

## RANCANG BANGUN APLIKASI DIAGNOSIS KERUSAKAN MESIN *PHOTOCOPY* DENGAN MENGGUNAKAN METODE *DEMPSTER SHAFER* DI SISWA *FOTOCOPY & PRINTING KOTA BLITAR*

Adam Bachtiar, Sri Lestanti, Saiful Nur Budiman

Teknik Informatika, Universitas Islam Blitar

Jalan Majapahit No 2-4 Blitar, Indonesia

adambachtiarc@gmail.com

### ABSTRAK

Mesin photocopier adalah perangkat penting dalam berbagai lingkungan kerja. Namun, kerusakan yang tidak terduga dapat menyebabkan gangguan operasional yang signifikan dan biaya perbaikan yang tinggi. Dalam konteks ini, pengembangan aplikasi diagnosis kerusakan mesin photocopier menjadi krusial untuk meningkatkan efisiensi pemeliharaan dan kinerja perangkat. Metode Dempster-Shafer adalah salah satu pendekatan yang menjanjikan dalam mengatasi ketidakpastian dalam diagnosis. Tujuan utama dari penelitian ini adalah merancang dan mengembangkan sebuah aplikasi berbasis metode Dempster-Shafer untuk diagnosis kerusakan mesin photocopier. Aplikasi ini bertujuan untuk memberikan diagnosis yang seakurat mungkin dan dapat diandalkan terhadap kerusakan-kerusakan yang umum terjadi pada mesin photocopier. Penelitian ini menggunakan metode pengumpulan data mengenai kerusakan mesin photocopier, mengidentifikasi berbagai indikator kerusakan, serta memanfaatkan teori Dempster-Shafer untuk menggabungkan bukti-bukti dan memberikan diagnosis. Aplikasi ini dikembangkan menggunakan bahasa pemrograman tertentu dan dilengkapi dengan antarmuka pengguna yang mudah digunakan. Hasil dari penelitian ini adalah aplikasi diagnosis kerusakan mesin photocopier yang mampu memberikan diagnosis dengan tingkat akurasi yang signifikan. Aplikasi ini dapat mengidentifikasi kerusakan mesin photocopier dengan mempertimbangkan berbagai indikator dan meminimalkan tingkat ketidakpastian. Pengujian aplikasi ini telah menunjukkan hasil yang menggembirakan dalam memberikan diagnosis yang tepat. Pengujian alpha testing telah dilakukan untuk mengevaluasi fungsionalitas dasar dan kelayakan aplikasi sebelum dirilis secara luas. Penelitian ini berhasil merancang dan mengembangkan aplikasi diagnosis kerusakan mesin photocopier berdasarkan metode Dempster-Shafer. Aplikasi ini memiliki potensi besar untuk meningkatkan pemeliharaan mesin photocopier, mengurangi biaya perbaikan yang tidak perlu, dan meningkatkan efisiensi operasional. Dengan demikian, aplikasi ini dapat menjadi alat yang sangat berharga dalam konteks pemeliharaan mesin photocopier di berbagai lingkungan kerja.

**Kata kunci :** Perancangan, Diagnosis, Mesin *Photocopy*, *Dempster Shafer*, *Alpha testing*

### 1. PENDAHULUAN

Perkembangan dalam duplikasi dokumen telah mengalami perubahan yang sangat signifikan, terutama dengan hadirnya mesin *photocopy*. Mesin ini digunakan untuk membuat salinan dari berbagai sumber, seperti buku, dokumen, dan lain-lain. Saat ini, mesin *photocopy* juga dapat digunakan untuk mencetak dokumen dari berkas digital. Namun, mesin *photocopy* juga sering mengalami kerusakan yang mengurangi produktivitas dan efisiensi. Menurut pendapat pengusaha yang bergerak di bidang *photocopy*, mereka mengalami berbagai permasalahan dalam mengidentifikasi kerusakan mesin *photocopy*. Berdasarkan kejadian lapangan di Siswa *Fotocopy & Printing*, diketahui bahwa untuk mengidentifikasi kerusakan mesin *photocopy* masih dilakukan secara manual atau dalam istilah lain disebut dengan *trial and error*.

Macam-macam Kendala yang dihadapi dalam mengidentifikasi kerusakan mesin *photocopy* yaitu sulitnya menemukan titik masalah yang mengalami kerusakan sehingga harus mengecek semua komponen dan dibutuhkan waktu yang banyak dan biaya yang cukup tinggi, jika memanggil teknisi dari luar dan juga

tidak setiap hari dan setiap saat teknisi bisa datang, hanya beberapa dari karyawan *photocopy* siswa yang dapat mengatasi kerusakan ringan pada mesin *photocopy*. Untuk mengetahui kerusakan mesin *photocopy* tersebut perlu dirancang sebuah aplikasi yang dapat membantu mendiagnosis kerusakan mesin *photocopy*. Salah satu cara yang dapat dipakai untuk mendiagnosis kerusakan mesin *photocopy* yaitu dengan Metode *Dempster Shafer*. *Dempster Shafer* digunakan karena pendekatan yang menggabungkan dan menyebarkan ketidakpastian, yang memiliki beragam karakteristik yang sesuai dengan pemikiran seorang pakar secara intuitif, tetapi didasarkan pada dasar matematika yang solid.

Metode ini diterapkan untuk menentukan dan mendefinisikan tingkat keyakinan serta fungsi yang rasional dalam mengevaluasi suatu kemungkinan [1]. Proses pengembangan aplikasi menggunakan model SDLC Waterfall, yang memiliki tahapan terstruktur seperti analisis kebutuhan, perancangan, penerapan, pengujian, dan pemeliharaan. Metode Waterfall dipilih karena proses yang jelas dan terstruktur, cocok untuk aplikasi diagnosis kerusakan mesin *photocopy* dengan metode *Dempster Shafer*. Tahapan analisis kebutuhan

dan desain dilakukan dengan studi literatur dan penelitian untuk menentukan gejala serta metode Dempster Shafer. Implementasi dan pengujian dilakukan dengan menerapkan metode Dempster Shafer dan menguji aplikasi untuk efektivitas dan akurasi. Pemeliharaan dilakukan dengan perbaikan dan pembaruan seiring perkembangan teknologi. Dempster-Shafer dipilih karena kemampuannya mengatasi ketidakpastian, mengintegrasikan berbagai jenis bukti, dan memberikan diagnosis yang lebih akurat dalam aplikasi diagnosis kerusakan mesin photocopy. Penelitian ini bertujuan untuk mengatasi masalah pada mesin photocopy dengan membuat aplikasi diagnosis berbasis metode Dempster Shafer.

**2. TINJAUAN PUSTAKA.**

**2.1. Mesin Photocopy**

Mesin *photocopy* adalah suatu mesin yang dibuat untuk menggandakan dokumen menjadi beberapa rangkap dokumen hitam putih. Mesin *photocopy* memiliki peran yang krusial dalam kebutuhan administratif individu, sehingga pada saat ini, setiap aspek administrasi membutuhkan mesin fotokopi untuk menggandakan dokumen[2].

Salah satu contoh tipe mesin *photocopy* yaitu Canon dengan tipe *IR 5000* dan Canon tipe *IR 6000*. Canon tipe *IR 5000* dan *IR 6000* adalah salah satu jenis mesin fotocopy terbaru dari Canon yang menawarkan kualitas mesin yang cukup baik dan harga yang lumayan terjangkau. Kecepatannya mencapai 60 lembar per menit, menjadikannya sangat ideal untuk copy dalam jumlah besar. Selain itu, mesin *photocopy* Canon *IR 5000* dan *IR 6000* khusus didesain untuk penggunaan copy dengan skala besar yang membutuhkan kehandalan.

**2.2. Dempster Shafer**

Dempster shafer adalah sebuah teori gabungan yang dapat mengatasi ketidakpastian dalam pengambilan keputusan dengan memodelkan tingkat keyakinan dalam berbagai aspek[3]. Metode *Dempster Shafer* merupakan suatu metode matematika yang secara umum digunakan untuk menghitung kemungkinan atau probabilitas. Metode ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer. Dalam menghadapi suatu permasalahan, sering ditemukan jawaban yang tidak memiliki kepastian penuh[4]. Metode tersebut merupakan sebuah bentuk umum dari teori probabilitas *Bayesian* subjektif, di mana untuk setiap pertanyaan yang ingin dijawab, diperlukan probabilitas yang sesuai dengan kebutuhan. Fungsi kepercayaan dalam metode ini didasarkan pada tingkat keyakinan (percaya atau tidak percaya) terhadap jawaban dari pertanyaan tersebut, yang kemudian digunakan untuk menghitung probabilitas jawaban dari pertanyaan tersebut [5]Aturan yang digunakan untuk menggabungkan bukti tersebut dikenal sebagai *Dempster's Rule of Combination*. Metode *Dempster Shafer* digunakan untuk mendapatkan kesimpulan dengan mencari

probabilitas setiap kerusakan berdasarkan nilai densitas gejala yang ada. Berikut adalah rumus belief atau kepercayaan :

$$Bel(x) = \sum_{v \subset x} m1(X)$$

Sedangkan untuk rumus Plausibility atau ketidakpercayaan adalah:

$$Pls(X) = 1 - Bel(x)$$

Sehingga rumus *Dempster Shafer* adalah sebagai berikut :

$$m3(Z) = \frac{\sum_{X \cap Y = Z} m1(X).m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X).m2(Y)}$$

Keterangan:

X = Kerusakan dengan gejala A

Y = Kerusakan dengan gejala B

Bel(X) = Nilai belief suatu kerusakan yang mengalami gejala A

Pls(X) = Nilai plausibility yang didapat dari 1- bel

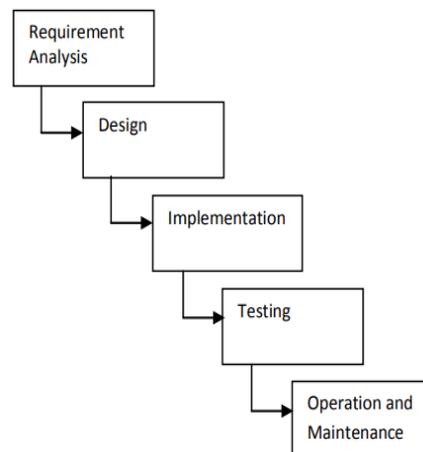
m1(X) = Tingkat kepercayaan dari suatu gejala

(Sumber : Kotimah, Chandra.,2022)

**2.3. Waterfall**

Sekitar tahun 1970, Winston Royce memperkenalkan model pengembangan perangkat lunak baru dengan nama Waterfall. Metode ini terus dipakai sampai sekarang hingga menjadi model paling populer dalam Software Engineering. Model air terjun atau disebut juga *waterfall* adalah suatu metode dalam pengembangan sistem yang mengadopsi pendekatan sekuensial. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing / verification, dan maintenance. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan[7].

Secara ringkas, model *waterfall* dapat dikatakan melibatkan fase-fase berikut : *requirement analysis, design, implementation, testing, and operation and maintenance* . Tahapan – tahapan metode SDLC *waterfall* digambarkan seperti berikut :



Gambar 1. Tahap - tahap pada SDLC *Waterfall* [8]

**2.4. Alpha testing**

Alpha testing merupakan bentuk pengujian perangkat lunak yang dilakukan oleh tim pengembang sebelum produk tersebut dirilis kepada pengguna eksternal. Pengujian ini berlangsung di lingkungan yang terkendali dan bertujuan untuk menemukan kecacatan atau bug yang mungkin terlewat selama proses pengembangan. Tujuan dari alpha testing adalah untuk menjamin bahwa perangkat lunak tersebut stabil, berfungsi dengan baik, dan memenuhi kebutuhan pengguna akhir.

Tahapan pengujian alpha pada proses pengembangan perangkat lunak dimulai dengan pengumpulan dan analisis kebutuhan pengguna terhadap produk yang akan dikembangkan. Setelah itu, dilakukan perencanaan dan persiapan untuk pengujian alpha, termasuk menentukan ruang lingkup pengujian, mengidentifikasi risiko, menyiapkan lingkungan pengujian, dan menyusun rencana pengujian. Selanjutnya, dilakukan desain pengujian yang melibatkan perancangan rincian skenario pengujian, yang mencakup uji coba fungsional dan pengujian non-fungsional, serta menentukan kriteria keberhasilan pengujian.

**3. METODE PENELITIAN**

**3.1. Analisa Kebutuhan**

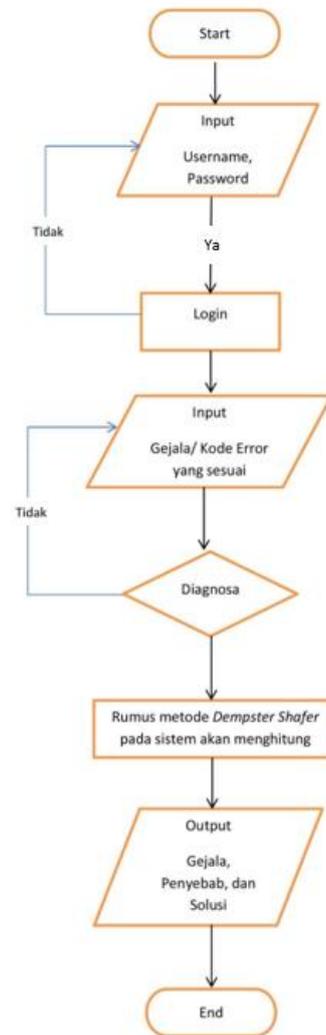
Kemajuan dalam bidang duplikasi dokumen telah mengalami perkembangan yang sangat penting, terutama dengan diperkenalkannya mesin fotokopi. Perangkat ini difungsikan untuk menghasilkan salinan dari berbagai jenis dokumen seperti buku, catatan, dan lainnya. Saat ini, mesin photocopy juga mampu mencetak dokumen dari file digital. Perangkat ini menjadi sangat vital dalam kegiatan administrasi dan pendidikan, serta ditemukan di berbagai lingkungan seperti perkantoran, lembaga pendidikan, dan pemerintahan. Kehadiran mesin fotokopi menjadi bagian integral dari infrastruktur sekolah, pemerintah, dan lembaga lainnya.

Peneliti memilih tempat ini karena selain tempat kerja peneliti, juga sistem mendiagnosis kerusakannya masih manual dan hanya bisa dilakukan oleh teknisi saja. Dalam penelitian ini diharapkan dapat memudahkan karyawan lainnya dalam melakukan diagnosis kerusakan mesin photocopy tanpa harus menunggu teknisi yang melakukannya dan dengan adanya aplikasi diagnosis kerusakan mesin photocopy berbasis web diharapkan dapat meminimalkan biaya maintenance pemilik mesin photocopy.

**3.2. Perancangan Sistem**

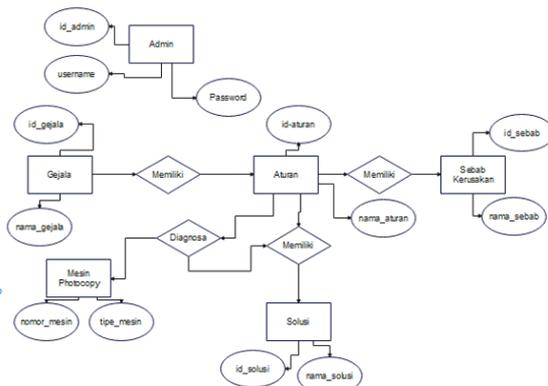
Pada tahap perancangan sistem, peneliti melakukan perancangan desain aplikasi diagnosis kerusakan mesin photocopy yaitu dengan menentukan bagaimana alur berjalan nya aplikasi tersebut dengan menggunakan flowchart, ERD, DFD Level 0, dan DFD Level 1

**3.3. Flowchart**



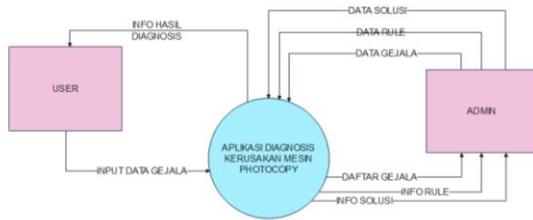
Gambar 2. Flowchart Software Design

**3.4. Entity Relationship Diagram**

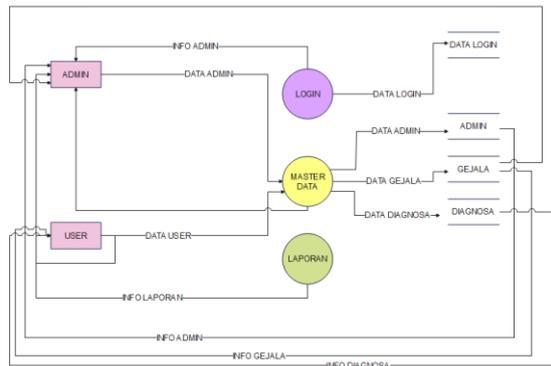


Gambar 3. Entity Relationship Diagram untuk Software Design

### 3.5. Data Flow Diagram



Gambar 4. Data Flow Diagram level 0 Aplikasi diagnosis kerusakan mesin *photocopy*



Gambar 5. Data Flow Diagram level 1 Aplikasi diagnosis kerusakan mesin *photocopy*

Berdasarkan pada gambar 5 dan gambar 6 proses pembuatan sistem aplikasi ini melibatkan beberapa entitas yang memiliki keterkaitan langsung dengan sistem tersebut. Entitas-entitas ini mencakup:

a. *Administrator*

*Administrator* berperan sebagai pembuat sistem aplikasi, dimana data-data yang akurat akan dimasukan dalam *database*. Selain itu *administrator* juga menyediakan fasilitas-fasilitas yang diperlukan untuk pengguna.

b. *User*

*User* yang berperan sebagai pengguna untuk mendiagnosis kerusakan pada mesin *photocopy* dengan bantuan aplikasi.

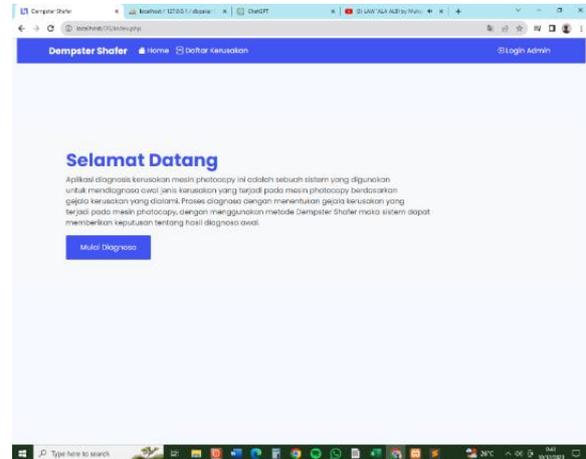
## 4. HASIL DAN PEMBAHASAN

### 4.1. Hasil

Tahap ini akan menguraikan mengenai penyusunan program untuk mendiagnosis kerusakan pada mesin *photocopy* dengan menggunakan metode *Dempster Shafer* yang berbasis web. Program ini dimaksudkan untuk dapat mengidentifikasi kerusakan dengan akurasi yang tinggi. Dikembangkan dengan menggunakan bahasa pemrograman web seperti HTML dan PHP, serta didukung oleh database MySQL. Hasil implementasi dari program untuk mendiagnosis kerusakan pada mesin *photocopy* dengan metode *Dempster-Shafer* ini akan dicapai setelah melewati serangkaian langkah dalam prosedur penelitian yang telah ditetapkan sebagai berikut:

#### 4.1.1. Halaman Home

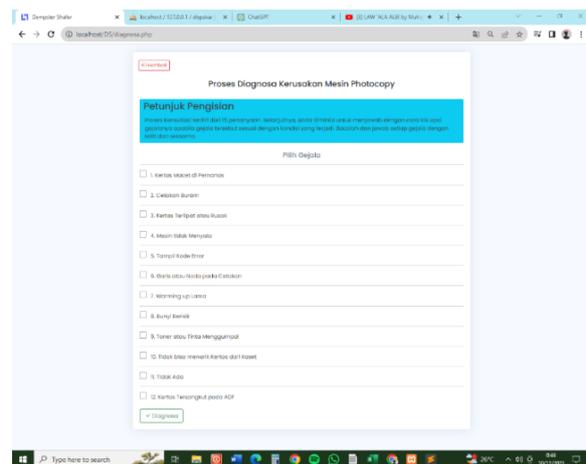
Tampilan halaman utama ini merupakan implementasi dari rancangan *home*. Pada gambar ini menjelaskan menu apa saja yang ada pada program Aplikasi Untuk Mendiagnosis Kerusakan Mesin *Photocopy* Menggunakan Metode *Dempster Shafer* ini. Bentuk dari halaman *home* ini dapat dilihat pada Gambar 6 berikut :



Gambar 6. Tampilan Halaman Home Aplikasi Diagnosis Kerusakan Mesin *Photocopy*

#### 4.1.2. Halaman Diagnosa

Pada bagian diagnosa berisi tentang bagaimana cara mendiagnosa dengan memberikan *checklist* pada gejala yang telah tersedia. Pilih gejala yang diinginkan lalu tekan *button* “Diagnosa” untuk menjalankan proses diagnosa. Hasil diagnosa akan keluar pada halaman hasil diagnosa sebagaimana yang ditampilkan pada Gambar 7 berikut :



Gambar 7. Tampilan Halaman Diagnosa Aplikasi Diagnosis Kerusakan Mesin *Photocopy*

#### 4.1.3. Halaman Hasil Diagnosa

Pada bagian hasil diagnosa ditampilkan hasil dari proses diagnosa yang telah dilakukan yaitu tampil kerusakan yang terjadi pada mesin *photocopy* beserta solusi yang dari kerusakan yang tampil pada halaman

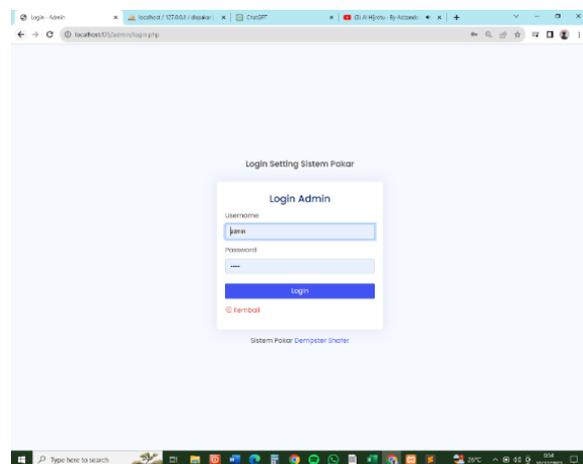
sebagaimana yang ditampilkan pada Gambar 8 berikut



Gambar 8. Tampilan Halaman Hasil Diagnosa Aplikasi Diagnosis Kerusakan Mesin Photocopy

**4.1.4. Halaman Login**

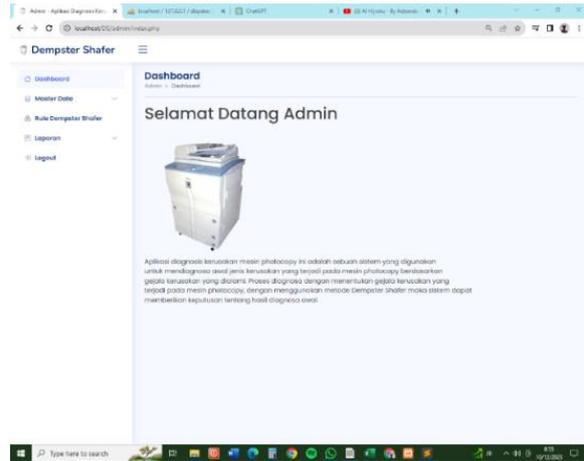
Pada Gambar 9 menunjukkan antarmuka halaman login yang dipergunakan oleh pengguna guna mengakses dashboard. Sebelum dapat melakukan login, pengguna harus mengisi kolom username dan password yang sudah terdaftar sebelumnya. Apabila formulir tersebut ditinggalkan tanpa diisi dan tombol login ditekan, akan timbul pesan kesalahan validasi yang meminta pengguna untuk melengkapi semua isian formulir sesuai dengan aturan yang telah ditetapkan. Peran pengguna dalam aplikasi ini adalah sebagai administrator yang mengelola data di dalamnya.



Gambar 9. Tampilan Halaman Login Admin Aplikasi Diagnosis Kerusakan Mesin Photocopy

**4.1.5. Halaman Dashboard Admin**

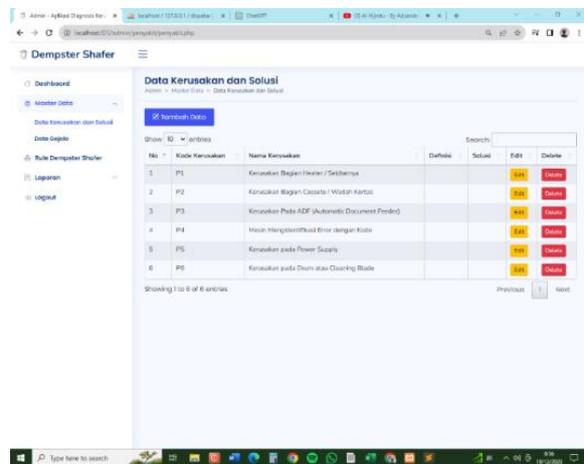
Gambar 10 menggambarkan keseluruhan dari halaman dashboard admin. Halaman ini menyajikan secara komprehensif informasi terkait ringkasan tujuan pengembangan aplikasi untuk mendiagnosis kerusakan pada mesin photocopy. Desain halaman dashboard ini bertujuan memberikan gambaran menyeluruh mengenai berbagai aspek yang terkait dengan aplikasi tersebut.



Gambar 10. Tampilan Halaman Dashboard Aplikasi Diagnosis Kerusakan Mesin Photocopy

**4.1.6. Halaman Data Kerusakan dan Solusi**

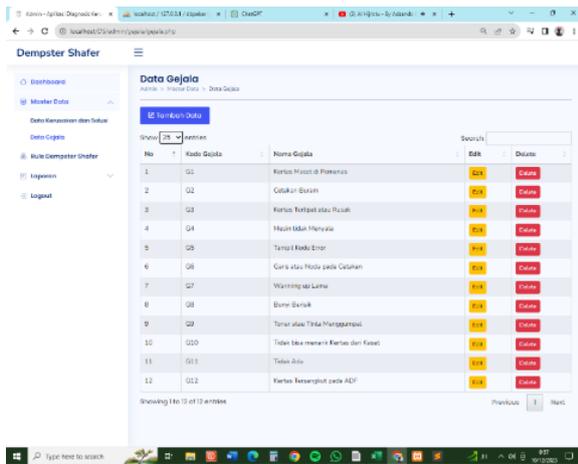
Pada Gambar 11 berisi Data Kerusakan dan solusi meliputi Kode Kerusakan, Nama Kerusakan, Definisi Kerusakan, dan Solusi untuk mengatasi Kerusakan pada Mesin Photocopy. Data kerusakan bisa ditambahkan dengan menekan button “Tambah” maka akan memunculkan pop-up untuk menambahkan data tersebut dan tersedia menu untuk mengedit dengan menekan button “Edit” dan menghapus data kerusakan tersebut dengan menekan button “Delete”.



Gambar 11. Tampilan Halaman Data Kerusakan dan Solusi Aplikasi Diagnosis Kerusakan Mesin Photocopy

**4.1.7. Halaman Data Gejala**

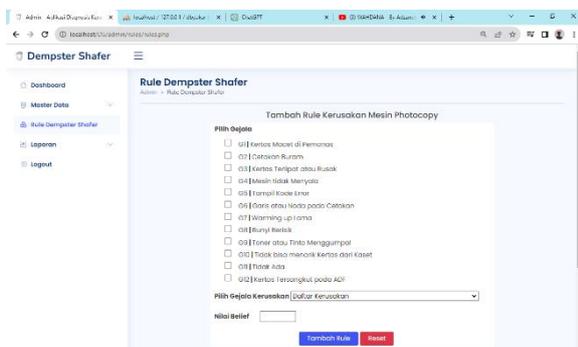
Pada Gambar 12 berisi Data Kerusakan dan solusi meliputi Kode Gejala, Nama Gejala, Data Gejala bisa ditambahkan dengan menekan button “Tambah” maka akan memunculkan pop-up untuk menambahkan data tersebut dan tersedia menu untuk mengedit dengan menekan button “Edit” dan menghapus data kerusakan tersebut dengan menekan button “Delete”.



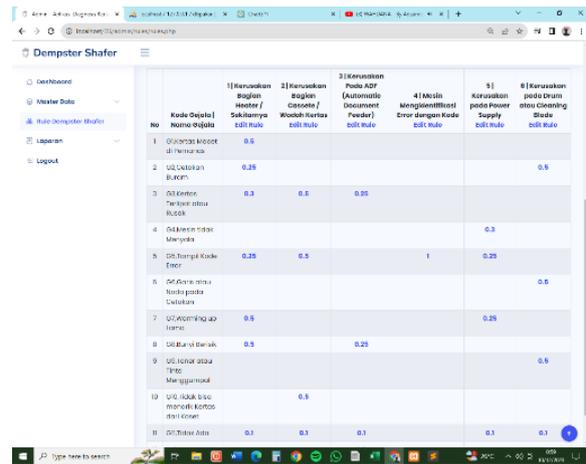
Gambar 12. Tampilan Halaman Data Gejala Aplikasi Diagnosis Kerusakan Mesin Photocopy

4.1.8. Halaman Rule Dempster Shafer

Pada Halaman Rule terdiri dari dua bagian yaitu bagian yaitu bagian penambahan rule kerusakan mesin photocopy dan tabel rule kerusakan mesin photocopy. Bagian penambahan rule berfungsi untuk menambahkan rule kerusakan mesin photocopy dengan cara memberi checklist pada gejala yang telah ditentukan lalu pilih kerusakan nya. Sedangkan pada bagian tabel berfungsi untuk menampilkan kerusakan dan gejala yang telah ditambahkan sebelumnya serta untuk mengedit nilai belief dan juga rule Dempster Shafer. Cara tersebut berguna untuk menentukan bagaimana rule Dempster Shafer diterapkan pada proses diagnosa yang dilakukan untuk mendiagnosa kerusakan mesin photocopy . Berikut gambar 13 dan 14 merupakan halaman rule Dempster Shafer :



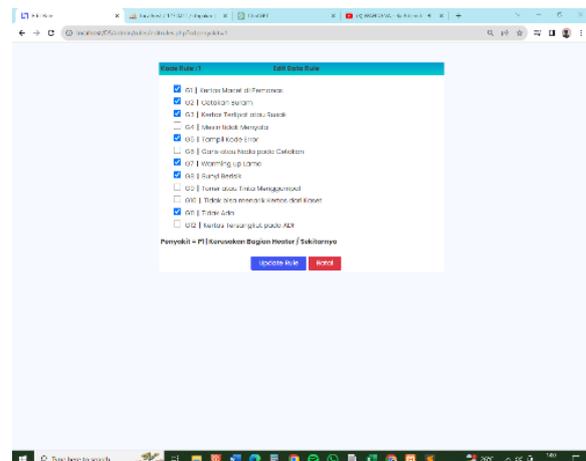
Gambar 13. Tampilan Halaman Rule Dempster Shafer Aplikasi Diagnosis Kerusakan Mesin Photocopy Bagian Penambahan Rule



Gambar 14. Tampilan Halaman Rule Dempster Shafer Aplikasi Diagnosis Kerusakan Mesin Photocopy Bagian Tabel Rule

4.1.9. Halaman Edit Rule

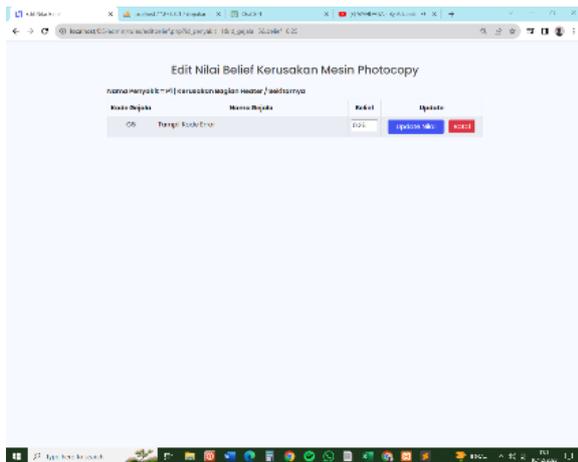
Pada gambar 15 merupakan halaman edit rule Dempster Shafer. Seperti yang telah penulis sampaikan sebelumnya, halaman ini akan muncul ketika admin meng-klik *linked text* "edit rule" pada bagian tabel rule. Halaman ini berfungsi untuk memperbaiki atau mengedit rule apabila rule yang telah ditulis pada tabel tidak sesuai dengan yang ditentukan.



Gambar 15. Tampilan Halaman Edit Rule Dempster Shafer Aplikasi Diagnosis Kerusakan Mesin Photocopy

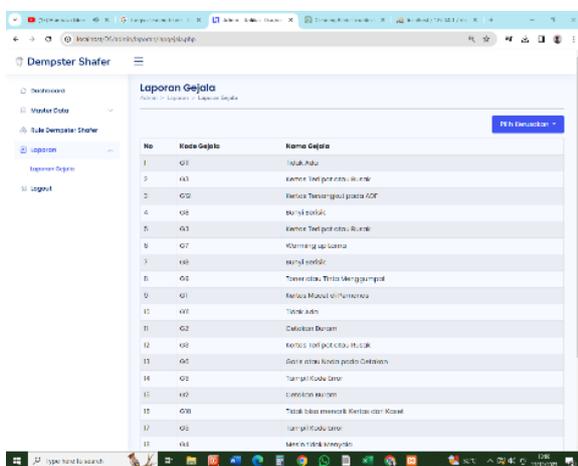
4.1.10. Halaman Edit Nilai Belief

Pada gambar 16 merupakan halaman edit nilai belief. Seperti yang telah dijelaskan sebelumnya, halaman ini akan muncul saat admin mengklik teks terhubung "edit nilai belief" di dalam tabel rule. Halaman edit nilai belief berperan dalam melakukan perbaikan atau pengeditan nilai belief jika nilai yang tercatat di dalam tabel tidak sesuai dengan yang telah ditentukan.



Gambar 16. Tampilan Halaman Edit Nilai Belief Aplikasi Diagnosis Kerusakan Mesin Photocopy

4.1.11. Halaman Laporan Gejala



Gambar 17. Tampilan Halaman Laporan Gejala Aplikasi Diagnosis Kerusakan Mesin Photocopy

Pada Halaman Laporan Gejala, tersedia informasi mengenai riwayat gejala yang telah

dimasukkan ke dalam aplikasi selama proses diagnosa. Halaman ini memberikan gambaran menyeluruh mengenai evolusi dan hasil diagnosa yang telah dilakukan sebelumnya. Halaman tersebut menunjukkan *interface* halaman rule *Dempster Shafer*. Ilustrasi tersebut memberikan pandangan lebih rinci mengenai peraturan dan variabel yang digunakan dalam proses pengambilan keputusan atau diagnosa berbasis Dempster Shafer. Halaman ini berperan sebagai pusat informasi penting yang mendukung pemahaman dan analisis lebih mendalam terkait metode dan hasil diagnosa yang diterapkan dalam aplikasi ini. Halaman Laporan Gejala dapat dilihat pada gambar 17.

4.2. PEMBAHASAN

Dari hasil yang telah disampaikan oleh penulis, selanjutnya aplikasi akan masuk dalam tahap pengujian perangkat lunak. Pengujian perangkat lunak yang digunakan adalah pengujian alfa (*alpha testing*). Pengujian alfa dimulai dengan tiga tahapan yaitu yang pertama adalah *pre alpha testing*, lalu *alpha testing*, dan terakhir adalah *post alpha testing*. Pada pengujian alfa (*alpha testing*) perlu menyiapkan beberapa langkah sebelum memasuki pengujian alfa yaitu menentukan skenario pengujian dan *test case* yang akan dibuat, menentukan metode pengujian, serta menentukan siapa saja yang akan menguji aplikasi. Pada Ketiga tahap pengujian akan dipaparkan sebagai berikut :

4.2.1. Pre Alpha testing

*Pre-alpha testing* adalah tahap pengujian yang dilakukan sebelum memasuki tahap *alpha testing*. Pada tahap ini, penulis akan menguji aplikasi menggunakan metode *blackbox* dengan tabel *test case* pengujian. Berikut merupakan *test case* yang telah dibuat untuk tahap pengujian ini :

Tabel 1. Test Case Pada Halaman Home

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Meng- Klik Tulisan Home	Tetap pada Halaman Home	Tetap pada Halaman Home	Valid
2	Meng -klik Button Mulai Diagnosa	Masuk ke Halaman Proses Diagnosa	Masuk ke Halaman Proses Diagnosa	Valid
3	Meng-Klik Tulisan Daftar Kerusakan	Masuk ke Halaman Daftar Kerusakan	Masuk ke Halaman Daftar Kerusakan	Valid

Tabel 2. Test Case Pada Halaman Diagnosa

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Memilih 1 gejala lalu klik button"Diagnosa"	Keluar notifikasi "Maaf, anda harus memilih minimal 2 gejala"	Keluar notifikasi "Maaf, anda harus memilih minimal 2 gejala"	Valid
2	Memilih 2 gejala lalu klik button"Diagnosa"	Diarahkan ke halaman hasil diagnosa	Diarahkan ke halaman hasil diagnosa	Valid

Tabel 3. Test Case Pada Halaman Daftar Kerusakan

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mem -blok teks dan menyalin teks	Teks dapat di salin dan di tempel	Teks dapat di salin dan di tempel	Valid
2	Meng- Klik button “Kembali”	Kembali ke halaman Home	Kembali ke halaman Home	Valid

Tabel 4. Test Case Pada Halaman Login

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Login tanpa username dan password	Keluar Notifikasi ‘Please Fill out the Field’	Keluar Notifikasi ‘Please Fill out the Field’	Valid
2	Login hanya menggunakan username	Keluar Notifikasi ‘Please Fill out the Field’	Keluar Notifikasi ‘Please Fill out the Field’	Valid
3	Login hanya menggunakan password	Keluar Notifikasi ‘Please Fill out the Field’	Keluar Notifikasi ‘Please Fill out the Field’	Valid
4	Login menggunakan username dan password	Masuk ke Halaman Dashboard admin	Masuk ke Halaman Dashboard admin	Valid

Tabel 5. Test Case Pada Halaman Data Kerusakan dan Solusi

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menulis salah satu huruf misal”a” pada search box	Keluar Data yang ada huruf tersebut	Keluar Data yang ada huruf tersebut	Valid
2	Meng-Klik Button “Delete” pada salah satu baris lalu klik ya	Data pada baris tersebut akan terhapus	Data pada baris tersebut akan terhapus	Valid
3	Meng-Klik Button “Edit” pada salah satu baris lalu meng edit isi nya dengan memberikan teks	Data akan ganti sesuai teks yang ditulis	Data akan ganti sesuai teks yang ditulis	Valid
4	Meng-Klik Button “Edit” pada salah satu baris lalu meng edit isi nya tanpa memberikan teks	Data pada baris yang di edit akan kosong	Data pada baris yang di edit akan kosong	Valid

Tabel 6. Test Case Pada Halaman Halaman Data Gejala

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menulis salah satu huruf misal”a” pada search box	Keluar Data yang ada huruf tersebut	Keluar Data yang ada huruf tersebut	Valid
2	Meng-Klik Button “Delete” pada salah satu baris lalu klik ya	Data pada baris tersebut akan terhapus	Data pada baris tersebut akan terhapus	Valid
3	Meng-Klik Button “Edit” pada salah satu baris lalu meng edit isi nya dengan memberikan teks	Data akan ganti sesuai teks yang ditulis	Data akan ganti sesuai teks yang ditulis	Valid
4	Meng-Klik Button “Edit” pada salah satu baris lalu meng edit isi nya tanpa memberikan teks	Data pada baris yang di edit akan kosong	Data pada baris yang di edit akan kosong	Valid

Tabel 7. Test Case Pada Halaman Rule Dempster Shafer

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menambahkan gejala dan Kerusakan dengan memberi checklist pada gejala dan memilih daftar kerusakan lalu menekan button “Tambah Rule”	Data akan ditambahkan pada Tabel Rule Dempster Shafer	Data akan ditambahkan pada Tabel Rule Dempster Shafer	Valid
2	Menekan button “ Tambah rule” dengan memberi checklist pada gejala tanpa memilih kerusakan	Keluar notifikasi anda harus memilih daftar kerusakan	Tetap menambahkan ke tabel yang tidak di definisikan	Tidak Valid
3	Menambahkan nilai yang sama pada data yang telah ada	Keluar notifikasi data telah ditambahkan	Sebaliknya data menumpuk dibawah data sebelum nya	Tidak Valid
4	Menekan button “ Tambah rule” tanpa memberi checklist pada gejala dan memilih kerusakan	Keluar notifikasi “Anda harus memilih minimal 1 gejala”	Keluar notifikasi “Anda harus memilih minimal 1 gejala”	Valid

**4.2.2. Alpha testing**

Pada tahap ini, pengujian alpha akan dilakukan dengan metode *blackbox* dengan memberikan beberapa skenario pengujian serta test case berdasarkan aplikasi yang telah dibuat. Pada pengujian ini tingkat keberhasilan pengujian dihitung menggunakan rumus metode *blackbox* sebagai berikut :

$$\text{Result} = \frac{\text{valid}}{\text{total}} \times 100\%$$

Keterangan :

Result = Persentase keberhasilan pengujian

Valid = Jumlah Pengujian Valid

Total = Total Jumlah Pengujian

(Sumber : Wicaksono, Yuana Chulkamdi, 2021)

Rumus menghitung Test case pada Metode Blackbox [9]

Pada Tahap *Pre Alpha testing* telah dipaparkan testcase untuk pengujian tahap berikutnya. Untuk menghitung tingkat keberhasilan pada pengujian berdasarkan test case yang telah dijabarkan berikut merupakan rumus metode *blackbox* :

- a. Rumus Menghitung Testcase pada Halaman Home

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{3}{3} \times 100\% \\ &= 100\% \end{aligned}$$

- b. Rumus Menghitung Testcase pada Halaman Diagnosa

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{2}{2} \times 100\% \\ &= 100\% \end{aligned}$$

- c. Rumus Menghitung Testcase pada Halaman Daftar Kerusakan

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{2}{2} \times 100\% \\ &= 100\% \end{aligned}$$

- d. Rumus Menghitung Testcase pada Halaman Login

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{4}{4} \times 100\% \\ &= 100\% \end{aligned}$$

- e. Rumus Menghitung Testcase pada Halaman Data Kerusakan dan Solusi

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{4}{4} \times 100\% \\ &= 100\% \end{aligned}$$

- f. Rumus Menghitung Testcase pada Halaman Data Gejala

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{4}{4} \times 100\% \\ &= 100\% \end{aligned}$$

- g. Rumus Menghitung Testcase pada Halaman Rule Dempster Shafer

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{2}{4} \times 100\% \\ &= 50\% \end{aligned}$$

Berdasarkan perhitungan yang dilakukan pada setiap test case yang telah dilakukan dapat disimpulkan hasil pengujian dengan metode *blackbox* sebagai berikut :

$$\begin{aligned} \text{Result} &= \frac{\text{valid}}{\text{total}} \times 100\% \\ &= \frac{21}{23} \times 100\% \\ &= 91,31\% \end{aligned}$$

Rumus Menghitung Testcase pada Keseluruhan Halaman

Dari 23 skenario pengujian yang diujikan pada aplikasi menunjukkan hasil 21 skenario berhasil dan 2 skenario gagal sehingga bisa disimpulkan aplikasi ini berjalan 91,31% sesuai dengan yang diharapkan.

**4.2.3. Post Alpha testing**

*Post-alpha testing* adalah tahap pengujian yang dilakukan setelah selesai tahap *alpha testing*. Pada tahap ini diperlukan pengisian kuesioner untuk menilai kualitas atau keefektifan sebuah website yang telah dibuat. Kuesioner terdiri dari empat pertanyaan yang ditujukan kepada lima responden. Responden akan menjawab empat pertanyaan yang diberikan terkait dengan program yang telah dibuat. Pada kuisisioner ini, skala likert digunakan dari Skala 1 sampai 4. Kriteria skor penilaian yang menggunakan skala likert pada tabel berikut :

Tabel 8. Skala Penilaian [10]

Tingkat Kepuasan	Skala
Sangat Baik	4
Baik	3
Cukup	2
Kurang	1

Sumber : Masripah, Ramayanti (2020)

Tabel 9. Pengujian Kuisisioner 1  
Bagaimana tampilan website aplikasi diagnosis kerusakan mesin *photocopy* ?

Pertanyaan	No	Keterangan	Skala (N)	Responden (R)	N*R
1	1	Sangat Baik	4	1	4
	2	Baik	3	2	6
	3	Cukup	2	1	2
	4	Kurang	1	1	1
Jumlah					13

Hasil pengujian dengan soal mengenai bagaimana tampilan website aplikasi diagnosis kerusakan mesin *photocopy* menunjukkan bahwa 1 dari 5 responden memberi nilai Sangat Baik, 2 dari 5 responden memberikan nilai Baik, 1 dari 5 responden memberi nilai Cukup, 1 dari 5 responden memberi

nilai Sangat Kurang, sehingga diperoleh nilai  $Y = \frac{13}{20} \times 100 = 65\%$ .

Tabel 10. Pengujian Kuisisioner 2  
Apakah aplikasi diagnosis kerusakan mesin *photocopy* mudah digunakan ?

Pertanyaan	No	Keterangan	Skala (N)	Responden (R)	N*R
2	1	Sangat Baik	4	0	0
	2	Baik	3	3	9
	3	Cukup	2	2	4
	4	Kurang	1	0	0
	Jumlah				

Hasil pengujian dengan soal mengenai apakah aplikasi diagnosis kerusakan mesin *photocopy* mudah digunakan menunjukkan bahwa 3 dari 5 responden memberi nilai Baik, 2 dari 5 responden memberikan nilai Cukup, sehingga diperoleh nilai  $Y = \frac{13}{20} \times 100 = 65\%$ .

Tabel 11. Pengujian Kuisisioner 3  
Seberapa akurat hasil diagnosis yang diberikan oleh aplikasi ?

Pertanyaan	No	Keterangan	Skala (N)	Responden (R)	N*R
3	1	Sangat Baik	4	0	0
	2	Baik	3	3	9
	3	Cukup	2	2	4
	4	Kurang	1	0	0
	Jumlah				

Hasil pengujian dengan soal mengenai apakah aplikasi diagnosis kerusakan mesin *photocopy* mudah digunakan menunjukkan bahwa 3 dari 5 responden memberi nilai Baik, 2 dari 5 responden memberikan nilai Cukup, sehingga diperoleh nilai  $Y = \frac{13}{20} \times 100 = 65\%$ .

Tabel 12. Pengujian Kuisisioner 4  
Apakah aplikasi cepat dalam memberikan hasil diagnosa ?

Pertanyaan	No	Keterangan	Skala (N)	Responden (R)	N*R
4	1	Sangat Baik	4	3	12
	2	Baik	3	1	3
	3	Cukup	2	1	2
	4	Kurang	1	0	0
	Jumlah				

Hasil pengujian dengan soal apakah aplikasi cepat dalam memberikan hasil diagnosa menunjukkan bahwa 3 dari 5 responden memberi nilai Sangat Baik, 1 dari 5 responden memberikan nilai Baik, 1 dari 5 responden memberikan nilai Cukup ,sehingga diperoleh nilai  $Y = \frac{17}{20} \times 100 = 85\%$ . Dari hasil pengujian dengan keseluruhan soal menunjukkan bahwa soal pertama mendapat nilai total 65%, soal kedua mendapat nilai total 65%, soal ketiga mendapat nilai total 65%, dan soal keempat mendapat nilai total 65% ,sehingga diperoleh nilai  $Y = \frac{65+65+65+65}{4} = 70$  , sehingga pada tahap *post alpha testing* ini aplikasi ini berjalan 70% dengan yang diharapkan.

Dari kesimpulan keseluruhan hasil pengujian menunjukkan bahwa tahap *alpha testing* mendapat nilai total 91,31%, dan pada tahap *post alpha testing* mendapat nilai total 70%,sehingga diperoleh nilai  $Y = \frac{91,31+70}{2} =$  , sehingga pada keseluruhan tahap pengujian ini aplikasi ini berjalan 80,65% dengan yang diharapkan.

## 5. KESIMPULAN DAN SARAN

Dari hasil pembuatan program sistem pakar diagnosis kerusakan mesin fotokopi dengan Metode Dempster Shafer menunjukkan beberapa kesimpulan penting. Pertama, dalam merancang aplikasi diagnosis, data gejala kerusakan dan jenis kerusakan mesin fotokopi harus dikumpulkan dari observasi di lapangan, dengan hanya mempertimbangkan kerusakan yang sering terjadi. Langkah selanjutnya meliputi pembuatan berbagai diagram dan struktur data seperti Flowchart, ERD, DFD, dan Database, sebelum mengembangkan program sistem pakar.

Kedua, penerapan metode Dempster Shafer dalam aplikasi ini merupakan pilihan yang tepat karena mampu mengatasi ketidakpastian dalam diagnosis kerusakan mesin fotokopi, yang sering melibatkan banyak faktor tidak pasti. Metode ini dapat memodelkan ketidakpastian dengan baik dan mengelola informasi yang tidak lengkap. Ketiga, pengujian aplikasi dilakukan menggunakan metode *alpha testing* yang mencakup uji input-output dan kuesioner responden melalui Google Form. Pengujian ini dilakukan dengan tiga tahap dan menggunakan kombinasi *blackbox testing* dan *closed beta test*.

Dari hasil analisis dan kesimpulan yang tercantum dalam laporan ini, beberapa rekomendasi yang ingin diajukan adalah sebagai berikut. Desain sistem saat ini masih tergolong sederhana, oleh karena itu, diperlukan inovasi baru untuk mengembangkannya ke arah yang lebih canggih. Hal ini akan memungkinkan pengembang untuk melakukan modifikasi yang lebih lanjut agar sistem menjadi lebih baik, dengan menghadirkan fitur baru dan mengurangi jumlah kesalahan yang terjadi. Selain itu, ada kemungkinan bagi pengembang untuk mengubah algoritma yang digunakan dalam aplikasi diagnosis kerusakan mesin *photocopy*, sehingga dapat meningkatkan kinerjanya secara keseluruhan.

## DAFTAR PUSTAKA

- [1] D. Aldo and S. E. Putra, "Sistem Pakar Diagnosis Hama Dan Penyakit Bawang Merah Menggunakan Metode Dempster Shafer," *Komputika J. Sist. Komput.*, vol. 9, no. 2, pp. 85–93, 2020, doi: 10.34010/komputika.v9i2.2884.
- [2] Y. Yuliana, "Sistem Pakar Diagnosis Kerusakan Mesin Fotocopy Minolta Cs Pro Ep5000 Dengan Menggunakan Metode Fordward Chaining," *J. Perangkat Lunak*, vol. 3, no. 2, pp. 51–59, 2021, doi: 10.32520/jupel.v3i2.1622.
- [3] M. Mustaqim and A. Iskandar, "Perbandingan Penggunaan Certainty Factor dan Pendekatan

- Dempster-Shafer dalam Sistem Expert untuk Mendiagnosis Kasus Cacar,” *J. Comput. Syst. Informatics*, vol. 5, no. 1, pp. 93–103, 2023, doi: 10.47065/josyc.v5i1.4618.
- [4] M. Syahputra, “Sistem Pakar Mendiagnosa Penyakit Ensefalitis Menggunakan Metode Dempster Shafer,” *J. SANTI (Sistem Inf. dan Teknol. Informasi)*, vol. 2, no. 1, 2022, doi: 10.35329/jp.v3i1.1232.
- [5] S. I. Pasaribu, “Sistem Pakar Mendeteksi Kerusakan Pada Mesin Printer Dengan Menggunakan Metode Dempster Shafer,” *J. Cyber Tech*, pp. 1–11, 2020.
- [6] K. Kotimah and A. Y. Chandra, “Sistem Pakar Penentuan Jenis Kulit Wajah Menggunakan Metode Dempster Shafer,” *J. Sains dan Teknol.*, vol. 2, no. 1, pp. 18–26, 2022, doi: 10.47233/jsit.v2i1.75.
- [7] A. A. Wahid, “Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,” *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.
- [8] U. S. Senarath, “Waterfall methodology, prototyping and agile development,” *Tech. Rep.*, no. June, pp. 1–16, 2021, doi: 10.13140/RG.2.2.17918.72001.
- [9] E. T. Wicaksono, H. Yuana, and M. T. Chulkamdi, “Perancangan Aplikasi Pemindai Harga Dan Rekomendasi Produk Menggunakan Metode Topsis Berbasis Android,” *J. Mnemon.*, vol. 4, no. 2, pp. 50–56, 2021, doi: 10.36040/mnemonic.v4i2.4169.
- [10] S. Masripah and L. Ramayanti, “Penerapan Pengujian Alpha Dan Beta Pada Aplikasi Penerimaan Siswa Baru,” *Swabumi*, vol. 8, no. 1, pp. 100–105, 2020, doi: 10.31294/swabumi.v8i1.7448.