

**PROPOSAL TUGAS AKHIR**

**APLIKASI SISTEM INFORMASI PENCARIAN DAN  
RESERVASI KAFE DI MEDAN DENGAN API MAPS  
BERBASIS MOBILE DAN WEB**

Oleh

**HANDOKO**  
NIM. 131112080

**RANDY MUHROJI**  
NIM. 131111661

Disetujui Oleh:

Dosen Pembimbing I,



Sunario Megawan, S.Kom., M.Kom.

Dosen Pembimbing II,



Felix, S.Kom., M.Kom.

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
MIKROSKIL  
MEDAN  
2017**

## 1. Latar Belakang

Kafe adalah tempat untuk makan dan minum sajian cepat saji dan menyuguhkan suasana santai atau tidak resmi, selain itu juga merupakan suatu tipe dari restoran yang biasanya menyediakan tempat duduk didalam dan diluar restoran. Kebanyakan kafe tidak menyajikan makanan berat namun lebih berfokus pada menu makanan ringan seperti kue, roti, sup, dan minuman (Marsum, 2005). Kafe biasanya dijadikan sebagai tempat pertemuan atau sekedar tempat kumpul untuk berdiskusi yang didominasi oleh kalangan remaja maupun mahasiswa. Namun maraknya kafe yang muncul membuat pengunjung kafe juga ikut meningkat, sebagaimana pada saat bulan ramadhan kafe-kafe banyak digunakan oleh masyarakat kota medan untuk melakukan kegiatan buka puasa bersama. Salah satu kesulitan untuk menentukan kafe dalam kegiatan buka puasa bersama adalah pada saat penentuan dan pencarian kafe yang ingin dikunjungi. Pada saat melakukan reservasi tempat pihak kafe tersebut mengharuskan setiap pelanggan untuk menentukan jumlah tempat yang tersedia beserta sejumlah menu yang dipesan. Pelanggan diharuskan datang langsung atau telepon untuk mengetahui menu yang tersedia pada kafe, maupun kapasitas meja yang tersedia pada kafe tersebut hal ini akan sangat sulit dilakukan jika hanya melalui telepon. Meski saat ini telah hadir aplikasi Zomato (aplikasi mencari resto dan tempat makan di Jakarta) aplikasi ini tidak memungkinkan pelanggan melakukan pemesanan secara online dan terfokus pada informasi kafe serta testimonial dari para pelanggan dan hanya terbatas di Jakarta.

Dari masalah tersebut dibutuhkan sebuah sistem yang dapat digunakan untuk melakukan pencarian kafe dan reservasi secara online, selain itu user juga dapat memesan makanan dan minuman sekaligus dan dapat melakukan pencarian kafe yang difilter berdasarkan nama dan lokasi kafe. Sistem sekaligus dapat memberikan kemudahan bagi pelanggan untuk melihat detail kafe maupun informasi detail menu berupa harga maupun kategori makanan (halal, non halal, *vegetarian*, dsb). Pada API google maps memungkinkan user menemukan kafe terdekat menggunakan *routing get direction* yang tersedia pada GPS smartphone pengguna. Sistem

pembayaran pada pemesanan kafe menggunakan sistem rekening bersama yang dilakukan via transfer antar rekening bank.

Berdasarkan uraian di atas, dibangunlah aplikasi yang dapat memenuhi kebutuhan pencarian dan reservasi kafe . Oleh sebab itu, Tugas Akhir ini diberi judul **“Aplikasi Sistem Informasi Pencarian Dan Reservasi Kafe Dengan API Maps Berbasis Mobile Dan Web”**

## **2. Rumusan Masalah**

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dijabarkan sebagai berikut :

1. Sulitnya mencari dan menentukan kafe yang sesuai untuk melakukan pertemuan, karena aplikasi saat ini tidak memberikan informasi jumlah tempat yang tersedia.
2. Aplikasi sistem informasi kafe saat ini hanya sebatas informasi menu makanan dan fasilitas yang tersedia dari kafe tersebut, dan tidak terdapat mekanisme reservasi yang memudahkan user untuk langsung melakukan pemesanan tempat.

## **3. Tujuan dan Manfaat**

Tujuan tugas akhir ini adalah mengembangkan sebuah aplikasi yang memungkinkan pelanggan melakukan pencarian lokasi kafe terdekat dengan menggunakan API google maps serta pemesanan kafe secara online, aplikasi mengetahui informasi detail kafe dan informasi detail menu yang terdapat pada kafe

Manfaat tugas akhir ini adalah sebagai berikut:

1. Menyediakan wadah kepada pemilik-pemilik kafe mempromosikan kafanya.
2. Mempermudah pencarian lokasi kafe terdekat dengan menggunakan *routing get direction* yang tersedia pada GPS *smartphone* User
3. Menyediakan fitur Reservasi kafe secara online, sehingga pelanggan tidak perlu datang ke kafe untuk melakukan reservasi dan mengetahui berapa meja yang tersedia pada kafe.

4. Mempermudah pelanggan mendapatkan informasi detail kafe serta promo-promo potongan harga sebelum berkunjung ke kafe.

#### **4. Batasan Masalah**

Batasan masalah pada tugas akhir ini adalah:

1. Fitur-fitur yang terbagi sesuai pada kategori pengguna:
  - a. Administrator
    - i. Mengelola data pengguna dan data kafe;
    - ii. Menverifikasi data kafe;
    - iii. Mengelola pembayaran reservasi kafe;
  - b. Administrator Kafe
    - i. Membuat akun untuk dapat mengakses fitur administrator kafe pada sistem, fitur ini tersedia pada *platform web*;
    - ii. Mengelola daftar menu pada Kafe, fitur ini tersedia pada *platform web*;
    - iii. Mengelola promo pada menu, fitur ini tersedia pada *platform web*;
  - c. Pengguna
    - i. Membuat akun untuk dapat mengakses semua pengguna;
    - ii. Melakukan reservasi tempat pada kafe, fitur ini tersedia pada platform web dan mobile dan reservasi dilakukan paling lama 1 hari sebelum waktu reservasi;
    - iii. Mengelola profil, komentar serta memberikan rating pada kafe, fitur ini tersedia pada platform web dan mobile;
    - iv. Melakukan pencarian kafe terdekat menggunakan *routing get direction* yang tersedia pada GPS smartphone pengguna, fitur ini hanya tersedia pada platform mobile;
  - d. Guest
    - i. Melihat informasi detail kafe dan menu;
    - ii. Melakukan pencarian kafe terdekat menggunakan *routing get direction* yang tersedia pada GPS smartphone pengguna, fitur ini hanya tersedia pada platform;

2. *Mobile Application* sebagai produk dari tugas akhir ini hanya dapat di-install / pasang pada *smartphone* yang menggunakan sistem operasi Android dan memiliki keterbatasan fitur dari sisi *administrator panel* yang hanya dapat diakses di *web application*.
3. *Web application* sebagai produk dari tugas akhir ini memiliki keterbatasan fitur dari sisi pencarian lokasi kafe terdekat.
4. Sistem pembayaran pada pemesanan kafe menggunakan sistem rekening bersama yang pengimplementasiannya bersifat simulasi, simulasi yang dimaksud adalah pembayaran dilakukan via transfer antar rekening bank, dengan pengelolaan rekening pihak ketiga dilakukan secara manual oleh administrator sistem.

## **5. Metodologi Pengembangan Sistem**

Untuk menyelesaikan tugas akhir ini, diimplementasikan *waterfall* sebagai metodologi pengembangan sistem. Berikut tahapan proses yang harus dilaksanakan:

### **1. Requirement**

Tahapan ini bertujuan mengumpulkan data yang diperlukan dalam proses bisnis. Beberapa proses yang terjadi dalam tahapan ini:

- a. Menentukan pengguna yang terlibat dalam aplikasi beserta fungsi kontrol masing-masing pengguna.
- b. Melakukan kajian terhadap proses bisnis yang terjadi dalam proses pemesanan menu yang terdapat pada kafe.

### **2. Analysis**

Analisis terhadap sistem yang sedang dibangun dilakukan dengan manfaat UML (*Unified Model Language*). Beberapa jenis UML yang digunakan dalam menganalisa sistem:

- a. *Usecase* digunakan untuk memperlihatkan interaksi yang terjadi antara *user* dan sistem.

- b. *Data Flow Diagram* digunakan untuk memperlihatkan aliran data yang terjadi antara *user*, sistem dan *datastore*.

### 3. *Design*

Desain yang dihasilkan dari tahapan ini berupa *mockup* atau rancangan tampilan dari sistem yang sedang dibangun.

### 4. *Implementation*

Tahapan ini dibagi dalam 2 (dua) proses utama:

- a. *Coding*

Mengimplementasikan analisis dan desain yang telah dirancang pada tahapan sebelumnya untuk membangun sistem.

- b. *Testing*

Bertujuan untuk mengevaluasi sistem yang telah dibangun.

## 6. **Tinjauan Pustaka**

### 6.1 **Kafe**

Kafe adalah tempat untuk makan dan minum sajian cepat saji dan menyuguhkan suasana santai atau tidak resmi, selain itu juga merupakan suatu tipe dari restoran yang biasanya menyediakan tempat duduk didalam dan diluar restoran. Kebanyakan kafe tidak menyajikan makanan berat namun lebih berfokus pada menu makanan ringan seperti kue, roti, sup, dan minuman (Marsum, 2005). Kafe pertama kali muncul di daerah barat.

Pengertian Kafe Pada dasarnya kafe itu sendiri mempunyai arti harafiah yaitu Restoran kecil yang melayani atau menjual makanan ringan dan minuman, kafe biasanya digunakan orang untuk rileks (Sumber: Dictionary of English Language and Culture, Longman).

Beberapa cara penyajian makanan pada kafe diantaranya adalah:

- 1. **Self Service**

Dimana pengunjung melakukan pelayanan bagi dirinya sendiri. Pengunjung datang kemudian mengambil makanan dan minuman yang mereka

inginkan kemudian menuju ke kasir dan membayar makanan mereka lalu duduk di tempat yang telah disediakan. Cara ini terkesan familiar dan bersahabat.

2. Waiter of Waitress Service to Table

Pengunjung datang lalu duduk pada kursi yang telah disediakan, kemudian pramusaji akan melayani mereka, mengantar menu dan makanan hingga membayar ke kasir, sehingga orang tidak perlu beranjak dari kursinya. Cara ini terkesan formal.

3. Counter Service

Dimana terdapat area khusus yang terdapat display makanan yang ada, biasanya digunakan untuk pelayanan yang cepat dan service tidak formal.

4. Automatic Vending

Menggunakan mesin otomatis. Pengunjung memasukkan koin lalu dari mesin keluar makanan yang dipilihnya.

## 6.2 Android Mobile Platform

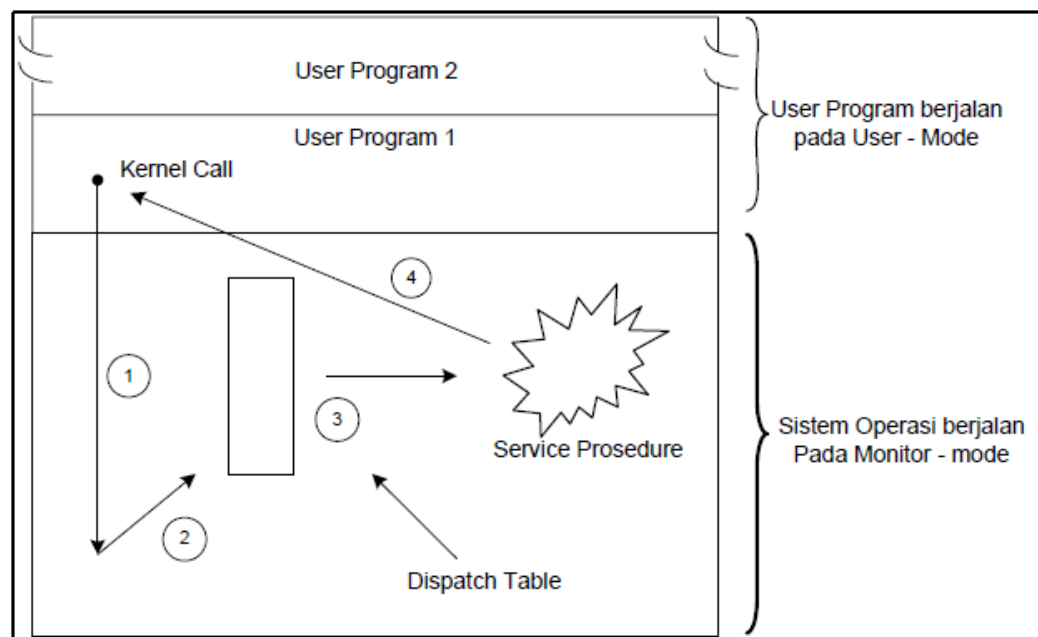
Android merupakan sebuah sistem operasi pada perangkat *mobile* seperti *smartphone* dan *PC Tablet* dan merupakan salah satu versi modifikasi dari Linux *Operating System*. Pada awalnya, Android dikembangkan oleh perusahaan *startup* yang bernama Android Inc. pada tahun 2005. Namun seiring perkembangan Android yang sangat pesat dan mampu bersaing dengan *mobile operating system* lainnya, Google Inc. mengakuisisi Android dan menjadikan Android *opensource operating system*. Pengembang dapat membuat aplikasi untuk platform android dengan menggunakan android SDK dengan menggunakan bahasa pemrograman java dan berjalan dalam Dalvik dengan mesin virtual yang telah tertanam dan berjalan di atas kernel linux. Hal ini menguntungkan bagi *vendor* (penyedia *hardware*) yang diberikan hak untuk menambahkan ekstensi yang mereka inginkan di dalam Android, seperti Oppo yang telah melakukan modifikasi pada Android yang digunakan pada *smartphone* mereka dan menamakannya ColorOS. Android juga merilis sebagian besar *code* di bawah Apache License Open-Source sehingga *mobile developer* dapat mengunduh *source code* Android secara keseluruhan. (Lee, W., M., 2011).

### 6.2.1 Arsitektur Android

Arsitektur perangkat lunak adalah struktur-struktur yang menjadi landasan untuk menentukan keberadaan komponen-komponen perangkat lunak, cara komponen perangkat lunak berinteraksi dengan perangkat keras dan organisasi antar perangkat. Arsitektur perangkat lunak diklasifikasikan menjadi 5 (lima) bentuk (Hafid, A., M., 2015):

#### 1. *Monolithic system.*

Merupakan stuktur sistem operasi sederhana yang dilengkapi dengan operasi “dual” pelayanan (*system call*) yang diberikan oleh sistem operasi. Modul *system call* dikerjakan dengan cara mengambil sejumlah parameter pada tempat yang telah ditentukan sebelumnya, seperti *register* atau *stack* dan kemudian mengeksekusi setiap instruksi *trap* pada *monitor mode*.



Gambar 1. Monolithic System

Seperti yang digambarkan pada Gambar 1., setiap layanan dieksekusi pada ruang memori yang sama sehingga mempercepat proses eksekusi. Di saat yang bersamaan penggunaan satu alamat memori membuat rumitnya



penanganan masalah karena sulit untuk dipisahkan dan dilokasikan. Kelemahan lain dari arsitektur adalah:

- a. Sulit menyediakan fasilitas pengamanan;
- b. Kurang efisien dalam penggunaan memori, karena setiap perangkat harus menjalankan seluruh kernel yang ada, sementara tidak semua kernel diperlukan untuk menjalankan suatu perangkat;
- c. Tidak berfungsinya sebuah kernel menyebabkan tidak berfungsinya sebuah perangkat secara total.

## 2. *Layered System*

Pendekatan yang digunakan pada sistem ini adalah modularisasi, yaitu membagi sistem operasi menjadi beberapa lapis (tingkat). Lapisan terendah (layer 0) adalah perangkat keras dan lapisan teratas (layer N) adalah user interface. Beberapa sistem operasi yang telah menggunakan pendekatan ini adalah THE, UNIX termodifikasi, Venus dan OS/2.

Beberapa kelebihan layered system:

- a. Semua keuntungan dari pendekatan modularisasi.
- b. Setiap modul dirancang, dibangun dan diuji secara independen. Sehingga memudahkan developer untuk melakukan penanganan kesalahan ketika terjadi kesalahan pada salah satu modul.

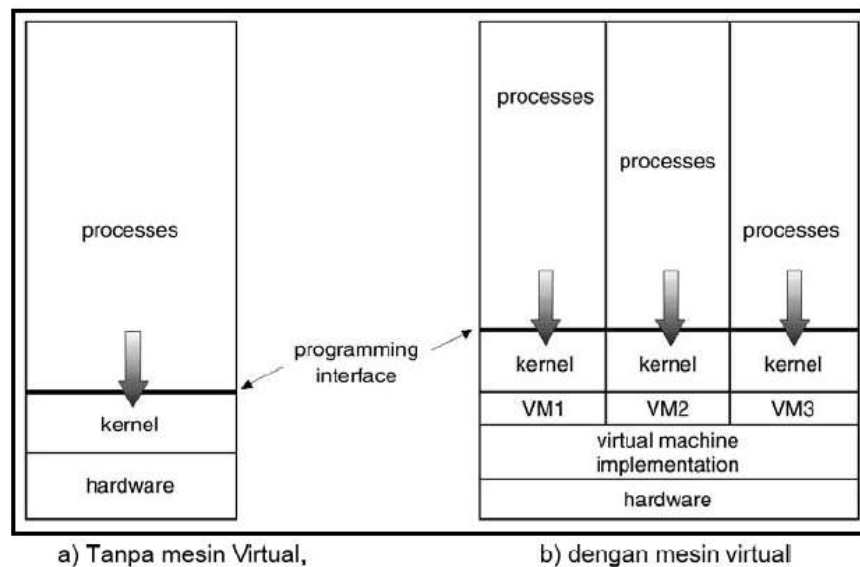
Sementara kelemahan layered system adalah adanya kemungkinan seluruh lapisan harus diprogram ulang apabila terjadi logic error pada salah satu fungsi utama pada sistem operasi. Hal ini dikarenakan layered system mengharuskan fungsi utama ditanam pada setiap lapisan.

<i>Layer</i>	<i>Nama</i>	<i>Fungsi</i>
<i>Layer – 0</i>	<i>Hardware</i>	Lapisan yang mengatur penggunaan <i>hardware</i>
<i>Layer – 1</i>	Penjadwalan CPU	Mengatur alokasi CPU dan <i>switching</i> pengaturan prosesor
<i>Layer – 2</i>	Manajemen Memori	Pengalokasian ruang memori
<i>Layer – 3</i>	<i>Console operator device driver</i>	Mengatur jalur komunikasi antar proses
<i>Layer – 4</i>	<i>Buffering untuk I/O device</i>	Penyederhana akses I/O pada lapisan atas
<i>Layer – 5</i>	<i>User Interface</i>	Mengatur tampilan aplikasi

*Gambar 2. Layered System*

### 3. *Virtual Machine System*

Pendekatan yang digunakan oleh arsitektur ini hampir sama dengan layered system, dengan tambahan berupa user interface yang menghubungkan hardware dengan kernel untuk tiap-tiap proses. Virtual machine menyediakan user interface yang identik untuk hardware yang ada. Arsitektur ini membuat ilusi atau virtual untuk beberapa proses dan mengeksekusi processor dan memori (virtual) masing-masing. Walau konsep ini cukup baik, namun sangat kompleks untuk dapat diimplementasikan karena system berjalan menggunakan metode dual mode, sementara virtual machine hanya dapat berjalan pada monitor mode atau user mode. Dampaknya adalah virtual machine harus dijalankan melalui physical user mode yang menyebabkan adanya transfer dari user mode ke monitor mode pada virtual machine. Namun arsitektur ini menghadirkan banyak fitur yang memanjakan penggunanya, misalnya penjadwalan CPU dapat membuat tampilan seolah-olah pengguna mempunyai processor sendiri atau spooling dan system file dapat menyediakan card reader virtual dan line printer virtual. Beberapa sistem operasi yang pernah menggunakan arsitektur ini adalah IBM S/370 dan IBM VM/370.



*Gambar 3. Perbandingan sitem operasi dengan dan tanpa virtual machine*

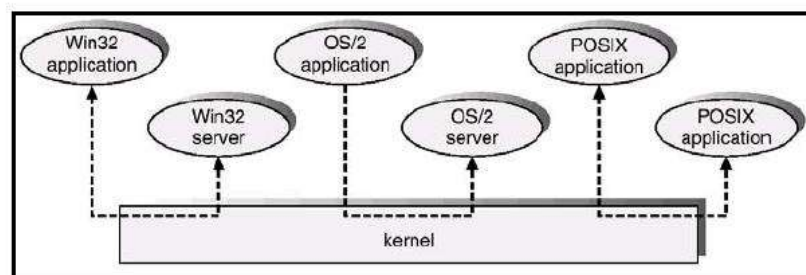
Pendekatan ini berkembang seiring perkembangan teknologi menjadi emulator operating system sehingga sistem operasi dapat menjalankan aplikasi-aplikasi untuk sistem operasi lain. Misalnya MS-Windows NT dapat menjalankan aplikasi untuk MS-DOS atau OS/3 text mode dan aplikasi Win16 dengan memanfaatkan Win32 API (system call di MS-Windows NT) salah satu kelebihan dari pendekatan ini adalah setiap proses bekerja secara independen karena dijalankan oleh virtual machine yang berbeda, sehingga kerusakan/kegagalan eksekusi pada salah satu proses tidak berdampak pada seluruh sistem yang ada. Kelebihan dan kekurangan lain dari pendekatan ini secara bersamaan adalah:

- Virtual machine menyediakan perlindungan yang lengkap bagi sumber daya sistem sehingga masing-masing virtual machine dari virtual machine yang lain. Isolasi ini berdampak pada tidak terjadinya pembagian sumber daya secara langsung.
- Arsitektur ini sangat cocok jika ingin melakukan riset dan pengembangan operating system. Pengembangan dikerjakan pada virtual machine sehingga tidak mengganggu main operating system.

- c. Konsep ini sangat sulit untuk mengimplementasikan kebutuhan dan duplikasi yang tepat pada mesin yang sebenarnya.

#### 4. *Client-server System*

Arsitektur client-server membagi setiap proses yang dikerjakan dalam beberapa modul server process yang akan mengeksekusi setiap permintaan dari user/client server untuk meminimalisir jumlah kernel yang mengeksekusi setiap proses yang ada. Karena arsitektur ini bekerja pada user mode process dan bukan monitor mode, maka server tidak dapat mengakses hardware secara langsung, yang berarti apabila terjadi kerusakan pada server file, maka kerusakan tidak menyebabkan gangguan bagian lain dari sistem, misalnya server terminal.



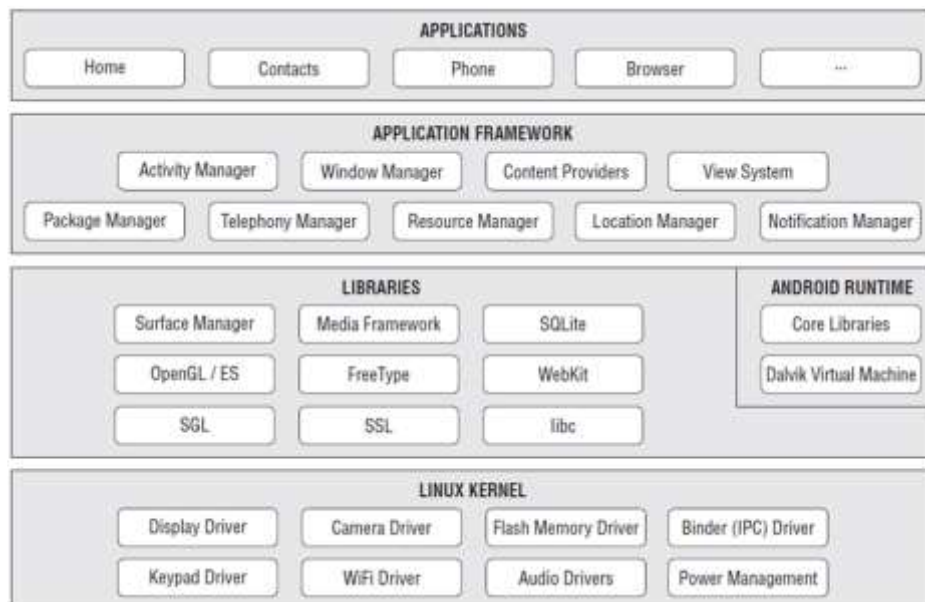
*Gambar 4. Client-Server System*

kelemahan pendekatan ini adalah lambatnya pertukaran informasi yang terjadi antara user/client dan server process menyebabkan lambatnya eksekusi masing-masing perintah. Sementara kelebihan dari arsitektur ini adalah:

- a. Dapat diadaptasikan pada sistem terdistribusi;
- b. Server tidak perlu tahu sumber (local atau by network) pesan yang dikirimkan oleh client ketika sedang menjalin komunikasi dengan server;
- c. Pengembangan dilakukan secara modular;
- d. Kesalahan pada suatu subsistem tidak akan mengganggu subsistem lain ataupun sistem secara utuh.

## 5. *Object Oriented System*

Setiap layanan akan menjadi object dengan proses, direktori, berkas sebagai properti dari objek tersebut. Beberapa sistem operasi yang menggunakan arsitektur ini adalah Eden, Choices, X-Kernel, Medusa, Clunds, Amoeba, Muse dan Sistem Operasi MS-Windows NT meski tidak secara keseluruhan. Sebagai sebuah platform, Android juga dibangun di atas platform yang dapat membantu penanganan setiap layanan. Berikut adalah gambar arsitektur Android:



*Gambar 5. Arsitektur Android mobile platform*

Berikut adalah penjelasan dari setiap lapisan pada arsitektur Android seperti Gambar 2.4. (Lee, W., M., 2011):

### a. Linux Kernel.

Lapisan ini merupakan kernel dasar dari Android yang berisi semua driver tingkat rendah untuk berbagai komponen hardware yang dipasangkan pada perangkat keras seperti smartphone. Kernel ini akan menjadi penerjemah bagi hardware untuk dapat melaksanakan fungsinya secara baik.

b. Libraries.

Mengandung seluruh code yang menyediakan fitur utama dari sistem operasi Android. Sebagai contoh, library SQLite yang menyediakan dukungan database yang dapat digunakan pula oleh aplikasi lain yang dipasang oleh pengguna di sistem operasi Android.

c. Android runtime.

Lapisan ini juga menyediakan kode yang disimpan sebagai library, namun memiliki fungsi dan tujuan yang lebih spesifik, yaitu library untuk mengembangkan aplikasi dengan menggunakan bahasa pemrograman java.

d. Application Framework.

Menjadi lapisan yang dapat digunakan oleh mobile apps developer ketika mengembangkan aplikasi yang dapat dipasang pada mobile device dengan android platform.

e. Applications.

Pada lapisan teratas ini, ditanamkan mobile applications yang sudah dipasangkan secara otomatis oleh pihak Android ataupun mobile applications yang dipasangkan oleh pengguna melalui Android market.

### 6.3 Website

Website atau situs web adalah sekumpulan halaman-halaman web yang saling berhubungan, termasuk didalamnya *homepage*, dan umumnya berada pada server yang sama, dan disiapkan dan dipelihara sebagai kumpulan informasi oleh seseorang, kelompok atau organisasi. *World Wide Web (WWW)* adalah suatu metode untuk menampilkan informasi di internet, baik berupa text, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen yang lainnya (*hypertext*) yang dapat diakses melalui *browser* (Yuhefizar, 2008). Salah satu keuntungan membangun aplikasi berbasis *website* adalah karena *website* tidak tergantung pada *platform* atau operating system perangkat yang digunakan, karena *website* dijalankan menggunakan *internet browser* yang dipasang pada operating system perangkat. Selain itu, informasi pada *website* terpusat pada *server*, sehingga ketika web

*developer* melakukan perubahan atau modifikasi terhadap *website*, pengguna tidak perlu melakukan *upgrade* terhadap aplikasi web, karena pengguna mengakses server tempat *website* diletakkan.

### **6.3.1 Klasifikasi Website**

Secara fisik, website adalah kumpulan komputer pribadi, web browser, koneksi ISP, komputer server, router dan switch yang digunakan untuk mengalirkan informasi yang menjadi wahana pertama bagi pihak terkait. Berdasarkan fungsinya, web diklasifikasikan menjadi 4 (empat) jenis, yaitu (Yuhefizar, 2008):

1. Search engine website. Jenis ini merupakan website yang memiliki kemampuan untuk melakukan pencarian berdasarkan kata kunci (keyword). Pencarian dilakukan pada server dengan tujuan menyuguhkan informasi yang lebih fariatif bagi pengguna. Website yang ditampilkan oleh search engine didasarkan pada SEO (Search Engine Optimization) yang dimiliki oleh setiap website. Menurut Matt Cutts (Head of Webmaster Google Spam), SEO dapat sangat berdampak pada index website anda ketika seseorang melakukan pencarian pada search engine jika dikombinasikan dengan optimasi lain dan berdampak pada pengalaman pengguna dan dapat mendongkrak kinerja website
2. Portal website. Jenis website ini berisi kumpulan link dan search engine terhadap seluruh halaman utama dari webiste tersebut. Salah satu contoh portal website yang ada saat ini adalah <http://it-ebooks.info>. Website ini berisi kumpulan ebooks tentang dunia IT yang dapat diunduh secara gratis dan menyediakan fitur search engine yang mengarahkan kita pada halaman Google yang khusus mencari tentang ebooks.
3. Content website. Digunakan oleh seseorang, sekelompok orang, organisasi, pemerintah bahkan negara untuk membagikan informasi 1 (satu) arah tentang pemilik webiste kepada pengguna. Salah satu contoh dari content website adalah <http://www.mikroskil.ac.id> yang berisi informasi tentang STMIK – STIE Mikroskil.

4. Application website. Website ini lebih kompleks dibandingkan 3 (tiga) jenis website yang lain karena application website memberikan wadah bagi pengguna untuk berinteraksi dengan pengguna untuk menjalankan fungsi utama dari website. Salah satu contoh dari website ini adalah <https://www.lazada.com> yang dapat digunakan oleh pengguna untuk membeli barang secara online.

#### **6.4 API (*Application program interface*)**

Application program interface (API) adalah sekumpulan perintah, fungsi, serta protokol yang digunakan oleh programmer untuk membangun sebuah aplikasi, API menentukan bagaimana komponen perangkat lunak harus berinteraksi. Selain itu, API digunakan ketika promrograman komponen pada Atar muka User (Grafical User Interface). Sebuah API yang baik akan membuatnya lebih mudah untuk mengembangkan program dengan menyediakan semua komponen, dan programmer kemudian menempatkan semua komponen secara bersama-sama (Beal, V., 2016). API bekerja seperti middleware pada sebuah aplikasi dengan tujuan yang spesifik dan tidak bekerja pada operating system/platform tempat aplikasi dibangun. beberapa keuntungan menggunakan API:

1. Portabilitas. Aplikasi yang dibangun dengan memanfaatkan API dapat dijalankan pada platform/operating system apapun. Hal ini dikarenakan cara kerja dari API adalah dengan menggunakan system call interface yang dibuat sendiri oleh vendor yang berfungsi sebagai penghubung antara API dan system call operating system. Sehingga programmer tidak perlu memikirkan system call operating system yang akan digunakan pada platform atau interface tertentu.
2. Lebih mudah dimengerti karena menggunakan bahasa yang lebih terstruktur dan mudah dimengerti dibanding system call operating system. Hal ini membuat programmer tidak perlu mempelajari system call dari setiap platform yang akan digunakan.



## 6.5 Google Maps API

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam web atau aplikasi yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Cara membuat Google Maps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta Java Script, serta koneksi Internet yang sangat stabil. Dengan menggunakan Google Maps API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia.

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, database, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman HTML, Javascript, dan AJAX, serta beberapa bahasa pemrograman lainnya. Gambar-gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan database pada web server Google untuk menampilkan gabungan dari potongan-potongan gambar yang diminta. Seluruh citra yang ada diintegrasikan ke dalam suatu database pada Google server yang nantinya akan dipanggil sesuai kebutuhan permintaan. Bagian-bagian gambar peta yang merupakan gabungan dari gambar-gambar yang berukuran 256 x 256 pixel. Tiap-tiap 256 x 256 tile mewakili gambar tertentu dalam longitude, latitude, dan zoom level tertentu.

Pada Google Maps API terdapat empat jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. Roadmap

Menampilkan peta biasa duadimensi pada tampilan Google Maps.

## 2. Satellite

Menampilkan foto satelit pada tampilan Google Maps.

## 3. Terrain

Menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai pada tampilan Google Maps.

## 4. Hybrid

Menunjukkan foto satelit yang diatasnya tergambar pula tampilan pada roadmap(jalan dan nama kota) pada tampilan Google Maps.

### 6.5.1 Places Library

Berbagai fungsi di *Google Places JavaScript Library* memungkinkan aplikasi Anda mencari tempat (didefinisikan dalam API ini sebagai tempat usaha, lokasi geografis, atau tempat menarik yang menonjol) yang terdapat dalam area yang d ditentukan, seperti batas-batas peta, atau di sekitar titik tetap. *Google Places API* menawarkan fitur *autocomplete* yang bisa Anda gunakan untuk memberikan aplikasi Anda perilaku prediksi-pencarian pada bidang pencarian *Google Maps*. Bila pengguna mulai mengetik alamat, *autocomplete* akan mengisi sisanya. Layanan *Places* adalah *library* terpisah dari kode utama Maps API JavaScript.

Dengan menggunakan layanan *Places* anda dapat melakukan 4 jenis pencarian:

### 1. *Nearby Search*

Mengembalikan daftar tempat terdekat berdasarkan lokasi pengguna.

### 2. *Text Search*

Mengembalikan daftar tempat terdekat berdasarkan string pencarian, misalnya "Pizza". Layanan ini mengembalikan daftar tempat yang cocok dengan string teks

dan kecondongan lokasi yang telah diatur. Respons pencarian akan menyertakan daftar tempat.

### 3. *Radar Search*

Mengembalikan daftar besar berisi banyak tempat dalam radius pencarian yang ditetapkan, namun kurang detail dibandingkan *Nearby Search* atau *Text Search*. *Radar Search* memungkinkan Anda mencari tempat dalam radius pencarian yang ditetapkan dengan kata kunci, ketikan, atau nama. *Radar Search* mengembalikan hasil lebih banyak daripada *Nearby Search* atau *Text Search*, namun berisi bidang yang lebih sedikit

### 4. *Place Details*

Mengembalikan informasi lebih detail tentang tempat tertentu, termasuk ulasan pengguna. Selain menyediakan daftar tempat dalam suatu area, layanan Places juga mengembalikan informasi detail tentang tempat tertentu. Setelah tempat dikembalikan dalam *respons* pencarian tempat, *placeID* bisa digunakan untuk meminta detail tambahan tentang tempat itu, seperti alamat lengkap, nomor telepon, *rating* pengguna dan ulasan, dll.

## 7. Daftar Pustaka

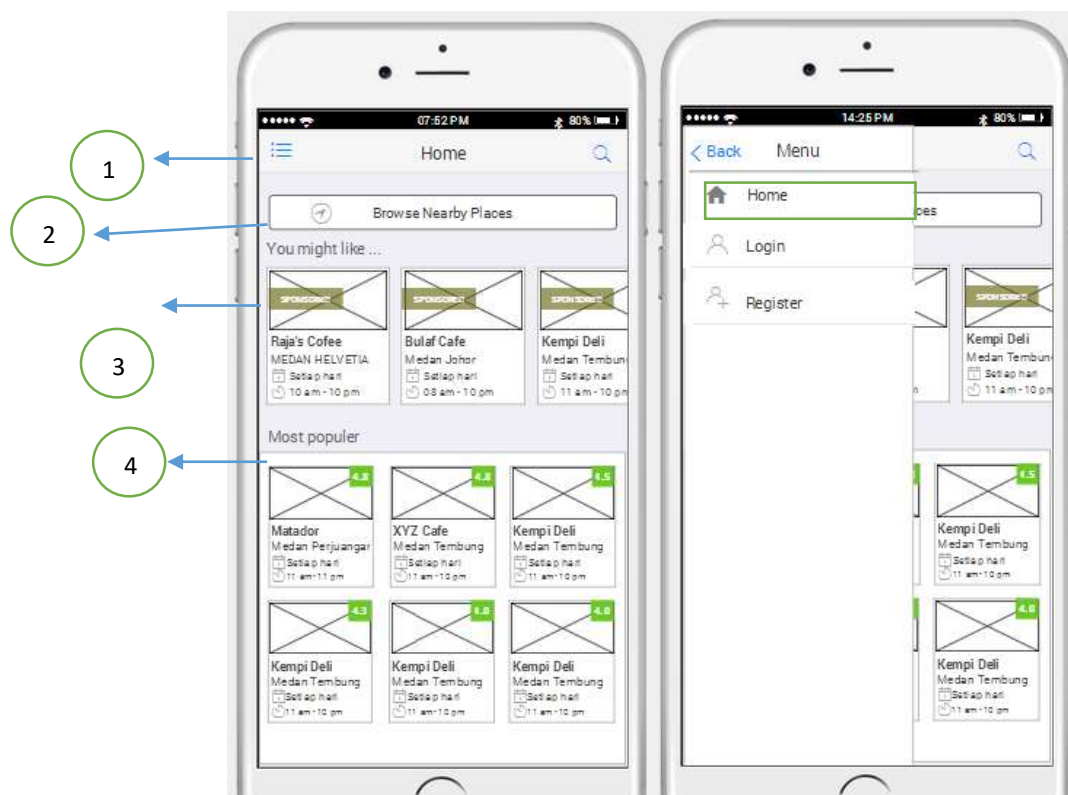
- Beal, V., 2016, API – *application programming interface*, tersedia pada: <http://www.webopedia.com/term/a/api.html>, tanggal akses: 8 Februari 2017
- The Free Dictionary by Farlex, tersedia pada: <http://www.thefreedictionary.com/Website>, tanggal akses: 8 Februari 2017
- Marsum WA, 2001. Restoran dan Segala Permasalahannya, Yogyakarta:ANDI Yogyakarta
- Pengertian Android Mobile, tersedia pada: <https://www.openhub.net/p/android>, tanggal akses : 20 Februari 2017
- Hafid, A., M., 2015, Arsitektur Sistem Operasi dan Penjelasan Lengkap, tersedia pada: <http://www.blog-hafid25.com/2015/09/arsitektur-sistem-operasi-dan.html>, tanggal akses 20 Februari 2017
- Baron S., Peter Z., Vadim T., High Performance MySQL, Edisi ke 3, tersedia pada [:ftp://203.157.240.9/pub/docs/High.Performance.MySQL.3rd.Edition.pdf](ftp://203.157.240.9/pub/docs/High.Performance.MySQL.3rd.Edition.pdf), tanggal akses: 2 Maret 2017
- Pengertian Framework Laravel dan MVC, tersedia pada [:https://library.binus.ac.id/eColls/eThesisd/doc/Bab2/2015-1-00359-IF%20Bab2001.pdf](https://library.binus.ac.id/eColls/eThesisd/doc/Bab2/2015-1-00359-IF%20Bab2001.pdf), tanggal akses: 2 Maret 2017
- Pengertian Ionic Framework tersedia pada [:http://digilib.mercubuana.ac.id/manager/n!@file\\_skripsi/Isi2355686478758.pdf](http://digilib.mercubuana.ac.id/manager/n!@file_skripsi/Isi2355686478758.pdf), tanggal akses: 2 Maret 2017
- Google Documentation, tersedia pada : [https://developers.google.com/maps/documentation/javascript/places?hl=id#loading\\_the\\_library](https://developers.google.com/maps/documentation/javascript/places?hl=id#loading_the_library), tanggal akses: 8 Maret 2017

## Lampiran

### 1. Mobile Version

#### 1.1. Homepage & menu (guest)

*Homepage* merupakan halaman yang pertama kali muncul pada saat aplikasi dijalankan, *guest* merupakan sebutan untuk user yang menggunakan aplikasi namun belum *login*. Pada aplikasi ini *guest* hanya bisa menggunakan fitur pencarian dan melihat informasi tentang kafe.



Gambar 6. Homepage & menu (guest)

Keterangan :

1. Merupakan *button* untuk *show-hide* menu.
2. *Browse nearby places* untuk menampilkan kafe terdekat dari lokasi pengguna
3. Kafe yang di rekomendasikan oleh sistem, sebagai wadah sponsor untuk pemilik kafe.
4. Tampilan kafe terpopuler berdasarkan rating, apabila kafe tersebut di klik maka akan menampilkan informasi tentang kafe tersebut.

## 1.2. *Login page & Register page*

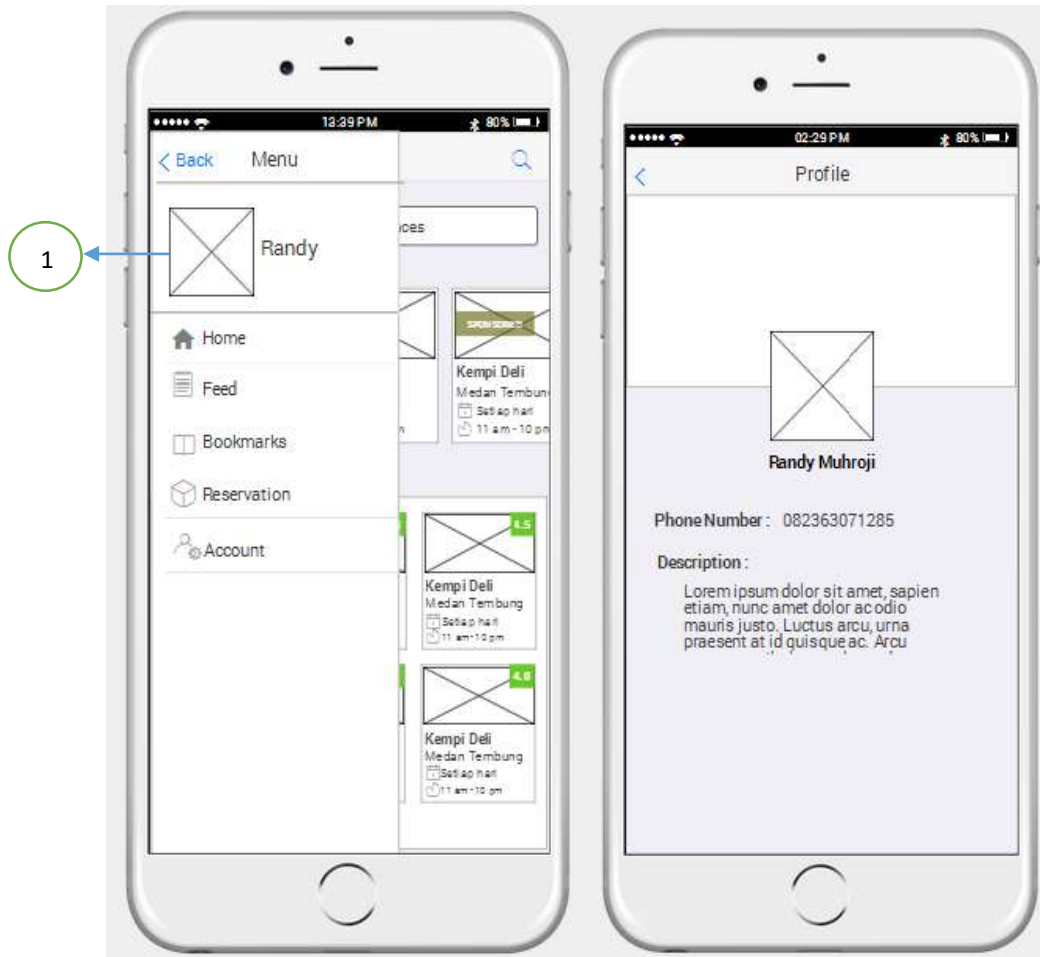


*Gambar 7. Register Page & Login Page*

Keterangan :

1. Apabila data yang dimasukkan telah benar, sistem akan otomatis membuat akun dengan *email* dan *password* tersebut untuk dapat *login* ke sistem.
2. Apabila *email* dan *password* telah benar, maka user akan diarahkan ke *homepage*.
3. User dapat login menggunakan akun facebook.

### 1.3. *Menu (user) & View profil*



*Gambar 8. Menu (user) & profil Page*

Keterangan :

1. Apabila di klik maka sistem akan mengarahkan aplikasi ke halaman *view profile*.

#### 1.4. Searchpage



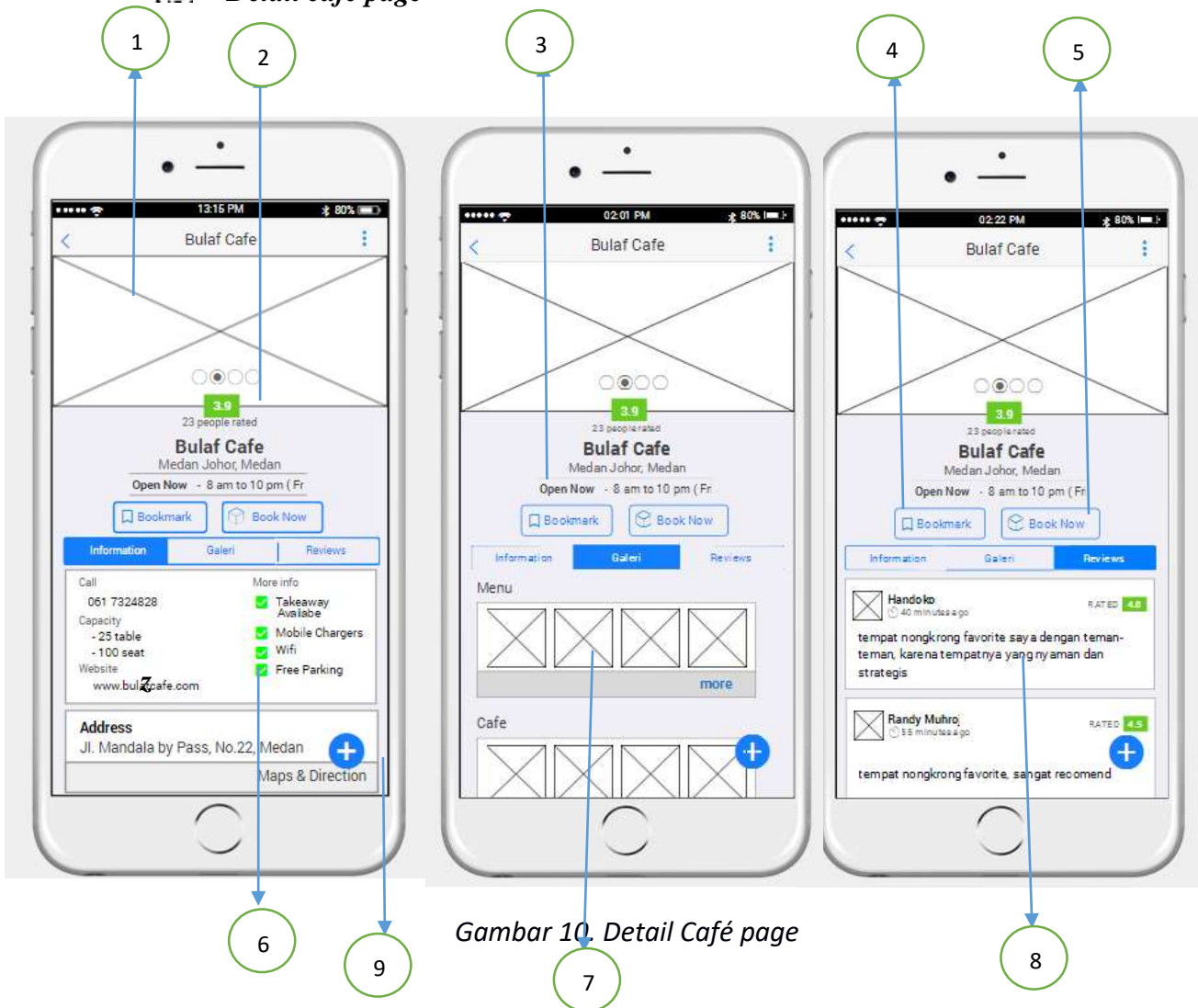
Gambar 9. Homepage & Search page

Keterangan :

1. Sistem akan mengarahkan aplikasi ke halaman pencarian.
2. *Form* untuk memasukkan *keyword* yang ingin dicari seperti nama kafe.
3. Hasil pencarian.
4. Untuk menampilkan menu filter
5. Daftar filter yang bisa dilakukan *user*.



### 1.5. *Detail café page*



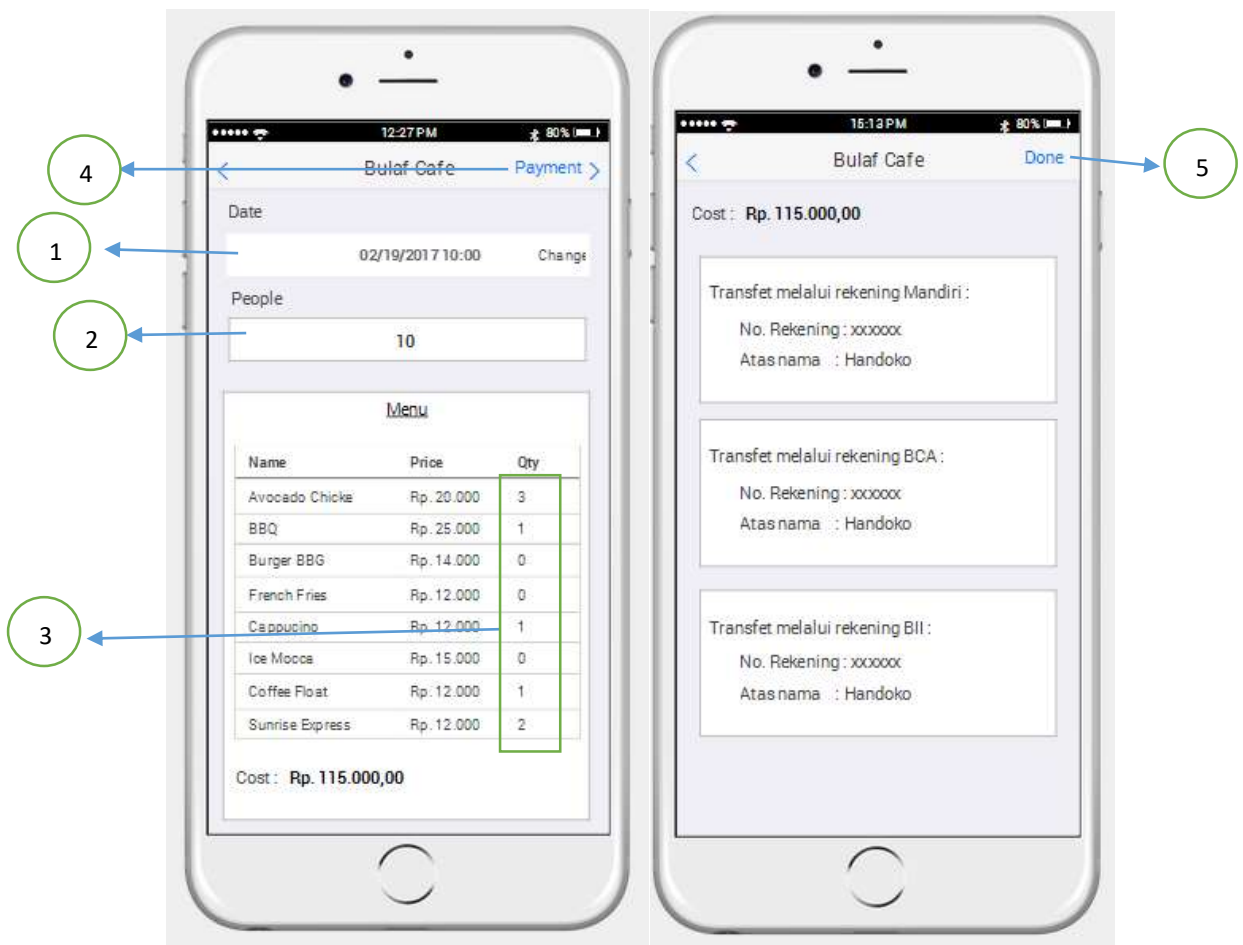
Gambar 10. *Detail Café page*

Keterangan :

1. *Slider* yang dimasukkan oleh pemilik kafe berupa tampilan kafe tersebut.
2. *Rating* yang dimiliki kafe tersebut.
3. Menampilkan status kafe pada waktu sekarang, apakah lagi buka atau tutup.
4. Bookmark kafe tersebut, yang berfungsi untuk menyimpan kafe tersebut, dan sistem secara otomatis menyimpan pada bookmarks page.
5. *Reservasi* kafe tersebut
6. Informasi tentang kafe seperti alamat, deskripsi, dan fasilitas yang dimiliki kafe tersebut

7. Galeri pada kafe tersebut, seperti gambar kafe, gambar menu, dan lain-lain.
8. Daftar *review* pada kafe tersebut.
9. Apabila user ingin memberikan *review* pada kafe tersebut,

#### 1.6. *Booking page & Payment page*



Gambar 11. *Booking page & Payment page*

Keterangan :

1. *User* dapat mengganti tanggal dan waktu untuk reservasi kafe.
2. *User* dapat mengubah banyak orang, yang nantinya akan dipersiapkan mejanya oleh pemilik kafe.
3. Daftat menu yang ingin dipesan oleh *user*.

4. Apabila data telah diisi, maka user akan diarahkan ke halaman payment
5. User akan diberikan waktu 1 jam untuk transer uang ke salah satu rekening tersebut, apabila telah ditransfer lalu user menekan *button* tersebut.

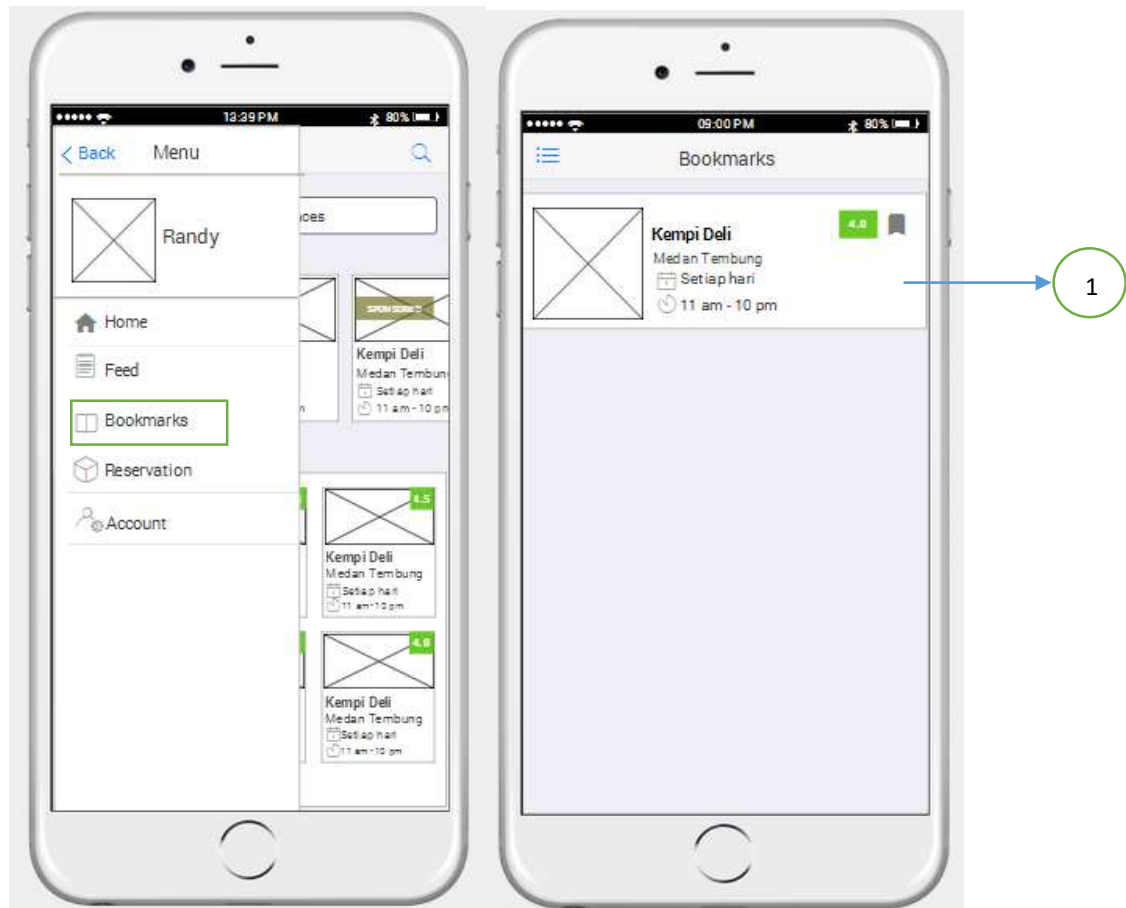
### 1.7. *Feed page*

*Feedpage* merupakan halaman yang menampilkan review dari semua user terhadap kafe, dan akan diurutkan berdasarkan yang terbaru.



Gambar 12. *Feedpage*

### 1.8. *Bookmarks page*

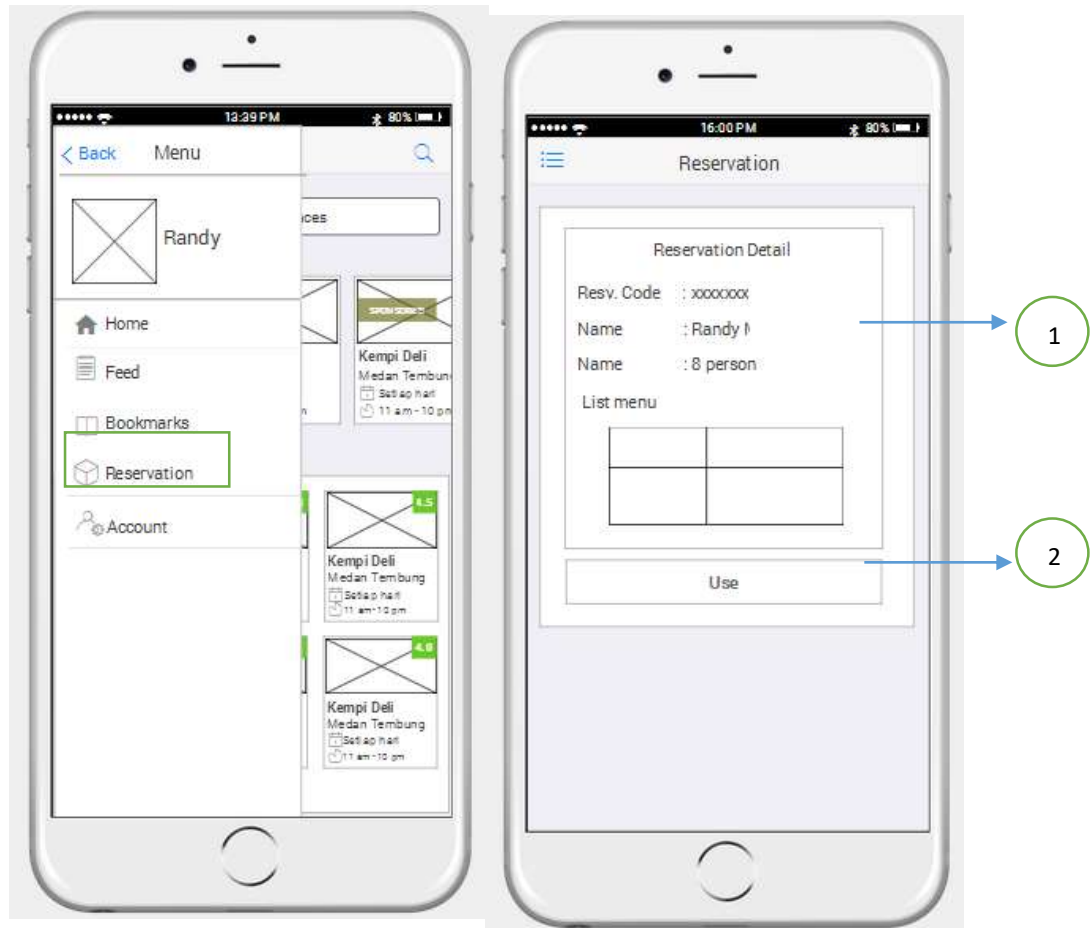


Gambar 13. *Bookmarks Page*

Keterangan :

1. Tampilan kafe yang telah di *bookmarks*.

### 1.9. *Reserv Page*

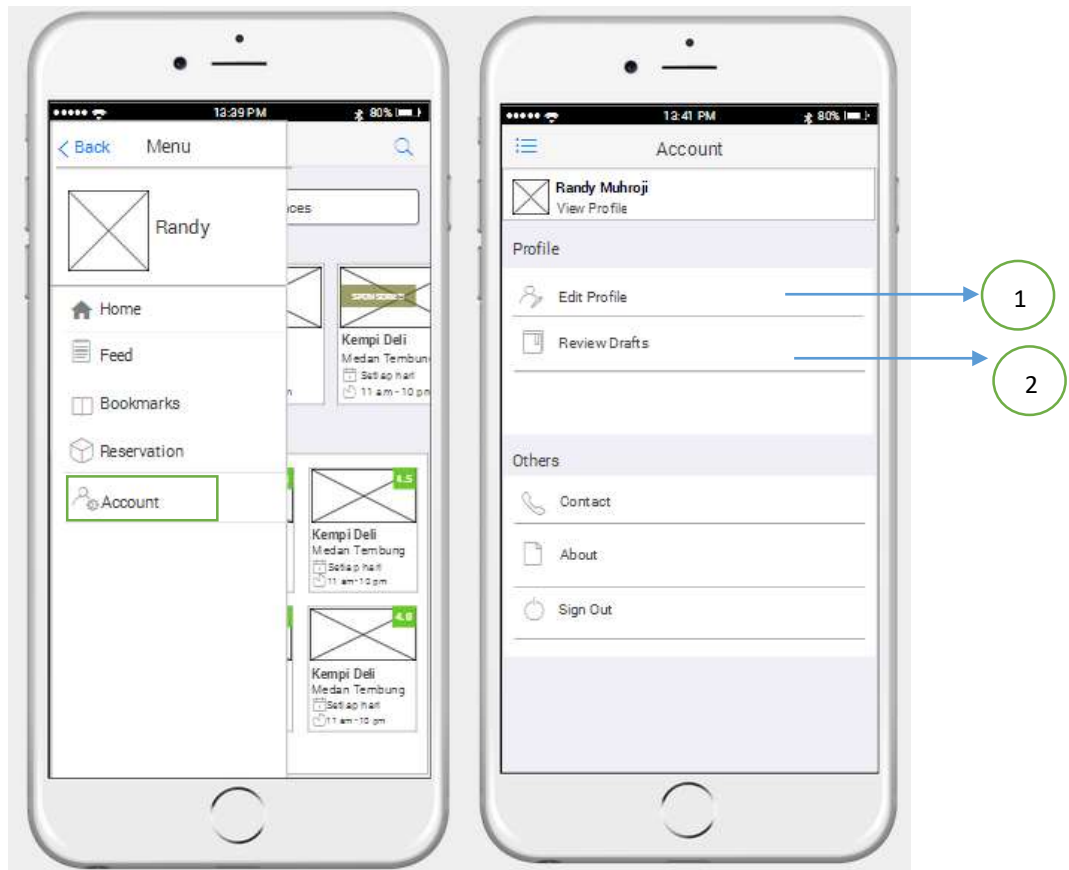


*Gambar 14. Reserv Page*

Keterangan :

1. Tampilan ini digunakan sebagai bukti reservasi kepada pemilik kafe, bahwa *user* telah melakukan reseervasi terhadap kafe tersebut.
2. Apabila *user* telah reservasi, maka pemilik kafe menekan *button* tersebut, sebagai bahwa bahwa struk reservasi telah digunakan.

### 1.10. Account page



Gambar 15. Account Page

Keterangan :

1. Untuk memperbaharui user profile seperti nama, nomot telephone, dan sebagainya.
2. Untuk menampilkan semua aktivitas review user terhadap kafe.

### 1.11. *Edit profile page & Review drafts page*



*Gambar 16. Review Drafts Page & Edit profile page*

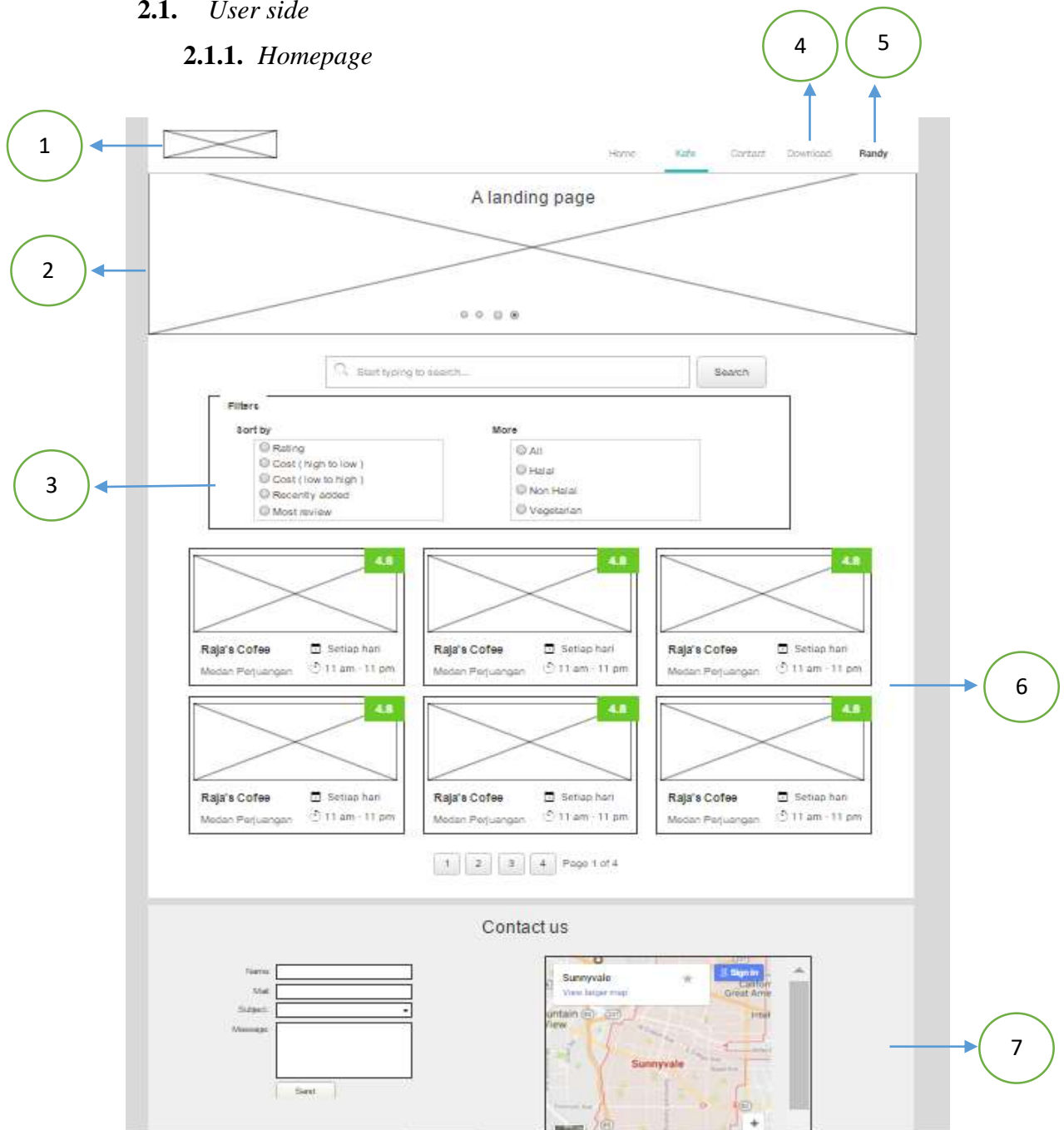
Keterangan :

1. Isi review yang dilakukan user.
2. Apabila user ingin memperbaharui isi review.
3. Apabila user ingin menghapus review.
4. Form untuk mengedit profile.

## 2. Web Version

### 2.1. User side

#### 2.1.1. Homepage



Gambar 17. Homepage

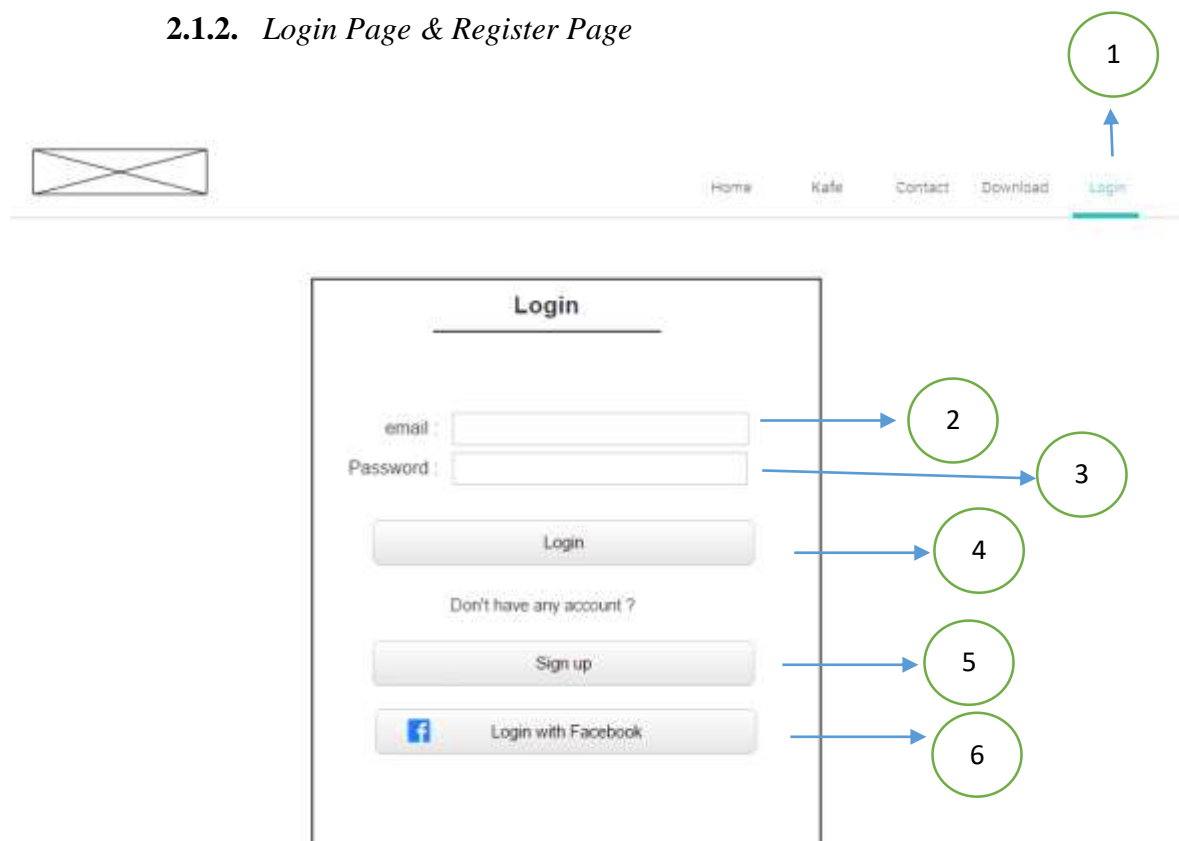
Keterangan :

1. Logo website
2. Slider image yang menampilkan informasi tentang website



3. Filter pencarian berguna untuk memudahkan *user* dalam melakukan pencarian
4. Menu “*Download*” berfungsi untuk mendownload aplikasi *mobile* dari website tersebut
5. Merupakan nama dari *user* yang telah *login*, apabila di klik menu tersebut, sistem akan mengarahkan user ke *profile page*.
6. Daftar kafe, apabila kafe tersebut di klik, sistem akan mengarahkan user ke detail *café*
7. Apabila user memiliki kritik dan saran dapat menghubungi admin melalui fitur “*Contact us*”

### 2.1.2. Login Page & Register Page



Gambar 18. Login page

Header: Home Kafe Contact Download Login

Register

email :

email :

Password :

Retry Password :

Sign up

Login with Facebook

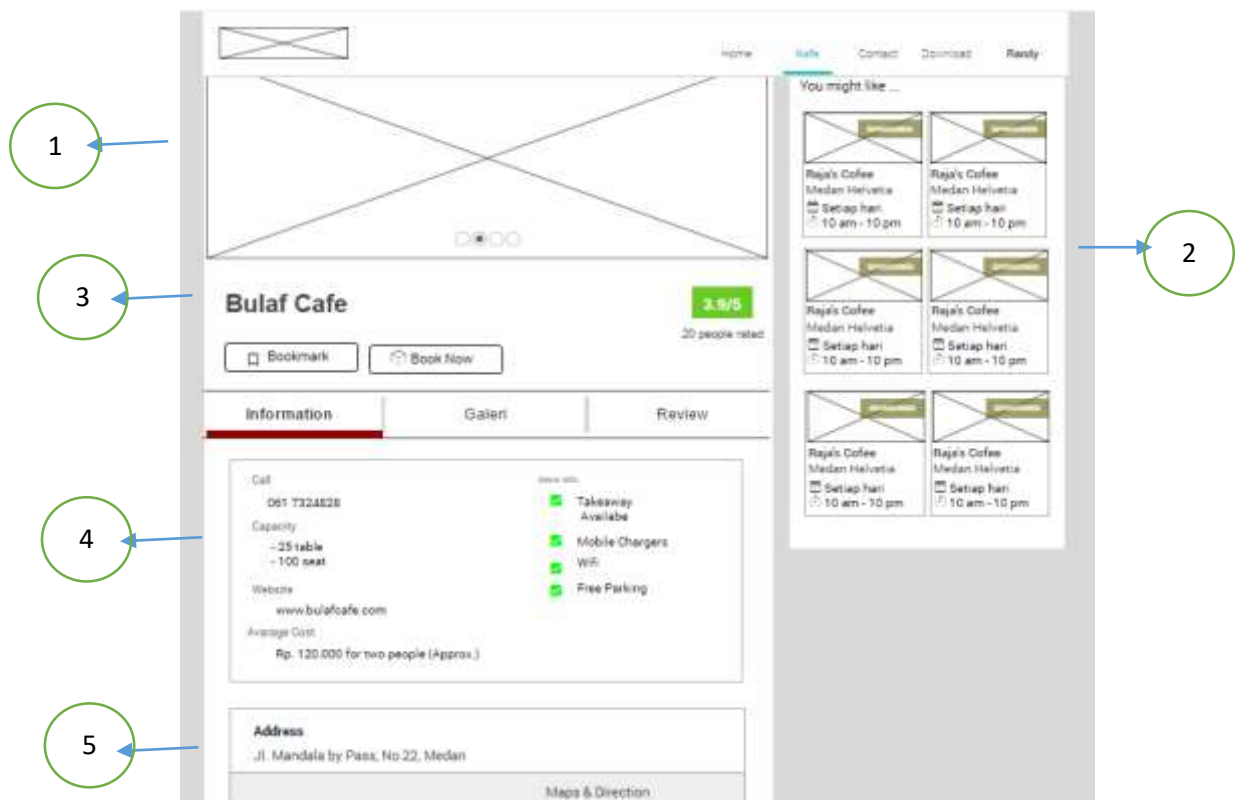
7

Gambar 19. Register Page

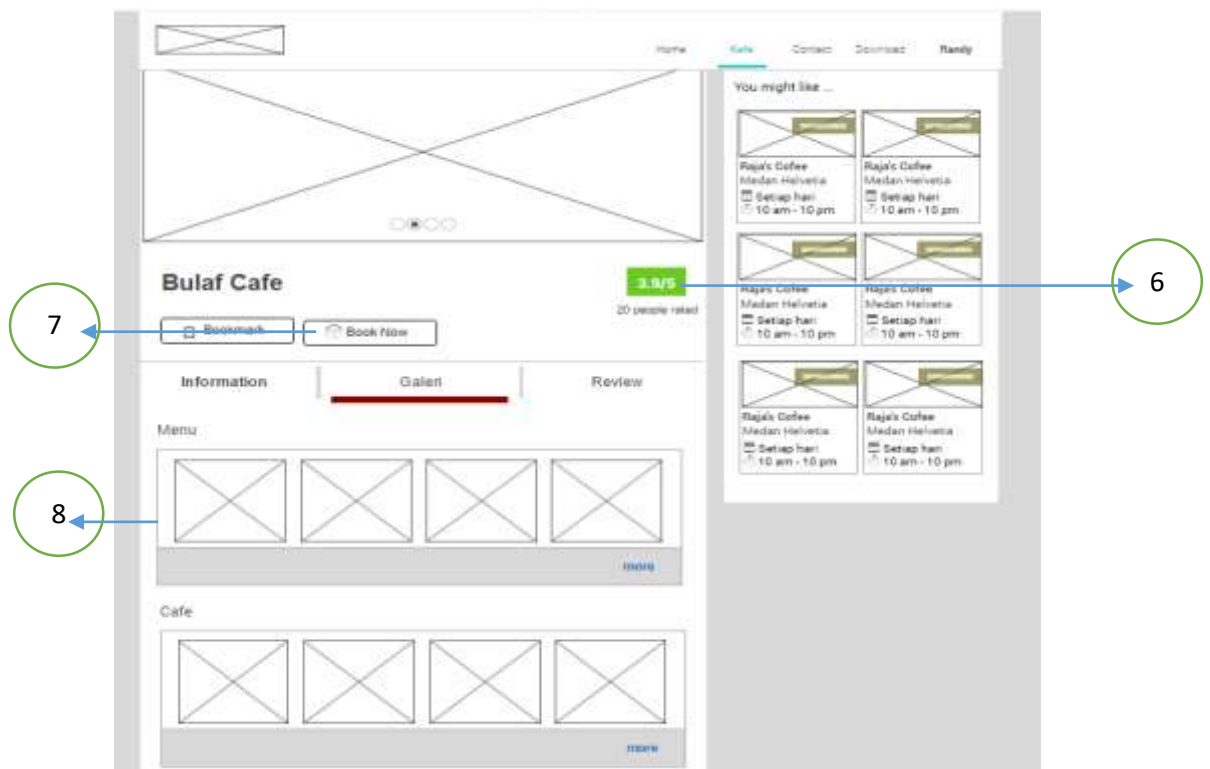
Keterangan :

1. *User* telah berada pada menu *login*.
2. *Form* untuk memasukan *email*.
3. *Form* untuk memasukkan *password*.
4. Apabila *email* dan *password* yang dimasukkan *user* valid, maka sistem akan mengarahkan *user* ke *homepage* seperti pada *gambar 17*.
5. Apabila *user* tidak memiliki akun maka terlebih dahulu harus *register*.
6. *User* dapat *login* menggunakan akun facebook.
7. *Form* yang harus diisi *user* apabila ingin mendapatkan akun.

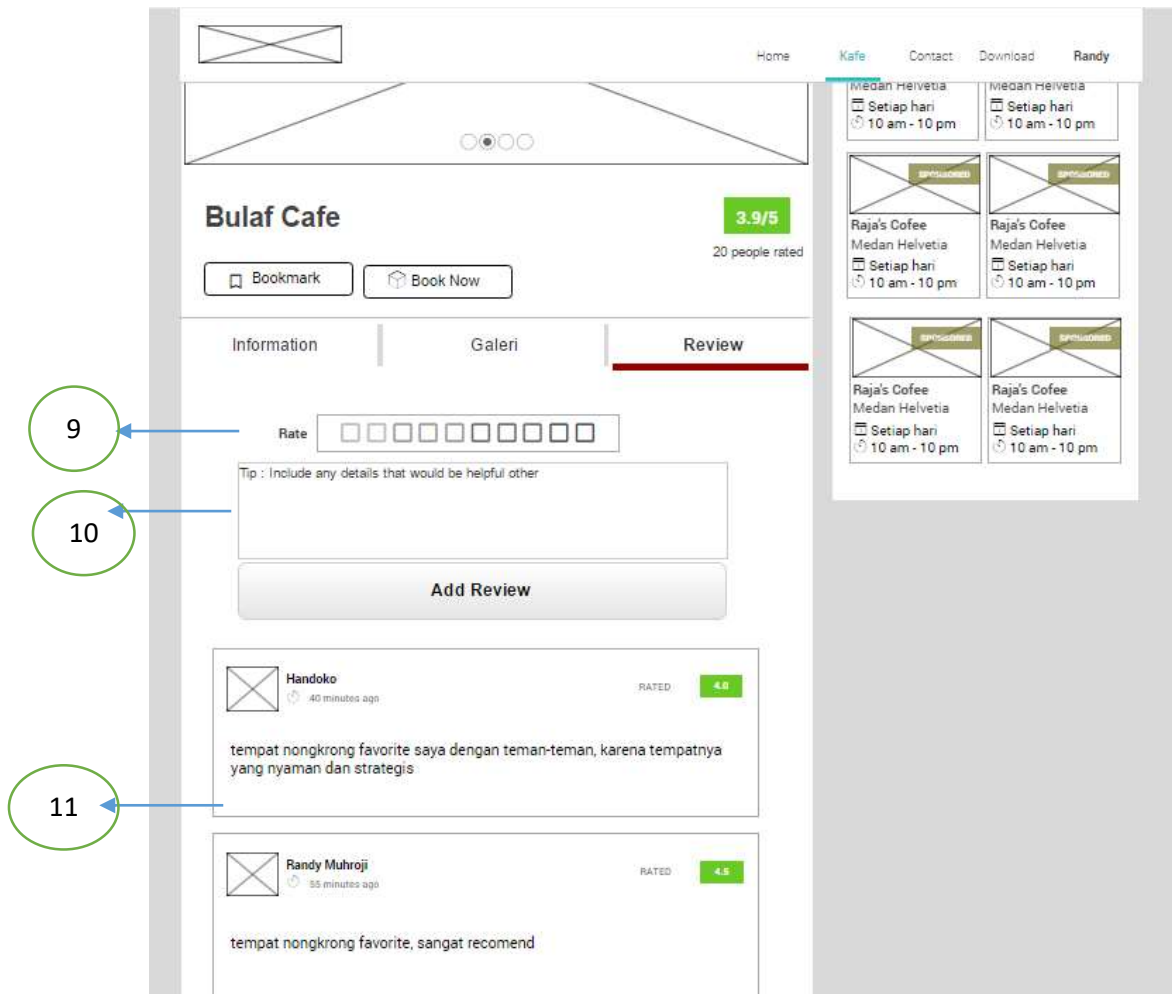
### 2.1.3. Detail Café



Gambar 20. Information Café



Gambar 21. Galeri Café



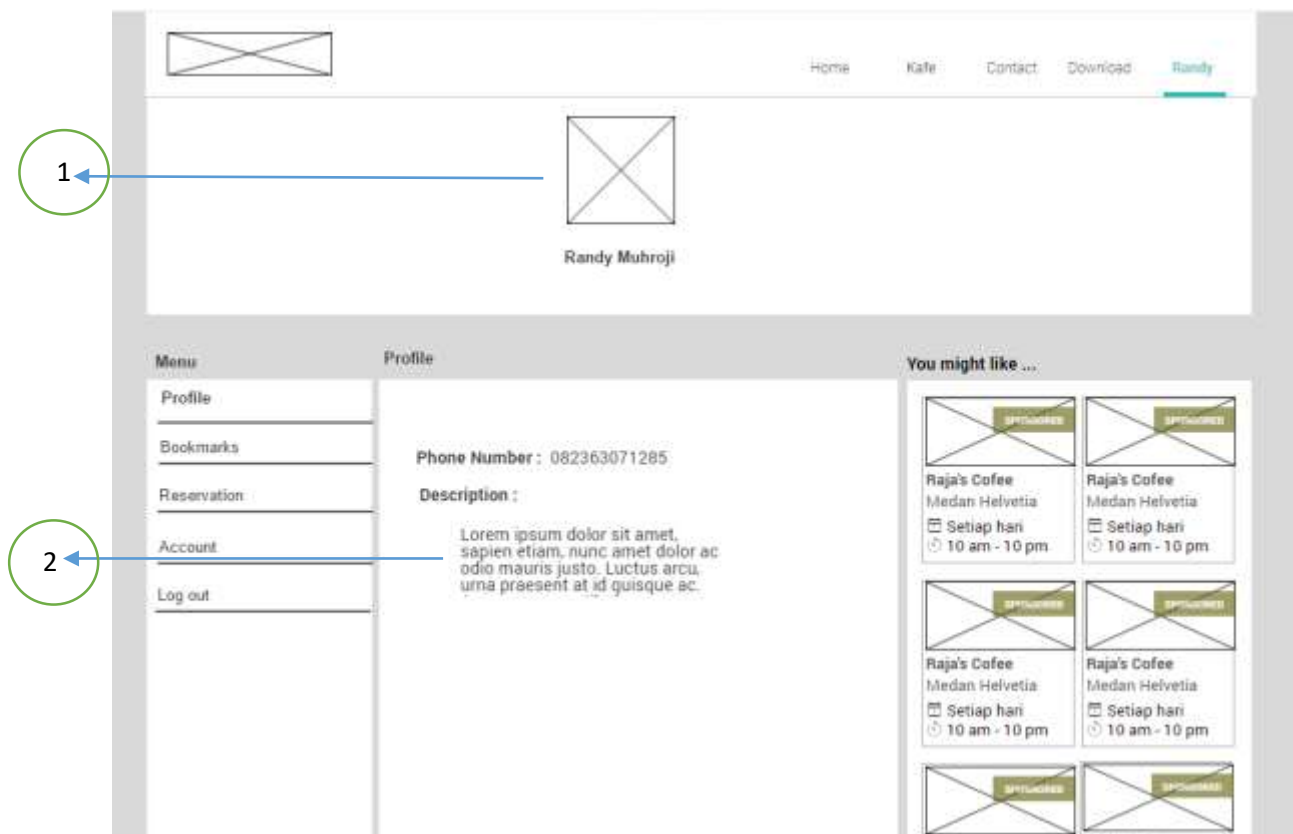
Gambar 22. Review Cafe

Keterangan :

1. *Slider image* yang menerangkan secara sederhana tentang café tersebut.
2. “Sponsor” merupakan daftar café yang direkomendasikan oleh sistem kepada *user*.
3. Apabila *user* tertarik dengan café tersebut, user dapat melakukan *bookmark*.
4. Informasi tentang café tersebut.
5. Lokasi café.
6. *Rating* yang dimiliki café tersebut.
7. Apabila *user* ingin melakukan pemesanan terhadap café tersebut, dapat menggunakan button “*Book now*”

8. Daftar gambar menu pada café tersebut.
9. Penilai *user* terhadap café tersebut.
10. Komentar *user* terhadap café tersebut.
11. Daftar *review* semua *user* terhadap café tersebut.

#### 2.1.4. User Profile

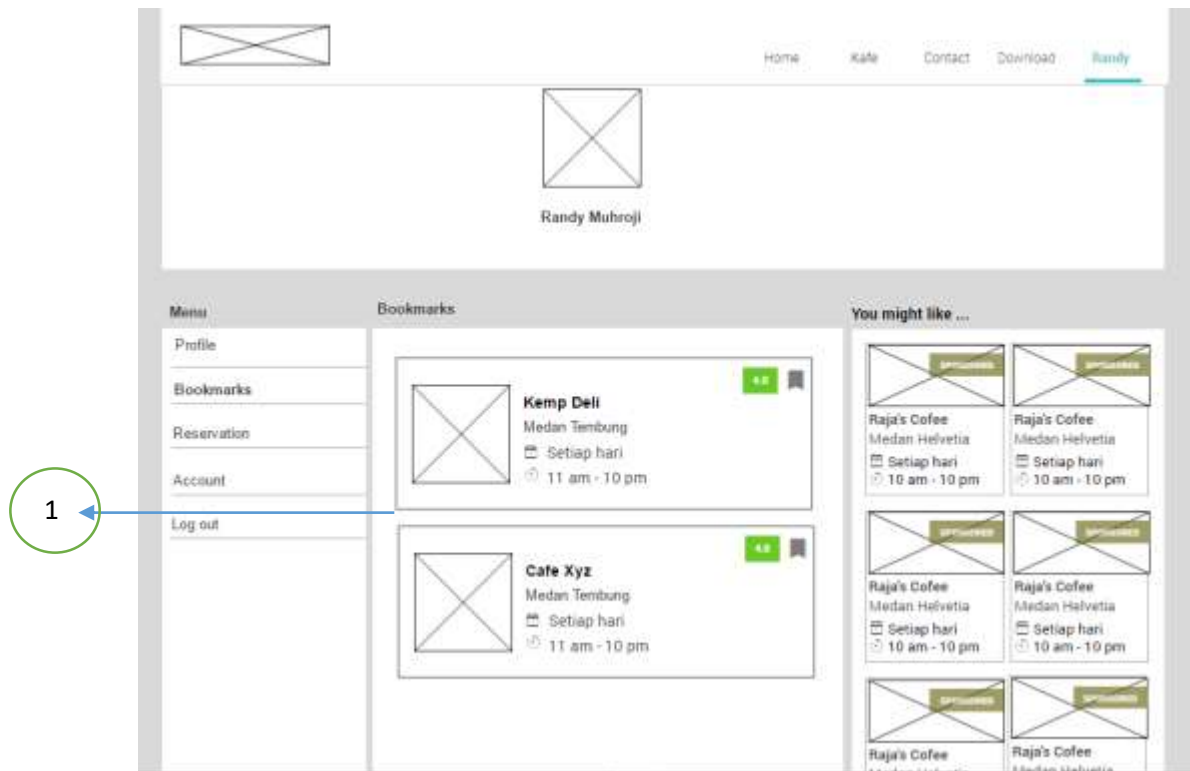


Gambar 23. User Profile

Keterangan :

1. Foto Profil *user*
2. Deskripsi singkat tentang *user*.

### 2.1.5. Bookmarks

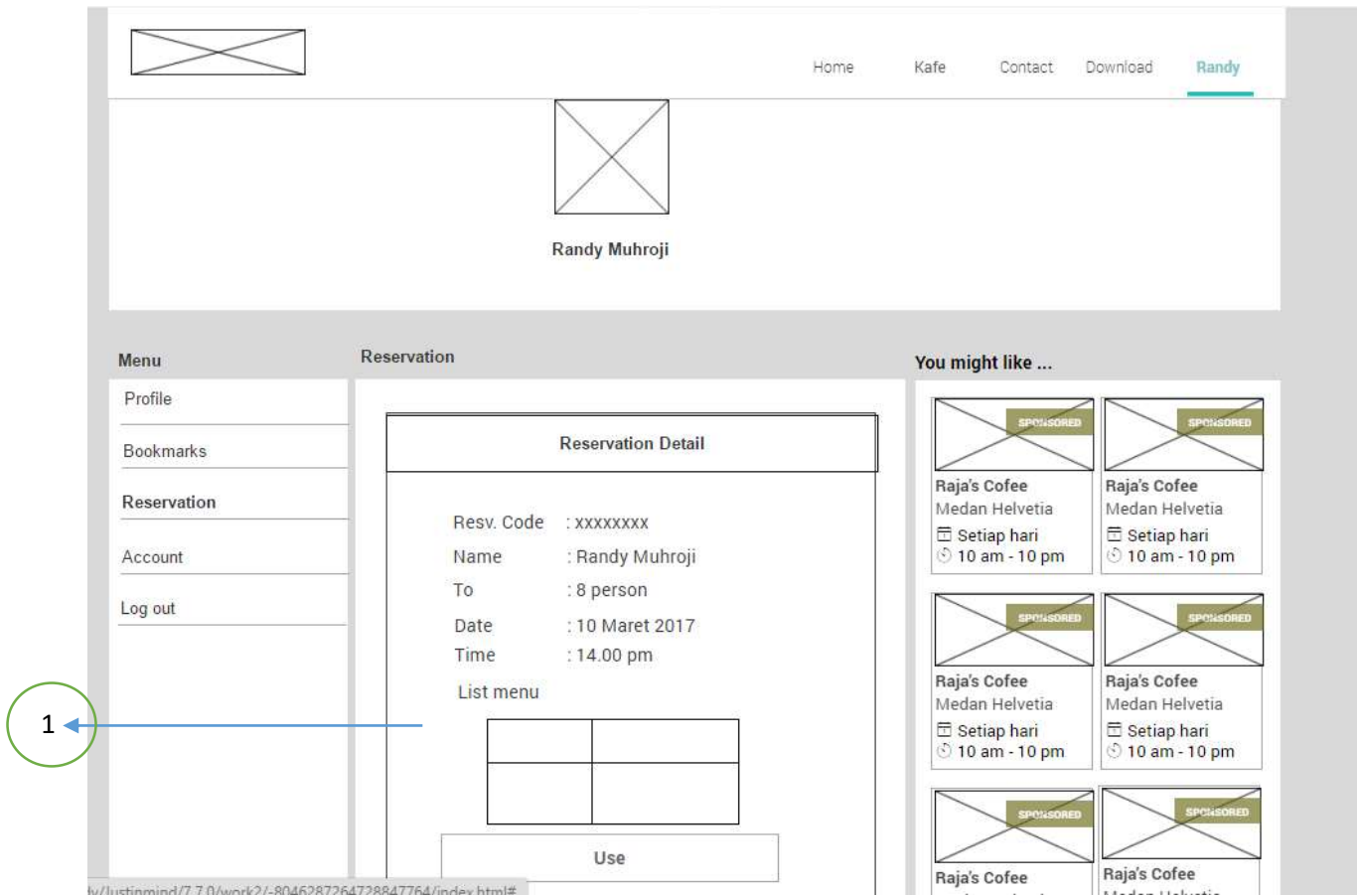


Gambar 24. Bookmarks

Keterangan :

1. Daftar yang pernah di *bookmarks* oleh *user*

### 2.1.6. Reservation

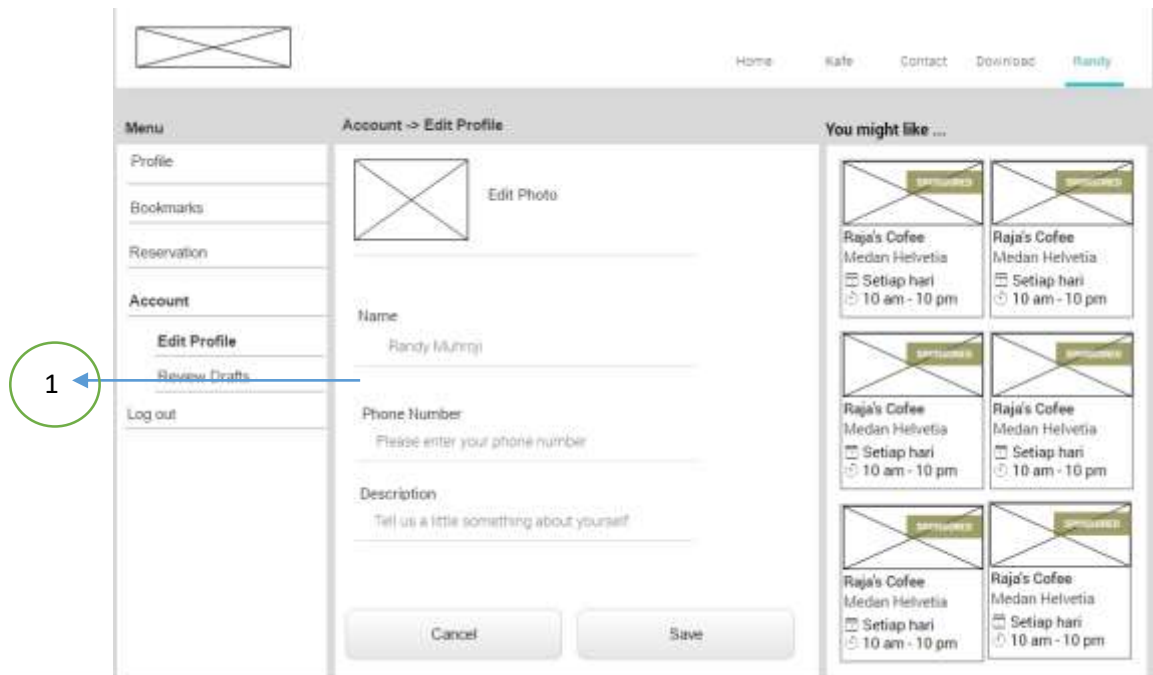


Gambar 25. Reservasi

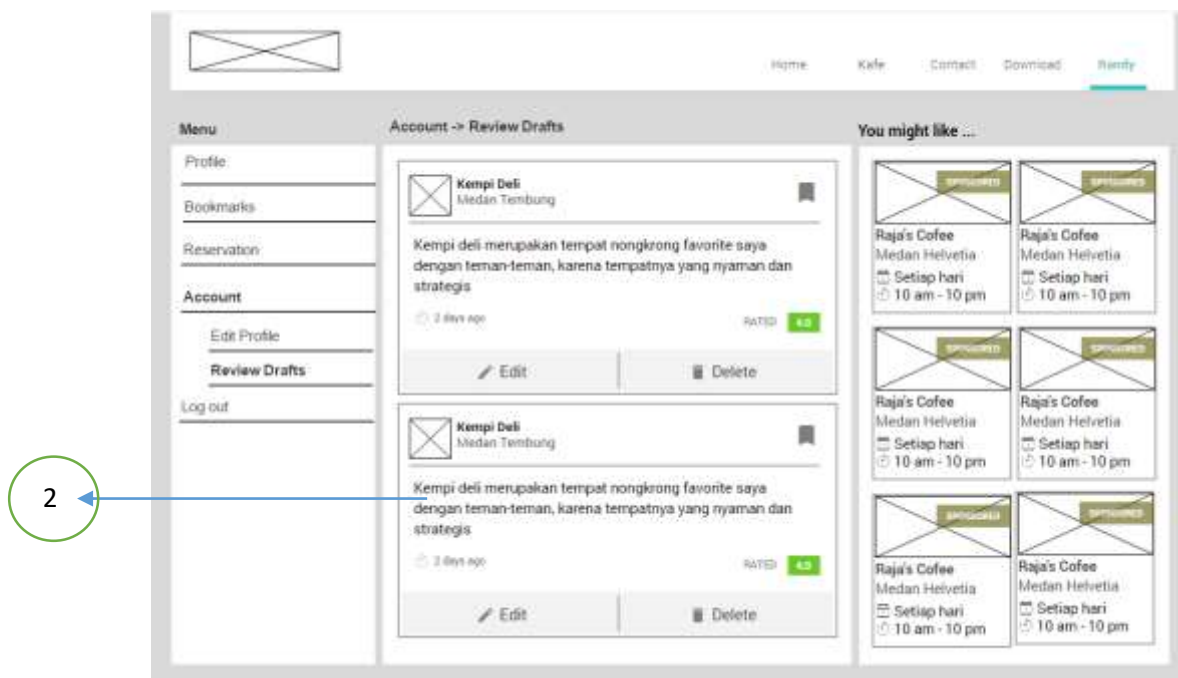
#### Keterangan :

1. Struk bukti bahwa user telah melakukan reservasi terhadap café.

### 2.1.7. Account



Gambar 26. Edit Profile



Gambar 27. Review Drafts



Keterangan :

1. *Form* untuk *edit profile*.
2. Daftar *review* yang telah dilakukan *user* terhadap *café*.

## 2.2. Owner Café side

### 2.2.1. Profile Café

1

2

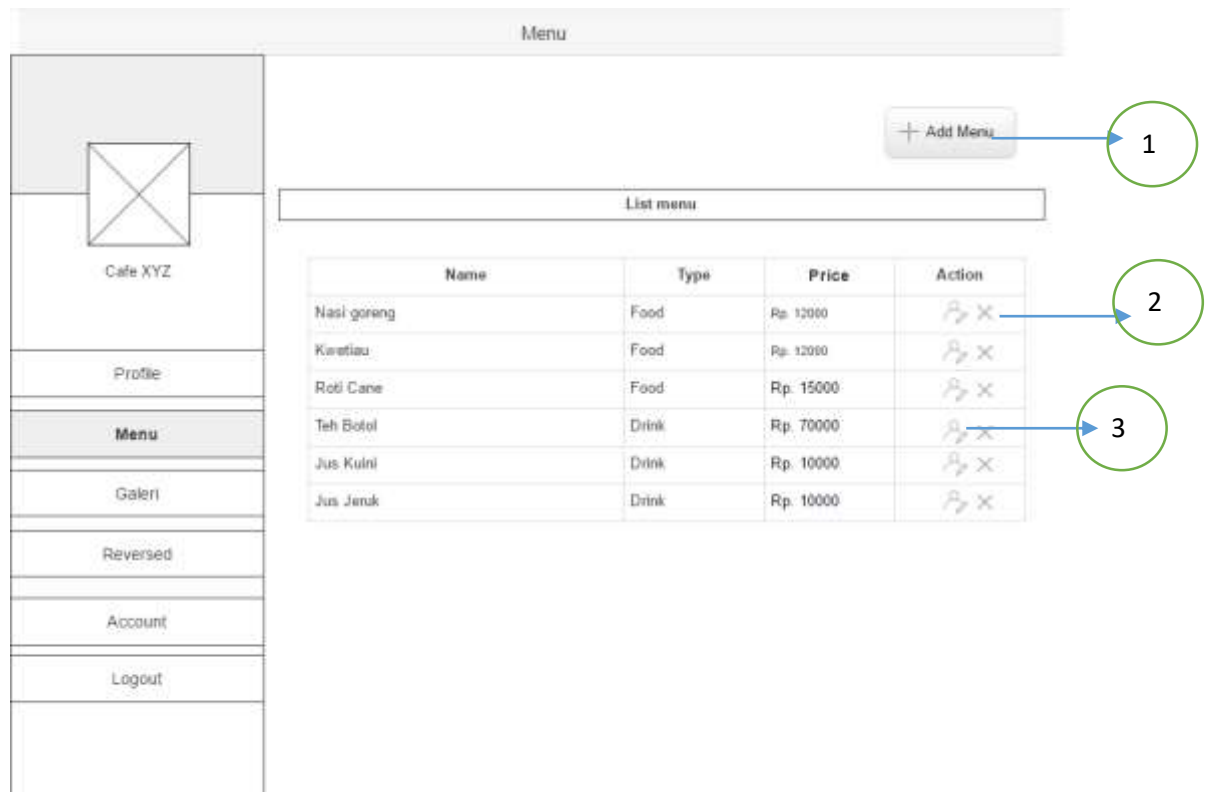
3

Gambar 28. Edit Profile Café

Keterangan :

1. Foto profil tentang *café*.
2. Untuk melihat tampil *profile café* yang akan dilihat *user*.
3. *Form* untuk memperbarui informasi pada *café*.

### 2.2.2. Daftar Menu Café

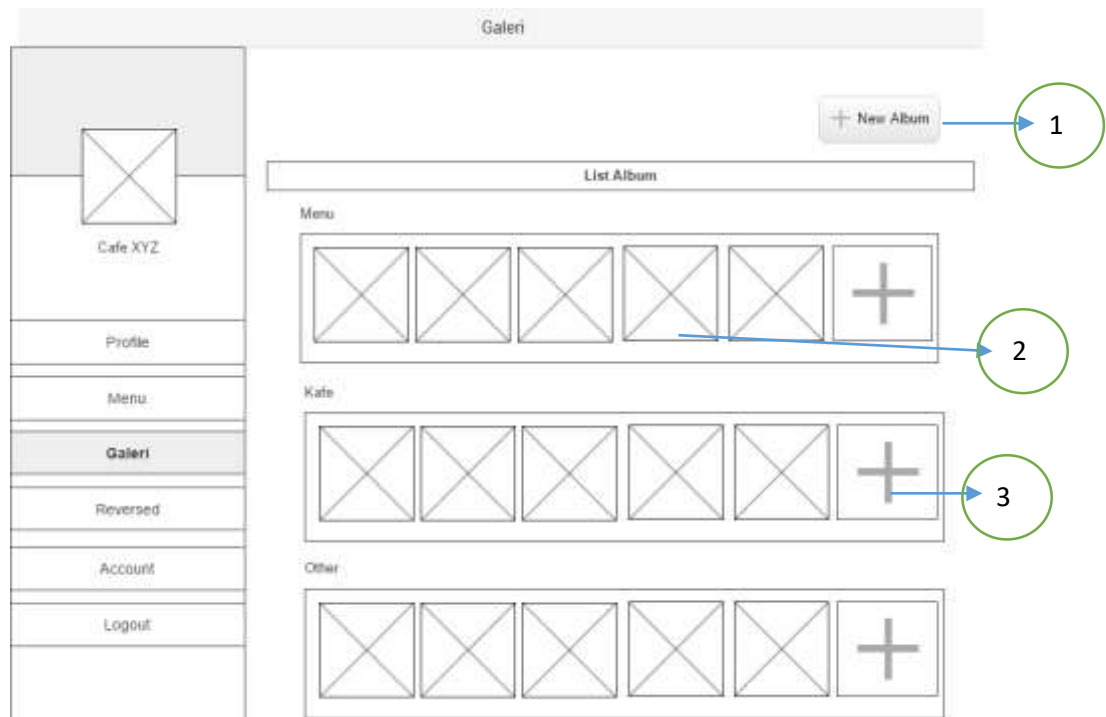


Gambar 29. Daftar menu mafé

Keterangan :

1. Fitur untuk menambah menu
2. Hapus menu
3. *Edit* menu

### 2.2.3. Galeri




Gambar 30. Galeri

Keterangan :

1. *Button* untuk menambahkan album baru.
2. Daftar photo pada album tersebut.
3. Untuk menambah foto pada album tersebut.

#### 2.2.4. Reserved

Reserved



Cafe XYZ

Profile

Menu

Galeri

**Reserved**

Account

Logout

Reservation Detail

Resv. Code : xxxxxxxx

Name : Randy Muhroji

To : 8 person

Date : 10 Maret 2017

Time : 14.00 pm

List menu



Status

Expired

Gambar 31. Reserve

#### 2.2.5. Account

Account



Cafe XYZ

Profile

Menu

Galeri

Reserved

**Account**

Logout

ID : Cafexyz001

Change Password

Old Password

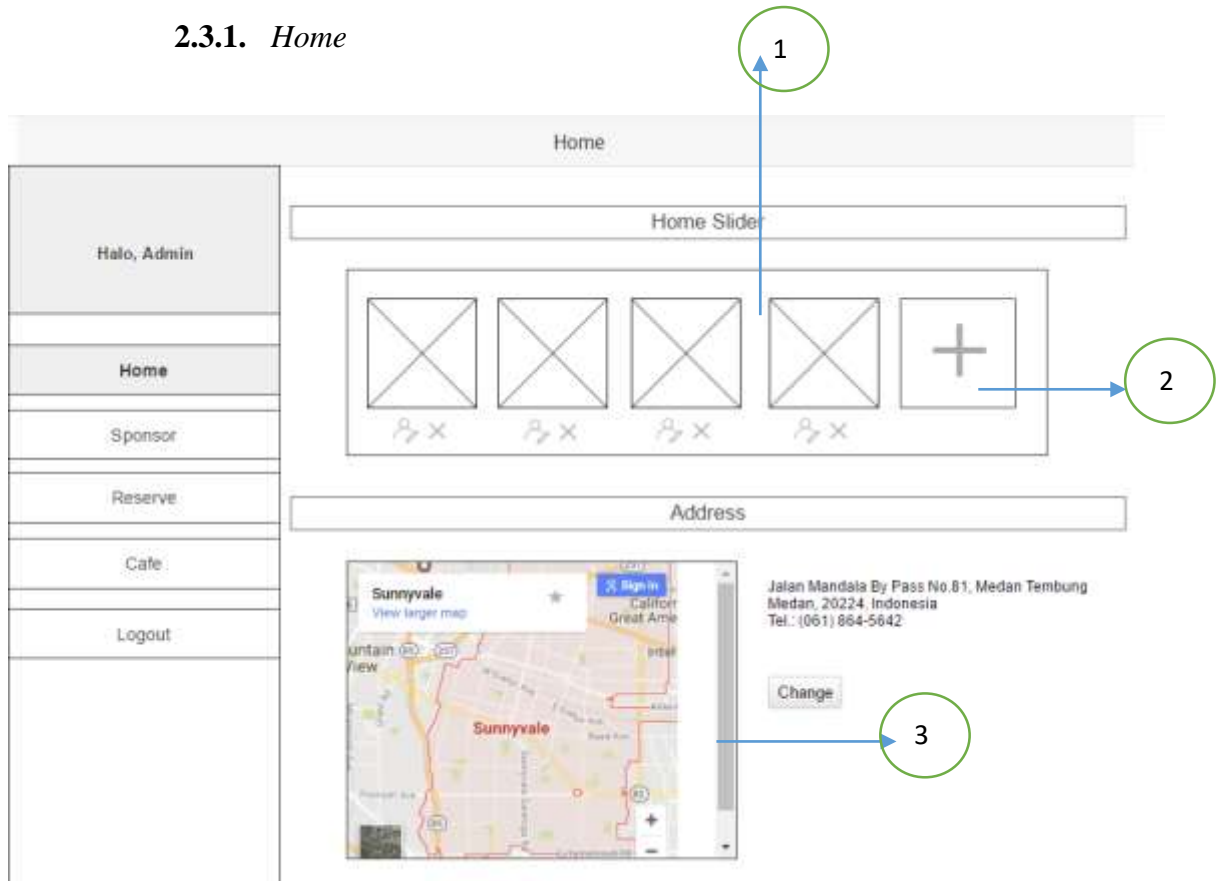
New Password

Re- Password

Gambar 32. Edit Password

## 2.3. Admin side

### 2.3.1. Home



Gambar 33. Home (Admin)

Keterangan :

1. Daftar gambar yang akan ditampilkan pada *Homepage*.
2. Untuk menambah gambar.
3. Alamat yang akan ditampilkan pada *Homepage* bagian *Contact us*

us

### 2.3.2. Sponsor

Halo, Admin

Home

**Sponsor**

Reserve

Cafe

Logout

Sponsor

+ Add Sponsor

List Sponsor

**Id cafe :** cafe001

**Name :** cafe XYZ

**Expired :** 10 Maret 2017

Show cafe

**Id cafe :** cafe001

**Name :** cafe XYZ

**Expired :** 10 Maret 2017

Show cafe

**Id cafe :** cafe001

**Name :** cafe XYZ

**Expired :** 10 Maret 2017

Show cafe

Gambar 34. Memasukkan sponsor

### 2.3.3. Reserve

Halo, Admin

Home

Sponsor

**Reserve**

Cafe

Logout

Reserve

Reservation Detail

Resv. Code : xxxxxxxx

**Name :** Randy Muhroji

**To :** 8 person

**Date :** 10 Maret 2017

**Time :** 14.00 pm

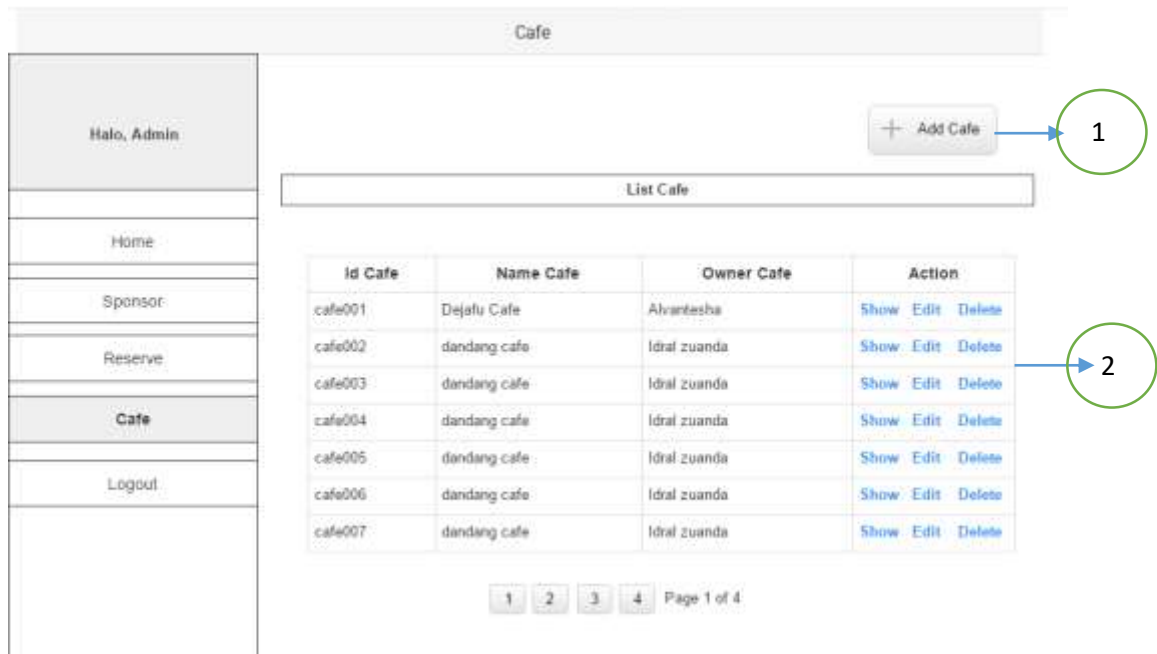
List menu


**Cost :** Rp. 120.000

Confirm

Gambar 35. Daftar pemesan yang dilakukan oleh user

### 2.3.4. Café



Gambar 36. Cafe

Keterangan :

1. Untuk menambahkan kafe baru.
2. Daftar kafe yang tersedia pada website tersebut.