



PROYEK AKHIR - VE180626

***Monitoring Mesin ICT (In Circuit Test) Berbasis
Internet of Things (IoT) Dan SQL Server di PT.
Schneider Electric Manufacturing Batam***

**Dhimas Adi Pangestu
NRP 1031160000048**

**Dosen Pembimbing
Ir. Joko Susila, M.T.
Slamet Budiprayitno, S.T., M.T.**

**DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

- Halaman ini sengaja dikosongkan -



FINAL PROJECT - VE180626

***ICT (In Circuit Test) Machine Monitoring Based On
Internet of Things (IoT) And SQL Server At PT.
Schneider Electric Manufacturing Batam***

**Dhimas Adi Pangestu
NRP 1031160000048**

Supervisor
Ir. Joko Susila, M.T.
Slamet Budiprayitno, S.T., M.T.

***ELECTRICAL OTOMATION ENGINEERING DEPARTMENT
Faculty of Vocation
Institut Teknologi Sepuluh Nopember
Surabaya 2019***

- Halaman ini sengaja dikosongkan -

**MONITORING MESIN ICT (IN CIRCUIT TEST) BERBASIS
INTERNET OF THINGS DAN SQL SERVER DI PT. SCHNEIDER
ELECTRIC MANUFACTURING BATAM**

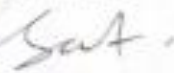
PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Pembimbing 1

Pembimbing 2


Ir. Joko Susila, M.T.
NIP. 19660606 199102 1 003


Slamet Badiprasimo, S.T., M.T.
NIP. 19781113 201012 1 002

**SURABAYA
JULI, 2019**

- Halaman ini sengaja dikosongkan -

***Monitoring Mesin ICT (In Circuit Test) Berbasis Internet of Things
(IoT) Dan SQL Server di PT. Schneider Electric Manufacturing
Batam***

Nama Mahasiswa : Dhimas Adi Pangestu
NRP : 1031160000048
Dosen Pembimbing 1 : Ir. Joko Susila, M.T.
NIP : 196606061991021001
Dosen Pembimbing 2 : Slamet Budiprayitno, S.T., M.T.
NIP : 197811132010121002

ABSTRAK

Pada industri manufaktur yang memproduksi PCB pasti membutuhkan sebuah mesin untuk melakukan uji coba kelayakan dan fungsional dari sebuah PCB. Hampir setiap industri manufaktur yang memproduksi PCB akan menggunakan mesin ICT sebagai *tester* untuk melakukan pengecekan terhadap PCB yang diproduksi. Sehingga keberadaan mesin ICT sangat penting dalam produksi PCB di industri.

Mesin ICT merupakan mesin yang harus berjalan dengan maksimal dan optimal. Saat ini mesin ICT pada PT. SEMB beroperasi sebagaimana mestinya, tidak ada pengawasan lebih intensif terhadap parameter-parameter yang sangat penting yang berhubungan dengan kinerja dari mesin ICT. Sehingga untuk meningkatkan dan mengoptimalkan kinerja dari mesin ICT harus dilakukan pengawasan secara khusus dengan membuat sebuah sistem *monitoring* yang berfungsi untuk mengawasi beberapa elemen penting dari mesin ICT sehingga dapat dilakukan analisis terhadap mesin ICT dan mencegah terjadinya masalah yang akan terjadi pada mesin ICT yang akan mengganggu kinerja dari mesin ICT. Agar mempermudah pengawasan terhadap mesin ini, pengawasan ini menggunakan konsep *Internet of Things* dan agar mempermudah analisis terhadap parameter-parameter mesin ICT, sistem ini dikombinasikan dengan metode *Statistical Process Control*.

Hasil dari sistem ini meningkatkan kinerja dari mesin ICT, dan mencegah terjadinya *machine breakdown* pada mesin ICT.

. **Kata Kunci:** Mesin *In Circuit Test*, *Internet of Things*, *Statistical Process Control*

- Halaman ini sengaja dikosongkan -

ICT (In Circuit Test) Machine Monitoring Based On Internet of Things (IoT) And SQL Server At PT. Schneider Electric Manufacturing Batam

Name Of Student 1 : **Dhimas Adi Pangestu**
Number Of Registration : **10311600000048**
Supervisor 1 : **Ir. Joko Susila, M.T.**
ID Number : **196606061991021001**
Supervisor 2 : **Slamet Budiprayitno, S.T., M.T.**
ID Number : **197811132010121002**

ABSTRACT

In Industrial Manufacturing which produces PCB need a machine for quality and functional testing. Every industrial manufacturing which produces PCB using ICT (In Circuit Test) Machine as a tester for the quality and functional testing. The existence of this machine is very important in PCB Assembly at industrial manufacturing.

ICT Machine is the important thing in PCB Assembly, because of that ICT Machine must running optimally. Now, in PT. SEMB, ICT Machine running without monitoring on it's important parameters that effect on the performance of the machine. To improve and optimize performance of ICT Machine, special monitoring must be carried out. Build a monitoring system to monitor important parameters of the ICT Machine, so the engineers can analyze the performance of the machine and prevent problems of the ICT machine that will inhibit ICT machine performance. To make this system easier, this monitoring system uses internet of things concept, and to analyze the ICT Machine performance easier, this system will be combined with statistical process control method.

For the result, this system will make the performance of the ICT Machine increased, and prevent the problems of the ICT Machine.

Key word: *In Circuit Test Machine, Internet of Things, Statistical Process Control*

- Halaman ini sengaja dikosongkan -

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan ke hadirat Allah SWT serta sholawat dan salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW sehingga penulis dapat menyelesaikan Proyek Akhir dengan judul:

Monitoring Mesin ICT (In Circuit Test) Berbasis Internet of Things (IoT) Dan SQL Server di PT. Schneider Electric Manufacturing Batam

Proyek Akhir ini merupakan salah satu persyaratan yang harus dipenuhi dalam menyelesaikan pendidikan program Diploma III di Jurusan Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam pembuatan Proyek Akhir ini, khususnya kepada:

1. Bapak, Ibu, serta seluruh keluarga yang telah memberi dukungan.
2. Bapak Ir. Joko Susila, MT. selaku dosen pembimbing 1 dan Kepala Departemen Teknik Elektro Otomasi atas bimbingan dan arahan selama penulis mengerjakan Proyek Akhir ini.
3. Bapak Slamet Budiprayitno, ST., MT. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan Proyek Akhir ini.
4. Seluruh dosen pengajar dan staff Jurusan Teknik Elektro Otomasi yang banyak memberikan ilmu dan pengalaman selama penulis menempuh kuliah.
5. Teman-teman angkatan 2016 yang senantiasa membantu dan mendukung pengerjaan Proyek Akhir ini.

Penulis sadar bahwa Proyek Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 18 Juli 2019

Dhimas Adi Pangestu

- Halaman ini sengaja dikosongkan -

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Metodologi	2
1.6. Sistematika Penulisan	3
1.7. Relevansi dan Manfaat	4
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG	6
2.1. Mesin ICT	6
2.2. Statistical Process Control	6
2.2.1. Jenis-Jenis Variasi	7
2.2.2. Diagram Kontrol Variabel	8
2.3. Internet of Things	9
2.3.1. Modular PC Box	10
2.3.2. Node-Red	10
2.3.3. Augmented Operator Advisor	11
2.3.4. Machine Advisor	12
2.3.5. Aveva Insight	13
2.4. Sensor CL110	13
2.5. ZBRN32	14
2.6. PowerTag	14
2.7. Acti 9 Smartink SI D	15
2.8. PLC TM241CEC24R	15
2.9. Keyence GP-M010	16
2.10. Visual Studio	17

2.11. Microsoft SQL Server	18
BAB III PERANCANGAN SISTEM	19
3.1. Prinsip Kerja	19
3.2. Perancangan Sensor Suhu Dan Kelembaban	19
3.3. Perancangan Sensor Energi	20
3.4. Perancangan Sensor Tekanan	21
3.5. Pembuatan Aplikasi Interface	22
3.5.1. Aplikasi Data Extractor	23
3.5.2. Aplikasi Interface	23
3.5.3. Perancangan Node-Red	24
3.5.4. Augmented Operator Advisor	26
3.5.5. Machine Advisor	29
3.5.6. Aveva Insight	32
BAB IV PENGUJIAN DAN ANALISA	33
4.1. Pengujian Komunikasi Sistem	33
4.1.1. Sensor Suhu Dan Kelembaban	33
4.1.2. Sensor Tegangan	35
4.1.3. Sensor Tekanan	38
4.2. Pengujian Sistem	40
4.2.1. Pengujian Aplikasi Data Extractor	40
4.2.2. Pengujian Aplikasi Interface	41
4.2.3. Pengujian Machine Advisor	44
4.2.4. Pengujian Augmented Operator Advisor	44
4.2.5. Pengujian IE Andon	45
BAB V PENUTUP	47
5.1. Kesimpulan	47
5.2. Saran	47
DAFTAR PUSTAKA	49
LAMPIRAN	51
BIODATA PENULIS	85

DAFTAR GAMBAR

Gambar 2.1 Mesin ICT	5
Gambar 2.2 Contoh Grafik Pengendali.....	9
Gambar 2.3 Modular PC Box.....	10
Gambar 2.4 Flow Node-Red	15
Gambar 2.5 Augmented Operator Advisor	15
Gambar 2.6 Machine Advisor	12
Gambar 2.7 Aveva Insight	13
Gambar 2.8 Sensor CL110	13
Gambar 2.9 ZBRN32	14
Gambar 2.10 PowerTag	14
Gambar 2.11 Acti 9 Smartlink SI D	15
Gambar 2.12 PLC TM241CEC24R.....	16
Gambar 2.13 Keyence GP-M010	17
Gambar 2.14 Visual Studio	18
Gambar 2.15 Microsoft SQL Server	18
Gambar 3.1 Rangkaian Sensor Suhu Dan Kelembaban.....	20
Gambar 3.2 Rangkaian Sensor Energy	21
Gambar 3.3 Rangkaian Sensor Tekanan	22
Gambar 3.4 Aplikasi Data Extractor.....	23
Gambar 3.5 Flow Node-Red Pengambilan Data dari Datalog	24
Gambar 3.6 Flow Node-Red Pengambilan Data dari SQL	24
Gambar 3.7 Flow Node-Red Acti9 3 Phase	25
Gambar 3.8 Flow Node-Red pengambilan data dari ZBRN32	25
Gambar 3.9 Flow Node-Red unggah file ke Aveva Insight.....	25
Gambar 3.10 Augmented Operator Advisor	26
Gambar 3.11 Augmented Operator Advisor Scene Builder.....	27
Gambar 3.12 Augmented Operator Advisor POI Builder.....	27
Gambar 3.13 Augmented Operator Advisor Variabel Builder.....	28
Gambar 3.14 Augmented Operator Advisor Project Builder	28
Gambar 3.15 Augmented Operator Advisor Builder.....	29
Gambar 3.16 Machine Advisor	29
Gambar 3.17 Fleet Machine Advisor.....	30
Gambar 3.18 Architecture Machine Advisor	30
Gambar 3.19 Config Machine Advisor.....	31
Gambar 3.20 Dashboard Aveva Insight.....	32
Gambar 4.1 Flow Node-Red Pengambilan Data CL110	33
Gambar 4.2 Konfigurasi Pada Node Serial	34

Gambar 4.3 Output pengambilan data sensor suhu dan kelembaban....	34
Gambar 4.4 Flow Node-Red Dengan Node Modbus	36
Gambar 4.5 Konfigurasi server IP Acti 9	36
Gambar 4.6 Konfigurasi Address.....	37
Gambar 4.7 Ladder Diagram	38
Gambar 4.8 Aplikasi Data Extractor	40
Gambar 4.9 Database SQL Server	41
Gambar 4.10 Halaman Utama Aplikasi Interface	41
Gambar 4.11 Halaman Grafik Pada Parameter Tekanan.....	42
Gambar 4.12 Halaman Grafik Pada Parameter Tegangan.....	42
Gambar 4.13 Halaman Grafik Pada Parameter Suhu	43
Gambar 4.14 Halaman Grafik Pada Parameter Kelembaban	43
Gambar 4.15 Dashboard Machine Advisor	44
Gambar 4.16 Augmented Operator Advisor.....	44
Gambar 4.17 IE Andon.....	45
Gambar 4.18 Dashboard IE Andon.....	46

DAFTAR TABEL

Tabel 4.1 Data Percobaan Sensor Suhu Dan Kelembaban	35
Tabel 4.2 Data Percobaan Sensor Tegangan Dan Tekanan	37
Tabel 4.3 Data Percobaan Sensor Tekanan	39

- Halaman ini sengaja dikosongkan -

BAB I

PENDAHULUAN

1.1. Latar Belakang

Era globalisasi, sebuah perkembangan dan kemajuannya berbagai macam teknologi. Pada beberapa tahun belakang ini kemajuan teknologi sudah berkembang dengan cepat dan pesat. Berbagai macam hal yang dahulu dikerjakan atau diterapkan secara manual atau menggunakan sumber daya manusia sebagai pelaksananya, saat ini sudah tergantikan dengan mesin mesin canggih yang mampu menggantikan pekerjaan yang dilakukan manusia.

Keberhasilan penemuan ini tentu akan memicu para ilmuwan-ilmuwan lain untuk menciptakan teknologi yang sangat terbarukan dan inovatif, terlebih persaingan dari para negara-negara yang sudah maju dan sudah mampu berhasil menciptakan teknologi untuk kepentingan individu, kelompok, negara, dan umum. Dengan persaingan dan kegengsiannya inilah yang akan membuat kita semua terdorong untuk menciptakan teknologi yang sangat berguna untuk menunjang kegiatan sehari-hari.

Schneider Electric merupakan perusahaan electric yang sudah sangat terkenal namanya dalam bidang peralatan-peralatan electrical, dan salah satu perusahaan yang ter-inovatif yang berpusat di Perancis dan memiliki banyak sekali anak-anak cabang salah satunya PT. Schneider Electric Manufacturing Batam yang terletak di Batam dan berbasis manufaktur yang memproduksi berbagai macam peralatan electrical yang berguna untuk menunjang suatu sistem atau alat yang digunakan untuk menciptakan teknologi.

Kualitas produksi pada perusahaan ini merupakan aspek yang sangat penting demi menunjang sasaran-sasaran produksi dan pemasaran. Kualitas ini yang harus dijaga sebaik mungkin agar para customer terus percaya terhadap produk yang dihasilkan perusahaan ini. Tetapi, penjagaan dan pengawasan kualitas produksi di perusahaan ini sebagian besar masih secara manual, terlebih pada era digital seperti saat ini harusnya sudah tidak manual atau sudah otomatis.

Atas dasar ini, penulis ingin meng-*improve* dan menerapkan teknologi yang saat ini sedang di gadang-gadang, yaitu *Internet of Things*. Konsep ini menawarkan sistem dimana informasi yang berupa data-data disambungkan atau dihubungkan dari perangkat satu ke perangkat lainnya dengan menggunakan konektivitas. Dengan konsep

seperti ini, untuk menjaga dan mengawasi suatu kualitas produksi akan sangat mudah. Yang selama ini jika ingin mengecek suatu kondisi mesin produksi atau hasil produksi harus langsung ditinjau di tempat, konsep *Internet of Things* atau yang lebih dikenal dengan sebutan IoT ini dapat memberikan sesuatu yang berbeda dan sangat efisien, yaitu kita dapat mengecek suatu kondisi mesin produksi dan hasil produksi hanya dengan perangkat elektroik atau digital. Dengan cara seperti itu akan sangat mengefesienkan waktu, dan terlebih akan meningkatkan jumlah produksi.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam Proyek Akhir ini adalah:

1. Bagaimana membuat sebuah sistem pengawasan yang baik pada mesin ICT yang sangat butuh pengawasan ekstra di suatu industri?
2. Bagaimana cara meningkatkan kualitas produksi di industri dengan berbasis industri 4.0?
3. Bagaimana meng-implentasikan industri 4.0 pada suatu sistem data mesin *monitoring* di suatu industri?

1.3. Batasan Masalah

Batasan masalah dalam Proyek Akhir ini adalah:

1. Sistem *monitoring* pada *smartphone* atau *gadget* hanya khusus staff Schneider Electric.
2. Tidak ada pengontrolan pada mesin yang bermasalah.

1.4. Tujuan

Tujuan penelitian pada Proyek Akhir ini adalah sebagai berikut:

1. Mampu mengetahui konsep sistem pengawasan atau *monitoring* yang baik.
2. Mampu mengimplementasikan industri 4.0 .
3. Mempunyai keterampilan dalam membuat suatu sistem *monitoring*.

1.5. Metodologi

Langkah langkah yang dikerjakan pada Proyek Akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Proyek Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. Perancangan *Hardware*

Pada tahap ini dirancang sebuah *hardware* yang terdiri atas sensor yang terdiri dari sensor suhu, kelembaban, tegangan, dan tekanan, serta *receiver* dari sensor suhu, kelembaban, dan tegangan yang terkoneksi ke PC IoT.

3. Perancangan *Software*

Pada tahap ini disusun *software* pada PC IoT untuk mengimplementasikan IoT pada mesin yang terkoneksi. Dan juga *software pada pc menggunakan visual studio 2017* untuk mengambil dan mengolah data.

4. Pengujian Sistem

Proses pengujian sistem dilakukan dengan membaca data sensor dan mengirimkannya ke database. Apabila data berhasil dikirim dan terbaca pada sistem IoT maka pengujian dinyatakan berhasil.

5. Analisa

Data-data yang diperoleh pada tahapan pengujian perlu di analisa lebih lanjut untuk mencari batasan-batasan pada metode SPC. Berdasarkan data-data tersebut juga dapat diketahui langkah-langkah pencegahan ataupun mengatasi masalah yang timbul.

6. Penulisan Laporan Proyek Akhir

Tahap penulisan laporan Proyek Akhir adalah tahapan terakhir dari proses pengerjaan Proyek Akhir ini. Laporan Proyek Akhir berisi seluruh hal yang berkaitan dengan Proyek Akhir yang dikerjakan meliputi pendahuluan, teori penunjang, perancangan sistem, pengujian dan penutup.

1.6. Sistematika Penulisan

Dalam buku Proyek Akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

❖ BAB I: PENDAHULUAN

Pada bagian ini menjelaskan tentang dasar penyusunan Proyek Akhir yang terdiri dari beberapa sub bagian, yaitu latar belakang, permasalahan dan batasannya, tujuan yang diharapkan,

metodologi pengerjaan Proyek Akhir, relevansi serta sistematika penulisan Proyek Akhir ini.

- ❖ **BAB II : TINJAUAN PUSTAKAN DAN TEORI PENUNJANG** Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Proyek Akhir ini, yang meliputi pengetahuan tentang Mesin ICT, *Statistical Process Control*, *Internet of Things*, sensor suhu, PLC, database *SQL Server*, dan dasar pengiriman data pada *visual studio*.
- ❖ **BAB III : PERANCANGAN SISTEM**
Pada bagian ini berisi perancangan sistem yang terdiri dari beberapa bagian, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).
- ❖ **BAB IV : PENGUJIAN**
Pada bagian ini akan menjelaskan mengenai beberapa jenis pengujian yang dilakukan dalam Proyek Akhir ini seperti pengujian sensor suhu, sensor kelembaban, tegangan, dan tekanan, pengujian pembacaan, pengolahan, dan pengiriman data sensor, pengujian *interface statistical process control* dengan data yang sudah ada.
- ❖ **BAB V : PENUTUP**
Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Proyek Akhir ini, serta saran-saran untuk pengembangan dari Proyek Akhir berikutnya.

1.7. Relevansi dan Manfaat

Metode *Statistical Parameter Control* ini diharapkan mampu membantu kebutuhan dunia industri pada divisi *Quality* sehingga produksi industri akan lebih berkualitas, ditambah konsep *Internet of Things* yang digunakan untuk mengawasi jalannya produksi di industri sehingga produksi industri diharapkan akan tepat sasaran. Selain itu metode-metode ini diharapkan nantinya dapat dikembangkan pada objek yang lain sehingga dapat memenuhi kebutuhan manusia di masa depan nantinya.

BAB II

TEORI PENUNJANG

Tinjauan pustaka dalam bab ini menjelaskan mengenai sistem *monitoring* pada mesin ICT dengan metode *Statistical Process Control* dan *Internet of Things*. SPC (*Statistical Process Control*) ini berfungsi untuk menunjukkan kualitas dari parameter yang diambil dan historikal pergerakan dari parameter tersebut. Data-data yang berguna untuk menunjang sistem ini diolah menggunakan berbagai macam aplikasi. Sedangkan *Internet of Things* ini berfungsi sebagai perantara antara parameter mesin yang diambil dengan sistem *statistical process control*, dan juga berfungsi sebagai *interface* dari keseluruhan proyek akhir ini. Berdasarkan rancangan sistem-sistem tersebut, teori penunjang pada Proyek Akhir ini meliputi penjelasan mengenai *statistical process control* dan macam-macam *control chart*-nya, serta *Internet of Things* dan cara merencangnya.

2.1. Mesin ICT (*In Circuit Test*)



Gambar 2.1 Mesin ICT[1]

ICT (*In Circuit Test*) adalah suatu alat atau mesin yang digunakan untuk menguji secara fungsional dan kelayakan suatu PCB yang sudah

terangkai dari berbagai komponen-komponen. Pada industri manufaktur yang memproduksi PCB pasti akan menggunakan mesin ini sebagai *tester* terhadap PCB yang dihasilkan.

Mesin ICT ini sangat diyakini oleh perusahaan industri besar dan ternama untuk melakukan pengujian terhadap PCB, karena mesin ICT ini mempunyai kemampuan uji yang sangat baik, mesin ini dapat mendeteksi kesalahan-kesalahan yang terdapat pada PCB dengan akurat. Mesin ICT juga dapat melakukan pengujian pada PCB dengan tingkat kerumitan yang sangat tinggi, dengan kemampuan dapat melakukan pengujian hingga 5000 titik pada PCB. Serta mesin ICT ini dapat melakukan pengujian pada jumlah PCB yang dapat dikatakan tidak sedikit[1].

Dengan kemampuan yang dimiliki mesin ini harus dilakukan perawatan dan pengawasan secara berkala, terlebih mesin ini sangat kritikal atau krusial dalam proses pembuatan PCB secara menyeluruh. Jika terjadi kesalahan atau masalah pada mesin ini, dapat dipastikan akan membuat produktivitas pada suatu produksi industri akan menurun yang akan berimbas pada kerugian pada perusahaan.

2.2. *Statistical Process Control*

Statistical Process Control (SPC) merupakan teknik penyelesaian masalah yang digunakan sebagai pemonitor, pengendali, penganalisis, pengelola, dan memperbaiki proses menggunakan metode – metode statistik. Filosofi pada konsep pengendalian kualitas proses statistik atau yang lebih dikenal dengan pengendalian proses statistik (*Statistical Process Control*) adalah output pada proses atau pelayanan dapat dikemukakan ke dalam pengendalian statistik melalui alat-alat manajemen dan tindakan perancangan. Pengendalian proses statistik merupakan penerapan metode-metode statistik untuk pengukuran dan analisis variasi proses. Teknik ini merupakan parameter-parameter pada proses dan analisis proses. Dengan menggunakan pengendalian proses statistik ini maka dapat dilakukan analisis dan minimasi penyimpangan atau kesalahan, mengkuantifikasikan kemampuan proses, menggunakan pendekatan statistik dengan dasar six – sigma, dan membuat hubungan antara konsep dan teknik yang ada untuk mengadakan perbaikan proses. Sasaran pengendalian proses statistik adalah terutama adalah mengadakan pengurangan terhadap variasi atau kesalahan – kesalahan proses. Selain itu, tujuan utama dalam pengendalian proses statistik adalah mendeteksi adanya khusus (*assignable cause* atau *special cause*)

dalam variasi atau kesalahan proses melalui analisis data dari masalah maupun masa mendatang. Variasi proses terdiri dari dua macam penyebab, yaitu penyebab umum (*random cause* atau *chance cause* atau *common cause*) yang sudah melekat pada proses, dan penyebab khusus (*assignable cause* atau *special cause*) yang merupakan kesalahan yang berlebihan. Idealnya, hanya penyebab umum yang ditunjukkan atau yang tampak dalam proses, karena hal tersebut menunjukkan bahwa proses berada dalam kondisi stabil dan dapat diprediksi. Kondisi ini menunjukkan variasi yang minimum.

Dalam setiap proses produksi, hal yang perlu dipahami adalah setiap produk ataupun jasa yang dihasilkan tidak akan 100% sama. Hal ini karena adanya variasi selama proses produksi berlangsung. Adanya variasi merupakan hal yang normal dan wajar, namun akan berpengaruh pada kualitas produk sehingga perlu dikendalikan.

Umumnya, metode statistik banyak digunakan dalam upaya pengendalian proses produksi. Pendekatan yang paling umum digunakan dalam dunia industri adalah melalui metode *Statistical Process Control* (SPC).

Statistical Process Control merupakan metode pengambilan keputusan secara analitis yang memperlihatkan suatu proses berjalan dengan baik atau tidak. SPC digunakan untuk memantau konsistensi proses yang digunakan untuk pembuatan produk yang dirancang dengan tujuan mendapatkan proses yang terkontrol.

2.1.1. Jenis-Jenis Variasi

Variasi didefinisikan sebagai ketidakseragaman produk atau jasa yang dihasilkan. Variasi dapat pula didefinisikan sebagai produk atau jasa yang dihasilkan tidak memenuhi spesifikasi standard yang telah ditetapkan. Variasi dikelompokkan menjadi 2 jenis:

1. Variasi Terkendali (*Controllable Variation*)

Variasi terkendali adalah variasi yang dapat dikendalikan atau variasi yang dapat dihilangkan atau diminimalisir jika dilakukan aktifitas perbaikan. Variasi jenis ini biasanya bersifat stabil, konsisten, kemungkinannya random, terprediksi, terjadi secara alamiah, inheren, sebab-sebab acak. Contoh jenis variasi ini adalah kurang homogenya bahan baku kurang cermatnya operator dan lain-lain.

2. Variasi Tidak Terkendali (*Uncontrollable Variation*)

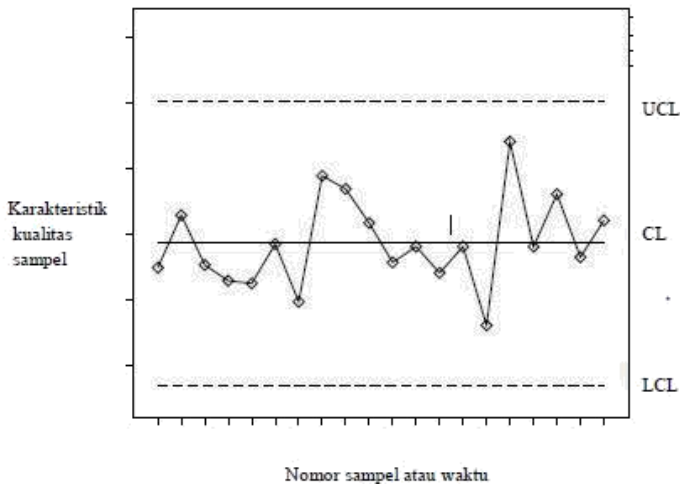
Variasi tidak terkendali adalah variasi yang tidak dapat dikendalikan. Variasi jenis ini biasanya bersifat tidak stabil, tidak konsisten, tidak terprediksi, dan umumnya terjadi karena faktor alam atau lingkungan, sehingga menyebabkan abnormalitas terhadap sistem dan dapat diperbaiki secara lokal. Contoh variasi jenis ini adalah kelembaban udara, suhu ruangan yang berubah-ubah, perubahan tegangan listrik, dan lain-lain.

2.1.2. Diagram Kontrol Variabel

Diagram kontrol adalah teknik pengendali proses pada jalur yang digunakan secara luas untuk menaksir parameter suatu proses produksi menentukan kemampuan dan memberi informasi yang berguna dalam meningkatkan proses tersebut (Montgomery, 1990).

Grafik pengendali atau disebut diagram kontrol adalah alat untuk menggambarkan dengan cara yang tepat apa yang dimaksudkan dengan pengendalian statistik, dengan itu dapat digunakan dalam berbagai cara. Grafik pengendali pertamakali ditemukan oleh Dr. Walter Andrew Shewhart, dari Bell Telephone Laboratories Amerika Serikat pada tahun 1924. Dalam banyak penerapan, ini digunakan untuk mengawasi proses pada jalur. Yakni, data sampel dikumpulkan dan digunakan untuk membentuk grafik pengendali. Grafik pengendali dapat diklasifikasikan ke dalam dua tipe umum. Apabila karakteristik kualitas dapat diukur dan dinyatakan dalam bilangan, dinamakan *variable*. Dalam hal seperti itu, tepat sekali untuk melukiskan karakteristik kualitas dengan ukuran tengah dan ukuran variabilitas. Grafik pengendali untuk nilai tengah dan variabilitas dinamakan *grafik pengendali variabel*.

Banyak karakteristik kualitas yang tidak diukur dengan skala kuantitatif. Dalam keadaan ini, dapat dinilai tiap unit produk sebagai sesuai atau tidak sesuai atas dasar apakah produk itu memiliki atau tidak memiliki sifat tertentu, atau dapat mencacah banyak yang tidak sesuai (cacat) yang tampak pada suatu unit produk. Grafik pengendali untuk karakteristik kualitas semacam itu dinamakan grafik pengendali sifat (atribut).



Gambar 2.2 Contoh Grafik Pengendali[2]

Bentuk dasar grafik pengendali ditunjukkan dalam gambar yang merupakan peragaan grafik suatu karakteristik kualitas yang telah diukur atau dihitung dari sampel terhadap nomor sampel atau waktu. Grafik itu memuat:

1. *Central Line* (CL) atau disebut garis tengah yang merupakan nilai rata – rata karakteristik kualitas yang berkaitan dengan keadaan terkendali.
2. *Upper Control Limit* (UCL) yaitu batas pengendali atas.
3. *Lower Control Limit* (LCL) yaitu batas pengendali bawah.

Selama titik – titik sampel terletak di dalam batas – batas pengendali, proses dianggap dalam keadaan terkendali, dan tidak perlu tindakan apapun. Jika titik berada di luar batas pengendali, diinterpretasikan sebagai proses tak terkendali dan diperlukan tindakan penyelidikan dan perbaikan untuk mendapatkan dan menyingkirkan sebab atau sebab – sebab tersangka yang menyebabkan tingkah laku itu[2].

2.3. *Internet of Things*

Internet of Things adalah suatu konsep dimana objek tertentu punya kemampuan untuk mengirim data lewat jaringan tanpa

memerlukan adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer.

Internet of Things lebih sering disebut dengan singkatannya yaitu IoT. IoT ini sudah berkembang pesat mulai dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan juga Internet.

IoT ini juga kerap diidentifikasi dengan RFID sebagai metode komunikasi. Walaupun begitu, IoT juga bisa mencakup teknologi-teknologi sensor lainnya, semacam teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita.

Adapun kemampuannya bermacam-macam contohnya dalam berbagi data, menjadi remote control, dan masih banyak lagi yang lainnya. Sebenarnya fungsinya termasuk juga diterapkan ke benda yang ada di dunia nyata, di sekitar kita. Contohnya adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif.

2.3.1. Modular PC Box

Modular PC Box atau yang lebih dikenal dengan IoT Box digunakan sebagai otak dari *Internet of Things* sendiri. Semua data-data dari perangkat yang terintegrasi dengan IoT akan diolah di IoT Box ini. Tidak hanya itu, IoT Box sendiri dapat digunakan untuk mendapatkan data secara langsung, dan banyak cara untuk mendapatkan data tersebut. Dapat berupa membaca *datalog* pada mesin, membaca data dengan komunikasi serial, serta dapat membaca langsung *address* dari PLC.

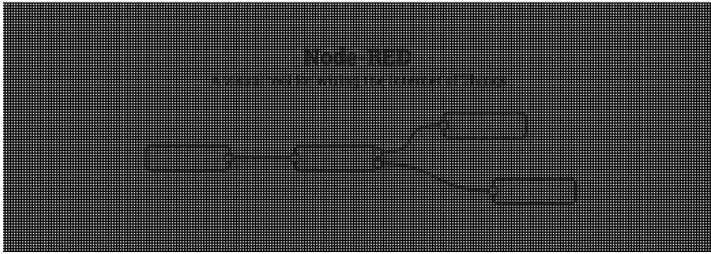


Gambar 2.3 Modular PC Box[3]

2.3.2. Node-Red

Node-RED adalah sebuah tool berbasis browser untuk membuat aplikasi *Internet of Things* (IoT) yang mana lingkungan pemrograman

visualnya mempermudah penggunaannya untuk membuat aplikasi sebagai “*flow*”. Ketahui lebih lanjut tentang Node-RED dan pelajari bagaimana kamu dapat membuat aplikasi IoT menggunakan flow.



Gambar 2.4 Flow Node-Red

Node-RED mengambil jalur alternatif tersebut untuk pengembangan software. Pertama, ia adalah bahasa pemrograman visual. Daripada membuat aplikasi sebagai barisan kodingan, Node-RED fokus ke program sebagai flow.

Flow ini terbentuk dari node yang saling berhubungan, di mana tiap node akan melakukan tugas tertentu. Walaupun Node-RED didesain untuk Internet of Things (IoT), ia juga dapat digunakan untuk keperluan umum dan untuk berbagai macam jenis aplikasi. Tujuan Node-RED adalah sebagai tool pemrograman IoT untuk keperluan umum, yang telah berkembang penggunaannya karena kesederhanaannya dan kekuatannya untuk berbagai jenis aplikasi.

2.3.3. Augmented Operator Advisor

Augmented Operator Advisor (AOA) adalah kombinasi kontel-ektual dan informasi dinamis lokal untuk pengguna *mobile*, yang memungkinkan pengguna memiliki pengalaman dari perpaduan kehidupan nyata dengan benda-benda visual.

Augmented Operator Advisor mengambil informasi secara real-time dimana-pun, dan kapan-pun dibutuhkan. Aplikasi yang mengimprovisasi efisiensi pengoperasian dengan *augmented reality*, sehingga operator dapat melihat dan memeriksa data saat itu juga pada virtual objek.



Gambar 2.5 Augmented Operator Advisor

Augmented Operator Advisor atau AOA merupakan salah satu dari 3 *IoT Implementation* di Schneider. AOA ini berkonsep dengan memindai barcode yang tertempel pada mesin atau gambar dari mesin.

AOA dapat menampilkan kondisi dari mesin secara *real-time* dan status dari hasil produksi. Selain berupa data, AOA juga dapat menampilkan dokumen-dokumen untuk menunjang produksi, seperti jika mesin terjadi *machine breakdown* atau ada masalah pada mesin, maka *engineer* atau *technician* tidak perlu membuka buku manual user lagi.

2.3.4. Machine Advisor



Gambar 2.6 Machine Advisor

Machine Advisor merupakan salah satu dari 3 *IoT Implementation* di Schneider. *Machine Advisor* ini menampilkan data-data mesin yang ada di produksi Schneider. Pada machine advisor ini terdapat banyak sekali mesin yang ada di Schneider yang digunakan untuk produksi, dan tidak hanya di Indonesia saja, karena sesuai dengan tujuan machine

advisor ini merupakan selain melihat kondisi mesin secara *real-time*, *machine advisor* juga dapat untuk mengkomparasi antara mesin di daerah satu dengan daerah yang lainnya.

Selain melihat kondisi mesin secara *real-time*, di *machine advisor* ini juga dapat melihat bagaimana mesin tersebut secara struktural.

2.3.5. Aveva Insight



Gambar 2.7 Aveva Insight

Aveva Insight atau wonderware online merupakan salah satu implementasi *Internet of Things* di Schneider. Sama halnya dengan implementasi yang lainnya, Aveva insight bertujuan untuk menampilkan data-data atau kondisi pada mesin secara *real-time*, namun berbasis smartphone. Untuk melihat data-data atau kondisi mesin secara real-time dapat dilihat di aplikasi pada smartphone.

2.4. Sensor CL110

CL110 merupakan salah satu sensor dari Schneider. CL110 merupakan sensor suhu dan kelembaban. Selain berfungsi sebagai sensor, CL110 ini sendiri sudah sekaligus menjadi transmitternya. CL110 ini sudah terfasilitasi dengan wireless, kemudian sensor suhu ini juga dapat menerima suhu hingga -25°C s/d 130°C , serta jangkauan pemancaran sinyalnya hingga 20 meter[4].



Gambar 2.8 Sensor CL110[4]

2.5. ZBRN32

ZBRN32 ini merupakan *receiver* dari sensor suhu dan kelembaban atau CL110. Tidak hanya itu, ZBRN32 ini dapat menangkap sinyal dari perangkat-perangkat yang lainnya. ZBRN32 ini mempunyai port serial yang digunakan untuk mengirimkan data yang didapat dan port LAN sebagai sarana konektivitas ke jaringan yang tersedia[5].



Gambar 2.9 ZBRN32[5]

2.6. PowerTag



Gambar 2.10 PowerTag[6]

PowerTag merupakan sebuah sensor untuk membaca yang berhubungan dengan energi. Energi ini sendiri banyak sekali macamnya, tegangan, arus, daya konsumsi, dan lain-lain. PowerTag sendiri dapat dipasang dengan 2 cara tergantung apa yang akan lebih diutamakan, yaitu pada incoming energinya atau *outgoing* energi tersebut. PowerTag ini dipasang menempel pada MCB, jika akan dipasang *incoming* maka PowerTag dipasang diatas MCB atau sebelum sumber masuk ke MCB,

dan jika dipasang secara outgoing maka PowerTag dipasang dibawah MCB atau setelah MCB menerima sumber aliran listrik.

PowerTag sendiri juga sebagai *transmitter* dari sensornya dan sudah terfasilitasi dengan adanya wireless pada PowerTag itu sendiri yang akan mengirimkan sinyal[6].

2.7. Acti 9 Smartlink SI D

Acti 9 Smartlink SI D atau Acti 9 merupakan receiver dari sensor energy atau PowerTag. Setelah mendapatkan data dari PowerTag maka Acti9 akan mengirimkan data tersebut ke IoT Box sebagai pengolah data dengan menggunakan konektivitas internet dengan menggunakan LAN sebagai sarananya[7].



Gambar 2.11 Acti 9 Smartlink SI D[7]

2.8. PLC TM241CEC24R

Programmable Logic Control merupakan suatu bentuk khusus pengontrol berbasis microprocessor yang memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi – instruksi dan untuk mengimplementasikan fungsi – fungsi semisal logika, *sequencing*, pewaktuan (*timing*), pencacahan (*counting*) dan aritmatika guna mengontrol mesin-mesin dan proses-proses serta dirancang untuk dioperasikan oleh para insinyur yang hanya memiliki sedikit pengetahuan mengenai komputer dan bahasa pemrograman. Sebagian besar industri telah menerapkan sistem otomatis dalam proses produksi. Pada umumnya sistem otomatis yang diterapkan terdiri atas dua metode yaitu otomatisasi berbasis kontrol relay dan otomatisasi berbasis *Programmable Logic Control* (PLC). Otomatisasi berbasis relay banyak digunakan pada mesin-mesin yang memiliki urutan-urutan (sekuens) yang sederhana, sedangkan otomatisasi PLC dapat memiliki sekuens yang lebih kompleks dari relay. Otomatisasi berbasis PLC dapat

diintegrasikan dengan sistem *monitoring*. Sistem *monitoring* berbasis PLC adalah suatu sistem yang berguna untuk mengontrol proses suatu kerja tertentu., dimana parameter atau inputan data diambil dan diolah oleh *Personal Computer* (PC) dan melalui sebuah program tertentu.



Gambar 2.12 PLC TM241CEC24R[8]

Sesuai namanya, PLC dapat dengan mudah diprogram ulang. Keunggulan PLC dibandingkan dengan sistem konvensional antara lain:

1. Relatif mudah untuk melakukan perubahan pada strategi kendali yang akan diterapkan, karena logika kendali yang digunakan diwujudkan dalam bentuk perangkat lunak.
2. Jumlah relay yang diperlukan dapat dikurangi sesuai dengan jumlah input maupun output yang diperlukan. Lebih mudah untuk proses instalasinya karena pengkabelan lebih sederhana.
3. Lebih mudah dalam menemukan kesalahan dan kerusakan, karena memiliki fasilitas self – diagnosis.
4. Tahan terhadap temperature tinggi, tekanan tinggi dan kelembaban yang tinggi apabila dipakai secara terus - menerus, dan ini banyak di jumpai padalingkungan industri[8].

2.9. Keyence GP-M010

Keyence GP-M010 merupakan suatu sensor *vacuum* (daya sedot) atau tekanan negatif. Komunikasi pada sensor ini menggunakan kabel konektor M12 dengan 4 kaki



Gambar 2.13 Keyence GP-M010[9]

Adapun spesifikasi dari sensor tersebut, sebagai berikut:

1. Rentang kekuatan -14.5 sampai +145.0 PSI, atau -0.1 sam-pai +1 MPa.
2. Objek pada sensor ini dapat berupa angin dan air.
3. Dapat bertahan pada suhu -20°C sampai +100°C.
4. Sumber tegangan 10-30 VDC[9]

2.10. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi *Windows*, ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Compiler yang dimasukkan ke dalam paket Visual Studio antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*[10].

Pada Proyek Akhir ini visual studio digunakan sebagai *user interface* (tampilan) dimana nantinya tampilan tersebut digunakan sebagai tampilan untuk memberikan informasi mengenai data dari masing-masing parameter dan nilai serta grafik dari variabel-variabel *statistical process control*. Logo visual studio dapat dilihat seperti pada Gambar 2.15.



Gambar 2.14 Visual Studio.[10]

2.11. Microsoft SQL Server



Gambar 2.15 Microsoft SQL Server.[11]

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data *mirroring* dan *clustering*[11].

BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan pembuatan sistem perangkat keras (*Hardware*) dan perangkat lunak (*Software*) dimana rancangan tersebut akan menghasilkan suatu sistem dengan kemampuan untuk menganalisa dan memonitoring suatu kinerja produksi pada industri. Sistem ini menggabungkan antara konsep *Statistical Process Control* dengan konsep *Internet of Things*. Berikut adalah perancangan sistem yang digunakan:

3.1. Prinsip Kerja

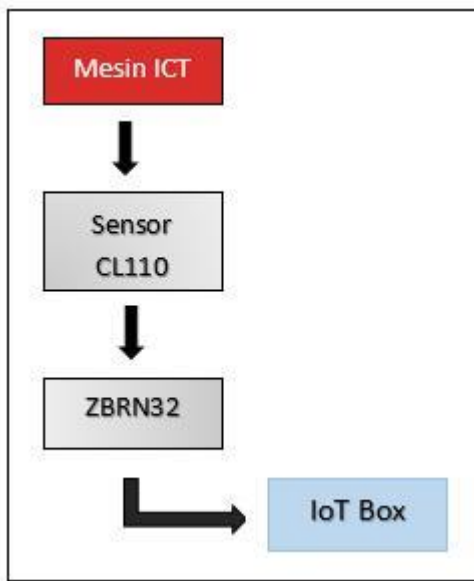
Menurut ilustrasi pada gambar diatas prinsip kerja dari sistem yang akan dirancang merupakan sebuah sistem pengawasan dimana dalam sistem ini mengawasi 4 parameter pada mesin produksi. Parameter-parameter tersebut diambil datanya menggunakan sensor-sensor. Sensor CL110 untuk mendapatkan parameter suhu dan kelembaban, kemudian setelah mendapatkan data atau informasi dari mesin, informasi tersebut dikirimkan melalui sinyal dan diterima ZBRN32 sebagai *receiver*. Sensor tegangan yaitu PowerTag ditempatkan pada MCB untuk mesinnya, tata letak PowerTag ini diletakkan dibawah MCB karena akan mengawasi tegangan *outgoing*-nya. Informasi yang sudah diterima PowerTag akan dikirimkan menggunakan sinyal dan akan diterima Acti9 sebagai *receiver*-nya. Keyence GP-M010 merupakan sensor tekanan, sensor ini diletakkan pada pipa saluran angin pada mesin, informasi dari sensor tersebut diambil menggunakan PLC. Setelah data semua didapat dan dikirima ke IoT Box, data-data tersebut direkam juga di database SQL Server, kemudian data-data tersebut diolah dan ditampilkan pada aplikasi *interface*.

3.2. Perancangan Sensor Suhu Dan Kelembaban

Rangkaian sistem sensor *temperature* dan *humidity* atau suhu dan kelembaban ini meliputi berbagai macam alat. Pada implementasi ini sensor suhu dan kelembaban atau CL110 ditempelkan pada bagian dari mesin ICT, karena CL110 ini mempunyai magnet pada bagian bawahnya sehingga harus ditempelkan pada bagian mesin ICT yang berbahan besi dan mempunyai suhu yang berpengaruh pada mesin. Lalu,

sebagai *receiver* dari CL110, ZBRN32 diletakkan pada panel yang terjangkau pada area sinyal daripada CL110.

Setelah mendapatkan sinyal data dari CL110, IoT Box akan membacanya melalui ZBRN32 dengan menggunakan komunikasi serial port COM. Kemudian, data yang sudah terbaca akan diolah dengan aplikasi Node-Red yang kemudian akan dikirim ke database dan *EcoStuxure* untuk pengembangan lebih lanjut.



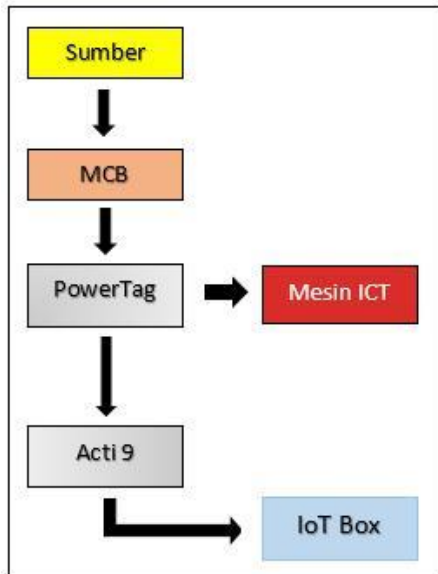
Gambar 3.1 Rangkaian Sensor Suhu Dan Kelembaban

3.3. Perancangan Sensor Energi

Pada perangkaian sistem sensor *energy* ini meliputi banyak *device* yang terlibat. Penempatan sensor *energy* atau PowerTag ini langsung pada MCB. Penempatannya pun bisa dilakukan dengan 2 posisi, yaitu sebelum atau diatas MCB dan sesudah atau dibawah MCB. Penempatan ini berdasarkan apa yang ingin dilakukan pengawasan, jika sumber *incoming* yang ingin dilakukan pengawasan maka ditempatkan diatas MCB atau tegangan sebelum masuk MCB, jika sumber *outgoing*-nya maka diletakkan sesudah sumber masuk ke MCB atau dibawah MCB

yang akan disambungkan pada power dari mesin. Pada ICT ini membutuhkan pengawasan pada *outgoing*.

Setelah mendapatkan aliran energy, PowerTag akan mengirimkan sinyal data yang akan diterima oleh *receiver*-nya yaitu Acti 9. Acti 9 terpasang pada panel dan berbeda dengan ZBRN32, untuk komunikasi Acti 9 ini menggunakan Ethernet yang terhubung pada jaringan. Kemudian, IoT Box akan mengambil data yang sudah diterima Acti 9 menggunakan Node-Red, dan untuk komunikasi pengambilan data ini digunakan node Modbus. Data diambil sesuai dengan kebutuhan, parameter apa yang harus dilakukan pengawasan. Data akan disimpan pada database dan akan diolah untuk diterapkan pada *EcoStuxure*.



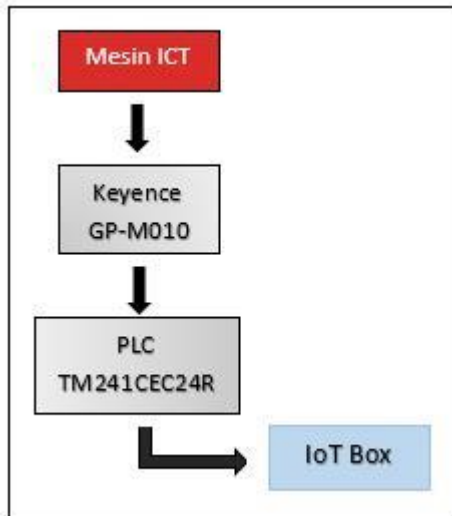
Gambar 3.2 Rangkaian Sensor *Energy*

3.4 Perancangan Sensor Tekanan

Sensor Keyence GP-M010 merupakan sensor tekanan untuk objek angin dan air. Tidak hanya dapat membaca tekanan positif, sensor ini juga mampu membaca tekanan negatif atau yang biasa dikenal dengan tekanan daya sedot atau *vacuum*. Pada mesin ICT ini membutuhkan daya sedot untuk menekan *fixture* sehingga komponen-komponen yang

ada pada *fixture* tersebut menempel dengan PCB yang akan dilakukan pengetesan.

Rangkaian sistem sensor tekanan ini lebih rumit daripada yang lainnya, karena harus memotong selang aliran angin agar dapat disambungkan dengan sensor tekanan. Setelah terpasang, sensor tekanan dilakukan *wiring* ke *controller* atau PLC. Karena sensor ini outputnya adalah analog, maka PLC harus ditambahkan I/O Analog agar dapat membaca data dari sensor.



Gambar 3.3 Rangkaian Sensor Tekanan

3.5. Pembuatan Aplikasi Interface

Interface pada proyek ini memerlukan 2 buah aplikasi sebagai penunjangnya. Aplikasi pertama yaitu aplikasi *data extractor* atau aplikasi pengolahan data-data dari berbagai macam sumber. Aplikasi ini mengumpulkan data dari berbagai macam sensor yang digunakan, lalu diolah agar menjadi data yang sesuai, kemudian direkam dengan cara dikirim ke database SQL Server. Aplikasi *extractor* ini ditanamkan pada setiap mesin ICT.

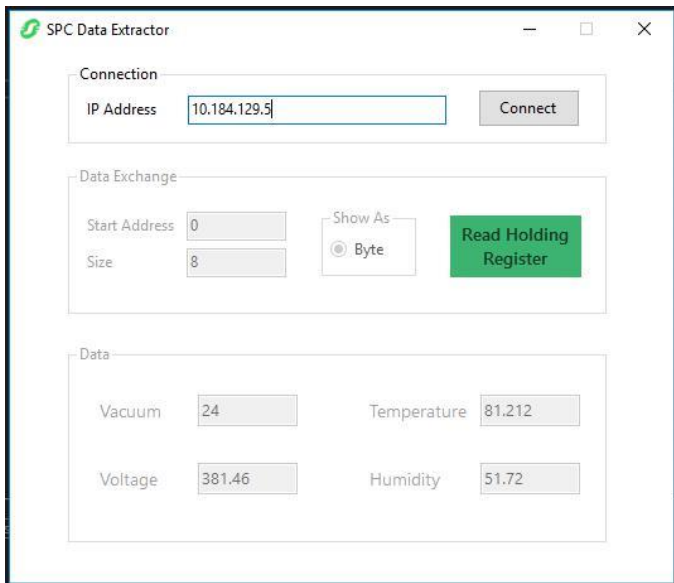
Aplikasi kedua yaitu halaman *interface* itu sendiri. Aplikasi ini menampilkan data-data dari berbagai macam parameter yang diambil

pada mesin ICT dan terdapat grafik sebagai penunjuk dari *Statistical Process Control*.

3.5.1. Aplikasi Data Extractor

Aplikasi *Data Extractor* ini berfungsi untuk mengumpulkan data-data dari bermacam-macam parameter penunjang proyek ini. Data dari sensor suhu dan kelembaban diambil dari datalog yang sudah dibuat oleh IoT Box, lalu data dari sensor energi diambil dengan aplikasi ini menggunakan sistem Modbus dari Acti9, kemudian data dari sensor tekanan diambil langsung pada Modbus dari PLC karena data dari sensor tekanan ini dikirimkan ke PLC sebagai kontrolernya.

Setelah data terkumpul semua, aplikasi *extractor* ini mengirimkan data-data tersebut ke database SQL Server agar semua data dapat terekam dan mudah untuk mengolah datanya kembali.



Gambar 3.4 Aplikasi *Data Extractor*

3.5.2. Aplikasi Interface

Interface ini digunakan untuk melihat data dari parameter yang sudah diambil oleh aplikasi *extractor*. Pada aplikasi ini memuat data dan

juga grafik yang menunjukkan variabel-variabel untuk *statistical process control*. Data-data tersebut diambil dari database SQL Server yang sudah terkoneksi dengan aplikasi ini.

Di dalam aplikasi ini terdapat 5 *form* yang terdiri dari halaman utama, halaman untuk parameter suhu, halaman untuk parameter kelembaban, halaman untuk parameter tekanan, dan halaman untuk parameter tegangan.

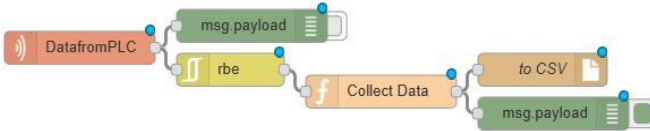
Untuk halaman utamanya menampilkan data-data dari 4 parameter secara *real-time* beserta dengan beberapa variabel *statistical process control*. Pada halaman ini juga dapat melihat beberapa data dari mesin yang sudah diimplementasikan sistem ini, seperti mesin ICT yang mempunyai 4 mesin yang sama.

3.5.3. Perancangan Node-Red

Saat ini, *real-time monitoring* ini sudah banyak diimplementasikan pada mesin-mesin produksi.

Sensor untuk mendapatkan informasi hasil produksi akan mengirimkan data menuju suatu *software*, begitu pula dengan sensor suhu dan kelembaban akan mengirimkan data menuju ZBRN32 (*receiver*) dan juga sensor energi mengirimkan data menuju Acti9 (*receiver*). Lalu ketiga data tersebut akan dibaca oleh IoT Box dengan aplikasi Node-Red.

Pada Node-Red, akan membaca dari Datalog dan Acti9 dengan menyambungkan IP Addressnya dengan ModbusTCP.



Gambar 3.5 Flow Node-Red Pengambilan Data dari Datalog

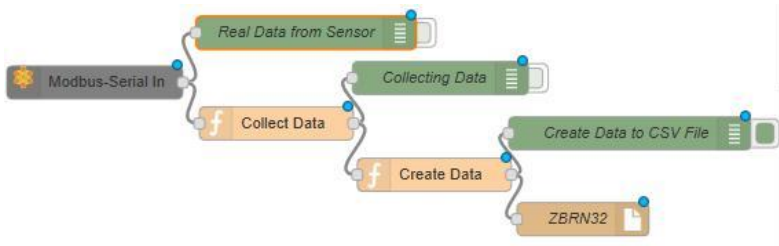


Gambar 3.6 Flow Node-Red Pengambilan Data dari SQL



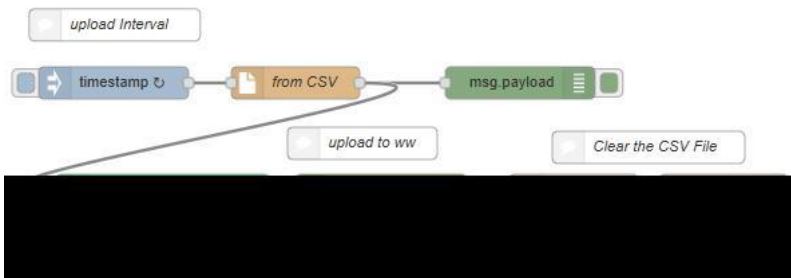
Gambar 3.7 Flow Node-Red Acti9 3 Phase

Lalu membaca data dari ZBRN32 dengan menyambungkan COMPort pada Modbus Serial.



Gambar 3.8 Flow Node-Red pengambilan data dari ZBRN32

Setelah data di proses dan diubah menjadi file CSV. File tersebut akan di *upload* menuju Wonderware Online dengan Node-Red. Namun sebelumnya, kita harus membuat data source terlebih dahulu pada wonderware online. Guna mendapatkan Bearer ID, dimana Bearer ID ini digunakan sebagai ID untuk lokasi penyimpanan file yang telah diupload.



Gambar 3.9 Flow Node-Red unggah file ke Aveva Insight

Data yang berhasil terupload pada Wonderware Online, secara langsung akan divisualisasikan menjadi bentuk chart.

3.5.4. Augmented Operator Advisor

Augmented Operator Advisor (AOA) adalah kombinasi kontekstual dan informasi dinamis lokal untuk pengguna *mobile*, yang memungkinkan pengguna memiliki pengalaman dari perpaduan kehidupan nyata dengan benda-benda visual.

Augmented Operator Advisor mengambil informasi secara real-time dimana-pun, dan kapan-pun dibutuhkan. Aplikasi yang mengimprovisasi efisiensi pengoperasian dengan *augmented reality*, sehingga operator dapat melihat dan memeriksa data saat itu juga pada virtual objek.

Augmented Operator Advisor atau AOA merupakan salah satu dari 3 *IoT Implementation* di Schneider. AOA ini berkonsep dengan memindai barcode yang tertempel pada mesin atau gambar dari mesin.

AOA dapat menampilkan kondisi dari mesin secara real-time dan status dari hasil produksi. Selain berupa data, AOA juga dapat menampilkan dokumen-dokumen untuk menunjang produksi, seperti jika mesin terjadi breakdown, maka *engineer* atau *technician* tidak perlu membuka buku *manual user* lagi.



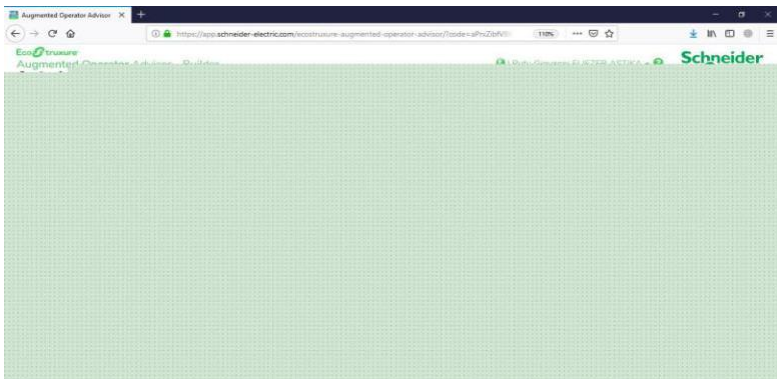
Gambar 3.10 Augmented Operator Advisor

Untuk membuat atau mengatur tampilan *Point Of Interest* (POI) pada AOA diperlukan *AOA Builder*. Di dalam *AOA builder* kita dapat mengatur apa saja data yang akan ditampilkan, dan mengatur berapa POI yang akan ditampilkan dan apa saja yang dimasukkan ke dalam masing-masing POI.



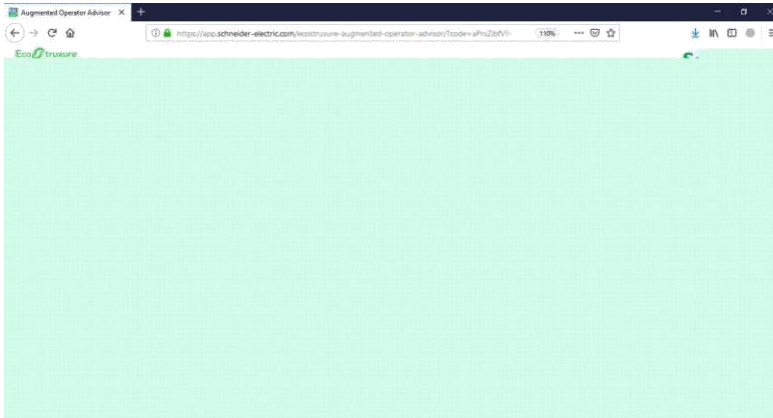
Gambar 3.11 Augmented Operator Advisor Scene Builder

Untuk membuatnya, siapkan gambar suatu objek yang akan di tampilkan datanya. Jika ingin menggunakan barcode sebagai objek untuk memindainya, tempelkan *barcode* pada objek/mesin. Kemudian pilih gambar yang sesuai dan pilih tipe pembacaan, *tag recognition* atau *image recognition*.



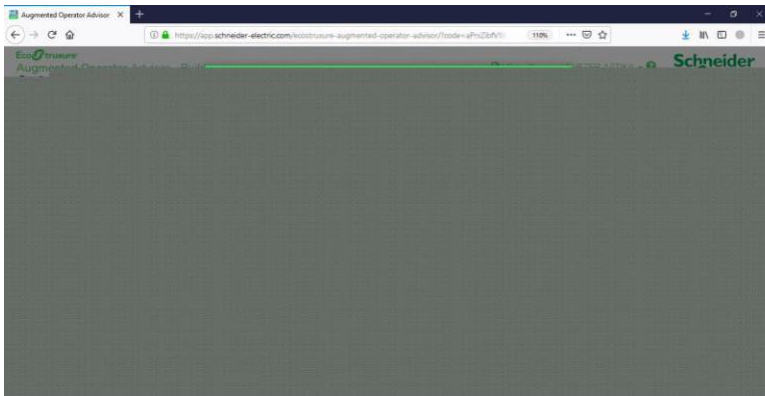
Gambar 3.12 Augmented Operator Advisor POI Builder

Untuk menampilkan data-data, gambar yang sudah tertampil diberi *Point of Interest*, kemudian POI tersebut diisi sesuai dengan data yang ingin ditampilkan.



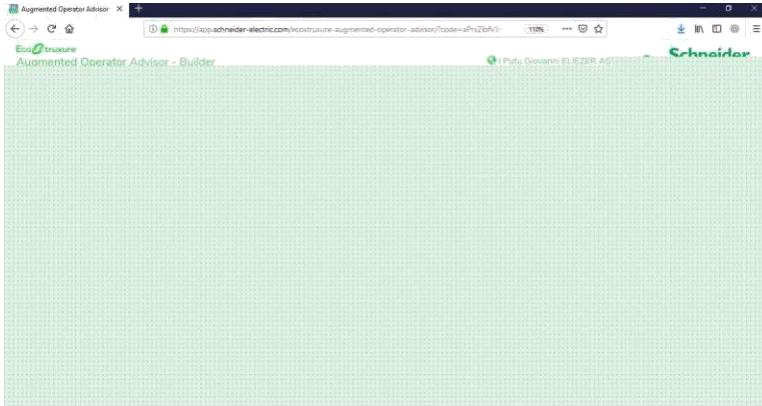
Gambar 3.13 Augmented Operator Advisor Variabel Builder

Jika ingin menampilkan data secara real-time maka pilih read variable dan masukkan nama variable yang akan ditampilkan sesuai dengan *flow* pada Node-Red, dan jika ingin menampilkan dokumen maka harus meng-*upload* dokumen terlebih dahulu ke AOA builder dan pilih dokumen tersebut.



Gambar 3.14 Augmented Operator Advisor Project Builder

Setelah selesai membuat tampilannya kemudian project tersebut di build.



Gambar 3.15 Augmented Operator Advisor Builder

Kemudian, setelah build selesai, *download* dan tempatkan hasil project tadi di server agar project yang di *download* terbaca.

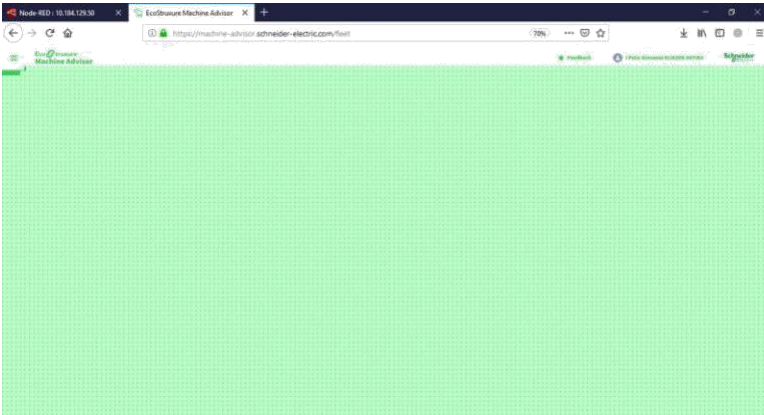
3.5.5. Machine Advisor

Machine Advisor merupakan salah satu dari 3 *IoT Implementation* di Schneider. Machine Advisor ini menampilkan data-data mesin yang ada di produksi Schneider. Pada machine advisor ini terdapat banyak sekali mesin yang ada di Schneider yang digunakan untuk produksi, dan tidak hanya di Indonesia saja, karena sesuai dengan tujuan machine advisor ini merupakan selain melihat kondisi mesin secara *real-time*, machine advisor juga dapat untuk mengkomparasi antara mesin di daerah satu dengan daerah yang lainnya.

Selain melihat kondisi mesin secara *real-time*, di machine advisor ini juga dapat melihat bagaimana mesin tersebut secara struktural.

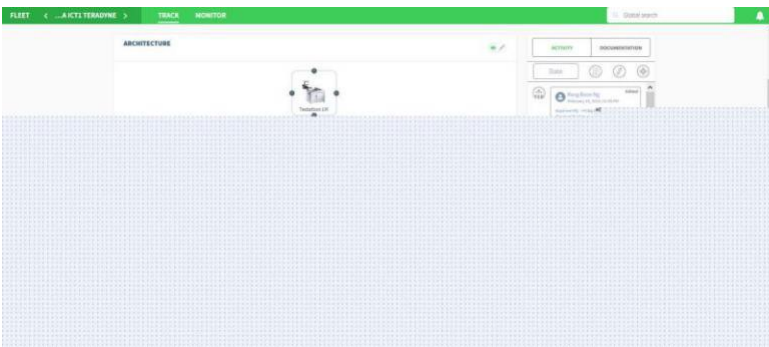


Gambar 3.16 Machine Advisor



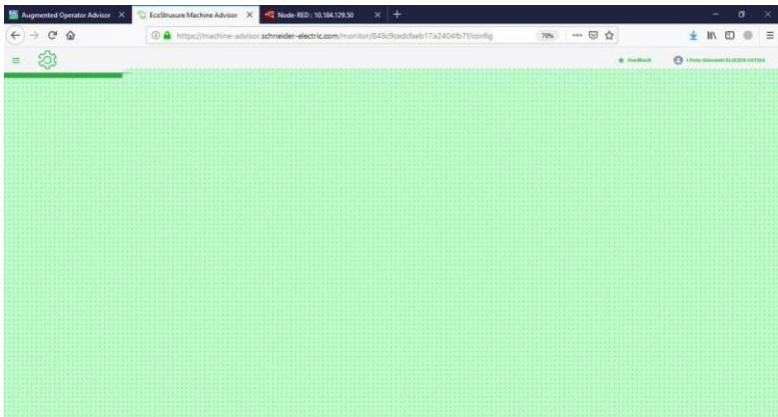
Gambar 3.17 Fleet Machine Advisor

Machine Advisor yang merupakan salah satu *Internet of Things* implementasi memiliki kemampuan untuk *monitoring* kondisi mesin dan produk di *in house* maupun di berbagai macam daerah atau negara lain. Seperti pada gambar di atas pada *dashboard Machine Advisor* menunjukkan tidak hanya mesin di Indonesia saja yang tertampil, ada dari beberapa negara lain sehingga dapat mengkomparasi mesin-mesinnya.



Gambar 3.18 Architecture Machine Advisor

Selain untuk *monitoring*, *Machine Advisor* berguna untuk melihat structural konsep dari mesin. Dan dapat ditambahkan *task management* yang berfungsi untuk membuat jadwal jika ingin dilakukan perawatan berkala. Selain itu juga dapat dilampirkan dokumen-dokumen yang berkaitan dengan mesin tersebut.



Gambar 3.19 Config Machine Advisor

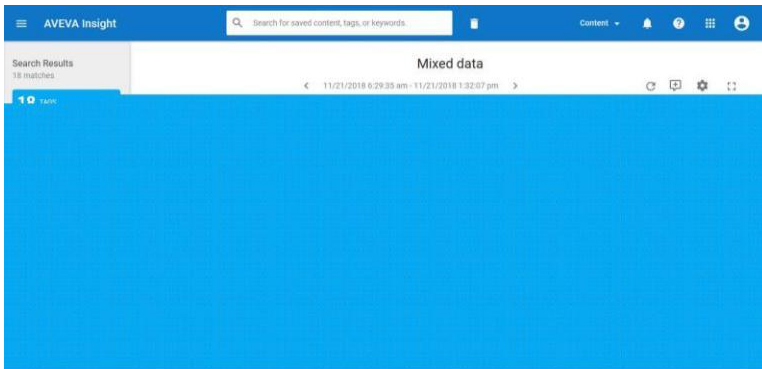
Agar data terbaca oleh *Machine Advisor*, ada beberapa konfigurasi yang harus disesuaikan pada saat perancangan Node-Red. Berikut merupakan konfigurasi yang diperlukan:

- a. *Mode : Connect Your Machine*
- b. *Gateway : Custom*
- c. *Format : Tango*
- d. *Transport : Https*

Kemudian *custom header2* dan server disalin di *function* Node-Red agar data dari Node-Red terkirim ke *Machine Advisor*.

Setelah data dari Node-Red terkirim dan terbaca di *Machine Advisor*, desain *dashboard* sesuai dengan keinginan dan sesuai dengan data-data yang ada. Ada banyak pilihan widget yang tersedia, gunakan sesuai dengan keperluan.

3.5.6. Aveva Insight



Gambar 3.20 Dashboard Aveva Insight

Perancangan aveva ini membutuhkan datalog dari mesin yang akan ditampilkan dan datalog tersebut dapat diolah di Node-Red. Kemudian dikirimkan melalui node khusus Aveva lalu setelah data masuk ke Aveva akan tertampil pada gambar diatas.

BAB IV PENGUJIAN DAN ANALISA

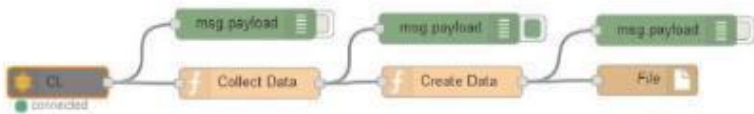
Pengujian yang dilakukan pada Proyek Akhir ini bertujuan untuk mengetahui tingkat keberhasilan metode yang digunakan pada sistem. Pengujian dimulai dengan pengujian komunikasi antara sensor dengan *receiver*-nya dan dikirimkan ke IoT Box, kemudian pengujian pengambilan dan pengiriman data dari sensor.

4.1. Pengujian Komunikasi Sistem

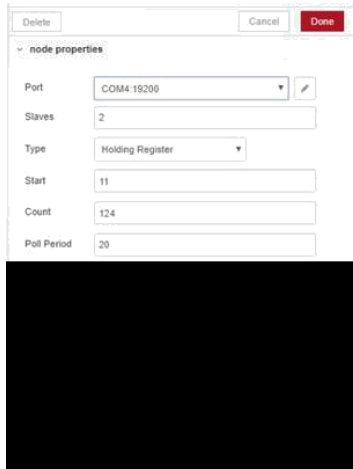
Komunikasi sistem yang dilakukan ada 2 jenis berdasarkan dari sensornya masing-masing, yaitu untuk sensor suhu dan kelembaban menggunakan komunikasi *serial com port*, sedangkan untuk sensor tekanan dan tegangan menggunakan Modbus sebagai komunikasi pengambilan datanya.

4.1.1. Sensor Suhu Dan Kelembaban

Komunikasi sistem yang dilakukan untuk mendapatkan data atau informasi dari sensor suhu dan kelembaban yaitu menggunakan komunikasi *serial com port*. CL110 yang merupakan sensor suhu dan kelembaban mendapatkan informasi data dari objek kemudian dikirimkan ke ZBRN32 sebagai *receiver* menggunakan sinyal, kemudian ZBRN32 mengirimkan informasi data ke IoT Box menggunakan komunikasi *serial com port*. Pembacaan data dari *serial com port* menggunakan *node-red* yang terdapat pada IoT Box.



Gambar 4.1 Flow Node-Red Pengambilan Data CL110



Gambar 4.2 Konfigurasi Pada Node Serial

Data yang didapat dari node serial tersebut masih dalam bentuk ASCII, sehingga data tersebut perlu dirubah menjadi bilangan data yang diinginkan. Pengolahan dan perubahan data tersebut dilakukan menggunakan node *function*. Data yang sudah dirubah akan menampilkan data sesuai dengan format data yang diinginkan, seperti data suhu akan ditampilkan dalam bentuk derajat celcius, dan data kelembaban akan ditampilkan dalam bentuk persentase. Berikut hasil pengujian yang didapatkan :



Gambar 4.3 Output pengambilan data sensor suhu dan kelembaban

Dari keluaran pada gambar diatas menunjukkan komunikasi yang dilakukan antara sensor dengan *receiver*, dan *receiver* dengan IoT Box maupun perubahan format data berhasil. Pada array 0 dan genap merupakan hasil data suhu dalam bentuk derajat Fahrenheit dari beberapa mesin, kemudian pada array ganjil merupakan hasil data kelembaban dalam bentuk persentase pada beberapa mesin.

Tabel 4.1 Data Percobaan Sensor Suhu Dan Kelembaban

Tanggal	Waktu	Suhu (°F)	Kelembaban (%)
4/18/2019	07:02:14	79.70	47.37
4/18/2019	07:07:26	75.74	55.34
4/18/2019	07:12:36	73.58	56.19
4/18/2019	07:17:44	80.42	46.21
4/18/2019	07:22:14	81.13	45.77
4/18/2019	07:27:26	80.74	46.63
4/18/2019	07:32:36	78.58	49.69
4/18/2019	07:37:44	82.27	44.15
4/18/2019	07:42:14	79.69	47.41
4/18/2019	07:47:26	82.34	43.94

Berdasarkan percobaan yang sudah dilakukan dan berdasarkan data pada Tabel 4.1, terlihat bahwa perbandingan antara suhu dan kelembaban berbanding terbalik. Jika suhu menurun maka tingkat kelembaban akan meningkat, dan jika suhu naik maka tingkat kelembaban akan menurun.

4.1.2. Sensor Tegangan

Komunikasi yang digunakan untuk mendapatkan data atau informasi dari sensor tegangan atau PowerTag adalah dengan menggunakan Modbus. PowerTag diletakkan pada *output* dari MCB karena akan mengambil data tegangan *outgoing*. PowerTag menerima data kemudian dikirimkan ke Acti9 sebagai *receiver* nya, Acti9 tersebut terhubung

dengan jaringan menggunakan LAN sehingga dapat di akses oleh *device* yang terhubung pada jaringan juga.

Pengambilan data dari Acti9 ini tetap menggunakan Node-Red yang ada didalam IoT Box. Berikut merupakan flow Node-Red untuk mendapatkan data tegangan:



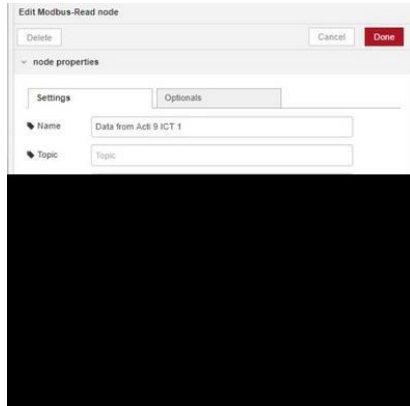
Gambar 4.4 Flow Node-Red Dengan Node Modbus

Node yang digunakan untuk mengambil data tegangan tersebut adalah node Modbus.



Gambar 4.5 Konfigurasi server IP Acti9

Pada Gambar 4.5 diatas menunjukkan konfigurasi server IP Acti9 yang akan diambil datanya.



Gambar 4.6 Konfigurasi Address

PowerTag merupakan tidak hanya sebagai sensor tegangan, melainkan sensor energi. PowerTag mampu membaca besar tegangan, arus, konsumsi sumber, energi parsial, total energi, dll. Sehingga untuk mengambil data tegangan membutuhkan address data tegangan yang sesuai pada *datasheet* dari Acti9. Gambar 4.6 diatas merupakan konfigurasi *address* dari tegangan.

Tabel 4.2 Data Percobaan Sensor Tegangan Dan Tekanan

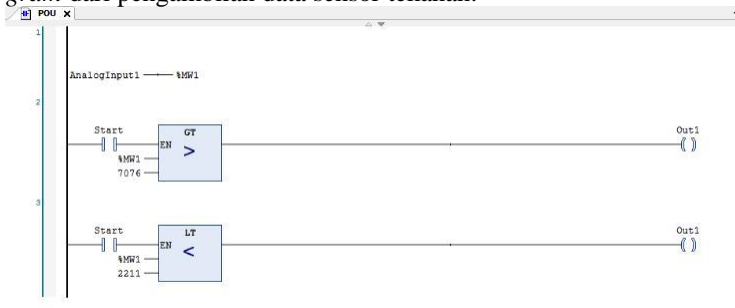
Tanggal	Waktu	Tegangan (Volt)	Tekanan (cm Hg)
4/17/2019	12:12:12	381.21	44.67
4/17/2019	12:16:25	381.62	45.19
4/17/2019	12:20:35	378.58	39.56
4/17/2019	12:24:43	378.10	39.15
4/17/2019	12:30:12	380.95	43.72
4/17/2019	12:33:25	378.44	39.34

4/17/2019	12:38:35	379.81	42.21
4/17/2019	12:42:43	381.29	44.72
4/17/2019	12:47:12	380.48	43.03
4/17/2019	12:52:25	379.66	41.65

Berdasarkan pengujian yang sudah dilakukan mendapatkan data-data dari parameter tegangan dan parameter tekanan seperti pada Tabel 4.2 diatas. Dilihat dari data-data pada table diatas terlihat bahwa jika tekanan yang dibutuhkan tinggi maka tegangan yang diperlukan akan meningkat, dan jika tekanan melemah maka tegangan yang dibutuhkan akan menurun.

4.1.3.Sensor Tekanan

Komunikasi yang digunakan untuk mendapatkan data dari sensor tegangan adalah Modbus karena dalam sistem ini menggunakan PLC sebagai kontroler dari sensor tersebut. Berikut merupakan *ladder diagram* dari pengambilan data sensor tekanan:



Gambar 4.7 Ladder Diagram

Keluaran dari sensor tekanan ini adalah keluaran analog, dan akan menampilkan data berupa arus. Pada *ladder* diatas terdapat *upper limit* dan *lower limit* yang berfungsi sebagai variabel pengontrol. Jika data yang keluar melebihi *limit* yang sudah ditentukan maka *output* akan menyala, yang akan mengirimkan data ke database bahwa terjadi masalah pada tekanannya.

Data yang terbaca pada PLC adalah berupa arus, sehingga harus dilakukan analisa agar mendapatkan data tekanan yang sesuai pada sensor dan mesin. Berikut adalah analisa dari data pada Tabel 4.3:

Tabel 4.3 Data Percobaan Sensor Tekanan

Percobaan	Data Arus (mA)	Data Sensor (Bar)
1	3098	0.467
2	3536	0.533
3	3364	0.507
4	2985	0.450
5	3244	0.489
6	3403	0.513
7	3350	0.505
8	3177	0.479
9	3649	0.550
10	3768	0.568

Total data tekanan dari 10 percobaan adalah = 5,061 Bar. Yang berarti mempunyai rata-rata = 0,506 Bar. Sedangkan, Total data arus pada 10 percobaan adalah = 33574 mA, dengan rata-rata 3357,4 mA. Jadi, dari data pada Tabel 4.3 dapat diambil nilai pengali-nya, dengan cara sebagai berikut :

$$\frac{0,506}{3357,4}$$

$$= \frac{0,506}{3357,4}$$

$$= 0,00015071186 \text{ atau}$$

$$= 15,07 \times 10^{-5}$$

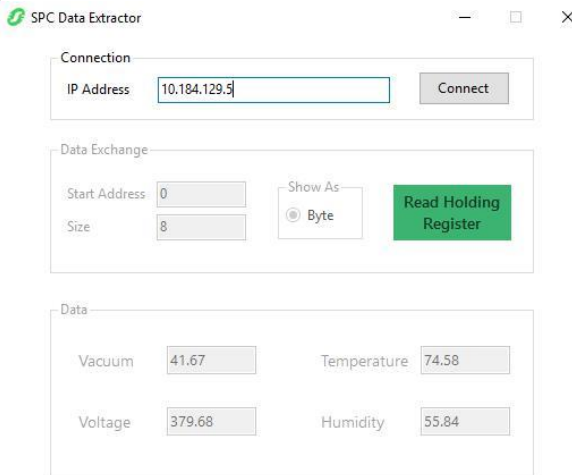
Sehingga dapat diartikan, untuk mendapatkan nilai tekanan dalam bentuk satuan Bar, maka arus yang diterima oleh I/O Analog PLC dikali dengan pengalinya yaitu $15,07 \times 10^{-5}$.

4.2. Pengujian Sistem

Pengujian Sistem ini berupa pengujian terhadap aplikasi *interface* dan aplikasi data *extractor* yang dibuat dengan menggunakan Visual Studio selain itu pengujian terhadap aplikasi *ecostruxure* dari Schneider. Pengujian ini dilakukan dengan menguji konektivitas antara aplikasi tersebut dengan sensor-sensor, dan dengan Microsoft SQL Server.

4.2.1. Pengujian Aplikasi Data *Extractor*

Pengujian aplikasi data *extractor* dilakukan dengan menguji pengambilan data dari macam-macam sumber data sensor, kemudian dikirimkan ke database SQL Server agar semua data tercatat dan tersimpan dengan baik dan aman.



The screenshot shows the SPC Data Extractor application window. It has three main sections: Connection, Data Exchange, and Data. The Connection section has an IP Address field with the value '10.184.129.5' and a 'Connect' button. The Data Exchange section has a 'Start Address' field with '0', a 'Size' field with '8', a 'Show As' dropdown menu with 'Byte' selected, and a green 'Read Holding Register' button. The Data section displays four sensor readings: Vacuum (41.67), Temperature (74.58), Voltage (379.68), and Humidity (55.84).

Gambar 4.8 Aplikasi Data *Extractor*

Pada pengujian aplikasi data *extractor* berhasil mendapatkan data dari berbagai sensor. Pada Gambar 4.8 terlihat bahwa data pada parameter tekanan atau tekanan sebesar 41.67 cmHg, data pada parameter tegangan sebesar 379.68 Volt, lalu data pada parameter suhu sebesar 74.58 Fahrenheit, kemudian data pada parameter kelembaban sebesar 55.84%. Pada *textbox IP Address* merupakan alamat dari PLC, yang digunakan untuk mengambil data dari *vacuum* atau tekanan. Kemudian aplikasi ini akan mengambil data dengan interval yang sudah ditentukan.

Setelah mendapatkan data dari masing-masing sensor, data tersebut dikirimkan ke *database* SQL Server.



Gambar 4.9 Database SQL Server

Berdasarkan Gambar diatas terlihat bahwa data-data yang diambil oleh aplikasi *extractor* dan dikirimnya ke SQL Server berhasil.

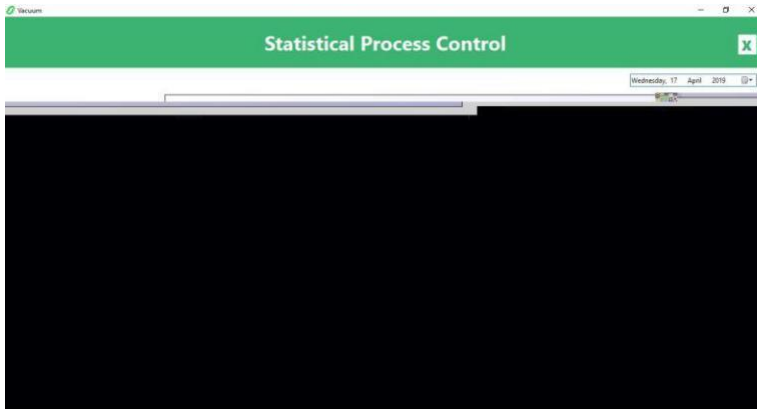
4.2.2. Pengujian Aplikasi *Interface*

Pada pengujian aplikasi *interface* ini dilakukan pengujian terhadap konektivitas antara aplikasi dengan *database* SQL Server dan pengolahan terhadap nilai-nilai dan grafik dari masing-masing variabel *SPC*.



Gambar 4.10 Halaman utama aplikasi *interface*

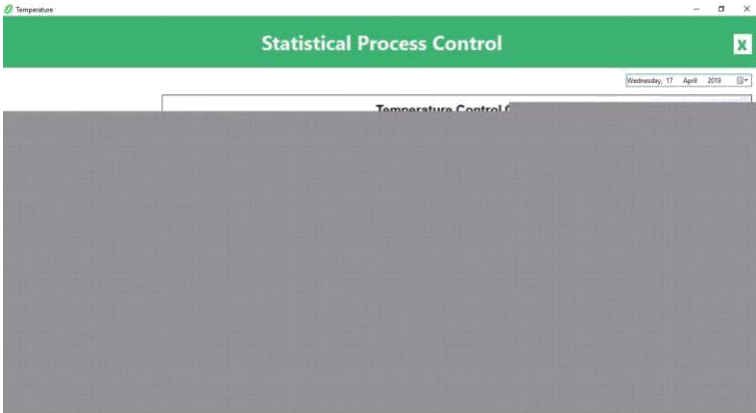
Pada halaman utama aplikasi ini menampilkan data *real-time* dari masing-masing parameter yang diambil dari *database* SQL Server. Selain nilai data dari masing-masing parameter, pada halaman utama ini menampilkan data dari variabel *UCL* atau *upper control limit* dan *LCL* atau *lower control limit* dari masing-masing parameternya agar mempermudah memonitoring kondisi mesin tersebut.



Gambar 4.11 Halaman grafik pada parameter tekanan



Gambar 4.12 Halaman grafik pada parameter Tegangan



Gambar 4.13 Halaman grafik pada parameter Suhu

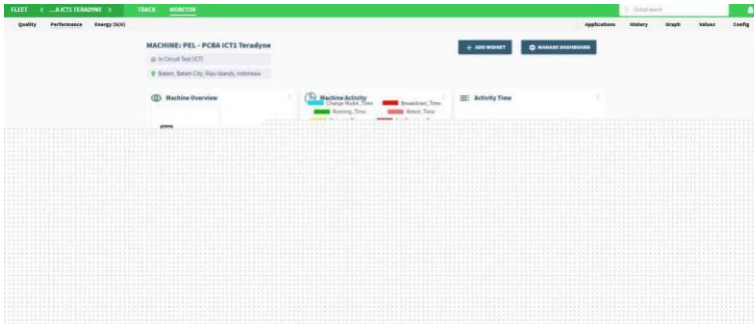


Gambar 4.14 Halaman grafik pada parameter Kelembaban

Pada halaman grafik dari masing-masing parameter menampilkan grafik dengan variabel-variabel *statistical process control*, yang mana pada gambar memnunjukkan garis merah berupa *upper control limit* dan *lower control limit*, kemudian garis lurus berwarna hitam adalah rata-rata dari data perjamnya, lalu garis berwarna hijau merupakan rata-rata data yang diambil per satu jam.

Pada pengujian aplikasi ini jika grafik tertampil maka pengujian ini berhasil dan juga dapat mengetahui fluktuatifitas dari masing-masing parameter.

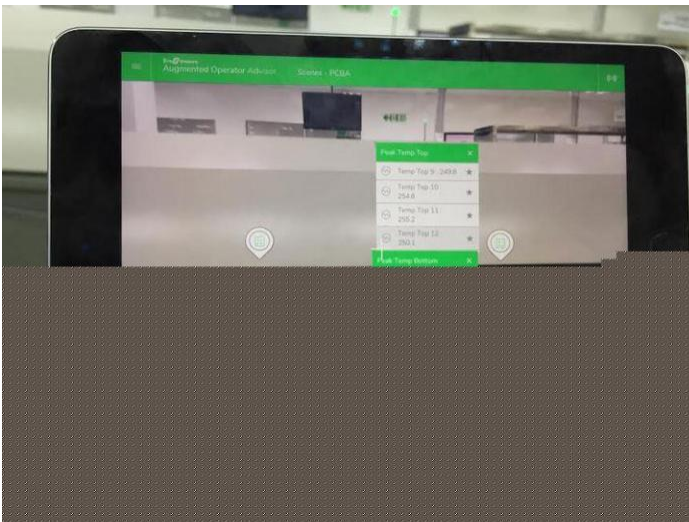
4.2.3. Pengujian Machine Advisor



Gambar 4.15 Dashboard Machine Advisor

Pada Machine Advisor, data-data dari masing-masing parameter yang sudah diolah dan diproses akan terlihat dalam *dashboard machine advisor* ini. Untuk tampilan *dashboard* dapat diatur sesuai keinginan dan penempatannya.

4.2.4. Pengujian Augmented Operator Advisor



Gambar 4.16 Augmented Operator Advisor

Untuk pengujian Augmented Operator Advisor ini dilakukan menggunakan *iPad* atau *Tablet Gadget* yang sudah terpasang aplikasi Augmented Operator Advisor. Untuk mengkoneksikan aplikasi tersebut terhadap server harus diinputkan ip server yang terpasang data dari AOA Builder. Kemudian *scan tag* yang sudah ditempelkan pada mesin, maka *Point of Interest* yang sudah dibuat akan muncul seperti pada Gambar 4.16. Dan jika nilai data tidak masuk maka akan ditampilkan simbol tanda tanya, dan jika data sudah masuk maka akan ditampilkan sesuai data yang diinputkan.

Selain nilai data dari mesin, AOA dapat juga menampilkan dokumen-dokumen mesin, seperti dokumen *manual user* mesin jika terjadi masalah maka tidak harus membuka buku tersebut lagi.

4.2.5. Pengujian IE Andon



Gambar 4.17 IE Andon

IE Andon merupakan sebuah sistem dimana sistem ini akan mencatat durasi waktu, dan data-data dari *station*. Setiap kegiatan di *station* tersebut akan dicatat dan disimpan ke dalam *database SQL Server*.

Sistem andon ini dibuat agar memudahkan para *supervisor* atau *engineer* menganalisa kondisi dan hasil produksi dari mesin atau *station* tersebut.

Pada pengujian ini dilakukan uji coba terhadap aplikasi andon seperti pada Gambar 4.17 yang akan dioperasikan oleh *operator* dari *station* di Mesin ICT. Pengujian ini dilakukan dengan menguji setiap *button* pada aplikasi mempunyai statusnya masing-masing dan statusnya tersebut akan dikirimkan ke *database*.

Date	Station	Operator	Reference	StartTime	EndTime	Duration_in_Minutes	Duration_in_Hour	Status	Hourly
2019-04-25	Wave1	DEWI,RIRI	MCBIYA	14:58:39.9270000				CHANGE SHIFT	14
2019-04-25	Wave1	ROSNA,WIRNA	LULEUY	14:54:02.8000000				RUNNING PRODUCTION	14
2019-04-25	Wave1	ROSNA,WIRNA	LULEUY	14:54:01.8000000	14:54:02.8770000	.02	.00	CHANGE SHIFT	14
2019-04-25	Wave1	DINA,WINDA	LULEUY	12:36:29.7670000	14:54:01.8600000	117.52	1.96	RUNNING PRODUCTION	12
2019-04-25	Wave1	DINA,WINDA	LULEUY	12:25:17.1170000	12:56:28.7530000	31.20	.52	Break	12
2019-04-25	Wave1	DINA,WINDA	LULEUY	11:34:09.6870000	12:25:17.9300000	51.12	.85	RUNNING PRODUCTION	11
2019-04-25	Wave1	DINA,WINDA	ULAPMKDC	11:34:00.5870000	11:34:06.6430000	.15	.00	Change Model Ref.	11
2019-04-25	Wave1	DEBY,DWI,RAHAYU	MCBIYA	10:30:27.9130000	14:58:39.8270000	266.18	4.47	RUNNING PRODUCTION	10
2019-04-25	Wave1	DINA,WINDA	ULAPMKDC	09:56:57.3630000	11:34:00.5570000	97.05	1.62	RUNNING PRODUCTION	9
2019-04-25	Wave1	DEBY,DWI,RAHAYU	MCBIYA	09:53:55.5530000	10:30:27.8900000	36.53	.61	Break	9
2019-04-25	Wave1	DINA,WINDA	ULAPMKDC	09:25:45.2400000	09:56:57.3500000	31.20	.52	Break	9
2019-04-25	Wave1	DINA,WINDA	ULAPMKDC	09:11:46.2300000	09:25:45.1330000	13.92	.23	RUNNING PRODUCTION	9
2019-04-25	Wave1	DINA,WINDA	ULAPMK200	09:11:34.5270000	09:11:46.2500000	.23	.00	Change Model Ref.	9
2019-04-25	Wave1	DEBY,DWI,RAHAYU	MCBIYA	08:22:10.8230000	09:53:55.5330000	91.73	1.53	RUNNING PRODUCTION	8
2019-04-25	Wave1	DEBY,DWI,RAHAYU	MCBIYA	07:16:00.1230000	08:22:10.8000000	66.17	1.10	Pallet Waiting/Problem	7
2019-04-25	Wave1	DEBY,DWI,RAHAYU	MCBIYA	06:53:26.7670000	07:16:00.5700000	22.55	.38	RUNNING PRODUCTION	6
2019-04-25	Wave1	DEBY,DWI,RAHAYU	S189938	06:48:46.7000000	06:53:26.7100000	4.67	.08	Change Model Ref.	6
2019-04-25	Wave1	DINA,WINDA	ULAPMK200	06:06:06.1000000	09:11:34.4830000	185.47	3.09	RUNNING PRODUCTION	6
2019-04-25	Wave1	DEBY,DWI,RAHAYU	S189938	06:02:38.8130000	06:48:45.9800000	46.12	.77	RUNNING PRODUCTION	6
2019-04-25	Wave1	DEBY,DWI,RAHAYU	S189938	06:02:37.5700000	06:02:38.7930000	.02	.00	CHANGE SHIFT	6
2019-04-25	Wave1	DINA,WINDA	ULAPMK200	06:01:13.4400000	06:06:06.6700000	4.87	.08	CHANGE SHIFT	6
2019-04-25	Wave1	NUFUS,DEFFI	S189938	05:33:09.6830000	06:02:37.2700000	29.45	.49	RUNNING PRODUCTION	5
2019-04-25	Wave1	NUFUS,DEFFI	S189938	05:20:01.7470000	05:33:09.6630000	13.12	.22	Move Out From Call	5

Gambar 4.18 Database IE Andon

Pada Gambar 4.18 terlihat sebuah *database* dari IE Andon yang mana data-data yang dikirimkan ke *database* ada banyak seperti tanggal, nama *station*, nama operator, *reference* dari produk yang sedang dilakukan pengetesan, *startTime*, *EndTime*, durasi waktu, dan status.

Dari data diatas dapat dilakukan analisa terhadap produk yang dihasilkan, jika produk yang dihasilkan tidak mencapai target yang sudah ditentukan maka akan dilakukan pengecekan terhadap setiap kegiatan yang dilakukan di *station* tersebut dengan melihat hasil dari IE Andon.

BAB V

PENUTUP

Setelah dilakukan rangkaian kegiatan perancangan sistem dan pengujian, penulis memperoleh kesimpulan dan memberikan beberapa saran sebagai berikut.

5.1. Kesimpulan

Kesimpulan yang diperoleh dalam Proyek Akhir ini adalah:

- a. Berdasarkan pengujian sensor suhu dan kelembaban dapat disimpulkan nilai suhu dari mesin akan bertolak belakang atau berbanding terbalik dengan persentase kelembaban.
- b. Berdasarkan pengujian sensor tegangan dan tekanan dapat disimpulkan bahwa tegangan akan semakin menguat jika mesin sedang melakukan penyedotan
- c. Mesin ICT akan melakukan pengecekan produk lebih banyak jika kondisi mesin stabil

5.2. Saran

Beberapa saran yang dapat penulis berikan untuk pengembangan Proyek Akhir adalah:

- a. Sistem perancangan dari masing-masing parameter untuk lebih dirapihkan atau ditata kembali agar lebih tertata dan tidak membahayakan atau mengganggu komponen-komponen yang lainnya.
- b. Sistem *Internet of Things* ini hanya masih dapat memonitoring berbagai macam elemen dan parameter, sehingga penulis menyarankan agar kedepannya sistem *Internet of Things* ini dapat mengontrol dari berbagai macam *device*.

- Halaman ini sengaja dikosongkan -

DAFTAR PUSTAKA

- [1], "**TestStation**", <https://www.teradyne.com/products/test-solutions/production-board-test/teststation>
- [2], "**Statistical process control (SPC)**", <http://repository.usu.ac.id/bitstream/handle/123456789/63555/Chapter%20II.pdf?sequence=4&isAllowed=y>
- [3], "**HMIBMPSI74D4801**", <https://www.schneider-electric.com/en/product/HMIBMPSI74D4801/modular-box-pc-hmibm-performance-ssd-dc-windows-10-4-slots/>
- [4], "**Datasheet CL110**", RM6 Cubicle CB Humidity Sensor installation (CL110), Schneider.
- [5], "**Datasheet ZBRN32**", Harmony XB5R ZBRN32, Schneider.
- [6], "**Datasheet PowerTag**", PowerTag Wireless Energy Sensor, Schneider.
- [7], "**Datasheet Acti9 Smartlink SID**", Acti9 PowerTag A9XMWA20, Schneider.
- [8], "**Datasheet PLC TM241CEC24R**", Logic Controller Modicon M241 TM241CEC24R, Schneider.
- [9], "**GP-M010**", <https://www.keyence.com/products/process/pressure/gp-m/models/gp-m010/index.jsp>
- [10] https://id.wikipedia.org/wiki/Microsoft_Visual_Studio
- [11] https://id.wikipedia.org/wiki/Microsoft_SQL_Server

- Halaman ini sengaja dikosongkan -

LAMPIRAN

1. Program Aplikasi Data Extractor

```
Imports System.Data.SqlClient
Imports ModbusTCP
Imports System.IO

Public Class Form1

    Private MBmaster As ModbusTCP.Master
    Private data As Byte()

    Public valVacuum_Xbar, valVacuum_XDbar, valTemp_Xbar, valTemp_XDbar,
    valHum_Xbar, valHum_XDbar, valVolt_Xbar, valVolt_XDbar

    Public SQLConn As New SqlConnection("Database=" &
    My.Settings.DatabaseSQL & "; Server=" & My.Settings.ServerSQL &
    ";Integrated Security=True")

    Private Shared Sub Main()
        Application.Run(New Form1)
    End Sub
```

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    SQL_Timer.Start()
    data = New Byte((0) - 1) {}
    SendDataToSQL()
    spc()
End Sub

Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing
    If (Not (MBmaster) Is Nothing) Then
        MBmaster.Dispose()
        MBmaster = Nothing
    End If
    Application.Exit()
End Sub

Private Sub MBmaster_OnResponseData(ID As UShort, unit As Byte, Func As Byte, values As Byte())
    If Me.InvokeRequired Then
        Me.BeginInvoke(New Master.ResponseData(AddressOf MBmaster_OnResponseData), New Object() {ID, unit, Func, values})
    Return
    End If
    Select Case ID
        Case 1
            data = values
            Dim dataVacuum = values(3) '=> %MW1
            txtVacuum.Text = 0.00015071186 * dataVacuum
    End Select
End Sub

```

```

Private Sub MBmaster_OnException(ID As UShort, unit As Byte, Func
As Byte, Exception As Byte)
    Dim exc As String = "Modbus Says Error: "

    Select Case Exception

        Case Master.excIllegalFunction : exc = (exc + "Illegal
Function!")

        Case Master.excIllegalDataAdr : exc = (exc + "Illegal Data
Address!")

        Case Master.excIllegalDataVal : exc = (exc + "Illegal Data
Value!")

        Case Master.excSlaveDeviceFailure : exc = (exc + "Slave
Device Failure!")

        Case Master.excAck : exc = (exc + "Acknowledge!")

        Case Master.excGatePathUnavailable : exc = (exc + "Gateway
Path Unavailable!")

        Case Master.excExceptionTimeout : exc = (exc + "Slave
Timeout!")

        Case Master.excExceptionConnectionLost : exc = (exc +
"Connection Is Lost!")

        Case Master.excExceptionNotConnected : exc = (exc + "Not
Cennected!")

    End Select

    MessageBox.Show(exc, "Modbus Slave Exception")

End Sub

Public Function ReadStartAdr() As UShort
    If txtStartAddress.Text.IndexOf("0x", 0, txtStar-
tAddress.Text.Length) = 0 Then
        Dim str As String = txtStartAddress.Text.Replace("0x", "")
        Dim hex As UShort = Convert.ToUInt16(str, 16)

    Else
        Convert.ToUInt16(txtStartAddress.Text)
    End If
End Function

```

```

Private Sub btnReadHoldingRegister_Click(sender As Object, e As EventArgs) Handles btnReadHoldingRegister.Click
    Dim ID As UShort = 3
    Dim unit As Byte = Convert.ToByte(txtUnit.Text)
    Dim StartAddress As UShort = ReadStartAdr()
    Dim Length As Byte = Convert.ToByte(txtSize.Text)

    MBmaster.ReadHoldingRegister(ID, unit, StartAddress, Length)
End Sub

Private Sub btnConn_Click(sender As Object, e As EventArgs) Handles btnConn.Click
    If btnConn.Text = "Connect" Then
        If txtIP.Text = "SuperAdmin" Then
            grpExchange.Enabled = True
            grpData.Enabled = True
            txtIP.Clear()
        Else
            Try
                MBmaster = New Master(txtIP.Text, 502)
                AddHandler MBmaster.OnResponseData, New ModbusTCP.Master.ResponseData(AddressOf MBmaster_OnResponseData)
                AddHandler MBmaster.OnException, New ModbusTCP.Master.ExceptionData(AddressOf MBmaster_OnException)

                grpExchange.Enabled = True
                grpData.Enabled = True

                txtIP.Enabled = False
                btnConn.Text = "Disconnect"

                SQL_Timer.Enabled = True
                SQL_Timer.Start()
                Modbus_Timer.Enabled = True
                Modbus_Timer.Start()

            Catch ex As Exception
                MsgBox(ex.ToString, MsgBoxStyle.Critical, "Modbus Connection Error")
                txtIP.Focus()
                txtIP.SelectAll()
            End Try
        End If
    Else
        If (Not (MBmaster) Is Nothing) Then
            MBmaster.Dispose()
            MBmaster = Nothing
        End If
        txtIP.Enabled = True
        txtIP.Focus()
        txtIP.SelectAll()
        btnConn.Text = "Connect"
    End If
End Sub

```



```

        SQL_Timer.Stop()
        SQL_Timer.Enabled = False

        Modbus_Timer.Stop()
        Modbus_Timer.Enabled = False

        grpExchange.Enabled = False
        grpData.Enabled = False
    End If
End If
End Sub

Sub SendDataToSQL()
    'Temperature and Humidity
    Dim filedirectorySuhu = My.Settings.FileDirectoryTemp Dim
    Getdata = File.ReadAllLines(filedirectorySuhu).Last() Dim
    data() = Getdata.Split(",")
    Dim Temp_ICT1 = data(My.Settings.ArrayData_Temperature)
    Dim Hum_ICT1 = data(My.Settings.ArrayData_Humidity)

    txtTemperature.Text = (Temp_ICT1 * 9 / 5) + 32
    txtHumidity.Text = Hum_ICT1

    'Voltage
    Dim getDataVolt =
    File.ReadAllLines(My.Settings.FileDirectoryVolt).Last()

    Dim dataVolt() = Split(getDataVolt, ",")

    Dim V1 = dataVolt(4)
    txtVoltage.Text = V1
    'Vacuum

    Dim query As String = "INSERT INTO
    SPC(Date,Time,MachineName,Vacuum,Voltage,Temperature,Humidity,H
    ourly) VALUES(getdate(),getdate(),' " & My.Settings.MachineName &
    "',' " & txtVacuum.Text & "',' " & txtVoltage.Text & "',' " &
    txtTemperature.Text & "',' " & txtHumidity.Text & "',' DATE-
    PART(hour,getdate()))"

    Dim upload As New SqlCommand(query, SQLConn)
    SQLConn.Open()

    upload.ExecuteNonQuery()
    SQLConn.Close()
End Sub

```

```

Private Sub SQL_Timer_Tick(sender As Object, e As EventArgs) Handles
SQL_Timer.Tick
    Try
        SendDataToSQL()
        spc()

        Catch ex As Exception

    End Try
End Sub

Private Sub Modbus_Timer_Tick(sender As Object, e As EventArgs) Handles
Modbus_Timer.Tick
    Call btnReadHoldingRegister_Click(sender, e)
End Sub

Private Sub txtIP_PreviewKeyDown(sender As Object, e As Pre-
viewKeyDownEventArgs) Handles txtIP.PreviewKeyDown

    If e.KeyCode = Keys.Enter Then
        Call btnConn_Click(sender, e)
    End If
End Sub

Sub spc()

    'Get Hourly
    Dim hourly As New SqlCommand("SELECT DATEPART(hour,getdate())",
SQLConn)
    If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
    Dim valHourly = hourly.ExecuteScalar().ToString()
    SQLConn.Close()

    'Cek Data
    Dim cek As New SqlCommand("SELECT X_Bar_Vacuum From
SPC_ControlChart WHERE Date = CONVERT(date,getdate()) AND Hour-
ly = '" & valHourly & "' AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

    If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
    Dim valCek = cek.ExecuteScalar()
    SQLConn.Close()

    If valCek Is Nothing Then

        'insert to table control chart
        Dim insertXbar As New SqlCommand("INSERT INTO
SPC_ControlChart(Date,Time,Hourly,MachineName,X_Bar_Vacuum,X_DBar_Vacuum,X_Bar_Temp,X_DBar_Temp,X_Bar_Hum,X_DBar_Hum,X_Bar_Y01,X_DBar_Y01)
VALUES(getdate(),getdate(),'" & valHourly & "','" & My.Settings.MachineName & "','" & 0,0,0,0,0,0,0,0)", SQLConn)

```

```

If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
insertXbar.ExecuteNonQuery()
SQLConn.Close()

Else

'X-Bar
Dim Vacuum_Xbar As New SqlCommand("SELECT AVG(Vacuum)
FROM SPC WHERE Date = convert(date,getdate()) and
Hourly='" & valHourly & "' AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

Dim Temp_Xbar As New SqlCommand("SELECT
AVG(Temperature) FROM SPC WHERE Date = con-
vert(date,getdate()) and Hourly='" & valHourly & "'
AND MachineName = '" & My.Settings.MachineName & "'",
SQLConn)

Dim Hum_Xbar As New SqlCommand("SELECT AVG(Humidity)
FROM SPC WHERE Date = convert(date,getdate()) and
Hourly='" & valHourly & "' AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

Dim Volt_Xbar As New SqlCommand("SELECT AVG(Voltage)
FROM SPC WHERE Date = convert(date,getdate()) and
Hourly='" & valHourly & "' AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()

valVacuum_Xbar = Vacuum_Xbar.ExecuteScalar()
valTemp_Xbar = Temp_Xbar.ExecuteScalar()
valHum_Xbar = Hum_Xbar.ExecuteScalar()
valVolt_Xbar = Volt_Xbar.ExecuteScalar()
SQLConn.Close()

'Update X-Bar
Dim insert_Xbar As New SqlCommand("UPDATE
SPC_ControlChart SET X_Bar_Vacuum='" & valVacuum_Xbar &
"',X_Bar_Temp='" & valTemp_Xbar & "',X_Bar_Hum='" &
valHum_Xbar & "',X_Bar_Volt='" & valVolt_Xbar & "' WHERE
Date = convert(date,getdate()) and Hourly='" & valHourly
& "' AND MachineName = '" & My.Settings.MachineName &
"'", SQLConn)

If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()

insert_Xbar.ExecuteNonQuery()
SQLConn.Close()

```

```

'X-DBar
Dim Vacuum_XDbar As New SqlCommand("SELECT
AVG(X_Bar_Vacuum) FROM SPC_ControlChart WHERE Date = con-
vert(date,getdate()) AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

Dim Temp_XDbar As New SqlCommand("SELECT AVG(X_Bar_Temp)
FROM SPC_ControlChart WHERE Date = con-
vert(date,getdate()) AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

Dim Hum_XDbar As New SqlCommand("SELECT AVG(X_Bar_Hum)
FROM SPC_ControlChart WHERE Date = con-
vert(date,getdate()) AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

Dim Volt_XDbar As New SqlCommand("SELECT AVG(X_Bar_Volt)
FROM SPC_ControlChart WHERE Date = con-
vert(date,getdate()) AND MachineName = '" &
My.Settings.MachineName & "'", SQLConn)

If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()

valVacuum_XDbar = Vacuum_XDbar.ExecuteScalar()
valTemp_XDbar = Temp_XDbar.ExecuteScalar()
valHum_XDbar = Hum_XDbar.ExecuteScalar()
valVolt_XDbar = Volt_XDbar.ExecuteScalar()
SQLConn.Close()

'Update X_DBar
Dim update_XDbar As New SqlCommand("UPDATE
SPC_ControlChart SET X_DBar_Vacuum = '" & valVacuum_XDbar
& "',X_DBar_Temp = '" & valTemp_XDbar & "',X_DBar_Hum =
'" & valHum_XDbar & "',X_DBar_Volt = '" & valVolt_XDbar &
'" WHERE Date = convert(date,getdate()) AND MachineName =
'" & My.Settings.MachineName & "'", SQLConn)

If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()

update_XDbar.ExecuteNonQuery()
SQLConn.Close()

End If

End Sub

End Class

```

2. Program Aplikasi *Interface*: Halaman Utama

```
Imports System.Data.SqlClient

Public Class Form1
    Private MBmaster As ModbusTCP.Master
    Private txtData As TextBox
    Private labData As Label
    Private data As Byte()
    Private pict As PictureBox
    Public Machine As String

    Public SQLConn As New SqlConnection("Database=TA;Server=DESKTOP-
S603KSA\SQLEXPRESS;Integrated Security=True")

    Private Shared Sub Main()
        Application.Run(New Form1)
    End Sub

    Sub ICT()
        Try

            '-----Vacuum-----
            'STDEV
            Dim stdev As New SqlCommand("SELECT STDEV(X_Bar_Vacuum)
FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND Machine-
Name = '" & Machine & "'", SQLConn)
            If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
            Dim ValSTDEV = stdev.ExecuteScalar()
            SQLConn.Close()

            Dim Vacuum_XDbar As New SqlCommand("SELECT X_DBar_Vacuum
FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND Machine-
Name = '" & Machine & "'", SQLConn)
            If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
            Dim valVacuum_XDbar = Vacuum_XDbar.ExecuteScalar()
            SQLConn.Close()

            'Upper Limit Control
            Dim Val_UCL = valVacuum_XDbar + 3 * ValSTDEV

            'Lower Limit Control
            Dim Val_LCL = valVacuum_XDbar - 3 * ValSTDEV
            lblVacuumUCL.Text = "UCL >> " & FormatNum-
ber(Val_UCL.ToString, 2)
            lblVacuumLCL.Text = "LCL >> " & FormatNum-
ber(Val_LCL.ToString, 2)
        End Try
    End Sub
End Class
```

```

'-----Temperature-----
'STDEV
Dim stdevTemp As New SqlCommand("SELECT STDEV(X_Bar_Temp)
FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND
Machine-Name = '" & Machine & "'", SQLConn)
If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
Dim ValSTDEVTemp = stdevTemp.ExecuteScalar()
SQLConn.Close()

Dim Temp_XDbar As New SqlCommand("SELECT X_DBar_Temp FROM
SPC_ControlChart WHERE Date= convert(Date,getdate()) AND MachineName
= '" & Machine & "'", SQLConn)
If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
Dim valTemp_XDbar = Temp_XDbar.ExecuteScalar()
SQLConn.Close()

'Upper Limit Control
Dim Val_UCL_Temp = valTemp_XDbar + 3 * ValSTDEVTemp

'Lower Limit Control
Dim Val_LCL_Temp = valTemp_XDbar - 3 * ValSTDEVTemp

lblTempUCL.Text = "UCL >> " & FormatNum-
ber(Val_UCL_Temp.ToString, 2)
lblTempLCL.Text = "LCL >> " & FormatNum-
ber(Val_LCL_Temp.ToString, 2)

'-----Humidity-----
'STDEV
Dim stdevHum As New SqlCommand("SELECT STDEV(X_Bar_Hum)
FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND
Machine-Name = '" & Machine & "'", SQLConn)
If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
Dim ValSTDEVHum = stdevHum.ExecuteScalar()
SQLConn.Close()

Dim Hum_XDbar As New SqlCommand("SELECT X_DBar_Hum FROM
SPC_ControlChart WHERE Date= convert(Date,getdate()) AND MachineName
= '" & Machine & "'", SQLConn)
If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
Dim valHum_XDbar = Hum_XDbar.ExecuteScalar()
SQLConn.Close()

'Upper Limit Control
Dim Val_UCL_Hum = valHum_XDbar + 3 * ValSTDEVHum

'Lower Limit Control
Dim Val_LCL_Hum = valHum_XDbar - 3 * ValSTDEVHum

```

```

        lblHumUCL.Text = "UCL >> " & FormatNumber(Val_UCL_Hum.ToString, 2)
        lblHumLCL.Text = "LCL >> " & FormatNumber(Val_LCL_Hum.ToString, 2)

        '-----Voltage-----
        'STDEV
        Dim stdevVolt As New SqlCommand("SELECT STDEV(X_Bar_Volt)
FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND MachineName = '" & Machine & "'", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim ValSTDEVVolt = stdevVolt.ExecuteScalar()
        SQLConn.Close()

        Dim Volt_XDbar As New SqlCommand("SELECT X_DBar_Volt FROM SPC_ControlChart WHERE Date= convert(Date,getdate()) AND MachineName = '" & Machine & "'", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim valVolt_XDbar = Volt_XDbar.ExecuteScalar()
        SQLConn.Close()

        'Upper Limit Control
        Dim Val_UCL_Volt = valVolt_XDbar + 3 * ValSTDEVVolt

        'Lower Limit Control
        Dim Val_LCL_Volt = valVolt_XDbar - 3 * ValSTDEVVolt

        lblVoltUCL.Text = "UCL >> " & FormatNumber(Val_UCL_Volt.ToString, 2)
        lblVoltLCL.Text = "LCL >> " & FormatNumber(Val_LCL_Volt.ToString, 2)

        'Temperature
        Dim cmdTemp As New SqlCommand("SELECT TOP 1 Temperature FROM SPC WHERE MachineName = '" & Machine & "' ORDER BY Date DESC,Time DESC", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim valTmp = cmdTemp.ExecuteScalar()
        SQLConn.Close()

        txtTemperature.Text = valTmp

        'Humidity
        Dim cmdHum As New SqlCommand("SELECT TOP 1 Humidity FROM SPC WHERE MachineName = '" & Machine & "' ORDER BY Date DESC,Time DESC", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim valHum = cmdHum.ExecuteScalar()

```



```

        SQLConn.Close()
        txtHumidity.Text = valHum

        'Voltage
        Dim cmdVolt As New SqlCommand("SELECT TOP 1 Voltage FROM
SPC WHERE MachineName = '" & Machine & "' ORDER BY Date DESC,Time
DESC", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
        Dim valVolt = cmdVolt.ExecuteScalar()
        SQLConn.Close()

        txtVoltage.Text = valVolt

        'Vacuum
        Dim cmdVacuum As New SqlCommand("SELECT TOP 1 Vacuum FROM
SPC WHERE MachineName = '" & Machine & "' ORDER BY Date DESC,Time
DESC", SQLConn)
        If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
        Dim valVacm = cmdVacuum.ExecuteScalar()
        SQLConn.Close()

        txtVacuum.Text = valVacm
    Catch ex As Exception

    End Try
End Sub

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    data = New Byte((0) - 1) {}
    Machine = "ICT1"
    ICT()
    Timer_ICT1.Start()

    Humidity.Machine = "ICT1"
    Humidity.loadData_ICT()
    Humidity.Timer1.Start()

    Vacuum.Machine = "ICT1"
    Vacuum.loadData()
    Vacuum.Timer1.Start()

    Temperature.Machine = "ICT1"
    Temperature.loadData()
    Temperature.Timer1.Start()

    Voltage.Machine = "ICT1"
    Voltage.loadData_ICT()
    Voltage.Timer1.Start()
End Sub

```

```

Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing
    If (Not (MBmaster) Is Nothing) Then
        MBmaster.Dispose()
        MBmaster = Nothing
    End If

    Application.Exit()
End Sub

Private Sub ModbusConnectionToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ModbusConnectionToolStripMenuItem.Click
    Connection.Show()
    Me.Hide()
    Connection.txtIP.Focus()
    Connection.txtIP.SelectAll()
End Sub

Private Sub btnVacuumGraph_Click(sender As Object, e As EventArgs) Handles btnVacuumGraph.Click
    Vacuum.Show()
    Me.Hide()

End Sub

Private Sub btnVoltageGraph_Click(sender As Object, e As EventArgs) Handles btnVoltageGraph.Click
    Voltage.Show()
    Me.Hide()

End Sub

Private Sub btnTempGraph_Click(sender As Object, e As EventArgs) Handles btnTempGraph.Click
    Temperature.Show()
    Me.Hide()

End Sub

Private Sub btnHumiGraph_Click(sender As Object, e As EventArgs) Handles btnHumiGraph.Click
    Humidity.Show()
    Me.Hide()

End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ExitToolStripMenuItem.Click
    If (Not (MBmaster) Is Nothing) Then
        MBmaster.Dispose()
        MBmaster = Nothing
    End If

    Application.Exit()

```

```

End Sub

Private Sub Timer_ICT1_Tick(sender As Object, e As EventArgs)
Handles Timer_ICT1.Tick
    Try
        ICT()
    Catch ex As Exception
        MsgBox("No Data Shown", MsgBoxStyle.Exclamation, "Error")
    End Try
End Sub

Private Sub ICT1ToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ICT1ToolStripMenuItem.Click
    Try

        Machine = "ICT1"
        ICT()
        LabelTitle.Text = "ICT 1 Teradyne"

        Humidity.Machine = "ICT1"
        Humidity.loadData_ICT()
        Humidity.Timer1.Start()

        Vacuum.Machine = "ICT1"
        Vacuum.loadData()
        Vacuum.Timer1.Start()

        Temperature.Machine = "ICT1"
        Temperature.loadData()
        Temperature.Timer1.Start()

    Catch ex As Exception
        MsgBox("No Data Shown", MsgBoxStyle.Exclamation, "Error")
    End Try
End Sub

Private Sub ICT2ToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ICT2ToolStripMenuItem.Click
    Try

        Machine = "ICT2"
        ICT()
        LabelTitle.Text = "ICT 2 Teradyne"

        Humidity.Machine = "ICT2"
        Humidity.loadData_ICT()
        Humidity.Timer1.Start()

        Vacuum.Machine = "ICT2"
        Vacuum.loadData()
        Vacuum.Timer1.Start()

        Temperature.Machine = "ICT2"

```

```

        Temperature.loadData()
        Temperature.Timer1.Start()

    Catch ex As Exception
        MsgBox("No Data Shown", MsgBoxStyle.Exclamation, "Error")
    End Try
End Sub

Private Sub ICT3ToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ICT3ToolStripMenuItem.Click
    Try
        Machine = "ICT3"
        ICT()
        LabelTitle.Text = "ICT 3 Teradyne"
        Humidity.Machine = "ICT3"
        Humidity.loadData_ICT()
        Humidity.Timer1.Start()
        Vacuum.Machine = "ICT3"
        Vacuum.loadData()
        Vacuum.Timer1.Start()

        Temperature.Machine = "ICT3"
        Temperature.loadData()
        Temperature.Timer1.Start()
    Catch ex As Exception
        MsgBox("No Data Shown", MsgBoxStyle.Exclamation, "Error")
    End Try
End Sub

Private Sub ICT4ToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ICT4ToolStripMenuItem.Click
    Try

        Machine = "ICT4"
        ICT()
        LabelTitle.Text = "ICT 4 Teradyne"

        Humidity.Machine = "ICT4"
        Humidity.loadData_ICT()
        Humidity.Timer1.Start()

        Vacuum.Machine = "ICT4"
        Vacuum.loadData()
        Vacuum.Timer1.Start()

        Temperature.Machine = "ICT4"
        Temperature.loadData()
        Temperature.Timer1.Start()

    Catch ex As Exception
        MsgBox("No Data Shown", MsgBoxStyle.Exclamation, "Error")
    End Try
End Sub
End Class

```

3. Program Aplikasi *Interface: Form Vacuum*

```
Imports ZedGraph
Imports System.Data.SqlClient

Public Class Vacuum

    Public SQLConn As New SqlConnection("Database=TA;Server=DESKTOP-
S603KSA\SQLEXPRESS;Integrated Security=True")
    Public Machine As String

    Private Sub Label11_Click(sender As Object, e As EventArgs) Handles
Label11.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub Vacuum_FormClosing(sender As Object, e As FormClos-
ingEventArgs) Handles Me.FormClosing
        Application.Exit()
    End Sub

    Private Sub CreateGraph(ByVal zgc As ZedGraphControl)

        Try

            DateTimePicker1.Format = DateTimePickerFormat.Custom
            DateTimePicker1.CustomFormat = "yyyy/MM/dd"

            Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

            'STDEV
            Dim stdev As New SqlCommand("SELECT STDEV(X_Bar_Vacuum)
FROM SPC_ControlChart WHERE Date='" & tgl & "' AND MachineName = '" &
Machine & "'", SQLConn)
            SQLConn.Open()
            Dim ValSTDEV = stdev.ExecuteScalar()
            SQLConn.Close()

            Dim valVacuum_XDbar = DataGridView1.Item(1, 1).Value

            'Upper Limit Control
            Dim Val_UCL = valVacuum_XDbar + 3 * ValSTDEV

            'Lower Limit Control
            Dim Val_LCL = valVacuum_XDbar - 3 * ValSTDEV
            zgc.AxisChange()

            Dim myPane As GraphPane = zgc.GraphPane

            ' Set the titles and axis labels
            myPane.Title.Text = "Vacuum Control Chart"
```

```

myPane.XAxis.Title.Text = ""
myPane.YAxis.Title.Text = "X-Bar"

' Make up some data points from the Sine function
Dim list_Xbar = New PointPairList()
Dim list_ControlLimit = New PointPairList()
Dim list_UCL = New PointPairList()
Dim list_LCL = New PointPairList()
Dim x, y_Xbar, y_ControlLimit, y_UCL, y_LCL

For x = 0 To DataGridView1.Rows.Count - 2

    y_Xbar = DataGridView1.Item(0, x).Value
    y_ControlLimit = DataGridView1.Item(1, x).Value
    y_UCL = Val_UCL
    y_LCL = Val_LCL

    list_Xbar.Add(x, y_Xbar)
    list_ControlLimit.Add(x, y_ControlLimit)
    list_UCL.Add(x, y_UCL)
    list_LCL.Add(x, y_LCL)
Next x

' Generate a blue curve with circle symbols, and "My Curve
2" in the legend
Dim X_Bar As LineItem = myPane.AddCurve("X-Bar", list_Xbar,
Color.FromArgb(1, 155, 1), SymbolType.Circle)
Dim ControlLimit As LineItem = myPane.AddCurve("Control
Limit", list_ControlLimit, Color.FromArgb(50, 50, 50), SymbolType.None)
Dim LCL As LineItem = myPane.AddCurve("Lower Control Lim-
it", list_LCL, Color.FromArgb(204, 0, 0), SymbolType.None)
Dim UCL As LineItem = myPane.AddCurve("Upper Control Lim-
it", list_UCL, Color.FromArgb(204, 0, 0), SymbolType.None)

' Adjust Line Width
X_Bar.Line.Width = 3.4F
ControlLimit.Line.Width = 3.4F
UCL.Line.Width = 3.4F
LCL.Line.Width = 3.4F

' Fill the area under the curve with a white-red gradient
at 45 degrees
UCL.Line.Fill = New Fill(Color.White, Color.LightBlue,
45.0F)
LCL.Line.Fill = New Fill(Color.White, Col-
or.LightGoldenrodYellow, 45.0F)

' Make the symbols opaque by filling them with white
LCL.Symbol.Fill = New Fill(Color.White)

```

```

        ' Fill the axis background with a color gradient
        myPane.Chart.Fill = New Fill(Color.White, Color.LightGoldenrodYellow, 45.0F)

        ' Fill the pane background with a color gradient
        myPane.Fill = New Fill(Color.White, Color.FromArgb(220, 220, 255), 45.0F)

        ' Calculate the Axis Scale
        Ranges zgc.AxisChange()
        zgc.Refresh()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        Catch ex As Exception
            Timer1.Stop()
            MsgBox("No Data Shown", MsgBoxStyle.Exclamation)

    End Try
End Sub

Private Sub SetSize()
    zg1.Location = New Point(10, 10)

    ' Leave a small margin around the outside of the control
    zg1.Size = New Size(ClientRectangle.Width - 20, ClientRectangle.Height - 20)
End Sub

Private Sub Vacuum_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    Me.WindowState = FormWindowState.Maximized

    loadData()
    CreateGraph(zg1)
    'SetSize()
End Sub

Sub loadData()

    DateTimePicker1.Format = DateTimePickerFormat.Custom
    DateTimePicker1.CustomFormat = "yyyy/MM/dd"

    Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

    Try

        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim dataset As New DataSet

```

```

Dim adapter As New SqlDataAdapter("SELECT
X_Bar_Vacuum,X_DBar_Vacuum FROM SPC_ControlChart WHERE Date = '" & tgl
& "' AND MachineName = '" & Machine & "' order by Time ASC", SQLConn)

adapter.Fill(dataset)
DataGridView1.DataSource = dataset.Tables(0)
SQLConn.Close()

DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
DataGridView1.Visible = False

Catch ex As Exception

End Try
End Sub

Private Sub DateTimePicker1_ValueChanged(sender As Object, e As
EventArgs) Handles DateTimePicker1.ValueChanged

zg1.GraphPane.CurveList.Clear()
zg1.GraphPane.GraphObjList.Clear()

zg1.Refresh()
loadData()
CreateGraph(zg1)
End Sub

Private Sub Timer_ICT_Tick(sender As Object, e As EventArgs) Han-
dles Timer_ICT.Tick

zg1.GraphPane.CurveList.Clear()
zg1.GraphPane.GraphObjList.Clear()

zg1.Refresh()
loadData()
CreateGraph(zg1)
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
zg1.GraphPane.CurveList.Clear()
zg1.GraphPane.GraphObjList.Clear()

zg1.Refresh()
loadData()
CreateGraph(zg1)
Timer1.Stop()
End Sub
End Class

```


4. Program Aplikasi *Interface: Form Temperature*

```
Imports ZedGraph
Imports System.Data.SqlClient

Public Class Temperature
    Public SQLConn As New SqlConnection("Database=TA;Server=DESKTOP-
S603KSA\SQLEXPRESS;Integrated Security=True")

    Public Machine As String

    Private Sub Label11_Click(sender As Object, e As EventArgs) Handles
Label11.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub Temperature_FormClosing(sender As Object, e As Form-
ClosingEventArgs) Handles Me.FormClosing
        Application.Exit()
    End Sub

    Private Sub CreateGraph(ByVal zgc As ZedGraphControl)
        Try

            DateTimePicker1.Format = DateTimePickerFormat.Custom
            DateTimePicker1.CustomFormat = "yyyy/MM/dd"

            Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

            'STDEV
            Dim stdev As New SqlCommand("SELECT STDEV(X_Bar_Temp) FROM
SPC_ControlChart WHERE Date='" & tgl & "' AND MachineName = '" & Ma-
chine & "'", SQLConn)
            If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
            Dim ValSTDEV = stdev.ExecuteScalar()
            SQLConn.Close()

            Dim valTemp_XDbar = DataGridView1.Item(1, 1).Value

            'Upper Limit Control
            Dim Val_UCL = valTemp_XDbar + 3 * ValSTDEV

            'Lower Limit Control
            Dim Val_LCL = valTemp_XDbar - 3 * ValSTDEV

            zgc.AxisChange()
        End Try
    End Sub
End Class
```

```

Dim myPane As GraphPane = zgc.GraphPane

' Set the titles and axis labels
myPane.Title.Text = "Temperature Control Chart"
myPane.XAxis.Title.Text = ""
myPane.YAxis.Title.Text = "X-Bar"

' Make up some data points from the Sine function
Dim list_Xbar = New PointPairList()
Dim list_ControlLimit = New PointPairList()
Dim list_UCL = New PointPairList()
Dim list_LCL = New PointPairList()
Dim x, y_Xbar, y_ControlLimit, y_UCL, y_LCL

For x = 0 To DataGridView1.Rows.Count - 2

    y_Xbar = DataGridView1.Item(0, x).Value
    y_ControlLimit = DataGridView1.Item(1, x).Value
    y_UCL = Val_UCL
    y_LCL = Val_LCL

    list_Xbar.Add(x, y_Xbar)
    list_ControlLimit.Add(x, y_ControlLimit)
    list_UCL.Add(x, y_UCL)
    list_LCL.Add(x, y_LCL)
Next x

' Generate a blue curve with circle symbols, and "My Curve
2" in the legend
Dim X_Bar As LineItem = myPane.AddCurve("X-Bar", list_Xbar,
Color.FromArgb(1, 155, 1), SymbolType.Circle)
Dim ControlLimit As LineItem = myPane.AddCurve("Control
Limit", list_ControlLimit, Color.FromArgb(50, 50, 50), SymbolType.None)
Dim LCL As LineItem = myPane.AddCurve("Lower Control Lim-
it", list_LCL, Color.FromArgb(204, 0, 0), SymbolType.None)
Dim UCL As LineItem = myPane.AddCurve("Upper Control Lim-
it", list_UCL, Color.FromArgb(204, 0, 0), SymbolType.None)

' Adjust Line Width
X_Bar.Line.Width = 3.4F
ControlLimit.Line.Width = 3.4F
UCL.Line.Width = 3.4F
LCL.Line.Width = 3.4F

' Fill the area under the curve with a white-red gradient
at 45 degrees
UCL.Line.Fill = New Fill(Color.White, Color.LightBlue,
45.0F)

```

```

        LCL.Line.Fill = New Fill(Color.White, Color.LightGoldenrodYellow, 45.0F)

        ' Make the symbols opaque by filling them with white
        LCL.Symbol.Fill = New Fill(Color.White)

        ' Fill the axis background with a color gradient
        myPane.Chart.Fill = New Fill(Color.White, Color.LightGoldenrodYellow, 45.0F)

        ' Fill the pane background with a color gradient
        myPane.Fill = New Fill(Color.White, Color.FromArgb(220, 220, 255), 45.0F)

        ' Calculate the Axis Scale
        Ranges zgc.AxisChange()
        zgc.Refresh()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        Catch ex As Exception
            Timer1.Stop()
            MsgBox("No Data Shown", MsgBoxStyle.Exclamation)

    End Try
End Sub

Private Sub SetSize()
    zg1.Location = New Point(10, 10)

    ' Leave a small margin around the outside of the control
    zg1.Size = New Size(ClientRectangle.Width - 20, ClientRectangle.Height - 20)
End Sub

Private Sub Temperature_Load(sender As Object, e As EventArgs)
    Handles MyBase.Load
        Me.WindowState = FormWindowState.Maximized
        loadData()
        CreateGraph(zg1)
End Sub

Sub loadData()

    DateTimePicker1.Format = DateTimePickerFormat.Custom
    DateTimePicker1.CustomFormat = "yyyy/MM/dd"

    Dim tg1 = DateTimePicker1.Value.ToString("yyyy-MM-dd")

    Try
        If SQLConn.State = ConnectionState.Closed Then SQLConn.Open()
        Dim dataset As New DataSet
    
```

```

        Dim adapter As New SqlDataAdapter("SELECT
X_Bar_Temp,X_DBar_Temp FROM SPC_ControlChart WHERE Date = '" & tgl & "'
AND MachineName = '" & Machine & "' order by Time ASC", SQLConn)

        adapter.Fill(dataset)
        DataGridView1.DataSource = dataset.Tables(0)
        SQLConn.Close()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        DataGridView1.Visible = False

    Catch ex As Exception

    End Try

End Sub

Private Sub DateTimePicker1_ValueChanged(sender As Object, e As
EventArgs) Handles DateTimePicker1.ValueChanged
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

    zg1.Refresh()
    loadData()
    CreateGraph(zg1)
End Sub

Private Sub Timer_ICT_Tick(sender As Object, e As EventArgs) Handles
Timer_ICT.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

    zg1.Refresh()
    loadData()
    CreateGraph(zg1)
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

    zg1.Refresh()
    loadData()
    CreateGraph(zg1)
    Timer1.Stop()
End Sub
End Class

```

5. Program Aplikasi *Interface: Form Humidity*

```
Imports ZedGraph
Imports System.Data.SqlClient

Public Class Humidity

    Public SQLConn As New SqlConnection("Database=TA;Server=DESKTOP-
S603KSA\SQLEXPRESS;Integrated Security=True")

    Public Machine As String

    Private Sub Label11_Click(sender As Object, e As EventArgs) Handles
Label11.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub Humidity_FormClosing(sender As Object, e As FormClos-
ingEventArgs) Handles Me.FormClosing
        Application.Exit()
    End Sub

    Private Sub CreateGraph_ICT(ByVal zgc As ZedGraphControl)
        Try

            DateTimePicker1.Format = DateTimePickerFormat.Custom
            DateTimePicker1.CustomFormat = "yyyy/MM/dd"

            Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

            'STDEV
            Dim stdev As New SqlCommand("SELECT STDEV(X_Bar_Hum) FROM
SPC_ControlChart WHERE Date='" & tgl & "' AND MachineName = '" & Machine
& "'", SQLConn)
            If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
            Dim ValSTDEV = stdev.ExecuteScalar()
            SQLConn.Close()

            Dim valHum_XDbar = DataGridView1.Item(1, 1).Value

            'Upper Limit Control
            Dim Val_UCL = valHum_XDbar + 3 * ValSTDEV

            'Lower Limit Control
            Dim Val_LCL = valHum_XDbar - 3 * ValSTDEV
```

```

zgc.AxisChange()

    Dim myPane As GraphPane = zgc.GraphPane

    ' Set the titles and axis labels
    myPane.Title.Text = "Humidity Control Chart"
    myPane.XAxis.Title.Text = ""
    myPane.YAxis.Title.Text = "X-Bar"

    ' Make up some data points from the Sine function
    Dim list_Xbar = New PointPairList()
    Dim list_ControlLimit = New PointPairList()
    Dim list_UCL = New PointPairList()
    Dim list_LCL = New PointPairList()
    Dim x, y_Xbar, y_ControlLimit, y_UCL, y_LCL

    For x = 0 To DataGridView1.Rows.Count - 2

        y_Xbar = DataGridView1.Item(0, x).Value
        y_ControlLimit = DataGridView1.Item(1, x).Value
        y_UCL = Val_UCL
        y_LCL = Val_LCL

        list_Xbar.Add(x, y_Xbar)
        list_ControlLimit.Add(x, y_ControlLimit)
        list_UCL.Add(x, y_UCL)
        list_LCL.Add(x, y_LCL)
    Next x

    ' Generate a blue curve with circle symbols, and "My Curve
    2" in the legend
    Dim X_Bar As LineItem = myPane.AddCurve("X-Bar", list_Xbar,
    Color.FromArgb(1, 155, 1), SymbolType.Circle)
    Dim ControlLimit As LineItem = myPane.AddCurve("Control Lim-
    it", list_ControlLimit, Color.FromArgb(50, 50, 50), SymbolType.None)
    Dim LCL As LineItem = myPane.AddCurve("Lower Control Limit",
    list_LCL, Color.FromArgb(204, 0, 0), SymbolType.None)
    Dim UCL As LineItem = myPane.AddCurve("Upper Control Limit",
    list_UCL, Color.FromArgb(204, 0, 0), SymbolType.None)

    ' Adjust Line Width
    X_Bar.Line.Width = 3.4F
    ControlLimit.Line.Width = 3.4F
    UCL.Line.Width = 3.4F
    LCL.Line.Width = 3.4F
    UCL.Line.Fill = New Fill(Color.White, Color.LightBlue,
    45.0F)
    LCL.Line.Fill = New Fill(Color.White, Col-
    or.LightGoldenrodYellow, 45.0F)

```

```

        ' Make the symbols opaque by filling them with
        white LCL.Symbol.Fill = New Fill(Color.White)

        ' Fill the axis background with a color gradient
        myPane.Chart.Fill = New Fill(Color.White, Col-
or.LightGoldenrodYellow, 45.0F)

        ' Fill the pane background with a color gradient
        myPane.Fill = New Fill(Color.White, Color.FromArgb(220, 220,
255), 45.0F)

        ' Calculate the Axis Scale
        Ranges zgc.AxisChange()
        zgc.Refresh()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        Catch ex As Exception
            Timer1.Stop()
            MsgBox("No Data Shown", MsgBoxStyle.Exclamation)

    End Try
End Sub

Private Sub SetSize()
    zg1.Location = New Point(10, 10)

    ' Leave a small margin around the outside of the control
    zg1.Size = New Size(ClientRectangle.Width - 20, ClientRectan-
gle.Height - 20)
End Sub

Private Sub Humidity_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
    Me.WindowState = FormWindowState.Maximized
    loadData_ICT()
    CreateGraph_ICT(zg1)

End Sub

Sub loadData_ICT()

    DateTimePicker1.Format = DateTimePickerFormat.Custom
    DateTimePicker1.CustomFormat = "yyyy/MM/dd"

    Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

    Try
        If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
        Dim dataset As New DataSet

```

```

        Dim adapter As New SqlDataAdapter("SELECT
X_Bar_Hum,X_DBar_Hum FROM SPC_ControlChart WHERE Date = '" & tgl & "'
AND MachineName = '" & Machine & "' order by Time ASC", SQLConn)

        adapter.Fill(dataset)
        DataGridView1.DataSource = dataset.Tables(0)
        SQLConn.Close()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        DataGridView1.Visible = False

    Catch ex As Exception

    End Try
End Sub

Private Sub DateTimePicker1_ValueChanged(sender As Object, e As
EventArgs) Handles DateTimePicker1.ValueChanged

    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
End Sub

Private Sub Timer_ICT_Tick(sender As Object, e As EventArgs) Handles
Timer_ICT.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()
    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()
    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
    Timer1.Stop()
End Sub
End Class

```


6. Program Aplikasi *Interface: Form Voltage*

```
Imports ZedGraph
Imports System.Data.SqlClient

Public Class Voltage

    Public SQLConn As New SqlConnection("Database=TA;Server=DESKTOP-
S603KSA\SQLEXPRESS;Integrated Security=True")

    Public Machine As String

    Private Sub Label11_Click(sender As Object, e As EventArgs) Handles
Label11.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub Voltage_FormClosing(sender As Object, e As FormClos-
ingEventArgs) Handles Me.FormClosing
        Application.Exit()
    End Sub

    Private Sub CreateGraph_ICT(ByVal zgc As ZedGraphControl)
        Try

            DateTimePicker1.Format = DateTimePickerFormat.Custom
            DateTimePicker1.CustomFormat = "yyyy/MM/dd"

            Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

            'STDEV
            Dim stdev As New SqlCommand("SELECT STDEV(X_Bar_Volt) FROM
SPC_ControlChart WHERE Date='" & tgl & "' AND MachineName = '" & Machine
& "'", SQLConn)
            If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
            Dim ValSTDEV = stdev.ExecuteScalar()
            SQLConn.Close()

            Dim valHum_XDbar = DataGridView1.Item(1, 1).Value

            'Upper Limit Control
            Dim Val_UCL = valHum_XDbar + 3 * ValSTDEV

            'Lower Limit Control
            Dim Val_LCL = valHum_XDbar - 3 * ValSTDEV
            zgc.AxisChange()
```

```

Dim myPane As GraphPane = zgc.GraphPane

' Set the titles and axis labels
myPane.Title.Text = "Voltage Control Chart"
myPane.XAxis.Title.Text = ""
myPane.YAxis.Title.Text = "X-Bar"

' Make up some data points from the Sine function
Dim list_Xbar = New PointPairList()
Dim list_ControlLimit = New PointPairList()
Dim list_UCL = New PointPairList()
Dim list_LCL = New PointPairList()
Dim x, y_Xbar, y_ControlLimit, y_UCL, y_LCL

For x = 0 To DataGridView1.Rows.Count - 2

    y_Xbar = DataGridView1.Item(0, x).Value
    y_ControlLimit = DataGridView1.Item(1, x).Value
    y_UCL = Val_UCL
    y_LCL = Val_LCL

    list_Xbar.Add(x, y_Xbar)
    list_ControlLimit.Add(x, y_ControlLimit)
    list_UCL.Add(x, y_UCL)
    list_LCL.Add(x, y_LCL)
Next x

' Generate a blue curve with circle symbols, and "My Curve
2" in the legend
Dim X_Bar As LineItem = myPane.AddCurve("X-Bar", list_Xbar,
Color.FromArgb(1, 155, 1), SymbolType.Circle)
Dim ControlLimit As LineItem = myPane.AddCurve("Control Lim-
it", list_ControlLimit, Color.FromArgb(50, 50, 50), SymbolType.None)
Dim LCL As LineItem = myPane.AddCurve("Lower Control Limit",
list_LCL, Color.FromArgb(204, 0, 0), SymbolType.None)
Dim UCL As LineItem = myPane.AddCurve("Upper Control Limit",
list_UCL, Color.FromArgb(204, 0, 0), SymbolType.None)

' Adjust Line Width
X_Bar.Line.Width = 3.4F
ControlLimit.Line.Width = 3.4F
UCL.Line.Width = 3.4F
LCL.Line.Width = 3.4F

' Fill the area under the curve with a white-red gradient at
45 degrees
UCL.Line.Fill = New Fill(Color.White, Color.LightBlue,
45.0F)

```

```

        LCL.Line.Fill = New Fill(Color.White, Color.LightGoldenrodYellow, 45.0F)

        ' Make the symbols opaque by filling them with white
        LCL.Symbol.Fill = New Fill(Color.White)

        ' Fill the axis background with a color gradient
        myPane.Chart.Fill = New Fill(Color.White, Color.LightGoldenrodYellow, 45.0F)

        ' Fill the pane background with a color gradient
        myPane.Fill = New Fill(Color.White, Color.FromArgb(220, 220, 255), 45.0F)

        ' Calculate the Axis Scale
        Ranges zgc.AxisChange()
        zgc.Refresh()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        Catch ex As Exception
            Timer1.Stop()
            MsgBox("No Data Shown", MsgBoxStyle.Exclamation)

        End Try
    End Sub

    Private Sub SetSize()
        zg1.Location = New Point(10, 10)

        ' Leave a small margin around the outside of the control
        zg1.Size = New Size(ClientRectangle.Width - 20, ClientRectangle.Height - 20)
    End Sub

    Private Sub Voltage_Load(sender As Object, e As EventArgs)
        Handles MyBase.Load
        Me.WindowState = FormWindowState.Maximized
        loadData_ICT()
        CreateGraph_ICT(zg1)
    End Sub

    Sub loadData_ICT()

        DateTimePicker1.Format = DateTimePickerFormat.Custom
        DateTimePicker1.CustomFormat = "yyyy/MM/dd"

        Dim tgl = DateTimePicker1.Value.ToString("yyyy-MM-dd")

    Try

```

```

        If SQLConn.State = ConnectionState.Closed Then SQL-
Conn.Open()
        Dim dataset As New DataSet
        Dim adapter As New SqlDataAdapter("SELECT
X_Bar_Volt,X_DBar_Volt FROM SPC_ControlChart WHERE Date = '" & tgl & "'
AND MachineName = '" & Machine & "' order by Time ASC", SQLConn)

        adapter.Fill(dataset)
        DataGridView1.DataSource = dataset.Tables(0)
        SQLConn.Close()

        DateTimePicker1.CustomFormat = "dddd, dd MMMM yyyy"
        DataGridView1.Visible = False

    Catch ex As Exception

    End Try
End Sub

Private Sub DateTimePicker1_ValueChanged(sender As Object, e As
EventArgs) Handles DateTimePicker1.ValueChanged

    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
End Sub

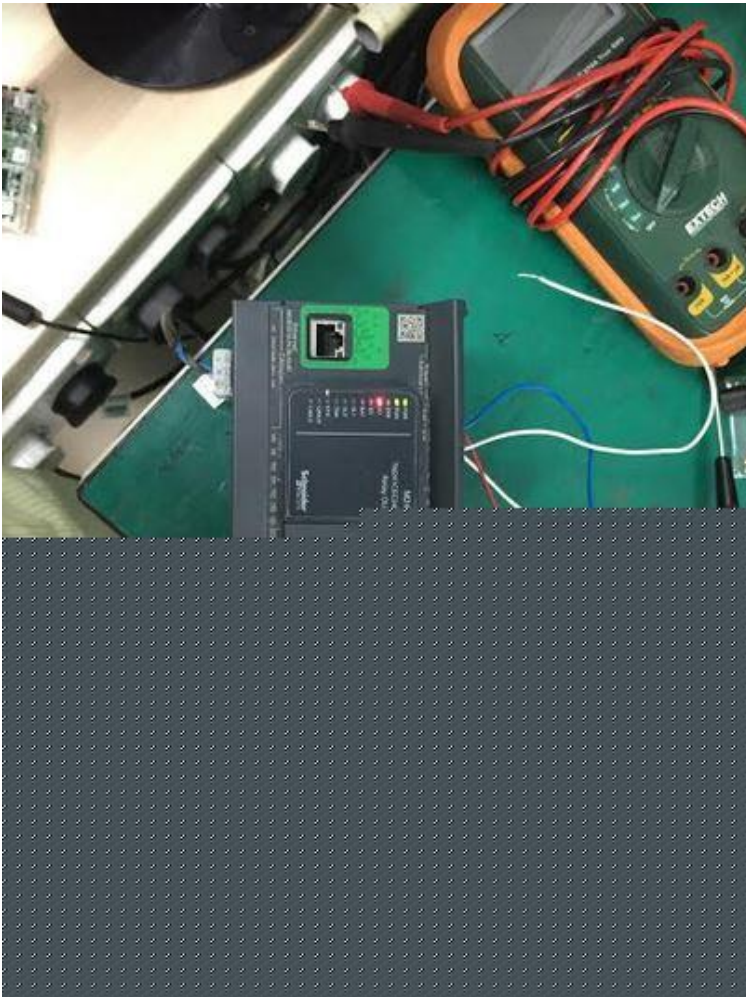
Private Sub Timer_ICT_Tick(sender As Object, e As EventArgs) Handles
Timer_ICT.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()
    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    zg1.GraphPane.CurveList.Clear()
    zg1.GraphPane.GraphObjList.Clear()

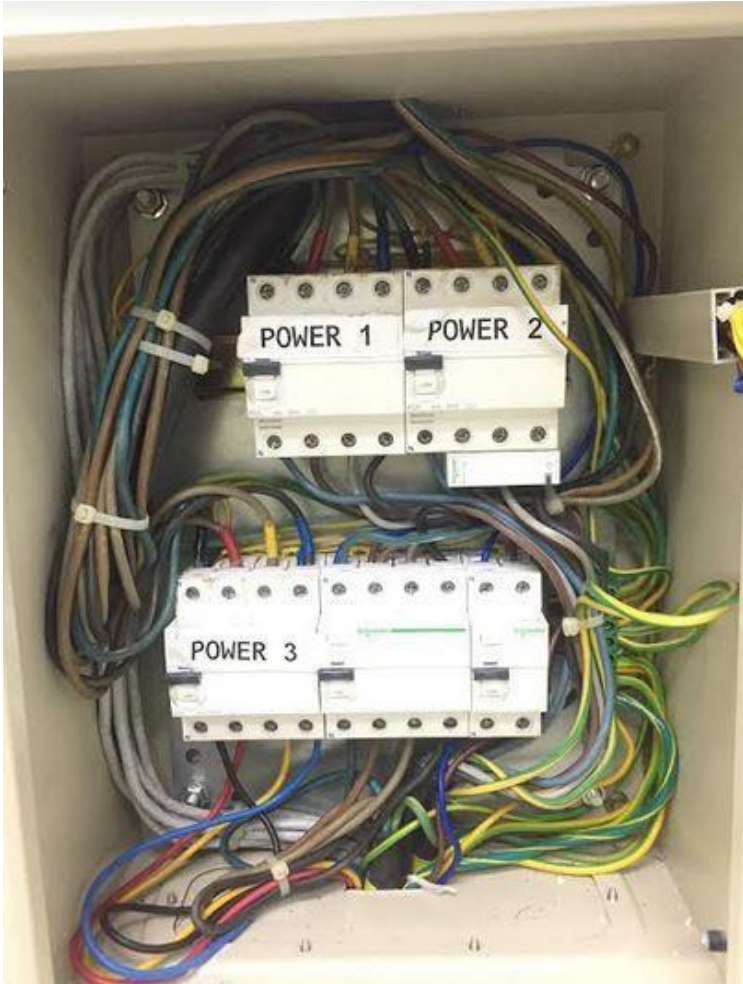
    zg1.Refresh()
    loadData_ICT()
    CreateGraph_ICT(zg1)
    Timer1.Stop()
End Sub
End Class

```

7. Percobaan PLC



8. Pemasangan PowerTag



- Halaman ini sengaja dikosongkan -

BIODATA PENULIS



Dhimas Adi Pangestu. Lahir di Bogor, Jawa Barat pada 1 Januari 1998. Penulis memulai jenjang pendidikan Sekolah Dasar di SD Muhammadiyah 1 Cileungsi dari tahun 2004-2010. Setelah itu penulis melanjutkan pendidikan di SMP Negeri 01 Cileungsi. Tahun 2013 penulis diterima sebagai siswa di SMA Negeri 01 Cileungsi. Pada tahun 2016, penulis melanjutkan jenjang pendidikan

Program Diploma di Institut Teknologi Sepuluh Nopember Surabaya dengan bidang studi Teknik Elektro.