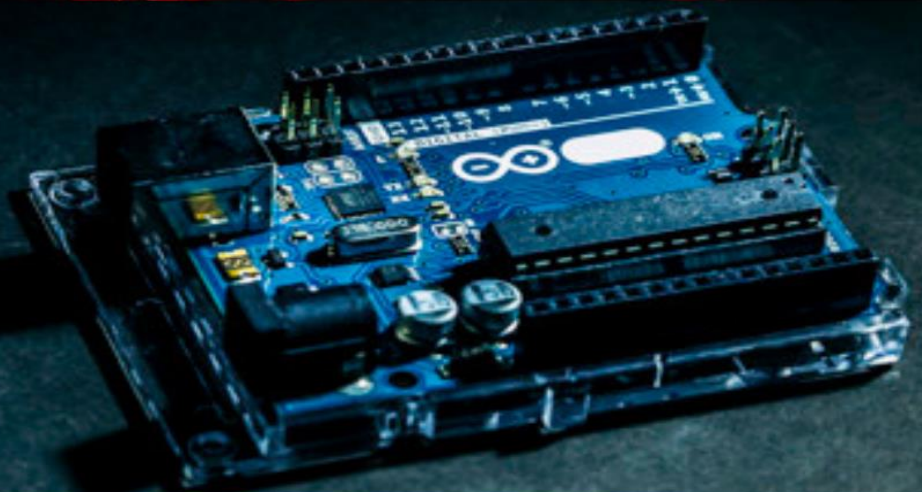
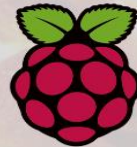




MODUL
PRAKTIKUM

ORGANISASI DAN ARSITEKTUR KOMPUTER



**LABORATORIUM KECERDASAN BUATAN
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL BANDUNG
2020**

PENGANTAR

Laboratorium Dasar Komputer setiap semester selalu melaksanakan kegiatan praktikum dan pada semester ganjil biasa dilakukan kegiatan praktikum Organisasi dan Arsitektur Komputer. Kegiatan praktikum ini dilaksanakan sebagai pendukung matakuliah yang sama yaitu Organisasi dan Arsitektur Komputer dengan prespektif pada perkembangan teknologi. Pada tahun ajaran 2020/2021 materi praktikum Organisasi dan Arsitektur Komputer adalah pemrograman Arduino dan norde MCU Esp8266. Kegiatan praktikum yang dilaksanakan 10 x pertemuan dengan 4 jam/pertemuan dan 4 orang asisten laboratorium untuk setiap kelas.

Pada praktikum Organisasi dan Arsitektur Komputer ajaran 2020/2021, pembuatan modul dan *jobsheet* praktikan dibuat sepenuhnya oleh **Koordinator Laboratorium** dan **Tim Asisten Laboratorium Dasar Komputer tahun ajaran 2015/2016**.

Koordinator : Milda Gustiana Husada, Ir, M.Eng

1	15.2017.012	Irsan Rasyidin
2	15.2017.015	Muh. Assidiq Fattah
3	15.2017.066	Lulu Rifqia Rachmaniar
4	15.2017.071	Revi Moch. Fikry
5	15.2017.084	Cindy Mawar Kasih
6	15.2017.096	Ni Komang Intan Tri Pujiani
7	15.2017.114	Anisa Putri Setyaningrum
8	15.2017.116	Moh. Faishal Dzaky
9	15.2017.130	Alhimny Dwinata Utama
10	15.2018.016	Moh. Muqiiit Faturahman
11	15.2018.021	Alifian Alvarez Firdlorizky
12	15.2018.082	Hazkia Kaikiba
13	15.2018.084	Annisa Olga Zerlinda
14	15.2018.109	Firly Taufikurohman
15	15.2018.118	Nafia Ruwaida Chosyyatillah
16	15.2018.130	Muh. Haikal

DAFTAR ISI

PENGANTAR	i
DAFTAR ISI.....	v
Deskripsi Alat yang digunakan praktikum	1
1.LED	1
2.SWITCH.....	2
3.BUZZER.....	3
4.Microphone (Mikrofon).....	4
5.Photo Transistor.....	4
6.Light Dependent Resistor (LDR)	5
7. Pengertian LCD I2C	6
MATERI – 1 Arsitektur	45
1.1 Arduino.....	45
1.1.1 Arsitektur Arduino.....	46
1.2 Raspberry PI	48
1.2.1 Arsitektur Raspberry pi.....	44
1.3 Node MCU Esp2866	45
1.3.1 Pinout NodeMCU V3	2
MATERI – 2 Bahasa C	14
2.1 Pengertian	14
2.2 Operator aritmatika , Comparison, dan logic di Bahasa C	14
2.3 Tipe data di Bahasa C.....	16
2.4 Penulisan IF, Else, IF bersarang	16
2.5 Penggunaa For dan while pada Bahasa C.....	17
MATERI – 3 PYTHON	24
3.1 Pengertian	24
3.2 Penulisan Kode Python.....	25
3.3 Operator aritmatika , Comparison, dan logic di Python.....	27
3.4 Tipe Data di Python.....	29
3.5 Penggunaan Variabel di Python	34
3.6 Penulisan IF, Else, Elif dan IF bersarang Python.....	36

3.7 Penggunaan For pada python	39
3.8 Penggunaan While Pada Python.....	40
MATERI – 4 LED	50
4.1. Percobaan 1 LED ON.....	50
4.2. Percobaan 2 LED OFF	51
4.3. Percobaan 3 LED Turn on With For	53
MATERI – 5 SWITCH.....	84
5.1.Percobaan 1 Switch Push Button.....	84
5.2.Percobaan 2 Switch + Buzzer.....	85
5.3.Percobaan 3 Switch + LED + Buzzer.....	87

Deskripsi Alat yang digunakan praktikum

1.LED

Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada Remote Control TV ataupun Remote Control perangkat elektronik lainnya.

Kegunaan

Berikut ini beberapa pengaplikasiannya LED dalam kehidupan sehari-hari.

1. Lampu Penerangan Rumah
2. Lampu Penerangan Jalan
3. Papan Iklan (Advertising)
4. Backlight LCD (TV, Display Handphone, Monitor)
5. Pemancar Infra Merah pada Remote Control (TV, AC, AV Player)

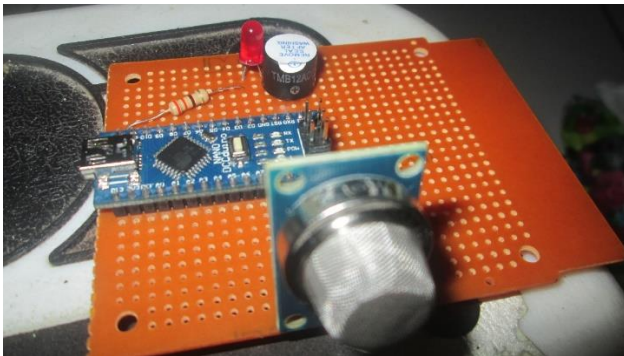
Manfaat

Manfaat dari penggunaan LED (Light Emitting Diode) adalah sebagai indikator dari berbagai peralatan dan saat ini mulai digunakan sebagai penerangan.

Contoh Produk

1. Alat Pendeteksi Kebakaran

LED dapat digunakan sebagai indikator saat terjadi kebakaran disuatu tempat.



Gambar 1 : Contoh LED 1

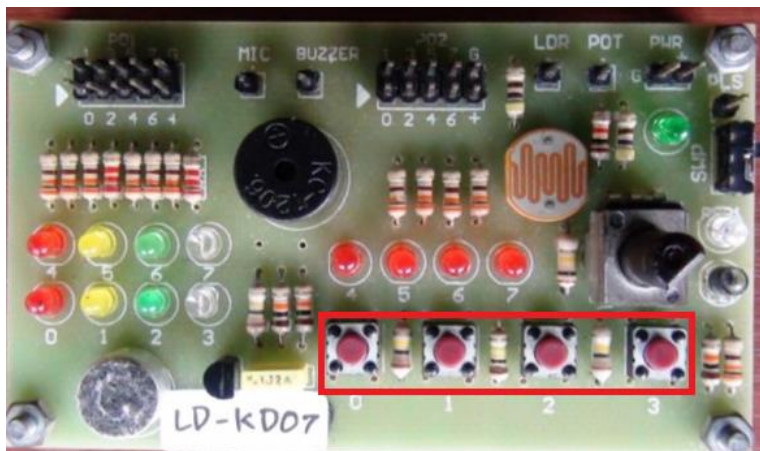
2. Keyboard dan Mouse

LED digunakan pada keyboard dan mouse misalkan, indikator untuk tombol Caps Lock on pada keyboard dan penanda mouse on atau off.



Gambar 2 : Contoh LED 2

2.SWITCH



Switch pada rangkaian elektronika sesungguhnya alat yang dapat atau memiliki fungsi untuk menghubungkan dan memutuskan aliran listrik (arus listrik) pada jaringan arus listrik kuat maupun jaringan arus listrik yang lemah.

Kegunaan

fungsi untuk menghubungkan dan memutuskan aliran listrik (arus listrik) pada jaringan arus listrik kuat maupun jaringan arus listrik yang lemah.

Manfaat

Dapat memutuskan atau menyambungkan sebuah alat yang terhubung dengan listrik

Contoh Produk

Saklar lampu.

3.BUZZER

Buzzer Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara.

Kegunaan

Pada umumnya, Buzzer yang merupakan sebuah perangkat audio ini sering digunakan pada rangkaian anti-maling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada kendaraan.

Manfaat

Manfaat dari penggunaan buzzer adalah sebagai perangkat peringatan bahaya.

Contoh produk

1. Alarm pintu

Buzzer dapat digunakan sebagai alarm pada pintu yang menandakan pintu sedang dibuka. Jadi jika pintu terbuka maka penghalang listrik akan terlepas dan listrik akan mengalir kemudian LED akan menyala dan buzzer akan berbunyi.



2. Indikasi motor

Buzzer dapat diaplikasikan sebagai alarm pada motor dalam keadaan diparkirkan yang tersenggol atau disenggol. Setelah motor tersenggol atau disenggol maka lampu LED akan menyala dan suara alarm akan berbunyi.



4. Microphone (Mikrofon)

Microphone atau dalam bahasa Indonesia disebut dengan Mikrofon adalah suatu alat atau komponen Elektronika yang dapat mengubah atau mengkonversikan energi akustik (gelombang suara) ke energi listrik (Sinyal Audio). Microphone (Mikrofon) merupakan keluarga Transduser yang berfungsi sebagai komponen atau alat pengubah satu bentuk energi ke bentuk energi lainnya. Setiap jenis Mikrofon memiliki cara yang berbeda dalam mengubah (konversi) bentuk energinya, tetapi mereka semua memiliki persamaan yaitu semua jenis Mikrofon memiliki suatu bagian utama yang disebut dengan Diafragma (Diaphragm)

Kegunaan

Mikrofon adalah menangkap gelombang suara dan mengubahnya menjadi getaran listrik sinyal Analog untuk selanjutnya diperkuat dan diolah sesuai dengan kebutuhan, pengolahan berikutnya dengan Power Amplifier dari suara yang berintensitas rendah menjadi lebih keras

Manfaat Mic

Menginput suara yang masuk sehingga dapat menghitung frekuensi dari suara tersebut

Contoh Produk

1. Mendeteksi frekuensi dari suatu suara
Alarm di kamar bayi – saat bayi menangis alarm akan berbunyi dan ortu si bayi akan datang melihat anaknya.

5. Photo Transistor

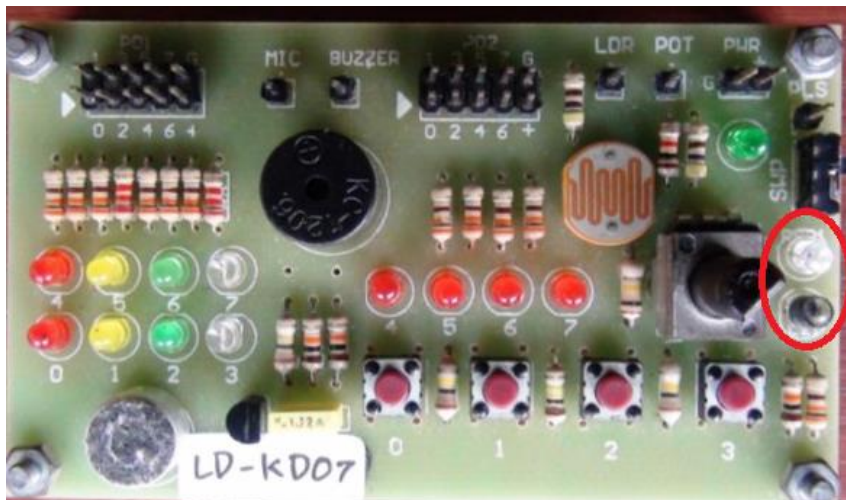


Photo Transistor adalah Transistor yang dapat mengubah energi cahaya menjadi listrik dan memiliki penguat (gain) Internal. Penguat Internal yang terintegrasi ini menjadikan sensitivitas atau kepekaan Photo Transistor terhadap cahaya jauh lebih baik dari komponen pendeteksi cahaya lainnya seperti Photo Diode ataupun Photo Resistor. Photo Transistor hanya memiliki dua kaki yaitu Kolektor dan Emitor sedangkan terminal Basisnya berbentuk lensa yang berfungsi sebagai sensor pendeteksi cahaya. Photo Transistor pada umumnya dikemas dalam bentuk transparan pada area dimana Photo Transistor tersebut menerima cahaya.

Kegunaan

Photo Transistor dirancang khusus untuk aplikasi pendeteksian cahaya sehingga memiliki Wilayah Basis dan Kolektor yang lebih besar dibanding dengan Transistor normal umumnya.

Manfaat

Sensor ini bisa di gunakan untuk menghidupkan lampu taman secara otomatis, jika kondisi Ruangan sudah mulai gelap, maka sensor cahaya akan deteksi, kemudian digunakan untuk mematikan lampu atau menyalakan lampu.

Contoh Produk

Lampu Otomatis

Aplikasi Sensor Fototransistor sebagai Pembaca Kode Barang berupa Kombinasi Garis Hitam dan Putih

6.Light Dependent Resistor (LDR)



Salah satu jenis resistor yang besar hambatannya tergantung dari cahaya yang diterimanya. Semakin besar intensitas cahaya yang datang maka semakin banyak elektron yang lepas dari ikatan. Sehingga intensitas akan turun saat cahaya meneranginya. LDR mempunyai nilai hambatan yang sangat besar ketika tak ada cahaya yang meneranginya (gelap) yakni mencapai 1 Mega ohm atau 1,000,000,000 m Ω . Sebaliknya jika terkena cahaya, nilai hambatan LDR akan turun secara drastis hingga beberapa puluh ohm saja. jarak efektif yang dapat diterima sensor 0.5 - 1 cm

Kegunaan : Untuk mendeteksi intensitas cahaya.

Manfaat :

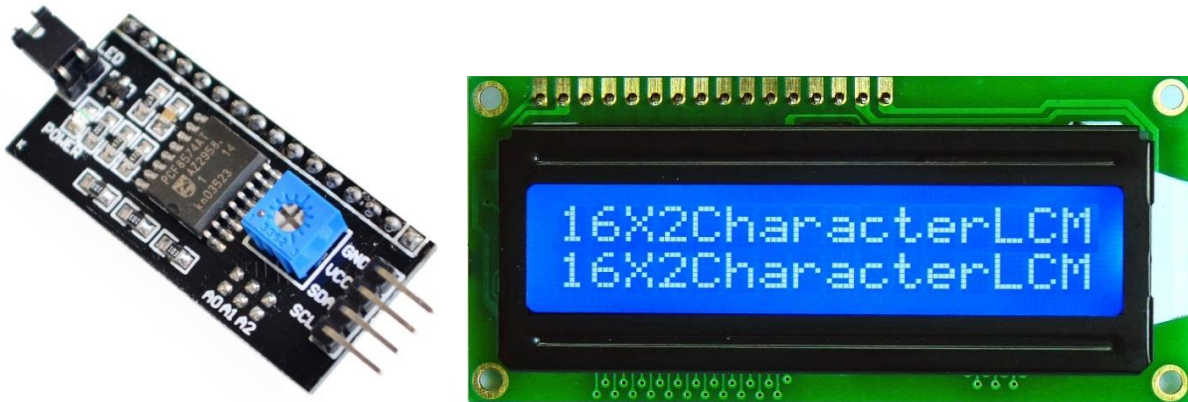
1. Untuk menghidupkan atau mematikan lampu jalan raya secara otomatis. Bayangkan jika lampu jalan raya berjumlah ribuan dan dimatikan secara manual dan akan sulit untuk mengontrol lampu itu semua. Dan semakin maju nya teknologi bahkan sekarang sudah dijadikan rangkaian untuk membangun smarthome.

2. Sistem otomatis yang dapat memudahkan pekerjaan menjadi lebih terkontrol dengan SDM yang terbatas.

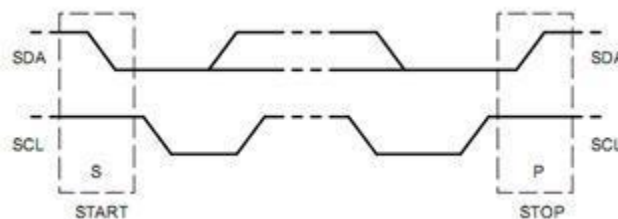
Contoh Produk :

1. Aplikasi sensor warna berdasarkan nilai intensitas yang dimiliki tiap warna
2. Atap otomatis pada aplikasi jemuran adalah sebuah alat yang akan bekerja apabila cuaca cerah maka pakaian akan dikeluarkan untuk di jemur tetapi sebaliknya apabila cuaca mendung atau hujan maka secara otomatis akan memasukkan pakaian sehingga tidak bisa terkena air hujan.
3. Aplikasi sistem penyiraman tanaman otomatis
4. Aplikasi Lampu pintar
5. Aplikasi buka tutup jendela & gordeng otomatis. Gordeng yang berfungsi membuka atau menutup gordeng ketika keadaan terang atau gelap.
6. Saklar otomatis pada smarthome

7. Pengertian LCD I2C

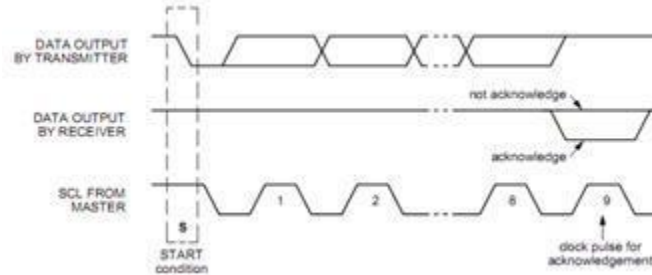


Inter Integrated Circuit atau sering disebut I^2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I^2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I^2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I^2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *transfer* data pada I^2C Bus dengan membentuk sinyal *Start*, mengakhiri *transfer* data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master*. Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal *Start* dan sinyal *Stop* seperti tampak pada Gambar 1.



Kondisi sinyal start dan stop

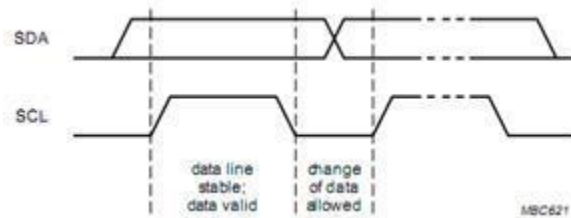
Sinyal dasar yang lain dalam I²C Bus adalah sinyal *acknowledge* yang disimbolkan dengan ACK Setelah transfer data oleh *master* berhasil diterima *slave*, *slave* akan menjawabnya dengan mengirim sinyal *acknowledge*, yaitu dengan membuat SDA menjadi “0” selama siklus *clock* ke 9. Ini menunjukkan bahwa *Slave* telah menerima 8 bit data dari *Master*. Kondisi sinyal *acknowledge* seperti tampak pada Gambar 2.



Sinyal ACK dan NACK

Dalam melakukan *transfer* data pada I²C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- *Transfer* data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- Selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal *Start* atau sinyal *Stop*.



Trasfer Bit pada I²C bus

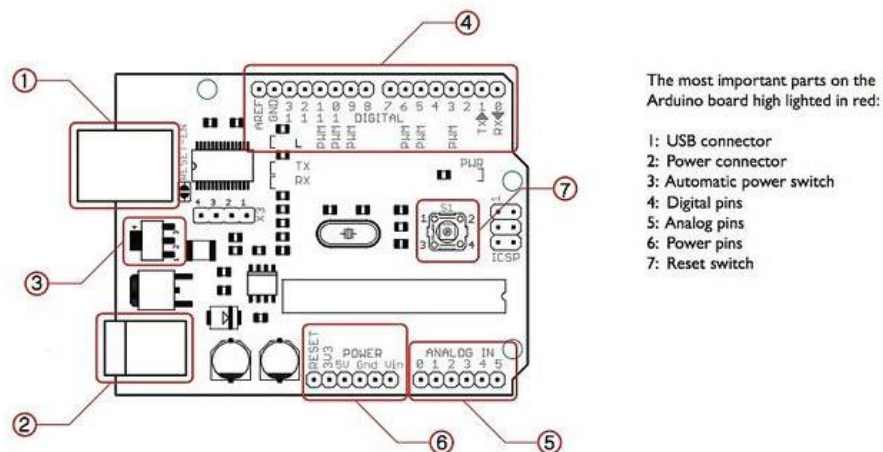
MATERI – 1 Arsitektur

1.1 Arduino

Arduino Uno adalah board berbasis mikrokontroler pada ATmega 328. Board ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, jack listrik dan tombol reset. Pin – pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya (Arduino, Inc., 2009). Arduino Uno R3 berbeda dengan semua board sebelumnya karena Arduino Uno R3 ini tidak menggunakan chipdriver FTDI USB-to-serial. Melainkan menggunakan fitur dari ATmega 16U2 yang diprogram sebagai konverter USB-to-serial.



Gambar 1 Arduino uno R3



Gambar 2 parts arduino

Konfigurasi pin papan Arduino Uno ditunjukkan di atas. Ini terdiri dari 14-pin i / o digital. Dimana 6 pin digunakan sebagai modulasi lebar pulsa o / ps dan 6 analog i / ps, koneksi USB, colokan listrik, osilator kristal 16MHz, tombol reset, dan header ICSP. Papan Arduino dapat diberi daya baik dari komputer pribadi melalui USB atau sumber eksternal seperti baterai atau adaptor. Board ini dapat beroperasi dengan suplai eksternal 7-12V dengan memberikan referensi tegangan melalui pin IOREf atau melalui pin Vin.

I / Ps Digital

Ini terdiri dari 14-pin I / O digital, masing-masing pin mengambil dan menyediakan arus 40mA. Beberapa pin memiliki fungsi khusus seperti pin 0 & 1, yang masing-masing bertindak sebagai pemancar dan penerima. Untuk komunikasi serial, pin-2 & 3 adalah interupsi eksternal, 3,5,6,9,11 pin memberikan PWM o / p dan pin-13 digunakan untuk menghubungkan LED.

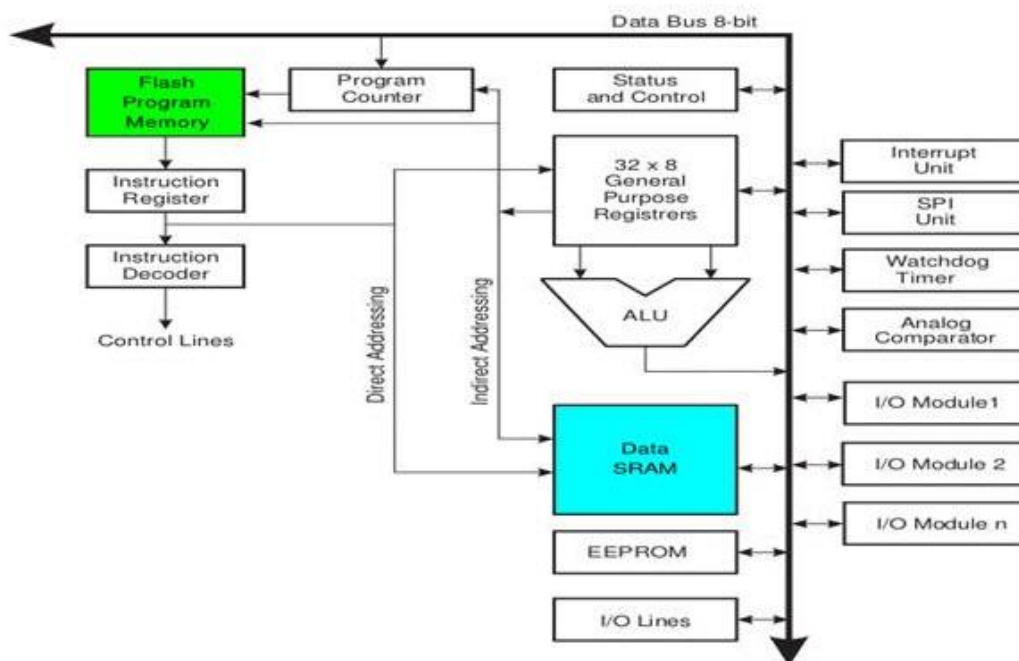
Analog i / ps: Memiliki 6-analog I / O pin, masing-masing pin memberikan resolusi 10 bit.

Aref: Pin ini memberikan referensi ke analog i / ps.

Reset: Ketika pin rendah, maka mikrokontroler akan disetel ulang.

1.1.1 Arsitektur Arduino

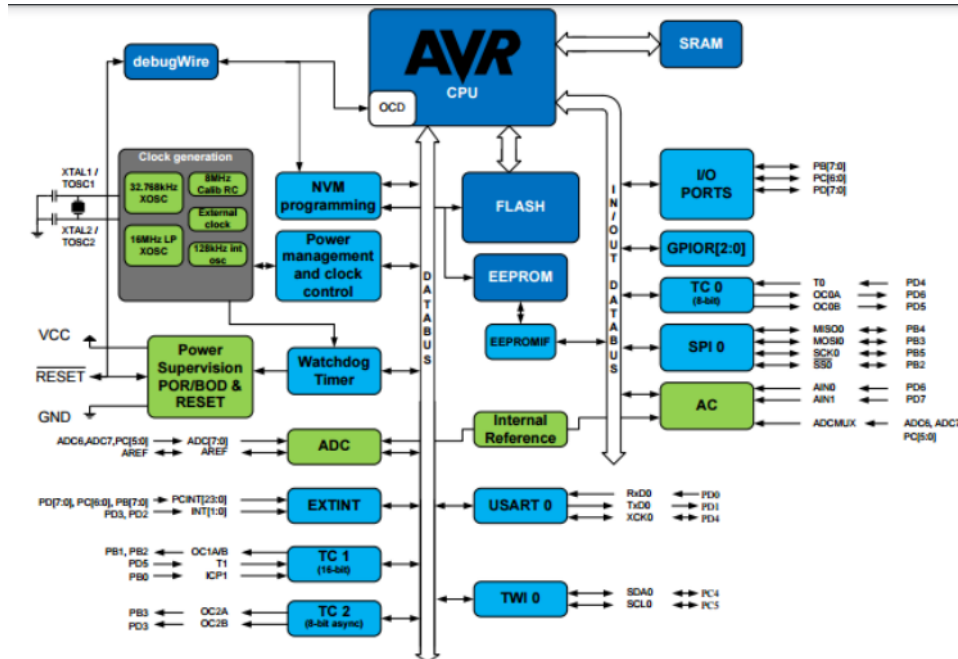
Pada dasarnya, pengolah papan Arduino menggunakan arsitektur Harvard dimana kode program dan data program memiliki memori terpisah. Ini terdiri dari dua memori seperti memori program dan memori data. Dimana data tersebut disimpan dalam memori data dan kodenya disimpan dalam memori program flash. Mikrokontroler Atmega328 memiliki memori flash 32kb, 2kb SRAM 1kb EPROM dan beroperasi dengan kecepatan clock 16MHz.



Gambar 3 Arsitektur arduino

1. ATEMEGA328

ATMega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (Reduce Instruction Set Computer) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (Completed Instruction Set Computer). Mikrokontroler berbasis AVR RISC 8-bit



Gambar 4 Scematik Atemega328

- Memori flash program 32KB
- EEPROM 1KB
- SRAM 2KB
- Frekuensi clock maks 20MHz
- 23 pin GPIO
- 32 general purpose registers
- 3 timers/counters
- Internal & external interrupts
- USART
- 2-wire serial interface
- SPI port
- 6-channel 10-bit A2D

2. AVR ARCHITECTURE

- Harvard architecture
- Separate memories & buses for program and data
- Instructions executed in single-level pipeline
- Fast-access register file
- 32 x 8 bit general purpose registers

- Single clock cycle access time
- Single cycle ALU operation

1.2 Raspberry PI

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal (Single Board Circuit /SBC) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game, bahkan bisa digunakan sebagai media player karena kemampuannya dalam memutar video high definition. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris.

Ide dibalik komputer mungil ini diawali dari keinginan untuk mencetak generasi baru programmer, pada 2006 lalu. Seperti disebutkan dalam situs resmi Raspberry Pi Foundation, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft, dari Laboratorium Komputer Universitas Cambridge memiliki kekhawatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Mereka lantas mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada 2009. Tiga tahun kemudian, Raspberry Pi Model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Kini, sekitar dua tahun kemudian, Raspberry Pi telah terjual lebih dari 2,5 juta unit ke seluruh dunia.

Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memory yang digunakan, Model A menggunakan memory 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan ethernet port (kartu jaringan) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang. Raspberry Pi merupakan komputer mini yang sangat murah, harganya hanya 25 dollar AS untuk Model A adapun 35 dollar AS untuk Model B per unit. Hardware Raspberry Pi tidak memiliki real-time clock, sehingga OS harus memanfaatkan timer jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi

real-time (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (General-purpose input/output) via antarmuka I²C (Inter-Integrated Circuit).

Raspberry Pi bersifat open source (berbasis Linux), Raspberry Pi bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa pemrograman Python. Salah satu pengembang OS untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian diklaim mampu memaksimalkan perangkat Raspberry Pi. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS.

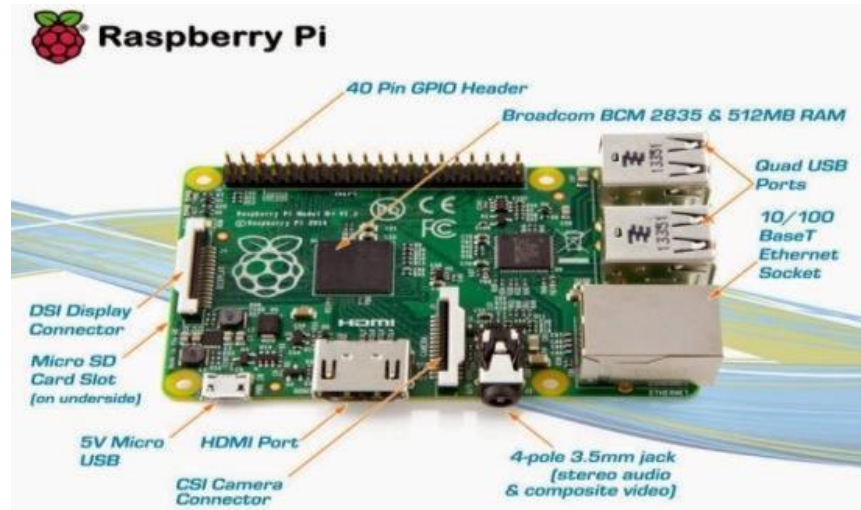
Specification

- Chip : Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
- CPU : 700 MHz ARM1176JZF-S core (ARM6 family)
- GPU : Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
- Memory (SDRAM) : 512 MB (shared with GPU)
- USB 2.0 ports : 2 (via integrated USB hub)
- Video outputs : Composite RCA (PAL & NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from * 640×350 to 1920×1200 plus various PAL and NTSC standards.
- Audio outputs : 3.5 mm jack, HDMI
- Onboard storage : SD / MMC / SDIO card slot
- Onboard network : 10/100 Ethernet (RJ45)
- Low-level peripherals: 8 × GPIO, UART, I²C bus, SPI bus with two chip selects, +3.3 V, +5 V, ground[58][63]
- Power ratings : 700 mA (3.5 W)
- Power source : 5 volt via MicroUSB or GPIO header
- Size : 85.60 × 53.98 mm (3.370 × 2.125 in)
- Weight : 45 g (1.6 oz)
- Operating systems : Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS

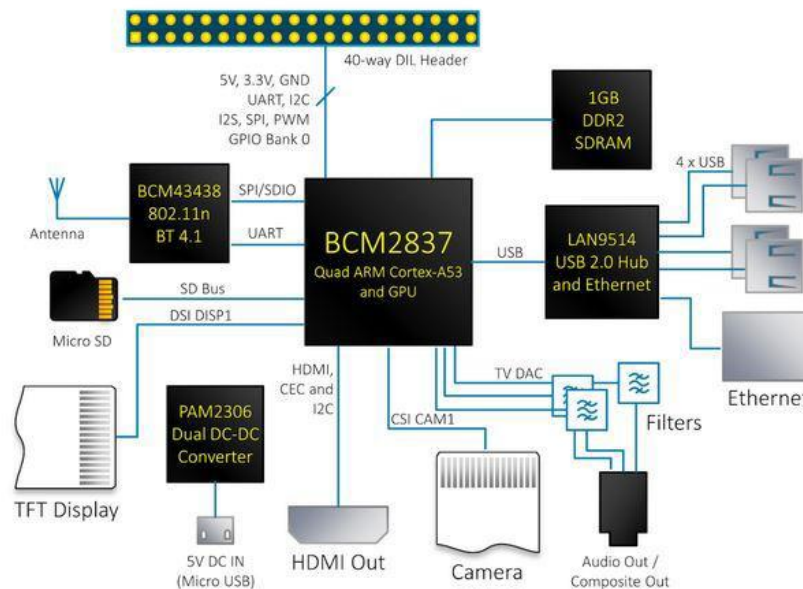
Sumber : https://id.wikipedia.org/wiki/Raspberry_Pi

1.2.1 Arsitektur Raspberry pi

Model B+ merupakan revisi terbaru dari Model B. Terdapat 4 *slot* USB dan 40 pin GPIO. *Slot Power micro* USB di ubah ke sebelah kanan dan *slot* kartu SD juga telah diganti dengan *slot micro* SD yang jauh lebih kuat.



A. Diagram Blok Raspberry Pi B+



1. **Broadcom BCM2837 ARM Cortex-A53**

Sebuah prosesor yang berfungsi sebagai pengendali *Raspberry Pi B+*

2. **HDMI out**

Merupakan port HDMI yang sama ditemukan di banyak televisi dan monitor komputer saat ini. Penggunaan kabel standard HDMI untuk menghubungkan antara *Raspberry Pi* dengan layar monitor yang sesuai.

3. **CSI connector camera**

Camera serial interface Raspberry Pi memiliki dua jalur data, satu jalur jam dan satu port I2C. Port I2C digunakan untuk mengkonfigurasi kamera dan mengirim perintah dan data gambar/video diterima melalui jalur data.

4. **Ethernet Out**

Ethernet Out adalah cara untuk terhubung dengan jaringan internet. Untuk terhubung dengan jaringan internet ialah dengan menggunakan kabel *Ethernet*. Kabel *ethernet* memungkinkan terhubung dengan akses internet yang stabil dan lebih cepat, hanya saja kurang nyaman untuk dibawa berpindah-pindah. Karena dibatasi dengan panjang dari kabel *ethernet* tersebut.

5. **USB 2.0**

Raspberry Pi memiliki empat buah port USB, yang memungkinkannya untuk dapat terhubung dengan *keyboard*, *mouse*, *Wi-Fi dongle*, dan *USB stick* yang berisikan berkas terkait, secara bersamaan.

6. **AUDIO OUTPUT**

Terdapat socket headphone 3,5 mm jack, yang memungkinkan *Raspberry Pi* terhubung dengan speaker.

7. **DSI Display connector**

Display Serial Interface memiliki 15 biasanya digunakan untuk display LCD seperti LCD pada ponsel.

8. **SD card slot**

SD card slot digunakan sebagai media penyimpanan dan *booting* dari *Raspberry Pi*. Dimana *operating system* yang digunakan tersimpan beserta dengan berkas lain yang diperlukan.

9. **Micro USB power**

Micro USB power adalah untuk *power*, artinya memungkinkan untuk menggunakan *charger smartphone* yang sesuai untuk *Raspberry Pi*

10. **GPIO Header**

Pin GPIO (*General Purpose Input/Output*) merupakan kumpulan pin yang dapat dimanfaatkan untuk banyak keperluan. Namun untuk fungsi utama dari GPIO sendiri ialah untuk menghubungkan *Raspberry Pi* dengan perangkat rangkaian elektronik lainnya. Untuk kemudian mengendalikan perangkat tersebut. Keseluruhan terdapat 40 pin GPIO dalam *single board computer* ini.

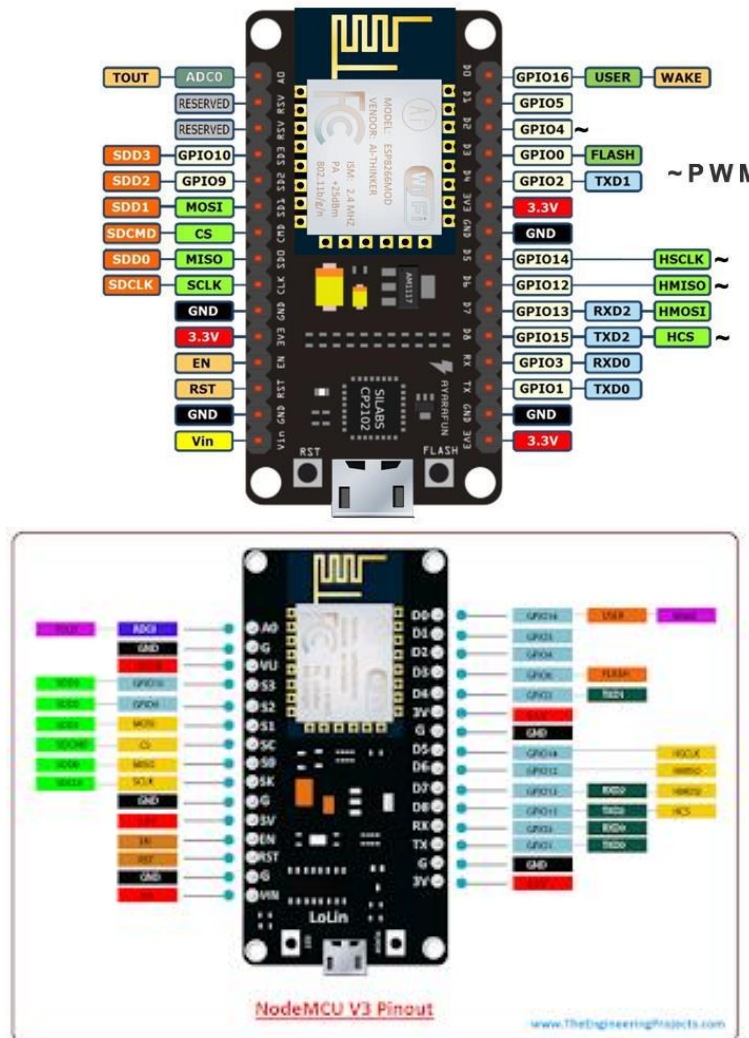
1.3 Node MCU Esp8266

NodeMCU V3 adalah firmware sumber terbuka dan kit pengembangan yang memainkan peran penting dalam mendesain produk IoT menggunakan beberapa baris skrip. Beberapa pin GPIO di papan memungkinkan kami untuk menghubungkan papan dengan periferal lain dan mampu menghasilkan komunikasi serial PWM, I2C, SPI, dan UART.

Antarmuka modul ini terutama dibagi menjadi dua bagian termasuk Firmware dan Perangkat Keras di mana yang pertama berjalan pada SoC Wi-Fi ESP8266 dan yang lebih baru didasarkan pada modul ESP-12. Firmware ini didasarkan pada Lua - Bahasa skrip yang mudah dipelajari, memberikan lingkungan pemrograman sederhana yang dilapisi dengan bahasa skrip cepat yang menghubungkan Anda dengan komunitas pengembang terkenal.

1.3.1 Pinout NodeMCU V3

NodeMCU V3 hadir dengan sejumlah Pin GPIO. Gambar berikut menunjukkan Pinout NodeMCU.



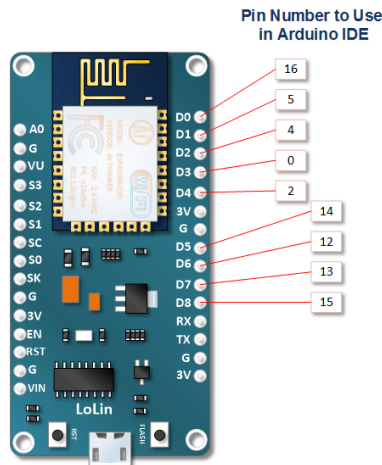
Ada perbedaan yang jelas antara Vin dan VU di mana yang pertama adalah tegangan yang diatur yang mungkin berdiri di suatu tempat antara 7 hingga 12 V sementara kemudian adalah tegangan daya untuk USB yang harus dijaga sekitar 5 V.

Features

1. Open-source
2. Arduino-like hardware
3. Status LED
4. MicroUSB port
5. Reset/Flash buttons
6. Interactive and Programmable
7. ESP8266 with inbuilt wifi
8. USB to UART converter

9. GPIO pins
10. Arduino-like hardware IO
11. Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware.
12. Code like arduino, but interactively in Lua script
13. Nodejs style network API
14. Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm*5mm sized MCU in Nodejs style.
15. Greatly speed up your IOT application developing process.
16. We provide the best platform for IOT application development at the lowest

Note: GPIO node MCU dipemogramkan Arduino Idle ada hal yang perlu di perhatikan dalam melakukan set terhadap pin yang akan digunakan dalam melakukan pemograman akan ditunjukkan oleh gambar berikut :



1.4 Uji Pemahaman

Jawablah pertanyaan di bawah ini :

1. Apa hubungan Arduino dengan Organisasi dan Arsitektur Komputer ?
2. Apa perbedaan Arduino dan Raspberry Pi ?
3. Apa perbedaan beberapa versi Raspberry Pi ?
4. Jelaskan IO pada Arduino !
5. Jelaskan IO pada Raspberry Pi !
6. Jelaskan Arsitektur pada Raspberry Pi !
7. Jelaskan Arsitektur pada Arduino !
8. Apa perbedaan RAM & ROM & EPROM ?
9. Apa yang dimaksud dengan "FirmWare" ?
10. Apa yang dimaksud pin digital dan pin analog ?
11. Ada berapa IO yang digunakan secara digital pada Arduino UNO R3 ?
12. Ada berapa IO yang digunakan secara analog pada Arduino UNO R3 ?
13. Ada berapa IO yang digunakan secara digital pada NodeMCU ?
14. Ada berapa IO yang digunakan secara analog pada NodeMCU ?
15. Jelaskan perbedaan istilah "mikroprosesor", "mikrokontroler" ?

MATERI – 2 Bahasa C

2.1 Pengertian

Bahasa pemrograman c merupakan bahasa pemrograman tingkat tinggi, itu dikarenakan bahasa c dapat dimengerti dan dipelajari dengan mudah karena kedekatannya dengan bahasa manusia. Tapi banyak orang juga mengatakan bahwa bahasa c adalah medium level programming language karena bahasa c juga dapat digunakan untuk memasukkan program ke mesin.

Bahasa C dirancang oleh Dennis M. Ritchie pada tahun 1972 di AT&T Bell Labs. Bahasa C dikembangkan dari bahasa BPCL (Basic Combined Programming Language) dan bahasa B. Bahasa BPCL di kembangkan oleh Martin Richard pada tahun 1967 sebagai bahasa system operasi dan compiler. Ken Thompson pada tahun 1970 telah merancang bahasa B dengan memasukkan feature BPCL. Bahasa B dirancang untuk membuat system operasi UNIX/LINUX untuk computer DEC PDP-7 pada Bell Laboratories.

2.2 Operator aritmatika , Comparison, dan logic di Bahasa C

1. Operator Aritmatika Bahasa C

Operator Aritmatika adalah operator yang digunakan untuk operasi bilangan seperti penjumlahan, pengurangan, perkalian, pembagian, modulus, increment dan decrement. Operator aritmatika bisa digunakan pada semua tipe bilangan seperti char, int, long int dan float. Operator aritmatika juga bisa menangani tipe signed dan unsigned. Increment adalah operasi bilangan dimana bilangan hasil merupakan bilangan asal ditambah satu, sedangkan decrement adalah operasi bilangan dimana bilangan hasil merupakan bilangan asal dikurang satu.

Tabel 1 Operator Aritmatika

Operator	Description	Example
+	Adds two operands.	$A + B = 30$
-	Subtracts second operand from the first.	$A - B = -10$
*	Multiplies both operands.	$A * B = 200$
/	Divides numerator by de-numerator.	$B / A = 2$
%	Modulus Operator and remainder of after an integer division.	$B \% A = 0$
++	Increment operator increases the integer value by one.	$A++ = 11$
--	Decrement operator decreases the integer value by one.	$A-- = 9$

1. Operator Comparison (Perbandingan)

Operator perbandingan (comparison) adalah operator yang digunakan untuk membandingkan dua buah nilai atau variabel. Operator ini terdapat dalam sebuah ekspresi yang selanjutnya akan menentukan benar atau tidaknya ekspresi tersebut. Nilai yang dibandingkan bisa berupa angka maupun string. Hasil dari perbandingan ini berupa nilai boolean, yaitu true (benar) yang bernilai 1 atau false (salah) yang bernilai 0.

Tabel 2 Operator Perbandingan

Operator	Description	Example	Keterangan
==	Sama dengan	a == b	benar jika a sama dengan b
!=	Tidak sama dengan	a != b	benar jika a berbeda dengan b
>	Lebih besar	a > b	benar jika a lebih besar dari b
<	Lebih kecil	a < b	benar jika a lebih kecil dari b
>=	Lebih besar atau sama dengan	a >= b	benar jika a lebih besar atau sama dengan b
<=	Lebih kecil atau sama dengan	a <= b	benar jika a lebih kecil atau sama dengan b

2. Operator Logic

Operator Logika (Logical Operator) Adalah operator yang digunakan untuk menangani tipe data boolean. Operator logika berfungsi untuk membandingkan dua operand bertipe Boolean untuk memecahkan masalah berdasarkan operator yang digunakan, dan hasil dari operasi operator ini akan menghasilkan nilai bertipe Boolean 1 (true) atau 0 (false), tetapi ada satu operator dalam macam-macam operator logika yang tidak bekerja untuk membandingkan yaitu operator dengan bentuk tanda '!', karena operator '!' hanya membutuhkan satu operand bertipe Boolean.

Tabel 3 Operator Logic

Operator	Description
	Jika ekspresi sisi kiri bernilai Benar (True) maka langsung menghasilkan Benar (True).
&&	Jika ekspresi sisi kiri bernilai Salah (False) maka langsung menghasilkan Salah (False).
!	Berfungsi untuk mengembalikan nilai sebenarnya

2.3 Tipe data di Bahasa C

Tipe data ialah klasifikasi data yang mengenalkan kompilator atau penerjemah bagaimana programmer bermaksud untuk menggunakan data. Sebagian besar bahasa pemrograman mendukung tipe data dasar akan bilangan integer, bilangan titik mengambang, karakter dan boolean. Sebuah tipe data menyediakan sekumpulan nilai-nilai dari ekspresi (yaitu variabel, fungsi) yang dapat mengambil nilai tersebut. Tipe data ini mendefinisikan operasi yang akan dilakukan pada data, maksud dari data dan jurusan nilai dari tipe tersebut dapat disimpan.

Tabel 4 Tipe data Bahasa C

Tipe data	Bahasa C
Text Type:	String, char
Numeric Types:	Int, unsight int, sight int, long int, float, double
Boolean Type:	True or False
Binary Types:	bit

2.4 Penulisan IF, Else, IF bersarang

Sama dengan bahasa pemrograman lainnya, Bahasa C menggunakan pernyataan if. Pernyataan if ini berisi sebuah ekspresi logika menggunakan data yang telah dibandingkan, dan menghasilkan sebuah keputusan yang dibuat berdasarkan perbandingan tersebut. Aturan penulisan untuk pernyataan if sebagai berikut:

Tabel 5 Penulisan IF dll

	Bahasa C
Satu kondisi	if (kondisi) {pernyataan benar;}
Dua kondisi	if (kondisi) {pernyataan benar1 ;} else {pernyataan salah ;}
Tiga kondisi atau lebih	If (kondisi 1) {pernyataan benar1 ;} else if (kondisi 2) {pernyataan benar2 ;} else if (kondisi 3) {pernyataan benar3 ;} else {pernyataan salah ;}

2.5 Penggunaa For dan while pada Bahasa C

Struktur Perulangan For dan while Bahasa C ,Struktur perulangan (atau dalam *bahasa inggris* disebut dengan loop) adalah instruksi kode program yang bertujuan untuk mengulang beberapa baris perintah.

Dalam merancang perulangan, kita setidaknya harus mengetahui 3 komponen:

1. Kondisi awal perulangan.
2. Kondisi pada saat perulangan.
3. Kondisi yang harus dipenuhi agar perulangan berhenti.

Tabel 6 Penulisan Program For dan While di Bahasa C

	Bahasa C
Syntax For	For (expresi 1; ekspresi2;ekspresi3) {pernyataan yang diulang;}
Syntax While	While (expresi) { Pernyataan yang diulang; }

MATERI – 3 PYTHON

Dalam bagian ini, akan dijelaskan sekilas tentang python dan bagaimana cara penggunaannya yang terdiri dari pengertian, cara penulisan kode python, operator aritmatika yang digunakan, tipe data, penggunaan variabel, penggunaan if, penggunaan else, penggunaan elif, penggunaan IF bersarang, penggunaan for, penggunaan while dan penggunaan break dan continue pada python.

3.1 Pengertian

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah:

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- OS/2
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Lihat sejarahnya di Python Copyright. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi Open Source maupun General Public License (GPL).

3.2 Penulisan Kode Python

Penulisan kode program pada python ada dalam tiga cara, yaitu (1) *mode interaktif*, (2) *mode skrip*, dan (3) *mode IDLE*.

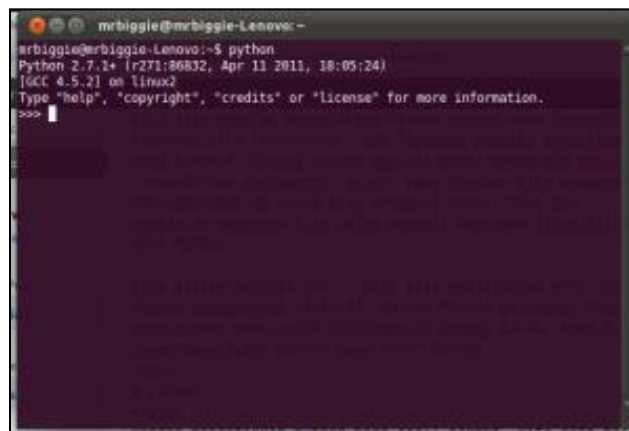
(1) Mode Interaktif

Setiap sistem operasi pasti mempunyai sebuah command-line interpreter, misal: pada Windows memiliki DOS, sedangkan di Linux memiliki Shell. Mode ini sangatlah membantu dalam menguji beberapa fitur-fitur dari Python.

Pada sistem operasi Linux, mode ini bisa dijalankan menggunakan Terminal. Karena Python merupakan bahasa pemrograman yang sudah terintegrasi dengan Linux, maka cukup menuliskan python pada shell prompt.

```
$ python
```

Dengan mengetikkan python pada shell prompt, maka akan masuk ke mode interaktif Python yang akan ditandai dengan ">>>".



Gambar. 1 Python Mode Interaktif

(2) Mode Skrip

Python juga dapat di jalankan dengan mengeksekusi langsung sebuah skrip program pada shell prompt. Cukup dengan menuliskan sebuah perintah:

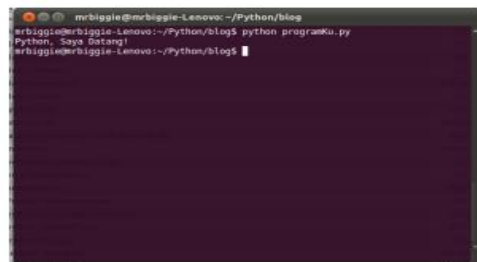
```
$ python programKu.py
```

Perintah ini akan mengeksekusi skrip *programKu.py* secara langsung dari direktori yang aktif sekarang.

Langkah pertama buat terlebih dahulu file *programKu.py*, kemudian simpan di folder kerja. Dalam menuliskan kode *programKu.py*, bisa dilakukan dengan menggunakan Text Editor yang telah disediakan oleh Linux, diantaranya *gEdit* atau *Geany*.

Berikut listing kode *programKu.py*:

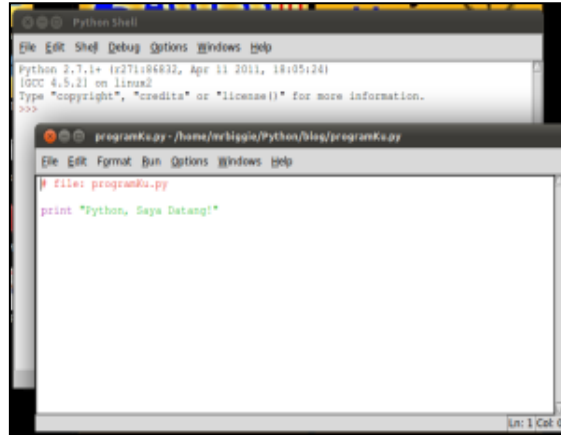
```
1 # file: programKu.py
2
3 print "Python, Saya Datang!"
```



Gambar . 2 Python Mode Skrip

(3) Mode IDLE

Python juga dapat dijalankan pada mode grafik yang disebut **mode IDLE**. IDLE berbasis **Tkinter**, untuk dapat menjalankannya harus terlebih dahulu menginstal *Tcl/Tk*. IDLE dijalankan langsung dari folder `/usr/lib/python2.7/idlelib/`. Berikut tampilan IDLE:



Gambar. 3 Python Mode IDLE

3.3 Operator aritmatika , Comparison, dan logic di Python

1. Operator Aritmatika

Peran operator dalam proses perhitungan matematika sangatlah penting. Selain operator Aritmatika, Python juga mendukung operator berkondisi yang berfungsi untuk membandingkan suatu nilai dengan nilai yang lain. Operator-operator yang didukung oleh Python yaitu operator Unari (+ dan –) dan operator Binari (+, -, *, /, %, dan **). Pada ekspresi Aritmatika berikut:

$$x = y + z$$

y dan z disebut sebagai operan dari operator +.

Tabel.1 di bawah ini menjelaskan tentang berbagai macam operator yang digunakan untuk segala perhitungan di Python.

Tabel. 1 operator perhitungan Python

Operator Aritmatika			
OPERATOR	DESKRIPSI	CONTOH	KETERANGAN
+	Menjumlahkan dua operan	$x = y + z$	Menjumlahkan nilai y dan z, kemudian hasilnya disimpan ke x
-	Mengurangi operan kiri dengan operan kanan	$x = y - z$	Mengurangi y dengan z, kemudian hasilnya disimpan ke x

**	Memangkatkan operan kiri sebesar operan kanan	$x = y ** z$	y dipangkatkan sebesar z, kemudian hasilnya disimpan ke x
*	Mengalikan dua operan	$x = y * z$	Mengalikan y dengan z, kemudian hasilnya disimpan ke x
/	Membagi operan kiri dengan operan kanan	$x = y / z$	Membagi nilai y dengan z, kemudian hasilnya disimpan ke x. Jika kedua operan bertipe integer, maka hasilnya akan berupa integer dengan pembulatan ke bawah. Tetapi jika salah satu atau kedua operan bertipe float, maka hasilnya akan berupa float (disertai tanda desimal).
%	Menghitung sisa dari pembagian integer	$x = y \% z$	Membagi nilai y dengan z, dan sisanya kemudian disimpan ke x

Jika sebuah ekspresi melibatkan lebih dari satu operator, Python secara otomatis akan memilih operator mana yang akan diutamakan dahulu. Sebagai contoh:

```
>>>                                     x=7+3*6
>>>                                     x
25
>>>                                     y=100/4*5
>>>                                     y
125
```

Untuk memahami contoh di atas, dapat dilihat table.2 pengurutan operator Aritmatika versi Python di bawah ini.

Tabel. 2 pengurutan operator Aritmatika versi Python

Tabel Urutan Operator Aritmatika		
TYPE	OPERATOR	URUTAN
Nilai Konstruksi	()	Dari dalam ke luar
Eksponensial	**	Tertinggi
Perkalian	//, *, /, %	Kiri ke kanan
Penjumlahan	+, -	Kiri ke kanan

Operator ****** memiliki urutan tertinggi diantara operator lainnya. Operator ***** mempunyai urutan lebih tinggi daripada operator **+**, dan operator **/** mempunyai urutan yang sama dengan operator *****. Pada ekspresi $x = 7 + 3 * 6$, bagian $3 * 6$ akan dieksekusi pertama kali menghasilkan 18, yang kemudian ditambahkan dengan 7. Sedangkan ekspresi $y = 100/4*5$, bagian $100/4$ dieksekusi terlebih dahulu karena operator **/** berada disebelah kiri dari operator *****.

2. Operator Comparison dan logic (Perbandingan)

Sesuai dengan namanya, operator perbandingan membandingkan nilai dari 2 **operand**. Hasilnya selalu salah satu dari **TRUE** atau **FALSE**. Hasil perbandingan akan bernilai **TRUE** jika kondisi perbandingan tersebut benar, atau **FALSE** jika kondisinya salah. Operand untuk operator perbandingan ini bisa berupa tipe data angka (**integer** atau **float**), maupun bertipe **string**. Operator perbandingan akan memeriksa **nilai** dan (untuk beberapa operator) **juga tipe data** dari **operand**.

Tabel. 3 operator Comparison versi Python

Nama	Operator Python
Sama Dengan	<code>==</code>
Tidak sama dengan	<code>!=</code>
Lebih besar	<code>></code>
Lebih kecil	<code><</code>
Lebih besar sama dengan	<code>>=</code>
Lebih kecil sama dengan	<code><=</code>
AND	<code>and</code>
OR	<code>Or</code>
NEGASI	<code>Not</code>

3.4 Tipe Data di Python

A. Menggunakan tipe Number

Tipe data Number digunakan untuk menyimpan nilai-nilai numerik. Tipe ini merupakan tipe data immutable, yang artinya jika mengubah nilai dari sebuah data, maka akan mengalokasikan obyek baru. Sama seperti tipe data lainnya, obyek Number dibuat ketika memberikan sebuah nilai padanya. Contoh:

```
>>>data = 1
```

Juga dapat mengubah nilai yang ada dalam variable data tersebut.

```
>>>data = data + 1
```

```
>>>data=3.50
```

```
>>>floatdat = 7.5
```

```
>>>data = floatdat
```

Sebuah obyek ataupun banyak obyek dapat dihapus dengan menggunakan pernyataan del.

Misalnya:

```
>>>del data
```

```
>>>del data, floatdat
```

Python mengelompokkan tipe Number dalam 4 macam, yaitu:

Plain Integer

Plain integer atau bilangan bulat merupakan tipe data yang sering ditemukan pada semua bahasa pemrograman. Integer ini mempunyai range nilai antara -2^{32} sampai $2^{31} - 1$. Tipe ini juga dapat ditulis dalam bentuk octal (di tanda awalan "0") maupun hexadecimal (ditandai awalan "0x" atau "0X"). Contoh:

```
10 100 6542 -784
```

```
083 -042 -0x43 0X61
```

Long Integer

Long integer sangat membantu untuk perhitungan di luar range nilai integer. Secara virtual, tidak ada batasan nilai tergantung besar virtual memory yang digunakan. Akhiran 'l' atau 'L' disetiap nilai bilangan bulat menandakan bahwa data tersebut bertipe long integer.

```
562718819L -0x526718L 012L -567299101L
```

Floating Point Real Number

Tipe ini sering disebut sebagai tipe real (atau float). Tipe ini sama dengan tipe double di C. Nilai float mempunyai dua bagian, bagian titik desimal dan bagian eksponensial. Tanda positif atau negatif diantara "e" merupakan tanda eksponen. Contoh nilai float:

```
0.0 14.5 -15.4 32.3+e18
```

```
-90.76712 -90. -32.54e100 70.2-E12
```

Complex Number

Sebuah bilangan kompleks biasanya ditunjukkan oleh bentuk $a + bj$, dimana a adalah bagian real dan b adalah bagian imajiner. Bagian imajiner merupakan bilangan di awal tanda “j” atau “J”. Berikut ini contoh bilangan kompleks:

3.14j 45j 54.56+12.1J 3e+36J

Bagian real dan imajiner dari bilangan kompleks dapat dipisahkan menggunakan data atribut, yaitu menggunakan `real` dan `imag`. Sedangkan untuk mendapatkan konjugasi dari bilangan kompleks tersebut, dapat menggunakan metode `conjugate()`.

```
>>> kompleks = 23.45-1.23J
>>> kompleks.real
23.45
>>> kompleks.imag
-1.23
>>> kompleks.conjugate()
(23.45+1.23j)
```

B. Menggunakan tipe karakter (String)

String merupakan salah satu tipe data yang sering digunakan dalam pemrograman Python. Sebuah string dapat dinyatakan sebagai kumpulan karakter yang dibatasi oleh satu atau dua tanda petik. Berikut contohnya,

```
>>> nama = "Klinik Python Indonesia"
>>> nama
'Klinik Python Indonesia'
>>> slm = 'Salam Python Dahsyat!'
>>> slm
'Salam Python Dahsyat!'
>>> print slm
Salam Python Dahsyat!
```

Dari contoh di atas, ketika akan memanggil variabel secara langsung maka akan ditampilkan

Menampilkan Tanda Petik Sebagai String

Di dalam sebuah string tidak dapat berisi tanda petik yang sama dengan tanda petik yang digunakan oleh string tersebut. Misalkan, ketika ingin menuliskan 'Python' maka akan muncul pesan kesalahan (syntax error). Agar tidak muncul pesan kesalahan, bisa dilakukan dengan cara mengganti tanda petik luarnya dengan tanda petik ganda, misalnya "Py'thon". Tanda petik juga dapat ditulis setelah tanda backslash (\) agar dapat ditampilkan sebagai string.

```
>>> str = "Py'thon"
>>> str
"Py'thon"
>>> str2 = 'Py"thon'
>>> str2
'Py"thon'
>>> "\"OK, \" sampai ketemu lagi.\"
'"OK, \" sampai ketemu lagi.'
```

Menggabungkan String

Untuk menggabungkan dua buah string atau lebih, dapat menggunakan operator +. Sedangkan untuk menggandakan string, dapat menggunakan operator *.

```
>>> blog = 'Klinik' + 'Python'
>>> blog
'KlinikPython'
>>> newblog = blog*5
>>> newblog
'KlinikPythonKlinikPythonKlinikPythonKlinikPythonKlinikPython'
>>> blog *= 4
>>> print blog
KlinikPythonKlinikPythonKlinikPythonKlinikPython
```

Jika dua string ditulis secara berurutan, maka secara langsung kedua string tersebut akan digabungkan.

```
>>> blog = 'Klinik' 'Python'
>>> blog
'KlinikPython'
```

Menentukan Panjang String

Panjang dari sebuah string dapat ditemukan dengan menggunakan fungsi `len()`.

```
>>> len(blog)
12
```

Memecah String

Tidak seperti bahasa lainnya, **Python** tidak mendukung *tipe Karakter*. Untuk mengambil satu karakter atau lebih dari sebuah string, dapat dilakukan dengan cara memecah string tersebut menggunakan *indeks* (disebut **Metode Irisan**). Irisan terdiri dari dua indeks yang dipisahkan tanda koma.

```
>>> buah = 'Nanas'
>>> buah[0]
'N'
>>> buah[0:2]
'Na'
>>> buah[0:4]
'Nana'
>>> buah[0:5]
'Nanas'
```

Dari contoh di atas, panjang string buah adalah 5. Ketika menghitung maju, indeks bernilai 0 sampai (panjang-string – 1) dimulai dari kiri ke kanan. Maka dari itu, dapat mengakses setiap karakter dalam range 0 sampai 4.

Sebuah string juga dapat dihitung mundur, dengan indeks -1 sampai (negatif panjang-string) dimulai dari kanan ke kiri. Berikut gambaran lengkapnya, baik itu penghitungan maju atau mundur.

0	1	2	3	4
N	a	n	a	s
-5	-4	-3	-2	-1

Contoh penggunaan penghitungan mundur,

```
>>> buah[-1]
's'
```



```

>>>                                     buah[-5]
'N'
>>>                                     buah[-5:-1]
'Nana'
>>>                                     buah[1:-1]
'ana'

```

Jika lupa berapa nilai indeks awala atau indeks akhir, dapat dilakukan dengan mengkosongkan indeks tersebut.

```

>>>                                     buah[:3]
'Nan'
>>>                                     buah[2:]
'nas'

```

Pengosongan indeks akan menyebabkan semua string ditampilkan.

```

>>>                                     buah[:]
'Nanas'

```

3.5 Penggunaan Variabel di Python

Tidak seperti pemrograman lainnya, variabel pada Python tidak harus dideklarasikan secara eksplisit. Pendeklarasian variabel terjadi secara otomatis ketika memberikan sebuah nilai pada suatu variabel. Seperti bahasa pemrograman pada umumnya, tanda sama-dengan (=) digunakan untuk memberikan nilai pada suatu variabel. Operan di sebelah kiri dari tanda (=) adalah nama variabel, sedangkan operan yang sebelah kanan dari tanda (=) adalah nilai yang diberikan pada variabel.

```

>>> harga = 100
>>> diskon = 25
>>> harga - diskon
75

```

Pada contoh di atas, 100 dan 25 merupakan nilai yang diberikan pada variabel harga dan diskon. Sedangkan pernyataan harga-diskon akan menghitung selisih antara harga dengan diskon. Variabel juga dapat menyimpan suatu nilai berupa teks (dibaca string).

```
>>>a = 'sekolah'
>>>b = 'dasar'
>>>a + b
'sekolahdasar'
```

Variabel juga dapat menyimpan dua nilai string atau lebih dengan menggunakan **operator (+)**.

```
>>>c = 'Py' + 'thon'
>>>c
'Python'
```

Jika telah memberikan nilai pada suatu variabel, variabel tersebut dapat digunakan dalam ekspresi yang lain.

```
>>>a = 2
>>>a = a + 3
>>>a
5
```

Juga dapat memberikan sebuah nilai untuk beberapa variabel.

```
>>>p=q=r=1
>>>p
1
>>>q
1
>>>r
1
```

Selain itu, dapat memberikan beberapa nilai untuk beberapa variabel (disebut multiple assignment).

```
>>>x, y, z = 1, 2, 'belajar Python'
>>>x
1
```

```
>>>y
2
>>>z
'belajar Python'
```

Bentuk lain dari contoh di atas, bisa menggunakan tanda kurung-buka kurung-tutup.

```
>>>(x, y, z) = (1, 2, 'belajar Python')
```

3.6 Penulisan IF, Else, Elif dan IF bersarang Python

(1) IF

Sama dengan bahasa pemrograman lainnya, Python juga menggunakan pernyataan if. Pernyataan if ini berisi sebuah ekspresi logika menggunakan data yang telah dibandingkan, dan menghasilkan sebuah keputusan yang dibuat berdasarkan perbandingan tersebut. Aturan penulisan untuk pernyataan if sebagai berikut:

```
1 | if kondisi:
2 |     pernyataan_benar
```

Semua pernyataan dituliskan dengan *indentasi* (alinea) setelah bagian kondisi bersyarat. Python menggunakan indentasi untuk mengelompokkan satu atau beberapa pernyataan.

Pada pernyataan if, syarat kondisi akan dijalankan pertama kali. Jika kondisi tersebut benar, maka pernyataan pada blok pernyataan_benar akan dijalankan. Berikut contohnya,

```
1 | >>> x = 10
2 | >>> if x>0:
3 | ...     print 'Hello'
```

Pada contoh di atas, pernyataan if akan menampilkan teks 'Hello' jika nilai variabel x lebih besar dari 0.

(2) Else

Pernyataan else ini biasanya digabungkan dengan pernyataan if di atas. Pernyataan else dapat berisi satu atau beberapa blok pernyataan (kode) yang mana akan dijalankan jika kondisi tersebut salah (atau tidak sesuai dengan syarat yang berlaku). Berikut syntax untuk if ... else,

```
1  if kondisi:
2  ...     pernyataan_benar
3  ... else:
4  ...     pernyataan_salah
```

Sebagai contoh, untuk mengetahui apakah suatu bilangan merupakan genap atau ganjil, bisa dilakukan dengan menggunakan kode seperti berikut ini,

```
1  >>> if (bilangan%2 == 0):
2  ...     print 'genap'
3  ... else:
4  ...     print 'ganjil'
```

Pada contoh di atas, '*genap*' akan ditampilkan jika sisa dari pembagian bilangan dengan 2 sama dengan 0. Dan sebaliknya jika tidak sama dengan 0, maka akan ditampilkan '*ganjil*'.

(3) Elif

Pernyataan elif memperbolehkan untuk menguji beberapa kondisi bersyarat untuk tiap-tiap nilai dan menjalankan kode pernyataan yang sesuai dengan syarat yang diberlakukan. Berikut syntax dari pernyataan elif,

```
1  if kondisi_1:
2  ...     pernyataan_1_benar
3  elif kondisi_2:
4  ...     pernyataan_2_benar
5  ...     :
6  ...     :
7  elif kondisi_N:
8  ...     pernyataan_N_benar
9  else:
10 ...     pernyataan_tidak_sesuai_syarat_di_atas
```

Contoh berikut ini menggunakan pernyataan elif untuk menampilkan apakah huruf yang diinput merupakan huruf vokal. Jika tidak, maka sebuah pesan lain akan ditampilkan.

```

1  huruf = raw_input("Masukkan sebuah huruf: ")
2
3  if (huruf == 'a'):
4      print "Ini adalah huruf vokal - a -"
5  elif (huruf == 'e'):
6      print "Ini adalah huruf vokal - e -"
7  elif (huruf == 'i'):
8      print "Ini adalah huruf vokal - i -"
9  elif (huruf == 'o'):
10     print "Ini adalah huruf vokal - o -"
11 elif (huruf == 'u'):
12     print "Ini adalah huruf vokal - u -"
13 else:
14     print "Ini bukan huruf vokal"

```

Pada contoh kode di atas, ketika seorang user mengetikkan sebuah huruf maka huruf tersebut akan diproses apakah huruf vokal atau bukan. Jika user mengetikkan huruf 'a', maka sesuai syarat yang berlaku maka akan ditampilkan 'Ini adalah huruf vokal - a -'. Jika kondisinya tidak sesuai, misalkan user mengetikkan huruf 'u', maka akan dilanjutkan kondisi bersyarat elif yang berikutnya sampai ditemukan kondisi yang sesuai dan akan ditampilkan hasilnya 'Ini adalah huruf vokal - u -'. Jika user mengetikkan huruf 'x', maka tidak ada syarat yang sesuai dengan kondisi bersyarat if dan elif. Dan pernyataan else yang dijalankan yaitu 'Ini bukan huruf vokal'.

(4) IF bersarang

Pada kondisi tertentu, dapat menggunakan pernyataan if bersarang. Maksudnya adalah dapat menuliskan pernyataan if..else di dalam pernyataan if..else. Berikut contoh penggunaan if bersarang pada program untuk menentukan apakah huruf yang dimasukkan user adalah huruf besar atau huruf kecil,

```
1 | huruf = raw_input("Masukkan sebuah huruf: ")
2 |
3 | if (huruf >= 'A'):
4 |     if (huruf <= 'Z'):
5 |         print "Ini adalah Huruf Besar"
6 |     elif (huruf >= 'a'):
7 |         if (huruf <= 'z'):
8 |             print "Ini adalah huruf kecil"
9 |         else:
10 |             print "Huruf > z"
11 |     else:
12 |         print "Huruf > z tapi < a"
13 | else:
14 |     print "Huruf < A"
```

3.7 Penggunaan For pada python

Pengulangan **for** pada **pemrograman Python** mempunyai kemampuan untuk melakukan iterasi data-data yang berurutan, seperti **list** atau **string**. Syntax umumnya adalah,

```
1 | for variabel in urutan:
2 |     pernyataan_yang_diulang
```

Jika urutan berupa sebuah list, maka urutan tersebut akan diuji dahulu. Kemudian data-data dari urutan tersebut akan dijadikan sebagai variabel. Selanjutnya, blok pernyataan_yang_diulang akan dieksekusi. Berikut akan diperlihatkan bagaimana pengulangan for bekerja dalam tipe urutan yang berbeda,

```
1 | >>> for huruf in 'Indonesia':
2 |     ...     print "Huruf: ", huruf
```

Keluaran yang ditampilkan dari kode di atas adalah:

<i>Huruf:</i>	<i>I</i>
<i>Huruf:</i>	<i>n</i>
<i>Huruf:</i>	<i>d</i>
<i>Huruf:</i>	<i>o</i>
<i>Huruf:</i>	<i>n</i>
<i>Huruf:</i>	<i>e</i>


```
Huruf: s
Huruf: i
Huruf: a
```

Ketika pengulangan `for` untuk iterasi `string`, variabel iterasi diasumsikan sebagai nilai dari setiap karakter pada `string`. Keluaran dari pengulangan `for` dalam bentuk satu karakter mengindikasikan bahwa urutan iterasi yang digunakan berupa `string`, bukan berupa obyek seperti `list`.

Jika ingin menampilkan sebuah daftar nama orang diikuti dengan panjangnya nama tersebut, maka lebih cocok menggunakan pengulangan `for` menggunakan urutan bertipe `list`. Berikut contohnya,

```
1 >>> nama = ['Fahmi', 'Dodit', 'Agus', 'Biggie']
2 >>> for x in nama:
3     print "Nama %-3s panjangnya %d karakter" %(x, len(x))
```

Pada kode di atas, bentuk `%` digunakan untuk memformat keluaran dan fungsi `len()` digunakan untuk menghitung panjang dari tiap-tiap item dalam daftar nama. Keluarannya seperti berikut,

```
Nama      Fahmi      panjangnya      5      karakter
Nama      Dodit      panjangnya      5      karakter
Nama      Agus       panjangnya      4      karakter
Nama Biggie panjangnya 6 karakter
```

3.8 Penggunaan While Pada Python

Pengulangan `while` akan berhenti jika pengujian suatu kondisi sudah tidak sesuai alias `False`. Yang mana, pengujian kondisi ini menggunakan *pengujian logika* dan akan memberikan nilai benar atau salah. Syntax umum dari pengulangan `while` adalah sebagai berikut,

```
1 while pengujian_kondisi:
2     pengulangan_kode
```

Berikut

contoh

penggunaan while,

```
1  angka = input("Masukkan sebuah angka yang lebih besar dari 1: ")
2
3  bilangan_1 = 0
4  bilangan_2 = 1
5
6  print bilangan_1
7
8  while(bilangan_2 < angka):
9      print bilangan_2
10
11     bilangan_2 = bilangan_1 + bilangan_2
12     bilangan_1 = bilangan_2 - bilangan_1
```

Hasil dari program di atas jika memasukkan angka 200 adalah,

```
Masukkan    sebuah    angka    yang    lebih    besar    dari    1:    200
0
1
1
2
3
5
8
13
21
34
55
89
144
```

Pada contoh di atas, bagian pengulangan_kode pada pengulangan while akan diulang beberapa kali sampai nilai bilangan_2 lebih besar daripada nilai angka. Pada setiap iterasi (pengulangan), nilai bilangan_2 akan bertambah sebesar bilangan_1, dan nilai bilangan_1 akan diubah menjadi selisih antara bilangan_2 dengan bilangan_1 itu sendiri.

MATERI – 4 LED

Dalam bagian ini, akan dijelaskan penggunaan raspberry pi untuk menyalakan LED pada kit display. Materi yang dibahas terdiri dari percobaan LED On, percobaan LED Off, percobaan LED Blinking 3 Times, Percobaan Blinking LED, Percobaan Turn On With Input From Terminal dan Percobaan LED With For.

4.1. Percobaan 1 LED ON

Pada sub-bab ini adalah percobaan LED On, akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

4.1.1. Highlight Materi

Menyalakan 2 lampu Led pada display kit dengan menggunakan Bahasa C pada Arduino menggunakan Pin 2 dan 3 pada Port digital.

4.1.2. Tujuan

Tujuan dalam modul ini yaitu agar praktikan dapat lebih memahami dalam penggunaan Port digital

4.1.3. Dasar Teori

- *Light Emitting Diode* atau biasa disebut LED sering kita jumpai dalam kehidupan sehari-hari, seperti lampu indikator standby di monitor, lampu indikator di charger, lampu indikator di dispenser, dan lain-lain.
- Pada bagian ini, akan dicoba bagaimana caranya agar LED dapat menyala melalui port digital yang anda gunakan

4.1.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit
- 4 buah kabel jumper

4.1.5. Langkah Kerja

a. On lampu led dengan menggunakan port Digital

- Siapkan semua alat dan bahan yang dibutuhkan
- Sambungkan kabel jumper dari pin 2 ke pin led 1
- Sambungkan kabel jumper dari pin 3 ke pin led 3
- Sambungkan kabel jumper dari 3,3 v ke + dan G ke ground
- Selesai pemasangan komponen, maka lanjutkan dengan menulis program menggunakan seperti dibawah ini :

```
const int led =2;
const int led1 =3;

void setup() {
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
}

void loop() {
  digitalWrite(led,HIGH);
  digitalWrite(led1,HIGH);
  delay(500);
}
```

- *Save as* program dengan nama led_on;
- Jalankan programnya, dengan upload program
- periksalah apakah LED menyala setelah dijalankan programnya. Namun, jika belum berhasil, silahkan di periksa pada bagian rangkaian maupun programnya.

4.2.Percobaan 2 LED OFF

Pada sub-bab ini adalah percobaan LED Off, akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

4.2.1. Highligt Materi

Mematikan 2 lampu Led pada display kit menggunakan Digital port 2 dan 3 pada Arduino dengan delay 500ms

4.2.2. Tujuan

Tujuan dalam modul ini yaitu agar praktikan dapat lebih memahami dalam penggunaan pin digital beserta pemberian delay (waktu). Dan menampilkan hasil print pada serial diarduino.

4.2.3. Dasar Teori

- *Light Emitting Diode* atau biasa disebut LED sering kita jumpai dalam kehidupan sehari-sehari, seperti lampu indikator standby di monitor, lampu indikator di charger, lampu indikator di dispenser, dan lain-lain.
- Pada bagian ini, akan dicoba bagaimana caranya agar LED dapat mati melalui digital Port yang anda gunakan

4.2.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit
- 4 buah kabel jumper

4.2.5. Langkah Kerja

a. Off lampu led dengan setmode broadcom

- Siapkan semua alat dan bahan yang dibutuhkan
- Sambungkan kabel jumper dari pin 2 ke pin led 1
- Sambungkan kabel jumper dari pin 3 ke pin led 3
- Sambungkan kabel jumper dari 3,3 v ke + dan G ke ground
- Selesai pemasangan komponen, maka lanjutkan dengan menulis program menggunakan seperti dibawah ini :

```

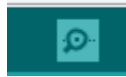
const int led =2;
const int led1 =3;

void setup() {
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  Serial.println("Led on");
  digitalWrite(led,HIGH);
  digitalWrite(led1,HIGH);
  delay(500);
  Serial.println("Led off");
  digitalWrite(led,LOW);
  digitalWrite(led1,LOW);
  delay(500);
}

```

- *Save as* program dengan nama led_onoff.py
- Jalankan programnya, dan on serial monitor yang ada pada Arduino atau klik ikon



- periksalah apakah LED menyala setelah dijalankan programnya. Namun, jika belum berhasil, silahkan di periksa pada bagian rangkaian maupun programnya.

4.3.Percobaan 3 LED Turn on With For

Pada sub-bab ini adalah Percobaan LED Turn on With For akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

4.3.1. Highligt Materi

Menyalakan 4 lampu Led dengan menggunakan logika for sederhana pada display kit dengan menggunakan port digital 2, 3, 4, dan 5 dengan delay (1000 ms).

4.3.2. Tujuan

Tujuan dalam modul ini yaitu agar praktikan diharapkan para praktikan dapat membuat program menyalakan 4 lampu led menggunakan logika for sederhana.

4.3.3. Dasar Teori

- *Light Emitting Diode* atau biasa disebut LED sering kita jumpai dalam kehidupan sehari-hari, seperti lampu indikator standby di monitor, lampu indikator di charger, lampu indikator di dispenser, dan lain-lain.
- Pada bagian ini, akan dicoba bagaimana caranya agar LED dapat mati melalui Port digital yang anda gunakan

4.3.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit
- 4 buah kabel jumper

4.3.5. Langkah Kerja

- Siapkan semua alat dan bahan yang dibutuhkan
- Sambungkan kabel jumper dari port 2 ke pin led 0, Sambungkan kabel jumper dari port 3 ke pin led 1, Sambungkan kabel jumper dari port 4 ke pin led 2, Sambungkan kabel jumper dari port 5 ke pin led 3
- Sambungkan kabel jumper dari 3,3 v ke + dan G ke ground
- Salin source code di bawah pada program Arduino lalu simpan Buat sebuah file dengan nama LED_blink_for.

```

int led [] ={2,3,4,5};

void setup() {
  for (int i=0; i< 4; i++){
    pinMode(led[i],OUTPUT);
    digitalWrite(led[i],LOW);
  }
  Serial.begin(9600);
}

void loop() {
  Serial.println("Led on");
  for(int j=0; j<4; j++){
    digitalWrite(led[j],HIGH);
  }
  delay (1000);
}

```

- Jalankan programnya dengan cara upload program
- periksalah apakah LED menyala setelah dijalankan programnya. Namun, jika belum berhasil, silahkan di periksa pada bagian rangkaian maupun programnya.

4.3.6. Uji Pemahaman

1. Menyalakan 4 lampu Led on off terus menerus dengan menggunakan logika for.dengan delay 500
2. Menyalakan 4 lampu led secara bergantian dengan menggunakan logika for
3. Menyalakan lampu zig zag dimulai dengan led 7 ke 0 pada display kit menggunakan for dan array.

MATERI – 5 SWITCH

Dalam bagian ini, akan dijelaskan penggunaan raspberry pi untuk melakukan sesuatu dengan SWITCH pada kit display. Materi yang dibahas terdiri dari percobaan Switch push button, percobaan Switch dan LED, percobaan Switch dan Buzzer dan percobaan Switch, LED serta Buzzer.

5.1.Percobaan 1 Switch Push Button

Pada sub-bab ini adalah Percobaan Switch Push Button, akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

5.1.1. Highligt Materi

Melakukan penekanan pada tombol switch yang menghasilkan output “button pressed” pada serial monitor. Dan menyalakn Led PO2 Pin 4.

5.1.2. Tujuan

Tujuan dalam modul ini yaitu agar praktikan diharapkan para praktikan dapat membuat program dengan menggunakan Switch Push Button sebagai input.

5.1.3. Dasar Teori

- *Switch Push Button* (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci).

5.1.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit
- 4 buah kabel jumper

5.1.5. Langkah Kerja

- Siapkan alat – alat dan bahan diatas
- Lalu Rangkai alat seperti di bawah :

Sambungkan 1 buah kabel jumper dari 3,3 V ke Pwr + sebagai power 3,3v.

Sambungkan 1 buah kabel jumper dari G ke Pwr G sebagai Ground.

Sambungkan 1 buah kabel jumper dari pin digital 5 ke pin P02.4 sebagai output

Sambungkan 1 buah kabel jumper dari pin digital 6 ke pin P02.0 sebagai input

- Salin source code di bawah pada program Python lalu simpan Buat sebuah file dengan name switch_push_button.

```
const int btn=6;
const int led1=5;

void setup() {
  pinMode(led1,OUTPUT);
  pinMode(btn,INPUT);
  Serial.begin(9600);
  //digitalWrite(btn,HIGH);
}

void loop() {
  if (digitalRead(btn)==0) {
    do{
    }
    while (digitalRead(btn)==0);
    digitalWrite(led1,digitalRead(led1));
    Serial.println("button pressed");
  }
}
```

- Run program yang telah dibuat lalu lihat hasil dan lihat pada serial monitor

5.2.Percobaan 2 Switch + Buzzer

Pada sub-bab ini adalah Percobaan Switch + Buzzer, akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

5.2.1. Highligt Materi

Menyalakan 1 buah Buzzer dengan menekan tombol switch yang menghasilkan output “button pressed and Buzzer ON” pada serial monitor arduino dengan menggunakan Port 6 dan 5 untuk menyalakan switch dan buzzer.

5.2.2. Tujuan

Pada percobaan ini diharapkan para praktikan dapat membuat program Menyalakan Buzzer dengan menggunakan Switch.

5.2.3. Dasar Teori

- *Switch Push Button* (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci).
- *Buzzer* adalah alat yang dapat mengeluarkan suara beep

5.2.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit
- 4 buah kabel jumper

5.2.5. Langkah Kerja

- Siapkan alat – alat dan bahan diatas
- Lalu Rangkai alat seperti di bawah :

Sambungkan 1 buah kabel jumper dari 3,3 V ke Pwr + sebagai power 3,3v.

Sambungkan 1 buah kabel jumper dari G ke Pwr G sebagai Ground.

Sambungkan 1 buah kabel jumper dari pin digital 5 ke pin P02.4 sebagai output

Sambungkan 1 buah kabel jumper dari pin digital 6 ke pin P02.0 sebagai input

- Salin source code di bawah pada program Python lalu simpan Buat sebuah file dengan nama switch_buzzer

```
const int btn=6;
const int buzzer=5;

void setup() {
  pinMode(buzzer,OUTPUT);
  pinMode(btn,INPUT);
  Serial.begin(9600);
  //digitalWrite(btn,HIGH);
}

void loop() {
  if (digitalRead(btn)==0){
    do{
    }
    while (digitalRead(btn)==0);
    digitalWrite(buzzer,!digitalRead(buzzer)); |
    Serial.println("button pressed");
    delay(500);
  }
}
```

- Run program yang telah dibuat lalu lihat hasil pada serial monitor dan kit display.

5.2.6. Uji Pemahaman

- Membuat Bunyi 4 kali pada buzzer dengan selang waktu 2 detik setelah menekan button pada switch 1 kali.

5.3.Percobaan 3 Switch + LED + Buzzer

Pada sub-bab ini adalah Percobaan Switch + LED + Buzzer, akan dibahas highlight percobaan yang akan dilakukan, tujuan dari percobaan, dasar teori, alat dan bahan dan langkah kerja.

5.3.1. Highlight Materi

Menyalakan 1 buah Buzzer dan led berjalan dari kiri (0-3) dengan menekan tombol switch yang menghasilkan output “button pressed and Buzzer ON and LED On” pada serial monitor.

5.3.2. Tujuan

Pada percobaan ini diharapkan para praktikan dapat membuat program menyalakan lampu led dan buzzer dengan menekan switch menggunakan raspberry.

5.3.3. Dasar Teori

- *Light Emitting Diode (LED)* adalah komponen elektronika yang bisa memancarkan cahaya monokromatik ketika diberikan suatu tegangan maju. **LED** masih termasuk dalam keluarga Dioda. **LED** terdiri dari sebuah chip dari bahan semikonduktor yang diisi penuh, atau di-dop, dengan ketidakmurnian untuk menciptakan sebuah struktur
- *Switch* button adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci)
- *Buzzer* adalah alat yang dapat mengeluarkan suara beep

5.3.4. Alat dan Bahan

Alat dan bahan yang digunakan adalah :

- Arduino
- Display Kit

- 4 buah kabel jumper

5.3.5. Langkah Kerja

- Siapkan alat – alat dan bahan diatas
- Lalu Rangkai alat seperti di bawah :

Sambungkan 1 buah kabel jumper dari 3,3 V ke Pwr + sebagai power 3,3v.

Sambungkan 1 buah kabel jumper dari G ke Pwr G sebagai Ground.

Sambungkan 1 buah kabel jumper dari pin digital 5 ke buzzer sebagai output

Sambungkan 1 buah kabel jumper dari pin digital 6 ke pin P02.0 sebagai input

Sambungkan 4 buah kabel jumper dari pin digital 2,3,4,7 ke pin P01 sebagai Output

Sambungkan 1 buah kabel jumper dari pin digital 8 ke pin P02.4 sebagai input

- Salin source code di bawah pada program Python lalu simpan Buat sebuah file dengan nama switch_led_buzzer.

```
int a[]={2,3,4,7};
const int led4=8;
const int btn=6;
const int buzzer=5;

void setup() {
  for (int i=0; i<4; i++){
    pinMode(a[i],OUTPUT);
    digitalWrite(a,LOW);
  }
  pinMode(led4,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(btn,INPUT);
  Serial.begin(9600);
  digitalWrite(btn,HIGH);
}

void loop() {
  if (digitalRead(btn)==0){
    do{
    }
    while (digitalRead(btn)==0);
    digitalWrite(led,!digitalRead(btn));
  }
  kiri();
}

void kiri(){
  if (digitalRead(led4)==1){
    digitalWrite(buzzer,HIGH);
    for (int j=0; j<4; j++){
      digitalWrite(a[j],HIGH);
      delay(100);
      digitalWrite(a[j],LOW);
      delay(100);
    }
    digitalWrite(buzzer,LOW);
  }
}
```


- Run program yang telah dibuat lalu lihat hasil pada serial monitor dan kit display.

Contoh: python switch_led_buzzer.py

5.3.6. Uji Pemahaman

Buatlah program dengan skema seperti berikut ini :

1. Terdapat 3 buah LED (merah,kuning,hijau), ketika menekan switch pertama untuk membuat ketiga LED tersebut menyala secara bergantian (*blinking*), switch kedua untuk mengatur durasi/delay menyala dari *blinking* . Semakin sering switch kedua ditekan maka akan semakin cepat *blinking*-nya.
2. Terdapat 4 switch, setiap switch berfungsi untuk membuat buzzer berbunyi dengan nada yang berbeda, gunakan fungsi `tone()`.