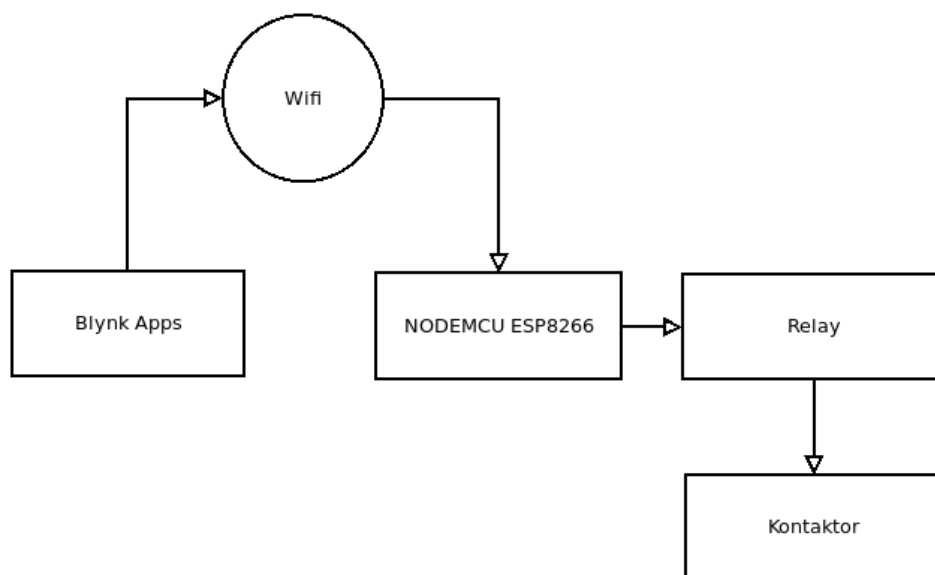


CONTACTOR WIRELESS CONTROLLING

Revolusi Industri 4.0 merupakan fenomena yang mengkolaborasikan teknologi cyber dan teknologi otomatisasi. Konsep penerapannya berpusat pada konsep otomatisasi yang dilakukan oleh teknologi tanpa memerlukan tenaga kerja manusia dalam proses pengaplikasiannya. Dalam industri berbagai alat dikendalikan dengan panel-panel kontrol yang mana secara umum terdiri atas kontaktor, push button, lampu indikator yang memerlukan tenaga manusia untuk mengoperasikan alat tersebut. Proses otomatisasi pada industri kebanyakan menggunakan plc sebagai kontrol, tetapi PLC memiliki biaya investasi yang tinggi. Selain itu memerlukan kabel untuk mengontrol output.

Dengan perkembangan teknologi kita dapat membuat sistem kendali jarak jauh yang dapat mengontrol panel-panel kontrol dengan melalui internet. Kendali jarak jauh ini dimaksudkan untuk memudahkan pengoperasian suatu alat dalam industri yang mana terdapat pada daerah-daerah yang cukup berbahaya bagi manusia seperti dapat terpapar radiasi nuklir, area suhu tinggi, maupun resiko tinggi seperti dapat meledak. Selain itu biaya investasi yang terbilang lebih terjangkau dibandingkan dengan PLC.

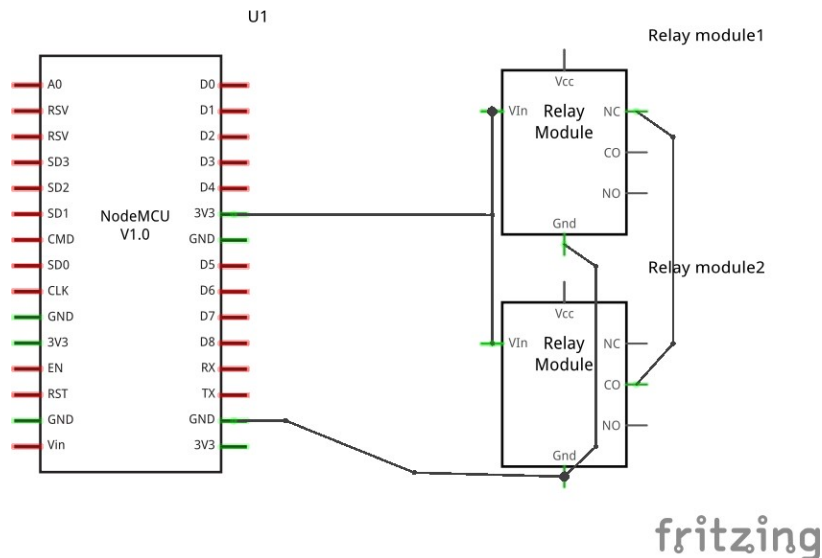
Contactor Wireless Controlling (CWC) menggunakan Nodemcu sebagai microcontroller yang dapat dioperasikan secara wireless menggunakan koneksi wifi dan dapat dikendalikan secara jarak jauh melalui smartphone. Dengan output yang mengendalikan relay sebagai pengganti push button pada panel kontrol dan terhubung dengan kontaktor. Konsep CWC dapat dilihat pada diagram block berikut:



Komponen yang digunakan pada CWC antara lain:

1. Nodemcu ESP8266
2. Modul Relay (Min 2 Channel)
3. Blynk App
4. Contactor Fuji SC-05

Berikut skematik Contactor Wireless Controlling (CWC)



Pemilihan Nodemcu sebagai microcontroller karena memiliki fitur wireless yang dapat dikoneksikan dengan internet menggunakan jaringan Wifi. Pemrogram menggunakan Arduino Ide dengan library Blynk dan aplikasi blynk pada smartphone. Berikut Sourcecode yang digunakan:

```
/******
```

Download latest Blynk library here:

<https://github.com/blynkkk/blynk-library/releases/latest>

Blynk is a platform with iOS and Android apps to control

Arduino, Raspberry Pi and the likes over the Internet.

You can easily build graphic interfaces for all your projects by simply dragging and dropping widgets.

Downloads, docs, tutorials: <http://www.blynk.cc>

Sketch generator: <http://examples.blynk.cc>

Blynk community: <http://community.blynk.cc>

Follow us: <http://www.fb.com/blynkapp>

http://twitter.com/blynk_app

Blynk library is licensed under MIT license

This example code is in public domain.

This example runs directly on NodeMCU.

Note: This requires ESP8266 support package:

<https://github.com/esp8266/Arduino>

Please be sure to select the right NodeMCU module
in the Tools -> Board menu!

For advanced settings please follow ESP examples :

- ESP8266_Standalone_Manual_IP.ino
- ESP8266_Standalone_SmartConfig.ino
- ESP8266_Standalone_SSL.ino

Change WiFi ssid, pass, and Blynk auth token to run :)

Feel free to apply it to any other example. It's simple!

*****/

```
/* Comment this out to disable prints and save space */
```

```
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>
```

```
#include <BlynkSimpleEsp8266.h>
```

```
// You should get Auth Token in the Blynk App.
```

```
// Go to the Project Settings (nut icon).
```

```
char auth[] = "*****";
```

```
// Your WiFi credentials.
```

```
// Set password to "" for open networks.
```

```
char ssid[] = "*****";
```

```
char pass[] = "*****";
```

```
void setup()
```

```
{
```

```
// Debug console
```

```
Serial.begin(9600);
```

```
Blynk.begin(auth, ssid, pass);
```

```
// You can also specify server:
```

```
//Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
```

```
//Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
```

```
}
```

```
void loop()
```

```
{
```

```
Blynk.run();  
}
```

Blynk app pada android digunakan sebagai pengendali jarak jauh, blynk app memiliki fitur yang memungkinkan kita untuk melakukan konfigurasi sesuai dengan peralatan yang kita miliki. Berikut tampilan dan konfigurasi pada blynk app :

