

PERANCANGAN APLIKASI PEMESANAN MAKANAN (STUDI KASUS: RESTORAN ANAK NATBERRY) BERBASIS WEBSITE

Achmad Aditya Ashadul Ushud

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260

Email: achmad.aditya@budiluhur.ac.id

Abstrak

Perkembangan teknologi informasi yang semakin canggih memunculkan banyak ide kreatif dalam berbagai bidang. Salah satunya dalam bidang bisnis. Seseorang atau perusahaan dituntut untuk bisa kreatif agar bisa disukai atau diterima oleh masyarakat atau pasar. Penggunaan informasi teknologi banyak diterapkan untuk mendukung kreatifitas tersebut. Salah satu yang menggunakannya adalah tempat makan khusus anak atau restoran anak Natberry. Seiring banyaknya tempat usaha sejenis, pengelola Natberry merasakan keuntungan usahanya semakin menurun. Agar tidak kalah bersaing dan meningkatkan keuntungan, Natberry berusaha menggunakan konsep baru dalam menyajikan makanan kepada pelanggannya, salah satunya melalui informasi teknologi. Sebuah aplikasi pemesanan makanan berbasis *web* diaplikasikan melalui gawai seperti tablet. Dengan tampilan yang menarik, mudah dimengerti dan digunakan, aplikasi pemesanan ini diharapkan juga dapat digunakan oleh anak-anak yang menginginkan memesan sendiri makanannya. Aplikasi ini juga dapat menyarankan menu makanan yang sesuai dengan kebutuhan pelanggan. Konsumen yang datang memberikan respon yang positif atas penerapan aplikasi ini. Pada akhirnya aplikasi ini akan membawa dampak keuntungan bagi Natberry. Aplikasi ini dibuat dengan PHP, menggunakan *framework* Laravel dan desain *bootstrap* 4.

Kata kunci : *web, aplikasi pemesanan makanan, PHP, Laravel, bootstrap*

I. PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi informasi sudah sedemikian maju, dan masuk ke dalam kehidupan sehari-hari. Penggunaan teknologi informasi dapat dengan mudah dijumpai baik dalam skala kecil seperti rumah tangga hingga skala besar seperti perusahaan. Di rumah misalnya ada *smart TV*, mesin cuci, *microwave*, dan telepon seluler. Penerapan teknologi informasi yang semakin meluas ini, memunculkan ide-ide kreatif dalam pemanfaatannya. Salah satunya dalam bidang bisnis makanan.

Pemanfaatan teknologi informasi dalam bidang bisnis makanan dalam hal ini warung atau tempat makan atau restoran banyak digunakan hanya sebagai pencatat pesanan, sampai penghitungan transaksi keuangan. Ada beberapa restoran yang telah menggunakan informasi teknologi lebih dari itu, misalnya menampilkan menu untuk dipilih kepada pelanggan.

Natberry adalah salah satunya, merupakan sebuah tempat makan atau restoran yang dikhususkan untuk anak-anak yang berusia 6 bulan ke atas sampai 5 tahun. Restoran yang berada di daerah Tangerang ini mencoba menawarkan pengalaman baru kepada pengunjungnya dengan memanfaatkan teknologi informasi secara lebih menarik. Menyadari bahwa keuntungan yang didapat dari usahanya semakin menurun, salah satunya disebabkan karena bermunculan usaha yang sejenis yang menjadi kompetitornya, Natberry harus memikirkan ide kreatif agar tempat usahanya berbeda dari yang sudah ada. Karena pasarnya adalah anak-anak, Natberry mencoba memanfaatkan segmen pasar tersebut dengan menerapkan tema anak-anak pada tempatnya. Di tempat tersebut disediakan tempat bermain dan sekaligus belajar, sehingga anak-anak merasa nyaman dan betah. Lebih jauh lagi, Natberry menginginkan adanya sebuah aplikasi berbasis *web* yang diaplikasikan ke dalam *tablet* yang disediakan untuk pengunjung. Aplikasi ini berfungsi untuk mencatat makanan yang dipesan oleh pengunjung sekaligus merekomendasikan menu yang pas untuk pelanggan. Selama ini proses pemesanan makanan berlangsung dengan cara biasa, dimana orang tua melihat menu dalam bentuk

lembaran, lalu memesan melalui pelayan. Anak-anak hanya bersifat pasif, menerima makanan sesuai dengan pilihan orang tua.

Untuk itulah melalui penelitian ini, dirancang Aplikasi Pemesanan Makanan dalam sebuah *website* yang bisa memenuhi kebutuhan yang dimaksud di atas. Aplikasi ini bertujuan mendapatkan respon positif dari pengunjung yang akan berdampak dengan keuntungan bagi Natberry.

1.2. Perumusan Masalah

Adapun perumusan masalah dalam penelitian Perancangan Aplikasi Pemesanan Makanan Berbasis Web sebagai berikut :

1. Bagaimana membuat tampilan yang menarik dan mudah digunakan untuk anak-anak?
2. Bagaimana mendapatkan data pelanggan lalu memproses data tersebut untuk langsung dapat ditampilkan menu yang direkomendasi oleh sistem?
3. Perangkat lunak apa yang digunakan untuk membuat Aplikasi Pemesanan Makanan yang baik dan mudah digunakan?

2. METODOLOGI PENELITIAN

Metode yang digunakan dalam pembuatan Aplikasi Pemesanan Makanan ini adalah model *Waterfall*. Menurut Rosa A.S dan M. Shalahuddin (2011) metode ini memiliki empat tahapan di dalamnya, yaitu.

1. Analisa Kebutuhan Perangkat Lunak

Analisa kebutuhan perangkat lunak ini dilakukan melalui diskusi dan wawancara terhadap pengguna. Pengguna yang akan menggunakan aplikasi ini adalah pelanggan dan karyawan Natberry. Untuk itu dibutuhkan data kebutuhan sistem dari mereka. Pengguna tersebut dibagi menjadi dua seperti berikut:

a. Administrator

Administrator adalah pengguna yang mempunyai hak akses penuh ke dalam aplikasi. Administrator yang mengatur data pelanggan, produk makanan yang dijual, sampai transaksi keuangan.

b. Pengguna umum

Pengguna umum adalah pengguna yang menjadi pelanggan Natberry untuk melihat atau memesan makanan. Pengguna umum tidak mempunyai hak akses untuk mengubah atau menambah data, kecuali pesanan yang dibeli.

2. Desain

Setelah mendapatkan Analisa kebutuhan perangkat lunak, maka dibuatlah rancangan *website* sesuai dengan spesifikasi analisa yang telah didapat. Rancangan tampilan layar dibuat setelahnya.

3. Implementasi

Website Aplikasi Pemesanan Makanan ini dibangun dengan menggunakan *framework* Laravel yang berbasis bahasa pemrograman PHP, untuk desain menggunakan Bootstrap 4.0.

4. Pengujian

Proses pengujian system menggunakan *Black Box Testing*. Menurut Roger S. Pressman (2002) *Black Box Testing* berfokus pada persyaratan fungsional dari perangkat lunak. Misalnya sistem navigasi, *login*, penambahan, *update* dan penghapusan program, data dan fasilitas lainnya. Teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. Pengujian sampai batasan dimana sistem harus berjalan sesuai harapan sehingga bisa mencapai tujuan yang diinginkan.

5. Operasi dan Pemeliharaan (*Maintenance*)

Selanjutnya tahap operasi dan pemeliharaan yang merupakan tahap yang terakhir. Aplikasi dipublikasikan di internet secara *online*, dan mulai digunakan. Untuk menjaga aplikasi tetap berjalan dengan baik, perlu dilakukan pemeliharaan sistem secara berkala.

3. HASIL DAN PEMBAHASAN

3.1. PHP

PHP *Hypertext Preprocessor* atau disingkat dengan PHP adalah bahasa pemrograman yang digunakan untuk membuat *website*. Sifat PHP adalah *server side scripting* atau berbasis *server*, artinya untuk menjalankan PHP harus menggunakan *web server*. PHP dapat diintegrasikan dengan bahasa pemrograman *web* lain seperti HTML, JavaScript, CSS. Dengan menggunakan PHP suatu *website* bisa dibuat menjadi *powerfull* yang dinamis dengan disertai manajemen *database*-nya menurut Alexander (2013).

3.2. Framework

Menurut Budi (2015) *framework* merupakan istilah yang sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali. Sedangkan menurut Betha Sidik (2012), *framework* adalah kumpulan instruksi-instruksi yang dikumpulkan dalam *class* dan *function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang, sehingga dapat menghemat waktu. Bisa dikatakan bahwa *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks.

Kelebihan menggunakan *framework*:

1. Tersedia fasilitas-fasilitas yang umum dipakai sehingga tidak perlu membangun dari awal.
2. Mempermudah pembangunan sebuah aplikasi *web*.
3. Membutuhkan waktu yang lebih singkat dalam membangun sebuah aplikasi *web*.
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.
5. Mempermudah proses pemeliharaan karena sudah ada pola tertentu.

3.3. Laravel

Laravel merupakan salah satu *framework* PHP. Laravel menekankan pada kesederhanaan dan fleksibilitas pada desainnya untuk membangun sebuah web yang dinami. Pada tahun 2013 dilakukan survei oleh situs Sitepoint.com yang memunculkan Laravel sebagai *framework* PHP yang paling populer. Tahun 2014 Laravel dinobatkan sebagai *framework* terbaik.

Laravel mempunyai keunggulan fitur seperti di bawah ini:

1. *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular dan telah tersedia berbagai *bundle* untuk digunakan ke dalam aplikasi.
2. *Eloquent ORM*, merupakan penerapan PHP lanjutan dari pola "*active record*". Terdapat metode internal untuk mengatasi kendala hubungan antara obyek dalam *database*.
3. *Application logic* merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* atau sebagai bagian dari deklarasi *route*.
4. *Reverse routing*, mendefinisikan hubungan antara *link* dan *route*, sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *link* yang dibuat dengan menggunakan nama-nama dari *route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
5. *Restful controllers*, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.

6. *Class auto loading*, menyediakan otomatis *loading* untuk *class-class* PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang yang tidak perlu.
7. *View composers* adalah kode unit *logical* yang dapat dieksekusi ketika sebuah *view* di-*load*.
8. *IoC Container* memungkinkan untuk obyek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik.
9. *Migrations* menyediakan versi sistem kontrol untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan dalam basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
10. Unit *Testing* mempunyai peran penting dalam *framework* Laravel, dimana unit *testing* ini mempunyai banyak tes untuk medeteksi dan mencegah regresi. Unit *testing* ini dapat dijalankan melalui utilitas “*artisan command-line*”.
11. *Automatic pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

3.4. Bootstrap

Bootstrap adalah framework yang mempunyai fungsi utama dalam mendesain tampilan. Menurut Husein Alatas, (2015) Bootstrap merupakan *framework* untuk membangun desain web secara responsif. Dengan menggunakan bootstrap tampilan *website* nanti akan menyesuaikan ukuran layar dari *browser* yang digunakan baik di *desktop*, *tablet* ataupun *mobile device*. Sehingga, pengguna akan mendapatkan pengalaman yang lebih baik dalam berselancar tanpa mempertimbangkan perangkat apa yang harus digunakan.

3.5. Analisa Kebutuhan Perangkat Lunak

Kebutuhan Pengguna Administrator

1. Melakukan *login* ke dalam *website*
2. Menginput data makanan
3. Menginput data admin dan pelanggan
4. Mengubah data makanan
5. Mengubah data admin dan pelanggan
6. Menghapus data pelanggan

3.6. Desain

1. Rancangan Tampilan Awal

Tampilan awal pada *website* ini adalah bagian *header* yang berisi judul *web*, kotak pencarian, menu *login* dan notifikasi kepada pengguna bila sudah *login*. Ditunjukkan dengan gambar 1 di bawah.

Header Menu		
Logo		
Member	Daftar	Guest
Footer		

Gambar 1. Rancangan tampilan awal

Menu *Member* diperuntukkan untuk *member* yang telah terdaftar. Bila menu ini dipilih, maka akan diarahkan ke *form login* untuk mengisi data *username* dan *password*. Sedangkan untuk menu *Daftar* digunakan untuk mendaftarkan *member* baru. Menu *Guest* berfungsi untuk memulai pemesanan menu tanpa harus mendaftar sebagai *member*.

2. Rancangan Tampilan *Member*

Menu *Member* pada tampilan awal berfungsi untuk masuk ke halaman *member*. Halaman ini dikhususkan untuk pelanggan yang sudah menjadi *member*. Pelanggan yang sudah menjadi *member* mempunyai keuntungan dibanding dengan yang bukan *member*. Misalnya bila ada promo tertentu khusus *member*, selain itu data *member* khususnya anaknya telah tersimpan, sehingga aplikasi telah mengetahui menu yang sesuai atau yang paling disukai oleh *member* tersebut. Tampilan masuk *member* ditunjukkan pada gambar 2 di bawah ini.

Header Menu			
<table style="margin: auto; border: none;"> <tr> <td style="border: 1px solid black; padding: 5px;">Your Username</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Enter Password</td> </tr> <tr> <td style="border: 1px solid black; padding: 10px; margin-top: 10px;">LOGIN</td> </tr> </table>	Your Username	Enter Password	LOGIN
Your Username			
Enter Password			
LOGIN			
Footer			

Gambar 2. Rancangan tampilan *member*

3. Rancangan Tampilan *Form* Daftar

Bila pelanggan ingin melakukan pendaftaran *member*, maka pada menu *Daftar* ditampilkan form pendaftaran. Pada gambar 3 di bawah ini ditampilkan rancangan layarnya.

Header Menu		
Pendaftaran		
Nama Lengkap	Tempat Lahir	
No Hp/WA	Tanggal Lahir	
Email	Alamat Lengkap	
Security		
Password	Ulangi Password	
<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 5px;">CANCEL</td> </tr> </table>		CANCEL
CANCEL		
Footer		

Gambar 3. Rancangan tampilan *form* pendaftaran *member*

4. Rancangan Tampilan *Guest*

Ketika pilihan *guest* dipilih, maka pengguna akan diarahkan pada tampilan pertanyaan-pertanyaan yang mengarah pada proses pemesanan menu makanan. Daftar pertanyaan yang diberikan merupakan bagian dari proses pemesanan makanan, sistem akan menentukan menu makanan yang cocok dengan pelanggan dari jawaban-jawaban yang diberikan. Berikut daftar pertanyaan yang diajukan:

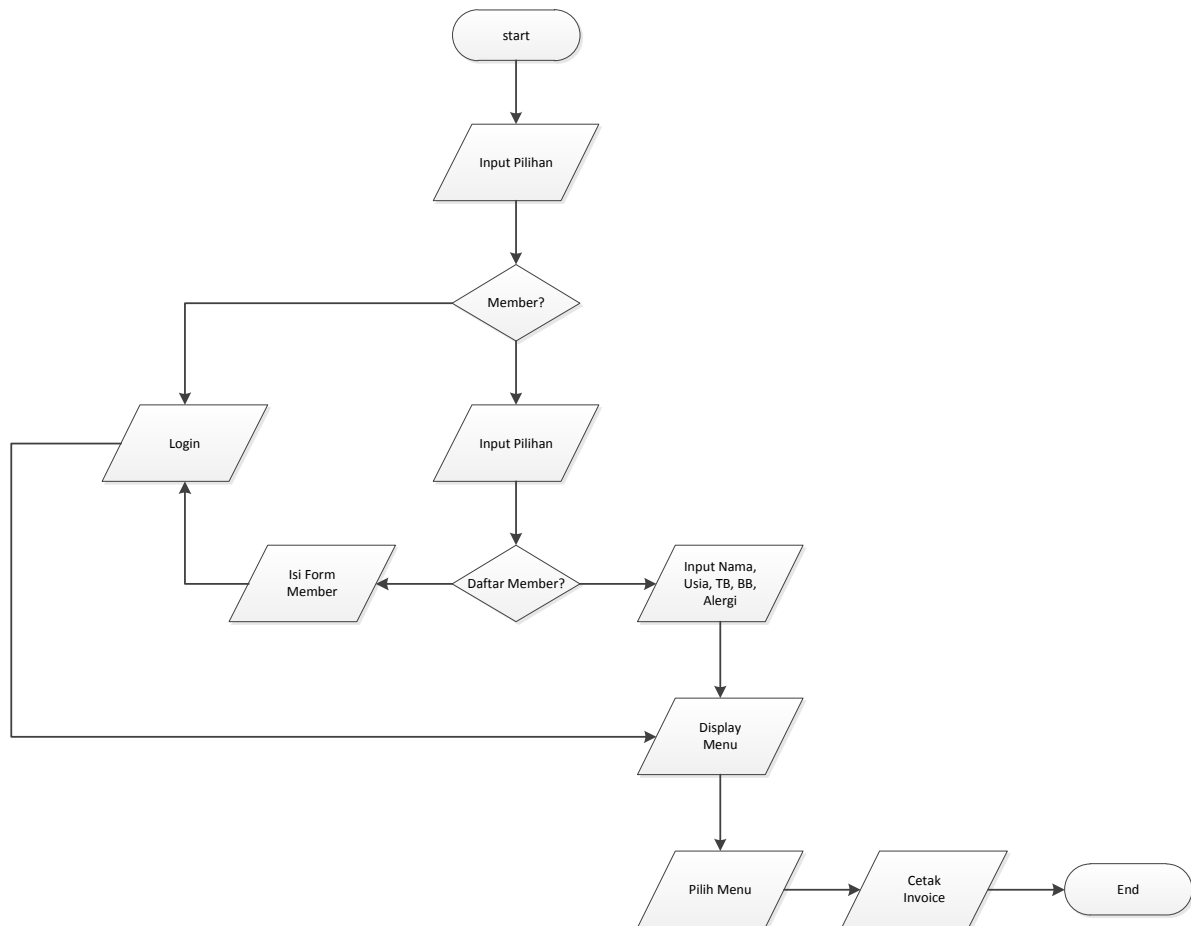
- a. Siapa namamu?
- b. Berapa usiamu?
- c. Apakah kamu punya alergi makanan?
- d. Apa yang membuatmu alergi?

Dari jawaban yang didapat di atas, sistem akan mencocokkan menu yang sesuai berdasarkan usia dan riwayat alergi. Sehingga pelanggan tidak perlu khawatir mengenai komposisi dari makanan yang diberikan.

3.7. Perancangan *Flowchart*

Alur proses aplikasi dimulai dari menerima *input* dari *user*. Ada dua jenis *user* yang mulai menggunakan aplikasi ini, *user* yang sudah menjadi *member* dan *user* yang belum menjadi *member*. *User* yang menjadi *member* diarahkan untuk *login* sebelum dapat melakukan pemesanan makanan. Sedangkan *user* yang belum *member* diberikan dua pilihan, mendaftar menjadi *member* atau menjadi *guest* dan langsung melakukan proses pemesanan makanan.

Untuk proses pemesanan makanan, *user* yang menjadi *guest* akan ditanya data yang akan dikonsumsi makanan, dengan tujuan memberikan rekomendasi menu yang cocok. Sementara untuk *member*, langsung diberikan rekomendasi menu dengan beberapa pilihan. Hal ini karena *member* sebelumnya telah memberikan data yang akan dikonsumsi makanan. Proses pemesanan makanan ini digambarkan melalui *flowchart* di bawah ini.



Gambar 4. Rancangan flowchart

3.8. Implementasi

Aplikasi Pemesanan makanan ini menggunakan *framework* Laravel PHP dan Bootstrap versi 4. Aplikasi ini sudah dipublikasi pada domain resto.natberry.id.

1. Implementasi Tampilan Awal

Ketika pertama kali membuka resto.natberry.id, maka tampilan awal adalah seperti gambar di bawah ini:

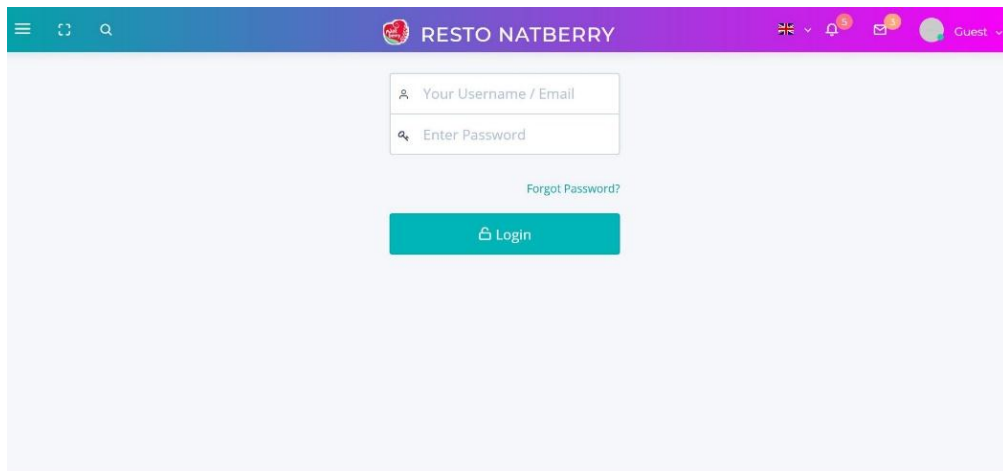


Gambar 5. Implementasi tampilan awal

Pengguna diberikan pilihan sebelum memulai pesanan. Menu *Member* apabila pengguna adalah *member* yang pernah mendaftar sebelumnya. Menu *Daftar* bila pengguna ingin mendaftar sebagai *member* sebelum melakukan pemesanan makanan. Menu *Guest* bila pengguna bukan *member* dan tidak ingin mendaftar, tapi langsung melakukan pemesanan makanan.

2. Implementasi Tampilan Member

Pengguna yang telah mendaftar sebagai *member*, diharuskan *login* terlebih dulu sebelum melakukan pemesanan makanan melalui menu *Member*.

The screenshot shows the login page for 'RESTO NATBERRY'. At the top, there is a navigation bar with a menu icon, a search icon, the restaurant name 'RESTO NATBERRY', and user profile information for a 'Guest'. The main content area features a login form with two input fields: 'Your Username / Email' and 'Enter Password'. Below these fields is a 'Forgot Password?' link and a teal 'Login' button with a lock icon.

Gambar 6. Implementasi tampilan *form login member*

3. Implementasi Tampilan Daftar

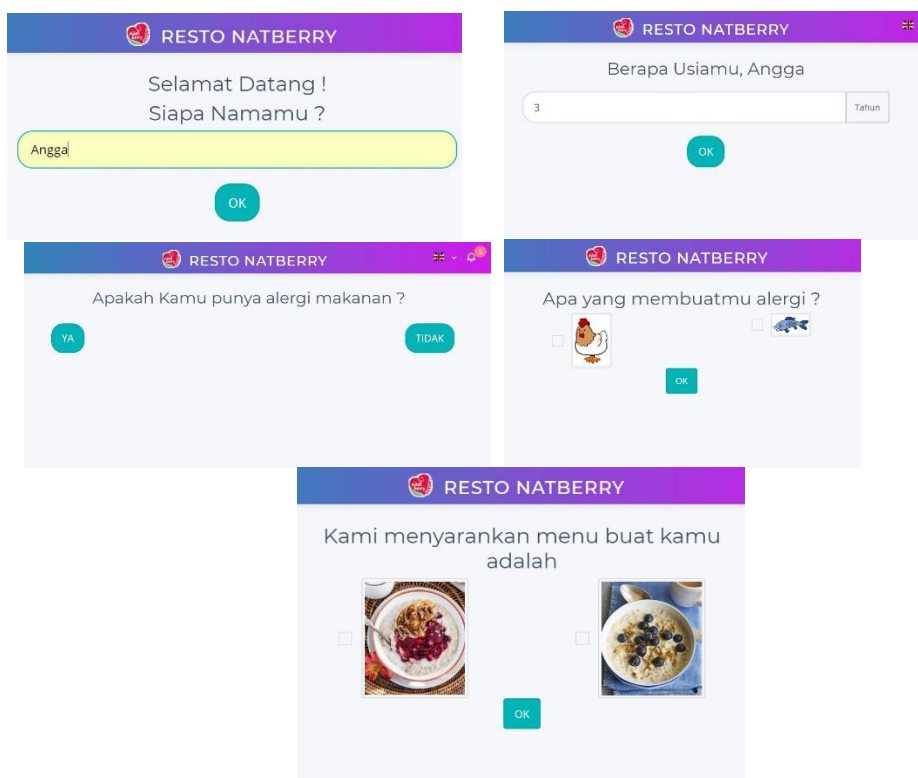
Bagi pengguna yang ingin mendaftar sebagai *member*, menu *Daftar* menjadi pilihan di awal tampilan. Pengguna diminta untuk mengisi *form* sebagai *member* seperti tampilan gambar 6 di bawah ini.

The screenshot shows the registration page for 'RESTO NATBERRY'. The page title is 'Pendaftaran'. The form is divided into two columns. The left column contains fields for 'Nama Lengkap' (with placeholder 'nama'), 'No HP/WA' (with placeholder 'no hp/wa'), and 'Email' (with placeholder 'email'). The right column contains fields for 'Tempat Lahir' (with placeholder 'tempat lahir'), 'Tanggal Lahir' (with placeholder 'dd----yyyy'), and 'Alamat Lengkap' (with placeholder 'alamat lengkap').

Gambar 7. Implementasi tampilan *form daftar member*

4. Implementasi Tampilan Pertanyaan dan Rekomendasi Menu

Di dalam aplikasi ini, pengguna yang akan memesan makanan akan disuguhkan menu yang direkomendasikan secara otomatis oleh sistem. Aplikasi akan mengetahui menu yang sesuai dengan keinginan pelanggan melalui jawaban-jawaban yang diberikan oleh pengguna melalui pertanyaan yang diberikan oleh sistem. Berikut tampilan pertanyaan dan rekomendasi menu tersebut.



Gambar 8. Implementasi tampilan pertanyaan dan rekomendasi menu

Penelitian ini telah menghasilkan sebuah perangkat lunak berupa *website* Aplikasi Pemesanan Makanan yang bisa diakses di resto.natberry.id.

3.9. Pengujian

Pengujian perangkat lunak ini dilakukan menggunakan data uji dari administrator selaku pengguna khusus.

1. Pengujian Login

Berikut tabel pengujian *login* administrator.

Tabel 1. Pengujian *login* administrator

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Username & Password</i> Administrator	Masuk ke halaman administrator dengan hak akses administrator	Berhasil masuk ke halaman administrator dengan hak akses administrator	[x] Diterima [] Ditolak

Tabel 2. Pengujian *login salah*

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Username & Password Tak terdaftar</i>	Tidak dapat masuk dan muncul peringatan kesalahan	Tidak dapat masuk dan muncul peringatan kesalahan	[x] Diterima [] Ditolak

2. Pengujian Input, Edit dan Delete Data

Berikut tabel pengujian *input*, *edit* dan *delete* data oleh administrator.

Tabel 3. Pengujian *input* data

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Input</i> data dan detailnya	Data masuk ke dalam <i>database</i> dan tampil di halaman <i>web</i>	Data ada di <i>database</i> dan tampil di <i>web</i>	[x] Diterima [] Ditolak

Tabel 4. Pengujian *input* data *member*

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Input member</i>	Data <i>member</i> masuk ke dalam <i>database</i>	Data <i>member</i> ada di <i>database</i>	[x] Diterima [] Ditolak

Tabel 5. Pengujian *edit* data

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Edit</i> data dan detailnya	Data berubah di dalam <i>database</i> dan perubahan tampil di halaman <i>web</i>	Data berubah di <i>database</i> dan perubahan tampil di <i>web</i>	[x] Diterima [] Ditolak

Tabel 6. Pengujian *edit* data *member*

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Edit member</i>	Data <i>member</i> berubah di dalam <i>database</i>	Data <i>member</i> berubah di <i>database</i>	[x] Diterima [] Ditolak

Tabel 7. Pengujian *delete* data

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Delete</i> data dan detailnya	Data hilang dari dalam <i>database</i> dan tidak tampil lagi di halaman <i>web</i>	Data hilang dari <i>database</i> dan tidak tampil di <i>web</i>	[x] Diterima [] Ditolak

4. SIMPULAN

Setelah dilakukan perancangan, pembuatan, pengujian dan analisa maka penelitian ini menyimpulkan beberapa hal terkait dengan rumusan masalah perancangan Aplikasi Pemesanan Makanan pada Restoran Natberry sebagai berikut :

1. Aplikasi pemesanan makanan berbasis web untuk Natberry ini telah disesuaikan desain tampilannya untuk dapat digunakan oleh anak-anak dengan tampilan yang sederhana, menarik dan mudah digunakan.
2. Pembuatan aplikasi pemesanan makanan ini dapat memenuhi kebutuhan Restoran Natberry dalam memberikan pengalaman baru kepada pelanggan sekaligus mencatat data dan transaksi melalui dengan menyuguhkan aplikasi untuk digunakan oleh pelanggan. Selain itu juga memudahkan pelanggan memilih menu yang tepat untuk anaknya.
3. Dengan menggunakan *framework* Laravel dan Bootstrap 4.0 sebagai *tools*, maka pembuatan *website* Aplikasi Pemesanan Makanan ini menjadi lebih mudah dan cepat, sehingga dapat langsung diaplikasikan ke dalam program.

DAFTAR PUSTAKA

Alexander. 2013. “*Web Programming Power Pack*”. Jakarta: Mediacom

Betha Sidik. 2012. “*Pemrograman Web dengan PHP*” Bandung : Informatika.

Budi, Raharjo. 2015. “*Belajar Otodidak Framework CodeIgniter*”. Bandung: Informatika.

Husein Alatas. 2015. “*Proyek Membangun Responsive Web Design dengan Bootstrap 3 dan 4*”. Yogyakarta: Loko Media.

Pressman, Roger S. 2002. “*Rekayasa Perangkat Lunak Pendekatan Praktisi*”. Yogyakarta : Andi.

Rosa A.S, dan M. Shalahuddin. 2011. “*Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Obyek)*”. Bandung: Modula.