



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1. Kedudukan dan Koordinasi

Proses kerja magang yang dilakukan penulis di PT SURE Indonesia berkedudukan sebagai *web developer*. Posisi *web developer* berada dibawah divisi IT bagian *IT Development* dimana divisi ini menangani permasalahan yang terkait dengan infrastruktur teknologi yang ada di PT SURE Indonesia. *IT Development* yang ada di PT SURE Indonesia membantu mengembangkan serta menangani permasalahan seperti pengembangan sistem, *website*, server.

Tugas dari *web developer* sendiri adalah mengembangkan *website* yang sudah/akan di *publish*. Tugas lainnya adalah menjadi penghubung antara sumber daya yang dibutuhkan untuk membuat *website*, seperti melakukan koneksi pada *database*, membuat halaman *website* yang dinamis, serta membantu *user/visitor website* untuk dapat berinteraksi menggunakan elemen-elemen yang ada di dalam *website*. Melakukan *maintenance* terhadap *website* yang sudah berjalan merupakan salah satu tugas juga dari seorang *web developer* untuk mencari *error/bug* pada *website* agar *website* dapat berjalan dengan lancar dan membantu juga dalam hal peningkatan keamanan pada *website*.

Penulis selama proses kerja magang mengerjakan *front-end website e-commerce* PT Kandel yang merupakan afiliasi dari PT Sure Indonesia. PT Kandel memiliki *website e-commerce* untuk melakukan penjualan terhadap produk-produk pendidikan yang diproduksi untuk memberikan materi pembelajaran dan pengajaran untuk murid maupun guru. Adapun sebelumnya, PT Kandel sudah memiliki *website e-commerce* yang sudah berjalan namun menginginkan adanya perubahan terhadap *website* yang sudah ada untuk diganti dengan *website* yang baru untuk memaksimalkan penjualan dari segi *online*. Selanjutnya penulis diberi kesempatan untuk membuat bagian *front-end websitenya* tersebut.

### 3.2. Tugas dan Proses Pelaksanaan Kerja Magang

Selama masa proses kerja magang, penulis sebagai *web developer* mengerjakan *website e-commerce* dimulai dari melakukan :

#### 1. Pengenalan Menggunakan *Codeigniter* sebagai *PHP framework*

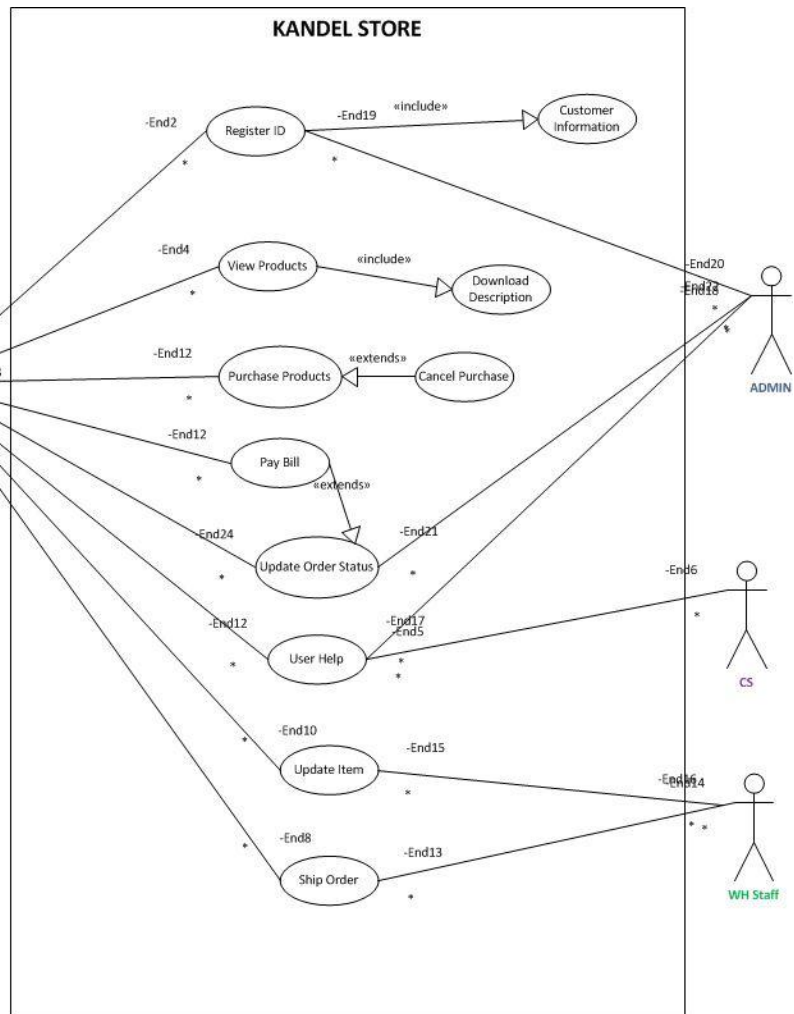
Penulis diberikan tugas untuk melakukan eksplorasi *framework* yang akan digunakan untuk melakukan proses pembuatan *website*. *Framework* yang digunakan untuk membuat *website* adalah *Codeigniter*. *Codeigniter* adalah *framework* berbasis *PHP* dan bersifat *open source* untuk membuat *website* secara dinamik. Dalam proses pembuatan maupun pengembangan *website* melalui *Codeigniter* ini, prosesnya akan berjalan lebih cepat dibandingkan dengan cara pembuatan *website* dengan cara *coding* secara manual. *codeigniter* dilengkapi dengan berbagai pustaka yang dibutuhkan *developer* dalam hal pembuatan *website*. Daftar pustaka tersebut siap pakai untuk berbagai kebutuhan, seperti melakukan koneksi *database*, *email*, *session* dan *cookies* serta keamanan dan lainnya. *Framework Codeigniter* menggunakan konsep *MVC (Models-Views-Controllers)* dimana proses *scripting* dapat dipisahkan antara logika pembuatan kode dengan pembuatan *template website*. Pembagian *scripting* tersebut yaitu *Models* untuk basis data, *views* untuk tampilan pada halaman *website* dan *controllers* untuk logika aplikasinya.

## 2. Menentukan *database* Kandel Store

Pada tahap ini penulis diberikan *database* dari *web store* sebelumnya untuk melakukan pembagian kategori jenis produk dan ketentuan lainnya sebelum ditambahkan ke dalam *database* agar tidak ada pengulangan data yang tidak diperlukan dan agar menghemat *resource*.

## 3. Mendesain *Use-Case Diagram* untuk Kandel Store

Setelah *database* tersebut selesai dikategorikan dibuatlah *use-case diagram* untuk mempermudah dalam mendeskripsikan sistem yang akan dibuat nantinya. Melalui *Microsoft Visio* penulis membuat *use-case diagram* untuk mendapatkan requirements sistem dan menggambarkan hubungan antar sistemnya. *Use-case diagram* dibuat sebagai pemodelan terhadap *behaviour* sistem yang nantinya akan dibuat dan dibutuhkan juga untuk tahap analisis. *Use-case diagram* dapat mendeskripsikan interaksi antara satu atau lebih *actor (user)* yang terlibat dalam sistem yang dibuat. Fungsinya adalah untuk mengetahui apa saja yang ada di dalam sebuah sistem dan siapa saja yang menggunakan fungsi-fungsi di dalam sistem tersebut. Secara garis besar *use-case diagram* menjelaskan secara sederhana fungsi dari sistem dari sudut pandang *user*.



**Gambar 3.1.** Use-Case Diagram Kandel Store

Dalam diagram use-case yang telah dibuat, penulis membagi menjadi 4 aktor yaitu :

- USER → sebagai *website e-commerce*, pihak USER akan memperoleh fungsi-fungsi yang dibutuhkan dalam hal menjadi pembeli ataupun pengunjung *web store* ini. Fungsi seperti *register* sampai ke *update* pembelian produk diberikan untuk USER.
- ADMIN → bertugas sebagai pusat kontrol terhadap *web store* yang berjalan dan diberikan *privilege* mengatur fungsi dari USER dan *update stock* yang dilakukan USER.

- CS → merupakan *customer service* yang ada di dalam *website* untuk memudahkan USER dalam hal bertanya dan memberikan info yang lebih cepat.
- WH Staff → bertugas untuk melakukan *update stock* barang yang masuk maupun keluar agar *stock* barang yang tertera pada *website* selalu *update* dan juga sebagai penyalur barang ke pengiriman ke USER.

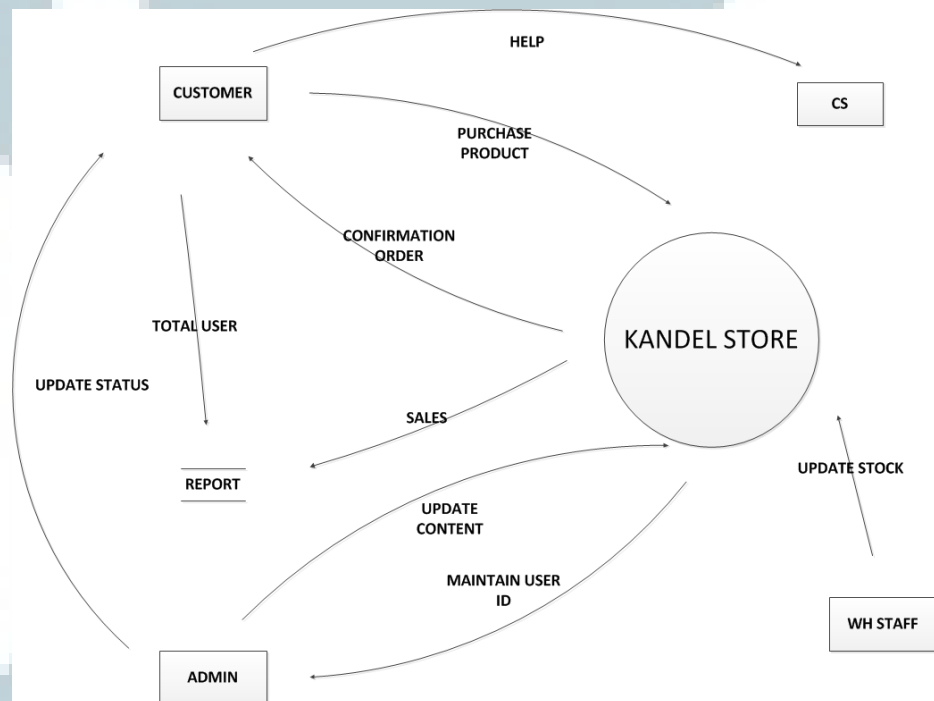
#### 4. Mendesain ERD Kandel Store

Dengan pemodelan menggunakan ERD dapat membantu mengorganisasikan data dalam suatu proyek ke dalam *entity-entity* dan menentukan hubungan antar *entity* tersebut. Dalam mendesain ERD terdapat 3 struktur umum dari ERD tersebut, yaitu *entity*, *attribute & relationship*. Berdasarkan *database* yang telah didapat sebelumnya, penulis menentukan *relationship* antar *entity* yang ada. Menggunakan *Microsoft visio* penulis mendapatkan *entity* yang sudah ditentukan dengan *primary key* pada masing-masing *entity*. ERD yang digambarkan terdapat di bagian lampiran pada halaman ke 7.

U  
M  
M  
N

## 5. Mendesain DFD Kandel Store (*Context Diagram*)

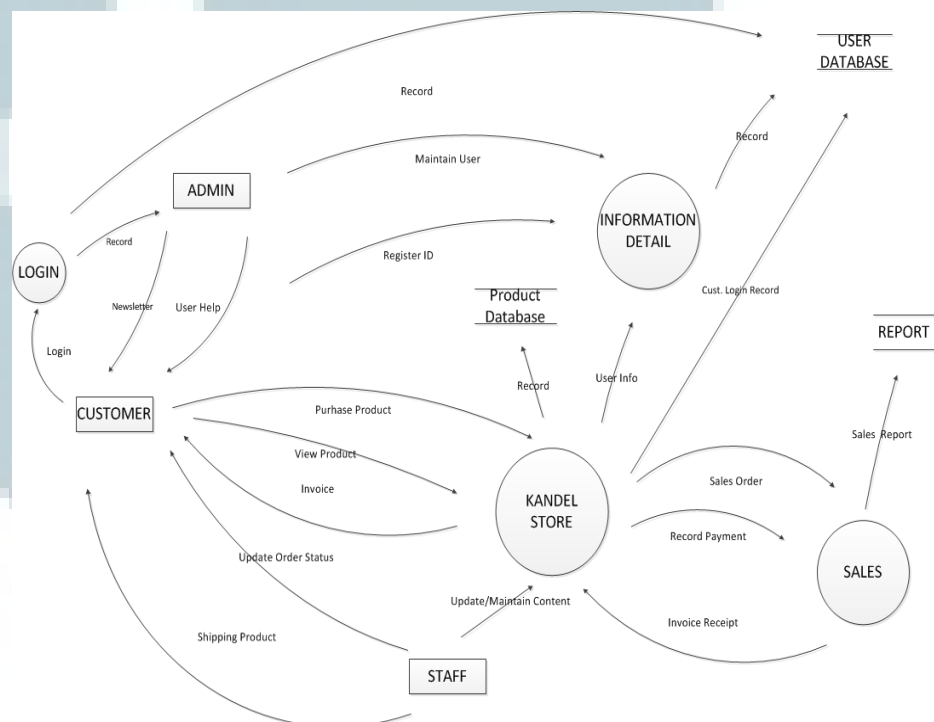
DFD (*Data Flow Diagram*) digunakan untuk menggambarkan pemodelan data pada sistem yang sudah ada atau sistem baru yang akan dikembangkan dengan cara menganalisa sistem yang akan dibuat secara visual sebagai suatu rangkaian aliran data yang saling berkaitan. Keuntungan dalam menggunakan DFD adalah memudahkan *user* yang tidak menguasai *programming* untuk bisa mengerti sistem yang akan dikerjakannya. DFD terdiri dari *context diagram* dan *level diagram*. *Context diagram* merupakan diagram yang terdiri dari proses yang menggambarkan jangkauan kerja suatu sistem dan mewakili seluruh proses yang terdapat di dalamnya. Biasanya hanya terdapat satu proses di dalam *context diagram*.



Gambar 3.2. Data Flow Diagram Kandel Store

## 6. Mendesain DFD Lv. 1 Kandel Store (Level Diagram)

*Level diagram* merupakan diagram yang menjelaskan proses apa yang ada di dalam diagram level sebelumnya. Dalam hal ini diagramnya ada pada tingkatan 1 atau DFD Level 1. DFD level 1 merupakan proses selanjutnya dari *context diagram* dimana pada tahap ini terjadi pemecahan proses dari *context diagram* yang memuat penyimpanan data dan menunjukkan proses-proses utama dari sistem yang sedang dibangun.

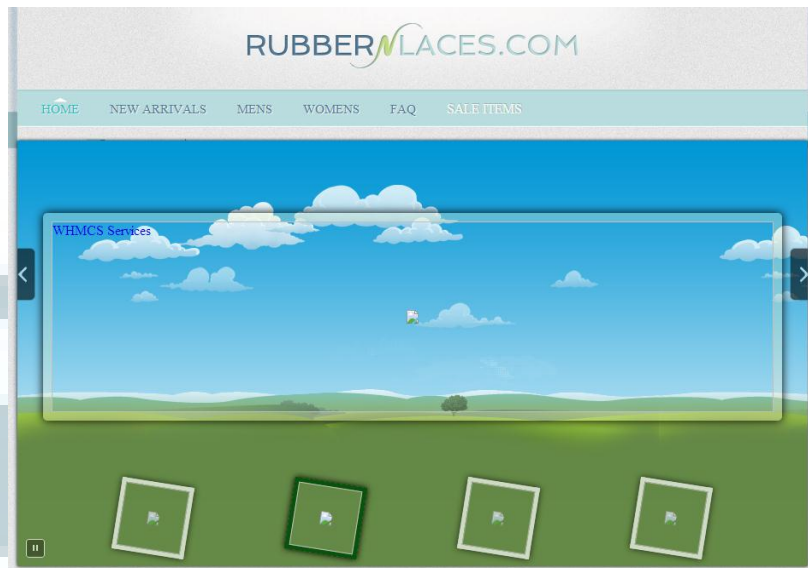


**Gambar 3.3.** Data Flow Diagram Level 1 Kandel Store

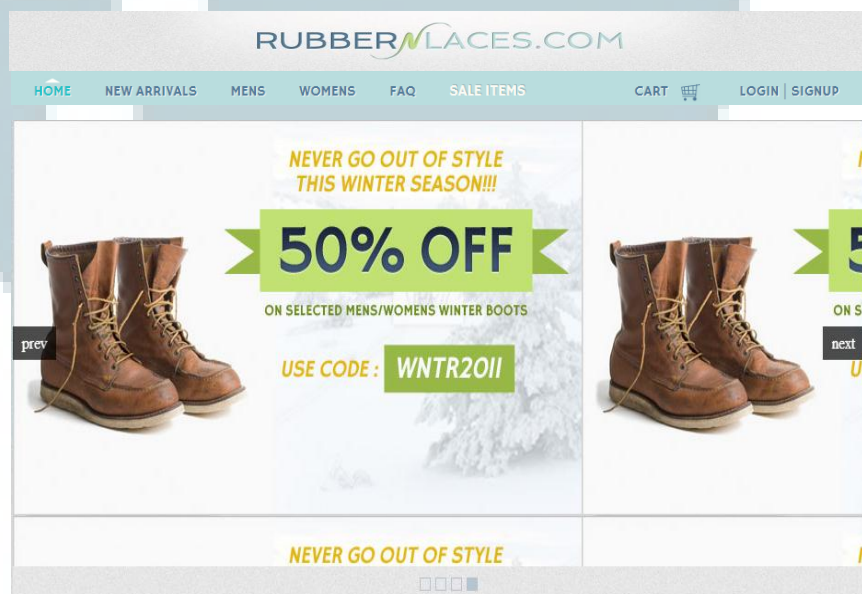
## 7. Editing Template Website

Penulis melakukan *editing* pada template *website* yang digunakan untuk *website* Kandel Store dimana pada proses *editing* ini berlangsung penulis melakukan perubahan di bagian *navigation bar* dan *content image slider*. Perubahan dari *parallax slider* menjadi *coin slider*.





**Gambar 3.4.** Parallax Slider (Slider di template sebelumnya)



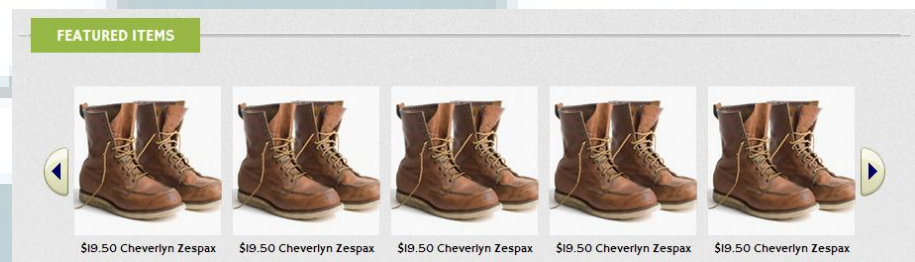
**Gambar 3.5.** Coin Slider (Slider di template baru)



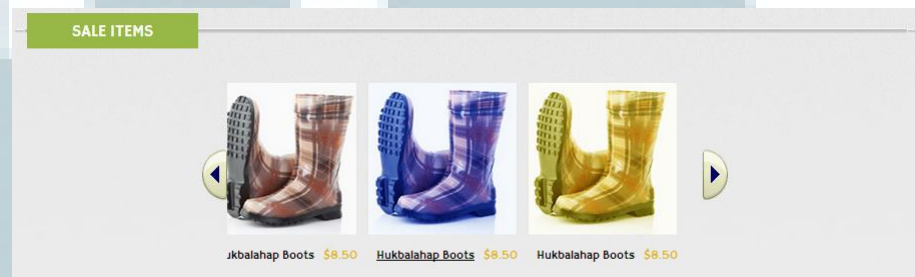
**Gambar 3.6.** Footer (Template Baru)

## 8. *Editing Javascript, JQuery Pada Template*

Menambahkan atribut *slider* di *middle* dan *bottom content* dengan *slider image* untuk bisa menggerakkan kolom *featured items* dan *sale items*.



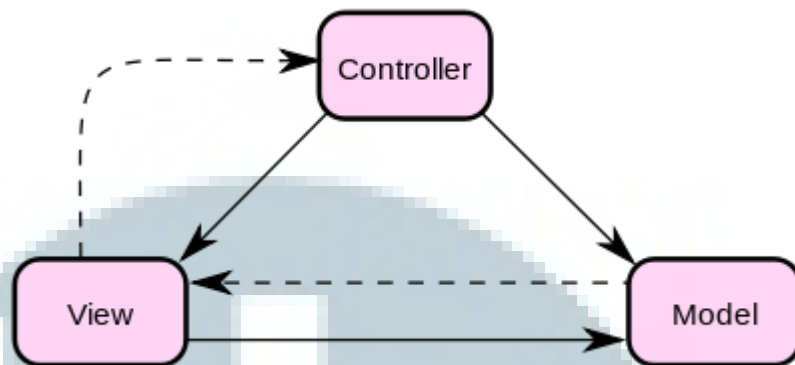
Gambar 3.7. *Featured Items*



Gambar 3.8. *Sale Items*

## 9. *Menggabungkan Page Website Untuk Seluruh Kategori*

Penulis melakukan *edit template website* dengan menggunakan *Codeigniter* agar lebih mudah dalam melakukan tahap *programming* karena *codeigniter* menggunakan pola MVC (*Models-Views-Controllers*) yang bertujuan untuk memisahkan antara proses, data dan tampilan. MVC merupakan sebuah metode dalam *programming* dengan memisahkan *model* (data) dari *view* (tampilan) dan cara melakukan *controller* (proses). MVC membantu dalam memisahkan komponen utama dalam proses *programming website/aplikasi* seperti proses *editing/programming*, *user interface* dan control dari *website*.



Gambar 3.9. Konsep MVC

Bagian dari MVC :

- a. **Model** → mewakili struktur data dari sistem aplikasi/web dan berisi fungsi pengolahan datanya melakukan koneksi ke basis data.
- b. **View** → bagian yang mengatur tampilan (*user interface*) ke *user*, dalam hal ini adalah *webpage*-nya. View berisi halaman *website* menggunakan HTML, CSS maupun JavaScript.
- c. **Controller** → bagian yang menjembatani *model* dan *view*. Berisi perintah untuk memproses data dan mengirimkannya ke *webpage*. Tugas dari *controller* sendiri menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data.

Urutan *request* yang berjalan dengan MVC adalah *user* yang berhubungan dengan *view* akan melakukan *request* dan diproses oleh *controller*. Validasi data nantinya akan terjadi di *controller* dan *Model* akan menyelesaikan *request*. Dari *model*, data akan dikirim kembali untuk di proses di *controller* dan ditampilkan di *view*.

#### 10. **Membuat Sign Up Form + JQuery Multistep Form**

Membuat tampilan dan *form sign up* yang interaktif dengan ditambahkannya *JQuery Multistep Form*. *Requirement* yang diinginkan adalah membuat *sign up form* dengan tampilan yang *real-time* tanpa perpindahan *page*. Tampilan dari *sign up form* ada di bagian lampiran halaman 1.

#### 11. **Membuat Login Form**

Membuat tampilan *login form* untuk *website Kandel Store*. Pada tahapan ini penulis membuat *login form* menggunakan HTML dan komposisi *login field* serta *button* melalui CSS. Tampilan dari *login form* ada di bagian lampiran halaman 2.

#### 12. **Membuat Checkout Form**

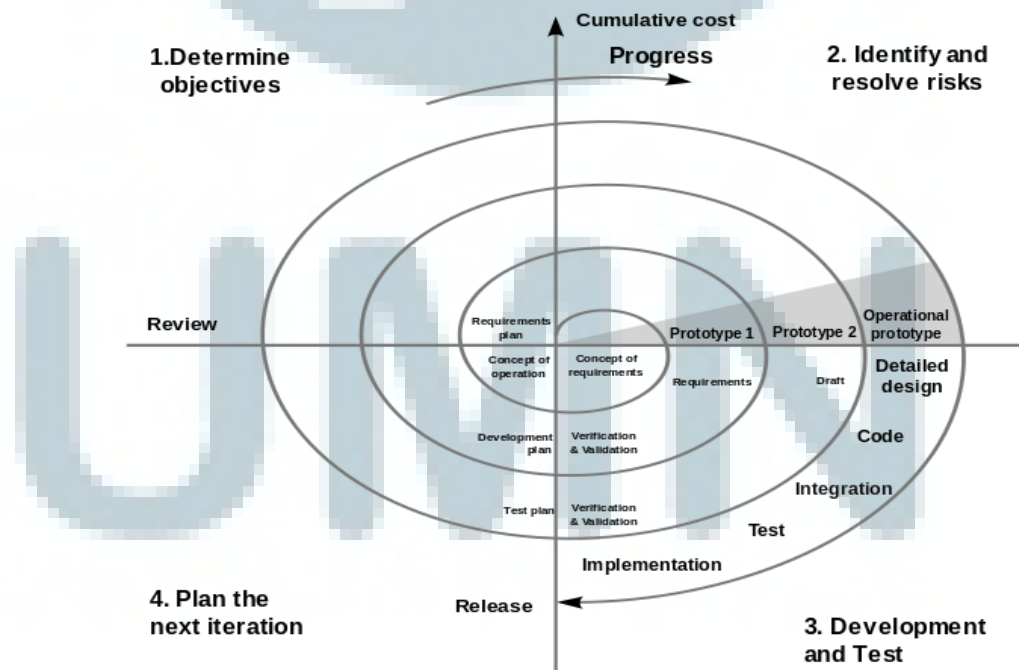
Membuat tampilan *checkout form* yang menampilkan fungsi *update* dan *checkout*. Penulis menggunakan HTML untuk membuat *checkout form* ini sesuai dengan *requirement* yang diinginkan yaitu menambahkan *update* dan *checkout*. Tampilan dari *checkout form* ada di bagian lampiran halaman 3.

#### 13. **Membuat Payment Process Form + JQuery**

Membuat tampilan untuk proses *checkout* dengan pilihan *Guest Checkout* atau *Register Account* dan ditambah menggunakan *smartwizard javascript*. Tampilan dari *payment process form* ada di bagian lampiran halaman 3.

### 3.3. Metode Perancangan Yang Digunakan

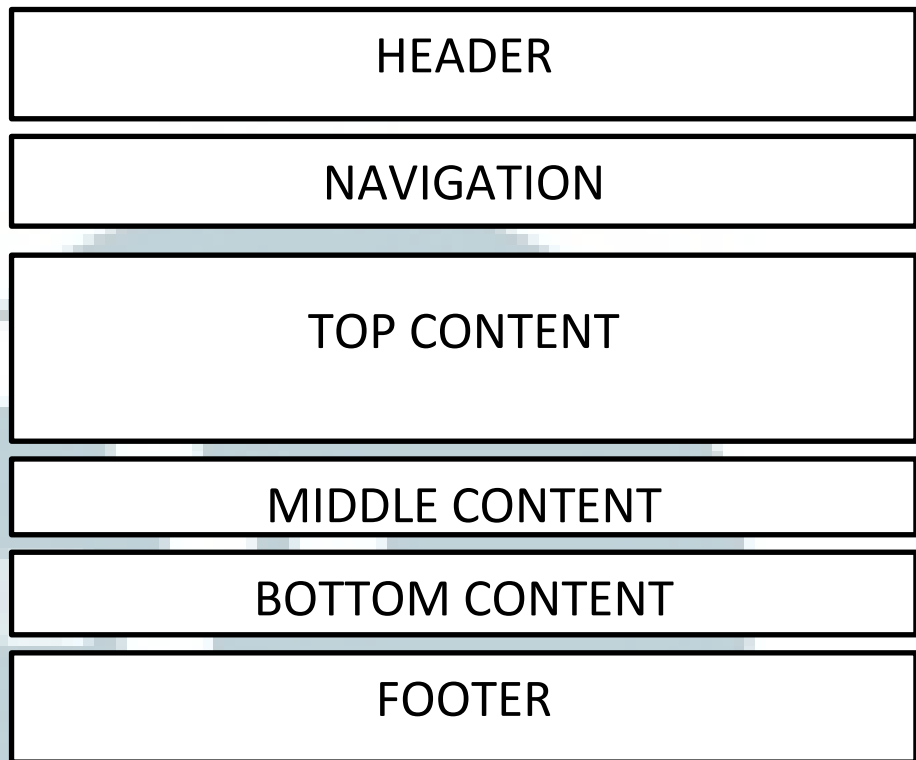
Dalam perancangan suatu sistem baru maupun *website* dibutuhkan metode tertentu sebagai bagian dari acuan untuk menyelesaikan pembuatan sistem atau *website*. metode yang digunakan dalam hal pembuatan *front-end website* ini adalah dengan metode *spiral*. Metode *spiral* merupakan suatu metodologi dalam pengembangan *software* yang merangkai sifat *iterative* dan *prototype* dengan cara kontrol dan aspek sistematis model sequensial linier. Metode ini pada awalnya diusulkan oleh Boehm. Metode ini memungkinkan developer dalam mengembangkan versi dari *software* sebelumnya menjadi lebih lengkap secara bertahap. Karakteristik menggunakan metode spiral ini adalah menggunakan prinsip iterasi dan setiap kali iterasi selalu diperhitungkan dengan manajemen risikonya. Setiap kali iterasi terjadi akan selalu dicek apakah sesuai dengan tujuan semula. Resiko-resiko yang akan muncul bila adanya perubahan akan dilihat alternatif lainnya yang sesuai. Selanjutnya menitikberatkan pada penanggulangan resiko-resiko yang dapat terjadi. Setiap resiko akan ditutup dengan pengekseskusan rencana.



Gambar 3.10. Metode Spiral

Dalam pengembangan *software & website* langkah – langkah yang harus diambil dalam metodologi *spiral* antara lain :

1. *Determine Objectives* → mencari tahu kebutuhan yang diperlukan dalam pengembangan *website* nantinya. Pada tahapan ini dalam pembuatan *website* dibutuhkan pencarian informasi sebanyak-banyaknya. Kebutuhan apa saja yang nantinya ada di *website* dan dapat ditampilkan ke dalamnya. Proses ini membutuhkan *user requirements* sebagai bentuk data apa yang diinginkan oleh *user* dan menjadi acuan kepada penulis untuk menterjemahkan ke *coding website* tersebut.
2. *Planning* → tugasnya adalah untuk mendefinisikan sumber daya yang sudah ada dengan ketepatan waktu perencanaan dan proyek informasi lain yang berhubungan.
3. *Identify and Resolve Risks* → pada tahap ini resiko-resiko apa saja yang akan ditemui dalam hal mengembangkan *website* sebelumnya menjadi lebih baru. Disertai juga bagaimana cara untuk menanggulangi resiko yang dapat terjadi nantinya.
4. *Development and Test* → proses *coding* merupakan penerjemahan kebutuhan dari *website* tersebut ke bahasa komputer. Dalam project yang dikerjakan penulis yaitu membuat *front-end website* PT Kandel menghasilkan sebagai berikut :
  - a. struktur tampilan web seperti *header, footer* dan *menu bar*. Dalam prosesnya masih menggunakan data *dummy* hanya sebagai acuan pembuatannya.



Gambar 3.11. Struktur Rancangan Tampilan



Gambar 3.12. Header



Gambar 3.13. Navigation bar

Disini proses *editing* yang berlangsung mengganti beberapa bagian. Seperti dibagian *footer* dengan menempatkan kolom untuk *social media*.

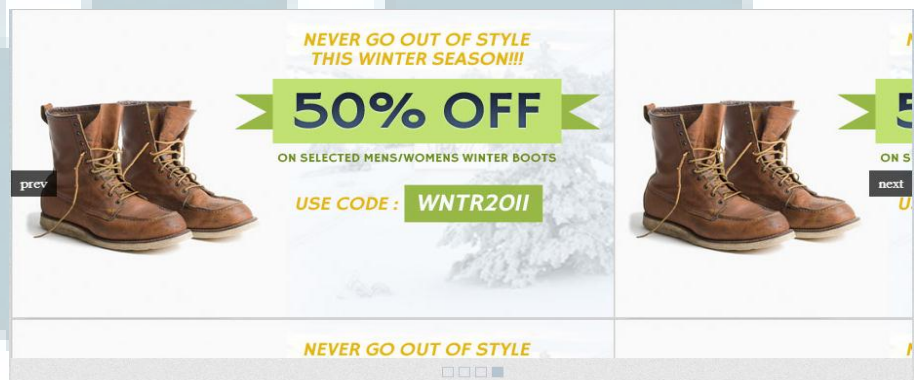


*Gambar 3.14. Footer*

Pada bagian footer ini dengan mengedit kolom *footer* tersebut, serta melebarkan bagian footer melalui CSS untuk mendapatkan ruang untuk kolom *social media*.

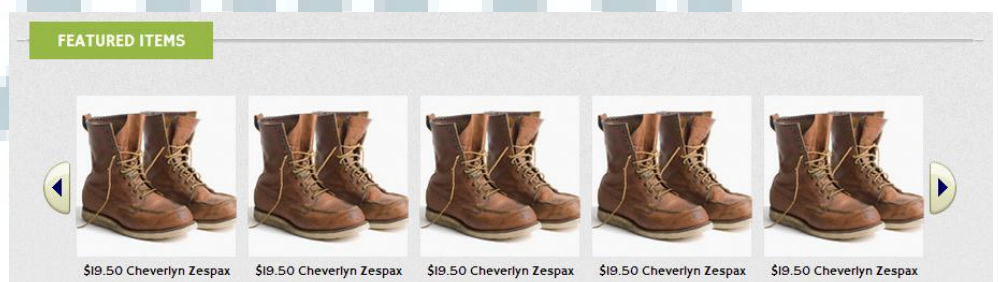
- b. Untuk selanjutnya dibagian *top content*, *middle content* serta *bottom content* lebih banyak menggunakan *javascript*.

Pada bagian *top content*, ditambahkan *image slider* dengan efek *coin slider*.



*Gambar 3.15. Coin slider pada top content*

Sedangkan dibagian *middle* dan *bottom content*, menggunakan *slider* biasa untuk melihat jenis barang dengan kategori *featured items* dan *sale items*.



*Gambar 3.16. Featured Items dengan slider*





**Gambar 3.17.** Sale Items dengan slider

- c. *Requirements* yang diinginkan oleh *user* salah satunya adalah membuat tampilan *website* yang baru dengan kesan interaktif dan tidak monoton. Menggunakan *javascript multistep form* agar memunculkan tampilan yang bergerak.

A screenshot of a 'Register Account' form. The form has a title 'Register Account' at the top. Below the title are three input fields. The first field contains the text 'andreyraharja' and has a label 'At least 6 characters.'. The second field is empty and has a label 'At least 6 characters. Use a mix of upper and lowercase for a strong password.'. The third field is empty and has a label 'Type Your Password Again.'. A red 'NEXT' button is located at the bottom right of the form. Below the form is a progress indicator showing '0% Complete'.

**Gambar 3.18.** Sign-up form

- d. Membuat tampilan *login form* untuk *website* Kandel Store. Pada tahapan ini penulis membuat *login form* menggunakan HTML dan komposisi *login field* serta *button* melalui CSS

**Login to Kandel Store**

Username or Email

Password

Remember me

[Login](#)

[Register](#)

[Forgot your password?](#)

© 2013 [PT. Kandel Multimedia](#)

*Gambar 3.19. Login form*

- e. *Requirements* lainnya yang diinginkan adalah tetap membuat adanya fungsi *update* pada fitur *checkout* dimana hal ini dapat memudahkan pembeli nantinya untuk melanjutkan memilih barang lainnya. Disini penulis membuat tampilan untuk *checkout* yang kemudian diserahkan kepada *web designer* untuk di edit kembali agar lebih menarik.

Remove	IMAGE	PRODUCT NAME	MODEL	QUANTITY	UNIT PRICE	TOTAL
<input type="checkbox"/>		Dummy Pict.	B 10120	<input type="text" value="1"/>	Rp200.000	Rp200.000

[Update](#)

[Continue Shopping](#)

**Sub Total : Rp200.000**

**Total : Rp200.000**

[CHECKOUT](#)

*Gambar 3.20. Checkout Form*

- f. Dengan menggunakan *smartwizard javascript* maka sistem *checkout* akan berupa *step by step* dimana jika pilihan *checkout* dipilih menggunakan *register account* maka akan dialihkan ke *login form*. Jika menggunakan *guest checkout* maka akan melanjutkan ke *step* selanjutnya untuk mengisi data-data calon pembeli.

**Checkout**

**1 Step 1**  
Checkout Options

**2 Step 2**  
Billing Details

**3 Step 3**  
Delivery Method

**4 Step 4**  
Payment Method

**5 Step 5**  
Confirmation Order

**Checkout Options**

**New Customer**

Checkout Options :

Register Account

Guest Checkout

Creating an account on Kandel Store will able to up to date your order status and track your orders

Previous Next Finish

**Gambar 3.21.** *Payment Process Form*

5. Testing program → tahapan dimana *website* sudah bisa dijalankan dan digunakan oleh *user* setelah melalui analisa, desain, dan *coding*. Pada tahap ini penulis tidak sempat melakukan pengujian *website* yang dibuat. Hal ini terjadi dikarenakan durasi waktu kerja magang sudah habis dan tidak melanjutkan sampai *website* berjalan (*Go Live*).

6. Plan the Next Iteration → tahapan akhir dari seluruh metode yaitu merencanakan kembali, namun membutuhkan evaluasi terlebih dahulu dari proyek sudah diselesaikan sebagai *feedback*.

### 3.4. Kelebihan dan Kekurangan Menggunakan Metode *Spiral*

Metode *spiral* memiliki kelebihan dalam membangun sistem dan *website*, antara lain :

1. Dapat disesuaikan agar *software/website* bisa dipakai dalam jangka waktu yang lama.
2. Cocok untuk pengembangan sistem dan *software* skala besar.
3. *User* dan *developer* dapat lebih mudah memahami dan bereaksi terhadap resiko yang terjadi karena *software* terus bekerja selama proses.
4. Menggunakan *prototype* sebagai mekanisme pengurangan resiko.
5. Mengikuti langkah-langkah dalam siklus dan memasukkannya ke dalam kerangka kerja iteratif.
6. Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi masalah.

Adapun kekurangan menggunakan metode ini antara lain :

1. Sulit untuk meyakinkan calon pelanggannya tentang pendekatan evolusioner ini bisa dikontrol.
2. Membutuhkan analisis resiko yang masuk akan dan dapat terjadi masalah jika resiko utamanya tidak ditemukan dan diatur.