

BAB I

PENDAHULUAN

A. Latar Belakang

Akhir-akhir ini kebutuhan informasi akan peta dalam berbagai bidang semakin dirasakan. Hal ini dikarenakan peta merupakan suatu komunitas informasi visual yang sangat representatif. Banyak hal dapat diinformasikan peta dan tidak dengan teks. Seiring perkembangan teknologi informasi dan komunikasi, peta sekarang dapat diolah dan disajikan secara dinamis dalam bentuk aplikasi *GIS (Geographic Information Systems)*, baik berbasis desktop, web dan mobile.

Pada kantor UPTD Bina Marga Wilayah II Parepare, pendataan Jalan dan Jembatan sudah menggunakan komputer yakni aplikasi *Microsoft Excel* dan *Word* tetapi dalam pengumpulan data, unit daerah masing-masing datang dan melaporkan data mereka secara langsung dengan berupa data laporan teks. Dengan adanya aplikasi *Web GIS* secara online dapat memudahkan penyampaian informasi data jalan dan jembatan secara visual berupa peta sehingga mengurangi kesalahpahaman penyampaian informasi lokasi data.

Peneliti akan menerapkan sebuah aplikasi *GIS (Geographic Information Systems)* sebagai wadah untuk menampung informasi data jalan dan jembatan. Adapun ruang lingkup UPTD Bina Marga wilayah II Parepare meliputi daerah Pinrang, Sidrap, Parepare dan Enrekang.

Berdasarkan masalah di atas maka penulis melakukan penelitian dengan mengambil judul yaitu “Aplikasi *web GIS* jalan dan jembatan rusak pada UPT Bina marga Seajatappareng”. Diharapkan dengan menggunakan aplikasi *web GIS* ini dapat mempermudah pekerjaan pegawai kantor masing-masing daerah dalam penyampaian informasi dan pembuatan laporan umum Jalan dan Jembatan bagian Provinsi.

B. Rumusan Masalah

Dari latar belakang diatas maka didapatkan rumusan masalah sebagai berikut :

Bagaimana membuat sebuah Aplikasi web GIS jalan dan jembatan rusak pada UPT Bina marga Seajatappareng?

C. Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah, maka penulis membatasi pokok permasalahan sebagai berikut.

Direncanakan aplikasi *WEB GIS* ini menampilkan koordinat awal lokasi dan informasi data umum Jalan dan Jembatan rusak bagian Ajatappareng.

D. Tujuan Penelitian

Adapun tujuan penulis antara lain :

Membuat sebuah aplikasi *WEB GIS* untuk kantor UPT Bina Marga Seajatappareng.

E. Manfaat Penelitian

Manfaat dari Aplikasi *WEB GIS* Jalan dan Jembatan Pada UPT Bina Marga Seajatappareng adalah :

1. Manfaat bagi pembaca

Memberikan pengetahuan atau ide baru dalam dunia teknologi dan sebagai motivasi dalam mengembangkan dan menciptakan teknologi yang baru dan berguna pula.

2. Manfaat bagi penulis

Mendapat pengetahuan baru dalam menerapkan dan mengembangkan ilmu pengetahuan dan teknologi yang akan memberikan kesan tampilan yang berbeda dengan sistem yang lama, dimana pegawai kantor dan pihak pegawai lapangan dapat bekerja sama dalam pembaharuan data dan pembuatan laporan jalan dan jembatan sehingga mempermudah proses kerja.

F. Sistematika Penulisan

Untuk mempermudah penulis dan pembahasan selanjutnya, maka uraian pembahasan di tulis secara sistematis sebagai berikut :

BAB I : PENDAHULUAN, bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI, bab ini berisi landasan teori yang berkaitan dengan penyelesaian masalah yang diambil sebagai referensi dalam perancangan program.

BAB III : METODOLOGI PENELITIAN, bab ini berisi waktu dan lokasi penelitian, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian.

BAB IV : PERANCANGAN SISTEM, bab ini membahas tentang analisis perancangan sistem, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, Perancangan *Database* , Rancangan *Input/Output*, dan Pengujian sistem dengan teknik pengujian *White Box* dan *black box*.

BAB V : PENUTUP, bab ini berisi kesimpulan untuk menjawab rumusan masalah yang dibuat dan membuat saran.

BAB II

LANDASAN TEORI

A. Aplikasi

”Aplikasi adalah sekelompok atribut yang terdiri dari beberapa *form*, *report* yang disusun sedemikian rupa sehingga dapat mengakses data”. Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jogiyanto menambahkan aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal (Jogiyanto Hartono, 2019).

Aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, dan *Microsoft Excel*.

Aplikasi adalah penerapan, penggunaan atau penambahan data. Dari pengertian di atas, dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

B. Web

World Wide Web (WWW), sering disingkat dengan web, adalah suatu layanan di dalam jaringan internet yang berupa ruang informasi. Dengan adanya web, *user* dapat memperoleh atau menemukan informasi yang diinginkan dengan

cara mengikuti *link (hyperlink)* yang disediakan di dalam dokumen yang ditampilkan oleh aplikasi *web browser*.

Dalam teknologi web, informasi akan disajikan dalam bentuk dokumen yang disebut halaman web. Halaman web merupakan hasil terjemahan kode-kode program dari *file-file* yang tersimpan di dalam suatu komputer yang berperan sebagai *server web (web server)*. Komputer yang mengakses informasi disebut *client* melalui aplikasi yang disebut dengan *web browser*, sering disingkat *browser*. Contoh *web browser* saat ini banyak digunakan oleh pengguna *web* adalah *Internet Explorer*, *Mozilla Firefox* dan lain-lain.

Aplikasi *web browser* adalah *software* yang berfungsi untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang tersimpan dalam suatu *web server*. Untuk mengakses informasi dari suatu situs web, kita perlu melakukan instalasi aplikasi *web browser* pada komputer-komputer klien (Raharjo Budi, 2020).

Adapun Struktur Pembentukan Jaringan adalah :

1. LAN (*Local Area Network*)

Sebuah LAN, adalah jaringan yang dibatasi oleh area yang *relative* kecil, umumnya dibatasi oleh area lingkungan seperti sebuah perkantoran di sebuah gedung, atau sebuah sekolah, dan biasanya tidak jauh dari sekitar 1 km persegi. Beberapa model konfigurasi LAN, satu komputer biasanya di jadikan sebuah *file server*. Yang mana digunakan untuk menyimpan perangkat lunak (*software*) yang mengatur aktifitas jaringan, ataupun sebagai perangkat lunak yang dapat digunakan oleh komputer-komputer yang terhubung ke dalam *network*.

Komputer-komputer yang terhubung ke dalam jaringan (*network*) itu biasanya disebut dengan *workstation*. Biasanya kemampuan *workstation* lebih di bawah dari *file server* dan mempunyai aplikasi lain di dalam hardisknya selain aplikasi untuk jaringan. Kebanyakan LAN menggunakan media kabel untuk menghubungkan antara satu komputer dengan komputer lainnya.

2. MAN (*Metropolitan Area Network*)

Sebuah MAN, biasanya meliputi area yang lebih besar dari LAN, misalnya antar wilayah dalam satu propinsi. Dalam hal ini jaringan menghubungkan beberapa buah jaringan-jaringan kecil ke dalam lingkungan area yang lebih besar, sebagai contoh yaitu : jaringan Bank dimana beberapa kantor cabang sebuah Bank di dalam sebuah kota besar dihubungkan antara satu dengan lainnya. Misalnya Bank BNI yang ada di seluruh wilayah Ujung Pandang atau Surabaya.

3. WAN (*Wide Area Network*)

Wide Area Network (WAN) / Jaringan area Skala Besar adalah jaringan yang lingkupnya biasanya sudah menggunakan sarana Satelit ataupun kabel bawah laut sebagai contoh keseluruhan jaringan BANK BNI yang ada di Indonesia ataupun yang ada di Negara-negara lain. Menggunakan sarana WAN, Sebuah Bank yang ada di Bandung bisa menghubungi kantor cabangnya yang ada di Hongkong, hanya dalam beberapa menit. Biasanya WAN agak rumit dan sangat kompleks, menggunakan banyak sarana untuk menghubungkan antara LAN dan WAN ke dalam Komunikasi Global seperti Internet. Tapi bagaimanapun juga antara LAN, MAN dan WAN tidak banyak berbeda dalam beberapa hal, hanya lingkup areanya saja yang berbeda satu diantara yang lainnya.

Yang paling populer dari ketiga struktur jaringan diatas adalah *Local Area Network* (LAN). LAN merupakan *type* jaringan yang paling banyak digunakan.

Kelebihan dari LAN adalah :

- 1) Pertukaran *file* dapat dilakukan dengan mudah (*File Sharing*)
- 2) Pemakaian printer dapat dilakukan oleh semua *client* (*Printer Sharing*)
- 3) *File-file* data dapat disimpan pada server, sehingga data dapat diakses dari semua *client* menurut otoritas sekuritas dari semua karyawan, yang dapat dibuat berdasarkan struktur organisasi perusahaan sehingga keamanan data terjamin.
- 4) *Throughput* yang tinggi
- 5) Relatif lebih murah
- 6) *File* data yang keluar/masuk dari/ke server dapat di *control*
- 7) Proses *backup* data menjadi lebih mudah dan cepat
- 8) Resiko kehilangan data oleh virus *computer* menjadi sangat kecil sekali
- 9) Komunikasi antar karyawan dapat dilakukan dengan menggunakan *Email & Chat*
- 10) Bila salah satu *client/server* terhubung dengan modem, maka semua atau sebagian *computer* pada jaringan LAN dapat mengakses ke jaringan internet atau mengirimkan *fax* melalui 1 modem.

C. GIS

Geographic Information System (bahasa Indonesia: Sistem Informasi Geografis) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spesial (bereferensi keruangan). Atau yang dalam arti sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalkan data yang diidentifikasi menurut lokasinya, dalam sebuah *database*. Para Praktisi juga memasukkan orang (yang membangun dan mengoprasikannya) dan data sebagai bagian dari sistem (Riyanto. 2022).

Perkembangan GIS di Indonesia berjalan tidak terlalu cepat, diawali terlebih dahulu dengan perkembangan penginderaan jauh. Lembaga yang terlibat sejak awal dalam pengembangan penginderaan jauh seperti *LAPAN* (Lembaga Antariksa Penerbangan Nasional) dan *BAKOSURTANAL* (Badan Survei dan Pemetaan Nasional) perannya sangat besar dalam mempercepat perkembangan GIS di Indonesia.

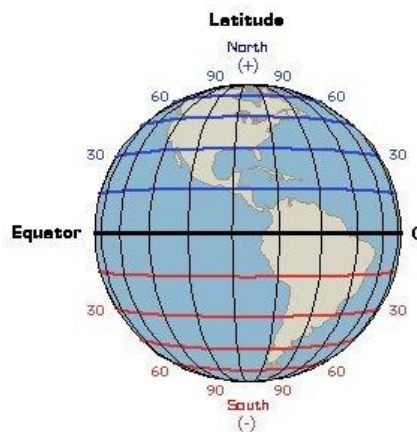
LAPAN adalah lembaga pemerintah Indonesia non-departemen yang menyediakan data digital khususnya untuk citra pengedaran jauh dengan satelit. Lembaga ini mempunyai stasiun bumi untuk merekam data pengedaran jauh seperti : citra *SPOT*, *MMS- Landsat* dan *TM- Landsat*. Data citra dari kelembagaan ini dapat di pesan baik dalam bentuk digital maupun dalam bentuk cetak. Sedangkan *BAKOSURTANAL* berperan mengkoordinasikan pemetaan dasar dan penyedia data digital berbentuk peta-peta dasar seluruh Indonesia. Pada lembaga ini juga tersedia layanan pembelian citra foto udara baik berupa data analog

maupun data gital, dan juga penyedia data yang bersifat grafis. Lembaga lain yang juga berperan mendorong perkembangan GIS dan pengindaran jauh di Indonesia adalah BPPT (Badan Pengkajian Dan Penerapan Teknologi).

1. Satuan Ukuran *Latitude* dan *Longitude*

- 1) *Latitude* adalah garis yang melintang di antara kutub utara dan kutub selatan, yang menghubungkan antara sisi timur dan barat bagian bumi. Garis ini memiliki posisi membentangi bumi, sama halnya seperti garis *equator* (khatulistiwa), tetapi dengan kondisi nilai tertentu. Garis lintang inilah yang dijadikan ukuran dalam mengukur sisi utara-selatan koordinat suatu titik di belahan bumi.

Latitude di bedakan menjadi 2 wilayah, yaitu utara atau yang biasa kita sebut lintang utara dan selatan atau yang biasa kita sebut lintang selatan, dimana nilai koordinat di bagian utara selalu positif dan nilai koordinat di bagian selatan adalah negatif.



Gambar 2. 1 Ukuran Garis Latitude

Berikut nilai-nilai yang dijadikan satuan ukuran garis lintang berikut :

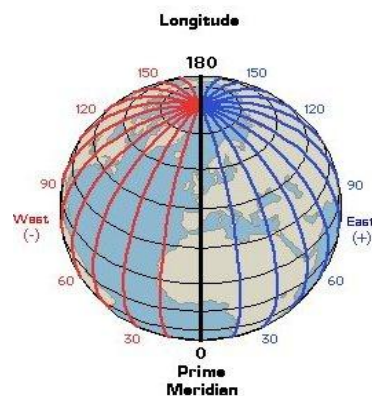
1. Garis paling atas (kutub utara) = 90 derajat
2. Garis paling tengah (*equator*) = 0 derajat
3. Garis paling bawah (kutub selatan) = -90 derajat

dengan mempersamakan derajat ke dalam bentuk satuan kilometer (km) maka ukurannya seperti berikut :

1 derajat *latitude* = 111 km

1 menit *latitude* = 1.85 km

- 2) Sedangkan *longitude* adalah garis membujur yang menghubungkan antara sisi utara dan sisi selatan bumi (kutub). Garis bujur ini digunakan untuk mengukur sisi barat-timur koordinat suatu titik di belahan bumi.



Gambar 2. 2 Ukuran Garis *Longitude*

Sama seperti *equator* pada *latitude* yang berada ditengah dan memiliki nilai 0 (nol) derajat, pada *longitude*, garis tengah yang bernilai 0 (nol) derajat disebut garis *prime meridian* (*garis bujur*).

Sedangkan garis yang berada paling kiri memiliki nilai -90 derajat, dan yang paling kanan memiliki nilai 90 derajat.

Longitude juga dibedakan menjadi 2 wilayah, yaitu bujur timur dan bujur barat, dimana koordinat yang berada di timur selalu bernilai negatif, dan sebaliknya yang berada di barat selalu positif. Nilai satuan ukuran derajat menjadi kilometer pada *longitude* juga sama seperti pada *latitude*.

Jadi, dalam metode pengukuran koordinat, suatu titik terlebih dulu diukur derajatnya berdasarkan *latitude* dan *longitude*-nya, setelah itu barulah di *translasikan* ke dalam bentuk satuan kilometer, baik itu dalam format *degree* (DDD) maupun *degree-minutes-second* (DMS).

D. UPTD Bina Marga Wilayah II

UPTD Bina Marga Wilayah II adalah Unit Pelaksana Teknis Dinas (UPTD) dari Bina Marga Provinsi Sulawesi Selatan sebagai penyelenggaraan urusan pemerintahan dan pelayanan umum di bidang Prasarana Jalan Transportasi yang berada di bawah naungan dan bertanggung jawab kepada Gubernur dan berdasarkan peraturan perundang-undangan dan azas otonomi yang berlaku.

Departemen Pekerjaan Umum, biasa disebut Departemen PU, sempat bernama “Departemen Permukiman dan Pengembangan Wilayah” (1999-2000) dan “Departemen Permukiman dan Prasarana Wilayah” (2000-2004), adalah departemen dalam Pemerintah Indonesia yang membidangi urusan pekerjaan umum. Departemen PU dipimpin oleh seorang Menteri Pekerjaan Umum yang sejak tanggal 21 Oktober 2004 dijabat oleh Ir. Djoko Kirmanto, Dipl. HE..

Pada awal terbentuknya Negara Kesatuan RI, susunan Kementerian berbeda. Dalam masa prolog G 30 S. PKI terjadilah dalam sejarah Pemerintahan RI suatu Kabinet yang besar disebut dengan nama Kabinet DwiKora atau Kabinet 100 Menteri, dimana pada masa ini dibentuk Koordinator Kementerian. Tidak luput Departemen PUT. yang pada masa itu ikut mengalami perubahan organisasi menjadi 5 Dept. dibawa Kompartemen PUT Kabinet Dwikora, dipimpin Jenderal Suprajogi. Adapun Kompartemen PUT ketika membawahi, antara lain:

1. Departemen Listrik dan Ketenagaan
2. Departemen Bina Marga
3. Departemen Cipta Karya Konstruksi
4. Departemen Pengairan Dasar
5. Departemen Jalan Raya Sumatera

Setelah peristiwa G30S. PKI Pemerintah segera menyempurnakan Kabinet Dwikora dengan menunjuk Ir. Soetami, sebagai menteri PUT untuk memimpin Kompartemen PUT. Kabinet yang disempurnakan itu tidak dapat lama dipertahankan.

E. Internet

Tidaklah mudah menjelaskan Internet ke dalam kata-kata. Dari satu sisi, Internet dapat di definisikan sebagai rajanya jaringan, orang bilang *networks-of networks*. Internet juga merupakan sistem jaringan yang paling tua, yang oleh beberapa orang disebut *The Grand Daddy of Computer Networks*. Sistem jaringan ini terbentuk tak kurang dari 40 juta komputer yang letaknya tersebar di lima benua. Tidak hanya kalangan industri yang memilikinya, tetapi termasuk pula

kalangan pendidikan, instansi pemerintah, badan riset, perkumpulan para *hobbyist*, dan bahkan para pemilik komputer pribadi (Ir. Pandapotan Sianipar, 2008).

Asal-usul *internet* berasal dari jaringan komputer yang dibentuk pada tahun 1970-an. Jaringan komputer tersebut disebut dengan *Arpanet*, yaitu jaringan komputer yang dibentuk oleh departemen pertahanan Amerika Serikat. Selanjutnya, jaringan komputer tersebut diperbaharui dan dikembangkan, dan sekarang penerusnya menjadi tulang punggung global untuk sumber daya informasi yang disebut dengan *internet*.

F. *PHP (HypertextPreprocessor)*

PHP singkatan dari *PHP Hypertext preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisipkan pada dokumen *HTML*. Penggunaan *PHP* memungkinkan *Web* dapat dibuat dinamis sehingga *maintenance* situs *Web* tersebut menjadi lebih mudah dan efisien. *PHP* merupakan *software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari situs resminya Kasiman (Peranginangin, 2019).

Pada awalnya *PHP* merupakan kependekan dari personal *home page* (situs personal). *PHP* pertama kali dibuat oleh *Rasmus Ledorf* pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreter (FI)*, yang wujudnya berupa sekumpulan *skrip* yang digunakan untuk mengolah data formulir dari *web*. Selanjutnya *Rasmus* merilis kode sumber tersebut untuk umum dan

menanamkannya *PHP/PI*. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*.

Pada November 1997, dirilis *PHP/PI* 2.0 pada rilis ini, *interpreter PHP* sudah diimplementasikan dalam program c. dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP/PI* secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter PHP* menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada juni 1998, perusahaan tersebut merilis *interpreter* baru untuk *PHP* dan meresmikan rilis tersebut sebagai *PHP* 3.0 dan singkatan *PHP* diubah menjadi akronim berulang *PHP : hypertext preprocessing*.

PHP memiliki 8 tipe data yaitu :

- 1) *Integer*
- 2) *Double*
- 3) *Boolean*
- 4) *String*
- 5) *Object*
- 6) *Array*
- 7) *Null*
- 8) *Nil*

Beberapa kelebihan *PHP* dari bahasa pemrograman *web* antara lain:

- 1) Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya,

- 2) *Web server* yang mendukung *PHP* dapat ditemukan dimana-mana dari mulai *Apache*, *IIS LIGHTTPD*, hingga *XITAMI* dengan konfigurasi yang relatif mudah.
- 3) Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
- 4) Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak,

PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux*, *unix*, *macintosh*, *windows*) dan dapat dijalankan secara *runtime* melalui konsule serta juga dapat menjalankan perintah-perintah sistem. Untuk menggunakan *PHP* ada beberapa *software* yang harus kita *install* diantaranya *apache server*, *PHP*, *PHP MY admin* dan *MySQL*.

G. Mysql

MYSQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar perlu mengakses *database* nya. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial) (Abdul Kadir, 2019).

MySQL merupakan program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan multi *user*.

MySQL pertama kali dirintis oleh seorang *programer database* bernama Michael Widenius. Selain *MySQL* ada beberapa program *database server* lain yang menggunakan standar *query* berupa *SQL*, antara lain adalah:

1. *Oracle*
2. *PostgreSQL*
3. *MySQL front*
4. *MySQL*
5. *SQL Server 97*

MySQL adalah sebuah *database* server, dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*, yang *open source* dengan kemampuan dapat berjalan baik di *OS (Operating Sistem)* manapun, dengan *Platform Windows* maupun *Linux*. *MySQL* termasuk jenis *RDBMS (Relational Database Management System)*. Itulah sebabnya, istilah seperti *table*, *baris* dan *kolom* digunakan pada *MySQL*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah *tabel*. *Tabel* terdiri atas sejumlah *baris* dan setiap *baris* mengandung satu atau beberapa *kolom*.

H. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari *X* (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya.

XAMPP adalah kepanjangan yang masing-masing hurufnya adalah :

- X : Program ini dapat dijalankan di banyak sistem operasi, seperti *Windows, Linux, Mac OS*, dan juga *Solaris*.
- A : *Apache*, merupakan aplikasi *web server*. Tugas utama *Apache* adalah menghasilkan halaman web yang benar kepada *user* berdasarkan kode *PHP* yang dituliskan oleh pembuat web. jika diperlukan juga berdasarkan kode *PHP* yang dituliskan, maka dapat saja suatu *database* diakses terlebih dahulu (misalnya dalam *MySQL*) untuk mendukung halaman web yang dihasilkan.
- M : *MySQL*, merupakan aplikasi *database server*. Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah *database* . *MySQL* dapat digunakan untuk membuat dan mengelola *database* beserta isinya. Kita dapat memanfaatkan *MySQL* untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database* .
- P : *PHP*, bahasa pemrograman web. Bahasa pemrograman *PHP* merupakan bahasa pemrograman untuk membuat web yang bersifat server-side scripting. *PHP* memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama *PHP* adalah *MySQL*. namun *PHP* juga mendukung sistem manajemen *database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL*, dan sebagainya.

P : *Perl* adalah bahasa pemrograman untuk segala keperluan, dikembangkan pertama kali oleh Larry Wall di mesin *Unix*. *Perl* dirilis pertama kali pada tanggal 18 Desember 1987 ditandai dengan keluarnya *Perl 1*. Pada versi-versi selanjutnya, *Perl* tersedia pula untuk berbagai sistem operasi varian *Unix* (*SunOS*, *Linux*, *BSD*, *HP-UX*), juga tersedia untuk sistem operasi seperti *DOS*, *Windows*, *PowerPC*, *BeOS*, *VMS*, *EBCDIC*, dan *PocketPC*. Dukungan terhadap pemrograman berbasis obyek (*object oriented programming/OOP*) ditambahkan pada *Perl 5*, yang pertama kali dirilis pada tanggal 31 Juli 1993. Proyek pengembangan *Perl 6* dimulai pada tahun 2000, dan masih berlangsung hingga kini tanpa tanggal yang jelas kapan mau dirilis. Ini dikatakan sendiri oleh Larry Wall dalam satu pidatonya yang dikenal dengan seri *The State of the Onion*. Dua di antara karakteristik utama *Perl* adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. *Perl* sangat populer digunakan dalam program-program *CGI (Common Gateway Interface)* dan berbagai *protokol Internet* lainnya. Seperti diketahui, *TCP/IP* sebagai basis bagi semua protokol *Internet* yang dikenal sekarang ini menggunakan format teks dalam komunikasi data. Seperti juga bahasa populer lainnya, *Perl* menerima banyak kritikan. Meski banyak di antaranya hanya berupa mitos, atau berlebih-lebihan, tapi terdapat juga sejumlah kritikan yang valid. Salah satunya adalah,

sintaksnya susah dibaca, karena banyak menggunakan simbol-simbol yang bukan huruf dan angka.

I. CSS

CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman *website (situs)*. Singkatnya dengan menggunakan metode CSS ini anda dengan mudah mengubah secara keseluruhan sekaligus memformat ulang situs (Prasetio Adhi, 2020).

CSS mempunyai 2 bagian utama yaitu *selectors* dan deklarasi. Yang dimaksud *selectors* biasanya *element HTML* yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut *style* yang ingin Anda ubah, dan setiap properti memiliki nilai

J. Javascript

Javascript adalah bahasa pemrograman yang digunakan untuk membuat web lebih dinamis dan interaktif. *Javascript* terintegrasi langsung dengan *HTML* (Prasetio Adhi, 2020).

Kode *Javascript* biasanya dituliskan dalam bentuk fungsi yang ditaruh di tag `<head>` yang dibuka dengan tag `<script type="teks/Javascript">`. Berbeda dengan *PHP* yang merupakan pemrograman *server side*, *Javascript* merupakan bahasa pemrograman *client side*. Untuk membuktikannya sangat mudah, coba anda buka *source code* sebuah *website* yang memiliki kode *PHP* dan *Javascript*. Anda tidak akan menemukan kode *PHP* dalam kode sumber mana pun, tetapi sangat besar kemungkinan anda akan menemukan kode *Javascript* seperti

layaknya kode *HTML*. Itu karena *Javascript* dijalankan di *client* (dalam hal ini *browser*), sedangkan *PHP* dijalankan di server.

Script Javascript akan dikirimkan ke browser bersamaan dengan kode *HTML* dan dijalankan oleh *browser*.

K. *Jquery*

JQuery merupakan suatu *framework (library) Javascript* yang menekankan bagaimana interaksi antara *Javascript* dan *HTML*. *JQuery* pertama kali dirilis pada tahun 2006 oleh John Resig. Pada perkembangannya *JQuery* tidak sekedar sebagai *framework Javascript*, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak *developer web* menggunakannya. *JQuery* memiliki slogan “*Write less, do more*” yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan *code*, tapi dengan hasil yang lebih banyak.

JQuery merupakan *library open source* dengan lisensi *GNU General Public License* dan *MIT License*. Dari sisi ukurannya, *framework JQuery* sungguh ramping, hanya sekitar 20 KB dan hanya terdiri dari satu *file*. Namun demikian, bagi yang menginginkan fungsi lebih, *JQuery* memungkinkan penambahan fungsionalitas dalam bentuk *plugin*. Saat ini tersedia ribuan *plugin* yang dapat diperoleh secara gratis di internet. Fitur utama dari *JQuery* diantaranya :

1. Dapat mengakses elemen dalam dokumen

Javascript khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan *Document Object Model* dan pengaksesan harus secara spesifik menyesuaikan dengan struktur *HTML*.

2. Mengubah tampilan halaman *website*

CSS (Cascading Style Sheet) menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman *web*.

3. Mengubah isi dari dokumen

Tidak hanya memberikan “kosmetik” pada halaman *web*, *JQuery* juga memberikan fasilitas untuk mengubah isi dari dokumen hanya dengan beberapa baris perintah.

4. Merespon interaksi *user*

Javascript sendiri memiliki beberapa *event-handling* seperti *onclick* untuk menangani *event* saat terjadi *click*.

5. Animasi pada dokumen

Animasi seringkali disertakan dalam suatu halaman *web* untuk menambah kecantikannya.

6. Mengambil informasi dari *server* tanpa harus *me-refresh* halaman merupakan salah satu konsep dasar yang dikenal dengan nama *AJAX (Asynchronous Javascript and XML)*.

7. Menyederhanakan penulisan *sintaks Javascript*

Semboyan *JQuery* adalah “*Write less, do more*” atau dengan kata lain kesederhanaan dalam penulisan *code*, tetapi menghasilkan tampilan yang lebih.

L. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah himpunan struktur data dan teknik untuk pemodelan dan desain program berorientasi objek (OOP) serta

aplikasinya. *UML* adalah metodologi untuk mengembangkan sistem OOP dan sekelompok *tool* untuk mendukung pengembangan sistem tersebut. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari tiga notasi yang telah ada sebelumnya yakni Grady Booch *OOD* (*Object-Oriented Design*), Jim Rumbaugh *OMT* (*Object Modeling Technique*), dan Ivan Jacobson *OOSE* (*Object-Oriented Software Engineering*).

Pada analisis aliran data, penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*

Tipe tipe *Diagram Unified Modeling Language (UML)* yang digunakan adalah sebagai berikut :

1. Diagram *Use Case*

Use Case diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor).

Menurut Prabowo Pudjo Widodo dan Herlawati (2020:16) “*Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya” *Use Case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Simbol *Use Case Diagram* dapat dilihat pada table 1

Tabel 2. 1 *Use Case Diagram*

Notasi	Keterangan	Simbol
<i>Actor</i>	<i>Actor</i> adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jika <i>actor</i> ini bisa berupa orang, perangkat keras, atau mungkin objek lain dalam sistem yang sama.	
<i>Use Case</i>	<i>Use Case</i> menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun <i>Use Case</i> hanya menjelaskan apa yang dilakukan oleh <i>actor</i> dan sistem, bukan bagaimana <i>actor</i> dan sistem melakukan kegiatan tersebut.	
Asosiasi	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>Use Case</i> Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>Actor</i> dengan <i>Use Case</i> .	

2. Diagram Activity

Menurut Prabowo Pudjo dan Herlawati (2020:143) Diagram *Activity* merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah lagi. orang yang melakukan masing-masing aktivitas.dan aliran sekuensial dari aktivitas-aktivitas tersebut. Diagram *Activity* digunakan untuk menjelaskan berbagai kegiatan pengguna (atau aplikasi), orang yang melakukan masing-masing aktivitas. dan aliran sekuensial dari aktivitas-aktivitas tersebut. Diagram *Activity* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang. bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi. dan bagaimana mereka berakhir. Adapun beberapa simbol Diagram *Activity* seperti pada tabel 2.

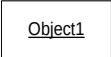
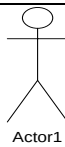
Tabel 2. 2 Diagram *Activity*

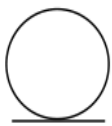
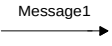
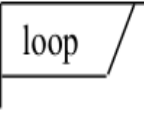
Notasi	Keterangan	Simbol
<i>Initial</i>	Titik awal, untuk memulai suatu aktivitas.	
<i>Final</i>	Titik akhir, untuk mengakhiri aktivitas.	
<i>Activity</i>	Menandakan sebuah aktivitas	
<i>Decision</i>	Pilihan untuk mengambil keputusan	
<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.	

3. Diagram *Sequence*

Menurut Prabowo Pudjo dan Herlawati (2020:150) *Sequence Diagram* digunakan untuk menggambarkan perilaku pada suatu *scenario*. Diagram ini menggunakan sejumlah contoh objek dan pesan yang diletakkan diantara objek didalam *usecase*. Adapun simbol *sequence Diagram* dapat dilihat pada tabel 3.

Tabel 2. 3 Diagram *Sequence*

Notasi	Keterangan	Simbol
<i>Object</i> (Partisipan)	<i>Object</i> atau biasa juga disebut partisipan merupakan instance dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>Object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.	

Notasi	Keterangan	Simbol
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>Object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah objek.	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	
<i>Boundary</i>	<i>Boundary</i> terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke <i>system</i> lainnya adalah termasuk dalam kategori.	
<i>Control</i>	berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi atau penanganan kesalahan.	
<i>Entity</i>	<i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.	
<i>Message</i>	<i>Message</i> , digambarkan dengan anakpanah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>Object-Object</i> .	
<i>Self-Message</i>	<i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.	
<i>Loop</i>	Operator <i>loop</i> adalah fragmen yang dapat mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi.	

M. Pengujian Sistem

Pengujian sistem merupakan proses pengeksekusian program atau sistem secara keseluruhan. Tujuan melakukan pengujian sistem adalah untuk memastikan bahwa komponen-komponen dari sistem telah berfungsi sesuai dengan yang

diharapkan. Pengujian ini dilakukan dengan menggunakan metode *Black box* dan *White Box testing*.

1. Metode *Black Box*

Black box adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interface* nya) , fungsionalitasnya.tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*).

2. Metode *White Box*

a. Uji Coba Sistem dan Program

Untuk menguji sistem informasi yang dirancang, maka di gunakan pendekatan *White Box* yang merupakan salah satu metode pengujian yang menggunakan struktur *control desain procedure* untuk memperoleh *test case*. Jadi dengan menggunakan metode ini, penulis dapat mengetahui cara kerja sistem yang dirancang secara terperinci sesuai spesifikasi dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah sesuai dengan baik dan benar.

Dengan pengujian ini dapat dipastikan bahwa :

- 1) Semua jalur *independent patch* telah digunakan minimal satu kali.

2) Telah menggunakan semua keputusan logika pada kedua sisinya baik itu *true* atau *false*.

3) Telah mengeksekusi semua *loop* pada batasan maksimal yang mungkin terjadi.

4) Menggunakan struktur data internal untuk menjamin validasi

Sebagai *alternative* pengembangan *white box* testing, digunakan metode *Basis Patch Testing* yang berguna untuk :

1) Mengukur Kompleksitas *logic* dari desain *procedure* dan menggunakannya sekaligus sebagai pedoman untuk mendapatkan konsisten dari jalur aplikasi.

2) Pengujian yang dilakukan dijamin menggunakan *statement* dalam program minimal satu kali selama pengujian.

3) Menghitung *cyclometris complexity* sebagai ukuran kontitif untuk menentukan jumlah *independent patch* sebagai jalur yang perlu di uji.

Dalam pengujian menggunakan basis *Patch*, ada beberapa elemen penting yang harus diketahui antara lain :

1) Node (N), yaitu symbol yang mewakili suatu proses yang ada pada setiap *flowchart*, simbol Node (N) adalah sebagai berikut

:



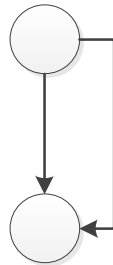
- 2) Edge (E), yaitu garis-garis yang menghubungkan node satu dengan node yang lainnya pada *flowchart*. Simbol *Edge* (E) adalah sebagai berikut ;



- 3) Prediksi (P), yaitu node yang mempunyai minimal dua buah edge Simbol Predikat (P) adalah sebagai berikut :



- a) *Region* (R), yaitu suatu wilayah tertutup yang terbentuk dari *Node* dan *Edge*. Simbol *Region* (R) adalah sebagai berikut :



- b) Rumusan yang digunakan dalam pengujian *Basis Patch* adalah sebagai berikut :

$$1) \quad V(G) = E - N + 2$$

$$2) \quad V(G) = P + 1$$

Pengujian pada suatu aplikasi dapat dikatakan berhasil jika hasil dari rumusan 1 dan rumusan 2 adalah sama.

b. Teknik Pengujian

1. *Basis Patch System*

Salah satu teknik pengujian menggunakan sistem *White Box* adalah *Basis Patch* Sistem. Metode *Basis Patch* digunakan untuk menentukan ukuran kompleksitas logika (*Logika complexity measure*) dari suatu desain. Ukuran kompleksitas logika berguna untuk menentukan banyaknya pengujian yang dilakukan untuk menjarin semua jalur di uji setidaknya sekali.

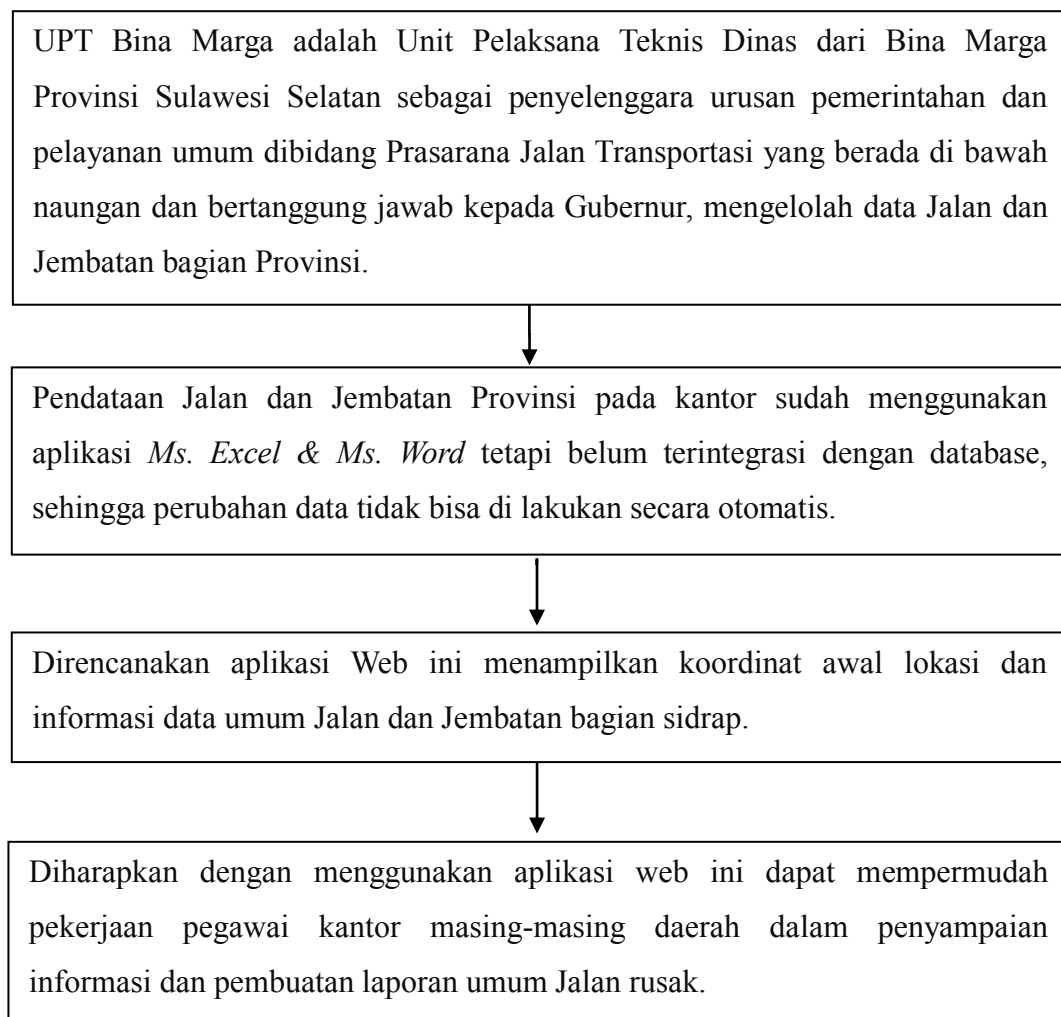
Teknik ini mempunyai 3 langkah dalam pelaksanaannya yaitu :

- 1) Teknik *procedural* dengan menggambarkan *flowgraph* perancangan procedural di beri nomor untuk memudahkan pembuatan *flowgraph*.
- 2) Menentukan *Cyclomatic Complexity* untuk *flowgraph* yang dibuat. Dengan rumusan $V(G) = E - N + 2$, dimana $V(G)$ adalah jumlah *region* pada *flowgraph*.
- 3) Menentukan *independent patch* pada *flowgraph*. *Independent patch* adalah jalur yang melintas dalam sistem dimana sekurang-kurangnya terdapat ada *flowgraph*, E adalah jumlah *Edge*, dan N adalah jumlah Node pada proses perintah yang baru. Dalam istilah *flowgraph*, *independent Path* harus bergerak sekurang-kurangnya pada satu *Edge*, dimana pada *Edge* yang akan

dilewatkan tidak dapat di lewati *edge* sebelum jalur tersebut didefinisikan.

Pengujian sistem merupakan proses eksekusi suatu program atau sistem secara keseluruhan. Pengujian menunjukkan bahwa fungsi dari perangkat lunak bekerja sesuai dengan spesifikasi dan bahwa persyaratan kinerja telah dipenuhi.

N. Kerangka Berpikir



Gambar 2. 3 Kerangka Pikir

O. Penelitian Terdahulu

1. Nida Nafsul Muthmainnah, 2021, Jurusan Teknik Informatika Fakultas Teknik Universitas Sriwijaya, **Analisis Perbandingan Kondisi Jembatan Menggunakan Program Bridge Management System (Bms) Dan Aplikasi Inspeksi Visual Jembatan (Invi – J) (Studi Kasus: Jembatan Pada Jalan Mayjen Yusuf Singadekane, Palembang)**, Salah satu cara merawat dan memelihara jembatan adalah dengan menggunakan beberapa sistem, salah satunya adalah *Bridge Management System (BMS)*. *Bridge Management System (BMS)* merupakan salah satu cara untuk dapat mempertahankan kondisi jembatan melalui proses investigasi berkala pada suatu jembatan sehingga dapat menentukan tahap perawatan dan perbaikan. Agar *BMS* dapat bekerja dengan efektif dan efisien sangat dibutuhkan informasi yang baik tentang jembatan tersebut. Informasi tersebut tergantung dari ukuran dan kompleksitas dari sistem yang akan dibangun, tetapi pada dasarnya semua sistem tersebut mempunyai hubungan dengan inventaris, inspeksi, perawatan dan keuangan.
2. Mustika Anindita, 2021, Universitas Telkom, **Aplikasi Pengelolaan Data Kerusakan Jalan Dan Jembatan Berbasis Web Dan Android Menggunakan Sistem Monitoring Google Maps Api (Studi Kasus : Dinas Pekerjaan Umum Kabupaten Magetan)**. Dinas Pekerjaan Umum Kabupaten Magetan adalah dinas yang berwenang mengatasi kerusakan jalan dan jembatan. Pengelolaan data kerusakan jalan dan jembatan seperti melakukan pengecekan ke tempat rusaknya jalan dan jembatan, pencatatan

dan pembuatan laporan masih secara manual. Melihat permasalahan tersebut maka muncul gagasan untuk membangun aplikasi berbasis *web* dan *android*. Metode yang digunakan dalam membangun aplikasi ini adalah metode *Waterfall*. Bahasa yang digunakan adalah *PHP* dengan *framework Codeigniter* dan *Java*, serta editor yang digunakan adalah *Notepad++* dan *Eclipse Luna SR2 (4.4.2)*. Untuk tampilan menggunakan *Bootstrap Bucket Admin*. *Database* yang digunakan adalah *MySQL*. Aplikasi ini digunakan oleh pegawai khususnya bidang bina marga dan kepala bidang bina marga. Dengan adanya aplikasi ini diharapkan dapat menangani permasalahan dalam melakukan pengecekan dan pengelolaan data kerusakan jalan dan jembatan.

3. **Andika Muhamad Rafi, 2020, UIN Sunan Gunung Djati Bandung, Aplikasi Jalan Cantik, Memperbaiki Jalan dan Jembatan Menjadi “Cantik” Kembali**, untuk mengatasi terjadinya hambatan dimana masyarakat memberikan masukan kepada pemerintah terkait infrastruktur jalan serta jembatan yang tidak layak supaya untuk segera diperbaiki maka perlu adanya suatu inovasi yang dilakukan oleh pemerintah kepada masyarakat terkait tanggapan untuk mengatasi kendala tersebut. Maka dari itu, pemerintah memanfaatkan *e-government* sebagai sarana untuk merespon masukan dari masyarakat serta dengan memberikan pelayanan yang baik kepada masyarakat. Mengapa perlu memanfaatkan *e-government*? Sebab di masa kini dengan perkembangan teknologi yang canggih serta masyarakat sudah paham akan teknologi

maka pemerintah pun harus mengikuti perkembangan zaman dengan memanfaatkan *e-government*, hal ini menjadi sebuah inovasi masyarakat untuk mendapatkan pelayanan yang baik dari pemerintah untuk memenuhi kebutuhan masyarakat.

BAB III

METODOLOGI PENELITIAN

A. Waktu Dan Lokasi Penelitian

Pelaksanaan penelitian dilakukan pada bulan Februari-Juni 2023 dan lokasi penelitian di kantor UPT Bina Marga Wilayah III Kota Parepare, Sulawesi Selatan.

B. Jenis Penelitian

Penulis dalam hal ini menggunakan metode penelitian Deskriptif yaitu kegiatan yang dilakukan dengan cara mengumpulkan data secara langsung dari objek penelitian melalui wawancara, pengamatan langsung dan pengumpulan dokumen.

C. Metode Pengumpulan Data

Di penelitian ini, penulis memperoleh data dengan metode sebagai berikut:

1. Observasi

Dilakukan dengan mengadakan penelitian langsung dengan kantor yang terkait untuk mengumpulkan informasi dan data yang dibutuhkan dalam menunjang permasalahan.

2. Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dengan responden. Komunikasi ini berlangsung dalam bentuk tanya jawab dalam hubungan tata muka guna memperoleh data langsung

3. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data dengan menggunakan buku-buku, laporan-laporan atau bacaan lain sebagai bahan referensi dalam penulisan laporan dan pembuatan sistem.

D. Alat Dan Bahan Penelitian

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan tersebut. Alat dan bahan yang diperlukan antara lain:

1. Perangkat keras

Perangkat keras yang digunakan dalam pembuatan Aplikasi Web Jalan dan Jembatan Pada UPT Bina Marga seajataappareng adalah sebuah laptop dengan spesifikasi minimal yang dapat dilihat pada tabel 4 berikut :

Tabel 3. 1 Spesifikasi Minimal Perangkat Keras

Jenis	Spesifikasi
<i>Laptop</i>	<i>ACER</i>
<i>Processor</i>	<i>Core-i3-3217U 1,8 GHz</i>
<i>Memory</i>	<i>4 GB</i>
<i>Hardisk</i>	<i>500 GB</i>

2. Perangkat Lunak

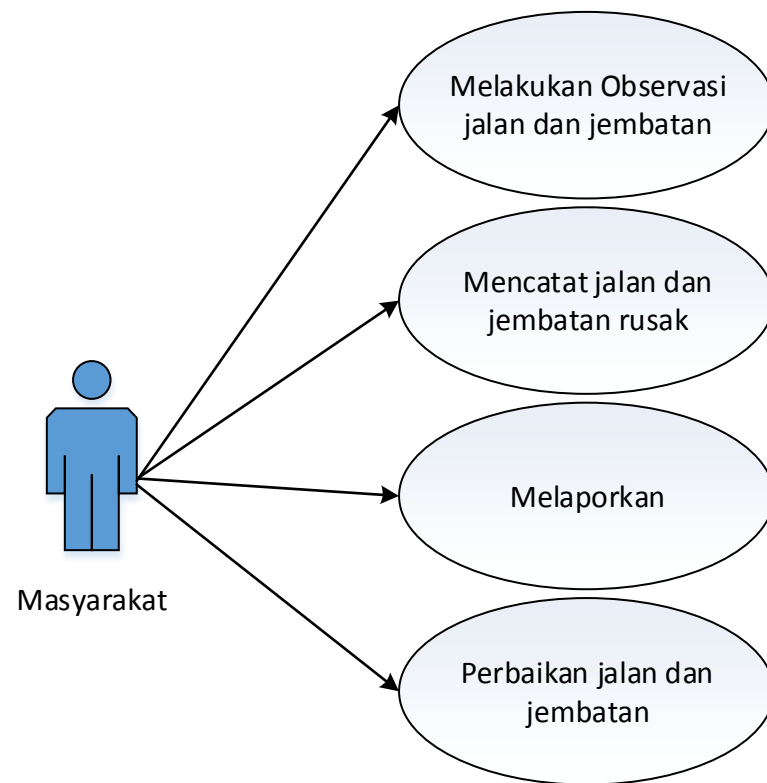
Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada Tabel 5 berikut :

Tabel 3. 2 Spesifikasi Minimal Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 11</i>
Bahasa Pemrograman	<i>PHP</i>
Database	<i>MySql, Notepad++, Netbeans, XAMPP</i>

E. Desain Sistem

1. Sistem Yang Berjalan



Gambar 3. 1 Sistem Yang Berjalan

BAB IV

ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Analisis dan perancangan sistem ialah tahapan pengumpulan kebutuhan data dan informasi dan kemudian diolah sehingga dapat di definisikan kebutuhan yang harus dipenuhi oleh sistem yang akan dibangun. Jenis data yang dikumpulkan selama proses pengambilan data berupa data sekunder. Data sekunder merupakan berupa data yang di peroleh langsung oleh *user* yang mendaftar pada aplikasi kami dengan cara memperkenalkan aplikasi kami pada grup grup diskusi yang secara *online*, baik grup *whatschap*, *facebook* dan lain lain.

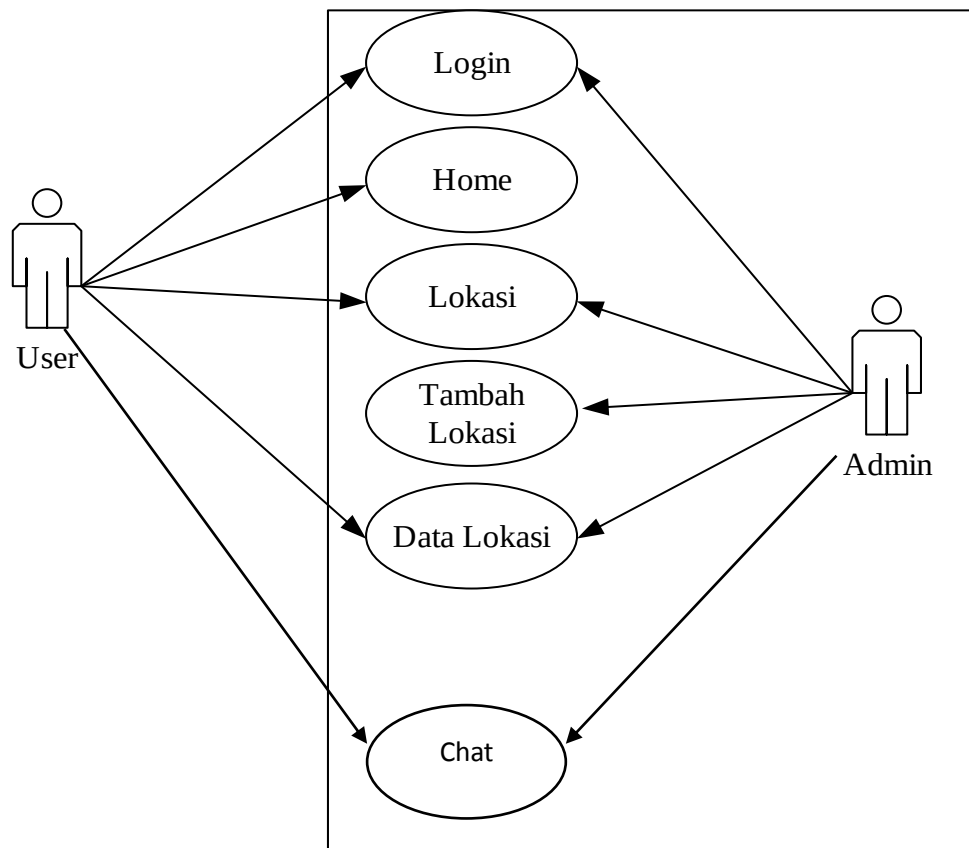
B. Use Case Diagram

Perancangan sistem adalah pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk. Adapun kegiatan dalam tahapan ini antara lain :

1. Use case diagram

Use Case Diagram adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem. Sebuah *use case diagram* merepresentasikan interaksi yang terjadi antara *user* dengan proses atau sistem yang dibuat. *Use Case diagram* menggambarkan kegiatan yang dilakukan pada *user* pada sistem yang akan

dibangun sehingga *use case diagram* dari sistem yang dibangun dapat dilihat pada gambar berikut ini :



Gambar 4. 1 *Use Case Diagram*

Pada gambar 4.1, dapat dijelaskan bahwa *User* dapat melakukan *Login* diri di aplikasi dan melihat lokasi yang telah disiapkan, setelah itu admin dapat melakukan mengecek data lokasi dan tambah lokasi.

2. Tabel Use case diagram

Penjelasan Use Case Diagram di atas :

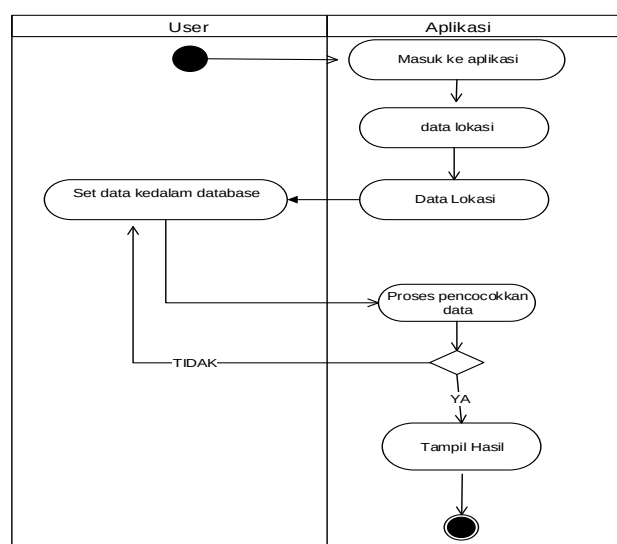
a. Actor Pengguna

Tabel 4. 1 Tabel *use case*

Nama Use Case	Deskripsi Use Case
<i>Login</i>	<i>Use Case</i> ini menjelaskan bahwa setelah <i>user</i> mengaktifkan aplikasi, akan terbuka tampilan awal sistem.
<i>Home</i>	<i>Use Case</i> ini menjelaskan bahwa setelah <i>user</i> mengaktifkan aplikasi, akan terbuka tampilan <i>home</i> yang disipkan aplikasi.
Lokasi	<i>Use Case</i> ini menjelaskan bahwa setelah <i>user</i> membuka aplikasi maka dapat melihat lokasi didalam aplikasi.
Tambah Lokasi	<i>Use Case</i> ini menjelaskan bahwa setelah admin mengisi Lokasi yang akan di input kedalam sistem.
Data Lokasi	<i>Use Case</i> ini menjelaskan bahwa setelah <i>admin</i> melakukan penginputan data lokasi maka akan ditampilkan semua data lokasi yang sudah di input oleh <i>admin</i> .
Chat	<i>Use Case</i> ini menjelaskan bahwa setelah <i>user</i> membuka aplikasi maka dapat melihat fitur chat didalam aplikasi.

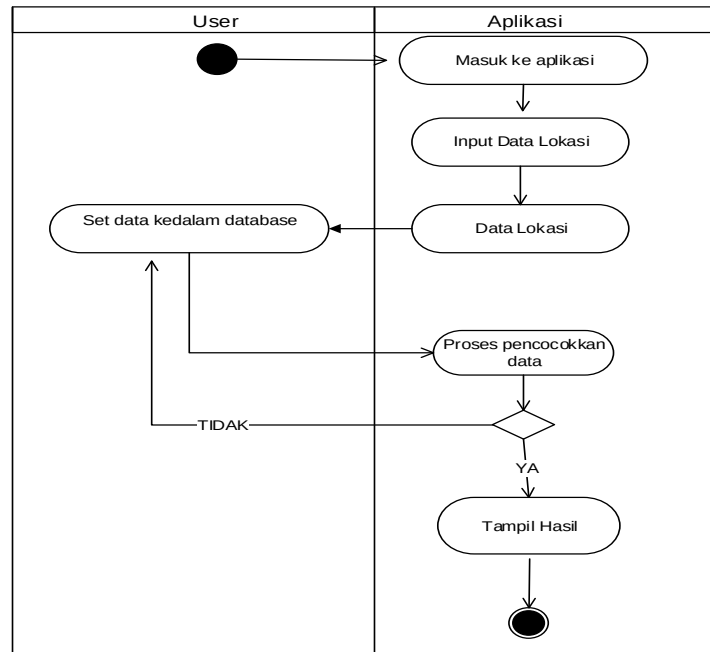
1. Activity Diagram

a. Activity Diagram User



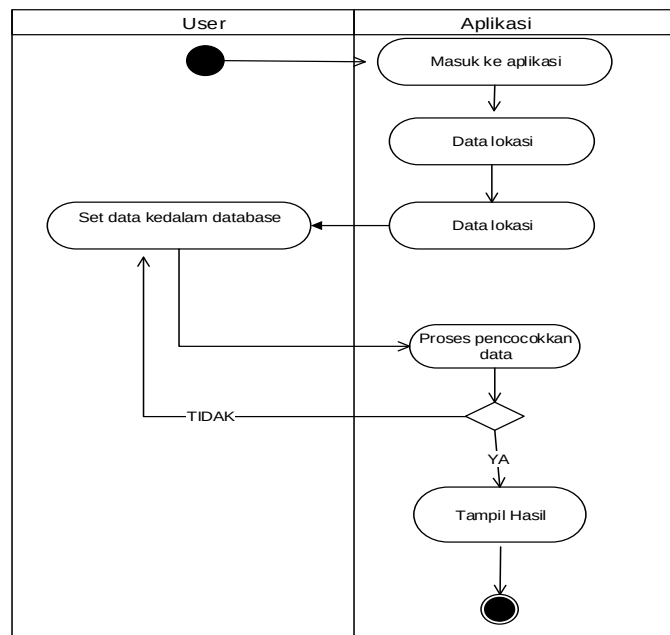
Gambar 4. 2 Activity Diagram user

b. Activity Diagram Input data Lokasi



Gambar 4. 3 Activity Diagram input data Lokasi

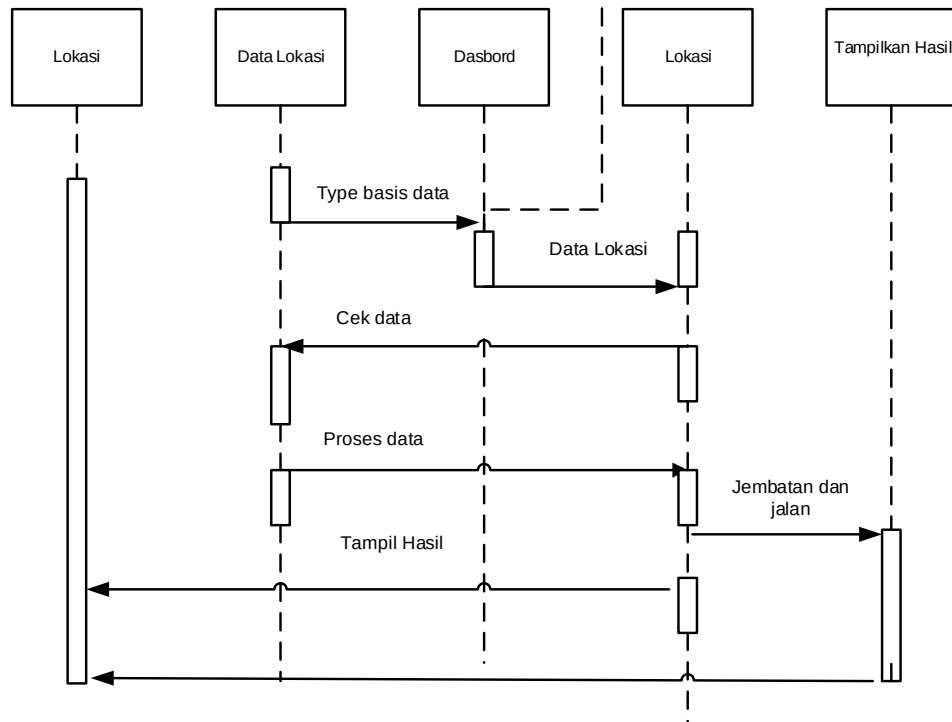
c. Activity Diagram data Lokasi



Gambar 4. 4 Activity Diagram data lokasi

2. Sequence Diagram

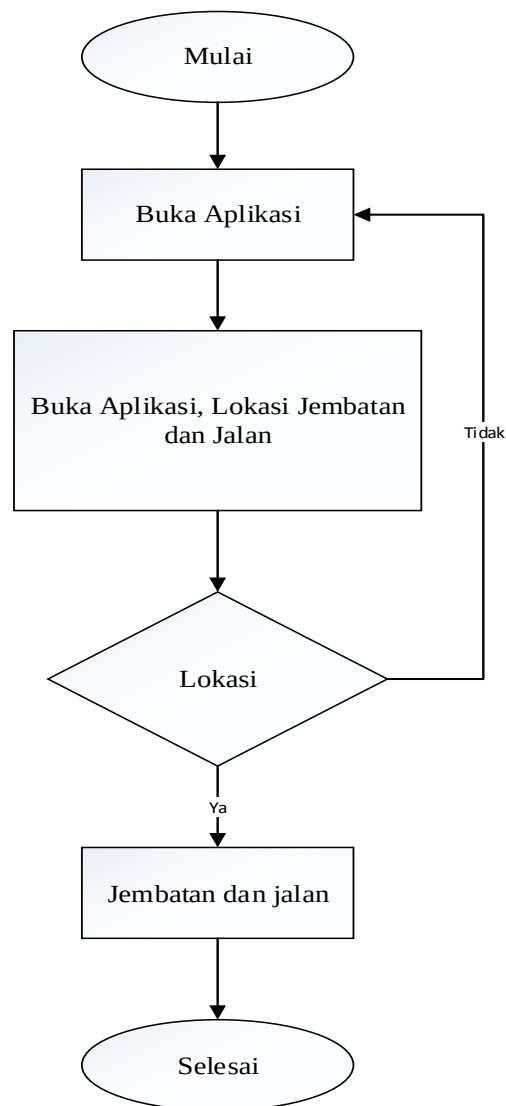
a. Sequence Diagram Aplikasi



Gambar 4. 5 Sequence Diagram

3. Rancangan Flowchart

Flowchart merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam alur penelitian. Pada *flowchart* dibawah ini akan menjelaskan mengenai cara aplikasi.



Gambar 4. 6 *Flowchart*

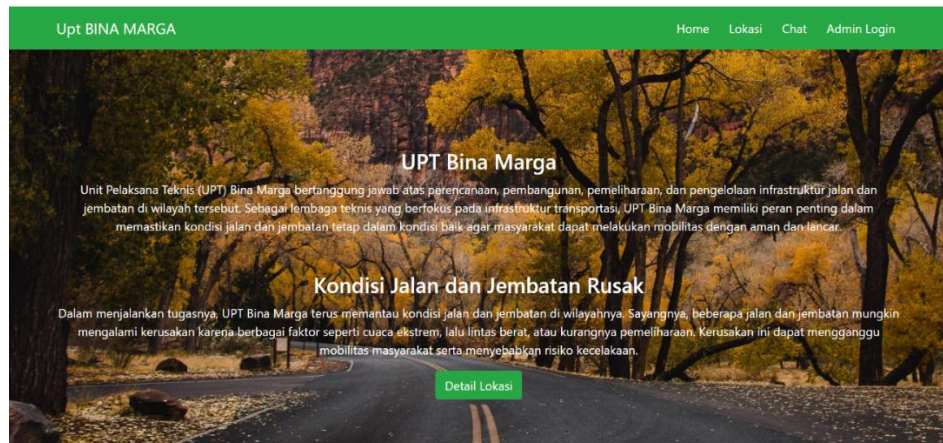
C. Implementasi

Program ini adalah aplikasi yang mempunyai fungsi untuk mempermudah seseorang untuk memperoleh data diri dari hasil aplikasi.

Implementasi aplikasi ini dilakukan setelah proses input data lokasi jembatan dan jalan diaplikasi selesai dilakukan. Untuk menjalankan aplikasi ini maka dibutuhkan sebuah perangkat *smartphone* atau *android*.

1. Tampilan menu utama

Ini adalah tampilan menu utama pada aplikasi sebagai berikut;



Gambar 4. 7 Tampilan Menu Utama

- Tombol detail lokasi, untuk melihat data jembatan dan jalan.
- Tombol *Admin login*, masuk untuk melakukan pengisian data Lokasi jembatan dan jalan yang sudah didata.

2. Tampilan Detail lokasi

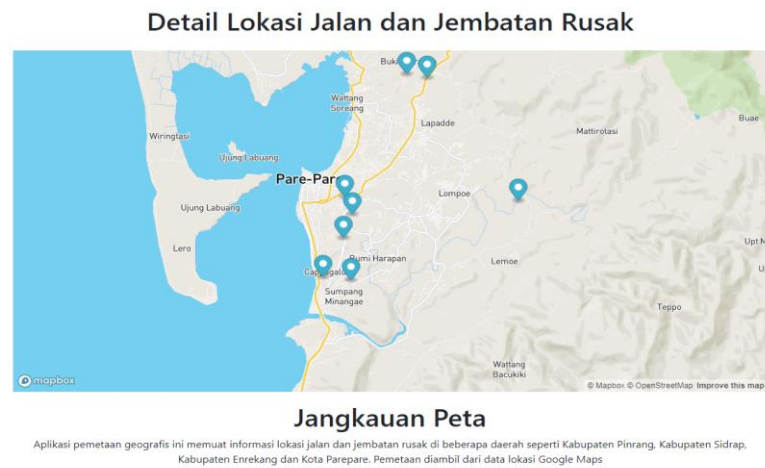
Setelah menekan Tombol detail lokasi dibawah maka *user* akan diarahkan ke tampilan lokasi jembatan dan jalan.

Data Lokasi					
No	Nama	Latitude	Longitude	Deskripsi	Gambar
1	Jl. Poros Pakajene-Anabanua (Sidrap)	-3.907106	119.975082	Jalan berlubang	
2	Jl. Anggrek (Sidrap)	-3.994155	119.808174	Jalan berbatu	
3	Jl. Poros Sidrap-Soppeng (Sidrap)	-3.994787	119.813171	Jalan berlubang dan berbatu	
4	Jl. Pabbaesseng (Sidrap)	-3.957528	119.680634	Jalan berlubang	
5	Jl. Jenderal Sudirman, Parepare	-4.023074	119.633095	Jalan berbatu	
6	Jl. Jend. Ahmad Yani, Parepare	-4.010817	119.633316	Jalan berlubang	
7	Jl. Persatuan, Parepare	-4.016136	119.635468	Jalan berbatu dan tidak rata	
8	Jl. Lambo, Parepare	-4.034866	119.627373	Jalan berbatu dan tidak rata	
9	Jl. Poros Pinrang-Rappang, Sidrap	-3.834866	119.778450	Jalan berlubang	
10	Jl. Jend. Sudirman (Dekat pasar Lawawoi, Sidrap)	-3.903788	119.735550	Jalan berbatu dan berlubang	
11	Jl. Laude, Pinrang	-3.816184	119.724113	Jalan berbatu	

Gambar 4. 8 Tampilan data Lokasi

3. Tampilan halaman lokasi

Setelah *user* memilih menu lokasi maka tampil lokasi tersebut didalam aplikasi.



Gambar 4. 9 Tampilan data Lokasi

4. Tampilan halaman detail jembatan dan jalan

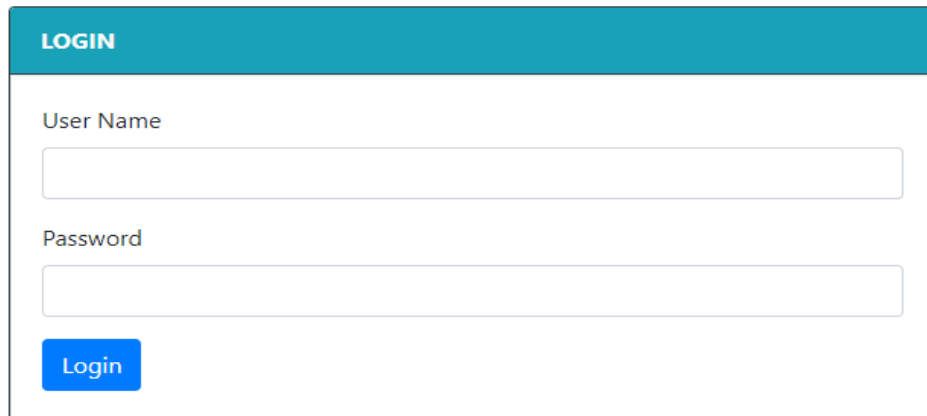
Setelah *user* memilih menu lokasi maka tampil lokasi tersebut didalam aplikasi.



Gambar 4. 10 Tampilan detail jembatan dan jalan

5. Tampilan *Login Admin*

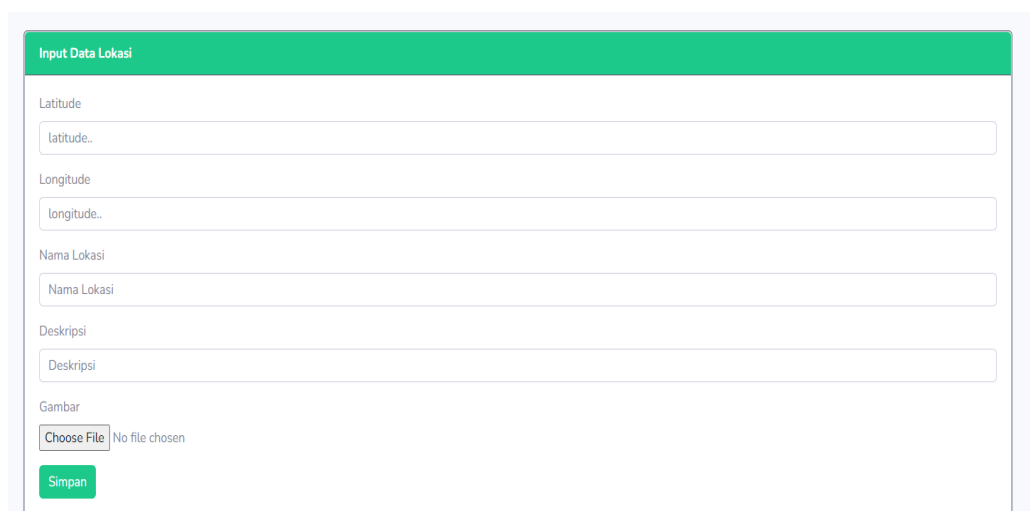
Tampilan pada Gambar dibawah *login admin* sedang melakukan input data lokasi.

A screenshot of a web form titled "LOGIN" in a teal header. Below the header, there are two input fields: "User Name" and "Password". Below the "Password" field is a blue "Login" button.

Gambar 4. 11 Tampilan *login admin*

6. Tampilan Tambah Lokasi jembatan dan jalan

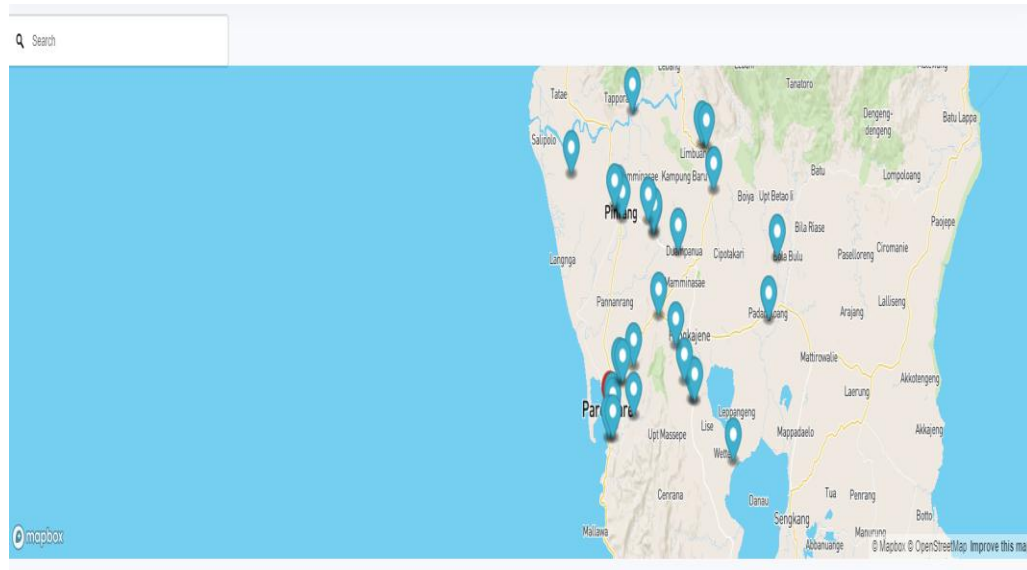
Tampilan pada Gambar dibawah Admin sedang melakukan input data Lokasi jembatan dan jalan.

A screenshot of a web form titled "Input Data Lokasi" in a green header. Below the header, there are five input fields: "Latitude", "Longitude", "Nama Lokasi", "Deskripsi", and "Gambar". The "Gambar" field has a "Choose File" button and the text "No file chosen". Below the "Gambar" field is a green "Simpan" button.

Gambar 4. 12 Tampilan tambah Lokasi

7. Tampilan data lokasi

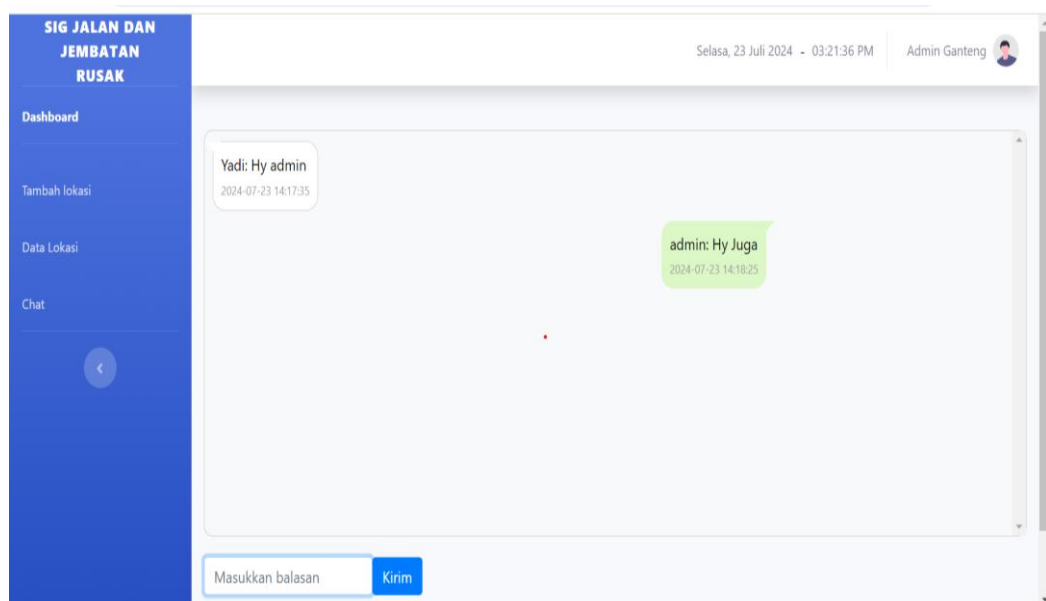
Tampilan pada Gambar dibawah Admin sedang menampilkan data lokasi.



Gambar 4. 13 Tampilan data lokasi

8. Tampilan Chat

Tampilan pada gambar dibawah *user* sedang chat dengan admin



Gambar 4. 14 Tampilan Chatting

9. Tampilan Data Laporan

Tampilan pada gambar dibawah admin sedang menampilkan data laporan

The image shows two screenshots of a web application interface. The top screenshot displays a table of reported road and bridge issues, while the bottom screenshot shows the form for adding new data.

Top Screenshot: Data Report Table

At the top, there is a search bar labeled "Keterangan" and a "Tambah Data" button. Below this is a date selection field "Pilih Tanggal" with a calendar icon and a "Filter" button. A "Download Laporan" button is also present.

NO	TANGGAL	MASALAH	SOLUSI	GAMBAR	KET	AKSI
1	2024-07-25	Jl. Poros Pangkajene	harus segera diperbaiki		Sudah diperbaiki	 Hapus
2	2024-03-29	jl. jend. Sudirman, Sidrap	harus segera diperbaiki		Sudah diperbaiki	 Hapus
3	2023-05-10	jl. poros sidrap-enrekang	harus segera diperbaiki		Sudah diperbaiki	 Hapus

Bottom Screenshot: Add New Data Form

The bottom screenshot shows the "Data Masalah dan Solusi" form. It includes a sidebar with navigation links: Dashboard, Tambah lokasi, Data Lokasi, Chat, and Laporan. The form fields are:

- Tanggal: mm/dd/yyyy
- Masalah: [Text input field]
- Solusi: [Text input field]
- Gambar: Choose File (No file chosen)
- Keterangan: [Text input field]

The top right of the bottom screenshot shows the date "Rabu, 28 Agustus 2024" and the user "Admin Ganteng".

Gambar 4. 15 Tampilan Data Laporan

10. Tampilan Laporan

Tampilan pada gambar dibawah adalah data laporan setelah di *download*

 PEMERINTAH PROVINSI SULAWESI SELATAN DINAS BINA MARGA DAN BINA KONTRUKSI WILAYAH III Jl. Karaeng Burane No. 2 Ujung Bulu, Kec. Ujung Kota Parepare					
LAPORAN MASALAH					
Nama Pelapor	Tanggal	Masalah	Solusi	Keterangan	Gambar
A. Astita	2024-07-25	Jl. Poros Pangkajene	harus segera diperbaiki	Sudah diperbaiki	
Muzakkir	2024-03-29	Jl. jend. Sudirman, Sidrap	harus segera diperbaiki	Sudah diperbaiki	
Muh. Femy	2023-05-10	Jl. poros sidrap-enrekang	harus segera diperbaiki	Sudah diperbaiki	
Pebrianto	2024-03-22	Jl. poros Parepare-Sidrap	harus segera diperbaiki	Sudah diperbaiki	
Erlina	2024-04-10	Jl.poros pinrang rapping	harus segera diperbaiki	Sudah diperbaiki	
Yadi Rahmat	2024-07-06	Jembatan Carawali	harus segera diperbaiki	Sudah diperbaiki	

Gambar 4. 16 Tampilan Detail Data Laporan

11. Panduan Pelaporan

Tampilan pada Gambar dibawah Admin sedang menampilkan panduan pelaporan



Gambar 4. 17 Tampilan Panduan Pelaporan

12. Rincian Pengerjaan

Tampilan pada Gambar dibawah Admin sedang menampilkan rincian pengerjaan



PEMERINTAH PROVINSI SULAWESI SELATAN
DINAS BINA MARGA DAN BINA KONTRUKSI WILAYAH III
 Jl. Karaeng Burane No. 2 Ujung Bulu, Kec. Ujung Kota Parepare

RINCIAN LAPORAN Pengerjaan

No	Nama Jalan	Dana	Waktu	Yang Bertanggung Jawab
1	Jl. Poros Pangkajene	APBN sebesar +-2 m	13/08/2022 – 25/07/2024	Dinas PU Kab. Sidrap
2	jl. jend. Sudirman, Sidrap	APBN sebesar +-1 m	05/12/2023 – 29/03/2024	Dinas PU Kab. Sidrap
3	jl. poros sidrap-enrekang	APBN sebesar +- 1 m	26/10/2022 – 10/05/2023	Dinas PU Kab. Enrekang
4	Jl. poros Parepare-Sidrap	APBN sebesar +-3 m	20/04/2022 – 22/03/2024	Dinas PU Wilayah II Kota Parepare
5	Jl.poros pinrang-rappang	APBN sebesar +-5 m	14/06/2020 – 10/04/2024	Dinas PU Kab. Pinrang
6	Jembatan Carawali	APBD sebesar +- 500 juta	19/03/2024 – 06/07/2024	Dinas PU Kab. Sidrap

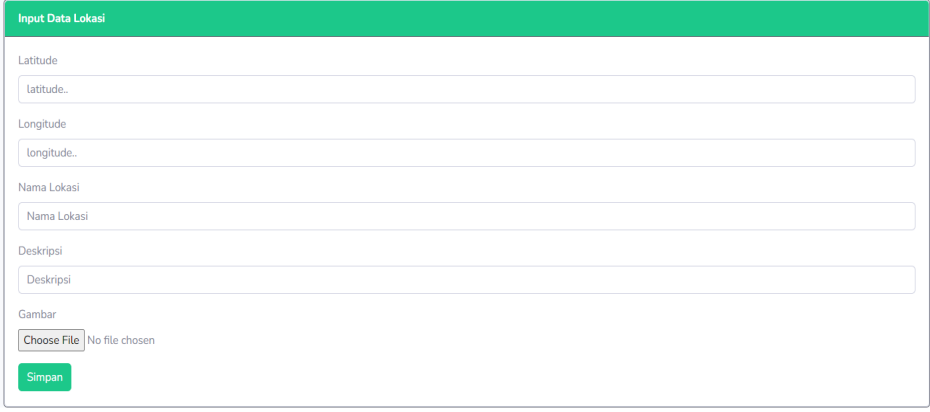
Gambar 4. 18 Tampilan Rincian Pengerjaan

D. PENGUJIAN

Pengujian Sistem dengan Pengujian *Black box*. Pengujian ini bersifat menguji waktu perintah sesuai dengan fungsinya yang tanpa ada kesalahan atau error.

1. *Black box* melakukan *input* data lokasi

Tabel 4. 2 *Black box* Melakukan proses *input* data lokasi

<i>Test Factor</i>	<i>Hasil</i>	<i>Kesimpulan</i>
Jika tombol tambah data Lokasi di tekan akan menampilkan proses pengisian tambah data lokasi	✓	Berhasil karena proses data lokasi dapat di tampilkan data detail pada sistem.
<i>Screenshot</i>		
		

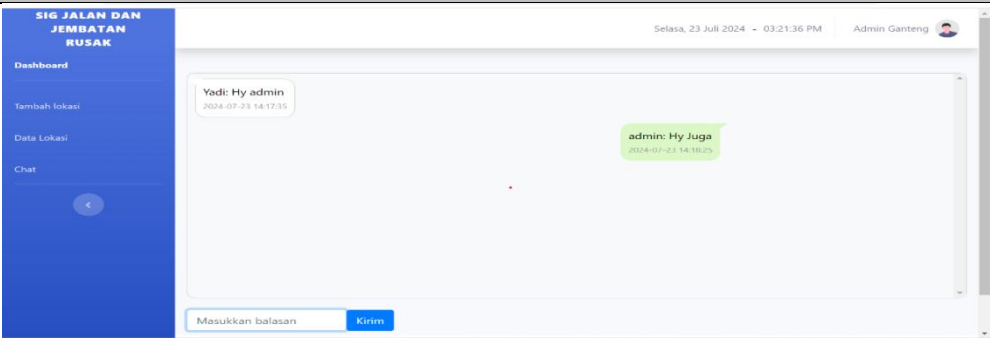
2. *Black box* login admin.

Tabel 4. 3 *Black box* Melakukan proses *login* admin

<i>Test Factor</i>	<i>Hasil</i>	<i>Kesimpulan</i>
Jika proses mengisi <i>user</i> dan <i>password</i> dilakukan maka <i>system</i> akan menampilkan data admin.	✓	Berhasil, karena data admin dapat dikenali oleh <i>system</i> secara otomatis
<i>Screenshot</i>		
		

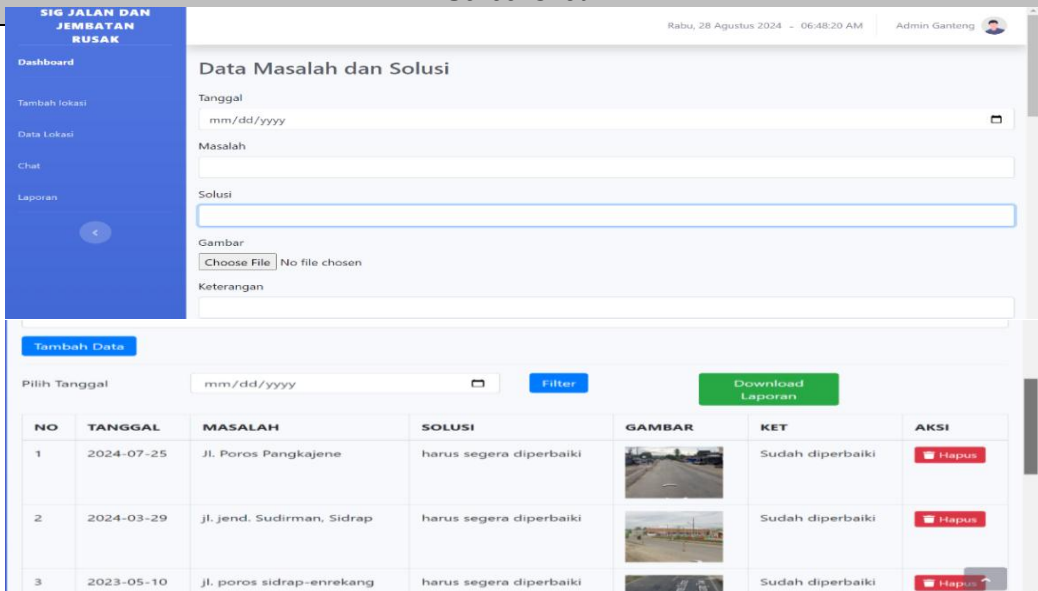
3. Black box Chat

Tabel 4. 4 *Black box* Melakukan proses *chatting*

Test Factor	Hasil	Kesimpulan
Jika tombol Chat di tekan akan menampilkan ruang obrolan	✓	Berhasil karena proses chatting dapat di tampilkan
Screenshot		
		

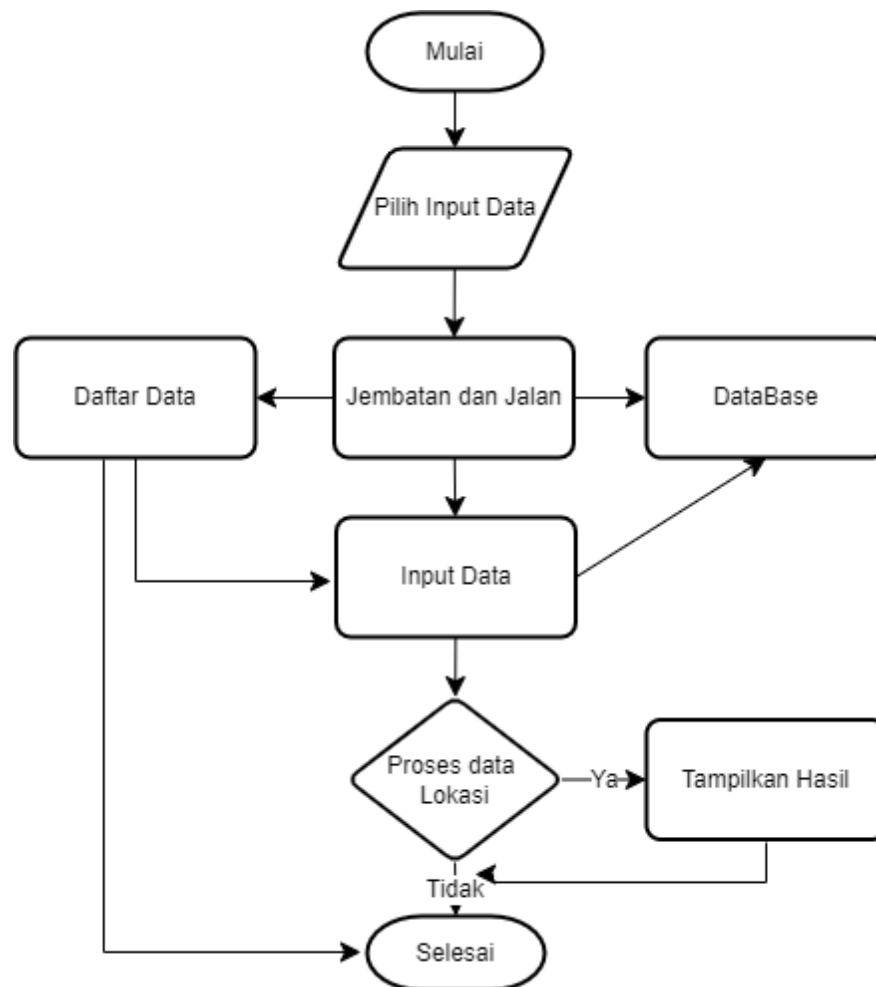
4. Black box Laporan

Tabel 4. 5 *Black box* Menampilkan data laporan penanganan

Test Factor	Hasil	Kesimpulan
Jika fitur laporan di tekan akan menampilkan laporan	✓	Berhasil karena data laporan penanganan dapat di tampilkan
Screenshot		
		

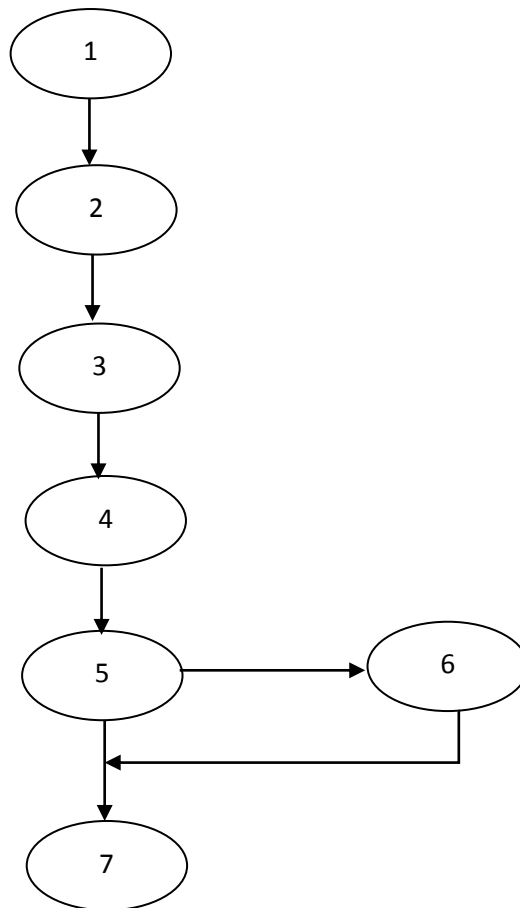
1. Pengujian *White box*

a. *Flowchart* Aplikasi



Gambar 4. 16 *Flowchart* Aplikasi

b. *Flow Graph Aplikasi*



Gambar 4. 17 *FlowGraph Aplikasi*

1) Proses Perhitungan

Dari Gambar *FlowGraph* di atas dapat di lakukan proses perhitungan sebagai berikut :

a. Menghitung *Cyclomatic Complexcity* $V(G) = E - N + 2$

$$N(\text{node}) = 7$$

$$E(\text{edge}) = 7$$

$$P(\text{predikat node}) = 1$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 7 - 7 + 2$$

$$= 2$$

$$\text{Predikat Node (N)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

b. Berdasarkan perhitungan *Cyclomatic Complexity* dari *FlowGraph* diatas memiliki *Region* = 2

c. *Independent Independent Path* pada *Flow Graph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

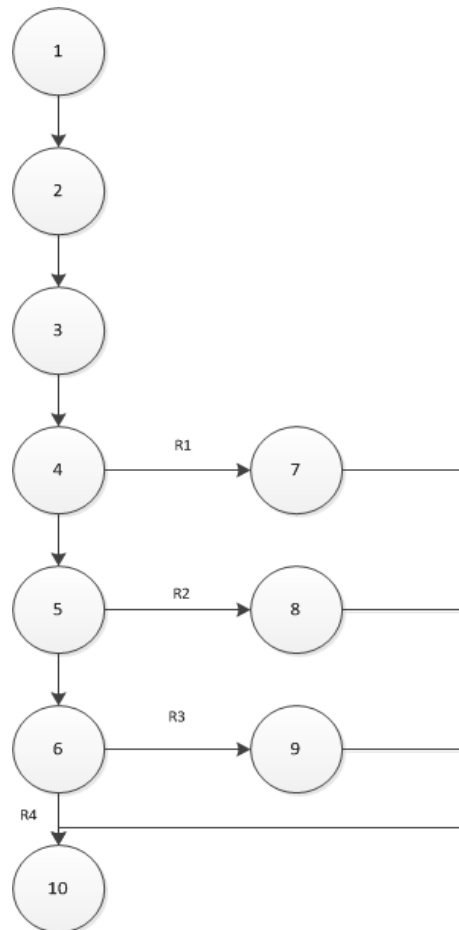
$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 7$$

d. *Grafik Matriks*

Tabel 4. 6 *Grafik Matriks*

	1	2	3	4	5	6	7	E - 1
1		1						1-1 = 0
2			1					1-1 = 0
3				1				1-1 = 0
4					1			1-1 = 0
5						1	1	2-1 = 1
6							1	1-1 = 1
7								
Zum (E+1)								1+1 = 2

1) *Flow Graph* menjalankan aplikasi



Gambar 4. 18 *Flowgraph* menjalankan aplikasi

2) Proses perhitungan menjalankan aplikasi

Dari gambar *FlowGraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N(node) = 10$

$E(edge) = 12$

$P(predikat\ node) = 3$

Penyelesaian : $V(G) = E - N + 2$

$$= 12 - 10 + 2$$

$$= 4$$

$$\text{Predikat Node (N)} = P + 1$$

$$= 3 + 1$$

$$= 4$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *FlowGraph* diatas memiliki *Region* = 4

c) *Independent Path* pada *FlowGraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 4 – 7 – 10

Path 2 = 1 – 2 – 3 – 4 – 5 – 8 – 10

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10

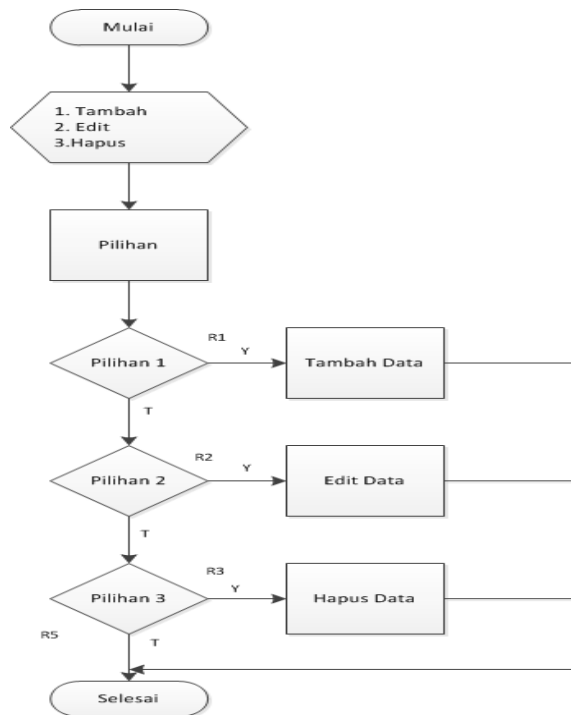
Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 10

d) *Grafik Matriks* Menjalankan Aplikasi

Tabel 4. 7 *Grafik Matriks* menjalankan aplikasi

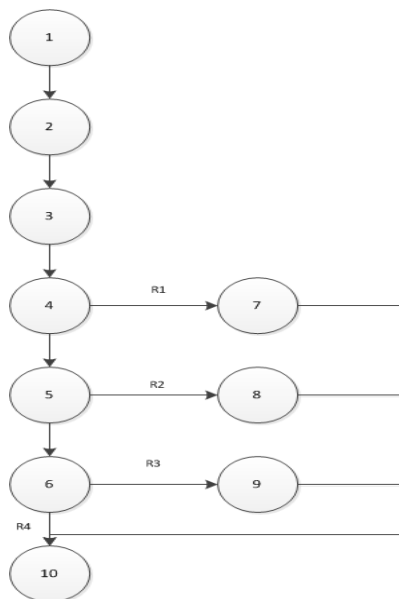
[illegible]

1. Flowchart input data



Gambar 4. 19 Flowchart input data

1) Flow Graph input data



Gambar 4. 20 Flow Graph input data

2) Proses perhitungan android data

Dari gambar *FlowGraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$$N(\text{node}) = 10$$

$$E(\text{edge}) = 12$$

$$P(\text{predikat node}) = 3$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 12 - 10 + 2 = 4$$

$$\text{Predikat Node } (N) = P + 1$$

$$= 3 + 1$$

$$= 4$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *FlowGraph*

diatas memiliki *Region* = 4

c) *Independent Path* pada *FlowGraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 7 - 10$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 8 - 10$$

$$\text{Path 3} = 1 - 2 - 3 - 4 - 5 - 6 - 9 - 10$$

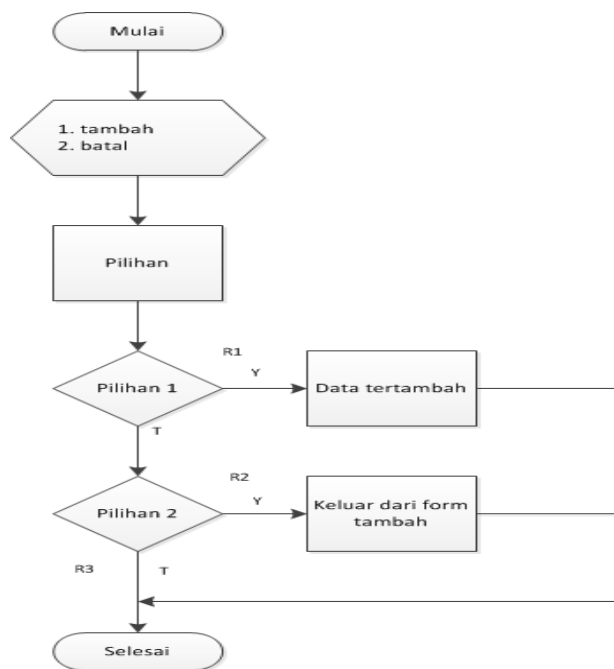
$$\text{Path 4} = 1 - 2 - 3 - 4 - 5 - 6 - 10$$

d) *Grafik Matriks*

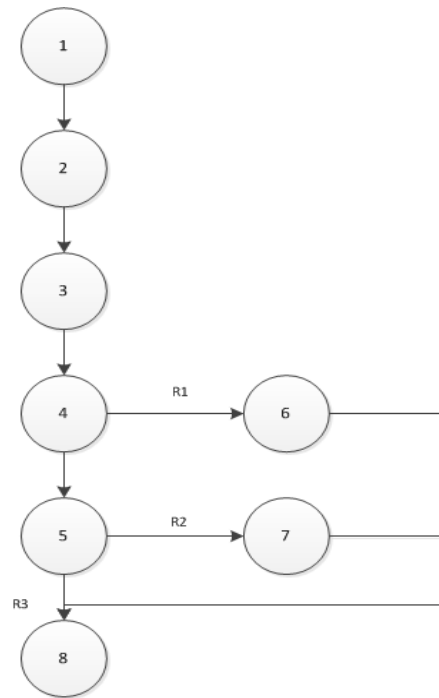
Tabel 4. 8 *Grafik Matriks*

	1	2	3	4	5	6	7	8	9	10	E-1
1		1									1-1=0
2			1								1-1=0
3				1							1-1=0
4					1		1				2-1=1
5						1		1			2-1=1
6									1	1	2-1=1
7										1	1-1=0
8										1	1-1=0
9										1	1-1=0
10											0
SUM(E+1)											3+1=4

a. *White box* berdasarkan *form* tambah

**Gambar 4. 21** *Form* tambah data

1) *FlowGraph form* tambah data



Gambar 4. 22 *Flowgraph form* tambah

2) Proses perhitungan *form* tambah

Dari gambar *FlowGraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexcity* $V(G)$ dari *Egde* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$$N(\text{node}) = 8$$

$$E(\text{edge}) = 9$$

$$P(\text{predikat node}) = 2$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 9 - 8 + 2$$

$$= 3$$

Predikat Node (N) = P + 1

$$= 2 + 1$$

$$= 3$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *FlowGraph* diatas memiliki *Region* = 3

c) *Independent path* pada *FlowGraph* diatas adalah :

$$Path\ 1 = 1 - 2 - 3 - 4 - 6 - 8$$

$$Path\ 2 = 1 - 2 - 3 - 4 - 5 - 7 - 8$$

Path 3 = 1 – 2 – 3 – 4 – 5 – 8

d) *Grafik Matriks Form* tambah

Tabel 4. 9 *matriks form* tambah

[illegible]

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil penelitian dan pembahasan aplikasi web gis jalan dan jembatan rusak pada UPT bina marga seajataappareng, maka dapat diambil kesimpulan antara lain :

- 1) Telah dihasilkan sebuah program pelayanan website yaitu Aplikasi Web Gis Jalan Dan Jembatan Pada UPTD Bina Marga Wilayah ajatappareng menggunakan perangkat lunak *PHP* dan *MySQL* sehingga dapat bekerja dalam jaringan (*online*) yang dapat membantu pegawai kantor dalam penyampaian informasi koordinat awal lokasi jalan dan jembatan provinsi.
- 2) Dalam pengujian ini sistem ini dilakukan dengan menggunakan 2 metode yakni *White Box* dan *Black Box*. Metode pengujian *White Box* adalah pengujian menggunakan algoritma program dan hasil yang diperoleh adalah jumlah path 18. Sedangkan *Black box* adalah merupakan pengujian *listing* program antara lain : *form login*, *form master daerah*, *form master admin user daerah*, *form master admin*, *form info daerah*, *form input data jalan dan jembatan*, dan *form peta*.

B. Saran

Berdasarkan kesimpulan dari hasil penelitian ini, dapat disarankan hal-hal berikut ini :

Diharapkan kedepannya bisa lebih baik dan memperbaiki segala kekurangan yang ada pada aplikasi ini dengan mengembangkan aplikasi menggunakan *Sistem Android*

DAFTAR PUSTAKA

- Andika Muhamad Rafi, 2020, UIN Sunan Gunung Djati Bandung, Aplikasi Jalan Cantik, Memperbaiki Jalan dan Jembatan Menjadi “Cantik” Kembali
- Jogiyanto Hartono, 2019, Aplikasi (*application*) *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft *Word*, dan *Microsoft Excel*
- Jogiyanto, Hartono. 2019. *Pengenalan Komputer*. Yogyakarta: C.V.ANDI OFFSET.
- Kadir, Abdul. 2019. *Dasar Pemrograman Web dinamis menggunakan PHP*. Yogyakarta : Andi.
- Mustika Anindita, 2021, Universitas Telkom, Aplikasi Pengelolaan Data Kerusakan Jalan Dan Jembatan Berbasis Web Dan Android Menggunakan Sistem Monitoring Google Maps Api (Studi Kasus : Dinas Pekerjaan Umum Kabupaten Magetan)
- Nida Nafsul Muthmainnah, 2021, Jurusan Teknik Informatika Fakultas Teknik Universitas Sriwijaya, Analisis Perbandingan Kondisi Jembatan Menggunakan Program Bridge Management System (Bms) Dan Aplikasi Inspeksi Visual Jembatan (Invi – J) (Studi Kasus: Jembatan Pada Jalan Mayjen Yusuf Singadekane, Palembang)
- Pandapotan Sianipar. 2022. *Panduan Menggunakan Internet*. Jakarta: PT Elex Media Komputindo.
- Peranginangin, Kasiman. 2019. *Aplikasi WEB dengan PHP dan MySQL*. Yogyakarta : Andi Offset.
- Prasetio, Adhi. 2020. *Buku Pintar Pemrograman WEB*. Bandung : Mediakita.
- Raharjo, Budi. 2020. *Belajar Pemrograman Web*. Bandung: MODULA.
- Riyanto. 2022. *Pengembangan Aplikasi Sistem Informasi Geografis*. Yogyakarta: Gava Media.
- Muhammad Andrian, 2019, Jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Parepare, Aplikasi Web Jalan dan Jembatan Rusak Pada UPT BINA MARGA Kabupaten Pinrang (Studi Kasus: Dinas Pekerjaan Umum Kabupaten Pinrang).