

Pengembangan Dan Implementasi Aplikasi Pencarian Lokasi Pelanggan Dan Penjadwalan Teknisi PT. Telkom Kandatel Pati Berbasis Android

Tri Nor Ahfandi¹⁾, Rinta Kridalukmana²⁾

Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro

Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstract— Telkom Group is the only telecommunications company as well as telecommunications service providers and the largest network in Indonesia. At this time the daily scheduling of technicians and customers do not have a location search memadahi information system. In some fields, the management is still done manually. Such as scheduling repairs to the customer given to technicians only reports of damage on the day yesterday so that customers in today's report will have to wait one day plus also a technician who does not mengetahui in detail the work area so that slowing customer service.

Development of information systems in this study using the waterfall method. The waterfall model is composed of: requirements, design, implementation, testing and maintenance. For the beginning of the design, developers need observation or interviews to determine the needs of the system that is being developed. Applications developed using CodeIgniter framework.

The results of this final project is an information system that can meet the needs of PT. Telkom Kandatel Patis in managing daily scheduling service technicians and search customer locations in accordance with existing business processes.

Keywords: *information systems, Android, CodeIgniter, Bootstrap*

I PENDAHULUAN

BUMN (Badan Usaha Milik Negara) merupakan badan usaha yang dimiliki oleh negara untuk memenuhi kebutuhan penduduk sebagai penunjang pengembangan suatu negara. Indonesia memiliki banyak BUMN yang bergerak dalam berbagai bidang. Telkom Group adalah satu-satunya BUMN telekomunikasi serta penyelenggara layanan telekomunikasi dan jaringan terbesar di Indonesia. Di Kabupaten Pati, kebutuhan telekomunikasi dan jaringan berada di bawah naungan PT. Telkom Kandatel Pati. Dengan jumlah pelanggan yang sangat besar, pengelolaan harus dilakukan dengan baik dan rapi. Dalam beberapa bidang, pengelolaan masih dilakukan secara manual. Seperti penjadwalan perbaikan ke pelanggan yang diberikan kepada teknisi hanya laporan kerusakan pada hari kemarin saja sehingga pelanggan yang melapor di hari ini harus menunggu satu hari ditambah juga teknisi yang tidak mengetahui secara detail daerah kerjanya sehingga memperlambat pelayanan terhadap pelanggan.

Menurut artikel internet (tribuntimur) menyatakan rendahnya kualitas pelayanan dapat dipengaruhi oleh beberapa faktor baik itu sumber daya manusia (SDM) maupun sarana dan prasarana yang ada untuk mendukung pelayanan. Rendahnya kualitas sumber daya manusia maupun sarana dan prasarana dapat berdampak kepada rendahnya kualitas pelayanan yang diberikan kepada masyarakat. Sumber daya manusia yang rendah terlihat dari rendahnya kemampuan dan

keterampilan serta kesadaran petugas dalam memberikan pelayanan yang terbaik.

Ilmu pengetahuan serta teknologi informasi dan komunikasi saat ini telah mengalami perkembangan yang sangat pesat. Faktor ketersediaan informasi dan kemudahan dalam mengakses informasi yang dapat diakses dimanapun dan kapanpun inilah yang menjadikan pemanfaatan teknologi diterapkan hampir diberbagai bidang. Informasi dan komunikasi harus dapat diakses dengan mudah. Kemudahan yang ditawarkan inilah yang membuat teknologi diterapkan di hampir setiap aspek kehidupan manusia. Penerapan Aplikasi ini akan sangat membantu perusahaan dalam melakukan penjadwalan kepada teknisi, pencarian lokasi dan pendataan informasi pelanggan.

Dengan adanya penjelasan diatas, maka penulis merancang desain aplikasi penjadwalan dan pencari lokasi pelanggan berbasis PHP dan Mobile yang diharapkan mampu memudahkan teknisi dalam mendapatkan jadwal baru dan pencarian lokasi pelanggan Telkom area Pati sehingga dapat meningkatkan pelayanan perusahaan kepelanggan.

II LANDASAN TEORI

A. Sistem Informasi

Sistem adalah entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai satu tujuan. Informasi merupakan fakta yang memiliki arti sehingga memungkinkan untuk dimanfaatkan. Sedangkan semua fakta tersebut yang tercatat, terekam atau terdokumentasikan disebut data.

Alter^[1] berpendapat untuk sistem informasi sebagai tipe khusus dari sistem kerja. Sistem kerja adalah suatu sistem di mana manusia dan atau mesin melakukan pekerjaan dengan menggunakan sumber daya untuk memproduksi produk tertentu dan atau jasa bagi pengguna. Sistem informasi adalah suatu sistem kerja yang kegiatannya ditujukan untuk pengolahan (menangkap, transmisi, menyimpan, mengambil, memanipulasi dan menampilkan) informasi. Contoh sistem informasi yang mudah ditemui adalah Sistem Informasi Akademik, Sistem Penjualan Toko, dan Sistem Tagihan Listrik.

B. Sistem Basis Data

Data merupakan fakta yang tercatat atau terekam. Segala sesuatu dapat dijadikan data namun data tersebut tidak selalu bermanfaat untuk dikumpulkan. Sebuah data mentah yang baru saja didapatkan bisa saja tidak memiliki arti sama sekali, tetapi setelah diolah data tersebut bisa menjadi informasi yang membantu proses bisnis. Data perlu dikumpulkan data sudah dilakukan sejak ribuan tahun lalu dengan menulis. Sekarang data disimpan dan dikumpulkan dengan bantuan komputer, kumpulan atau koleksi ini dikenal dengan basis data.

Menurut Fathansyah^[2] sistem basis data merupakan sistem yang terdiri atas kumpulan tabel data yang saling

berhubungan (dalam sebuah basis data di sebuah komputer) dan sekumpulan program (yang biasa disebut *Data Base Management System*) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi tabel data tersebut. Contoh dari *Data Base Management System* antara lain SQL Server, MySQL, Oracle dan DB2.

C. Aplikasi Web

Hales^[3] mendefinisikan aplikasi *web* adalah kumpulan dari HTML dokumen, komponen *web* dan sumber daya lainnya yang disusun dalam sebuah hierarki. Aplikasi *web* berada di sebuah *server* yang menyediakan layanan ke berbagai pengguna.

Aplikasi *web* menjadi populer karena kemudahan yang ditawarkan untuk mengakses aplikasi. Akses dapat dijalankan dengan *web browser* yang disebut sebagai suatu *thin client*. Kemampuan untuk memperbarui dan memelihara aplikasi *web* tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada ribuan komputer klien merupakan faktor utama aplikasi *web* semakin populer. Contoh aplikasi *web* yang umum adalah email, toko online, wiki, papan diskusi, log.

D. PHP

Lerdorf^[9] menerangkan PHP adalah bahasa *scripting server-side* yang dirancang untuk pengembangan web, tetapi juga dapat digunakan sebagai bahasa pemrograman yang bertujuan umum. Awalnya dibuat oleh Rasmus Lerdorf pada tahun 1994.

Kode PHP dapat hanya dicampur dengan kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai mesin *template* dan kerangka kerja *web*. Kode PHP biasanya diproses oleh interpreter PHP, yang biasanya diimplementasikan sebagai modul asli *web server* atau *Common Gateway Interface* (CGI) *executable*. Setelah kode PHP ditafsirkan dan dijalankan, *server web* mengirimkan dihasilkan keluaran ke klien, biasanya dalam bentuk bagian dari halaman *web* yang dihasilkan - misalnya, kode PHP dapat menghasilkan kode halaman *web* HTML, gambar, atau beberapa lainnya data. PHP juga telah berkembang untuk menyertakan antarmuka baris perintah (CLI) kemampuan dan dapat digunakan dalam aplikasi grafis mandiri.

Interpreter PHP kanonik, didukung oleh Zend Engine, adalah perangkat lunak gratis yang dirilis dibawah lisensi PHP. PHP telah banyak diimpor dan dapat digunakan pada kebanyakan *server web* pada hampir setiap sistem operasi dan platform, gratis.

Meskipun sangat populer, tidak ada spesifikasi tertulis atau standar yang ada untuk bahasa PHP; sebagai gantinya, PHP interpreter kanonik berfungsi sebagai *standar de facto*.

E. Arsitektur MVC

Freeman^[12] menjelaskan bahwa Arsitektur MVC adalah teknik pengembangan aplikasi *web* dengan dasar membagi konsentrasi dari sebuah aplikasi menjadi beberapa domain. MVC dibagi dalam 3 konsentrasi yaitu: *model* untuk abstraksi data, *controller* untuk logika pengolahan dan *view* untuk antarmuka pengguna. Masing-masing *domain* konsentrasi tersebut dapat bekerja secara terpisah. Tujuan dari pemisahan konsentrasi tersebut adalah memudahkan pengembangan, pengujian dan pemeliharaan. Arsitektur ini dikembangkan sejak akhir 1970 oleh studio Smalltalk.

F. Kerangka Kerja CodeIgniter

Codeigniter adalah sebuah *framework* bahasa pemrograman PHP. Codeigniter diperkenalkan kepada publik pada tanggal 28 Februari 2008. Codeigniter menawarkan kemudahan serta standarisasi dalam proses pengembangan *website* dan aplikasi berbasis *web*. Dengan Codeigniter proses pengembangan *website* menjadi lebih cepat dan terstandar. Standar ini sangat berguna dalam

pengembangan *website* yang dikerjakan oleh *tim*. Codeigniter juga telah menyediakan *library* dan *helper* yang berguna dan mempermudah proses *development*. Terdapat beberapa kelebihan penggunaan Codeigniter dalam pengembangan suatu proyek, yaitu : 1.Menghemat Waktu 2.*Code Reuse* 3.Dukungan Komunitas 4.Kumpulan Best Practice Codeigniter dibangun menggunakan konsep *Model-View-Controller development pattern*, yaitu sebuah teknik pemrograman yang memisahkan logika bisnis, logika data, dan logika presentasi^[13].

G. Kerangka Kerja Ionic

Ionic^[14] adalah kerangka kerja *Mobile HTML5* untuk mengembangkan aplikasi perangkat bergerak lintas-platform *hybrid native* menggunakan HTML, JavaScript dan CSS.

Ionic dikembangkan untuk memudahkan pengembang mengembangkan aplikasi perangkat bergerak *hybrid native* daripada membuat situs perangkat bergerak. Ionic merupakan kerangka kerja yang fokus pada interaksi dengan pengguna melalui HTML5 tetapi dengan kecepatan aplikasi *native*. Ionic tidak menggunakan kerangka kerja CSS Bootstrap maupun *library* JavaScript jQuery mobile melainkan menggunakan kerangka kerja JavaScript angularJS yang mempunyai kecepatan proses lebih cepat dari jQuery mobile.

H. MySQL Server

Huda^[15] menjelaskan bahwa MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh pengguna maupun program-program aplikasinya. Sebagai server basis data, MySQL dapat dikatakan lebih unggul dibandingkan *server* basis data lainnya dalam *query* data.

I. GPS (Global Positioning System)

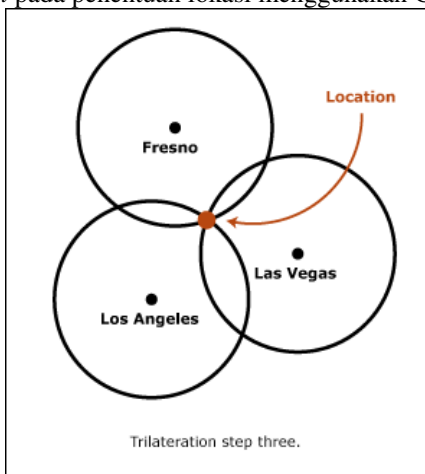
Global Positioning System (GPS) adalah suatu sistem radio navigasi penentuan posisi menggunakan satelit.^[16] GPS dapat memberikan posisi suatu objek di muka bumi dengan akurat dan cepat (koordinat tiga dimensi x,y,z) dan memberikan informasi waktu serta kecepatan bergerak secara kontinyu di seluruh dunia.

Pada awalnya, ada 24 satelit MEO (*Medium Earth Orbit*) yang digunakan untuk membantu melacak posisi, dengan membagi wilayah bumi menjadi 3 bagian yang masing – masing diisi oleh 8 satelit. Dalam perkembangannya, bumi dibagi menjadi 6 bagian dengan 4 satelit melayani per wilayah. Karena mengorbit di MEO (ketinggian rata – rata di permukaan bumi sekitar 20.200 km), maka posisi orbit satelit kadang berubah (bandingkan dengan GEO stasioner yang berada di ketinggian 36.000 km).

Dengan adanya 24 satelit di angkasa, 4 sampai dengan 10 satelit GPS setiap saat akan selalu diamati di seluruh permukaan bumi. Dari wilayah Indonesia, umumnya 6 sampai 9 satelit GPS akan dapat ditangkap dengan sudut diatas 10⁰. Setiap satelit GPS secara kontinyu memancarkan sinyal – sinyal gelombang pada 2 frekuensi *L-band* (dinamakan L1 dan L2). Dengan mengamati sinyal – sinyal dari satelit dalam jumlah dan waktu yang cukup, kemudian data yang diterima tersebut dapat dihitung untuk mendapatkan informasi posisi,

kecepatan, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India.

Khususnya pada perangkat bergerak, dengan perangkat lunak dan paket data dari penyedia telekomunikasi, fitur GPS mampu bekerja untuk memberikan informasi lokasi kita, menunjukkan arah tujuan yang kita kehendaki dan dapat juga memberikan informasi tentang lokasi – lokasi di sekitar kita seperti, hotel, SPBU, dan bahkan ATM. Agar GPS dapat menentukan lokasi kita, paling tidak harus berada dalam jangkauan 3 satelit. Untuk menentukan lokasi secara akurat, penerima sinyal GPS menggunakan prinsip *trilateration*. *Trilateration* merupakan metode penentuan lokasi berdasarkan perpotongan tiga lingkaran. Titik hasil persinggungan ketiga bola inilah yang menjadi lokasi penerima sinyal GPS, atau dalam hal ini merupakan lokasi orang yang membawa alat penerima tersebut. Gambar 1 merupakan gambaran metode *Trilateration* pada penentuan lokasi menggunakan GPS.



Gambar 1. *Trilateration* GPS

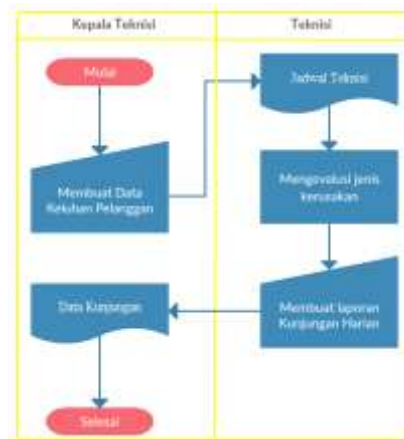
III PERANCANGAN SISTEM

A. Analisis Kebutuhan

Penyusunan analisis kebutuhan dilakukan untuk mengurangi risiko gagal dikembangkan. Kegagalan pengembangan system umumnya terjadi karena system tidak memenuhi kebutuhan pengguna atau system tidak sesuai dengan proses bisnis yang berjalan. Studi kelayakan perlu dilakukan dalam penyusunan analisis kebutuhan untuk mengurangi risiko tersebut. Studi kelayakan dapat dilakukan dengan cara wawancara, observasi dan membaca dokumen. Hasil analisis kebutuhan berupa dokumen *System Requirements Specifications* (SRS). Pengembangan ini menggunakan struktur SRS tersebut untuk menyampaikan analisis kebutuhan.

1) Kondisi Awal Penjadwalan

Proses selama wawancara dan observasi memberikan gambaran kondisi yang terjadi di lapangan. Pada proses penjadwalan teknisi, terdapat beberapa aktivitas yang harus dilakukan. Gambar 2 Menunjukkan *flowchart* aktifitas kondisi awal proses penjadwalan teknisi.



Gambar 2. *Flowchart* aktifitas kondisi awal proses penjadwalan teknisi.

Penjelasan pada *flowchart* diatas adalah sebagai berikut :

1. Kepala teknisi membuat data keluhan baru, data yang dibuat berisi informasi id speedy, nama pelanggan, alamat pelanggan, nomor telepon rumah, nomor telepon *selular*, perangkat MSAN (*Multi Service Access Network*), DP (*Distribution Point*), keterangan keluhan, jenis keluhan.
2. Kepala teknisi mengelompokkan data keluhan pelanggan sesuai tempat atau lokasi pelanggan untuk dibagikan kepada teknisi.
3. Teknisi mendapat informasi jadwal kerja yang telah di bagikan oleh kepala teknisi sebelumnya.
4. Teknisi mengevaluasi kerusakan dan memperbaiki kerusakan yang terjadi di pelanggan.
5. Teknisi memberikan hasil laporan kunjungan kepada kepala teknisi.

Pembuatan SRS dapat dilakukan setelah data yang dikumpulkan pada kondisi awal sudah cukup menjelaskan keadaan sebelum sistem dikembangkan.

2) Deskripsi umum sistem yang dikembangkan

Deskripsi umum sistem diperlukan untuk menjelaskan prespektif produk akhir hasil pengembangan, fungsi produk yang diharapkan untuk berjalan, kelompok pengguna, lingkungan pengoprasian serta lingkungan pengguna yang akan menggunakan sistem.

1. Prespektif produk akhir

Produk pengembangan adalah aplikasi dapat digunakan oleh Divisi Teknik Telkom Kandatel Pati dalam melakukan manajemen penjadwalan pelanggan. Aplikasi tersebut digunakan untuk mengelola jadwal perbaikan kerusakan pelanggan.

2. Fungsi Produk

Aplikasi dapat membagikan jadwal baru kepada teknisi dengan lebih cepat tanpa harus menunggu satu hari kerja dan juga teknisi dapat membuat laporan kunjungan. Data yang disimpan dalam sistem dapat diolah untuk menghasilkan laporan.

3. Kelompok Pengguna

Pengguna yang menggunakan sistem terdiri dari 4 jenis, yaitu : manajer, kepala teknisi, operator, teknisi. Keempat pengguna tersebut memiliki aktivitas yang berbeda dalam menggunakan sistem.

- a. Manajer adalah pengguna yang bertugas atau berwenang menambhkan pegawai baru. Manager berperan sebagai kepala Telkom Kandatel Pati.
- b. Kepala Teknisi adalah pengguna yang bertugas melakukan tambah teknisi, pengelolaan penjadwalan teknisi, dan data pelanggan. Kepala teknisi berperan sebagai kepala divisi teknik di Telkom Kandatel Pati.

- c. Operator adalah pengguna yang bertugas melakukan pengolahan penjadwalan teknisi dan data pelanggan. Operator berperan sebagai asisten kepala teknisi
 - d. Teknisi adalah pengguna yang melakukan aktivitas perbaikan kerusakan pelanggan dan pencatatan kerusakan. Teknisi adalah pengguna yang merupakan pekerja lapangan di bawah divisi teknik di Telkom Kandatel Pati.
4. Lingkup pengoperasian
Aplikasi dioperasikan di area kerja Telkom Kandatel Pati. Telkom Kandatel Pati menggunakan koneksi internet dalam mengoperasikan aplikasi. Aplikasi bias digunakan sepanjang hari selama *server* dan jaringan tidak mengalami gangguan.
5. Lingkup Pengguna
Pada sisi operator aplikasi berupa aplikasi berbasis *web*. Pengguna bias mengakses sistem melalui *web browser* dari perangkat komputer, sedangkan dari sisi pengguna teknisi aplikasi berupa aplikasi ponsel pintar yang telah di pasang sebelumnya.

3) Kebutuhan Fungsional

Sistem yang dikembangkan diharapkan dapat memenuhi kebutuhan fungsional agar dapat digunakan sesuai dengan proses bisnis yang telah berjalan pada sisi pada divisi teknik di PT. Telkom Kandatel Pati. Berikut adalah kebutuhan fungsional yang harus dipenuhi oleh sistem.

1. Sistem pada sisi manajer dapat menyimpan, mengubah dan menghapus data pegawai dengan indeks pegawai. Fungsi tersebut bertujuan agar data pegawai yang masuk tercatat sebagai data pegawai atas nomor indeks pegawai yang dimaksud.
2. Sistem pada sisi manajer dapat melihat laporan kunjungan teknisi secara bulanan maupun periodik.
3. Sistem pada sisi manajer dapat melihat laporan kinerja teknisi secara bulanan maupun periodik.
4. Sistem pada sisi kepala teknisi dapat menyimpan, mengubah dan menghapus data teknisi yang digunakan atribut pegawai dengan nomor indeks pegawai. Fungsi tersebut bertujuan agar data pegawai dapat diawasi langsung oleh kepala teknisi.
5. Sistem pada sisi kepala teknisi dan operator dapat menyimpan, mengubah dan menghapus data pelanggan yang digunakan atribut pelanggan dengan nomor indeks pegawai. Fungsi tersebut bertujuan agar teknisi dapat menyimpan informasi keluhan sesuai dengan indeks pelanggan yang dimilikinya.
6. Sistem pada sisi kepala teknisi dan operator dapat mengolah data jadwal sesuai dengan urutan waktu keluhan dan kondisi (*state*) yang dimiliki oleh teknisi tersebut.
7. Sistem pada sisi kepala teknisi dan Operator dapat mengolah data keluhan pelanggan yang digunakan atribut keluhan dengan nomor indeks *kat_keluhan*. Fungsi tersebut bertujuan agar dapat membuat jadwal.
8. Sistem pada sisi teknisi dapat menyimpan data laporan kunjungan sesuai nomor index jadwal yang dimilikinya.
9. Sistem pada sisi teknisi dapat melihat jadwal harian kunjungan teknisi secara langsung.

4) Kebutuhan Non-Fungsional

Selain kebutuhan fungsional sistem juga harus memenuhi kebutuhan non-fungsional agar sistem yang berjalan dapat digunakan dengan baik oleh pengguna pada sisi manajer, kepala teknisi, Operator, dan teknisi PT. Telkom Kandatel Pati. Berikut merupakan kebutuhan fungsional yang harus dipenuhi oleh sistem.

1. Operasional

- a. Sistem yang bekerja pada manajer, kepala teknisi, dan Operator dapat diakses dari perangkat computer.
 - b. Sistem memiliki fitur pemberitahuan pesan kesalahan muncul jika pengguna melakukan kesalahan pemakaian sistem.
 - c. Sistem dapat memperbarui data secara otomatis.
2. Keamanan
- a. Sistem memiliki fungsi login untuk memberikan akses ke pengguna.
 - b. Sistem memiliki fungsi role untuk membedakan hak pengguna.
 - c. Sistem mengimplementasikan metode *soft delete* untuk menjaga validasi pengolahan data

B. Desain Sistem

SRS yang dihasilkan dari analisis kebutuhan digunakan sebagai bahan perancangan sistem. Kebutuhan fungsional di dalam SRS menentukan bagaimana sistem harus dibangun sehingga mampu memenuhi kebutuhan yang diharapkan. Spesifikasi teknis dibuat pertama kali dalam perancangan untuk menetapkan lingkungan pengembangan. Pemodelan dilakukan setelah spesifikasi sistem selesai ditentukan. Penulisan kode belum bisa dimulai sebelum pemodelan selesai dibuat. Penulisan kode tanpa memperhatikan pemodelan akan menimbulkan risiko kegagalan pengembangan. Desain antarmuka dilakukan setelah pemodelan selesai dibuat sehingga sistem mampu dioperasikan pengguna dengan mudah. Pengembangan aplikasi yang mengabaikan desain antarmuka *user friendly* cenderung susah digunakan.

1) Spesifikasi Teknis

1. Arsitektur Sistem

Sistem yang dikembangkan berupa aplikasi berbasis *web* dan aplikasi ponsel pintar. *Web server* dibutuhkan untuk menjalankan aplikasi tersebut. Data yang diolah sistem disimpan pada basis data. *Server* basis data juga dibutuhkan untuk menjalankan basis data yang digunakan sistem. Manajer, kepala teknisi, Operator dan teknisi dapat mengakses sistem melalui *web browser* dengan cara memanggil URL aplikasi. Gambar 3 menunjukkan arsitektur sistem yang menjelaskan bagaimana sistem bekerja.

Web Server dan basis data berada pada cloud yang didukung oleh IaaS DigitalOcean. Sedangkan aplikasi perangkat bergerak pada ponselnya hanya ditujukan kepada teknisi.

2. Arsitektur Aplikasi

Deskripsi umum SRS menjelaskan bahwa sistem berupa aplikasi web pada sisi manajer, kepala teknisi, operator dan aplikasi perangkat bergerak pada teknisi PT. Telkom Kandatel Pati. Arsitektur yang digunakan untuk membangun aplikasi tersebut adalah Restful. Bahasa yang dipakai adalah PHP dengan kerangka kerja CodeIgniter 3.0, Javascript dengan kerangka kerja AngularJS dan Ionic.. Gambar 3.3 menunjukkan arsitektur aplikasi yang digunakan pada pengembangan ini.

3. Kebutuhan Perangkat Keras

Sistem berjalan dengan posisi server berada di *cloud* dengan dukungan IaaS sebagai penyedia platform dan infrastruktur. Dengan arsitektur tersebut maka kebutuhan perangkat keras untuk *server* dapat sepenuhnya diabaikan untuk dapat mendukung platform yang akan dikembangkan.

Sedangkan untuk sisi operator PT. Telkom Kandatel Pati memerlukan PC yang dapat menjalankan *web browser*, dari berbagai macam kebutuhan minimal *web browser*, maka perangkat keras yang dibutuhkan pada sisi *admin* dengan alah sebagai berikut :

- Prosesor : Intel Pentium 4
- RAM : 512 MB

- HDD : 4 GB

Untuk sistem yang berjalan pada pengguna petugas penjadwalan teknisi PT. Telkom Kandatel Pati memerlukan perangkat bergerak dengan spesifikasi minimal RAM 512 MB.

4. Kebutuhan Perangkat Lunak

Aplikasi membutuhkan perangkat lunak pendukung agar aplikasi bisa berjalan di server. Kebutuhan minimal perangkat lunak server adalah :

- Ubuntu 12.04
- LAMP Stack Server

Manajer, kepala teknisi, operator juga memerlukan perangkat lunak pendukung untuk bisa mengakses aplikasi. Kebutuhan minimal perangkat lunak manajer, kepala teknisi, operator PT. Telkom Kandatel Pati adalah :

- Web Browser dengan dukungan Javascript

Perangkat yang digunakan oleh teknisi PT. Telkom Kandatel Pati juga memerlukan perangkat lunak pendukung untuk bisa menggunakan aplikasi yang dikembangkan. Kebutuhan minimal perangkat lunak teknisi PT. Telkom Kandatel Pati adalah menggunakan Sistem Operasi Android versi 4.1 disesuaikan dengan kebutuhan minimal kerangka kerja Ionic yang digunakan untuk mengembangkan sistem.

2) Desain ERD

Sistem yang dikembangkan merupakan sistem yang berfungsi untuk mengolah data. Pemodelan perlu dilakukan untuk menentukan data yang diolah. Pemodelan tersebut dilakukan setelah spesifikasi teknis ditentukan. ERD digunakan untuk memodelkan data yang akan disimpan dalam basis data. Langkah yang dilakukan dalam merancang ERD adalah menentukan entitas yang ada, lalu entitas tersebut dihubungkan dalam notasi diagram.

1. Menentukan Entitas

Entitas dapat ditentukan dari SRS. Kebutuhan yang dijelaskan memberikan gambaran data-data yang diolah. Data tersebut merupakan entitas yang akan digunakan dalam merancang ERD. Tabel 1 menunjukkan entitas yang telah ditentukan.

Tabel 1. Tebel Entitas ERD

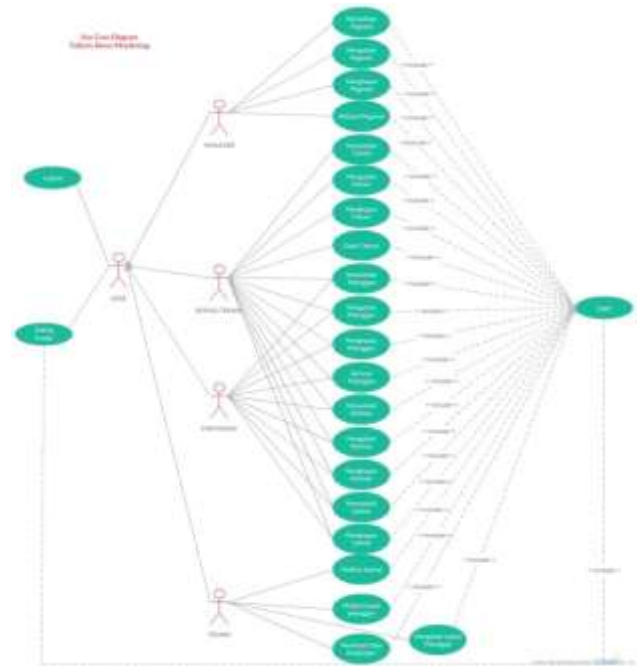
| No | Nama | Keterangan |
|-----|-------------|---|
| 1. | dp | Data DP (<i>Distribution Point</i>) |
| 2. | Jadwal | Data jadwal kunjungan teknisi |
| 3. | Kat_keluhan | Data kategori keluhan |
| 4. | Keluhan | Data keluhan pelanggan |
| 5. | kunjungan | Data kunjungan teknisi |
| 6. | login | Data login pegawai |
| 7. | msan | Data MSAN (<i>Multi Service Access Network</i>) |
| 8. | pegawai | Data pegawai |
| 9. | pelanggan | Data pelanggan |
| 10. | user | Data user |

3) Desain UML

Fungsi yang ditunjukkan pada SRS belum memiliki detail yang cukup untuk dikembangkan. Pemodelan UML memberikan ruang untuk mengembangkan detail fungsi yang dibutuhkan oleh sistem sebelum penulisan kode dimulai. Kesalahan yang dilakukan pada pemodelan akan lebih mudah dianalisis dan diperbaiki daripada saat sistem sudah ditulis dengan bahasa pemrograman.

1. Diagram Use case Sistem

Bagaimana pengguna menggunakan sistem dapat digambarkan dengan diagram *use case*. Diagram *use case* dibuat berdasarkan skenario yang dilakukan pengguna ketika menggunakan fungsi sistem. Diagram *use case* terlihat pada Gambar 4.

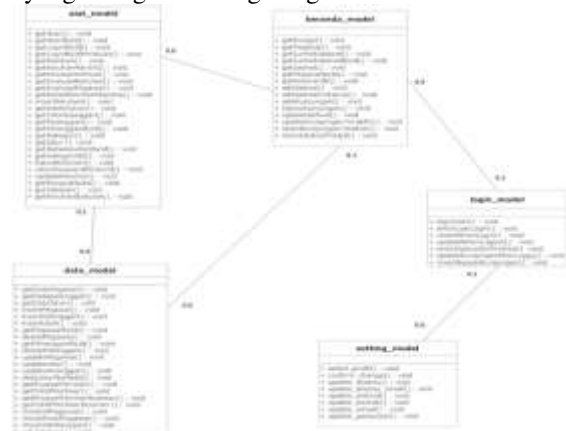


Gambar 3. Diagram *use case* sistem

Pemodelan diagram *use case* menunjukkan bahwa sistem terdapat 4 aktor yang menggunakan sistem. Sistem memiliki sembilan belas skenario umum. Aktor user (pengguna) adalah bentuk umum dari aktor manajer, kepala teknisi, operator dan teknisi.

2. Diagram Kelas

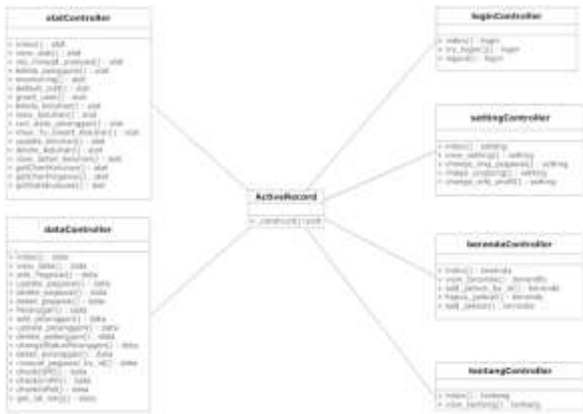
Diagram kelas pada gambar 4 menunjukkan kelas dengan atribut yang saling terhubung dengan kelas lain.



Gambar 4. Diagram kelas tipe model keseluruhan sistem

Kelas yang ada pada Gambar 4 tersebut merupakan kelas dengan tipe model yang berfungsi sebagai abstraksi basis data. Semua model yang ada merupakan *extend* dari *active record* yang berfungsi sebagai penghubung dengan kelas tipe *controller*.

Diagram kelas pada gambar 5 menunjukkan kelas *controller* yang berfungsi sebagai pengolah *Business logic*.



Gambar 5. Diagram kelas tipe *controller* keseluruhan sistem

Kelas *controller* dapat berinteraksi dengan data di dalam basis data dengan *Repository Pattern*. Deskripsi masing – masing kelas ditunjukkan pada tabel 2.

Tabel 2. Deskripsi diagram kelas

| No | Kelas | Deskripsi |
|----|-------------------|---|
| 1 | login_model | Model data login |
| 2 | setting_model | Model data setting pegawai |
| 3 | beranda_model | Model data beranda |
| 4 | alat_model | Model data alat |
| 5 | data_model | Model data data |
| 6 | alatController | <i>Controller</i> yang mengelola data alat |
| 7 | berandaController | <i>Controller</i> yang mengelola data beranda |
| 8 | dataController | <i>Controller</i> yang mengelola data data |
| 9 | loginController | <i>Controller</i> yang mengelola data login |
| 10 | tentangController | <i>Controller</i> yang mengelola data tentang |
| 11 | settingController | <i>Controller</i> yang mengelola data setting |

Pemodelan yang dibuat sudah cukup untuk digunakan menjelaskan bagaimana sistem akan memenuhi kebutuhan sistem. Diagram aktivitas memberikan gambaran utama bagaimana sistem akan digunakan dalam proses bisnis. Diagram *use case* memberikan informasi bagaimana pengguna menggunakan sistem. Diagram *sequence* menampilkan perilaku sistem ketika digunakan pengguna. Diagram kelas menunjukkan struktur sistem.

Kode yang digunakan untuk membangun aplikasi bisa dikerjakan setelah tahap ini. Spesifikasi teknis dan pemodelan dapat memberikan gambaran dan arahan bagaimana kode ditulis. Penulisan kode kemudian dikerjakan hingga selesai.

IV PENGUJIAN SISTEM

Pengujian unit dilakukan dalam pengujian sistem ini. Pengujian unit adalah melakukan analisis dari masing – masing fungsi di dalam sistem. Tujuan pengujian unit adalah membatasi objek pengujian dalam skala kecil sehingga tahap perbaikan akan lebih cepat jika ditemukan kesalahan. Pengujian unit dilakukan dengan menjalankan setiap fungsi di dalam sistem. Teknik yang digunakan dalam pengujian unit ini adalah teknik kotak hitam. Teknik tersebut adalah teknik untuk memastikan masing – masing fungsi individu berjalan dengan cara diamati ketika fungsi dieksekusi.

1. Login sistem

Login sistem adalah aktivitas yang dilakukan ketika memulai akses ke dalam sistem. pengujian dilakukan dengan memasukkan username dan password pada form login yang ditunjukkan pada Gambar 6



Gambar 6. Pengujian aktivitas *login* sistem

Hasil yang diharapkan ketika pengguna memasukkan *username* dan *password* yang benar adalah pengguna akan mendapatkan akses masuk menuju halaman depan sistem. Gambar 7 menunjukkan antarmuka ketika pengguna berhasil *login*.



Gambar 7. Hasil pengujian login sistem berhasil

Hasil yang diharapkan ketika pengguna salah atau gagal memasukkan *username* dan *password* adalah halaman login memberikan pesan kepada pengguna. Pesan yang diberikan pengguna ditunjukkan pada Gambar 8.



Gambar 8. Hasil pengujian login sistem gagal

2. Membuat data pegawai

Membuat data pegawai adalah kegiatan untuk membuat data baru pegawai. Pengujian dengan memasukkan data pegawai baru pada form data yang ditunjukkan pada Gambar 9.

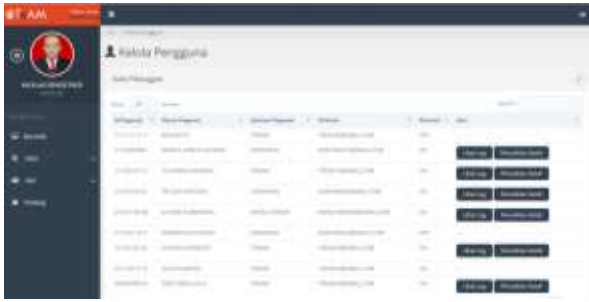


Gambar 9. Hasil pengujian tambah pegawai

Hasil yang diharapkan sistem ketika tombol *save* ditekan adalah data masuk ke dalam basis data. Hasil pengujian ini adalah menunjukkan data berhasil disimpan.

3. Kelola data pegawai

Kelola data pegawai adalah tindakan yang dilakukan untuk mengelola data pegawai. Pengujian dilakukan dengan menekan tombol Alat → Kelola Pengguna. Gambar 10 menunjukkan kelola data pegawai yang dibaca sistem.



Gambar 10. Hasil pengujian kelola pegawai

Hasil yang diharapkan dari pengujian ini adalah seluruh data kelola pegawai yang memiliki status aktif dan tidak aktif ditampilkan dalam bentuk tabel. Gambar 10 Menunjukkan pengujian berhasil.

4. Mengubah status pegawai

Dalam halaman kelola pegawai pengguna bisa melakukan pengaktifan pegawai dengan menekan *off*. Gambar 11 menunjukkan aktivasi pengguna dengan status *off*



Gambar 11. Hasil pengujian status pegawai (status *off*)

Hasil yang diharapkan ketika pengguna sudah menekan *off*, maka status pengguna akan berubah menjadi *on* jika ingin mengubah status menjadi *off* kembali maka pengguna tinggal menekan *on* sehingga status berubah menjadi *off*. Pesan yang diberikan pengguna ditunjukkan pada Gambar 12



Gambar 12. Hasil pengujian status pegawai (status *on*)

5. Membaca data pegawai

Membaca data pegawai digunakan ketika pengguna mencari data pegawai yang telah dibuat sebelumnya. Pengujian dilakukan dengan menekan tombol Data → Pegawai. Gambar 13 menunjukkan seluruh data pegawai yang dibaca sistem.



Gambar 13. Hasil pengujian membaca data pegawai

Hasil yang diharapkan dari pengujian ini adalah seluruh data pegawai yang memiliki status aktif ditampilkan dalam bentuk tabel. Gambar 13 Menunjukkan pengujian berhasil.

6. Mengubah data pegawai

Mengubah data pegawai adalah tindakan yang dilakukan untuk mengantisipasi kesalahan masukan data pegawai. Pengujian dilakukan dengan menekan tombol ubah informasi pada tabel pegawai lalu memasukkan data dalam *form data*. *Form* ubah data pegawai ditunjukkan pada Gambar 14.

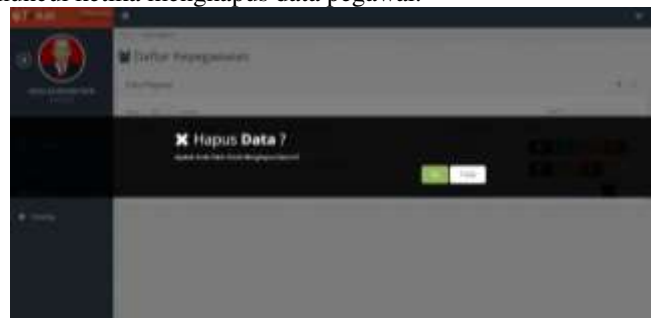


Gambar 14. Hasil pengujian mengubah data pegawai

Hasil yang diharapkan dari pengujian ini adalah ketika data baru disimpan maka data yang lama akan diperbarui dengan data yang baru. Hasil pengujian ini menunjukkan data berhasil diperbarui.

7. Menghapus data pegawai

Menghapus data pegawai adalah kegiatan ketika data di dalam sistem tidak diperlukan lagi. Pengujian ini dilakukan dengan menekan hapus data pegawai pada halaman daftar kepegawaian. Gambar 15 menunjukkan *form* konfirmasi yang muncul ketika menghapus data pegawai.



Gambar 15. Hasil pengujian menghapus data pegawai

Hasil yang diharapkan dalam pengujian ini adalah data pegawai akan dihapus ketika tombol Hapus ditekan. Hasil pengujian ini menunjukkan bahwa data pegawai berhasil dihapus.

8. Membaca *log* pegawai

Membaca *log* pegawai yang telah dilakukan merupakan salah satu tindakan untuk mengamati detail informasi *login* dan *logout* pegawai. Pengujian dilakukan dengan menekan tombol lihat *log* pada tabel kelola pengguna. Gambar 16 menunjukkan tampilan detail *log* pegawai.



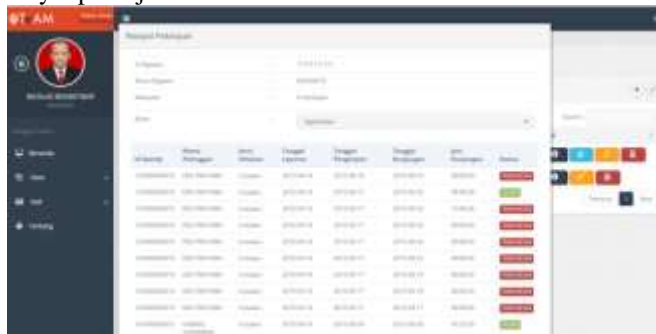
Gambar 16. Hasil pengujian membaca log pegawai

Hasil pengujian yang diharapkan dari pengujian ini adalah semua data *log* pegawai ditampilkan dengan informasi detail. Pengujian menunjukkan dokumen berhasil menampilkan seluruh *login* dan *logout* pegawai.

9. Membaca riwayat pekerjaan teknisi

Membaca riwayat pekerjaan pegawai yang telah dilakukan merupakan salah satu tindakan untuk mengamati

detail informasi pekerjaan yang diterima teknisi. Pengujian dilakukan dengan menekan tombol lihat riwayat teknisi pada tabel data pegawai. Gambar 17 menunjukkan tampilan detail riwayat pekerjaan teknisi.



Gambar 17. Hasil pengujian membaca riwayat pekerjaan teknisi

Hasil pengujian yang diharapkan dari pengujian ini adalah semua data riwayat pekerjaan teknisi ditampilkan dengan informasi detail. Pengujian menunjukkan dokumen berhasil menampilkan seluruh riwayat pekerjaan teknisi.

10. Membaca evaluasi

Membaca evaluasi merupakan salah satu tindakan untuk mengamati detail informasi pekerjaan, keluhan dan kerusakan yang diterima. Pengujian dilakukan dengan menekan tombol Alat → Evaluasi. Gambar 18 menunjukkan tampilan evaluasi.



Gambar 18. Hasil pengujian membaca riwayat pekerjaan teknisi

11. Data keluhan pelanggan

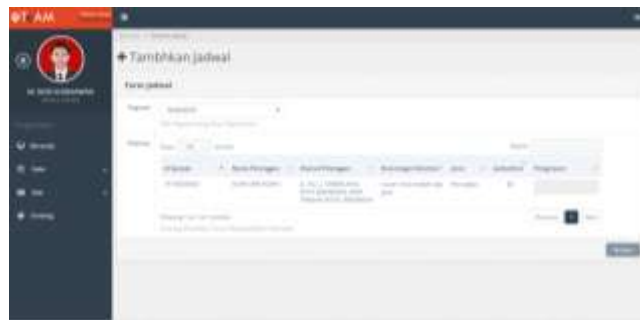
Kepala teknisi dan operator membuat data keluhan baru. Tampilan pembuatan data keluhan baru ditunjukkan pada Gambar 19.



Gambar 19 Pengujian integrasi membuat data keluhan Data keluhan baru yang dibuat oleh kepala teknisi akan diubah menjadi data jadwal harian teknisi.

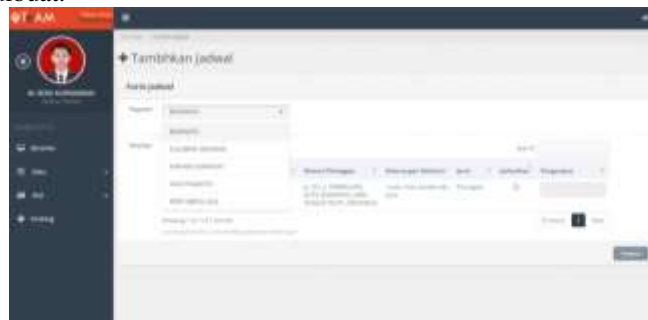
12. Kepala Teknisi membuat jadwal harian teknisi

Kepala teknisi membuat jadwal harian teknisi sesuai dengan keluhan yang telah masuk sebelumnya. Tampilan pembuatan data jadwal harian baru ditunjukkan pada Gambar 20



Gambar 20 Pengujian integrasi membuat jadwal harian teknisi baru

Data jadwal harian teknisi baru yang dibuat oleh kepala teknisi menjadi data jadwal yang akan dilakukan pengerjaan keluhan oleh teknisi sesuai dengan data jadwal yang telah dibuat.



Gambar 21 Pengujian integrasi memilih teknisi untuk jadwal harian teknisi baru

13. Melihat jadwal harian teknisi

Jadwal yang telah dibuat dapat dilihat oleh pengguna sistem melalui menu Beranda. Gambar 22 menunjukkan jadwal harian teknisi disisi Web.



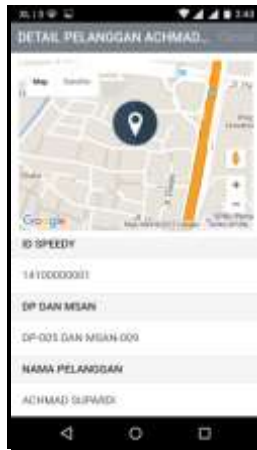
Gambar 22 Pengujian integrasi jadwal harian teknisi di Web



Gambar 23 Pengujian integrasi jadwal harian teknisi Budianto

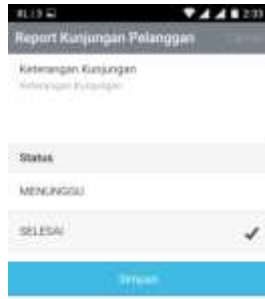
14. Melihat lokasi pelanggan

Teknisi dapat melihat lokasi pelanggan yang telah terjadwal pada jadwal harian teknisi. Gambar 24 menunjukkan tampilan lokasi pelanggan.



Gambar 24 Pengujian integrasi lokasi pelanggan teknisi

15. Membuat laporan kunjungan
 Teknisi dapat membuat laporan kunjungan yang bertujuan agar teknisi dapat menyimpan informasi selama melakukan aktivitas perbaikan kerusakan selain itu sekaligus memberi tahu kepala teknisi mengenai informasi tersebut. Gambar 25 menunjukkan aktivitas membuat laporan kunjungan.



Gambar 25 Pengujian integrasi membuat laporan kunjungan
 Hasil pengujian yang diharapkan dari pengujian ini adalah semua data evaluasi ditampilkan dengan informasi detail. Pengujian menunjukkan dokumen berhasil menampilkan seluruh data evaluasi.

Lima belas pengujian yang ditampilkan menunjukkan fungsi yang memiliki pola berbeda. Sistem memiliki fungsi yang memiliki pola sama seperti tambah pelanggan, tambah jadwal, hapus jadwal, hapus pelanggan, edit pelanggan dan edit keluhan. Pengujian unit dilakukan seluruh fungsi tersebut. Daftar lengkap pengujian unit ditampilkan pada Tabel 3 Disertai hasil analisisnya.

Tabel 3. Daftar hasil pengujian

| Pengujian | Hasil yang diharapkan | Analisis |
|---------------------------|--|----------|
| Login | Masuk halaman depan, jika <i>username</i> dan <i>password</i> salah muncul pemberitahuan | Berhasil |
| Membuat data pegawai baru | Data disimpan dalam sistem | Berhasil |
| Kelola data pegawai | Halaman menampilkan tabel data kelola pegawai | Berhasil |
| Mengubah status pegawai | Status pegawai dapat diubah dari <i>off</i> ke <i>on</i> | Berhasil |
| Membaca data pegawai | Halaman menampilkan tabel data | Berhasil |
| Mengubah data pegawai | Data lama diperbarui dengan data baru | Berhasil |

| | | |
|---------------------------------------|--|----------|
| Menghapus data pegawai | Data dihapus dari sistem | Berhasil |
| Membaca <i>log</i> pegawai | Halaman menampilkan tabel data | Berhasil |
| Membaca riwayat pekerjaan teknisi | Halaman menampilkan tabel riwayat pekerjaan teknisi | Berhasil |
| Pemulihan sandi | Sandi berubah menjadi sandi yang sesuai dengan alamat email sebelum "@" | Berhasil |
| Membaca evaluasi | Halaman menampilkan data evaluasi | Berhasil |
| Menambah data pelanggan baru | Data disimpan dalam sistem | Berhasil |
| Mengubah status pelanggan | Status pelanggan dapat diubah dari <i>off</i> ke <i>on</i> | Berhasil |
| Membaca data pelanggan | Halaman menampilkan tabel data | Berhasil |
| Mengubah data pegawai | Data lama diperbarui dengan data baru | Berhasil |
| Menghapus data pelanggan | Data dihapus dari sistem | Berhasil |
| Menambah data keluhan baru | Data disimpan dalam sistem | Berhasil |
| Membaca data keluhan | Halaman menampilkan tabel data | Berhasil |
| Mengubah data keluhan | Data lama diperbarui dengan data baru | Berhasil |
| Menghapus data keluhan | Data dihapus dari sistem | Berhasil |
| Membaca data keluhan <i>finish</i> | Halaman menampilkan tabel data | Berhasil |
| Menambah Jadwal baru | Data disimpan dalam sistem | Berhasil |
| Melihat jadwal harian seluruh teknisi | Halaman menampilkan tabel data | Berhasil |
| Menghapus data jadwal hari ini | Data dihapus dari sistem | Berhasil |
| Melihat lokasi pelanggan | Halaman menampilkan peta lokasi pelanggan | Berhasil |
| Teknisi Melihat jadwal harian | Halaman menampilkan data jadwal sesuai teknisi yang melakukan <i>login</i> | Berhasil |
| Melihat detail pelanggan | Halaman menampilkan data pelanggan | Berhasil |
| Mengubahlo kasi pelanggan | Data lama diperbarui dengan data baru | Berhasil |
| Membuat | Data disimpan dalam sistem | Berhasil |

| | | |
|-------------------|---|----------|
| laporan kunjungan | | |
| Setting profile | Data lama diperbaharui dengan data baru | Berhasil |
| Logout | Pengguna keluar dari sistem | Berhasil |

V PENUTUP

1) Kesimpulan

Selama pengembangan sistem ini terdapat beberapa hal yang bisa disimpulkan. Kesimpulan yang didapatkan antara lain:

1. Aplikasi *mobile* pencarian lokasi pelanggan dan penjadwalan teknisi PT. Telkom Kandatel Pati mempermudah teknisi mencari lokasi pelanggan.
2. Aplikasi *mobile* pencarian lokasi pelanggan dan penjadwalan teknisi PT. Telkom Kandatel Pati mempermudah teknisi mendapatkan informasi jadwal lebih cepat.
3. Sistem yang dikembangkan dapat membantu mengawasi kinerja teknisi dalam melaksanakan pekerjaan.
4. Sistem yang dikembangkan dapat mengurangi masa tunggu pelanggan dalam penanganan kerusakan oleh teknisi dimana keluhan pelanggan akan langsung didistribusikan keteknisi lapangan.
5. Sistem yang dikembangkan dapat memfasilitasi kegiatan pencarian lokasi dan penjadwalan oleh teknisi sesuai kebutuhan PT. Telkom Kandatel Pati.

2) Saran

1. Sistem informasi ini memiliki banyak ruang untuk dikembangkan seperti integrasi dengan sistem manajemen marketing.
2. Fitur pelanggan baru pada sistem manajemen marketing belum bisa diimplementasikan karena dibutuhkan perancangan dan pengembangan sistem yang berbeda.

DAFTAR PUSTAKA

- 1] S. Alter, *The Work System Method: Connecting People, Processes, and IT for Business Results*, CA: Works System Press, 2006.
- 2] Fathansyah, *Basis Data*, Bandung: Informatika, 2012.

- 3] W. Hales, *HTML5 and JavaScript Web Apps*, California: O'Reilly Media, Inc, 2013.
- 4] R. Earp dan S. Bagui, *Database Design Using Entity-Relationship Diagrams*, Florida: Auerbach, 2003.
- 5] A. S. Rosa dan M. Shalahuddin, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Bandung: Modula, 2011.
- 6] B. K. Hamilton dan R. Miles, *Learning UML 2.0*, California: O'Reilly Media, Inc, 2006.
- 7] OMG, "UML Infrastructure Specification, v2.4.1," 2011. [Online]. Available: <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>.
- 8] D. P. Oktavian, *Membuat Web Powerfull Menggunakan PHP*, Yogyakarta: Mediakom, 2013.
- 9] R. Lerdorf, K. Tatroe dan P. MacIntyre, *Programming PHP*, California: O'Reilly Media, Inc, 2006.
- 10] W. Willard, *HTML: A Beginner's Guide*, California: McGraw-Hill Osborne Media, 2013.
- 11] A. Taufik, *Pemrograman Grafik dengan Java*, Bandung: Informatika, 2010.
- 12] A. Freeman, *Pro ASP. NET MVC 5*, New York: Apress, 2013.
- 13] W. Komputer, *Mudah membuat Aplikasi SMS Gateway dengan CodeIgniter*, Jakarta: Elex Media Komputindo, 2014.
- 14] driftyco, "What is Ionic," 2014. [Online]. Available: <https://github.com/driftyco/ionic>. [Diakses 17 10 2014].
- 15] M. Huda dan B. Komputer, *Membuat Aplikasi Database Dengan Java, MySql dan NetBeans*, Jakarta: Elex Media Komputindo, 2010.
- 16] Riyanto, *Sistem Informasi Geografis Berbasis Mobile*, Yogyakarta : GAVA MEDIA, 2010.
- 17] R. Pressman, 2012, Yogyakarta: ANDI, *Rekayasa Perangkat Lunak*.
- 18] R. Tantra, *Manajemen Proyek Sistem Informasi*, Yogyakarta: ANDI, 201.