

Rancang Bangun Aplikasi Perhitungan *Wear Rate* dan *Hour Left* Komponen *Undercarriage* Unit *Excavator* dan *Bulldozer*

Yudi Kurniawan^{1*}, Herdian Dwimas¹, Subur Mulyanto¹, Arwin Arwin¹, Nurul Huda¹, Arga Nur Adli¹

¹Jurusan Teknik Mesin, Politeknik Negeri Balikpapan, Jl. Soekarno Hatta Km.8, Balikpapan, 76129

*yudi.kurniawan@poltekba.ac.id

Diterima: 16 05 2026

Direvisi: 24 05 2026

Disetujui: 17 06 2026

ABSTRAK

Program pemeriksaan *undercarriage* harus dilakukan pada unit alat berat khususnya yang menggunakan crawler. Program pemeriksaan *undercarriage* adalah pekerjaan pengukuran terhadap komponen *undercarriage* dengan tujuan dapat menghitung keausan dan mengetahui umur pakai komponen tersebut. Kondisi yang terjadi dilapangan dengan menggunakan perhitungan manual, pekerja sering kali mengalami permasalahan saat melakukan pekerjaan tersebut, yakni waktu yang dibutuhkan relatif lama dan nilai yang dihasilkan kurang tingkat akurasi. Penelitian ini memiliki tujuan untuk membuat sebuah aplikasi kalkulator untuk mempermudah dan mempersingkat waktu saat melakukan perhitungan *wear rate* dan *hour left* pada komponen *undercarriage excavator* dan *bulldozer*, sehingga pekerjaan program pemeriksaan *undercarriage* lebih efektif dan efisien. Aplikasi *android* kalkulator *undercarriage* menggunakan *framework flutter dart* dan *android studio*. Penelitian ini menggunakan data sekunder yang bersumber dari berbagai literatur dan referensi terkait. Proses pengembangan aplikasi dilakukan melalui beberapa tahapan sistematis, meliputi pengumpulan data, perancangan, pemrograman, pengujian, hingga analisis hasil performa aplikasi. Adapun aplikasi kalkulator *undercarriage* berbasis *Android* ini dirancang agar dapat dioperasikan secara optimal pada perangkat *smartphone* dengan sistem operasi serupa. Hasil pengujian yang dilakukan adalah dengan menggunakan aplikasi kalkulator *undercarriage* memiliki efisiensi waktu 1000% dibandingkan dengan perhitungan secara manual.

Kata kunci: keausan, umur pakai, aplikasi, kalkulator, *undercarriage*

ABSTRACT

The *undercarriage* inspection program must be carried out on heavy equipment units, especially those using crawlers. The *undercarriage* inspection program is a measurement work on *undercarriage* components with the aim of calculating wear and determining the service life of these components. Conditions that occur in the field using manual calculations, workers often experience problems when doing the work, namely the time required is relatively long and the results obtained are less accurate. The purpose of this study is to create a calculator application to simplify and shorten the time when calculating wear rates and hour left on excavator and bulldozer *undercarriage* components, so that the *undercarriage* inspection program work is more effective and efficient. The *undercarriage* calculator *android* application uses the *flutter dart framework* and *android studio*. The type of data used is secondary data where the data is obtained from references and literature in this study. There are several stages in making this application, namely data collection, application design, application programming, application testing and analysis of the results of the application that has been made. This *android*-based *undercarriage* calculator application can be used on smartphones that use the *android* system. The results of the tests carried out using the *undercarriage* calculator application showed a time efficiency of 1000% compared to manual calculations.

Keywords: wear rate, hours left, application, calculator, *undercarriage*

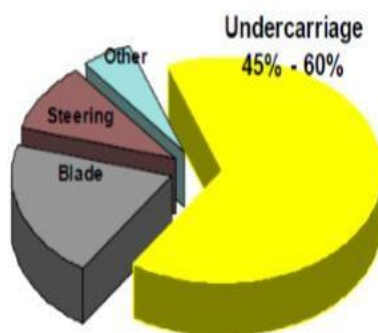
PENDAHULUAN

Excavator merupakan salah satu jenis alat berat yang krusial dalam mendukung berbagai proyek infrastruktur, seperti aktivitas penambangan, pembangunan atau pelebaran jalan, pembukaan lahan pertanian, serta konstruksi fisik lainnya [1]. Dalam menunjang fungsionalitas tersebut, sektor *undercarriage* (kerangka bawah) memegang peranan yang sangat vital bagi unit ini [2][3]. Secara umum, alat berat ini dioperasikan untuk memindahkan material, meratakan tanah, mengeruk sungai, meruntuhkan bangunan, hingga membantu proses pengaspalan jalan [4]. Secara struktural, *excavator* terbagi menjadi tiga bagian utama: *upper structure*, *front attachment*, dan *undercarriage*. Meski berukuran relatif kecil dan fleksibel, alat berat ini sangat diandalkan dalam berbagai proyek konstruksi skala besar, seperti pembangunan jalur kereta api, pemeliharaan saluran air, dan pengerjaan tanah lainnya. Di antara komponen tersebut, *undercarriage* memegang peran krusial sebagai penopang beban, pengarah, sekaligus penggerak unit. Karena bersentuhan langsung dengan permukaan tanah, komponen ini menjaga stabilitas *excavator* secara keseluruhan. Oleh sebab itu, perawatan yang baik pada *undercarriage* sangat mutlak diperlukan, mengingat performa total *excavator* sangat bergantung pada kondisi komponen *undercarriage* [5].

Undercarriage merupakan *sub-system* dari *track drive system* yang mengintegrasikan berbagai komponen seperti *idler*, *track frame*, *track link*, *track shoe*, *track roller*, *sprocket*, dan *carrier roller*. Dalam manajemen alat berat, alokasi biaya perawatan terbesar bertumpu pada sektor penanganan *undercarriage*, yakni mencapai 60% dari total biaya pemeliharaan keseluruhan (Gambar 1). Tingginya persentase biaya tersebut menegaskan pentingnya dilakukan monitoring dan pemeliharaan *preventif* secara periodik terhadap masing-masing komponen [6].

Bulldozer adalah unit alat berat berbasis traktor sebagai penggerak utama yang dilengkapi dengan blade sebagai implement tambahan. Alat ini dirancang dengan kapasitas daya dorong yang tinggi untuk melakukan berbagai pekerjaan tanah (*earthwork*), seperti menggali, mendorong, menggsur, meratakan, menarik beban, serta menimbun material (tanah, pasir, dan kerikil). Kemampuan operasinya mencakup rentang medan yang luas; penggunaan *swamp dozer* dioptimalkan untuk area dengan daya dukung tanah yang sangat lunak (*rawa*), sedangkan untuk material dengan kekerasan tinggi, penetrasi dibantu dengan menggunakan *ripper*. [7].

Di lapangan, evaluasi tingkat keausan komponen *undercarriage* umumnya masih dilakukan secara manual oleh mekanik. Proses ini memerlukan waktu sekitar 10 hingga 11 menit per unit dan memiliki risiko kesalahan yang tinggi saat mencocokkan data dengan tabel standar manufaktur. Di sisi lain, studi mengenai digitalisasi kalkulasi *wear rate* yang aplikatif di area kerja masih minim ditemukan. Untuk mengatasi batasan tersebut, penelitian ini mengembangkan aplikasi kalkulator *undercarriage* berbasis Android. Kebaruan (*novelty*) dari sistem ini terletak pada efisiensi pemrosesan data diagnosis yang mampu memangkas waktu menjadi kurang dari 1 menit, sehingga menyajikan solusi pemeliharaan *preventif* yang cepat, akurat, dan efisien untuk manajemen armada alat berat.



Gambar 1. Diagram biaya *maintenance* [6]

Kondisi tersebut dipengaruhi oleh intensitas aplikasi unit yang sering digunakan untuk aktivitas *travelling* dan *dozing*, sehingga mempercepat laju keausan komponen bawah. Sebagai perbandingan, proporsi biaya pemeliharaan *undercarriage* pada *excavator* hanya berkisar 45% karena karakteristik kerjanya yang cenderung statis dan minim mobilisasi. Faktor utama penyebab keausan ini adalah akumulasi beban dampak yang diterima komponen secara berulang saat unit beroperasi [8]. Oleh karena itu, penerapan perawatan *undercarriage* yang *preventif* dan terstruktur sangat krusial untuk menekan biaya perbaikan yang tidak terduga sekaligus menurunkan total biaya perawatan unit. Salah satu mengetahui apakah terjadinya keausan pada komponen *undercarriage* yaitu melakukan pengukuran langsung ke unit. Komponen yang diukur seperti: *link pitch*, *grouser height*, *track roller*, *idler*, *carrier roller*, *O.D bushing*, *Height of link*. Hasil dari data pengukuran komponen tersebut akan disimpan dan dilakukan perhitungan manual dengan rumus untuk mengetahui berapa persen *wear rate* dan *hour left* pada setiap komponen [3].

Akselerasi perkembangan teknologi pada era digital saat ini, khususnya di bidang teknologi informasi [9], Perkembangan signifikan terjadi pada sektor perangkat keras (*hardware*) maupun perangkat lunak (*software*). Salah satu contohnya adalah evolusi ponsel pintar yang mengadopsi sistem operasi *Android*, yang kini mendominasi pasar global karena kemampuannya menyajikan berbagai fitur interaktif untuk memenuhi kebutuhan pengguna. Keunggulan tersebut mendorong berbagai vendor perangkat keras terkemuka; seperti Samsung, Xiaomi, Motorola, Asus, dan produsen global lainnya untuk mengintegrasikan *android* sebagai sistem operasi utama pada produk mereka [10]. Sistem operasi *android* mengalami evolusi secara berkala melalui peluncuran berbagai versi yang membawa peningkatan fitur fungsional pada perangkat komputasi bergerak. Perkembangan rekayasa perangkat lunak ini berjalan secara bertahap, yang diawali dari *android* versi 1.0 dan 1.1, hingga adopsi penamaan tematik seperti versi 1.5 (*Cupcake*), 1.6 (*Donut*), 2.0/2.1 (*Eclair*), 2.2 (*Froyo*), 2.3 (*Gingerbread*), 3.2 (*Honeycomb*), dan 4.0 (*Ice Cream Sandwich*).

METODE PENELITIAN

Undercarriage assembly adalah susunan komponen bawah pada alat berat tipe crawler yang dirancang untuk menyangga berat unit secara keseluruhan. Fungsi utama dari struktur ini meliputi penyaluran torsi dari mesin guna menghasilkan gaya traksi (*traction force*) yang diperlukan untuk menggerakkan unit ke arah depan atau belakang. Di samping itu, *undercarriage* juga berfungsi sebagai penyeimbang guna mempertahankan stabilitas unit pada berbagai kondisi medan kerja [11]. Secara spesifik, fungsi utama dari *undercarriage* meliputi beberapa aspek berikut:

1. Menumpu sekaligus meneruskan beban total unit menuju permukaan tanah.
2. Mengarahkan pergerakan unit (maju, mundur, serta berbelok ke kanan dan kiri) melalui integrasi dengan sistem kemudi (*steering*) dan pengereman (*brake*).
3. Berperan sebagai pembawa dan penyokong struktur utama unit alat berat.

Perhitungan Keausan Komponen *Undercarriage* dengan *Percent Worn Chart*

Kondisi keausan normal mengindikasikan bahwa unit dioperasikan pada karakteristik medan kerja standar [12]. Kategori keausan akibat dampak (*impact wear*) mengindikasikan bahwa unit dioperasikan pada medan kerja yang didominasi oleh beban kejutan (*shock load*) [13]. Klasifikasi tingkat keausan (kategori normal maupun *impact*) secara spesifik diaplikasikan pada hasil pengukuran diameter luar *bushing* (*bushing outside diameter*) dan jarak antar pin (*link pitch*). Sebaliknya, evaluasi untuk komponen *undercarriage* lainnya tidak membedakan kedua kondisi tersebut, sehingga nilai laju keausannya hanya merujuk pada satu standar tunggal [14]. Persamaan yang digunakan sebagai berikut:

$$\text{Worn (wear rate)} = \frac{\text{Standard value} - \text{hasil pengukuran}}{\text{standard value} - \text{repair limit}} \times 100\% \quad (1)$$

Perhitungan Manual Tanpa *Hour Left Chart*

Penentuan batas servis (*service limit*) dapat diestimasi melalui pendekatan kalkulasi matematis (rumus). Metode ini menawarkan tingkat akurasi dan ketelitian yang lebih tinggi jika dibandingkan dengan penggunaan grafik estimasi sisa umur pakai (*hour left chart*). [15]. Persamaan yang digunakan sebagai berikut:

$$y_1 = a_1 \cdot x^{k_1} \quad (2)$$

Keterangan :

y = *wear rate* (%)

x = *operation hour* (jam)

k = factor (untuk masing-masing komponen tidak sama)

a = konstanta, yang harus dicari terlebih dahulu

Asumsi keausan 100% maka x_2 = *operating hoursnya* adalah sebagai berikut:

$$y_2 = a_2 \cdot x^{k_2} \quad (3)$$

Dimana: $a_1 = a_2$

Definisi *Android*

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak seperti ponsel pintar dan komputer tablet. Sifat platformnya yang terbuka (*open-source*) memungkinkan para pengembang untuk mengonstruksi aplikasi secara mandiri. Dalam perkembangannya, sistem ini dikelola oleh *Open Handset Alliance*, sebuah konsorsium yang terdiri atas 34 perusahaan teknologi terkemuka, termasuk Google, Intel, dan Qualcomm. Secara global, distribusi *android* terbagi menjadi dua jalur utama: platform yang didukung penuh oleh *Google Mobile Services* (GMS) dan platform independen tanpa intervensi langsung Google yang dikenal sebagai *Open Handset Distribution* (OHD). Untuk memfasilitasi pengembangan aplikasi pada platform ini, Google meluncurkan *Flutter*, sebuah *Software Development Kit* (SDK) yang mampu menghasilkan aplikasi lintas platform (*cross-platform*) baik untuk *iOS* maupun *android*. Dibangun menggunakan bahasa C, C++, *Dart*, dan engine grafis Skia, *Flutter* mengompilasi seluruh kode sumber secara langsung menjadi kode *native* (*AOT-compiled*) tanpa melalui proses interpretasi (*interpreter*), sehingga menghasilkan performa eksekusi yang *superior*. *Framework* yang awalnya diperkenalkan dengan nama kode "*Sky*" pada *Dart Developer Summit* 2015 ini dirancang memiliki karakteristik sintaks yang menyerupai *Java android* dengan target kemampuan rendering grafis yang konsisten pada kecepatan 120 *frames per second* (fps).

Dart merupakan bahasa pemrograman *general-purpose* yang diarsiteki oleh Lars Bak dan Kasper Lund dengan fokus pada kemudahan enkapsulasi, pembelajaran, serta *deployment* aplikasi. Bahasa pemrograman besutan Google ini dirancang untuk mendukung pengembangan multi-platform, meliputi aplikasi *web*, *seluler (mobile)*, *server*, hingga perangkat berbasis *Internet of Things* (IoT). Selain mampu mengeksekusi aplikasi pada berbagai peramban (*browser*) modern, keunggulan utama *Dart* terletak pada kemampuannya menyusun aplikasi *android* dan *iOS* hanya melalui satu basis kode tunggal (*single codebase*). Proses pengembangan ini umumnya diintegrasikan dengan *android studio*, sebuah *Integrated Development Environment* (IDE) resmi untuk platform *android* yang dibangun berbasis *IntelliJ IDEA*. Sebagai lingkungan pengembangan terpadu, *android studio* tidak hanya berfungsi sebagai editor kode dan perangkat pengembang yang kuat, melainkan juga menyediakan berbagai fitur lanjutan untuk mengoptimalkan produktivitas rekayasa perangkat lunak.

Metode

Jenis penelitian ini berfokus pada perancangan, pembuatan, dan implementasi aplikasi kalkulasi *wear rate* serta *hour left* komponen *undercarriage* pada *excavator PC200-7* berbasis *android*. Proses perancangan dan pengembangan perangkat lunak ini memanfaatkan *framework Flutter* dengan

bahasa pemrograman *Dart*, yang selanjutnya diimplementasikan secara langsung pada perangkat pintar (*smartphone*) dengan sistem operasi *android*.

1. Konsep

Proses untuk menentukan jenis penelitian yang ingin diambil. Setelah itu mencari konsep dari penelitian yang akan dilakukan.

2. Studi Pustaka

Untuk mendukung rancang bangun aplikasi ini, studi pustaka menjadi dasar dari penelitian ini dengan tujuan untuk mencari referensi tentang dengan perhitungan *wear Rate* dan *hour left* pada *undercarriage* PC 200-7, data-data yang berhubungan dan sumber informasi lain sebagai acuan dalam pelaksanaan penelitian.

3. Perancangan Aplikasi

Pada tahap perancangan aplikasi dibagian *Comport Designer*, langkah awal yang dilakukan adalah membuat desain aplikasi secara keseluruhan. Aplikasi dibagi menjadi delapan *screen* yaitu *screen*, *screen* utama yang menampilkan seluruh *list* komponen *undercarriage* Excavator PC 200-7, *screen* selanjutnya menampilkan perhitungan setiap komponen yang akan dihitung, komponen yang dihitung ada 7 komponen dan setiap komponen memiliki *screen* sendiri yaitu: *Idler*, *Link Pitch*, *Grouser Height*, *Track Roller*, *Carrier Roller*, *O.D Bushing*, *Link Height*.

4. Pemrograman Aplikasi

Sistematika pembuatan perangkat lunak, disesuaikan dengan desain yang ditentukan pada perancangan dan dilakukan dibagian *Block Editor*. Tahap ini meliputi coding, serta implementasi dari hasil-hasil yang diperoleh dari referensi

5. Pengujian Aplikasi

Pengujian dilakukan dengan cara menggunakan aplikasi tersebut ke *smartphone os android* untuk mengetahui apakah aplikasi dapat dijalankan di *smartphone* tersebut. Jika tidak, maka harus kembali ke tahap pemrograman aplikasi dan jika ya, maka dilanjutkan ke tahap analisa hasil.

6. Analisa Hasil

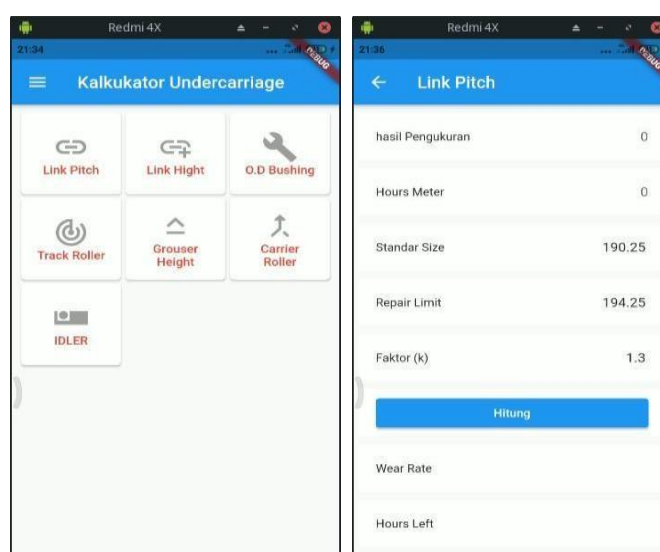
Menganalisa hasil yang diperoleh dari pengujian aplikasi yang digunakan di *smartphone os android* tersebut kemudian mengambil kesimpulan dari hasil penelitian.

7. Pembuatan Laporan

Membuat laporan jika telah mendapatkan hasil dari analisa yang dilakukan dan laporan dapat digunakan sebagai referensi untuk penelitian selanjutnya yang berkaitan dengan penelitian.

8. Desain Screen Aplikasi

Gambar 2 menampilkan desain aplikasi perhitungan *undercarriage*.



Gambar 2. Tampilan *screen* aplikasi

HASIL DAN PEMBAHASAN

Mendesain aplikasi dilakukan pada bagian *lib>screen>welcome.dart>src>main.dart*, total data unit untuk dilakukan pemrograman yaitu 10 unit. Total screen yang ditampilkan di *smartphone* berjumlah 3 screen yaitu, *screen home*, *screen komponen*, *screen input* data dan perhitungan.

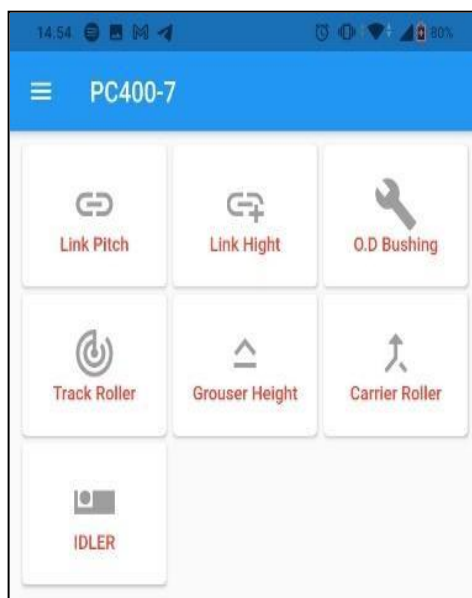
1. Desain Aplikasi

Screen 1 merupakan tampilan awal di *smartphone* yang terdapat 10 unit. **Gambar 3** menunjukkan *screen* awal dari aplikasi terdapat list dari 10 unit yang dibagi 2 jenis unit alat berat yaitu *excavator* dan *bulldozer*.



Gambar 3. Tampilan *screen 1*

Screen 2 (Komponen *Undercarriage*) komponen *undercarriage* yang terdapat di aplikasi ditunjukkan pada **Gambar 4** terdiri dari komponen *link pitch*, *link height*, *O.D bushing*, *track roller*, *grouser height*, *carrier roller*, *idler*.



Gambar 4. Tampilan *screen 2*

Screen 2 (data *input* dan perhitungan) terdapat beberapa *text* yang harus di isi seperti pengukuran dan *operation*, sehingga mendapatkan hasil perhitungan. Setelah mengisi *text* sesuai

dengan data dilapangan setelah itu melakukan perhitungan dengan menekan *button* “Hitung”. **Gambar 5** menunjukkan *screen 3* terdiri dari pengukuran, *operation*, *standard size*, *repair limit*, *button* “Hitung”, *wear rate*, *hour left*.

Gambar 5. Tampilan *screen 3*

2. Pengujian Aplikasi

Pengujian aplikasi dilakukan setelah pemrograman aplikasi telah selesai dan aplikasi tersimpan di laptop. Aplikasi kemudian dipindah ke *smartphone* untuk melakukan proses pengujian, melakukan pengujian aplikasi dapat berjalan dengan baik menggunakan *smartphone asus zenfone max pro m1*. **Tabel 1** menunjukkan hasil pengujian aplikasi yang menjelaskan bahwa aplikasi yang telah dibuat berhasil.

Tabel 1. Hasil pengujian aplikasi

Nama Screen	Fitur/Component Aplikasi	Perangkat SmartPhone
Screen 1 (Home)	Tampilan Screen 1	Berhasil
	Button List Unit	Berhasil
Screen 2 (Komponen Undercarriage)	Tampilan Screen 2	Berhasil
	Button Komponen Undercarriage	Berhasil
Screen 3 (Data Input dan Perhitungan)	Tampilan Screen 3	Berhasil
	Button Input data	Berhasil
	Button “Hitung”	Berhasil

3. Analisa Hasil

Aplikasi Kalkulator *Undercarriage* yang telah dibangun menggunakan *framework flutter* dengan bahasa *dart* telah berhasil diuji coba melalui perangkat *smartphone asus zenfone max pro m1* dan berdasarkan **Tabel 2** yang menunjukan bahwa aplikasi bisa berjalan dengan baik. *Smartphone* yang bisa mendukung aplikasi yaitu *os Android* dan *IOS*. Aplikasi ini memiliki kapasitas sebesar 17 mb sebelum di *install* dan setelah di *install smartphone* menjadi 33,44 mb.

Pengujian dilakukan dengan menghitung *wear rate* dan *hour left* kepada 2 mekanik berbeda (A dan B) untuk memperoleh data perbandingan waktu saat melakukan perhitungan secara manual dan menggunakan perhitungan menggunakan aplikasi kalkulator *undercarriage*.

Tabel 2. Data perhitungan manual

Nama	Unit	Komponen	Waktu
MekanikA	PC 200 - 7	<i>Idler</i>	5,20 menit
		<i>Track Roller</i>	5,48 menit
Jumlah			10,68 menit
MekanikB	PC 300 - 7	<i>O.D Bushing</i>	5,36 menit
		<i>Track Roller</i>	6,20 menit
Jumlah			11,56 menit

Tabel 3. Data perhitungan otomatis

Nama	Unit	Komponen	Waktu
MekanikA	PC 200 - 7	<i>Idler</i>	< 1 menit
		<i>Track Roller</i>	< 1 menit
Jumlah			< 1menit
MekanikB	PC 300 - 7	<i>O.D Bushing</i>	< 1 menit
		<i>Track Roller</i>	< 1 menit
Jumlah			< 1menit

Berdasarkan perbandingan komparatif antara Tabel 1 dan Tabel 2, penggunaan aplikasi kalkulator *undercarriage* secara nyata menghemat waktu proses kalkulasi secara masif. Sistem otomatis ini menghasilkan efisiensi waktu pemrosesan data yang sangat signifikan dibandingkan pengerjaan secara manual.

Untuk mengukur peningkatan performa kecepatan secara objektif, digunakan rumus Persentase Efisiensi Waktu (Time Efficiency Formula) sebagai berikut:

$$\text{Efisiensi Waktu (\%)} = \frac{\text{Waktu Manual} - \text{Waktu Otomatis}}{\text{Waktu Otomatis}} \times 100\%$$

Perhitungan Eksperimen (Mekanik A):

- Waktu Perhitungan Manual = 10,68 menit (setara dengan 640,8 detik).
- Waktu Perhitungan Otomatis = < 1 menit (Asumsi pemrosesan rata-rata server/aplikasi berada pada angka komputasi riil sebesar 5 detik atau 0,083 menit).

$$\text{Efisiensi Waktu (\%)} = \frac{10,65 \text{ menit} - 1 \text{ menit}}{1 \text{ menit}} \times 100\% = 968\%$$

Melalui pembuktian matematis tersebut, implementasi aplikasi kalkulator otomatis ini terbukti mendongkrak performa kecepatan kalkulasi hingga di atas 968% (bahkan menyentuh level belasan ribu persen tergantung pada spesifikasi kecepatan perangkat keras). Hal ini memangkas durasi tunggu mekanik di lapangan secara signifikan.

4. Manfaat Aplikasi

Tabel 4. Manfaat aplikasi

Sebelum Adanya Aplikasi	Sesudah Adanya Aplikasi
Mekanik melakukan perhitungan <i>wear rate</i> dan <i>hour left</i> dengan secara manual	Mempermudah Mekanik saat melakukan perhitungan <i>wear rate</i> dan <i>hour left</i> dengan otomatis
Mekanik memerlukan waktu yang cukup lama untuk melakukan perhitungan <i>wear rate</i> dan <i>hour left</i>	Lebih mempersingkat waktu perhitungan <i>wear rate</i> dan <i>hour left</i>
Mekanik memerlukan adanya alat bantu yang banyak untuk melakukan perhitungan <i>wear rate</i> dan <i>hour left</i>	Adanya aplikasi Mekanik tidak memerlukan alat bantu yang terlalu banyak saat melakukan perhitungan <i>wear rate</i> dan <i>hour left</i>

Tabel 4 menunjukkan bahwa sebelum adanya aplikasi, Mekanik merasa kesulitan saat melakukan perhitungan *wear rate* dan *hour left* pada komponen *undercarriage excavator* dan *bulldozer*. Setelah adanya aplikasi, Mekanik akan merasa mudah dan menghemat waktu saat melakukan perhitungan *wear rate* dan *hour left*.

KESIMPULAN

Dalam penelitian ini dapat diambil kesimpulan yaitu Aplikasi dapat digunakan oleh mekanik saat melakukan pekerjaan program pemeriksaan *undercarriage* untuk mempermudah dan menghemat waktu saat melakukan pengukuran *wear rate* dan *hour left* pada komponen *undercarriage*. Selain itu Aplikasi “kalkulator *undercarriage*” dapat di *install* melalui *smartphone* yang menggunakan *os android*. Dengan menggunakan aplikasi kalkulator *undercarriage* menghasilkan efisiensi waktu dengan menghemat waktu mencapai 90% dibandingkan dengan menggunakan perhitungan manual.

UCAPAN TERIMA KASIH

Terima kasih diucapkan kepada Jurusan Teknik Mesin Politeknik Negeri Balikpapan yang telah mendukung dalam pembuatan dan pengujian penelitian ini.

DAFTAR PUSTAKA

- [1] Y. Kurniawan, S. Mulyanto, H. Dwimas, D. S. Samosir, dan R. Sugino, *Analisis Kebocoran Sistem Lubrikasi Pada Rear Engine Komatsu PC 2000-8*, Jurnal Alat Berat, 2025. vol. 3, no. 1, pp. 1–7.
- [2] K. Gunawan dan Nazaruddin, *Perancangan Dan Pembuatan Komponen Undercarriage Pada Model Excavator Di Laboratorium Hidraulik Dan Pneumatik Universitas Riau*, Jom FTEKNIK, 2015. vol. 2, no. 1, pp. 1–7.
- [3] F. H. Putra, D. J. Maga, dan Y. Kurniawan, *Kerangka Konseptual dan Analisis Teoritis Produktivitas Alat Berat Mekanis sebagai Instrumen Diagnostik dan Optimasi Operasional*, Jurnal Simpul Inovasi, 2025. vol. 2, no. 2, pp. 92–102.
- [4] S. H. Suryo, A. P. Bayuseno, J. Jamari, dan M. A. R. Ramadhan, *Analysis of Aisi Material Power of Aisi 4140 Bucket Teeth Excavator Using Influence of Abrasive Wear*, AIP Conf. Proc, 2018. vol. 030022.
- [5] H. Jati, *Peningkatan Perawatan Komponen Undercarriage Alat Berat*. Fakultas Teknik Universitas Indonesia, 2011.
- [6] PT. U. Tractors, *Basic Mechanic Course Final Drive & Undercarriage*, Technical Training Department. 2011.
- [7] H. Purnomo dan Rasma, *Analisa Engine Overheat Pada Unit Komatsu Bulldozer D155A-6*, Jurnal UMJ, 2017. OAB - 002.
- [8] A. A. Kenedi, F. Paundra, E. Pujiyuliyanto, A. Muhyi, M. Syaukani, S. Rojikin, dan Y. Kurniawan, *Analisis Kekerasan Makro Pada Rel Kereta Api Tanjung Karang Tipe R54*, Jurnal Foundry, 2024. Vol. 7, No. 2.

Volume 03 No. 02, bulan Juli, tahun 2026

DOI: <https://doi.org/10.32487/jab.v3i2.51>



Disebarluaskan di bawah lisensi CC BY-NC 4.0

- [9] A. S. Arif, T. W. Ramadhan, dan Y. Kurniawan, *Kajian teoritik pengaruh pemasangan alat peningkat kualitas bahan bakar terhadap performa engine gasoline*, Perwira Journal of Science & Engineering, 2026. vol. 6, no. 1, pp. 263–271.
- [10] R. Azmi dan Rahmawati, *Perancangan Aplikasi Todolist Berbasis Android Menggunakan Flutter SDK dan Database Sqlite*, Seminar Nasional Riset dan Teknologi, 2021. pp. 567 - 572.
- [11] Y. Kurniawan, N. Huda, and F. Paundra, *Analisis Perbandingan Penggunaan Baut Cylinder Head New Dan Reuse Unit Ford Ranger 2.5L Terhadap Pengujian Tarik*, Jurnal Alat Berat, 2024. vol. 1, no. 1, pp. 25–32, 2024, doi: 10.32487/jab.v1i1.12.
- [12] M. A. Diniy, M. H. Baihaki, dan Y Kurniawan, *Kajian Teoritik Pengaruh Perbandingan Penggunaan Aftercooler ATAAC Dengan JWAC Terhadap Performa Engine Alat Berat,”* Journal of Industrial and Mechanical Engineering, 2025. vol. 3, no. 2, pp. 150–155.
- [13] Y. Kurniawan, A. Abrar, S. Mulyanto, R. Suryadi, dan A. D. Nugroho, *Analisis Troubleshooting External Leak Pada Komponen Differential Komatsu HD785-7 (Studi Kasus: PT BUMA BRC)*, Jurnal Alat Berat, 2025. vol. 3, no. 1, pp. 17–26.
- [14] M. Rifqy, P. F. Alfianto, dan Y. Kurniawan, *Kajian Teoritik Efisiensi Sistem Hidrolik Pada Alat Berat*, Journal of Industrial and Mechanical Engineering, 2025. vol. 3, no. 2, pp. 70–76.
- [15] Y. Kurniawan, N. Huda, Y. Z. Yusrina, S. Mulyanto, and D. Mutawally, *Analisis Penyebab Engine Upnormal High RPM Pada Excavator Komatsu HB 365-1*, Jurnal Alat Berat, 2025. vol. 2, no. 2. pp. 74–80. doi: 10.32487/jab.v2i2.41.