

PERANCANGAN APLIKASI SOLUSI KERUSAKAN SEPEDA MOTOR KARBURATOR BERBASIS WEB MENGGUNAKAN METODE WATERFALL

Adimas Chandra, Kurnia Paranita Kartika R, Dimas Fanny H.P
Program Studi Teknik Informatika S1, Fakultas Teknologi Informasi
Universitas Islam Balitar Blitar, Jalan Majapahit No.2-4 Blitar, Indonesia
candraadimas@gmail.com

ABSTRAK

Penelitian ini dibangun bertujuan untuk dapat diagnosa kerusakan pada salah satu komponen sepeda motor dan memberikan solusi penanganan perbaikan, dengan menerapkan metode *waterfall* yang terdiri dari 5 tahap yakni *requirement and analysis*, *design*, *coding/implementation*, *testing*, dan *maintenance*. Data yang digunakan untuk membangun sistem ini diperoleh dari jurnal penelitian terdahulu sebagai data sekunder dan data hasil wawancara dari pihak pemilik maupun mekanik instansi bengkel bengkel. Hasil yang diperoleh dari penelitian ini adalah sebuah sistem yang dapat diagnosa kerusakan yang terjadi pada sepeda motor jenis karburator. Sistem ini telah dilakukan pengujian agar mendapatkan kesimpulan yang diinginkan dengan cara menentukan hasil presentase pengujian. Untuk pengujian yang dilakukan menggunakan metode *black box* dan *beta testing* ditambah dengan pengujian validasi ahli IT. Untuk hasil dari pengujian *black box* mendapatkan nilai total rata-rata 93,08%, yang berarti website layak untuk digunakan. Sedangkan untuk pengujian *beta* oleh user dan ahli IT mendapatkan nilai total rata-rata 78% dan 76%, yang berarti website juga sudah layak untuk digunakan. Hasil dari pengujian didapatkan kesimpulan bahwa sistem sudah layak untuk digunakan, akan tetapi masih dibutuhkan pembenahan pada tata letak fitur dan kesalahan sistem.

Kata kunci : Diagnosa kerusakan, *Waterfall*, *black box*, *beta testing*

1. PENDAHULUAN

Seperti halnya pada zaman sekarang teknologi terbilang sudah semakin berkembang, dimana mudahnya untuk mendapatkan kendaraan sepeda motor saat ini, membuat hampir seluruh kalangan masyarakat memiliki sepeda motor. Terlebih lagi dari segi perawatan tentunya juga sangat penting untuk diterapkan. Minimnya pengetahuan akan kerusakan pada sepeda motor mengakibatkan banyaknya masyarakat memilih bengkel menjadi tujuan untuk memperbaiki kerusakan pada sepeda motor [1].

Kondisi mesin pada kendaraan sepeda motor yang terbilang kurang bagus juga dapat mengakibatkan berbagai macam masalah. Untuk itu dapat dinilai dari kurangnya kenyamanan pada saat berkendara, mesin sepeda motor sering bermasalah, atau bahkan dapat mengakibatkan terjadinya kecelakaan [2].

Penelitian ini akan berfokus pada perancangan aplikasi solusi kerusakan sepeda motor karburator. Alasan mengapa peneliti mengambil judul ini dikarenakan banyak pengguna/pengendara sepeda motor pada saat kendarannya mengalami suatu gejala kerusakan, kendaraan tersebut akan langsung dibawa ke bengkel untuk diperbaiki tanpa tahu bahwa gejala kerusakan tersebut dapat di ditangani oleh pengendara itu sendiri. Sehingga peneliti bertujuan untuk membuat sebuah aplikasi yang dapat memberikan solusi kerusakan pada salah satu komponen sepeda motor sehingga pengendara dapat dengan sendirinya memperbaiki kerusakan tersebut tanpa harus di bawa ke bengkel.

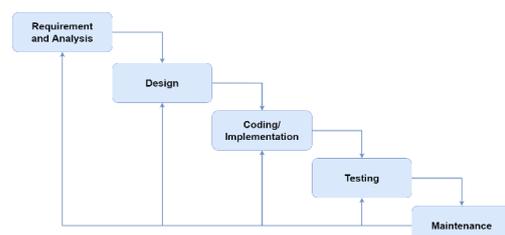
hasil penelitian ini diharapkan dapat membantu pengendara sepeda motor dalam mengenali kerusakan dan juga memberikan solusi penanganan perbaikan dari kerusakan salah satu komponen sepeda motor melalui media online berbasis web. Khususnya sepeda motor karburator (manual).

2. TINJAUAN PUSTAKA

2.1. *Waterfall*

Pembangunan sistem secara keseluruhan dilakukan melalui beberapa tahapan/langkah. Metode *waterfall* merupakan metode pengembangan perangkat lunak tertua sebab sifatnya yang natural. Urutan dalam metode *waterfall* bersifat serial yang dimulai dari proses menganalisa kebutuhan hingga perawatan pada sistem [3].

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap *requirement and analysis*, *design*, *coding/implementation*, *testing*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut *waterfall* [4].



Gambar 1. Alur *waterfall*

Berikut merupakan tahapan dari metode *waterfall*, yaitu:

a. *Requirement and analysis*

Pada tahap ini peneliti harus mengetahui seluruh informasi mengenai kebutuhan *software* seperti kegunaan *software* yang diinginkan oleh pengguna dan batasan *software*. Informasi tersebut biasanya diperoleh dari wawancara, *survey*, ataupun diskusi. Setelah itu informasi dianalisa sehingga mendapatkan data-data yang lengkap mengenai kebutuhan pengguna akan *software* yang akan dikembangkan.

b. *Design*

Tahap selanjutnya yaitu Desain. Desain dilakukan sebelum proses *coding* dimulai. Ini bertujuan untuk memberikan gambaran lengkap tentang apa yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem yang diinginkan. Sehingga membantu menspesifikasikan kebutuhan *hardware* dan sistem, juga mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan.

c. *Coding/implementation*

Proses penulisan code ada di tahap ini. Pembuatan *software* akan dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum. Dan dilanjutkan hingga ke Penerapan program, penerapan program di sini *software* yang sudah jadi akan dijalankan atau dioperasikan oleh penggunanya.

d. *Testing*

Pada tahap keempat ini akan dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya. Setelah itu akan dilakukan pengujian yang bertujuan untuk mengetahui apakah *software* sudah sesuai desain yang diinginkan dan apakah masih ada kesalahan atau tidak.

e. *Maintenance*

Pada tahap kelima ini selanjutnya *software* akan dilakukan perawatan yang melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

2.2. *Flowchart*

Flowchart adalah mempunyai tampilan bentuk secara simbolik dari suatu algoritma atau tata cara untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan pada bagian-bagian yang terlupakan dalam menganalisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek [5].

2.3. *Data Flow Diagram*

Data Flow Diagram (DFD) adalah representasi grafis dari aliran data melalui sistem informasi. DFD memungkinkan untuk memvisualisasikan bagaimana sistem beroperasi, apa sistem menyelesaikan dan bagaimana itu akan dilaksanakan, bila disempurnakan dengan spesifikasi lebih lanjut. *Data flow diagram* digunakan oleh analis sistem untuk merancang sistem pemrosesan informasi tetapi juga sebagai cara untuk model seluruh organisasi [6].

2.4. *Usability Testing*

Usability testing merupakan bentuk metode dalam evaluasi usability yang digunakan untuk mengevaluasi sebuah produk dengan mengujinya langsung pada pengguna sebagai user. Sehingga bertujuan untuk mengidentifikasi masalah uji ketergunaan seperti, mengumpulkan data kualitatif maupun kuantitatif, mengukur kemudahan, mengukur efisiensi dan menentukan kepuasan pengguna dengan produk [7].

3. METODE PENELITIAN

Suatu penelitian tentunya dilakukan dengan menggunakan suatu metode atau langkah-langkah yang sesuai dengan tema yang dipakai, hal ini bertujuan untuk melakukan pembatasan yang jelas dan tegas untuk menentukan objek penelitian, teknik pengambilan data, maupun analisis data. Untuk itu pada penelitian ini menggunakan pengembangan *research and development* (RnD).

Pengembangan *research and development* (RnD) adalah salah satu metode penelitian yang digunakan untuk menghasilkan suatu produk tertentu dan menguji keefektifannya [8]. Selanjutnya untuk model yang digunakan adalah metode *waterfall* yang digunakan untuk menghasilkan sebuah produk berupa sistem aplikasi. Berikut merupakan langkah-langkah pada metode *waterfall* yang digunakan:

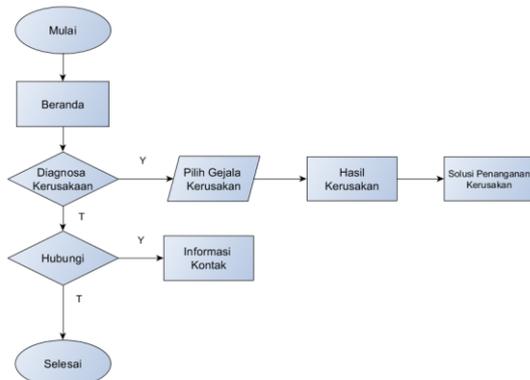
3.1. *Requirement and Analysis*

Tahap pertama yang dilakukan adalah analisis kebutuhan, pada tahap ini yang dilakukan adalah pengumpulan data sehingga data dapat digunakan untuk membangun sistem. Berikut ini merupakan data gejala kerusakan dan data kerusakan yang dialami kendaraan sepeda motor yang telah didapatkan berdasarkan wawancara pada pihak bengkel dan kepustakaan pada Tabel 1:

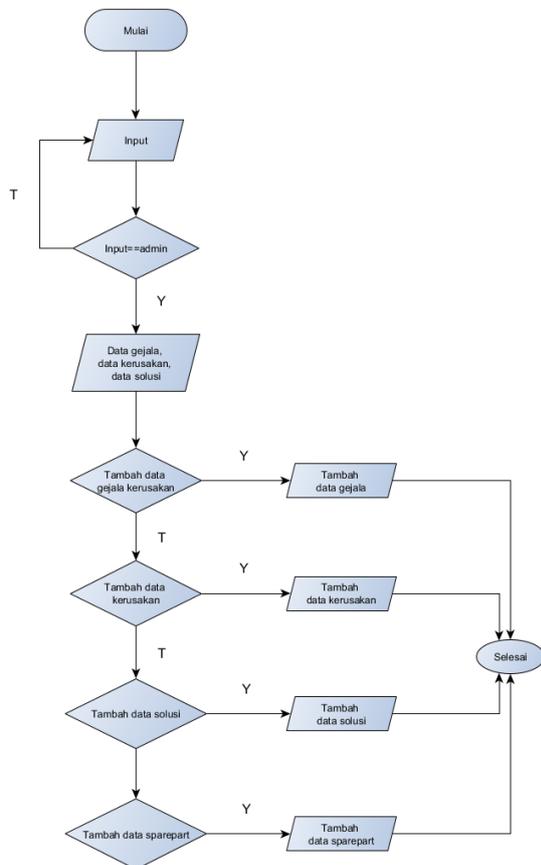
Tabel 1. Tabel gejala dan kerusakan sepeda motor

No	Nama Kerusakan	Gejala Kerusakan
1	Busi	<ul style="list-style-type: none"> • Mesin susah dihidupkan • Tidak ada percikan api • Mesin meledak-ledak ketika berjalan
2	Koil	<ul style="list-style-type: none"> • Mesin susah dihidupkan • Percikan api pada busi berwarna kemerahan • Mesin tiba-tiba mati ketika sedang berjalan

No	Nama Kerusakan	Gejala Kerusakan
3	CDI	<ul style="list-style-type: none"> Mesin susah dihidupkan Mesin brebet di rpm tinggi Percikan busi pendek Busi sering mati
4	Spul Pengisian	<ul style="list-style-type: none"> Mesin susah dihidupkan Lampu spidometer, sen, dll mati Aki kurang daya
5	Kiprok	<ul style="list-style-type: none"> Bohlam lampu sering putus/mati Aki cepat soak Mesin tiba-tiba mati saat dihidupkan
6	Aki	<ul style="list-style-type: none"> Lampu spidometer, sen, dll mati Starter tidak berfungsi Lampu signal redup Klakson tidak bersuara



Gambar 2. Flowchart Aplikasi Untuk Pengguna



Gambar 3. Flowchart Untuk Admin

3.2. Flowchart Aplikasi Untuk Pengguna

Bedasarkan hasil dari analisa dibutuhkan perancangan pada aplikasi solusi kerusakan sepeda motor karburator (manual). Aplikasi ini memiliki dua alur sistem, yaitu alur dari pengguna dan alur dari admin. berikut ini merupakan flowchart alur pengguna. Untuk penjelasan lebih rincinya, flowchart untuk pengguna digambarkan pada Gambar 2.

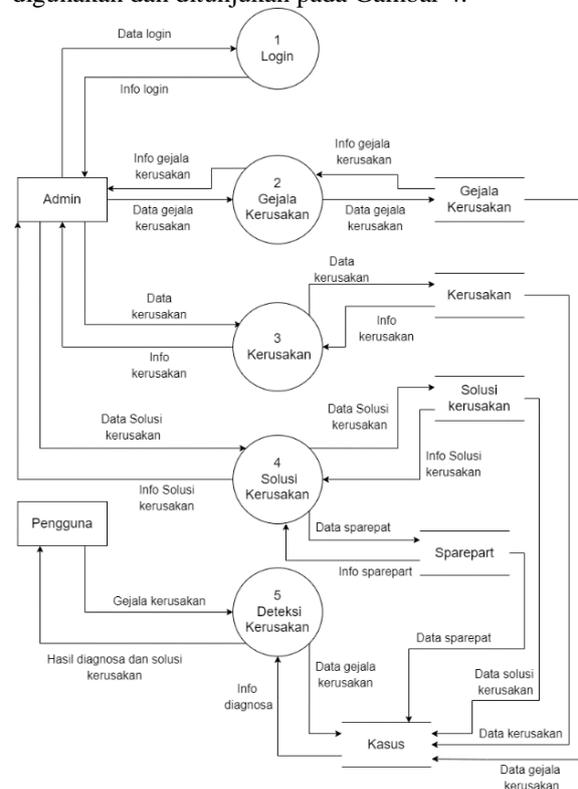
3.3. Flowchart Untuk Admin

Flowchart untuk admin sendiri berperan untuk menambah data gejala kerusakan, kerusakan, solusi kerusakan, dan sparepart. Berikut ini flowchart alur admin. Untuk penjelasan lebih rincinya, flowchart untuk admin digambarkan pada Gambar 3.

3.4. Merancang Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafis dari aliran data melalui sistem informasi. Data flow diagram digunakan oleh analis sistem untuk merancang sistem pemrosesan informasi tetapi juga sebagai cara untuk model seluruh organisasi.

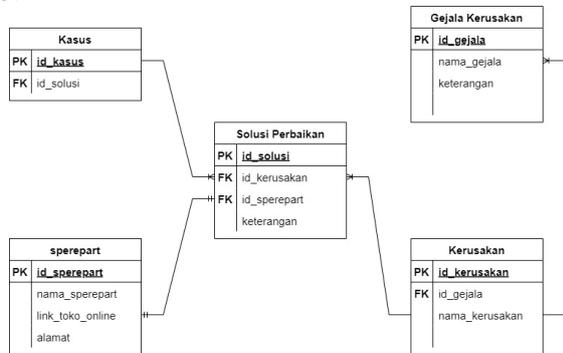
Proses pertama adalah login, merupakan proses untuk masuk kedalam sistem. Selanjutnya Proses kedua gejala kerusakan, merupakan proses pengolahan data gejala kerusakan. Proses ketiga kerusakan merupakan proses pengolahan data kerusakan. Proses keempat solusi kerusakan, merupakan proses pengolahan data solusi kerusakan. Dan yang terakhir proses kelima deteksi kerusakan merupakan proses untuk mendeteksi kerusakan yang terjadi pada sepeda motor. Berikut keterangan untuk DFD sistem yang digunakan dan ditunjukkan pada Gambar 4.



Gambar 4. Data Flow Diagram

3.5. Merancang Entity Relationship Diagram (ERD)

Untuk memulai perancangan sistem dibutuhkan konsep sederhana untuk membuat database. ERD merupakan bentuk gambaran diagram untuk diterapkan pada konsep pembuatan database yang akan dirancang. Berikut ERD ditunjukkan pada Gambar 5.



Gambar 5. Entity Relationship Diagram

4. HASIL DAN PEMBAHASAN

4.1. Implementation

Tahap lanjut dalam membangun waterfall merupakan tahapan implementasi. Pada tahap ini dilakukan pengkodean sistem yang digunakan untuk membuat tampilan layout pada sistem. Setelah melakukan pengkodean sistem selanjutnya dilakukan pengujian untuk menilai kelayakan sistem yang dibuat. Untuk tahap akhir pada waterfall adalah dilakukan pemeliharaan pada sistem.

Pada halaman sistem untuk pengguna sebagai user terdapat beberapa halaman, diantaranya: halaman utama, halaman deteksi kerusakan, halaman solusi kerusakan, dan halaman hubungi. Berikut ini hasil implementasi berbasis website solusi kerusakan sepeda motor karburator:

4.2. Halaman Beranda

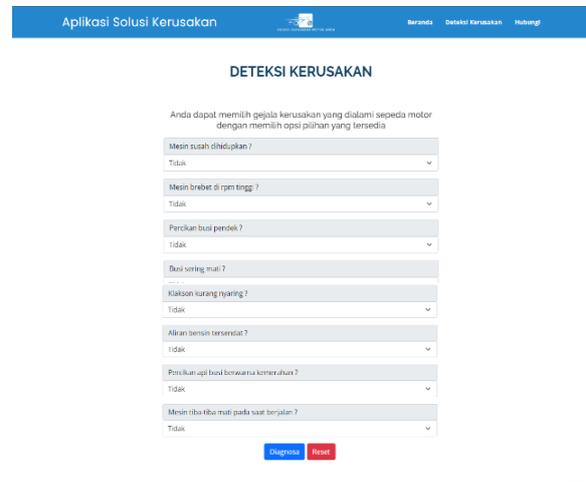
Halaman ini merupakan bentuk tampilan beranda pada website, halaman beranda ini menyediakan informasi mengenai kegunaan dan keunggulan yang di terapkan pada website. Dapat dilihat pada Gambar 6.



Gambar 6. Tampilan halaman utama

4.3. Halaman Deteksi Kerusakan

Pada halaman deteksi kerusakan menyediakan form pilihan untuk mengidentifikasi gejala kerusakan yang dialami pada sepeda motor. Dapat dilihat pada Gambar 7.



Gambar 7. Tampilan halaman deteksi kerusakan

4.4. Halaman Solusi Kerusakan

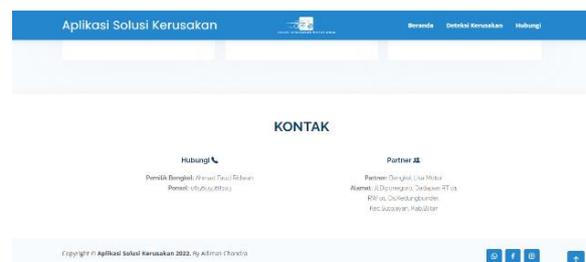
Halaman solusi kerusakan adalah halaman yang menampilkan informasi solusi penanganan perbaikan pada salah satu kerusakan, dari hasil mengidentifikasi gejala yang dialami sepeda motor. Dapat dilihat pada Gambar 8.



Gambar 8. Tampilan halaman solusi kerusakan

4.5. Halaman Hubungi

Pada halaman hubungi menyediakan informasi tentang kontak dari pemilik bengkel dan alamat bengkel. Dapat dilihat pada Gambar 9.



Gambar 9. Tampilan halaman kontak

4.6. Testing

Pada tahap ini merupakan pengujian pada sistem, pengujian ini dilakukan untuk menilai layaknya sebuah sistem agar dapat diterapkan secara *public*. Pengujian black box dikelompokkan berdasarkan fungsionalitas dan halamannya, yang dimana komponen yang akan diujikan berupa: pengujian tombol menu / navigasi, pengujian halaman utama, pengujian halaman login, pengujian sisi admin dan pengujian halaman diagnosa serta halaman hasil diagnosa.

Tabel 2. Pengujian *black box* fungsional fitur dan input data pada aplikasi

Pengujian Sistem aplikasi	Skenario Pengujian Sistem Aplikasi	Hasil Yang Diharapkan	Kesimpulan
Menu navigasi deteksi kerusakan	Pengguna mencoba menekan menu navigasi deteksi kerusakan	Sistem akan mengarahkan pengguna ke halaman seteksi kerusakan	Sesuai
Tombol diagnosa	Pengguna mencoba menekan tombol diagnosa	Pengguna akan diarahkan ke halaman solusi kerusakan setelah pengguna melakukan pemilihan gejala kerusakan	Sesuai
Input data	Admin mencoba menambah data jenis kerusakan	Kolom pada aplikasi dapat di isi	Sesuai
Input data	Admin mencoba menambah data kerusakan yang sama dengan kerusakan yang sudah ada	Sistem akan otomatis mendeteksi kerusakan yang sudah ada dengan memunculkan notif	Tidak Sesuai

Total seluruh pengujian ada 49 pengujian yang mana dari seluruh pengujian tersebut terdapat tiga pengujian yang tidak sesuai. Jadi dapat disimpulkan dari pengujian tersebut diperoleh:

$$\text{Hasil pengujian } black\ box = \frac{\text{pengujian berhasil}}{\text{total pengujian}} \times 100 = \frac{46}{49} \times 100\% = 93,08\%$$

Dapat disimpulkan bahwa pada pengujian blackbox mendapatkan hasil 93,08% dari total 49 skenario pengujian. Baik pengujian yang dilakukan dari sisi pengguna maupun admin.

Tahap selanjutnya pengujian ini dilakukan dengan mengisi *form kusioner* penilaian yang telah disediakan oleh peneliti, dengan cara menjawab pertanyaan yang tertera pada form kusioner. Untuk pengujian validasi ahli digunakan metode pengujian *usability testing*.

Tabel 3. Aspek *usability testing*

No	Indikator	Butir Pertanyaan
1	<i>Learnability</i>	5
2	<i>Memorability</i>	2
3	<i>Efficiency</i>	2
4	<i>Errors</i>	2
5	<i>Satisfaction</i>	4

Sumber : [9]

Bedasarkan hasil kuisisioner, akan ditentukan nilai presentase setiap jawaban dengan menggunakan rumus dibawah ini.

$$Y = \frac{\Sigma(N.R)}{\text{Skor Ideal}} \times 100\%$$

Keterangan:

Y = nilai presentase yang dicari

X = jumlah kategori jawaban yang dikalikan dengan frekuensi ($\Sigma = N.R$).

N = nilai dari setiap jawaban.

R = Frekuensi.

Skor ideal = nilai tertinggi dikalikan dengan jumlah sampel.

Untuk hasil perhitungan dinilai dari keriteria kelayakan, dapat dilihat pada Tabel 3 berikut : [10]

Tabel 4. Tabel kriteria kelayakan

Persentase Nilai Rata-Rata	Kategori	Keterangan
85%—100%	Sangat Layak	Sangat baik untuk digunakan
69%—84%	Layak	Boleh digunakan dengan revisi kecil
53%—68%	Cukup Layak	Boleh digunakan setelah revisi besar
37%—52%	Kurang Layak	Tidak boleh digunakan
20%—36%	Tidak Layak	Sangat Tidak boleh digunakan

4.7. Hasil Perhitungan

Hasil penyebaran form kusioner didapatkan 3 responden yang telah mengisi dan menilai website. Hasil pengisian ini kemudian dilakukan pengujian validasi ahli IT menggunakan *usability testing*. Berdasarkan uji validasi ahli IT menggunakan penilaian form kusioner menghasilkan jumlah responden sebanyak 3 orang, dengan kategori penilaian dilakukan pada orang yang sudah ahli pada bidang IT. Maka didapatkan jumlah Y=nilai presentase yang dicari 76%. dapat dilihat pada tabel 5.

Tabel 5. Hasil pengujian validasi ahli IT

No	Indikator	Kode Pertanyaan	Indikator Pertanyaan Untuk Aplikasi	$\Sigma N.R$	Y
1	<i>Learnability</i>	A1	Fitur dan isi konten mudah untuk dipelajari	10	83%
		A2	Penempatan informasi tidak membingungkan	9	75%
		A3	Penyajian konten mudah untuk dipahami	11	91%
		A4	Alur navigasi mudah untuk dioperasikan	9	75%
		A5	Tidak membutuhkan instruksi khusus untuk menjalankan aplikasi	11	91%
2	<i>Memorability</i>	B1	Website mudah untuk diingat	9	75%
		B2	Alur navigasi mudah diingat	10	83%
3	<i>Efficiency</i>	C1	Pada website kecepatan dalam mengakses menu, dan fitur mudah digunakan	9	75%
		C2	Pada website kecepatan mendapatkan informasi sangat mudah	9	75%

No	Indikator	Kode Pertanyaan	Indikator Pertanyaan Untuk Aplikasi	ΣN.R	Y
4	Errors	D1	Tidak menemukan kesalahan pada website	7	58%
		D2	Keaktifan fungsional menu pada website tidak mengalamai kesalahan	8	66%
5	Satisfaction	E1	website memiliki kenyamanan tampilan pada semua halaman website	9	75%
		E2	Website memiliki kenyamanan dalam penjelajahan aplikasi lebih detail	8	66%
		E3	Keluwesan tata letak komponen dan konten pada website sudah sesuai	8	66%
		E4	Website memiliki kesesuaian dengan judul penelitian yang telah dipilih dan konten yang disajikan	11	91%
			Total	76%	

$$\frac{\text{Rata - Rata Nilai Y}}{\text{Jumlah pertanyaan}} \times 100\% = \frac{1.145}{15} \times 100\% = 76\%$$

Dengan melihat hasil uji validasi ini Y=nilai presentase yang dicari mempunyai nilai rata-rata total sebesar 76%, selanjutnya nilai tersebut akan dicocokkan pada tabel kriteria sehingga mendapatkan kategori dinyatakan layak dengan keterangan website sudah dapat digunakan.

4.8. Maintenance

Pada tahap ini aplikasi terbilang sudah siap untuk digunakan atau diujicobakan. Namun, pada penelitian ini proses pemeliharaan tidak dilakukan, dikarenakan aplikasi belum diterapkan secara nyata kepada user, serta masih dibutuhkan pengujian lebih lanjut.

4.9. Pembahasan

Bedasarkan hasil pengujian yang sudah didapatkan, penelitian ini masih perlu dikembangkan lebih lanjut dan dilakukan validasi ulang terhadap data pada sistem yang digunakan. Dikarenakan terdapat hasil yang terbilang kurang baik sehingga mendapatkan penilaian rata-rata 58% pada salah satu pertanyaan dinilai langsung oleh ahli IT itu sendiri.

5. KESIMPULAN DAN SARAN

Hasil dari penelitian yang dilakukan dengan metode waterfall dapat ditarik kesimpulan bahwa sistem yang telah dibuat sangat membantu dalam mendeteksi kerusakan pada salah satu komponen sepeda motor. Akan tetapi penelitian ini masih perlu dikembangkan ulang lebih lanjut dari kelengkapan data dan kelayakan sistem.

Adapun saran yang diberikan kepada peneliti, dan dari peneliti saran tersebut dapat digunakan pengembangan penelitian selanjutnya. sarannya sebagai berikut: Masih terdapat beberapa fungsi yang belum memuaskan, khususnya pada fitur inti. Yang dimana dapat lebih ditingkatkan lagi keakuratan pada solusi yg ditawarkan dari gejala-gejala yang akan dipilih, Untuk dari segi list pertanyaan diharapkan

dapat dibagi tiap komponen secara umum, misalnya pada jenis gejala kerusakan diharapkan dapat dibedakan dari segi bagian kelistrikan maupun mesin.

DAFTAR PUSTAKA

- [1] A. Syaputra dan D. Setiadi, "Sistem Pakar Diagnosa Kerusakan Sepeda Motor Yamaha Matic Menggunakan Metode Forward Chaining," *Jusikom J. Sist. Komput. Musirawas*, vol. 5, no. 2, hal. 126–135, 2020, doi: 10.32767/jusikom.v5i2.1039.
- [2] A. Mugiprakoso dan N. Hidayat, "Identifikasi Kerusakan Mesin Pada Sepeda Motor Menggunakan Metode Modified K-Nearest Neighbor (MKNN)," vol. 3, no. 4, 2019.
- [3] G. Wiro Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *J. Inform. Pengemb. IT*, vol. 2, no. 1, hal. 6–12, 2017.
- [4] I. Widiaty, L. S. Riza, Ana, A. G. Abdullah, M. Abdullah, dan S. R. Mubaroq, "Web-based digital learning application of iconic batik in batik learning at vocational high school," *J. Eng. Sci. Technol.*, vol. 14, no. 5, hal. 2475–2484, 2019.
- [5] S. Santoso dan R. Nurmalina, "Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas," *J. Integr.*, vol. 9, no. 1, hal. 84, 2017, doi: 10.30871/ji.v9i1.288.
- [6] F. Soufitri, "Perancangan Data Flow Diagram Untuk Sistem Informasi Sekolah (Studi Kasus Pada Smp Plus Terpadu)," *Ready Star*, vol. 2, no. 1, hal. 240–246, 2019.
- [7] N. Luh Putri Ari Wedayanti, N. Kadek Ayu Wirdiani, dan I. Ketut Adi Purnawan, "Evaluasi Aspek Usability pada Aplikasi Simalu Menggunakan Metode Usability Testing," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 7, no. 2, hal. 113, 2019, doi: 10.24843/jim.2019.v07.i02.p03.
- [8] S. Purnama, "Metode Penelitian Dan Pengembangan (Pengenalan Untuk Mengembangkan Produk Pembelajaran Bahasa Arab)," *LITERASI (Jurnal Ilmu Pendidikan)*, vol. 4, no. 1, hal. 19, 2016, doi: 10.21927/literasi.2013.4(1).19-32.
- [9] A. Setiawan dan R. A. Widyanto, "Evaluasi Website Perguruan Tinggi menggunakan Metode Usability Testing," *J. Inform. J. Pengemb. IT*, vol. 3, no. 3, hal. 295–299, 2018, doi: 10.30591/jpit.v3i3.912.
- [10] Vidia Sari H dan Suswanto H, "Pengembangan Media Pembelajaran Berbasis Web untuk Mengukur Hasil Belajar Siswa Pada Mata Pelajaran Komputer Jaringdasar Program Keahlian Teknik Komputer Dan Jaringan," *J. Pendidik. Teor. Penelitian, dan Pengemb.*, vol. 2, hal. 1008–1016, 2017.