

Perancangan Aplikasi Pengecekan Kerusakan *Capacitor Discharge Ignition (CDI)* Sepeda Motor Berbasis IOT

Muhammad Hidayat¹, Nonong Rahimah²

^{1,2} STKIP PGRI Banjarmasin

e-mail: ¹ hidayat90@stkipbjm.ac.id, ² nonongrahimah6@gmail.com

Abstrak

Penerapan *Internet of Things* (IoT) dalam pengujian dan perbaikan kerusakan pada *Capacitor Discharge Ignition* (CDI) pada sepeda motor matic Honda digunakan untuk memeriksa dan membaca data dari sensor-sensor pada CDI sepeda motor. Namun, keterbatasan dalam mengambil data pada CDI standar menyebabkan informasi yang diperoleh tidak terperinci. Oleh karena itu, penulis mencoba membuat alat diagnosa CDI sendiri berbasis IoT dengan modul Arduino Mega 2560 serta modul sim 808 untuk memudahkan mekanik dalam mendiagnosis dan memperbaiki kerusakan CDI pada sepeda motor injeksi matic, dalam pengujian kali ini menggunakan tipe *scooter* matic Honda Scoopy Fi tahun 2016. Aplikasi pengecekan kerusakan CDI ini diharapkan dapat meningkatkan kualitas layanan, menghemat waktu dan biaya perbaikan, serta memperpanjang umur pemakaian CDI. Kesimpulan perancangan Aplikasi kerusakan CDI menghasilkan 7 *form user interface* dan 5 Tabel dalam rancangan *database* sedangkan dari pengujian dengan metode McCall menghasilkan presentase sebesar 84,6% (sangat baik), sedangkan dalam analisis perangkat keras dan perangkat lunak menunjukkan bahwa semua perangkat berjalan dengan baik, tetapi terkadang Arduino Mega mengalami lost respon dalam koneksi ke sensor, hal ini menjadi fokus pengembangan Aplikasi kerusakan CDI selanjutnya untuk bisa menjadi lebih baik lagi.

Kata Kunci : IoT, CDI, Arduino Mega, Perancangan Aplikasi

Abstract

The application of the Internet of Things (IoT) in testing and repairing damage to Capacitor Discharge Ignition (CDI) on Honda automatic motorcycles is used to check and read data from sensors on motorcycle CDIs. However, the limitations in retrieving data on a standard CDI cause the information obtained is not detailed. Therefore, the author tries to make an IoT-based CDI self-diagnosis tool with an Arduino Mega 2560 module and an 808 sim module to make it easier for mechanics to diagnose and repair CDI damage on matic injection motorbikes, in this test using the 2016 Honda Scoopy Fi matic scooter type. This CDI damage checking application is expected to improve service quality, save time and repair costs, and extend the service life of CDI. The conclusion of the design of the CDI damage application produces 7 user interface forms and 6 tables in the database design while from testing with the McCall method results in a percentage of 84.6% (very good), while in the analysis of hardware and software shows that all devices are running well, but sometimes Arduino Mega experiences a lost response in connection to the sensor, this is the focus of further development of the CDI damage application to be even better.

Keywords: IoT, CDI, Arduino Mega, Application Design.

1. Pendahuluan

Dengan perkembangan teknologi informasi dan komunikasi saat ini, terutama teknologi Internet-of-Things (IoT) telah membawa perubahan signifikan dalam berbagai sektor kehidupan (Prakoso, 2022).

Salah satu sektor yang tak luput dari dampak positif IoT adalah industri sepeda motor. Dalam upaya memperbaiki dan meningkatkan performa sepeda motor, penting bagi para penggemar dan mekanik sepeda motor untuk dapat mengidentifikasi

dan memperbaiki kerusakan dengan cepat dan efisien.

Sepeda motor merupakan salah satu alat transportasi utama bagi sebagian orang dalam menjalankan aktivitas sehari-hari (L.A. Dewi, 2022). Sepeda motor merupakan alat transportasi yang digerakkan oleh mesin (Ilyas, 2022). Salah satu komponen vital dalam sistem pengapian sepeda motor adalah *Capacitor Discharge Ignition* (CDI). CDI bertanggung jawab mengatur aliran listrik yang diperlukan untuk memicu ledakan bahan bakar dalam ruang bakar mesin. Namun, kerusakan pada CDI sering kali sulit untuk dideteksi secara tepat dan memerlukan keahlian khusus dari mekanik. 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 (A. Mugiprakoso, 2019).

Tingkat kesulitan yang dialami mekanik dalam mengetahui atau menentukan jenis kerusakan pada sepeda motor injeksi (Sitio, 2021) bisa saja bersumber pada kerusakan CDI, terkadang luput dari perhatian mekanik. Dengan adanya aplikasi pengujian dan pemeriksaan kerusakan CDI sepeda motor berbasis IoT, diharapkan akan lebih mudah dan efisien bagi mekanik sepeda motor untuk mendiagnosis dan memperbaiki kerusakan pada CDI. Hal ini akan membantu meningkatkan kualitas layanan, menghemat waktu dan biaya perbaikan, serta meningkatkan kepuasan pelanggan. Selain itu, aplikasi ini juga dapat membantu mengurangi potensi kerusakan yang lebih parah dan memperpanjang umur pemakaian CDI pada sepeda motor. Dengan pemaparan yang telah di jelaskan, penulis memandang penting permasalahan yang ada saat ini untuk memudahkan teknisi dan meningkatkan pelayanan terhadap konsumen, terkendala kode etik dan hak cipta suatu produk untuk mencoba memodifikasi alat dari pihak yamaha, dengan ini penulis mencoba membuat alat diagnosa CDI sendiri yang dapat menyimpan suatu kejadian khususnya pada bagian sistem elektronik injeksi berdasarkan pada data sensor-sensor dan waktu kejadian, dengan menggunakan teknologi IoT (*Internet Of Things*)

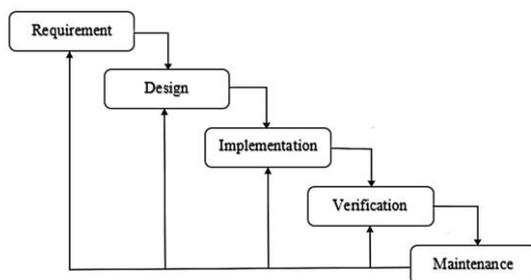
memanfaatkan *web server* sebagai penyimpanan data sehingga dapat di akses kapanpun dan dimanapun selama terkoneksi dengan internet.

2. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode deskriptif. Data yang diperoleh akan dideskripsikan secara kualitatif dan kuantitatif. Penelitian deskriptif bertujuan untuk membuat deskripsi secara sistematis, faktual dan akurat mengenai fakta-fakta dan sifat populasi di wilayah tertentu. Data kualitatif yang dihasilkan akan dapat memberikan jawaban terhadap penelitian yang dilakukan. Penelitian ini dilakukan pada model waterfall sebagai metodologi pengembangan Aplikasi Kerusakan CDI pada sepeda motor injeksi. Langkah awal pengumpulan data menggunakan studi pustaka. Pada metode pengembangan sistem penulis menggunakan metode SDLC (*System Development Life Cycle*), SDLC terdiri dari sejumlah tahapan yang dilaksanakan secara berurutan (Rozaq, 2018), selanjutnya untuk model yang digunakan adalah metode *waterfall* yang digunakan untuk menghasilkan sebuah produk berupa sistem aplikasi (C. Adimas, 2022).

Software Development Life Cycle (SDLC) adalah sebuah proses yang menggambarkan metode dan strategi seperti bagaimana caranya mengembangkan desain dan memelihara proyek perangkat lunak serta memastikan bahwa semua tujuan, sasaran, fungsional, dan kebutuhan pengguna terpenuhi (N.Arora, 2016).

Model *Waterfall* (model air terjun) merupakan suatu model pengembangan secara sekuensial. Model *Waterfall* bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan (Pressman, 2012). Tahapan dari metode *waterfall* tersebut dapat dilihat pada Gambar 1, sedangkan penerapan model *waterfall* dapat dijelaskan melalui alur penelitian yang dapat dilihat pada Gambar 2.



Gambar 1. Metode *Waterfall*
Sumber : Pressman (2012)

Tahapan Metode *Waterfall*.

a. *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

b. *Design*.

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

c. *Implementation*.

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

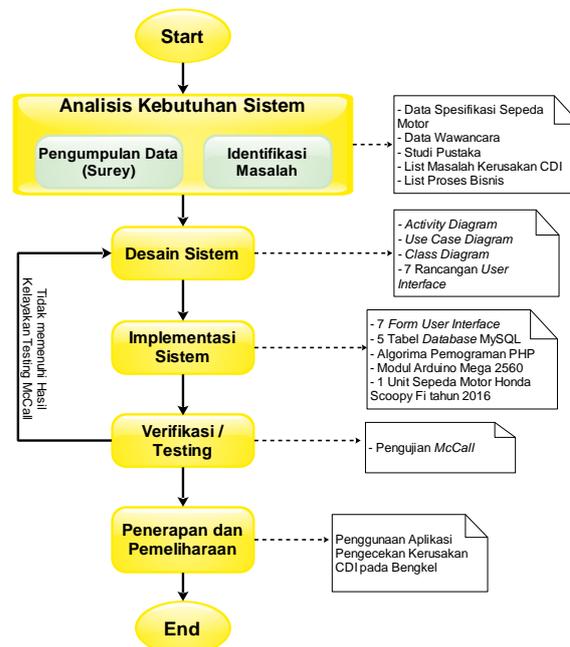
d. *Verification*.

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

e. *Maintenance*.

Ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk

dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

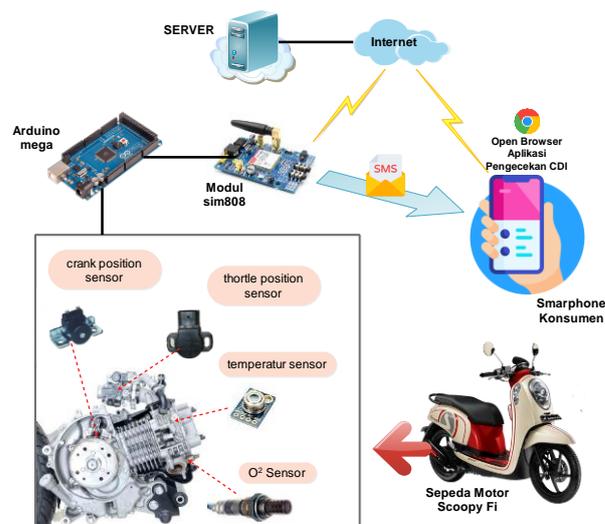


Gambar 2. Alur Penelitian

3. Hasil dan Pembahasan

3.1 Arsitektur Sistem

Analisis arsitektur sistem merupakan suatu proses yang bertujuan untuk menjelaskan secara rinci tentang struktur fisik dari sistem yang akan dikembangkan, termasuk komponen-komponen utama dan pendukungnya, yang dapat dilihat pada Gambar 3 berikut ini.



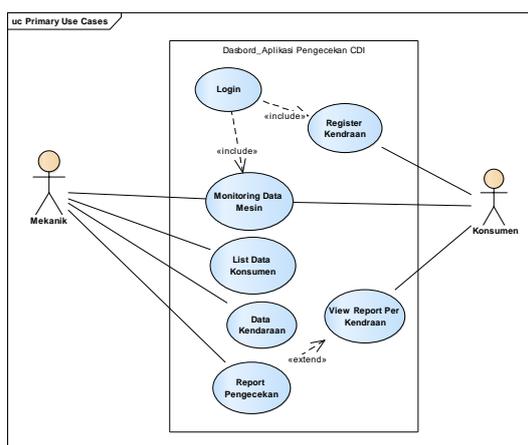
Gambar 3. *Computing Network Diagram* Aplikasi Pengecekan CDI

3.2 Rancangan Diagram Aktivitas

Diagram aktivitas ini memberikan menggambarkan langkah-langkah dalam proses perancangan Aplikasi Kerusakan CDI dengan tools *Unified Modeling Language* (UML), UML adalah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain,sertamenggambarkan arsitektur dalam pemrograman berorientasi objek (R.A. Sukamto, 2018). Komponen UML yang digunakan diantaranya yaitu : *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

a. Use Case Dashboard Aplikasi Kerusakan CDI.

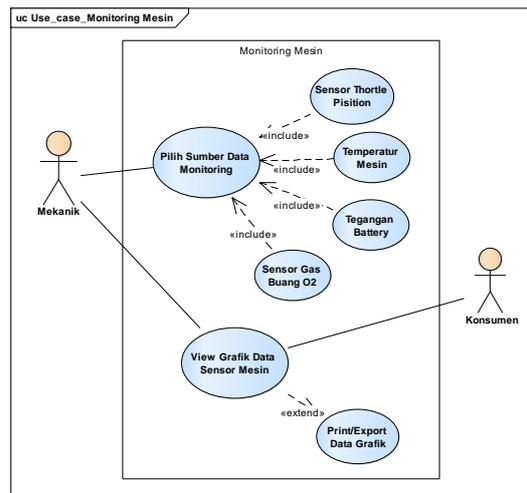
Use case ini menjelaskan tentang antarmuka yang ditujukan untuk Teknisi atau Mekanik dan konsumen yang menggunakan aplikasi diagnose data mesin dan melihat hasil report yang dijelaskan pada Gambar 4.



Gambar 4. Use Case Dashboard Aplikasi Kerusakan CDI

b. Use Case Monitoring Mesin.

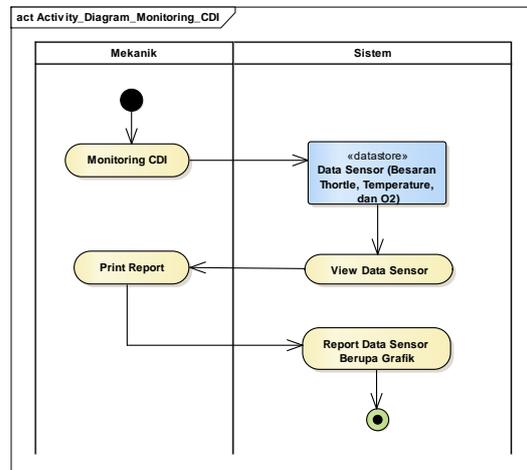
Use case ini menjelaskan tentang antarmuka yang ditujukan untuk Teknisi atau Mekanik dan konsumen yang menggunakan aplikasi untuk monitoring mesin dan melihat grafik dari perolehan sensor pada mesin yang dijelaskan pada Gambar 5.



Gambar 5. Use Case Monitoring Mesin

c. Activity diagram Monitoring CDI.

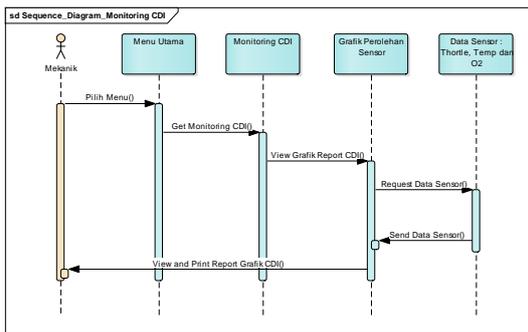
Activity diagram monitoring CDI digunakan untuk menggambarkan proses pemantauan kerusakan pada sistem CDI pada sepeda motor. Diagram ini memperlihatkan langkah-langkah yang terlibat dalam pemantauan CDI dan bagaimana informasi kerusakan diperoleh yang dapat dilihat pada Gambar 6.



Gambar 6. Activity diagram Monitoring CDI

d. Sequence Diagram Monitoring CDI.

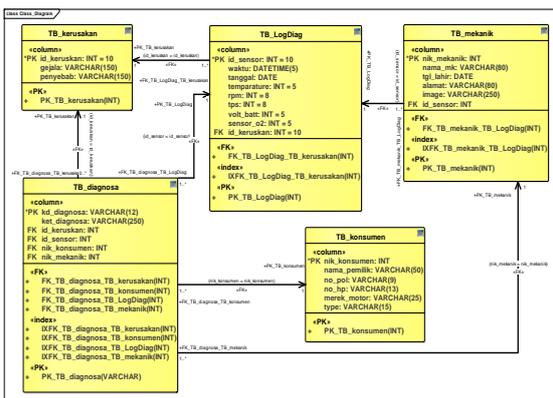
Sequence diagram monitoring CDI digunakan untuk menggambarkan interaksi antara objek-objek terkait dalam proses pemantauan sistem CDI pada sepeda motor. Diagram ini menunjukkan urutan pesan antara objek-objek yang terlibat dalam pemantauan CDI yang dapat dilihat pada Gambar 7.



Gambar 7. Sequence diagram Monitoring CDI

e. Class Diagram Aplikasi Kerusakan CDI.

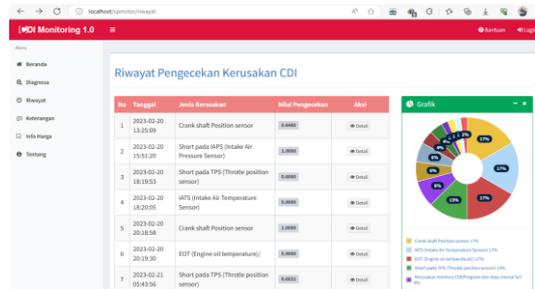
Diagram ini memberikan gambaran tentang struktur kelas dan interaksi antar kelas dalam Aplikasi. Dalam proses perancangan database, dibutuhkan tabel-tabel yang normal, yang tidak memiliki anomali pada setiap tabelnya. Normalisasi ini dilakukan agar proses pencarian/query data dapat dilakukan dengan mudah (R.Fitri, 2020). Rancangan class diagram tersebut dapat dilihat pada Gambar 8 beriku ini.



Gambar 8. Class Diagram Aplikasi Kerusakan CDI

f. Tampilan WEB Form Riwayat Aplikasi Kerusakan CDI.

Berikut tampilan WEB dari Aplikasi kerusakan CDI pada bagian form Riwayat pengecekan kerusakan CDI pada bagian ini berisikan informasi tentang histori jenis kerusakan pada sepeda motor matic ijeksi honda, selain itu juga diberikan informasi tentang grafik tentang kerusakan tersebut, semua fitur tersebut dapat dilihat pada Gambar 9.



Gambar 9. Tampilan WEB Form Riwayat Kerusakan CDI

3.3 Pengujian Web Aplikasi Kerusakan CDI.

Pengujian Aplikasi Kerusakan CDI pada penelitian ini menggunakan metode McCall. Metode McCall merupakan salah satu model yang menjelaskan Software Quality Factor atau factor kualitas perangkat lunak. Model ini memiliki tiga perspektif utama yaitu product operation (sifat-sifat operasional dari software), product revision (kemampuan software dalam menjalani perubahan), dan product transition (daya adaptasi software terhadap lingkungan baru). Product operation meliputi beberapa faktor yaitu correctness, reliability, usability, integrity, dan usability (Suhari, 2021). Hasil pengujian Aplikasi kerusakan CDI dapat dilihat pada Tabel 1 berikut ini.

Tabel 1. Hasil Pengujian WEB Aplikasi Kerusakan CDI

No.	Indikator	Keterangan	Nilai
1.	Correctness (Ketepatan) 0,3	Completeness (Kelengkapan)	3,44
		Consistency (Konsistensi)	5,27
		Traceability (lacak)	1,4
2.	Usability (Kegunaan) 0,2	Communicativeness (Komunikatif)	5,04
		Operability (Operabilitas)	3,36
		Training (Pelatihan)	3,36
3.	Integrity (Integritas) 0,3	Security (Keamanan)	2,68
4.	Reliability (Kehandalan) 0,2	Accuracy (Akurasi)	9,5
		Error Tolerancy (Toleransi kesalahan)	1,52
		Simplicity	2,37
5.	Efficiency (Efisiensi) 0,2	Execution Efficiency (Kemudahan Eksekusi)	3,72

Penghitungan masing-masing faktor kualitas dilakukan berdasarkan kriteria yang telah ditentukan adalah sebagai berikut :

Fa1 = 3,37
 Fa2 = 3,92
 Fa3 = 2,68
 Fa4 = 4,46
 Fa5 = 3,72

$$\sum = \frac{(0,3 \times Fa1) + (0,2 \times Fa2) + (0,3 \times Fa3) + (0,2 \times Fa4) + (0,2 \times Fa5)}{\text{Nilai Maksimum}} \times 100\%$$

$$\sum = \frac{(0,3 \times 3,37) + (0,2 \times 3,92) + (0,3 \times 2,68) + (0,2 \times 4,46) + (0,2 \times 3,72)}{5} \times 100\%$$

$$\sum = \frac{4,23}{5} \times 100\%$$

$$\sum = 84,6\%$$

Tabel 2 Kategori Kelayakan

Kategori	Presentase
Sangat Baik	81% - 100%
Baik	61% - 80%
Cukup Baik	41% - 60%
Tidak Baik	21% - 40%
Sangat Tidak Baik	< 21%

Dengan berdasarkan ketentuan bobot kelayakan yang sudah dijelaskan dengan skor 84,6%, maka Aplikasi Kerusakan CDI ini termasuk dalam kategori sangat baik.

4. Kesimpulan

Berdasarkan perancangan sistem hingga pengujian yang telah dilakukan serta hasil pengujian dan analisa perangkat keras dan perangkat lunak yang telah dibangun dapat disimpulkan bahwa:

- Aplikasi kerusakan CDI menghasilkan 7 form user interface dan 5 Tabel dalam rancangan database sedangkan
- Hasil dari pengujian dengan metode *McCall* menghasilkan presentase sebesar 84,6% (sangat baik).
- Pembacaan diagnosa kerusakan tidak dapat di lakukan pada saat mesin dalam keadaan hidup.
- Arduino Mega berjalan dengan baik tetapi sesekali mengalami *lost* respon dalam koneksi ke sensor.

Referensi

- A. Mugiprakoso & N. Hidayat. (2019). "Identifikasi Kerusakan Mesin Pada Sepeda Motor Menggunakan Metode Modified K-Nearest Neighbor (MKNN)," vol. 3, No. 4.

- C. Adimas, K. P. Kartika R, F. Dimas. (2022). Perancangan Aplikasi Solusi Kerusakan Sepeda Motor Karburator Berbasis Web Menggunakan Metode Waterfall. Vol. 6 No. 2. <https://doi.org/10.36040/jati.v6i2.5617..>

- Ilyas, M., Hutagalung, J., & Suparmadi, S. (2022). Penerapan Metode Forward Chaining Pada Sistem Pakar Untuk Diagnosa Kerusakan Sepeda Motor Berbasis Web. *Building of Informatics, Technology and Science (BITS)*, 4(2), 932–942. <https://doi.org/10.47065/bits.v4i2.2163>.

- L.A. Dewi & W.W. Wahyu. (2022). Expert Systems: Web-Based Motorcycle Detection Solutions. *International Journal of Computer and Information System (IJCIS)*. Vol 3, No 4. 148-152. <https://doi.org/10.29040/ijcis.v3i4.91>.

- N. Arora, R. & Arora. (2016). "Analysis of SDLC Model," *International Journal of Current Engineering and Technology*, vol. 6, no. 1, pp. 268–272, 2. Available: <http://inpressco.com/category/ijcet/>

- R.A. Sukanto & M. Shalahuddin. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung

- R. Fitri. (2020). *Pemrograman Basis Data Menggunakan MySQL*, 1st ed. Banjarmasin: Deepublish

- Rozaq, A., Hardinto, R. K., Annurrahman, A., & Susanti, D. (2018). Sistem Informasi Pembayaran Tambahan Penghasilan Berdasarkan Beban Kerja Pada Dinas Pendidikan Dan Kebudayaan Kabupaten Tanah Bumbu. *POSITIF : Jurnal Sistem Dan Teknologi Informasi*, 4(1), 1–11.

- Sitio, A. S., & Sianturi, F. A. (2021). Implementasi Metode Certainty Factor dalam Mengetahui Kerusakan Sepeda Motor Type Injeksi. *Jurnal Sains Dan Teknologi*, 3(1), 1-7. <https://doi.org/10.55338/saintek.v3i1.199>

- Suhari Camara M, A., Aelani, K., & Dwi Juniar S, F. (2021). Pengujian Kualitas Website menggunakan Metode McCALL Software Quality

- (studi kasus smkn4bdg.sch.id).
Journal of Information Technology,
3(1), 25–32.
<https://doi.org/10.47292/joint.v3i1.4>.
- Prakoso, A., & Wellem, T. (2022).
Perancangan dan Implementasi
Sistem Pemantauan Kualitas Udara
berbasis IoT menggunakan Wemos
D1 Mini dan Android. Building of
Informatics, Technology and
Science (BITS), 4(3), 1246–1254.
<https://doi.org/10.47065/bits.v4i3.2498>.
- Pressman, & Roger, S. (2012). Rekayasa
Perangkat Lunak: Buku I,
Pendekatan Praktisi (Edisi 7).
Yogyakarta: Andi.