

IMPLEMENTASI FRAMEWORK GTK DALAM PEMBUATAN APLIKASI DESKTOP MONITORING MESIN PRODUKSI



PUBLIKASI ILMIAH

**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada
Program Studi Teknik Informatika Fakultas Komunikasi dan Informatika**

Oleh:

SHANDIKA GALIH NUR AMIN

L200190078

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2023**

HALAMAN PERSETUJUAN

**IMPLEMENTASI FRAMEWORK GTK DALAM
PEMBUATAN APLIKASI DESKTOP MONITORING MESIN
PRODUKSI**

PUBLIKASI ILMIAH

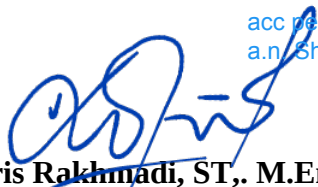
oleh:

SHANDIKA GALIH NUR AMIN

L200070150

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



acc pendadaran
a.n Shandika Galih

Aris Rakhmadi, ST., M.Eng

NIK.983

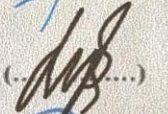
HALAMAN PENGESAHAN
IMPLEMENTASI FRAMEWORK GTK DALAM
PEMBUATAN APLIKASI DESKTOP MONITORING MESIN
PRODUKSI

OLEH
SHANDIKA GALIH NUR AMIN
L200190078

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Selasa, 13 Februari 2023
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Aris Rakhmadi, S.T., M.Eng
(Ketua Dewan Penguji)
2. Dimas Aryo Anggoro, S.Kom., M.Sc.
(Anggota I Dewan Penguji)
3. Diah Priyawati, S.T., M.Eng
(Anggota II Dewan Penguji)


(.....)

(.....)

(.....)



Dekan
Fakultas Komunikasi dan Informatika

Murniyatna S.T. M.Sc. Ph.D
NIDN 0612076901



Ketua
Program Studi Informatika

Jedi Gunawan, S.T., M.Sc, Ph.D
NIDN. 0602048602

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 27 Januari 2022



SHANDIKA GALIH NUR AMIN

L200190078

IMPLEMENTASI FRAMEWORK GTK DALAM PEMBUATAN APLIKASI DESKTOP MONITORING MESIN PRODUKSI

Abstrak

Aplikasi berbasis desktop saat ini sangat populer, karena memiliki beberapa kelebihan yaitu aplikasi desktop dapat diakses secara offline, aplikasi desktop memiliki respon yang cepat, aplikasi desktop kaya akan user experience dan sekarang juga banyak dikembangkan demi meningkatkan optimalisasi kinerja dari mesin yang terintegrasi dengan alat industri seperti alat yang dapat mengotomatisasi proses produksi PT. Stechoq Robotika Indonesia adalah salah satu industrial manufacturing yang memiliki produk utama berupa Digital Control System. Dalam prosesnya, aplikasi yang dibangun dengan desktop dan framework electron ini berfungsi untuk memonitoring mesin produknya. Dari hasil observasi yang sudah dilakukan penulis, terdapat kondisi yang perlu diadakan peningkatan yaitu berupa pergantian framework electron yang dirasa saat pembangunan aplikasi desktopnya memerlukan resource yang besar. Hasil penelitian yang telah dilakukan, bahwa sistem monitoring yang mengimplementasikan framework gtk sebagai kerangka kerja dari aplikasi desktop monitoring lebih efektif dan efisien karena memiliki size lebih kecil yaitu hanya sebesar 394,7 kb dan penggunaan cpu yang lebih sedikit hanya 0.78% sedang pada aplikasi yang dibangun menggunakan framework electron memiliki size yang lebih besar dengan 232,2 mb dan memakan cpu sebesar 1.20 %.

Kata Kunci: Desktop, Framework Gtk, Monitoring

Abstract

Desktop-based applications are currently very popular due to several advantages such as being able to access the application offline, having a fast response time, rich user experience, and now they are also developed to improve performance optimization of machines integrated with industrial tools such as automated production process machinery. PT. Stechoq Robotika Indonesia is one of the industrial manufacturing companies that has a main product of Digital Control System. In the process, the application built with desktop and electron framework serves to monitor its product machines. From the observations that have been made by the author, there is a condition that needs improvement, which is the replacement of the electron framework which is considered to require a large amount of resources during desktop application development. Based on the research that has been conducted, the monitoring system that implements the gtk framework as the framework for the desktop monitoring application is more effective and efficient because it has a smaller size of 394.7 kb and uses less CPU at only 0.78%, while the application built using the electron framework has a larger size of 232.2 mb and uses CPU at 1.20%.

Keywords: Desktop, Gtk Framework, Monitoring

1. PENDAHULUAN

Teknologi yang berkembang saat ini sangatlah pesat, perkembangan otomatisasi dalam bidang industri yang berfungsi dalam mempermudah pekerjaan manusia atau juga sebagai pengingat saat kesalahan terjadi. Banyak proses monitoring di industri masih dilakukan secara manual informasi yang diperoleh secara manual tentu tidak reliabel karena sering terjadi kesalahan pada operator, yang tentu bisa berdampak pada sistem produksi dan bisa merugikan perusahaan (Maulana & Pancono, 2021). Dalam usaha untuk meningkatkan efisiensi dan efektifitas proses industri, banyak perusahaan yang mempertimbangkan untuk menggunakan sistem otomatisasi dalam proses produksi yang bertujuan untuk mengurangi kesalahan dalam melakukan proses produksi dengan sistem yang terintegrasi. Karena kebutuhan proses otomatisasi maka industri manufaktur yang memproduksi peralatan kontrol otomasi saat ini berkembang dengan pesat dan banyak sekali produk yang dikeluarkan oleh perusahaan internasional untuk memenuhi kebutuhan penggunaan sistem otomasi pada perusahaan perusahaan di Indonesia (Ban et al., n.d.).

Aplikasi berbasis desktop saat ini sangat populer karena memiliki beberapa kelebihan yaitu aplikasi desktop dapat diakses secara *offline* (Robiatul Adwiya, 2020). Aplikasi desktop memiliki respon yang cepat, kaya akan *user experience* dan sekarang banyak dikembangkan demi meningkatkan optimalisasi kinerja dari mesin yang terintegrasi dengan alat industri seperti alat yang dapat mengotomatisasi proses produksi, jadi perlu sebuah aplikasi desktop yang dapat memonitoring alat produksi yang berjalan otomatis itu (Bustamin, 2021).

PT. Stechoq Robotika Indonesia adalah salah satu industrial manufaktur yang memiliki produk utama berupa *Digital Control System*. Dalam prosesnya, aplikasi yang dibangun menggunakan desktop framework electron ini memiliki size yang besar. Aplikasi ini berfungsi untuk memonitoring mesin produksi, adapun fitur dari aplikasinya memuat *list order* untuk memilih order yang akan di

distribusi, *confirm* produk untuk mulai memproduksi, Istirahat untuk waktu masuk ke istirahat maksimal 30 menit jika lebih masuk ke downtime dan fitur lainnya yang memonitoring mesin produksi.

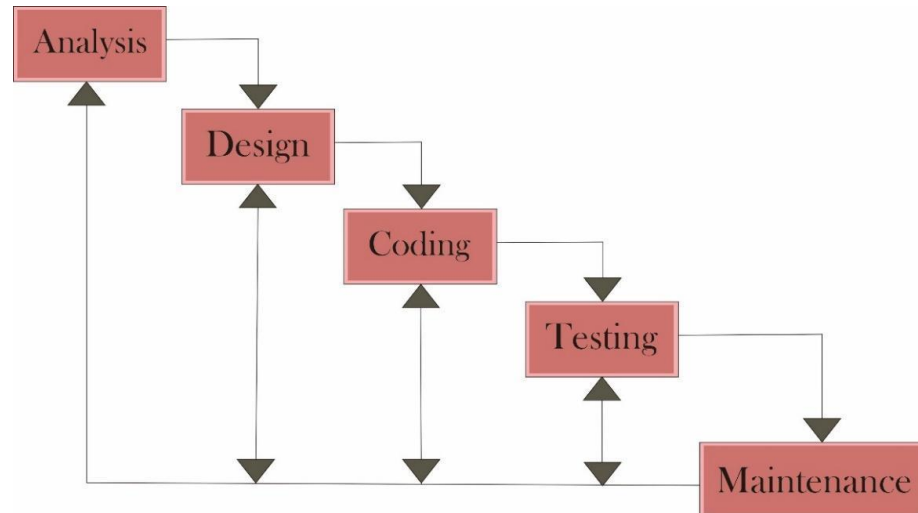
Dari hasil observasi yang sudah dilakukan, terdapat kondisi yang perlu diadakan peningkatan yaitu berupa pergantian *framework electron*, sebab saat proses pembangunan aplikasi desktop memerlukan *resource* yang besar karena menggabungkan file *Chromium* dan *Node.js* menjadi satu runtime saat pembangunan aplikasinya (Hołowiński & Pańczyk, 2020) yang membuat aplikasi memiliki size yang besar dan memakan banyak cpu saat dijalankan. Untuk itu perlu mengganti *framework electron* ke *framework GTK* agar efisiensi saat pembangunan sistem dapat memperingan, mempercepat pekerjaan dalam memonitor dan memperoleh waktu yang efisien (Sasikumar, 2008). Karena GTK merupakan kerangka kerja pemrograman berbasis GUI (Hoemig, n.d.) yang digunakan untuk membangun aplikasi desktop dan memiliki keunggulan tersendiri yang dimana *resource* yang digunakan lebih ringan (Porras.J, 2005). Dengan adanya penelitian ini maka diharapkan dapat mengimplementasikan *framework gtk* dalam pembuatan aplikasi monitoring mesin produksi yang bisa mempermudah dan optimalisasi pekerjaan.

2. METODE

GTK (GIMP Toolkit) adalah sebuah framework yang populer digunakan dalam pengembangan aplikasi desktop di sistem operasi Linux. Framework ini sering digunakan untuk membuat aplikasi berbasis GUI (Graphical User Interface) yang bertujuan untuk mempermudah proses data collection dan data analysis.

Pengembangan aplikasi desktop ini menggunakan metode *SDLC Waterfall* (*Software Development Life Cycle Waterfall*). Tujuan dari peneliti menggunakan metode ini agar penelitian berjalan dengan terarah dan teratur juga dengan metode ini dapat mengurangi terjadinya kesalahan dalam pengembangan sistem yang dibuat penulis. Model air terjun (*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan dimulai dari analisis,

desain, pengkodean, pengujian dan tahap pemeliharaan software (*maintenance*). Berikut adalah bagan *SDLC Waterfall* yang digunakan (Balaji, 2012).



Gambar 1. SDLC Waterfall

Dalam meneliti proses SDLC Waterfall pada gambar 1, saya melakukan metode observasi untuk memahami bagaimana setiap tahap dalam proses ini berlangsung, mulai dari perencanaan, analisis, desain, coding, pengujian, dan pemeliharaan. Mulai dari mengamati bagaimana proyek ini melaksanakan tugas-tugas, mengkomunikasikan dengan satu sama lain, dan memastikan bahwa setiap tahap selesai sebelum memulai tahap berikutnya.

2.1 Analysis

Analisis kebutuhan sangat penting untuk menentukan fitur apa saja yang akan kita masukan dalam sebuah sistem aplikasi dalam menentukan kebutuhan sistem.

A. Kebutuhan Fungsional

Kebutuhan Fungsional merupakan jenis kebutuhan yang berisikan proses- proses yang diberikan oleh sistem kepada pengguna saat mengakses aplikasi ini.

1. Halaman *Login*, pada halaman ini user dapat masuk sebagai operator yang akan melakukan monitoring pada mesin.

2. Halaman *Home*, menu ini digunakan oleh operator untuk menentukan dan memilih order barang yang akan diproduksi oleh mesin.
 3. Halaman Konfirmasi, pada menu ini banyak sekali fitur seperti shift time untuk memonitoring kerja operator, *running timer* untuk memonitoring kerja mesin, *downtime* untuk mesin yang memiliki keadaan *down*.
 4. Halaman Dandori, fitur ini digunakan untuk memonitoring lama waktu operator dalam melakukan proses persiapan pada mesin sebelum mesin dioperasikan.
 5. *SCW (Stop-Call-Wait)*, fitur ini digunakan untuk memberikan informasi adanya *downtime* pada mesin kepada tim maintenance dan manager produksi.
 6. Istirahat, merupakan fitur untuk memonitoring waktu istirahat operator.
- B. Kebutuhan Non Fungsional

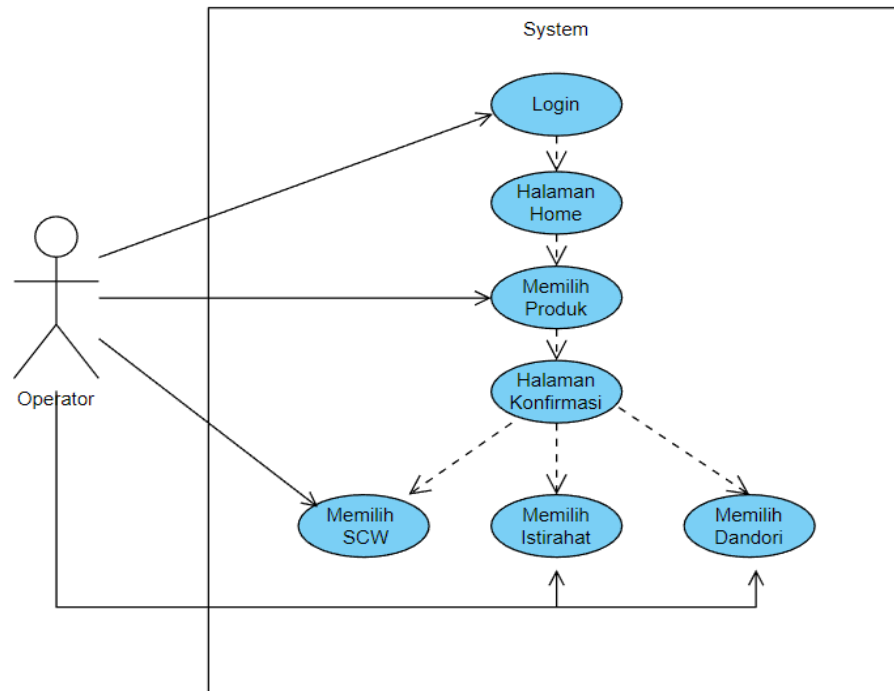
Untuk dapat mengakses aplikasi desktop ini memerlukan sistem operasi linux, karena dibangun menggunakan framework GTK yang belum mensupport windows.

2.2 Design

Proses ini menganalisa bentuk dan desain dari aplikasi yang akan dibuat (Khafid & Putri, 2020) melalui kebutuhan-kebutuhan yang dihasilkan dari proses analisis sistem dan *software* menjadi representasi ke dalam bentuk prototype atau tampilan *software* sebelum nantinya masuk ke tahap pemrograman atau *coding*. *Software* yang digunakan dalam membuat prototype dan desain layout aplikasi adalah *Figma* dan *app diagram*.

2.2.1 Use case Diagram

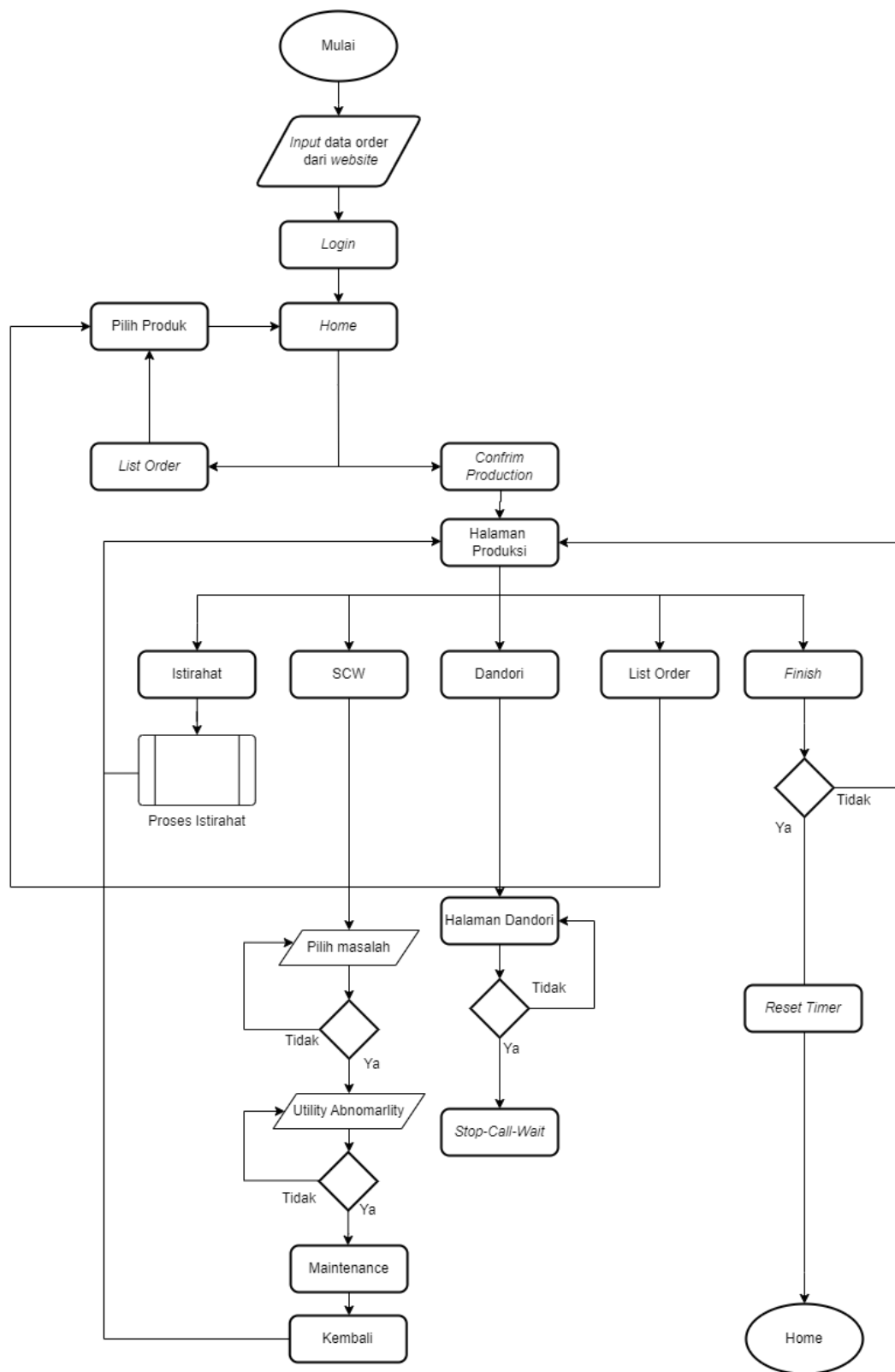
Dalam use case ini, operator dapat memilih produk yang akan diproduksi ke sistem aplikasi. Proses ini melibatkan penginputan informasi produk seperti nama produk, deskripsi, dan gambar yang dapat dilihat pada gambar 2.



Gambar 2. Use Case Diagram

2.2.2 Flow Chart System

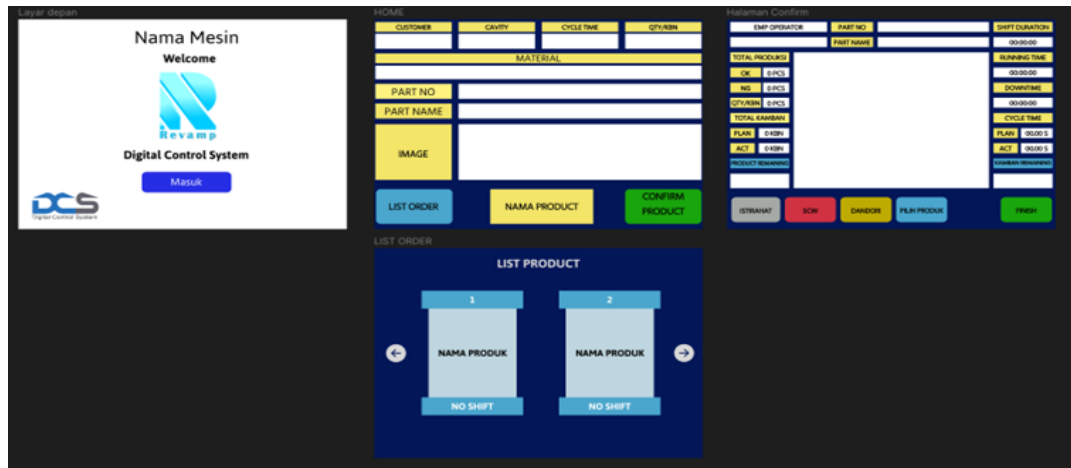
Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Untuk *flowchart* programnya bisa di lihat pada gambar 3, sebelum operator memulai aplikasi monitoring ini operator terlebih dahulu *order* produk melalui *website* yang berisi mengenai komponen yang akan ditampilkan pada halaman *home* pada aplikasi desktop yang dibuat, lalu operator dapat mengganti produk di dalam aplikasi monitoring dan halaman konfirmasi operator dapat memilih fitur seperti istirahat, dandori dan scw serta dapat menyelesaikan shift dengan menekan finish untuk mereset timer kembali ke 0.



Gambar 3. Flowchart System

2.2.2 Halaman depan

Halaman depan *user* dapat masuk sebagai operator yang akan menjalankan aplikasi mesin monitoring produksi.



Gambar 4. Desain Halaman

2.2.3 Halaman Home

Halaman *home* ini memiliki informasi seperti nama customer, *cavity cycle time*, *qyt/kbn* material yang dipakai nomor dan nama part serta gambar mesin yang akan dicetak.

2.2.4 Halaman List Order

Halaman *list order* ini adalah halaman untuk list produk yang akan dicetak, jadi operator akan memilih jenis produk yang akan dicetak terlebih dahulu.

2.2.5 Halaman Confirm

Halaman confirm ini memiliki banyak fitur yang ditampilkan seperti *shift time* yang merupakan fitur untuk memonitoring waktu kerja operator, *shift time* akan terus berjalan meskipun masuk istirahat, *SCW*, dan mesin mengalami masalah. *Running Time*, merupakan fitur yang memonitoring waktu mesin bekerja dan akan berhenti apabila memasuki waktu istirahat, mesin mengalami *down* ataupun dandori sedang dilakukan. *Downtime*, merupakan fitur untuk memonitoring waktu mesin mengalami *down* atau masuk dalam keadaan SCW, Fitur ini digunakan untuk mengetahui berapa lama mesin perbaiki oleh tim *maintenance*.

2.2.6 Halaman Istirahat

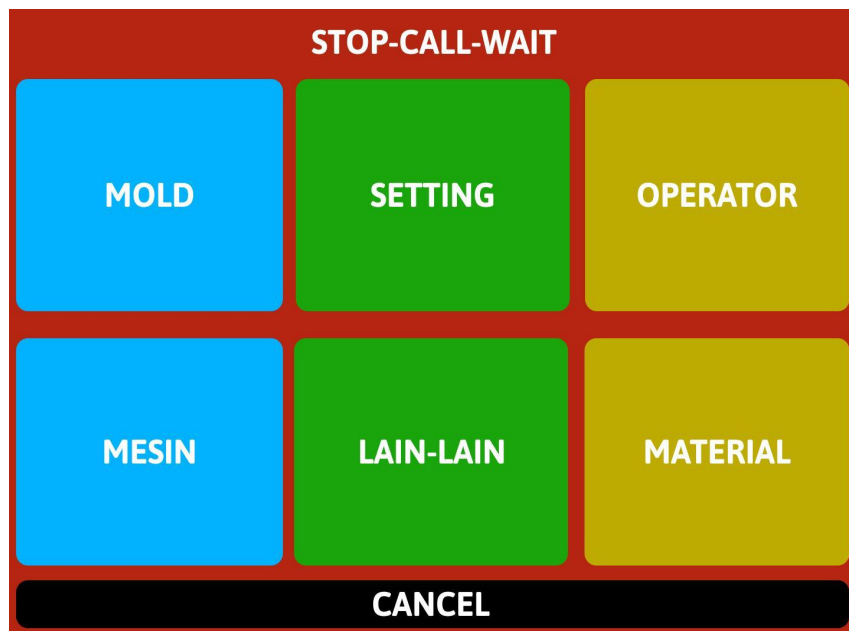
Halaman istirahat, merupakan fitur untuk memonitoring waktu istirahat operator. Waktu maksimal istirahat operator adalah 30 menit. Apabila lebih dari 30 menit, maka waktu itu akan terhitung ke dalam *downtime*.



Gambar 5. Desain Halaman Istirahat

2.2.7 Halaman SCW (Stop Call Wait)

SCW (Stop-Call-Wait), fitur ini digunakan untuk memberikan informasi adanya *downtime* pada mesin kepada tim *maintenance* dan *manager* produksi. Selama proses ini berlangsung, operator akan melakukan perhitungan barang produksi tidak bagus yang dihasilkan oleh mesin.



Gambar 6. Desain Halaman SCW

2.2.8 Halaman Dandori

Dandori, fitur ini digunakan untuk memonitoring lama waktu operator dalam melakukan proses inisiasi dan persiapan pada mesin sebelum mesin dioperasikan. Jika waktu persiapan dandori memerlukan perbaikan maka saat dandori dimulai bisa langsung memilih scw yang ada disitu untuk memperbaiki mesin yang belum siap digunakan.



Gambar 7. Desain Halaman Dandori

2.3 Coding

Mengimplementasikan rancangan desain ke dalam bahasa pemrograman adalah tahapan yang sangat penting dalam pembuatan sebuah aplikasi desktop (Andry & Stefanus, 2020), karena aplikasi berbasis desktop menggunakan Framework gtk ini tidak bisa menjalankan semua bahasa pemrograman dan penulis memilih satu untuk pembuatan aplikasi desktop ini, Adapun program yang digunakan untuk membuat aplikasi monitoring ini adalah sebagai berikut:

- a. Framework GTK 3, sebagai kerangka kerja dalam pembangunan aplikasi desktop
- b. *Glade*, digunakan untuk merancang antarmuka pengguna (user interface) aplikasi dengan mudah.
- c. *Python*, sebagai bahasa pemrograman yang dipakai untuk membangun aplikasi.
- d. *Visual Studio Code*, aplikasi untuk merancang kodingan dari pembuatan aplikasi.

2.4 Testing

Tahap ini semua fungsi-fungsi software harus dites menggunakan metode black box yang didasarkan pada sistem yang telah dibuat agar fungsionalitas dari aplikasi dapat berjalan dengan baik. Aplikasi diuji secara menyeluruh untuk memastikan tidak adanya *bug* atau *error* pada *software* (Suparyanto dan Rosad (2015, 2020). Pada tahap ini juga dilakukan evaluasi terhadap hasil uji coba yang dilakukan. Hasil dari evaluasi tersebut digunakan untuk memperbaiki dan menyempurnakan aplikasi agar dapat berfungsi sesuai dengan yang diharapkan.

2.5 Maintenance

Kemungkinan bahwa sistem akan mengalami perubahan ketika sudah digunakan oleh user itu akan terjadi (Purnia et al., 2019), maka pemeliharaan *software* menjadi tahap yang bertujuan untuk jangka panjang. Pada tahap ini juga terdapat tahap *update* untuk penambahan fitur dan pengembangan aplikasi untuk memastikan aplikasi sesuai dan berguna agar dapat berjalan dengan baik seperti yang diharapkan sebelumnya.

3. HASIL DAN PEMBAHASAN

Dari hasil yang telah dilakukan penulis telah berhasil mengimplementasikan framework GTK ke dalam aplikasi desktop monitoring untuk produksi dengan menggunakan metode *SDLC Waterfall*.

CUSTOMER	CAVITY	CYCLE TIME	QTY/KBN
Shandika	25	10	100
MATERIAL			
PP 66 ZYTEL			
PART NO	003		
PART NAME	T-Valve Female		
IMAGE			
LIST ORDER	T-Valve Female		CONFIRM PRODUCT

Gambar 8 home dan konfirmasi menampilkan data dari website

Hasil yang didapat aplikasi ini berhasil menampilkan data dari order yang dilakukan oleh *website* dan bisa mengfungsikan *timer* untuk kondisi dimana saat operator masuk dan operator ingin membuka menu seperti dandori, istirahat dan scw maka fungsi dari *running time* dan *shift duration* akan berjalan menyesuaikan menu yang dipilih.

EMP OPERATOR	PART NO.	001	SHIFT DURATION
Shift 1	PART NAME	T-Valve Male	00:00:19.1
TOTAL PRODUKSI			RUNNING TIME
OK	0 PCS		00:00:08.6
NG	0 PCS		DOWNTIME
QTY/KBN	100 PCS		00:00:10.4
TOTAL KANBAN			CYCLE TIME
PLAN	100 KBN	PLAN	18.0 S
ACT	0 KBN	ACT	18.0 S
PRODUCT REMAINING		KANBAN REMAINING	
1 ITEMS			100 KBN
ISTIRAHAT SCW DANDORI LIST ORDER FINISH			

Gambar 9 halaman konfirmasi setelah membuka menu

Timer akan berjalan ketika memasuki halaman konfirmasi. Halaman istirahat dan dandori dibuka maka timer dari *running time* akan berhenti dan shift durasi tetap berjalan, lalu jika scw dibuka maka *running time* berhenti dan *timer downtime* akan berjalan selama halaman scw dibuka.

Pengujian aplikasi yang diuji menggunakan metode *black box* diuji menggunakan sistem operasi Ubuntu 22.04.1 LTS. Terlihat aplikasi yang telah dibangun menggunakan framework gtk ini telah berhasil menampilkan fitur yang ada pada tabel 1.

Tabel 1 Pengujian Black Box

NO	Fitur	Deskripsi	Hasil yang diinginkan	Hasil Pengujian
1	Login, Halaman Home, Halaman Konfirmasi	Menekan tombol pada layar	Halaman berpindah <i>layer</i> saat tombol ditekan	Berhasil
2	Istirahat, SCW, Dandori, List Order, Finish	Menekan tombol pada layar	Halaman berpindah <i>layer</i> saat tombol ditekan	Berhasil
3	Data dari api backend	Menampilkan data dari website order	Data akan muncul saat orderan telah diinputkan	Berhasil
4	Timer	Menekan confirm product akan memulai timer	Menampilkan timer yang dapat berjalan dan berhenti sesuai kondisi	Berhasil

Aplikasi *Digital Control System* yang dibangun menggunakan framework GTK ini memiliki fungsi untuk memonitoring sistem kerja dari proses produksi yang dilakukan oleh industri. Aplikasi ini memiliki keunggulan dimana performa dan pemakaian aplikasi ini lebih optimal dibandingkan aplikasi yang dibangun menggunakan framework electron. Terlihat pada tabel 2 perbandingan performa, cpu yang dipakai pada electron lebih besar dibanding dengan GTK lalu pada ukuran aplikasinya terlihat sangat besar yaitu 232,2 mb dan aplikasi yang dibangun menggunakan framework GTK hanya memiliki ukuran sebesar 395,4 kb.

Tabel 2 Perbandingan performa

Aplikasi Framework GTK		Aplikasi Framework Electron	
Virtual Memory	749,6 MiB	Virtual Memory	4,5 GiB
CPU	0,75 %	CPU	1.20 %
CPU Time	0:44.38	CPU Time	0:19.24
Size File	394,7 kB	Size File	232.2 MB

4. PENUTUP

Kesimpulan:

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem monitoring yang mengimplementasikan framework gtk sebagai kerangka kerja dari aplikasi desktop monitoring lebih efektif dan efisien karena memiliki *size* lebih kecil yaitu hanya sebesar 394,7 kb dan penggunaan cpu yang lebih sedikit hanya 0.78 % sedang pada aplikasi yang dibangun menggunakan framework electron memiliki *size* yang lebih besar dengan 232,2 mb dan memakan *cpu* sebesar 1.20 %.

Aplikasi desktop yang dibangun menggunakan framework gtk ini memiliki fungsionalitas dan fitur yang berjalan dengan baik. Namun aplikasi ini belum digunakan dan diimplementasikan kedalam *hardware* dari mesin produksi yang ada, sehingga perlu pengujian lebih dalam untuk dapat mengetahui efisiensi dan efektifitas dari aplikasi monitoring yang dibuat menggunakan framework gtk ini. Selain itu, dapat dikembangkan lebih lanjut dan penambahan fitur seperti *Rfid* Agar dapat mengoptimalkan aplikasi tanpa perlu melakukan login secara manual.

DAFTAR PUSTAKA

- Andry, J., & Stefanus, M. (2020). Pengembangan Aplikasi E-learning Berbasis Web Menggunakan Model Waterfall Pada SMK Strada 2 Jakarta. *Jurnal Fasikom*, 10(1), 1–10. <https://doi.org/10.37859/jf.v10i1.1878>
- Balaji, S. (2012). Waterfall vs v-model vs agile : A comparative study on Sdlc. *Waterfall Vs V-Model Vs Agile : A Comparative Study On Sdlc*, 2(1), 26–30.
- Ban, G., Berbasis, P. C. R., & Net, V. B. (n.d.). *Perancangan Aplikasi Sistem Informasi Stok Stiker di*. 8(2), 200–212.
- Bustamin, S. (2021). Aplikasi desktop multi platform untuk redis client framework Electron JS dan React JS. *Dewantara Journal of Technology*, 2(1), 21–25. <http://jurnal.atidewantara.ac.id/index.php/djtech/article/view/83>
- Hoemig, A. S. (n.d.). *Diploma Thesis Feasibility Study and Prototypical Implementation of a Window Framework for Digital Tables*.
- Hołowiński, M., & Pańczyk, B. (2020). Comparative analysis of the technology used to create multi-platform applications on the example of NW.js and Electron. *Journal of Computer Sciences Institute*, 17(November), 396–400. <https://doi.org/10.35784/jcsi.2380>
- Khafid, B., & Putri, D. A. P. (2020). Pesma Apps as Android-based Integrated Applications for Mahasantri Pesma KH Mas Mansur UMS. *Khazanah Informatika: Jurnal Ilmu Komputer Dan Informatika*, 6(2), 95–102. <https://doi.org/10.23917/khif.v6i2.10494>
- Maulana, G. G., & Pancono, S. (2021). Proses Control System Pada Mesin Stamping Berbasis Internet Of Thing. *CESS (Journal of Computer Engineering, System and Science)*, 6(1), 48. <https://doi.org/10.24114/cess.v6i1.21180>
- Purnia, D. S., Rifai, A., & Rahmatullah, S. (2019). Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android. *Seminar Nasional Sains Dan Teknologi 2019*, 1–7.
- Robiatul Adwiya. (2020). Rancang Bangun Aplikasi Penjualan Handphone Berbasis Desktop. *Jurnal Informatika Kaputama*, 4(2), 197–206.
- Sasikumar, M. (2008). Automatic Generation of Speech Interface for GUI Tools /

Applications using Accessibility Framework. *Techshare India*, 1–7.
<http://naveenk.wdfiles.com/local--files/publications/AccessibleSpeech.pdf>
Suparyanto dan Rosad (2015. (2020). 済無No Title No Title No Title. *Suparyanto*
Dan Rosad (2015, 5(3), 248–253.
Thesis, M. (2005). *GTK + toolkit on mobile Linux devices*. 358(050).