

BAB II
KAJIAN PUSTAKA

A. Studi Pustaka

Sebagai acuan dalam perancangan sistem Pengaduan Kawasan Kumuh Berbasis Web GIS, maka penulis mencari referensi terhadap beberapa penelitian sejenis yang berkaitan dengan penelitian ini. Berikut ini ada beberapa penelitian yang dilakukan oleh peneliti terdahulu antara lain :

Tabel 2. 1 Perbandingan Beberapa Penelitian

No	Peneliti	Judul	Metode dan Software	Tujuan dan Hasil Penelitian
1.	Dery Satya Pramdhana (2019)	Perancangan Dan Implementasi Sistem Informasi Geospasial Menggunakan Aplikasi Geoserver	<i>Copsmet95 dan TKME (TK Metadata Editor)</i>	Tujuan penelitiannya adalah membangun aplikasi untuk mendukung pemetaan online (Web-Mapping). Hasil dari sistem ini adalah aplikasi yang dapat digunakan dalam sistem informasi geospasial dan dapat memudahkan dalam proses pencarian data geospasial. Melalui sistem informasi geospasial berbasis Open Source diharapkan dapat memudahkan pencarian data geospasial secara cepat dan mudah.
2.	Tiara Surtikanti (2018)	Sistem Pelaporan Dan Monitoring Kerusakan Jalan Berbasis Web Menggunakan Model	- <i>Model View Controller (MVC)</i> - <i>PHP</i> - <i>Database MySql</i>	Tujuan penelitiannya adalah untuk merancang dan membangun sistem pelaporan kerusakan jalan berbasis web menggunakan MVC. Penelitian ini menghasilkan sistem yang dapat menampilkan informasi

		View Controller (MVC)		dari pengguna jalan mengenai infrastuktur jalan yang mengalami kerusakan dan dapat dijadikan acuan dalam penanganan kerusakan jalan dengan tepat.
3.	Helmi Bahar Alim (2016)	Sistem Informasi Geografis Jalan dan Jembatan di Kabupaten Wonogiri	- <i>SDLC</i> - <i>UML</i> - <i>PostGis database version 2.2</i> - <i>LeafletJS 1.0</i>)	Tujuan penelitiannya adalah membuat Sistem Informasi Geografis yang dapat memberikan informasi tentang kondisi jalan dan jembatan di Kabupaten Wonogiri Hasil dari penelitian ini adalah sistem informasi yang dapat menyajikan segala informasi mengenai jalan dan jembatan melalui suatu peta jaringan jalan dalam sebuah aplikasi Sistem Informasi Geografis (SIG) mengenai sistem informasi geografis jaringan jalan dan jembatan.

Pada penelitian ini, penulis bermaksud mengangkat judul “ Teknik Visualisasi Data Spasial Pada Sistem Informasi Pengaduan Kawasan Kumuh Berbasis Web GIS (Di Program KOTAKU Cilacap - Ditjen Cipta Karya Kementerian PUPR) “ dengan tujuan untuk membangun sistem informasi pengaduan kawasan kumuh berbasis web GIS dengan teknik visualisasi data spasial. Sistem ini dirancang dengan menggunakan metode teknik visualisasi data spasial, tahap pengembangan sistem model *waterfall* dan bahasa pemrograman PHP. Dalam membuat sistem ini penulis menggunakan *framework CodeIgniter (Ci)* dan *Object Oriented Analysis and Design (OOAD)*. *Interface* penghubung beberapa aplikasi menggunakan *Google Application Programming Interface (API)* dan database MySQL. Hasil dari penelitian ini adalah sistem diharapkan sebagai fitur untuk menyampaikan aspirasi masyarakat dalam

pengaduan kawasan kumuh (kondisi bangunan gedung, kondisi jalan lingkungan, kondisi penyediaan air minum, kondisi drainase lingkungan, kondisi pengelolaan air limbah, kondisi pengelolaan persampahan, kondisi pengamanan proteksi kebakaran) dan menyajikan informasi dengan efektif, cepat dan lebih mudah.

Berdasarkan uraian beberapa penelitian diatas, maka dapat disimpulkan bahwa perbedaan penelitian yang dilakukan oleh penulis dengan penelitian lainnya yaitu tidak hanya pengaduan kerusakan jalan tetapi juga permasalahan kondisi bangunan gedung, kondisi penyediaan air minum, kondisi drainase lingkungan, kondisi pengelolaan air limbah, kondisi pengelolaan persampahan, dan kondisi pengamanan proteksi kebakaran. Masyarakat juga dapat melihat progres data pengaduan. Penulis juga menggunakan *Google Maps Application Programming Interface (API)* untuk memvisualisasikan data spasial.

B. Landasan Teori

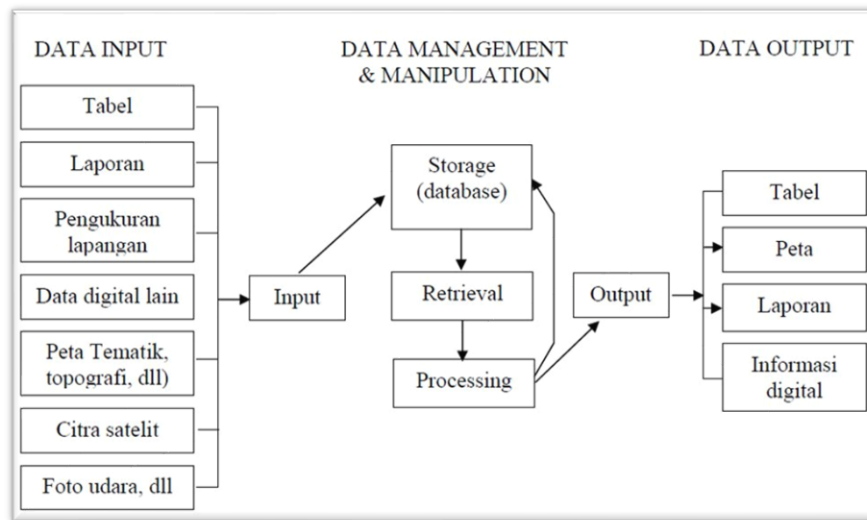
2.1 Sistem Informasi Geografis (SIG)

2.1.1 Definisi Sistem Informasi Geografis (SIG)

Menurut Ahaliki 2018, Sistem Informasi Geografis (SIG) atau sering juga disebut dengan Sistem Informasi Geospasial merupakan suatu sistem informasi yang digunakan untuk menyusun, menyimpan, merevisi dan menganalisa data dan atribut yang bereferensi kepada lokasi atau posisi obyek-obyek di bumi.

SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, meng-update, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografis. Dari beberapa definisi diatas, dapat disimpulkan bahwa SIG merupakan sistem informasi yang didasarkan pada kerja komputer yang untuk dikelola, memanipulasi dan mengidentifikasi data yang berhubungan dengan permukaan bumi untuk dianalisis dan menampilkan data dalam suatu informasi berbasis geografis (Ahaliki, 2018).

Berikut ini merupakan gambaran dari cara kerja Sistem Informasi Geografis (SIG):



Gambar 2. 1 Cara Kerja Sistem Informasi Geografis

2.1.2 Model Data SIG

2.1.2.1 Model Data Spasial

Data spasial (informasi lokasi) merupakan sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya. Data spasial berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) maupun koordinat Cartesian XYZ (absis, koordinat dan ketinggian), termasuk diantaranya referensi permukaan dan sistem proyeksi (Hanita, 2014). Dalam SIG, data spasial terbagi dalam dua format, yaitu:

- a. Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (merupakan titik perpotongan antara dua buah garis)
- b. Data raster (atau disebut juga dengan sel grid) yaitu untuk menampilkan, menempatkan, dan menyimpan konten data spasial dengan menggunakan struktur matriks atau susunan piksel-piksel yang membentuk suatu grid. Entitas-entitas spasial model raster juga dapat disimpan dalam sejumlah layer yang secara fungsionalitas direlasikan dengan unsur-unsur petanya. Sebagai ilustrasi, beberapa sumber entitas spasial raster adalah citra satelit.

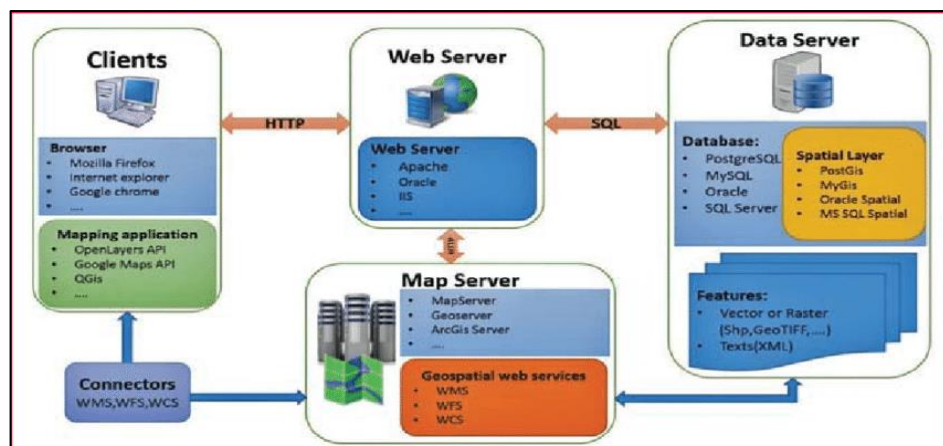
2.1.2.2 Model Data Non Spasial

Data non spasial atau data atribut memberikan gambaran atau menjelaskan informasi berkaitan dengan fitur peta atau coverage SIG. Data atribut dapat disimpan dalam format angka (numbers) maupun karakter (Hanita, 2014).

2.1.3 SIG Berbasis Web (webGIS)

Menurut Akbar, webGIS merupakan kombinasi SIG dan internet yang mengelola serta menampilkan proses informasi geografis dan data spasial. WebGIS merupakan sebuah web pemetaan atau mapping dengan menggunakan internet, webGIS tidak berarti hanya menampilkan peta yang berupa gambar statis tetapi juga memiliki kemampuan dalam menampilkan informasi data atribut (Akbar, 2015).

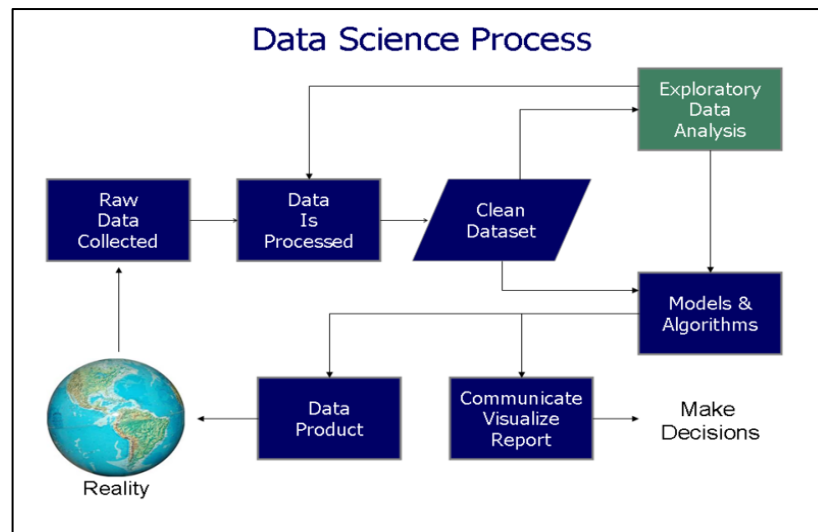
WebGIS memiliki dua layanan menampilkan informasi geospasial yaitu *Web Map Service (WMS)* dan *Web Feature Service (WFS)*. WMS adalah standar dari *Open Geospatial Consortium (OGC)* yang digunakan untuk pengiriman dan penerimaan data gambar geospasial melalui protokol HTTP. Gambar ini dapat bersumber dari data vektor dan data raster. WMS menyediakan sebuah standar interface untuk pemanggilan gambar geospasial. Manfaat utama dari WMS adalah bahwa klien dapat meminta gambar dari beberapa server, dan kemudian menggabungkannya dalam satu tampilan. (Akbar, 2015). Berikut ini adalah ilustrasi alur perancangan webGIS :



Gambar 2. 2 Ilustrasi Alur Perancangan WebGIS

2.2 Visualisasi Data

Visualisasi data adalah sebuah metode untuk mengkomunikasikan sebuah informasi dengan berbagai cara baik itu dalam bentuk gambar, diagram atau animasi dan visualisasi digunakan untuk menampilkan data dalam jumlah yang besar kemudian dipresentasikan dengan berbagai model. Visualisasi data merupakan menyajikan data secara visual yang berinteraksi langsung dengan pengguna untuk melakukan eksplorasi dan memperoleh informasi yang terdapat dalam data (Akbar, 2015). Tujuan dari visualisasi data adalah untuk menghitung atau menyampaikan informasi baik itu dalam bentuk 1 dimensi (dokumen teks), 2 dimensi (data bidang, peta geografis atau data peta), 3 dimensi (objek nyata seperti bentuk bangunan, dsb) dan temporal data yaitu untuk menampilkan data dalam periode tertentu (Sri Mulyana dan Edi Winarko, 2011). Berikut adalah tampilan *data visualization process* atau proses visualisasi data :

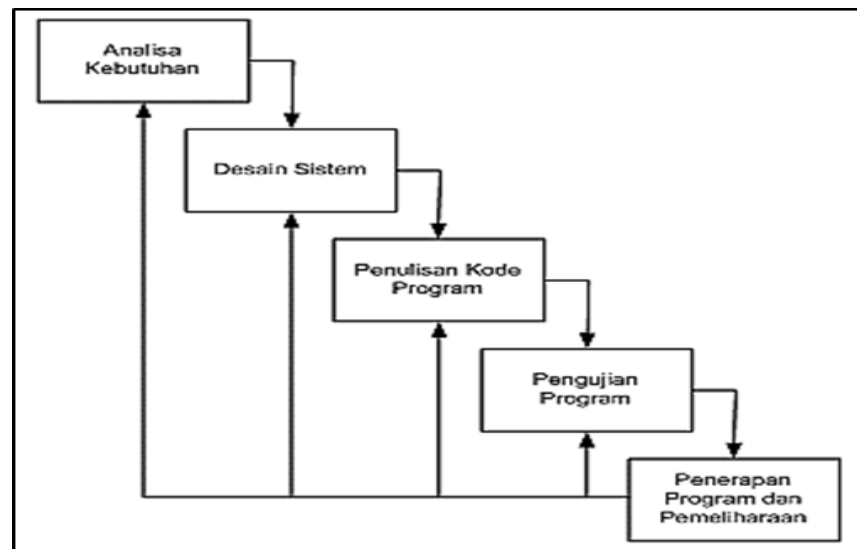


Gambar 2. 3 Proses Visualisasi Data

2.3 Waterfall model

Prosedur selalu mendefinisikan kontrol yang membantu keyakinan kualitas dan perubahan koordinasi, dan mendefinisikan keluaran (berupa dokumen, laporan dan formulir yang dibutuhkan). Proses rekayasa perangkat lunak secara linier biasanya disebut dengan model air terjun (*waterfall model*). *Waterfall model* mengusulkan

sebuah pendekatan yang sistematis dan sekuensial. Tahapan tersebut proses tersebut adalah sebagai berikut (Yuniarti, 2015) :



Gambar 2. 4 Pemodelan Waterfall Menurut Sommerville

Penjelasan tahapan-tahapan dari metode waterfall :

1. Analisa Kebutuhan (Analysis)

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data pada tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. Desain Sistem (Design)

Proses desain sistem akan menterjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen inilah yang akan digunakan programmer untuk melakukan aktifitas pembuatan sistemnya.

3. Penulisan Kode Program (Coding)

Coding merupakan penerjemah design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. Pengujian Program (Testing)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

5. Penerapan Program dan Pemeliharaan (Maintenance)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau dengan sistem operasi yang baru) atau karena pelanggan membutuhkan perkembangan fungsional.

2.4 Google Maps API (Application Programming Interface)

2.4.1 Definisi Google Maps

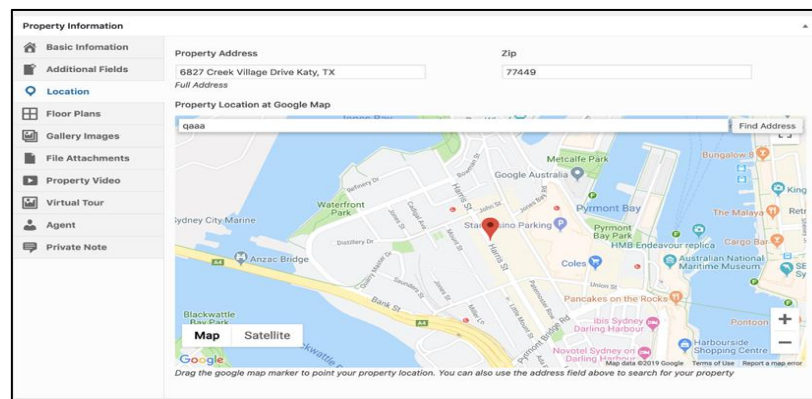
Google maps adalah sebuah peta virtual yang disediakan oleh google dan bisa di akses secara gratis di <https://www.google.com/maps>. Google maps menampilkan gambar peta yang diambil dari database pada web server yang dimiliki oleh google untuk menampilkan gambar yang diminta. Pada google maps , pengguna internet dapat mencari informasi grafis seperti berikut :

a. Satelit Maps

Pengguna dapat menikmati gambar permukaan bumi dari foto satelit. Pengguna dapat melihat foto satelit lebih detail dengan cara zoom pada daerah yang diinginkan.

- b. Hasil Pencarian integritas
Mencari lokasi, bisnis, peta buatan pengguna, dan *real estate*
- c. *Draggable Maps* (Peta Topografi)
Peta digital *mapping* yang *draggable* (bisa digeser) dengan menggunakan bantuan mouse.
- d. *Terrain Maps* (Peta Topografi)
Terrain Maps menyediakan informasi fitur peta fisik atau peta topografi yang biasa digunakan pada atlas.
- e. *Earth Maps*
Pengguna dapat melihat peta bumi dimana bumiterlihat secara utuh dan jika *di zoom* pengguna akan melihat permukaan bumi yang ditutupi dengan awan beserta pulau dan laut yang tampak lebih nyata dari ketinggian.
- f. *My Location*
Meggunakan fitur ini pengguna dapat mengetahui dimana letak lokasipengguna tersebut berada.

Berikut ini adalah tampilan mengenai Google Maps API (Application Programming Interface) :



Gambar 2. 5 Google Maps API (Application Programming Interface)

2.4.2 Definisi *Application Programming Interface*

Menurut Akbar, *Application Programming Interface (API)* atau antarmuka pemrograman aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem tertentu. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi

lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol (HTTP)* dan mendapatkan respon berupa *Extensible Markup Language (XML)* atau *JavaScript Object Notation (JSON)*. Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda. Format pemanggilan fungsi ini diatur pada masing-masing penyedia API (Akbar, 2015).

2.4.3 Langkah-langkah menggunakan *Google Maps API*

Google Maps API adalah suatu library yang berbentuk *JavaScript*. Cara membuat google maps untuk ditampilkan pada situs web / blog sangat mudah, hanya dengan membutuhkan pengetahuan tentang *HTML* serta *JavaScript*, serta koneksi internet yang stabil. Dengan menggunakan *google maps API*, kita dapat menghemat waktu dan membangun aplikasi peta digital yang handal, sehingga kita bisa fokus hanya pada data-data yang akan ditampilkan. Dalam pembuatan program *google maps API* menggunakan urutan sebagai berikut (Agustinus, 2017) :

1. Memasukkan *maps API JavaScript* kedalam *HTML*
2. Membuat element div dengan nama *maps_canvas* untuk menampilkan peta
3. Membuat beberapa objek literal untuk properti-properti pada peta
4. Menulis fungsi *JavaScript* untuk membuat objek peta
5. Meng-inisiasi peta dalam tag-body *HTML* dengan event *onload*

2.5 Pengaduan Masyarakat

2.5.1 Definisi Pengaduan Masyarakat

Pengaduan adalah pernyataan secara lisan atau tertulis atas ketidakpuasan pelanggan terhadap pelayanan yang diberikan oleh suatu sistem pelayanan (Jamaluddin, 2020). Pengaduan masyarakat merupakan akibat dari penyimpangan dalam penyelenggaraan pelayanan publik. Ketidakpuasan masyarakat dalam menerima pelayanan menyebabkan masyarakat melakukan pengaduan atas keluhan yang mereka alami. Pada dasarnya pengaduan merupakan masukan positif yang bersifat konstruktif. Meskipun banyak organisasi yang telah menanggapi pengaduan sebagai hal positif, namun tidak dapat dipungkiri jika masih terdapat organisasi yang menanggapi hal tersebut sebagai ancaman bagi keberlangsungan organisasi mereka.

2.6 *Permukiman*

2.6.1 Definisi Permukiman

Permukiman adalah kawasan yang didominasi oleh lingkungan yang dilengkapi dengan prasarana dan sarana lingkungan dan tempat kerja yang memberikan pelayanan dan kesempatan kerja yang terbatas untuk mendukung perikehidupan dan penghidupan, sehingga fungsinya dapat berdaya guna dan berhasil guna. Permukiman ini dapat berupa permukiman perkotaan maupun permukiman perdesaan (Usman, 2021).

2.6.2 Definisi Kumuh

Pengertian Kumuh prasarana yang ada tidak sesuai, Kumuh adalah kesan atau gambaran standar yang berlaku, baik standar secara umum tentang sikap dan tingkah laku yang rendah dilihat dari standar hidup persyaratan rumah sehat, kepadatan bangunan, kebutuhan sarana dan penghasilan kelas menengah. Dengan air bersih, sanitasi maupun persyaratan kata lain, kumuh dapat diartikan sebagai kelengkapan prasarana jalan, ruang tanda atau cap yang diberikan golongan terbuka, serta kelengkapan fasilitas sosial atas yang sudah mapan kepada golongan bawah yang belum mapan, (Usman, 2021).

2.6.3 Kawasan Kumuh

Kawasan kumuh adalah tempat tinggal yang kumuh, pendapatan yang rendah dan tidak menentu, serta lingkungan yang tidak sehat dan bahkan membahayakan dan hidup penuh resiko dan senantiasa dalam ancaman penyakit dan kematian (Naurotun, 2017).

Wilayah kawasan kumuh (Usman, 2021) merupakan bagian yang terabaikan dalam pembangunan perkotaan. Hal ini ditunjukkan dengan kondisi sosial demografis di kawasan kumuh seperti kepadatan penduduk yang tinggi, kondisi lingkungan yang tidak layak huni dan tidak memenuhi syarat serta minimnya fasilitas pendidikan, kesehatan dan sarana prasarana sosial budaya. Tumbuhnya kawasan kumuh terjadi karena tidak terbendungnya arus urbanisasi



Gambar 2. 6 Kawasan Kumuh Kel. Cilacap (13/01/2020)

2.7 Database Management System (DBMS)

DBMS adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol pengaksesan *database* (Yuniarti, 2015).Kelebihan menggunakan DBMS :

1. Mengontrol Duplikasi Data.
2. Menyediakan Kemampuan Berbagai Data.
3. Meningkatkan Integritas Data.
4. Mengatur Keamanan Data.

Dengan menggunakan perangkat lunak ini pengelolaan data menjadi lebih mudah dilakukan. Selain itu perangkat lunak ini juga menyediakan berbagai piranti yang berguna. Misalnya piranti yang memudahkan dalam membuat berbagai bentuk laporan

MySQL adalah sistem manajemen database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database dan SQL database manajemen sistem (DBMS), database ini dibuat untuk keperluan sistem database yang cepat, handal, dan mudah digunakan. Ulf Michael Widenius adalah penemu awal versi pertama MySQL yang kemudian pengembangan selanjutnya dilakukan oleh perusahaan MySQL AB. MySQL AB yang merupakan sebuah perusahaan komersial yang didirikan oleh para pengembang MySQL. Berikut ini beberapa kelebihan MySQL sebagai database server antara lain (Iit, 2015):

1. Source MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan database dapat dilakukan dengan mudah.
4. MySQL merupakan program yang multithreaded, sehingga dapat dipasang pada server yang memiliki multiCPU.
5. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dsb.
6. Bekerja pada berbagai platform. (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jumlah kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
8. Memiliki sistem security yang cukup baik dengan verifikasi host.
9. Mendukung ODBC untuk sistem operasi windows.
10. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

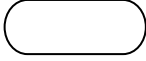
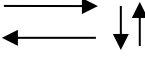
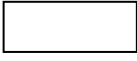
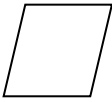
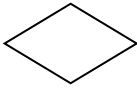


2.8 Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak. Selain itu *UML* adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. *UML* menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. *UML* tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Novianto, 2016).

2.8.1 Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma (Novianto, 2016) . Simbol *flowchart* dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol Flowchart




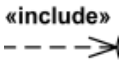
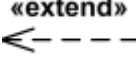

No	Simbol	Nama Simbol dan Keterangan
1.		Simbol Terminal yaitu untuk menyatakan permulaan atau akhir suatu program.
2.		Simbol Arus / <i>Flow</i> yaitu untuk menyatakan jalannya arus suatu proses (arah aliran program).
3.		Simbol Proses yaitu untuk menyatakan proses perhitungan / proses pengolahan data.
4.		Simbol <i>Input - Output</i> yaitu untuk memasukan data (proses <i>input</i>) maupun menunjukan hasil (<i>output</i> data) dari suatu proses.
5.		Simbol <i>Decision</i> / Logika yaitu untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak.
6.		Simbol <i>Document</i> yaitu untuk mencetak laporan ke printer.
7.		Simbol <i>Manual Operation</i> adalah simbol berfungsi untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.

2.8.2 Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan

terlihat di mata *user*. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client (Novianto, 2016).

Tabel 2. 3 Simbol-simbol pada *use case* diagram






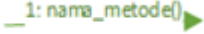

No	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> adalah menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Use Case</i> menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu.
3.		<i>Association</i> penghubung antara <i>Actor</i> dan <i>Use Case</i>
4.		<i>Include</i> yaitu perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
5.		<i>Extend</i> yaitu perilaku suatu <i>use case</i> memperluas perilaku <i>use case</i> lain. <i>Use case</i> boleh dilakukan boleh tidak (optional)
6.		<i>Sistem Boundary</i> yaitu batasan dari sebuah sistem.



2.8.3 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima oleh objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram

sequence juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada diagram sequence (Novianto, 2016).

Tabel 2. 4 Simbol-simbol pada sequence diagram

No	Simbol	Keterangan
1.		<i>aktor</i> , orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
2.		Garis hidup/ <i>lifeline</i> , menyatakan kehidupan suatu objek
3.		objek, menyatakan objek yang berinteraksi pesan
4.		Waktu aktif, menyatakan objek dalam keadaan aktif dan berinteraksi
5.		Pesan tipe <i>create</i> , menyatakan suatu objek membuat objek lain arah panah mengarah pada objek yang dibuat
6.		Pesan tipe <i>call</i> , menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri
7.		Pesan tipe <i>send</i> , menyatakan bahwa suatu objek mengirim data atau masukan atau informasi ke objek lainnya arah panah mengarah pada objek yang dikirim

8.		Pesan tipe <i>return</i> , menyatakan bahwa suatu objek yang telah menjalankan operasi atau metode menghasilkan suatu kembalian ke objek tertentu arah panah mengarah pada objek yang menerima kembalian
9.		Pesan tipe <i>destroy</i> , menyatakan suatu objek mengakhiri hidup objek yang lain arah panah mengarah pada objek yang diakhiri sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

2.8.4 Class Diagram




Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan. *Class diagram* memberi kita gambaran tentang perangkat lunak dan relasi-relasi yang ada didalamnya. Menurut (Whitten L. Jeffery et al 2004:432) *Class Diagram* adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk *UML* (Novianto, 2016).

Atribut dan operation dapat memiliki salah satu sifat berikut :

- a. *Private* hanya bisa dipanggil dari dalam kelas itu sendiri. Atribut diawali “-“.
- b. *Protected* hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. Atribut diawali dengan tanda “#”.
- c. *Public* dapat dipanggil dari semua objek atribut diawali tanda “+”

Ada beberapa simbol *relationships* antar class yg digunakan pada diagram class (Herdiansyah, 2010). Berikut adalah beberapa simbol-simbol yang ada pada class diagram dapat dilihat pada Tabel 2.5

Tabel 2. 5 Simbol-simbol Class Diagram

No	Simbol	Nama dan Keterangan
1.		Asosiasi adalah hubungan antar kelas.
2.		<i>Generalization</i> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus).
3.		<i>Composition</i> adalah bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yg menjadi part ikut musnah.

Multiplicity adalah jumlah banyaknya objek sebuah kelas yang berelasi dengan sebuah objek dari kelas lain yang berasosiasi dengan kelas tersebut (Novianto, 2016).
Macam-macam *multiplicity* disajikan dalam Tabel 2.6

Tabel 2. 6 Macam-macam Multiplicity

Nilai Kardinalitas	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nol atau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$)
0..n	Nol sampai n (dengan $n > 1$)
1..n	Satu sampai n (dengan $n > 1$)

2.9 Object Oriented Analysis and Design (OOAD)

Object Oriented Analysis and Design (OOAD) atau Pemrograman berorientasi objek adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk

menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah entitas yang memiliki atribut, karakter (*behaviour*) dan kadang kala disertai kondisi (*state*) (Fatin, 2020).

Konsep Dasar Pemrograman Berorientasi Objek :

a. Kelas (*class*)

Kelas (*class*) merupakan penggambaran satu set objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data pada pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas pada umumnya. Kelas merupakan jantung dalam pemrograman berorientasi objek.

b. Objek (*Object*)

Objek merupakan teknik dalam menyelesaikan masalah yang kadang muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek. Objek adalah entitas yang memiliki atribut, karakter dan kadangkala disertai kondisi.

c. Abstraksi (*Abstraction*)

Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan. Setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakkan kelebihan diterapkan.

d. *Encapsulation* (Pembungkusan)

Pembungkusan merupakan penggabungan potongan-potongan informasi dan perilaku-perilaku spesifik yang bekerja pada informasi tersebut, kemudian mengemasnya menjadi sesuatu yang disebut objek. Enkapsulasi adalah proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai

prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berintegrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

e. *Polimorfisme*

Polimorfisme merupakan suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.

f. *Inheritance* (Pewarisan)

Konsep *inheritance* mempunyai fungsi mengatur polimorfisme dan encapsulation dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa mengimplementasikan perilaku tersebut.

2.10 Pengertian *Framework*

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

2.11 Pengertian *Codeigniter*

Codeigniter adalah Sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal. (Yanuar Saputri, 2020)

Framework Codeigniter dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. kelebihan dari framework codeigniter jika dibandingkan dengan framework lain adalah sebagai berikut :

1. Open-Source

Kerangka kerja Codeigniter memiliki lisensi dibawah Apache/BSD open-source sehingga bersifat bebas atau gratis.

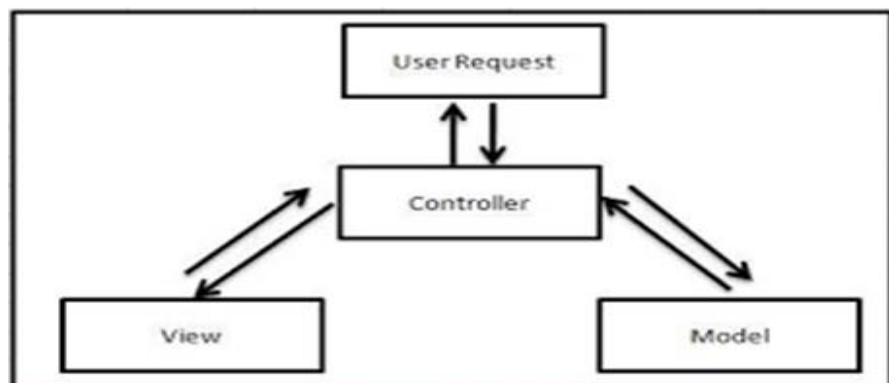
2. Berukuran kecil

Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan framework lain yang berukuran besar dan membutuhkan resource yang besar dan juga dalam eksekusi maupun penyimpanannya.

3. Menggunakan konsep M-V-C

Codeigniter merupakan konsep M-V-C (ModelView-Controller) yang memungkinkan pemisahan antara layer application-logic dan presentation. Dengan konsep ini kode PHP, query Mysql, Javascript dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau maintenance.

- a. Model Kode merupakan program (berupa OOP class) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (input-edit-delete).
- b. View Merupakan kode program berupa template atau PHP untuk menampilkan data pada browser.
- c. Controller merupakan Kode program (berupa OOP class) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan view. Adapun alur dari program aplikasi berbasis codeigniter yang menggunakan konsep M-V-C ditunjukkan pada gambar berikut :



Gambar 2. 7 Konsep Aliran M-V-C