BAB II
TEORI PENUNJANG

2.1 Kerja Paruh Waktu

Kerja paruh waktu (*Part Time*) menurut Organisasi Buruh Internasional adalah *"a form of employment that carries fewer hours per week than a full-time job. Workers are considered to be part-time if they commonly work fewer than 30 or 35 hours per week"* [3]. Berdasarkan definisi di atas, dapat disimpulkan pengertian kerja paruh waktu adalah suatu pekerjaan yang tugasnya hanya dalam sebagian waktu dari ketentuan waktu kerja atau hari kerja normal, misalnya orang yang ditunjuk staf ahli atau jabatan lain pada suatu perusahaan yang hanya bekerja tiga hari dalam seminggu. Pekerjaan paruh waktu biasanya tidak memiliki kontrak kerja seperti pegawai tetap di kantor dan pekerja paruh waktu bisa membuat kesepakatan kepada pihak yang memberikan pekerjaan mengenai upah, serta fasilitas-fasilitas yang mungkin akan didapat selama mengerjakan pekerjaan tersebut.

2.2 *Employee Seeker*

Employee Seeker dalam bahasa Indonesia diartikan sebagai pencarian pekerja, yaitu proses penentuan/pemilihan pegawai, karyawan, atau pekerja yang dilakukan oleh pencari kerja yang memenuhi syarat dan ketentuan yang ditentukan pencari kerja (Nawawi, 2011) [5]. Seseorang atau suatu badan usaha yang sedang mencari dan memerlukan pekerja bisa masuk sebagai kategori *Employee Seeker.*

2.3 *Employee*

Employee dalam Kamus Besar Bahasa Indonesia (KBBI) diartikan sebagai pekerja, karywan, atau pegawai. Menurut Undang-Undang Nomor 14 Tahun 1969 tentang Ketentuan Ketentuan Pokok Mengenai Tenaga Kerja dalam pasal 1 bahwa tenaga kerja adalah tiap orang yang mampu melakukan pekerjaan baik di dalam

maupun di luar hubungan keja guna menghasilkan jasa atau barang untuk memenuhi kebutuhan masyarakat. Sama halnya Berdasarkan Ketentuan Pasal 1 Angka 2 UU No. 13 Tahun 2003, tenaga kerja adalah setiap orang yang mampu melakukan pekerjaan guna menghasilkan barang dan atau jasa, baik untuk memenuhi kebutuhan sendiri maupun untuk masyarakat (Wijayanti,2010:01) [6].

Pegawai/ karyawan adalah sumber daya manusia/ penduduk yang bekerja di suatu institusi baik pemerintah maupun swasta. Menurut Wirawan, sumber daya manusia merupakan sumberdaya yang digunakan untuk menggerakkan dan mensinergikan sumberdaya lain untuk mencapai tujuan organisasi. Tanpa SDM sumberdaya lain menganggur (*idle*) dan kurang bermanfaat dalam mencapai tujuan organisasi (Abdullah, 2014) [7]. Singkatnya karyawan atau sumber daya manusia (SDM), di satu sisi berfungsi sebagai sumberdaya organisasi disamping sumberdaya-sumberdaya organisasi lainnya (uang, mesin, bahan baku, dan metode) dengan kemampuannya yang leading (berada dimuka) untuk berperan melaksanakan fungsi manajerial (menggerakkan) sumberdaya-sumberdaya organisasi lainnya (uang, mesin, bahan baku, dan metode) (Abdullah, 2014) [7].

Pegawai terbagi menjadi dua macam, pegawai tetap dan pegawai tidak tetap/ tenaga kerja lepas atau yang biasa disebut pekerja paruh waktu (*Part Time*), pekerja paruh waktu adalah orang yang mencari pekerjaan tambahan untuk memenuhi kebutuhan pribadi, pekerja paruh waktu biasanya adalah orang orang yang sedang menempuh pendidikan yang memenuhi syarat usia, seperti mahasiswa dan pelajar.

2.4 Unified Modelling Language (UML)

Unified Modelleing Language atau disingkat UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen- komponen

yang diperlukan dalam sistem software. [8] Menurut Nugroho (2010:6), “UML (Unified Modeling Language) adalah perangkat lunak yang berparadigma “berorientasi objek”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami”. [9]

2.4.1 Bagian – bagian dari UML

Berikut ini adalah bagian – bagian dari Unified Modelling Language, antara lain:

1. **View**

   View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. Ada beberapa jenis view dalam UML antara lain sebagai berikut:

   a. Use Case View

   Use Case View adalah bentuk fungsionalitas dari sistem yang diinginkan oleh user (dalam hal ini aktor). View ini digambarkan dalam user case diagram dan seringkali juga dibuat dalam class diagram. View digunakan dalam pelanggan, perancang, pengembang dan penguji sistem.

   b. Logical View

   Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object dan ralationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. View digunakan untuk perancang dan pengembang.

   c. Component View

   Mendeskripsikan implementasi dan ketergantungan modul. Komponen ini ialah tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrasi lainnya. View digunakan untuk pengembang.

d. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat dan bagaimana hubungannya dengan lainnya. View digunakan untuk pengembang, pengintegrasi dan penguji.

2. Diagram

Diagram ialah presentasi grafis dari sekumpulan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu. Adapun jenis diagram antara lain:

a. Use Case Diagram

Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case ialah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol namun dapat juga dilakukan dalam activity diagrams. Use case digambarkan hanya dilihat dari luar actor dan bukan fungsi yang ada di dalam sistem.

b. Class Diagram

Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara, yaitu: associated, dependent, specialed atau package.

c. Statechart Diagram

Menggambarkan semua state yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik.

d. Sequence Diagram

Menggambarkan semua state yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object

lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequnce diagram. Dalam menunjukkan pertukaran pesan, collaboration diagram menggambarkan object dan hubungannya.

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

g. Component Diagram

Menggambarkan alokasi semua kelas dan object kedalam komponen-komponen dalam desain fisik sistem software. Diagram ini memperlihatka pengaturan dan ketergantungan antara komponen-komponen software seperti source code, binary code dan komponen yang tereksekusi.

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat satu sama lain dan jenis hubungannya.

2.5 Algoritma Haversine

Formula Haversine adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Teorema haversine digunakan untuk menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur (lattitude) dan garis lintang (longitude). Istilah Haversine ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Ríos menggunakan haversine pertama kali dalam penelitiannya tentang “Masalah Utama

Astronomi Nautical" Proc. Royal Soc, Dec 22. 1796. Haversine digunakan untuk menemukan jarak antar bintang. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi.

Menurut Widiyatmoko (2010) dalam penelitian yang berjudul "Pemanfaatan Geolocation dan Haversine Formula dalam Perancangan Sistem Informasi Geografis (GIS) (Studi Kasus: Pariwisata Kabupaten Semarang)", formula Haversine merupakan salah satu persamaan yang sangat akurat untuk menentukan jarak antara dua titik di bumi. Formula Haversine menghitung jarak antara dua titik di bumi berdasarkan panjang garis lurus antara dua titik tanpa mengabaikan kelengkungan yang di miliki bumi [10]. Persamaan haversine merupakan sebuah solusi agar lokasi-lokasi terdekat tampak disekitar pengguna dengan mengabaikan bentuk geografis bukit maupun lembah [11].

Berikut merupakan rumus Haversine formula:

Jarak = Acos(Sin(Lat1) x Sin(Lat2) + Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1) x R

Dimana:

Lat1: Nilai Latitude Lokasi Pengguna

Lat2: Nilai Latitude Lokasi yang dituju

Long1: Nilai Longitude Lokasi Pengguna

Long2: Nilai Longitude Lokasi yang dituju

R: Radius Bumi (rata-rata radius = 6,371 kilometer)

Berikut ini merupakan contoh implementasi algoritma Haversine pada bahasa pemrograman Java:

```
public class HaversineDistance {
public static void main(String[] args) {

final int R = 6371;
Double lat1 = Double.parseDouble(args[0]);
Double lon1 = Double.parseDouble(args[1]);
Double lat2 = Double.parseDouble(args[2]);
Double lon2 = Double.parseDouble(args[3]);
Double latDistance = toRad(lat2-lat1);
Double lonDistance = toRad(lon2-lon1);
Double a = Math.sin(latDistance / 2) * Math.sin(latDistance / 2) +
Math.cos(toRad(lat1)) * Math.cos(toRad(lat2)) *
Math.sin(lonDistance / 2) * Math.sin(lonDistance / 2);
Double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
Double distance = R * c;
System.out.println("The distance between two lat and long is::" +
distance);
}
private static Double toRad(Double value) {
return value * Math.PI / 180;
 }
}
```

Contoh perhitungan jarak dua titik dengan menggunakan rumus Haversine formula: Misalkan untuk mencari jarak antara lokasi *Employee Seeker (misal berada di Dago)* terhadap *Employee* (*misal berada di Dipatiukur*) jika diketahui:

1. Latitude 1 (Employee) -6.886776 dan Longitude 1 (Employee) 107.615246.
2. Latitude 2 (Employee Seeker) -6.908358 dan Longitude 2 (Employee Seeker) 107.610745.

Jarak = ACos(Sin(Lat1) x Sin(Lat2) x Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1)) x R

= ACos(Sin(-6.886776 x pi / 180) x Sin(-6.908358 x pi / 180)

+ Cos(-6.886776 x pi / 180) x Cos(-6.908358 x pi / 180)

x Cos((107.610745 x pi / 180) – (107.615246 x pi / 180)) x 6.371

= Acos(0.897524175) x 6.371 = 2.909 km

2.6 Google Maps

Google Maps adalah suatu layanan pemetaan digital bersifat non komersial yang dibuat oleh perusahaan teknologi Google. Layanan ini memberikan citra satelit, pemetaan jalan, potret 360 derajat, keadaan lalu lintas sampai dengan pembuatan rute jalan otomatis. Sebelumnya Google Maps merupakan aplikasi desktop dengan basis kode C++ yang dikembangan oleh perusahaan Where 2 Technologies yang kemudian pada tahun 2004 layanan tersebut diakuisisi oleh Google. Layanan ini menggunakan teknologi AJAX, XML, dan Javascript serta memungkinkan untuk memasukan data peta oleh aplikasi pihak ketiga melalui Google Maps API.

2.7 Google Maps API

API atau Application Programming Interface merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan programmer untuk “membongkar” suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan

programmer menggunakan suatu fungsi dari sebuah sistem. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari HTML, Javascript dan AJAX serta XML, memungkinkan untuk menampilkan peta Google Maps di website lain. Google juga menyediakan layanan Google Maps API yang memungkinkan para pengembang untuk mengintegrasikan Google Maps ke dalam website masingmasing dengan menambahkan data point sendiri. Dengan menggunakan Google Maps API, Google Maps dapat ditampilkan pada web site eksternal.

2.8 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. [13] Android merupakan sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi *mobile*. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013. [14]

2.9 Java

Java merupakan bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk *smartphone*. Awalnya bahasa ini dibuat oleh James Gosling ketika masih bergabung di Sun Microsystems yang saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

2.10 Android Studio

Android Studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi Android, yang dibangung diatas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android. Android studio sendiri pertama kali diumumkan di Google I/O *conference* pada tanggal 16 Mei 2013. Ini merupakan tahap preview dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014. Versi liris stabil yang pertama diliris pada December 2014, dimulai sejak versi 1.0. Sedangkan versi stabil yang sekarang adalah versi 3.13 yang diliris pada Juni 2018. Fitur Fitur yang tersedia saat ini dalam stable version.

2.11 Firebase

Firebase memiliki produk utama, yaitu menyediakan database *realtime* dan *backend* sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* Firebase ini. Firebase menyediakan *library* untuk berbagai client platform yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service*) dengan konsep *realtime*. Semua data Firebase Realtime Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di awan. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. [15]

2.12 JSON

JavaScript Object Notation (JSON) merupakan *lightweight* data *interchange* yang berbasis *JavaScript Programming Language*. JSON berbasis teks/tulisan dengan format yang dapat dibaca dan dikenali oleh manusia untuk merepresentasikan struktur data sederhana dan array asosiatif. JSON merupakan bahasa independen yang lengkap dan menggunakan konvensi yang familiar bagi para programmer bahasa C, antara lain bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python, dan lainnya (JSON,2015). [15]

2.13 Skala Likert

Skala Likert secara umum adalah indikator berganda atau beberapa item dari beberapa rangkai sikap yang berkaitan dengan area tertentu. Tujuan dari skala Likert adalah untuk mengukur intensitas perasaan tentang area yang dimaksud. Secara umum, pengukuran ini terdiri dari serangkaian pernyataan (dikenal sebagai ‘item’) yang berfokus pada isu atau tema tertentu. Jadi dalam pengukuran sebuah sikap menggunakan beberapa item pertanyaan sebagai indikatornya. Setiap

responden kemudian diminta untuk menunjukkan tingkat persetujuannya dengan pernyataan tersebut. Biasanya, format untuk menunjukkan tingkat persetujuan adalah skala lima poin yang berasal dari ‘sangat setuju’ sampai ‘sangat tidak setuju’, selain itu bisa juga menggunakan skala tujuh poin. Format lain juga boleh digunakan. Namun, biasanya ada posisi tengah (netral) dalam setiap item, misalnya ‘ragu-ragu’ atau lainnya yang menunjukkan netralitas. Sehingga jumlah keseluruhan bentuk persetujuan itu adalah ganjil.

Respons dari masing-masing responden pada setiap item diubah menjadi skor, dan kemudian skor dari setiap item dikumpulkan untuk membentuk skor keseluruhan. Dengan kata lain skor dari masing-masing item tersebut dijumlahkan menjadi skor total. Biasanya, karena skala mengukur intensitas, penilaian dilakukan sehingga tingkat intensitas perasaan yang tinggi pada masing-masing indikator mendapatkan skor yang tinggi. Misalnya, pada skala lima poin, skor 5 untuk perasaan positif yang kuat tentang suatu hal dan skor 1 untuk perasaan yang sangat negatif. [16] [17]