



WordPress Theme Development



Part 1: Dasar

Mulai membuat theme - I

- Buka code editor dan command line terminal
- Jalankan WordPress di Localhost
 - Wp-cli `wp core download` atau download di wordpress.org
- Clone dari [repository](#) di folder wp-content/themes
 - `git clone git@github.com:tonjoo/WordPress-theme-from-scratch.git`
 - `Cd themes/WordPress-theme-from-scratch`
 - `git checkout be0edf4`
 - Install Fakerpress -> generate post dan term (category dan tag)

Mulai membuat theme - II

Masuk ke folder themes `git clone git@github.com:tonjoo/WordPress-theme-from-scratch.git`

```
● bismilllah@tonjoo-yayan:~/htdocs/basic-theme/wp-content/themes$ git clone git@github.com:tonjoo/WordPress-theme-from-scratch.git
Cloning into 'WordPress-theme-from-scratch'...
remote: Enumerating objects: 2963, done.
remote: Counting objects: 100% (2963/2963), done.
remote: Compressing objects: 100% (2236/2236), done.
remote: Total 2963 (delta 724), reused 2963 (delta 724), pack-reused 0
Receiving objects: 100% (2963/2963), 6.50 MiB | 2.30 MiB/s, done.
Resolving deltas: 100% (724/724), done.
● bismilllah@tonjoo-yayan:~/htdocs/basic-theme/wp-content/themes$ cd WordPress-theme-from-scratch/
● bismilllah@tonjoo-yayan:~/htdocs/basic-theme/wp-content/themes/WordPress-theme-from-scratch$ git log --oneline
fatal: unrecognized argument: --oneline
● bismilllah@tonjoo-yayan:~/htdocs/basic-theme/wp-content/themes/WordPress-theme-from-scratch$ git log --oneline
60de019 (HEAD -> master, origin/master, origin/HEAD) Part 3, Step 2
156feb4 Part 3, Step 1
8efec1e Part 2, Step 2 : widget
a23df50 Part 2, Step 1 : daftarkan menu di functions.php
c6d005a Step 5 : Single.php
19c2493 Step 4 : Membuat Excerpt otomatis di functions.php
007c73f Step 3 : Query Loop
fbde939 Step 2 : functions.php
bf92316 Step 1 : Base Theme
● bismilllah@tonjoo-yayan:~/htdocs/basic-theme/wp-content/themes/WordPress-theme-from-scratch$ git checkout be0edf4
```

Dengan melakukan step diatas tema sudah terinstal dan hanya perlu mengaktifkan

Mulai membuat theme - III

Identitas Theme

Setelah itu, kita perlu membuat identitas untuk theme baru kita.

Pertama, buat file bernama `style.css` di dalam direktori `tonjoo-tutorial`.

Menggunakan file ini, kita bisa mengatur nama theme, versi, author, dan lain-lain. List lengkap dapat dilihat [disini](#).

 **style.css**  362 Bytes

```
1  /*
2  Theme Name: Tutorial
3  Theme URI: https://tonjoo.com/
4  Author: Tonjoo
5  Author URI: https://tonjoo.com/
6  Description: Theme tutorial
7  Version: 1.0.0
8  Tested up to: 6.1
9  Requires PHP: 5.6
10 License: GNU General Public License v2 or later
11 License URI: http://www.gnu.org/licenses/gpl-2.0.html
12 Text Domain: tutorial
13 Tags: custom-menu, featured-images, translation-ready
14 */
```

Mulai membuat theme - IV

Index.php

Sebuah theme minimal memerlukan index.php untuk bekerja

```
<?php
/**
 * The main template file
 *
 * @package Tutorial
 */
get_header();
?>
<H1>Hello Word</H1>

<?php
get_footer();
```

Cheatsheet Git

1. Link step Percommit

[commit step](#)

2. Langkah Memulai Coding

- mulai code (tidak perlu sampai selesai) ke step ke 2 kemudian save tidak perlu di commit
- setelah itu `git restore .` (dilakukan sebelum checkout ke step selanjutnya)
- Jika ada file yang ditambahkan run `git clean -df` untuk menghapusnya langsung dan menyamakan status code dengan commit an yg sekarang kita checkout

3. Untuk checkout ke step lain

- step 1 `git checkout bf92316``
- step 2 `git checkout fbde939``
- step 3 `git checkout 007c73f``
- step 4 `git checkout 19c2493``
- step 5 `git checkout c6d005a``
- part 2 step 1 `git checkout a23df50``
- part 2 step 2 `git checkout 8efec1e``
- part 3 step 1 `git checkout 156feb4``
- part 3 step 2 `git checkout 60de019``

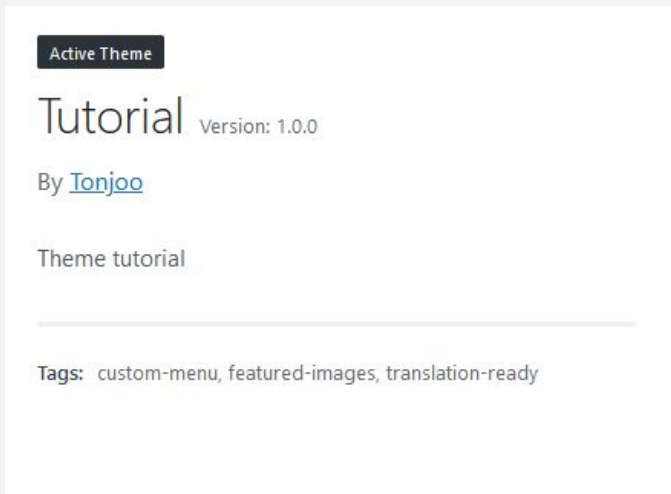
1 – Theme Basics

```
git checkout 502420d
```

Identitas Theme – Hasil

Setelah itu, buka halaman dashboard WordPress → Themes. Sekarang Anda dapat melihat theme Tutorial sudah tampil di list theme.

Mari kita aktifkan theme lalu lihat bagian front end. Anda akan melihat bahwa websitenya hanya blank screen, karena memang di theme yang kita buat belum ada apa-apa. Karena itu, coba kita implement HTML template menjadi theme WordPress



1 – Theme Basics

```
git checkout 502420d
```

Index.php

Copy [index.html](#) dari template HTML ke direktori tonjoo-tutorial lalu di-rename menjadi index.php. Copy juga direktori assets dan file-filenya di direktori yang sama.

Coba refresh tampilan front end, Anda akan melihat tampilan html yang tadi kita copy sudah muncul.

Selanjutnya, coba kita panggil file CSS dan JS yang ada di template melalui theme.

Anthony's Blog

Hi, my name is Anthony Doe. Briefly introduce yourself here. You can also provide a link to the about page.

[Read out more about me](#)

-
-
-
-

- [Blog Home \(current\)](#)
- [Blog Post](#)
- [About Me](#)

[Get in Touch](#)

DevBlog - A Blog Template Made For Developers

Welcome to my blog. Subscribe and get my latest blog post in your inbox.

Your email

image

Top 3 JavaScript Frameworks

Published 2 days ago 5 min read [comments](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Donec quam felis, ultricies...

[Read more](#)

image

About Remote Working

Published 3 months ago 3 min read [comments](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Donec quam felis, ultricies...

[Read more](#)

image

A Guide to Becoming a Full-Stack Developer



2 – Functions.php, Header, Footer

`git checkout fbde939`

[functions.php](#)

Untuk menambahkan fungsi-fungsi di theme WordPress, kita menggunakan functions.php. Hal-hal yang dapat diatur melalui functions.php contohnya (dan tidak terbatas pada):

- Inisiasi fitur pada tema
- Inject CSS atau JS
- Enable/disable fitur WordPress
- Register menu, sidebar, post type, dan lain-lain

2 – Functions.php, Header, Footer

`git checkout fbde939`

[functions.php](#) - Best Practice

Agar lebih mudah, kita coba define beberapa constant yang digunakan di beberapa tempat di functions.php:

- Theme Version
- Template directory
- Template directory URI

```
8  define( 'TUTORIAL_VERSION', wp_get_theme()->get( 'Version' ) );
9  define( 'TUTORIAL_DIR', get_template_directory() );
10 define( 'TUTORIAL_URL', get_template_directory_uri() );
```

2 – Functions.php, Header, Footer

`git checkout fbde939`

[functions.php](#) - Best Practice

Coba kita aktifkan beberapa fitur WordPress theme seperti:

- Translasi/text domain
- Theme support rss link tag -> otomatis generate link rss tag
- Theme support title tag -> Otomatis generate title tag

```
12 /**
13  * Sets up theme defaults and registers support for various WordPress features.
14  */
15 function tutorial_setup() {
16     /*
17      * Make theme available for translation.
18      * Translations can be filed in the /languages/ directory.
19      * If you're building a theme based on Tutorial, use a find and replace
20      * to change 'tutorial' to the name of your theme in all the template files.
21      */
22     load_theme_textdomain( 'tutorial', TUTORIAL_DIR . '/languages' );
23
24     // Add default posts and comments RSS feed links to head.
25     add_theme_support( 'automatic-feed-links' );
26
27     /*
28      * Let WordPress manage the document title.
29      * By adding theme support, we declare that this theme does not use a
30      * hard-coded <title> tag in the document head, and expect WordPress to
31      * provide it for us.
32      */
33     add_theme_support( 'title-tag' );
34 }
35 add_action( 'after_setup_theme', 'tutorial_setup' );
36
```

2 – Functions.php, Header, Footer

`git checkout fbde939`

[functions.php](#) - Inject CSS atau JS

Untuk meng-inject CSS atau JS ikuti sample di bawah atau pada source code yang tersedia. Perhatikan, kita menggunakan `wp_enqueue_style()` untuk me-load CSS and `wp_enqueue_scripts()` untuk JS. Perhatikan juga bagaimana kita menggunakan constant `TUTORIAL_URL` dan `TUTORIAL_VERSION`.

```
37  /**
38   * Enqueue scripts and styles.
39   */
40  function tutorial_scripts() {
41      wp_enqueue_style( 'tutorial-style', get_stylesheet_uri(), array(), TUTORIAL_VERSION );
42      wp_enqueue_style( 'tutorial-theme-1', TUTORIAL_URL . '/assets/css/theme-1.css', array(), TUTORIAL_VERSION );
43
44      wp_enqueue_script( 'fontawesome', TUTORIAL_URL . '/assets/fontawesome/js/all.min.js', array(), '6.1.1', true );
45      wp_enqueue_script( 'popper', TUTORIAL_URL . '/assets/plugins/popper.min.js', array(), '2.11.5', true );
46      wp_enqueue_script( 'bootstrap', TUTORIAL_URL . '/assets/plugins/bootstrap/js/bootstrap.min.js', array(), '5.2.0', true );
47  }
48  add_action( 'wp_enqueue_scripts', 'tutorial_scripts' );
```

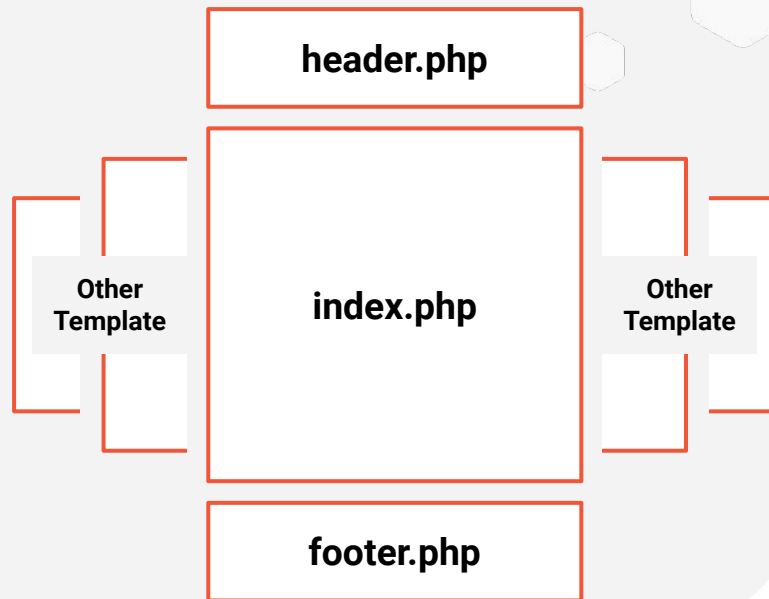
2 – Functions.php, Header, Footer

`git checkout fbde939`

Pemisahan Template

Mari kita pisahkan header dan footer dengan index.

1. Buat file header.php and footer.php.
2. Pindahkan bagian header dari source code di index.php ke dalam header.php
3. Pindahkan bagian footer dari source code di index.php ke dalam footer.php



2 – Functions.php, Header, Footer, Index

`git checkout fbde939`

Index.php Memanggil [header](#) dan [footer](#) di halaman index

Gunakan `get_header()` untuk memanggil header.php.
Letakkan fungsi tersebut di awal index.php

Gunakan `get_footer()` untuk memanggil footer.php.
Letakkan fungsi tersebut di akhir file index.php

```
1 <?php
2 /**
3  * The main template file
4  *
5  * @package Tutorial
6  */
7
8 get_header();
9 ?>
```

```
124 <?php
125 get_footer();
```

2 – Functions.php, Header, Footer

`git checkout fbde939`

Fungsi WordPress pada [Header](#)

Beberapa fungsi WP yang dapat di-implement:

- [language_attributes\(\)](#)
- [bloginfo\(\)](#)
- [wp_head\(\)](#)
- [body_class\(\)](#)
- [wp_body_open\(\)](#)

Setiap theme WP wajib memiliki `wp_head()` dan `wp_body_open()`. Sisanya di contoh adalah tambahan namun secara best practice lebih baik dipanggil juga sesuai posisinya.

```
10 <html <?php language_attributes(); ?>
11 <head>
12     <meta charset="<?php bloginfo( 'charset' ); ?>">
13     <meta name="viewport" content="width=device-width, initial-scale=1">
14     <link rel="profile" href="https://gmpg.org/xfn/11">
15
16     <?php wp_head(); ?>
17 </head>
18
19 <body <?php body_class(); ?>
20 <?php wp_body_open(); ?>
```




2 – Functions.php, Header, Footer

`git checkout fbde939`

Putting some PHP on [Footer](#)

Pada footer `wp_footer()` wajib dipanggil tepat sebelum tutup tag `</body>`.

```
18
19
20  <?php wp_footer(); ?>
21
22  </body>
23  </html>
```

2 – Functions.php, Header, Footer

`git checkout 531c3bb`

Hasil render di footer

```
180 </div><!--//main-wrapper-->
181
182
183 <script type='text/javascript' src='http://localhost/wordpress/wp-content/themes/tutorial/assets/fontawesome/js/all.min.js?ver=6.1.1' id='fontawesome-js'></script>
184 <script type='text/javascript' src='http://localhost/wordpress/wp-content/themes/tutorial/assets/plugins/popper.min.js?ver=2.11.5' id='popper-js'></script>
185 <script type='text/javascript' src='http://localhost/wordpress/wp-content/themes/tutorial/assets/plugins/bootstrap/js/bootstrap.min.js?ver=5.2.0' id='bootstrap-js'></script>
186
```



2 – Functions.php, Header, Footer

`git checkout fbde939`

Kesimpulan

- Header & footer terpisah dari index
- CSS dan JS sekarang sudah dipanggil menggunakan functions.php dan ditampilkan melalui wp_head() and wp_footer()

Di materi selanjutnya, kita akan belajar cara me-render konten WordPress di front end.

3 – Query Loop

`git checkout 007c73f` [index.php](#)

The Loop

Adalah rangkaian kode PHP yang digunakan WordPress untuk menampilkan posts. Menggunakan The Loop, WordPress memproses setiap post untuk ditampilkan di halaman, dan diformat menggunakan kriteria yang diinginkan di dalam loop.

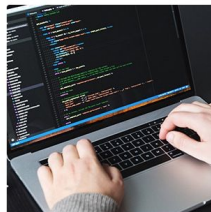
```
1 <?php
2 //Check if the page / post have "posts"
3 if ( have_posts() ) {
4     //do magic loop
5     while ( have_posts() ) {
6         //assign $post variable to local variable
7         the_post();
8
9         //
10        // Post Content here
11        //
12    } // end while
13 } else{
14     echo "No Post match this criteria";
15 }
16
17
18 >
```

3 – Query Loop

`git checkout 007c73f` [index.php](#)

Komponen Loop

- Judul
- Konten
- Link URL (Read more)
- Thumbnail
- Info komentar
- (custom) tanggal publish
- (custom) durasi membaca



Top 3 JavaScript Frameworks

Published 2 days ago • 5 min read • 8 comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies...

[Read more →](#)

About Remote Working

Published 3 months ago • 3 min read • 26 comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies...

[Read more →](#)

3 – Query Loop

[git checkout 007c73f index.php](#)

Post Thumbnail → `the_post_thumbnail()`

```
<div class="col-2 col-xl-3">
    <?php the_post_thumbnail( 'thumbnail', array( 'class' => 'img-fluid post-thumb' ) ); ?>
</div>
```

Judul → `the_title()` **Post URL** → `the_permalink()`

Konten → `the_permalink()` **Info komentar** → `comments_popup_link()`

```
<h3 class="title mb-1"><a class="text-link" href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h3>
<div class="meta mb-1">
    <span class="date"><?php tutorial_publish_date(); ?></span>
    <span class="time"><?php tutorial_reading_time(); ?></span>
    <span class="comment"><?php comments_popup_link( false, false, false, 'text-link' ); ?></span>
</div>
<div class="intro"><?php the_content(); ?></div>
<a class="text-link" href="<?php the_permalink(); ?>">Read more &rarr;</a>
```

3 – Query Loop

`git checkout 007c73f` [functions.php](#)

Fungsi custom

- `tutorial_publish_date()` → dibuat “xx days ago, yy months ago, etc”
- `tutorial_reading_time()` → menghitung durasi membaca berdasarkan 161 kata per menit

```
60 function tutorial_publish_date() {
61     echo sprintf(
62         /* translators: %s: time */
63         esc_html__( 'Published %s ago', 'tutorial' ),
64         esc_html( human_time_diff( get_the_time( 'U' ), current_time( 'timestamp' ) ) )
65     );
66 }
67 /**
68  * Display reading time.
69  */
70 function tutorial_reading_time() {
71     $content = get_the_content( null, false, get_the_ID() );
72     $content = strip_shortcodes( $content );
73     $content = wp_strip_all_tags( $content );
74
75     $word_count = count( preg_split( '/\s+/', $content ) );
76
77     $wpm = 161;
78     $time = ceil( $word_count / $wpm );
79     $time = max( 1, $time );
80
81     $read_time = sprintf(
82         /* translators: %s: Number of minutes. */
83         __( '%s min read', 'tutorial' ),
84         $time
85     );
86     echo esc_html( $read_time );
87 }
```



3 – Query Loop

Kesimpulan

- Menampilkan konten dinamis di halaman index
- Konten masih ditampilkan semuanya, kita butuh memotong untuk halaman index.

Pada materi selanjutnya, kita akan belajar bagaimana cara meng-kustom output dari konten WordPress menggunakan filter.

4 – Using Filter Hooks

`git checkout 19c2493` [functions.php](#)

Mengubah output `the_content()`

Kita akan mencoba mengubah output [the_content\(\)](#) dari menampilkan konten secara full menjadi hanya menampilkan 36 kata dan menambahkan '...' di akhir hanya di halaman arsip dan homepage.

Kita menggunakan [wp_trim_words\(\)](#) untuk memotong konten, dan menggunakan [add_filter\(\)](#) untuk diimplement ke fungsi `the_content()`.

```
88 +
89 + /**
90 + * Generate a trimmed-down version of the full post content for archive & home
91 + *
92 + * @param string $content Content of the current post.
93 + * @return string
94 + */
95 + function tutorial_the_content( $content ) {
96 +     if ( is_archive() || is_home() ) {
97 +         $content = wp_trim_words( $content, 36 );
98 +     }
99 +     return $content;
100 + }
101 + add_filter( 'the_content', 'tutorial_the_content' );
```

4 – Using Filter Hooks

`git checkout 19c2493` [functions.php](#)

Result



Top 3 JavaScript Frameworks

Published 2 weeks ago • 2 min read • No Comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis,...

[Read more →](#)

5 – Single.php

`git checkout c6d005a`, [single.html](#), [single.php](#)

Membuat halaman single/detail konten

Karena kita sudah bisa membuat halaman index untuk menampilkan list artikel, mari kita buat halaman detail artikelnya.

Mari kita buat file bernama `single.php` dan copy bagian body dari template `blog-post.html` ke dalam file php tersebut.

Why Every Developer Should Have A Blog

Published 2 days ago • 6 min read • 4 comments



Image Credit: [made4dev.com](#) (Premium Programming T-shirts)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula,

5 – Single.php

[git checkout c6d005a](#), [single.html](#), [single.php](#)

Menambahkan Logic (1)

Catatan: Anda harus menerapkan the loop (while (have_posts()) : the_post();) di dalam template agar fungsi-fungsi konten bekerja.

Gunakan the_title(), tutorial_publish_date(), tutorial_reading_time(), and comments_popup_link() seperti biasa.

```
<h2 class="title mb-2"><?php the_title(); ?></h2>
<div class="meta mb-3">
  <span class="date"><?php tutorial_publish_date(); ?></span>
  <span class="time"><?php tutorial_reading_time(); ?></span>
  <span class="comment"><?php comments_popup_link( false, false, false, 'text-link' ); ?></span>
</div>
```

5 – Single.php

`git checkout c6d005a`, [single.html](#), [single.php](#)

Menambahkan Logic (2)

Thumbnail: gunakan `the_post_thumbnail()` untuk menampilkan thumbnail yang di-wrap kondisional fungsi `has_post_thumbnail()` untuk mengecek apakah post tersebut memiliki featured image atau tidak. Anda juga akan menjumpai fungsi kustom `tutorial_post_thumbnail_caption()`, yang akan dijelaskan kemudian.

```
<div class="blog-post-body">
  <?php if ( has_post_thumbnail() ) : ?>
    <figure class="blog-banner">
      <?php the_post_thumbnail( 'large', array( 'class' => 'img-fluid' ) ); ?>
      <?php tutorial_post_thumbnail_caption(); ?>
    </figure>
  <?php endif; ?>

  <?php the_content(); ?>
</div>
```

5 – Single.php

`git checkout c6d005a` [function.php](#)

Fungsi kustom caption

`tutorial_post_thumbnail_caption()` adalah fungsi kustom untuk menampilkan caption gambar sesuai dengan struktur HTML template yang digunakan.

```
103 /**
104  * Display featured image caption
105  */
106 function tutorial_post_thumbnail_caption() {
107     $thumbnail_id = get_post_thumbnail_id();
108     $thumbnail    = get_post( $thumbnail_id );
109     if ( ! empty( $thumbnail->post_excerpt ) ) {
110         echo '<figcaption class="mt-2 text-center image-caption">';
111         echo wp_kses_post( $thumbnail->post_excerpt );
112         echo '</figcaption>';
113     }
114 }
```

5 – Single.php

`git checkout c6d005a` [function.php](#)

Navigasi Prev/Next

Untuk menampilkan navigasi (prev/next) kita membuat fungsi kustom bernama `tutorial_post_navigation()`. Lebih detail, bisa dicek di source code

```
116 /**
117  * Display single post navigation
118  */
119 function tutorial_post_navigation() {
120     $previous = get_previous_post_link(
121         '%link',
122         esc_html__( 'Previous', 'tutorial' ) . '<i class="arrow-prev fas fa-long-arrow-alt-left"></i>'
123     );
124     $next = get_next_post_link(
125         '%link',
126         esc_html__( 'Next', 'tutorial' ) . '<i class="arrow-next fas fa-long-arrow-alt-right"></i>'
127     );
128     $previous = str_replace( 'rel="prev"', 'rel="prev" class="nav-link-prev nav-item nav-link rounded-left"', $previous );
129     $next      = str_replace( 'rel="next"', 'rel="next" class="nav-link-next nav-item nav-link rounded-right"', $next );
130
131     if ( $previous || $next ) {
132         echo '<nav class="blog-nav nav nav-justified my-5">';
133         echo wp_kses_post( $previous );
134         echo wp_kses_post( $next );
135         echo '</nav>';
136     }
137 }
```



Previous

Next



5 – Single.php

```
git checkout c6d005a
```

Hasil

Template dapat menampilkan:

- Judul
- Tanggal publish
- Waktu membaca
- Informasi komentar
- Thumbnail dan caption
- Konten
- Navigasi next/prev

About Remote Working

Published 3 days ago • 1 min read • No Comments



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.



Previous

Next





Part 2: Menu & Widget

1 – Implement WordPress Menu

`git checkout a23df50`

Register Menu [function.php](#)

Untuk menampilkan menu untuk WordPress pertama kali diregister dulu menggunakan fungsi `register_nav_menus()`.

Memanggil Menu [header.php](#)

Gunakan `wp_nav_menu` untuk mengganti template HTML navigation menu di dalam `header.php` menggunakan fungsi `wp_nav_menu()`

```
// This theme uses wp_nav_menu() in one location.  
register_nav_menus(  
    array(  
        'menu-1' => esc_html__( 'Primary', 'tutorial' ),  
    )  
);
```

```
<?php  
wp_nav_menu(  
    array(  
        'theme_location' => 'menu-1',  
        'container'      => false,  
        'menu_class'     => 'navbar-nav flex-column text-start',  
    )  
);  
?>
```

1 – Implement WordPress Menu

`git checkout a23df50`

Tambahan - Menambah Class ke List Item [function.php](#)

Karena kita butuh menambahkan class `nav-item` ke menu, gunakan filter bernama `nav_menu_css_class`.

```
146 + /**
147 + * Add classes to menu item list
148 + *
149 + * @param array $classes Classes.
150 + * @param obj $item Menu item object.
151 + * @param obj $args Menu arguments.
152 + * @return array
153 + */
154 + function tutorial_menu_item_list_class( $classes, $item, $args ) {
155 +     if ( 'menu-1' === $args->theme_location ) {
156 +         $classes[] = 'nav-item';
157 +     }
158 +     return $classes;
159 + }
160 + add_filter( 'nav_menu_css_class', 'tutorial_menu_item_list_class', 10, 3 );
```

1 – Implement WordPress Menu

`git checkout a23df50`

Tambahan - Menambah Class ke Anchor Item [function.php](#)

Karena kita butuh menambahkan class `nav-link` ke item navigasi <a>, kita gunakan filter `nav_menu_link_attributes`.

Also, for menu item that's currently active, we add `active` class to it.

```
162 + /**
163 + * Add classes to menu item link
164 + *
165 + * @param array $atts HTML attributes.
166 + * @param obj   $item Menu item object.
167 + * @param obj   $args Menu arguments.
168 + * @return array
169 + */
170 + function tutorial_menu_item_link_class( $atts, $item, $args ) {
171 +     if ( 'menu-1' === $args->theme_location ) {
172 +         $atts['class'] = 'nav-link';
173 +         if ( in_array( 'current-menu-item', $item->classes ) ) {
174 +             $atts['class'] .= ' active';
175 +         }
176 +     }
177 +     return $atts;
178 + }
179 + add_filter( 'nav_menu_link_attributes', 'tutorial_menu_item_link_class', 10, 3 );
```

2 – Implement WordPress Widget Area

`git checkout 8efec1e`

Register Sidebar [function.php](#)

Untuk me-register sidebar cukup dengan fungsi `register_sidebar()` lalu ikuti dokumentasi yang tersedia. Kita namakan sidebar ini `Sidebar`.

Hasilnya dapat dilihat pada gambar di samping, dari menu WP Admin → **Appearances** → **Widgets**

```
register_sidebar(  
    array(  
        'name'           => esc_html__( 'Sidebar', 'tutorial' ),  
        'id'             => 'sidebar-1',  
        'description'    => esc_html__( 'Add widgets here.', 'tutorial' ),  
        'before_widget' => '<section id="%1$s" class="widget %2$s">',  
        'after_widget'  => '</section>',  
        'before_title'  => '<h2 class="widget-title">',  
        'after_title'   => '</h2>',  
    )  
);
```



2 – Implement WordPress Widget Area

```
git checkout 8efec1e
```

Implement Widget Area on Page [header.php](#)

Cukup simpel, kita cukup memanggil fungsi `dynamic_sidebar()`.

Setelah itu anda dapat menggunakan widget HTML untuk mengimplement HTML sidebar seperti gambar di kanan. Namun, bagaimana kalo kita membuat satu widget custom baru untuk template seperti ini?

```
<?php dynamic_sidebar( 'sidebar-1' ); ?>
```



Hi, my name is Anthony Doe. Briefly introduce yourself here. You can also provide a link to the about page.

[Find out more about me](#)



2 – Implement WordPress Widget Area

`git checkout 8efec1e`

Membuat Widget Baru

Kita gunakan fungsi `register_widget()` untuk me-register suatu widget. Detail kodenya dapat dicek di source code direktori [inc/class-profile-widget.php](#), jadi jangan lupa untuk dipelajari ya!

```
register_widget( 'Tutorial\Profile_Widget' );
```

2 – Implement WordPress Widget Area

`git checkout 8efec1e`

Hasil Membuat Widget Baru

Sekarang Anda dapat menggunakan widget baru bernama `Profile` dan digunakan untuk menampilkan section profil. Anda dapat men-set gambar, bio, dan link social media menggunakan field yang tersedia.

Profile

Image URL:

Bio:

Hi, my name is Anthony Doe. Briefly introduce yourself here. You can also provide a link to the about page.
Find out more about me

Facebook URL:

Linkedin URL:

Github URL:

Stackoverflow URL:

Codepen URL:



Part 3: Finishing

3 –Paginasi di Homepage

[git checkout 156feb4](#)

- [function.php](#)

Secara default, WordPress menampilkan satu halaman berisi 10 artikel. Setelah 10 artikel tersebut, artikel lain akan ditampilkan pada halaman berikutnya.

Untuk menampilkan tombol navigasi (prev/next) di halaman home, archive, dan search results, bisa dengan menggunakan custom function berikut ini.

```
/**
 * Display posts navigation
 */
function tutorial_posts_navigation() {
    global $paged, $wp_query;
    if ( $wp_query->max_num_pages > 1 ) {
        if ( ! $paged ) {
            $paged = 1;
        }
        $nextpage = (int) $paged + 1;
        $max_page = $wp_query->max_num_pages;

        $prev_link = previous_posts( false );
        $next_link = next_posts( $max_page, false );

        echo '<nav class="blog-nav nav nav-justified my-5">';
        if ( $paged > 1 ) {
            echo '<a class="nav-link-prev nav-item nav-link rounded" href="' . esc_attr(
                $prev_link ) . '">' . esc_html__( 'Previous', 'tutorial' ) . '<i class="arrow-prev fas fa-long-arrow-alt-
                left"></i></a>&nbsp;';
        }
        if ( $nextpage <= $max_page ) {
            echo '<a class="nav-link-next nav-item nav-link rounded" href="' . esc_attr(
                $next_link ) . '">' . esc_html__( 'Next', 'tutorial' ) . '<i class="arrow-next fas fa-long-arrow-alt-
                right"></i></a>';
        }
        echo '</nav>';
    }
}
```

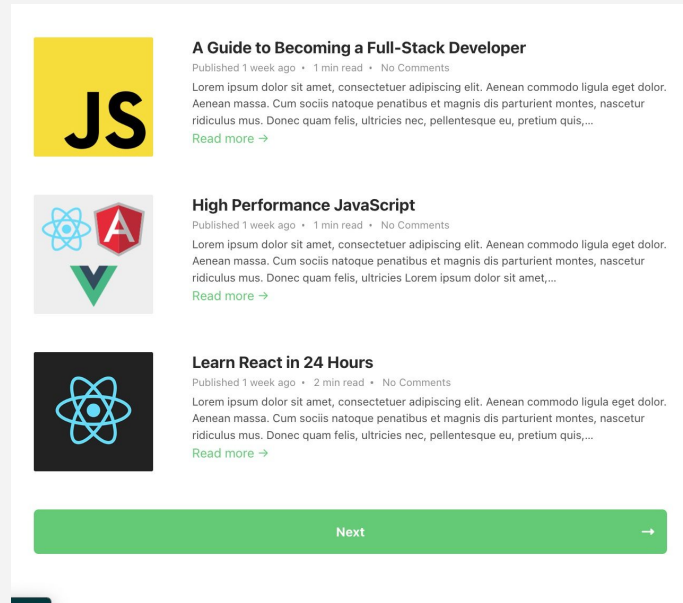
3 –Paginasi di Homepage

`git checkout 156feb4`

- [index.php](#)

Apabila jumlah artikel sudah lebih dari 10, maka tombol-tombol navigasi tersebut baru akan tampil. Tombol Next akan tampil di halaman selain halaman terakhir, sedangkan tombol Prev akan tampil di halaman selain halaman pertama.

Untuk mengganti jumlah artikel yang tampil tiap halaman, bisa melalui Settings -> Reading -> Blog pages show at most.



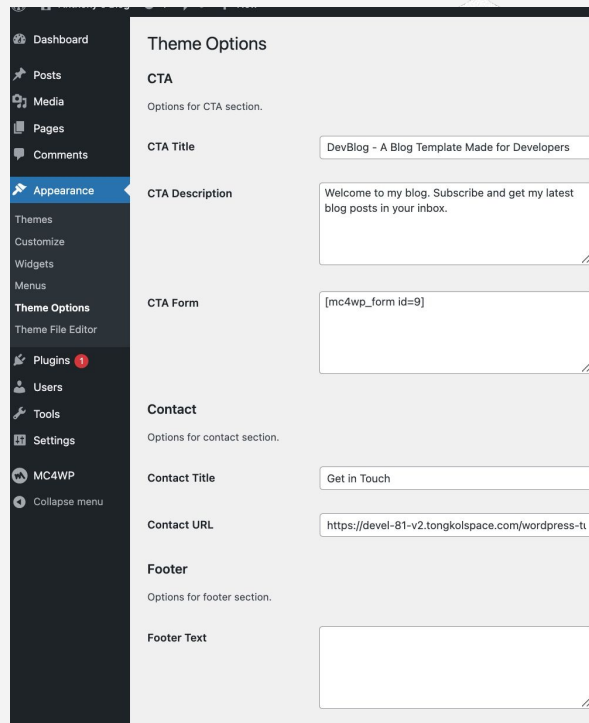
3 – Theme Options

`git checkout master`

Theme Options adalah halaman admin untuk melakukan pengaturan website tanpa menyentuh file atau kode.

Theme options dapat dibuat menggunakan plugin seperti Advanced Custom Field Pro atau manual.

Untuk cara manual, dapat melihat source code di direktori </inc/class-options.php>.



The screenshot shows the WordPress admin interface with the 'Theme Options' page selected. The left sidebar contains a menu with 'Appearance' highlighted, and 'Theme Options' is the active sub-menu. The main content area is titled 'Theme Options' and contains several sections for configuring the theme:

- CTA**: Options for CTA section.
 - CTA Title**: DevBlog - A Blog Template Made for Developers
 - CTA Description**: Welcome to my blog. Subscribe and get my latest blog posts in your inbox.
 - CTA Form**: [mc4wp_form id=9]
- Contact**: Options for contact section.
 - Contact Title**: Get in Touch
 - Contact URL**: https://devel-81-v2.tongkolospace.com/wordpress-ti
- Footer**: Options for footer section.
 - Footer Text**: (Empty text area)