

BAB II

TINJAUAN LITERATUR

2.1 Penelitian Terdahulu

Untuk mendapatkan informasi tentang pengenalan atau pemasaran, peneliti telah melakukan penyelidikan dan meninjau berbagai sumber penelitian yang membahas topik yang sama. Beberapa penelitian sebelumnya yang sejalan dengan penelitian ini dapat digunakan sebagai referensi untuk pemahaman. Studi sebelumnya terkait dengan pengenalan atau pemasaran *Guest House*. Beberapa studi sebelumnya yang menjadi acuan peneliti adalah sebagai berikut: Studi pertama, "Analisis Strategi Pemasaran *Guest House* Tengkawang *Residence* Samarinda", ditulis oleh Daniel Purnama Candra. (Dwi Putra, 2024). Metode kualitatif dan analisis data deskriptif digunakan dalam penelitian ini. Penelitian ini berbeda dengan penelitian *Guest House* Gintribu dalam hal penerimaan tamu. Penelitian Daniel meneliti *Guest House* Tengkawang Candra, yang memungkinkan tamu lokal dan internasional, tetapi penelitian *Guest House* Gintribu mengutamakan tamu dari berbagai negara yang diminta untuk mendaftar.

Kedua, Jurnal yang dituliskan oleh (Ekalista, 2019) dengan judul “Strategi Komunikasi Pemasaran Hotel Kartika Graha Malang dalam Meningkatkan Jumlah Pengunjung”. Kesamaan pada penelitian sebelumnya , Penelitian ini menggunakan metode kualitatif deskriptif dan strategi untuk meningkatkan pengunjung. Namun, perbedaan terletak pada lokasi dan objek penelitian.

Ketiga, jurnal berjudul "Strategi Pemasaran dalam Meningkatkan Daya

Tarik Tamu di *Grand Bunda Hotel Syariah Bukit tinggi*" (Oktavia, 2021) memiliki kesamaan dengan penelitian sebelumnya karena penelitian ini menggunakan metode kualitatif deskriptif dan menggunakan strategi pemasaran dengan buaran produk, harga, lokasi, dan promosi. Namun, yang membedakan adalah lokasi penelitian yang dilakukan peneliti sebelumnya adalah hotel, sedangkan lokasi penelitian yang dilakukan peneliti saat ini adalah *Guest House*.

Penginapan *Guest House* biasanya lebih kecil dari hotel dan dijalankan sendiri oleh pemiliknya yang tinggal di dekat usahanya. Banyak dipilih oleh wisatawan mancanegara karena konsepnya sederhana dan mirip dengan hunian rumah sendiri bagi pelanggan. Pengunjung atau wisatawan adalah kelompok atau individu yang mengunjungi suatu tempat di luar tempat tinggalnya dengan tujuan rekreasi, bisnis, atau alasan lainnya. Wisatawan terdiri dari wisatawan lokal dan internasional. (Dwi Putra, 2024)

2.1.1 Website

Menurut (Rochman, 2020). *Website* merupakan sarana penyampaian informasi secara *Online* yang memiliki fasilitas dasar internet yang mampu menembus batasan ruang dan waktu. Sementara ditengah pesatnya perkembangan dunia yang penuh persaingan, informasi merupakan bagian yang sangat penting.

Menurut (Nopita, 2021). Bahwa *Website* adalah sekumpulan halaman web dan dapat digunakan untuk menghubungkan beberapa file yang saling terhubung. Dalam *Website* ini ada sebuah halaman utama atau beranda

yang pertama kali dilihat ketika pengguna mengakses web tersebut. Dari halaman utama, pengunjung dapat mengakses fitur-fitur pada menu utama untuk berpindah pada menu lainnya yang ada pada *Website* tersebut.

Menurut (Rochman, 2020). *Website* adalah halaman informasi yang disediakan melalui jalur internet, sehingga bisa diakses di manapun selama terkoneksi dengan jaringan internet. *Website* merupakan kumpulan komponen yang terdiri dari teks, Gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi.

2.1.2 *Guest House sebagai Alternatif Akomodasi*

Menurut (Angga Hernanda, 2022). Penginapan adalah salah satu bisnis yang memiliki banyak peluang dan prospek yang menjanjikan. Memiliki manajemen hotel yang tepat dan berpengalaman akan meningkatkan kualitas pariwisata dan menjaga kelangsungan usaha hotel di Kota Bumi Lampung Selatan. Namun, persaingan antar hotel yang ketat saat ini membutuhkan kegiatan promosi dan pengelolaan penyediaan kamar hotel yang baik.

Menurut (Dwi Putra, 2024). Penginapan *Guest House* biasanya lebih kecil dari hotel dan dijalankan sendiri oleh pemiliknya yang tinggal di dekat usahanya. Banyak dipilih oleh wisatawan mancanegara karena konsepnya sederhana dan mirip dengan hunian rumah sendiri bagi pelanggan. Pengunjung atau wisatawan adalah kelompok atau individu yang mengunjungi suatu tempat di luar tempat tinggalnya dengan tujuan rekreasi, bisnis, atau alasan lainnya. Wisatawan terdiri dari wisatawan

lokal dan internasional. Perusahaan berusaha untuk menginformasikan, mengingatkan, dan menarik perhatian pelanggan mengenai industri dan barang yang mereka jual melalui proses komunikasi pemasaran.

2.1.3 Framework

Menurut (Sallaby, 2020). *Framework* adalah kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu.

Menurut (Kansha, 2023). *Framework* adalah sebuah software yang mempunyai susunan kerangka kerja yang bisa digunakan oleh developer untuk mengembangkan sebuah aplikasi berbasis web, mobile, maupun desktop. *Framework* berisi perintah dan sebuah fungsi dasar yang umum dan bisa digunakan untuk mengembangkan sebuah software aplikasi dan diharapkan aplikasi bisa dikembangkan dengan lebih cepat serta tersusun dan terstruktur dengan sangat rapi. *Framework* mempunyai konsep yang menyediakan sebuah susunan fungsional generik atau suatu susunan fungsi yang umum dan terdapat pada suatu aplikasi. Programmer bisa menggunakan susunan fungsi tersebut pada aplikasi yang sedang dikembangkan. Programmer juga bisa mengubah susunan fungsi secara selektif menggunakan kode tertentu.

2.1.4 Codeigniter

Menurut (Sahi, 2020). *Codeigniter* adalah sebuah *Framework PHP*

yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis *PHP* dibanding jika menulis semua kode program dari awal.

Menurut (Sallaby, 2020). *Codeigniter* adalah Sebuah *Framework PHP* yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal.

2.1.5 Composer

Menurut (Aipina, 2022). *Composer* adalah tool yang di dalamnya terdapat dependencies dan libraries yang dibutuhkan pada saat membuat project laravel dan saat menginstall *library* yang kita butuhkan selanjutnya. Kumpulan dependencies dan *library* yang kita gunakan disimpan dalam sebuah file dengan nama *Composer.json* yang nantinya berada dalam folder utama aplikasi yang kita buat.

Menurut (Kansha, 2023). *Composer* adalah adalah sebuah tools dependency manager khusus *PHP* yang mempunyai fungsionalitas seperti Gem (Ruby) atau Maven (Java). Suatu *library* dapat diinstall di *Composer* dan *Composer* akan secara otomatis bisa menginstall ataupun meng-*update library* yang sangat dibutuhkan.

2.1.6 Guest House

Menurut (Maita, 2017). *Guest House* adalah jenis akomodasi penginapan yang menyediakan layanan tempat tinggal sementara dengan

suasana yang lebih pribadi dan nyaman dibandingkan hotel. *Guest House* biasanya dikelola secara mandiri oleh pemilik atau keluarga dan umumnya memiliki fasilitas yang lebih sederhana namun tetap memadai untuk kebutuhan tamu. *Guest House* adalah sebuah tempat yang dibangun untuk penginapan, perbedaan yang mendasar dengan hotel adalah disini biasanya para pengelola menawarkan untuk sewa kamar harian dan bulanan. Sedangkan Hotel dapat didefinisikan sebagai penggabungan dari semua definisi usaha pariwisata, akomodasi, boga, hiburan serta bisnis. Semua itu bisa terjadi dalam suatu hotel. Hotel memiliki ciri khas yang tidak dimiliki oleh usaha pariwisata yang mirip hotel, yaitu adanya layanan oleh pramusaji, tatagraha, *dry cleaning*, serta berbagai fasilitas lain dalam satu properti.

1. Ciri-ciri *Guest House*:

a. Ukuran Lebih Kecil:

Kapasitas *Guest House* biasanya lebih terbatas dibandingkan hotel, dengan jumlah kamar yang lebih sedikit.

b. Suasana Keluarga:

Memberikan nuansa yang lebih akrab dan hangat, mirip seperti tinggal di rumah sendiri.

c. Layanan yang Personal:

Pelayanan di *Guest House* cenderung lebih *Personal* karena pemilik atau pengelola sering berinteraksi langsung dengan tamu.

d. Fasilitas Standar:

Memberikan fasilitas dasar seperti kamar tidur, kamar

mandi, dapur bersama, ruang tamu, dan tempat parkir. Beberapa *Guest House* mungkin memiliki tambahan fasilitas seperti Wi-Fi dan sarapan gratis.

e. Lokasi:

Biasanya terletak di lingkungan yang lebih tenang, dekat tempat wisata atau pusat kota.

2. Perbedaan *Guest House* dengan Hotel:

Tabel 2.1 Perbedaan *Guest House* dan Hotel

Aspek	<i>Guest House</i>	Hotel
Fasilitas	Sederhana	Lengkap
Kapasitas	Terbatas	Besar
Suasana	<i>Personal</i>	Formal
Harga	Lebih Terjangkai	Lebih Mahal
Layanan	Sederhana	Lengkap
Ukuran	Kecil	Besar

Guest House adalah penginapan kecil yang biasanya dikelola secara pribadi dengan suasana yang lebih santai dan akrab, mirip seperti tinggal di rumah seseorang. Fasilitasnya sederhana, seperti kamar tidur, kamar mandi (kadang berbagi), dan mungkin dapur bersama. Layanan yang diberikan pun terbatas dan tidak seformal hotel. Hotel, di sisi lain, adalah penginapan yang berskala lebih besar dan profesional, sering kali memiliki banyak kamar serta fasilitas lengkap seperti restoran, layanan kamar, kolam renang, dan pusat kebugaran. Layanannya lebih formal dan

ditujukan untuk kenyamanan serta privasi tamu.

3. Keunggulan Guest House

- a. Harga lebih terjangkau
- b. Suasana yang lebih nyaman

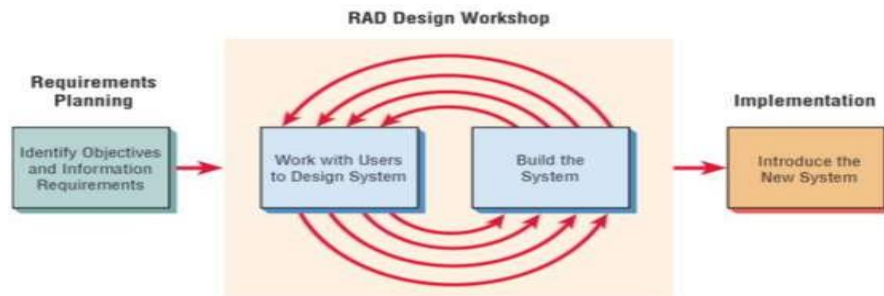
Cocok untuk wisatawan yang ingin pengalaman lokal yang lebih dekat

Guest House menjadi pilihan populer bagi *backpacker*, keluarga kecil, atau wisatawan yang mencari penginapan ekonomis dengan nuansa yang lebih Personal.

2.1.7 Rapid Application Development

Menurut (Santoso, 2022). *Rapid Application Development* (RAD) adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Pada umumnya, untuk mengembangkan sebuah sistem informasi membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam kurun waktu 30 sampai 90 hari. Pada saat proses pengimplementasian RAD, para pengguna bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan cara bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan sistem tersebut. Dengan metode RAD, suatu sistem dapat dengan cepat dihasilkan karena sistem yang dikembangkan dapat memenuhi keinginan dari para pengguna sehingga dapat mengurangi waktu untuk pengembangan ulang setelah tahap implementasi.

Pengembangan aplikasi pada pengembangan ini menggunakan salah satu metode yang merupakan bagian dari *System Development Life Cycle* (SDLC) yakni *Rapid Application Development* (RAD) sebagai metode pengembangan perangkat lunak yang menekankan pada pengembangan dalam waktu singkat dan menggunakan metode *iteratif* (berulang) dimana model kerjanya dikonstruksikan pada awal tahap pengembangan untuk menetapkan kebutuhan pengguna dan selanjutnya disingkirkan. Beberapa tahapan pengembangan RAD seperti terlihat pada Gambar 1 di antaranya meliputi fase perencanaan kebutuhan (*Requirement Planning Phase*), *workshop Design RAD* (*RAD Design workshop*), dan fase implementasi (*Implementation Phase*).



Gambar 2. 1 *Rapid Application Development*

2.1.8 *Flowchart*

Menurut (Syamsiah, 2019). *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan *Flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol yang standar yang dapat di mengerti

oleh programmer. Tahapan penyelesaian masalah yang disajikan harus tepat, sederhana, dan jelas.

Menurut (Agustinus et al., 2023). *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* sistem merupakan suatu urutan proses dalam system dengan menunjukkan alat dari media *input*, *output* serta jenis media yang digunakan untuk penyimpanan dalam proses pengolahan data sedangkan *flowchart* program merupakan suatu bagan dengan simbol-simbol tertentu yang menggambarkan suatu urutan dari proses secara detail dan berhubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

2.1.9 User Interface

Menurut (Rochmawati, 2019). *User interface* bukan hanya mengenai tampilan saja, akan tetapi lebih menekankan kepada usaha dalam kemudahan interaksi antara pengguna dengan sistem aplikasi, disesuaikan dengan tujuan interaksi manusia dengan sistem. Tujuan utama perancangan *user interface* adalah untuk memudahkan pengoprasian sistem dalam penyampaian konten informasi.

Menurut (Hartawan, 2022). *User Interface* (UI) adalah saat sistem dan pengguna dapat saling berinteraksi satu dengan lainnya melalui perintah seperti halnya menggunakan konten dan memasukan data.

2.1.10 Black Box

Menurut (Shaleh et al., 2021). Metode black box merupakan

sebuah metode yang biasa digunakan untuk menguji sebuah program dengan tidak harus memperhatikan setiap rincian dari program yang akan diuji. Di dalam uji black box ini hanya mengecek value dari masukan masing – masing. Dan tidak ada sebuah langkah atau upaya untuk mencari kode program yang digunakan untuk output. Keuntungan dalam menggunakan metode Black Box adalah dalam pelaksanaan pengujiannya tidak perlu memiliki pengetahuan yang dalam tentang pemrograman tertentu .

Menurut (Utomo et al., 2020). metode uji coba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu uji coba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat – syarat fungsional suatu program. Uji coba *black box* bukan merupakan alternatif dari uji coba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*. Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi