

RANCANG BANGUN SISTEM INTEGRASI PENGOLAHAN DATA PELAPORAN *DRUM* *TEST* BERBASIS WEB

Anri Antonius Mardaup¹⁾
Teknik Elektronika, Politeknik Gajah Tunggal
anriantonius123@gmail.com

Adi Setyo Prasajo²⁾
Teknik Elektronika, Politeknik Gajah Tunggal
adisetyoprasajo@yahoo.com

Ilham Assegaf³⁾
Teknik Elektronika, Politeknik Gajah Tunggal
ilhamassegaf28@gmail.com

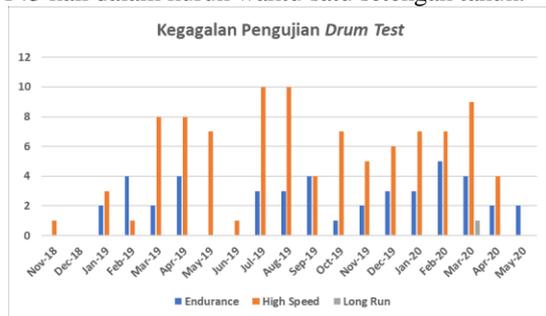
ABSTRAK

Tire Performance Testing merupakan sub departemen yang bertugas untuk menguji performa ban. Pengujian dilakukan dengan melakukan setting untuk setiap parameter pengujian tersebut. Keberhasilan pengujian sangat dipengaruhi oleh nilai dari setting parameter tersebut. Kajian ini bertujuan untuk membangun sebuah sistem integrasi pelaporan drum test berbasis aplikasi web. Aplikasi ini dapat mengurangi rugi losstime dalam pengujian khususnya drum test. Hal ini disebabkan karena masih adanya manual input ketika akan melakukan pengujian dan pembuatan laporan. Aplikasi berbasis web ini menggunakan framework laravel. Kemudian untuk database yang digunakan adalah dengan menggunakan MySQL sebagai server data dari pengujian. Data yang dipakai pada aplikasi ini bersumber langsung dari mesin drum test yang berekstensi *.mdb (file Ms. Access). Tranformasi dan pengubahan data tersebut ke dalam bentuk data di MySQL memanfaatkan aplikasi lain seperti pentaho. Hasil dari kajian ini menunjukkan bahwa aplikasi sistem integrasi tersebut dapat mengurangi rugi losstime dan dapat mengurangi terjadinya salah setting. Hal ini dikarenakan sistem tersebut menghilangkan proses pencatatan manual dan menggantikannya dengan menampilkan data secara otomatis yang bersumber dari database sistem untuk kemudian dapat dicetak sebagai pelaporan dari pengujian.

Kata Kunci: *Aplikasi web, Framework laravel, Database, Pelaporan*

I. PENDAHULUAN

Tire Performance Testing merupakan salah satu sub departemen di *Quality Assurance* bagian laboratorium yang bertugas dalam pengujian performa dari ban. Pengujian tersebut ditujukan untuk memastikan kualitas ban sudah sesuai dengan standar yang ditetapkan. Salah satu jenis pengujian yang dilakukan adalah pengujian *drum test*. Pengujian *drum test* sendiri memiliki berbagai jenis pengujian, yaitu *endurance test* dan *high speed test* dan *long run test*. Dalam pengujian tersebut terdapat kegagalan pengujian sebanyak 143 kali dalam kurun waktu satu setengah tahun.



Gambar 1. Grafik Data Hasil Pengujian Pada *Drum Test*

Kegagalan pengujian tersebut dikarenakan cara perhitungan pada parameter nilai *setting* pengujian masih manual. Proses manual tersebut menjadi hal yang perlu dikaji agar performa pengujian *drum test* tersebut meningkat. Pengkajian yang dapat dilakukan untuk mengurangi kesalahan dalam proses tersebut adalah dengan mengganti proses manual menjadi otomatis.

Proses otomatis ini dapat berupa pengembangan sistem lama maupun pembuatan sistem yang baru. Pengembangan maupun penggantian terkait sistem tersebut pada intinya adalah dapat meningkatkan hasil dengan mengurangi kerugian waktu maupun biaya yang digunakan.

II. LANDASAN TEORI

II.1. *Drum Test*

Drum Test adalah suatu alat penguji berbentuk *drum* yang dibuat sebagai *prototype* jalan raya dengan *temperature ambient* dan *temperature* ruangan sesuai dengan regulasi yang digunakan. [1] Jenis-jenis pengujian pada *drum test*:

a. *High Speed*

Pengujian terhadap kecepatan tinggi dengan kondisi *test* beban tetap dan kecepatan bertambah naik.

b. *Endurance*

Pengujian terhadap beban dengan kondisi *test* kecepatan tetap dan beban bertambah naik.

c. *Bead Durability / Long Run*

Pengujian performa ban untuk tujuan khusus contohnya mengetahui performa *bead* yang digunakan pada ban yang di uji.

II.2. *Microsoft Access (Ms. Access)*

Microsoft access merupakan salah satu *software* pengolah *database* yang berjalan di bawah sistem windows. *Microsoft Access* merupakan salah satu produk *Office* dari *Microsoft* yang dapat menangani *database* dengan skala besar maupun kecil. Dalam pengolahan *database*, *Microsoft Access* ini memiliki sarana atau objek-objek yang dapat mempermudah pekerjaan pengguna. [2]

II.3. *Structured Query Language (SQL)*

Structured Query Language (SQL) adalah sistem manajemen *database* relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada umumnya terdapat 3 (tiga) jenis perintah SQL yang bisa digunakan oleh SQL, yaitu: *Data Definition Language (DDL)*, *Data Manipulation Language (DML)*, dan *Data Control Language (DCL)*.

II.4. *Laravel*

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep *model, view, and controller (MVC)*. MVC adalah sebuah pendekatan yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen aplikasi seperti: [3]

a. Model

Model mewakili struktur data. Berisi fungsi-fungsi pengelolaan basis data.

b. View

View adalah bagian yang mengatur tampilan ke pengguna.

c. Controller

Controller merupakan bagian yang menjembatani *model* dan *view*.

II.5. *Basis Data (Database)*

Basis data dapat didefinisikan atau diartikan sebagai kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (*software*) program atau aplikasi untuk menghasilkan informasi. Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data untuk diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data,

menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga *update* yang rumit.

II.6. XAMPP

XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server *localhost*, dengan sistem operasi *Apache HTTP Server*, *MySQL database*, *PHP* dan *Perl*. [4]

II.7. Unified Modelling Language (UML)

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi objek dan desain, yaitu *Object Oriented Analysis and Design* (OOAD&D). UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek. Namun, UML juga digunakan untuk menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Bahasa pemodelan UML memiliki beberapa bagian utama, yaitu *view*, *diagram*, *model element*, dan *general mechanism*. [5]

II.8. Waterfall Process Model

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*) [6].



Gambar 2. Model Proses Waterfall [6]

III. KONSEP PERANCANGAN

III.1. Prosedur Pengembangan

a. Analisis Kebutuhan Sistem

Perancangan sistem integrasi pengolahan data pelaporan *drum test* berbasis web di *Sub. Dept. Tire Performance Testing* diawali dengan analisis kebutuhan sistem. Analisis ini diperlukan agar dalam proses pengembangan tepat sasaran dan dapat difungsikan dengan baik sebagai sistem pengolahan data pelaporan *drum test* berbasis web. Kebutuhan minimal yang harus ada menurut hasil observasi dan wawancara adalah sebagai berikut:

1) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk mengolah data pelaporan *drum test*.

2) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk mencari informasi *recipe* dari pengujian.

3) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk melakukan input *recipe* melalui aplikasi web.

4) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk mengolah data *recipe* dari pengujian.

5) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk membuat pelaporan *drum test* secara otomatis.

6) Sistem integrasi pengolahan data pelaporan *drum test* dapat digunakan untuk mengolah dan menampilkan

pelaporan *drum test* terkait metode uji yang dipilih.

b. Analisis Kebutuhan *Hardware* dan *Software*

Analisis spesifikasi kebutuhan *hardware* dan *software* dilakukan untuk mengetahui kebutuhan agar sistem informasi sekolah dapat dijalankan dengan baik tanpa terkendala. Spesifikasi dari alat-alat yang digunakan untuk menjalankan sistem ini antara lain:

1) PC/Laptop

2) Framework Laravel

3) Framework CSS Bootstrap

4) *Web Server* XAMPP versi 5.6.3

5) *Database server* MySQL

6) *Web browser*

c. Desain Sistem

Proses desain sistem menggunakan UML ini dibagi menjadi satu bagian, yaitu dari sisi *admin* atau operator. Sebagai *admin* akan memiliki hak akses sepenuhnya mengenai fungsi dan fitur sistem yang akan dibuat berdasarkan kebutuhan-kebutuhan sebagai *admin* dalam mengelola dan menjalankan sistem.

d. Implementasi / *Code System*

Pada sistem ini implementasi kode dibuat ke dalam bahasa pemrograman PHP dan ke *database* MySQL. Implementasi kode untuk membentuk fungsi-fungsi yang dibutuhkan oleh program dibuat dengan bantuan *framework* laravel dan *framework* bootstrap agar sesuai dengan keinginan pengguna dan implementasi dapat berjalan cepat dan mudah.

e. Pengujian Sistem

Sistem yang sudah selesai dibuat diuji dengan aspek *functionality*, *efficiency*, *reliability*, dan *usability*. Pada aplikasi pelaporan ini diuji hanya sampai tahap *functionality*. Pada tahap ini terdiri dari proses uji kelayakan sistem oleh seorang *web developer*. Uji kelayakan ini dilakukan menggunakan *test case* berupa *checklist* dengan sasaran ahli media yang memiliki keahlian pada bidang pemrograman dan desain aplikasi *web*.

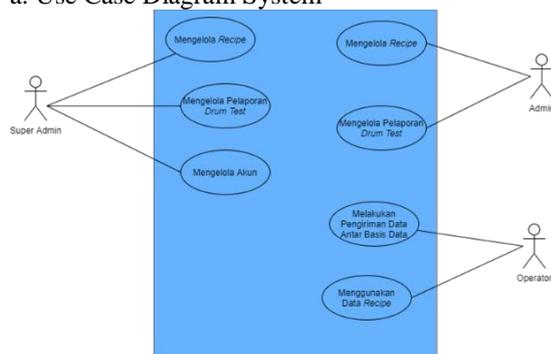
III.2. Metode Pengumpulan Data

- a. Observasi
- b. Studi Literatur

IV. HASIL DAN PEMBAHASAN

IV.1. Tahap Desain

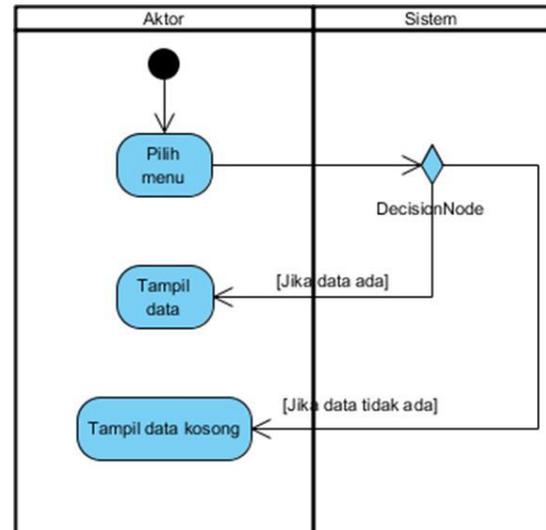
a. Use Case Diagram System



Gambar 3. Use Case Diagram System

Dalam sistem informasi yang dikembangkan terdapat tiga aktor yaitu *super admin*, *admin*, dan *operator*. *Super admin* memiliki fungsi diantaranya dapat mengelola data pelaporan *drum test*, mengelola *recipe*, mencetak laporan pelaporan *drum test* dan mengelola akun. *Admin* memiliki fungsi diantaranya dapat mengelola data pelaporan *drum test*, mengelola *recipe*, dan mencetak laporan pelaporan *drum test*. Sedangkan operator memiliki fungsi dapat menggunakan data *recipe* yang sudah di-input *admin* ke dalam mesin *drum test* dan melakukan pemindahan data antar basis data. Fungsi yang ada pada actor *super admin* dan *admin* harus melalui *use case login* terlebih dahulu, sedangkan untuk mengakses fungsi operator harus diakses melalui komputer yang terhubung dengan mesin *drum test*.

b. Activity Diagram

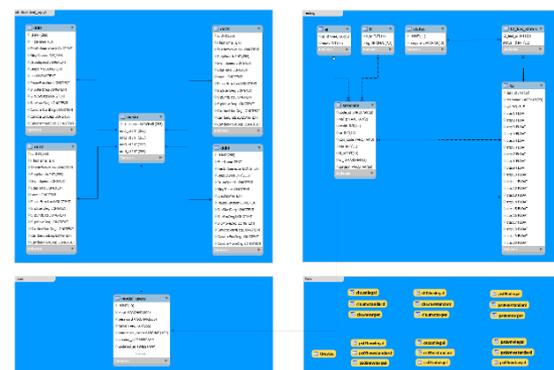


Gambar 4. Activity Diagram Tampil Data

Aktor memilih menu yang ada di sistem, kemudian sistem akan menampilkan data berupa sesuai dengan menu yang dipilih oleh aktor. Setelah itu sistem akan menampilkan daftar data, jika data ada dalam *database* maka data akan ditampilkan, jika tidak ada maka sistem tidak akan menampilkan daftar data.

c. Sequence Diagram

Aktor pertama kali harus melakukan *login* dengan memasukkan *username* dan *password*. Setelah berhasil *login*, aktor memilih melihat data. Kemudian aktor memilih data yang akan dilihat, sistem akan menerima masukan dan



Gambar 6. Perancangan Basis Data

Perancangan *database* terdiri dari 11 tabel. Kesebelas tabel tersebut yaitu tabel *model_users*, tabel mesin, tabel *ddt1*, tabel *ddt2*, tabel *ddt3*, tabel *ddt4*, tabel *li*, tabel *si*, tabel *sizecode*, tabel status, dan tabel *t2*. Tabel-tabel tersebut berisi informasi yang dibutuhkan pada sistem aplikasi. Informasi-informasi tersebut mulai dari informasi *user* yang terdapat *password*, *email*, nama, dan lain-

lain sampai dengan informasi pengujian yang dilakukan atau yang terdapat dalam sistem, seperti tabel t2 yang berisi informasi mengenai nama metode *test* yang dilakukan dan tabel status yang berisi mengenai *purpose* dari pengujian, apakah pengujian tersebut termasuk ke dalam kategori legal, standar ataukah target. Selain daripada tabel-tabel tersebut, terdapat pula tabel-tabel *view* yang merupakan gabungan dari beberapa tabel yang memuat informasi pengujian yang nantinya akan digunakan pada proses pencetakan pelaporan dari *drum test* itu sendiri.

e. Perancangan Desain *Interface Dashboard* Sistem



Gambar 7. Desain *Interface Dashboard*

Setelah *user login* maka yang akan ditampilkan adalah halaman *dashboard*. Halaman ini terdiri dari menu-menu *dashboard* yang diantaranya informasi tes, dan *input recipe*. Menu tersebut terdapat pada sisi kanan (*sidebar*).

f. Perancangan Halaman *Tambah Recipe*



Gambar 8. Desain Halaman *Input Recipe*

Halaman ini terdiri dari berbagai kolom seperti *Files Name*, *Test Name*, *Test Location*, *Method*, *Tire Type*, *Tire Code*, *Tire Size* dan *Brand*. Kolom tersebut dapat diisi sesuai dengan data yang di dapatkan, lalu di bawah kolom tersebut juga terdapat tombol “*SAVE*” yang berguna untuk menyimpan data yang sudah dimasukkan pada kolom tersebut.

g. Perancangan Halaman *Pelaporan Test*

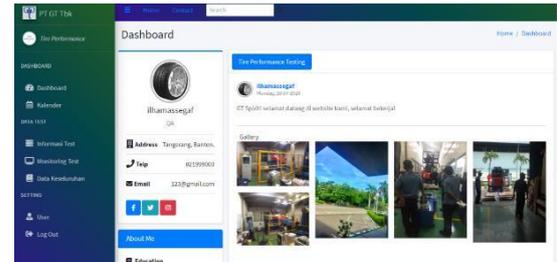


Gambar 9. Desain Halaman *Pelaporan Test*

Halaman ini terdiri dari berbagai kolom seperti *Files Name*, *Method*, dan *Test Name*. Kolom tersebut dapat diisi sesuai dengan data yang di dapatkan, lalu di bawah kolom tersebut juga terdapat tombol *PRINT* yang berguna untuk mencetak *form* pelaporan *drum test* berdasarkan pilihan kolom tersebut.

IV.2. Tahap Implementasi

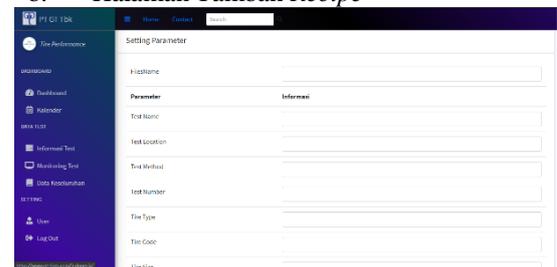
a. Halaman *Dashboard*



Gambar 10. Hasil Implementasi Halaman *Dashboard*

Halaman *dashboard* pada digunakan untuk mengakses menu tertentu.

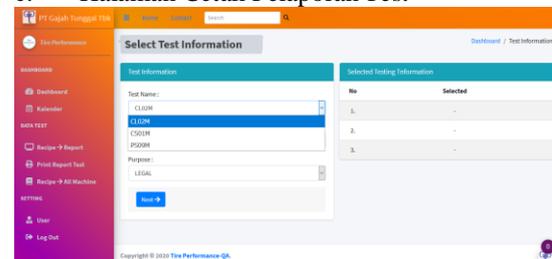
b. Halaman *Tambah Recipe*



Gambar 11. Hasil Implementasi Halaman *Tambah Recipe*

Halaman *tambah recipe* untuk menyimpan *recipe* ke dalam basis data.

c. Halaman *Cetak Pelaporan Test*



Gambar 12. Hasil Implementasi Halaman *Cetak Recipe*

Halaman cetak pelaporan *test* pada sistem digunakan untuk mencetak hasil pengujian dalam bentuk berkas atau dokumen *hardcopy*.

IV.3. Tahap *Pengujian Sistem*

a. Uji *Functionality*

No.	Fungsi	LOLOS	
		YA	TIDAK
I.	User Admin		
A	Akun		
1.	Login	3	0
2.	Mengubah password	3	0
3.	Lupa password	3	0
4.	Logout	3	0
B	Mengelola Data <i>Recipe</i>		
5.	Melihat informasi <i>recipe</i>	3	0
6.	Menambah data <i>recipe</i>	3	0
7.	Mengedit data <i>recipe</i>	3	0
8.	Menghapus data <i>recipe</i>	3	0
9.	Mencari data <i>recipe</i>	3	0
C	Mengelola Pelaporan <i>Drum Test</i>		
10.	Membuat pelaporan <i>drum test</i>	3	0
11.	Mengedit laporan <i>drum test</i>	3	0
12.	Mencari laporan <i>drum test</i>	3	0
13.	Menghapus laporan <i>drum test</i>	3	0
14.	Mencetak laporan <i>drum test</i>	3	0
D	Mengelola Basis Data		
15.	Perbandingan antara basis data	3	0

Gambar 13. Hasil Uji *Functionality*

Hasil dari uji *functionality* menunjukkan bahwa program dapat berjalan dan berfungsi sebagaimana mestinya.

V. KESIMPULAN DAN SARAN

V.1 Kesimpulan

Rancang bangun sistem integrasi pengolahan data pelaporan *drum test* berbasis web berhasil dibuat berdasarkan hasil uji *functionality*.

V.2 Saran

Peneliti dapat memberikan saran:

1. Penerapan sistem integrasi tersebut perlu dikembangkan untuk cakupan selain daripada *drum test*.
2. Diharapkan penelitian selanjutnya dapat mengkaji kehandalan dari sistem tersebut dengan menggunakan teknik pengujian kualitas *software* atau *tools* khusus.
3. Pengembangan sistem integrasi tersebut, selanjutnya dapat dilengkapi fitur yang lebih lengkap sehingga sistem ini bisa memberikan manfaat yang lebih dari sistem yang telah dikembangkan.

VI. DAFTAR PUSTAKA

- [1] Handoyo, Yopi. (2014). *Analisis Performance Ban Dengan Alat Drum Test*. Bekasi: Universitas 45.
- [2] Haer Talib. (2014). *Panduan Lengkap Ms Access 2013*. Cetakan Pertama, Jakarta: PT Elex Media Komputindo
- [3] Bean, M. (2015). *Laravel 5 Essentials*. Birmingham: Packt Publishing Ltd.
- [4] Nugroho. (2013). *Mengenal XAMPP Awal*. Yogyakarta: MediaKom.
- [5] Darwiyanti, S. (2003). *Pengantar Unified Modeling Language*. Ilmu

- komputer.
- [6] Sembiring, Kiniulin Br. (2010). "Metode SDLC untuk pengembangan system informasi WEB". Fakultas Ilmu Komputer Sistem Informasi, Universitas Sriwijaya.