

LAPORAN PRAKTEK KERJA LAPANGAN

**SISTEM INFORMASI GOFOODIES LOMBOK BERBASIS WEBSITE
MENGUNAKAN FRAMEWORK LARAVEL DAN BOOTSTRAP**



Disusun oleh:

MUHAMMAD KHAIDAR RAHMAN

F1D 018 086

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2021

LEMBAR PENGESAHAN
LAPORAN PRAKTEK KERJA LAPANGAN
SISTEM INFORMASI GOFOODIES LOMBOK BERBASIS WEBSITE
MENGGUNAKAN FRAMEWORK LARAVEL DAN BOOTSTRAP

Disusun oleh:

MUHAMMAD KHAIDAR RAHMAN

F1D 018 086

Telah disetujui oleh:

1. **Dosen Pembimbing**

1. Tanggal: 8 Maret 2021

Arik Aranta, S.Kom., M.Kom.

NIP. 199402202019031004

2. **Pembimbing Lapangan**

2. Tanggal: 8 Maret 2021



Raden Bagus Faizal Irary Sidharta, SE, MM.

NIP. -

Mengetahui,

Sekretaris Program Studi Teknik Informatika

Fakultas Teknik

Universitas Mataram



Andy Hidayat Jatmika, ST., M.Kom.

NIP.198312092012121001

KATA PENGANTAR

Puji syukur penulisan panjatkan ke hadirat Allah SWT, Tuhan semesta alam, karena dengan berkat, rahmat dan limpahan karunianya sehingga penulis dapat menyelesaikan Laporan Praktek Kerja Lapangan ini sebagaimana mestinya.

Adapun Praktek Kerja Lapangan (PKL) ini penulis laksanakan di Intansi Gojek Indonesia cabang Lombok dengan judul “Sistem Informasi Gofoodies Lombok berbasis website menggunakan *Framework Laravel* dan *Bootstrap*”. *Website* GoFoodies Lombok bertujuan meningkatkan loyalitas *Merchant* (Penjual makanan yang bermitra dengan Gojek) terhadap Gojek, meningkatkan kualitas informasi kepada *Merchant* dan meningkatkan keterikatan antara *Merchant* dengan Gojek. Dalam pembuatan laporan ini penulis berpedoman pada bahan kuliah, petunjuk dari pembimbing lapangan, dosen pembimbing, referensi dan literatur yang terkait dengan penulisan laporan.

Penulis menyadari laporan ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik serta saran yang bersifat membangun agar dapat menghasilkan karya yang lebih baik dimasa mendatang. Semoga laporan ini dapat bermanfaat bagi penulis dan pembaca sekalian.

Mataram, 8 Maret 2021



Penulis

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat	3
BAB II TINJAUAN INSTANSI TEMPAT PKL	4
2.1 Profil Gojek Indonesia.....	4
2.2 Visi dan Misi Gojek Indonesia.....	5
2.3 Lokasi Gojek Indonesia Cabang Lombok	5
2.4 Struktur Organisasi Gojek Indonesia Cabang Lombok.....	6
BAB III LANDASAN TEORI	7
3.1 Sistem Informasi.....	7
3.2 Website.....	8
3.3 Basis Data.....	9
3.4 ERD.....	9
3.5 <i>Framework Laravel</i>	11
3.6 <i>Framework Bootstrap</i>	11
3.7 <i>Use case Diagram</i>	12
3.8 <i>Activity Diagram</i>	13
3.9 <i>Sequence Diagram</i>	14
3.10 <i>Web Server</i>	15
3.11 <i>Visual Studio Code</i>	15
3.12 <i>MySQL</i>	16
BAB IV PEMBAHASAN	17
4.1 Metode Perancangan Sistem.....	18
4.2 Desain Sistem.....	19
4.3 Implementasi Sistem.....	20
4.4 Hasil Pengujian.....	46
BAB V PENUTUP	48
5.1 Kesimpulan.....	48

5.2 Saran.....	48
DAFTAR PUSTAKA.....	49
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 Lokasi Gojek Indonesia Cabang Lombok.....	6
Gambar 2. 2 Struktur Organisasi Gojek Cabang Lombok.....	7
Gambar 3. 1 Entitas kuat.....	10
Gambar 3. 2 Entitas lemah.....	10
Gambar 3. 3 <i>Simple attribute</i>	10
Gambar 3. 4 <i>Composite attribute</i>	10
Gambar 3. 5 <i>Multi value attribute</i>	10
Gambar 3. 6 Contoh relasi dari dua buah entitas	11
Gambar 4. 1 Model <i>waterfall</i>	17
Gambar 4. 2 <i>Use case diagram gofoodies lombok</i>	18
Gambar 4. 3 <i>activity diagram login admin</i>	20
Gambar 4. 4 <i>activity diagram</i> tambah data artikel.....	20
Gambar 4. 5 <i>activity diagram</i> ubah data artikel bantuan.....	21
Gambar 4. 6 <i>activity diagram</i> hapus data artikel	22
Gambar 4. 7 <i>activity diagram</i> tambah data FAQ.....	22
Gambar 4. 8 <i>activity diagram</i> ubah data FAQ.....	23
Gambar 4. 9 <i>activity diagram</i> hapus data FAQ	23
Gambar 4. 10 <i>activity diagram</i> hapus data kritik dan saran	24
Gambar 4. 11 <i>activity diagram</i> logout admin.....	24
Gambar 4. 12 <i>activity diagram</i> melihat daftar gofood.....	25
Gambar 4. 13 <i>activity diagram</i> melihat cara promo	25
Gambar 4. 14 <i>activity diagram</i> melihat artikel bantuan	26
Gambar 4. 15 <i>activity diagram</i> melihat FAQ	26
Gambar 4. 16 <i>activity diagram</i> tambah data kritik dan saran	27
Gambar 4. 17 <i>sequence diagram login admin</i>	28
Gambar 4. 18 <i>sequence diagram</i> tambah data artikel bantuan	28
Gambar 4. 19 <i>sequence diagram</i> ubah data artikel bantuan laman Hasil Akhir	29
Gambar 4. 20 <i>sequence diagram</i> hapus data artikel bantuan	30
Gambar 4. 21 <i>sequence diagram</i> tambah data FAQ.....	30
Gambar 4. 22 <i>sequence diagram</i> ubah data FAQ.....	31
Gambar 4. 23 <i>sequence diagram</i> hapus data FAQ	31
Gambar 4. 24 <i>sequence diagram</i> hapus data kritik.....	32

Gambar 4. 25 <i>sequence diagram</i> tambah data kritik dan saran	33
Gambar 4. 26 <i>sequence diagram</i> melihat daftar gofood.....	33
Gambar 4. 27 <i>sequence diagram</i> melihat cara promo	34
Gambar 4. 28 <i>ERD sistem informasi Gofoodies Lombok</i>	35
Gambar 4. 29 Halaman <i>Homepage</i>	37
Gambar 4. 30 Halaman Daftar Gofood.....	37
Gambar 4. 31 Halaman Cara Promo	38
Gambar 4. 32 Halaman artikel bantuan	39
Gambar 4. 33 Halaman kontak kami	40
Gambar 4. 34 Halaman FAQ	41
Gambar 4. 35 Halaman <i>login</i> admin.....	42
Gambar 4. 36 Halaman dashboard.....	42
Gambar 4. 37 Halaman Kelola Data artikel.....	43
Gambar 4. 38 Halaman tambah data artikel	43
Gambar 4. 39 Halaman ubah data artikel	44
Gambar 4. 40 Halaman Kelola data FAQ.....	44
Gambar 4. 41 Halaman tambah data FAQ.....	45
Gambar 4. 42 Halaman ubah data artikel	45
Gambar 4. 43 Halaman Kelola data Kritik	46
Gambar 4. 44 Persentase hasil pengujian	47

DAFTAR TABEL

Tabel 3. 1 <i>Use case Diagram</i>	11
Tabel 3. 2 <i>Activity Diagram</i>	12
Tabel 3. 3 <i>Sequence Diagram</i>	13

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi saat ini merupakan sebuah kebutuhan dalam melakukan setiap aktivitas di hampir setiap bidang, terutama di dalam sebuah instansi. Aplikasi komputer merupakan teknologi yang berkembang pesat pada saat ini, kemajuan teknologi dalam membantu pengolahan data atau informasi yang tersedia dapat berlangsung secara cepat dan efisien serta akurat. Kemajuan aplikasi dapat dilihat dari banyaknya aplikasi yang dapat memudahkan penggunaannya dalam membantu pekerjaannya. Salah satu model aplikasi yang sangat berperan penting adalah aplikasi pengolahan data yang sangat bermanfaat dan membantu dalam penyimpanan data, perubahan data dan pencarian data. Tuntutan setiap aktivitas ini yang menjadikan teknologi informasi sebagai bagian terpenting di dalam keberlangsungan kinerja yang sedang dijalankan oleh sebuah instansi. Salah satu hal ini menyebabkan teknologi khususnya aplikasi komputer menjadi pengaruh di dalam sebuah kinerja instansi, baik dalam bentuk sistem informasi berbasis teknologi komputer atau *website*, teknologi informasi dapat memberikan nilai tambah bagi organisasi sehingga kinerja sebuah organisasi menjadi efisien dan efektif.

Salah satu instansi yang menggunakan dan memanfaatkan teknologi informasi ini adalah PT Gojek Indonesia Cabang Lombok. PT Gojek Indonesia Cabang Lombok merupakan sebuah perusahaan teknologi asal Indonesia yang memberikan layanan di berbagai bidang seperti jasa ojek, bisnis makanan, layanan pembayaran uang digital, jasa pengantaran, jasa pemijatan, jasa layanan pembersihan rumah, layanan pembelian pulsa dan layanan-layanan lainnya, Khususnya di wilayah Kota Mataram dan Lombok Barat. Berbagai aktivitas yang berkaitan dengan kinerja instansi ini harus dilaksanakan dengan sebaik mungkin. Untuk itu saat ini PT Gojek Indonesia Cabang Lombok terus melakukan inovasi di bidang teknologi informasi dengan memberikan kesempatan pada setiap divisinya untuk membuat sebuah karya inovasi baru yang berhubungan dengan teknologi informasi guna membantu pekerjaan yang terdapat di instansi tersebut. Kemudian

permasalahan yang terdapat pada PT Gojek Indonesia Cabang Lombok yaitu pengelolaan tentang pemberian dan pengelolaan data informasi kepada *merchant* masih di *handle* oleh satu divisi saja dan pendataan untuk daftar menjadi *merchant* masih harus menggunakan layanan *google form* dan kebanyakan *merchant* masih kebingungan terhadap konten informasi yang diberikan sehingga sering manual bertanya melalui aplikasi Whatsapp. dengan adanya permasalahan ini, menjawab permasalahan tersebut maka diberikan solusi dengan membuat aplikasi Gofoodies Lombok berbasis website untuk memudahkan penyampaian informasi Gojek terhadap *merchant* dan pendataan informasi dari *merchant* bisa lebih teratur.

Sistem Informasi Gofoodies Lombok berbasis website dengan menggunakan *framework Laravel* dan *bootstrap* yang digunakan untuk mengelola konten informasi terhadap *merchant* dan menampung informasi dari *merchant*, sehingga nantinya hasil dari pengolahan data tersebut dapat digunakan sebagai bahan untuk meningkatkan loyalitas *merchant* terhadap Gojek, meningkatkan kualitas Informasi Kepada *merchant* dan meningkatkan keterikatan antara *merchant* dengan Gojek. Untuk mendapatkan hasil tersebut divisi pada bagian *merchant* tidak perlu lagi membuat *google form* secara manual untuk *merchant* dan mengurangi intensitas *merchant* yang bertanya melalui kontak whatsapp divisi bagian *merchant*. Dengan adanya Aplikasi Gofoodies Lombok membuat divisi bagian *merchant* cukup memberikan arahan kepada *merchant* untuk menggunakan aplikasi ini dan melihat informasinya pada aplikasi Gofoodies Lombok sehingga membantu pekerjaan divisi bagian *merchant* menjadi lebih mudah dan efektif.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka dapat dirumuskan pokok permasalahan yaitu :

1. Bagaimana cara meningkatkan loyalitas *merchant* terhadap Gojek ?
2. Bagaimana upaya yang diperlukan guna meningkatkan kualitas informasi terhadap *merchant* ?
3. Bagaimana sistem yang akan dibangun dapat meningkatkan keterikatan *merchant* terhadap Gojek ?

1.3 Batasan Masalah

Adapun batasan masalah berdasarkan rumusan masalah di atas yaitu :

1. Sistem yang dibangun hanya memberikan konten dan tampilan informasi menjadi lebih menarik sehingga memudahkan *merchant*.
2. Sistem yang dirancang menggunakan *framework Laravel* dan *Bootstrap*.
3. Sistem yang dibangun khususnya pada bagian menu admin hanya dapat diakses oleh pihak yang memiliki akun.
4. Sistem yang dibangun berfokus untuk memberikan fungsionalitas berupa memberikan informasi pendaftaran gofood dan melakukan promo serta artikel bantuan yang dibutuhkan oleh *merchant*.

1.4 Tujuan

Tujuan dari pembuatan sistem informasi gofoodies Lombok berbasis website ini adalah sebagai berikut :

1. Meningkatkan Loyalitas *merchant* terhadap Gojek.
2. Meningkatkan Kualitas Informasi Kepada *merchant*.
3. Meningkatkan Keterikatan antara *merchant* dengan Gojek.

1.5 Manfaat

Manfaat dari pembuatan sistem informasi gofoodies Lombok berbasis website ini adalah sebagai berikut:

1. Bagi Instansi

Dapat meningkatkan kualitas informasi kepada *merchant*, meningkatkan keterikatan antara *merchant* dengan Gojek, mempermudah penyampaian informasi dari gojek kepada *merchant* dan meningkatkan loyalitas *merchant* terhadap gojek.

2. Bagi Pegawai

Dapat memudahkan memantau, mengolah dan memberikan konten informasi yang lebih baik terhadap *merchant*.

BAB II

TINJAUAN INSTANSI TEMPAT PKL

2.1 Profil Gojek Indonesia

Gojek Indonesia (sebelumnya ditulis GO-JEK) merupakan sebuah perusahaan teknologi asal Indonesia yang berawal dari layanan transportasi, sekarang Gojek Indonesia memiliki lebih dari 20 layanan yang menjadi solusi buat tantangan sehari-hari. Berkat itu juga, Gojek menjadi salah satu platform teknologi terbesar yang melayani jutaan pengguna di Asia Tenggara dengan mengembangkan tiga Super-app yaitu untuk customer, mitra driver dan mitra *merchant*. Perusahaan ini didirikan pada tahun 2010 di Jakarta oleh Nadiem Makarim. Saat ini, Gojek telah tersedia di 50 kota di Indonesia. Pada tahun 2021, aplikasi Gojek sudah diunduh sebanyak lebih dari 50 juta kali di Google Play pada sistem operasi Android, dan telah tersedia di App Store. Gojek juga mempunyai layanan pembayaran digital yang bernama Gopay. Selain di Indonesia, layanan Gojek kini telah tersedia di Thailand, Vietnam dan Singapura.

Gojek didirikan oleh Nadiem Makarim, warga negara Indonesia lulusan Master of Business Administration dari Harvard Business School. Ide mendirikan Gojek muncul dari pengalaman pribadi Nadiem Makarim menggunakan transportasi ojek hampir setiap hari ke tempat kerjanya untuk menembus kemacetan di Jakarta. Saat itu, Nadiem masih bekerja sebagai Co-Founder dan Managing Director Zalora Indonesia dan Chief Innovation Officer Kartuku. Gojek memulai perjalanannya pada tahun 2010 dengan layanan pertama gojek yaitu pemesanan ojek melalui call-center. Pada tahun 2015, Gojek berkembang pesat setelah meluncurkan sebuah aplikasi dengan tiga layanan, yaitu: GoRide, GoSend, dan GoMart. Sejak saat itu, laju Gojek semakin cepat dan terus beranjak hingga menjadi grup teknologi terkemuka yang melayani jutaan pengguna di Asia Tenggara.

Tujuan dari Gojek Indonesia ialah memberi dampak ekonomi dan menghilangkan kesulitan kesusahan sehari-hari masyarakat Indonesia dan Asia Tenggara dengan menerapkan layanan gojek yang cepat dan tanggap, selalu berinovasi untuk membantu masyarakat dan menaikkan kesejahteraan masyarakat.

2.2 Visi dan Misi Gojek Indonesia

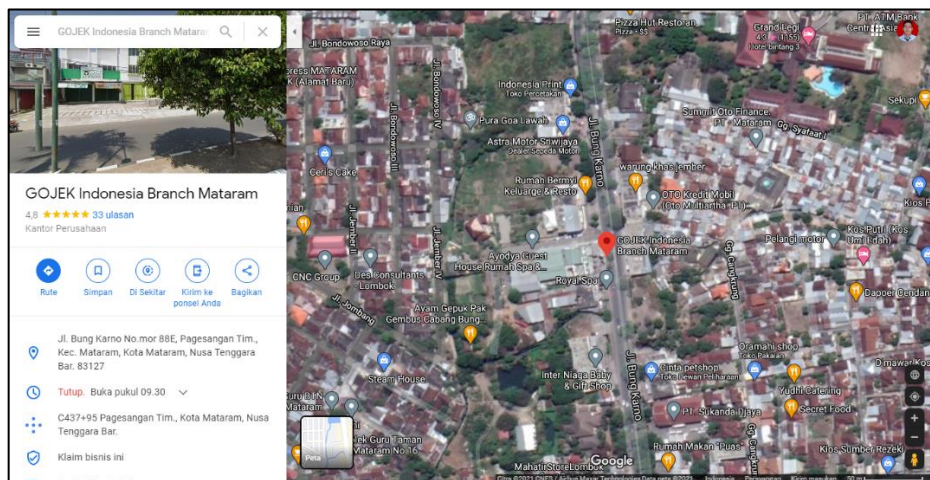
Visi:

Pasti ada jalan (Membantu dan menghilangkan Kesusahan Sehari-hari Masyarakat Indonesia dan Asia tenggara).

Misi:

1. *Speed* (Layanan Gojek dengan cepat dapat membantu kehidupan masyarakat).
2. Inovasi (Gojek dituntut selalu berinovasi untuk membantu kehidupan masyarakat)
3. *Sosial Impact* (Menaikkan taraf hidup jutaan masyarakat dengan meningkatkan kesejahteraan)

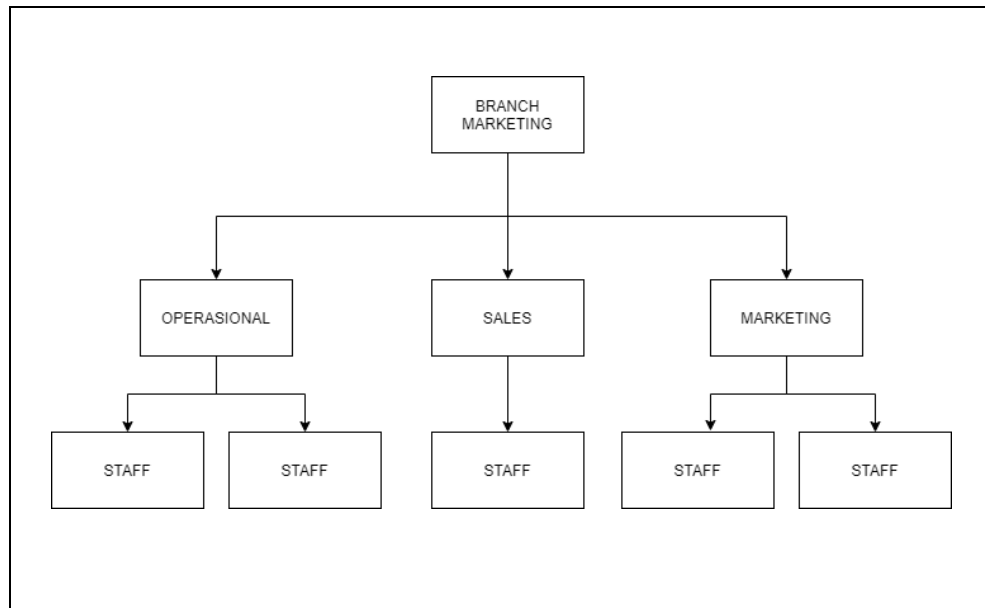
2.3 Lokasi Gojek Indonesia Cabang Lombok



Gambar 2.1 Lokasi Gojek Indonesia Cabang Lombok

2.4 Struktur Organisasi Gojek Indonesia Cabang Lombok

Berikut merupakan struktur Organisasi Gojek Cabang Lombok.



Gambar 2.2 Struktur Organisasi Gojek Cabang Lombok

BAB III

LANDASAN TEORI

3.1 Sistem Informasi

Sistem informasi terdiri dari dua kata, yaitu sistem dan informasi. Sistem menurut Fat adalah sebuah himpunan “benda” nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan (*Unity*) untuk mencapai suatu tujuan tertentu secara efisien dan efektif. Menurut Davis, G.B, sistem secara fisik adalah kumpulan dari elemen-elemen yang beroperasi bersama-sama untuk menyelesaikan suatu sasaran. Berdasarkan penjelasan di atas, dapat disimpulkan bahwa sistem merupakan kumpulan dari berbagai macam komponen-komponen atau elemen-elemen yang saling berkaitan satu dengan lainnya yang beroperasi secara bersama-sama untuk mencapai suatu tujuan tertentu. Sedangkan pengertian informasi menurut Sutanta yaitu hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Oleh sebab itu dapat disimpulkan bahwa sitem informasi adalah himpunan dari berbagai macam komponen-komponen yang saling berkaitan satu denan lainnya yang berfungsi untuk melakukan pengolahan data dan informasi [2].

Sistem informasi terdiri dari komponen-komponen atau yang biasa disebut dengan blok bangunan (*building block*) sebagai berikut [2]:

1. Blok masukan (*input block*)

Blok masukan ini mencakup data yang masuk ke dalam sistem, metode-
meode serta media yang digunakan untuk mendapatkan data tersebut.

2. Blok model (*model block*)

Blok ini terdiri dari prosedur, logika dan metode matematik yang akan memanipulasi input dan data yang tersimpan untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Mencangkup informasi yang telah diolah yang akan berguna bagi pemakai sistem informasi.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk mendapatkan *input*, menjalankan model, menyimpan dan mengakses data serta mengeluarkan *output*. Teknologi terdiri dari unsur utama yaitu:

- a. Teknisi (*human ware* atau *brain ware*)
- b. Perangkat lunak (*software*)
- c. Perangkat keras (*hard ware*)

5. Blok basis data (*database block*)

Blok basis data yaitu merupakan kumpulan data yang saling berinteraksi dan memiliki hubungan serta dapat dimanipulasi.

6. Blok kendali (*control block*)

Pengendali ini perlu dirancang agar dapat mencegah dan mengatasi hal-hal yang dapat merusak sistem.

3.2 Website

Website adalah kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu page dengan page yang lain yang sering disebut sebagai hyperlink [3]. Website biasanya terangkum dalam sebuah domain atau subdomain yang tempatnya berada di dalam World Wide Web (www) di internet. Sebuah Web Page adalah dokumen yang ditulis dalam format HTML yang hampir selalu bisa diakses melalui HTTP, yaitu protocol yang bisa menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Semua publikasi dari website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari website akan bisa diakses melalui sebuah URL yang bisa disebut Homepage. URL ini mengatur halaman-halaman situs untuk menjadi

sebuah hirarki, meskipun hyperlink yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa website membutuhkan subskripsi (data masukan) agar para user bisa mengakses sebagian atau keseluruhan isi web tersebut. Contohnya, ada beberapa situs-situs bisnis, situs-situs email gratisan yang membutuhkan subskripsi agar kita bisa mengakses situs tersebut [6].

3.3 Basis Data

Basis data merupakan kumpulan beberapa data yang disusun dalam suatu tabel yang saling berelasi ataupun tidak berelasi dan tersimpan di suatu media. Keberadaan sistem basis data di dalam sistem informasi adalah mutlak karena suatu sistem informasi tidak akan terealisasi tanpa adanya keterlibatan sistem basis data [3].

Terdapat beberapa operasi dasar pada basis data, yaitu:

1. Membuat basis data (*create database*)
2. Menghapus basis data (*drop database*)
3. Membuat tabel (*create table*)
4. Menghapus tabel (*drop table*)
5. Memasukkan data (*insert*)
6. Memperbaharui data (*update*)
7. Menghapus data (*drop*)

3.4 ERD

ERD (*entity relationship diagram*) berfungsi untuk memfasilitasi perancangan basis data. ERD terdiri dari komponen-komponen sebagai berikut [3]:

1. Entitas

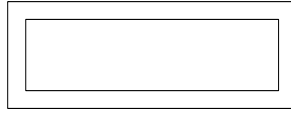
Entitas adalah sesuatu berupa objek di dunia nyata yang dapat dibedakan satu dengan yang lainnya. Entitas dibedakan menjadi dua jenis yaitu :

- a. Entitas kuat (*strong entity*), yaitu suatu entitas yang keberadaannya tidak tergantung dengan entitas lainnya. Entitas kuat digambarkan sebagai berikut



Gambar 3.1 Entitas kuat

- b. Entitas lemah (*weak entity*), yaitu suatu entitas yang keberadaannya bergantung dengan entitas lainnya. Entitas lemah digambarkan sebagai berikut :

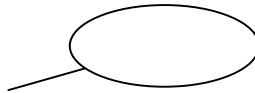


Gambar 3.2 Entitas lemah

2. Atribut

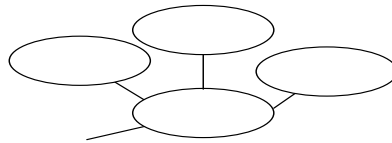
Atribut adalah karakteristik yang dimiliki oleh entitas. Berikut merupakan jenis-jenis atribut yang ada :

- a. *Simple attribute*, yaitu atribut yang nilainya tidak dapat dibagi menjadi bentuk yang lebih kecil.



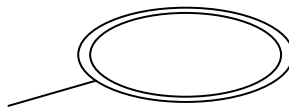
Gambar 3.3 *Simple attribute*

- b. *Composite attribute*, yaitu atribut yang nilainya dapat dibagi menjadi bagian yang lebih kecil.



Gambar 3.4 *Composite attribute*

- c. *Single valued attribute*, yaitu atribut yang hanya boleh mengandung satu nilai tertentu.
- d. *Multi valued attribute*, yaitu atribut yang boleh mengandung lebih dari satu nilai.

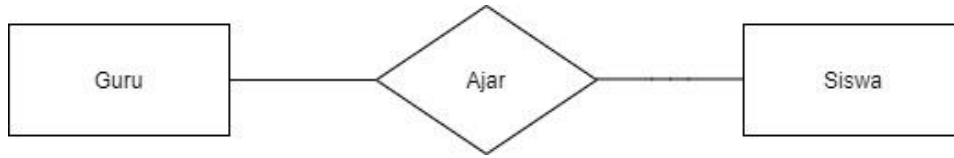


Gambar 3.5 *Multi valued attribute*

- e. *Derivated attribute*, yaitu atribut yang nilainya diperoleh dari dua atau lebih atribut.

3. Relasi

Relasi adalah hubungan yang terjadi antara satu atau lebih entitas. Relasi tidak memiliki keberadaan fisik, melainkan mewarisi hubungan antara entitas tersebut.



Gambar 3.6 Contoh relasi dari dua buah entitas

3.5 *Framework Laravel*

Framework adalah kumpulan instruksi-instruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang. Hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar tercipta nya *source code* yang bersih dan terstruktur.

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (Model View Controller). *Laravel* adalah pengembangan website berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu Model, View dan Controller.

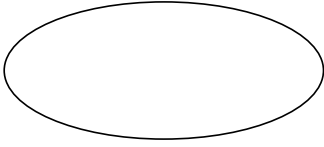
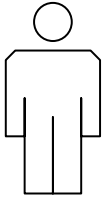



3.6 *Framework Bootstrap*

Bootstrap adalah paket aplikasi siap pakai untuk membuat front-end sebuah website. Bisa dikatakan, *bootstrap* adalah template desain web dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan *bootstrap* [7].

3.7 Use case Diagram

Use case diagram merupakan *diagram* yang menggambarkan kelakuan (*behavior*) sistem informasi yang dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang berhak melakukan fungsi-fungsi tersebut. Berikut merupakan simbol-simbol yang terdapat dalam *use case diagram*:

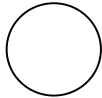


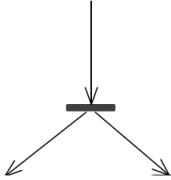
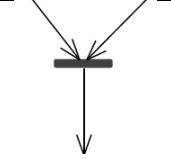
Tabel 3.1 Tabel *use case diagram*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan yang dinyatakan dengan kata kerja.
	Aktor merupakan orang atau sistem yang mengaktifkan fungsi yang ada di sistem. Di dalam <i>use case diagram</i> , setiap aktor dapat berinteraksi dengan <i>use case</i> yang ada, tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor atau <i>use case</i> digambarkan dengan garis lurus tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung.
	<i>Include</i> merupakan simbol yang digunakan oleh <i>use case</i> untuk memanggil <i>use case</i> lain. Contohnya adalah pemanggilan fungsi lain oleh program.
	<i>Extend</i> merupakan perluasan dari <i>use case</i> jika kondisi atau syarat lain terpenuhi.

3.8 Activity Diagram

Activity diagram berfungsi untuk menggambarkan aliran kerja atau aliran aktivitas dari suatu sistem. Berikut merupakan simbol-simbol yang terdapat dalam *activity diagram*:


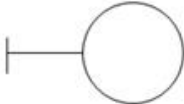

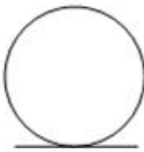
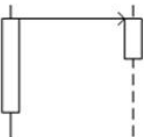
Tabel 3.2 Tabel *activity diagram*

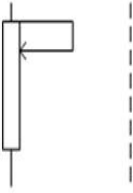
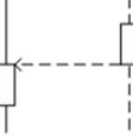

Gambar	Keterangan
	<i>Start point</i> merupakan awal dari aktivitas.
	<i>End point</i> merupakan simbol yang digunakan sebagai penanda akhir aktivitas.
	Merupakan simbol dari aktivitas yang terdapat di dalam sistem
	<i>Fork</i> atau percabangan digunakan untuk menunjukkan adanya dekomposisi.
	<i>Join</i> atau penggabungan digunakan untuk menggabungkan kegiatan paralel menjadi satu.

3.9 Sequence Diagram

Sequence Diagram menurut Munawar (2005) adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertical.

Tabel 3.3 Simbol *SequanceDiagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem
	<i>Boundary</i>	Menggambarkan interaksi antara satu atau lebih <i>actor</i> dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.
	<i>Control</i>	Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem
	<i>Entity</i>	Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem)
	<i>Object Message</i>	Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi

	<p><i>Generalization</i></p>	<p>Mengambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi</p>
	<p><i>Return Message</i></p>	<p>Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.</p>
	<p><i>Lifeline</i></p>	<p>Eksekusi obyek selama <i>sequence</i> (<i>message</i> dikirim atau diterima dan aktifasinya)</p>

3.10 Web Server

Web merupakan salah satu teknologi informasi yang menghubungkan data dari banyak sumber dan layanan yang bermacam-macam di internet. Sedangkan *web server* merupakan *server* yang digunakan untuk menyimpan halaman-halaman dari sebuah *website* [5].

3.11 Visual Studio Code

Visual Studio Code adalah sebuah *software* aplikasi/editor text untuk bahasa pemrograman multi bahasa termasuk pemrograman PHP dan *Javascript* yang akan digunakan untuk mengembangkan sistem, *Visual Studio Code* mendukung banyak bahasa pemrograman dan bahasa *markup* serta fungsinya dapat ditambah dengan plugin. Kelebihan *Visual Studio Code* yaitu:

- a. *Multiple selection*
- b. *Command pallete*
- c. *Distraction free mode*
- d. *Find in project*
- e. *Multi platform*

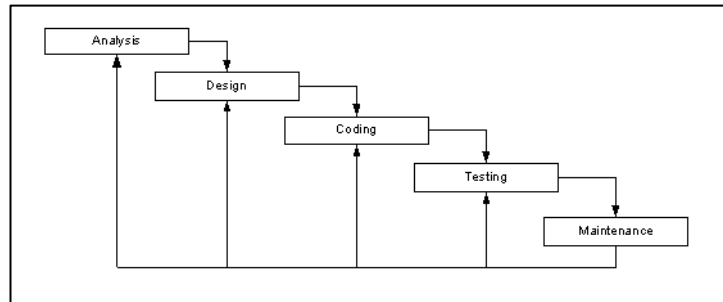
3.12 MySQL

MySQL merupakan salah satu jenis basis data yang pada umumnya sering digunakan dan sangat terkenal. Basis data yang tersimpan dalam basis data MySQL terdiri dari sejumlah tabel-tabel yang memiliki sejumlah baris dan sejumlah kolom [5].

BAB IV PEMBAHASAN

4.1 Metode Perancangan Sistem

Pengembangan sebuah perangkat lunak pada umumnya dilakukan dengan menggunakan kaidah *software engineering* (rekayasa perangkat lunak). Dalam teori *software engineering* terdapat berbagai macam model dari proses pengembangan perangkat lunak. Pada penelitian ini menggunakan model *waterfall*. Model *waterfall* atau yang sering disebut dengan model *classic life cycle* menunjukkan pengembangan perangkat lunak secara terstruktur dan sistematis dimulai dari tahap analisis kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing/verification, dan maintenance [3].



Gambar 4.1 Model *waterfall*

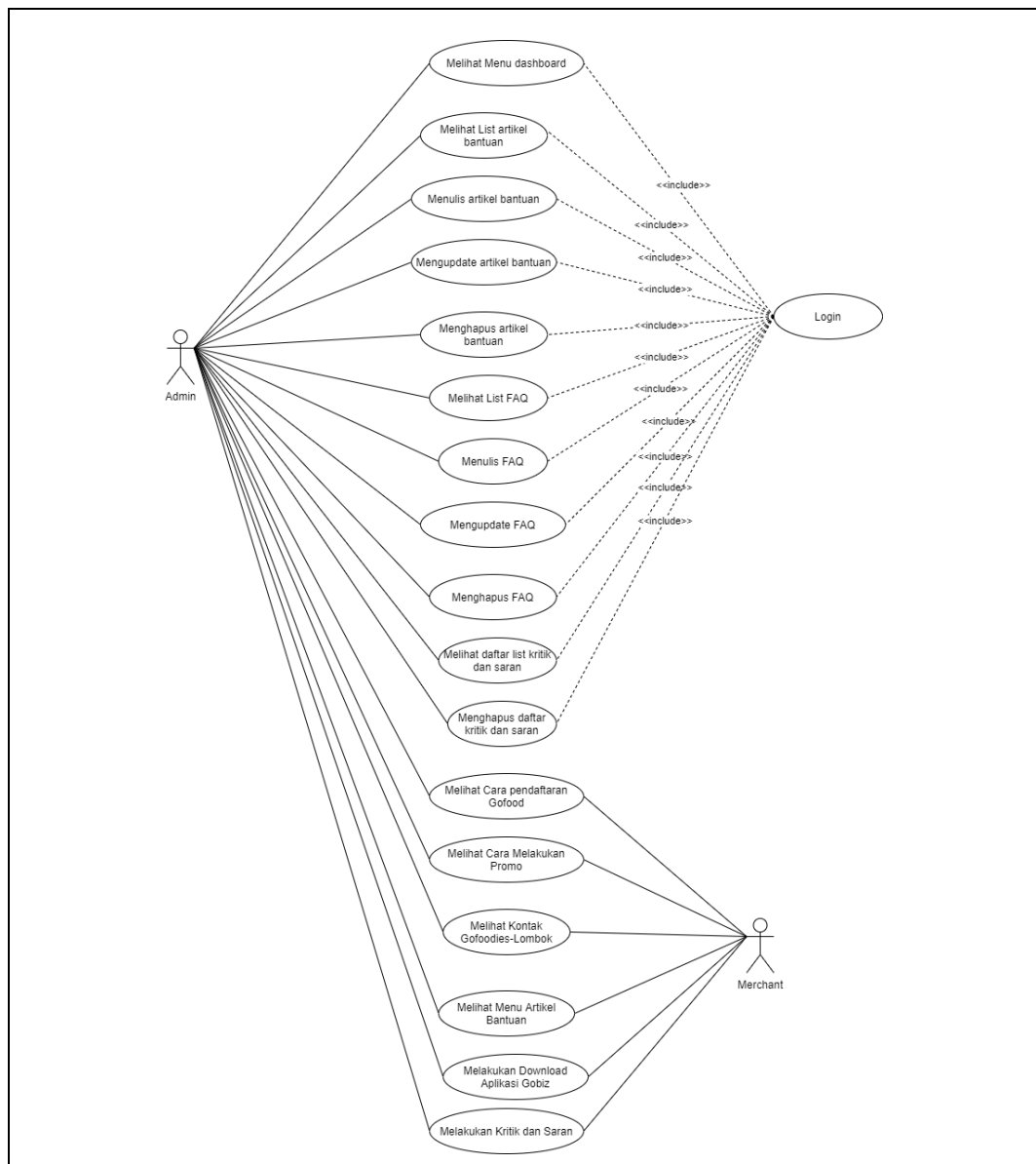
Untuk membuat perancangan *system* dilakukan analisis terhadap *system* yang akan dibuat. Hal yang pertama kali dilakukan adalah menganalisis *Use case diagram* untuk interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri. Selanjutnya menganalisis *activity diagram* pada sebuah *system* yang akan dirancang agar alur dari kerja suatu sistem dari awal hingga akhir dapat dimengerti oleh perancang *system*. Setelah itu buat perancangan untuk *sequence diagram* yang harus sesuai dengan *use case* yang ada pada *system*. Kemudian merancang *class diagram* untuk menjelaskan struktur dari program yang akan dibuat, biasanya *class diagram* dirancang pada *system* yang menggunakan konsep OOP. Perancangan dari segi *database* menggunakan penggambaran ERD juga perlu, untuk menggambarkan data-data yang ada dalam *system* [3].

4.2 Desain Sistem

Pada Sistem Informasi Gofoodies Lombok berbasis *web* ini terdapat 4 (empat) *diagram* yaitu *Use case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Entity Relationship Diagram*, berikut penjelasannya:

4.2.1 Use case Diagram

Berikut ini merupakan *Use case Diagram* dari Sistem Informasi Gofoodies Lombok yang ditunjukkan pada gambar 4.2.



Gambar 4.2 Use case diagram gofoodies lombok

Gambar 4.2 merupakan *usecase diagram* pada Sistem Informasi Gofoodies Lombok. Terdapat 2 (dua) jenis *user* dari sistem ini adalah:

1. *Merchant*

Berfungsi sebagai user *merchant* yang dapat melakukan:

- a. Melihat Cara pendaftaran Gofood
- b. Melihat Cara Melakukan Promo
- c. Melihat Kontak Gofoodies Lombok
- d. Melihat Menu FAQ bantuan
- e. Melihat Menu Artikel Bantuan
- f. Melakukan Download Aplikasi Gobiz
- g. Melakukan Kritik dan Saran

2. Admin

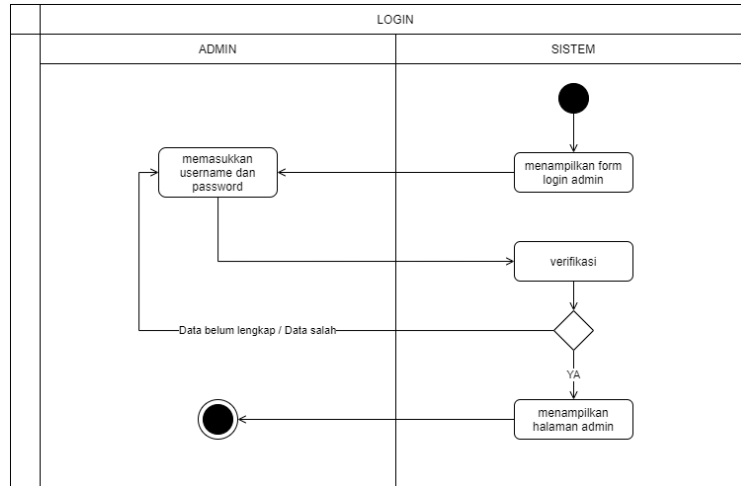
Berfungsi sebagai admin yang dapat melakukan:

- a. *Login* sebagai admin
- b. Melakukan semua yang bisa dilakukan oleh *merchant*
- c. Melihat Menu *dashboard*
- d. Melihat list artikel bantuan
- e. Menulis artikel bantuan
- f. Mengupdate artikel bantuan
- g. Menghapus artikel bantuan
- h. Melihat list FAQ
- i. Menulis FAQ
- j. Mengupdate FAQ
- k. Menghapus FAQ
- l. Melihat daftar list kritik dan saran
- m. menghapus daftar kritik dan saran
- n. Logout dari system

4.2.2 Activity Diagram

Berikut merupakan *Activity Diagram* dari Sistem Informasi Gofoodies Lombok.

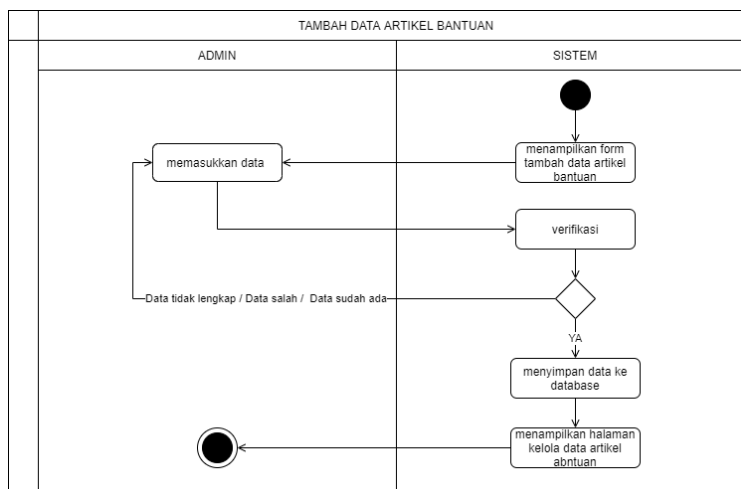
1. *Login* sebagai admin



Gambar 4.3 activity diagram login admin

Gambar 4.3 merupakan *activity diagram* dari proses *login* sebagai admin yang dimulai dari sistem menampilkan *form login*. Kemudian pengguna memasukkan *username* dan *password*. Setelah itu, sistem memverifikasi data yang telah dimasukkan, apabila data sudah lengkap dan sudah valid maka sistem akan menampilkan halaman *dashboard* admin. Apabila data tidak ditemukan atau data yang diisi salah maka admin harus memasukkan ulang data *username* dan *password* yang sesuai.

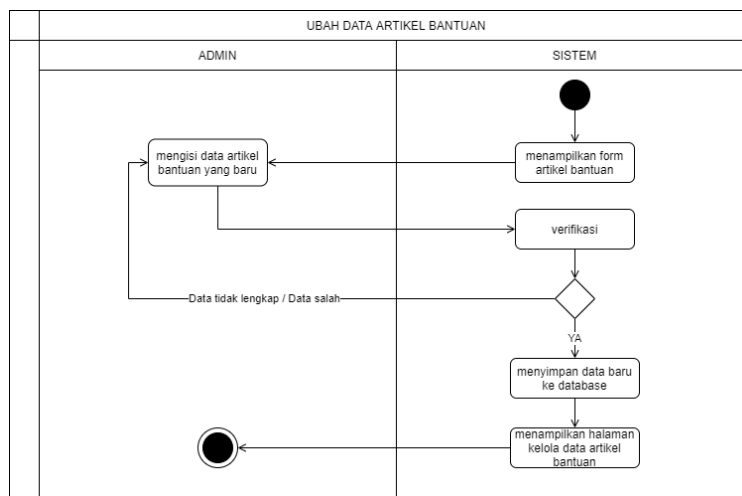
2. Proses penambahan data artikel bantuan



Gambar 4.4 activity diagram tambah data artikel

Gambar 4.4 merupakan *activity diagram* dari proses tambah data artikel bantuan. Proses ini berawal dari sistem yang menampilkan form tambah data artikel, kemudian admin mengisi data sesuai yang dibutuhkan. Data yang telah dimasukkan akan diverifikasi. Jika tidak terjadi duplikasi data, data telah benar dan terisi semua maka data disimpan ke dalam database dan jika tidak, maka admin harus melengkapi data atau memperbaiki data tersebut. Setelah menyimpan data ke dalam *database*, selanjutnya sistem menampilkan halaman kelola data artikel bantuan.

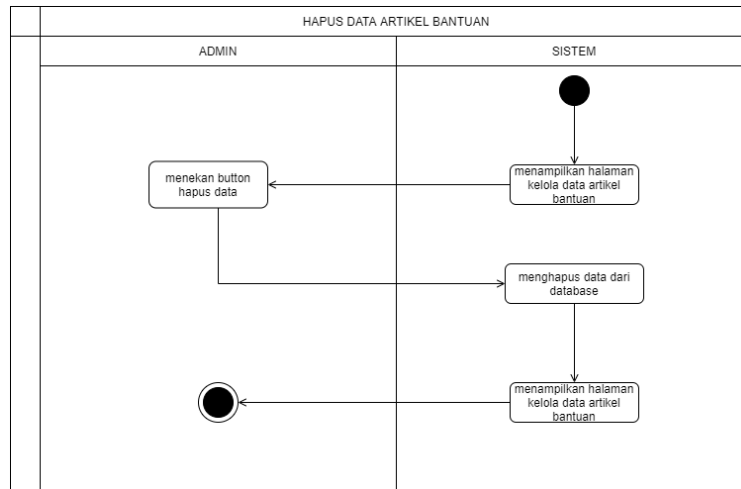
3. Proses mengubah data artikel bantuan



Gambar 4.5 *activity diagram* ubah data artikel bantuan

Gambar 4.5 merupakan *activity diagram* dari proses ubah data artikel bantuan. Proses ini berawal dari sistem menampilkan *form* ubah data artikel bantuan, lalu admin mengisi data baru yang ingin diubah. Data yang telah diubah diverifikasi, apabila data tersebut telah lengkap dan benar, maka data disimpan ke dalam *database* dan sistem menampilkan halaman kelola data artikel bantuan. Apabila data tidak lengkap atau salah, maka admin kembali memasukkan data yang benar dan lengkap.

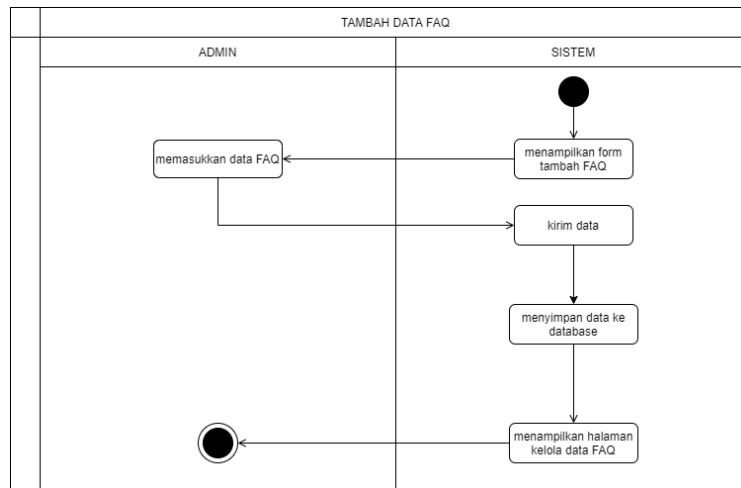
4. Proses menghapus data artikel bantuan



Gambar 4.6 activity diagram hapus data artikel

Gambar 4.6 merupakan *activity diagram* dari proses menghapus data artikel bantuan. Proses ini berawal dari sistem menampilkan halaman kelola data artikel bantuan. Admin menekan tombol hapus pada data yang dipilih di halaman kelola data artikel bantuan. Setelah itu, data yang terpilih akan dihapus dari *database* dan sistem menampilkan halaman kelola data artikel bantuan.

5. Proses penambahan data FAQ

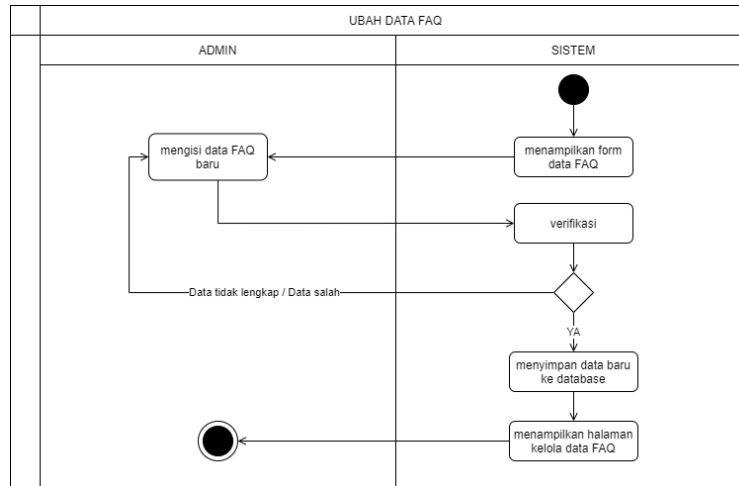


Gambar 4.7 activity diagram tambah data FAQ

Gambar 4.7 merupakan *activity diagram* dari proses tambah data FAQ. Proses ini berawal dari sistem yang menampilkan form tambah data FAQ, kemudian admin mengisi data sesuai yang dibutuhkan. Data yang telah dimasukkan akan diverifikasi. Jika tidak terjadi duplikasi data, data telah

benar dan terisi semua maka data disimpan ke dalam *database* dan jika tidak, maka admin harus melengkapi data atau memperbaiki data tersebut. Setelah menyimpan data ke dalam *database*, selanjutnya sistem menampilkan halaman kelola data FAQ.

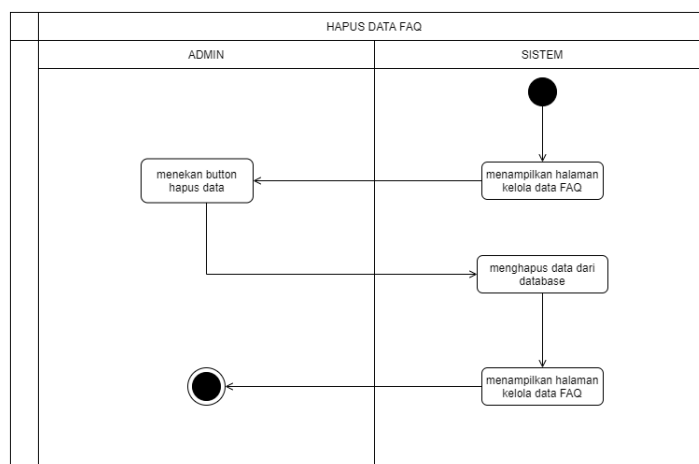
6. Proses mengubah data artikel FAQ



Gambar 4.8 activity diagram ubah data FAQ

Gambar 4.8 merupakan *activity diagram* dari proses ubah data FAQ. Proses ini berawal dari sistem menampilkan *form* ubah data FAQ, lalu admin mengisi data baru yang ingin diubah. Data yang telah diubah diverifikasi, apabila data tersebut telah lengkap dan benar, maka data disimpan ke dalam *database* dan sistem menampilkan halaman kelola data FAQ. Apabila data tidak lengkap atau salah, maka admin kembali memasukkan data yang benar dan lengkap.

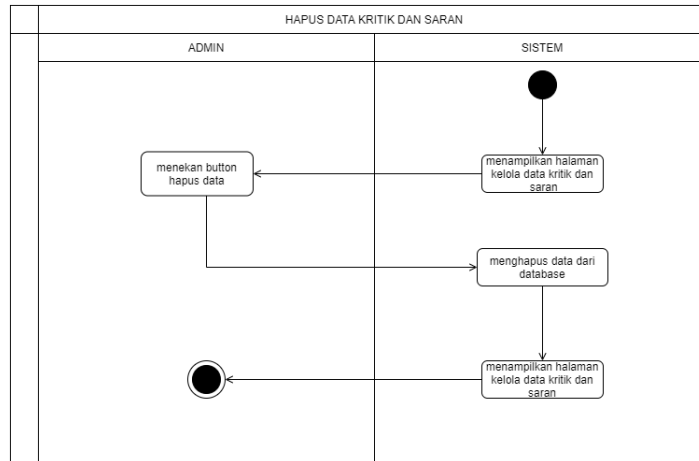
7. Proses menghapus data FAQ



Gambar 4.9 activity diagram hapus data FAQ

Gambar 4.9 merupakan *activity diagram* dari proses menghapus data FAQ. Proses ini berawal dari sistem menampilkan halaman kelola data FAQ. Admin menekan tombol hapus pada data yang dipilih di halaman kelola data FAQ. Setelah itu, data yang terpilih akan dihapus dari *database* dan sistem menampilkan halaman kelola data FAQ.

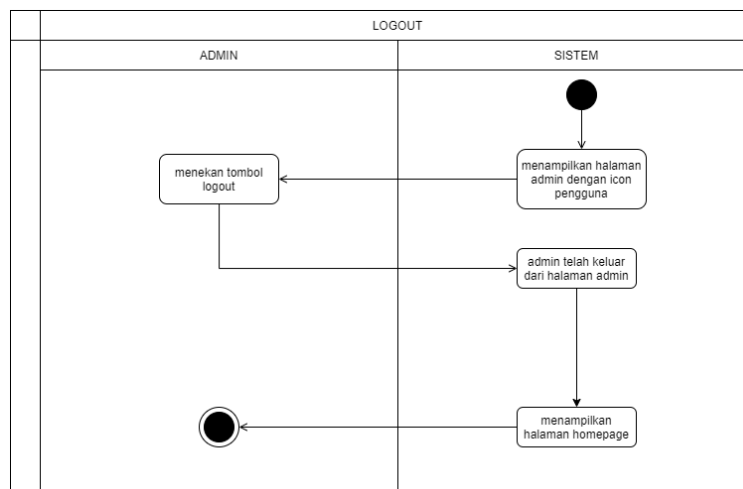
8. Proses menghapus data kritik dan saran



Gambar 4.10 *activity diagram* hapus data kritik dan saran

Gambar 4.10 merupakan *activity diagram* dari proses menghapus data kritik dan saran. Proses ini berawal dari sistem menampilkan halaman kelola data kritik dan saran. Admin menekan tombol hapus pada data yang dipilih di halaman kelola data kritik dan saran. Setelah itu, data yang terpilih akan dihapus dari *database* dan sistem menampilkan halaman kelola data kritik dan saran.

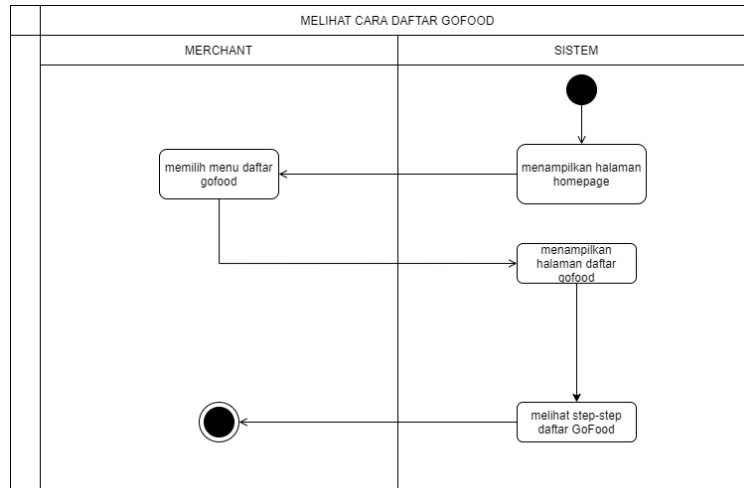
9. *Logout* sebagai admin



Gambar 4.11 *activity diagram* *logout* admin

Gambar 4.11 merupakan *activity diagram* dari proses *logout* sebagai admin yang dimulai dari sistem menampilkan halaman admin dengan *icon* pengguna. Kemudian pengguna menekan tombol *logout*. Setelah itu, sistem mengarahkan untuk menampilkan halaman *homepage* bahwa admin sudah *logout*.

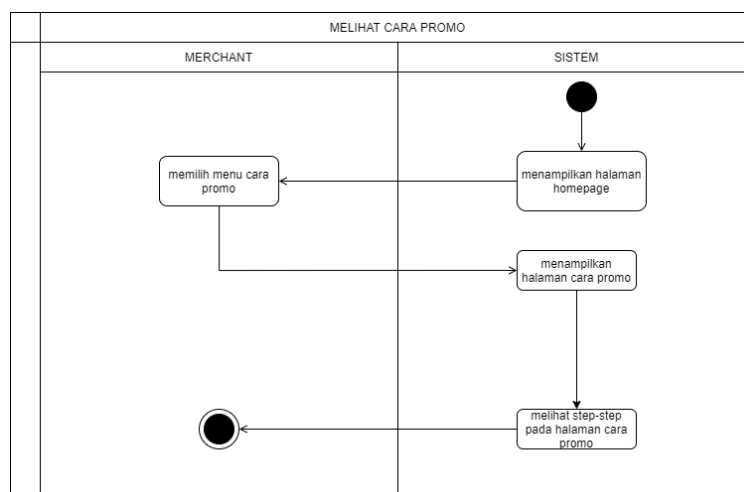
10. Proses melihat Daftar *Gofood*



Gambar 4.12 *activity diagram* melihat daftar *gofood*

Gambar 4.12 merupakan *activity diagram* dari proses melihat daftar *gofood* sebagai *merchant* yang dimulai dari sistem menampilkan halaman *homepage*. Kemudian pengguna memilih menu navigasi daftar *gofood*. Setelah itu, sistem mengarahkan untuk menampilkan halaman daftar *gofood*.

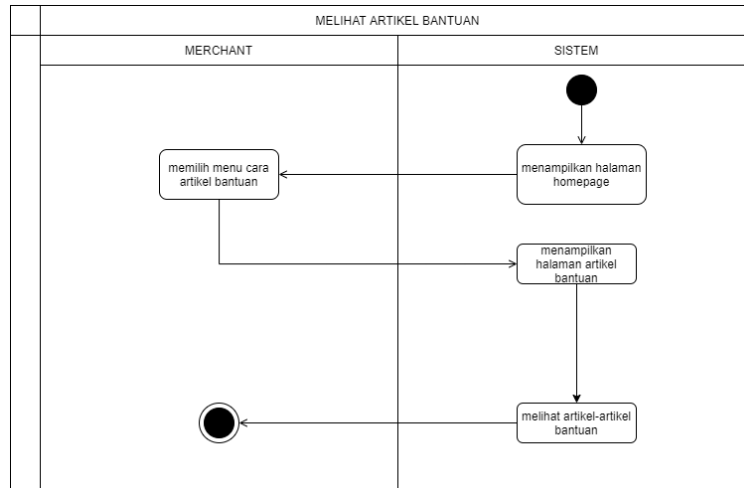
11. Proses melihat Cara Promo



Gambar 4.13 *activity diagram* melihat cara promo

Gambar 4.13 merupakan *activity diagram* dari proses melihat cara promo sebagai *merchant* yang dimulai dari sistem menampilkan halaman *homepage*. Kemudian pengguna memilih menu navigasi cara promo. Setelah itu, sistem mengarahkan untuk menampilkan halaman cara promo.

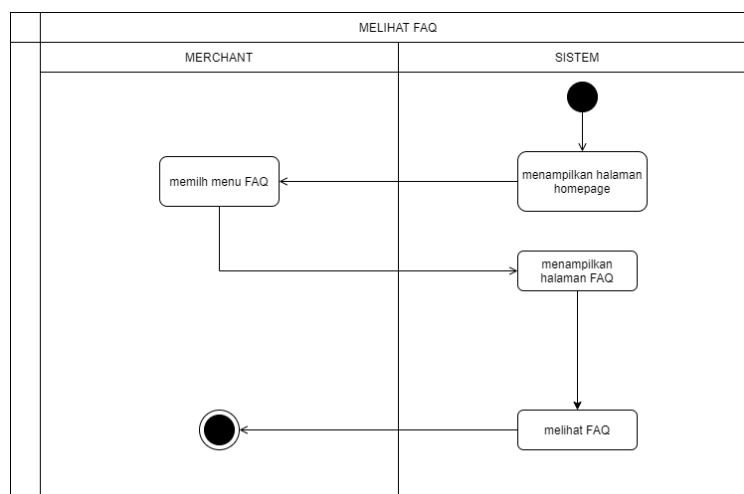
12. Proses melihat artikel bantuan



Gambar 4.14 *activity diagram* melihat artikel bantuan

Gambar 4.14 merupakan *activity diagram* dari proses melihat artikel bantuan sebagai *merchant* yang dimulai dari sistem menampilkan halaman *homepage*. Kemudian pengguna memilih menu navigasi artikel bantuan. Setelah itu, sistem mengarahkan untuk menampilkan halaman artikel bantuan.

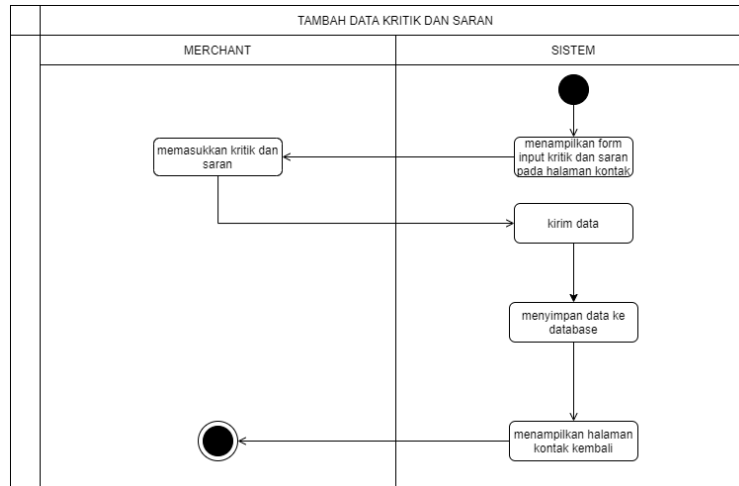
13. Proses melihat FAQ



Gambar 4.15 *activity diagram* melihat FAQ

Gambar 4.15 merupakan *activity diagram* dari proses melihat FAQ sebagai *merchant* yang dimulai dari sistem menampilkan halaman *homepage*. Kemudian pengguna memilih menu navigasi FAQ. Setelah itu, sistem mengarahkan untuk menampilkan halaman FAQ.

14. Proses penambahan data kritik dan saran oleh *merchant*



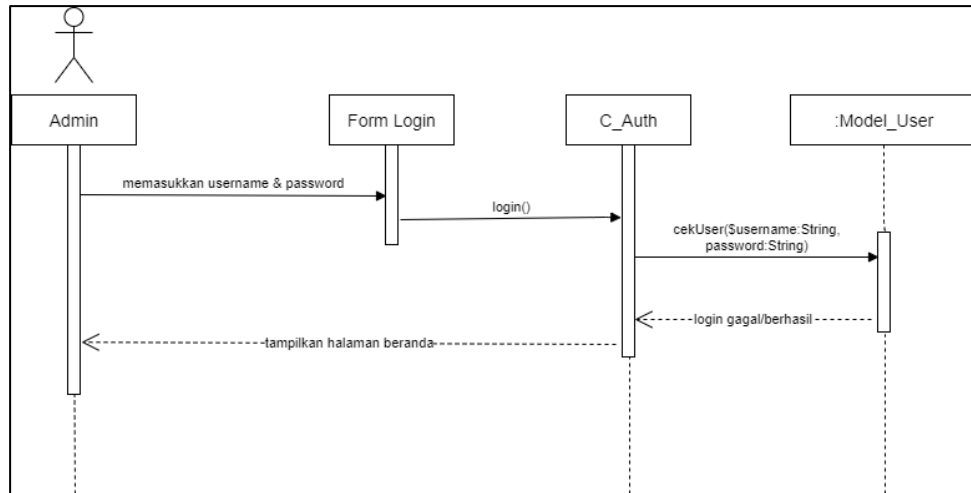
Gambar 4.16 *activity diagram* tambah data kritik dan saran

Gambar 4.16 merupakan *activity diagram* dari proses tambah data kritik dan saran. Proses ini berawal dari sistem yang menampilkan form tambah data kritik dan saran, kemudian *merchant* mengisi data sesuai yang dibutuhkan. Jika data telah benar dan terisi semua maka data disimpan ke dalam *database* dan jika tidak, maka *merchant* harus melengkapi data atau memperbaiki data tersebut. Setelah menyimpan data ke dalam *database*, selanjutnya sistem menampilkan halaman kontak kembali.

4.2.3 Sequence Diagram

Berikut merupakan *sequence diagram* dari sistem informasi Gofoodies Lombok:

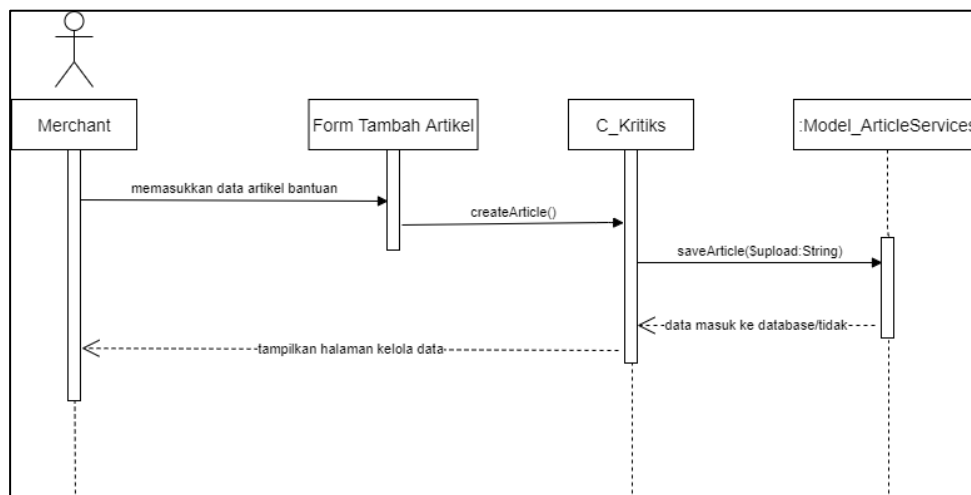
1. Login



Gambar 4.17 *sequence diagram* login admin

Gambar 4.17 merupakan *sequence diagram* dari proses *login*. Proses ini diawali dari form *login*, admin memasukkan *username* dan *password* lalu memanggil fungsi '*login()*' pada *Controller Auth*. Kemudian memanggil fungsi '*cekAkun()*' pada Model *Model_User*. Dari Model, mengirimkan nilai *login* berhasil atau gagal. Apabila berhasil, *Controller* akan menampilkan halaman beranda *dashboard*.

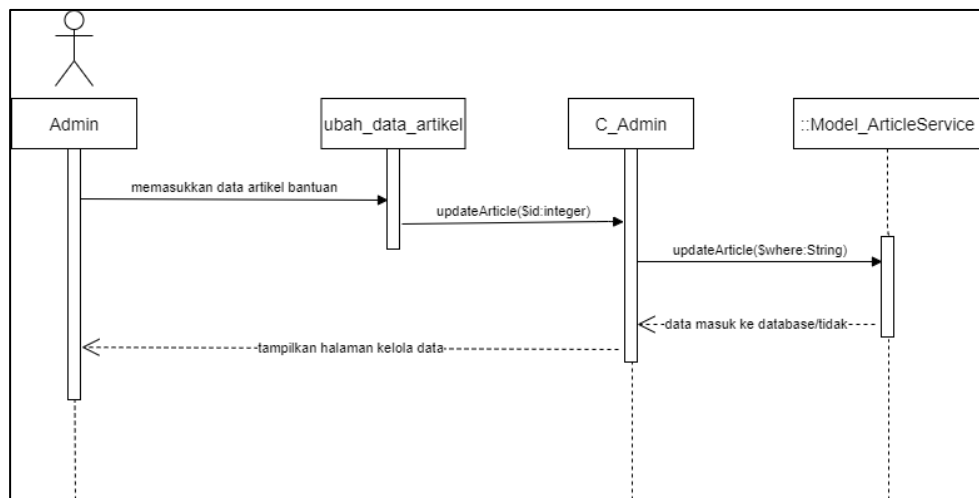
2. Penambahan data artikel bantuan



Gambar 4.18 *sequence diagram* tambah data artikel bantuan

Gambar 4.18 merupakan *sequence diagram* dari proses tambah artikel bantuan. Proses ini diawali tampilan form tambah artikel. Admin memasukkan data artikel kemudian memanggil fungsi “createArticle()” pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi “saveArticle()” pada Model Model_ArticleServices. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data artikel bantuan.

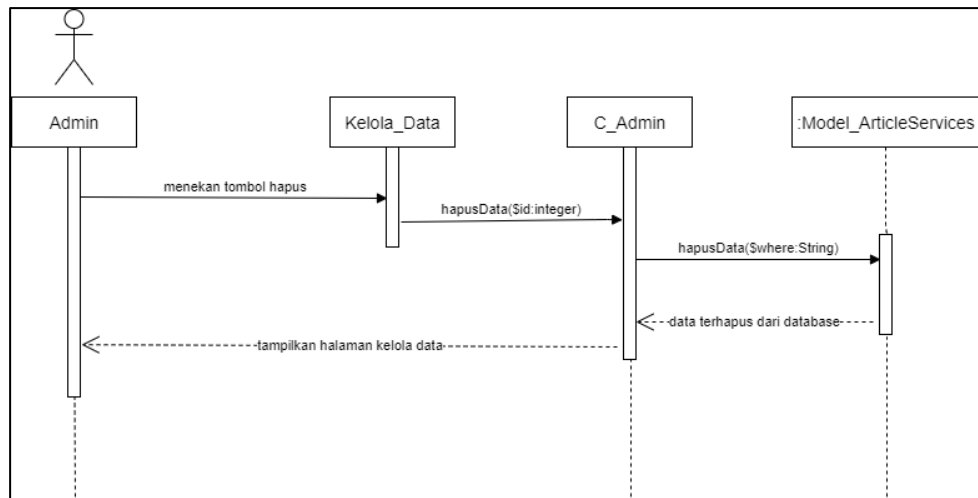
3. Perubahan data artikel bantuan



Gambar 4.19 *sequence diagram* ubah data artikel bantuan

Gambar 4.19 merupakan *sequence diagram* dari proses ubah artikel bantuan. Proses ini diawali tampilan form ubah artikel bantuan. Admin memasukkan data artikel kemudian memanggil fungsi “updateArticle()” pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi “updateArticle()” pada Model Model_ArticleServices. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data artikel bantuan.

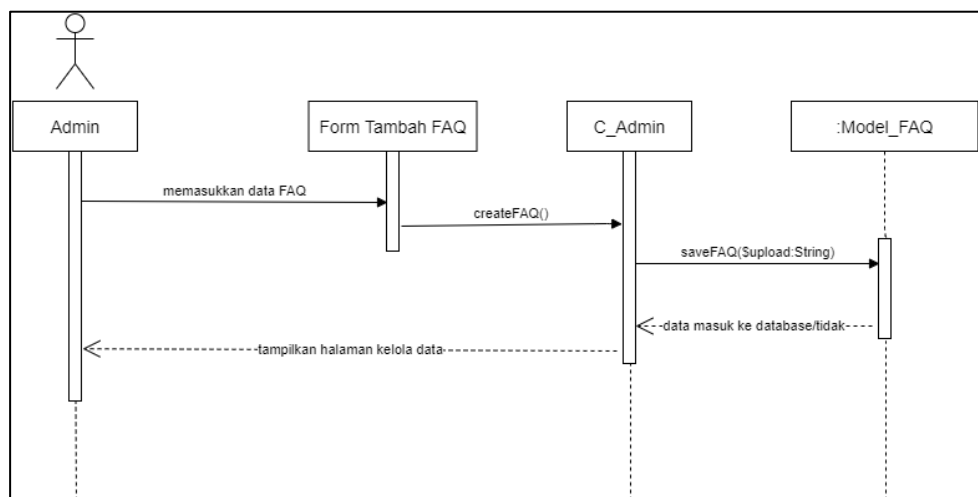
4. Penghapusan data artikel bantuan



Gambar 4.20 *sequence diagram* hapus data artikel bantuan

Gambar 4.20 merupakan *sequence diagram* dari proses hapus artikel bantuan. Proses ini diawali tampilan form Kelola data artikel bantuan. Admin menghapus data artikel kemudian memanggil fungsi “hapusData ()” pada *Controller* Admin dan data yang dihapus dikirim pada fungsi “hapusData ()” pada Model Model_ArticleServices. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data artikel bantuan.

5. Penambahan data FAQ

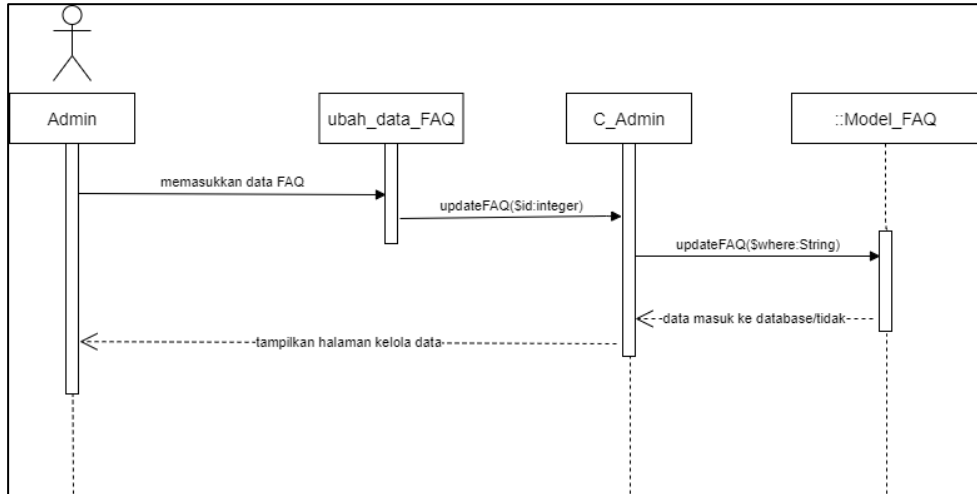


Gambar 4.21 *sequence diagram* tambah data FAQ

Gambar 4.21 merupakan *sequence diagram* dari proses tambah FAQ. Proses ini diawali tampilan form tambah FAQ. Admin memasukkan data artikel kemudian memanggil fungsi “createFAQ ()” pada *Controller* Admin

dan data yang dimasukkan dikirim pada fungsi “`saveFAQ()`” pada Model `Model_FAQ`. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data FAQ.

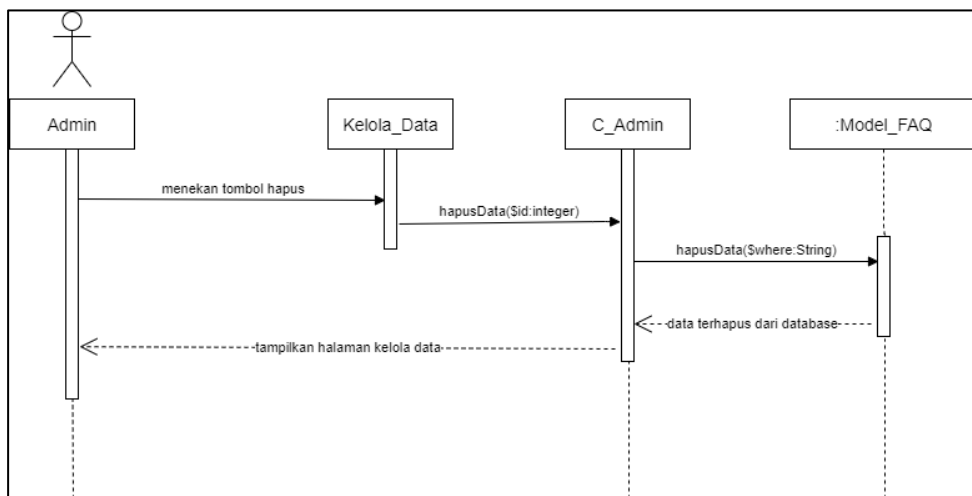
6. Perubahan data FAQ



Gambar 4.22 *sequence diagram* ubah data FAQ

Gambar 4.22 merupakan *sequence diagram* dari proses ubah FAQ, Proses ini diawali tampilan form ubah FAQ. Admin memasukkan data artikel kemudian memanggil fungsi “`updateFAQ()`” pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi “`updateFAQ()`” pada Model `Model_FAQ`. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data FAQ.

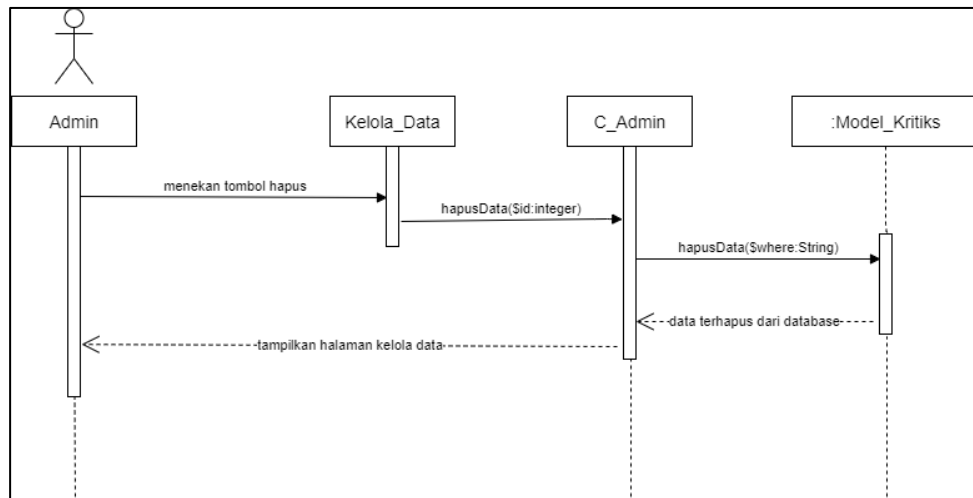
7. Penghapusan data FAQ



Gambar 4.23 *sequence diagram* hapus data FAQ

Gambar 4.23 merupakan *sequence diagram* dari proses hapus FAQ. Proses ini diawali tampilan form data FAQ. Admin menghapus data FAQ kemudian memanggil fungsi “hapusData ()” pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi “hapusData ()” pada Model Model_FAQ. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data FAQ.

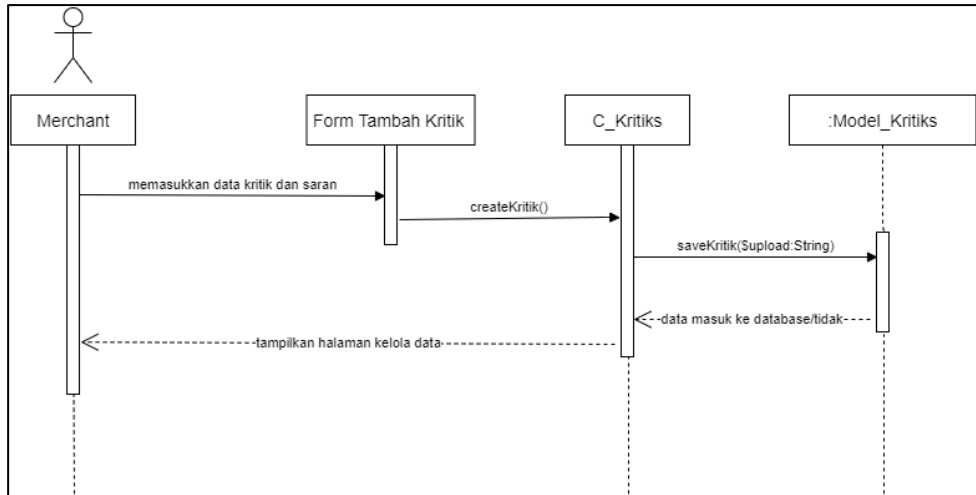
8. Penghapusan data kritik dan saran



Gambar 4.24 *sequence diagram* hapus data kritik

Gambar 4.24 merupakan *sequence diagram* dari proses hapus kritik dan saran. Proses ini diawali tampilan kelola data kritik dan saran. Admin menghapus data kritik dan saran kemudian memanggil fungsi “hapusData ()” pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi “hapusData ()” pada Model Model_Krtiks. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data kritik dan saran.

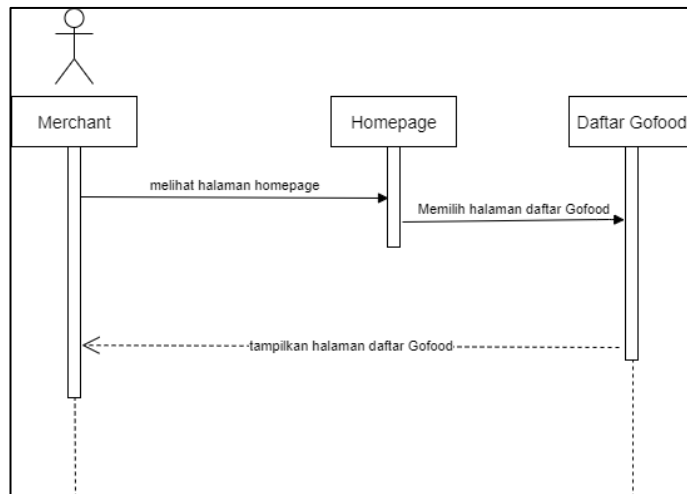
9. Penambahan data kritik dan saran



Gambar 4.25 *sequence diagram* tambah data kritik dan saran

Gambar 4.25 merupakan *sequence diagram* dari proses tambah kritik dan saran oleh *merchant*. Proses ini diawali tampilan form tambah kritik. *merchant* memasukkan kritik dan saran kemudian memanggil fungsi “*createKritik()*” pada *Controller* Kritiks dan data yang dimasukkan dikirim pada fungsi “*saveKritik()*” pada Model Model_FAQ. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kontak Kembali dan data tersebut ditampilkan pada halaman Kelola data kritik dan saran.

10. Melihat cara daftar *Gofood*

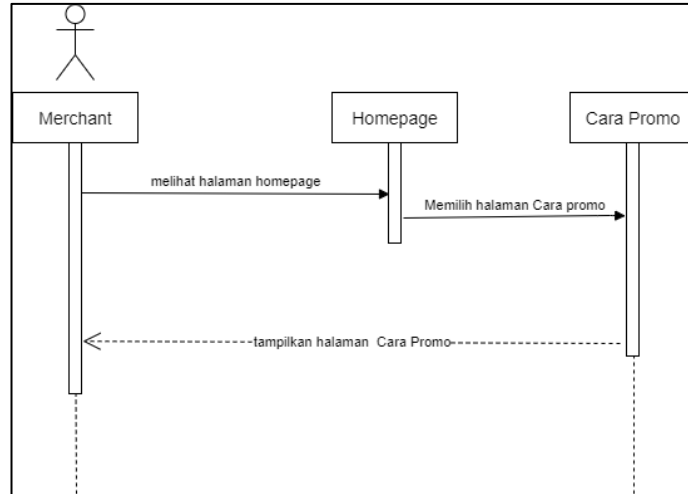


Gambar 4.26 *sequence diagram* melihat daftar gofood

Gambar 4.26 merupakan *sequence diagram* dari proses melihat daftar *gofood* proses ini diawali dengan *merchant* yang mengakses halaman

Homepage kemudian memilih navigasi untuk halaman daftar *gofood*.
Kemudian ditampilkan halaman daftar *gofood*.

11. Melihat cara promo

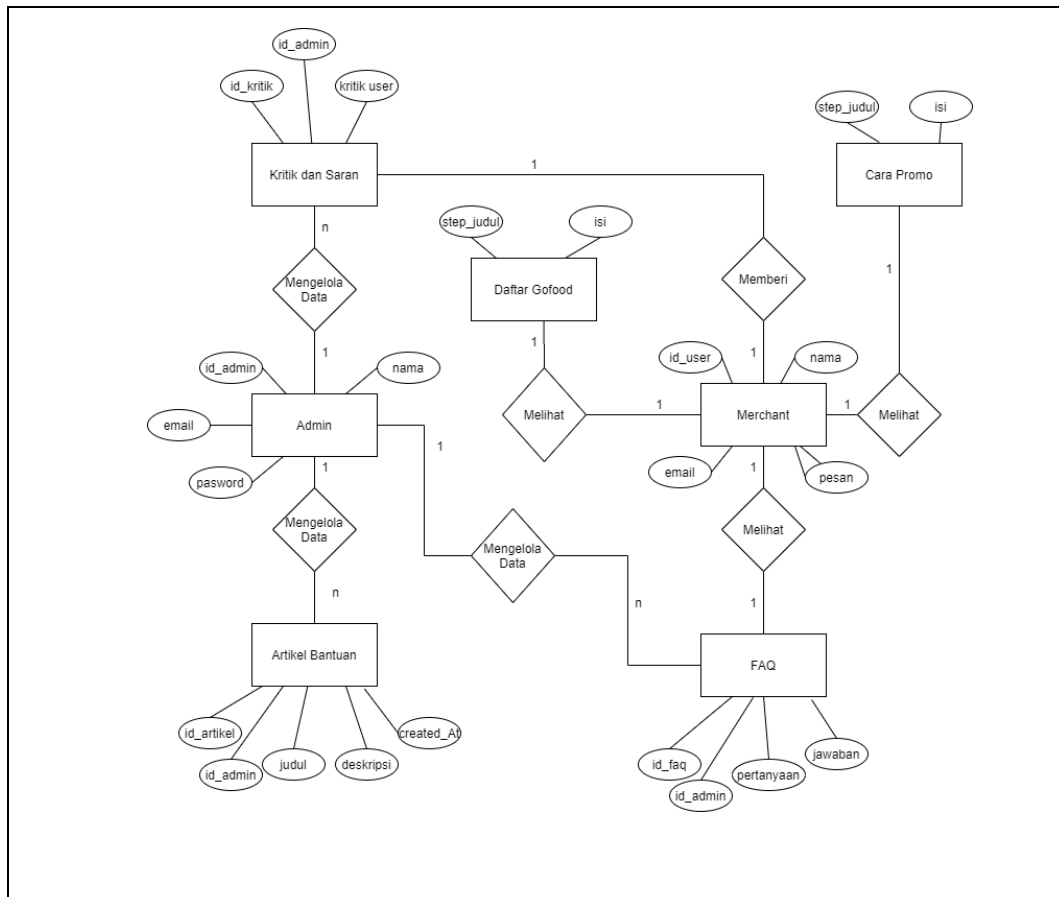


Gambar 4.27 *sequence diagram* melihat cara promo

Gambar 4.27 merupakan *sequence diagram* dari proses melihat cara promo oleh *merchant* proses ini diawali dengan *merchant* yang mengakses halaman *Homepage* kemudian memilih navigasi untuk halaman cara promo. Kemudian ditampilkan halaman cara promo.

4.2.4 Entity Relationship Diagram (ERD)

Berikut pada gambar 4.26 adalah ERD (*EntityRelationship Diagram*) dari sistem informasi Gofoodies Lombok dimana terdapat 7 (tujuh) entitas yaitu tabel admin, merchant, kritik dan saran, artikel bantuan, cara promo, FAQ dan daftar gofood.



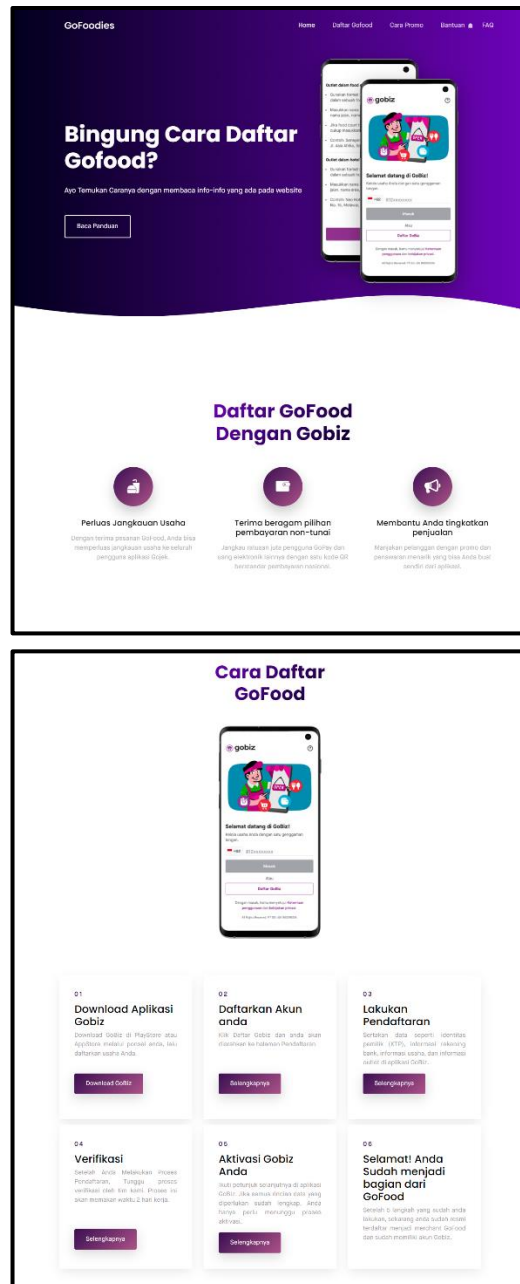
Gambar 4.28 ERD sistem informasi Gofoodies Lombok

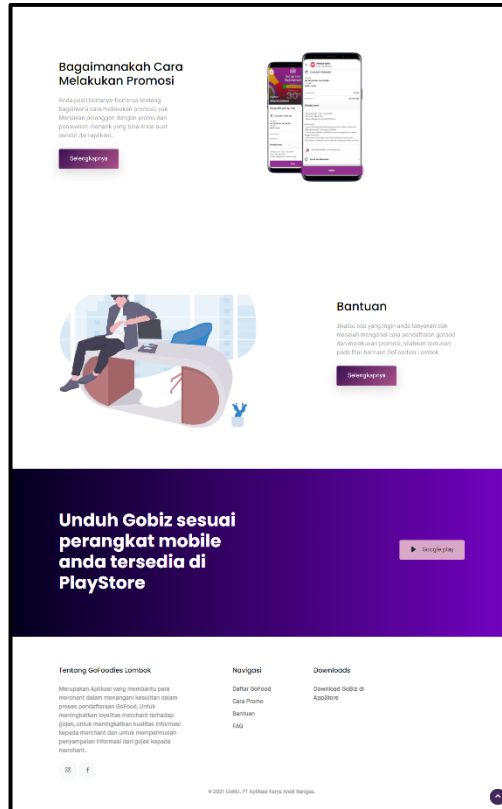
4.3 Implementasi Sistem

4.3.1 Implementasi *Interface* Sistem

Berikut pada gambar adalah implementasi tampilan (*interface*) program dari sistem Gofoodies Lombok.

1. Halaman Homepage

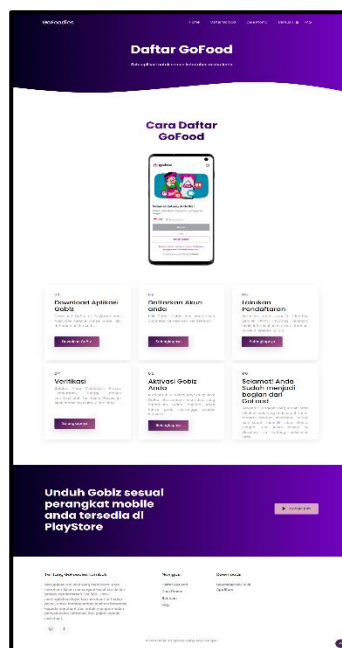




Gambar 4.29 Halaman *Homepage*

Pada Gambar 4.29 merupakan implementasi dari halaman *homepage* yang muncul pertama kali ketika *merchant* mengakses *website*.

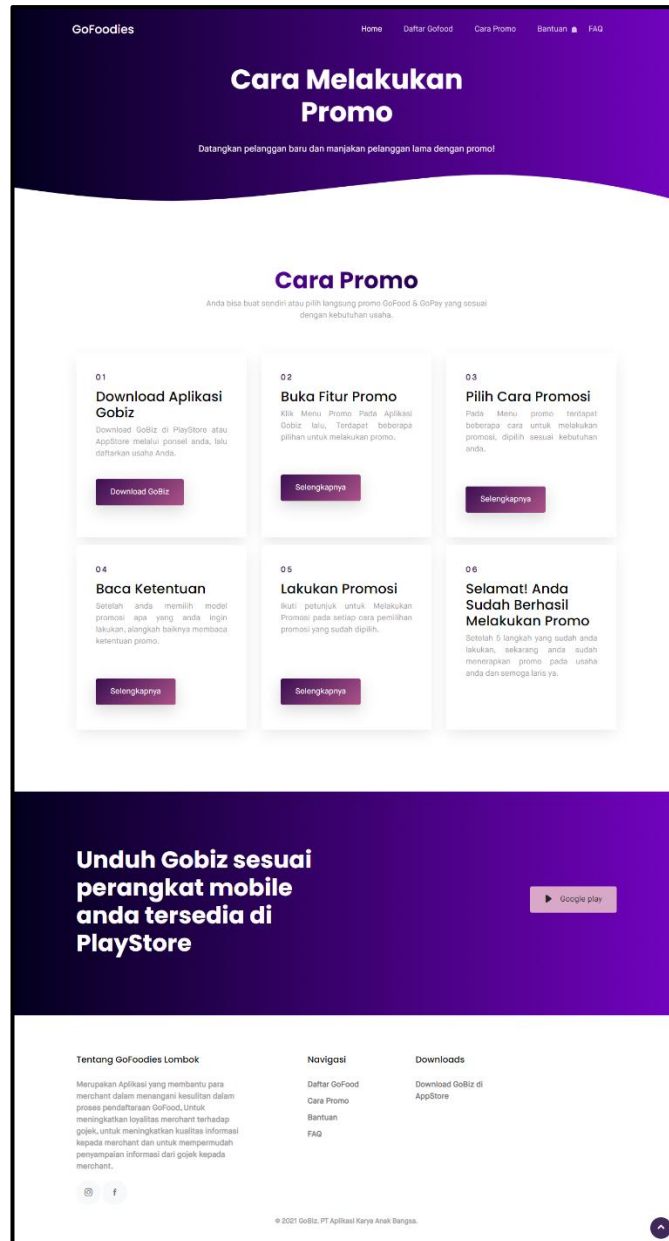
2. Halaman Daftar Gofood



Gambar 4.30 Halaman Daftar Gofood

Pada Gambar 4.30 merupakan implementasi dari halaman Daftar Gofood yang digunakan bagi *merchant* untuk mengetahui info cara-cara mendaftar gofood.

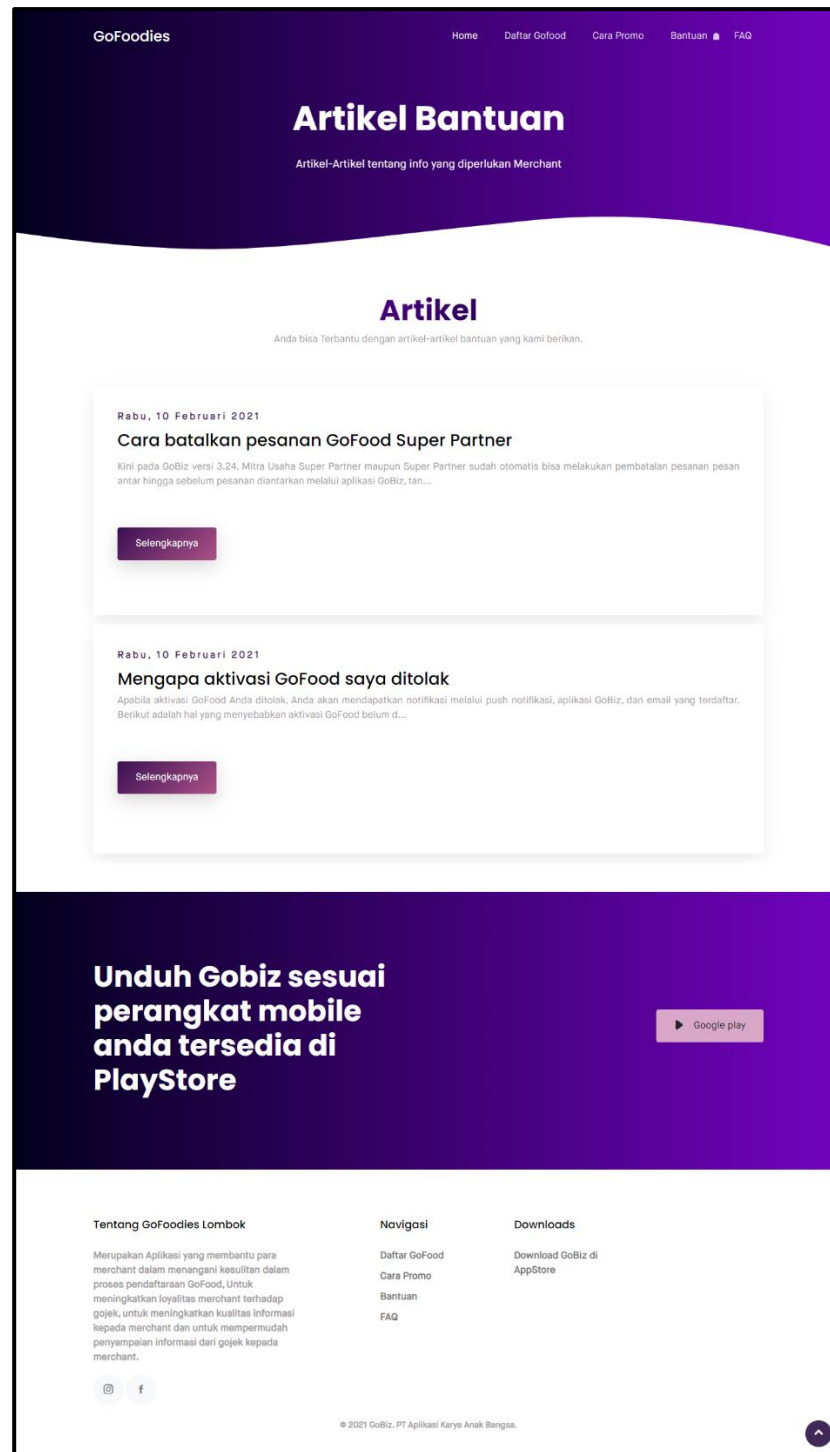
3. Halaman Cara Promo



Gambar 4.31 Halaman Cara Promo

Pada Gambar 4.31 merupakan implementasi dari halaman cara promo yang digunakan bagi *merchant* untuk mengetahui info cara-cara melakukan promosi.

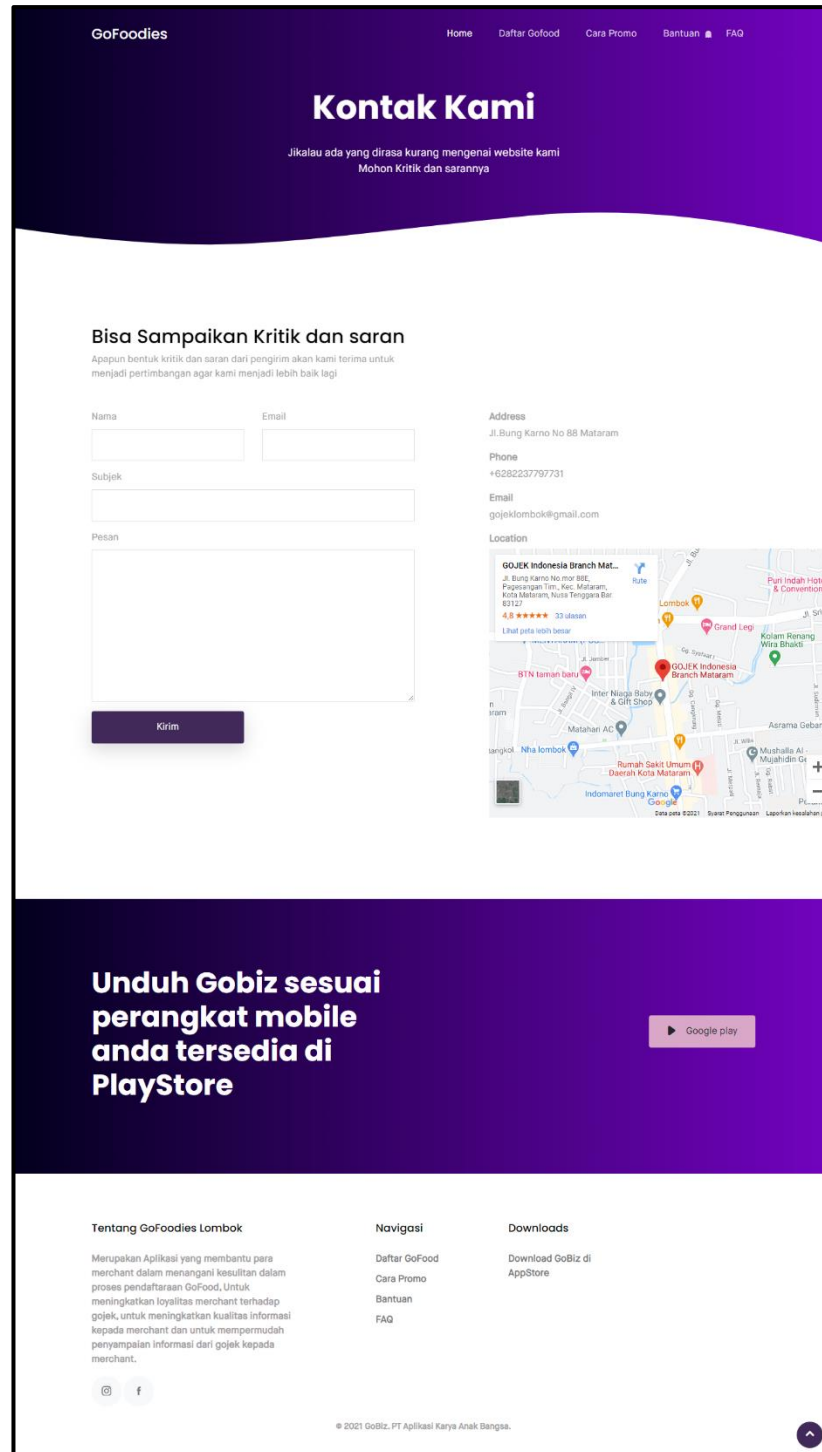
4. Halaman Artikel Bantuan



Gambar 4.32 Halaman artikel bantuan

Pada Gambar 4.32 merupakan implementasi dari halaman artikel bantuan yang digunakan bagi *merchant* untuk mengetahui info artikel bantuan untuk masalah *merchant*.

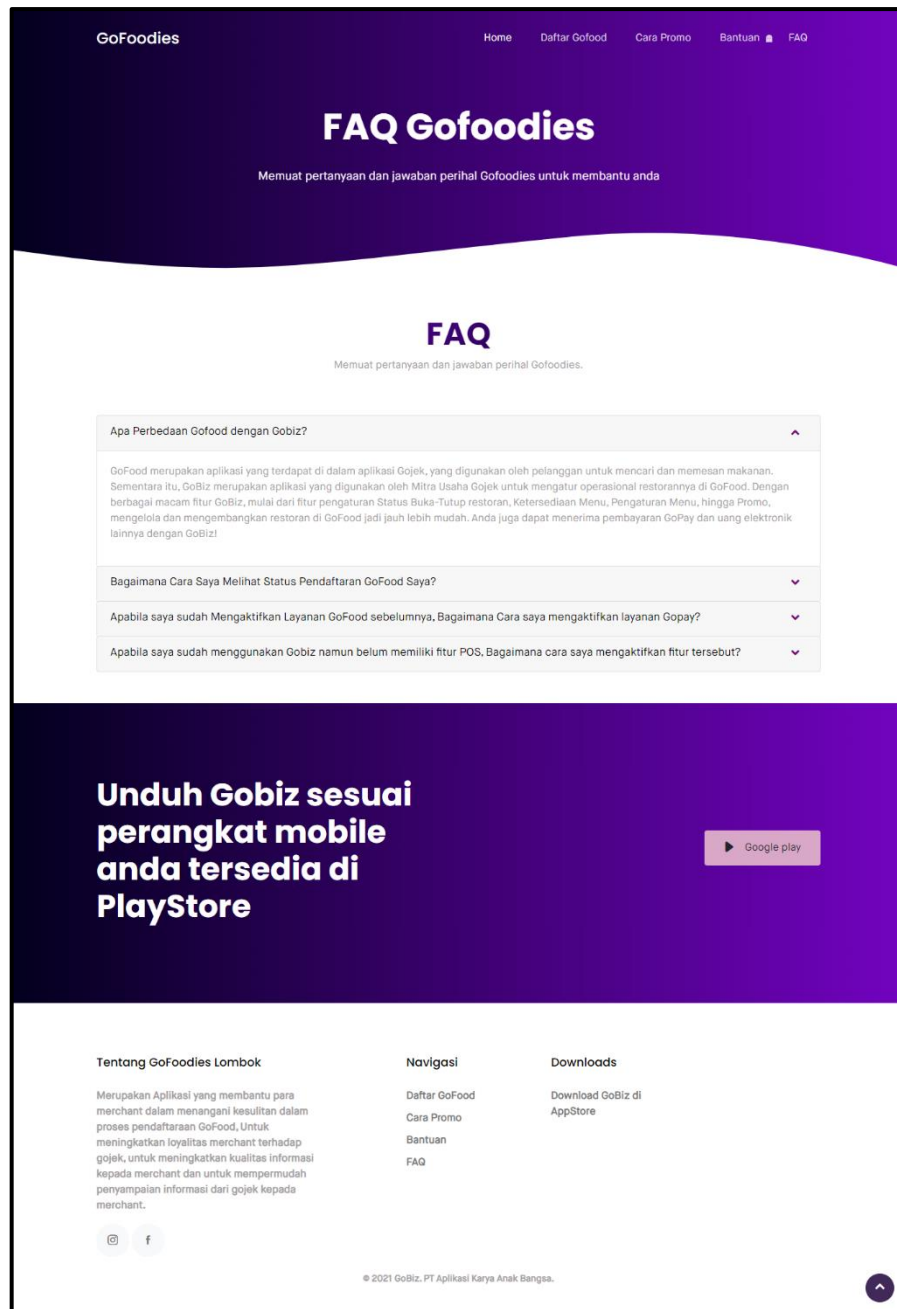
5. Halaman kontak kami



Gambar 4.33 Halaman kontak kami

Pada Gambar 4.33 merupakan implementasi dari halaman kontak kami yang digunakan bagi *merchant* untuk dapat pelayanan kontak bila ada pertanyaan maupun kritik dan saran.

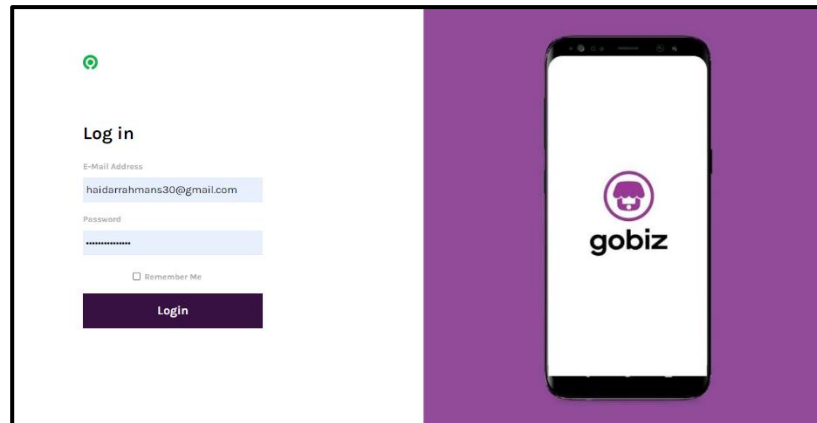
6. Halaman FAQ



Gambar 4.34 Halaman FAQ

Pada Gambar 4.34 merupakan implementasi dari halaman FAQ yang digunakan bagi *merchant* untuk mengetahui info FAQ untuk masalah *merchant*.

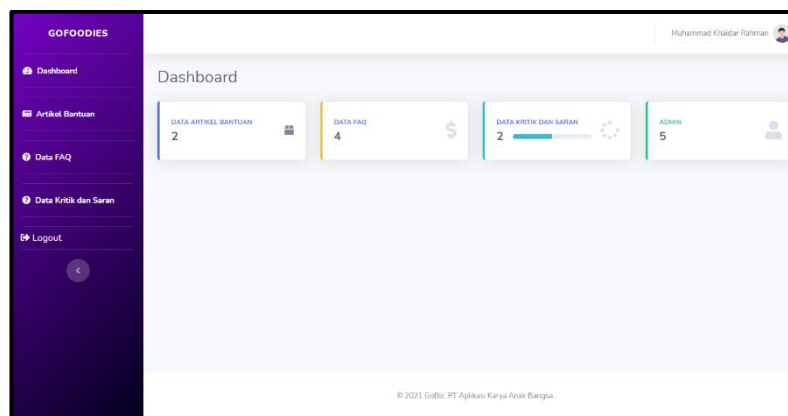
7. Halaman *Login* admin



Gambar 4.35 Halaman *login* admin

Pada Gambar 4.35 merupakan implementasi dari halaman *login* yang digunakan oleh admin untuk masuk ke dalam halaman admin.

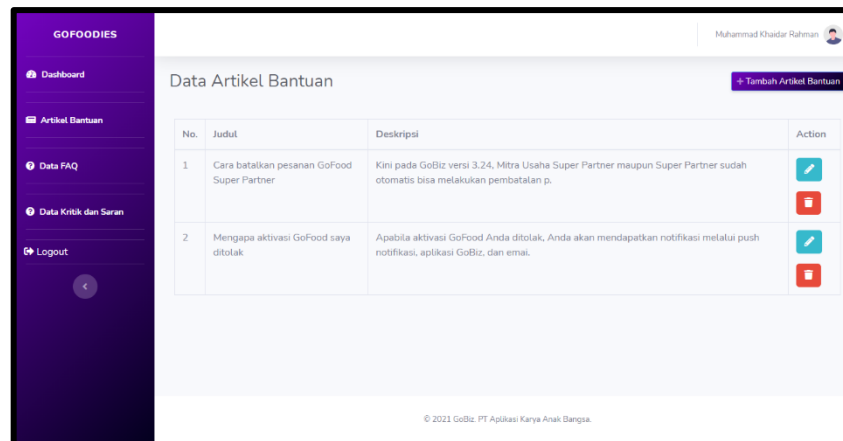
8. Halaman *dashboard* admin



Gambar 4.36 Halaman *dashboard*

Pada Gambar 4.36 merupakan implementasi dari halaman *dashboard* admin yang akan muncul pertama kali ketika admin sudah masuk ke dalam sistem. Terdapat beberapa pilihan fitur seperti kelola artikel bantuan, kelola FAQ, kelola kritik dan saran dan melihat berapa jumlah admin pada sistem gofoodies lombok.

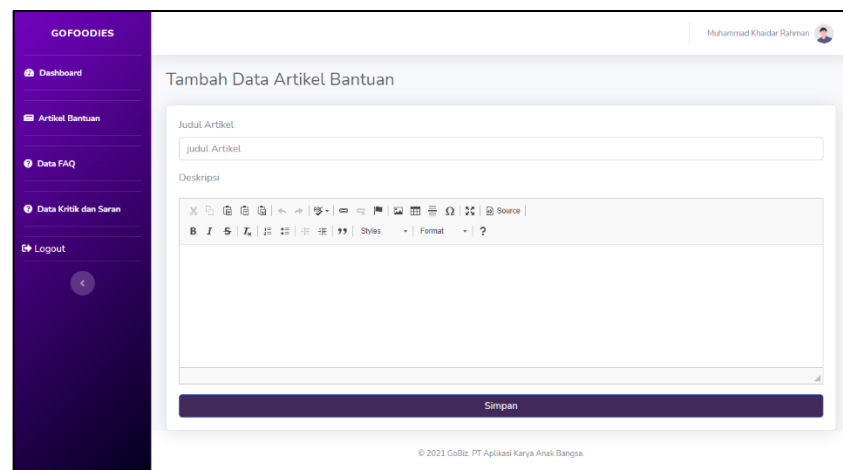
9. Halaman Kelola Data Artikel



Gambar 4.37 Halaman Kelola Data artikel

Pada Gambar 4.37 merupakan implementasi dari halaman Kelola Data artikel, dimana pada halaman ini admin dapat melakukan pengolahan data berupa melihat data, mengedit dan menghapus data artikel.

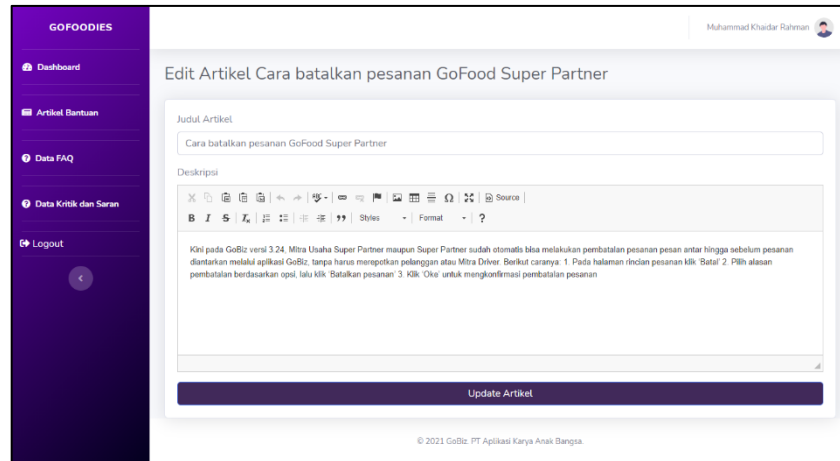
10. Halaman tambah data artikel



Gambar 4.38 Halaman tambah data artikel

Pada Gambar 4.38 merupakan implementasi dari proses tambah data artikel yang dimana admin mengisi form tambah data artikel bantuan dan menekan tombol simpan jika sudah mengisi form tambah data artikel bantuan.

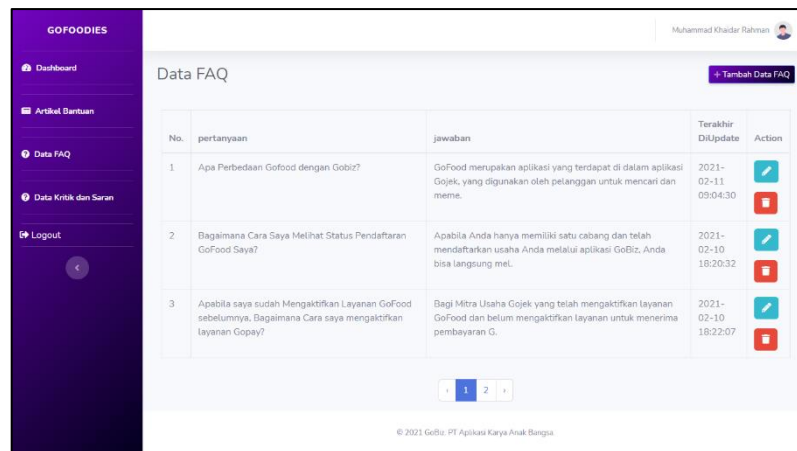
11. Halaman ubah data artikel



Gambar 4.39 Halaman ubah data artikel

Pada Gambar 4.39 merupakan implementasi dari proses ubah data artikel yang dimana admin mengisi form tambah data artikel bantuan dan menekan tombol simpan jika sudah mengisi form tambah data artikel bantuan.

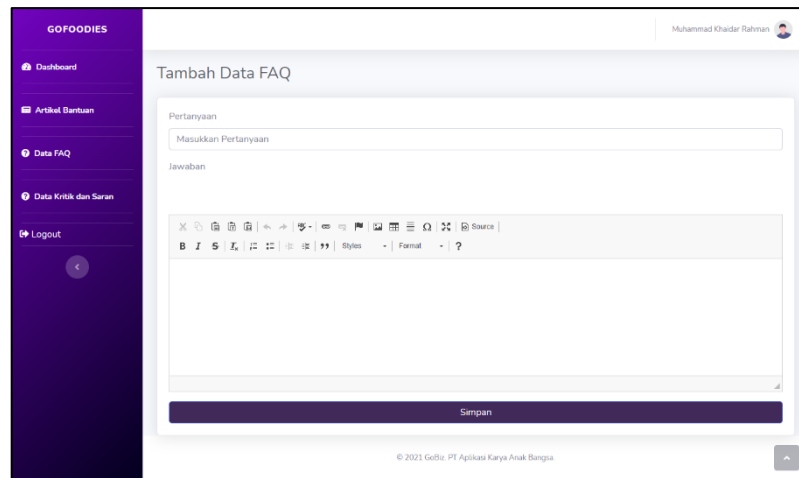
12. Halaman Kelola data FAQ



Gambar 4.40 Halaman Kelola data FAQ

Pada Gambar 4.40 merupakan implementasi dari halaman Kelola Data FAQ, dimana pada halaman ini admin dapat melakukan pengolahan data berupa melihat data, mengedit dan menghapus data FAQ.

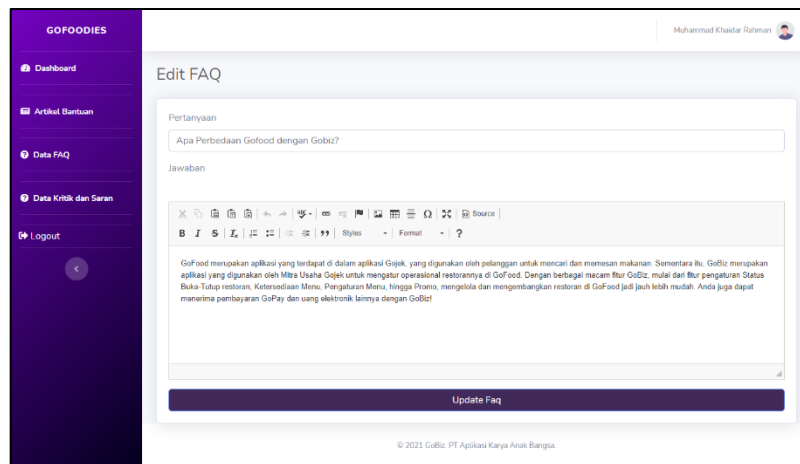
13. Halaman tambah data FAQ



Gambar 4.41 Halaman tambah data FAQ

Pada Gambar 4.41 merupakan implementasi dari proses tambah data FAQ yang dimana admin mengisi form tambah data FAQ dan menekan tombol simpan jika sudah mengisi form tambah data FAQ.

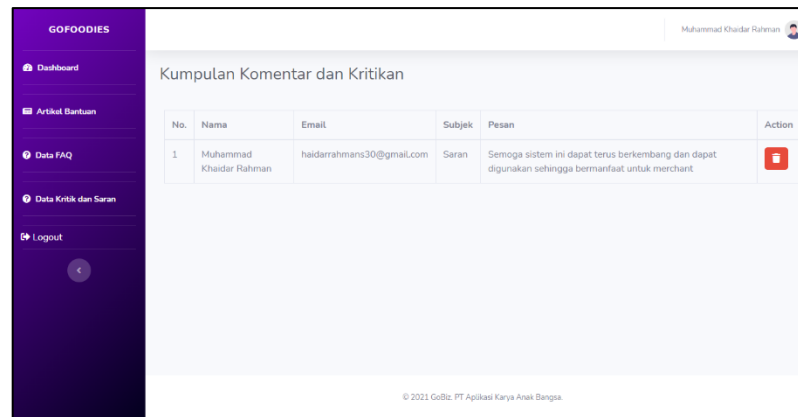
14. Halaman ubah data FAQ



Gambar 4.42 Halaman ubah data artikel

Pada Gambar 4.42 merupakan implementasi dari proses ubah data FAQ yang dimana admin mengisi form tambah data FAQ dan menekan tombol simpan jika sudah mengisi form tambah data artikel bantuan.

15. Halaman Kelola data Kritik



Gambar 4.43 Halaman Kelola data Kritik

Pada Gambar 4.43 merupakan implementasi dari halaman Kelola Data kritik, dimana pada halaman ini admin dapat melakukan pengolahan data berupa melihat data, mengedit dan menghapus data kritik.

4.4 Hasil Pengujian

Setelah dilakukan pengimplementasian sistem selanjutnya dilakukan pengujian sistem. Pengujian dengan menggunakan kuesioner dilakukan dengan cara mencari responden yang merupakan pegawai Gojek Indonesia Cabang Lombok untuk melakukan uji coba sistem dan menjawab pernyataan pada kuesioner yang diberikan. Parameter pengujian yang digunakan pada metode ini adalah :

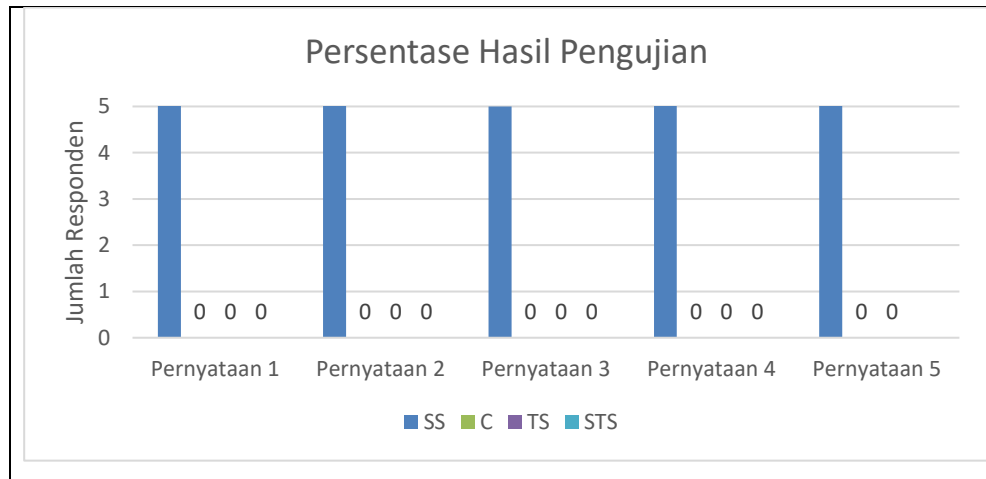
- Apakah Tampilan Interface pada Web Gofoodies menarik dan mudah dimengerti ?
- Fitur-fitur yang ada sudah lengkap dan berjalan sesuai fungsionalitasnya?
- Apakah Kualitas informasi Kepada *merchant* jadi lebih baik dan efektif ?
- Apakah sistem mudah dioperasikan ?
- Apakah Sistem dapat memudahkan pekerjaan Divisi *merchant*?

Responden akan diberikan pilihan jawaban dari pertanyaan-pertanyaan di atas, yaitu :

- SS = Sangat Setuju
- S = Setuju
- C = Cukup

- d. TS = Tidak Setuju
- e. STS = Sangat Tidak Setuju

Hasil rekapitulasi jawaban responden dihitung dari jawaban berdasarkan pengisian kuesioner dari masing-masing pernyataan. Berikut pada gambar 4.44 merupakan grafik persentase jawaban rata-rata hasil pengujian dari 5 responden dari divisi *merchant* dan divisi *marketing* :



Gambar 4.44 Persentase hasil pengujian

Pada Gambar 4.44 merupakan hasil dari persentase pengujian dari kuesioner yang diberikan mendapatkan nilai untuk pernyataan 1 nilai Sangat Setuju diberikan oleh 5 orang responden, untuk pernyataan 2 nilai Sangat Setuju diberikan oleh 5 orang responden, untuk pernyataan 3 nilai Sangat Setuju diberikan oleh 5 orang responden, untuk pernyataan 4 nilai Sangat Setuju diberikan oleh 5 orang responden dan untuk pernyataan 5 nilai Sangat Setuju diberikan oleh 5 orang responden. Sehingga dapat disimpulkan bahwa dari responden setuju dengan sistem yang telah dibangun.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil Praktek Kerja Lapangan di Gojek Indonesia Cabang Lombok berupa Sistem Informasi Gofoodies Lombok berbasis *website* dengan menggunakan *framework Laravel* dan *Bootstrap*, maka didapatkan kesimpulan sebagai berikut :

1. Sistem Informasi Gofoodies Lombok berbasis *website* yang dibangun memiliki layanan yang dibutuhkan oleh *merchant* seperti bagaimana cara daftar GoFood, cara melakukan promo dan memberikan layanan bantuan-bantuan untuk *merchant* sehingga dapat meningkatkan loyalitas *merchant* terhadap Gojek.
2. Rancangan Sistem Informasi Gofoodies Lombok berbasis *website* dibuat berdasarkan hasil analisa kualitas penyajian informasi terhadap *merchant*, sehingga dengan adanya Rancangan sistem ini dapat meningkatkan kualitas informasi kepada *merchant*.
3. Sistem Informasi Gofoodies Lombok berbasis *website* yang dibangun memiliki layanan yang dibutuhkan oleh *merchant* seperti memberikan layanan untuk memberikan kritik dan saran dan memberikan layanan bantuan-bantuan berupa artikel dan FAQ untuk *merchant* sehingga dapat meningkatkan keterikatan *merchant* terhadap Gojek.

5.2 Saran

Adapun saran yang dapat diberikan penulis agar Sistem Informasi Gofoodies Lombok berbasis *website* ini menjadi lebih baik di masa yang akan datang adalah sebagai berikut:

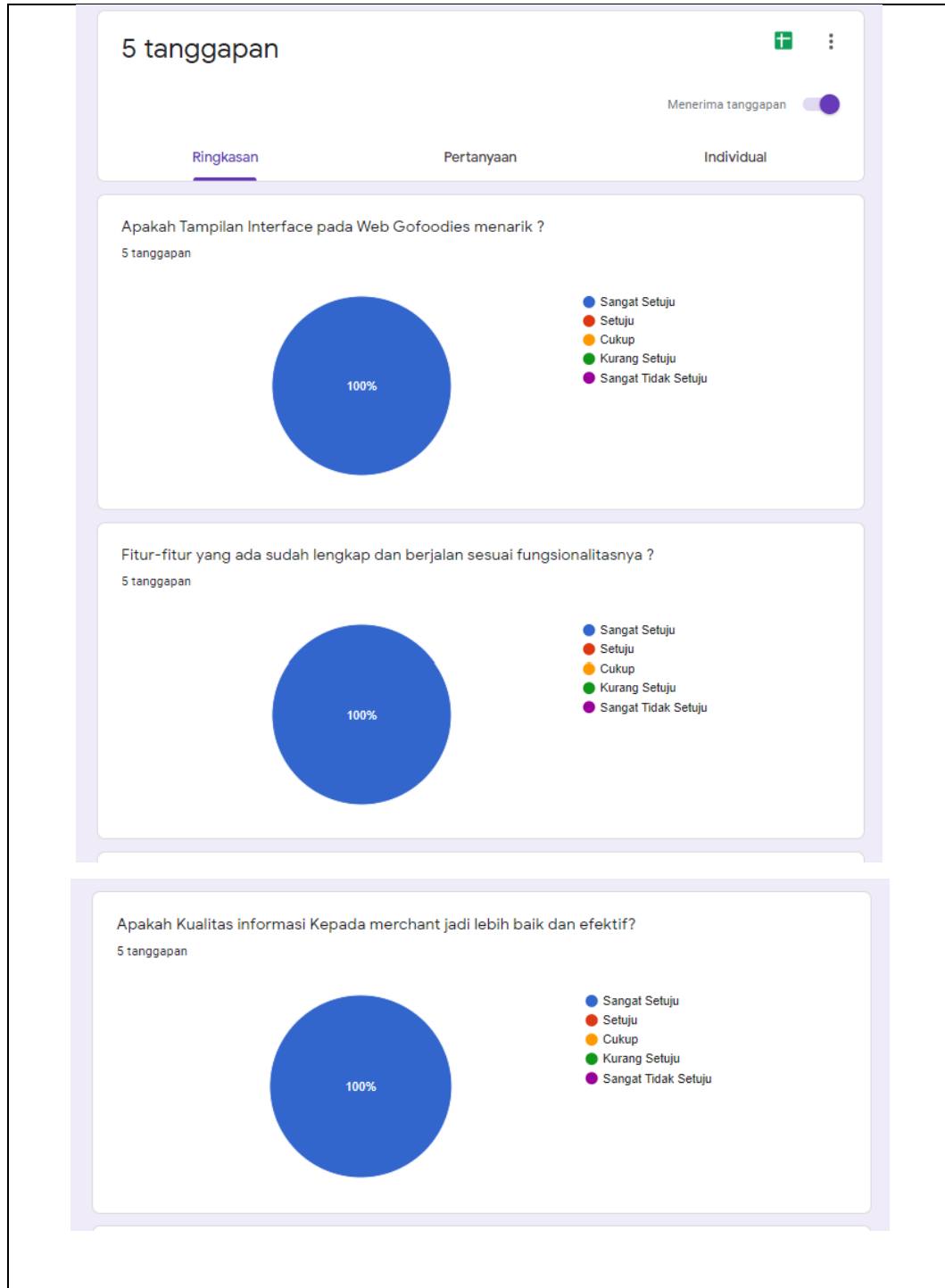
1. Perlunya dilakukan analisa dan perancangan lebih mendalam terhadap kebutuhan *merchant* guna membangun sistem yang sesuai dengan kebutuhan *merchant* Gojek dalam waktu yang lebih singkat.
2. Dalam pengembangan selanjutnya, diharapkan Sistem Informasi Gofoodies Lombok berbasis *website* dapat menjadi lebih kompleks dalam menu maupun fungsinya agar dapat dimanfaatkan lebih maksimal.

DAFTAR PUSTAKA

- [1] Hendini, Ade. 2016. “Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak).” *Jurnal Khatulistiwa Informatika* IV(2): 107–16.
- [2] Hutahaean, Jeperson. 2015. *Konsep Sistem Informasi*. Sleman: deepublish.
- [3] Pamungkas, Canggih Aji. 2017. *Pengantar Dan Implementasi Basis Data*. Sleman: deepublish.
- [4] Sumiati, S.E. Anjarwani, M.A. Albar. 2018. “Rancang Bangun Sistem Informasi Pengelolaan Surat dan Kearsipan pada Sekertariat Daerah Provinsi NTB berbasis WEB.” *J-COSINE* 1(2): 9–11.
- [5] Sutopo, Priyo, Dedi Cahyadi, and Zainal Arifin. 2016. “Sistem Informasi Eksekutif Sebaran Penjualan Kendaraan Bermotor Roda 2 Di Kalimantan Timur Berbasis Web.” 11(1).
- [6] H. Hidayat, *Cara Instan Menguasai Pemrograman Website Secara Otodidak*. Jakarta Barat: Agogos Publishing, 2016.
- [7] Rozi A. Zaenal & Community Smit Dev, 2015. “*Bootstrap Design Framework*”, Jakarta: PT Elex Media Komputindo.

LAMPIRAN

Lampiran 1.



Apakah sistem mudah dioperasikan ?

5 tanggapan



- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah interface sistem mudah dimengerti ?

5 tanggapan



- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah Sistem dapat memudahkan pekerjaan Divisi merchant ?

5 tanggapan



- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Pengujian Sistem Gofoodies

Perlunya dilakukan pengujian terhadap sistem gofoodies yang sudah dirancang agar mengetahui apakah sudah memenuhi tujuan dari target pembuatan sistem.

Apakah Tampilan Interface pada Web Gofoodies menarik ?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Fitur-fitur yang ada sudah lengkap dan berjalan sesuai fungsionalitasnya ?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah Kualitas informasi Kepada merchant jadi lebih baik dan efektif?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah sistem mudah dioperasikan ?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah interface sistem mudah dimengerti ?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Apakah Sistem dapat memudahkan pekerjaan Divisi merchant ?

- Sangat Setuju
- Setuju
- Cukup
- Kurang Setuju
- Sangat Tidak Setuju

Kirim

Konten ini tidak dibuat atau didukung oleh Google. [Laporkan Penyalahgunaan](#) - [Persyaratan Layanan](#) - [Kebijakan Privasi](#)

Google Formulir

Lampiran 2.

