

PEMBUATAN SISTEM *MONITORING* SELURUH *LINE MACHINING* BERBASIS PLC OMRON CJ1M DAN *VISUAL BASIC* DI PT FEDERAL IZUMI MANUFACTURING

Afianto, S.T., M.T., M.Sc.

Program Studi Teknik Produksi & Proses Manufaktur, Politeknik Manufaktur Astra
Jl. Gaya Motor Raya No.8, Sunter II, Jakarta 14330, Jakarta

Email: afianto@polman.astra.ac.id

Abstrak

PT. Federal Izumi Manufacturing (PT. FIM) merupakan salah satu anak perusahaan PT. Astra Otoparts Tbk. yang bergerak dalam bidang produksi piston. Piston merupakan bagian penting dalam sebuah kendaraan. Pencatatan *downtime* menjadi hal yang sangat diperlukan bagi suatu perusahaan sebagai tolak ukur performa produksi dan acuan perencanaan produksi selanjutnya. Namun pada penerapannya, data *downtime* yang didapatkan tidak akurat dikarenakan proses pencatatan yang masih dilakukan manual. Oleh karena itu, penulis membuat sistem monitoring berbasis PLC Omron CJ1M dan *visual basic* yang dapat mencatat secara akurat data *downtime* seluruh *line machining* pada gedung A, B, dan C kedalam database. Serta dapat memantau kondisi line secara *real time*. Dengan dibuatnya sistem ini, akurasi pencatatan *downtime* meningkat sebesar 3.01% dari data *maintenance*. Selain itu, *performance rate* pada proses produksi juga dapat ditingkatkan.

MAKING OF MONITORING SYSTEM ALL MACHINING LINE USING PLC OMRON CJ1M AND VISUAL BASIC IN PT FEDERAL IZUMI MANUFACTURING

PT. Federal Izumi Manufacturing is a branch company of PT. Astra Otoparts Tbk and engaged in manufacture of piston. Piston is an important part in vehicle engine that functionate as air surpressor. Recording downtime becomes an indispensable thing for a company as a benchmark of production performance and reference production planning next. However, in practice, the downtime data obtained is not accurate recording process is still done manually. Therefore, the authors make a monitoring system based on Omron CJ1M PLC and visual basic that can accurately record the downtime data of all line machineries on buildings A, B, and C into the database. And can be in outline in real time. With data theft downtime increased by 3.01% of maintenance data. In addition, the level of performance in the production process can also be improved.

Keywords: system monitoring, visual basic, PLC, ethernet, device net

I. PENDAHULUAN

1.1 Latar Belakang

PT. Federal Izumi Manufacturing adalah Perusahaan yang berada dalam grup Astra Otopart. Bergerak di bidang pembuatan komponen kendaraan bermotor berupa piston. Piston adalah sumbat geser yang terpasang di dalam silinder mesin pembakaran. Piston yang di produksi PT. FIM adalah piston untuk kendaraan roda dua seperti Honda, Yamaha, Suzuki. Juga kendaraan roda empat berupa mobil bahan bakar bensin, bus dan truk bermesin diesel bahan bakar solar. Semua kegiatan perusahaan ini berlokasi di Cileungsi, Bogor.

Untuk meningkatkan kinerja serta performa mesin pada *line machining*, Departemen Maintenance malakukan tindakan preventive dengan membuat jadwal

Maintenance Day pada setiap *line machining*. Salah satu cara untuk menentukan *Line* yang akan di lakukan tindakan *maintenance* adalah dengan melihat jumlah *downtime* yang terjadi dan jenis masalah yang timbul.

Sistem pencatatan yang diterapkan di PT. FIM saat ini masih dilakukan secara manual oleh operator, yaitu pada saat produksi terhenti akibat masalah yang terjadi pada mesin, operator akan melihat jam dan mencatat waktu awal terjadinya *downtime*, kemudian operator akan melaporkannya kepada *leader* dan akan diteruskan ke departemen *maintenance* untuk dilakukan proses perbaikan. Setelah proses perbaikan selesai dan mesin dapat kembali beroperasi, operator akan mencatat kembali waktu akhir terjadinya *downtime*.

Dengan sistem pencatatan yang digunakan ini sangat memungkinkan waktu *downtime* yang dicatat tidak

akurat, karna didasari oleh perkiraan operator. Masalah lain yang timbul adalah ketika terjadi masalah pada mesin yang dapat diselesaikan sendiri oleh operator, sehingga operator tersebut tidak mencatat *downtime* yang terjadi karena dapat diselesaikan dalam kurun waktu kurang dari 5 menit. Jika hal ini terus berulang selama kurun waktu 1 hari saja, maka akumulasi *downtime* mesin tersebut tidak akan tercatat maksimal. Disebabkan oleh sistem pencatatan yang dilakukan tidak akurat.

Dari permasalahan yang ada, maka salah satu solusinya adalah dengan dibuat sebuah sistem *monitoring* yang dapat memantau kinerja mesin pada *Line* selama 24 jam dan melakukan pencatatan *downtime* secara akurat.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang ada, maka perumusan masalah yang dibahas adalah Bagaimana membuat Sistem Monitoring yang dapat memantau seluruh *Line Machining* selama 24 jam dan mencatat waktu *downtime* secara akurat menggunakan PLC OMRON CJ1M dan Visual Basic 6?

1.3 Pembatasan Masalah

Pada penulisan tugas akhir ini penulis akan membahas :

1. Membahas sistem kerja pencatatan *downtime*
2. Membahas *wiring* sistem monitoring
3. Membahas program PLC dan *Visual Basic*
4. Tidak membahas perhitungan arus pada sistem kontrol

1.4 Tujuan

Tujuan penelitian ini adalah:

- a. Meningkatkan akurasi pencatatan *downtime*
- b. Kalkulasi *downtime* dihitung secara otomatis

II. LANDASAN TEORI

2.1 Perangkat Lunak

2.2.1 Visual Basic

Visual BASIC (*Beginners All-Purpose Symbolic Instruction Code*) merupakan sebuah bahasa pemrograman yang dapat digunakan untuk membuat suatu aplikasi dalam Microsoft Windows. Visual BASIC menggunakan metode *Graphical User Interface* (GUI) dalam pembuatan program aplikasi (*project*). Istilah visual mengacu pada metode pembuatan tampilan program (Interface) atau objek pemrograman yang biasa dilakukan secara langsung terlihat oleh programmer. Dalam Visual BASIC, pembuatan program aplikasi harus dikerjakan dalam sebuah *project*. Sebuah *Project* dapat terdiri dari File *Project* (.vbp), File Form (.frm), File data binary (.frx), Modul Class (.cls), Modul Standar (.bas), dan file resource tunggal (.res). Bahasa yang digunakan adalah bahasa BASIC yang sangat populer pada era sistem operasi DOS.

- a) Antar Muka *Visual basic*

Interfaces atau antar muka *visual basic* berisi menu, *toolbar*, *toolbox*, *properties*, jendela *form*, jendela *project*, jendela *lay-out*.

Pembuatan program aplikasi menggunakan *Visual basic* dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat *project* baru, membuka *project* dan menyimpan *project*.

- b) Konsep Dasar Pemrograman Dalam *Visual basic*

Konsep dasar pemrograman Visual Studio adalah pembuatan form dengan mengikuti aturan pemrograman Property, Metode dan *Event*. Hal ini berarti:

1. Property

Setiap komponen di dalam pemrograman Visual Studio dapat diatur propertinya sesuai dengan kebutuhan aplikasi.

2. Metode

Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

3. Event

Setiap komponen dapat beraksi melalui *event*, seperti *event click* pada *command button* yang tertulis dalam layar *script Command1_Click*.

2.2.2 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system DBMS*). *DBMS* merupakan sistem perangkat lunak yang memungkinkan pengguna *database* (*database user*) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh *DBMS*. Ada beberapa fungsi yang harus ditangani *DBMS* yaitu mengolah pendefinisian data, menangani permintaan pemakai untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh *DBA* (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun disk dan menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan tetap bisa diambil (diakses) secara efisien. Pertimbangan efisien di sini adalah bagaimana merancang struktur data yang kompleks tetapi masih tetap bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya.

2.2 Device Net

¹Device Net merupakan protokol komunikasi yang digunakan pada industri yang bekerja secara otomatis yang menghubungkan antara peralatan sensor, transmitter dan peralatan kontrol terhubung dalam jaringan komunikasi. Protokol ini menggunakan CAN atau (Kontroller Area Network) sebagai tulang punggung teknologi (backbone technology) dan mendefinisikan layer aplikasi untuk mengcover sejumlah peralatan yang dikendalikan. Aplikasi yang menggunakan protocol ini mencakup pertukaran informasi (information exchange), safety devices, dan jaringan kontrol peralatan Input/Output yang besar. Device Net awalnya dikembangkan oleh Perusahaan Amerika yaitu Allen-Bradley (sekarang berganti nama menjadi Rockwell Automation)

2.3.1 Protokol Device Net

Device Net menggunakan ²CAN (Kontroller Area Network) sebagai protokol komunikasi. CAN adalah standard serial komunikasi yang digunakan berbagai perangkat untuk saling berkomunikasi. CAN memiliki baud rate maksimal sebesar 1 mbps, tidak seperti serial komunikasi lain yang dapat mentransfer data hingga ribuan atau jutaan data bit dalam satu frame. Namun penggunaan pada industri hanya memerlukan baud rate dengan kecepatan sekitar 128 kbps. Baud rate adalah kecepatan aliran data, atau seberapa banyak bit data yang dapat ditransfer dalam waktu 1 detik. Bit atau binary digit adalah satuan digital terkecil dan hanya dapat memiliki nilai 1 atau 0. Sementara byte adalah kumpulan dari beberapa buah bit, 1 byte terdiri dari 8 buah bit berbeda. Keunggulan dari CAN tidak terletak pada kecepatan transfer data, melainkan minimnya overhead yang dibutuhkan untuk mentrasfer data. Inilah yang menyebabkan CAN digunakan pada Device Net. CAN memiliki beberapa field layer dengan fungsi yang berbeda, seperti start of frame, identifier field. Format data frame dari CAN adalah sebagai berikut ini.

S O F	11-bit Identifier	R T R	I D E	r0	DLC	0...8 Bytes Data	CRC	ACK	E O F	I F S
-------------	-------------------	-------------	-------------	----	-----	------------------	-----	-----	-------------	-------------

Gambar 2 Error! No text of specified style in document..1 Format Data Frame CAN

Gambar 2.2 menunjukkan format data frame yang dimiliki oleh CAN.

- SOF atau start of frame merupakan bit yang menandakan permulaan dari sebuah pesan. Digunakan untuk sinkronisasi dari beberapa node pada bus.
- Identified field berguna untuk menentukan prioritas pesan. Semakin kecil nilai binernya, maka semakin besar prioritasnya. Frame ini yang digunakan oleh Device Net.
- RTR atau Remote Transmission Request digunakan saat informasi diperlukan dari node lain. Saat semua node menerima permintaan, identifier berfungsi untuk menentukan node secara spesifik.
- IDE atau Single Identifier Extension menandakan bahwa sebuah identifier CAN tanpa ekstension sedang ditransmisikan.
- r0 merupakan bit spare untuk penggunaan yang akan datang.
- DLC merupakan 4-bit data length code yang berisi nomor byte yang sedang di transmisikan.
- Data field, menandakan bahwa 8 bytes data atau 64 bit data aplikasi yang dapat ditransmisikan.
- CRC atau Cyclic Redundancy Check berfungsi untuk aplikasi data yang mendeteksi error.
- ACK berfungsi untuk memeriksa integritas dari data yang ditransmisikan.
- EOF atau End Of Frame menandakan berakhirnya pesan frame CAN.
- IFS atau Interframe Space berisi waktu yang dibutuhkan kontroler untuk memindahkan frame ke posisi yang seharusnya pada area pesan.

Meskipun CAN memiliki banyak spesifikasi pesan frame yang berbeda seperti, data, remote, overload, error, Device Net hanya menggunakan frame identifier field dari CAN.

IDENTIFIER BITS										HEX RANGE	IDENTITY USAGE	
10	9	8	7	6	5	4	3	2	1			0
0	Group 1 Message ID		Source MAC ID							000 - 3ff	Message Group 1	
1	0	MAC ID			Group 2 Message ID				400 - 5ff	Message Group 2		
1	1	Group 3 Message ID		Source MAC ID						600 - 7bf	Message Group 3	
1	1	1	1	1	Group 4 Message ID (0 - 2f)					7c0 - 7ef	Message Group 4	
1	1	1	1	1	1	1	X	X	X	X	7f0 - 7ff	Invalid CAN Identifiers
10	9	8	7	6	5	4	3	2	1	0		

Gambar 2.2 Alokasi data Device Net pada CAN Identifier Field

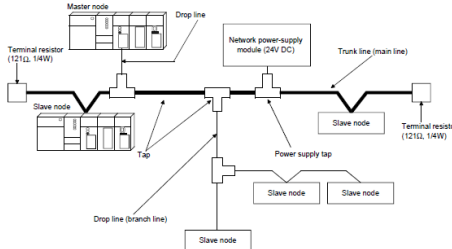
Gambar 2.3 menunjukkan alokasi data yang digunakan Device Net pada identifier field 11-bit CAN. Data tersebut terbagi menjadi 4 grup, tiga grup pertama terdiri dari 2 bagian; bagian pertama untuk 6-bit Media Access Core Identifier (MAC ID) dan yang lainnya untuk message ID. Identifier field inilah yang digunakan untuk mendeteksi node pada setiap perangkat Device Net. CAN identifier field juga memiliki mekanisme otomatis yang dapat mendeteksi keberadaan node dengan alamat yang sama.

¹ Omron, PLC Analog I/O dan Networking Manual

² Steve Corrigan, Introduction to The Kontroller Area Network

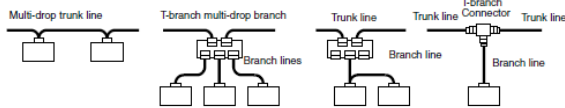
2.3.2 Karakteristik Device Net

Jaringan *Device Net* bisa disambungkan hingga 64 buah unit modul, termasuk unit *master* maupun unit *slave*. Setiap *nodes* atau unit dapat dikoneksikan langsung dari unit *master* atau dicabangkan terlebih dahulu menggunakan terminal.



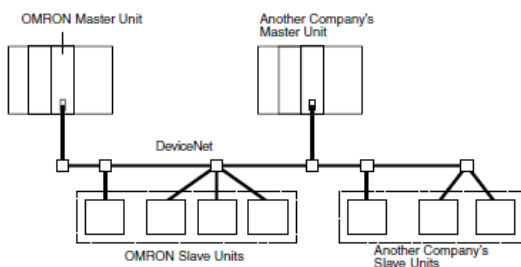
Gambar 2.3 Konfigurasi Device Net

Gambar 2.4 menunjukkan contoh sebuah konfigurasi *Device Net* yang terdiri dari sebuah unit *master* yang terhubung dengan total lima buah unit *slave*. Gambar 2.4 juga menunjukkan bahwa terdapat beberapa unit *device* yang dapat digunakan sebagai unit *master* maupun unit *slave*.



Gambar 2.4 Jenis Koneksi Device Net

Pada gambar 2.5 kita dapat melihat bahwa *Device Net* dapat dikoneksikan dengan berbagai cara, seperti koneksi *Multi-drop* maupun koneksi *T-brach*. Hal ini tentunya mempermudah kita untuk dapat memilih koneksi seperti apa yang akan kita gunakan. Selain itu *Device Net* juga dirancang agar dapat bekerja dengan perangkat dari berbagai merk berbeda. Hal ini tentu memudahkan kita untuk membuat sistem yang lebih ekonomis.



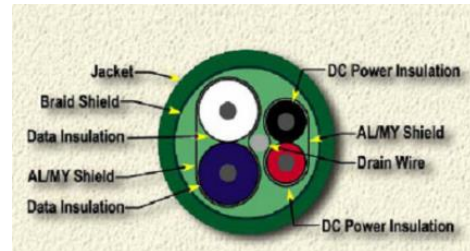
Gambar 2.5 Device Net multi-vendor networks

Karakteristik dari jaringan *Device Net* membutuhkan kabel yang dapat menyediakan sambungan untuk sinyal dan sumber tegangan dalam satu kabel. Sumber tegangan ini harus disuplai dari sebuah *power supply* yang tersambung secara langsung ke jaringan yang dapat memberikan tegangan ke semua *nodes* yang tersambung. Jumlah *baudrate* yang digunakan bergantung dari ukuran kabel pada jaringan. Tabel 1.1 menunjukkan nilai *baudrate* untuk ukuran kabel tertentu.

Tabel 1.1 Nilai Baudrate

Baud Rate	Network Size	Derivation	
		Maximum	Total
125kbps	500m	6m	156m
250kbps	250m		78m
500kbps	100m		39m

Kabel yang digunakan untuk jaringan ini adalah kabel *shielded 5-wire*. Kabel ini terdiri dari dua buah kabel sinyal, dua buah kabel *power*, dan sebuah kabel *shield*.

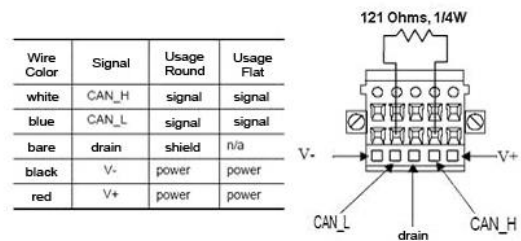


Gambar 2.6 Kabel Device Net

Tabel 2.2 Warna Kabel Device Net

Wire Color	Signal	Round Cable	Flat Cable
White	CAN_H	DN Signal	DN Signal DN
Blue	CAN_L	DN Signal	DN Signal
Naked wire	Drain	Shield	Not used
Black	V-	Power	Power
Red	V+	Power	Power

Konektor yang digunakan untuk menyambungkan unit *master* dan unit *slave* adalah sebagai berikut ini.



Gambar 2.7 Konektor Device Net

Pada gambar 2.8 ditunjukkan perlu adanya penambahan satu buah unit *resistor* 121 ohm yang dipasang diantara *socket CAN_L* dan *CAN_H*. *Resistor* ini berguna untuk mengurangi pemantulan komunikasi saat data disalurkan.

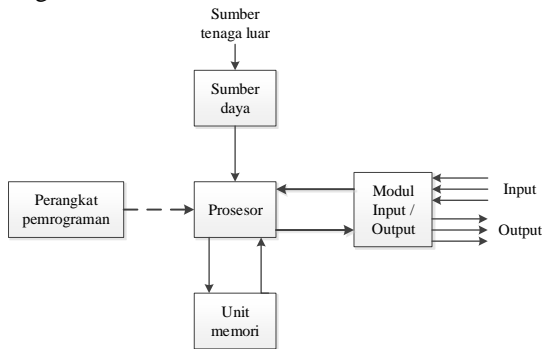
Sumber tenaga yang digunakan untuk koneksi *Device Net* adalah sebagai berikut ini.

Tabel 2.3 Power Supply

Power Supply (V _{dc})		
Minimum	Maximum	Recommended
11	30	24
Current (mA)		
Average	Maximum	
30	60	

2.3 PLC

³Programmable Logic Kontrol (PLC) adalah sistem berbasis mikroprosesor yang menggunakan memori yang dapat diprogram untuk menyimpan instruksi-instruksi dan mengimplementasikan fungsi-fungsi seperti fungsi logika, pengurutan, pewaktu, pencacahan dan aritmetika untuk mengontrol mesin-mesin dan proses, serta dirancang untuk dioperasikan oleh insinyur yang mungkin memiliki kemampuan dan pengetahuan terbatas mengenai komputer dan bahasa pemrograman.



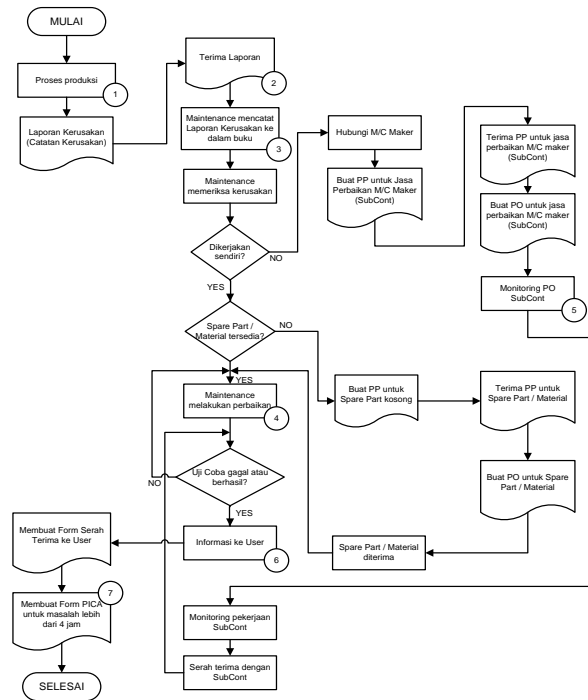
Gambar 2.8 Skematik sistem PLC⁴

PLC pada dasarnya adalah sistem komputer digital yang mampu melaksanakan proses kerja yang kompleks. Penggunaan PLC di lingkungan industri terhitung lebih praktis dan efisien daripada sistem kontrol konvensional. Saat terjadi perubahan proses, *user* hanya perlu mengubah instruksi-instruksi program yang dimasukkan didalamnya tanpa perlu perubahan *wiring*. Selain itu dalam hal pengabelan PLC mudah untuk dimodifikasi. PLC memiliki kemampuan untuk mengontrol sistem proses atau mesin yang kompleks seperti *Human Machine Interface* (HMI) dengan kecepatan respon yang tinggi dan efisien.

III. PENGUMPULAN DATA DAN PERANCANGAN

3.1 Analisa Permasalahan

Dalam mendapatkan data pencatatan downtime, tim *Breakdown maintenance* memiliki prosedur seperti gambar berikut.



Gambar 3.1 Alur proses *breakdown* mesin

Dari alur proses diatas dapat dilihat bahwa tidak ada proses yang mengatur untuk mencatat *downtime* yang timbul, hasil pencatatan *downtime* sangat bergantung kepada peranan operator. Dimana operator harus membuat laporan kerusakan kepada divisi *maintenance* dan awal mula waktu kerusakan berdasarkan perkiraan operator.

Berdasarkan penjabaran diatas, pencatatan *downtime* menjadi sangat tidak efektif karena akan timbul *loss time* dari waktu operator membuat laporan. Dalam gambar 3.1 hanya menjelaskan pencatatan downtime ketika mesin mengalami *breakdown* dan membutuhkan bantuan dari departemen *maintenance*, sehingga gangguan lain yang terjadi diluar keadaan *breakdown* tidak tercatat.

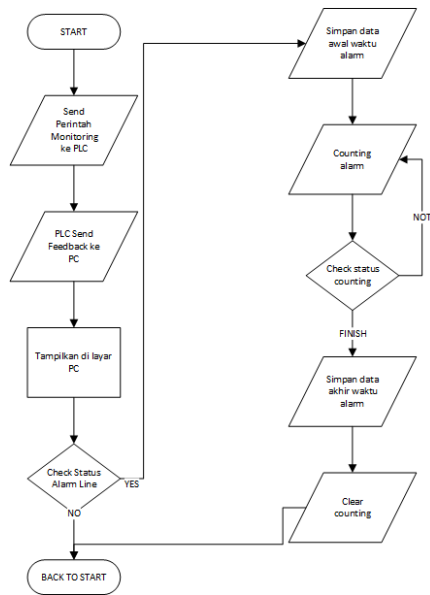
Dapat dilihat pada gambar diatas, akibat dari kelalaian operator dan sistem yang tidak efektif, muncul perbedaan nilai *downtime* yang dicatat oleh divisi *maintenance* dan divisi PPC.

Berdasarkan permasalahan yang ada, maka dibutuhkan suatu sistem yang mampu mencatat downtime secara otomatis. Agar *downtime* dapat tercatat secara akurat, maka sistem tersebut harus mampu mencatat *downtime* mesin dan memonitoring kinerja line secara *real time* dan akurat untuk 35 line *machining* di gedung A, B, dan C.

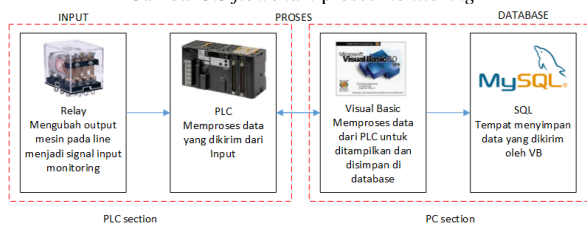
3.2 Perancangan

Pada sub bab ini penulis merancang sebuah sistem yang dapat memonitoring kinerja line secara *real time* serta menyimpan data *breakdown* kedalam database pada jaringan *server*. Penulis membuat *flowchart* dan sistematika *monitoring* yang akan diterapkan. Berikut ini adalah *flowchart* dan sistematika *monitoring* yang dirancang.

³ W. Bolton, Sistem Instrumentasi dan Sistem Kontrol (Jakarta: Erlangga, 2009), Halaman 158



Gambar 3.3 flowchart proses monitoring



Gambar 3.4 Sistematika monitoring

Dari gambar diatas dapat dilihat bahwa proses monitoring dan pencatatan downtime diawali dari :

1. PC mengirimkan perintah berupa monitoring ke PLC melalui kabel jaringan
2. Kemudian perintah tersebut akan dieksekusi oleh PLC, dan PLC akan mengirim feedback berupa data yang diminta oleh PC
3. PC mengolah dan menampilkan data yang telah diterima dari PLC. Jika terdapat alarm pada data yang diterima, maka PC akan memulai counting dan mencatat waktu terjadi alarm kedalam database.

Setelah alarm selesai, PC mencatat kembali waktu berhentinya alarm kedalam database.

IV. PEMBUATAN DAN PENGUJIAN

4.1 Pengabelan Sistem Kontrol

4.1.1 Layout Panel Master

Panel Master dirancang sebagai tempat komponen elektrik seperti power supply 24VDC, circuit protector, blok terminal, serta modul PLC yang terdiri dari CJ1M, ETN21, dan DRM21 yang nantinya akan terhubung pada komponen yang berada di panel Slave. Berikut adalah gambar Panel Master dan layout dari komponen yang ada pada Panel Master ditunjukkan pada gambar.



Gambar 4. 1 Layout panel master

4.1.2 Layout Panel Slave

Panel Slave dirancang sebagai tempat komponen elektrik seperti power supply 24VDC, blok terminal, relay, dan DRT21 yang terhubung dengan DRM21 pada Panel Master. Fungsi dari panel ini adalah menerima input dari line machining berupa tegangan 220VAC dan 24VDC, kemudian disetarakan menggunakan relay dan diteruskan ke modul DRT21 yang sudah terhubung dengan DRM21 pada panel master. Berikut adalah gambar dari panel slave dan layout komponen elektronik yang digunakan ditunjukkan pada Gambar 4.2.

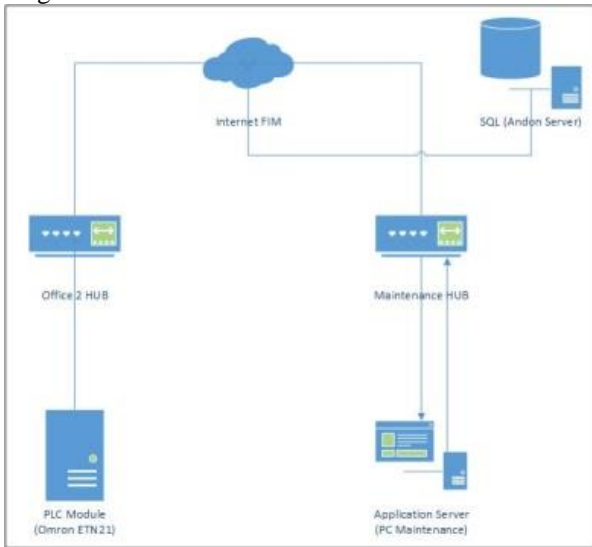


Gambar 4. 2 Layout Panel Slave

4.2 Pengabelan Jaringan Ethernet

Sistem monitoring ini memerlukan sebuah perangkat sebagai media penghubung atau komunikasi dengan master, dalam hal ini master adalah personal computer dan slave adalah PLC. Untuk lebih jelasnya

dapat dilihat pada figure 4.3 tentang *layout wiring* kabel jaringan ethernet.

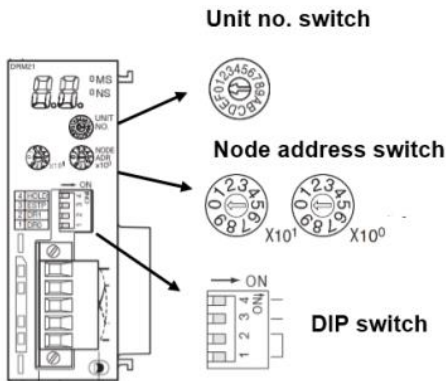


Gambar 4.3 Layout jaringan ethernet

Dari gambar diatas dapat dilihat bahwa ethernet PLC terhubung dengan HUB yang terhubung dengan jaringan internet PT.FIM dan *personal computer* juga terhubung dengan HUB yang terhubung dengan jaringan internet PT. FIM. Maka untuk pengalaman ethernet PLC dan computer mengikuti IP yang sudah disediakan oleh PT. FIM.

4.3 Pengalaman Device Net

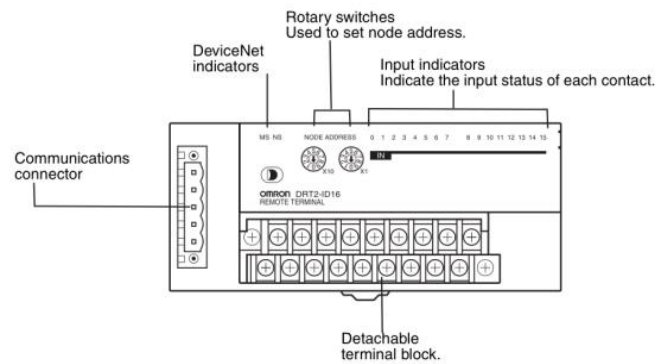
Terdapat beberapa parameter yang harus diatur pada unit *Master*.



Gambar 4.4 Parameter DRM21

Parameter bagian unit no. *switch* berada pada nilai 0, hal ini menunjukkan bahwa perangkat ini adalah modul PLC nomor 1. Kemudian parameter *node address switch* berapa pada nilai 0, hal ini menunjukkan bahwa perangkat ini memiliki nilai node 0. Ini juga berarti bahwa perangkat ini akan berfungsi sebagai unit *master*. Realisasi pengalaman unit *master* dapat dilihat pada gambar berikut. Kemudian pada bagian *dip switch* nomor 2 diatur pada posisi *ON*, hal ini menunjukkan bahwa *baudrate* yang digunakan dapat mencapai 500 kbps.

Pengaturan yang dilakukan pada unit slave adalah bagian switch node parameter.



Gambar 4.5 Konstruksi DRT2-ID16

Penentuan node yang diatur harus sesuai dengan yang telah di bahas pada bab sebelumnya. Berikut adalah alokasi nilai node yang telah diatur.

Tabel 4.2 Alokasi Nilai Node

Device	Nilai node	Input
Panel Slave A1	1	11
Panel Slave A2	2	12
Panel Slave B1	4	14
Panel Slave B2	3	13
Panel Slave C1	5	15
Panel Slave C2	6	16

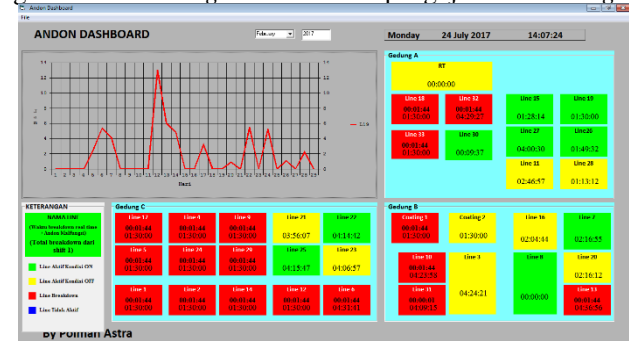
4.4 Pembuatan Program VB

Program VB digunakan untuk memonitoring perangkat PLC yang telah menyimpan data dari tiap – tiap Line Machining dan untuk menyimpan serta menampilkan data downtime yang akan dicatat.

4.5 Pengujian Sistem Kerja Monitoring

Pengujian ini bertujuan untuk mengetahui apakah program yang telah dibuat telah benar dan dapat melakukan *monitoring* data pada PLC. Pengujian dilakukan dengan cara melihat secara visual keadaan PLC dengan VB.

Monitoring dinyatakan ok apabila VB dapat melakukan *monitoring* data yang ada pada PLC dan dinyatakan not good (NG) apabila VB tidak dapat melakukan *monitoring* data di PLC. Berikut adalah gambar *monitoring* VB serta tabel pengujian *monitoring*.



Gambar 4.93 Tampilan Andon Dashboard

V. KESIMPULAN

Pembuatan dan perancangan sistem *monitoring line machining* berbasis *visual basic* dan plc omron CJ1M menghasilkan kesimpulan yang dapat menjawab perumusan masalah yang ada dalam tugas akhir ini, yaitu

sistem *monitoring* yang penulis buat, telah memenuhi rancangan yang direncanakan. Dimana, sistem *monitoring* dapat memonitor kondisi line secara real-time dan dapat menaikkan akurasi pencatatan *downtime* sebanyak 3.01% dari jumlah total jam kerja mesin selama satu bulan.

Daftar Acuan

- [1] Bolton, W. 2009. *Sistem Instrumentasi dan Sistem Kontrol* (Jakarta: Erlangga).
- [2] Mardiani, Gentisya Tri. *Sitem Monitoring Data Aset dan Inventaris PT. Telkom Cianjur Berbasis Web*. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) , ISSN : 2089-9033, Vol. 2. No. 1, Maret 2013
- [3] Jimmy, Mada. 2010. *Panduan Praktikum PLC (Programmable Logic Controller)*. Jakarta: Politeknik Manufaktur Astra.
- [4] Pangestu, Danu Wira. 2009. *Modul Pelatihan Visual Basic – Access*. [Online]. Tersedia di: <http://depositfiles.com/files/8jqu57lgi>.
- [5] Law, Robert D. 2010, *DeviceNet Book of Knowledge*, Rockwell Automation.