Perancangan dan Pembangunan Aplikasi *Electronic Menu* Restoran

Finsa Nurpandi¹, Diki M Rosid Liki²
Program Studi Teknik Informatika
Universitas Suryakancana
finsa@unsur.ac.id¹ dikymury@gmail.com²

Abstract

Electronic Menu or abbreviated as eMenu is an application that designed to improve the performance of service processes in a business sector, especially restaurants. eMenu aims to change the food ordering service process that was originally traditional using paper, into web-based digital media. Not only handling food ordering services, eMenu also handles almost all business processes in a restaurant such as menu management, managing information on orders received by waiters and chef, payment transactions, to sales reports. To handle payment transaction, eMenu using third party application payment gateway, called Duitku. eMenu is built using Single-Page Application concept that use AJAX calls which allows time efficiency in displaying the required data. By using the Single-Page Application, the browser does not reload the entire requested page but only part of data is displayed dynamically. The Single-Page Application uses Vue.js as Front-end stack, and Laravel to manage Back-end. eMenu application is responsive design, so that user can access via smartphone or desktop computer with properly.

Keywords: *electronic menu, eMenu, single-page application, vue.js, laravel*

Abstrak

Electronic Menu atau disingkat eMenu merupakan aplikasi yang dibuat untuk meningkatkan kinerja proses pelayanan di suatu bidang usaha khususnya restoran. eMenu bertujuan untuk mengubah proses pelayanan pemesanan makanan yang awalnya tradisional menggunakan kertas, menjadi media digital yang berbasis web. Tidak hanya menangani pelayanan pemesanan makanan saja, eMenu juga menangani hampir keseluruhan proses bisnis di suatu restoran seperti pengelolaan menu, pengelolaan informasi pesanan yang diterima pelayan maupun penyaji, transaksi pembayaran, hingga laporan penjualan. Untuk menangani transaksi pembayaran, didukung oleh aplikasi pihak ketiga yaitu Duitku. eMenu dibangun dengan menggunakan konsep Single-Page Application yang dengan menggunakan AJAX calls memungkinkan efisiensi waktu dalam menampilkan data yang dibutuhkan. Karena dengan menggunakan Single-Page Application, browser tidak memuat ulang seluruh halaman yang diminta tetapi hanya sebagian data saja yang ditampilkan secara dinamis. Single-Page Application ini menggunakan Vue.js sebagai stack Front-end, dan Laravel untuk mengelola Back-end. Aplikasi eMenu bersifat responsive, sehingga pengguna bias mengakses melalui smartphone maupun komputer desktop.

Kata kunci: *electronic menu, eMenu, single-page application, vue.js, laravel*

1. PENDAHULUAN

Banyak restoran yang masih melakukan kegiatan pelayanan secara manual. Seperti pelayanan pemesanan makanan hingga pembayaran makanan masih dilakukan secara manual. Dalam hal pelayanan pemesanan makanan, kebanyakan restoran masih menggunakan menu biasa dan pelanggan rata-rata diberikan sehelai kertas untuk mencatat pesannya. Sehingga proses pelayanannya masih membutuhkan pelayan sepenuhnya. Walaupun proses pembayaran dikasir sudah menggunakan alat khusus kasir, tetapi sebagian besar proses bisnis pada restoran masih ditangani secara manual.

Dengan perkembangan teknologi saat ini yang berkembang pesat, mendorong setiap area bidang usaha untuk terus mengikuti perkembangan teknologi. Setiap bidang usaha yang tidak dapat beradaptasi, akan

mengalami kesulitan dalam mengembangkan usahanya. Selain perkembangan teknologi ini dapat digunakan untuk mempromosikan kegiatan usahanya, teknologi dapat digunakan untuk meningkatkan efektifitas kerja pada setiap proses bisnisnya dan melakukan efisiensi terhadap waktu dan biaya. Khususnya untuk usaha restoran, penerapan teknologi informasi biasanya digunakan untuk mempermudah proses pemesanan makanan dan pembayaran karena kedua proses tersebut yang berhubungan langsung dengan pelanggan.

Penerapan teknologi diharapkan dapat meningkatkan efektifitas pada proses bisnis yang dijalankan. Pada bidang usaha restoran, penggunaan teknologi diharapkan dapat mempermudah dalam proses pemesanan makanan, penyampaian informasi pesanan pelanggan, proses pembayaran yang lebih mudah, dan pembuatan laporan penjualan yang lebih praktis dan mengurangi kesalahan. Berdasarkan pada kebutuhan tersebut, penelitian ini akan

membahas tentang bagaimana merancang dan membangun aplikasi Elektronik Menu (eMenu) yang dapat diterapkan pada suatu restoran. Sehingga aplikasi ini diharapkan dalam meningkatkan efektifitas dan efisiensi pada proses bisnis yang dijalankan.

1.1 eMenu

Electronic Menu (eMenu) merupakan sebuah aplikasi yang digunakan untuk memudahkan pemilik restaurant dalam menjalankan proses bisnisnya dan juga mempermudah proses transaksi dengan menghilangkan proses manual seperti penggunaan kertas pada umumnya dan beralih menggunakan media digital seperti smartphone ataupun komputer [1].

eMenu menyediakan informasi tambahan tentang menu yang tersedia, dengan gambar yang interaktif dirasa akan lebih baik dibandingkan dengan menggunakan kertas menu tradisional. Penggunaan eMenu berfungsi untuk mengurangi kesalahan dalam penulisan pesanan yang dilakukan oleh pelayan. Sehingga koki yang berada di dapur tidak lagi bingung karena mendapatkan informasi yang sangat jelas mengenai pesanan tamu.

Berikut di bawah ini merupakan salah satu contoh kerangka kerja pada aplikasi E-restaurant Management System yang diajukan oleh Soon Nyeon Cheong, dkk [2].

Gambar 1: Kerangka Kerja E-Restaurant Management System [2]

Terdapat juga kerangka sistem yang diajukan oleh Kitsiri Chongchiang, dkk [3] membahas tentang aplikasi pelayanan restaurant dimulai dari menangani antrian pelanggan, memanggil antrian, menangani pesanan, memantau status pesanan, mengatur urutan pembuatan makanan, mengatur pelayanan pemesanan, melakukan kalkulasi terhadap pesanan, dan mengelola pembayaran. Kerangka system dapat dilihat pada Gambar 2.

Gambar 2: *One Stop Restaurant Service Application* [3]

1.2 Single-Page Application

Single-Page Application (SPA) merupakan aplikasi web yang hanya memuat satu halaman HTML dan secara dinamis memperbaharui halaman tersebut sesuai dengan interaksi pengguna dengan aplikasi tersebut [4].

SPA menggunakan AJAX dan HTML5 untuk membuat aplikasi web yang responsive. Karena menggunakan Javascript, artinya SPA lebih berfokus pengembangannya ke area *Front-end*. Sudah banyak Framework Javascript yang mendukung pengembangan SPA, seperti React.js, Angular.Js, dan Vue.Js.

Pada aplikasi web tradisional, setiap kali permintaan dari pengguna yang berinteraksi dengan aplikasi akan memanggil server untuk menjalankan perintah. Server akan menampilkan halaman HTML baru, dengan begitu halaman pada browser harus dimuat ulang. Pada SPA, setelah memuat ulang halaman pertama kali, semua interaksi dengan server dilakukan melalui AJAX. AJAX mengembalikan data menggunakan format JSON, dan aplikasi menggunakan data JSON untuk memperbaharui halaman secara dinamis tanpa memuat ulang halaman. Pada gambar di bawah ini menjelaskan mengenai perbedaan aplikasi web tradisional dengan SPA.

Gambar 3: Aplikasi Web Tradisional dan SPA [4]

2. METODE PENELITIAN

Pada pembangunan aplikasi ini, metode pengembangan perangkat lunak menggunakan metode *Waterfall*. Menurut Pressman [5], metode *Waterfall* adalah model klasik yang bersifat sistatis, berurutan dalam membangun perangkat lunak. Berikut gambar di bawah ini adalah gambaran bagaimana pembuatan sebuah perangkat lunak dari sebuah model *Waterfall*.

Gambar 4: Model Waterfall [5]

Berikut di bawah ini adalah tahapan-tahapan dalam metode *Waterfall*.

a. Communication

Tahapan ini, peneliti akan melakukan observasi mengenai kebutuhan perangkat lunak yang akan dibangun. Yang dihasilkan pada proses ini adalah daftar kebutuhan perangkat lunak beserta tujuan akhir dari perangkat lunak.

b. Planning

Tahapan ini peneliti melakukan perencanaan mengenai jadwal pelaksanaan pembangunan perangkat lunak. Tidak hanya menghasilkan jadwal pelaksanaan, biasanya pada tahapan ini juga akan menghasilkan rincian biaya pelaksanaan pembangunan perangkat lunak.

c. Modeling

Tahapan ini, peneliti akan melakukan analisis dan merancang sistem. Pada pembangunan perangkat lunak ini, akan menggunakan UML untuk perancangan perangkat lunaknya.

d. Construction

Tahapan ini merupakan lanjutan dari tahapan sebelumnya. Tahapan ini melakukan implementasi terhadap hasil perancangan dengan melakukan pengkodean. Selain melakukan pengkodean, tahapan ini juga melakukan pengujian terhadap hasil kode yang telah dibuat.

e. Deployment

Tahapan ini merupakan tahapan dalam menerapkan aplikasi yang telah dibuat dari lingkungan pengembangan ke dalam lingkungan yang sebenarnya.

3. HASIL PENELITIAN

3.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan tahapan dalam menganalisis kebutuhan-kebutuhan yang diperlukan dalam membangun sistem dengan mengumpulkan dan merincikan persyaratan tingkat tinggi dari sebuah sistem serta mendefinisikan fungsi-fungsi dasar yang akan dijalankan oleh perangkat lunak.

Kebutuhan fungsional mendeskripsikan layanan, fitur, atau fungsi yang disediakan oleh sistem untuk pengguna.

Pada tabel di bawah ini merupakan kebutuhan fungsional pada sistem yang akan dibangun.

No	Kebutuhan Fungsional	Fungsi
1	Membuat pesanan	1. Memilih menu 2. Checkout 3. Pengisian pilihan untuk kritik dan saran
2	Memproses pembayaran Online	1. Pembayaran Online
3	Mengelola penjualan	1. Login sebagai kasir 2. Mengelola penjualan
4	Mengelola pesanan	1. Login sebagai penyaji 2. Mengelola pesanan
5	Mengelola laporan penjualan	1. Login sebagai manager 2. Mengelola laporan penjualan
6	Mengelola pengguna	1. Login sebagai manager 2. Mengelola pengguna
7	Mengubah pengaturan	1. Login sebagai manager 2. Mengubah pengaturan
8	Mengelola menu	1. Login sebagai manager 2. Mengelola menu

TABEL 1: Kebutuhan Fungsional

3.2 Analisis Pengguna

Pada analisis pengguna ini akan mendefinikan siapa saja yang akan menggunakan system, dan apa saja yang dapat dilakukan oleh pengguna di dalam system. Berikut

di bawah ini merupakan pengguna dari aplikasi yang dibangun:

- Tamu: sebagai konsumen restoran
- Duitku: Sebagai aplikasi pihak ketiga yang digunakan untuk *Payment Gateway* untuk memproses pembayaran secara Online
- Pelayan: Sebagai orang yang melayani tamu
- Kasir: Sebagai orang yang menerima dan memproses pembayaran pesanan secara luring
- Penyaji: Sebagai koki yang membuat pesanan tamu
- Manajer: Sebagai orang yang memiliki peran tertinggi untuk mengatur aplikasi dimulai dari pengaturan kategori hingga pengaturan umum.

3.3 Perancangan Sistem

Perancangan sistem pada aplikasi eMenu menggunakan model UML yang dirancang dari diagram *Use Case*, *Activity*, dan *Class Diagram*. Pembuatan Diagram Entity Relationship untuk menggambarkan rancangan basis data yang akan digunakan. Dengan adanya rancangan system ini, akan memudahkan dalam tahap pengkodean selanjutnya.

3.3.1 Use Case Diagram

Use Case dibuat berdasarkan pada pengguna (*actor*) dan tujuannya (*goals*) dalam menggunakan sistem. *Use Case Diagram* menyediakan gambaran yang komprehensif dari kebutuhan sistem [6]. *Use Case Diagram* berisi *actor*, *use cases*, hubungan antar *use cases* dan *actors*, dan catatan lainnya. Berikut di bawah ini merupakan Use Case Diagram pada aplikasi eMenu.

Gambar 5: Use Case Diagram

Pada gambar 2 menjelaskan tentang Use Case Diagram dari Aplikasi E-Menu Restoran terdapat 19 (sembilan belas) Use Case. Tamu hanya dapat mengakses Use Case membuat pesanan dan tentang kami, Kasir hanya dapat mengakses Use Case Login sebagai Kasir, Mengelola Penjualan, Duitku hanya bisa memproses Use Case Pembayaran Online, Penyaji hanya bisa mengakses Use Case Login sebagai Penyaji, Use Case Mengelola Pesanan, Pelayan hanya bisa mengakses Use Case Login sebagai Penyaji, Use Case Lihat Pesanan, Manager hanya bisa mengakses Use Case Login sebagai Manager, Use Case Melihat Penjualan, Use Case Laporan Penjualan, Use Case Mengubah Pengaturan, Use Case Mengubah Tentang Kami, Use Case Mengelola Banner, Use Case Mengelola Kategori, Use Case Mengelola Menu, Use

Case Mengelola Pengguna, Use Case Melihat Data Kritik & Saran.

3.3.2 Activity Diagram

Activity Diagram pada UML menjelaskan tentang aliran atau proses pada sistem [6]. Dengan menggunakan *Activity Diagram* memungkinkan perancang untuk (a) merancang proses bisnis dari suatu organisasi yang menggambarkan fungsi bisnis dijalankan, (b) melihat aliran pada *Use Case* dengan membuat pemetaan secara visual dari dokumentasi *Use Case* itu sendiri, (c) Ketergantungan antara *Use Case* dengan menggunakan notasi aktivitas untuk menggambarkan *Use Case*. Berikut di bawah ini merupakan activity diagram pada perancangan aplikasi eMenu yang dibagi menjadi beberapa bagian.

a. *Activity Diagram* Membuat Pesanan

Gambar 6: *Activity Diagram* Membuat Pesanan

b. *Activity Diagram* Memproses Pembayaran Online

Gambar 7: *Activity Diagram* Mengelola Pembayaran Online

dengan memanggil fungsi yang telah disediakan oleh Duitku, dan Duitku akan melakukan proses dan menampilkan hasil berdasarkan permintaan.

Pada proses aktivitas pembayaran Online, proses verifikasi dan validasi pembayaran dilakukan oleh pihak ketiga yaitu Duitku. Sistem hanya melakukan permintaan

c. *Activity Diagram* Mengelola Penjualan

Gambar 8: *Activity Diagram* Mengelola Penjualan

d. *Activity Diagram* Mengelola Pesanan

Gambar 9: *Activity Diagram* Mengelola Pesanan

e. *Activity Diagram* Melihat Laporan Penjualan

Gambar 10: *Activity Diagram* Melihat Laporan Penjualan

f. *Activity Diagram* Mengelola Menu

Gambar 11: *Activity Diagram* Mengelola Menu

3.3.3 Class Diagram

Berikut di bawah ini merupakan Class Diagram dari aplikasi eMenu.

Gambar 12: Class Diagram

3.3.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) menjelaskan tentang hubungan antar entitas. ERD ini nantinya akan digunakan sebagai panduan dalam membangun basis data

untuk aplikasi eMenu. Berikut di bawah ini merupakan rancangan ERD dari aplikasi eMenu.

Gambar 13: Entity Relationship Diagram

4. PEMBAHASAN

4.1 Teknologi Yang Digunakan

Teknologi yang digunakan dalam pembuatan aplikasi eMenu ini adalah dengan menerapkan konsep teknologi *Single-Page Application*. Dengan menggunakan SPA, permintaan yang diminta tidak lagi harus memuat ulang semua halaman tetapi hanya data yang dibutuhkan saja yang dimuat ulang. Aplikasi dibagi menjadi dua bagian yaitu *Front-end* dan *Back-end*. Agar dapat terhubung antara *Front-end* dan *Back-end*, komunikasi diantara keduanya dihubungkan dengan menggunakan REST API. Sehingga data yang disimpan dalam basis data MySQL dapat diminta melalui *Front-end* dan disediakan oleh *Back-end*.

Bagian *Front-end* menggunakan Framework Vue.js yang sangat mampu memberikan dukungan dalam pembangunan *Single-Page Application*. Pada pembangunan aplikasi ini menggunakan vue-cli yang mengarahkan pengembang untuk menginstall node.js sebagai pendukungnya. Vue.js memiliki sistem *reactive data binding* yang berfungsi agar data dan *Document Object Model* (DOM) tetap terkait. Dengan adanya proses *binding*, DOM akan bersinkronisasi dengan data, dan jika terjadi perubahan, DOM akan segera memperbaruinya. Berikut di bawah ini adalah proses sinkronisasi DOM dan data pada Vue.js.

Gambar 14: Sinkronisasi DOM pada Vue.js [7]

Bagian *Back-end* menggunakan Framework PHP Laravel. Laravel dikenal sebagai Framework yang tangguh yang menganut konsep *clean code* dalam pengembangannya. Laravel juga sangat mendukung pengembangan *Single-Page Application*, yang terbukti pengembang bisa dengan sangat mudah dalam membuat API dari Laravel. Laravel menggunakan *design pattern* Modal-View-Controller (MVC). Berikut di bawah ini merupakan rancangan MVC dalam Laravel.

Gambar 15: Laravel MVC

Pada gambar di atas, pola MVC Laravel akan memungkinkan pengguna untuk mengakses Route yang telah disediakan. Jika Route tersebut valid, maka akan langsung diarahkan ke Controller yang menangani Route tersebut. Controller akan berinteraksi dengan Model dengan cara meminta data yang dibutuhkan dari basis data. Setelah data diterima, Controller mengirimkan data tersebut ke View untuk dapat di *render* dan ditampilkan ke pengguna sesuai dengan yang diminta.

4.2 Implementasi Antarmuka

Berikut di bawah ini merupakan implementasi hasil antarmuka yang telah dibuat.

a. Halaman Buat Pesanan

Gambar 16: Halaman Buat Pesanan

b. Halaman Tentang Kami

Gambar 17: Halaman Kelola Tentang Kami

c. Halaman Login

Gambar 18: Halaman Login

d. Halaman Kelola Penjualan

Gambar 19: Halaman Kelola Penjualan

e. Halaman Kelola Pesanan Pelayan

Gambar 20: Halaman Kelola Pesanan Pelayan

f. Halaman Kelola Pesanan Penyaji

Gambar 21: Halaman Kelola Pesanan Penyaji

g. Halaman Kelola Menu

Gambar 22: Halaman Kelola Menu

h. Halaman Penjualan

Gambar 23: Halaman Penjualan

i. Halaman Laporan Penjualan

Gambar 24: Halaman Laporan Penjualan

4.3. Pengujian

Pengujian merupakan bagian yang paling penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal. Pengujian perangkat lunak ini menggunakan metode pengujian black box.

No.	Halaman	Deskripsi	Kesesuaian
1.	Halaman Login	Jika <i>email</i> , <i>password</i> dan tipe user yang dimasukkan sesuai maka user akan diarahkan kepada halaman utama. Namun jika <i>email</i> , <i>password</i> dan tipe user tidak sesuai maka akan muncul pesan kesalahan.	Sesuai
2.	Halaman Buat Pesanan	Jika menekan tombol filter dan urutan menampilkan data yang diinginkan, jika menekan tombol tambah ke keranjang akan tampil tombol checkout.	Sesuai
3.	Halaman Checkout	Jika tamu mengubah menu maka data akan langsung di ubah dan jika klik tombol lanjutkan diteruskan ke tab form pemesanan dan pembayaran, jika klik tombol buat pesanana maka akan di alihkan ke halaman bukti pemesanan	Sesuai

No.	Halaman	Deskripsi	Kesesuaian
4.	Halaman Kelola Penjualan	Jika user menekan tombol filter dan urutkan maka akan muncul data yang diinginkan, jika user menekan tombol bayar maka akan muncul form pembayaran jika admin menekan tombol simpan maka akan menyimpan pembayaran, jika menekan tombol rincian maka akan muncul modal rincian.	Sesuai
5.	Halaman Kelola Pesanan	Jika user menekan tombol filter maka muncul form filter, jika user menekan tombol terapkan maka akan muncul data yang diinginkan, jika user menekan tombol urutkan maka akan muncul data yang diurutkan, jika menekan tombol rincian maka akan muncul modal rincian	Sesuai
6.	Halaman Penjualan	Jika user menekan tombol rincian maka akan muncul modal rincian	Sesuai
7.	Halaman Laporan Penjualan	Jika user menekan tombol rincian maka akan muncul modal rincian, jika user menekan tombol ekspor maka akan mendownload berkas laporan.	Sesuai
8.	Halaman Kelola Banner	Jika user menekan tombol tambah, ubah dan hapus berfungsi dengan baik.	Sesuai
9.	Halaman Kelola Pengguna	Jika user menekan tombol tambah, ubah dan hapus berfungsi dengan baik.	Sesuai
10.	Halaman Tentang Kami	Jika manager menekan menu tombol perbaharui maka akan muncul pesan tentang kami berhasil diperbaharui	Sesuai
11.	Halaman Pengaturan	Jika manager menekan menu tombol perbaharui maka akan muncul pesan pengaturan berhasil diperbaharui	Sesuai
12.	Halaman Kelola Menu	Jika user menekan tombol tambah, ubah dan hapus berfungsi dengan baik.	Sesuai

TABEL 2: Pengujian Sistem

5. KESIMPULAN

Aplikasi eMenu yang dibangun secara *Single-Page Application* dengan menggunakan Vue.js telah memiliki

fitur yang dapat menunjang semua proses bisnis secara umum pada tempat usaha rumah makan. Penggunaan yang sangat mudah karena berbasis web, sehingga pengguna hanya membutuhkan *Web Browser* untuk dapat mengakses sistem. Selain itu juga, aplikasi ini tidak akan melalui banyak kendala jika diakses menggunakan perangkat *Mobile*, karena dibangun secara *Responsive* dan tampilan akan menyesuaikan dengan perangkat yang mengakses aplikasi tersebut.

6. REFERENSI

- [1] D. Vinayak, V. Ranjan, N. Masiwal, and N. Verma, "e-Restaurant: Online Restaurant Management System for Android," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 1, 2013.
- [2] S. N. Cheong, W. W. Chiew, and W. J. Yap, "Design and development of Multi-touchable E-restaurant Management System," *CSSR 2010 - 2010 Int. Conf. Sci. Soc. Res.*, no. C SSR, pp. 680–685, 2010.
- [3] K. Chochiang, P. Ung, and N. Bunsaman, "One Stop Restaurant Service Application," *17th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. ECTI-CON 2020*, pp. 482–485, 2020.
- [4] M. Wasson, "Single Page Application, build modern responsive web apps with asp .Net," 2013. .
- [5] R. S. Pressman, *Software Engineering A Practitioner's Approach 8th Edition*. New York: McGraw-Hill Education, 2015.
- [6] B. Unhelkar, *Software Engineering with UML*. Boca Raton: CRC Press, 2018.
- [7] E. You, "Vue.js Concept Overview," 2015. .