

BAB II

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang cuci tangan otomatis sudah pernah dilakukan oleh beberapa peneliti. Beberapa ringkasan *Studi Literatur* digunakan untuk mengetahui sejauh mana penelitian tersebut sudah dilakukan.

1.(Pilarogo, 2018) Rancang Bangun Sistem Alat Pencuci Dan Pengering Tangan Otomatis Menggunakan Mikrokontroler At89s52. Proses pencucian tangan berdasarkan pendeteksian gerakan tangan oleh Sensor PIR (Passive Infra Red) yang sudah berbentuk modul yang merupakan input dari mikrokontroler. Jarak maksimal dari obyek yang dapat dideteksi oleh sensor PIR adalah 5 m. Debit air yang keluar diatur oleh pengaturan waktu (setting timer) yang terdapat pada mikrokontroler. Mikrokontroler yang digunakan pada alat ini adalah AT89S52. Di dalam mikrokontroller AT89S52 terdapat 32 jalur I/O yang berguna sebagai keluaran dari program yang telah dimasukkan ke dalamnya. Pada alat ini menggunakan motor DC sebagai penggerak pompa sabun, motor AC sebagai penggerak pompa air, dan filamen pemanas sebagai penghasil udara panas untuk pengering tangan.

1.(Guminsar Rico Martogi Manullang, 2019) Rancang Bangun Dan Implementasi Sistem Pencuci Tangan (Hand Washer) Dan Pengering Tangan (Hand Dryer) Otomati. Alat pencuci dan pengering tangan otomais ini merupakan bentuk aplikasi dari sistem kontrol yang menggunakan mikrokontroler sebagai pengendali utamanya, mikrokontroler yang digunakan pada alat ini yaitu mikrokontroler. Alat ini akan melakukan proses pencucian tangan secara otomatis dengan memakai sensor Passive Infra Red (PIR) untuk mendeteksi ada atau tidak adanya tangan user. Debit air yang keluar dari kran tersebut tergantung dengan delay waktu yang dimasukan pada program dimikrokontroler. Setelah proses pencucian

tangan selesai maka proses selanjutnya yaitu masuk ke proses pengeringan tangan yang dilakukan oleh Dryer secara otomatis setelah delay waktu pencucian tangan selesai.

1.(Wulandari, 2018) Sistem Otomatisasi Kran Pencuci Tangan Hasil pemrosesan dari data sensor infra merah ditampilkan pada display LCD dalam bentuk tulisan "KRAN PENCUCI TANGAN OTOMATIS". Pada saat air mengalir maka dalam LCD tertulis instruksi "CUCILAH TANGAN" dan pada saat kipas menyala maka dalam LCD tertulis instruksi "KERINGKAN TANGAN"

1.(Guminsar Rico Martogi Manullang, 2019) Rancang Bangun Alat Pencuci Tangan Dan Pengering Tangan Otomatis Dengan Human Modul Interface (Hmi) Menggunakan T Ft Adafruit 2.8" Berbasis Arduino 2560. Alat pencuci tangan dan pengering tangan otomatis ini dilengkapi menggunakan Human Module Interface (HMI) menggunakan TFT 2.8" adafruit untuk dapat memilih salah satu dari tiga mode operasi kerja yaitu mode otomatis, mode manual dan mode sanitizer. Alat ini dapat bekerja ketika user memilih salah satu mode operasi pada layar TFT dan sensor IR (infrared) mendeteksi adanya tangan. Mode otomatis dirancang bekerja secara otomatis untuk mengalirkan air, mengeluarkan sabun dan mengeringkan tangan dalam satu rangkaian kerja berdasarkan waktu yang telah ditetapkan pada source code. Mode manual alat bekerja secara terpisah dalam mengalirkan air, mengeluarkan sabun cair dan mengeringkan tangan tanpa dibatasi waktu. Mode sanitizer bekerja dengan mengalirkan sanitizer secara otomatis ketika sensor IR mendeteksi tangan, serupa dengan mode manual pada mode sanitizer bekerja tanpa dibatasi waktu sehingga pengguna dapat menyesuaikan banyaknya penggunaan sanitizer untuk mencuci tangan.

2.2 Perangkat Keras Yang Digunakan

2.2.1 Sensor Ultrasonik HC-SR04



Gambar 2.1 Ultrasonik HC-SR04

(Sumber <https://www.elektronka.com>,2015)

Sensor HC-SR04 adalah sensor pengukur jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini mirip dengan radar ultrasonik. Gelombang ultrasonik di pancarkan kemudian di terima balik oleh receiver ultrasonik. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor ini cocok untuk aplikasi elektronik yang memerlukan deteksi jarak termasuk untuk sensor pada robot. Sensor HC-SR04 adalah versi low cost dari sensor ultrasonic PING buatan parallax. Perbedaannya terletak pada pin yang digunakan. HC-SR04 menggunakan 4 pin sedangkan PING buatan parallax menggunakan 3 pin. Pada Sensor HC-SR04 pin trigger dan output diletakkan terpisah. Sedangkan jika menggunakan PING dari Parallax pin trigger dan output telah diset default menjadi satu jalur. Tidak ada perbedaan signifikan dalam pengimplementasiannya. Jangkauan karak sensor lebih jauh dari PING buatan parllax, dimana jika ping buatan parllax hanya mempunyai jarak jangkauan maksimal 350 cm sedangkan sensor HC-SR04 mempunyai kisaran jangkauan maksimal 400-500cm.

1. Jangkauan deteksi: 2cm sampai kisaran 400 -500cm
2. Sudut deteksi terbaik adalah 15 derajat
3. Tegangan kerja 5V DC
4. Resolusi 1cm
5. Frekuensi Ultrasonik 40 kHz
6. Dapat dihubungkan langsung ke kaki mikrokontroler

2.2.2 Motor Servo

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem closed *feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Contoh motor servo, teori motor servo, definisi motor servo, bentuk motor servo, dasar teori motor servo, pengertian motor servo, analisa motor servo. Motor servo disusun dari sebuah motor DC, gearbox, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo.



Gambar 2.2 Motor Servo
(ROBOTIKA, Endra Pitowarno)

2.2.3 Modul Arduino Nano

Arduino nano dapat berupa sebuah mikroprosesor single board yang bersifat open source yang dirancang untuk memudahkan sebuah aplikasi pengguna elektronik dalam merancang sebuah alat, arduino ini bertujuan agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan, Berikut ini gambar arduino dari tampak depan dan belakang, disebut sebagai papan pengembangan karena board ini memang berfungsi sebagai arena prototype sirkuit mikrokontroller. dengan menggunakan papan pengembangan akan lebih mudah dalam merangkai rangkaian elektronika mikrokontroller ATmega328 (Slamet Haryono, 2016). Mengapa prototype ini menggunakan Arduino nano? Apa bedanya dengan arduino Nano? terdapat perbedaan pada sebuah tegangan input pada kaki (VIN) yang dilengkapi dengan

Jack (DC) dan sedangkan arduino nano tidak, karena arduino nano dapat menggunakan catu daya langsung dari *mini-USB port* dari luar. Adapun kelebihan dari arduino nano tidak perlu lagi menggunakan kabel USB ASP tapi cukup dengan menggunakan kabel USB Mini. Adapun system kerja dari arduino nano ini adalah dengan menggunakan pin analog di papan arduino yang menggunakan besaran tegangan sebesar 5V. Gambar 2.1 arduino nano atmega328.



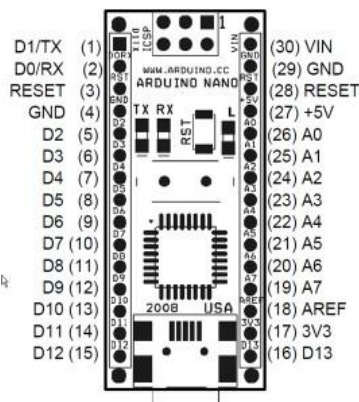
Gambar 2.3. Arduino Nano Atmega328

(Sumber <https://www.arduino.com>,2016)

2.2.3.1 Blog Arduino Nano

Dengan mengambil sebuah contoh blok diagram arduino nano, dapat dijelaskan bagian bagian kaki pin arduino nano sebagai berikut. Dapat dilihat seperti gambar 2.2.

Gambar 2.4. Blok Diagram Arduino Nano



(Sumber <https://www.arduino.com>,2016)

1. **Serial** 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin yang sesuai dari chip FTDI USB-to-TTL Serial.
2. **External Interrupt** (Interupsi Eksternal): Pin 2 dan pin 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
3. **PWM** : Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi `analogWrite()`.
4. **SPI** : Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI. Sebenarnya komunikasi SPI ini tersedia pada hardware, tapi untuk saat belum didukung dalam bahasa Arduino.
5. **LED** : Pin 13. Tersedia secara built-in pada papan Arduino Nano. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala, dan ketika pin diset bernilai LOW, maka LED padam.
6. **I2C** : Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C (TWI) menggunakan perpustakaan `Wire`.
7. **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
8. **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino

2.3 Perangkat Lunak Yang Digunakan

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

2.3.1 *Software* Mikrokontroller Arduino

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

1. Editor Program

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

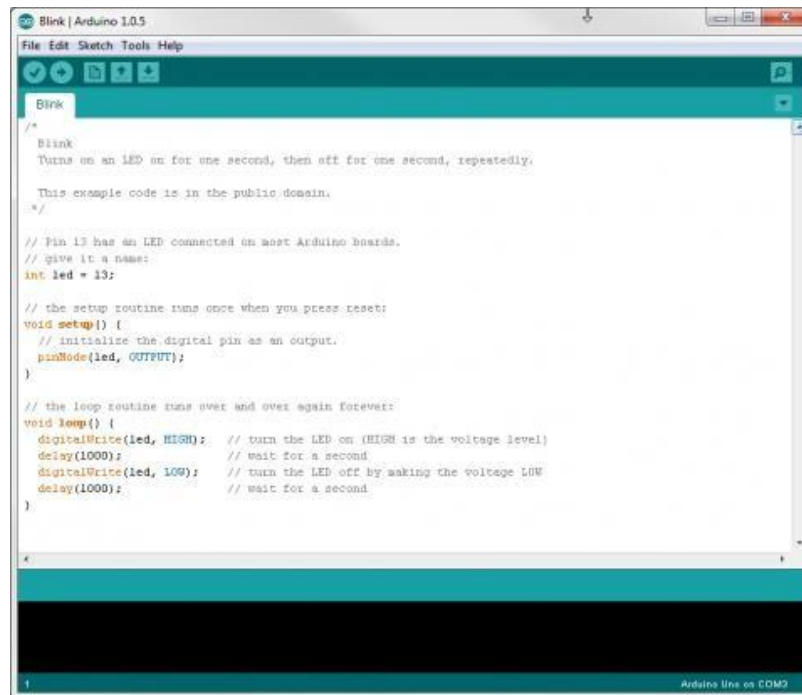
2. *Compiler*

Berfungsi untuk kompilasi *sketch* tanpa unggah ke *board* bisa dipakai untuk pengecekan kesalahan kode *sintaks sketch*. Sebuah modul yang mengubah kode program menjadi kode *biner* bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

3. *Uploader*

Berfungsi untuk mengunggah hasil kompilasi *sketch* ke *board* target. Pesan *error* akan terlihat jika *board* belum terpasang atau alamat *port* COM belum terkonfigurasi dengan benar. Sebuah modul yang memuat kode *biner* dari komputer ke dalam *memory* didalam papan *arduino*.(Sumber: B.Gustomo, 2015).

2.3.1.1 Program Arduino Ide

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and running. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

The status bar at the bottom right indicates "Arduino Uno on COM3".

Gambar 2.5. Tampilan Program Arduino Nano

Kode Program *Arduino* biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau *sketch* yang sudah selesai ditulis di *Arduino IDE* bisa langsung *dcompile* dan *diupload* ke *Arduino Board*. Secara sederhana, *sketch* dalam *Arduino* dikelompokkan menjadi 3 blok (lihat gambar di atas):

1. Header
2. Setup
3. Loop

2.3.1.2 Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan *library* dan pendefinisian *variable*. *Code* dalam blok ini dijalankan hanya sekali pada waktu *compile*. Di bawah ini contoh *code* untuk mendeklarasikan *variable led* (integer) dan sekaligus di isi dengan angka 13

```
int led = 13;
```


2.3.1.3 Setup

Di sinilah awal program *Arduino* berjalan, yaitu di saat awal, atau ketika *power on Arduino board*. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai *input* atau *output*, menggunakan perintah *pinMode*. Inisialisasi *variable* juga bisa dilakukan di blok ini

```
// the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }
```

OUTPUT adalah suatu makro yang sudah didefinisikan *Arduino* yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);`

Suatu pin bisa difungsikan sebagai *OUTPUT* atau *INPUT*. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai *INPUT*, pin tersebut memiliki *impedance* yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

2.3.1.4 Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol *power Arduino* di matikan. Di sinilah fungsi utama program *Arduino* kita berada.

```
void loop() {  
  
digitalWrite(led, HIGH); // nyalakan LED delay(1000); // tunggu 1000 milidetik  
digitalWrite(led, LOW); // matikan LED delay(1000); // tunggu 1000 milidetik }  

```

Perintah *digitalWrite*(pinNumber,nilai) akan memerintahkan *arduino* untuk menyalakan atau mematikan tegangan di *pinNumber* tergantung nilainya. Jadi perintah di atas `digitalWrite(led,HIGH)` akan membuat pin nomor 13 (karena di header dideklarasikan led = 13) memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai *digitalWrite* yaitu *HIGH* atau *LOW* yang sebetulnya adalah nilai integer 1 atau 0. Kalau sudah dibuat program di atas, selanjutnya kita ambil kabel USB yang diikutsertakan pada saat membeli *Arduino*, pasangkan ke

komputer dan *board arduino*, dan *upload* programnya. Lampu LED yg ada di *Arduino* board kita akan kelap-kelip. Sekedar informasi, sebuah LED telah disediakan di *board Arduino Nano* dan disambungkan ke pin 13.

Selain blok *setup()* dan *loop()* di atas kita bisa mendefinisikan sendiri blok fungsi sesuai kebutuhan. Kita akan jumpai nanti pada saat pembahasan proyek. (Sumber: Septa Ajjie, 2016).

