

BAB 2

TINJAUAN PUSTAKA

2.1 Pemantauan

Pemantauan adalah sebuah kegiatan untuk mengamati perkembangan, pelaksanaan rencana pembangunan, mengidentifikasi serta mengantisipasi permasalahan yang timbul dan/atau akan timbul untuk dapat diambil tindakan sedini mungkin.

Pemantauan adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan, yang fokus pada proses dan keluaran. Monitoring menyediakan data dasar untuk menjawab permasalahan, sedangkan evaluasi adalah memposisikan data-data tersebut agar dapat digunakan dan diharapkan memberikan nilai tambah. Evaluasi adalah mempelajari kejadian, memberikan solusi untuk suatu masalah, rekomendasi yang harus dibuat, menyarankan perbaikan. Namun tanpa monitoring, evaluasi tidak dapat dilakukan karena tidak memiliki data dasar untuk dilakukan analisis, dan dikhawatirkan akan mengakibatkan spekulasi, oleh karena itu Monitoring dan Evaluasi harus berjalan seiring

2.2 Gunung

Gunung merupakan suatu bentuk permukaan tanah yang menjulang dan letaknya jauh lebih tinggi daripada tanah-tanah di daerah sekitarnya. Pada umumnya gunung lebih besar dibandingkan dengan bukit, tetapi beberapa bukit memiliki ketinggian yang lebih tinggi dibandingkan dengan gunung.

Gunung biasanya memiliki lereng yang curam dan tajam, bisa juga dikelilingi oleh puncak-puncak atau pegunungan. Iklim dipegunungan bisa berbeda beda tergantung ketinggian dari gunung tersebut, dan hanya beberapa jenis tumbuhan yang bisa hidup di sana. Gunung memiliki ketinggian berkisar antara 500-600m diatas permukaan laut.

2.2.1 Terbentuknya Gunung

Pada dasarnya gunung terbentuk karena dipengaruhi oleh gaya tektonik. Gaya tektonik yang bekerja dalam bumi yang disebut dengan orogenesis dan epeirogenesis. Gunung yang terbentuk oleh gaya tektonik orogenesis adalah sedimen tanah yang terkumpul dan menjadi berubah bentuk karena mendapat gaya tekan dari tumbukan lempeng tektonik yang ada. Ada tiga tipe tumbukan lempeng tektonik, antara lain yaitu lempeng busur, kepulauan dan benua, lautan dan benua, dan antara benua dengan benua. Tumbukan lempeng lautan dan benua menimbulkan deposit sedimen laut terhadap tepi lempeng benua. Tumbukan antara lempeng busur kepulauan dengan benua berakibat lempeng lautan menyusup ke lapisan asthenosfir dan batuan vulkanik dan sedimen menumpuk pada sisi benua sehingga terjadilah pegunungan Sierra Nevada di California pada zaman Mesozoic. Sedangkan tumbukan lempeng benua dengan benua merupakan proses pembentukan sistem pegunungan Himalaya dan Ural. Sedangkan dalam proses epeirogenesis merupakan gerakan yang membentuk benua yang bekerja sepanjang jari-jari bumi. Proses ini juga disebut gerakan radial karena gerakan mengarah atau menjauhi titik pusat bumi dan terjadi pada daerah yang sangat luas sehingga prosesnya lebih lambat dibandingkan dengan proses orogenesis. Pembentukan dataran rendah (*graben*) dan dataran tinggi (*horts*) adalah salah satu contoh proses epeirogenesis

2.2.2 Ranger

Ranger adalah pengelola gunung yang bertugas untuk mengelola area jalur pendakian. Ranger bertanggung jawab untuk melakukan perawatan area pendakian, membuat tanda jalur pendakian, memberikan pengarahan juga pengawasan pada pendaki, membantu evakuasi dan lain - lain.

1. Pengetahuan

Ranger biasanya memiliki pengetahuan dan pengalaman pada area pendakian seperti navigasi darat, jenis makanan dan obat-obatan di alam.

2. Perlengkapan

Ranger biasanya memakai pakaian mudah kering, tas carrier untuk logistik, kompas juga peta kontur atau alat navigasi digital, dan alat komunikasi radio walkie talkie.

3. Basecamp Ranger

Ranger biasanya memantau area pendakian dan melakukan pemeliraharan jalur namun kebanyakan ranger sering berkumpul di pos awal pendakian yaitu pos registrasi untuk melakukan verifikasi pendakian.

4. Evakuasi

Untuk menjalankan proses evakuasi, ranger biasanya menunggu laporan dari pendaki yang turun melapor, atau mengecek waktu pendakian setiap pendaki, jika ditemukan ada yang melebihi batas waktu pendakian, maka akan dilakukan langkah evakuasi awal. Jika terkonfirmasi hilang ranger biasanya akan melapor ke tim SAR untuk proses evakuasi lebih lanjut.

2.2.3 Pendaki

Pendaki adalah seorang yang melakukan proses pendakian kesuatu gunung baik itu untuk berwisata maupun untuk melakukan sebuah penelitian. Pendaki biasanya membawa sejumlah peratan daki sesuai dengan tipe dan iklim dari gunung yang didaki. Biasanya pendaki membawa tas carrier, Tenda, alat memasak dan alat navigasi.

2.2.4 Proses Pendakian Gunung di Indonesia

Proses pendakian gunung di Indonesia dibagi menjadi beberapa tahapan yaitu sebagai berikut:

1. Proses pengajuan izin Pendakian / SIMAKSI (Surat Izin Masuk Kawasan Konservasi)

Saat di Indonesia menerapkan system izin manual dan juga beberapa gunung sudah melakukan izin secara online.

2. Proses verifikasi sebelum pendakian

Proses verifikasi adalah tahapan pemeriksaan perlengkapan dan kesehatan pendaki sebelum melakukan pendakian

3. Pendakian

Pendakian adalah proses pendaki melakukan kegiatan pendakian di area pendakian gunung yang dituju.

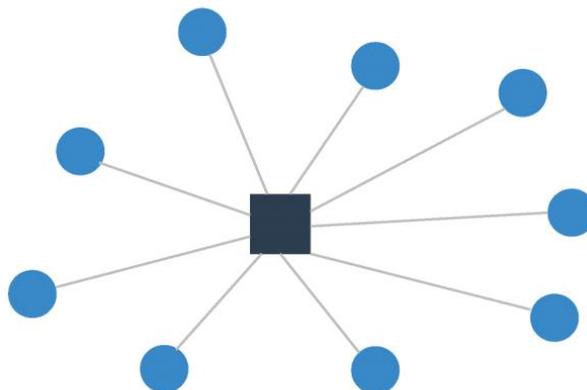
4. Proses verifikasi selesai pendakian

Proses ini adalah proses verifikasi bahwa pendaki telah selamat dan juga verifikasi pendaki tidak membawa hewan atau tumbuhan yang dilindungi dari area pendakian.

2.3 Lora (Long Range)

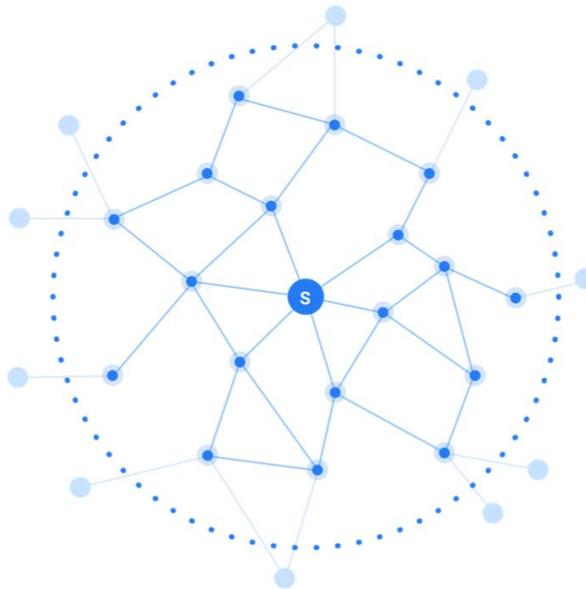
LoRa (kependekan dari *Long Range*) adalah sebuah teknologi *low power wide area network (LPWAN)* yang bekerja pada frekuensi radio 415 Mhz dan 915 Mhz dengan memanfaatkan teknik modulasi *spread spectrum* yang berasal dari teknologi *chirp spread spectrum (CSS)*. Perangkat LoRa dan teknologi frekuensi radio nirkabel Semtech adalah platform nirkabel berdaya rendah yang telah menjadi salah satu teknologi pada jaringan *Internet of Things (IoT)*.

Perangkat LoRa salah satunya mempunyai protokol yang dinamakan LoRaWAN, yaitu sebuah protokol yang memungkinkan untuk aplikasi IoT memecahkan beberapa masalah seperti: manajemen energi, pengurangan sumber daya alam, pengendalian polusi, efisiensi infrastruktur, pencegahan bencana, dan banyak lagi. Perangkat LoRa Semtech dan protokol LoRaWAN telah mengumpulkan digunakan diantaranya untuk *smartcity, smarthome / smart building, smart agriculture*, dan banyak lagi.



Gambar 2.1 Arsitektur LoRaWAN

Selain loraWAN kita juga bisa mengembangkan protocol komunikasi sendiri yang berjalan di atas teknologi LoRa. Salah satu alternatif dari protocol komunikasi loraWan yang menggunakan topologi Star adalah protocol komunikasi dengan topologi Mesh. Saat ini salah satu protocol yang bekerja pada topologi mesh dikembangkan secara terbuka oleh komunitas dengan proyek *Meshtastic*.



Gambar 2.2 Arsitektur LoRa Mesh

Meshtastic adalah proyek protocol komunikasi yang berjalan pada perangkat keras LoRa dan dapat diinstall pada microcontroller *opensource* seperti Arduino atau ESP32 untuk keperluan *outdoor*. Seperti hiking, ski, paralayang atau keperluan lainnya di area yang tidak memiliki jaringan yang bagus.

2.3.1 Algoritma LoRa Mesh Meshtastic

Protokol Mesh pada meshtastic menggunakan pendekatan algoritma *dynamic source routing (DSR)* dan *ad hoc on-demand distance vector routing (AODV)*, yang merupakan sebuah algoritma routing yang ditemukan oleh C. Perkins di Nokia Research Center pada tahun 2003 dan dikhususkan untuk perangkat wireless. Pada 2007 protokol ini algoritma ini dipublikasikan dengan status experimental di IETF kode RFC 4728.

2.3.2 Bluetooth API Service Meshtastic

Meshtastic menggunakan media bluetooth sebagai komunikasi antara LoRa Node dengan smartphone. Untuk bisa berkomunikasi dengan Node, Meshtastic menyediakan Bluetooth API yang bisa dilihat pada tabel berikut:

Tabel 2.1 Bluetooth API

UUID	Properti	Deskripsi
6ba1b218-15a8-461f-9fa8-5dcae273eafd	read,write	Informasi Node
8ba2bcc2-ee02-4a55-a531-c525c5e454d5	read	Baca pesan
f75c76d2-129e-4dad-a1dd-7866124401e7	write	Tulis pesan
e74dd9c0-a301-4a6f-95a1-f0e1dbea8e1e	write,read	Informasi Firmware
e272ebac-d463-4b98-bc84-5cc1a39ee517	write	OTA Update
4826129c-c22a-43a3-b066-ce8f0d5bacc6	write	OTA untuk selesai update
5e134862-7411-4424-ac4a-210937432c77	read,notify	OTA update notifikasi
GATT_UUID_SW_VERSION_STR/0x2a28	read	Informasi Software Version
GATT_UUID_MANU_NAME/0x2a29	read	Informasi Manufacture
GATT_UUID_HW_VERSION_STR/0x2a27	read	Informasi Hardware Version

2.4 Geographic Information System (GIS)

Geographic Information System (GIS) adalah sistem informasi yang mempunyai data berspasial yang diambil berdasarkan letak geografis suatu wilayah untuk proses analisis, penyimpanan dan visualisasi. GIS merupakan pengelolaan data geografis yang didasarkan pada kerja komputer (mesin).

Pada Tahun 1967 merupakan awal pengembangan SIG yang bisa diterapkan di Ottawa, Ontario oleh Departemen Energi, Pertambangan dan Sumber Daya. Dikembangkan oleh Roger Tomlinson, yang kemudian disebut CGIS (Canadian GIS - SIG Kanada), digunakan untuk menyimpan, menganalisis dan mengolah data yang dikumpulkan untuk Inventarisasi Tanah Kanada (CLI - Canadian land Inventory) - sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada dengan memetakan berbagai informasi pada tanah, pertanian, pariwisata, alam bebas, unggas dan penggunaan tanah pada skala 1:250000. Faktor pemeringkatan klasifikasi juga diterapkan untuk keperluan analisis.

CGIS merupakan sistem pertama di dunia dan hasil dari perbaikan aplikasi pemetaan yang memiliki kemampuan tumpang susun (overlay), penghitungan,

pendijitalan/pemindaian (digitizing/scanning), mendukung sistem koordinat nasional yang membentang di atas benua Amerika, memasukkan garis sebagai arc yang memiliki topologi dan menyimpan atribut dan informasi lokasional pada berkas terpisah. Pengembangnya, seorang geografer bernama Roger Tomlinson kemudian disebut "Bapak SIG".

CGIS bertahan sampai tahun 1970-an dan memakan waktu lama untuk penyempurnaan setelah pengembangan awal, dan tidak bisa bersaing dengan aplikasi pemetaan komersial yang dikeluarkan beberapa vendor seperti Intergraph. Perkembangan perangkat keras mikro komputer memacu vendor lain seperti ESRI, CARIS, MapInfo dan berhasil membuat banyak fitur *GIS*, menggabungkan pendekatan generasi pertama pada pemisahan informasi spasial dan atributnya, dengan pendekatan generasi kedua pada organisasi data atribut menjadi struktur database. Perkembangan industri pada tahun 1980-an dan 1990-an memacu lagi pertumbuhan *GIS* pada workstation UNIX dan komputer pribadi. Pada akhir abad ke-20, pertumbuhan yang cepat di berbagai sistem dikonsolidasikan dan distandardisasikan menjadi platform lebih sedikit, dan para pengguna mulai mengeksport menampilkan data *GIS* lewat internet, yang membutuhkan standar pada format data dan transfer.

2.5 Microprocessor

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input-output. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkoumputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer.

2.5.1 Arduino

Arduino adalah salah satu jenis kit microprocessor opensource yang saat ini terkenal dan banyak digunakan untuk industri maupun pembelajaran. Arduino bisa

diprogram menggunakan sebuah software bernama arduino IDE dan menggunakan bahasa pemrograman C/C++.

2.5.2 ESP32

ESP32 merupakan varian dari arduino yang di kembangkan *opensource* oleh komunitas. ESP32 bisa menggunakan perangkat pengembangan yang sama dengan arduino yaitu arduino ide dan bahasa pemrograman C/C++, Yang membedakan adalah ESP32 merupakan varian arduino komunitas yang sudah terdapat perangkat wifi dan bluetooth didalamnya serta sudah menggunakan prosessor dual-core dengan harga murah dibandingkan varian arduino yang sama.

2.6 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat di gunakan oleh berbagai macam peranti penggerak. Awalnya Google Inc.Membeli Android Inc. Pendatang baru yang membuat perangkat lunak untuk telepon genggam. Kemudian untuk mengembangkan Android di bentuklah Open Handset Alliance yang merupakan gabungan dari 34 perusahaan peranti keras,peranti lunak dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, Tmobile, dan Nvidia [17].

Pada saat perilisan perdana Android pada tanggal 5 November 2007,Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Terdapat dua jenis distributor sistem operasi Android.pertama yang dapat dukungan penuh dari Google atau Google Mail Service (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau di kenal sebagai Open Handset Distribution (DHD) [17]. Berikut ini adalah sejarah perjalanan Android:

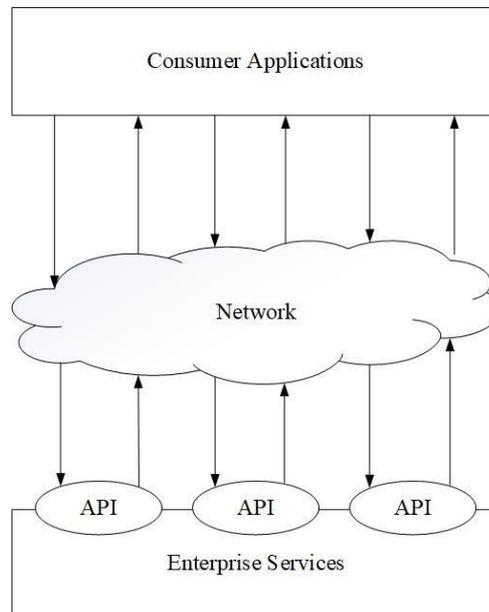
1. Android Versi 1.1
2. Android Versi 1.5 (Cupcake)
3. Android Versi 1.6 (Donut)

4. Android Versi 2.0/2.1 (Eclair)
5. Android Versi 2.2 (Froyo)
6. Android Versi 2.3 (Gingerbread)
7. Android Versi 3.0 (Honeycomb)
8. Android Versi 4.0 (Ice Cream Sandwich)
9. Android Versi 4.0 (Jelly Bean)
10. Android Versi 4.4 (Kitkat)
11. Android Versi 5.0 (Lollipop)
12. Android Versi 6.0 (Marshmallow)
13. Android Versi 7.0 (Nougat)
14. Android Versi 8.0 (Oreo)

Android digunakan dalam penelitian ini dengan alasan berdasarkan kuesioner yang telah dilakukan, terdapat banyak responden yang mengaku menggunakan perangkat smartphone dengan OS Android.

2.7 API (Application Programming Interface)

API merupakan software-to-software interface yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana suatu software dapat berinteraksi dengan software melalui jaringan tanpa interaksi pengguna. API dibuat untuk membantu mengekspos layanan bisnis atau aset perusahaan kepada para pengembang. Aplikasi dapat dipasang dan diakses dari berbagai jenis perangkat, seperti smartphone, Tablet, dan perangkat lainnya. Berikut adalah gambaran dari cara kerja API dapat dilihat pada Gambar 2.3



Gambar 2. 3 Interaksi API Terhadap Aplikasi Client

Pada penelitian ini API dimanfaatkan untuk mengintegrasikan sistem dari sub sistem web ke sub sistem Android untuk get dan request data seperti data artikel, data rekomendasi, dan penyimpanan data hasil perhitungan.

2.8 REST (Representational State Transfer)

REST adalah jenis arsitektur yang terdapat pada web untuk melayani suatu service. REST mempresentasikan cara interaksi antara server dan client untuk melakukan proses pertukaran informasi melalui media yang sama.

Dalam suatu jaringan, agar suatu resource dapat diakses, maka diperlukan identifikasi dan suatu bentuk manipulasi. Dapat digunakan URI (Uniform Resource Identifier) yang digunakan untuk mengidentifikasi resource yang ada pada suatu jaringan, dan dapat membuat resource menjadi addressable yang berarti resource dapat diketahui lokasinya dan dapat dimanipulasi dengan menggunakan suatu aplikasi. REST dapat digunakan sebagai interface dari API untuk mengakses suatu resource. API yang mengikuti prinsip dari REST architecture memberikan kemudahan bagi developer untuk tidak perlu mengetahui bagaimana struktur dari API di dalam server. Dalam hal ini, server akan memberikan informasi bagaimana agar client dapat mengakses service melalui API yang telah disediakan.

Penggunaan protokol HTTP pada REST architecture untuk komunikasi antara client dan server terletak pada HTTP method, yaitu GET, POST, PUT, dan DELETE. Method ini dapat digunakan untuk mengakses resources yang ada pada server, bergantung dari instruksi yang diberikan oleh server.

2.9 JSON (Javascript Object Notation)

JSON merupakan format pertukaran data yang digunakan untuk bertukar data antar platform. Format pertukaran data lain yang biasanya berupa XML. JSON merupakan format pertukaran data yang telah disetujui oleh banyak sistem untuk mengkomunikasikan data. JSON didasarkan pada literatur objek Javascript. Berikut adalah contoh penggunaan JSON.

Tabel 2. 3 Contoh Penggunaan JSON

```
{
  "id": 1,
  "username": "Siliwangi",
  "tempat_lahir": "Padjajaran",
}
```

Pada penelitian ini JSON digunakan sebagai output dari API dari data yang diakses oleh aplikasi.

2.10 SQLite

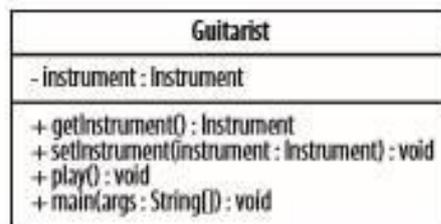
SQLite adalah sebuah perangkat lunak yang ditulis dalam bahasa C. SQLite dikembangkan oleh D. Richard Hipp pada tahun 2000 yang pada awalnya merupakan bagian dari kontrak dengan Angkatan Laut AS yang diimplementasikan oleh General Dynamics. SQLite ada di domain publik, jadi bisa digunakan oleh siapa saja. SQLite adalah perpustakaan yang dirancang untuk digunakan oleh satu pengguna untuk menangani fungsionalitas database [23].

Pada penelitian ini SQLite digunakan untuk menyimpan data device yang sudah terkoneksi dengan aplikasi agar aplikasi tidak melakukan proses scanning wearable berulang-ulang.

2.11 UML

UML (Unified Modeling Language) adalah standar bahasa pemodelan untuk pengembangan perangkat lunak dan sistem. Dalam sebuah perancangan perangkat lunak pemodelan merupakan salah satu bagian penting. Pemodelan dapat membantu mengkomunikasikan aspek penting dari perancangan sistem tersebut. Model merupakan abstraksi dari hal yang nyata. Pemodelan dilakukan untuk menyederhanakan sistem yang sebenarnya, sehingga memungkinkan perancangan dan kelangsungan suatu sistem mudah untuk dipahami, dievaluasi dan dikritik lebih cepat daripada harus menggali sendiri sistem yang sebenarnya.

Bahasa pemodelan dapat terdiri dari pseudo-code, kode aktual, gambar, diagram atau uraian yang panjang dan itu bisa digunakan untuk membantu seseorang menggambarkan sistemnya. Elemen yang membentuk bahasa pemodelan disebut notasi. Berikut merupakan contoh dari notasi UML, dapat dilihat pada Gambar 2.4



Gambar 2. 4 Contoh Notasi UML

Setiap pendekatan pemodelan memiliki kelebihan dan kekurangan yang berbeda, namun UML memiliki enam kelebihan utama yaitu, sebagai berikut:

1. Bahasa formal, Setiap elemen bahasa memiliki makna yang sangat jelas, sehingga dapat mudah dipahami ketika sedang memodelkan segi tertentu dari suatu sistem.
2. Singkat, Seluruh bahasa terdiri dari notasi sederhana dan mudah dipahami.
3. Komprehensif, Dapat menggambarkan semua aspek penting dari sebuah sistem.
4. Scaleable, dapat digunakan untuk memodelkan sistem dengan skala besar maupun skala kecil.

5. Built on Lessons Learned, UML adalah puncak dari praktik terbaik di komunitas *object-oriented* selama 15 tahun.
6. Standar, UML dikontrol oleh kelompok standar terbuka dengan kontribusi aktif dari sekelompok vendor dan akademisi di seluruh dunia. Standar ini memastikan kemampuan transformasi dan interoperabilitas UML, yang berarti tidak terikat dengan produk tertentu.

Penggunaan UML pada penelitian ini bermaksud untuk mempermudah dalam penggambaran perancangan sistem yang akan dibangun pada tahap analisis. Program yang dibangun menggunakan bahasa pemrograman Java yang sangat cocok digambarkan dengan UML karena menggunakan Object Oriented Programming.

2.12 Entity Relationship Model

Entity Relationship Model merupakan sebuah konsep didasarkan pada persepsi dunia nyata yang terdiri dari kumpulan objek dasar yang disebut entitas dan hubungannya antar objek yang disebut dengan relationship. Pada *Entity Relationship Model*, data yang ada dapat ditransformasikan kedalam perangkat konseptual dengan bantuan diagram yaitu ERD (*Entity Relationship Diagram*).

ERD menggambarkan arti dari aspek data seperti bagaimana entity-entity, atribut-atribut, dan relationship tersebut disajikan. Sebelum membuat ERD, tentunya ada beberapa aspek yang harus dipahami yaitu data yang diperlukan dan juga ruang lingkupnya. Dalam membuat ERD ada yang perlu diperhatikan dalam menentukan sesuatu konsep apakah merupakan suatu entity, atribut atau relationship.

Terdapat beberapa notasi yang harus diperhatikan dalam membuat sebuah ERD. Berikut ini merupakan beberapa notasi yang ada pada saat pembuatan ERD:

1. Entitas (*Entity*)

Entitas merupakan sebuah objek yang dapat dibedakan dengan yang lain dalam dunia nyata. Entity dapat berupa objek secara fisik seperti orang, rumah, atau kendaraan. Entity dapat juga berupa objek secara konsep, seperti pekerjaan, perusahaan, dan sebagainya. Entitas digambarkan dengan bentuk persegi panjang.

2. Atribut

Atribut merupakan karakteristik yang terdapat dalam entitas atau relationship, yang menyediakan penjelasan yang mendetail mengenai entitas atau relationship tersebut. Atribut digambarkan dengan bentuk oval yang dapat dilihat pada gambar 2.6

3. Relationship

Relationship merupakan hubungan yang terjadi antara satu atau lebih entitas. *Relationship* digambarkan dengan bentuk belah ketupat.

4. Cardinality (Kardinalitas)

Kardinalitas menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lainnya. Terdapat tiga jenis kardinalitas relasi yaitu satu ke satu (1:1), satu ke banyak (1:N / N:1) dan banyak ke banyak (N:N).

a. Satu ke Satu (1:1)

Misalkan terdapat entitas A dan B. Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan entitas B, demikian juga sebaliknya.

b. Satu ke Banyak atau Banyak ke Satu (1:N / N:1)

Misalkan terdapat entitas A dan B. Setiap entitas pada himpunan A dapat berelasi dengan banyak entitas pada himpunan B, tetapi tidak sebaliknya atau banyak entitas pada himpunan A dapat berelasi dengan paling banyak satu entitas pada entitas B, tetapi tidak sebaliknya.

c. Banyak ke Banyak (N:N)

Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.

5. Entitas Lemah (*Weak Entity*)

Entitas lemah adalah suatu entitas dimana keberadaannya tergantung dari keberadaan entitas lainnya. Entitas Lemah digambarkan dengan bentuk Persegi panjang yang didalamnya terdapat persegi panjang lainnya.

2.13 Kamus Data

Menurut Jogiyanto (2001:725) kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Kamus data dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut:

1. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di diagram arus data, maka nama dari arus data juga harus dicatat di kamus data, sehingga yang membaca diagram arus data dapat langsung mencarinya dengan mudah di kamus data.

2. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3. Bentuk Data

Bentuk dari data yang mengalir dapat berupa:

- a. Dokumen dasar atau formulir
- b. Dokumen hasil cetakan komputer
- c. Laporan tercetak
- d. Tampilan di layar monitor
- e. Variabel
- f. Parameter
- g. Field

Bentuk data ini perlu dicatat di kamus data, karena dapat digunakan untuk mengelompokkan kamus data ke dalam kegunaannya sewaktu perancangan sistem.

4. Arus Data

Arus data menunjukkan dari mana dan kemana data mengalir. Keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini.

5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan, dan kapan laporan-laporan harus dihasilkan.

7. Volume

Volume yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume ini digunakan untuk mengidentifikasikan besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah alat input, alat pemroses dan alat output.

8. Struktur Data

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

2.14 Skala Likert

Skala *Likert* adalah merupakan sebuah skala psikometrik (Pengukuran Psikologis) yang sering digunakan dalam angket dan merupakan skala yang paling banyak digunakan untuk keperluan riset yang berupa survei. Istilah skala ini diambil dari nama Rensis Likert, yaitu seseorang yang menerbitkan suatu laporan

yang menjelaskan penggunaannya. Ketika menanggapi pernyataan dalam skala *Likert*, responden menentukan tingkat persetujuan terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang ada. Biasanya tingkat persetujuan disediakan sebanyak lima pilihan dengan format sebagai berikut:

- a. Sangat Setuju
- b. Setuju
- c. Ragu-Ragu
- d. Tidak Setuju
- e. Sangat Tidak Setuju

Untuk keperluan analisis kuantitatif, maka untuk setiap jawaban dari responden dapat diberi skor, misalnya:

- | | |
|------------------------|---|
| a. Sangat Setuju | 5 |
| b. Setuju | 4 |
| c. Ragu-Ragu | 3 |
| d. Tidak Setuju | 2 |
| e. Sangat Tidak Setuju | 1 |

Instrumen penelitian dengan menggunakan skala *likert* ini dapat dibuat dalam bentuk *checklist* atau juga bisa dalam bentuk pilihan ganda.

2.15 Metode Pengujian Sistem

Metode pengujian sistem merupakan sebuah proses untuk mengetahui efektifitas dari aplikasi yang dipakai selain memberikan kesempatan kepada pengguna untuk menggunakan dan juga melakukan pengecekan terhadap laporan yang dihasilkan melalui *software*.

2.16 Pengujian Black Box

Pengujian Black Box merupakan pengujian yang berfokus pada apakah unit program yang dibuat sudah memenuhi kebutuhan yang sudah disebutkan dalam spesifikasi. Pada pengujian black box, cara pengujian dilakukan dengan mengeksekusi unit atau modul, setelah itu diamati untuk mengetahui apakah unit

itu sesuai keluarannya. Proses yang terdapat pada pengujian black box adalah analisis batasan nilai yang berlaku pada setiap data