

BAB II

LANDASAN TEORI

2.1. Penelitian Terkait

1. Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi.[1]

Sistem absensi saat ini perlu memerlukan pembaharuan sistem yang baru. Karena dengan melakukan hal ini akan berdampak sangat baik bagi perusahaan dilihat dari sisi efisien, efektif, dan cepat. Dalam hal ini perusahaan berlomba membuat sistem absensi yang dapat berjalan pada teknologi saat ini seperti smartphone android. Hal tersebut membutuhkan sebuah design rancangan sebelum sistem ini dibuat sehingga dapat mengetahui permasalahan dan solusi pada sistem yang akan dibuat. Design sistem absensi online digunakan untuk mempercepat absen hanya melalui smartphone android yang dimiliki oleh masing-masing karyawan. Hal ini membutuhkan jaringan lokal yang berjarak hanya didalam lingkungan perusahaan sehingga karyawan tidak dapat absen diluar perusahaan. Guna membuat sebuah laporan dengan metode sistem diatas, memerlukan metode pengumpulan data sistem yang berjalan, wawancara, observasi, analisa sistem yang diusulkan, dan metode prototipe dan testing sebagai pembentukan sistem usulan yang memadai sebagai alternative solusi terbaik dari permasalahan yang ada.

2. Penerapan Sistem Absensi *Online* Berbasis Android (Studi Kasus Pada Kantor Pemerintah Daerah Kabupaten Majalengka Jawa Barat). [2]

Sistem absensi sangat penting untuk menilai kinerja pegawai, terutama dilihat dari sisi kedisiplinan dan tanggung jawab pegawai. Aplikasi absensi *online* berbasis android mulai sering digunakan oleh perusahaan, karena loyalitas dan kinerja pegawai sulit diukur jika sistem absensi yang digunakan oleh perusahaan masih menggunakan sistem absensi manual. Tujuan penelitian ini yaitu untuk mengetahui sistem absensi *online* berbasis android pada Kantor Pemerintah Daerah Kabupaten Majalengka, Jawa Barat. Metode yang digunakan dalam penelitian ini menggunakan kualitatif dengan pendekatan studi kasus. Hasil penelitian menunjukkan bahwa absensi sistem *online* berbasis android diharapkan dapat meningkatkan disiplin dan tanggung jawab pegawai negeri sipil di Kantor Pemerintah Daerah Kabupaten Majalengka Jawa Barat.

3. Aplikasi GIS Menggunakan Metode *Location Based Service* (Lbs) Berbasis Android.[3]

Kota Bandar Lampung adalah salah satu kota yang ada di Provinsi Lampung dan merupakan ibu kota dari Provinsi Lampung. Pada kota Bandar Lampung masih ada lokasi seperti objek wisata, rumah sakit, hotel/penginapan, pendidikan (sma/smk/universitas), restoran, dan mall/pasar tradisional yang belum terexplor sehingga banyak wisatawan lokal maupun non-lokal tidak mengetahui lokasi tersebut. Sehingga Penulis merancang suatu aplikasi berbasis android yaitu aplikasi profil

kota Bandar Lampung (Probal) yang bertujuan untuk membantu atau memudahkan masyarakat kota Bandar Lampung maupun wisatawan yang datang dari berbagai macam daerah untuk dapat lebih mudah menemukan lokasi-lokasi yang ada di kota Bandar Lampung dengan menggunakan metode location based service (LBS).

2.2. Pengertian Absensi

Absensi adalah daftar kehadiran pegawai/siswa, yang berisi jam datang, jam pulang, serta alasan/keterangan kehadiran pegawai. Absensi juga merupakan suatu daftar pendataan kehadiran seseorang dari suatu aktifitas disebuah institusi yang diatur dan disusun sedemikian rupa sehingga dapat digunakan oleh pihak yang berkepentingan. Dimana data dari pegawai yang tidak hadir akan tercatat di daftar kepegawaian dan dapat di periksa kapan saja oleh pihak instansi tersebut. Dalam bahasa inggris, pemakaian kata absen sering digunakan sebagai istilah *List of Absent*, yang berarti seseorang yang tidak hadir dalam suatu pertemuan sedangkan pemakaian kata kehadiran sering digunakan sebagai istilah *List of Presence* atau *List of Participants*.^[4]

2.3. Pengertian 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 perangkat keras, perangkat lunak, dan telekomunikasi yang

bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel *Android* pertama mulai dijual pada bulan Oktober 2008.[5]

Android merupakan sistem operasi untuk telepon seluler yang berbasis *linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. *Android* umumnya digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi *Symbian* di *nokia*, *iOS*, *Apple*, dan *BlackBerry OS*.[5]

2.4. Pengertian *Located Base Service*

Layanan Berbasis Lokasi atau lebih dikenal dengan *Location Based Service* (LBS) istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. LBS adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut[1]. Terdapat dua unsur utama pada LBS yaitu:

a. *Location Manager* (API Maps)

Menyediakan *tools/source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan, memanipulasi maps/peta beserta *feature* lainnya seperti tampilan satelit, *street* (jalan). Paket ini berada pada *com.google.android.maps*.

b. *Location Provider* (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device/perangkat*. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API Location berada pada paket

android yaitu dalam paket *android.location*. Dengan *Location Manager*, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan



Gambar 2.1 Teknologi *Located Base Service*

Location Based Service dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu: *Geographic Information System*, *Internet Service*, dan *Mobile Devices*. Teknologi *Location Based Services* berfokus dari peralatan yang gunakan atau disebut dengan metode *positioning*. Jika berbicara tentang *positioning*, beberapa sistem yang digunakan ini dapat dibedakan dalam tiga jenis, yaitu :

a. Secara Manual

Metode manual adalah cara-cara konvensional yang selama ini sering dilakukan baik yang melalui *yellow pages*, telepon bantuan operator, dan sebagainya. Cara-cara ini tentunya sangat merepotkan dan sudah mulai ditinggalkan. Munculnya internet memberi perspektif baru bagi fasilitas pencarian ini yang lebih luas. Ditambah dengan munculnya komunikasi selular, tentunya mobilitas pun semakin meningkat. Dengan

menggabungkan kedua teknologi ini, sudah pasti metode manual akan ditinggalkan.

b. Melalui GPS (*Global Positioning System*)

GPS merupakan system navigasi radio di seluruh dunia yang memanfaatkan 24 satelit beserta stasiun bumi. Melalui system ini, bumi dibagi menjadi kotak-kotak dengan masing-masing memiliki alamat yang unik sehingga dapat mengidentifikasi dengan tepat setiap lokasi tersebut.

c. Melalui *Cellular Based Station* atau BTS (*Base Transceiver Station*)

Teknologi *Cellular Based Station*, berbasis pada jaringan telekomunikasi selular yang memungkinkan digunakan dalam ruangan atau *indoor*. Sebuah *handphone* dapat ditentukan posisinya saat ini. Dengan menggunakan prinsip *triangulasi*, posisi *handphone* dapat terdeteksi. *Cellular Based Station* memiliki akurasi yang sangat kurang baik jika dibandingkan dengan menggunakan GPS.

2.5. Android Studio

Android Studio adalah sebuah lingkungan pengembangan terpadu IDE (*Integrated development Environment*) untuk mengembangkan pada platform android. *Android Studio* merupakan *software* yang dapat meningkatkan produktivitas dan mempermudah pekerjaan dalam membuat aplikasi *android*. *Android studio* menyediakan berbagai fitur dan peralatan yang sangat dibutuhkan oleh para *developer* (pengembang) dengan pemrograman java. *Android studio* di perkenalkan oleh google secara resmi pada tahun 2013. Dalam penelitian ini penulis menggunakan bahasa java sebagai bahasa pemrograman.[4]

2.6. PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML. PHP merupakan singkatan dari “PHP : *Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML, sekaligus bekerja di sisi *server* (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga *script* tak tampak disisi *client*. PHP dirancang untuk dapat bekerja sama dengan database *server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah.[6]

Tujuan dari bahasa ini adalah untuk membuat aplikasi di mana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Saat ini PHP cukup populer sebagai piranti pemrograman web di lingkungan linux. Walaupun demikian PHP sebenarnya juga dapat berfungsi pada server-server yang berbasis UNIX, Windows, dan Macintosh. Kelebihan PHP dari bahasa pemrograman web, antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dari mulai IIS, *apache*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis dan *developer* yang siap membantu dalam pengembangan.

4. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Macintosh, Windows*) dan dijalankan secara *runtime*.

PHP memiliki 8 (delapan) tipe data yaitu :

a. *Integer*

Integer merupakan tipe data yang berguna untuk menyimpan bilangan bulat, bukan desimal. Sebagai contoh 1,2,-1.

b. *Double*

Double merupakan tipe data berguna untuk menyimpan nilai desimal. Contoh nilai desimal adalah 0,1; 0,2; dll.

c. *Object*

Tipe data *object* bisaanya digunakan pada bilangan, variabel, ataupun fungsi.

d. *Boolean*

Tipe data *boolean* digunakan untuk menyimpan nilai *true* atau *false*, biasanya tipe data ini mayoritas digunakan untuk melakukan pengecekan kondisi. Contohnya pada *if-else statement*.

e. *String*

Tipe data *string* digunakan untuk menyimpan karakter, angka, atau bisa juga digunakan untuk menyimpan kalimat atau gabungan dari angka dan kalimat.

f. *Array*

Tipe data *array* mampu menyimpan lebih dari satu data tetapi tiap element data dalam *array* dibedakan menurut nomor indeks.

g. *Null*

NULL adalah tipe data yang tidak memuat apapun, setiap variabel yang diset menjadi tipe data *NULL* ini akan menjadikan variabel tersebut kosong, seperti penggunaan fungsi *unset ()*.

h. *Resource*

Tipe data *resource* digunakan untuk menyimpan sumber atau alamat. Variabel tersebut hanya diciptakan oleh suatu fungsi yang mengembalikan nilai berupa *resource* seperti *mysql_connect*.

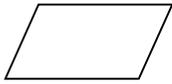
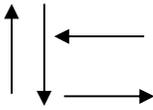
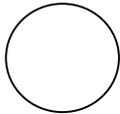
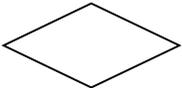
2.7. SQL

SQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*) (Biasta, 2011). Setiap orang bebas untuk menggunakan *MySQL*. *MySQL* Sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data. Keandalan *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah *SQL*, yang dibuat oleh *user* maupun program aplikasinya.[6]

2.8. Konsep Perancangan Flowchart

Flowchart (bagan alir) merupakan representasi secara grafik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah[8]. Adapun simbol yang digunakan adalah sebagai berikut :

Tabel 2.1 Simbol Flowchart

No	Simbol	Keterangan
1		Simbol <i>Input/Output</i> Simbol ini digunakan untuk mewakili data <i>input/output</i> .
2		Simbol Proses Simbol ini digunakan untuk mewakili suatu proses
3		Simbol Alir/Arah Simbol ini digunakan untuk menunjukkan arah/alir dari proses.
4		Simbol Penghubung Digunakan untuk sambungan dari alir yang terputus di halaman yang sama.
5		Simbol Keputusan Simbol yang digunakan untuk suatu penyelesaian kondisi dalam program.
6		Simbol Awal /Akhir (Terminator) Simbol ini digunakan untuk menunjukkan awal dan akhir dari proses.
7		Simbol Persiapan Simbol yang digunakan untuk memberikan nilai awal dari proses.
8		Simbol Predefined Proses Simbol ini digunakan untuk proses yang detailnya terpisah.
9		<i>Off Page Connector Symbol</i> Penghubung pada halaman yang berbeda

2.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang bersifat *general purpose* yang digunakan untuk membangun, dan

mendokumentasikan dari sebuah *system* perangkat lunak. UML menangkap keputusan serta pemahaman mengenai sistem. UML ditujukan untuk penggunaan pada semua metode pengembangan[7].

a. *Use Case Diagram*

Use Case Diagram adalah diagram yang secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

Sebuah *use case* adalah deskripsi statis dari beberapa cara dimana sistem yang digunakan, oleh perusahaan pelanggan, penggunaanya, atau dengan sistem lain. Setiap gelembung pada *use case* merupakan diagram *use case* dan setiap orang mewakili tingkat pengguna.

Tabel 2.2 Simbol *Usecase Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Actor</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Sequence Diagram*

Sequence diagram bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

Tabel 2.3 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi

3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
---	---	----------------	--

c. *Activity Diagram*

Diagram aktivitas menunjukkan dependensi antara (paralel) kegiatan sebagaimana kita bergerak dari titik awal ke tujuan yang diinginkan. Mereka mirip dengan diagram alir, tradisional digunakan untuk aliran model program atau kegiatan manusia.

Tabel 2.4 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana setiap kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

d. *Class Diagram*

Bersifat statis, diagram ini memperlihatkan himpunan kelas, antarmuka, kolaborasi, serta relasi. Diagram ini umum dijumpai pada pemodelan sistem

berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Bentuk dari diagram kelas dapat terlihat pada tabel 2.5 dibawah ini.

Tabel 2.5 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya