : "; var_dump($larik1 != $larik3); echo "<br>";
35 echo "larik1 != larik4 : "; var_dump($larik1 != $larik4); echo "<br>";
36 echo "larik1 != larik5 : "; var_dump($larik1 != $larik5); echo "<br><br>";
37
38 echo "ketidaksamaan larik<br>";
39 echo "larik1 < larik2 : "; var_dump($larik1 < $larik2); echo "<br>";
40 echo "larik1 < larik3 : "; var_dump($larik1 < $larik3); echo "<br>";
41 echo "larik1 < larik4 : "; var_dump($larik1 < $larik4); echo "<br>";
42 echo "larik1 < larik5 : "; var_dump($larik1 < $larik5); echo "<br><br>";
43
44 echo "ketidakidentikan larik<br>";
45 echo "larik1 !== larik2 : "; var_dump($larik1 !== $larik2); echo "<br>";
46 ;
47 echo "larik1 !== larik3 : "; var_dump($larik1 !== $larik3); echo "<br>";
48 ;
49 echo "larik1 !== larik4 : "; var_dump($larik1 !== $larik4); echo "<br>";
50 ;
51 echo "larik1 !== larik5 : "; var_dump($larik1 !== $larik5); echo "<br><br>";
52
53 ?>

```

Hasil keluaran dari kode di atas adalah seperti pada gambar [2.13](#) berikut :

Gambar 2.13: Hasil Keluaran Operator *Array* / Larik

2.4 Kesimpulan

Bahwa variabel dapat dibentuk untuk menyimpan sebuah data dengan tipe data tertentu, yang kemudian datanya dapat diolah menggunakan operator sederhana untuk menghasilkan data baru.

2.5 Tugas

Cobalah untuk membuat beberapa variabel dengan data sebagai berikut :

Nama Variabel	Keterangan
nama	diisikan nama pribadi
jurusan	diisikan jurusan program studi
usia	diisikan usia sekarang

Kemudian cetak hasilnya ke *browser* dengan menggunakan `echo`.