

PERANCANGAN SISTEM PAKAR DIAGNOSA KERUSAKAN *SMARTPHONE* ANDROID BERBASIS WEB MENGGUNAKAN *METODE FORWARD CHAINING*

Feby Hamdani Dipraja¹, Abdurahman Fauzi²

¹Universitas Adhirajasa Reswara Sanjaya
e-mail: febyhamdanidipraja3@gmail.com

²Universitas Adhirajasa Reswara Sanjaya
e-mail: abdurahman.auz@ars.ac.id

ABSTRAK

Sistem Pakar adalah sistem Berbasis pengetahuan . untuk membuat sistem pakar identifikasi masalah atau gejala, mulai dari input data pertanyaan , data jawaban , pendaftaran pengguna , hingga proses konsultasi . Pada Penelitian ini sedikit berbeda tidak menggunakan pertanyaan user cukup Cek Kerusakan dengan memilih list kerusakan dan list diagnose yang menghasilkan Solusi Untuk memperbaiki kerusakan dengan tahapan yang ringan, digunakannya Metodologi yang berupa aturan (rule) dengan teknik Forward Chaining dan menggunakan Topology pohon Pakar Sebagai Pencarian Pertama. Dengan website sistem pakar ini dapat menyediakan informasi tentang kerusakan Smartphone, dan memberikan sebuah solusi untuk mengatasi masalah dengan tepat dan akurat.

Kata Kunci: Sistem Pakar, Diagnosa Kerusakan *Smatphone*, *Waterfall* , *Forward Chaining*

ABSTRACT

Expert System is a knowledge Based system. to create an expert system for identifying problems or symptoms, starting from inputting question data, answer data, user registration, to the consultation process. In this study, it is a little different, not using user questions, simply checking for damage by selecting a list of damage and a list of diagnoses that produce a solution. . With this expert system website, it can provide information about smartphone damage, and provide a solution to solve the problem appropriately and accurately.

Keywords: Expert System, *Smatphone* Damage Diagnosis, *Waterfall*, *Forward Chaining*

1. Pendahuluan

1.1. Latar belakang

Seiring dengan berkembang nya jaman, perkembangan teknologi juga semakin pesat, diantaranya adalah perkembangan dalam bidang komputer, internet. Melalui komputer dan internet sistem informasi yang di sampaikan akan lebih efektif dan efisien. Tentunya diperlukan sumber daya manusia yang handal, agar sistem informasi tersebut dapat berjalan semaksimal mungkin, sesuai yang di harapkan (Sartini, 2015).

Pemakaian telepon genggam (*handphone*) sangat meningkat, yang dahulu

hanya berfungsi untuk menelepon dan mengirimkan pesan sekarang berkem bang penerima setiap informasi secara luas menjadi telepon pintar (*smartphone*) yang bisa. Smartphone bukan hanya sebagai alat komunikasi namun juga berfungsi sebagai alat yang memudahkan untuk mengakses internet ataupun mempermudah dalam transaksi online kebutuhan pengguna Menurut Stefanus dalam (Kurnia & Setiyanto, 2017),

Sistem pakar merupakan suatu sistem komputer yang dapat meniru atau mengimbangi kemampuan seorang pakar, yaitu kecerdasan buatan yang

menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan dari penelitian ini adalah membuat aplikasi sistem pakar berbasis web untuk mendeteksi kerusakan Smartphone. Dengan menggunakan teknik penalaran *forward chaining*, deteksi dilakukan dengan memulai dari sekumpulan gejala-gejala, nantinya dapat melihat kesimpulan jenis kerusakan pada handphone. penggunaannya hanya tinggal membawa smartphone nya tersebut ke tempat *service* terdekat untuk mengetahui jenis kerusakan apa yang terjadi pada perangkat tersebut. Waktu perbaikan yang habis terpakai selama handphone pengguna di tempat *service* juga dapat menyita waktu pengguna. Belum juga biaya yang akan dikeluarkan untuk memperbaiki perangkat tersebut serta penipuan yang banyak terjadi dikala kita membawa handphone ke tempat *service* (Pangkey et al., 2016).

Berdasarkan permasalahan yang di paparkan di atas, sangat di butuhkan nya sebuah aplikasi sistem pakar yang dapat menangani berbagai masalah kerusakan yang di timbulkan oleh Smartphone. Aplikasi yang dibuat harus mampu menangani masalah jarak, waktu, tenaga dan biaya yang dikeluarkan serta mudah digunakan bagi seluruh kalangan pengguna handphone (Pangkey et al., 2016).

Dengan adanya masalah itu maka kami akan membuat sebuah aplikasi yang akan memprediksi golongan darah menurut sifat dan karakter manusia yang di bantu oleh Metode *Forward Chaining* dan berbasis Web, dengan berbasis *Web* saya berharap aplikasi ini dapat dengan mudah di akses untuk semua kalangan Masyarakat.

Berdasarkan penelitian sebelumnya, maka penulis tertarik melakukan penelitian dengan mengangkat judul

"Perancangan Sistem Pakar Diagnosa Kerusakan Smartphone Android Berbasis Web Menggunakan Metode Forward Chaining".

1.2. Rumusan Masalah

Berdasarkan penjelasan di latar belakang sebelumnya, maka rumusan masalah yang teridentifikasi sebagai berikut:

1. Bagaimana Cara Merancang Aplikasi Sistem Pakar yang dapat Mendiagnosa Kerusakan *Smartphone* yang bisa diakses Secara *Mudah*.
2. Bagaimana Cara Membangun aplikasi Sistem Pakar Yang Memberikan

Kemudahan kepada Masyarakat atau *User Smartphone*.

3. Bagaimana Menerapkan Metode *Forward Chaining* untuk Mendiagnosa Kerusakan *Smartphone*.

1.3. Landasarn Teori

1.3.1. Pengertian Sistem Pakar

Menurut Siswanto dalam (Fazrie, 2017) sistem pakar atau *expert system* yaitu program-program yang bertingkah laku seperti manusia pakar/ahli (*human expert*). Secara umum sistem pakar (*expert system*) adalah sebuah sistem yang berusaha, mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan permasalahan yang dihadapi manusia atau pakar juga sebagai sarana untuk menyimpan pengetahuan para pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan orang awam.

Ciri - ciri Sistem pakar sistem pakar menurut Arhami, M dalam (Wiguna & Harianto, 2017) adalah sebagai Berikut :

1. Terbatas pada *domain* keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan - alasan yang diberikannya dengan cara yang dapat di pahami.
4. Berdasarkan pada kaidah atau ketentuan atau *rule* tertentu.
5. Dirancang dapat untuk dapat dikembangkan secara bertahap.
6. pengetahuan dan mekanisme penalaran (*interfance*) jelas terpisah.
7. keluarnya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntunoleh dialog dengan *user*.

1.3.2. Pengertian Program

Menurut Minir & Rinaldi dalam (Fridayanthie & Charter, 2016) "Program adalah *algoritma* yang ditulis dalam bahasa komputer". Pemrograman adalah proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan bahasa pemrograman. Penulisan program biasanya menggunakan menggunakan program *editor* yang telah

disediakan oleh bahasa pemrograman yang dipilih.

1.3.3. Pengertian Forward Chaining

Menurut Hayadi, B.H. & Rukun, K. dalam (Wijaya & Tanamal, 2019) Metode Runtut Maju atau istilahnya *Forward Chaining* adalah metode dimana pencocokan pernyataan atau faktanya dimulai dari fakta atau IF terlebih dahulu, lalu dilanjutkan dengan *THEN* yang berisi dengan hipotesisnya.

1.3.4. Pengerian Web

Menurut Rohi Abdullah dalam (Destiningrum & Adrian, 2017) web adalah : "Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet". Berdasarkan uraian, penulis menyimpulkan bahwa web adalah Sebuah software yang berfungsi untuk menampilkan dokumen - dokumen pada suatu web yang membuat pengguna dapat mengakses internet melalui software yang terkoneksi dengan internet.

1.3.5. Pengertian PHP

PHP adalah sebuah bahasa open source dari sisi server yang digunakan untuk membuat website yang dinamis. PHP dapat dimasukkan ke dalam HTML. PHP biasanya digunakan bersamaan dengan database MySQL untuk linux/unix server. PHP merupakan bahasa yang paling digemari saat ini (IndoSite, 2017).

1.3.6. Pengertian Framework

Menurut Betha Sidik dalam (Destiningrum & Adrian, 2017) Framework adalah : "kumpulan intruksi-intruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu".

1.3.7. Pengertian Codeigniter

Menurut Betha Sidik dalam (Destiningrum & Adrian, 2017) CodeIgniter adalah : "Sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk

memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal".

1.3.8. Xampp

Menurut MADCOMS dalam (Ayu & Permatasari, 2018) "Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain."

1.3.9. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (IndoSite, 2017).

Menurut MADCOMS dalam (Ayu & Permatasari, 2018) "MySQL adalah sistem manajemen Database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multiuser dan SQL Database managemen system (DBMS)".

Menurut Adi Nugroho dalam (Destiningrum & Adrian, 2017) MySQL (My Structured Query Language) adalah: " Suatu sistem basis data relation atau Relational Database managemnt System (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga sapat digunakan untuk aplikasi multi user (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (General Public License). Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan closed source atau komersial".

2. Metode Penelitian

2.1. Metode Pengumpulan Data

1. Observasi

Pada tahap ini Penulis dapat pengumpulan data yang kompleks karena melibatkan berbagai faktor dalam pelaksanaannya. Dalam *participant observation*, peneliti terlibat secara langsung dalam kegiatan sehari-hari orang atau situasi yang diamati sebagai sumber data.

2. Studi Pustaka

Penulis melakukan penelitian dengan mencari sumber-sumber dari dokumen yang berkenaan dengan proses pekerjaan tersebut.

2.2. Metode Pengembangan Sistem

Proses perancangan perangkat lunak pada penelitian ini menggunakan metode *waterfall*, pada metode ini terdapat beberapa tahapan yaitu:

1. Analisis Kebutuhan (*requirements analysis*)

Analisis kebutuhan sistem pada tahap ini yaitu terdapat *form register* untuk *admin* dan *user*, *form login* untuk *admin* dan *user*, *form* Utama yang melampirkan pilihan menu untuk *admin* dan *user*, *form* daftar kerusakan untuk menginput dan menampilkan list kerusakan dalam tabel yang hanya bisa di akses oleh *admin*, *form Diagnosa* kerusakan untuk *user*, *form* ubah dan hapus tugas untuk *admin*, *form* petunjuk untuk *user* dan *admin*, dan *form* profil untuk *user* dan *admin*.

2. Desain (*design*)

Desain dalam tahap ini yaitu menggunakan ERD dan UML.

3. Pembuatan Code (*Code Generation*)

Pembuatan *code* pada tahap ini menggunakan bahasa pemrograman *php*.

4. Integrasi dan Pengujian (*Integration & Testing*)

Integrasi dan pengujian dalam tahap ini menggunakan *black box testing*.

5. Pendukung (*Support*)

Software yang digunakan untuk merancang aplikasi ini yaitu Visual Code, Xampp dan Mysql.

2.3. Metode Sistem Pakar

Metode *Forward Chaining* atau Runut Maju adalah salah satu metode inferensi yang sangat penting dalam sistem pakar. Lalu apa itu mesin inferensi? Secara singkatnya, metode inferensi adalah proses untuk menghasilkan informasi dari fakta yang diketahui / diasumsikan.

Informasi tersebut adalah *konklusi* logis atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi ini dilakukan dalam modul mesininferensi.

Tabel 1. Diagnosa

No ID	Diagnosa
1	Loding <i>Xiaomi</i> stuck di logo Animasi
2	HP Sering <i>restart</i> Sendiri
3	Tidak bisa masuk pada <i>menu</i> utama namun selang beberapa detik ponsel mengalami <i>hank</i> dan kemudian ponsel me- <i>restart</i> kembali
4	Tidak Bisa masuk <i>Recovery</i>
5	Hp seringkali mengalami <i>lag</i>
6	Memori <i>internal</i> pada hp <i>xiaomi</i> sering cepat penuh
7	Lupa <i>Password</i> akun <i>MI</i>
8	Hp <i>xiaomi</i> mengalami lupa <i>pola</i>
9	Data pada file manager tidak bisa di hapus
10	Ketika membuka aplikasi sering mengalami <i>force close</i>
11	Tema pada aplikasi <i>xiaomi</i> tidak bisa di pasang

Proses Analisa menghasilkan berupa Berbagai jenis Kerusakan *Handphone*. Jenis Kerusakan yang digunakan dalam penelitian ini terlihat pada tabel 2

Tabel 2. Daftar Kerusakan

Kode	Jenis Kerusakan
K1	<i>Bootlop</i>
K2	<i>Lemot/Lag</i>
K3	<i>Human Error</i>
K4	<i>Aplikasi/Software Error</i>

Tabel keputusan merupakan cara untuk mendokumentasikan pengetahuan. Tabel keputusan juga merupakan matrik kondisi yang dipertimbangkan dalam memprediksikan kaidah. Walaupun kaidah

secara langsung dapat dihasilkan keputusan tetapi untuk menghasilkan keputusan yang efisien terdapat satu langkah yang harus dibuat yaitu pohon keputusan. Dari pohon keputusan dapat di ketahui *atribut* (kondisi) yang dapat di reduksi sehingga menghasilkan kaidah yang efisien dan optimal (Kesumaningtyas, 2017). Contoh tabel dan pohon keputusan sebagai berikut:

Tabel 3. Keputusan

ID Diagnosa	Kerusakan			
	K1	K2	K3	K4
1	✓			
2	✓			
3	✓			
4	✓			
5		✓		
6		✓		
7			✓	
8			✓	
9				✓
10				✓
11				✓

Bentuk (*IF-THEN*) merupakan bentuk dari suatu kaidah sebagai hubungan dua implikasi *IF* (jika) bagian premis dan *THEN* (maka) bagian konklusi. Adapun *rule-rule* pakar sebagai berikut:

Tabel 4. Rule – rule Pada Pakar

QUEUE	RULE	KONKLUSI
1,2,3,4	R1	K1
5,6	R2	K2
7,8	R3	K3
9,10,11	R4	K4

Rule Keputusan:

1. *Rule 1*

IF Loading Xiaomi stuck di logo Animasi **AND** HP Sering restart Sendiri **AND** Tidak bisa masuk pada menu utama namun selang beberapa detik ponsel mengalami hang dan kemudian ponsel me-restart kembali **AND** Tidak Bisa masuk Recovery **THEN** Bootlop

2. *Rule 2*

IF Hp seringkali mengalami lag **AND** Memori internal pada hp xiaomi sering cepat penuh **THEN** Lag

3. *Rule 3*

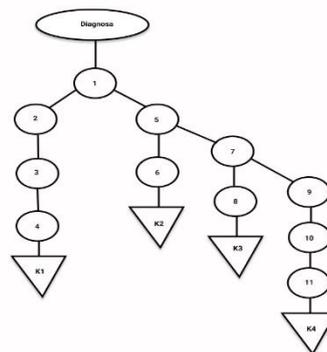
IF Lupa Password akun MI **AND** Hp xiaomi mengalami lupa pola **THEN** Human Error

4. *Rule 4*

IF Data pada file manager tidak bisa di hapus **AND** Ketika membuka aplikasi sering

mengalami **fc AND** Tema pada aplikasi xiaomi tidak bisa di pasang **THEN** Aplikasi/Software Error

Suatu pohon keputusan adalah *hierarki* struktur yang terdiri dari node (simpul) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan *node*. Pohon keputusan pakar yang digunakan pada sistem pakar (Merlina, 2016), pakar ini adalah sebagai berikut:



Gambar 1. Pohon Keputusan Pakar

3. Hasil dan Pembahasan

Hasil dari penelitian yaitu Sebuah Aplikasi konsultasi kerusakan yang membantu masyarakat dan Teknisi dalam menangani Suatu kerusakan pada *Smartphone Android* secepat mungkin, dengan proses Diagnosa Kerusakan Untuk mengetahui jenis kerusakan.

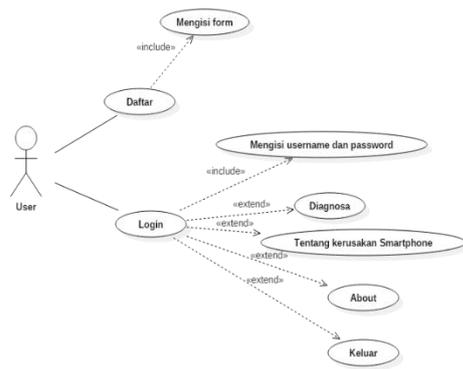
3.1 Analisis dan Perancangan Sistem

3.1.1. Use Case Diagram

Menggambarkan sejumlah external actors dan hubungan nya ke usecase yang diberikan oleh sistem. Usecase adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari usecase symbol namun dapat juga dilakukan dalam activity diagram. Usecase digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada didalam sistem (Pratama, 2019).

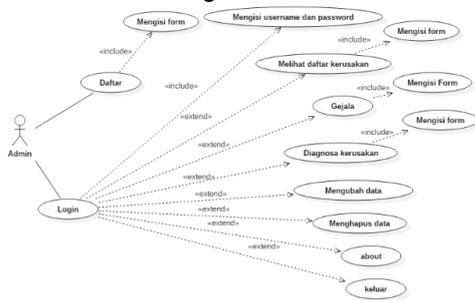
Pada penelitian ini terdapat dua *user* yaitu *User* dan *Admin* yang masing-masing dari *user* memiliki aplikasi yang terpisah. Berikut *usecase* diagramnya:

1. Use Case Diagram User



Gambar 2. Use Case User

2. Use Case Diagram Admin



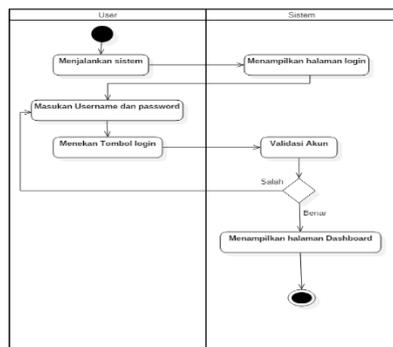
Gambar 3. Use Case Admin

3.1.2. 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.

1. Activity Diagram User

a) Activity Diagram Login

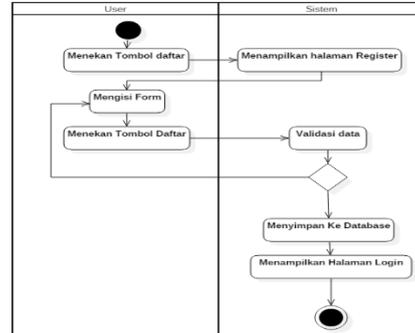


Gambar 5. Activity Login

Di *activity diagram* ini, *user* akan masuk ke Web *Login* terlebih dahulu, kemudian memasukkan *username* dan *password* lalu sistem akan mengecek ke

database apakah data yang dimasukkan oleh *user* sesuai atau tidak. Jika sesuai maka akan masuk ke halaman utama dan jika tidak akan kembali ke halaman *login* untuk mengisi data yang sesuai

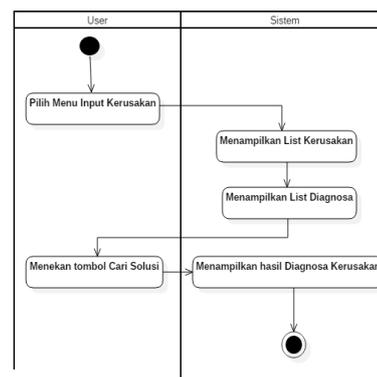
b) Activity Diagram Register



Gambar 4. Activity Register

Di halaman *register* untuk guru, maka guru akan memasukkan data nama, email, mata pelajaran dan *password*. Halaman *register* untuk *user* maka *user* akan memasukkan data nama, *username* dan *password*. Ketika data yang dimasukkan telah sesuai maka sistem akan menyimpan data tersebut ke *database*. Jika terdapat kesalahan seperti nama yang dimasukkan tidak sesuai karena sama maka sistem akan kembali ke halaman *register* agar *user/admin* dapat memasukkan data yang benar.

c) Activity Diagram Diagnosa kerusakan

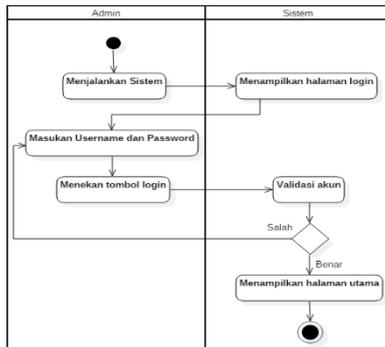


Gambar 6. Activity Diagnosa kerusakan

Pada *Activity* ini, *User* akan melakukan *Diagnosa Kerusakan* Dengan Memilih menu yang sudah di sediakan.

2. Activity Diagram Admin

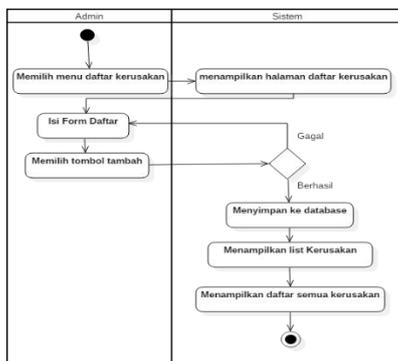
a) Activity Diagram Login



Gambar 8. Activity Login

Di *activity diagram* ini, *Admin* akan masuk ke *Web Login* terlebih dahulu, kemudian memasukkan *username* dan *password* lalu sistem akan mengecek ke *database* apakah data yang dimasukkan oleh *Admin* sesuai atau tidak. Jika sesuai maka akan masuk ke halaman utama dan jika tidak akan kembali ke halaman *login* untuk mengisi data yang sesuai.

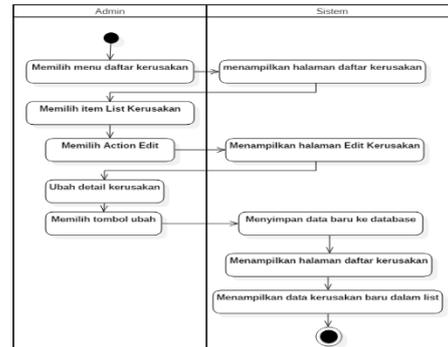
b) Activity Diagram tambah kerusakan



Gambar 10. Activity tambah kerusakan

Pada *activity diagram* tambah Kerusakan ini, admin akan menambahkan Kerusakan, yang disimpan di *database* dan kemudian bisa diakses oleh admin dan ditampilkan di aplikasi untuk user. Setelah melakukan *login* admin bisa memilih menu Kerusakan kemudian pilih mata pelajaran, kemudian sistem akan menampilkan halaman list Kerusakan, kemudian admin memilih tombol tambah Kerusakan maka data tugas seperti Kode kerusakan, nama kerusakan, dan tipe device, kemudian simpan maka sistem akan menambah data yang ada di *database* dan menampilkan halaman list Kerusakan Kembali.

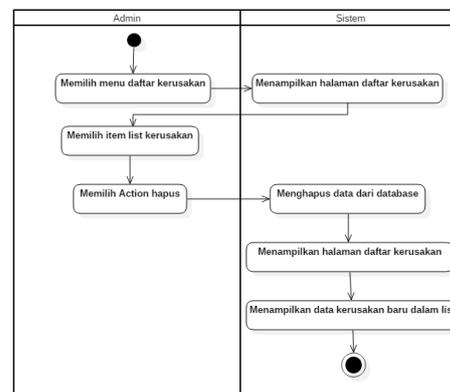
c) Activity Diagram ubah daftar kerusakan



Gambar 11. Activity ubah Daftar kerusakan

Pada *activity diagram* ubah kerusakan, hanya bisa dilakukan oleh admin. Bertujuan untuk mengoreksi atau mengubah detail kerusakan bila ada kesalahan. Sebelumnya admin harus *login* terlebih dahulu dan memilih menu kerusakan, sistem akan menampilkan halaman list kerusakan, admin memilih item kerusakan yang ingin diubah, sistem akan menampilkan halaman detail kerusakan, ubah detail kerusakan dan pilih tombol simpan, sistem akan menyimpan data baru di *database*

d) Activity Diagram Hapus kerusakan

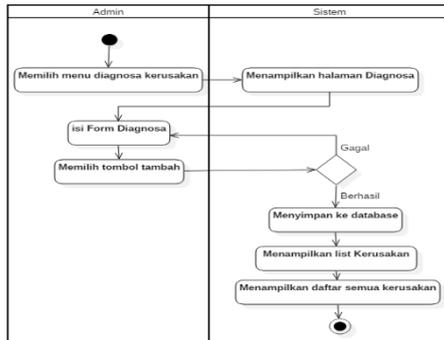


Gambar 12. Activity Hapus kerusakan

Pada *activity diagram* hapus kerusakan, hanya bisa dilakukan oleh admin. Bertujuan untuk menghapus kerusakan yang sudah tidak berlaku agar tidak menumpuk di *database*. Sebelumnya admin harus *login* terlebih dahulu dan memilih menu kerusakan, kemudian sistem akan menampilkan halaman list kerusakan, admin memilih item tugas yang ingin dihapus, pilih tombol hapus,

maka sistem akan menghapus data yang ada di *database*.

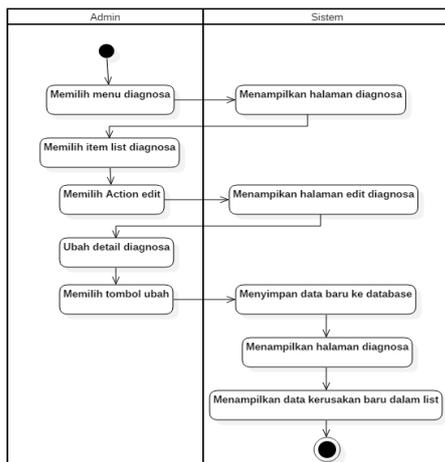
e) Activity Diagram tambah diagnosa



Gambar 13. Activity Tambah Diagnosa

Pada *activity* diagram tambah diagnosa ini, admin akan menambahkan Diagnosa, yang disimpan di *database* dan kemudian bisa diakses oleh admin dan ditampilkan di aplikasi untuk user. Setelah melakukan *login* admin bisa memilih menu diagnosa kemudian pilih mata pelajaran, kemudian sistem akan menampilkan halaman list diagnosa, kemudian admin memilih tombol tambah diagnosa maka data tugas seperti nama kerusakan dan , nama diagnosa, kemudian simpan maka sistem akan menambah data yang ada di *database* dan menampilkan halaman list diagnosa Kembali.

f) Activity Diagram ubah daftar diagnosa

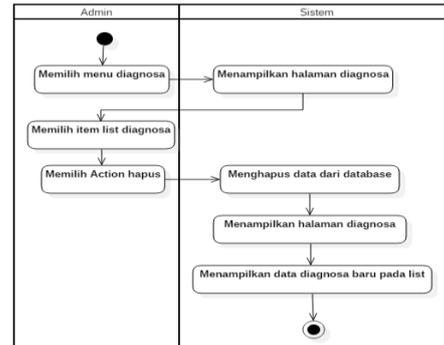


Gambar 14. Ubah Daftar diagnosa

Pada *activity* diagram ubah diagnosa, hanya bisa dilakukan oleh admin. Bertujuan untuk mengoreksi atau mengubah detail diagnosa bila ada kesalahan. Sebelumnya

admin harus *login* terlebih dahulu dan memilih menu diagnosa, sistem akan menampilkan halaman list diagnosa, admin memilih item diagnosa yang ingin diubah, sistem akan menampilkan halaman detail diagnosa, ubah detail diagnosa dan pilih tombol simpan, sistem akan menyimpan data baru di *database*

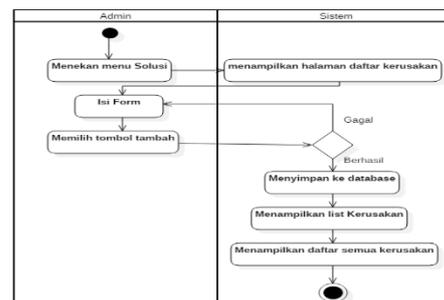
g) Activity Diagram Hapus daftar diagnose



Gambar 15. Activity Hapus Daftar diagnose

Pada *activity* diagram hapus diagnosa, hanya bisa dilakukan oleh admin. Bertujuan untuk menghapus diagnosa yang sudah tidak berlaku agar tidak menumpuk di *database*. Sebelumnya admin harus *login* terlebih dahulu dan memilih menu diagnosa, kemudian sistem akan menampilkan halaman list diagnosa, admin memilih item tugas yang ingin dihapus, pilih tombol hapus, maka sistem akan menghapus data yang ada di *database*.

h) Activity Diagram tambah Solusi

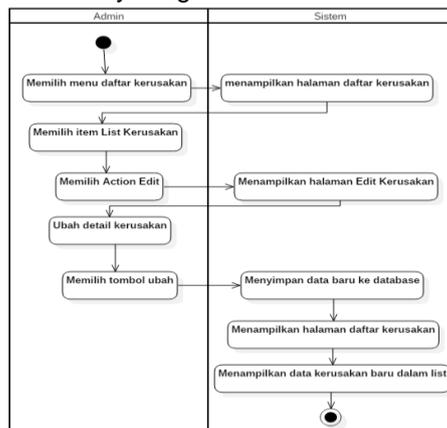


Gambar 16. Activity Tambah Solusi

Pada *activity* diagram tambah solusi ini, admin akan menambahkan solusi, yang disimpan di *database* dan kemudian bisa diakses oleh admin dan ditampilkan di aplikasi untuk user. Setelah melakukan *login* admin bisa memilih menu solusi kemudian sistem akan menampilkan halaman list

solusi, kemudian admin memilih tombol tambah solusi maka data solusi seperti nama diagnosa dan solusi, kemudian simpan maka sistem akan menambah data yang ada di *database* dan menampilkan halaman list solusi baru.

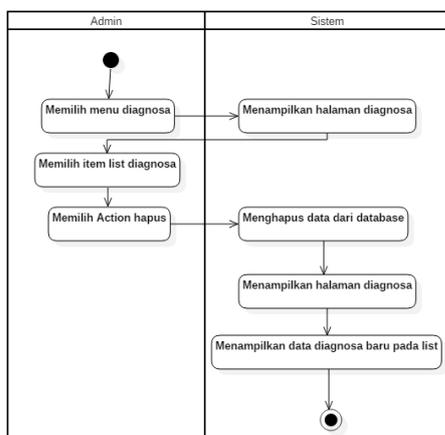
i) Activity Diagram ubah daftar Solusi



Gambar 17. Activity Ubah Solusi

Pada *activity* diagram ubah solusi, hanya bisa dilakukan oleh admin. Bertujuan untuk mengoreksi atau mengubah detail solusi bila ada kesalahan. Sebelumnya admin harus *login* terlebih dahulu dan memilih menu solusi, sistem akan menampilkan halaman list solusi, admin memilih item solusi yang ingin diubah, sistem akan menampilkan halaman detail solusi, ubah detail solusi dan pilih tombol simpan, sistem akan menyimpan data baru di *database*

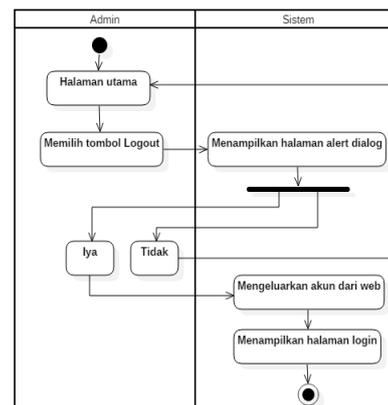
j) Activity Diagram Hapus daftar Solusi



Gambar 18. Activity Hapus Solusi

Pada *activity* diagram hapus solusi, hanya bisa dilakukan oleh admin. Bertujuan untuk menghapus solusi yang sudah tidak berlaku agar tidak menumpuk di *database*. Sebelumnya admin harus *login* terlebih dahulu dan memilih menu solusi, kemudian sistem akan menampilkan halaman list solusi, admin memilih item tugas yang ingin dihapus, pilih tombol hapus, maka sistem akan menghapus data yang ada di *database*.

k) Activity Diagram Keluar



Gambar 19. Activity Keluar

Kerika *admin* keluar, maka sistem akan mengeluarkan akun yang ada pada aplikasi tersebut.

3.2. Perancangan Basis Data

3.2.1. Entity Relationship Diagram

Diagram E-R digunakan untuk menggambarkan hubungan suatu entitas terhadap entitas lain, dimana entitas merupakan media untuk menghasilkan *instance* yang harus memiliki atribut yang dapat menjelaskan entitas tersebut (Krismawati, 2016).

ERD adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi (Sidik, 2016). ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

1. Entaity

Entitas adalah objek dalam dunia nyata yang dapat dibedakan dengan objek lain, sebagai contoh mahasiswa, dosen, departemen. Entitas terdiri atas beberapa atribut sebagai contoh atribut dari entitas mahasiswa adalah nim, nama, alamat, email,

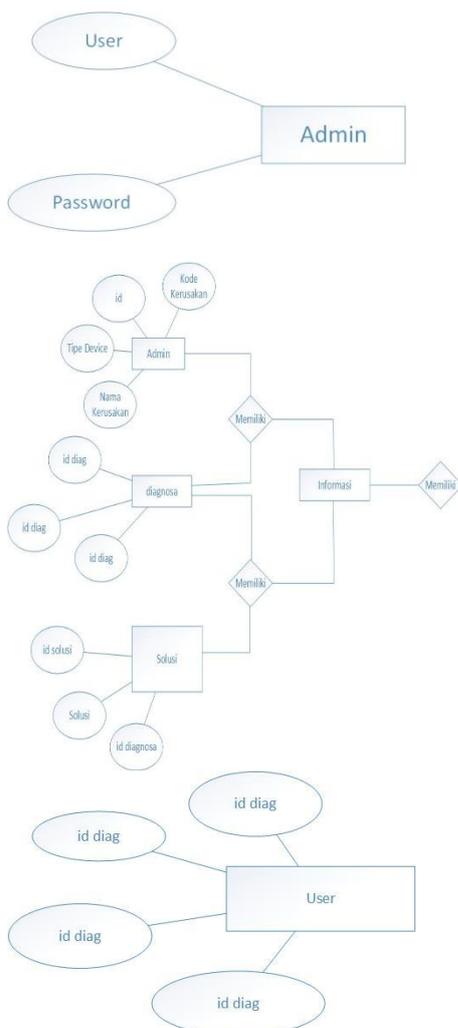
dll. Atribut nim merupakan unik untuk mengidentifikasi atau membedakan mahasiswa yg satu dengan yg lainnya. Pada setiap entitas harus memiliki 1 atribut unik atau yang disebut dengan *primary key*.

2. Attribute

Atribut adalah Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

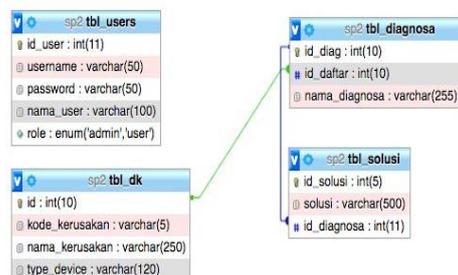
3. Relation

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda



Gambar 20. Rancangan ERD

3.2.2. Relasi Antar Tabel



Gambar 21. Relasi Antar Tabel

3.2.3. Arsitektur Sistem



Gambar 22. Arsitektur Sistem

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan telah diuraikan dalam laporan tentang pembuatan teknologi Sistem Pakar Berbasis web, dengan ini maka dapat disimpulkan beberapa hal sebagai berikut:

1. Proses perancangan aplikasi *Expert System* Berbasis web Menggunakan Metode *Forward Chaining* dilakukan dengan lima tahapan prosedur, pertama Analisa Kebutuhan, kedua Desain, ketiga Pembuatan Kode, keempat Integrasi dan Pengujian, dan yang kelima *Support*.
2. Aplikasi *Expert System* Diagnosakerusakan Smartphone Dalam Penelitian ini dengan menggunakan Platform *Website* dengan bahasa pemrograman PHP.
3. Berdasarkan pengujian aplikasi dengan menggunakan *Black Box* maka aplikasi *Expert System* Diagnosa kerusakan Smartphone layak digunakan karena menggunakan Aplikasi Berbasis web dengan sistem yang mudah dikelola oleh user.

Referensi

- Ayu, F., & Permatasari, N. (2018). Perancangan Sistem Informasi Pengolahan Data Praktek Kerja Lapangan (Pk) Pada Devisi Humas Pt. Pegadaian. *Intra-Tech*, 2(2), 12–26.
- Destiningrum, M., & Adrian, Q. J. (2017).

- Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre). *Jurnal Teknoinfo*, 11(2), 30. <https://doi.org/10.33365/jti.v11i2.24>
- Fazrie, M. (2017). Aplikasi Android Pendeteksi Masalah Komputer Berbasis Windows. *Faktor Exacta*, 10(2), 144–153.
- Fridayanthie, E. W., & Charter, J. (2016). *ISSN 1978-2136 | Rancang Bangun Sistem ... ISSN 1978-2136 | Rancang Bangun Sistem ... XIII(2)*, 63–71.
- IndoSite. (2017a). *Pengertian MySQL*. <https://www.indosite.com/pengertian-mysql/>
- IndoSite. (2017b). *Pengertian Tentang PHP*. <https://www.indosite.com/pengertian-tentang-php/>
- Krismawati. (2016). *PERANCANGAN SISTEM INFORMASI NILAI SISWA BERBASIS WEB PADA SMK NUSA JAYA TANGERANG*. <https://widuri.raharja.info/index.php?title=SI1211473306>
- Kurnia, L., & Setiyanto, N. A. (2017). Perangkat Bantu Pendeteksi Kerusakan Smartphone Android Jaringan 3G menggunakan Metode Forward Chaining Support Device to Detect Any 3G Android Smartphone Damage Using Forward Chaining Method. *Jl. Nakula No 5-11 Semarang*, 9(1), 3517261.
- Pangkey, M., Poekoel, V., & Lantang, O. (2016). Sistem Pakar Pendeteksi Kerusakan Handphone Berbasis Android. *Jurnal Teknik Informatika*, 8(1). <https://doi.org/10.35793/jti.8.1.2016.12825>
- Pratama, A. R. (2019). *Belajar Unified Modeling Language (UML) - Pengenalan*. <https://www.codepolitan.com/unified-modeling-language-uml>
- Sartini. (2015). *SISTEM PAKAR IDENTIFIKASI KERUSAKAN HARDWARE HANDPHONE DENGAN MENGGUNAKAN METODE FORWARD CHAINING. I NO. 2*.
- Sidik, B. F. (2016). *Mengenal Entity Relationship Diagram*. <https://www.kursuswebsite.org/mengenal-entity-relationship-diagram-2/>
- Wiguna, A. S., & Harianto, I. (2017). Sistem Pakar Diagnosa Kerusakan Sepeda Motor Matic Injeksi Menggunakan Metode Forward Chaining Berbasis Android. *SMARTICS Journal*, 3(1), 25–30. <https://doi.org/10.21067/smartics.v3i1.1933>
- Wijaya, B., & Tanamal, R. (2019). Rancang Bangun Aplikasi Sistem Pakar Berbasis Android Menggunakan Metode Forward Chaining Untuk Mendiagnosis Kerusakan Pada Hardware Laptop. *Teknika*, 8(1), 25–35. <https://doi.org/10.34148/teknika.v8i1.150>
- Ayu, F., & Permatasari, N. (2018). Perancangan Sistem Informasi Pengolahan Data Praktek Kerja Lapangan (Pkl) Pada Devisi Humas Pt. Pegadaian. *Intra-Tech*, 2(2), 12–26.
- Destiningrum, M., & Adrian, Q. J. (2017). Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre). *Jurnal Teknoinfo*, 11(2), 30. <https://doi.org/10.33365/jti.v11i2.24>
- Fazrie, M. (2017). Aplikasi Android Pendeteksi Masalah Komputer Berbasis Windows. *Faktor Exacta*, 10(2), 144–153.
- Fridayanthie, E. W., & Charter, J. (2016). *ISSN 1978-2136 | Rancang Bangun Sistem ... ISSN 1978-2136 | Rancang Bangun Sistem ... XIII(2)*, 63–71.
- IndoSite. (2017a). *Pengertian MySQL*. <https://www.indosite.com/pengertian-mysql/>
- IndoSite. (2017b). *Pengertian Tentang PHP*. <https://www.indosite.com/pengertian-tentang-php/>
- Krismawati. (2016). *PERANCANGAN SISTEM INFORMASI NILAI SISWA BERBASIS WEB PADA SMK NUSA JAYA TANGERANG*. <https://widuri.raharja.info/index.php?title=SI1211473306>
- Kurnia, L., & Setiyanto, N. A. (2017). Perangkat Bantu Pendeteksi Kerusakan Smartphone Android Jaringan 3G menggunakan Metode Forward Chaining Support Device to Detect Any 3G Android Smartphone Damage Using Forward Chaining Method. *Jl. Nakula No 5-11 Semarang*, 9(1), 3517261.
- Pangkey, M., Poekoel, V., & Lantang, O. (2016). Sistem Pakar Pendeteksi Kerusakan Handphone Berbasis

- Android. *Jurnal Teknik Informatika*, 8(1).
<https://doi.org/10.35793/jti.8.1.2016.12825>
- Pratama, A. R. (2019). *Belajar Unified Modeling Language (UML) - Pengenalan*.
<https://www.codepolitan.com/unified-modeling-language-uml>
- Sartini. (2015). *SISTEM PAKAR IDENTIFIKASI KERUSAKAN HARDWARE HANDPHONE DENGAN MENGGUNAKAN METODE FORWARD CHAINING. I NO. 2*.
- Sidik, B. F. (2016). *Mengenal Entity Relationship Diagram*.
<https://www.kursuswebsite.org/mengenal-entity-relationship-diagram-2/>
- Wiguna, A. S., & Harianto, I. (2017). Sistem Pakar Diagnosa Kerusakan Sepeda Motor Matic Injeksi Menggunakan Metode Forward Chaining Berbasis Android. *SMARTICS Journal*, 3(1), 25–30.
<https://doi.org/10.21067/smartics.v3i1.1933>
- Wijaya, B., & Tanamal, R. (2019). Rancang Bangun Aplikasi Sistem Pakar Berbasis Android Menggunakan Metode Forward Chaining Untuk Mendiagnosis Kerusakan Pada Hardware Laptop. *Teknika*, 8(1), 25–35.
<https://doi.org/10.34148/teknika.v8i1.150>