

BAB II

LANDASAN TEORI

2.1 Visualisasi Data

Visualisasi data adalah proses yang efektif untuk mempresentasikan data dan informasi secara grafis dan bergambar. Visualisasi data berhubungan dengan pengembangan, perancangan, dan pengaplikasian representasi data secara grafis dan membuatnya lebih mudah mengerti maksud dari data. Terdapat dua kategori dari visualisasi data, yakni eksplanasi dan eksplorasi. Visualisasi data eksplorasi sangat berguna ketika data tersedia secara kuantitas namun pengetahuan mengenai data itu sangat kecil serta memiliki tujuan yang kurang jelas. Sementara visualisasi data explanasi berguna ketika data tersedia secara kuantitas dan kita mengerti mengenai data tersebut. Visualisasi data membantu dalam mengkomunikasikan data yang rumit dengan akurasi, klarifikasi, dan efisiensi. Kesuksesan visualisasi sangat bergantung pada obyektif yang jelas dan spesifik, hanya mengandung informasi yang relevan, data yang fokus, serta mempresentasikan data sesuai dengan pola dan relasi antar data [4].

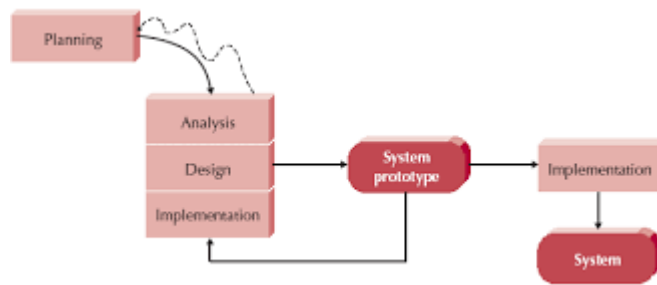
2.2 Bencana Alam

Menurut Undang-undang Nomor 24 Tahun 2007 Tentang Penanggulangan Bencana, bencana adalah peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam dan/atau faktor nonalam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda, dan dampak psikologis. Sementara itu, bencana alam bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang disebabkan oleh alam antara lain berupa gempa bumi, tsunami, gunung meletus, banjir, kekeringan, angin topan, dan tanah longsor. [5]

Bencana alam juga didefinisikan sebagai dampak yang cepat dan terjadi secara langsung atau mendalam dari lingkungan alam terhadap kehidupan sosio-ekonomi manusia sekitarnya. Bencana alam yang terjadi di negara-negara berkembang seperti Indonesia dapat menyebabkan lebih banyak korban jiwa. Hal itu terjadi karena di dalam negara berkembang terdapat banyak daerah perkotaan yang padat akan penduduk sehingga menjadi tempat yang berbahaya bila terkena bencana alam dan daerah perdesaan yang lingkungannya telah dieksploitasi besar-besaran. Pertumbuhan penduduk yang semakin cepat memperparah dampak dari bencana alam yang terjadi di negara berkembang. [6].

2.3 Prototyping

Prototyping merupakan metode yang prosesnya melalui tahapan analisis, perancangan, dan implementasi dilakukan secara bersamaan dan ketiga tahapan tersebut dilakukan berulang kali dalam satu siklus sampai sistem selesai. Keuntungan utama dari metodologi berbasis *prototyping* adalah bahwa ia sangat cepat menyediakan sistem yang dapat berinteraksi dengan pengguna, meskipun sistem belum siap dipakai secara luas. Tahap-tahap prototyping dilakukan pertama-tama mengidentifikasi dan menganalisis kebutuhan *user*. Lalu, pengembang membuat *prototype* di mana kebutuhan *user* diimplementasi di dalamnya. Kemudian *prototype* yang sudah jadi ditunjukkan kepada *user* agar *user* memberikan umpan balik terhadap sistem. Dari umpan balik tersebut, maka kembali dari tahap analisis, perancangan, dan implementasi sampai *user* merasa puas akan sistem tersebut. Setelah *prototype* terpasang, maka perbaikan akan terus dilakukan hingga sistem diterima sebagai sebuah sistem yang baru [7].



Gambar 2.1 Proses Metode Prototipe
Sumber: Dennis et al., 2015, p. 10

2.4 Python

Python merupakan salah satu bahasa pemrograman yang dapat dipakai untuk berbagai tujuan yang bisa dipakai secara efektif untuk membangun berbagai program komputer tanpa memerlukan akses langsung pada perangkat keras komputer. Bahasa pemrograman ini bisa dijalankan di berbagai *platform* yang berbeda. *Python* termasuk standar *library* yang komprehensif dari modul yang diuji yang bisa dengan berkoperasi dengan program kita. *Graphical User Interface* (GUI) yang disediakan *Python* cukup kuat dan mudah dipakai sehingga pengembangan aplikasi *user interface* cukup mudah dibuat dengan menggunakan *python*. *Python* juga mendukung *scripting* yang cocok untuk *rapid application development* (RAD). Banyak pengembang menggunakan *Python* karena *python* memiliki berbagai keuntungan yakni sederhana, mudah dipelajari, gratis dan *open-source*, dan sebagainya [8]. Dalam tugas akhir ini, *Python* digunakan untuk melakukan *web scrapping* pada situs web BMKG.

2.5 Power BI

Power BI adalah kumpulan layanan perangkat lunak, aplikasi, dan konektor yang bekerja bersama untuk mengubah sumber data Anda yang tidak terkait menjadi wawasan yang koheren, imersif secara visual, dan interaktif. Data Anda mungkin berupa lembar bentang Excel, atau kumpulan gudang data hibrid berbasis cloud dan lokal. Power BI memungkinkan Anda terhubung dengan mudah ke sumber data Anda, memvisualisasikan dan menemukan apa yang penting, dan membagikannya dengan siapa pun atau semua orang yang Anda inginkan [9].

Dalam tugas akhir ini, power BI dipakai untuk menerapkan visualisasi data bencana geologi di Indonesia yang berupa peta persebaran dan diagram batang.

2.6 Hypertext Markup Language

Hypertext Markup Language (HTML) merupakan sebuah bahasa yang berguna untuk mengatur tampilan konten di dalam *website*. Di dalam HTML, terdapat tag-tag tertentu di mana berfungsi menyediakan informasi terkait dengan sifat dan struktur konten serta referensi untuk gambar maupun media lainnya. Seluruh tag HTML bersifat dinamis yang berarti kode HTML tidak dapat dijadikan sebagai *file executable program*. Hal tersebut dikarenakan, HTML hanyalah sebuah bahasa script yang dapat berjalan jika dijalankan di dalam browser. [10]. Sekarang ini, versi terbaru dari HTML ialah versi 5.

2.7 Bootstrap

Bootstrap merupakan *framework* yang diciptakan oleh Twitter pada tahun 2010 yang memiliki tujuan untuk memudahkan pengembangan *website*. *Bootstrap* mencakup template yang berisikan CSS, HTML, dan *library* Javascript. Pada awalnya *bootstrap* hanya digunakan oleh Twitter sendiri namun baru dipublikasikan pada 19 Agustus 2011 [11]. Sekarang ini versi *bootstrap* terbaru ialah 4.5.0. Dalam pengembangan aplikasi usulan ini, menggunakan *bootstrap* versi 4.5.0 untuk memudahkan dalam desain tampilan *website*.

2.8 Hypertext Preprocessor

Hypertext Preprocessor (PHP) merupakan *script* yang terintegrasi dengan HTML dan berbasis *server* dimana *script* tersebut digunakan untuk membuat *website* yang dinamis. Dalam pembuatan *website*, PHP dapat dipakai dengan tujuan apapun di mana cocok untuk mengembangkan aplikasi *web* berbasis *server* di mana PHP dijalankan di *server web* [10]. Selain itu, PHP juga dapat dipakai untuk mengolah basis data. Basis data menyimpan data yang diolah oleh bahasa PHP yang ada di *web server* lalu data tersebut dapat ditampilkan dengan *web*

browser dengan cara menulis alamat sesuai dengan file yang terdapat di *web server* [12].

2.9 Web Scrapping

Web Scrapping digunakan di berbagai bidang untuk mengumpulkan data yang format lainnya sulit ditemukan. Teknik ini sangat berguna apabila seseorang ingin mengambil data dari *website* namun *website* tersebut tidak menyediakan API untuk mengakses data atau API yang disediakan tidak sesuai dengan data yang diinginkan. *Web scraping* berkaitan dengan aktivitas mengunduh data terstruktur dari *web*, memilih beberapa data itu, dan meneruskan semua yang dipilih ke proses lain. Melakukan *web scrapping* merupakan hal yang legal bila seseorang memakai data tersebut untuk keperluan pribadi. Namun lain halnya bila data yang dikoleksi tersebut hendak diterbitkan ulang. Upaya ini secara tak langsung dapat menjatuhkan *website* empunya data dimaksud [13]. *Web scrapping* yang dilakukan dalam tugas akhir ini menggunakan bahasa pemrograman Python untuk mengoleksi data dari situs *web* resmi BMKG (Badan Meteorologi, Klimatologi dan Geofisika) (<https://www.bmkg.go.id/gempabumi/gempabumi-terkini.bmkg>).

2.10 Unified Modeling Language

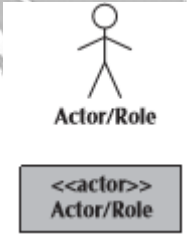

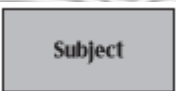



Unified Modeling Language (UML) merupakan kumpulan pemodelan berbasis objek untuk menggambarkan sistem yang dikembangkan. Tujuan dari UML ialah untuk menyediakan kosakata umum tentang istilah yang berorientasi objek dan teknik pembuatan diagram yang cukup bervariasi untuk memodelkan setiap proyek pengembangan sistem dari analisis melalui implementasi. Pada UML versi 2.5, terdapat 2 kelompok utama yakni pemodelan perilaku dan pemodelan struktural [7].


2.10.1 Pemodelan Perilaku

2.10.1.1 Use Case Diagram

Use case diagram merupakan gambaran sebuah proses bisnis secara luas dalam sebuah sistem informasi bisnis. Use case diagram digambar ketika mengumpulkan dan mendefinisikan kebutuhan sistem. Oleh karena itu, use case diagram menyediakan cara komunikasi yang langsung dan sederhana kepada pengguna seperti apa sistem yang diinginkan [7].

Tabel 2.1 Elemen-elemen *Use Case Diagram*

No.	Notasi	Elemen
1		<p>Actor</p> <ul style="list-style-type: none"> • Adalah seseorang atau sistem yang memperoleh keuntungan dari dan berada di luar subjek • Digambarkan dengan <i>stick figure</i> dan untuk nonhuman <i>actor</i> digambarkan dengan kotak yang bertuliskan <<actor>> • Diberi label sesuai dengan perannya • Dapat berasosiasi dengan actor lain dengan menggunakan asosiasi <i>specialization/superclass</i> • Diletakkan di luar batas subjek
2		<p>Use Case</p> <ul style="list-style-type: none"> • Menggambarkan sebagian besar fungsi sistem • Dapat <i>extend use case</i> lain • Dapat <i>include use case</i> lain • Diletakkan di dalam batas sistem • Diberi label dengan menggunakan frase kata kerja deskriptif-kata benda
3		<p>Subject Boundary</p> <ul style="list-style-type: none"> • Nama subjek ditulis di dalam atau di atas • Mewakili <i>scope</i> dari subjek
4		<p>Generalization Relationship</p> <ul style="list-style-type: none"> • Mewakili <i>use case</i> khusus ke <i>use case</i> yang lebih umum • Tanda panah terarah ke <i>parent/base use case</i>
5		<p>Include relationship</p> <ul style="list-style-type: none"> • Mewakili penyertaan fungsionalitas satu <i>use case</i> dengan <i>use case</i> lain • Tanda panah terarah ke sub <i>use case</i>
6		<p>Extend Relationship</p> <ul style="list-style-type: none"> • Mewakili perilaku opsional • Tanda panah terarah ke <i>parent/base use case</i>






No.	Notasi	Elemen
7		Association Relationship Digunakan untuk menghentikan setiap <i>control flow</i> dan <i>object flow</i> di dalam sebuah aktivitas (atau <i>action</i>)

Sumber: Dennis et al., 2015, p. 122




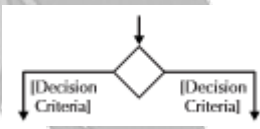
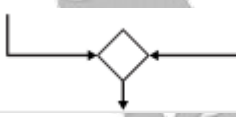



2.10.1.2 Activity Diagram

Activity diagram merupakan gambaran yang dipakai untuk menambah pemahaman dari proses bisnis dan model dari *use case*. Secara teknis, *activity diagram* dapat digunakan untuk berbagai jenis aktivitas pemodelan proses. *Activity diagram* dan *use case* merupakan model logis di mana model yang menggambarkan sebuah aktivitas domain bisnis tanpa menunjukkan bagaimana mereka dilakukan secara detail. Model ini menyediakan informasi yang dibutuhkan untuk membangun sebuah sistem. Analyst bisa berfokus pada bagaimana bisnis berjalan tanpa terganggu dengan detail implementasi bila berfokus pada aktivitas logis terlebih dahulu [7].

Tabel 2.2 Elemen-elemen *Activity Diagram*

No.	Notasi	Elemen
1		Action <ul style="list-style-type: none"> Adalah suatu bentuk perilaku yang sederhana dan tidak bisa didekomposisi. Diberi label sesuai dengan namanya
2		Activity <ul style="list-style-type: none"> Mewakili satu set <i>action</i> Diberi label sesuai dengan namanya
3		Object Note <ul style="list-style-type: none"> Mewakili sebuah objek yang terhubung pada satu set <i>object flows</i> Diberi label sesuai dengan nama kelasnya
4		Control Flow <ul style="list-style-type: none"> Menunjukkan urutan eksekusi
5		Object Flow <ul style="list-style-type: none"> Menunjukkan <i>flow</i> sebuah objek dari satu aktivitas (atau <i>action</i>) ke aktivitas lain (atau <i>action</i>)

Tabel 2.2 Elemen-elemen *Activity Diagram* (lanjutan)

No.	Notasi	Elemen
6		Initial Node <ul style="list-style-type: none"> Menjelaskan awal dari serangkaian action atau aktivitas
7		Final Node <ul style="list-style-type: none"> Digunakan untuk menghentikan setiap <i>control flow</i> dan <i>object flow</i> di dalam sebuah aktivitas (atau <i>action</i>)
8		Final-flow Node <ul style="list-style-type: none"> Digunakan untuk menghentikan <i>control flow</i> dan <i>object flow</i> tertentu di dalam sebuah aktivitas (atau <i>action</i>)
9		Decision Node <ul style="list-style-type: none"> Digunakan untuk mewakili kondisi pengujian untuk memastikan aliran kontrol atau arus objek hanya turun satu jalur Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur yang spesifik
10		Merge Node <ul style="list-style-type: none"> Digunakan untuk mengembalikan jalur keputusan yang berbeda yang terbentuk dari <i>decision node</i>
11		Fork Node <ul style="list-style-type: none"> Digunakan untuk memisahkan perilaku menjadi serangkaian arus aktivitas yang paralel atau bersamaan
12		Join Node <ul style="list-style-type: none"> Digunakan untuk menggabungkan kembali serangkaian arus aktivitas yang paralel atau bersamaan
13		Swimlane <ul style="list-style-type: none"> Digunakan untuk memecah <i>activity diagram</i> menjadi beberapa baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) kepada individu atau objek yang bertanggung jawab dalam melaksanakan aktivitas (atau tindakan) Diberi label dengan nama individu atau objek yang bertanggung jawab

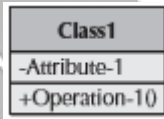
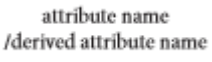


Sumber: Dennis et al., 2015, p. 132

2.10.2 Pemodelan Struktural




2.10.2.1 Class Diagram

Class diagram merupakan model statis yang dibuat untuk menunjukkan kelas dan hubungan di antara kelas yang konstan dalam sistem dalam waktu ke waktu. *Class diagram* menggambarkan kelas, yang mencakup perilaku dan status, dengan hubungan antara kelas [7].

Tabel 2.3 Elemen-elemen *Class Diagram*

No.	Notasi	Elemen
1		<p>Class</p> <ul style="list-style-type: none"> • Mewakili seseorang, tempat, atau hal tentang sistem, yang dibutuhkan untuk mengambil dan menyimpan informasi • Diberi label dengan nama yang dicetak tebal dan diletakkan di bagian atas • Memiliki daftar atribut di bagian tengah • Memiliki daftar operasi di bagian bawah • Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas
2		<p>Attribute</p> <ul style="list-style-type: none"> • Mewakili properti yang mendeskripsikan kondisi dari sebuah objek • Bisa diturunkan dari atribut lain, dengan menunjukkan garis miring "/" sebelum nama atribut
3		<p>Association</p> <ul style="list-style-type: none"> • Mewakili sebuah hubungan antara beberapa kelas atau dengan dirinya sendiri • Dilabeli menggunakan frase kata kerja atau nama peran • Bisa berada di antara satu atau lebih kelas • Berisi beragam simbol, yang merepresentasikan waktu minimum dan maksimum sebuah class instance dapat diasosiasikan dengan <i>class intance</i> terkait
4		<p>Operation</p> <ul style="list-style-type: none"> • Mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas • Dapat diklasifikasikan sebagai <i>constructor</i>, <i>query</i>, atau pembaharuan operasi • Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi

Tabel 2.3 Elemen-elemen *Class Diagram*(lanjutan)

No.	Notasi	Elemen
5		Generalization <ul style="list-style-type: none"> Mewakili hubungan antar beberapa kelas
6		Aggregation <ul style="list-style-type: none"> Mewakili bagian logis dari hubungan antara beberapa kelas atau dengan dirinya sendiri Merupakan bentuk asosiasi khusus
7		Composition <ul style="list-style-type: none"> Mewakili bagian fisik dari hubungan antara beberapa kelas atau dengan dirinya sendiri Merupakan bentuk asosiasi khusus

Sumber: Dennis et al., 2015, p. 178

2.11 Testing

Testing merupakan hal yang harus dilakukan pada pengembangan sistem, terutama saat tahap proses pemrograman sudah berakhir. *Testing* dilakukan untuk mengungkap perbedaan antara apa yang sebenarnya bisa dilakukan oleh sistem dengan apa yang harus dilakukan oleh sistem. Melakukan *testing* juga lebih penting pada sistem yang berorientasi pada obyek dari pada sistem lainnya. Ada empat tahap pengujian umum, yaitu *unit test*, *integration test*, *system tests*, dan *acceptance test*. *Unit test* berfokus pada satu unit-kelas. Ada dua pendekatan untuk pengujian unit: *black box testing* dan *white box testing*. Dalam penelitian ini, *testing* akan dilakukan dengan menggunakan tipe uji *black box testing*. *Black box testing* adalah tipe uji yang paling umum digunakan karena masing-masing kelas merepresentasikan objek yang dienkapsulasi. Tipe uji ini menggunakan sudut pandang user di mana kelas diperlakukan seperti sebuah kotak hitam dan user sebagai penguji berfokus untuk menentukan apakah kelas tersebut memenuhi persyaratan yang telah dibuat atau tidak. Dalam hal ini, rencana uji dikembangkan secara langsung dari spesifikasi kelas di mana setiap butir spesifikasi dijadikan ujian dan beberapa kasus uji dikembangkan untuk itu [7].

2.12 Penelitian Terlebih Dahulu

Tabel 2.4 Penelitian Terlebih Dahulu

No.	Judul	Tahun	Publikasi	Deskripsi	Aplikasi Usulan
1	Visualisasi Kualitas Penyebaran Informasi Gempa Bumi di Indonesia Menggunakan Twitter [14]	2019	Jurnal	Dalam penelitian ini, membahas mengenai perancangan untuk membangun visualisasi yang berguna untuk menganalisis kualitas informasi bencana yang disebar BMKG melalui <i>Twitter</i> .	Dalam aplikasi usulan, peneliti menggunakan sumber data dari <i>website</i> BMKG dan BNPB serta laporan dari pengguna. Pengumpulan data dari <i>website</i> menggunakan <i>web scrapping</i> . Aplikasi usulan bertujuan untuk membangun visualisasi data yang lebih baik.
2	Perancangan Sistem Informasi Bencana di Kabupaten Padang Pariaman [15]	2019	Jurnal	Dalam penelitian ini, membahas mengenai perancangan untuk membangun sebuah sistem informasi untuk mengolah data bencana berbasis <i>web</i> .	Dalam aplikasi usulan, informasi bencana meliputi seluruh Indonesia serta menggunakan <i>python</i> dalam membangun aplikasi.
3	Geovisual Analytics of Spatio-Temporal Earthquake Data in Indonesia [16]	2019	Jurnal	Dalam penelitian ini, membahas mengenai perancangan untuk membangun visualisasi <i>dashboard</i> untuk data historis gempa bumi yang dikombinasi dengan data spasial dan temporal	Dalam aplikasi usulan, aplikasi dibuat tanpa menggunakan <i>AppBuilder</i> . Selain itu, jenis bencana yang termasuk tidak hanya gempa bumi.

Tabel 2.4 Penelitian Terlebih Dahulu (lanjutan)

No.	Judul	Tahun	Publikasi	Deskripsi	Aplikasi Usulan
4	A quick earthquake disaster estimation system with fast urban earthquake simulation and interactive visualization [17]	2014	Jurnal	Dalam penelitian ini membahas mengenai perancangan untuk membangun sistem estimasi berdasarkan simulasi bencana gempa bumi cepat, bersama dengan metode visualisasi data interaktif yang untuk GPU workstation.	Dalam aplikasi usulan, visualisasi bukan ditujukan untuk mengestimasi datangnya bencana, namun memvisualisasi data bencana yg telah/sedang terjadi.
5	Sistem Deteksi Dini Bencana Tanah Longsor Berbasis 3D WebGIS [18]	2018	Jurnal	Dalam penelitian ini membahas mengenai perancangan untuk membangun sistem deteksi dini bencana tanah longsor berbasis 3D WebGIS dengan fitur real time monitoring system dan visualisasi data spasial.	Dalam aplikasi usulan, visualisasi data berupa 2D dan menampilkan data bencana yang telah terjadi.