# BAB II
# TINJAUAN PUSTAKA

## 2.1 Sekilas Tempat Penelitian

PT. Solusi Cipta Integrasi merupakan sebuah perusahaan konsultan IT yang bergerak dalam bidang kesehatan, khususnya rumah sakit, yang memiliki *client* di berbagai wilayah di Indonesia, khususnya di Jakarta, Belitung, dan Papua. Produk unggulan yang dimiliki oleh PT Solusi Cipta Integrasi adalah Avicenna-Hospital Information System, yaitu sistem yang dirancang untuk memenuhi semua kebutuhan informasi mulai dari tingkat operasional sampai pada tingkat *management*, serta menyediakan fasilitas analisa pada jajaran *management* yang dapat membantu menentukan langkah dalam pengambilan keputusan secara cepat dan akurat.

### 2.1.1 Visi Dan Misi PT. Solusi Cipta Integrasi

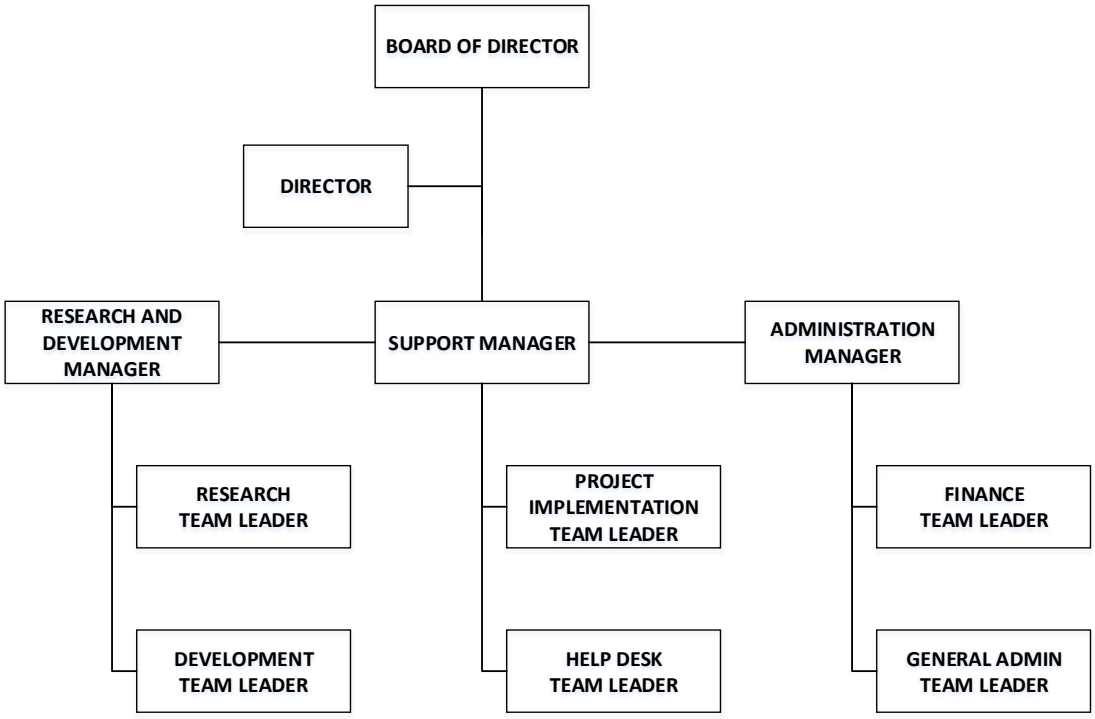
Menurut Wibisono, Visi adalah serangkaian kata-kata bahkan rangkaian kalimat mengungkapkan impian, cita-cita, rencana, harapan sebuah perkumpulan, perusahaan, organisasi yang ingin dicapai di masa mendatang [2]. Visi dari PT Solusi Cipta Integrasi adalah "Menjadi Solusi Yang Komprehensif Yang Mencakup Seluruh Aspek Dalam Bisnis Rumah Sakit".

Sedangkan Wheelen yang dikutip oleh Wibisono [2], misi adalah untaian kalimat yang berisi tujuan dan alasan keberadaan suatu organisasi yang memuat apa yang disediakan oleh perusahaan/ organisasi kepada masyarakat, berupa produk dan jasa. Berikut ini merupakan misi dari PT Solusi Cipta Integrasi, yaitu:

1. Membantu menyederhanakan, mengotomatisasi, dan mengoptimalkan fungsi-fungsi dalam meningkatkan efisiensi usaha dan tingkat kontrol
2. Mendukung rangkaian pelayanan di semua fungsi dan departemen di dalam organisasi
3. Menciptakan sistem yang dapat diimplementasikan sesuai dengan tujuan yang ingin dicapai rumah sakit dalam penerapan *hospital information system*

### 2.1.2 Struktur Organisasi

Struktur organisasi dapat diartikan sebagai kerangka kerja formal organisasi dimana kerangka kerja tersebut meliputi tugas-tugas pekerjaan yang dibagi-bagi, dikelompokkan, dan dikoordinasikan [3]. Berikut di bawah ini ini merupakan struktur organisasi dari PT Solusi Cipta Integrasi:




**Gambar 2.1 Struktur Organisasi PT Solusi Cipta Integrasi**

### 2.1.3 Deskripsi Kerja

Deskripsi pekerjaan adalah rekaman tertulis mengenai tanggung jawab dari pekerjaan tertentu [4]. Berikut ini adalah deskripsi kerja dari struktur organisasi PT Solusi Cipta Integrasi yang telah dijelaskan di atas, yaitu:

1. *Board of Directors*
    a. Melakukan pengawasan atas jalannya perusahaan dan memberikan nasihat kepada direktur
    b. Mengacu kepada kepentingan perusahaan dan sesuai dengan maksud dan tujuan perusahaan

c. Memiliki kewenangan khusus untuk diamanatkan dalam anggaran dasar untuk melaksanakan tugas-tugas tertentu direktur, apabila direktur berhalangan atau dalam keadaan tertentu
d. Membuat risalah rapat *board of director* dan menyimpan salinan rapat.

2. *Director*
   a. Memimpin perusahaan dengan menerbitkan kebijakan-kebijakan perusahaan
   b. Memilih, menetapkan, mengawasi tugas dari karyawan dan kepala bagian (manajer)
   c. Menyetujui anggaran tahunan perusahaan
   d. Mengkoordinasikan dan mengendalikan kegiatan pengadaan dan peralatan perlengkapan
   e. Memimpin rapat umum, dalam hal untuk memastikan pelaksanaan tata-tertib, keadilan dan kesempatan bagi semua untuk berkontribusi secara tepat, menyesuaikan alokasi waktu per item masalah, menentukan urutan agenda, menjelaskan dan menyimpulkan tindakan dan kebijakan
   f. Bertindak sebagai perwakilan organisasi dalam hubungannya dengan dunia luar
   g. Memainkan bagian terkemuka dalam menentukan komposisi dari *board of director*, sehingga tercapainya keselarasan dan efektivitas
   h. Mengambil keputusan sebagaimana didelegasikan oleh *board of director* atau pada situasi tertentu yang dianggap perlu, yang diputuskan, dalam meeting-meeting *board of director*
   i. Menjalankan tanggung jawab sebagai direktur perusahaan sesuai dengan standar etika dan hukum

3. *Research and Development Manager*
   a. Bertanggung jawab dalam seluruh proses penelitian dan pengembangan perusahaan
   b. Mengawasi dan memastikan kualitas performansi dalam perusahaan sesuai dengan standar yang telah ditetapkan perusahaan
   c. Membuat laporan berkala kepada direktur untuk menunjukkan status kemajuan penelitian dan pengembangan

d. Perumusan rencana dan program penelitian dan pengembangan

4. *Research Team Leader*
   a. Merumuskan perencanaan untuk peningkatan kualitas kinerja setiap divisi dan perusahaan
   b. Menetapkan target – target kinerja divisi
   c. Mengevaluasi hasil kinerja para divisi
   d. Mengawasi dan mengevaluasi kegiatan penelitian perusahaan
5. *Development Team Leader*
   a. Mengadakan proses pengembangan perusahaan.
   b. Meneliti kemungkinan-kemungkinan untuk ikut dalam pengembangan teknologi perusahaan.
   c. Melakukan analisa pengembangan pemasaran secara umum untuk membantu kemajuan perusahaan.
   d. Menerbitkan laporan-laporan dari penerbit-penerbit lainnya mengenai aktivitas perusahaan, dalam rangka aktivitas pengembangan perusahaan
   e. Memberikan saran-saran atau pertimbangan kepada direktur sesuai dengan langkah-langkah atau tindakan yang perlu diambil dibidang tugasnya.
6. *Support Manager*
   a. Bertanggung jawab pada kesiapan dan ketersediaan sistem komputer/aplikasi dalam lingkungan perusahaan
   b. Membuat dan/atau implementasi semua sistem dan aplikasi
   c. Merancang, mengelola, dan mengawasi serta mengevaluasi operasional dari sistem informasi (*software* dan aplikasi) dan pendukungnya (*hardware*, infrastruktur, telekomunikasi)
   d. Memberikan rekomendasi tentang solusi sistem informasi dan pendukungnya
   e. Memberikan arahan pada bawahan mengenai penggunaan dan solusi teknologi
   f. Memberikan laporan bulanan kepada bagian keuangan mengenai semua aspek yang berhubungan dengan *support*

7. *Project Implementation Team Leader*
    a. Melaksanakan tugas-tugas pengarahan dan pendayagunaan serta pengawasan atas semua personil proyek dalam bidang pelaksanaan
    b. bertanggungjawab kepada direktur dengan menjalin koordinasi sebaik-baiknya dengan bagian-bagian yang lain
    c. Pelaksanaan kewajiban-kewajiban perusahaan sesuai perjanjian dengan pemilik proyek
    d. Pelaksanaan yang sesuai dengan rencana kerja dengan ekonomis menguntungkan perusahaan dengan hasil yang dapat dipertanggungjawabkan
8. *Help Desk Team Leader*
    a. Mencatat, menyampaikan, dan menangani permasalahan (sesuai flowchart)
    b. Membuat Pelaporan/Rekapitulasi penyelesaian permasalahan
    c. Menangani account pengguna
9. *Administration Manager*
    a. Menyambut seluruh tamu VIP, klien, konsultan, ataupun tamu eksternal lain
    b. Mencatat notulensi rapat maupun dokumen penting sesuai standar
    c. Mengawasi kinerja staff penunjang lainnya
    d. Manajemen tata usaha, menjaga lingkungan kantor agar tertib dalam administrasi
    e. Mengatur rapat mingguan dan bulanan staff keuangan dan administrasi
10. *Finance Team Leader*
    a. Pembuatan laporan keuangan atau laporan kas perusahaan, laporan pergudangan, laporan bobot prestasi perusahaan, daftar hutang, dan lain-lain
    b. Membuat dan melakukan verifikasi bukti-bukti pekerjaan yang akan dibayar oleh perusahaan
    c. Membuat laporan akuntansi perusahaan dan menyelesaikan perpajakan serta retribusi perusahaan
    d. Membantu kepala bagian implementasi proyek terutama dalam hal keuangan sehingga kegiatan pelaksanaan proyek dapat berjalan dengan baik

11. *General Admin Team Leader*
    a. Melakukan seleksi atau perekrutan pekerja di proyek untuk pegawai bulanan sampai dengan pekerja harian dengan spesialisasi keahlian masing-masing sesuai posisi organisasi proyek yang dibutuhkan
    b. Melayani tamu-tamu intern perusahaan maupun ekstern dan melakukan tugas umum
    c. Mengisi data-data kepegawaian, pelaksanaan, dan asuransi tenaga kerja
    d. Menyimpan data kepegawaian karyawan dan pembayaran gaji serta tunjangan karyawan

## 2.2 Landasan Teori

Pada Sub bab ini berisi tentang beberapa teori yang mendukung dalam proses analisis dan implementasi berdasarkan masalah yang diangkat dalam pembangunan *web service* untuk aplikasi pelayanan rumah sakit *mobile*. Teori-teori yang digunakan yaitu sistem informasi, komponen sistem informasi, Android, *web service*, arsitektur *web service*, JSON (*JavaScript Object Nation*), API (*Application Programing* Interface), SOAP (*Simple Object Access Protocol*), WSDL (*Web Service Description Language*), UDDI (*Universal Description Discovery and Integration*), LBS (*Location Based Service*), Pencarian Rute Terpendek.

### 2.2.1 Sistem Informasi

Sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Sistem terdiri dari unsur-unsur seperti masukan(*input*), pengolahan(*processing*), serta keluaran(*output*). [5]

Pada bukunya yang berjudul *Management Information Systems*, Informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. [6]

Menurut Kertahadi, sistem informasi adalah alat untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya [7]

#### 2.2.3.1 KOMPONEN SISTEM INFORMASI

Menurut Robbin dan Coulter pada bukunya yang berjudul Manajemen, komponen sistem informasi terdiri dari beberapa hal sebagai berikut [3]:

1. Blok Masukan

   Masukan disini mewakili data yang masuk ke dalam sistem informasi. Masukan disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari prosedure, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

   Blok keluaran menghasilkan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan “kotak alat” dalam sistem informasi. Teknologi digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan, kegagalan sistem, kesalahan-kesalahan, ketidak efisiensian, sabotase dan lain sebagainya.

### 2.2.2 Android

*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi *Android* diluncurkan untuk umum pada musim gugur di tahun 2008. *Android* sangat berkembang pesat di industri karena dua aspek utama yaitu bersifat *opensource* dan model arsitekturya. Sebagai sebuah proyek yang bersifat *opensource*, memungkinkan *Android* untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada *bug* program hingga *hardware*.

#### 2.2.2.1 KEUNGGULAN ANDROID

Menurut Nazruddin beberapa keunggulan yang dimiliki oleh Android adalah: [8]

1. Lengkap (*Complete Platform*)

   Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang pengembangan aplikasi.

2. Terbuka (*Open Source Platform*)

   Platform Android disediakan melalui lisensi open source. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux kernel 2.6.

3. Free (*Free Platform*)

   Android adalah platform/aplikasi yang bebas untuk developer. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada platform Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

### 2.2.3 Web Service

Menurut W3C *Web services Architecture Working Group,* pengertian *web service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar Web. [9]

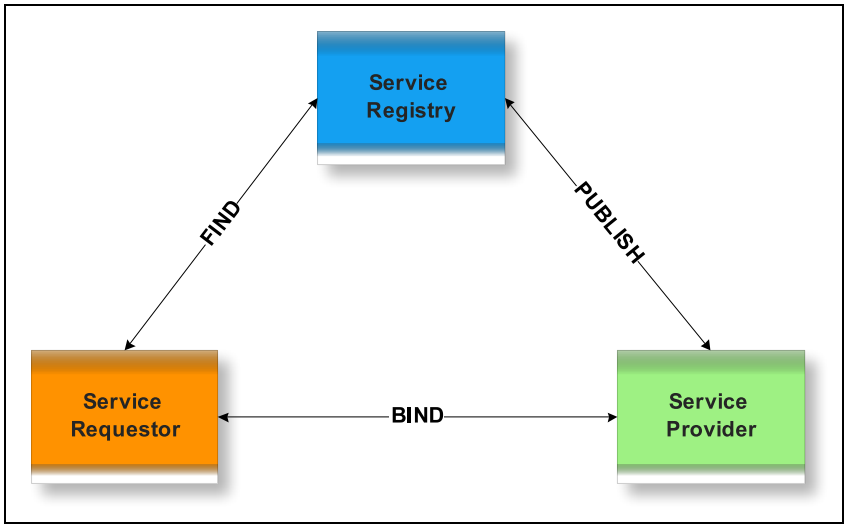
Secara umum, *web service* dapat diidentifikasikan dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *web* pada umumnya, URL *web service* hanya menggandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi. W*eb service* dapat pula diartikan sebagai perangkat lunak atau bagian dari perangkat lunak yang dapat diakses oleh berbagai *device* untuk menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

#### 2.2.3.1 ARSITEKTUR *WEB SERVICE*

Konsep arsitektur yang mendasari teknologi *web service* mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam sistem, yaitu:

1. *Service provider*, yaitu suatu entitas yang menyediakan interface terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu.
2. *Service requestor*, yaitu aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan ke *service provider*.
3. *Service registry*, yaitu tempat dimana *service provider* mempublikasikan layanannya.

Arsitektur *web service* secara umum dapat dilihat pada gambar di bawah ini:




**Gambar 2.2 Arsitektur *Web Service* [9]**

### 2.2.3.2 *SIMPLE OBJECT ACCESS PROTOCOL* (SOAP)

SOAP merupakan protokol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP merupakan gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan *protocol* HTTP sebagai sarana transport datanya dan data akan dipertukarkan ditulis dalam format XML. Karena SOAP mengunakan HTTP dan XML, maka SOAP memungkinkan pihak-pihak yang mempunyai *platform*, sistem operasi, dan perangkat lunak yang berbeda dapat saling mempertukarkan datanya. SOAP mengatur bagaimana *request* dan *response* dari suatu *web service* bekerja. [10]

Secara sederhana, SOAP dapat diartikan sebagai format standard dokumen berbentuk XML yang digunakan untuk saling bertukar informasi antara *web service* dengan aplikasi yang memanggilnya.

### 2.2.3.3 *WEB SERVICE DESCRIPTION LANGUAGE* (WSDL)

WSDL merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan *web service* dan menggambarkan bagaimana cara untuk mengakses *web service* tersebut. Fungsi utama WSDL dalam *web service* adalah untuk mengotomasi mekanisme komunikasi *business-to-business* dalam *web service* melalui protokol Internet.

WSDL merupakan representasi kontrak antara *requestor* dan *provider*-nya. Secara teknis merupakan representasi kontrak antara kode klien dan kode di server. Dengan menggunakan WSDL klien dapat memanfaatkan fungsi-fungsi publik yang disediakan oleh server. [10]

Di dalam WSDL dijelaskan *method-method* apa saja yang tersedia dalam *web service*, parameter apa saja yang diperlukan untuk memanggil sebuah *method*, dan apa hasil atau tipe data yang dikembalikan oleh *method* yang dipanggil tersebut.

#### 2.2.3.4 *UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION* (UDDI)

Menurut Ariba, IBM, dan Microsoft (2000) UDDI merupakan suatu cara untuk mempublikasikan dan menemukan informasi tentang *web service*. [11]

Untuk mencari sebuah *web service*, *service requestor* terlebih dahulu mengirimkan *query* ke UDDI *registry* untuk mencari *service* yang diinginkan. Kemudian, *service requestor* akan mendapatkan deskripsi WSDL. Dengan menggunakan deskripsi WSDL yang telah diperoleh tersebut, *service requestor* dapat membuat sebuah pesan SOAP untuk berkomunikasi dengan *web service* yang diinginkan.

### 2.2.4 *Javascript Object Nation* (JSON)

JSON (*JavaScript Object Nation*) adalah format pertukaran data ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemograman JavaScript, Standar ECMA-262 Edisi ke-3-Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Phyton dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. [12]

### 2.2.5 *Application Programming Interface* (API)

*Application programming interface* (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak.

Dalam suatu pemrograman dibutuhkan setidaknya ribuan *system call* per detik oleh karena itu banyak programmer yang menggunakan API. Didalam API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam s*ystem calls* dengan bahasa yang lebih mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system call* sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari *system call* sama dengan nama API.

Kelebihan pemograman menggunakan API adalah:

1. Portabilitas

   Programmer yang menggunakan API dapat menjalankan programnya dengan sistem operasi mana saja asalkan sudah terinstal API tersebut.

2. Mudah Dimengerti

   API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*.

### 2.2.6 *Location Based Service* (LBS)

*Location based service* adalah *service* yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* (GPS) dan *Google's Cell-based Location*. *Map* dan layanan berbasis lokasi menggunakan garis lintang dan garis bujur untuk menentukan letak geografis, namun sebagai *user* kita membutuhkan alamat atau posisi real time kita bukan nilai garis lintang dan garis bujur. *Android* menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, anda dapat mengkorversi nilai garis lintang dan garis bujur menjadi alamat dunia nyata atau sebaliknya [13].

### 2.2.7 Metode Haversine

Metode Haversine digunakan untuk menghitung jarak lingkaran antara 2 titik, yaitu jarak terpendek di atas permukaan bumi berdasarkan garis bujur dan garis lintang. [14]

Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari 6.367, 45 km, dan lokasi dari 2 titik di koordinant bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut: [15]

```
x = (lon2-lon1) * cos ((lat1+lat2)/2);
y= (lat2-lat1);
d= sqrt(x*x+y*y)*R
```

Keterangan:

$x = longitude\ (lintang)$

$y = latitude\ (bujur)$

$d = jarak$

$R = radius\ bumi = 6371\ km$

$1° = 0.0174532925$ radian

**Contoh Perhitungan Haversine Formula**

Lokasi 1 : $lon1 = 119.800801$

$lat1 = -0.790175$

Lokasi 2 : $lon2 = 119.8428$

$lat2 = -0.8989$

Konversi $lon1, lat1, lon2, lat2$ ke dalam bentuk radian:

$lat1 = -0.790175 * 0.0174532925 = -0.013791155$ radian

$lon1 = 119.800801 * 0.0174532925 = 2.090918422$ radian

$lat2 = -0.8989 * 0.0174532925 = -0.01569$ radian

$lon2 = 119.8428 * 0.0174532925 = 2.091651$ radian

$$x = \left(lon2 - lon1) * \cos(\frac{lat1 + lat2}{2})\right)$$
$$= \left(2.091651 - 2.090918422) * \cos(\frac{((-0.013791155 \ + -0.01569}{2})\right)$$
$$= \ 0.0007329412$$

$$y = (lat2 - lat1)$$
$$= (-0.01569 - (-0.013791155))$$
$$= -0.001897609$$

$$d = \sqrt{(x * x) + (y * y)} * R$$
$$= \sqrt{(0.0007329412 \ * \ 0.0007329412) + (-0.001897609 \ * \ -0.001897609)} * 6371$$
$$= \sqrt{0.0000041381} * 6371$$
$$= \mathbf{12.96009233\ km}$$

### 2.2.8 *Structure Analysis And Design Technique* (SADT)

*Structure analysis and design technique* atau SADT adalah metode untuk pengembangan sistem terstruktur yang dikembangkan pada tahun 1969 dan dikembangkan lebih lanjut pada tahun 1974. SADT membagi sistem kedalam 2 hal yaitu [16]

1. Sebagai benda (*object*): objek, dokumen, dan data.
2. Sebagai kejadian (*event*): kegiatan yang dilakukan manusia, perangkat lunak, dan mesin.

### 2.2.9 *Entity Relationship Diagram* (ERD)

ERD digunakan untuk menyatakan jenis data dan hubungan data yang ada diantara jenis data yang terdapat dalam sistem [17].

Komponen-komponen yang digunakan dalam ERD adalah:

1. *Entity* (Entitas)

   Suatu objek yang terdiri dari kumpulan elemen atau data yang dapat digunakan dari objek lainnya. Entitas terbagi menjadi 2 macam, yaitu :

a. Entitas kuat dimana keberadaannya tidak tergantung dari entitas lain,
b. Entitas lemah yang keberadaannya bergantung pada eksistensinya dalam sebuah relasi terhadap entitas lainnya serta tidak memiliki *primary key*.

2. *Relationship set*

   Relasi antar *entity* merupakan hubungan alamiah yang terjadi antar *entity*.

3. *Attribute*

   Atribut merupakan sifat atau karakteristik suatu entitas yang menyediakan penjelasan detail tentang entitas tersebut. Atribut terbagi menjadi beberapa macam yaitu :

   a. *Simple attribute*

      Merupakan atribut sederhana yang tidak dapat dibagi dalam beberapa bagian.

   b. *Composite attribute*

      Merupakan atribut yang dapat dibagi lagi dalam beberapa bagian. Contohnya ‘Nama’ yang bisa terbagi menjadi ‘Nama Depan’, ‘Nama Tengah’ dan ‘Nama Belakang’

   c. *Single-valued attribute* dan *multi-valued attribute.*

      *Single-valued attribute* merupakan atribut yang paling banyak memiliki satu nilai untuk setiap baris data, sedangkan *multi-valued attribute* merupakan atribut yang dapat diisi lebih dari satu nilai yang berjenis sama.

   d. *Derived attribute*

      Merupakan atribut yang diperoleh dari pengolahan atribut lain. Contohnya ‘Umur’ dapat diperoleh dari pengolahan ‘Tanggal Lahir’.

   e. *Mandatory* dan *non mandatory attribute*

      *Mandatory attribute* merupakan atribut yang harus diisi (tidak boleh kosong atau not null), sedangkan *non mandatory attribute* merupakan atribut yang boleh tidak diisi (kosong atau null).

4. Garis penghubung

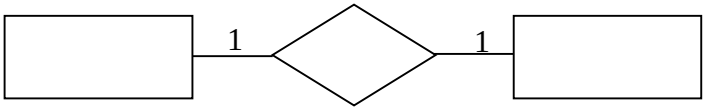
   Menghubungkan antara entitas satu dengan entitas yang lainnya.

5. *Cardinality* (Kardinalitas)

Merupakan tingkat hubungan yang terjadi antara *entity* didalam sebuah sistem, dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, terdapat 3 macam *cardinality* yaitu:

a. *One to One*

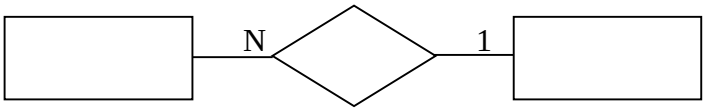
Kardinalitas *One to One* adalah tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua atau sebaliknya. Berikut ini adalah bentuk gambar kardinalitas *one to one*:




**Gambar 2.3 *Kardinalitas One to One***

b. *Many to One*

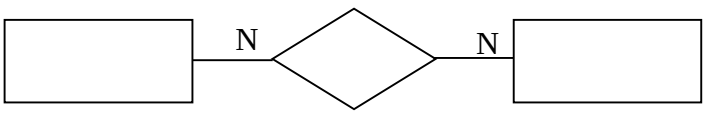
Kardinalitas *many to one* adalah tingkat hubungan dengan ke satu atau sama banyak satu ke banyak. Tergantung pada arah mana hubungan itu dilihat. Berikut ini adalah bentuk gambar kardinalitas *many to one*:




**Gambar 2.4 Kardinalitas *One to Many***

c. *Many to Many*

Kardinalitas *many to many* adalah tingkat hubungan banyak ke banyak terjadi jika kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi *entity* ke pertama maupun dilihat dari sisi *entity* ke kedua. Berikut ini adalah bentuk gambar kardinalitas *many to one*:




**Gambar 2.5 Kardinalitas *Many to Many***

6. Kunci (*Key*)

   Kunci merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan entitas yang lainnya dalam suatu himpunan entitas. Dari sudut pandang basis data, perbedaan diantara mereka harus dicerminkan lewat perbedaan dalam nilai atributnya. Nilai–nilai atribut kunci dapat secara unik mengidentifikasi suatu entitas terhadap entitas yang lainnya. Dengan kata lain, tidak ada lebih dari satu entitas memiliki nilai–nilai yang sama untuk semua atributnya. Macam–macam jenis kunci (*Key*) diantaranya:

   a. *Primary Key* (Kunci Primer)

      Adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian spesifik, tapi juga dapat mewakili setiap kejadian dari suatu *entity*. *Primary Key* mempunyai tiga kriteria yakni:

      i. *Key* tersebut lebih natural digunakan sebagai acuan.
      ii. *Key* tersebut lebih sederhana.
      iii. *Key* tersebut terjamin keunikannya.

   b. *Foreign Key* (Kunci Tamu)

      Merupakan sembarang atribut yang menunjuk kepada Primary Key pada tabel yang lain. *Foreign Key* terjadi pada suatu relasi yang memiliki *Cardinality one to many* atau *many to many*. *Foreign Key* biasanya selalu diletakan pada tabel/relasi yang mengaruh ke banyak.

   c. Candidate Key (Kunci Kandidat)

      Merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah relasi secara unik.

### 2.2.10 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari *Data Flow Diagram* yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Diagram konteks memberi gambaran tentang keseluruan sistem.

Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses [18].

### 2.2.11 *Data Flow Diagram*

*Data flow diagram* atau DFD adalah alat untuk menggambarkan komponen-komponen sebuah sistem, aliran data pada sistem, dimana data berasal, tujuan data serta penyimpanan data. Terdapat 4 komponen utama dalam DFD yaitu [16]:

1. Entitas eksternal (*External Entity*)
2. Arus data (*Data Flow*)
3. Proses (*Process*)
4. Penyimpanan data (*Data Store*)

DFD dapat dikatakan turunan dari diagram konteks, analisis lebih detail dari sistem dapat dilakukan dengan menurunkan DFD ke level 1,2,3 dan seterusnya. Penjelasan simbol pada diagram konteks dan DFD dapat dilihat di daftar simbol.

### 2.2.12 Spesifikasi Proses

Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD. Spesifikasi Proses berfungsi mendeskripsikan tahapan yang dilakukan untuk merubah masukan menjadi keluaran. Spesifikasi proses ditampilkan dalam bentuk form, logika proses yang ditulis pada spesifikasi proses harus jelas [19]. Berikut adalah tabel 2.1 bentuk contoh form dari spesifikasi proses:

**Tabel 2.1 Form Spesifikasi Proses**

| No. | Proses | Keterangan |
|---|---|---|
| 1. | No Proses | (Menyatakan nomor proses) |
| | Nama Proses | (Menyatakan nama proses) |
| | Sumber | (Menyatakan sumber data yang menuju ke proses) |
| | Input | (Menyatakan isi data yang masuk ke proses) |
| | Output | (Menyatakan informasi yang keluar dari proses) |
| | Tujuan | (Menyatakan tujuan informasi keluaran dari proses) |
| | Logika Proses | (Menyatakan algoritma dari proses) |

### 2.2.13 Kamus Data

Kamus data adalah deskripsi formal mengenai seluruh komponen-komponen yang ada pada data flow diagram. Pada tahap perancangan, komponen-komponen pada kamus data akan menjadi bahan untuk menyusun basis data [20].

### 2.2.14 *Object Oriented Analysis Design*

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [21]

### 2.2.15 *Unified Modelling Language* (UML)

*Unified Modeling Language* (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka [22]. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah :

a. *Use case Diagram*

   *Use case* adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case diagram* memperlihatkan hubungan diantara pengguna dan fungsionalitas sistem.

b. *Activity Diagram*

   *Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir

c. *Class diagram*

*Class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem

d. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior.

e. *Statechart Diagram*

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima.

f. *Conceptual diagram*

*Conceptual diagram* menggambarkan konsep-konsep yang ada di dalam aplikasi.

g. *Collaboration Diagram*

*Collaboration Diagram* menggambarkan interaksi antar objek.

h. *Object Diagram*

*Object Diagram* menggambarkan struktur objek.

i. *Component Diagram*

*Component Diagram* menggambarkan komponen objek.

j. *Deployment Diagram*

*Deployment Diagram* menggambarkan distribusi aplikasi.

### 2.2.16 Wawancara

Wawancara adalah metode pengumpulan data yang melibatkan penyajian pertanyaan dan jawaban secara lisan. Metode ini dapat digunakan melalui wawancara pribadi secara langsung dan, jika mungkin, melalui wawancara telepon.

Wawancara memiliki beberapa jenis metode. Berikut di bawah ini merupakan beberapa jenis metode wawancara: [23]

1. Wawancara Personal

Metode wawancara personal yaitu metode wawancara yang dilakukan dengan cara tatap muka. Wawancara dapat dilakukan secara terencana dengan berpedoman pada daftar pertanyaan yang telah dipersiapkan sebelumnya, maupun tidak.

2. Wawancara Telepon

   Metode wawancara telepon yaitu metode wawancara yang dilakukan melalui telepon. Metode ini memiliki keuntungan lebih flexibel dilakukan dibandingkan melalui surat, lebih cepat, dan lebih murah dibandingkan dengan wawancara personal.

### 2.2.17 Kuesioner

#### 2.2.17.1 Pengertian Kuesioner

Terdapat beberapa pengertian kuesioner menurut para ahli. Berikut ini merupakan beberapa pengertian kuesioner menurut para ahli, yaitu:

1. Menurut Nazir, *kuesioner* atau *daftar pertanyaan* adalah sebuat set pertanyaan yang secara logis berhubungan dengan masalah penelitian, dan tiap pertanyaan merupakan jawaban-jawaban yang mempunyai makna dalam menguji hipotesis. [24]
2. Menurut Suharsimi Arikunto, kuesioner/angket adalah pernyataan tertulis yang digunakan untuk memperoleh informasi dari responden dalam arti laporan tentang pribadi atau hal-hal yang ia ketahui. [25]
3. Menurut Sugiyono kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan secara tertulis kepada responden untuk dijawabnya. [26]

Dengan demikian dapat disimpulkan bahwa *angket/kuesioner* adalah daftar pertanyaan yang disiapkan oleh peneliti dimana tiap pertanyaannya berkaitan dengan masalah penelitian. Angket tersebut pada akhirnya diberikan kepada responden untuk dimintakan jawaban.

Menurut pendapat Sutrisno Hadi, anggapan yang perlu dipegang oleh penelitian dalam menggunakan metode interview dan juga kuesioner adalah sebagai berikut: [27]

1. Bahwa subyek (responden) adalah orang yang paling tahu tentang dirinya sendiri
2. Bahwa apa yang dinyatakan oleh subyek kepada peneliti adalah benar dan dapat dipercaya
3. Bahwa interpretasi subyek tentang pertanyaan-pertanyaan uang diajukan peneliti kepadanya adalah sama seperti yang dimaksud oleh peneliti

Menurut Suharsimi Arikunto, terdapat beberapa keuntungan yang dapat diperoleh dengan menggunakan angket, antara lain: [25]

1. Tidak memerlukan hadirnya peneliti
2. Dapat diberikan secara serempak kepada banyak responden
3. Dijawab oleh responden menurut kecepatan masing-masing dan menurut waktu senggang responden
4. Dapat dibuat anonim sehingga responden bebas, jujur, dan tidak malu-malu menjawab
5. Dapat dibuat berstandar sehingga semua responden dapat diberi

#### 2.2.17.2 Jenis-Jenis Pertanyaan Dalam Kuesioner

Ada tiga jenis pertanyaan dalam kuesioner, yakni pertanyaan tertutup, terbuka, dan gabungan tertutup dan terbuka.

1. Pertanyaan dengan jawaban tertutup

   Pertanyaan jenis ini merupakan pertanyaan dimana semua alternatif jawaban responden sudah disediakan oleh peneliti. Responden tinggal memilih alternatif jawaban yang dianggapnya sesuai. Contohnya yaitu sebagai berikut:

| |
|---|
| Atasan Anda mendelegasikan tugas dengan jelas:<br>1. Sangat Setuju Sekali<br>2. Sangat Setuju<br>3. Setuju<br>4. Tidak Setuju<br>5. Sangat Tidak Setuju |

2. Pertanyaan dengan jawaban terbuka

   Pertanyaan jenis ini merupakan pertanyaan yang memberikan kebebasan penuh kepada responden untuk menjawabnya. Disini peneliti tidak memberikan satupun alternatif jawaban. Contohnya yaitu sebagai berikut:

A. Sebutkan lima sifat pemimpin yang Anda sukai:

1. ……
2. ……
3. ……
4. ……
5. ……

B. Bagaimana pendapat Anda tentang kepemimpinan supervisor Anda?

_____

_____

_____

_____

### 2.2.17.3 Merancang Kuesioner Penelitian

Merancang yaitu mendesain kuesioner yaitu suatu cara untuk membuat atau menyusun suatu kuesioner yang baik yang dapat mengumpulkan informasi sesuai dengan tujuan penelitian. Bagian-bagian kuesioner ada 2 bagian pokok yaitu :

1. Bagian Pengantar,

   Bagian pengantar terdiri atas penjelasan, tujuan kuesioner, dan identitas peneliti (dibuat cover). Tujuan kuesioner dibuat untuk menjelaskan pada responden mengenai tujuan pemberian kuesioner atau tujuan penelitian, data apa yang diharapkan, manfaat apa yang dapat diperoleh masyarakat dari hasil penelitian, kewenangan peneliti, dan kerahasiaan jawaban responden.

2. Bagian Isi

   Bagian isi terdiri atas identitas umum responden dan pertanyaan-pertanyaan utama. Identitas umum responden harus memuat hanya data mengenai responden yang

benar-benar diperlukan saja. Sedapat mungkin jangan menanyakan hal-hal yang sensitif apabila tidak benar-benar merupakan bagian dari data yang perlu dikumpulkan. Misalnya pertanyaan mengenai agama atau status perkawinan yang kalau tidak memang perlu sebaiknya tidak ditanyakan. Diluar identitas umum responden, adalah inti dari kuesioner itu sendiri yang berupa pertanyaan-pertanyaan mengenai data pokok yang diperlukan penelitian, yang kerangkanya telah lebih dahulu dibuat berdasarkan tujuan penelitian dan prasurvei sebelumnya.

#### 2.2.17.4 Uji Coba Instrumen Kuesioner

Sebelum kuesioner disebarkan kepada responden, ujicobakanlah lebih dahulu kepada sejumlah kecil responden. Ini gunanya untuk mengetahui validitas dan reliabilitas alat ukur dimaksud. Selain itu, ini juga bisa digunakan untuk mengetahui kemungkinan diterima atau ditolaknya hipotesis yang telah dirumuskan. Selain itu, jika ternyata dalam uji coba ini terdapat banyak kesalahan, maka peneliti bisa mengubah atau menyempurkannya.

#### 2.2.17.5 Hal Yang Perlu Diperhatikan Dalam Merancang Kuesioner

Ada beberapa hal yang harus diperhatikan dalam merancang kuesioner. Berikut di bawah ini merupakan beberapa hal yang perlu diperhatikan dalam merancang suatu kuesioner, yaitu;

1. Gunakan kalimat yang pendek dan sederhana (mudah dipahami)
2. Gunakan kalimat yang efektif (tidak mengulang kalimat)
3. Jumlah pertanyaan disesuaikan dengan kebutuhan (tidak terlalu banyak)
4. Pastikan bahwa responden mempunyai pengetahuan yang diperlukan
5. Pertanyaan detail digunakan sesuai dengan porsinya
6. Hindari isu yang sensitif
7. Meminimalisir bias