

BAB II

TINJAUAN PUSTAKA

2.1 Studi Literatur

Sejumlah penelitian sebelumnya telah memberikan kontribusi signifikan dalam pengembangan sistem dan aplikasi waktu shalat. Salah satu area yang telah dieksplorasi adalah pengembangan aplikasi penunjuk waktu shalat otomatis, yang berfokus pada perancangan perangkat keras yang dilengkapi dengan pengingat suara atau visual. Teknologi penampil dan pengingat waktu shalat digital telah dikembangkan untuk menekankan pentingnya perangkat keras dalam menyampaikan informasi waktu shalat kepada pengguna (Hidayat Kurniawan et al., 2021). Jenis penelitian ini memberikan dasar pemahaman tentang bagaimana sistem waktu shalat digital dapat diimplementasikan dalam bentuk perangkat keras yang berdiri sendiri.

Penelitian lainnya yang menyoroti sistem penjadwalan shalat digital dengan menggunakan running text sebagai media informasi. Penelitian ini menekankan visualisasi data dan bagaimana teknologi seperti running text dapat menyampaikan jadwal shalat di ruang publik. Sistem pengatur running text berbasis android telah dikembangkan menggunakan modul bluetooth hc-05 dan arduino uno, yang memungkinkan pengguna memperbarui informasi secara nirkabel tanpa memerlukan perangkat komputer. Sistem ini diuji dengan berbagai panjang karakter dan menunjukkan hasil yang efisien dengan waktu tunda yang relatif singkat (Wijaya et al., 2023).

Akurasi perhitungan waktu shalat juga menjadi perhatian penting dalam berbagai studi. Perancangan sistem penjadwalan waktu shalat yang akurat telah dilakukan dengan metode *Julian Date* dan scanning untuk memastikan keandalan informasi yang disampaikan (Suroso et al., 2022). Pemilihan metode perhitungan yang tepat menjadi elemen krusial dalam sistem ini.

Penelitian mengenai akurasi jam digital di masjid juga memberikan kontribusi penting dalam konteks sosial dan teknis. Pendekatan partisipatif digunakan untuk membangun kesadaran masyarakat terhadap pentingnya akurasi jadwal waktu shalat pada jam digital di masjid, serta pentingnya pelibatan masyarakat dalam penggunaan teknologi di lingkungan keagamaan (Rohmah, 2024).

Pengembangan aplikasi mobile, telah dirancang aplikasi android yang menyediakan informasi arah kiblat dan waktu shalat, serta memanfaatkan teknologi GPS untuk menentukan lokasi pengguna secara akurat (Fadlil et al., 2021). Penelitian ini menunjukkan bagaimana aplikasi mobile mampu memberikan kemudahan akses informasi keagamaan.

Beberapa penelitian juga mengeksplorasi teknologi nirkabel untuk sistem waktu shalat digital. Perancangan display LED dot matrix yang dikendalikan melalui aplikasi android via Wi-Fi memperlihatkan potensi penggunaan koneksi nirkabel untuk pembaruan informasi (Nataprawira et al., 2020).

Salah satu penelitian yang mengimplementasikan geolocation pada sistem presensi mobile berbasis Android. Mereka memanfaatkan GPS internal perangkat untuk menentukan lokasi presensi yang valid, serta menggunakan REST API untuk pengelolaan data pengguna secara efisien dan ringan. Setiap permintaan dari aplikasi diproses secara stateless, memungkinkan transaksi cepat dan terpisah antar request. Mekanisme ini sangat cocok diadaptasi untuk sistem digital waktu shalat yang juga membutuhkan integrasi antara lokasi pengguna dan server penyedia informasi waktu ibadah. Dengan pendekatan ini, sistem dapat menyesuaikan jadwal berdasarkan lokasi dan tetap menyediakan jalur verifikasi dari sumber resmi melalui API (Dwi Kurniawan & Nur Laila, 2023).

Penelitian ini berbeda dari penelitian-penelitian sebelumnya yang terletak pada sisi tampilan digitalnya yang menggunakan seven segment. Fokus utama terletak pada perancangan aplikasi remote untuk mengontrol jam digital waktu shalat, yang

menyediakan fitur pengaturan jadwal shalat, waktu iqomah, buzzer, relay, dan lokasi. Selain itu, penelitian ini menggunakan MIT App Inventor sebagai platform pengembangan, memungkinkan proses desain aplikasi android yang visual dan intuitif. Oleh karena itu, penelitian ini memberikan kontribusi unik dalam pengembangan sistem waktu shalat digital yang fleksibel, mudah digunakan, dan dapat disesuaikan dengan kebutuhan pengguna.

2.2 Dasar Teori

2.2.1 Sistem Remote

Sistem remote adalah suatu mekanisme kendali jarak jauh yang memungkinkan pengguna untuk mengoperasikan perangkat lain tanpa kontak fisik langsung. Dalam konteks penelitian ini, sistem remote diterapkan melalui koneksi nirkabel bluetooth hc-05, yang memungkinkan aplikasi android mengirimkan data ke mikrokontroler untuk mengatur waktu shalat, buzzer, relay, dan pengaturan lainnya.

Penggunaan sistem remote berbasis aplikasi sangat relevan untuk perangkat jam digital masjid karena meningkatkan efisiensi dan kenyamanan pengelolaan waktu shalat tanpa harus menyentuh perangkat secara langsung. Komunikasi serial antara bluetooth hc-05 dan mikrokontroler menjadi elemen penting agar sistem berjalan dengan andal dan responsif.

Penggunaan MIT App Inventor dalam perancangan aplikasi remote berbasis bluetooth hc-05 terbukti efektif dalam mengendalikan perangkat secara nirkabel, seperti yang ditunjukkan oleh aplikasi pengendali robot melalui bluetooth yang dikembangkan oleh Top dan Gökbulut (TOP & GÖKBULUT, 2021).

2.2.2 Waktu Shalat

Shalat lima waktu merupakan bagian integral dari ibadah umat islam. Mengetahui waktu shalat dengan tepat adalah kebutuhan mendasar bagi umat muslim. Secara tradisional, waktu-waktu shalat ditentukan melalui pengamatan langsung terhadap

fenomena alam seperti pergerakan matahari. Namun, seiring dengan perkembangan teknologi, proses penentuan waktu shalat menjadi lebih akurat dan efisien dengan bantuan perangkat elektronik dan sistem informasi berbasis Lokasi (Suroso et al., 2022).

Tabel 2. 1 Contoh Waktu Shalat Berdasarkan Zona

Zona Waktu	Lokasi (Contoh Kota)	Subuh	Dzuhur	Ashar	Magrib	Isya
WIB	Jakarta (106.8° BT)	04:38	11:57	15:15	17:58	19:08
WIT	Makasar (119.4° BT)	04:56	12:07	15:26	18:09	19:19
WITA	Jayapura (140.7° BT)	04:24	11:42	14:51	17:24	18:32

Sumber: Kementerian Agama Republik Indonesia – Jadwal Salat Digital 2024.
Diakses dari: (<https://bimasislam.kemenag.go.id/jadwalshalat>)

Shalat wajib dalam Islam terdiri dari lima waktu utama, masing-masing memiliki ketentuan astronomis yang spesifik sebagai berikut:

1. Subuh: Dimulai ketika cahaya fajar pertama terlihat, saat matahari berada sekitar 20° di bawah ufuk.
2. Dzuhur: Dimulai saat matahari melewati titik tertinggi (kulminasi), yaitu ketika bayangan paling pendek.
3. Ashar: Dimulai ketika panjang bayangan suatu benda sama dengan panjang benda tersebut atau lebih, tergantung pada mazhab.
4. Maghrib: Dimulai saat matahari terbenam sepenuhnya di bawah ufuk (0°).
5. Isya: Dimulai saat langit benar-benar gelap, yaitu ketika matahari berada sekitar 18° di bawah ufuk.

2.2.3 Acuan Dalam Perhitungan Waktu Shalat

Acuan perhitungan waktu shalat menjadi dasar penting dalam perancangan sistem remote penentu waktu shalat digital. Terdapat beberapa metode yang digunakan dalam menentukan waktu shalat, masing-masing memiliki karakteristik dan keunggulan yang berbeda. Berikut ini adalah beberapa metode yang umum digunakan dalam perhitungan waktu shalat.

1. Metode pertama adalah hisab astronomi.

Metode ini memanfaatkan perhitungan posisi matahari yang mencakup deklinasi matahari dan hour angle, serta *Julian Date* sebagai dasar waktu astronomi. Seluruh data astronomi ini kemudian disesuaikan dengan koordinat geografis pengguna seperti *latitude*, *longitude*, dan *altitude*. Hasilnya dikonversi ke waktu lokal sesuai zona waktu (WIB, WITA, atau WIT), sehingga waktu shalat yang ditampilkan selalu akurat dan relevan. Metode hisab astronomi dianggap sangat sesuai untuk sistem digital karena dapat diproses dengan cepat, akurat, dan mendukung penyesuaian otomatis jika pengguna berpindah lokasi. Hal ini dijelaskan dalam penelitian yang melakukan simulasi perhitungan awal waktu shalat berbasis NOAA Solar Calculator dalam sistem digital (Taufikurrahman, 2023). Penelitian lain juga menunjukkan bahwa metode hisab astronomi dengan data ephemeris memberikan keakuratan tinggi dan cocok untuk integrasi dalam aplikasi android berbasis remote system (Emye Tegar Handhita, 2023).

2. Metode kedua adalah perhitungan manual menggunakan almanak atau jadwal tetap.

Metode ini biasanya disusun oleh lembaga-lembaga resmi seperti Kementerian Agama dan diterbitkan setiap tahun. Jadwal shalat yang dihasilkan sudah dihitung untuk satu tahun penuh, sehingga pengguna tinggal membacanya. Kelebihan metode ini adalah kemudahannya dalam penggunaan sehari-hari karena sudah tersedia dalam bentuk cetak atau digital. Namun, metode ini memiliki kelemahan yaitu sifatnya statis dan

tidak mampu menyesuaikan waktu shalat jika terjadi perpindahan lokasi yang signifikan dari lokasi awal data jadwal tersebut. Hal ini juga disebutkan oleh penelitian yang menyoroti bahwa data jadwal manual hanya akurat untuk lokasi asal dan tidak mendukung integrasi digital secara real-time (Akurasi Penentuan Jadwal Waktu Shalat Perspektif Ilmu Falak et al., 2022).

3. Metode ketiga adalah rukyat atau pengamatan langsung terhadap posisi matahari atau bulan.

Metode ini lebih banyak digunakan dalam penentuan awal bulan Hijriyah atau awal Ramadhan, dan jarang digunakan untuk menghitung waktu shalat harian. Rukyat memerlukan pengamatan visual langsung yang dilakukan oleh pengguna, sehingga tidak praktis jika diterapkan dalam sistem digital otomatis yang diharapkan dapat beroperasi setiap waktu secara mandiri. Penelitian menyebutkan bahwa meskipun rukyat memiliki nilai penting dalam konteks ibadah, untuk kebutuhan real-time sehari-hari, metode hisab astronomi jauh lebih praktis dan tepat guna (Ardi, 2021).

Dari ketiga metode di atas, sistem yang dirancang dalam penelitian ini memilih metode hisab astronomi karena akurasinya yang tinggi, kemudahan dalam pemrosesan data digital, dan kemampuannya menyesuaikan lokasi pengguna secara otomatis. Hal ini juga didukung oleh penelitian yang menunjukkan bahwa rumus trigonometri segitiga bola dalam hisab astronomi dapat diterapkan secara efisien dalam pemrograman mikrokontroler untuk perhitungan waktu shalat (Nur Eka Rahmawati et al., 2021). Dengan demikian, metode hisab astronomi menjadi landasan utama yang dipilih untuk mendukung sistem remote waktu shalat digital yang dinamis dan efisien.

2.2.4 Perhitungan Waktu Shalat Julian Date

Julian Date (JD) digunakan dalam astronomi untuk menghitung waktu secara kontinu sejak 1 Januari 4713 SM pukul 12:00 UT. Dalam perhitungan waktu

shalat, JD digunakan untuk menentukan parameter astronomi seperti deklinasi matahari, *equation of time* (ET), *altitude* matahari, transit, dan *hour angle* (HA). Berikut adalah langkah-langkah perhitungan waktu shalat dengan metode Julian Date:

1. Hitung *Julian Date* (JD)

Rumus:

$$JD = 1720994.5 + \text{INT}(365.25 \times Y) + \text{INT}(30.6001 \times (M + 1)) + B + D$$

Keterangan:

- a) **Y** = Tahun
- b) **M** = Bulan
- c) **D** = Tanggal
- d) **B** = Koreksi kalender (0 untuk kalender Gregorian setelah 15 Oktober 1582)
- e) **Langkah:**

Tanggal Masehi (D/M/Y) diubah ke JD untuk keperluan astronomi.

2. Hitung Nilai U (Jumlah Abad dari Epoch 2000)

Rumus:

$$U = (JD - 2451545) / 36525$$

Keterangan:

- a) **2451545** adalah JD pada 1 Januari 2000 pukul 12:00 UT.

3. Hitung Bujur Rata-Rata Matahari (L_0)

Rumus:

$$L_0 = 280.46607 + 36000.7698 \times U$$

Keterangan:

- a) **L_0** = Mean Solar Longitude dalam derajat.

4. Hitung *Equation of Time* (ET)

Rumus:

$$ET = [-(1789 + 237U) \sin L_o - (7146 - 62U) \cos L_o + (9934 - 14U) \sin (2L_o) - (29 + 5U) \cos (2L_o) + (74 + 10U) \sin (3L_o) + (320 - 4U) \cos (3L_o) - 212 \sin (4L_o)] / 1000$$

Keterangan:

- a) **ET** = Selisih antara waktu matahari rata-rata dengan waktu matahari sebenarnya (dalam menit).

5. Hitung Sudut Deklinasi Matahari (Δ)

Rumus:

$$\Delta = 0.37877 + 23.264 \sin(57.297 T - 79.547) + 0.3812 \sin(2 \times 57.297 T - 82.682) + 0.17132 \sin(3 \times 57.297 T - 59.722)$$

Keterangan:

- a) **T** = Hari ke- dari awal tahun (misalnya 25 Mei = hari ke-145).

6. Hitung Transit (Dzuhur)

Rumus:

$$\text{Transit} = 12 + Z - (B / 15) - (ET / 60)$$

Keterangan:

- a) **Z** = Zona waktu (contoh: WIB = +7)
b) **B** = Bujur lokasi (longitude)

7. Hitung Altitude Matahari

- a) Subuh & Isya

Rumus:

$$\text{Altitude} = -\theta$$

Keterangan:

- a. **θ** = Sudut matahari di bawah ufuk (18° – 20°)

b) Magrib

Rumus:

$$\text{Altitude} = -0.8333 - 0.0347 \sqrt{H}$$

Keterangan:

- a. **H** = Ketinggian lokasi (m).

c) Ashar

Rumus:

$$\text{Altitude} = \text{ARCCOT}(KA + \tan |\Delta - L|)$$

Keterangan:

- a. **KA** = 1 (Mazhab Syafi'i) atau 2 (Mazhab Hanafi).
 b. **L** = Lintang tempat (latitude).

Disini ada dua pendapat. Pendapat Mazhab Syafi'i menyatakan panjang bayangan benda saat ashar = tinggi benda + panjang bayangan saat dzuhur. Sementara pendapat Mazhab Hanafi menyatakan panjang bayangan benda saat ashar = dua kali tinggi benda + panjang bayangan saat dzuhur.

8. Hitung Hour Angle (HA)

Rumus:

$$\cos(HA) = [\sin(\text{Altitude}) - \sin(L) \times \sin(\Delta)] / [\cos(L) \times \cos(\Delta)]$$

Keterangan:

- a) **HA** = Sudut jam (jam matahari).
 b)

9. Hitung Waktu Shalat

Rumus:

$$\begin{aligned} \text{Subuh} &= \text{Transit} - (\text{HA_Subuh} / 15) \\ \text{Dzuhur} &= \text{Transit} \\ \text{Ashar} &= \text{Transit} + (\text{HA_Ashar} / 15) \\ \text{Maghrib} &= \text{Transit} + (\text{HA_Maghrib} / 15) \\ \text{Isya} &= \text{Transit} + (\text{HA_Isya} / 15) \end{aligned}$$

Keterangan:

- a) Konversi **HA/15** digunakan karena bumi berotasi 15° per jam.

10. Contoh Perhitungan Waktu Shalat (Bandar Lampung, 25 Mei 2025)

Data Input:

Lintang (L) = -5.3769°

Bujur (B) = 105.24949°

Zona Waktu (Z) = +7 (WIB)

Ketinggian (H) = 203.7

- a) Hitung Julian Day (JD)

Rumus:

$$\begin{aligned} \text{JD} &= 1720994.5 + \text{INT}(365.25 \times Y) + \text{INT}(30.6001 \times (M+1)) + B + \text{DJD} = \\ &1720994.5 + \text{INT}(365.25 \times Y) + \text{INT}(30.6001 \times (M+1)) + B \\ &+ \text{DJD} = 1720994.5 + \text{INT}(365.25 \times Y) + \text{INT}(30.6001 \times (M+1)) + B + D \end{aligned}$$

Hasil:

Dengan Y = 2025, M = 5, D = 25 $\rightarrow$ **JD = 2460822.5**

- b) Hitung U, L_o , ET, dan Deklinasi (Δ)

$$U = (\text{JD} - 2451545) / 36525 = 0.25399589$$

$$L_o = 280.46607 + 36000.7698 \times U = 64.49^\circ. \text{ (setelah mod } 360^\circ)$$

$$\text{ET} \approx -7.038 \text{ menit (dihitung dari } L_o \text{ dan } U)$$

$$\Delta \approx +15.90^\circ \text{ (dihitung dari posisi matahari pada hari ke-145)}$$

- c) Hitung Transit (Dzuhur)

Rumus:

$$\text{Transit} = 12 + Z - B/15 - \text{ET}/60$$

Hasil:

$$\text{Transit} = 12 + 7 - (105.24949/15) - (-7.038/60) = 12:06 \text{ WIB}$$

- d) Hitung Hour Angle (HA)

Hour Angle adalah jarak waktu dari Dzuhur.

- a. HA Subuh (dengan $h = -20^\circ$):
 $HA_{\text{Subuh}} \approx 7.28$ jam atau 7 jam 17 menit.
- b. HA Ashar:
 $HA_{\text{Ashar}} \approx 3.37$ jam atau 3 jam 22 menit.
- c. HA Magrib (dengan $h = -0.8333^\circ$):
 $HA_{\text{Magrib}} \approx 5.96$ jam atau 5 jam 57 menit
- d. HA Isya (dengan $h = -18^\circ$):
 $HA_{\text{Isya}} \approx 7.15$ jam atau 7 jam 9 menit

e) Hitung Waktu Shalat

Rumus:

$$\text{Subuh} = 12:06 - 07:17 = 04:49 \text{ WIB}$$

$$\text{Dzuhur} = 12:06 \text{ WIB}$$

$$\text{Ashar} = 12:06 + 03:22 = 15:28 \text{ WIB}$$

$$\text{Maghrib} = 12:06 + 05:57 = 18:03 \text{ WIB}$$

$$\text{Isya} = 12:06 + 07:09 = 19:15 \text{ WIB}$$

Tabel 2. 2 Hasil Contoh Penentu Waktu Shalat Kota Bandar Lampung

Tanggal	Longitude, Latitude, Altitude (m)	JD	U	Lo (°)	ET (menit)	Transit (WIB)	HA Subuh (jam)	HA Ashar (jam)	HA Magrib (jam)	HA Isya (jam)	Waktu Shalat (WIB)
25 Mei 2025	-5.3769°, 105.24949, 203.7 m	246089 9.5	0.256 11225	105.25 188°	-5.097 Menit	12:06	7.28				Subuh (04:49)
25 Mei 2025	-5.3769°, 105.24949, 203.7 m	246089 9.5	0.256 11225	105.25 188°	-5.097 Menit	12:06					Dzuhur (12:06)
25 Mei 2025	-5.3769°, 105.24949, 203.7 m	246089 9.5	0.256 11225	105.25 188°	-5.097 Menit	12:06		3.37			Ashar (15:28)
25 Mei 2025	-5.3769°, 105.24949, 203.7 m	246089 9.5	0.256 11225	105.25 188°	-5.097 Menit	12:06			5.96		Magrib (18:05)
25 Mei 2025	-5.3769°, 105.24949, 203.7 m	246089 9.5	0.256 11225	105.25 188°	-5.097 Menit	12:06				7.15	Isya (19:15)

Perhitungan ini menggunakan rumus astronomi internasional (*Jean Meeus*) untuk menghitung posisi matahari, namun sudut ketinggian matahari (altitude) telah disesuaikan dengan kriteria Badan Hisab Rukyat Kementerian Agama RI (Kemenag RI) dan rekomendasi LAPAN. Misalnya, sudut matahari untuk Subuh adalah -20° dan untuk Isya adalah -18° sesuai standar Kemenag RI.

Rumus *Julian Date* ini digunakan sebagai dasar konversi waktu dari kalender masehi ke sistem waktu astronomis universal. Perhitungan ini digunakan dalam banyak sistem jadwal shalat digital karena memberikan presisi tinggi berdasarkan pergerakan benda langit (Suroso et al., 2022).

2.2.5 Selisih dalam Perhitungan Waktu Shalat

Selisih waktu shalat adalah perbedaan antara hasil perhitungan hisab dengan jadwal resmi Kementerian Agama RI (Kemenag) atau lembaga falak lainnya. Selisih ini wajar terjadi karena faktor perbedaan metode perhitungan, data ephemeris, refraksi atmosfer, ketinggian lokasi, serta pembulatan waktu. Menurut Kemenag RI dalam pedoman penentuan jadwal waktu shalat sepanjang masa dan ephemeris hisab rukyat, selisih hingga $\pm 1-2$ menit masih berada dalam batas toleransi (iḥtiyāt) dan tidak memengaruhi keabsahan ibadah.

Faktor Penyebab Selisih:

- 1) Perbedaan sudut matahari (h_0): Menurut pedoman hisab Kementerian Agama RI, waktu Subuh dihitung berdasarkan sudut depresiasi matahari sebesar -20° , sedangkan waktu Isya menggunakan -18° . Metode ini berlaku untuk penentuan jadwal shalat dalam sistem nasional. Penelitian dan kajian hisab modern menunjukkan bahwa variasi sudut ini misalnya antara -17° hingga -20° dapat menimbulkan deviasi waktu shalat sebesar $\pm 1-2$ menit, yang masih dianggap berada dalam batas toleransi (jawāz) falakiah.
- 2) Ketinggian lokasi (altitude): Wilayah di dataran tinggi mengalami terbit dan terbenam matahari lebih cepat.

- 3) Refraksi atmosfer: Pembelokan cahaya matahari oleh atmosfer mempengaruhi posisi semu matahari.
- 4) Metode hisab: Penggunaan algoritma Julian Date atau data ephemeris (NOAA/NASA) menimbulkan deviasi kecil.
- 5) Pembulatan waktu: Jadwal resmi dibulatkan ke menit terdekat, sehingga detik dapat menambah atau mengurangi ± 1 menit.

Tabel 2. 3 Selisih Dalam Perhitungan Waktu Shalat

Selisih Waktu	Keterangan
0–1 menit	Sangat akurat, sesuai dengan jadwal resmi
1–2 menit	Dalam toleransi (iḥtiyāt)
2–3 menit	Masih dapat diterima, butuh koreksi minor
>3 menit	Perlu peninjauan ulang perhitungan

Deviasi antara jadwal hasil hisab mandiri dengan jadwal resmi Kementerian Agama umumnya berada dalam rentang 1–2 menit. Selisih ini masih dapat diterima dalam praktik falakiah (jawāz), sementara selisih lebih dari 3 menit sebaiknya dikaji ulang untuk menghindari kesalahan penentuan waktu shalat.

2.2.6 Application Programming Interface (API)

Application Programming Interface (API) merupakan antarmuka perangkat lunak yang memungkinkan suatu aplikasi untuk berkomunikasi dengan sistem atau layanan eksternal. Dalam pengembangan aplikasi android, API sangat berperan dalam menghubungkan aplikasi klien dengan server untuk pertukaran data secara real-time menggunakan protokol seperti HTTP dan format data standar seperti JSON.

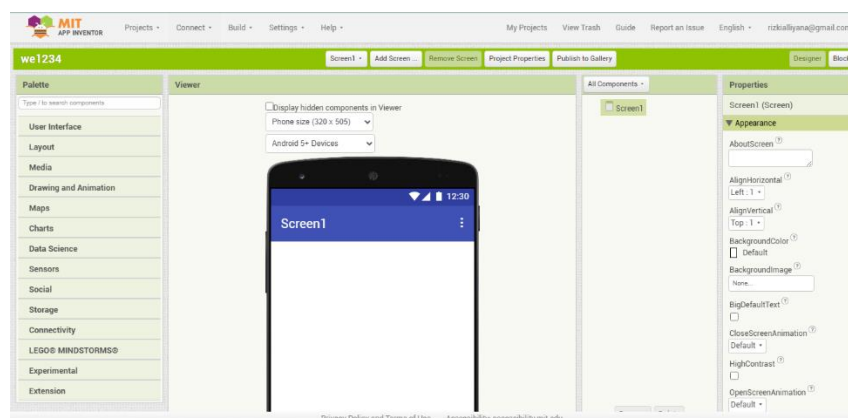
API memberikan kemudahan dalam manajemen data, karena proses pengolahan data dilakukan di sisi server, sementara aplikasi hanya menampilkan data yang

diperoleh. Hal ini menjadikan aplikasi lebih ringan, efisien, dan mudah diperbarui tanpa perlu menyematkan keseluruhan logika bisnis di sisi pengguna. Dalam penelitian ini, API digunakan untuk melakukan pengecekan jadwal waktu shalat berdasarkan wilayah administratif (provinsi dan kabupaten) melalui layanan dari (<https://api.myquran.com/v2>). Jadwal dari API ditampilkan di aplikasi android sebagai pembanding terhadap hasil hisab lokal berbasis data GPS internal.

Penggunaan API jenis REST (*Representational State Transfer*) sangat mendukung fleksibilitas sistem karena bersifat stateless dan dapat diakses lintas platform, sehingga mempermudah pengembangan aplikasi android yang terhubung dengan layanan web server (Sulistyo et al., 2024)

2.2.7 MIT App Inventor

MIT App Inventor merupakan sebuah platform pemrograman berbasis web yang dirancang untuk mempermudah pembuatan aplikasi android. Sistem ini menggunakan tampilan antarmuka berbasis blok sehingga memudahkan pengguna, khususnya pemula, dalam memahami konsep dasar pemrograman. Dengan metode ini, proses belajar menjadi lebih intuitif dan menyenangkan, bahkan bagi siswa yang tidak memiliki latar belakang teknis sekalipun (Wahyu Kurniawan et al., 2024).



Gambar 2. 1 MIT App Inventor
(<https://ai2.appinventor.mit.edu/#4606808550342656>)

MIT App Inventor memiliki lingkungan kerja yang terdiri dari beberapa komponen utama yang mendukung proses pengembangan aplikasi. Komponen-komponen ini terbagi menjadi tiga bagian utama, yaitu Desainer, Block Editor, dan Emulator.

- 1) Komponen Desainer merupakan bagian dari App Inventor yang berjalan di dalam browser. Fungsinya adalah untuk memilih dan menempatkan komponen visual yang dibutuhkan serta mengatur properti dari setiap komponen tersebut. Pada bagian ini terdapat lima elemen penting, yaitu Palette, Viewer, Component, Media, dan Properties.
 - a. Palette berisi daftar komponen yang dapat digunakan dalam pengembangan aplikasi, seperti tombol, label, sensor, serta konektivitas seperti Bluetooth dan Web.
 - b. Viewer berfungsi sebagai area kerja tempat pengguna menyusun tampilan antarmuka aplikasi dengan menyeret dan meletakkan komponen dari Palette.
 - c. Component menampilkan daftar komponen yang telah digunakan dalam proyek saat ini.
 - d. Media digunakan untuk menambahkan file media seperti gambar, audio, atau video yang diperlukan dalam proyek.
 - e. Properties digunakan untuk mengatur atribut atau karakteristik dari masing-masing komponen, seperti ukuran (width, height), nama, warna, dan parameter lainnya.
- 2) Block Editor adalah bagian dari App Inventor yang digunakan untuk menyusun logika dan perilaku dari aplikasi yang sedang dibuat. Pada editor ini, pengguna dapat membuat alur program melalui blok-blok visual yang saling terhubung, tanpa menulis kode secara manual. Setiap komponen dari Desainer akan memiliki blok-blok tersendiri yang dapat diatur sesuai kebutuhan. Block Editor berbasis JavaScript dan bekerja langsung di browser

modern, sehingga tidak lagi memerlukan instalasi JDK dan JRE seperti versi awal App Inventor.

- 3) Emulator merupakan alat bantu yang digunakan untuk menjalankan dan menguji aplikasi yang telah dibuat, terutama bagi pengguna yang belum memiliki perangkat android. Dengan emulator ini, pengembang dapat mensimulasikan jalannya aplikasi secara langsung di komputer, sehingga dapat mengetahui apakah desain dan logika aplikasi sudah berjalan sesuai dengan harapan sebelum diunggah ke perangkat sesungguhnya.

Kelebihan MIT App Inventor:

- 1) Antarmuka visual yang Intuitif
Pengguna tidak perlu menulis kode secara manual, sehingga sangat cocok untuk pemula.
- 2) Mendukung koneksi ke perangkat eksternal
MIT App Inventor dapat berkomunikasi dengan perangkat arduino melalui bluetooth, wifi, dan sensor lainnya.
- 3) Cepat dalam pengembangan
Aplikasi bisa dibuat dan diuji dengan cepat karena berbasis drag and drop dan dapat langsung diuji melalui companion app.
- 4) Open source dan gratis
Dapat diakses secara bebas tanpa biaya lisensi.
- 5) Komunitas yang aktif
Banyak dokumentasi, tutorial, dan forum yang mendukung pembelajaran.

Kekurangan MIT App Inventor:

- 1) Terbatas pada platform android
Tidak mendukung pengembangan untuk platform ios atau desktop secara langsung.
- 2) Kurang fleksibel untuk aplikasi kompleks

Tidak cocok untuk pengembangan aplikasi dengan fitur yang sangat kompleks atau yang memerlukan kontrol tingkat rendah.

3) Ukuran aplikasi yang besar

Aplikasi yang dihasilkan bisa lebih besar dari aplikasi sejenis yang dikembangkan dengan java atau kotlin.

4) Terbatas dalam desain UI/UX

Opsi penyesuaian antarmuka pengguna lebih sederhana dibandingkan pengembangan menggunakan android studio.

Proyek MIT App Inventor berupaya untuk mendemokratisasi pengembangan perangkat lunak dengan memberdayakan semua orang, terutama kaum muda, untuk beralih dari konsumsi teknologi ke penciptaan teknologi.

2.2.8 GPS (*Global Positioning System*) Internal Ponsel

GPS (*Global Positioning System*) yang tertanam dalam ponsel digunakan untuk menentukan lokasi geografis pengguna dengan cara menangkap sinyal dari minimal empat satelit navigasi. Dari hasil perhitungan jarak ke satelit-satelit tersebut, GPS dapat menentukan titik koordinat pengguna secara presisi. Titik koordinat ini terdiri dari tiga parameter utama, yaitu lintang (*latitude*), bujur (*longitude*), dan ketinggian (*altitude*).

- 1) Latitude (derajat lintang) adalah posisi utara atau selatan terhadap garis khatulistiwa, bernilai positif (+) untuk belahan bumi utara dan negatif (−) untuk belahan bumi selatan.
- 2) Longitude (derajat bujur) adalah posisi timur atau barat terhadap garis bujur nol yang melewati Greenwich, Inggris, bernilai positif (+) untuk wilayah timur dan negatif (−) untuk wilayah barat.
- 3) Altitude (meter di atas permukaan laut) adalah ketinggian lokasi yang dapat memengaruhi perhitungan waktu shalat, khususnya untuk Magrib dan Subuh, karena ketinggian berpengaruh terhadap sudut terbenam atau terbit matahari.



Gambar 2. 2 GPS

Penggunaan GPS internal mampu memberikan lokasi yang akurat dan mendukung fleksibilitas sistem pengingat shalat berbasis mobile (Fadlil et al., 2021).

Kelebihan Penggunaan GPS dalam Sistem:

1. Memberikan data posisi yang akurat tanpa tergantung jaringan internet.
2. Cocok untuk aplikasi berbasis lokasi seperti penentuan waktu shalat.
3. Dapat digunakan secara terus-menerus (*real-time tracking*).

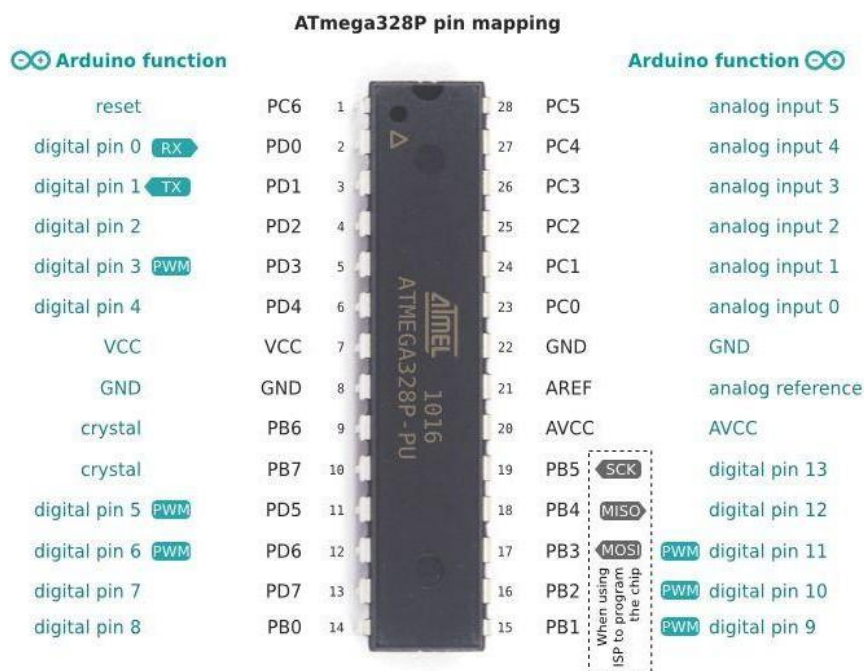
Kekurangan GPS:

1. Membutuhkan waktu untuk mendapatkan sinyal fix, terutama di dalam ruangan.
2. Konsumsi daya relatif lebih tinggi dibanding sensor pasif.
3. Akurasi dapat terganggu oleh kondisi cuaca atau hambatan fisik seperti gedung tinggi.

2.2.9 Mikrokontroler ATmega328P

Mikrokontroler adalah sebuah komputer mini yang dikemas dalam bentuk chip IC (*Integrated Circuit*) dan dirancang untuk mengendalikan berbagai perangkat elektronik secara mandiri. Chip ini mampu menerima sinyal input dari sensor atau perangkat lain, memprosesnya sesuai program yang tertanam di dalam memori, lalu mengirimkan sinyal output untuk menggerakkan aktuator, menampilkan informasi, atau mengendalikan modul lain. ATmega328P memiliki memori program sebesar 32 KB, SRAM 2 KB, EEPROM 1 KB, serta mendukung berbagai antarmuka komunikasi seperti UART, SPI, dan I²C. Mikrokontroler ini sering digunakan pada papan pengembangan arduino karena kemudahan

pemrograman dan ketersediaan banyak library pendukung (Nuran & Sofjan Firdausi, 2022). Pada proyek ini, mikrokontroler dibuat secara khusus dengan menggunakan ATmega328P sebagai mikrokontroler utama untuk mengontrol tampilan waktu shalat dan komunikasi dengan aplikasi remote.



Gambar 2. 3 Mikrokontroler ATmega328P
(<https://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html>)

Sisi performa, ATmega328P mampu beroperasi hingga kecepatan 20 MHz dengan bantuan kristal eksternal sebagai sumber clock. Frekuensi ini sudah cukup tinggi untuk menangani kebutuhan pemrosesan pada berbagai sistem kendali digital, baik untuk aplikasi sederhana seperti pengatur waktu maupun sistem otomatisasi berskala kecil hingga menengah. Selain kemampuan teknisnya, ATmega328P juga dirancang untuk efisiensi daya, menjadikannya ideal bagi sistem portabel yang menggunakan catu daya baterai. Terdapat beberapa mode hemat daya yang dapat dimanfaatkan seperti Idle Mode, ADC Noise Reduction, Power-down, dan Standby, di mana sistem dapat beroperasi dengan konsumsi energi minimum saat tidak sedang menjalankan tugas utama. Fitur tambahan seperti watchdog timer berfungsi sebagai pengaman sistem yang dapat melakukan reset otomatis apabila program tidak merespons, sedangkan brown-out detector

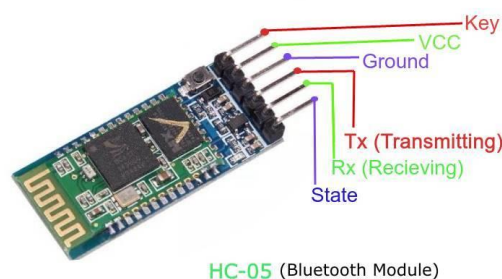
berperan menjaga kestabilan operasi saat tegangan suplai turun di bawah ambang batas yang ditentukan.

Salah satu keunggulan utama ATmega328P adalah kompatibilitasnya dengan platform Arduino Uno, sebuah papan pengembangan mikrokontroler yang banyak digunakan di berbagai bidang seperti pendidikan, penelitian, dan pengembangan prototipe. Dengan dukungan Arduino IDE yang mudah digunakan serta ketersediaan pustaka (*libraries*) yang sangat banyak, mikrokontroler ini sangat cocok bagi pemula hingga kalangan profesional dalam pengembangan sistem kendali berbasis mikrokontroler.

Secara keseluruhan, ATmega328P merupakan mikrokontroler yang serbaguna, efisien, dan memiliki integrasi tinggi, sehingga dapat digunakan pada berbagai aplikasi seperti sistem kendali otomatis, perangkat IoT, robotika, dan sistem monitoring konsumen. Kombinasi antara memori yang cukup, pin I/O yang fleksibel, fitur komunikasi digital, serta efisiensi daya menjadikan IC ini pilihan yang ideal dalam perancangan perangkat embedded modern.

2.2.10 Bluetooth HC-05

Bluetooth HC-05 adalah modul komunikasi nirkabel yang memungkinkan pengiriman data antara JWSN dan aplikasi *mobile*. Modul ini memungkinkan pengguna mengontrol dan memperbarui jadwal sholat tanpa harus mengakses perangkat secara fisik (TOP & GÖKBULUT, 2021).



Gambar 2. 4 Modul Bluetooth HC-05

(<https://www.geeksforgeeks.org/digital-logic/all-about-hc-05-bluetooth-module-connection-with-android/>)

Menghubungkan modul ini dengan sistem, HC-05 dilengkapi dengan beberapa pin yang masing-masing memiliki fungsi spesifik. Pin-pin tersebut meliputi Key, VCC, Ground (GND), Tx (Transmitting), Rx (Receiving), dan State. Berikut adalah penjelasan fungsi masing-masing pin:

1) Pin Key

Berfungsi untuk mengubah modul ke *AT Command Mode* atau mode konfigurasi. Saat pin ini diberi logika HIGH (3,3V atau 5V, tergantung spesifikasi modul) sebelum modul dinyalakan, HC-05 akan masuk ke mode konfigurasi yang memungkinkan pengguna mengatur parameter seperti *baud rate*, *device name*, dan *password pairing*. Jika pin ini tidak aktif, modul akan berjalan pada mode komunikasi data normal.

2) Pin VCC

Merupakan pin suplai daya untuk modul. Pin ini umumnya dihubungkan ke sumber tegangan 5V, namun beberapa varian HC-05 juga dapat bekerja dengan tegangan 3,3V. Modul HC-05 memiliki regulator internal sehingga aman digunakan langsung dengan suplai 5V dari mikrokontroler seperti Arduino.

3) Pin Ground (GND)

Berfungsi sebagai referensi tegangan dan jalur negatif dari sumber daya. Pin ini harus dihubungkan ke ground sistem mikrokontroler agar komunikasi dan suplai daya bekerja dengan benar.

4) Pin Tx (Transmitting)

Jalur keluaran data serial dari modul HC-05 menuju mikrokontroler atau perangkat penerima lainnya. Pin ini bekerja pada level logika TTL 3,3V, sehingga dapat langsung dihubungkan ke Arduino yang memiliki input tolerant 5V, namun tidak sebaliknya.

5) Pin Rx (Receiving)

Jalur masukan data serial ke modul HC-05 dari mikrokontroler atau perangkat pengirim lainnya. Karena modul bekerja pada logika 3,3V, sinyal yang masuk ke pin ini sebaiknya diturunkan tegangannya

menggunakan pembagi tegangan atau *level shifter* jika berasal dari perangkat dengan logika 5V, untuk mencegah kerusakan modul.

6) Pin State

Berfungsi sebagai indikator status koneksi Bluetooth. Pin ini akan memberikan logika HIGH ketika modul sedang terhubung dengan perangkat lain (*paired*), dan logika LOW ketika tidak ada koneksi. Pin ini sering digunakan untuk mengontrol LED indikator atau logika program yang memerlukan deteksi status koneksi.

Kelebihan:

1. Kemudahan integrasi: Mudah diintegrasikan dengan mikrokontroler seperti arduino, dengan banyak sumber daya dan tutorial yang tersedia.
2. Biaya rendah: Modul ini relatif murah, menjadikannya pilihan populer untuk proyek DIY dan prototipe.
3. Dukungan mode master/slave: Fleksibilitas dalam konfigurasi jaringan komunikasi.
4. Kompatibilitas luas: Dapat terhubung dengan berbagai perangkat yang mendukung bluetooth, seperti smartphone dan komputer.

Kekurangan:

1. Jangkauan Terbatas: Jarak komunikasi efektif sekitar 10meter, yang dapat menjadi kendala dalam aplikasi tertentu.
2. Tidak Mendukung BLE: HC-05 tidak mendukung Bluetooth Low Energy (BLE), sehingga konsumsi dayanya lebih tinggi dibandingkan modul BLE.
3. Kompatibilitas Terbatas dengan iOS: Modul ini tidak kompatibel dengan perangkat iOS karena keterbatasan dukungan SPP pada platform tersebut.

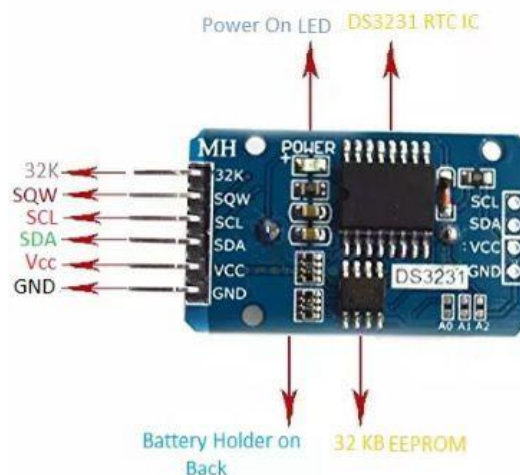
Cara Kerja:

Bluetooth HC-05 beroperasi pada frekuensi 2.4 GHz dan menggunakan modulasi GFSK (*Gaussian Frequency Shift Keying*). Modul ini memiliki dua mode operasi utama:

1. Mode *AT Command*: Digunakan untuk konfigurasi modul, seperti mengatur nama perangkat, *baud rate*, dan mode operasi (*master/slave*).
2. Mode Data (*Transparent Mode*): Digunakan untuk pertukaran data secara langsung antara perangkat yang terhubung.

2.2.11 RTC DS3231

RTC (*Real Time Clock*) adalah chip IC yang berfungsi menghitung waktu secara akurat mulai dari detik, menit, jam, hari, tanggal, bulan dan tahun. Untuk menjaga atau menyimpan data waktu yang telah dihidupkan modul memiliki sumber listrik sendiri yaitu baterai jam (Saputra & Hasna Tuddar Putri, 2023). Pada proyek ini RTC akan di pasang pada mikrokontroler JWSD.



Gambar 2. 5 RTC (Real Time Clock)

(<https://www.theengineeringprojects.com/2019/03/introduction-to-ds3231.html>)

Modul RTC (*Real Time Clock*) adalah modul waktu nyata yang digunakan untuk menyimpan dan menghitung waktu secara mandiri, bahkan ketika perangkat utama dimatikan, karena dilengkapi baterai cadangan. DS3231 memiliki akurasi tinggi dengan deviasi waktu yang sangat kecil berkat osilator kristal terkompensasi suhu (TCXO) yang tertanam di dalamnya. Modul ini menggunakan komunikasi I²C sehingga hanya membutuhkan dua pin data untuk bertukar informasi dengan mikrokontroler.

1) 32K

Pin keluaran sinyal frekuensi 32.768 kHz yang dapat digunakan sebagai referensi waktu eksternal untuk perangkat lain. Pin ini jarang digunakan pada aplikasi umum, tetapi bermanfaat untuk sinkronisasi sistem.

2) SQW

Pin keluaran gelombang persegi (square wave) yang dapat diatur frekuensinya melalui perintah dari mikrokontroler. Pin ini juga bisa difungsikan sebagai interrupt untuk memberi sinyal saat terjadi alarm pada modul RTC.

3) SCL

Pin *Serial Clock Line*, digunakan untuk menerima sinyal clock dari mikrokontroler dalam komunikasi I²C. Pin ini memastikan data terkirim dan diterima dengan sinkronisasi waktu yang tepat.

4) SDA

Pin *Serial Data Line*, digunakan untuk mengirim dan menerima data antara modul RTC dan mikrokontroler. Bersama dengan SCL, pin ini menjadi jalur utama komunikasi I²C.

5) VCC

Pin suplai daya untuk modul RTC. Tegangan kerja umumnya 3.3V hingga 5V DC, tergantung spesifikasi modul. Pin ini memberikan energi untuk rangkaian utama DS3231 saat perangkat utama menyala.

6) GND

Pin ground atau jalur negatif catu daya yang berfungsi sebagai referensi tegangan untuk modul. Pin ini harus dihubungkan ke ground mikrokontroler agar komunikasi I²C dapat berjalan dengan benar.

Keunggulan RTC DS3231:

1. Akurasi tinggi dengan kompensasi suhu otomatis.
2. Konsumsi daya rendah, sehingga cocok untuk aplikasi jangka panjang.
3. Dapat dihubungkan ke mikrokontroler melalui komunikasi I²C.

2.2.12 Buzzer

Buzzer adalah komponen elektronika yang berfungsi menghasilkan suara sebagai bentuk sinyal atau alarm. Komponen ini bekerja dengan mengubah sinyal listrik menjadi getaran mekanis yang kemudian menghasilkan gelombang suara. Buzzer sering digunakan pada berbagai perangkat elektronik sebagai indikator peringatan, penanda waktu, atau alarm sistem.



Gambar 2. 6 Buzzer
(<https://www.pcboard.ca/>)

Buzzer digunakan sebagai alarm atau indikator suara. Dalam sistem *JWSD*, buzzer digunakan untuk memberikan peringatan sebelum adzan atau iqamah (Wahono et al., 2024). Terdapat dua jenis buzzer yang umum digunakan, yaitu buzzer aktif dan buzzer pasif.

- 1) Buzzer aktif memiliki rangkaian osilator internal, sehingga dapat menghasilkan suara hanya dengan diberi tegangan DC.
- 2) Buzzer pasif tidak memiliki osilator internal, sehingga memerlukan sinyal gelombang (PWM) dari mikrokontroler untuk menghasilkan suara.

Konfigurasi pin Buzzer:

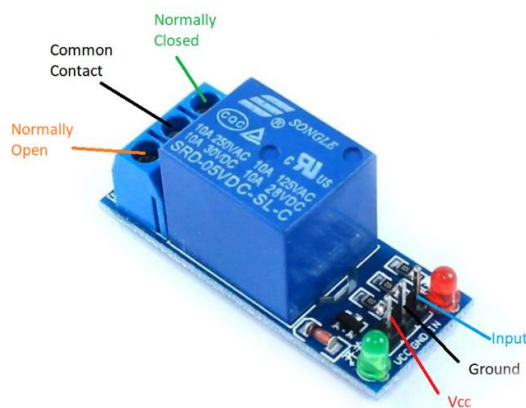
- 1) Pin (+) VCC terhubung ke output digital atau sumber tegangan positif untuk menerima sinyal atau tegangan penggerak.
- 2) Pin (-) GND terhubung ke jalur ground sistem.

Buzzer akan menghasilkan suara sesuai sinyal yang diterimanya, baik berupa bunyi kontinu maupun pola bunyi tertentu. Pola ini dapat diatur melalui rangkaian

pengendali atau program pada mikrokontroler. Penggunaan buzzer memberikan notifikasi suara yang efektif untuk menarik perhatian pengguna pada suatu kondisi tertentu, misalnya alarm peringatan, indikator proses, atau tanda waktu.

2.2.13 Relay 5V DC

Relay merupakan komponen saklar (*switch*) yang dikendalikan menggunakan listrik dan merupakan salah satu komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni kumparan magnet (*coil*) dan bagian mekanik (kontak saklar/*switch*). Relay bekerja berdasarkan prinsip elektromagnetik untuk menggerakkan kontakannya sehingga hanya dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi (Rohmah, 2024).



Gambar 2. 7 Relay 5V DC

(<https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet>)

Modul ini dilengkapi dengan beberapa pin dan terminal yang masing-masing memiliki fungsi berbeda.

1) VCC

Pin untuk suplai daya modul relay. Pin ini biasanya memerlukan tegangan sebesar 5V DC yang berasal dari mikrokontroler seperti arduino. Tegangan ini digunakan untuk memberi daya pada rangkaian pengendali dan koil elektromagnet yang menggerakkan saklar internal relay.

2) *Ground (GND)*

Pin ground atau jalur negatif dari catu daya. Pin ini menjadi referensi tegangan untuk seluruh rangkaian dan harus dihubungkan ke ground pada mikrokontroler agar sistem bekerja dengan benar.

3) *Input*

Pin kendali yang digunakan untuk mengaktifkan atau menonaktifkan relay. Pin ini menerima sinyal logika dari mikrokontroler, baik logika HIGH maupun LOW, tergantung tipe modul (active-high atau active-low). Saat mendapat sinyal aktif, koil elektromagnet dalam relay akan menarik saklar sehingga posisi kontak berubah.

4) *Common Contact (COM)*

Terminal pusat pada saklar relay yang menjadi titik hubung utama. Terminal ini dapat terhubung ke kontak *Normally Open* (NO) atau *Normally Closed* (NC) tergantung posisi kerja relay pada saat tertentu.

5) *Normally Open (NO)*

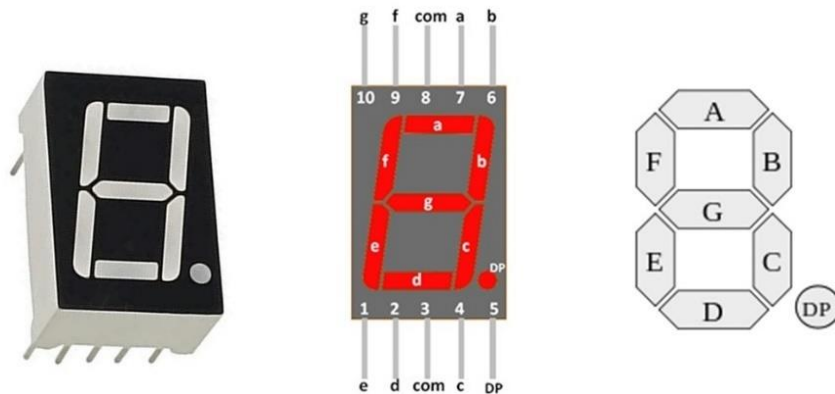
Terminal yang dalam kondisi default (relay mati) tidak terhubung dengan COM. Saat relay aktif, terminal NO akan terhubung dengan COM sehingga arus dapat mengalir ke beban. Terminal ini biasanya digunakan untuk mengendalikan perangkat yang hanya menyala ketika relay diaktifkan.

6) *Normally Closed (NC)*

Terminal yang secara default terhubung dengan COM ketika relay tidak aktif. Saat relay aktif, koneksi ini akan terputus sehingga arus berhenti mengalir. Terminal ini cocok digunakan untuk perangkat yang secara default menyala dan akan mati ketika relay diaktifkan.

2.2.14 Seven Segment

Seven Segment Display adalah komponen elektronika yang digunakan untuk menampilkan angka atau karakter tertentu dengan menggunakan tujuh buah segmen LED yang disusun membentuk angka 8. Setiap segmen diberi label huruf a hingga g, sedangkan titik desimal disebut DP (Decimal Point).



Gambar 2. 8 Seven Segment
<https://www.elektronikahendry.com>

Menurut (Januar et al., 2022), penggunaan seven segment display masih menjadi pilihan populer dalam perangkat elektronik sederhana maupun sistem terintegrasi karena kemudahan pengendalian dan keterbacaannya yang tinggi, meskipun tantangan seperti pengaturan intensitas cahaya dan keterbatasan bentuk karakter tetap menjadi perhatian dalam perancangan. Terdapat dua jenis seven segment berdasarkan konfigurasi pin:

1. Common Cathode (CC) semua kaki katoda segmen terhubung menjadi satu ke ground, dan setiap segmen dinyalakan dengan memberi logika HIGH pada anoda.
2. Common Anode (CA) semua kaki anoda segmen terhubung menjadi satu ke VCC, dan setiap segmen dinyalakan dengan memberi logika LOW pada katoda.

Keterangan Pin:

1. Pin 1 (e) Mengontrol segmen e.
2. Pin 2 (d) Mengontrol segmen d.
3. Pin 3 (COM) Pin common (common cathode atau anode).
4. Pin 4 (c) Mengontrol segmen c.
5. Pin 5 (DP) Mengontrol titik desimal (Decimal Point).
6. Pin 6 (b) Mengontrol segmen b.
7. Pin 7 (a) Mengontrol segmen a.

8. Pin 8 (COM) Pin common kedua (sama fungsi dengan pin 3).
9. Pin 9 (f) Mengontrol segmen f.
10. Pin 10 (g) Mengontrol segmen g.

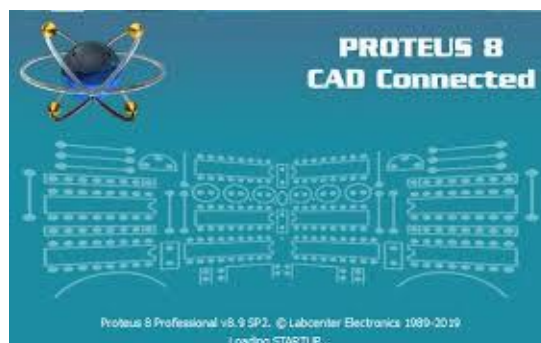
Cara Kerja:

Dengan mengatur logika pada masing-masing pin segmen, kita dapat membentuk angka 0–9 atau karakter tertentu. Misalnya, untuk menampilkan angka "2", segmen yang harus dinyalakan adalah a, b, g, e, d.

Seven segment display banyak digunakan pada jam digital, penghitung, alat ukur, dan berbagai perangkat elektronik yang memerlukan tampilan numerik sederhana.

2.2.15 Proteus 8 Pro

Software Proteus 8 Professional merupakan salah satu software elektronik yang digunakan untuk membantu para designer dalam merancang dan mensimulasikan suatu rangkaian elektronik (Saputra & Hasna Tuddar Putri, 2023). Proteus mendukung simulasi rangkaian analog dan digital, termasuk integrasi langsung dengan berbagai mikrokontroler seperti Arduino, PIC, AVR, dan ARM. Kelebihannya terletak pada kemampuan menjalankan *firmware* dalam lingkungan virtual, sehingga kesalahan dapat dideteksi lebih awal tanpa biaya pembuatan prototipe fisik.



Gambar 2. 9 Proteus 8 Pro

Pustaka komponen yang luas, mulai dari resistor hingga modul tampilan seperti seven segment dan LCD, menjadikan Proteus cocok untuk pengembangan sistem tertanam. Selain simulasi, Proteus juga menyediakan fitur perancangan PCB, sehingga proses dari desain skematik hingga tahap manufaktur dapat dilakukan dalam satu platform yang efisien dan terintegrasi.

2.2.16 Arduino IDE Software

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain *Arduino IDE* sebagai media untuk pemrograman pada board yang ingin diprogram. *Arduino IDE* ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu (Mahanin Tyas et al., 2023).



Gambar 2. 10 Arduino IDE
(<https://blog.indobot.co.id/fitur-utama-arduino-ide/>)

Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++ (wiring), yang membuat operasi input/output lebih mudah. Kelebihan Arduino IDE adalah kesederhanaan antarmuka dan kemudahan integrasi pustaka, sehingga pemula maupun pengembang berpengalaman dapat dengan cepat membuat prototipe. Selain itu, IDE ini mendukung berbagai sistem operasi seperti Windows, macOS, dan Linux, serta tersedia versi *offline* dan *web editor*.