**USULAN TUGAS AKHIR**

**SISTEM INFORMASI GEOGRAFIS UNTUK PEMETAAN LOKASI PASAR TRADISIONAL DI KOTA MATARAM BERBASIS WEBSITE DENGAN METODE AGILE**

**A close up of a logo

Description automatically generated**

**Oleh:**

**SUBHAN**

**F1D016082**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**Desember 2021**

# BAB I PENDAHULUAN

## Latar Belakang

Pasar merupakan tempat bertemunya antara penjual dan pembeli yang juga ditandai dengan adanya proses transaksi atau tawar menawar barang yang diperjual belikan. Keberadaan pasar di suatu wilayah mampu meningkatkan perekonomian dan pendapatan masyarakat. Pasar tradisional merupakan salah satu pasar yang ramai dikunjungi oleh masyarakat sekitar karna menyediakan banyak sekali produk mentah maupun produk jadi yang memiliki harga yang relatif lebih murah dan kualitas yang bagus. Selain itu, terdapat produ-produk yang unik yang hanya terdapat di pasar tradisional Kota Mataram.

Kota Mataram adalah kota yang memiliki jumlah masyarakat luar daerah maupun turis yang selalu meningkat setiap tahunnya. Penyebab melonjaknya penduduk ini, diakibatkan karena Kota Mataram adalah kota pariwisata, pendidikan serta sebagai tempat diselenggarakan event-event besar. Saat ini terdapat banyak sekali pasar tradisional yang tersebar di beberapa kelurahan dan kecamatan di Kota Mataram. Namun terkadang, sedikit masyarakat luar daerah maupun turis yang mengetahui informasi dan lokasi Pasar Tradisional di Kota Mataram. Hal ini disebabkan karna beberapa pasar tradisional di Kota Mataram memiliki letak yang jauh dan sulit diketahui oleh masyarakat luar daerah atau turis tersebut.

Metode pengembangan dalam sistem informasi ini dirancang menggunakan metode *Dynamic System Development Model* (DSDM). DSDM merupakan salah satu metode pengembangan perangkat lunak dari agile *software* method. Dalam pengembangan DSDM terbagi menjadi dua rilis yaitu rilis fitur dasar sistem yang berfungsi mengidentifikasi peran pengguna dan rilis fitur pendukung untuk melengkapi kebutuhan sistem yang dikembangkan secara menyeluruh. Salah satu tujuan dari pengembangan ini adalah untuk pencarian lokasi terdekat dan pengolahan data yang mengedepankan kecepatan, kemudahan dan fleksibilitas sehingga pengembangan yang bersifat *agile* sangat diperlukan salah satunya adalah DSDM.

Oleh karena itulah, maka dibutuhkanlah sebuah sistem yang mampu membantu menggambarkan dan mencari jalur menuju lokasi pasar tradisional di Kota Mataram hingga informasi lengkap di pasar tersebut. Dibandingkan dengan *Google Maps* yang hanya menampilkan peta dan rute menuju lokasi saja, pada system yang akan dibuat pada penelitian ini memiliki keunikan atau kelebihannya yaitu tidak hanya menampilkan peta dan rute menuju lokasi saja tetapi juga dapat menampilkan informasi detail pasar seperti nama pasar, alamat serta deskripsi lainnya sesuai dengan data pasar yang ada pada Dinas Perdagangan Provinsi NTB. Sehingga dapat mempermudah masyarakat luar daerah maupun turis mengetahui informasi seperti produk apa saja yang khas ada disana dan jalur menuju lokasi pasar tradisional terdekat yang di Kota Mataram dalam bentuk peta digital.

## Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dipaparkan maka dirumuskan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana proses perancangan sistem informasi berbasis website yang tepat untuk menampilkan infromasi dan lokasi pasar tradisioanl di Kota Mataram dan dapat diakses secara *online* oleh masyarakat lokal maupun luar?
2. Bagaimana menentukan dan menggambarkan titik lokasi pasar tradisional dalam bentuk peta digital dan memberikan rute/jalur perjalanan dari posisi *user* berada menuju lokasi pasar yang diinginkan?

## Batasan Masalah

Dari permasalahan yang telah diuraikan berikut adalah batasan masalah dalam merancang dan membangun aplikasi pada penelitian ini :

1. Sistem informasi ini hanya menyediakan informasi berbasis website yang dapat diakses secara *online* oleh publik.
2. Pada penelitian ini, sistem hanya digunakan oleh masyarakat umum untuk melihat lokasi dan informasi mengenai pasar tradisional di Kota Mataram.
3. Pada penelitian ini, sistem hanya berfokus untuk melihat lokasi dan informasi mengenai Pasar Tradisional yang ada di Kota Mataram saja. Jadi sistem tidak berfokus pada pasar yang berada di luar Kota Mataram.
4. Pembuatan peta lokasi lapangan pada aplikasi ini menggunakan peta dari Leaflet dan untuk peta rute/jalur perjalanannya menggunakan *Google Maps*. Berbeda dengan *Google Maps*, sistem yang akan dibuat memiliki data dan informasi yeng lebih detail. Sehingga informasi yang diberikan lebih banyak dan detail.

## Tujuan Penelitian

Berdasarkan rumusan masalah di atas, berikut adalah tujuan dari penelitian ini sebagai berikut:

1. Membangun sebuah sistem yang dapat menampilkan informasi seperti lokasi, jalur dan informasi lengkap mengenai Pasar Tradiasional yang ingin dituju.
2. Membangun sebuah sistem informasi yang dapat membantu memberikan informasi kepada masyarakat lokal maupun luar Kota Mataram ataupun turis asing mengenai titik dan jalur menuju lokasi Pasar Tradisional di Kota Mataram dalam bentuk peta digital.

## Manfaat Penelitian

Berdasarkan tujuan dari penelitian ini, berikut adalah beberapa manfaat penelitian ini antara lain:

1. Dapat memudahkan masyarakat lokal maupun masyarakat luar dalam mencari tau informasi dan lokasi mengenai pasar tradisional yang ada di Kota Mataram.
2. Dapat membantu dinas perdagangan terkait dengan pengelolaan pasar yang ada di Kota Mataram.

## Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penyusunan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat serta sistematika penulisan laporan tugas akhir.

1. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan tentang penelitian-penelitian terkait yang sudah dilakukan sebelumnya. Serta membahas teori-teori yang berhubungan dengan topik penelitian.

1. BAB III METODE PERANCANGAN

Bab ini menguraikan tentang metode penelitian yang digunakan dalam merancang sistem. Seperti rencana pelaksanaan, alat, bahan, jalannya perancangan dengan metode yang telah ditentukan, perhitungan untuk hasil yang diharapkan serta cara pengujian sistem.

1. BAB IV HASIL DAN PEMBAHASAN

Pada bab ini merupakan pembahasan tentang analisis dan pembahasan sistem yang sudah dibuat, meliputi tampilan database dan implementasi sistem, serta hasil dari pengujianpengujian yang telah dilakukan untuk mendapatkan hasil yang sesuai.

1. BAB V KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dan saran dari hasil penelitian yang telah dilakukan Tuliskan sistematika penulisan laporan penelitian yang dimulai dari pendahuluan hingga kesimpulan. Sistematika dapat mengacu pada pedoman penulisan skripsi yang berlaku di PSTI.

# BAB II TINJAUAN PUSTAKA

## Penelitian Terkait

Beberapa penelitian yang terkait dengan rancang bangun sistem informasi Sistem Informasi Geografis Untuk Pemetaan Lokasi Pasar Tradisional Di Kota Mataram Berbasis Website Dengan Metode Aghile adalah sebagai berikut.

Murinto (2012), pernah melakukan penelitian dengan judul Pemanfaatan Sistem Informasi Geografis Berbasis Web Untuk Pemetaan Lokasi Pasar Dan Pusat Perbelanjaan Di Kota Solo. Penelitian ini menggunakan metode pengumpulan data dengan cara observasi, dokumentasi, wawancara dan pencarian di internet. Metode yang dilakukan dalam merancang sistem informasi geografis pasar tradisional dan pasar modern (pusat perbelanjaan) di kota Solo berbasis web adalah dengan menganalisa kebutuhan sistem, merancang sistem, melakukan digitasi peta dengan perangkat lunak ArcView 3.3, mengimplementasikan program dengan PHP dan MySQL, dan menguji program. Untuk melakukan evaluasi terhadap aplikasi ini dilakukan dengan metode black box test dan alpha test. Hasil dari penelitian ini adalah sebuah aplikasi Sistem Informasi Geografis untuk Pemetaan Lokasi Pasar Tradisional dan Pasar Modern (Pusat Perbelanjaan) di Kota Solo Berbasis Web yang dapat digunakan dinas pasar dan masyarakat untuk mengetahui lokasi pasar, barang yang dijual dipasar, dan jumlah kios.

Penelitian selanjutnya dilakukan oleh Ervan Heri Jatmika (2014) dengan judul Sistem Informasi Geografis Pemetaan Pasar Parangtritis Kabupaten Bantul Berbasis Web. Metode pengumpulan data dilakukan dengan cara wawancara, kuisioner, serta kajian pustaka. Untuk mempertimbangkan kelayakan sistem yang akan digunakan dengan meninjau kembali sistem berdasarkan beberapa hal yang dinilai berhubungan dengan informasi maupun fungsi sistem. Pengujian sistem dilakukan dengan uji coba, yaitu Black box test (Pengujian Kotak Hitam) dan Alpha test (Pengujian Alfa). Hasil dari penelitian ini adalah Sistem Informasi Geografis dengan visualisasi data spasial yang berisi informasi tentang pasar parangtritis berbasis web.

Vandha Pradwiyasma Widartha (2013) melakukan penelitian dengan judul Sistem Informasi Geografis untuk Perencanaan Penempatan Toko Modern di Kota Jember Menggunakan Metode AHP. Penelitian tersebut menggunakan 2 metode

yakni metode pengolahan data dan metode pengolahan sistem. Metode pengolahan data menggunakan metode AHP untuk menghasilkan rekomendasi lokasi pendirian toko modern baru, sementara metode pengembangan sistemnya menggunakan metode perancangan prototype. Hasil dari penelitian tersebut berupa sistem informasi geografis untuk merekomendasikan lokasi pendirian toko modern baru yang ada di Kota Jember.

Perbedaan dari beberapa penelitian sebelumnya dengan penelitian yang akan dilakukan penulis yaitu, lokasi yang akan digunakan penulis mencakup satu wilayah Kota Mataram. Metode penguumpulan data yang dilakukan mengguunakan metode observasi, wawancara, dan studi pustaka. Hasil dari penelitian yang akan dilakukan penulis berupa sebuah sistem informasi geografis untuk memetakan sebaran pasar tradisional yang ada di Kota Mataram. Proses dalam pembuatan sistem dilakukan dengan metode Aghile (*Dynamic System Development Model*).

## 2.2 Teori Penunjang

Berikut merupakan teori-teori penunjang atau umum yang digunakan dalam tugas akhir ini.

* + 1. **Google Maps API**

*Svennerberg* telah mencatat bahwa, Google Maps API adalah API yang paling popular di internet. Hasil pencatatan pada bulan Mei tahun 2010 menyatakan bahwa 43% mashup (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps API. Beberapa tujunnya dari penggunaan Googlea Maps API adalah melihat lokasi, serta mencari alamat dan lain sebagainya [1].

Google maps API adalah fungsi-fungsi pemrograman yang disediakan oleh Google maps agar Google maps bisa di integrasikan kedalam Web atau aplikasi yang sedang buat. Contoh sederhanya misalkan penulisingin membuat Sistem informasi Geografis kampus di Jogja, dengan memanfaatkan Google Maps API penulis bisa membuat GIS tanpa perlu memikirkan Peta Jogja, penulis tinggal pake Google maps dan memanggil fungsi fungsi yang dibutuhkan seperti menampilkan peta, menempatkan *marker* dan lain sabagainya [2].

* + 1. **GPS**

*Global Positioning System* (GPS) berfungsi untuk mengetahui letak tempat yang akan dituju dan mengetahui dimana pengguna berada, memberikan informasi yang tepat dan akurat mengenai posisi, kecepatan, arah, dan waktu dengan bantuan sinyal satelit [1].

* + 1. ***Website***

*Website* merupakan sebuah kumpulan halaman-halaman *web* beserta *file-file* pendukungnya, seperti *file* gambar, *video*, dan *file* digital lainnya yang disimpan pada sebuah *web server* yang umumnya dapat diakses melalui internet. Atau dengan kata lain, *website* adalah sekumpulan *folder* dan *file* yang mengandung banyak perintah dan fungsi fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data, dan lain sebagainya [3].

* + 1. **Sistem Informasi**

Sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan. Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada. Sistem Informasi juga dapat didefinisiskan sebagai perangkat elemen yang bekerja mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengawasan, analisis, dan visualisasi dalam organisasi [4].

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang menudukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [5].

* + 1. **Sistem Informasi Geografis**

Sistem Informasi Geografis merupakan suatu sistem informasi yang dirancang untuk bekerja dengan data yang bereferensi spasial atau berkordinat geografis atau dengan kata lain SIG adalah suatu sistem basis data dengan kemampuan khusus untuk menangani data yang bereferensi keruangan (spasial) bersamaan dengan seperangkat operasi kerja.

Sistem informasi geografis memiliki tiga unsur yakni sebagai berikut.

1. Sistem, dapat diartikan sebagai berbagai hal yang saling berkaitan atau saling mempengaruhi dalam mengerjakan proses untuk satu tujuan dalam SIG.
2. Informasi, sesuai dengan karakter SIG, informasi disini tentu saja adalah informasi tentang bumi (geografis) dengan apa yang ada di bumi. SIG merupakan media untuk menggambarkan apa yang ada di bumi dengan segala yang ada sesuai dengan tenpat atau lokasi dia berada. Informasi inilah yang menjadi obyek kerja SIG.
3. Geografis, geografis dalam SIG berarti sifat dari informasinya yaitu mengenai obyek-obyek atau hal-hal yang ada atau terjadi atau diperkirakan terjadi di muka bumi, tepatnya disuatu lokasi entah itu wilayah yang luas atau kecil kecil, bisa rumah, kampung, desa, kota, hutan, sawah, negara, bahkan dunia, tergantung dari maksud. Geografis atau informasi geografis bisa juga ditandai dengan data-data seperti koordinat [6].
   * 1. ***Framework CodeIgniter***

*Codeigniter* adalah sebuah *framework* PHP yang dibuat berdasarkan *design pattern* *model view controller* atau biasa disingkat MVC. *Design Pattern* adalah kumpulan penjelasan mengenai metode-metode bagaimana cara menyelesaikan suatu masalah yang umum ditemui dalam proses perancangan perangkat lunak (*Software Design*) [3].

* + 1. **PHP**

PHP Merupakan singkatan *recursive* dari PHP : *Hypertext Prepocessor*. Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994 [7]. PHP merupakan bahasa pemograman untuk membuat web. PHP dapat digunakan untuk membuat halaman *web* yang dinamis. PHP dapat digunakan pada sistem operasi Windows, Mac OS, Linux, dan sistem operasi yang lainnya.

* + 1. **XAMPP**

XAMPP adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket yaitu Apache, MySQL,dan PhpMyAdmin, Dengan Xampp pekerjaan menjadi sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis. Xampp telah mengalami perkembangan dari waktu ke waktu.versi yang terbaru adalah revisi dari yang terdahulu, sehingga lebih baik dan lebih lengkap. Aplikasi utama dalam paket Xampp yakni terdiri atas *web server* Apache, MySQL, PHP, dan PhpMyAdmin.

* + 1. ***Apache***

*Apache* adalah sebuah *web server open source*, jadi semua orang dapat menggunakannya secara gratis, bahkan anda bisa mengedit kode programnya. fungsi utama dari *apache* yakni menghasilkan halaman *web* yang benar sesuai dengan yang dibuat oleh seorang *web programmer*, dengan menggunakan kode PHP [8].

* + 1. ***Web server***

*Web Server* berfungsi untuk memberikan layanan *protocol* http, contoh aplikasi *web server* yaitu : apache, Microsoft IIS, Tomcat, Nginx, dll. [9].

* + 1. **PHP My Admin**

PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun *internet*. phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), *indeks*, pengguna (*users*), perijinan (*permissions*), dan lain-lain [10].

* + 1. **MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia[7]. MySQL adalah sistem manajemen *database* yang sering digunakan bersama PHP. PHP juga mendukung pada Microsoft Access, Database Oracle, dBase, dan sistem manajemen database lainnya. SQL (*Structured Query Language*) adalah bahasa terstruktur yang digunakan secara khusus untuk mengolah database, dan MySQL merupakan sebuah sistem manajemen *database* [8].

* + 1. ***Visual Studio Code***

*Visual Studio Code* menyediakan pengembang dengan pilihan baru alat pengembang yang menggabungkan kesederhanaan dan pengalaman ramping dari *code editor* yang terbaik dari apa yang pengembang butuhkan untuk siklus kode-builddebug inti mereka. *Visual Studio Code* adalah editor kode pertama, dan alat pengembangan lintas *platform* pertama - mendukung OS X, Linux, dan Windows [11].

* + 1. **UML**

Menurut Windu Gata, Grace (2013:4),Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupaka nmetodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem [12]. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. *Use Case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu:

Tabel 2.1 Tabel simbol *use case* diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Actor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. |
|  | Use Case | Fungsionalitas yang disediakan sistem sebagai unit- unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case. |
|  | Association | Komunikasi antara use case dan aktor yang berpartisipasi pada use case atau use case yang berinteraksi dengan aktor. |
| ---<<extend>>--- | Extend | Relasi *usecase* tambahan ke sebuah use case dimana *usecase* yang ditambahkan dapat berdiri sendiri walau tanpa *usecase* tambahan itu. Mirip dengan inheritance pada pemrograman berorientasi objek. |
| ---<<include>>--- | Include | Relasi use case tambahan dimana *usecase* yang ditambahkan memerlukan *usecase* ini untuk menjalan fungsinya. |

1. *Entity Relationship* Diagram (ERD)

*Entity Relationship* Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol [14]. Adapun elemen-elemen diagram ER adalah sebagai berikut:

1. Entitas (*Entity*)

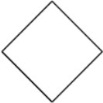
Entitas adalah objek yang dapat dibedakan dalam dunia nyata, sebagai contoh mahasiswa, dan dosen. Entitas terdiri atas beberapa atribut contohnya atribut dari entitas mahasiswa adalah nim, nama, dan alamat, selain itu pada setiap entitas harus memiliki satu buah atribut yang disebut dengan *primary key. Entity* disimbolkan dengan persegi panjang seperti Gambar 2.1.



Gambar 2.1 Simbol entitas

1. Relasi (*Relationship*)

Relasi adalah hubungan antara satu atau lebih entitas, contohnya yaitu relasi antara entitas mahasiswa dan mata kuliah dimana setiap mahasiswa bisa mengambil beberapa mata kuliah. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan, kardinalitas terdiri dari one to one, many to many, dan one to many. Contohnya Mahasiswa bisa mengambil banyak mata kuliah. *Relationship* disimbolkan dengan jajar genjang seperti pada Gambar 2.2 berikut [15].



Gambar 2.2 Simbol relasi

1. Atribut

Atribut adalah karakteristik dari tiap *entity* atau *relationship* yang menyediakan penjelasan detail mengenai *entity* atau *relationship* tersebut. Nilai dari atribut adalah data aktual atau informasi yang disimpan pada suatu atribut di dalam entity atau relationship, dimana tiap atribut memiliki domain (*value set*) sendiri. domain (*value set*) adalah batas-batas nilai yang diperbolehkan bagi suatu atribut.[15]. Simbol dari atribut adalah seperti Gambar 2.3.



Gambar 2.3 Simbol atribut

1. *Class* Diagram

Class diagram adalah diagram yang menggambarkan struktur dan deskripsi kelas serta hubungan antar kelas. Class diagram terdiri dari 3 bagian utama yaitu nama, atribut, dan operasi/method. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Pada class diagram terdapat beberapa simbol dan beberapa cara penulisan diantaranya adalah seperti pada Tabel 2.2.

Tabel 2.2 Tabel relasi class diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Class* | *Class* adalah sebuah objek yang menggambarkan sebuah keadaan nyata. Class memiliki: Nama kelas, atribut, dan method. |
|  | Association | Garis yang menghubungkan antara dua kelas atau lebih dan menunjukkan bahwa kelas yang terhubung menunjukkan sebuah relasi seperti : one-to-one (1 – 1), one-to-many (1 – M), many-to-many (M – M). |
|  | Aggregation | Menujukkan sebuah bagian relasi agregasi. |

1. *Sequence* Diagram

*Sequence* Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence* Diagram dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel simbol *sequence* diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Actor | Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. |
|  | Object Message | Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi. |
|  | Return Message | Menggambarkan pesan / hubungan antar obyek, yang menunjukan urutan kejadian yang terjadi. |

1. *Activity* diagram

*Activity* Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity* Diagram dapat dilihat pada tabel 2.4

Tabel 2.4 Tabel simbol *activity* diagram

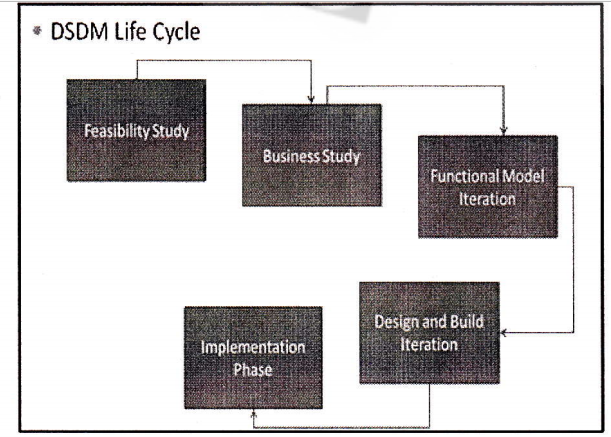
|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Activity* | Menunjukkan sebuah aktivitas yang dilakukan |
|  | *Initial node* | Awal dari aktivitas suatu sistem atau program |
|  | *Activity final node* | Akhir dari suatu aktivitas |
|  | *Decision* | Menujukkan sebuah pilihan |
|  | *Line connector* | Menghubungkan antara simbol yang satu dan yang lainnya |

* + 1. ***Black Box Testing***

*Black Box* adalah teknik pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengetesan pada spesifikasi fungsional. Tujuan *Black Box Testing* untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang telah diharapkan dan apakah informasi yang disimpan serta eksternal selalu dijaga kemutakhirannya. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih sebagai pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* mengidentifikasi jenis kesalahan dalam beberapa kategori antara lain fungsi yang tidak benar atau tidak ada, kesalahan antarmuka (*interface errors*), kesalahan pada struktur data dan akses basis data, kesalahan performansi (*performance errors*) dan kesalahan inisialisasi maupun terminasi [16].

* + 1. ***Dynamic System Development Model (DSDM)***

DSDM adalah sebuah kerangka kerja yang mengutamakan keterlibatan pengguna secara berkesinambungan dengan pendekatan pengembangan secara berulang dan bertambah, yang menangani proyek secara efektif dan efisien. DSDM memfasilitasi sebuah kerangka kerja untuk mengembangkan fungsi dengan cara yang lebih baik, memberikan fungsionalitas secara efisien dan efektif, dan memenuhi kebutuhan yang nyata dari suatu projek. DSDM dikembangkan dalam 5 fase, antara lain:[4]



Gambar 2.3 DSDM *Life Cycle*

1. ***Feasibility Study***

Kesesuaian proyek awal dinilai dalam fase ini. Fase ini membantu untuk mengidentifikasi kelayakan proyek. Fase ini membantu untuk mengidentifikasi

jawaban untuk beberapa pertanyaan seperti:

1. Apakah DSDM berlaku untuk proyek ini?
2. Apa saja kebergantungan yang muncul dalam proyek ini?
3. Apakah ada tantangan teknis?
4. Apakah ada keterbatasan sumber daya?
5. Apakah ada masalah organisasi yang berdampak dalam proyek?
6. Apakah ada risiko yang muncul, Jika demikian apa saja resiko tersebut?
7. Perkiraan tingkat tinggi dari skala waktu dan biaya.

Ruang lingkup dari studi kelayakan adalah untuk mengumpulkan rincian yang diperlukan tentang apakah solusi yang layak ada atau tidak. Analisis rinci dilakukan pada tahap selanjutnya. Laporan kelayakan (Feasibility report) adalah laporan tingkat tinggi yang memungkinkan komite pengarah proyek untuk memutuskan masa depan proyek, dan kebutuhan untuk studi kelayakan lebih lanjut.

1. ***Business Study***

Setelah melakukan analisis kelayakan pada langkah 7, langkah selanjutnya adalah menganalisis karakteristik bisnis dan teknologi. Studi Bisnis memberikan dasar untuk semua karya karya berikutnya. Fase ini mengarah pada garis rinci proses bisnis yang terkena dampak dan informasi yang mereka butuhkan. Fase ini mencoba menjawab beberapa pertanyaan kunci, seperti :

1. Apa Lingkup bisnis Proyek?
2. Adakah risiko yang harus ditinjau kembali?
3. Apa saja kebutuhan non-fungsionalnya (kinerja, kendala, subyektif)?
4. Prototype masa depan
5. Dasar untuk perkembangan teknologi
6. Prioritas persyaratan yang diidentifikasi pada langkah 1 dan langkah 2.

Fase ini menghasilkan sesuatu yaitu dalam mendefinisikan definisi area bisnis, kebutuhan prioritas, definisi arsitektur sistem dan rencana pengembangan.

1. ***Functionul Model lteration***

Fase ini memiliki tujuan untuk memberikan model fungsional yang terdiri dari kedua prototipe perangkat lunak yang bekerja dan model statis. Fase ini menghasilkan pengolahan informasi yangdiperoleh dalam penelitian bisnis.Fase ini menghasilkan modelfungsional, non fungsional, time *box plan*, dan *functional model review records.*

1. ***Design and Build Iteration***

Fase ini menyempurnakan *prototype* fungsional yang dikembangkan pada langkah 3 untuk memenuhi kebutuhanfungsional. Pada fase ini secara utamamengembangkan sistem untuk memenuhikebutuhan pengguna. Sebuah produk ujicoba adalah hasil utama dari fase ini.

Iterasi desain dan build ini terdiri dari ernpat kegiatan.

1. Mengidentifikasi persyaratan modul.
2. Merencanakan dan melakukan rencana sesuai dengan kebutuhan.
3. Mengembangkan modul, dan
4. Validasi fungsi modul. Design andlteration Build tercapai dalam sebuah rencana kotak waktu (*time box plan*), sistem yang diuji, prototipe desain, dan catatan pengujian.
5. ***Implementation Phase***

Fase ini meliputi transisi dari lingkungan pengernbangan untuk lingkungan operasional. Tujuan utama daritahap ini adalah untuk menempatkansistem yang diuji ke dalam lingkunganpengguna dan melatih individu – individuuntuk menggunakannya.

## BAB III

## METODE PENELITIAN

1. **Diagram AlurPembuatan Sistem**

Pembuatan sistem dengan menggunakan metode DSDM terdiri dari 3 tahap utama yaitu pra proyek, siklus hidup dan post proyek. Pada tahap pra proyek, semua data, anggaran dan komitmen proyek sudah terjamin. Sehingga dapat mengatasi masalah pada tahap akhir proyek. Lalu tahap kedua yaitu siklus hidup, dimana pada tahap ini memiliki 5 sub tahapan yaitu :

|  |
| --- |
|  |

Gambar 3. ER diagram SIG Pasar

Berikut merupakan penjelasan mengenai tahapan pengembangan sistem berdasarkan pada Gambar 3.1.

1. ***Feasibility Study***

Fase ini bertujuan untuk mempelajari dan mengetahui proses kelayakan dari sebuah proyek pada pasar yang ada di Kota Mataram. Pada fase ini, peneliti melakukan pembelajaran dari proses pembuatan jalur terdekat ke lokasi dan mendata informasi-informasi terkait pasar tersebut dengan cara mendatangi pasar dan mewawancarai salah satu pedagang di sana. Selain itu, peneliti juga mencari informasi-informasi penting seperti sejarah dan gambar-gambar di Google. Hal ini dilakukan agar proyek yang akan dibuat memiliki gambaran yang jelas dan maksimal.

1. ***Bussiness Study***

Pada fase ini, peneliti melakukan analisa karakteristik bisnis dan teknologi untuk lebih mengenal dan mempelajari proses bisnis yang ada pada setiap pasar. Hal ini bertujuan agar peneliti dapat mengetahui secara rinci tentang arsitektur sistem yang akan dibuat, kebutuhan-kebutuhan apa saja yang dibutuhkan serta dasar-dasar untuk rencana pengembangannya. \

1. ***Function Model Iteration***
2. ***Use case Diagram***

|  |
| --- |
|  |

Gambar 3. Use case diagram SIG Pasar

Gambar 3.2 merupakan *use case diagram* dari Sistem Informasi Geografis Pasar. Terdapat dua *user* yang akan menggunakan sistem ini, yaitu Administrator dan Masyarakat. Berikut adalah *use specification case* untuk seluruh *user* pada Sistem Sistem Informasi Geografis Pasar Kota Mataram.

* + 1. *Use Specification Case* pada Administrator

Tabel 3. *Use Specification Case* pada Administrator

| **Aktifitas** | **Penjelasan** |
| --- | --- |
| *Login* | Aktifitas *Login* merupakan suatu aktifitas yang pertama kali dilakukan oleh administrator untuk mengakses dan menggunakan Sistem Informasi Geografis Pasar. Administrator harus melakukan *Login* terlebih dahulu untuk dapat melakukan aktifitas-aktifitas yang lainnya. |
| Tambah Data Pasar | Aktifitastambah data pasar ini merupakan aktifitas yang hanya bisa dilakukan setelah administrator melakukan aktifitas *Login* pada Sistem Informasi Geografis pasar. Hal ini menandakan bahwa aktifitas tambah data program merupakan “*include*” dari aktifitas *Login*. |
| Hapus Data Pasar | Aktifitas untuk menghapus data pasar merupakan suatu aktifitas yang dilakukan dari sisi administrator yang hanya dapat dilakukan setelah administrator melakukan aktiftas *Login.* Hal ini menandakan bahwa aktifitas menghapus data pasar ini merupakan “*include*” dari aktifitas *Login* administrator. Pada *use case* tersebut (pada gambar 3.2) aktifitas menghapus data pasar ini juga merupakan “*extend*” dari aktifitas tambah data pasar, artinya untuk melakukan aktifitas menghapus data pasar ini administrator harus melakukan aktifitas tambah data pasar. |
| Edit Data Pasar | Aktifitas untuk mengedit data pasar merupakan suatu aktifitas yang dilakukan dari sisi administrator yang hanya dapat dilakukan setelah administrator melakukan aktiftas *Login.* Hal ini menandakan bahwa aktifitas mengedit data pasar ini merupakan “*include*” dari aktifitas *Login* administrator. Pada *use case* tersebut (pada gambar 3.2) aktifitas mengedit data program ini juga merupakan “*extend*” dari aktifitas tambah data pasar, artinya untuk melakukan aktifitas mengedit data pasar ini administrator harus melakukan aktifitas tambah data pasar. |
| Lihat Data pasar yang terverifikasi dan tidak terverifikasi | Aktifitas untuk melihat data pasar merupakan suatu aktifitas yang dilakukan dari sisi administrator yang hanya dapat dilakukan setelah administrator melakukan aktiftas *Login.* Hal ini menandakan bahwa aktifitas melihat data pasar ini merupakan “*include*” dari aktifitas *Login* administrator. Pada *use case* tersebut (pada gambar 3.2) aktifitas melihat data pasar ini juga merupakan “*extend*” dari aktifitas tambah data pasar, artinya untuk melakukan aktifitas melihat data pasar ini administrator harus melakukan aktifitas tambah data pasar. |
| Lihat Detail Data, Rute Lokasi dan Validasi Data Pasar | Aktifitas untuk melihat detail data, rute lokasi dan validasi data pasar merupakan suatu aktifitas yang dilakukan dari sisi administrator yang hanya dapat dilakukan setelah administrator melakukan aktiftas *Login.* Hal ini menandakan bahwa aktifitas melihat data pasar ini merupakan “*include*” dari aktifitas *Login* administrator. Pada *use case* tersebut (pada gambar 3.2) aktifitas melihat detail data, rute lokasi dan validasi data pasar ini juga merupakan “*extend*” dari aktifitas melihat data pasar dan merupakan “*extend*” juga dari aktifitas menambah data pasar, artinya untuk melakukan aktifitas melihat detail data, rute lokasi dan validasi data pasar ini administrator harus melakukan aktifitas tambah data pasar. |
| *Logout* | Aktifitas *Logout* merupakan aktifitas yang dilakukan untuk keluar dari seluruh aktifitas yang ada pada Sistem Informasi Geografis Pasar Kota Mataram. Aktifitas *logout* ini merupakan “*include*” dari aktifitas *login* yang artinya aktifitas *logout* hanya bisa dilakukan apabila administrator sudah melakukan *login* pada sistem. |

* + 1. *Use Specification Case* pada Masyarakat

Tabel 3. *Use Specification Case* pada Masyarakat

| **Aktifitas** | **Penjelasan** |
| --- | --- |
| Lihat Data Pasar yang terverifikasi | Aktifitas Lihat Data pasar yang terverfikasi merupakan suatu aktifitas yang dilakukan dari sisi masyarakat dan juga administrator tanpa harus melakukan *login.* Pada aktifitas ini administrator maupun masyarakat dapat melihat detail data pasar dan rute menuju lokasi pasar. |
| Tambah data pasar | Aktifitas tambah data pasar merupakan suatu aktifitas yang dilakukan dari sisi masyarakat yang dapat dilakukan tanpa harus melakukan *login.* Pada aktifitas ini masyarakat dapat menambah data pasar namun menunggu validasi dari administrator untuk dapat ditampilkan ke dalam system. |

1. **Class Diagram**

Sistem Informasi Geografis Pasar merupakan sebuah sistem yang dibuat dengan menggunakan *framework Code Igniter* yang berbasis MVC (*Model View Controller*), maka untuk *class diagram* pada pembahasan ini dibagi atas 2 *class diagram*, yaitu *class diagram controller* dan *class diagram model*. Berikut adalah bentuk *class diagram* dari Sistem Informasi Geografis Pasar.

Pada Gambar 3.3 dapat dilihat keterhubungan antar kelas pada *controller*, *model* dan *view* yang terdapat pada Sistem Informasi Geografis Pasar.

|  |
| --- |
|  |

Gambar 3. *Class diagram* keseluruhan sistem

1. ***Activity* diagram**

*Activity diagram* pada Sistem Informasi Geografis Pasar ini digunakan untuk menggambarkan semua proses yang ada pada sistem informasi ini. Berikut adalah *activity diagram* dari seluruh proses yang ada pada Sistem Informasi Geografis Pasar.

* 1. Proses *Login* Administrator

Pada Sistem Informasi Geografis Pasar ini, *user* administrator dapat melakukan *login* terlebih dahulu sebelum melakukan aktivitas yang lainnya terhadap sistem. Gambar 3.4menjelaskan aktivitas untuk masuk ke dalam sistem atau *login*  sistem. Proses pertama adalah sistem akan menampilkan halaman *login*, kemudian administrator diminta untuk meng-*input*-kan *username* dan *password* yang telah terdaftar di dalam sistem. Sistem akan memeriksa apakah *field* masih kosong atau tidak, jika masih kosong, maka administrator akan diminta lagi untuk mengisi semua form yang ada, jika form sudah terisi semua, maka proses *login* akan diproses sistem. Jika login gagal sistem akan menampilkan pesan kesalahan dan administrator diminta untuk memasukkan username dan password yang sesuai, jika *login*  berhasil, maka administrator akan dialihkan ke halaman administrator.



Gambar 3. *Activity diagram* untuk proses *login* admin

* 1. Proses Menambah Data Pasar

Pada Sistem Informasi Geografis Pasar ini, administrator dapat menambah data pasar. Gambar 3.5menjelaskan aktivitas untuk melakukan penambahan data pasar pada sistem. Proses pertama adalah sistem akan menampilkan form data pasar dan admin diminta untuk mengisi form kemudian system akan melakukan validasi form jika tidak sesuai maka admin akan diarahkan kembali mengisi form, jika sesuai maka data pasar akan tersimpan dalam *database* dan system akan menampilkan daftar data pasar*.*



Gambar 3. *Activity diagram* untuk proses menambah data pasar

* 1. Proses Menghapus Data Pasar

Pada Sistem Informasi Geografis Pasar ini, administrator dapat menghapus data pasar. Gambar 3.6 menjelaskan aktivitas untuk melakukan penghapusan data pasar pada sistem. Proses pertama adalah sistem akan menampilkan halamandaftar data pasar langsung pada sisi administrator, kemudian administrator dapat melakukan penghapusan data program. Kemudian setelah proses tersebut, maka perubahan data akan dikirim ke *database*. Setelah itu data pasar diperbarui di *database*.

.

Gambar 3.  *Activity diagram* untuk proses menghapus data pasar

* 1. Proses Mengedit Data Pasar

Pada Sistem Informasi Geografis Pasar ini, administrator dapat mengedit data pasar. Gambar 3.7 menjelaskan aktivitas untuk melakukan pengeditan data pasar pada sistem. Proses pertama adalah sistem akan menampilkan halamandata pasar langsung pada sisi administrator, kemudian administrator dapat melakukan pengeditan data pasar. Kemudian setelah proses tersebut, maka perubahan data akan dikirim ke *database*. Setelah itu data pasar diperbarui di *database*.

.

Gambar 3. *Activity diagram* untuk proses mengedit data pasar

* 1. Proses Melihat Rute Lokasi Pasar

Pada Sistem Informasi Geografis Pasar ini, Gambar 3.8 menjelaskan masyarakatdan administrator untuk dapat melakukan aktivitas melihat rute menuju lokasi pasar. Proses pertama adalah sistem akan menampilkan peta, kemudian masyarakat atau admindapat memilih salah satu lokasi pasar, kemudian sistem akan menampilkan rute menuju lokasi pasar yang diinginkan.



Gambar 3. *Activity diagram* untuk proses melihat rute lokasi pasar

### Sequence Diagram

*Sequence diagram* pada Sistem Informasi Geografis Pasar ini digunakan untuk menggambarkan kolaborasi antara *model*, *view*, dan *controller* untuk semua proses yang ada pada sistem. Berikut adalah penjelasannya.

* 1. *Login* administrator

Proses *login* dimulai ketika adminmemasukan *username* dan *password* pada *form login.* Lalu adminmengakses fungsi login() yang ada pada *controller* *C\_auth*, kemudian akan melakukan validasi apakah *username* dan *password* yang dimasukan sesuai dengan yang ada pada *database* menggunakan fungsi cek\_login(). Jika data yang dimasukkan tidak sesuai maka admin akan diarahkan kembali ke halaman login, jika data sesuai dengan data yang ada pada *database,* maka admin akan diarahkan ke halaman admin. *Sequence diagram* untuk proses login admin dapat dilihat pada Gambar 3.9.

|  |
| --- |
|  |

Gambar 3. *Sequence diagram* untuk proses login *admin*

* 1. Menambah Data Pasar

Prosesmenambah data pasar dimulai ketika *administrator* mengisi form data pasar*.* Kemudian admin mengakses fungsi tambah\_pasar() yang pada pada *Controller C\_admin*, kemudian system akan menjalankan fungsi tambah\_pasar()yang ada *Model M\_create* yang berfungsi untuk menyimpan data pasar ke dalam *database*. Setelah data tersimpan admin akan langsung diarahkan ke halaman *View V\_pasar*. *Sequence diagram* untuk proses menambah data pasar dapat dilihat pada Gambar 3.10.

|  |
| --- |
|  |

Gambar 3. *Sequence diagram* untuk menambah data pasar

* 1. Menghapus Data Pasar

Prosesmenghapus data programdimulai ketika *user* administrator memilih baris data pasar yang akan dihapus pada tabel data pasar*.* Administrator akan meng-klik hapus kemudian semua nilai akan dihapus melalui fungsi hapusdata() pada *Controller C\_admin*. Proses penghapusan juga melibatkan *Model M\_delete* untuk menghapus data pasar dari dalam database. Setelah data pasar terhapus, admin akan diarahkan ke halaman *View V\_pasar*. *Sequence diagram* untuk proses menghapus data pasar dapat dilihat pada Gambar 4.11.

|  |
| --- |
|  |

Gambar 3. *Sequence diagram* untuk menghapus data pasar

* 1. Mengedit Data Pasar

Prosesmengedit data pasardimulai ketika administrator meng-klik edit kemudian semua nilai yang di edit melalui fungsi edit\_pasar() pada *Controller C\_admin* akan dikirm ke fungsi edit\_pasar() yang ada pada *Model M\_update* untuk kemudian perubahannya disimpan ke dalam database. Setelah data pasar yang baru telah tersimpan, admin akan diarahkan ke halaman *View V\_pasar. Sequence diagram* untuk proses mengedit data pasar dapat dilihat pada Gambar 4.12.

|  |
| --- |
|  |

Gambar 3. *Sequence diagram* untuk proses mengedit data pasar

* 1. Mencari Rute Lokasi Pasar

Proses mencari rute menuju lokasi pasar dimulai ketika *user* memasukkan *keyword* dan menekan tombol “search”, kemudian *user* akan mengakses fungsi get\_pasar() yang ada pada *Controller C\_dashboard* dan get\_pasar() yang ada pada *Model M\_read* yang berfungsi untuk mengecek kesesuaian antara *keyword* yang dimasukkan dengan yang ada di *database*. Jika tidak *user* diarahkan untuk memasukkan *keyword* yang sesuai, jika sesuai maka sistem akan menampilkan hasil pencarian. *Sequence diagram* untuk proses melihat rute menuju lokasi pasar dapat dilihat pada Gambar 4.13.

|  |
| --- |
|  |

Gambar 3. *Sequence diagram* untuk proses melihat rute lokasi pasar

### Entity Relationship Diagram

Berikut adalah ERD (*Entity* *Relationship Diagram*) dari Sistem Informasi Geografis Pasar ini dapat dilihat pada gambar 3.14.

|  |
| --- |
|  |

Gambar 3. ER diagram Sistem Informasi Geografis Pasar

Berdasarkan ERD pada gambar 4.14 maka didapatkan tabel-tabel *database* sebagai berikut:

1. Tabel Admin

Tabel 3. 3 Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_admin | Int(5) | *Primary Key* |
| 2. | *Username* | Varchar(10) |  |
| 3. | *Password* | Varchar(10) |  |
| 4. | Nama\_admin | Varchar(30) |  |

1. Tabel Pasar

Tabel 3. Tabel Pasar

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_pasar | Int(5) | *Primary Key* |
| 2. | Nama\_pasar | Varchar(30) |  |
| 3. | Alamat\_pasar | Varchar(100) |  |
| 4. | Deskripsi\_pasar | Text |  |
| 5. | Foto\_pasar | Varchar(100) |  |
| 6. | *Latitude* | Varchar(25) |  |
| 7. | *Longitude* | Varchar(25) |  |
| 8. | Id\_status | Varchar(5) | *Foreign key* |
| 9. | Produk\_pasar | Text |  |

1. Tabel Status

3. Tabel Status

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_status | Int(5) | *Primary Key* |
| 2. | Nama\_status | Varchar(30) |  |
| 3. | Icon\_status | Varchar(100) |  |

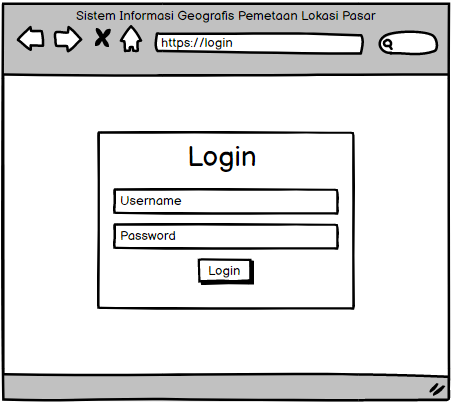
1. ***Design And Build Iteration***

### Rancangan Interface Sistem

Berikut adalah rancangan *interface* dari Sistem Informasi Geografis Pasar ini

1. Rancangan *Interface* Halaman Administrator
   1. Halaman *Login*

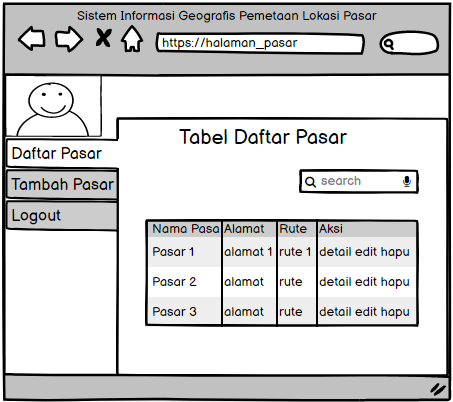
Halaman *login* merupakan halaman yang digunakan administrator untuk masuk ke dalam sistem. Rancangan *interface* halaman *login* admin dapat dilihat seperti pada gambar 3.15.



Gambar 3. Rancangan *interface* halaman *login*

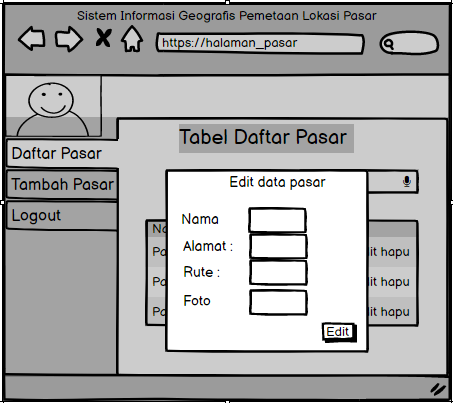
* 1. Halaman Data Pasar

Halaman Data Pasar merupakan halaman yang digunakan administrator untuk dapat melihat, mengedit dan menghapus data pasar serta melihat rute lokasi pasar. Rancangan *interface* halaman data pasar pada sisi admin dapat dilihat seperti pada gambar 3.16.

  
Gambar 3. Rancangan *interface* halaman Data Pasar

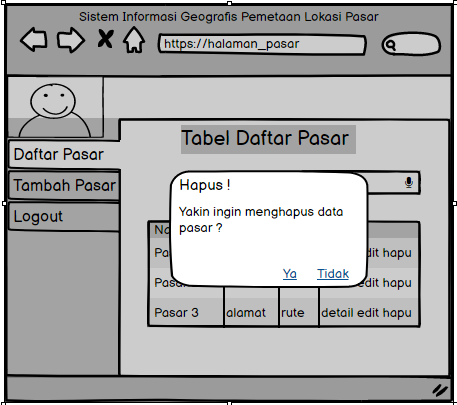
* 1. Halaman Edit Pasar

Halaman Data Pasar merupakan halaman yang digunakan administrator untuk dapat mengedit data pasar. Rancangan *interface* halaman edit data pasar dapat dilihat seperti pada gambar 3.17.

  
Gambar 3. Rancangan *interface* halaman Edit Data Pasar

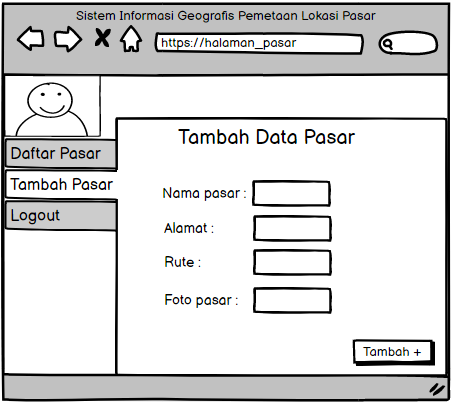
* 1. Halaman Hapus Pasar

Halaman Hapus Pasar merupakan halaman yang digunakan administrator untuk dapat menghapus data pasar. Rancangan *interface* halaman hapus data pasar dapat dilihat seperti pada gambar 3.18.

  
Gambar 3. Rancangan *interface* halaman Hapus Data Pasar

* 1. Halaman Tambah Pasar

Halaman Peta Pasar merupakan halaman yang digunakan administrator untuk menambah data pasar. Rancangan *interface* halaman tambah data pasar pada sisi admin dapat dilihat seperti pada gambar 3.19.

  
Gambar 3. Rancangan *interface* halaman Tambah Pasar

* 1. Halaman Validasi Data Pasar

Halaman validasi data pasar merupakan halaman yang digunakan administrator untuk melakukan validasi terhadap data-data pasar yang ditambahkan oleh masyarakat. Validasi data pasar dilakukan dengan cara menuju langsung ke titik lokasi sesuai yang telah ditambahkan oleh masyarakat. Validasi data dilakukan bertujuan untuk mengetahui apakah data pasar yang ditambahkan oleh masyarakat memang benar ada atau tidak. Rancangan *interface* halaman validasi data pasar pada sisi admin dapat dilihat seperti pada gambar 3.20.

  
Gambar 3. Rancangan *interface* halaman validasi data pasar

1. Rancangan *Interface* Halaman Masyarakat
   1. Halaman Rute Lokasi Pasar

Halaman rute lokasi pasar merupakan halaman yang digunakan masyarakat maupun admin untuk dapat melihat peta lokasi pasar yang ada pada system serta dapat melihat rute menuju lokasi pasar yang diinginkan. Rancangan *interface* halaman Rute Lokasi Pasardapat dilihat seperti pada gambar 3.21.



Gambar 3. Rancangan *interface* halaman *dashboard*

* 1. Halaman Detail Data Pasar

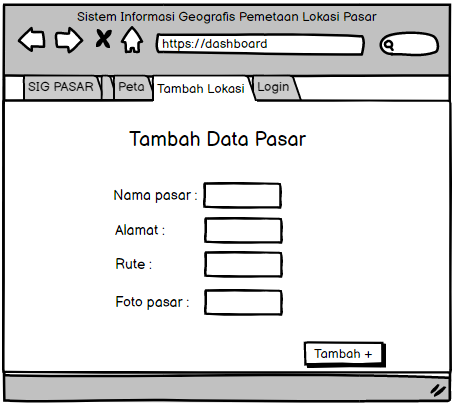
Halaman detail data pasar merupakan halaman yang digunakan masyarakat maupun admin untuk dapat melihat informasi detail mengenai data-data pasar diinginkan. Rancangan *interface* halaman detail data pasardapat dilihat seperti pada gambar 3.22.



Gambar 3. Rancangan *interface* halaman detail data pasar

* 1. Halaman Tambah Pasar

Halaman Tambah Pasar merupakan halaman yang digunakan masyarakat untuk menambah data pasar. Namun data yang ditambahkan oleh masyarakat perlu divalidasi oleh admin terlebih dahulu untuk dapat ditampilkan ke dalam peta lokasi yang ada pada system. Rancangan *interface* halaman tambah data pasar pada sisi masyarakat dapat dilihat seperti pada gambar 3.23.

  
Gambar 3. Rancangan *interface* halaman Tambah Pasar sisi masyarakat

# DAFTAR PUSTAKA

[1] W. N. Nugroho, “APLIKASI PENCARIAN MASJID TERDEKAT DI KOTA BANDAR LAMPUNG BERBASIS MOBILE MENGGUNAKAN ALGORITMA DIJKSTRA,” 2017.

[2] R. S. Hamsyah, “RANCANG BANGUN APLIKASI GO-BAN UNTUK MENCARI DAN MEMANGGIL TEKNISI TAMBAL BAN MENGGUNAKAN GOOGLE MAPS API,” 2018.

[3] Suhartini, M. Sadali, Y. K. Putra, and E. Al, “Sistem Informasi Berbasis Web Sma Al- Mukhtariyah Mamben Lauk Berbasis Php Dan Mysql Dengan Framework Codeigniter,” *infotek*, vol. 3, no. 1, pp. 79–84, 2020.

[4] M. Arifin and R. H. H. Hs, “Perancangan Sistem Informasi Pusat Karir Sebagai Upaya Meningkatkan Relevansi Antara Lulusan Dengan Dunia Kerja Menggunakan UML,” *IC-Tech*, vol. XII, no. 2, pp. 42–49, 2017.

[5] T. Sutabri, *KONSEP SISTEM INFORMASI*. Yogyakarta, 2012.

[6] G. S. Perdana, “Sistem informasi geografis tempat olahraga di provinsi daerah istimewa yogyakarta berbasis web,” 2017.

[7] A. Solichin, “Pemrograman Web dengan PHP dan MySQL,” pp. 1–122.

[8] U. Budi, “Aplikasi Peminjaman Dan Pengembalian Barang Menggunakan Rfid Dan Barcode Scanner Berbasis Web,” 2019.

[9] O. Suryana, “Server dan Web Server,” no. August, pp. 14–23, 2018.

[10] R. E. Standsyah and I. S. Restu, “Implementasi Phpmyadmin Pada Rancangan Sistem Pengadministrasian,” *J. UJMC, Vol. 3, Nomor 2, Hal. 38 - 44*, vol. 3, pp. 38–44, 2017.

[11] T. Kahlert and K. Giza, “Visual Studio Code - Code Editing. Redefined,” *Mikrosoft*, vol. 1, no. March, pp. 1–26, 2016.

[12] A. Hendini, “Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak),” *J. Khatulistiwa Inform.*, vol. 4, no. 2, p. 108, 2016, doi: 10.2135/cropsci1983.0011183x002300020002x.

[13] A. Hendini, “Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak),” *J. KHATULISTIWA Inform.*, vol. IV, no. 2, pp. 107–116, 2016.

[14] Y. Irawan and U. Rahmalisa, “Sistem Database Pemasyarakatan Studi Kasus Lapas Kelas II A Pekanbaru,” *J. Technopreneursh. Inf. Syst.*, vol. 2, no. 19, pp. 59–67, 2019.

[15] S. Agung, A. Kusyanti, M. Data, and E. Al, “Database Entity Relationship Diagram,” pp. 2–7, 2011.

[16] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, “Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 155, 2019, doi: 10.32493/jtsi.v2i4.3771.

[17] G. W. Sasmito, “Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal,” *J. Inform. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.