

BAB II

TINJAUAN PUSTAKA

1.1 Studi Literatur

Penelitian tentang keamanan Kendaraan sudah pernah dilakukan oleh beberapa peneliti. Ringkasan *Studi Literatur* yang dilakukan untuk mengetahui sejauh mana penelitian yang sudah ada dapat dilihat sebagai berikut:

1. (Saputro, 2016) dengan judul Rancang Bangun Pengaman Pintu Otomatis Menggunakan E-KTP Berbasis Mikrokontroler Atmega328 sistem kerja dari alat ini yaitu RFID reader yang digunakan memiliki frekuensi 13,56 MHz yang diletakkan dalam box dengan tebal 2mm dapat membaca ID E-KTP dengan jarak maksimal 1,8 cm. Solenoid dapat membuka pengunci pintu apabila ID E-KTP sesuai dengan memori mikrokontroler ATmega328, solenoid akan mengunci kembali dalam waktu 10 detik Alat pengaman pintu otomatis menggunakan E-KTP ini mampu membaca ID E-KTP dengan jarak maksimal 1.8cm dengan sensor RFID *reader* MFRC522 yang memiliki frekuensi 13,56 MHz.
2. (Kurnia Dwi Artika, 2017) dengan judul Rancang Bangun Sistem Pengaman Pada Sepeda Motor Dengan Memanfaatkan Sensor Encoder Dan Sensor Ping Rancang Bangun Sistem Pengaman Pada Sepeda Motor Denganmemanfaatkan Sensor Encoder Dan Sensor Ping. Sensor encoder (jarak) dapat bekerja untuk membunyikan alarm pada saat jumlah ketukan 35 titik (lubang cakram) atau sejauh 4,92 meter. Sensor ping (tinggi) dapat bekerja untuk membunyikan alarm pada *counter* data 1040 atau ketinggian 0,4 meter dari permukaan/jalan.
3. (Nurdin Bagenda & Indra, 2017) dengan judul Prototipe sistem keamanan dan pengendalian Kendaraan Roda Empat berbasis mikrokontroler atmega8535. Sistem kerja dari alat ini yaitu Dengan penggunaan modem Wavecom fastrack M1306B berfungsi mengirim informasi sepeda motor melalui SMS kepada pemilik kendaraan sehingga pemilik dapat dengan mudah mendapat informasi. Selain itu system kemanan ini menggunakan sensor ultrasonic sebagai

pendeteksi stang kendaraan, sensor getar untuk mendeteksi getaran kendaraan. Mikrokontroler atmega8535 sebagai pusat pengontrol atau pengedali dari sensor- sensor yang digunakan. Code vision AVR sebagai perangkat lunak media penghubung antara program yang akan diisikan ke mikrokontroler atmega8535 dengan menggunakan bahasa pemrograman C.

4. (Rizky Muhammad Syafii, 2017) dengan judul Desain Dan Implementasi Sistem Keamanan Locker Menggunakan E-Ktp Berbasis Arduino Pro Mini hasil pengujian dari alat ini yaitu sistem ini mampu merekam lebih dari 2 pengguna locker yaitu sebanyak 15 pengguna locker dengan jarak baca efektif untuk tag jenis key chain dengan tingkat keberhasilan 100% jarak bacanya = 2,5 cm dan e-KTP jarak baca efektif dengan tingkat keberhasilan 100% berjarak = 3,5 cm, untuk jenis tag RFID white card dengan tingkat keberhasilan 100% jarak baca = 5 cm, tag white card merupakan jarak baca paling terjauh diantara e-KTP dan key chain sesuai dengan data sheet reader RFID RC522.
5. (Ariesta Adhitama Satya Negara, Ufi Najib, Jenny Putri Hapsari, 2016) dengan judul Pemanfaatan E-Ktp Untuk Pengaktifan Sepeda Motor Berbasis Arduino Uno sistem kerja dari alat ini yaitu dengan memanfaatkan E-KTP untuk pengaktifan sepeda motor ini menggunakan Arduino UNO sebagai sistem kendali dan RFID untuk alat scanning kartu. Alat ini akan bekerja untuk menggantikan kunci konvensional yang digunakan sepeda motor selama ini. Pertama, alat akan dihidup dengan mengambil sumber tegangan dari akumulator sehingga lampu indikator menyala. Kedua, E-KTP yang sudah terkonfigurasi discan dan akan muncul suara dari buzzer yang mengindikasikan bahwa motor siap untuk diaktifkan. EKTP dapat discan dengan baik bila terdapat pada jarak kurang lebih 10 cm.
6. (Abdi Darmawan dan Nurfiana, Desember 2010) dengan judul Sistem Pengamanan Ganda Pada Kendaraan Bermotor Menggunakan Kartu Secara Elektronik Berbasis Mikrokontroler. Penelitian ini akan bagaimana untuk menghasilkan keamanan kendaraan bermotor roda dua sangat efektif dan jauh lebih murah jika diproduksi. Peralatan akan dibuat dapat ditampilkan oleh LCD (Liquid Crystal Display), yang menarik dan mudah untuk memodifikasi kartu sebagai password. Cara kerjanya roda dua sistem keamanan kendaraan

adalah dengan memasukkan kartu sebagai masukan untuk mematikan sinyal alarm yang diterima dari sinyal input melalui sensor gerak (switch). Jika roda dua stang posisi berubah dari posisi awal, dan stang berubah dengan kekerasan maka akan memberikan masukan ke mikrokontroler untuk menginformasikan sensor inframerah (kartu input), jika kartu yang salah maka mikrokontroler akan mengaktifkan tanduk (LED).

7. (Novi Herawadi Sudiby, 2019) Sistem Pengaman Kendaraan Roda Empat Menggunakan E-Ktp Berbasis Mikrokontroler . Peneliti menggunakan raider RFID sebagai inputan dalam membaca nomer ID pada E-KTP yang akan diproses oleh arduino nano sehingga akan menghasilkan *output* menyalakan kelitrikan, stater dan mematikan kendaraan. Jika E-KTP yang ditemplkan salah sebanyak 3 kali maka GSM Shield akan mengirimkan SMS (*Short Massage Service*) kepada pemilik mobil. Dari hasil uji coba dapat diketahui yaitu jika hanya melakukan 1 kali scan maka relay 1 akan HIGH untuk menyalakan kelistrikan sedangkan pada ujicoba ke 2 melakukan 2 kali *scan* yang berarti pemilik akan menyalakan kelistrikan dan stater kendaraan yaitu *relay* 1 dan 2 akan *HIGH* dan pada ujicoba ke 3 melakukan 3 kali scan ang artinya kendaraan dimatikan. Serta jika kesalahan dalam melakukan scan sebanyak 3 kali maka GSM *Shield* akan mengirimkan SMS kepada pemilik kendaraan dan sistem akan diblokir selama 5 menit mengirimkan SMS apabila dalam melakukan scan E-KTP salah sebanyak 3kali.
8. (Bayu Nugroho, 2011) dengan judul Aplikasi Sistem Pendeteksi Kadar Gas Buang Kendaraan Bermotor sistem kendali diterapkan untuk *Tingkat deteksi* aplikasi dari sistem pembuangan kendaraan bermotor yang dibuat digunakan untuk mendeteksi jumlah kadar gas NO dan gas CO dengan menggunakan sensor TGS2201. Data dari sensor diolah oleh mikrokontroler dan hasilnya ditampilkan pada PC melalui port serial, desain program aplikasi dengan menggunakan bahasa pemrograman Delphi.
9. (Melia Gripin Setyawati, Abdi Darmawan, 2019) Rancang Bangun Kunci Loker Masjid Alat yang digunakan adalah Arduino Uno, Sensor RFID, Sensor *IR Obstacle*, *Solenoid Door Lock*, *Relay* dan *Buzzer*. Metode yang digunakan Peneliti adalah *Observasi*, *Studi Literatur*, Perancangan dan Rancang Bangun.

Hasilnya RFID dan Sensor *IR Obstacle* sebagai *input* membuka dan menutup loker dan *Solenoid* sebagai output pengunci pintu loker dan semua itu dikontrol oleh pengendali Arduino. Hasil ujicoba sistem Ketika kartu RFID yang valid didekatkan ke sensor dengan jarak minimal 4 cm maka akan terbaca oleh sensor RFID dan *buzzer* berbunyi satu kali pendek sehingga pintu loker bisa terbuka. Ketika kartu didekatkan dengan jarak lebih dari 4 cm dari sensor maka kartu tidak akan terbaca dan tidak ada bunyi *buzzer*. Kartu RFID lain tidak akan terbaca oleh sensor RFID karena IDnya tidak dimasukkan di kode program ditandai dengan adanya bunyi *buzzer* panjang selama 30 detik

2.2 Dasar Teori

2.2.1 Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energy untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu setentitas yang berinteraksi, dimana suatu model matematika seringkali biasa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara yang berperan sebagai penggerak yaitu rakyat yang berada di negara tersebut. Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka (Sidarta, 2016).

2.2.2 Pengertian Keamanan

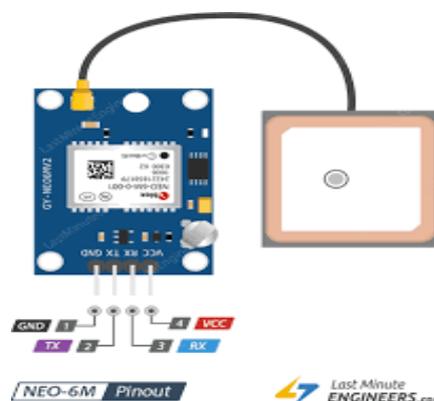
Keamanan adalah keadaan bebas dari bahaya. Istilah ini bisa digunakan dengan hubungan kepada kejahatan, segala bentuk kecelakaan, dan lain-lain. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap serangan teroris, keamanan komputer terhadap hacker atau *cracker*, keamanan rumah

terhadap maling dan penyusup lainnya, keamanan finansial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya. Kebutuhan dasar manusia prioritas kedua berdasarkan kebutuhan fisiologis dalam hirarki *Maslow* yang harus terpenuhi selama hidupnya, sebab dengan terpenuhinya rasa aman setiap individu dapat berkarya dengan optimal dalam hidupnya. Mencari lingkungan yang betul-betul aman memang sulit, maka konsekuensinya promosi keamanan berupa kesadaran dan penjagaan adalah hal yang penting. Keamanan fisik (*Biologic safety*) merupakan keadaan fisik yang aman terbebas dari ancaman kecelakaan dan cedera (*injury*) baik secara mekanis, thermis, elektris maupun bakteriologis. Kebutuhan keamanan fisik merupakan kebutuhan untuk melindungi diri dari bahaya yang mengancam kesehatan fisik, yang pada pembahasan ini akan difokuskan pada *providing for safety* atau memberikan lingkungan yang aman . (Sutris, 2017).

2.3 Perangkat Keras Yang Digunakan

2.3.1 Module GPS UBlox Neo 6M

Modul GPS yang digunakan adalah berjenis uBlox Neo 6M, jenis GPS ini cukup dapat diandalkan karena memiliki keakuratan yang cukup baik dan juga beberapa fitur yang cukup menguntungkan di antaranya terdapat baterai cadangan data, built-in elektronik kompas, dan built-in antena keramik untuk menangkap sinyal dengan kuat. Kemudian untuk dapat mengkomunikasikan GPS ini dengan Arduino diperlukan sebuah library yang bernama “TinyGPS++.h”. Bentuk dari modul GPS uBlox Neo 6M dapat dilihat pada Gambar 2.4 di bawah ini.



Gambar 2.1 Gambar GPS uBlox Neo 6M

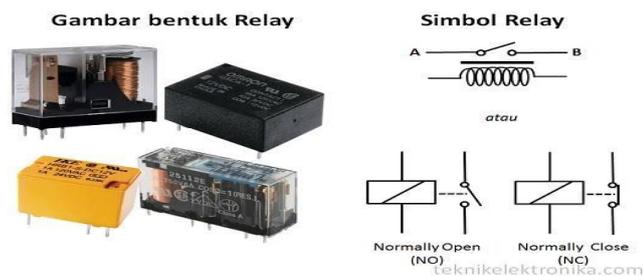
(Sumber: Arranda Ferdian D. 2017)

2.3.2 Relay

Relay merupakan bentuk hambatan terdiri atas titik-titik kontak bawah dengan gulungan *spool*-nya tidak bergerak dan titik kontak bagian atas yang bergerak. Prinsip kerja hambatan adalah menghubungkan titik-titik kontak bagian bawah dengan titik bagian atas yaitu terletak gulungan *spool* dialiri arus listrik yang timbul elektromagnet. (Handy Wicaksono,1996,1-12). *Relay* merupakan bentuk hambatan terdiri atas titik-titik kontak bawah dengan gulungan *spool*-nya tidak bergerak dan titik kontak bagian atas yang bergerak. Prinsip kerja hambatan adalah menghubungkan titik-titik kontak bagian bawah dengan titik bagian atas yaitu terletak gulungan *spool* dialiri arus listrik yang timbul elektromagnet. (Handy Wicaksono,1996,1-12). Secara sederhana *relay* elektromekanis ini didefinisikan sebagai berikut :

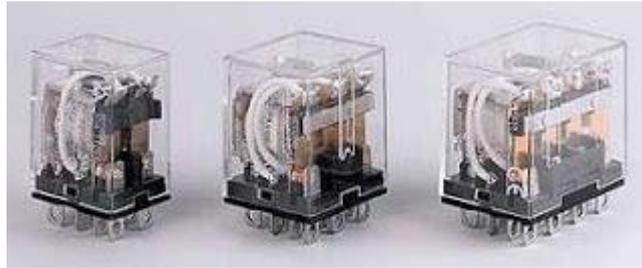
1. Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
2. Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Dibawah ini adalah gambar fisik, bentuk dan Simbol Relay yang sering ditemukan di Rangkaian Elektronika.



Gambar 2.2 Gambar dan Simbol Relay

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)



Gambar 2.3 Relay

(Sumber : Kilian, Christopher T, Modern Control Technology, (West Published Co : 1996)

Bagian titik kontak dibagi menjadi 2 bagian yaitu bagian kontak utama dan kontak bantu yaitu : Bagian kontak utama gunanya untuk menghubungkan dan memutuskan arus listrik bagian yang menuju beban/pemakai. Bagian kontak bantu gunanya untuk menghubungkan dan memutuskan arus listrik ke bagian yang menuju bagian pengendali. Kontak Bantu mempunyai 2 kontak yaitu kontak hubung (NC) dan kontak putus (NO) menandakan masing-masing kontak dan gulungan *spool*. Secara umum, *relay* digunakan untuk memenuhi fungsi-fungsi berikut :

1. *Remote control* : dapat menyalakan atau mematikan alat dari jarak jauh.
2. Penguatan daya : menguatkan arus atau tegangan.
3. Pengatur logika kontrol suatu sistem. Susunan kontak pada *relay* adalah:
4. *Normally Open* : *Relay* akan menutup bila dialiri arus listrik.
5. *Normally Close* : *Relay* akan membuka bila dialiri arus listrik.
6. *Changeover* : *Relay* ini memiliki kontak tengah yang akan melepaskan diri dan membuat kontak lainnya berhubungan.

2.3.2.1 Prinsip Kerja Relay

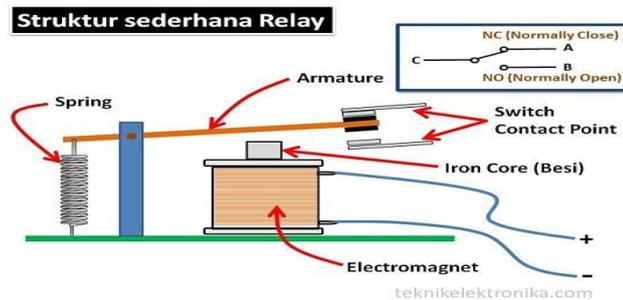
Pada dasarnya, *Relay* terdiri dari 4 komponen dasar yaitu :

1. *Electromagnet (Coil)*
2. *Armature*
3. *Switch Contact Point (Saklar)*
4. *Spring*

Seperti saklar, *relay* juga dibedakan berdasar *pole* dan *throw* yang dimilikinya.

1. *Pole* : banyaknya *contact* yang dimiliki oleh *relay*

2. Throw : banyaknya kondisi (*state*) yang mungkin dimiliki contact.
Berikut ini merupakan gambar dari bagian-bagian *Relay* :



Gambar 2.4 Struktur Sederhana Relay

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

Kontak normally open akan membuka ketika tidak ada arus mengalir pada kumparan, tetapi tertutup secepatnya setelah kumparan menghantarkan arus atau diberi tenaga. Kontak *normally close* akan tertutup apabila kumparan tidak diberi tenaga dan membuka ketika kumparan diberi daya. Masing-masing kontak biasanya digambarkan sebagai kontak yang tampak dengan kumparan tidak diberi tenaga atau daya .

Relay terdiri dari 2 terminal trigger, 1 terminal *input* dan 1 terminal *output*.

1. Terminal trigger : yaitu terminal yang akan mengaktifkan *relay*, seperti alat elektronik lainnya *relay* akan aktif apabila di aliri arus + dan arus -. Pada contoh *relay* yang kita gunakan terminal trigger ini adalah 85 dan 86.
2. Terminal *input* : yaitu terminal tempat kita memberikan masukan, pada contoh adalah terminal 30.
3. Terminal *output* : yaitu tempat keluarnya output pada contoh adalah terminal 87.

2.3.2.2 Jenis-jenis Relay

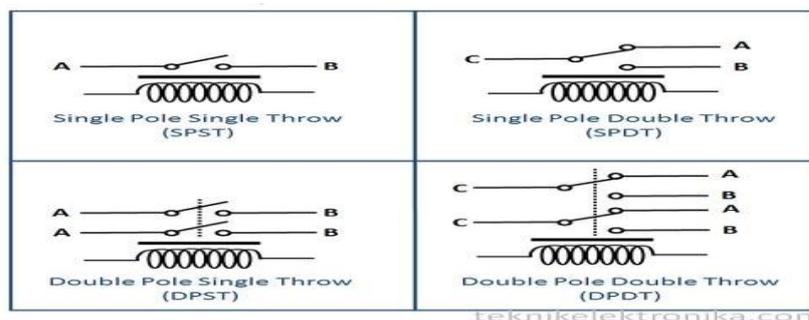
Berikut ini penggolongan relay berdasar jumlah pole dan throw :

1. DPST (*Double Pole Single Throw*), *relay* golongan ini memiliki 6 terminal, diantaranya 4 terminal yang terdiri dari 2 pasang terminal saklar sedangkan
2. terminal lainnya untuk *coil*. Relay DPST dapat dijadikan 2 saklar yang dikendalikan oleh 1 *coil*.

- A. *SPST (Single Pole Single Throw)*, relay golongan ini memiliki 4 terminal, 2 terminal untuk saklar dan 2 terminalnya lagi untuk *coil*.
- B. *SPDT (Single Pole Double Throw)*, relay golongan ini memiliki 5 terminal, 3 terminal untuk saklar dan 2 terminalnya lagi untuk *coil*.
- C. *DPDT (Double Pole Double Throw)*, relay golongan ini memiliki terminal sebanyak 8 terminal, diantaranya terminal yang merupakan 2 pasang *relay SPDT* yang dikendalikan oleh 1 (*single*) *coil*. Sedangkan 2 terminal lainnya untuk *coil*.

Selain golongan *relay* diatas, terdapat juga *relay-relay* yang *Pole* dan *Throw*-nya melebihi dari dua. Misal-nya *3PDT (Triple Pole Double Throw)* ataupun *4PDT (Four Pole Double Throw)* dan lain sebagainya.

Berikut ini merupakan gambar dari jenis *Relay* berdasarkan *Pole* dan *Throw*-nya :



Gambar 2.5 Jenis Relay berdasarkan Pole dan Throw

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

2.3.2.3 Fungsi-Fungsi Relay

Beberapa fungsi *relay* yang telah umum diaplikasikan kedalam peralatan elektronika diantara-nya adalah :

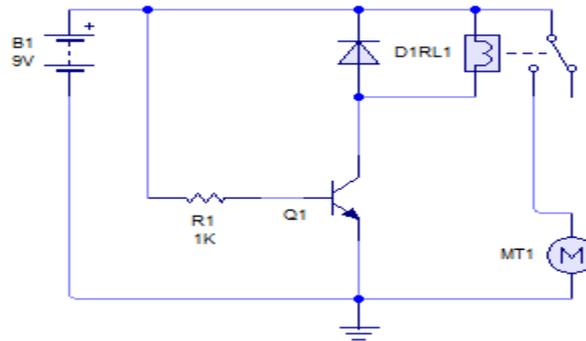
1. *Relay* digunakan untuk menjalankan fungsi logika (*logic function*).
2. *Relay* digunakan untuk memberikan fungsi penundaan waktu (*time delay function*).
3. *Relay* digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari signal tegangan rendah.
4. Ada juga *relay* yang berfungsi untuk melindungi motor ataupun komponen lainnya dari kelebihan tegangan ataupun hubung singkat (*short*).

2.3.2.4 Driver Relay

Rangkaian *driver relay* berfungsi untuk mengendalikan motor arus searah (dc) yang dihasilkan dari *port* paralel I/O. Sinyal dari keluaran port biasanya berupa sinyal-sinyal yang kecil, sehingga tidak mampu untuk menggerakkan sistem daya berupa motor arus searah. Untuk dapat dimanfaatkan sinyal keluaran port, diperlukan suatu rangkaian *driver relay* agar sinyal yang kecil dapat dipergunakan untuk penggerak objek yang akan dikendalikan dari jarak jauh. Rangkaian *driver relay* ini dibangun oleh suatu komponen utama yaitu transistor dan *relay*. Transistor di rangkain *driver relay* difungsikan sebagai penguat sinyal dan *switching*, serta *relay* sebagai penggerak motor dc. *Driver relay* ini selain sebagai sebagai penguat dan *switching*, sekaligus difungsikan untuk mengendalikan motor dc dalam sistem pembalik putaran. Jadi, *driver relay* ini dapat mengatur arah putaran motor *forward* dan *reverse*. Semua *driver relay* pada sistem ini memiliki rangkaian dan karakteristik yang sama. Saat *relay* 1 bekerja maka posisi positif motor akan mendapat sumber tegangan positif dan posisi negatif motor terhubung dengan kutub negatif sumber tegangan. Sehingga, motor akan berputar dengan arah putaran searah jarum jam (*clockwise*). Dengan cara yang sama untuk menggerakkan kontak *relay* 2, maka terjadi kondisi yang berkebalikan yaitu motor akan berputar dengan arah putaran yang berlawanan arah jarum jam (*counter clockwise*).

Penggunaan *driver relay* ini menjadi pilihan karena *driver relay* mudah dikontrol, dapat diberi beban yang besar baik beban AC maupun DC serta sebagai isolator yang baik antara rangkaian beban dengan rangkaian kendali. Rangkaian *driver relay* dapat dibangun menggunakan konsep transistor sebagai saklar. Teknik antara *relay* dengan rangkaian digital atau mikrocontroller adalah rangkaian *driver relay* dengan menggunakan transistor sebagai penguat.

Berikut merupakan contoh dari gambar rangkaian *Driver Relay* :



Gambar 2.6 Rangkaian Driver Relay

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

Pada rangkaian menyerupai sirkuit diatas, dapat dilihat untuk mengoperasikan transistor sebagai saklar transistor dalam keadaan sepenuhnya "OFF" (*cut-off*) atau dalam keadaan "ON" (saturasi).. Namun, ketika dinyalakan dalam kondisi ON (saturasi) , maka aliran arus maksimum. Dalam prakteknya ketika transistor diaktifkan "OFF", arus kebocoran akan kecil ketika mengalir melalui transistor dan ketika diaktifkan "ON" maka rangkaian tersebut akan memiliki tegangan saturasi kecil (V_{CE}) Meskipun transistor tidak dalam saklar yang sempurna, baik di *cut-off* dan daerah saturasi. Agar arus Basis mengalir, terminal *input* Basis harus dibuat lebih positif daripada *Emitter* dengan meningkatkan itu di atas 0,7 volt yang dibutuhkan untuk perangkat silikon. Dengan memvariasikan *Base-Emitter* ini tegangan V_{BE} arus basis juga mengontrol jumlah arus kolektor yang mengalir melalui transistor.

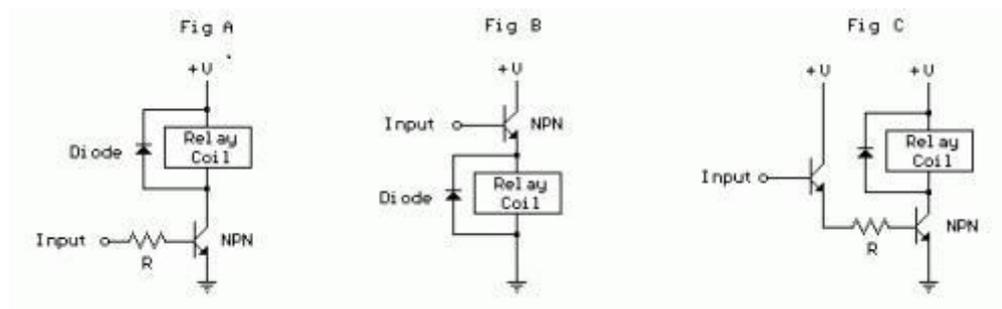
Ketika arus kolektor maksimum mengalir maka transistor dikatakan saturasi. Nilai dari resistor Basis menentukan berapa banyak masukan tegangan yang diperlukan dan sesuai saat Basis untuk beralih transistor sepenuhnya "ON".

Transistor BC108 adalah transistor umum NPN *bipolar junction* (BJT) digunakan untuk memperkuat daya rendah atau aplikasi *switching*. Hal ini dirancang untuk arus yang rendah sampai medium, daya yang rendah , tegangan medium, dan dapat beroperasi pada kecepatan yang cukup tinggi.

2.3.2.5 Interface Driver Relay

Penggunaan *relay* sering menjadi pilihan karena *relay* mudah dikontrol, *relay* dapat diberi beban yang besar baik beban AC maupun DC, dan sebagai

isolator yang baik antara rangkaian beban dengan rangkaian kendali. Rangkaian *interface relay* dapat dibangun menggunakan konsep transistor sebagai saklar. Transistor yang digunakan untuk *driver relay* dapat dikonfigurasi dengan *common emitor*, *emitor follower* atau transistor *darlington*. Teknik *interface* antara *relay* dengan rangkaian digital atau rangkaian *microcontroller* dapat dilihat pada gambar dibawah ini :



Gambar 2.7 Rangkaian Interface Driver Relay

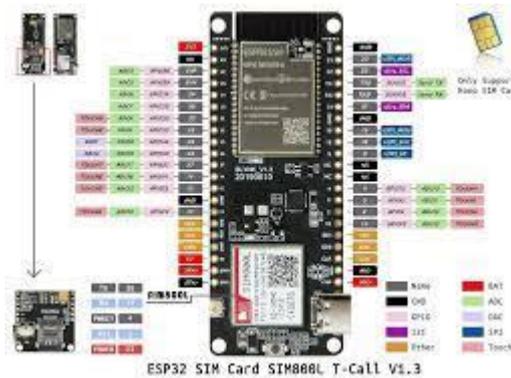
(<http://elektronika-dasar.web.id/teori-elektronika/interface-relay-ke-rangkaian-digital>)

Rangkaian *inteface* antar *relay* dengan rangkaian digital pada gambar diatas ada 3 jenis interface yang dapat digunakan. Bagian dan fungsi komponen dari rangkaian *interface relay* diatas sebagai berikut :

1. Rangkaian pada gambar A, rangkaian pada gambar A tersebut menggunakan mode *common emitor*, apabila basis mendapat sinyal *input* logika 1 (sumber tegangan positif) maka transistor pada gambar A akan mendapat bias maju, sehingga transistor ON dan memberikan sumber tegangan ke *relay* dan *relay* menjadi ON.
2. Rangkaian pada gambar B adalah *interface relay* yang menggunakan transistor teknik *emior follower* dimana *relay* diletakan pada kaki emitor trnasistor. Fungsi dioda yang dipasangkan pada rangkaian *interface* tersebut digunakan untuk menyerap tegangan induksi yang dihasilkan oleh *relay*.
3. Rangkaian pada gambar C merupakan teknik *inteface relay* ke rangkaian digital menggunakan transistor yang dirangkai secara *darlington*.

2.3.3 ESP32 Sim800L

ESP32 Sim800L adalah mikrokontroler sebagai penerus dari mikrokontroler ESP8266 dan ESP32. Pada mikrokontroler ini sudah tersedia modul wifi dan bluetooth sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. Memiliki 18 ADC (Analog Digital Converter), 2 DAC, 16 PWM, 10 Sensor sentuh, 2 jalur antarmuka UART, pin antarmuka I2C, I2S, dan SPI.



Gambar 2.8 ESP32 Sim800L

(Sumber: Arranda Ferdian D. 2017)

2.4 Perangkat Lunak Yang Digunakan

Pengertian perangkat lunak atau biasa disebut *software* adalah sekumpulan data elektronik yang sengaja disimpan dan diatur oleh komputer berupa program ataupun instruksi yang akan menjalankan sebuah perintah. Perangkat lunak atau *software* disebut juga sebagai penerjemah perintah-perintah yang dijalankan oleh user untuk diteruskan dan diproses oleh perangkat keras (*hardware*). Dengan adanya perangkat lunak inilah sebuah sistem mampu menjalankan perintah.

2.4.1 Software Mikrokontroler Arduino Uno

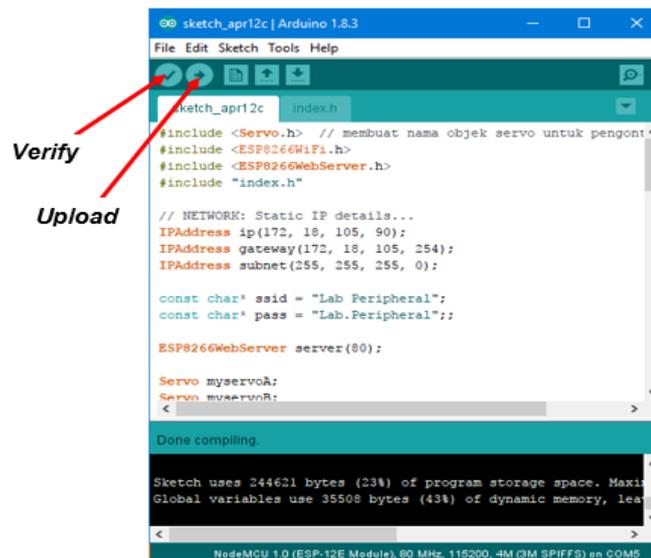
Software arduino yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. *Integrated Development Environment* (IDE), suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau *sketsa* program untuk papan *Arduino*. IDE *arduino* merupakan *software* yang sangat canggih ditulis dengan menggunakan *java*. IDE arduino terdiri dari.

2.4.2 Prangkat Lunak Arduino IDE

IDE merupakan kependekan dari *Integrated Development Environment*. IDE merupakan program yang digunakan untuk membuat program pada Arduino Uno. Program yang ditulis dengan menggunakan *Software* Arduino (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi.ino.

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Software* Arduino IDE, menunjukkan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan (Arranda Ferdian D, 2017).

- Verify/Compile*, berfungsi untuk mengecek apakah *sketch* yang dibuat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang dibuat akan *dikompilasi* kedalam bahasa mesin.
- Upload*, berfungsi mengirimkan program yang sudah dikompilasi ke *Arduino Board*.



Gambar 2.9 Arduino IDE

(Sumber: Arranda Ferdian D. 2017)

2.4.3 *Internet of Things*

Internet of Things, atau dikenal juga dengan singkatan **IOT**, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT. Dan kini IoT menjadi salah satu tugas bagi seorang mahasiswa di sebuah perguruan tinggi.



Gambar 2.10. Ilustrasi dari *Internet Of Things*

(Sumber : <https://www.meccanismocomplexo.org/en/iot-internet-of-things/>,
Diakses
Tanggal 6 Maret 2017)

2.4.4 **Android**

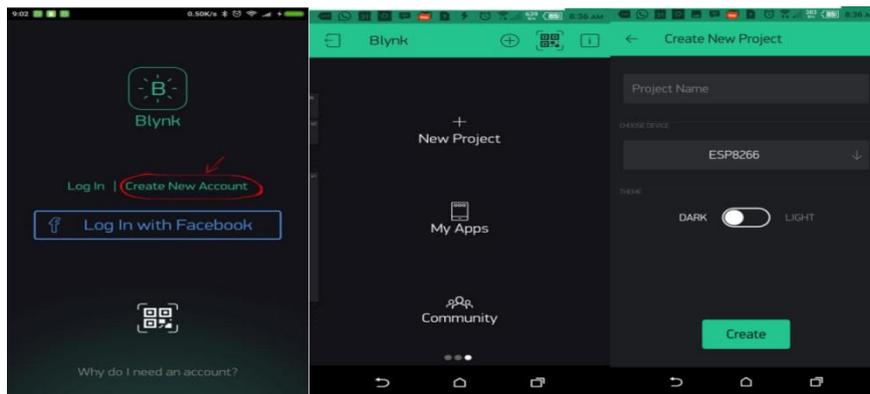
Android adalah sistem operasi untuk perangkat selular yang berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc.pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. Pada saat perilis perdana Android, November 2007,

Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Dilain pihak, Google merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

2.4.5 Aplikasi Blynk

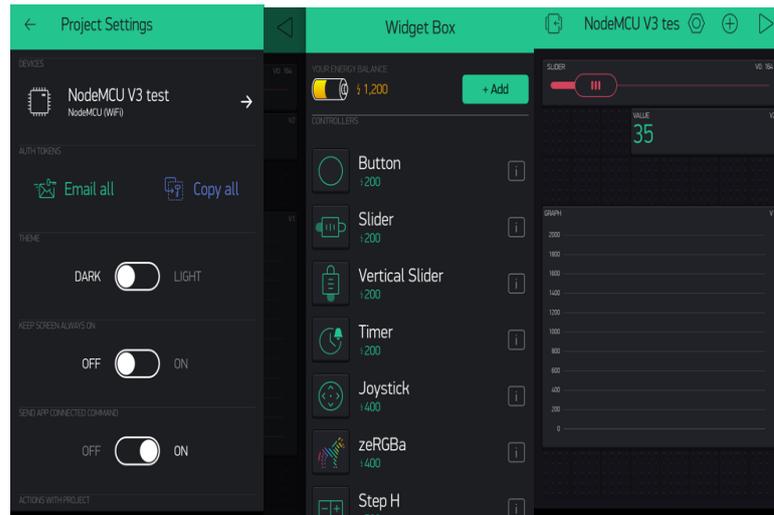
Blynk adalah *aplikasi* untuk IOS dan OS Android untuk mengontrol Arduino, ESP32Sim800L, *Raspberry Pin* dan sejenisnya melalui Internet. Aplikasi ini dapat digunakan untuk mengendalikan perangkat *hardware*, menampilkan data sensor, menyimpan data, visualisasi, dan lain-lain. *Aplikasi Blynk* memiliki 3 komponen utama, yaitu *Aplikasi*, *Server*, dan *Libraries*. *Blynk server* berfungsi untuk menangani semua komunikasi diantara *smartphone* dan *hardware*. *Widget* yang tersedia pada *Blynk* diantaranya adalah *Button*, *Value Display*, *History Graph*, *Twitter*, dan *Email*. *Blynk* tidak terikat dengan beberapa jenis microcontroller namun harus didukung *hardware* yang dipilih. ESP32Sim800L dikontrol dengan Internet melalui WiFi, chip ESP8266, *Blynk* akan dibuat online dan siap untuk *Internet of Things*. Cara pembuatan *user interface* pada *Blynk* sebagai berikut :

Membuka *aplikasi blynk*, pertama membuat akun untuk mendapatkan *auth token* yang dikirim melalui email. Setelah itu membuat project dengan diberi nama “ MONITORING” dan *hardware* yang digunakan, kemudian pilih *create* seperti pada Gambar :



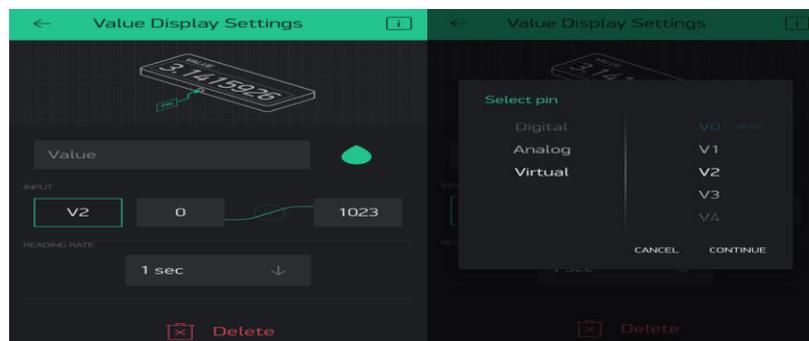
Gambar 2.11 Membuat Akun Dan Project Pada Aplikasi Blynk

Untuk menghubungkan *device IOT* dengan *server blynk* dibutuhkan kode keamanan *Authentication* yang dikirimkan dari *server blynk* ke email melalui *Project Setting* pada menu *auth token*. Menu *Project Setting* terdapat pada icon nomor 3 dari kanan . Menu yang lainnya adalah segitiga digunakan untuk *play aplikasi project* dan menu plus digunakan untuk menambah komponen dalam *project aplikasi blynk*. Kode *auth token* dapat didapatkan melalui pengiriman email ataupun langsung dicopy melalui aplikasi *blynk*. *Auth token* yang dikirimkan melalui email atau langsung copy dari aplikasi nanti akan dimasukkan kode program yang dimasukkan dalam ESP8266 untuk menambah komponen input *output project* dapat menggunakan menu plus yang ada didalam lingkaran. Terdapat bermacam-macam komponen diantaranya *Button* , *Slider* , *Vertical Slider* , *ValueDisplay* dan juga komponen *graphic*. Berbagai macam komponen yang tersedia disesuaikan dengan kredit power yang masih tersisa. kredit power pada saat registrasi diberikan sejumlah 2000. Untuk topup kredit power dapat menggunakan *google play* kredit.



Gambar 2.12 Auth Token dan Widget Pada Aplikasi Blynk

Menambahkan komponen *value display* dengan *caradrag and drop* pada komponen yang tersedia, selanjutnya melakukan konfigurasi komponen *value display pin* menjadi *virtual pin V1*. Komponen ini digunakan untuk menampilkan data yang nanti akan dikirimkan dari *hardware* ke Aplikasi *Blynk*.



Gambar 2.13 Value Display

Menambahkan komponen *Slider Display* dengan cara *drag and drop* pada komponen yang tersedia, selanjutnya melakukan konfigurasi komponen *value display pin* menjadi *Virtual Pin V0*. Komponen *Slider* ini akan digunakan untuk mengirimkan data dari *Aplikasi Blynk ke hardware*. (Sumber: Arranda Ferdian D. 2017)

2.4.6 Flowchart

Menurut Sulindawati (2010:8), “*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program”. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengopersian.

2.4.6.1 Jenis Flowchart

Menurut Sulindawati (2010:8), *Flowchart* terbagi atas lima jenis, yaitu:

1. *Flowchart* Sistem (*system flowchart*)
2. *Flowchart* peperwork (*documen flowchart*)
3. *Flowchart* Skematik (*Schematic Flowchart*)
4. *Flowchart* Program (*Program Flowchart*)
5. *Flowchart* proses (*process flowchart*)

2.4.6.2 Flowchart Sistem (System Flowchart)

Flowchart Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam system secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain, *flowchart* ini merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi yang membentuk sistem. *Flowchart* sistem terdiri dari tiga data yang mengalir melalui sistem dan proses yang mentransformasikan data itu. Data dan proses dalam *flowchart* sistem dapat digambarkan secara *online* (dihubungkan langsung dengan komputer) atau *offline* (tidak dihubungkan langsung dengan komputer, misalnya mesin tik, *cash register* atau kalkulator).

2.4.6.3 Flowchart Paperwork (Document Flowchart)

Flowchart Paperwork menelusuri alur dari data yang ditulis melalui sistem. *Flowchart* Paperwork sering disebut juga dengan *Flowchart* Dokumen. Kegunaan utamanya adalah untuk menelusuri alur form dan laporan sistem dari satu bagian ke bagian lain baik bagaimana alur form dan laporan diproses, dicatat atau disimpan.

2.4.6.4 Flowchart Skematik (*Schematic Flowchart*)

Flowchart Skematik mirip dengan *Flowchart* Sistem yang menggambarkan suatu sistem atau prosedur. *Flowchart* Skematik ini bukan hanya menggunakan simbol-simbol *flowchart* standart, tetapi juga menggunakan gambar-gambar komputer, *peripeheral*, form-form atau peralatan lain yang digunakan dalam sistem. *Flowchart* Skematik digunakan sebagai alat komunikasi antara analis sistem dengan seseorang yang tidak familiar dengan simbol-simbol *flowchart* yang konvensional. Pemakaian gambar sebagai ganti dari simbol-simbol *flowchart* akan menghemat waktu yang dibutuhkan oleh seseorang untuk mempelajari simbol abstrak sebelum dapat mengerti *flowchart*.

2.4.6.5 Flowchart Program (*Program Flowchart*)

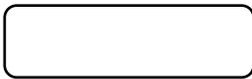
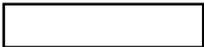
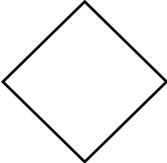
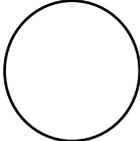
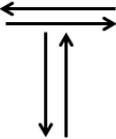
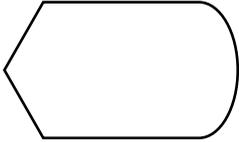
Flowchart Program dihasilkan dari *Flowchart* Sistem. *Flowchart* Program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan. *Flowchart* ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi. *Programmer* menggunakan *Flowchart* Program untuk menggambarkan urutan instruksi dari program komputer. Analisa sistem menggunakan *flowchart* program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi.

2.4.6.6 Flowchart Proses (*Process Flowchart*)

Flowchart Proses merupakan teknik menggambarkan rekayasa industrial yang memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem. *Flowchart* Proses memiliki lima simbol khusus. *Flowchart* Proses digunakan oleh perekayasa industrial dalam mempelajari dan mengembangkan proses-proses manufacturing. Dalam analisis sistem, *Flowchart* ini digunakan secara efektif untuk menelusuri alur suatu laporan.

2.4.6.7 Simbol – Simbol Dalam *Flowchart*

Tabel 2.1 Simbol – Simbol Dalam *Flowchart*

NO	SIMBOL	KETERANGAN
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol Input/Output yang mendefinisikan masukkan dan keluaran proses.
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
5.		Simbol konektor untuk keluar-masuk /menyambung proses dalam lembar yang sama
6.		Simbol konektor untuk keluar-masuk /menyambung proses dalam lembar yang berbeda.
7.		Simbol untuk menghubungkan antara simbol yang satu dengan yang simbol yang lain.
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual.

