

# Bel Sekolah Otomatis Berbasis Arduino yang Dikontrol Menggunakan Aplikasi *Mobile*

## (Automatic School Bell Based on Arduino Controlled By Mobile Application)

Muh Pauzan<sup>1</sup>, Indri Yanti<sup>2</sup>

**Abstract**—The school bell is one of the important devices in school. It has been used from years ago to the present. Bell has been improved along with time. Nowadays, research about school bell has been focused on the bell's controller, such as using software in a computer, placing a keypad on the bell, and even controlling the bell's schedule, which can only be performed using coding. However, these controllers are not practical; therefore, an efficient school bell is made by using an Android-based application as the controller. Bluetooth is used as a communication tool between the smartphone and the bell. The research method was by conducting a literature study and doing a survey to SDN 1 Sarapati Indramayu. Based on the method, it has been decided to make four modes on the Android application, i.e., regular mode, exam mode, free, dan emergency mode. Regular mode is used in a normal condition, and exam mode is used when the students are doing an exam. If teachers want to ring the bell directly, the free mode is operated. There are four types of bell ring in free mode, i.e., once, twice, three, and four-time rings. Emergency mode is useful when the school is in an emergency, such as earthquakes, floods, and fires. Results show that the maximum range to control the bell is 9 m.

**Intisari**—Bel sekolah adalah salah satu alat yang penting di sekolah, digunakan dari dahulu sampai saat ini. Seiring dengan perkembangan zaman, bel sekolah juga mengalami kemajuan. Penelitian tentang bel sekolah saat ini banyak yang difokuskan pada pengontrol bel, seperti menggunakan perangkat lunak di komputer, menggunakan keypad yang ditempatkan pada bel, bahkan mengatur jadwal bel berbunyi yang hanya dapat dilakukan melalui pemrograman. Pengontrol-pengontrol tersebut tidak praktis, sehingga dibuat bel dengan pengontrol yang lebih praktis, yaitu aplikasi berbasis Android. Bluetooth digunakan sebagai alat komunikasi *smartphone* dengan bel. Metode yang dilakukan adalah melakukan studi literatur dan melakukan survei di SDN 1 Sarapati, Indramayu. Berdasarkan kedua hal tersebut, dibuat empat mode pada aplikasi Android, yaitu mode reguler, ujian, bebas, dan mode darurat. Mode reguler digunakan pada kondisi normal, sedangkan mode ujian digunakan jika siswa sedang melaksanakan ujian. Jika guru hendak membunyikan bel secara langsung, digunakan mode bebas. Pada mode ini, dibuat empat tipe bunyi bel, seperti bunyi sekali, dua kali, tiga kali, dan empat kali. Mode darurat digunakan ketika sekolah dalam keadaan darurat, seperti terjadinya gempa bumi, banjir, dan

kebakaran. Hasil pengujian menunjukkan bel dapat dikontrol pada jarak maksimum 9 m.

**Kata Kunci**—Bel Sekolah, Pengontrol Bel, Aplikasi Berbasis Android, Bluetooth.

### I. PENDAHULUAN

Bel merupakan penanda suatu kegiatan dimulai atau berhenti di sekolah. Lima belas sampai dua puluh tahun yang lalu, bel listrik sudah digunakan secara manual, dengan dipencet oleh petugas atau guru. Saat ini, bel sekolah yang tersedia sudah semakin canggih, seperti memiliki fitur otomatisasi, yaitu petugas atau guru tidak perlu lagi memencet setiap masuk, istirahat, ganti jam pelajaran, ataupun saat waktu pulang. Penelitian terkait bel sekolah juga sudah banyak dilakukan, seperti mengenai bel sekolah berbasis komputer [1]. Komputer berfungsi sebagai pengontrol waktu bel dibunyikan. *User interface* dibangun menggunakan Delphi dan dari perangkat lunak inilah jadwal bel berbunyi diatur. Sinyal keluaran dari komputer berfungsi untuk mengontrol *relay*. Jika kondisi *relay on*, speaker akan menyala. Penelitian yang sama juga telah dilakukan, yaitu membuat bel otomatis berbasis *Personal Computer* (PC) dengan aplikasi *user interface* menggunakan Delphi [2]. Penelitian lainnya membuat bel sekolah menggunakan PC, tapi perangkat lunak yang digunakan untuk membuat *user interface* adalah Visual Studio [3]. Bel yang dibuat pada penelitian-penelitian tersebut memiliki kekurangan seperti boros energi, yaitu daya komputer berkisar dari 65 watt sampai 250 watt. Selain itu, pengontrol yang digunakan adalah aplikasi di komputer. Hal tersebut tentu tidak praktis karena setiap kali jadwal bel perlu diubah, petugas harus membuka aplikasi di komputer. Selain tidak praktis pengontrol bel menggunakan komputer juga berdimensi besar dan berat. Berikutnya, dibuat bel otomatis berbasis Raspberry Pi [4]. Tentu saja bel ini lebih praktis dari segi ukuran dan lebih murah dari segi harga, jika dibandingkan dengan yang menggunakan PC. Akan tetapi, kelemahan pengontrol bel ini adalah pengaturan jadwal bel berbunyi harus dilakukan menggunakan komputer atau laptop yang dihubungkan menggunakan kabel ke bel [4]. Penelitian lainnya melakukan pembuatan bel otomatis berbasis Arduino Uno. Ditinjau dari segi harga, penggunaan Arduino Uno lebih murah, dengan kemampuan sama dengan Raspberry Pi. Namun, sistem ini tidak praktis karena pengontrol jadwalnya menggunakan *push button* [5] dan aplikasi di komputer [6]. Penelitian berikutnya menjawab kekurangan tersebut, dengan membuat bel otomatis menggunakan Arduino Uno yang dikontrol menggunakan *web browser* [7]. Jadwal bel dapat diatur di mana pun karena berbasis internet. Penelitian lain juga memanfaatkan *website*

<sup>1</sup>Departemen Teknik Komputer Fakultas Teknik Universitas Wiralodra, Jl. Ir. H. Juanda Km.03 Singaraja 45213 INDONESIA (telp: 0234-275946; fax: 0234-273040; e-mail: muhpauzan.ft@unwir.ac.id)

<sup>2</sup>Departemen Teknik Sipil Fakultas Teknik Universitas Wiralodra, Jl. Ir. H. Juanda Km.03 Singaraja 45213 INDONESIA (e-mail: indriyanti.ft@unwir.ac.id)

TABEL I  
KONEKSI *PIN* RTC DS3231 DENGAN ARDUINO NANO

RTC DS3231	Arduino Nano
SDA	A4
SCL	A5
VCC	+5V
GND	GND

TABEL II  
KONEKSI *PIN* I2C LCD DENGAN ARDUINO NANO

I2C LCD	Arduino Nano
SDA	A4
SCL	A5
VCC	+5V
GND	GND

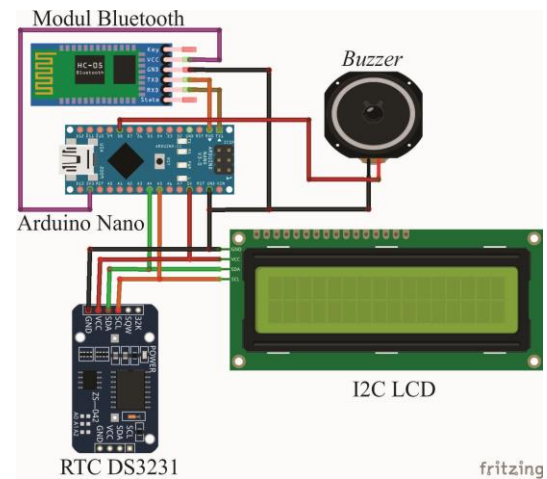
TABEL III  
KONEKSI Hc-05 DENGAN ARDUINO NANO

Hc-05	Arduino Nano
RXD	TXD
TXD	RXD
VCC	3,3V
GND	GND

sebagai pengontrol belnya, tetapi menggunakan Arduino Mega pada bel [8]. Pengontrol bel berbasis website memerlukan biaya yang cukup tinggi karena biayanya harus ditambah untuk penyewaan *hosting*, pembelian *router*, dan pembelian Ethernet Shield. Untuk menghemat biaya dan agar sesuai dengan kebutuhan, sebenarnya tetap dapat dibuat bel yang dikontrol secara *wireless* dengan jarak dekat tanpa mengurangi kemudahan untuk mengontrol jadwal bel berbunyi. Untuk mengatasi masalah tersebut, dirancang dan dibangun bel otomatis menggunakan Arduino Nano yang dikontrol melalui aplikasi *mobile* berbasis Android. Aplikasi dibuat menggunakan MIT App Inventor. Konsep dari aplikasi ini ditentukan berdasarkan hasil studi literatur dan survei yang dilakukan di SDN 1 Sarapati, Indramayu. Aplikasi tersebut memiliki fitur pengaturan jadwal yang terdiri atas mode reguler, mode ujian, mode bebas, dan *emergency*. *Board* yang digunakan adalah Arduino Nano, yang dipilih karena ukurannya yang paling kecil serta harga yang lebih murah dibandingkan dengan *board* Arduino lainnya. Dengan menggunakan Arduino Nano, bobot bel menjadi lebih ringan dan dimensi bel menjadi lebih kecil. Hal baru lainnya adalah komunikasi antara bel dengan aplikasi di *smartphone* dilakukan menggunakan Bluetooth, sehingga baik dari segi fungsi maupun kepraktisannya, tidak jauh berbeda dengan sistem yang menggunakan sambungan internet.

## II. METODOLOGI

Proses pembuatan bel sekolah dimulai dari membaca artikel-artikel ilmiah tentang bel sekolah. Hal tersebut bertujuan untuk menganalisis kekurangan dan kelebihan bel yang sudah ada. Selain itu, dilakukan survei ke SDN 1 Sarapati, Kecamatan Karanganyar, Indramayu, untuk mengetahui secara detail jadwal bel dibunyikan, mulai dari bel masuk sampai dengan bel pulang, baik saat jadwal pelajaran (hari normal) maupun saat



Gbr. 1 Koneksi DS3231, Hc-05, I2C LCD, dan *buzzer* dengan Arduino Nano.

ada jadwal ujian. Data yang diperoleh kemudian dijadikan rujukan untuk membuat aplikasi berbasis Android serta pembuatan program pada *Integrated Development Environment* (IDE) Arduino. Pembuatan alat secara detail dijabarkan sebagai berikut.

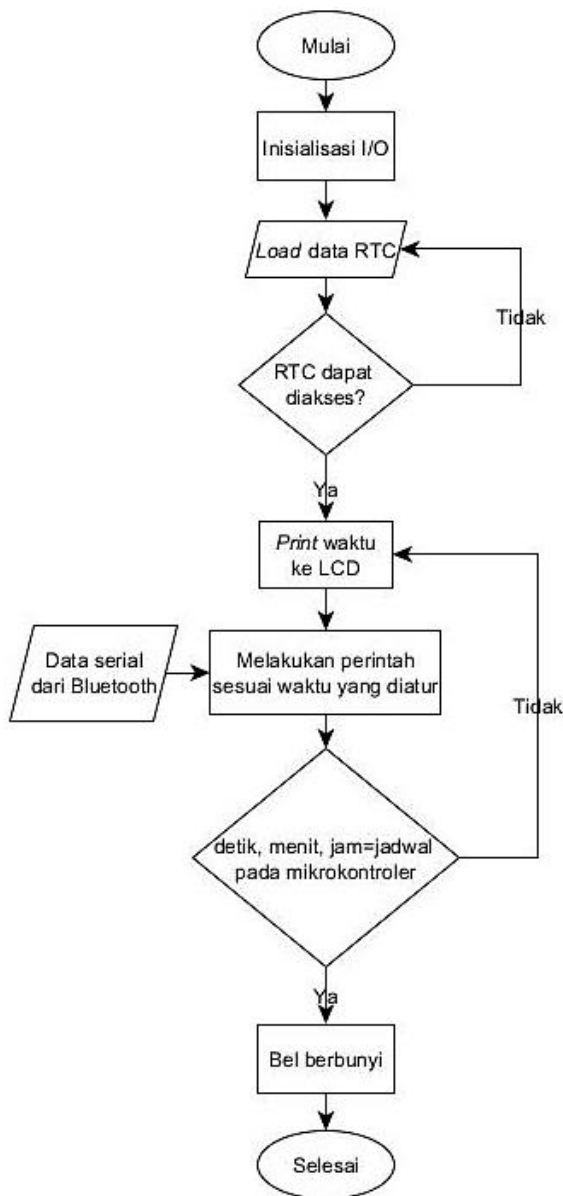
### A. Rancangan Perangkat Keras

Pada tahap ini, dibuat skema rangkaian antara Arduino Nano dengan RTC DS3231, I2C LCD 16x2, modul Bluetooth Hc-05, dan *buzzer*. Koneksi *pin* antara RTC DS3231 dengan Arduino Nano diperlihatkan pada Tabel I, koneksi *pin* antara I2C LCD 16x2 dengan Arduino Nano disajikan pada Tabel II, sedangkan koneksi *pin* modul Bluetooth Hc-05 dengan Arduino Nano ditunjukkan pada Tabel III. Koneksi tiap *pin* pada komponen-komponen tersebut diperlihatkan pada Gbr. 1.

Berdasarkan Gbr. 1, hal yang perlu diperhatikan adalah koneksi RTC DS3231 ke Arduino Nano antara SDA ke *pin* A4 dan SCL ke *pin* A5 [9]. SDA adalah *pin* untuk data serial. *Pin* ini berfungsi sebagai masukan/keluaran untuk I2C *serial interface*. Oleh karena itu, *pin* SDA pada DS3231 dihubungkan ke *pin* SDA I2C LCD. *Pin* SDA pada Arduino Nano berada pada A4. SCL adalah *pin* untuk *serial clock*, yang berfungsi sebagai *clock* untuk masukan ke I2C yang ada pada LCD. *Pin* ini digunakan untuk menyinkronkan pergerakan data dalam *serial interface*. *Pin* SCL berada pada *pin* A5 di *board* Arduino Nano. *Pin* TXD pada modul Bluetooth Hc-05 dihubungkan ke *pin* RXD pada *board* Arduino Nano, sedangkan *pin* RXD dihubungkan ke *pin* TXD. Hal ini dilakukan karena saat mengirim data dari aplikasi berbasis Android yang dibuat, data tersebut diterima pertama kali oleh Hc-05. Hc-05 meneruskan data ke Arduino Nano melalui *pin* TXD [10]. *Pin* ini berfungsi meneruskan data serial dengan kecepatan 9600 bps. Karena Arduino Nano bertugas menerima data, maka digunakan *pin* RXD pada *board*-nya.

### B. Rancangan Perangkat Lunak

Pada tahap ini ditentukan konsep yang akan diterapkan berkaitan dengan program pada *board* Arduino Nano melalui perangkat lunak IDE Arduino. Urutan kerja alat yang dibuat ditunjukkan pada Gbr. 2.



Gbr. 2 Diagram alir cara kerja alat.

Berdasarkan Gbr. 2, saat bel dinyalakan, Arduino Nano menginisiasi *pin* I/O. Inisiasi ini berada di blok *void setup()* di IDE Arduino. Data berupa jam, tanggal, bulan, dan tahun didapatkan dari RTC DS3231, kemudian data tersebut ditampilkan melalui LCD. Selanjutnya, Arduino Nano menunggu perintah dari aplikasi di *smartphone*. Komunikasi keduanya dilakukan menggunakan Bluetooth. Aplikasi di *smartphone* mengirimkan data serial, lalu Arduino menerima dan mencocokkannya dengan program di dalamnya. Jika data yang diterima sama dengan program untuk menyalakan bel satu kali, maka Arduino akan membunyikan bel satu kali. Demikian juga untuk mode yang lain.

C. Perancangan Aplikasi Mobile Berbasis Android

Aplikasi *mobile* dibuat menggunakan aplikasi MIT App Inventor. MIT App Inventor adalah sistem pembuatan aplikasi Android berbasis *website*. App inventor diciptakan oleh Google

tetapi saat ini dikelola oleh Massachusetts Institute of Technology (MIT). MIT App Inventor bersifat *open-source*, serta berbasis *visual block programming*. Pada MIT App Inventor, digunakan empat konsep saat membuat aplikasi *mobile* pengontrol bel, yaitu jadwal reguler, jadwal ujian, bel dapat dibunyikan kapan saja, serta tombol *emergency*. Dari konsep inilah dihasilkan *user interface* seperti pada Gbr. 3. Tampilan aplikasi setelah pengguna menekan tombol “lihat jadwal reguler” dan “lihat jadwal ujian” diperlihatkan pada Gbr. 4.

Cara penggunaan aplikasi *mobile* ini adalah cukup dengan menekan tombol yang dikehendaki. Jadwal reguler dan jadwal ujian mengikuti hasil survei yang dilakukan ke SDN 1 Sarapati. Cara kerja aplikasi ini adalah saat salah satu tombol ditekan, data bertipe *char* dikirim ke modul Hc-05. Sebagai contoh, data yang dikirim adalah “2” untuk menyalakan bel satu kali. Hc-05 kemudian meneruskan “2” ke Arduino Nano. Program pada *board* tersebut sudah diatur agar jika data yang diterima berupa “2” dengan tipe data *char*, bel dinyalakan satu kali. Demikian juga untuk mode lainnya.

D. Cara Kerja Alat

Cara kerja alat ditunjukkan pada Gbr. 5. Penjelasan cara kerja alat tersebut adalah sebagai berikut.

1. Saat bel dinyalakan, Arduino Nano pertama kali akan mengakses waktu dan tanggal dari RTC DS3231.
2. Data tersebut kemudian diteruskan ke I2C LCD sehingga akan muncul jam serta hari, tanggal, bulan, dan tahun pada layar I2C LCD.
3. Jika modul Bluetooth Hc-05 mendapatkan data serial dari aplikasi di *smartphone*, data tersebut akan diteruskan ke Arduino Nano
4. Arduino akan mencocokkan data yang diperoleh dengan program yang sudah ada di dalam mikrokontrolernya.
5. Jika data yang diterima sama dengan data di dalamnya, Arduino Nano akan langsung mengeksekusi blok program yang ada di dalam data tersebut.

III. HASIL DAN PEMBAHASAN

A. Realisasi Perangkat Keras dan Lunak

Setelah komponen-komponen yang diperlukan dirangkai, selanjutnya dilakukan unduh program dari IDE Arduino ke *board* Arduino Nano. Kemudian, dilakukan instalasi aplikasi pengontrol bel pada *smartphone*. Langkah terakhir adalah menghubungkan bel dengan aplikasi menggunakan Bluetooth. Tampilan bel tampak depan diperlihatkan pada Gbr. 6(a), sedangkan rangkaian di dalam bel ditunjukkan pada Gbr. 6(b).

Berdasarkan Gbr. 6(a), *smartphone* yang sudah terkoneksi dengan modul Hc-05 selanjutnya diuji coba dengan menekan salah satu tombol di aplikasi *smartphone*.

B. Keputusan pada Setiap Kondisi

Jika pengguna menekan tombol “bunyi satu kali” di aplikasi yang dibuat, dari *smartphone* dikirim data dengan nilai “2” bertipe data *char*. Arduino Nano akan melakukan eksekusi karena sesuai dengan program di dalamnya. *Syntax* program pada IDE Arduino untuk kondisi ini adalah sebagai berikut.



Gbr. 3 Tampilan *user interface*, (a) sebelum *smartphone* terkoneksi dengan bel, (b) setelah *smartphone* terkoneksi dengan bel.



Gbr. 4 Contoh tampilan beberapa mode, (a) jadwal bel berbunyi saat sekolah normal, (b) jadwal bel berbunyi saat ujian.

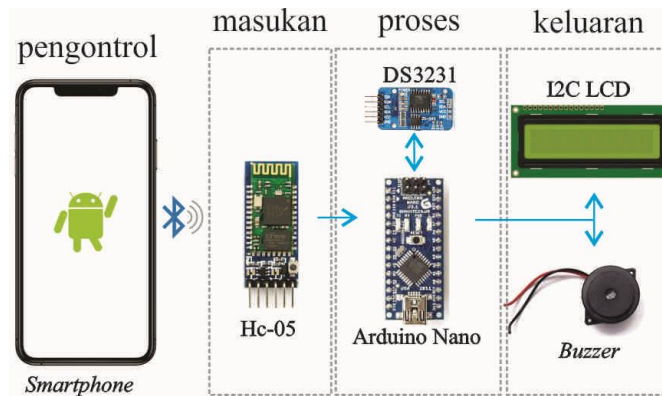
```

if (Serial.available() > 0)
{
  incomingValue = Serial.read();
  Serial.println(incomingValue);
  //Serial.print("\n");
  if (incomingValue == '0')
  if (incomingValue == '2')
  {
    BelBunyi1();
  }
}

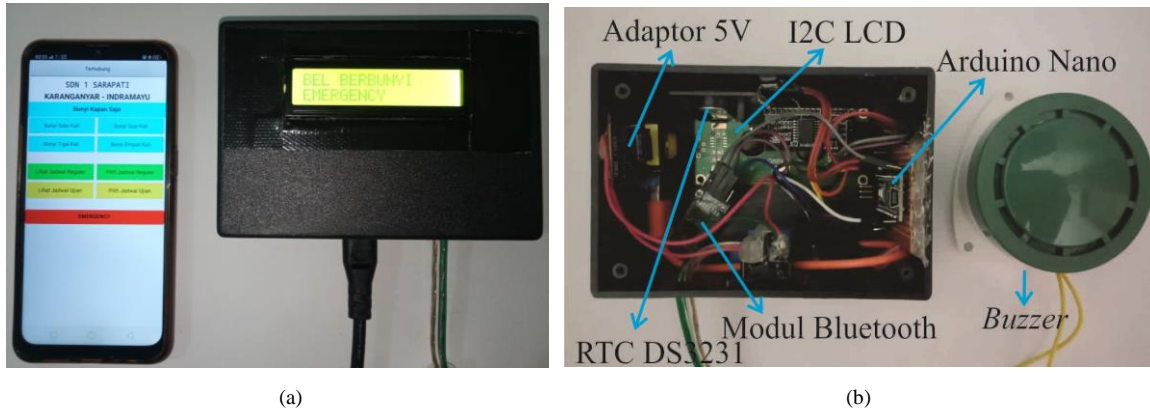
```

Fungsi `BelBunyi1()` adalah fungsi yang dideklarasikan di tempat lain. Fungsi ini membunyikan bel sekali dengan durasi 5 detik. Berdasarkan *syntax* di atas, jika data serial yang diterima Arduino Nano tidak sama dengan “2”, blok *if* yang ada di dalamnya tidak dilakukan.

Untuk tombol “Bunyi Dua Kali”, data yang dikirim oleh aplikasi di *smartphone* adalah “3”. Supaya bel berbunyi dua kali, dideklarasikan sebuah fungsi bernama `BelBunyi2()`.



Gbr. 5 Cara kerja bel.



Gbr. 6 Perangkat keras bel, (a) bel tampak depan, (b) rangkaian pada bel.

Pada fungsi ini, bel diatur untuk berbunyi 2,5 detik, lalu mati 400 milidetik, kemudian berbunyi lagi selama 2,5 detik. Tombol “Bunyi Tiga Kali” dan “Bunyi Empat Kali” mengirim data, berturut-turut, “4” dan “5” ke bel.

Pada tombol “Emergency” di aplikasi *smartphone*, data yang dikirim bernilai “6” dengan tipe data *char*. Arduino Nano akan mengeksekusi program jika mendapatkan data bernilai “6”, dengan *syntax* program adalah sebagai berikut.

```

if (Serial.available()>0)
{
    incomingValue=Serial.read();
    Serial.println(incomingValue);
    //Serial.print("\n");
    if (incomingValue=='0')
if (incomingValue=='6')
{
    Emergency();
}
}
    
```

Berdasarkan *syntax* tersebut, jika diterima data bernilai “6”, maka fungsi *Emergency()* akan dieksekusi. Di dalam tubuh fungsi ini terdapat instruksi untuk menyalakan bel selama 400 milidetik, lalu mati selama 400 milidetik. Hal tersebut berulang sampai 41 kali perulangan.

*Syntax* untuk membunyikan bel secara langsung di atas tidak dapat langsung diterapkan untuk membunyikan bel yang sudah terjadwal seperti pada tombol “Pilih Jadwal Reguler” atau pada tombol “Pilih Jadwal Ujian” karena saat dipilih tombol-tombol

yang dimaksud, Arduino Nano akan langsung mengeksekusi program. Untuk mengatasi masalah tersebut, dideklarasikan variabel bernama *a* untuk mengeksekusi tombol “Pilih Jadwal Reguler”. *Syntax* program tersebut adalah sebagai berikut.

```

char a=0;
void loop()
{
    if (a==9)
    {
        //sintaks untuk bunyi bel di hari senin
        if (rtc.getDOWStr()=="Monday")
        {
            Reguler();
        }
    }
    if (Serial.available()>0)
    {
        incomingValue=Serial.read();
        Serial.println(incomingValue);
        Serial.print("\n");
        if (incomingValue=='0')
        {
            a=9;
        }
    }
}
    
```

Berdasarkan *syntax* di atas, jika Arduino Nano memperoleh data dari Hc-05 bernilai “9”, maka variabel *a* yang sebelumnya bernilai “0” berubah menjadi “9”. Jadi, komunikasi Bluetooth dengan bel hanya berfungsi untuk mengubah nilai variabel *a*

saja. Setelah variabel *a* bernilai “9”, maka blok yang ada di dalam statemen *if* (*a*==9) akan dieksekusi. Blok yang ada di dalamnya memanggil fungsi *Reguler()*. *Syntax* dari fungsi ini adalah sebagai berikut.

```
void Reguler ()
{
  if ((Jam==07) && (Menit==00) && (Detik==00))
  {
    BelBunyi3();
  }
  if ((Jam==9) && (Menit==15) && (Detik==00))
  {
    BelBunyi2();
  }
  if ((Jam==9) && (Menit==45) && (Detik==00))
  {
    BelBunyi3();
  }
  if ((Jam==11) && (Menit==45) && (Detik==00))
  {
    BelBunyi2();
  }
  if ((Jam==12) && (Menit==10) && (Detik==00))
  {
    BelBunyi3();
  }
  if ((Jam==13) && (Menit==00) && (Detik==00))
  {
    BelBunyi4();
  }
}
```

Berdasarkan fungsi *Reguler()* tersebut, bel dibunyikan dari jam masuk pagi sampai jadwal pulang. Saat masuk kelas, bel dibunyikan dua kali, saat istirahat dibunyikan tiga kali, dan saat pulang dibunyikan empat kali. Hal ini diwakili oleh fungsi *BelBunyi2()*, *BelBunyi3()*, dan *BelBunyi4()* berturut-turut.

Tombol “Pilih Jadwal Ujian” digunakan untuk membunyikan bel pada waktu siswa melaksanakan ujian. Digunakan cara yang sama dengan *syntax* pada tombol “Pilih Jadwal Reguler” di atas, hanya saja dideklarasikan variabel bernama *b* dengan nilai awal sebesar “0”. Jika Arduino Nano menerima data dari Hc-05 bernilai “1”, maka nilai variabel *b* berubah menjadi “11”. Jika *b* bernilai “11”, maka bel dibunyikan sesuai dengan jadwal ujian.

Tipe data sangat penting untuk dipahami. Tipe data yang digunakan untuk komunikasi serial antara aplikasi di *smartphone* dengan Arduino Nano adalah tipe data *char*. Pada percobaan dengan tipe data yang lain, data yang dikirim diterima berbeda oleh Arduino Nano. Hasil analisis menunjukkan bahwa data yang dikirim dari *smartphone* diubah ke bentuk *American Standard Code for Information Interchange* (ASCII) saat data tersebut diterima oleh Arduino Nano sehingga perintah yang diberikan ke bel tidak dapat dieksekusi.

### C. Pengujian Bel

Selanjutnya, dilakukan uji coba dengan mengirim data ke bel pada jarak bervariasi, mulai dari 1 m, lalu 2 m, 3 m, 4 m, 5 m, 6 m, 7 m, 8 m, 9 m, dan 10 m. Jarak dari 1 m sampai 10 m dipilih karena dalam *datasheet* Hc-05 disebutkan bahwa rentang maksimum komunikasi adalah < 100 m [10] dan

TABEL IV  
MENGUJI BATAS KONEKSI ANTARA *SMARTPHONE* DENGAN BEL

Jarak (m)	Status
1	Pengontrol bel bekerja dengan baik
2	Pengontrol bel bekerja dengan baik
3	Pengontrol bel bekerja dengan baik
4	Pengontrol bel bekerja dengan baik
5	Pengontrol bel bekerja dengan baik
6	Pengontrol bel bekerja dengan baik
7	Pengontrol bel bekerja dengan baik
8	Pengontrol bel bekerja dengan baik
9	Pengontrol bel bekerja dengan baik
10	Pengontrol tidak dapat bekerja

Bluetooth di *smartphone* dapat berkomunikasi dalam jarak yang lebih pendek [11]. Pengujian dilakukan dengan cara menghubungkan *smartphone* dengan bel melalui koneksi Bluetooth. Hasilnya pengujian ditunjukkan pada Tabel IV.

Berdasarkan Tabel IV, pada jarak 10 m komunikasi antara *smartphone* dengan bel tidak berjalan, meskipun di *datasheet* Hc-05 memiliki frekuensi radio 2,4 GHz [10], yang artinya rentang komunikasi bisa mencapai > 10 m jika dihubungkan dengan perangkat Bluetooth lain yang memiliki spesifikasi sama. Akan tetapi, perangkat Bluetooth yang ada di *smartphone* didesain untuk komunikasi dalam jarak dekat [11], sehingga berdasarkan uji coba yang dilakukan, diperoleh jarak maksimum 9 m. Dengan jarak tersebut, guru/petugas dapat mengontrol bel dengan jarak sekitar tiga ruang kelas dari bel ditempatkan.

## IV. KESIMPULAN

Berdasarkan hasil rancang bangun bel sekolah otomatis yang telah dilakukan, dapat disimpulkan bahwa data yang dikirim oleh *smartphone* ke modul Bluetooth Hc-05 mensyaratkan data dalam bentuk *char*. Jika digunakan format data selain *char*, data dari *smartphone* akan dikonversi ke bentuk ASCII oleh Hc-05. Kemudian, pengontrol dapat bekerja dengan baik dengan jarak maksimum 9 m dari bel ditempatkan.

## UCAPAN TERIMA KASIH

Disampaikan terima kasih kepada Lembaga Penelitian dan Pengabdian (LPPM) Universitas Wiralodra yang telah membiayai penelitian ini melalui skema Hibah Penelitian Internal Tahun 2020.

## REFERENSI

- [1] M.S. Sungkar, “Perancangan Otomatisasi Bel Sekolah dengan Autopower Menggunakan Interface Berbasis Desktop,” *J. Inform. J. Pengemb. IT Poltek Tegal*, Vol. 2, No. 2, hal. 84–89, 2017.
- [2] A.L. Affandy, “Rancang Bangun Bel Sekolah Otomatis Berbasis Personal Computer (PC),” *J. Informatika*, Vol. 8, No. 3, hal. 194–206, 2017.
- [3] M.S. Riyadi, “Implementasi Aplikasi Bel Sekolah Otomatis untuk Membantu Pergantian Jam Belajar pada SDN 2 Brebes,” *Publicitas*, Vol. 1, No. 1, hal. 1–5, 2015.
- [4] M. Subianto, “Sistem Bel Otomatis Terprogram Berbasis Raspberry Pi,” *SMATIKA*, Vol. 05, No. 1, hal. 5–12, 2015.
- [5] Sarmidi dan A. Nurtado, “Simulasi Bel Sekolah Otomatis Berbasis Arduino Uno,” *J. Manaj. dan Tek. Inform.*, Vol. 3, No. 1, hal. 121–130, 2019.

- [6] A. Linarta dan N. Nurhadi, "Aplikasi Bel Sekolah Otomatis Berbasis Arduino Dilengkapi dengan Output Suara," *INFORMATIKA*, Vol. 10, No. 2, hal. 1–7, 2018.
- [7] D. Satria, Y. Yanti, dan M. Maulinda, "Rancang Bangun Sistem Penjadwalan Bel Sekolah Berbasis Arduino Uno dengan Antarmuka Berbasis Web Menggunakan Ethernet Web Server," *J. Serambi Eng.*, Vol. II, No. 3, hal. 141–147, 2017.
- [8] A.T. Setiawan, "Rancang Bangun Bel Sekolah Otomatis Menggunakan Mikrokontroler Arduino Mega dengan Antarmuka Berbasis Web," *Sem. Nas. Inov. Teknol.(SNITek 2019)*, 2019, hal. 68–80.
- [9] "RTC DS 3231 Datasheet," Maxim Integrated, 2015.
- [10] "Hc-05 Bluetooth Module," ITEad Studio, 2010.
- [11] J.F. Ensworth dan M.S. Reynolds, "Every Smart Phone is a Backscatter Reader: Modulated Backscatter Compatibility with Bluetooth 4.0 Low Energy (BLE) Devices," *2015 IEEE Int. Conf. RFID (RFID 2015)*, 2015, hal. 78–85.