ISBN: 978–623–91277–6–3

PERANCANGAN MONITORING *MACHINE CONDITION* DENGAN RASPBERRY Pi-ARDUINO WEMOS D1

Adi Rusdi Widya[1]*, Ahmad Turmudi Zy[2], dan Hendra Arya Syaputra[3]
Teknik Informatika, Universitas Pelita Bangsa, Bekasi [1,2]
Teknik Informatika, UPN "Veteran" Yogyakarta, Yogyakarta [3]
*E-mail: adirusdiw@pelitabangsa.ac.id

Abstract

The use of equipment for monitoring of a situation is needed to ensure the level of accuracy of data and the actual conditions, currently the use of information technology-based products and IoT equipment, makes it easy for designers and developer to create, design and develop a system application. The use of Raspberry, Arduino is one of the tools for developing and designing a machine condition monitoring system. Realtime information is currently needed to take quick and appropriate actions as needed in determining important and strategic decisions. Inaccurate information becomes an obstacle when dealing with problems that must be resolved requiring important and actual data. The development of raspberries and Arduino is used for monitoring system status of the engine (running/stop), Raspberry in Codesys functions as a PLC (master controller) and Arduino as a slave that is integrated directly with the engine. Communication between Codesys, raspberries, and Arduino uses wireless. The design of research using Raspberry-Pi, Arduino Wemos-D1 is one of the development tools and the design of machine monitoring condition (MMC) used for monitoring system status of the machine (running/stop). Human Machine Interface (HMI) for monitoring this system is web-based so it can be accessed through a browser. All devices (tablets/smartphones/notebooks) incorporated in one wireless network can access the machine system monitoring page.

Keywords: Arduino, MMCS, Raspberry, PLC, HMI.

Abstrak

Penggunaan peralatan untuk pengawasan (monitoring) suatu keadaan diperlukan untuk menjamin tingkat akurasi data dan kondisi yang sebenarnya, saat ini penggunaan produk berbasis teknologi informasi dan peralatan IoT, memberikan kemudahan bagi para perancang dan pengembangan untuk membuat, merancang dan mengembangkan suatu aplikasi sistem. Penggunaan Raspberry, Arduino merupakan salah satu tools pengembangan dan perancangan sistem monitoring machine condition. Informasi secara realtime saat ini diperlukan untuk melakukan tindakan cepat dan tepat sesuai kebutuhan dalam menentukan keputusan penting dan strategis. Informasi yang tidak akurat menjadi kendala saat menghadapi permasalahan yang harus diselesaikan membutuhkan data penting dan actual. Perancangan raspberry dan Arduino digunakan untuk system monitoring status mesin (running dan stop), Raspberry pada Codesys berfungsi sebagai PLC (master controller) dan Arduino sebagai slave yang diintegrasikan langsung dengan mesin. Komunikasi antara Codesys, raspberry, dan arduino menggunakan wireless. Perancangan penelitian menggunakan Raspberry-Pi, Arduino Wemos-D1 merupakan salah satu tools pengembangan dan perancangan monitoring machine

ISBN: 978–623–91277–6–3

condition (MMC) digunakan untuk sistem monitoring status mesin (running dan stop). Human Machine Interface (HMI) untuk monitoring system ini berbasis web sehingga dapat diakses melalui browser. Semua Device berupa tablet, smartphone, notebook) yang tergabung dalam satu jaringan wireless dapat mengakses halaman monitoring machine condition system.

Kata kunci: *Arduino, MMCS, Raspberry, PLC, HMI.*

1. PENDAHULUAN

Penelitian ini dilakukan untuk membantu mempercepat jalannya informasi mengenai kondisi mesin dan mesin yang mengalami kerusakan kepada pihak *maintenance* secara *real time*. Penggunaan komputer dan aplikasi yang dapat menunjang pekerjaan memberikan nilai lebih dalam menunjang proses perbaikan mesin sehingga proses produksi tidak terhambat. Perancangan *Monitoring Machine Condition System* (MMCS) diperlukan untuk menghilangkan proses informasi yang lama dari pengguna mesin kepada bagian perbaikan *maintenance*.

Penelitian ini bertujuan untuk:

1. Melakukan proses informasi secara *realtime* saat terjadi kerusakan mesin.
2. Membangun perancangan *Monitoring Machine Condition System* (MMCS) secara *wireless* dengan *hardware* & *software* yang murah, mudah, dan aman.

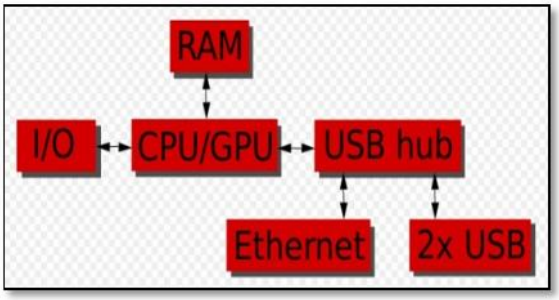
1.1 Raspbery

RaspberryPi adalah modul *micro computer* yang juga mempunyai *input output digital port* seperti pada *board micro controller*. Diantara kelebihan *Rasberry Pi* dibanding *board micro controller* yang lain yaitu mempunyai *port*/koneksi untuk *display* berupa TV atau *Monitor PC* serta koneksi USB untuk *Keyboard* serta *Mouse* (seperti tampak pada Gambar 2 dan 4). *Raspberry Pi* dibuat di Inggris oleh *Raspberry Pi Foundation*. Pada awalnya *Raspberry Pi* ditujukan untuk modul pembelajaran ilmu komputer disekolah.

1.1.1 *Raspberry Pi* Board

Raspberry Pi board dibuat dengan type yang berbeda yaitu *Raspberry Pi type* A, *A+ Raspberry Pi type B, B+ Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi zero*. Perbedaannya antara lain pada *RAM* dan *Port LAN*. Type A *RAM* = 256 MB dan tanpa *port LAN* (*ethernet*), type B = 512 MB dan terpasang *port* untuk *LAN*.

1.1.2 Blok Diagram *Raspberry Pi*



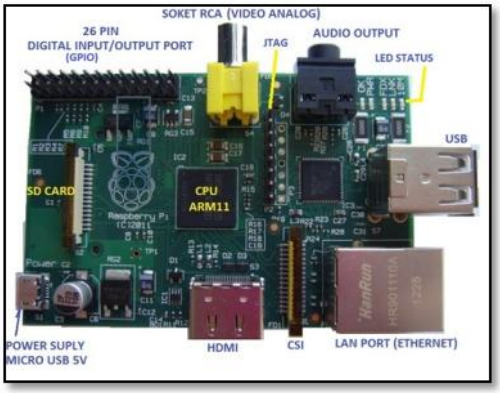
Gambar 1. Blok Diagram *Raspberry Pi*

Raspberry Pi board mempunyai *input dan output* antara lain:

a. *HDMI*, dihubungkan ke *LCD TV* yang mempunyai *port HDMI* atau dengan *cable converter HDMI to VGA* dapat dihubungkan ke *monitor PC.*
b. Video analog (*RCA port*), dihubungkan ke Televisi sebagai alternatif jika tidak memilih *monitor PC*.
c. *Audio output*
d. 2 buah *port USB* digunakan untuk *keyboard* dan *mouse*

ISBN: 978–623–91277–6–3

e. 26 *pin I/O digital*
f. *CSI port* (*Camera Serial Interface*)
g. *DSI* (*Display Serial Interface*)
h. *LAN port* (*network*)
i. *SD Card slot* untuk *SD Card* memori yang menyimpan sistem operasi berfungsi seperti *hardisk* pada *PC.*



Gambar 2. *Raspberry Pi Board*

GPIO merupakan sederet pin yang terdiri dari 26 pin dengan berbagai fungsi diantaranya:

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Gambar 3.*GPIO* 26 Pin

Selain sebagai *input output* pada beberapa pin GPIO juga berfungsi sebagai komunikasi serial diantaranya I2C, SPI dan serial komunikasi UART. Berikut ini merupakan contoh *Raspberry Pi board* dengan *casing* tampak lebih indah.

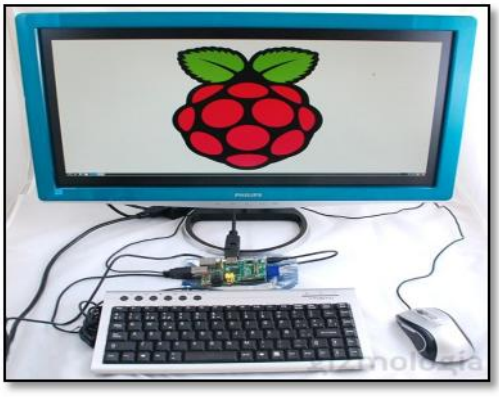
Gambar 4. *Raspberry Pi*

1.2. Tipe *Raspberry*

1.2.1 *Raspberry Pi Operating System*

Dalam menggunakan *Raspberry pi* kita memerlukan *operating system* (contoh OS: *windows, linux, mac, Unix,* dst) yang dijalankan dari *SD card* pada *board Rasberrry* tidak seperti pada *board microcontroller AVR* yang selama ini dipakai tanpa OS. *Operating System* yang banyak dipakai antara lain *Linux distro Raspbian*. OS disimpan di *SD card* dan saat proses *boot* OS hanya bisa dari SD card tidak dari lokasi lain.

OS yang bisa dijalankan di *Raspberryboard* antara lain: *Arch Linux ARM, Debian GNU/Linux, Gentoo, Fedora, FreeBSD, NetBSD, Plan 9, Inferno, Raspbian OS, RISC OS* dan *Slackware Linux.* Jadi dalam menggunakan *microcomputer Raspberry Pi* ini seperti menggunakan PC yang berbasis *linux plus* yang mempunyai *input output digital* seperti yang ada di *board microcontroller*. Berikut ini contoh 1 set *micro computer Raspberry Pi* dengan *OS LINUX Rasbian* yang siap pakai (Gambar 5).



Gambar 5. *Raspberry Pi* Siap digunakan

ISBN: 978–623–91277–6–3

1.3. Pengertian Arduino

Arduino adalah sebuah pengendali *mikro board* tunggal yang memiliki sifat terbuka (*open source*) yang diturunkan dari *platform* berbasis *Wiring*. Pengendali ini dirancang untuk mempermudah penggunaan dalam berbagai bidang elektronik. *Hardware arduino* mengandung prosesor jenis Atmel AVR, dan memiliki bahasa pemrograman tersendiri.

Perlu diketahui bahwa *Arduino* masih masuk dalam keluarga mikrokontroler *AT Mega* buatan *Atmel*. Namun seiring perkembangannya, banyak perusahaan lain yang membuat kloningan dari *arduino* dengan jenis mikrokontrol lainnya. Banyak pemula menggunakan *arduino* karena dianggap lebih mudah dipelajari maupun digunakan. Akan tetapi, tak jarang para professional menggunakan *arduino* untuk dikembangkan menjadi berbagai macam aplikasi elektronik. Sekedar informasi bahwa *arduino* menggunakan bahasa pemrograman arduino dengan *syntax* menyerupai bahasa pemrograman C. Karena sifatnya yang *open source,* semua orang bebas mengunduh skema *hardware*nya untuk dikembangkan.

Kelebihan *arduino* dibandingkan dengan pengendali mikro lain diantaranya adalah harganya yang relatif murah, pemrogramannya yang bersifat mudah dan sederhana, bebas digunakan karena bersifat *open source*, tak memerlukan *hardware* tambahan seperti *chip,* konektor USB, dan masih banyak lagi yang lainnya. Arduino juga bisa langsung terkoneksi dengan modul lain seperti GPS dan *ethernet*.

Arduino juga memiliki beberapa jenis seperti *arduino uno, arduino due, arduino mega, arduino leonardo, arduino fio, arduino lilypad, arduino nano, arduino mini, arduino micro, arduino ethernet, arduino esplora*, dan *arduino robot.* Masing-masing *arduino* tersebut memiliki ciri yang berbeda-beda.

1.3.1 Fungsi *Arduino*

Secara umum *arduino* memiliki fungsi memudahkan penggunaan dalam berbagai bidang elektronik seperti pembuatan aplikasi *running LED, traffict LED, mobile robot,* dan masih banyak lagi yang lainnya. Dengan menggunakan *arduino,* pembuatan aplikasi-aplikasi tersebut menjadi lebih praktis, mudah, dan murah.

1.4. Codesys

Codesys adalah akronim untuk sistem pengembangan pengendali, *Codesys* yang sebelumnya bergaya) adalah lingkungan pengembangan untuk aplikasi pengontrol pemrograman sesuai dengan standar industri internasional IEC 61131-3.

Codesys dikembangkan dan dipasarkan oleh perusahaan perangkat lunak Jerman 3S-Smart *Software Solutions* yang berlokasi di Kota Kempten, Bavaria. Versi 1.0 dirilis pada tahun 1994.

Lisensi *Codesys* tidak dipungut biaya dan dapat dipasang secara legal tanpa perlindungan salinan pada *workstation* lebih lanjut. Perangkat lunak mencakup berbagai aspek teknologi otomasi industri dengan satu permukaan.

Semua lima bahasa pemrograman untuk pemrograman aplikasi yang didefinisikan dalam IEC 61131-3 tersedia di lingkungan pengembangan *Codesys.* (Anonim, 2019)

a. IL (*Instruction List*) adalah assembler seperti bahasa pemrograman (sekarang sudah usang, tapi tersedia untuk kompatibilitas mundur.
b. ST (*Text Structure*) mirip dengan pemrograman di Pascal atau C.
c. LD (*Ladder Diagram*) memungkinkan pemrogram untuk menggabungkan

ISBN: 978–623–91277–6–3

kontak *relay* dan gulungan secara keseluruhan.

d. FBD (*Function Block Diagram*) memungkinkan pengguna untuk dengan cepat memprogram ekspresi Boolean dan analog.
e. SFC (*Squential Function Chart*) mudah digunakan untuk memprogram proses sekuensial dan arus editor grafis tambahan tersedia di *CODESYS* yang tidak didefinisikan dalam standar IEC.
f. CFC (*Continuous Function Chart*) adalah semacam editor *FBD freehand*. Selain editor FBD yang berorientasi jaringan dimana koneksi antara *input, operator*, dan *output* ditetapkan secara otomatis, mereka harus ditarik oleh *programmer*. Semua kotak dapat ditempatkan secara bebas yang memungkinkan untuk memprogram *loop* umpan balik tanpa variabel interim.

Komposer aplikasi *Codesys* berfungsi untuk membuat aplikasi dengan menggunakan modul yang ada. *User composes, parameterizes*, dan menghubungkan modul yang dibutuhkan untuk membentuk aplikasi yang lengkap. Konfigurasi ini tidak memerlukan pengetahuan pemrograman PLC dan oleh karena itu dapat dilakukan oleh teknisi tanpa pengalaman pemrograman. Generator internal membuat aplikasi *IEC 61131-3* yang lengkap dan terstruktur dengan baik termasuk pemetaan I dan visualisasi. Komposer Aplikasi memerlukan lisensi untuk mengembangkan dan menyusun modul.Selanjutnya ada modul yang dapat digunakan secara gratis (yaitu, Kegigihan Manager, *Device* Diagnosis), yang dapat digunakan tanpa lisensi.

1.4.1 *Runtime*

Setelah menerapkan Codesys *Control Runtime System*, perangkat cerdas dapat diprogram dengan Codesys. Tool kit yang dibebankan untuk menyediakan sistem *runtime* ini sebagai sumber dan kode obyek. Hal ini dapat *porting* ke *platform* yang berbeda.

1.4.2 Teknologi *Fieldbus*

Bus lapangan yang berbeda dapat digunakan secara langsung dalam sistem pemrograman *Codesys*. Adapun tujuan alat ini untuk mengintegrasikan konfigurator untuk sistem yang paling umum seperti *PROFIBUS, CANopen, Ether CAT, PROFINET* dan EtherNet/IP. Untuk beberapa sistem, tumpukan protokol tambahan tersedia dalam bentuk *library Codesys* yang dapat dimuatkan selanjutnya. Dengan menggunakan *plugin* perangkat lunak dalam aplikasi *Frame* FDT (Perangkat Perangkat Lapangan), antarmuka pengguna khusus perangkat tambahan dari pemasok pihak ketiga dapat diintegrasikan. Komunikasi antar muka ini akan diwujudkan melalui *Communication Device Type Manager* (DTM).

1.4.3 Visualisasi

Editor terintegrasi membantu pengguna membuat masker visualisasi yang kompleks secara langsung dalam sistem pemrograman *Codesys* dan menghidupkannya berdasarkan *variable* aplikasi. Untuk menyederhanakan prosedur, elemen visualisasi terpadu tersedia. *Toolkit* opsional memungkinkan pengguna membuat elemen visualisasi sendiri. Masker yang dibuat antara lain digunakan untuk tes aplikasi dan *commissioning* selama pengoperasian sistem pemrograman secara *online*.

Dalam kombinasi dengan klien visualisasi opsional, masker juga dapat digunakan untuk mengoperasikan mesin atau tanaman, contohnya pada kontroler dengan

ISBN: 978–623–91277–6–3

tampilan terintegrasi, di *web browser* atau *runtime* portabel di bawah *Windows* atau *Linux*.

1.4.4 Soft Motion

Solusi modular opsional untuk mengendalikan gerakan kompleks dengan pengendali terprogram *IEC 61131-3* juga terintegrasi sepenuhnya dalam sistem pemrograman *Codesys*. Solusi modular meliputi:

a. Editor untuk perencanaan gerak, e. g. dengan deskripsi *CAMC* atau *DIN 66025 CNC*.
b. Modul perpustakaan untuk *decoder*, *interpolator*, untuk eksekusi program, e.g. Menurut PLC *open Motion Control*, untuk transformasi kinematis dan *template* visualisasi *Safety* untuk mencapai tingkat integritas keselamatan (SIL) yang dibutuhkan setelah analisis risiko, semua komponen sistem harus mematuhi tingkat ini.

Komponen perangkat lunak pra-sertifikasi di dalam *Codesys* mempermudah produsen perangkat menyetel pengontrol SIL2 atau SIL3 mereka. Oleh karena itu, *Codesys Safety* terdiri dari komponen dalam sistem pemrograman dan sistem *runtime*, sedangkan perencanaan proyek benar-benar terintegrasi dalam lingkungan pemrograman *IEC 61131-3*.

Penggunaan industri lebih dari 250 produsen perangkat dari berbagai sektor industri menawarkan perangkat otomasi cerdas dengan sebuah antarmuka pemrograman *Codesys*. Akibatnya, ribuan pengguna akhir seperti pembangun mesin atau pabrik di seluruh dunia menggunakan *Codesys* untuk segala jenis tugas otomasi. Direktori Perangkat *Codesys* menawarkan ikhtisar tentang hampir 400 perangkat yang dapat diprogram dengan *Codesys* dari produsen terkemuka dari sektor teknik kontrol, komponen otomasi dan *embedded system*.

Penelitian yang dilakukan oleh Ayu Kusuma dkk, 2018, mengenai Rancang Bangun *Smarthome* Menggunakan *Wemos D1 R2 Arduino Compatible* Berbasis *ESP8266 ESP-12F*, dimana penelitian ini bertujuan untuk membangun perangkat lunak dan perangkat keras *Smarthome Wemos D1 R2 Arduino compatible* berbasis *ESP8266 ESP-12F*. Dengan metode eksperimen dan studi pustaka, penelitian ini telah berhasil merancang bangun *Smarthome* menggunakan *Wemos D1 R2 Arduino compatible* berbasis *ESP8266 ESP-12F*. Hasil penelitian ini sudah sesuai dengan tujuannya yaitu membangun perangkat lunak dengan alamat IP yang digunakan 192.168.43.52 (setiap *Wemos* memiliki *IP address* yang berbeda-beda) dan perangkat keras dengan mensimulasikan *smarthome* menggunakan LED maupun alat-alat elektronika.

Penelitian yang dilakukan oleh Ardianto A, dkk 2016, mengenai Sistem Monitoring Pencemaran Polutan Kendaraan Via Gadget Berbasis *Arduino*, yang dimana Tingkat deteksi aplikasi dari sistem pembuangan kendaraan bermotor yang dibuat digunakan untuk mendeteksi jumlah kadar gas H_2, gas EtOH dan gas CO dengan menggunakan sensor *TGS2201*. Data dari sensor diolah oleh *arduino* dan hasilnya ditampilkan pada PC melalui *port serial*, desain program aplikasi menggunakan Bahasa C.

Penelitian yang dilakukan oleh Kusuma dan Tirta, 2018, mengenai Perancangan Sistem Monitoring Infus Berbasis *Mikrokontroler Wemos D1 R2*, dimana pada penelitian dilakukan untuk mengatasi keterlambatan tersebut dibantu

ISBN: 978–623–91277–6–3

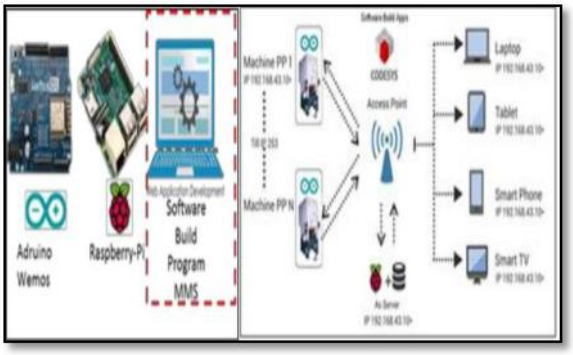
dengan menggunakan mikrokontroler, studi literatur dan melakukan eksplorasi terhadap perangkat keras seperti; papan *arduino*, sensor, modul-modul, dan perangkat lunak yang digunakan.

Penelitian yang dilakukan oleh Supegina dkk, 2017, mengenai Rancang Bangun IOT *Temperature Controller* untuk *Enclosure* BTS Berbasis *Microcontroller Wemos* dan *Android*, dimana *Enclosure* pada perangkat *Base Transceiver Station* (BTS) dirancang dan didesain untuk kondisi *outdoor* terutama cuaca panas. Alat ini dirancang menggunakan sensor suhu *DHT11* sebagai sumber informasi data untuk diolah mikrokontroler *Wemos*. Apabila suhu melebihi batas suhu yang ditentukan, maka akan otomatis menggerakkan kipas DC dan bila suhu kembali normal, maka secara otomatis kipas DC akan berhenti berputar. Selain itu, alat ini juga dirancang dapat bekerja secara manual dan dikontrol melalui *App Blynk* dari *smartphoneAndroid* secara *wireless*.

Penelitian yang dilakukan oleh Rachmat dkk, 2014, mengenai Implementasi *Counter Production Monitoring* pada mesin, dimana sistem ini digunakan pada mesin pemintal benang di industri tekstil untuk mengetahui efisiensi kerja mesin dan hasil produksi benang. Fungsi mikrokontroler pada penelitian ini yaitu untuk menghitung tiga besaran produksi dan menampilkan hasil pada *display LCD* melalui pengaturan 3 buah tombol serta mengirimkan ke PC melalui komunikasi serial untuk ditampilkan.

2. METODE PENELITIAN

Dari beberapa referensi penelitian yang telah dilakukan, maka penelitian yang akan dikembangkan adalah penggunaan *Raspberry-Pi* dan *Wemos-D1* untuk Perancangan "*Monitoring Machine Condition System (MMCS)*" dengan tujuan membantu pihak *user* (Produksi &*Maintenance*) dapat menurunkan *down time*apabila terjadi kerusakan mesin. Perancangan yang dilakukan ditunjukkan pada Gambar 6.



Gambar 6. Perancangan MMC system

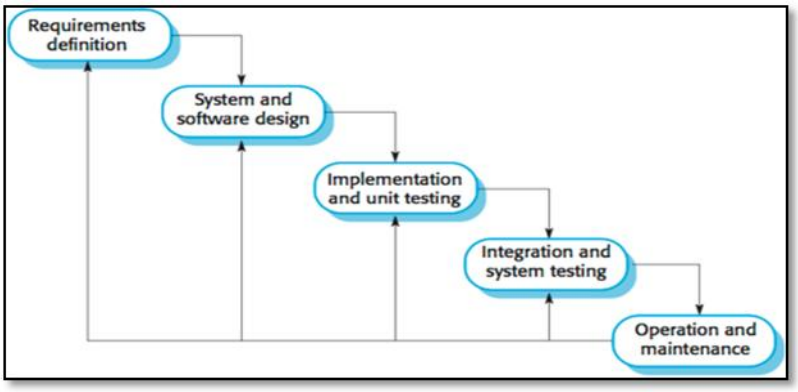
3. HASIL DAN DISKUSI

Pembuatan program terbagi dalam 4 bagian, yaitu:

a. Pembuatan *hotspot* dan *setting* alamat IP
b. Setting *Raspberry*
c. Pemrograman *Codesys*
d. Pemrograman *Arduino*

Salah satu metode pengembangan perangkat lunak yaitu *waterfall*/air terjun. Menurut Sommerville (2011) [7], tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*. Detail metode seperti pada Gambar 7.

ISBN: 978–623–91277–6–3



Gambar 7. Metode *Waterfall* versi Sommerville

Tahapan-tahapan dari Metode *Waterfall* adalah sebagai berikut:

1. *Requirement Analysis and Definition*

Pada fase perencanaan sistem ini kita terlebih dahulu harus merencanakan tentang *project* apa yang akan kita buat atau dengan kata lain kita harus mendefinisikan masalah yang harus dipecahkan. Hal tersebut meliputi tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design*

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya. Pada fase ini akan dilakukan desain pada sistem sebelum melakukan pengkodean. Tahap ini bertujuan untuk memberikan gambaran apa yang harus dikerjakan dan bagaimana tampilannya. Tahap ini juga membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation and Unit Testing*

Pada tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu *set* program atau unit program. Pembuatan *software* dipecah menjadi modul-modul kecil yang akan digabungkan dalam tahap berikutnya. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and System Testing*

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada dan dilakukan pengujian untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak. Setelah itu sistem akan dikirim ke pengguna sistem.

5. *Operation and Maintenance*

Pada fase *maintenance software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Pembuatan aplikasi pada penelitian ini hanya sampai pada tahap *integration and testing* saja.

3.1 Pembuatan *Hotspot*

Rancangan sistem ini menggunakan komunikasi *via wireless*, maka hal pertama yang harus dilakukan adalah membuat *hotspot*. *Hotspot* dapat dibuat melalui *router*/*access point* atau *internal hotspot via* PC. Adapun yang gunakan saat ini adalah

ISBN: 978–623–91277–6–3

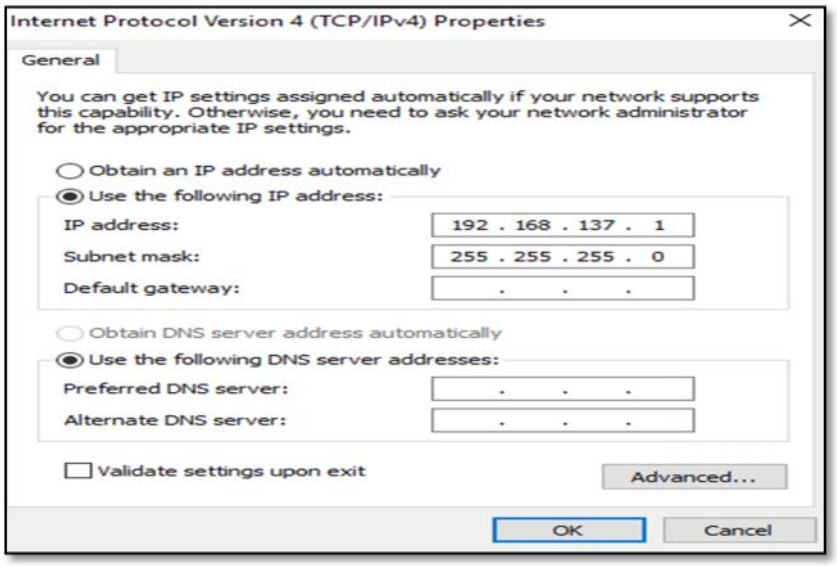
internal hotspot. Untuk Windows 10, *hotspot* dapat dibuat melalui menu *command prompt*.

Netsh wlan set hostednetwork mode=allow ssid=nabilla key=12345678

SSID dan *password* dapat diganti sesuai keinginan. Kemudian aktifkan *hotspot* yang telah dibuat dengan perintah:

```
netsh wlan start hostednetwork
```

Setelah itu atur alamat IP seperti berikut:



Gambar 8. Pembuatan *Hotspot*

3.2 Setting Raspbery

Untuk pengaturan pada *raspberry*, pertama-tama, sambungkan *raspberry* dengan *hotspot* yang telah dibuat.

a. Aktifkan *VNC viewer* pada terminal window, ketik:

```
sudo raspi-config
```

Pilih *Interfacing option*, kemudian *enable VNC server*.

Note: *VNC viewer* berfungsi untuk mengakses *raspberry via* PC.

b. *Setting IP raspberry* pada terminal window, ketik:

```
sudo nano /etc/dhcpcd.conf
```

Kemudian tambahkan perintah ini:

```
interface wlan0
static ip_address=192.168.137.62
static netmask=255.255.255.0
static gateway=192.168.137.1
```

Kemudian ketik:

```
sudo nano /etc/network/interfaces
```

Cari *line iface wlan0 inet dhcp*, ganti dengan:

```
iface wlan0 inet static
address 192.168.137.62
netmask 255.255.255.0
gateway 192.168.137.1
```

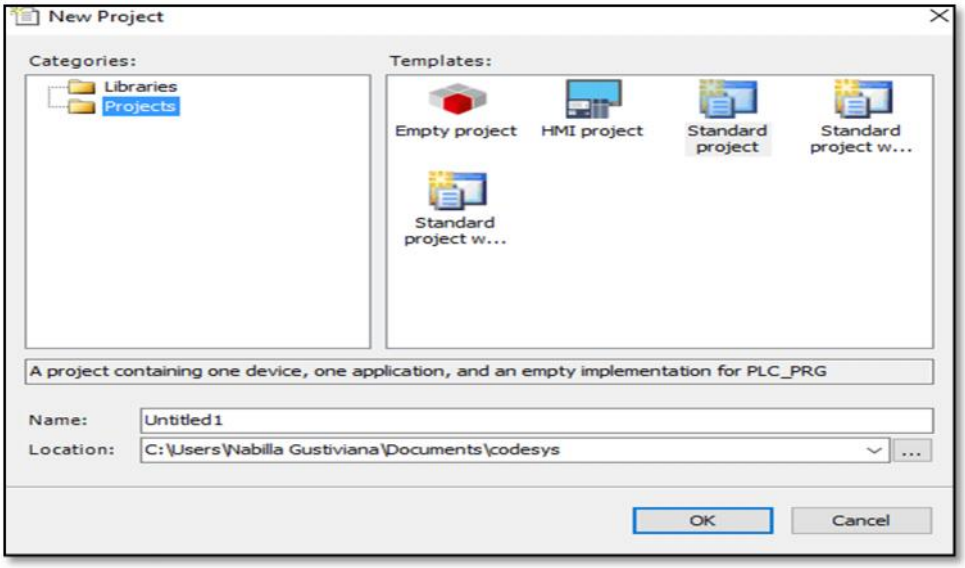
Untuk memastikan bahwa *raspberry* telah terhubung dengan *hotspot*, cek dengan menggunakan *IP scanner*, apakah *IP raspberry* telah berada dalam jaringan. Jika ingin mengakses *raspberry via* PC, buka aplikasi *VNC viewer* pada PC, ketikkan *IP raspberry* (*user=pi password=raspberry*).

ISBN: 978–623–91277–6–3

3.3 Pemrograman *Codesys*

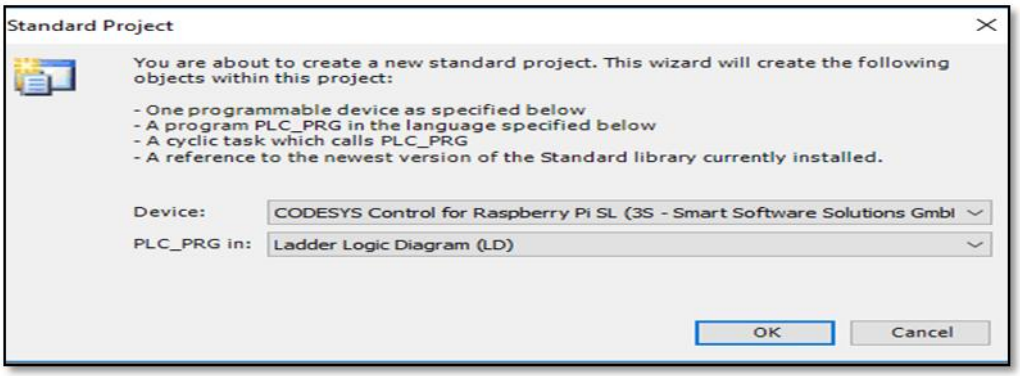
Berikut langkah pemrograman *Codesys*:

a. Buat file baru dengan menu *File*, *New Project*, pilih *Standard Project* seperti pada Gambar 9.



Gambar 9. Pemograman *Codesys*

b. Pilih menu *Codesys Control for Raspberry Pi*, seperti pada Gambar 10.



Gambar 10. Pilih Menu *Codesys*

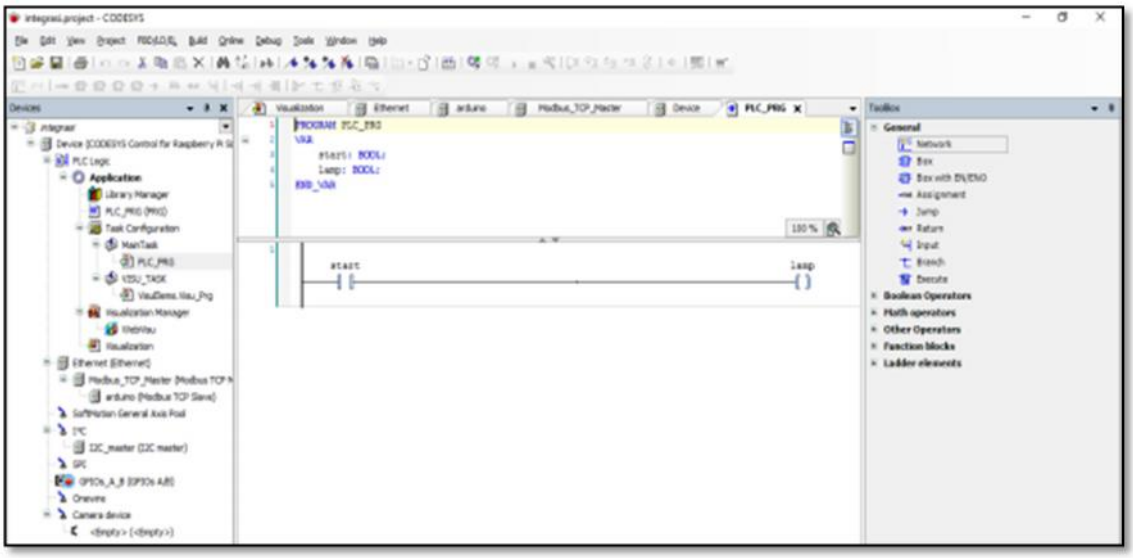
Pemograman dengan *Codesys* dapat dibuat dalam beberapa Bahasa, seperti *Ladder, Structured Text, Block Diagram,* dan lain lain. Disini yang digunakan yaitu *Ladder Diagram*.

c. *Download package raspberry* pada *Codesys store*. *Install package* tersebut pada menu *Tools*, *Package Manager*.

d. Pada menu *Tools*, *Update Raspberry Pi*, ketikkan alamat *IP raspberry* di kolom *IP address*. Kemudian isi kolom *login* dengan *username pi* dan *password raspberry*.

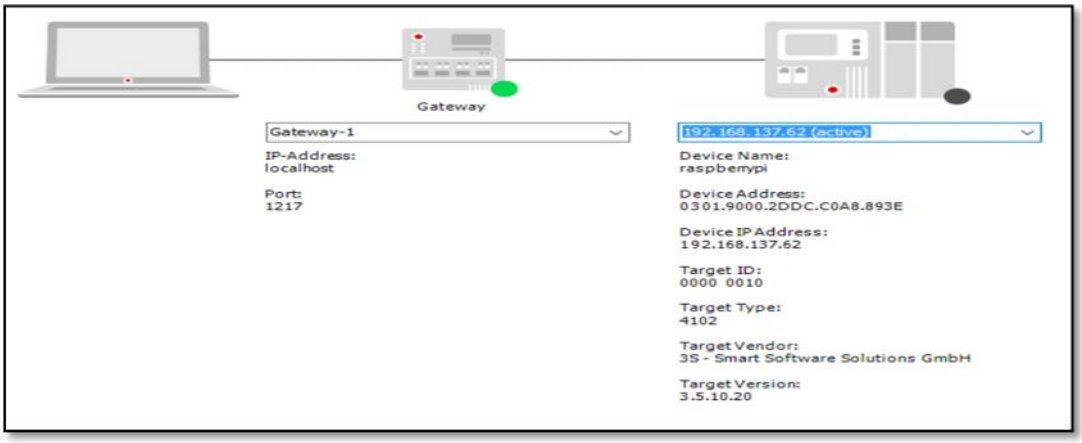
e. Buat program pada *sub menu PLC_PRG*. Sebagai contoh program untuk mengaktifkan lampu (Gambar 11).

ISBN: 978–623–91277–6–3



Gambar 11. Contoh Program PLC

f. Untuk membuat halaman HMI, klik kanan pada menu *Application*, *Add Object*, kemudian *Visualization*.
g. Elemen pada halaman HMI dapat diambil dari *toolbox* dengan melakukan *drop* & *drag*.
h. Masukkan variabel pada tiap elemen HMI sesuai dengan program *ladder* yang telah dibuat.
i. Untuk melakukan koneksi pada *raspberry* klik menu *Device*, kemudian ketikkan *IP addres* dari *raspberry.*



Gambar 12. IP Address

Setelah koneksi berhasil, klik menu *Online*, lalu *Login*. Kemudian menu *Debug*, lalu *Start*.

3.4 Pemrograman *Arduino*

Pemrograman *arduino* dilakukan dengan *Arduino IDE*. Berikut adalah program *Arduino*:

```
#include <ESP8266WiFi.h>
#include <Modbus.h>
#include "ModbusIP_ESP8266.h"

//Modbus Registers Offsets (0-9999)
const int LED_COIL1 = 0010;
const int BUTTON = 0000;

//Used Pins
const int ledPin1 = D7;
```

ISBN: 978–623–91277–6–3

```
const int buttonPin = D5;

//ModbusIP object
ModbusIP mb;
```

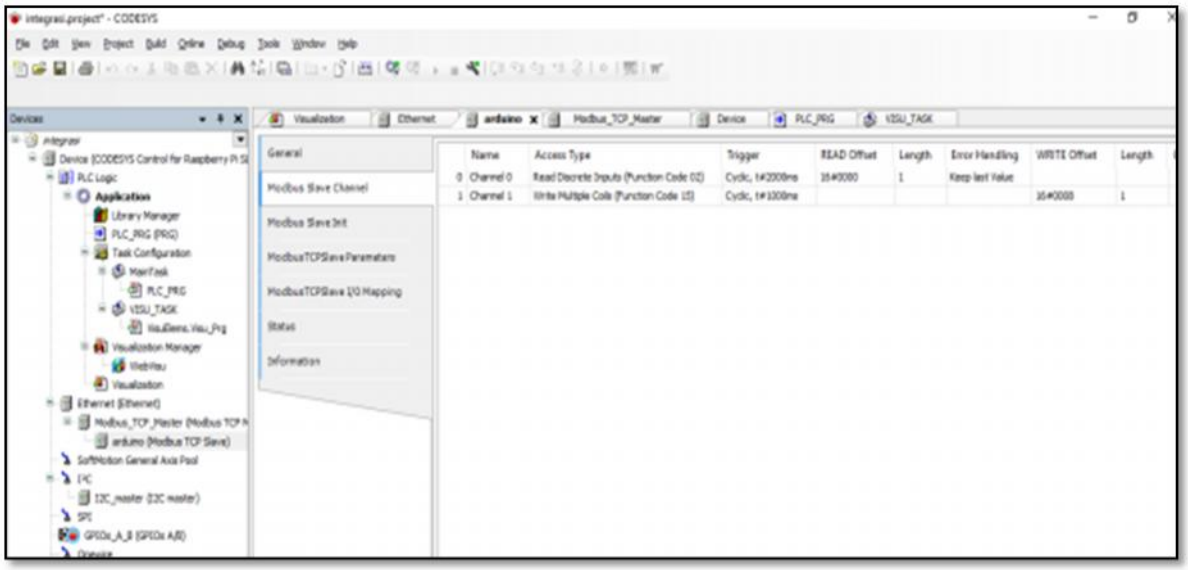
```
void setup()
{
Serial.begin(115200);
mb.config("nabilla", "12345678");

while(WiFi.status()!=WL_CONNECTED) {
delay(500);
Serial.print(".");
  }

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

pinMode(ledPin1, OUTPUT);
pinMode(buttonPin, INPUT);
mb.addCoil(LED_COIL1);
mb.addIsts(BUTTON, LOW);
}

void loop()
{
```

```
   //Call once inside loop() - all magic here
mb.task();

   //Attach buttonPin to BUTTON register
mb.Ists(BUTTON,digitalRead(buttonPin));

   //Attach ledPin to LED_COIL register
digitalWrite(ledPin1,mb.Coil(LED_COIL1));

Serial.print("pin D5= ");
Serial.println(digitalRead(buttonPin));
Serial.print("pin D7= ");
Serial.println(digitalRead(buttonPin));
}
```

Keterangan:

a. Lampu di*setting* dengan *register 0010* (pengaturan pada *Codesys*) pada *pin D7* di *board Arduino*.
b. *Push button* di*setting* dengan *register 0000* (pengaturan pada *Codesys*) pada *pin D5* di *board Arduino*.
c. *Arduino* dikoneksikan pada jaringan *hotspot* yang telah dibuat.

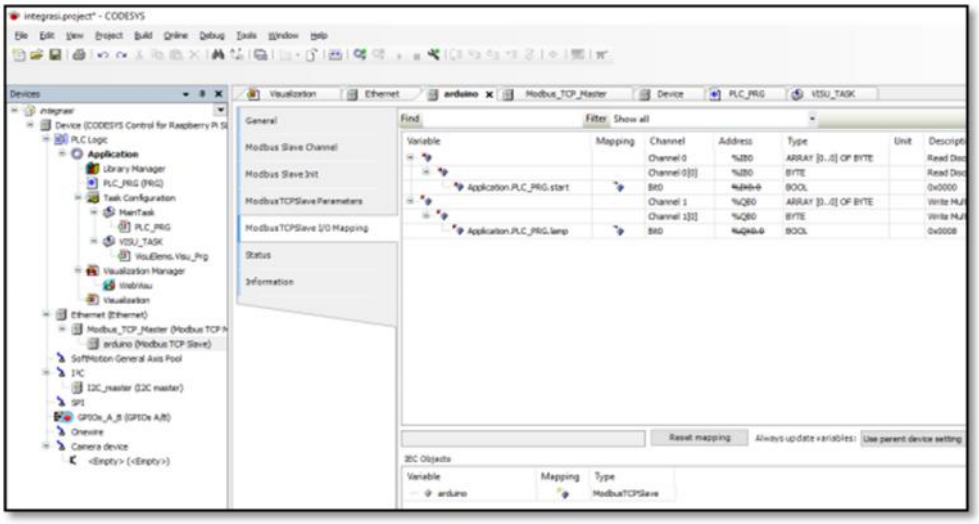
Pada pengaturan di *Codesys*, sesuaikan *register* dengan *register* yang telah di*setting* pada *Arduino*, seperti Gambar 13.



Gambar 13. *Setting Codesys*

ISBN: 978–623–91277–6–3

Terakhir, *setting I/O*:



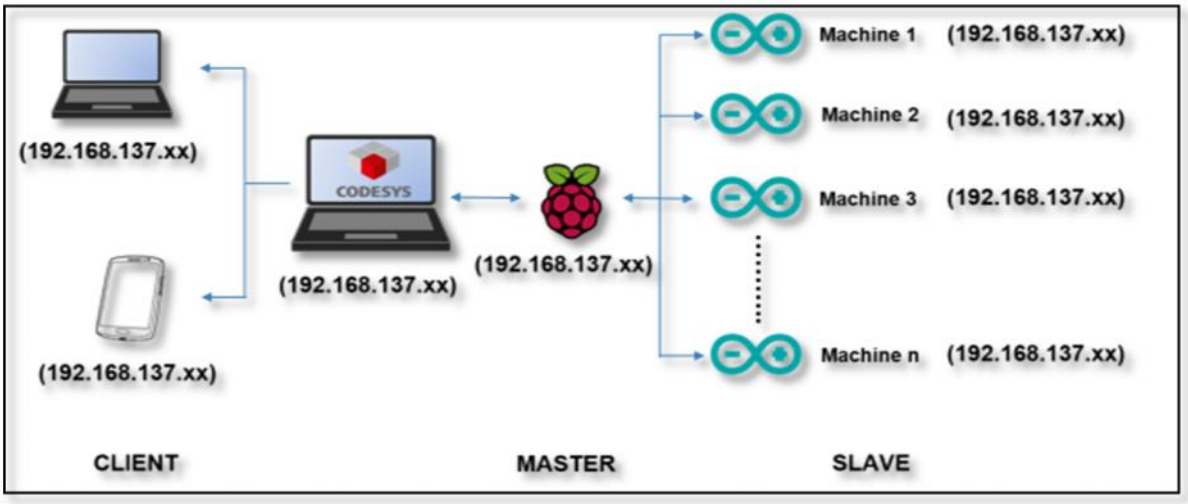
Gambar 14. *Setting I/O*

Program ini, dapat diakses melalui *browser* dengan mengetik:

192.168.137.62:8080/webvisu.htm

*dapat diganti dengan alamat IP pada *raspberry*.

Konfigurasi *Monitoring machine condition* dapat dilihat pada Gambar 15.

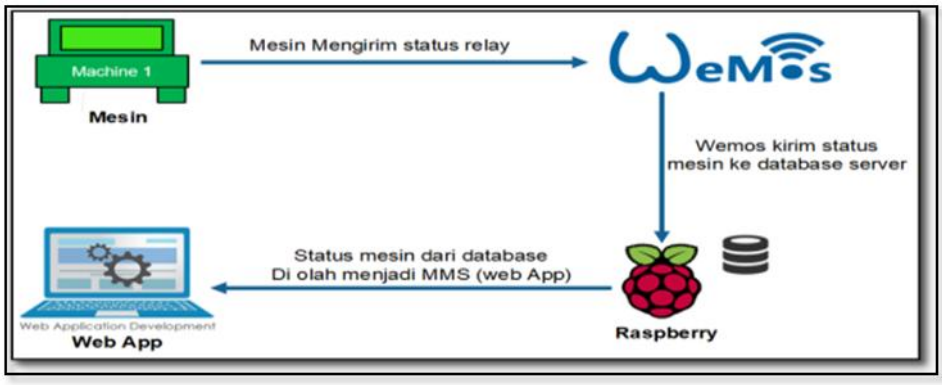


Gambar 15. Konfigurasi *MMC System*

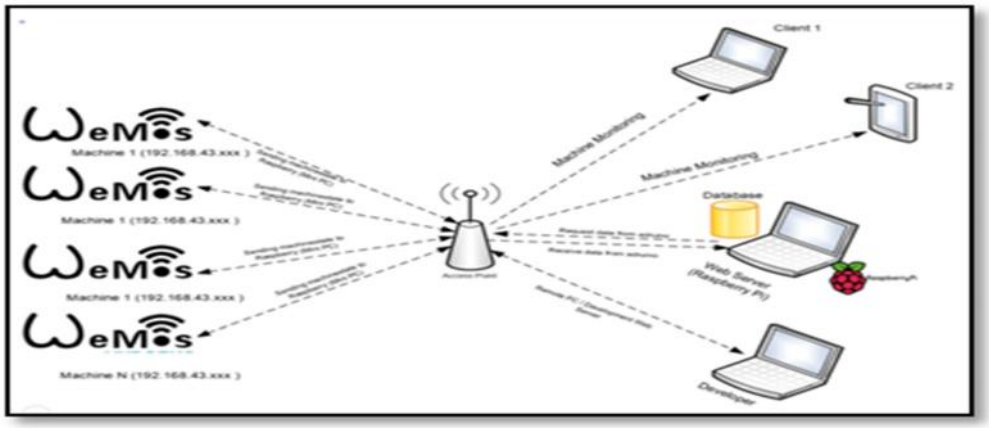
3.5 Konsep dan Rancangan Topologi Sistem MMCS *Web App*

Konsep dari *Machine Monitoring System* (MMCS) dengan menggunakan *Web Application Development* dapat dilihat pada Gambar 16. Sedangkan topologi sistem yang akan dibangun seperti yang terlihat pada Gambar 17.

ISBN: 978–623–91277–6–3



Gambar 16. Konsep MMCS dengan *Web Application Development*



Gambar 17. Topologi Sistem MMCS

Pada Gambar 17, dapat dijelaskan bahwa Wemos akan dipasang pada mesin. Kemudian mesin akan mengirim status mesin ke wemos, dan status tersebut akan diteruskan menuju ke *database* pada *Raspberry-Pi* melalui perantara *access point*. Setelah itu data yang ada di *Raspberry-Pi* akan diolah menjadi data visual menjadi MMCS oleh *developer* dan Kemudian dapat dimonitoring dalam berbagai *platform* seperti android, PC, dan lain sebagainya.

4. KESIMPULAN

Kesimpulan dari penelitian ini adalah:

a. Perancangan *Monitoring Machine Condition* sangat berguna untuk memberikan informasi status mesin yang sedang digunakan sehingga proses produksi dapat di monitor secara akurat.
b. Perancangan dapat dikembangkan dan disesuaikan dengan kebutuhan yang ada di dunia industri dan perkembangan monitoring dengan menggunakan alat monitoring berupa PC, *Notebook,* dan *smartphone device*.

Saran

Perancangan ini membutuhkan perbaikan dengan mencari dan membuat *monitoring system* yang dapat mudah diaplikasikan, akurat dan murah dalam pembiayaannya, sehingga dapat menyesuaikan dengan kebutuhan pengguna industri dan jasa, kerjasama tetap harus di tingkatkan dengan beberapa pihak sehingga pengembangan alat *monitoring* dapat menjadi alternatif pilihan dan sama fungsinya dengan seperti *system SCADA* dengan berbiaya lebih murah.

ISBN: 978–623–91277–6–3

Penghargaan/Ucapan Terima Kasih

Peneliti mengucapkan kepada para pihak dan Tim, berkaitan dengan perancangan MMCS, sehingga dapat selesai melakukan penelitian dan penulisan hasil penelitian.

DAFTAR PUSTAKA

Anonim. *Codesys Application Composer*. Diakses dari https://www.Codesys.com/products/Codesys-engineering/application-composer.html**,** tanggal 21 Agustus 2019.

Ardianto, A., Khasanah, U., Murdianto, Brian, D., Wulandari, Bekti, 2016. Sistem Monitoring Pencemaran polutan Kendaraan *via Gadget* Berbasis *Arduino*. Teknik, Universitas Negeri Yogyakarta.

Ian Sommerville. 2011. *Software Engineering*: Rekayasa Perangkat Lunak. Edisi 6. Jakarta: Penerbit Airlangga.

Kusuma, T., Tirta, Mulia M., 2018. Perancangan Sistem Monitoring Infus Berbasis *Mikrokontroller Wemos D1 R2*. Universitas Pasundan, Bandung.

N. A. Ayu Kusuma, E. Yuniarti, dan A. Aziz, "Rancang Bangun *Smarthome* Menggunakan *Wemos D1 R2 Arduino Compatible* Berbasis *ESP8266 ESP-12F*," Program Studi Fisika. Fakultas Sains Dan Teknologi. Universitas Islam Negeri Syarif Hidayatullah Jakarta, vol. 1, p. 8, Apr. 2018.

Rachmat, Hendi H., Asril, dan Hariandi, 2014. Implementasi *Counter Production Monitoring* pada Mesin Tekstil Berbasis *Mikrokontroller*. Teknik Elektro, Institut Teknologi Nasional Bandung, Bandung.

Supegina, Fina, Setiawan, dan Eka Jovi. 2017. Rancang Bangun *IOT Temperature Controller* untuk *Enclosure* BTS Berbasis *Microcontroller Wemos* dan *Android*. Universitas Mercu Buana, Jakarta.

ISBN: 978–623–91277–6–3

HALAMAN INI SENGAJA DIKOSONGKAN