

## **BAB 2**

### **TINJAUAN PUSTAKA**

Pembangunan sistem aplikasi lembur kerja yang ditujukan untuk PT Motiolabs Digital Indonesia ini dibutuhkan Tinjauan Pustaka untuk membantu proses pembuatan aplikasi. Berikut beberapa tinjauan pustaka yang digunakan dalam pembuatan aplikasi:

#### **2.1 Profile PT Motiolabs Digital Indonesia**

PT Motiolabs Digital Indonesia merupakan anak perusahaan PT. Swamedia Informatika, pengembang perangkat lunak kustom ternama yang berlokasi di Jl. Sidomulyo no 31, Sukaluyu, Bandung, Jawa Barat. Perusahaan ini berdiri sejak tahun 2016, dengan misi inti membantu perusahaan dalam perjalanan transformasi digital.

##### **2.1.1 Visi dan Misi**

Berikut merupakan Visi dan Misi dari PT Motiolabs Digital Indonesia :

###### **1. Visi**

Memberikan solusi teknologi untuk mendorong kemajuan transformasi digital bagi entitas bisnis di Indonesia.

###### **2. Misi**

- a. Menyediakan solusi digital dan layanan pendukung untuk beberapa industri utama di Indonesia.
- b. Menjadi perusahaan yang disegani dan berkembang pesat dengan tetap menjaga profitabilitas.
- c. Berkontribusi secara aktif untuk mengatasi tantangan klien kami dalam menerapkan transformasi digital.
- d. Menjadi organisasi tempat karyawan dapat berkembang dan bekerja dengan nyaman.
- e. Memberikan dampak dan kontribusi positif bagi komunitas, masyarakat, dan Indonesia.

### 2.1.2 Struktur Organisasi

Berdasarkan hasil wawancara dengan direktur PT Motiolabs Digital Indonesia, pada gambar 2.1 merupakan struktur organisasi dari PT Motiolabs Digital Indonesia, susunan per Februari 2023 :



**Gambar 2.1 Struktur Organisasi PT Motiolabs Digital Indonesia**

Berdasarkan struktur organisasi pada Gambar 2.1 digambarkan secara sederhana dimana ada 6 bagian dibawah direktur yaitu aplikasi, platform, branding, operation, people dan partnership

### 2.1.3 Logo PT Motiolabs Digital Indonesia

Berikut pada Gambar 2.2 merupakan logo yang digunakan oleh PT Motiolabs Digital Indonesia:



**Gambar 2.2 Logo PT Motiolabs Digital Indonesia**

### 2.1.4 Prosedur Lembur kerja PT Motiolabs Digital Indonesia

1. Perusahaan berhak untuk menugaskan karyawan untuk melakukan pekerjaan di luar waktu kerja tersebut sebagaimana dimaksud dalam pasal 4 ayat (2) untuk menyelesaikan pekerjaan yang tertunda dan/atau pekerjaan yang mendesak yang diperhitungkan sebagai waktu kerja lembur dan karenanya karyawan berhak untuk memperoleh upah lembur ("Waktu Lembur kerja").

2. Waktu lembur kerja hanya dapat dilakukan paling banyak 3 (tiga) jam dalam 1 (satu) hari dan 14 (empat belas) jam dalam 1 (satu) minggu, tidak termasuk waktu lembur kerja yang dilakukan pada waktu istirahat mingguan pada hari sabtu dan hari minggu.
3. Karyawan yang melakukan lembur kerja berhak untuk memperoleh upah lembur kerja yang perhitungannya dilakukan setiap bulan berjalan bersamaan dengan perhitungan dan pembayaran gaji dengan ketentuan besarnya upah lembur kerja adalah sebesar upah lembur kerja sesuai pelaksanaan lembur kerja dalam 1 (satu) bulan dengan ketentuan sebagai berikut:
  - a. Waktu lembur kerja pada hari kerja dihitung sejak 30 (tiga puluh) menit setelah karyawan memenuhi waktu kerja dalam 1 (satu) hari sebanyak 8 (delapan) jam.
  - b. Besaran upah lembur kerja

Waktu Lembur kerja			Upah Lembur kerja (per jam)
Hari Kerja	17:30	23:00	10000
	23:01	08:00	20000
Istirahat Mingguan & Libur Resmi	08:00	17:00	15000
	17:00	08:00	25.000

- c. Karyawan yang melakukan lembur berhak untuk mendapatkan uang makan dan uang transportasi dengan ketentuan sebagai berikut:

Waktu Lembur kerja	Uang Makan (per hari)	Uang Transfortasi (per hari)
Hari Kerja	20000	-
Istirahat Mingguan & Libur Resmi	20000	15000

- d. Lembur kerja posko, yaitu lembur kerja yang dilakukan oleh karyawan pada libur hari raya idul fitri dan akhir tahun yang ditentukan oleh pemerintah dengan ketentuan sebagai berikut:
- i. Posko *Oncall*, yaitu lembur kerja posko yang dilakukan diluar lingkungan perusahaan dan karyawan berhak untuk mendapatkan upah lembur kerja sebesar 200.000,- (dua ratus ribu rupiah) per hari
  - ii. Posko *Onsite*, yaitu lembur kerja posko yang dilakukan dilingkungan perusahaan dan karyawan berhak mendapatkan upah lembur kerja dengan ketentuan sebagai berikut:

Posko <i>Onsite</i>	Upah Lembur kerja (per hari)
<i>Onsite</i> hari H	300000
<i>Onsite</i> selain hari H (libur yang ditetapkan sebagai cuti bersama oleh pemerintah	250000

- e. Dalam hal karyawan menolak lembur kerja yang ditugaskan oleh perusahaan akan dianggap pelanggaran tata tertib perusahaan.
- f. Dalam perhitungan upah, dihitung berdasarkan jumlah jam yang sudah dilakukan, dan upah akan dihitung jika lebih dari 30menit akan dianggap 1jam.

## 2.2 Lembur kerja

Lembur kerja merupakan pekerjaan karyawan yang dilakukan pada hari libur mingguan karyawan atau pekerjaan yang melebihi jam kerja atas perintah atasan. Dalam perhitungan jam lembur kerja setiap perusahaan pasti berbeda-beda. Secara Yuridis dalam Pasal 1 angka 1 Dalam Ketentuan Keputusan Menteri Tenaga Kerja Dan Transmigrasi Republik Indonesia Nomor KEP.102 /MEN/VI/2004 Tentang Waktu Lembur kerja Dan Upah Lembur kerja Waktu lembur kerja adalah waktu kerja yang melebihi 7 (tujuh) jam sehari dan 40 (empat puluh) jam 1 (satu) minggu untuk 6 (enam) hari kerja dalam 1 (satu) minggu atau 8 (delapan) jam sehari, dan 40 (empat puluh)

jam 1 (satu) minggu untuk 5 (lima) hari kerja dalam 1 (satu) minggu atau waktu kerja pada hari istirahat mingguan dan atau pada hari libur resmi yang ditetapkan Pemerintah [10].

### **2.3 Sistem**

Sistem merupakan sebuah rangkaian kerja dari hasil penggabungan elemen - elemen yang saling berguna dengan tujuan tertentu. Dalam sebuah sistem, elemen-elemen yang terlibat saling berkaitan dan memiliki hubungan yang erat. Setiap elemen dalam sistem memiliki peran dan fungsi spesifik yang berkontribusi terhadap keseluruhan tujuan sistem tersebut. Penggabungan elemen-elemen ini dilakukan dengan tujuan menciptakan sinergi dan keterkaitan yang meningkatkan kinerja dan hasil dari sistem secara keseluruhan [11].

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu :

- a. Setiap sistem terdiri dari unsur-unsur.
- b. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
- c. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
- d. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

### **2.4 Absensi**

Absensi adalah sebuah proses yang dilakukan untuk mengumpulkan informasi terkait kehadiran dan ketidakhadiran karyawan perusahaan. Tujuan utama dari kegiatan ini adalah untuk memperoleh data yang akurat mengenai kehadiran setiap karyawan di perusahaan tersebut. Melalui proses absensi, suatu perusahaan dapat mencatat waktu kedatangan serta kepergian karyawan secara teratur. Data yang terkumpul ini dapat mencakup informasi tentang jam masuk dan jam pulang, serta jumlah jam kerja yang telah dilakukan oleh setiap individu.

Data tersebut akan digunakan untuk menghitung upah karyawan sesuai dengan jam kerja yang telah dilakukan, serta untuk memantau tingkat

kehadiran dan ketidakhadiran karyawan secara keseluruhan. Selain itu, dengan adanya informasi yang akurat mengenai kehadiran karyawan, perusahaan juga dapat mengambil tindakan yang tepat dalam menangani kehadiran yang tidak teratur atau keterlambatan yang berulang [12].

## **2.5 Karyawan**

Karyawan merupakan elemen berharga yang dimiliki oleh perusahaan dan menjadi salah satu faktor kunci dalam mencapai tujuan perusahaan. Mereka bukan hanya sumber daya manusia, tetapi juga dianggap sebagai aset berharga yang memiliki peran sentral dalam kesuksesan operasional dan pertumbuhan perusahaan.

Sebagai aset, karyawan memberikan kontribusi nyata dalam berbagai aspek bisnis. Mereka adalah tenaga kerja yang memiliki pengetahuan, keterampilan, dan pengalaman yang relevan dengan bidang pekerjaan mereka. Selain itu, karyawan juga membawa nilai-nilai perusahaan, seperti dedikasi, integritas, kerjasama tim, dan inisiatif yang berdampak pada budaya perusahaan.

## **2.6 Web**

*Web* adalah sekumpulan halaman yang terdapat dalam suatu domain atau subdomain yang ada di *World Wide Web (WWW)*, sebuah jaringan yang meluas di internet. Setiap halaman *web* merupakan dokumen yang dibuat menggunakan format *HTML (Hyper Text Markup Language)* dan terdiri dari berbagai komponen seperti teks, gambar, suara, dan animasi. Saat ini, hampir semua *website* dapat diakses melalui protokol HTTP, yang bertugas menyampaikan informasi dari server website kepada pengguna melalui web browser. Tujuan utama pembuatan sebuah website adalah untuk memfasilitasi interaksi dan keterhubungan antara pengguna dengan informasi yang disajikan, serta memungkinkan akses yang luas bagi para pengguna. *Website* juga digunakan sebagai sarana menyajikan informasi mengenai perusahaan atau topik lainnya yang relevan, sehingga dapat berfungsi sebagai sistem informasi yang berguna. [13][14].

## 2.7 Internet

Internet merupakan jaringan komputer yang dapat digunakan secara luas dan dapat terhubung dari negara ke negara lain. Pada awalnya internet maupun website hanya digunakan untuk kepentingan militer dalam mengirim pesan lewat satelit. Akan tetapi seiring berkembangnya teknologi, akhirnya meluas ke berbagai bidang dan pada saat ini internet sudah populer dengan telepon. Informasi yang dikirimkan lewat internet dapat diakses di seluruh dunia hanya dengan menggunakan browser [14].

Penjelasan lain dari Internet (*Interconnected Networking*) merupakan sekumpulan komputer yang terhubung menjadi satu dalam sebuah jaringan komputer yang berbeda di seluruh dunia. Dalam Internet, pengguna dapat mengakses berbagai layanan dan sumber daya yang tersedia secara online, seperti situs web, email, media sosial, aplikasi, dan banyak lagi. Internet juga menjadi fondasi bagi teknologi seperti streaming video, komunikasi real-time, perdagangan elektronik, dan komputasi awan, yang semuanya memberikan dampak yang signifikan terhadap cara kita bekerja, berkomunikasi, dan menjalani kehidupan sehari-hari. [15].

## 2.8 Visual Studio Code (VS Code)

*Visual Studio Code* merupakan *text editor* yang paling populer dikalangan *developer* dikarenakan banyak fitur yang sangat membantu. Text editor ini dikembangkan oleh *Microsoft* untuk sistem operasi multiplatform, bisa digunakan di *windows*, *Linux*, dan *macOS*. Text editor ini mendukung bahasa pemrograman *Javascript*, *Typescript*, dan *Node.js* serta bahasa pemrograman lainnya [16].

*Visual Studio Code* memiliki banyak fitur yang dapat membantu para *developer*, seperti *Intellinse*, *Git Integration*, *Debugging*, dan fitur yang bisa mempercepat kemampuan text editor. Fitur tersebut akan terus bertambah dan berkembang seiring versi yang di perbaharui. Versi *Visual Studio Code* ini di perbaharui secara berkala setiap bulan. *Visual Studio Code* ini bersifat *open source*, yang mana sumber *code* dapat dilihat dari kontribusi pengembangnya [17].

## **2.9 Hypertext Markup Language ( HTML )**

*HTML* kepanjangan dari (*Hyper Text Markup Language*) merupakan sebuah bahasa markup yang digunakan untuk untuk menulis halaman web. Kode *HTML* membantu para programmer untuk memastikan format teks dan gambar yang benar di situs web yang dibangun. Tanpa adanya *HTML* dalam pembangunan website, browser tidak dapat menampilkan struktur dan format situs web dengan benar. Selain menyediakan struktur dasar halaman web, *HTML* sering digunakan bersamaan dengan *Cascading Style Sheets (CSS)* untuk mengontrol tampilan konten situs web ketika digunakan bersama dengan bahasa lain seperti *JavaScript*, *HTML* dapat membuat semua jenis situs *web*, dari yang paling sederhana hingga yang paling kompleks. Elemen-elemen ini digunakan untuk menentukan tampilan, tata letak, dan fungsi file *HTML* [18].

## **2.10 JavaScript**

*JavaScript* merupakan bahasa pemrograman dengan kumpulan *script* yang berjalan pada dokumen *HTML*. Javascript digunakan untuk menyempurnakan tampilan dan sistem pada halaman web agar menjadi dinamis. Beberapa karakteristik dari bahasa javascript seperti bahasa pemrograman berjenis *high-level programming*, bersifat *client-side*, berorientasi pada objek, bersifat *loosely typed*.

*JavaScript* awal mulanya ditemui oleh Brendan Eich dari Netscape dibawah nama Mocha pada tahun 1995, setelah itu dikembangkan untuk Netscape 2, serta jadi standar ECMA-262 pada tahun 1997. Setelah ECMA menerima javascript dari Netscape, perkembangan javascript terus berkembang pada browser firefox [19].

## **2.11 Framework VueJS**

*VueJS* merupakan salah satu *framework javascript* yang banyak digunakan oleh programmer untuk mengembangkan aplikasi dengan mudah dan cepat. Fungsi dari framework *VueJS* ini adalah menerima dan mengirim data, kemudian ditampilkan dalam bentuk *UI (User Interface)*. Penggunaan *vuejs* mudah untuk di integrasikan dengan *library* yang lain. Menginisialisasi



komponen *HTML* dengan *VueJS* bisa menggunakan *id*, *class*, atau *name*. Pertama kali *VueJS* dikembangkan oleh Evan You pada tahun 2013 dan pertama rilis tahun 2014. Ada banyak fitur didalam *Vue js* yang banyak digunakan, seperti *components*, *computed property*, *data binding*, dll. *Sintaks* *Vue js* dianggap lebih mudah untuk dipelajari daripada *framework* lain seperti *ReactJS*, karena sintaksnya adalah *HTML* dan lebih terlihat seperti *HTML*, *CSS* dan *Javascript* seperti halnya halaman *web* tradisional [20].

### **2.12 PostgreSQL**

*PostgreSQL* merupakan salah satu dari database besar yang menawarkan skalabilitas, keluwesan, dan kinerja yang tinggi. Banyak didukung bahasa pemrograman dan penggunaannya sudah meluas di berbagai *platform*. Bagi masyarakat IT (Teknologi Informasi) di Indonesia, *PostgreSQL* sudah digunakan untuk berbagai aplikasi seperti *web*, *billing system*, dan sistem informasi besar lainnya. Pengenalan *PostgreSQL* merupakan manajemen database dengan skala besar yang memiliki kinerja dan skalabilitas yang tinggi. Penggunaan *PostgreSQL* banyak didukung oleh bahasa pemrograman. Di Indonesia, *PostgreSQL* banyak digunakan di berbagai platform, seperti *web*, sistem *billing*, dan yang lainnya. Berdasarkan pada *PostgreSQL* versi 4.2, *PostgreSQL* merupakan sebuah *Object Relational Database Management System (ORDBMS)* yang dikembangkan di universitas California pada *Berkeley Computer Science Department* [21].

### **2.13 Framework ExpressJS**

*ExpressJS* adalah sebuah *framework javascript* yang digunakan dari sisi *back-end* untuk membangun sebuah aplikasi web. Framework ini banyak digunakan di kalangan *developer* untuk membangun web dari sisi *back-end*. *Expressjs* merupakan *framework* minimal yang sangat fleksibel. Pemakaian *ExpressJS* dapat membuat website server *HTML*, server file statik, aplikasi chat, search engine, sosial media, layanan website dengan akses lewat *REST API* ataupun aplikasi hybrid ialah tidak hanya pengguna mempunyai akses lewat *REST API* pula memiliki akses ke *HTML page* [22].

### **2.14 Framework Tailwind CSS**

*Tailwind CSS* merupakan kerangka kerja *CSS* untuk membangun antarmuka dengan cepat yang didalamnya berisi kumpulan *utility classes*. Selain itu *tailwind* memberikan kemudahan kepada developer untuk mengubah suatu nilai pada *class* ketika nilai atau value pada *class* yang disediakan oleh *tailwind* tidak tersedia hal ini dapat dilakukan dengan cara penggunaan *arbitrary value* dimana penulisannya hampir sama dengan *CSS* pada umumnya hanya saja ditulis pada *class* yang disediakan oleh *tailwind* dengan menggantikan value dengan tanda kurung kotak yang didalamnya dapat diisi value yang kita inginkan. Berbeda dengan framework *CSS* lainnya, *tailwind* tidak memiliki tema *default* maupun *component UI (User Interface)* bawaan, agar pengguna bisa membuat dan mengubah tampilan *UI* sesuai dengan keinginan sendiri. *Tailwind CSS* resmi dirilis dengan versi alpha pada tanggal 1 November 2017 oleh Adam Wathan sebagai CEO dan rekannya Steve Schoger dan sampai saat ini *tailwind CSS* sudah mencapai versi 3.2.4 [23].

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

*Application Programming Interface* merupakan sebuah konsep pemrograman aplikasi untuk dapat diakses dan digunakan oleh pihak lain tanpa mengubah struktur kode maupun database sistem, dan memudahkan komunikasi antar aplikasi sistem walaupun berbeda platform [24]. Cara kerja *API* ada hubungannya dengan klien dan server, klien merupakan pesan berupa permintaan yang dikirim oleh aplikasi sedangkan server merupakan aplikasi yang mengirimkan respons dari klien. cara kerja *API* terbagi menjadi empat, berikut penjelasannya:

#### **1. API SOAP**

*Simple Object Access Protocol (SOAP)* merupakan standar protokol yang digunakan sistem untuk berkomunikasi melalui *HTTP* dan *XML*. *SOAP* juga digunakan untuk interaksi, membuat, memperbarui, dan menghapus suatu data atau objek

## 2. *API RPC*

Remote Procedure Call (RPC) adalah metode yang digunakan untuk membangun aplikasi client-server. Remote Procedure Call (RPC) adalah metode yang digunakan untuk membangun aplikasi client-server. API ini didasarkan pada pemanggilan prosedur pada server, dan server akan mengembalikan outputnya ke klien

## 3. *API Websocket*

API Websocket adalah pengembangan API web modern yang menggunakan objek JSON untuk mengirimkan data. Websocket API mendukung komunikasi antara klien dan server.

## 4. *API REST*

*Representasi State Transfer (REST)* merupakan dasar dari arsitektur *web* modern. *REST* memberikan panduan yang memungkinkan identifikasi kelemahan dari arsitektur sebelumnya dan validasi ekstensi sebelum penyebaran. *REST* adalah seperangkat kendala arsitektur terkoordinasi yang berusaha untuk meminimalkan latensi dan komunikasi jaringan serta memaksimalkan independensi dan skalabilitas implementasi komponen. Ini dicapai dengan menempatkan kendala pada semantik konektor, berbeda dengan gaya lain yang berfokus pada semantik komponen. Prinsip dasar *REST* adalah menghubungkan jalur komunikasi antar mesin atau aplikasi melalui *HTTP* sederhana. Sebelum adanya *REST*, komunikasi antar mesin atau aplikasi dilakukan dengan menggunakan mekanisme atau protokol *middleware* yang cukup kompleks. Sebelum adanya *REST*, komunikasi antar mesin atau aplikasi dilakukan dengan menggunakan mekanisme *middleware* yang lebih kompleks seperti *DCE*, *CORBA*, *RPC* atau *SOA* [25].

### **2.16 *Google Maps API***

*Google Maps* adalah sebuah layanan peta global virtual gratis dan online yang diberikan oleh *Google*, yang dapat diakses melalui <http://maps.google.com/>. *Google Maps API* merupakan antarmuka aplikasi yang dapat diakses menggunakan Bahasa pemrograman *JavaScript*, sehingga *Google Maps* dapat ditampilkan pada halaman *web* yang sedang dibangun.

Untuk dapat mengakses *Google Maps*, Anda diperlukannya *API Key* terlebih dahulu dengan data pendaftaran berupa nama domain web yang sedang dibangun [26].

Melalui integrasi *Google Maps API*, pengembang dapat menampilkan peta interaktif, mencari lokasi, menampilkan rute, serta menambahkan fungsi-fungsi lainnya yang disediakan oleh *Google Maps*. Hal ini memungkinkan pengguna untuk melihat dan berinteraksi dengan peta secara langsung di dalam halaman web yang sedang dibangun, dengan memanfaatkan fitur-fitur seperti *zoom*, perpindahan peta, dan informasi lokasi yang terkait.

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

Penggunaan *LBS* ini akan digunakan untuk aplikasi web absensi lembur kerja, dimana ketika user menginputkan absensi maka akan otomatis akan tersubmit lokasi lembur kerja karyawan.

*LBS* adalah layanan berbasis lokasi, yaitu sebuah layanan berbasis internet yang berfungsi untuk mencari dengan teknologi *Global Positioning Service (GPS)* dan *Google's cell based location*. Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis. *Location based service* atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan.

#### **2.17.1 Unsur Utama LBS :**

##### 1. Location Manager (API Maps)

Menyediakan perangkat bagi sumber atau source untuk *LBS*, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan atau memanipulasi peta.

##### 2. Location Providers (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API Location* berhubungan dengan data *GPS (Global Positioning System)* dan data lokasi real-time [27].

### 2.18 Data Flow Diagram (DFD)

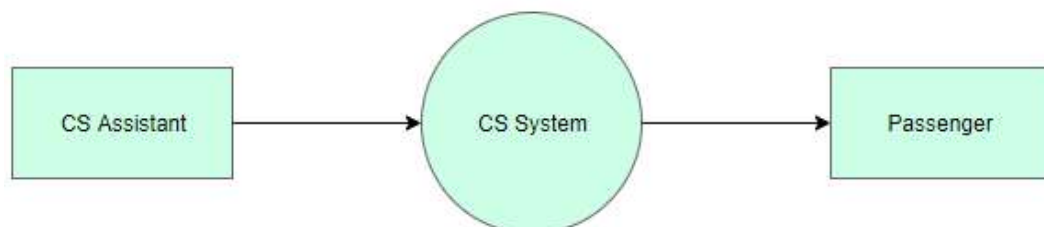
Notasi *DFD*, yang awalnya diperkenalkan dalam penelitian operasional untuk menggambarkan alur kerja dalam sebuah organisasi, berkaitan dengan teori grafik. *DFD* berasal dari *Activity Diagram* yang diterapkan dalam metodologi *SADT (Structured Analysis and Design Technique)* pada akhir tahun 1970-an. Beberapa tokoh terkenal yang turut mempopulerkan *DFD* antara lain *Edward Yourdon, Larry Constantine, Tom DeMarco, Chris Gane, dan Trish Sarson.*

*Data Flow Diagram* adalah sebuah model logika data atau proses yang digunakan untuk menggambarkan asal dan tujuan data yang keluar dari sistem, serta proses dan interaksi yang terjadi pada data tersebut. *Data Flow Diagram (DFD)* digunakan oleh profesional sistem untuk membentuk model sistem yang terdiri dari rangkaian proses fungsional yang saling terhubung melalui jalur data, baik secara manual maupun komputerisasi. *DFD* terdiri dari diagram context kemudian diagram konteks ini diturunkan sampai bentuk yang lebih detail [28].

Berikut merupakan penjelasan notasi yang ada pada *DFD* :

#### 1. Diagram Konteks

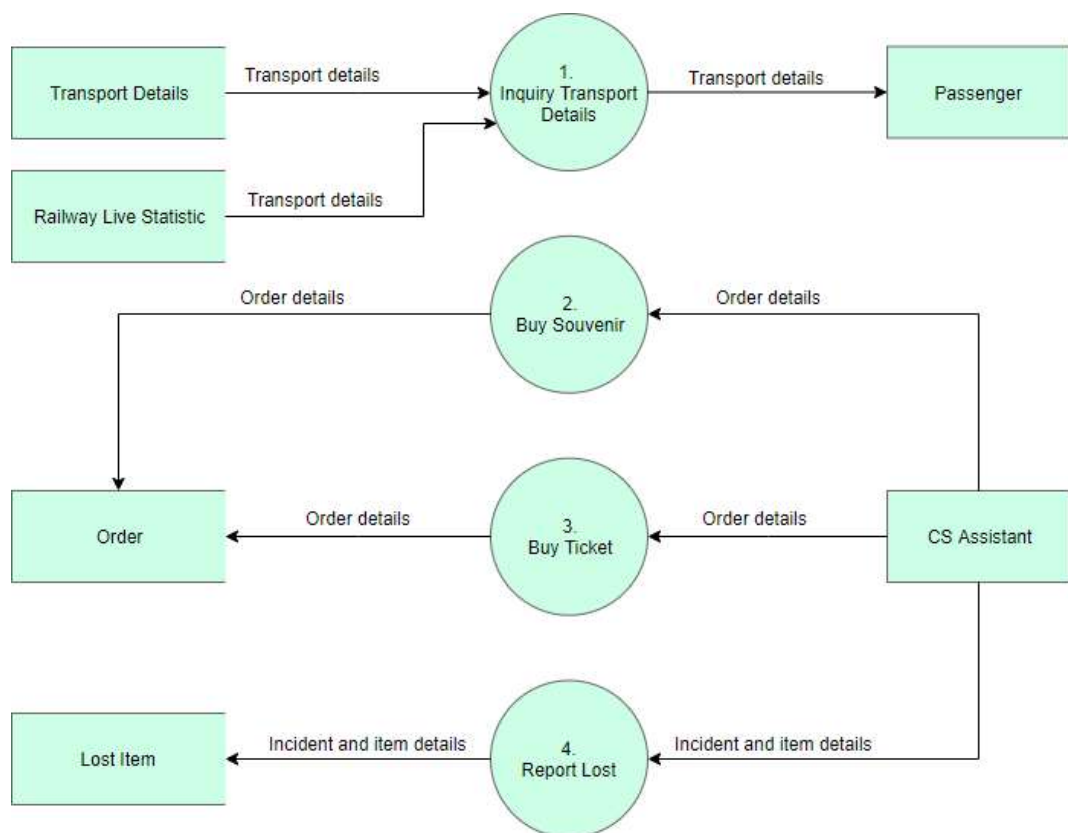
Diagram Konteks Merupakan sebuah gambaran yang terdiri dari satu proses dan menjelaskan lingkup dari suatu sistem. Diagram ini merupakan level tertinggi dari *Data Flow Diagram (DFD)* yang menggambarkan seluruh input yang masuk ke dalam sistem, atau keluaran dari sistem, sehingga memberikan gambaran tentang keseluruhan sistem [29]. Berikut pada Gambar 2.3 merupakan contoh diagram konteks :



**Gambar 2.3 Contoh Diagram Konteks**

## 2. DFD Level 1

DFD Level 1 merupakan rincian detail dari diagram konteks. DFD Level 1 menggambarkan aliran data yang terjadi pada setiap proses di dalam diagram konteks yang dapat dipecah menjadi beberapa subproses. DFD Level 1 juga dapat menggambarkan bagaimana data diproses di dalam sistem secara lebih detail dan dapat membantu dalam memahami sistem dengan lebih mendalam. Berikut pada Gambar 2.4 merupakan contoh DFD level 1 :



**Gambar 2.4 Contoh DFD Level 1**