TUGAS AKHIR

PROTOTYPE SISTEM PARKIR OTOMATIS BERBASIS RFID MENGGUNAKAN HARDWARE

Oleh :

Fernando Ratuela

11 023 018

Dosen Pembimbing :

Herry Makapedua,ST,MT

NIP. 195611131993031001

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI MANADO

JURUSAN TEKNIK ELEKTRO

2016

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Praktis, cepat, dan aman beberapa kata tersebut sangat dibutuhkan oleh penguna kendaraan roda empat yang ada di kota-kota besar seperti di kota manado. Dan dengan semakin berkembangnya jaman semakin banyak pula bangunan-bangunan seperti pusat perbelanjaan, apartemen dan lain sebagainya yang sering di gunakan masyarakat, dan tentunya tak terlepas dari pengguna lahan parkir. Yang saat ini sangat dibutuhkan oleh pengguna kendaraan bermotor khususnya kendaraan beroda empat, di kota-kota besar saat ini ketersediaan tempat untuk membangun gedung dengan lahan parkir yang luas sangatlah sulit, sehingga membuat pengguna kendaraan beroda empat kesulitan mencari lahan parkir, dan kemudian memarkir kendaraan di sembarang tempat yang tidak terjamin keamanannya. Dan juga banyak waktu yang terbuang untuk mencari lahan parkir akibat masih kurang efektifnya tempat parkir yang aman, pokok permasalahan ini yang membuat penulis merasa perlu untuk memnciptakan / membuat solusi lahan parkir yang modern dengan menggunakan system parkir yang aman, System parkir otomatis yang lebih aman, dengan menggunakan Kartu Radio frekuensi Identifikasi (RFID). Adapun keuntungan parikir otomatis ini adalah pemilik diberikan kartu (RFID) yang nantinya akan berfungsi untuk memerintahakan sensor parkir otomatis untuk membawa mobil ke tempat pemarkiran atau ke tempat semula dimana mobil sebelumnya di pindahkan. Hingga saat ini di kota manado belum di temui tempat parkir otomatis dengan dengan system seperti ini.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan di atas maka rumusan masalah dalam penelitian ini adalah, sebagai berikut

- Bagaimana menciptakan software sistem parkir yang bekerja otomatis sesuai algoritma yang di inginkan
- Bagaimana membuat program system parkir otomatis dengan menggunakan RFID yang kemudian dapat terintegrasi dengan Board ARDUINO
- Bagaimana membuat program (coding) menggunakan aplikasi ARDUINO IDE untuk menjalankan sistem parkir otomatis
- Bagaimana mengolah data analog input dari sensor dan kemudian di konversi ke digital untuk menyalakan instrument pada papan lintasan

1.3 Batasan Masalah

Agar tidak menyimpang dari topik yang akan di bahas maka penulis menekankan batasan masalah yang akang di bahas hanya terbatas pada :

- Bagaimana system kerja dari software pada sistem parkir parkit otomatis ini
- Bahasa pemograman yang di gunakan adalah bahasa C++ yang berbasis pada library dan sintaks dari software Arduino IDE
- Bagaimana mengolah data dari reader RFID dan kemudian bisa di gunakan sebagai navigasi pergerakan robot ketika memarkir kendaraan.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitan ini adalah, sebagai berikut :

- Untuk merancang software dari sistem parkir otomatis dengan mengunakan RFID dengan berbasiskan ARDUINO

1.5 Manfaat penelitian

Setelah semua tujuan telah di uraikan di atas maka manfaat yang bisa diperoleh adalah sebagai berikut :

- Program yang di buat dapat di aplikasikan pada system parkir yang menggunakan RFID sebagai identifikasi di area parkir
- Merancang system parkir yang aman bagi penggunna dan kendaraan itu sendiri.
  Dalam artian, membuat software yang dapat menjalankan system mekanikal secara otomatis, sehingga kesalahan – kesalahan yang sering terjadi di area parkir konvensional seperti, tabrakan akibat kelalaian driver, memarkir kendaraan yang tidak teratur dsb, dapat di minimalisir.
-

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Rangkaian Elektronika

Rangkaian elektronika adalah rangkaian yang di bangun dari berbagai komponen-komponen yang menbentuk satu kesatuan yang menghasilkan sebuah keluaran yang dapat mengontrol sesuatu atau membangkitkan sesuatu, rangkaian elektronik bisa sangat kecil dan sederhana. Komponen-komponen yang terletak dalam sebuah rangkaian elektronika sangatlah bervariasi. Beberapa komponen yang biasa kita jumpai dan nantinya akan di pakai dalam pembuatan prototype ini :

2.2 Integrated Cirtcuit

Sirkuit terpadu (*integrated circuit*) atau IC adalah komponen dasar yang terdiri dari resistor, transistor dan lain-lain. IC adalah komponen yang dipakai sebagai otak peralatan elektronika.Pada komputer, IC yang dipakai adalah mikroprosesor. Dalam sebuah mikroprosesor Intel Pentium 4 dengan ferkuensi 1,8 trilyun getaran per detik terdapat 16 juta transistor, belum termasuk komponen lain. Fabrikasi yang dipakai oleh mikroprosesor adalah 60nm.Sirkuit terpadu dimungkinkan oleh teknologi pertengahan abad ke-20 dalam fabrikasi alat semikonduktor dan penemuan eksperimen yang menunjukkan bahwa alat semikonduktor dapat melakukan fungsi yang dilakukan oleh tabung vakum. Pengintegrasian transistor kecil yang banyak jumlahnya ke dalam sebuah chip yang kecil merupakan peningkatan yang sangat besar bagi perakitan tube-vakum sebesar-jari. Ukuran IC yang kecil, tepercaya, kecepatan "switch", konsumsi listrik rendah, produksi massal, dan kemudahan dalam menambahkan jumlahnya dengan cepat menyingkirkan tabung vakum

Hanya setengah abad setelah penemuannya, IC telah digunakan dimana-mana. Radio, televisi, komputer, telepon selular, dan peralatandigital lainnya yang merupakan bagian penting dari masyarakat modern. Contohnya, sistem transportasi, internet, dll tergantung dari keberadaan alat ini. Banyak skolar percaya bahwa revolusi digital yang dibawa oleh sirkuit terpadu merupakan salah satu kejadian penting dalam sejarah umat manusia.

IC mempunyai ukuran seukuran tutup pena sampai ukuran ibu jari dan dapat diisi sampai 250 kali dan digunakan pada alat elektronika seperti:

- Telepon
- Kalkulator
- Ponsel
- Radio

2.3 Pengenalan ATMEL AVR Mikrokontroler

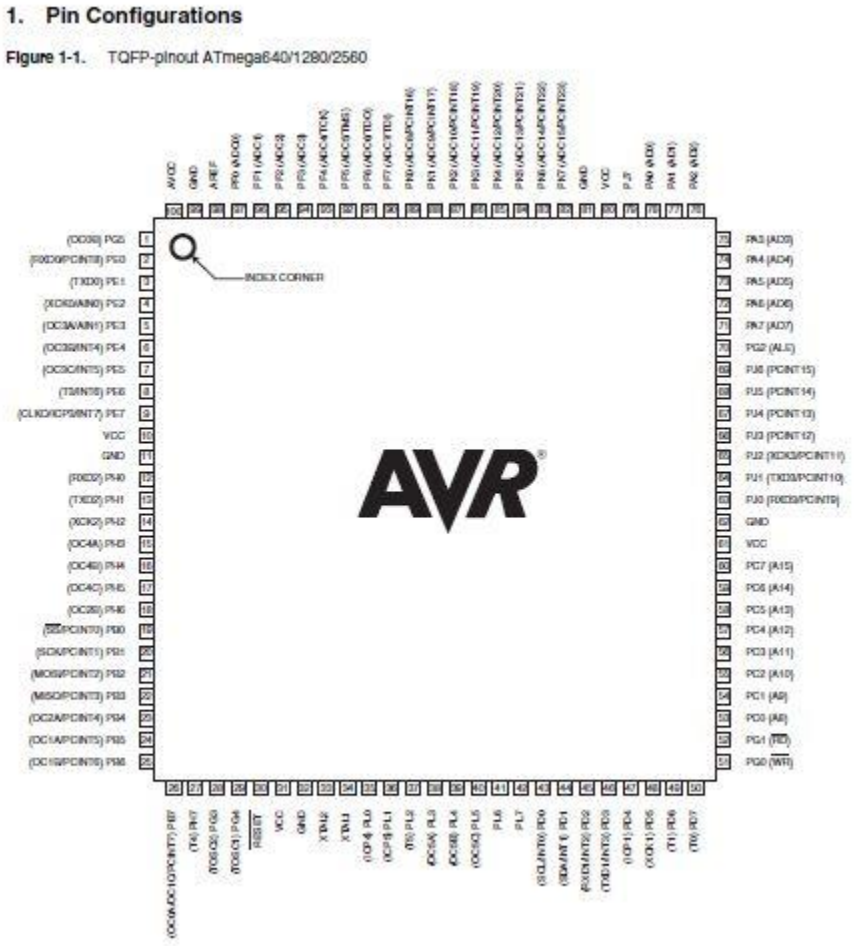
Mikrokontroler adalah sebuah computer kecil (special purpose computers) di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan parallel, Port input / output, dan ADC (Analog Digital Converter). Saat ini penggunaan mikrokontroler dapat di temui pada berbagai peralatan, Misalnya Televisi, telepon digital, mesin cuci, system keamanan rumah dan gedung, dll. Mikrokontroler juga dapat kita gunakan untuk otomasi industry, akuisisi data, telekomunikasi dll. Keuntungan menggunakan mikrokontroler adalah harganya murah dan mudah di dapat, dan dapat di progam secara berulang-ulang sesuai dengan keinginan kita.

Mikrokotroler AVR (Alf and Vegard’s Risc Prosesor) standar memiliki arsitektur 8 bit, dimana semua instruksi di kemas dalam kode 16-bit dan sebagian besar instruksi dip roses dalam satu siklus clock. AVR dapat di kelompokan menjadi empat kelas yaitu :

- Keluarga ATtiny

- Keluarga AT90Sxx
- Keluarga ATMega
- Keluarga AT86RFxx.

Jenis controller yang akan di pakai untuk pembuatan miniatur simulasi parkir otomatis adalah ATMega 2560, berikut adalah penjelasan umum susunan kaki dari ATMega 2560



Gambar 2.1. Arduino Mega 2560

Berikut ini penjelasan umum susunan kaki dari dari ATMega 2560 :

Gambar 2.2. Arduino Mega 2560
Sumber : https//www.google.com

- VCC merupakan pin masukan positive catu daya. Setiap device elekronika digital tentunya butuh sumber catu daya yang umumnya sebesar 5 V, oleh karena itu biasanya di PCB Kit Mikrokontroler, selalu ada IC regulator 7805.

- GND sebagai Pin Ground

- Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai input atau output, menggunakan fungsi pinMode , digitalWrite, dan digitalRead. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial : 0 (RX) dan 1 (TX); **Serial 1** : 19 (RX) dan 18 (TX); **Serial 2** : 17 (RX) dan 16 (TX); **Serial 3** : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.

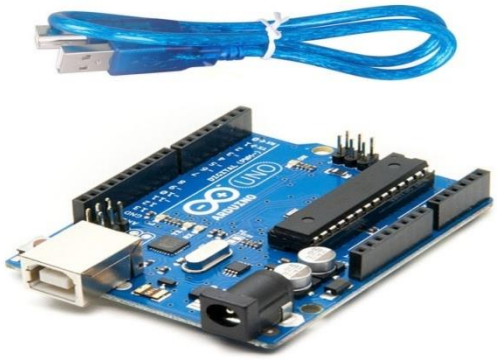
- Eksternal Interupsi : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi

untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.

- **SPI** : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
- **LED** : Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.
- Arduino Mega2560 memiliki 16 pin sebagai analog input, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi analogReference
- **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi analogReference
- **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.
- XTAL1 dan XTAL2 sebagai pin masukan Clock External. Suatu mikrokontroler membutuhkan sumber detak (Clock) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai cristalnya maka, semakin cepat mikrokontroler tersebut
- AVCC sebagai pin masukan tegangan untuk ADC
- AREF sebagai pin masukan tegangan referensi.

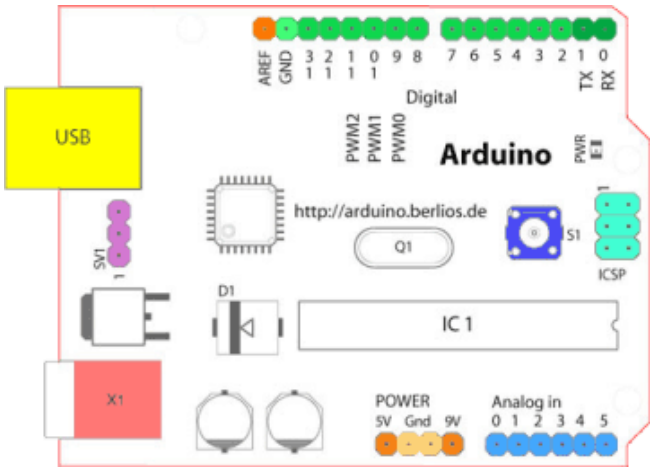
2.3.1 Sekilas tentang Arduino Uno

Arduino Uno adalah salah satu produkberlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keping yang secara fungsional bertindak sebagai sebuah computer). Peranti ini data di gunakan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat di implementasikan dengan menggunakan papan yang berukuran relative kecil ini (lihat gambar 10). Bahkan, dengan komponen tertentu, peranti ini dapat di pakai untuk pemantauan jarak jauh melalui internet, misalnya pemantauan kondisi pasien di rumah sakit dan pengendalian alat" di rumah.



Gambar 2.3. Arduino Mega 2560
Sumber : https//www.google.com

Arduino Uno mengandung mikroprosesor (berupa Atmel AVR) dan di lengkapi dengan oscillator 16 Mhz (yang memungkinkan operasi berbasis waktu dapat di lakukan dengan tepat), dan regulator (pembangkit Tegangan) 5 volt. Sejumlah pin tersedia di papan . Pin 0 sampai Pin 13 di gunakan untuk isyarat digital, yang hanya bernilai 0 atau 1. Pin A0-A5 digunakan untuk isyarat analog.



Gamar 2.4 , Tata letak komponen pada board Arduino

Setiap 13 pin digital pada Arduino Uno dapat digunakan sebagai input dan output, menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Fungsi-fungsi tersebut beroperasi di tegangan 5 Volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor pull-up (terputus secara default) 20-50 kOhm. Selain itu, beberapa pin mempunyai fungsi-fungsi spesial:

1) Serial: 0 (RX) dan 1 (TX). Digunakan untuk penerima (RX) dan pemancar (TX) serial data TTL (Transistor-Transistor Logic). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari chip Serial Atmega8U2 USB-ke-TTL.
2) External Interrupts: 2 dan 3. Pin-pin ini dapat dikonfigurasikan untuk dipicu sebuah interrupt (gangguan) pada sebuah nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai. Lihat fungsi attachInterrupt() untuk lebih jelasnya.
3) PWM: 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM output dengan fungsi analogWrite().
4) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini menghubungkan komunikasi SPI menggunakan SPI library.
5) LED: 13. Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai HIGH LED menyala, ketika pin bernilai LOW LED mati.

Arduino UNO mempunyai 6 input analog, diberi label A0 sampai A5, setiapnya memberikan 10 bit resolusi (contohnya 1024 nilai yang berbeda). Secara default, 6 input analog tersebut mengukur dari ground sampai tegangan 5 Volt, dengan itu mungkin untuk mengganti batas atas dari rangenya dengan menggunakan pin AREF dan fungsi analogReference(). Di sisi lain, beberapa pin mempunyai fungsi spesial:

1) TWI: pin A4 atau SDA dan pin A5 atau SCL. Mensupport komunikasi TWI dengan menggunakan Wire library
2) Ada sepasang pin lainnya pada board:
3) AREF. Referensi tegangan untuk input analog. Digunakan dengan analogReference().
4) Reset. Membawa saluran ini LOW untuk mereset mikrokontroler. Secara khusus, digunakan untuk menambahkan sebuah tombol reset untuk melindungi yang memblock sesuatu pada board.

Bagian strukutur program Arduino ini meliputi kerangka program, sintaks program, kontrol aliran program, dan operator

2.3.2 Kerangka Program

Kerangka program Arduino sangat sederhana, yaitu terdiri atas dua blok. Blok yang pertama adalah **void setup()** yang berisi kode program yang hanya dijalankan sekali sesaat setelah Arduino dihidupkan atau di-*reset*, dan blok kedua
adalah **void loop()** yang berisi kode program yang dijalankan terus menerus yang merupakan tempat untuk program utama.

2.3.3 Sintaks Program

Blok **void setup()**, **void loop()** maupun **function** harus diberi tanda kurung kurawal buka "**{**" sebagai tanda awal program di blok itu dan kurung kurawal tutup "**}**" sebagai tanda akhir program. Tanda kurung kurawal tersebut juga digunakan pada blok kontrol program, seperti **if**, **if-else**, **for-loop**, **while-loop**, dan **do-whileloop**. Untuk menandai akhir sebuah baris kode program digunakan tanda titik koma ";". Kurangnya

tanda kurung kurawal buka dan kurawal tutup ataupun titik koma akan menyebabkan *compile error*. Untuk lebih jelasnya fungsi IDE Arduino ditunjukan oleh tabel 3.2

Tabel 2.1 Fungsi IDE Arduino

No.	Tombol	Nama	Fungsi
1.		*Verify*	Menguji apakah ada kesalahan pada program atau *sketch*. Apabila *sketch* sudah benar,maka akan dikompilasi. Kompilasi adalah proses mengubah kode program dalam kode mesin.
2.		*Upload*	Mengirimkan kode mesin hasil kompilasi ke *board* Arduino.
3.		*New*	Membuat *sketch* atau lembar kerja halaman baru.
4.		*Open*	Berfungsi untuk membuka halaman kerja yang sudah ada.
5.		*Save*	Berfungsi untuk menyimpan halaman kerja *(sketch)*.
6.		*Serial Monitor*	Menampilkan data yang dikirm dan diterima melalui komunikasi serial.

2.3.4 Kontrol Aliran Program

Kontrol aliran program ini meliputi instruksi-instruksi yang digunakan untuk membuat percabangan dan perulangan. Instruksi percabangan diantaranya adalah **if**, **if-else**, **switch-case**, **break**, **continue**, **return**, dan **goto**. Sedangkan instruksi perulangan diantaranya adalah **for-loop**, **while-loop**, **do-while-loop**. Instruksi **if** dan **if-else** akan menguji apakah kondisi tersebut dipenuhi atau tidak. Jika tidak dipenuhi maka instruksi berikutnya akan dilompati, tetapi jika dipenuhi

maka instruksi tersebut akan dijalankan. Instruksi **return** digunakan untuk menghentikan proses sebuah blok **Function** dan kembali ke program utamanya, sambil membawa hasil proses apabila blok **Function** tersebut menghasilkan data. Instruksi **goto** sama seperti

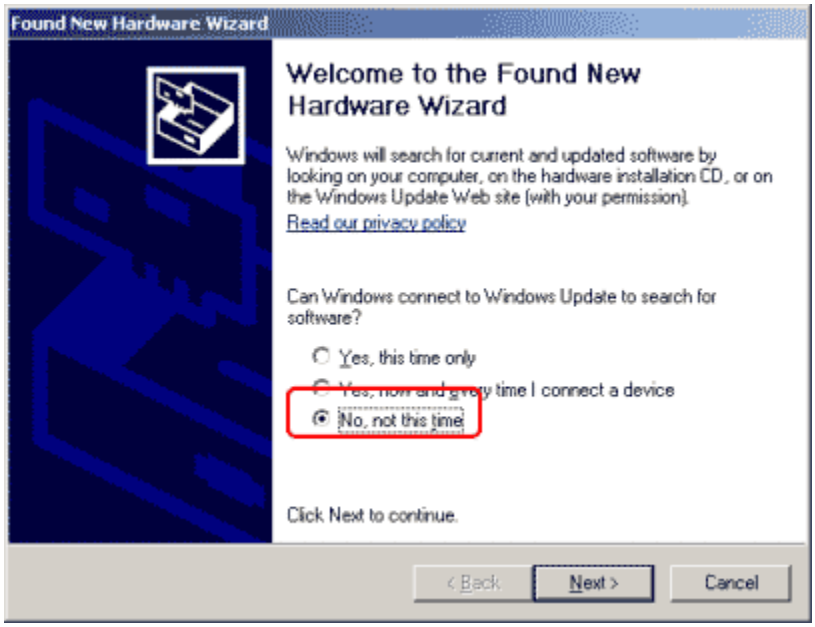
instruksi **break** yaitu digunakan untuk melompat keluar dari perulangan, hanya bedanya lokasi lompatan **goto** bisa diatur dengan cara menempatkan labelnya pada lokasi yang diinginkan.

Bagian fungsi meliputi fungsi *input output* digital dan analog, advanced I/O, fungsi waktu, fungsi matematika (termasuk *random,* instruksi byte dan bit) serta fungsi komunikasi. Untuk kaki *output* analog, sebenarnya Arduino tidak memiliki kaki *output* analog. Namun dengan metode PWM *(Pulse Width Modulation)* maka beberapa kaki I/O digital Arduino dapat digunakan untuk menghasilkan sinyal analog. Berbeda dengan *input output* digital yang harus diatur fungsinya sebagai INPUT atau sebagai OUTPUT dengan instruksi **pinMode()**, maka pada *input output* analog ini tidak diperlukan pengaturan seperti itu, karena kaki input dan outputnya terpisah. Fungsi komunikasi digunakan untuk berkomunikasi dengan komputer melalui *port* serial.

Menginstal arduino IDE

Di bawah ini akan dijelaskan langkah-langkah instalasi driver USB pada Windows XP (*red*: Pada Windows versi baru, sistem operasi akan mengenali, mengunduh dan menginstall driver secara otomatis, cukup sambungkan Arduino dengan komputer lewat kabel USB)

1. Sambungkan papan Arduino dengan sebuah komputer melalui kabel USB.
2. Dengan segera komputer akan mendeteksi kehadiran sebuah perangkat baru yang belum ia kenal dan Windows akan menampilkan sebuah window wizard seperti

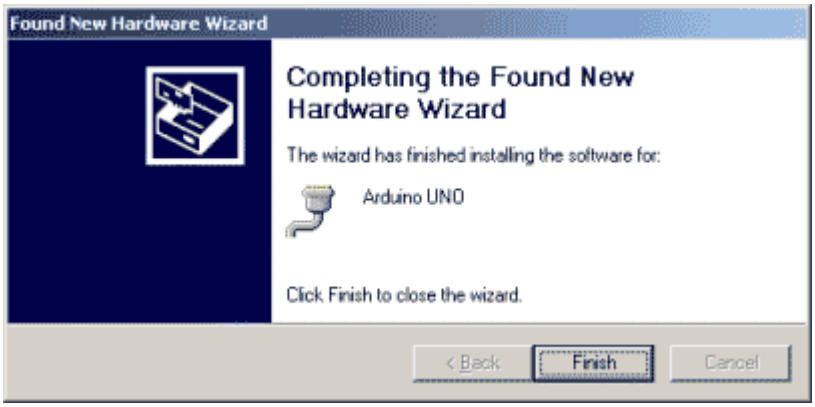


Gambar 2.5 Jendela install software yang baru

berikut ini.Jawab dengan "***No, not this time***" dan tekan ***Next***.

3. Wizard akan mencari software driver untuk perangkat tersebut. Silakan menjawab dengan "***Install from a list or specific location (Advance)***". Lanjutkan dengan ***Next***.
4. Tentukan lokasi dimana software Arduino ditempatkan pada komputer. Silakan sesuaikan lokasinya sesuai dengan hasil ekstrak software Arduino pada komputer Anda. Di dalam lokasi tersebut terdapat sebuah direktori bernama drivers, arahkan wizard untuk mencari driver di dalam direktori tersebut. Klik ***Next*** untuk melanjutkan. Jika muncul sebuah window peringatan bahwa "*Arduino UNO has not passed Windows Logo testing… dst.*", jawab dengan "***Continue Anyway***".
5. Jika driver Arduino selesai diinstal pada komputer maka pada akhir proses akan tampil sebuah pesan berhasil seperti berikut ini. Tekan ***Finish*** untuk menutup wizard.

Drive relah berhasil diinstall.

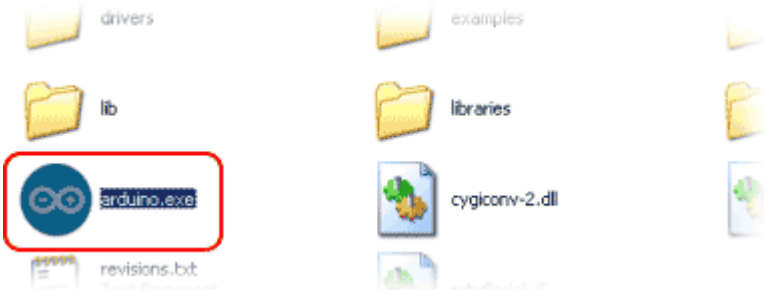


Gambar 2.6 jendela proses instalasi software telah selasai

Menguji koneksi komputer dan papan pada arduino.

Sekalipun sebuah papan Arduino dapat bekerja dengan mendapat asupan daya dari sebuah komputer, namun hal itu tidak berarti ia dapat berkomunikasi dengan komputer tersebut. Untuk memastikan Arduino telah terpasang dengan benar dan dapat berkomuniasi dengan interaktif maka ia perlu diuji.

1. Jalankan IDE Arduino dengan menjalankan sebuah file bernama **arduino.exe** pada lokasi software Arduino.

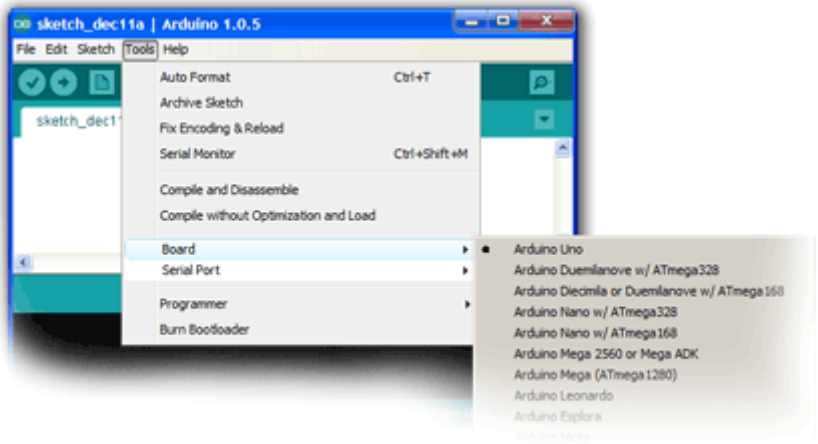


Gambar 2.7 Shortcut aplikasi Arduino IDE

Walaupun tampak seperti program Windows pada umumnya, namun sebetulnya program ini adalah sebuah program Java. Jika Anda menemukan sebuah pesan

kesalahan kemungkinan besar pada komputer belum terinstal Java Runtime Environment (JRE)atau Java Development Kit (JDK).

2. Jalankan menu **Tools → Board**, kemudian pilih tipe papan yang sesuai



Gambar 2.8 Jendela pemilihan bord Arduino

3. Jalankan menu **File → Examples → 01. Basic → Blink**. Ini adalah program sederhana yang fungsinya adalah membuat lampu LED menyala berkedip-kedip seperti yang telah dijelaskan sebelumnya.

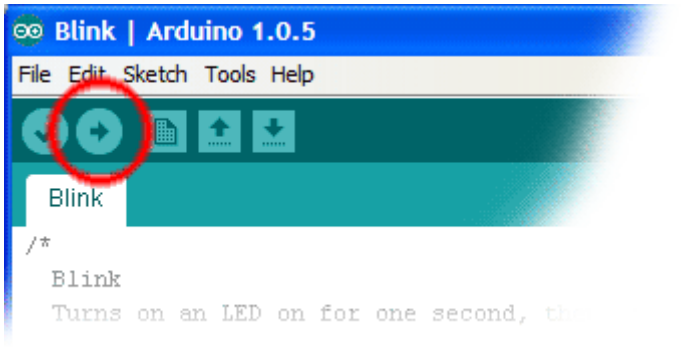
```
Blink
Turns on an LED on for one second, then off for one second,
repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
```

Gambar 2.9 Contoh program (codding) pada Arduino

Bagian yang di highligh adalah perintah untuk menunda aliran program selama satu detik (1000 milidetik). Jadi bila lampu LED diperintahkan menyala pada baris sebelumnya, maka dengan perintah delay() lampu itu akan bertahan menyala selama satu detik sebelum ia diperintahkan untuk padam pada baris berikutnya.

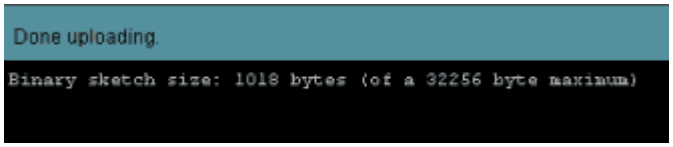
Semakin kecil angka pada delay maka interval nyala-padam menjadi lebih cepat.

4. Pada toolbar klik tombol Upload untuk memuat sketch tersebut ke dalam papan Arduino. Jika sketch berhasil dimuat akan ditandai dengan pesan berhasil seperti di bawah ini.



Gambar 2.10 Jendela untuk upload program

Jika sketch berhasil dimuat akan ditandai dengan pesan berhasil seperti di bawah ini.



Gambar 2.11 Jika program sudah selesai diupload

Namun jika sketch gagal dimuat (seperti pada kebanyakan kasus umumnya) maka akan muncul pesan kesalahan seperti berikut:

avrdude: stk500_getsync(): not in sync: resp=0x30

Solusinya cukup mudah, yaitu cukup mengganti pilihan serial port melalui menu **Tools → Serial Port**. Jika Anda tidak yakin pada port nomor berapa papan Arduino itu terhubung, coba pilih sebuah nomor port lalu jalankan upload seperti langkah sebelumnya. Jika pesan kesalahan masih muncul, ganti nomor port-nya dan lakukan berulang-ulang sampai upload berhasil.

Saat sketch yang sudah dimodifikasi tersebut berhasil dimuat ke dalam papan Arduino maka tampak lampu LED menyala dan padam dengan frekuensi yang lebih cepat.

2.4 Radio Frequency Identification (RFID)

merupakan sistem yang mengirimkan identitas tertentu (berbentuk serial number unik) dari objek tertentu secara nirkabel, menggunakan gelombang frekuensi radio. Teknologi ini merupakan bagian dari teknologi identifikasi otomatis seperti barcode, optical character readers, dan beberapa teknologi biometric seperti retinal scan. Teknologi identifikasi otomatis telah digunakan untuk mengurangi waktu dan tenaga dalam menginput data secara manual dan meningkatkan akurasi data. Di beberapa negara maju, tag RFID kini telah tergabung dengan uang tunai, pakaian, atau bahkan manusia. Ancaman bahwa teknologi ini dapat membaca informasi terkait personal tanpa disadari kini menjadi perhatian serius untuk privasi manusia Tag, atau label, RFID digunakan pada banyak industri. Tag RFID seringkali ditempel pada industri otomotif selama produksi untuk digunakan melacak perkembangan pada lini perakitan. Selain itu, RFID juga sering digunakan pada bidang farmasi dan pertanian untuk melacak stok (obat dan hewan) pada gudang sekaligus membantu proses operasional. Komponen Teknologi RFID ada 6 komponen utama pada teknologi RFID:

1. Tag
2. Reader
3. Antena

2.4.1 Tag

Tag RFID merupakan komponen untuk menandai objek yang ingin dikenali. Tag dapat berupa passive, active, atau battery-assisted passive. Active tag memiliki baterai on-board dan secara periodik mengirimkan sinyal ID. Battery-assited *passive* (BAP) tag memiliki baterai on-board dan diaktifkan ketika terdapat RFID reader. Passive tag sama sekali tidak menggunakan energi listrik, sehingga menjadi lebih kecil dan lebih murah karena tanpa baterai. Meski begitu, passive tag membutuhkan level energi sinyal yang lebih kuat tiga kali agar dapat beroperasi sehingga sistem ini rentan interferensi dan radiasi gelombang mikro.

Tag dapat bekerja dalam sistem read-only atau read/write. Read only berarti tag tidak dapat ditulis program atau data tertentu, sementara read/write dapat diprogram dan dibaca berkali-kali. Tag RFID terdiri dari bagian seperti IC dan memori untuk menyimpan dan memproses informasi, modulasi dan demodulasi sinyal frekuensi radio (RF), mengumpulkan energi DC dari pengirim sinyal, fungsi tertentu lainnya, serta antena untuk mengirim dan menerima sinyal.

2.4.2 Reader

Selain tag, komponen penting lainnya adalah reader. Secara umum terdapat dua sistem RFID jika dikelompokkan menurut tag dan reader: Passive Reader Active Tag (PRAT), Active Reader Passive Tag (ARPT), dan Active Reader Active Tag (ARAT).Sistem PRAT memiliki reader pasif yang hanya menerima sinyal radio dari active tag. Jangkauan sistem reader PRAT dapat disesuaikan pada jarak 0,3 – 610 meter, sehingga lebih fleksibel pada aplikasi menjangkau jarak yang luas seperti mengawasi aset-aset pada gudang penyimpanan.Sistem ARPT terdiri dari active reader yang mengirimkan sinyal pemeriksa dan juga menerima sinyal balasan otentikasi dari passive tag.Sistem ARAT menggunakan active tag yang dibangkitkan oleh sinyal pemeriksa dari active reader.

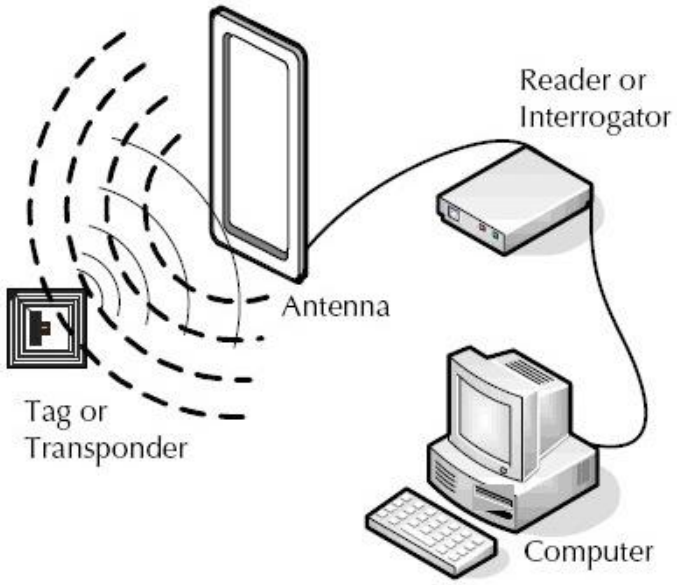
2.4.3 Antena

Antena pada RFID berfungsi untuk sebagai pengirim dan penerima sinyal. Antena yang digunakan dapat bervariasi bergantung pada biaya, fungsi, aplikasi, dan frekuensi operasi. Jumlah multiplexer merupakan variabel yang dipertimbangkan pada sebanyak apa antena yang akan digunakan. Konfigurasi dari multiplexer juga membutuhkan perangkat komunikasi lainnya seperti RS-485 dan perangkat eksternal lainnya yang mungkin juga dibutuhkan.Sistem kabel pada antena juga merupakan aspek penting agar RFID bekerja dengan baik. Terdapat efek penurunan amplitudo sinyal yang terjadi karena jarak antara reader dan antena. Sistem RFID terkadang membutuhkan kabel berkualitas bagus yang bisa jadi mahal dan memiliki batasan jarak. Kombinasi antara radar, antena, dan multiplexer merupakan syarat utama agar membuat sinyal terbaca.

2.4.4 Cara Kerja RFID

Terdapat tiga bagian utama pada pada sistem kerja RFID:

- Antena.
- Transceiver untuk men-decode data.
- Transponder yang telah diprogram dengan informasi tertentu berbentuk tag RFID.



Gambar 2.12 Rangkaian RFID

Antena berfungsi melakukan komunikasi via sinyal dengan tag RFID, dan menyediakan energi bagi RFID tag (hanya pada kasus tag RFID pasif). Ketika tag RFID mendapat dengan sinyal antena, perangkat ini akan mendeteksi sinyal aktivasi dari antena yang "membangunkan" chip RFID. Chip ini akan mengirim informasi untuk diterima antena.

Tag RFID ini tidak perlu ditempel pada permukaan objek, sehingga tidak harus ditempel atau digunakan menyatu dengan objek. Tag ini dapat dibaca waktu kurang dari 100 milisekon. Selain itu, melalui komunikasi dengan antena, host controller dapat membaca sejumlah tag dalam satu waktu sehingga lebih praktis dan lebih cepat.

2.5.1 Bahasa Pemograman

Sejarah C++

Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories awal tahun 1980-an berdasarkan C ANSI (American National Standard Institute). Pertama kali, prototype C++ muncul sebagai C yang dipercanggih dengan fasilitas kelas. Bahasa

tersebut disebut C dengan kelas (C wih class). Selama tahun 1983-1984, C dengan kelas disempurnakan dengan menambahkan fasilitas pembebanlebihan operator dan fungsi yang kemudian melahirkan apa yang disebut C++. Symbol ++ merupakan operator C untuk operasi penaikan, muncul untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C. Borland International merilis compiler Borland C++ dan Turbo C++. Kedua compiler ini sama-sama dapat digunakan untuk mengkompilasi kode C++. Bedanya, Borland C++ selain dapat digunakan dibawah lingkungan DOS, juga dapat digunakan untuk pemrograman Windows. Selain Borland International, beberapa perusahaan lain juga merilis compiler C++, seperti Topspeed C++ dan Zortech C++.

Contoh Program C :

```
# include <stdio.h>
Main ( )
{
Char pesan [ ] = “Hai, C programmers !” ;
Printf (pesan) ;
Return 0 ;
}
```

Contoh Program C++ :

```
# include <iostream.h>
Main ( )
```

```
{
Char pesan [ ] = “Hai, C programmers !” ;
Cout << pesan ;
      Return 0 ;
```

Tentang C++

C++ diciptakan untuk mendukung pemrograman berorientasi pada objek (Object Oriented Pragramming/OOP) yang tidak dimiliki C. sementara C merupakan bahasa pemrograman terbaik dilingkungannya, bahasa ini tidak memiliki kemampuan OOP. Reputasi C tidak diragukan lagi dalam menghasilkan program .EXE berukuran kecil, eksekusi yang cepat, antarmuka (interfacing) yang sederhana dengan bahasa lain dan fleksibilitas pemrograman. Apa yang membuat C tampak sukar dipelajari mungkin karena tiadanya pemeriksaan tipe. Sebagai contoh, dapat mencampur bilangan bulat dengan string untuk menghasilkan karakter. Namun, justru dsitu letak fleksibilitas C, dapat mengolah data C sebebas mengolah data dalam bahasa assembly.

Borland C++

Dibandingkan compiler C++ yang lain, Borland C++ memiliki keunggulan terutama dalam hal kecepatan dan efisiensi kompilasi. Disamping itu, Borland C++ mendukung beberapa system operasi yaitu DOS, Windows 16bit (Window 3.0) dan windows 32 bit (Windows NT). Meskipun demikian compiler Borland C++ juga memiliki kelemahan bila dibandingkan compiler C++ yang lain, misalnya : pemrograman dengan Borland C++ terutama yang menyangkut tampilan jauh lebih sulit daripada pemrograman dengan Microsoft Visual C++.

Struktur Bahasa C++

Program C maupun C++ selalu tersusun dari 4 (empat) bagian utama, yaitu :

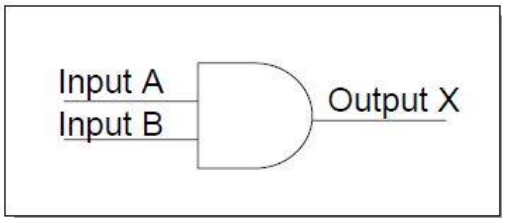
1. Bagian komentar yang ditandai dengan symbol // dan pasangan /* … */
2. Bagian pengarah compiler yang ditandai dengan symbol #
3. Bagian deklarasi
4. Bagian definisi

2.6 Gerbang logika dasar

Gerbang Logika ⟶ Blok dasar untuk membentuk rangkaian elektronika digital

- Sebuah gerbang logika mempunyai satu terminal output dan satu atau lebih terminal input
- Output-outputnya bisa bernilai HIGH (1) atauLOW (0) tergantung dari level-level digital padaterminal inputnya.
- Ada7 gerbang logika dasar: AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR

2.6.1 Gerbang logika AND



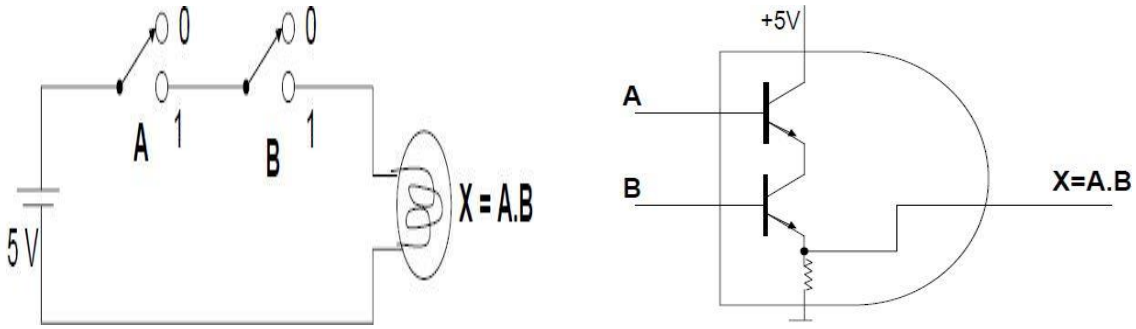
Gambar 2.13 Simbol Gerbang Logika AND

OperasiAND :

- Jika Input A dan B keduanya **HIGH**, maka output X akan **HIGH**
- Jika Input A atau B salah satu atau keduanya **LOW** maka output X akan **LOW**

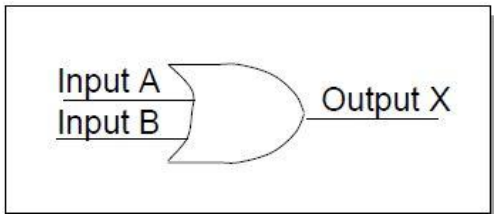
Table 2.2 Kebenaran Gerbang AND-2 Input

INPUT		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



Gambar 2.14 Analogi Elektrikal Gerbang AND

2.6.2 Gerbang OR



Gambar 2.15 Simbol Gerbang Logika OR

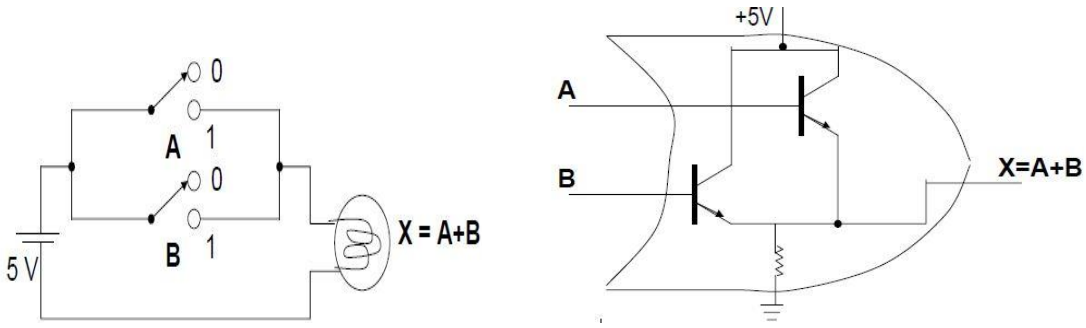
Operasi OR :

- Jika Input A atau B atau keduanya **HIGH**, maka output X akan **HIGH**

- Jika Input A dan B keduanya **LOW** maka output X akan **LOW**

Tabel 2.3 Kebenaran gerbang OR –2 input

INPUT		*Output*
A	**B**	**X**
0	0	0
0	1	1
1	0	1
1	1	1



Gambar 2.16 Analogi Elektrikal Gerbang OR

2.6.3 Gerbang NOT / INVERTER

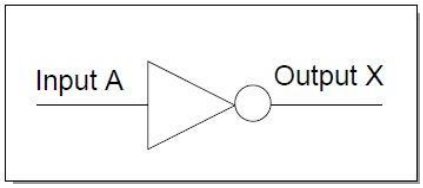


Table 2.4 kebenaran Gerbang Logika NOT

INPUT	*Output*
A	**X**
0	1
1	0

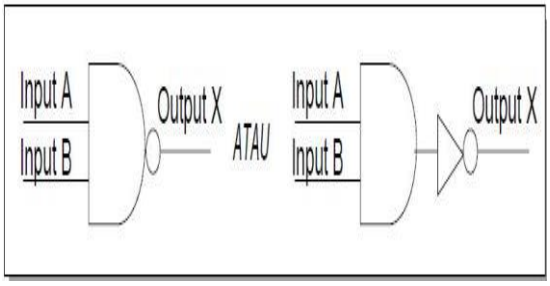
Gambar 2.17 Simbol gerbang logika NOT

OperasiNOT :

- Jika Input A **HIGH**, maka output X akan **LOW**
- Jika Input A **LOW,** maka output X akan **HIGH**

2.6.4 Gerbang NAND

Table 2.5 kebenaran gerbang NAND



INPUT		Output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

$X = \overline{A.B}$

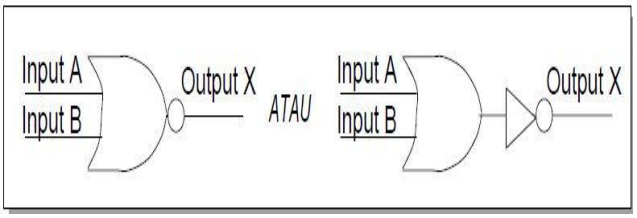
Gambar 2.18 Gerbang Logika NAND

OperasiNAND :

- Merupakan Inversi (kebalikan) dari operasi AND
- Jika Input A dan B keduanya **HIGH,**maka output X akan **LOW**
- Jika Input A atau B atau keduanya **LOW**, maka output X akan **HIGH**

2.6.5 Gerbang NOR

Table 2.6 kebenaran gerbang NOR



INPUT		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

$X = \overline{A+B}$

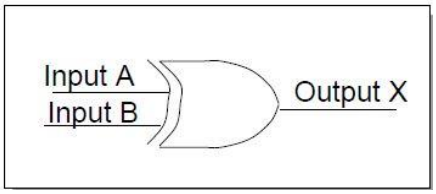
Gambar 2.19 Gerbang Logika NOR

OperasiNOR :

- Merupakan Inversi (kebalikan) dari operasi OR

- Jika Input A dan B keduanya**LOW,**maka output X akan **HIGH**
- Jika Input A atau B salah satu atau keduanya **HIGH**, maka output X akan **LOW**

2.6.6 Gerbang Ex-OR



Gambar 2.20 Gerbang Logika Ex-OR

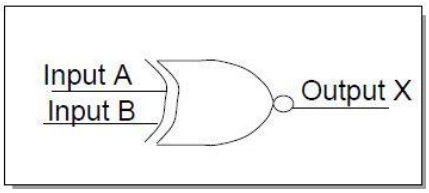
Table 2.7 kebenaran gerbang Ex-OR

INPUT		OUTPUT
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

OperasiEx-OR :

- Ex-OR adalah kependekan dari Exclusive OR
- Jika salah satu dari kedua inputnya **HIGH** (bukan kedua-duanya), maka output X akan **HIGH**
- Jika kedua inputnya bernilai **LOW** semua atau **HIGH** semua, maka output X akan **LOW**

2.6.7 Gerbang Ex-NOR



Gambar 2.21 Gerbang Ex-NOR

Table 2.8 kebenaran gerbang Ex-NOR

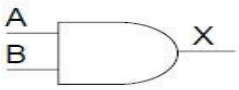
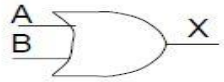
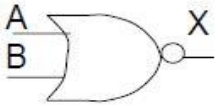
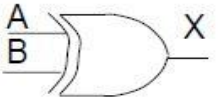
INPUT		OUTPUT
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

Operasi Ex-NOR :

- Ex-NOR merupakan kebalikan dari Ex-OR
- Jika salah satu dari kedua inputnya **HIGH** (bukan kedua-duanya), maka output X akan **LOW**
- Jika kedua inputnya bernilai **LOW** semua atau **HIGH** semua, Maka output X akan **HIGH**

RINGKASAN JENIS GERBANG LOGIKA

Tabel 2.9 Ringkasan Jenis-Jenis Gerbang logika

No	NAMA	TIPE IC	Simbol Logika	Persamaan	Tabel Kebenaran
1	AND	7408		$X=A.B$	INPUT A B / Output X: 0 0 0; 0 1 0; 1 0 0; 1 1 1
2	OR	7432		$X=A+B$	INPUT A B / Output X: 0 0 0; 0 1 1; 1 0 1; 1 1 1
3	NOT	7404		$X=\overline{A}$	INPUT A / Output X: 0 1; 1 0
4	NAND	7400		$X=\overline{A.B}$	INPUT A B / Output X: 0 0 1; 0 1 1; 1 0 1; 1 1 0
5	NOR	7402		$X=\overline{A+B}$	INPUT A B / Output X: 0 0 1; 0 1 0; 1 0 0; 1 1 0
6	Ex-OR	7486		$X=A\oplus B$	INPUT A B / OUTPUT X: 0 0 0; 0 1 1; 1 0 1; 1 1 0
7	Ex-NOR			$X=\overline{A\oplus B}$	INPUT A B / OUTPUT X: 0 0 1; 0 1 0; 1 0 0; 1 1 1