Rancang Bangun Perangkat Lunak *Monitoring* Menggunakan Sensor Timbangan Dinamis Terhadap Muatan Kendaraan Dan Penindakan Pada Gerbang Tol

Yuni Arkhiansyah[1], Muhammad Hidayat[2]

[1,2]Institut Informatika dan Bisnis Darmajaya

[1]*yuniarki@darmajaya.ac.id*, [2]*dayatt.1711010174@mail.darmajaya.ac.id*

ABSTRAK

Perkembangan teknologi informasi dan komunikasi di era sekarang ini begitu cepat, kebutuhan untuk melakukan aktivitas dan komunikasi secara *realtime* sangat dibutuhkan didalam beberapa bidang industri seperti misalnya di Gerbang Tol. Dan salah satu medianya adalah alat sensor, adapun alat sensor yang dimaksud adalah timbangan kendaraan dinamis (sensor) yang mampu mendeteksi beban keseluruhan kendaraan tanpa menimbulkan antrian yang panjang digerbang tol. Metode pengembangan perangkat lunak yang digunakan dalam sistem ini adalah metode Prototype. Dimana tahapan Protoype terdiri dari Mendengarkan Pelanggan, Membangun dan Memperbaiki Prototype serta Uji Coba. Metode prototype digunakan untuk menggali kebutuhan secara lebih tepat dan melibatkan pengguna secara langsung yang dalam hal ini ditemukan beberapa masalah. Hasil Penelitian ini dijelaskan mengenai Hasil dan Implementasi progam dari berbagai tahapan yang telah dirancang sebelumnya. Perlu diperhatikan sebelum Admin mulai menjalankan perangkat lunak untuk melakukan *monitoring*, pastikan perangkat (*personal computer*) memiliki sistem operasi *Windows 10 Pro*, dan terknoneksi dengan jaringan , dan juga perangkat lunak *Backend* telah di konfigurasi oleh seorang Teknisi. Setelah poin-poin yang disebutkan sudah tidak ada masalah, maka Admin dapat menjalankan perangkat lunak *monitoring* tersebut.

Kata kunci: *Weigh in Motion Scale Sensors, Desktop Application,* Perangkat Lunak Monitoring, *Prototype, Windows 10.*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi di era sekarang ini begitu cepat, kebutuhan untuk melakukan aktivitas dan komunikasi secara *realtime* sangat dibutuhkan didalam beberapa bidang industri seperti misalnya di Gerbang Tol. Dan salah satu medianya adalah alat sensor, adapun alat sensor yang dimaksud adalah timbangan kendaraan dinamis (sensor) yang mampu mendeteksi beban keseluruhan kendaraan tanpa menimbulkan antrian yang panjang digerbang tol. Penelitian ini akan menggunakan antarmuka *desktop* (*desktop application*) sebagai tampilan yang dapat dilihat oleh *user*/pengguna dan sistem pendukung yang bergerak dibelakang layar sebagai basis penelitian yang mendukung kinerja penggunaan alat sensor, yang dimana dari kedua perangkat lunak yang akan dibangun berjalan di sistem operasi *Windows 10*. Dalam membangun perangkat lunak yang akan dibuat penulis menggunakan *Microsoft Visual Studio* sebagai *IDE* yang digunakan untuk membangun perangkat lunak.

IDE yang digunakan dalam membangun perangkat lunak adalah *Microsoft Visual Studio*, karena perangkat lunak yang akan digunakan akan berjalan di sistem operasi *Windows 10*. Dan lebih mudah jika ingin melakukan integrasi dengan perangkat lunak lainnya yang berbasis *Windows*. Sehingga penulis memilih *Microsoft Visual Studio* ini untuk dapat menyelaraskan perangkat lunak yang akan dibangun untuk Gerbang Tol Trans Sumatera, dengan yang sudah ada agar dapat saling terintegrasi dengan baik.

Jalan Tol Trans Sumatera Bakauheni - Terbanggi Besar sepanjang 141 KM yang dikelola PT. Hutama Karya (Persero) dibangun pada tahun 2019. Terhitung sejak mulai dioperasikannya hingga akhir tahun 2020 telah terjadi kasus kecelakaan lalulintas.

Faktor penyebab kecelakaan disebabkan oleh *human error* dan juga kendaraan dengan beban muatan yang melebihi batas sehingga menyebabkan kerusakan ruas jalan. Sehingga terjadi kasus kecelakaan pada kendaraan tersebut dan juga kendaraan lain yang melintas, dan juga membuat umur jalan yang seharusnya mencapai angka efektif selama sepuluh tahun, ini menjadi berkurang. Untuk membatasi kendaraan dengan beban muatan berlebih yang mengunakan jalan tol diperlukan suatu alat yang dapat mengidentifikasi beban kendaraan yang melintas sehingga membantu pengelola jalan tol untuk dilakukan penindakan terhadap kendaraan yang melebihi batas beban muatan (*Overload*).

Dari uraian diatas masih terdapat beberapa kekurangan dari alat timbangan dinamis yang digunakan, salah satunya kurang beberapa *item* guna untuk melakukan *monitoring* dan juga memberikan data beban kendaraan untuk dilakukan penindakan apakah kendaraan tersebut *Overload* atau tidak, maka dari itu diperlukan adanya tambahan perangkat lunak pendukung untuk melakukan *monitoring* data kendaraan secara *realtime* untuk melengkapi kekurangan dari alat timbangan yang digunakan.

2. LANDASAN TEORI

2.1 Perangkat Lunak

Perangkat Lunak/*software* merupakan sekumpulan perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer agar melaksanakan sesuatu, (Ladjamudin, 2005).

2.2 Aplikasi

Aplikasi merupakan sekelompok atribut yang terdiri dari beberapa *form*, *report* yang disusun sedemikian rupa sehingga dapat mengakses data. Aplikasi merupakan program yang berisi perintah untuk melakukan olah data, secara umum aplikasi adalah suatu proses dari cara manual yang dipindahkan ke dalam komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal, (Rahman dan Santoso, 2015).

2.3 Desktop Application

Aplikasi desktop adalah aplikasi yang berdiri sendiri yang diinstal pada komputer desktop atau laptop. Ini bisa berfitur lengkap seperti Microsoft Excel atau melakukan satu atau dua fungsi seperti aplikasi kalender. Biasanya, aplikasi desktop dibatasi oleh perangkat keras yang dijalankannya. Dikombinasikan dengan aplikasi terminal mainframe, mereka memiliki antarmuka pengguna yang tidak kompleks. Mereka juga sulit untuk diperbarui, terutama jika peningkatan perangkat keras diperlukan agar aplikasi berfungsi. (Arif Fahrudin, 2020).

Aplikasi berbasis *desktop* merupakan aplikasi yang dijalankan pada masing-masing komputer atau klien. Aplikasi berbasis *desktop* harus di*install* terlebih dahulu ke dalam komputer agar dapat digunakan. Berdasarkan pengertian diatas penulis menyimpulkan bahwa aplikasi *desktop* adalah aplikasi yang berjalan pada komputer yang dapat digunakan secara langsung ketika kode program selesai dikompilasi. (Rafyrp101, 2013).

2.4 Basis Data (Database)

Basis Data (*Database*) adalah tempat penyimpanan data dan informasi secara terstruktur dan teratur, yang digunakan sebagai pangkalan data atau tempat berkumpulnya data secara digital. Pada *database* terdapat sebuah

sistem yang disebut *DBMS* (*Database Management System*), *DBMS* adalah sebuah sistem untuk membantu aplikasi dan pengguna di dalam melakukan manajemen data dan *database*. Beberapa contoh *DBMS* yang umum digunakan di dunia komputer, antara lain adalah *MySQL, Postgre SQL, Oracle, Microsoft SQL Server, Maria DB, Mongo DB*, dan lainnya, (I Putu Agus Eka Pratama, 2018).

2.5 Pengertian Bahasa C#

C# (*C sharp*) adalah" sebuah bahasa pemrograman berbasis obyek yang didukung oleh *Microsoft .NET Framework*". *Microsoft .NET Framework* adalah perantara agar aplikasi dengan bahasa pemrograman yang didukung dapat berkomunikasi dengan sistem operasi yang digunakan oleh komputer kebanyakan orang. Selain itu, *.NET Framework* juga memungkinkan *C#* untuk berkomunikasi dengan bahasa pemrograman lainnya yang juga didukung oleh *.NET Framework* seperti *VB .NET, F#,* atau *C++*", (Handoyo, 2011).

2.6 API (Application Programming Interface)

Application Programming Interface (API) adalah alat yang membuat data situs web dapat dicerna untuk komputer. Karakteristik yang membuat situs *web* optimal bagi manusia, namun membuat sulit digunakan oleh komputer sehingga dibutuhkanlah sebuah API. Melalui API, komputer dapat melihat dan mengedit data, sama seperti seseorang yang dapat memuat halaman dan submit formulir. Apa yang mungkin membutuhkan waktu berjam-jam dengan manusia untuk menyelesaikannya, namun dapat memakan waktu beberapa detik dengan komputer melalui API. Ketika dua sistem terhubung melalui API, dapat dikatakan bahwa dua sistem tersebut terintegrasi. Satu sisi yaitu server dan sisi lain yaitu klien. Dalam sisi server sebenarnya yang menyediakan API. Jika dalam sisi klien, dapat mengetahui data apa yang tersedia melalui API dan dapat memanipulasinya, biasanya atas permintaan dari pengguna. *Request method* nantinya akan memberitahu *server* tindakan apa yang diinginkan pengguna untuk diambil oleh *server*, (Girish M. Rama, Avinash Kak, 2015).

Terdapat empat method yang sering digunakan pada API, yaitu:

a. *GET* : Meminta *server* untuk mengambil suatu data.
b. *POST* : Meminta *server* untuk menambah suatu data.
c. *PUT* : Meminta *server* untuk memperbaharui suatu data.
d. *DELETE* : Meminta *server* untuk menghapus suatu data.

2.7 Pengujian Kotak Hitam (Black Box Testing)

Pengujian *Black-Box Testing* (Kotak Hitam), juga disebut dengan Pengujian Prilaku. Berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian *Black-Box Testing* memungkinkan untuk membbuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian *Black-Box Testing* merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap metode lainnya, (Roger S. Pressman, 2012).

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut:

1. Uji *Interface.*
2. Uji Fungsi Menu dan Tombol.
3. Uji Struktur dan *Database.*

2.8 Metode Prototype

Prototype digunakan untuk menggali kebutuhan secara lebih cepat. Biasanya saat pembuatan *ptototype*, keterlibatan user sangat dibutuhkan. Manfaat utama *prototype* adalah untuk mengurangi resiko tidak diterimanya hasil pengembangan suatu perangkat lunak serta pengulangan kerja di kemudian hari, (Munawar, 2018).

2.9 Unified Modelling Language (UML)

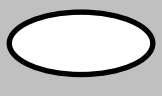
UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sagat handal di dunia pengembangan *system* yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Medeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan interatif, yaitu : identifikasi kelas-kelas dan objek-objek, identifikasi semantic dari hubungan objek dan kelas tesebut, perincian *interface* dan implementasi. UML dibangun atas model 4+1 *view*. Yaitu *LogicalView, Development View, Process View, Physical View* dan *Scenario*. Model ini di dasarkan pada fakta bahwa struktur sebuah system dideskripsikan dalam 5 *View*, yang salah satunya adalah *Scenario*, (Munawar, 2018).

2.9.1 Use Case Diagram

Use Case Diagram adalah deskripsi fungsi sebuah *system* dari perspektif pengguna. *Use Case Diagram* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan *system* disebut sebagai *scenario*. Setiap *scenario* menggambarkan urutan kejadian. Setiap urutan di inisialisasi oleh orang, *system* yang lain, perangkat keras atau urutan waktu. Dengan demikian, secara singkat bias dikatakan *Use Case Diagram* adalah serangkaian *scenario* yang digabungkan bersaa-sama oleh tujuan umum pengguna. *Use Case* dibuat berdasarkan kebutuhan Aktor. *Use Case Diagram* harus merupakan "apa" yang dikerjakan *software* aplikasi, bukan "bagaimana" *software* aplikasi mengerjakannya, (Munawar, 2018).

Tabel 1 pada halaman berikut ini adalah Simbol-simbol yang digunakan dalam *Use Case Diagram*:

Tabel 1. *Use Case Diagram*

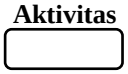
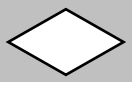
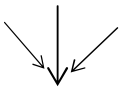
Simbol	Keterangan
Nama Use Case 	Fungsionalitas yang disediakan system sebagai unit yang saling bertukar pesan antara unit atau factor, biasanya dinyatakan dengan menggunakan kata kerja awal frase nama Use Case
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan system yang akan dibuat di luar system yang akan dibuat itu sendiri. Jadi, walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi	Komunikasi antara aktor dan Use Case, atau Use Case dan Aktor
Generalisasi	Hubungan Generalisasi dan Spesialisasi (Umum - Khusus) antara dua buah Use Case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Ekstensi / Extend	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambahkan dapat berdiri sendiri walaupun tanpa Use Case tambahan.

Include	Relasi Use Case tambahan ke sebuah Use Case, dimana Use Case yang ditambahkan memerlukan Use Case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan Use Case ini.

2.9.2 Activity Diagram

Munawar (2018: 127) mengurakan bahwa *Activity Diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari Sistem. Logika Prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah di deskripsikan dalam *Activity Diagram. Activity Diagram* mempunyai peran seperti halnya *Flowchart*, akan tetapi perbedaannya dengan *Flowchart* adalah *Activity Diagram* bisa mendukung perilaku paralel sedangkan *Flowcharti* tidak bisa.

Tabel 2. *Use Case Diagram*

Simbol	Keterangan
Status Awal 	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
Aktivitas 	Aktivitas yang diilakujan suatu sistem.
Percabangan / *Decision* 	Symbol ini digunakan jika ada pilihan aktivitas lebih dari satu atau menggabungkan dua kegiatan pararel menjadi satu.
Penggabungan / *Join* 	Penggabungan / *Join* digunakan untuk menunjukkan adanya kegiatan yang digabungkan.
Status Akhir 	Status Akhir, akhir dari aktifitas sebuah sistem.

3. METODOLOGI

3.1 Metode Pengembangan Perangkat Lunak

Pada tahapan membangun perangkat lunak, penelitian ini dilakukan dengan menggunakan metode yang telah dipilih, yaitu Metode *Prototype*. Dikarenakan dalam konteks ini sebelum melanjutkan ke tahap perilisan perangkat lunak yang sudah siap digunakan secara penuh, diperlukan adanya suatu *testing* perangkat lunak terlebih dahulu guna peninjauan sebelum masuk ke tahap akhir. Proses ini juga dilakukan secara intensif untuk menspesifikasi kebutuhan perangkat lunak agar dapat memenuhi keinginan *User* (Pengguna).

3.1.1 Komunikasi

Dalam tahapan komunikasi dari model metode *prototype* guna untuk mengidentifikasi permasalahan-permasalahan yang ada, dan juga informasi-informasi lain yang akan diperlukan dalam membangun perangkat lunak.

Pengumpulan data yang digunakan dalam menyusun serta melengkapi data adalah dengan cara observasi, wawancara dan studi pustaka, antaranya:

a. Observasi

Pengamatan langsung untuk memperoleh data yang dilakukan pada tempat penelitian yang terkait dengan penelitian dilakukan pada lajur yang terdapat timbangan dinamis dijalan tol milik PT. Hutama Karya (Persero).

b. Wawancara

Wawancara dilakukan dengan cara berkomunikasi langsung dengan pihak yang bertanggung jawab dibagian timbangan dinamis ini baik dari pihak pengelola tol maupun dari pihak penyedia timbangan dinamis.

c. Studi Pustaka

Studi pustaka dilakukan untuk memperoleh data dan informasi dengan membaca berbagai bahan penulisan, karya ilmiah serta sumber-sumber lain mengenai permasalahan yang berhubungan dengan penulisan.

3.1.2 Perencanaan Secara Tepat

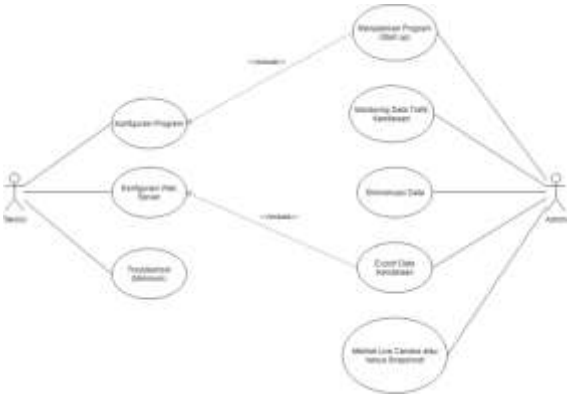
Tahapan ini dikerjakan dengan kegiatan penentuan sumber daya, spesifikasi untuk membangun berdasarkan kebutuhan sistem, dan tujuan berdasarkan pada hasil komunikasi yang dilakukan agar pembangunan perangkat sesuai dengan yang diharapkan pengguna.

3.1.3 Permodelan Perancangan Secara Tepat

Tahapan selanjutnya ialah representasi atau menggambarkan model sistem yang akan dikembangkan seperti proses dengan perancangan menggunakan *Unified Modeling Language* (UML) yang menerapkan *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Dalam tahap ini, *Prototype* yang dibangun dengan sistem rancangan sementara kemudian dievaluasi terhadap klien apakah sudah sesuai dengan yang diinginkan atau masih perlu dilakukan untuk di evaluasi kembali. Setelah perangkat lunak dianggap sesuai dengan apa yang diharapkan klien, langkah berikutnya yaitu masuk ke tahap pembuatan perangkat lunak / perangkat lunak dari rancangan sistem yang dibuat diterjemahkan ke dalam bahasa pemrograman *C#* yang diintegrasikan dengan pengguna basis data *MySQL*.

a. *Use Case Diagram*

Use Case Diagram digunakan untuk pemodelan kegiatan pada sistem yang akan dibuat. Gambar 1 berikut ini adalah rancangan *Use Case Diagram* yang akan dibuat:



Gambar 1. *Use Case Diagram Monitoring Software*

Berdasarkan *Use Case Diagram* pada gambar 1 dapat dijelaskan fungsi masing-masing dari *User Use Case* sebagai berikut:

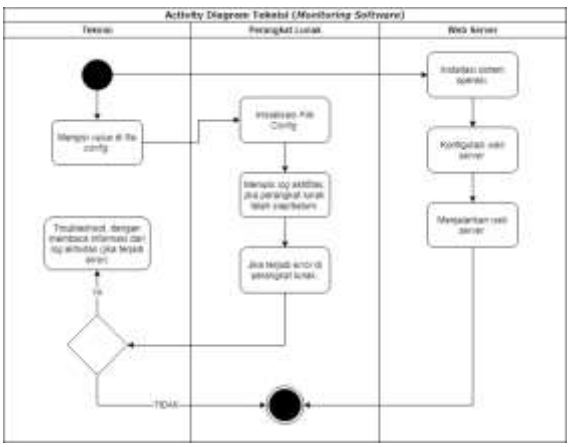
1. Teknisi melakukan konfigurasi awal perangkat lunak.
2. Teknisi menyiapkan sebuah *web server* untuk digunakan di perangkat lunak.
3. Teknisi melihat *log* yang di tulis oleh perangkat lunak, dan digunakan untuk melakukan *troubleshooting*.
4. Admin yang menjalankan perangkat lunak
5. Admin dapat melihat data kendaraan yang melintas di perangkat lunak *monitoring*.
6. Proses ketika terjadi ketidak sinkronan perangkat lunak *monitoring* yang disinkronisasi manual oleh Admin.
7. Admin dapat melakukan *export data* kendaraan yang telah melintasi lajur timbangan.
8. Admin melihat proses *media* dari kamera yang ingin ditampilkan di perangkat lunak *monitoring*.

b. *Activity Diagram*

Digunakan untuk menggambarkan alur dari awal sebuah sistem, melakukan, dan mengakhiri proses. *Activity Diagram* yang akan diterapkan pada sistem ini adalah sebagai berikut:

1. Activity Diagram Teknisi

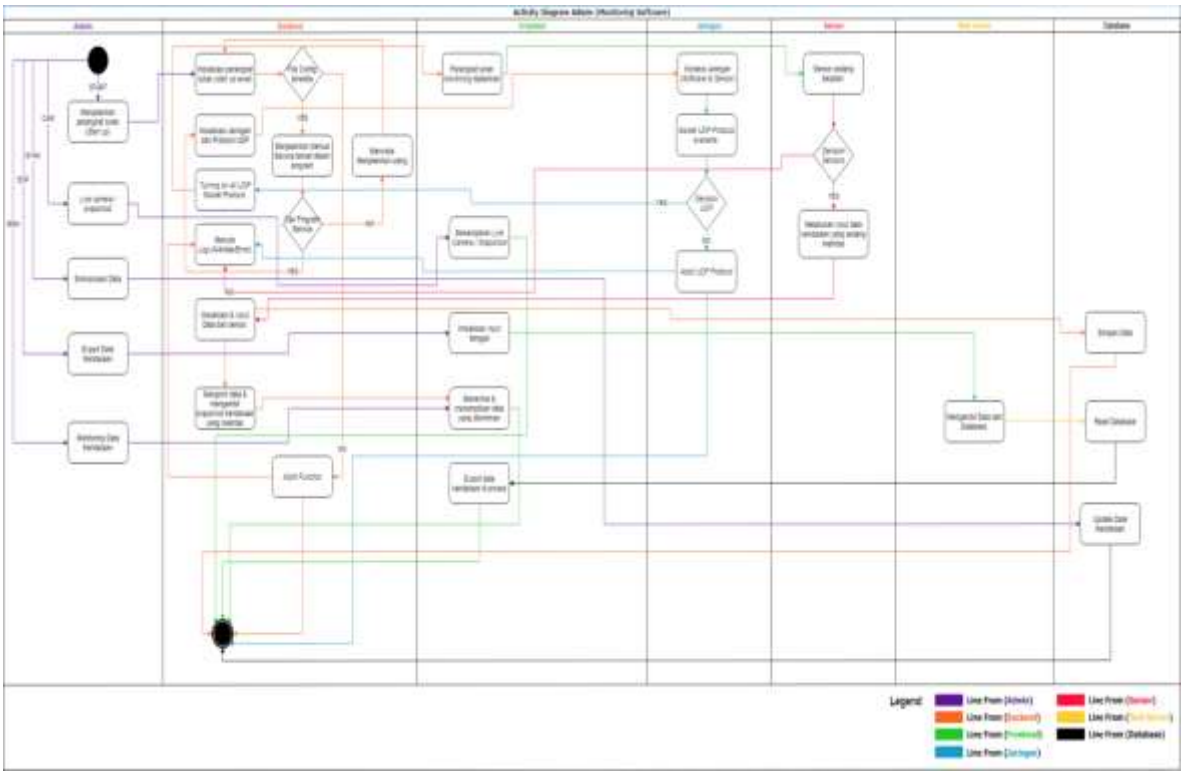
Activity Diagram Teknisi merupakan urutan langkah-langkah yang dilakukan Teknisi dalam melakukan tugasnya. Mulai dari melakukan konfigurasi perangkat lunak, konfigurasi *Web Server*, dan *Troubleshooting*. *Activity Diagram* Teknisi. Gambar 2 pada halaman berikut adalah *Activity Diagram* Teknisi dari perangkat lunak *Monitoring* yang akan dibuat:



Gambar 2. *Activity Diagram* Teknisi *Monitoring Software*

2. *Activity Diagram* Admin

Activity Diagram Admin merupakan urutan langkah-langkah yang dilakukan Teknisi dalam melakukan tugasnya. Mulai dari menjalankan perangkat lunak, *monitoring* data trafik kendaraan, sinkronisasi data, *export* data kendaraan, dan melihat *Live Cam* atau *Snapshoot*. *Activity Diagram* Admin. Gambar 3 pada halaman berikut adalah *Activity Diagram* Admin dari perangkat lunak *Monitoring* yang akan dibuat:



Gambar 3. *Activity Diagram Admin Monitoring Software*

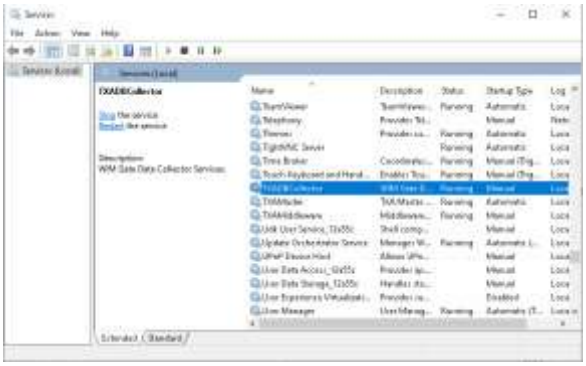
4. HASIL PEMBAHASAN

4.1 Hasil Penelitian

Hasil Penelitian ini dijelaskan mengenai Hasil dan Implementasi progam dari berbagai tahapan yang telah dirancang sebelumnya. Perlu diperhatikan sebelum Admin mulai menjalankan perangkat lunak untuk melakukan *monitoring*, pastikan perangkat (*personal computer*) memiliki sistem operasi *Windows 10 Pro*, dan terknoneksi dengan jaringan , dan juga perangkat lunak *Backend* telah di konfigurasi oleh seorang Teknisi. Setelah poin-poin yang disebutkan sudah tidak ada masalah, maka seorang Admin dapat menjalankan perangkat lunak *monitoring* tersebut.

4.1.1 Perangkat Lunak Backend

Perangkat lunak *Backend* dalam penelitian ini berbentuk *Windows Service* yang berjalan di sistem operasi *Windows 10 Pro*. Gambar 4 berikut ini merupakan perangkat lunak *backend* atau bisa disebut sebagai otak dari perangkat lunak *front end*.



Gambar 4. Perangkat Lunak *Backend*

4.1.2 Tampilan Frontend Utama

Tampilan utama merupakan yang akan selalu dilihat oleh Admin ketika perangkat lunak *monitoring* telah dijalankan dan sedang bekerja. Gambar 5. berikut ini merupakan Tampilan Utama perangkat lunak *monitoring*:

Gambar 5. Tampilan *Frontend* Utama *Monitoring*

4.2 Pembahasan

Proses pengujian merupakan tahap akhir dimana sistem akan diuji kemampuan dan keefektifannya. Pengujian Perangkat lunak dilakukan dengan menggunakan metode *Black-Box Testing*, yang merupakan salah satu cara pengujian perangkat lunak yang mengutamakan pengujian terhadap fungsi dari suatu program dan melibatkan pengguna sebagai alat ukur sebuah perangkat lunak. Dalam tahap pengujian perangkat lunak ini dilakakukan pada perangkat yang terdapat program *monitoring* dan sudah terkonfigurasi.

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan, maka dapat disimpulkan bahwa Penulis telah berhasil membangun perangkat lunak "Rancang Bangun Perangkat Lunak *Monitoring* Menggunakan Sensor Timbangan Dinamis Terhadap Muatan Kendaraan dan Penindakan Pada Gerbang Tol" sebuah perangkat lunak yang dapat melakukan proses *monitoring* data kendaraan secara *realtime*. Perangkat lunak dibangun menggunakan *IDE Microsoft Visual Studio 2019* dengan

bahasa pemrograman *Backend* maupun *Frontend* yang digunakan adalah C# (C Sharp), dan *MySQL* sebagai DBMS-nya serta menggunakan metode pengembangan perangkat lunak *Prototype*. Perangkat lunak yang dibangun dapat melakukan *monitoring* terhadap kendaraan yang melintas dan tentunya masih banyak kendaraan dengan muatan berlebih untuk dilakukan penindakan pada kendaraan dengan muatan berlebih (*Overload*) yang sering membuat jalan rusak, sehingga pengelola jalan tol terlalu sering melakukan perbaikan jalan dengan *cost* yang tidak sedikit.

5.2 Saran

Saran yang dapat diberikan sebagai perbaikan untuk penelitian yang lebih lanjut adalah sebagai berikut:

1. Penelitian selanjutnya dapat menemukan penyebab *error* yang belum diketahui sehingga dapat melakukan perbaikan – perbaikan di beberapa *error* yang terkadang masih terjadi, sehingga perangkat lunak dapat berjalan secara *full automatic*.
2. Optimalisasi *Synchronizing Data* agar dapat selalu sesuai dengan penulisan *Marking* pada gambar kendaraan (*Snapshoot*) sehingga mencapai 0 persen tingkat *error* dalam penulisan data.
3. Diharapkan penelitian selanjutnya dapat menerapkan perangkat lunak ini dengan *DBMS NoSQL* atau melakukan optimasi *database* agar lebih cepat dalam melakukan pemrosesan data.

DAFTAR PUSTAKA

Al-Bahra Bin Ladjamudin., 2005, Analisis dan Desain Sistem Informasi, Graha Ilmu, Yogyakarta

Al-Bahra Bin Ladjamudin., 2006, Rekayasa Perangkat Lunak, Graha Ilmu, Yogyakarta

Fakhrudin, Arif., 2020. Aplikasi Desktop Aplikasi Desktop (javatekno.co.id) dikutip pada 4 Juni 2020

I Putu Agus Eka Pratama, S.T., M.T., 20018. Handbook of Datawarehouse, Practice & Tourism base on Open Source, Informatika.

Handoyo, E. D., & Risal, L., 2011. Pemrograman Berorientasi Objek C#. Bandung: Penerbit Informatika.

Munawar., 2018. Analisis Perancangan Sistem Berorientasi Objek dengan Unified Modeling Language (UML). Depok: Informatika.

Roger S. Pressman., 2012. Buku 1 Rekayasa Perangkat Lunak Pendekatan Praktisi Edisi 7. Buku Rekayasa Perangkat Lunak Edisi 7 halaman 50 Tentang Metode Prototype.

Rama, Girish Maskeri dan Avinash Kak., 2013. Software – Practice and Experience. New Jersey: Wiley Online Library.

Rahman, F. and Santoso., 2015. Aplikasi Pemesanan Undangan Online, Jurnal Sanis dan Informatika, 1(2), pp. 78–87.