

IMPLEMENTASI MIKROKONTROLER BERBASIS ARDUINO SEBAGAI KUNCI (*DONGLE*) APLIKASI PERANGKAT LUNAK

Ikhsan Shobari, Indarzah MP, Jos Budi Sulisty, Usep Setia Gunawan
Pusat Rekayasa Fasilitas Nuklir – BATAN
Gedung 71 Kawasan Puspiptek Serpong, Tangerang Selatan 15314
ishobary@batan.go.id

ABSTRAK

IMPLEMENTASI MIKROKONTROLER BERBASIS ARDUINO SEBAGAI KUNCI (*dongle*) APLIKASI PERANGKAT LUNAK. Perkembangan teknologi informasi sangat pesat, yang diikuti dengan perkembangan pembuatan perangkat lunak program aplikasi, dan perangkat keras pendukungnya. Pemanfaatan kode kunci (*dongle*) yang disisipkan pada pembuatan program aplikasi dapat berfungsi untuk melindungi duplikasi dan penggandaan perangkat lunak ataupun perangkat keras. Telah dilakukan implementasi penambahan kode kunci (*dongle*) sebagai pengaman untuk melindungi duplikasi perangkat lunak aplikasi data logger yang dipasangkan dengan modul perangkat keras akusisi data. Hasil pengujian menunjukkan bahwa perangkat lunak aplikasi yang dilindungi kode kunci (*dongle*) dapat digandakan, tetapi hanya dapat berfungsi ketika dipasangkan dengan modul perangkat keras akusisi data pasangannya

*Kata kunci: mikrokontroler, kunci (*dongle*), aplikasi perangkat lunak,*

ABSTRACT

IMPLEMENTATION OF APPLICATION SOFTWARE DONGLE BASED ON ARDUINO MICROCONTROLLER BOARD. The rapid development of information technology is accompanied by the development of application software packages and their accompanying hardware. The use of the key code (*dongle*) that secures an application software package can protect the package and its associated hardware from duplication and copying. In this work, a *dongle* has been implemented to protect a data logger application software package installed in a data acquisition hardware module. Test results showed that the software packages to which the *dongle* is added can still be duplicated; however, they function only when paired with the intended data acquisition hardware modules they are intended for.

*Keywords: microcontroller, key (*dongle*), application software*

1. PENDAHULUAN

Saat ini, teknologi instrumentasi berkembang sangat pesat, baik dalam desain produk maupun terapannya. Perkembangan ini tidak bisa dilepaskan dari kemajuan iptek secara menyeluruh, khususnya di bidang komponen elektronika serta teknologi informasi dan komunikasi (ICT). Implementasi komponen elektronika modern pada sebuah produk sistem instrumentasi, akan menjadikan dia dapat bekerja secara lebih cepat, lebih akurat, lebih presisi, serta lebih efisien. Keberadaan sistem instrumentasi tidak mungkin dipisahkan dalam perkembangan iptek modern. Pengembangan sebuah sistem instrumentasi akan memberikan solusi bagi ketersediaan sistem pengukuran atau pengendalian secara menyeluruh^[1]. Seiring dengan perkembangan ini, pembuatan aplikasi dan pengembangan modul instrumentasi dan akusisi data banyak diperlukan. Pembuatan aplikasi perangkat lunak dan perangkat akusisi data sebagai perangkat instrumentasi perlu dilindungi dengan menambahkan kode kunci (*dongle*). Kode kunci *dongle* ini berfungsi untuk melindungi program aplikasi ataupun perangkat keras akusisi data yang dibuat. Kode kunci (*dongle*) dapat ditanamkan pada mikrokontroler yang berfungsi tunggal sebagai *dongle* ataupun pada perangkat mikrokontroler yang dijadikan sebagai perangkat akusisi data, dengan menambahkan

fitur *password* untuk mengakses aplikasi perangkat lunak^[2]. Penambahan fitur perlindungan semacam ini untuk mengatasi dan mengurangi pembajakan dan duplikasi perangkat secara ilegal, yang menyebabkan kerugian secara global sangat besar, mencapai 62,7 Milyar USD, pada tahun 2013^[3].

Algoritma dan sistem untuk melindungi perangkat lunak dan perangkat keras telah banyak dikembangkan. Aplikasi perangkat lunak atau berupa file untuk membuka dan menjalankan aplikasi^[4], proteksi file yang akan dijalankan^[5], atau bahkan file dengan format video juga telah dikembangkan untuk dilindungi dengan dienkripsi^[6]. Proteksi dan perlindungan file yang dilakukan dengan menggunakan aplikasi enkripsi yang dijalankan baik dari perangkat flashdisk menggunakan kombinasi *serial XML* dan USB *dongle* ataupun dari *hardsik*, dengan algoritma Rijndael^[7], dengan algoritma El Gamal^[8], ataupun dengan Algoritma Triple DES^[9].

Pada makalah ini disampaikan implementasi mikrokontroler berbasis Arduino Uno sebagai perangkat akuisisi data, ditambahkan kode kunci sebagai *dongle* untuk perangkat lunak aplikasi data *logger* yang dibuat sebagai pasangannya. Implementasi kode kunci (*dongle*) pada perangkat lunak program aplikasi dan modul perangkat keras akuisisi data yang berbasis Arduino, menambahkan kode *handshake* diantara keduanya. Penambahan kode ini memungkinkan program aplikasi perangkat lunak data *logger* masih dapat dilakukan duplikasi dengan di-copy, tetapi hanya bisa dipasangkan dengan modul perangkat keras akuisisi data yang dipasangkan, demikian sebaliknya.

2. DASAR TEORI

Proses pelaksanaan implementasi mikrokontroler berbasis Arduino sebagai kunci (*dongle*) aplikasi perangkat lunak dimulai dengan melakukan kajian dan literasi. Hasil dari kajian dan literasi diaplikasikan pada perancangan program aplikasi data akuisisi dan data *logger*. Implementasi dilakukan dengan menyiapkan perangkat keras data akuisisi dan data *logger*. Program aplikasi pada perangkat keras yang ditanamkan pada mikrokontroler, dibangun dengan menggunakan aplikasi IDE Sketch Arduino versi 1.8.9^[10]. Program aplikasi perangkat lunak akuisisi dan data *logger* dibangun dengan menggunakan program Aplikasi C# pada Microsoft Visual Studio 2019 Community^[11]. Pengujian dilakukan dengan menggunakan perangkat lunak program aplikasi data *logger* yang dibuat dan dihubungkan dengan modul perangkat keras akuisisi data berbasis Arduino. Perangkat keras akuisisi data berbasis Arduino yang digunakan untuk pengujian menggunakan program yang angka serinya (kode angka unik) telah dinotifikasi pada perangkat lunak aplikasi data *logger*, dan yang tanpa angka seri (kode angka unik). Pengujian ini untuk memastikan bahwa program aplikasi data *logger* hanya dapat digunakan pada modul perangkat keras akuisisi data yang angka serinya sama atau dikenali. Pengujian juga dilakukan pada modul perangkat keras dengan menggunakan aplikasi data *logger* lainnya seperti Teraterm^[12]. Pengujian ini dilakukan untuk memastikan bahwa perangkat lunak aplikasi data *logger* dan modul perangkat keras data akuisisi hanya dapat digunakan dan berfungsi ketika keduanya angka serinya sama atau dikenali.

3. TATA KERJA

Proses pelaksanaan implementasi mikrokontroler berbasis Arduino sebagai kunci (*dongle*) aplikasi perangkat lunak dimulai dengan melakukan kajian dan literasi. Hasil dari kajian dan literasi diaplikasikan pada perancangan program aplikasi data akuisisi dan data *logger*. Implementasi dilakukan dengan menyiapkan perangkat keras data akuisisi dan data *logger*. Program aplikasi pada perangkat keras yang ditanamkan pada mikrokontroler, dibangun dengan menggunakan aplikasi IDE Sketch Arduino versi

1.8.9^[10]. Program aplikasi perangkat lunak akuisisi dan data *logger* dibangun dengan menggunakan program aplikasi C# pada Microsoft Visual Studio 2019 Community^[11]. Pengujian dilakukan dengan menggunakan perangkat lunak program aplikasi data *logger* yang dibuat dan dihubungkan dengan modul perangkat keras akuisisi data berbasis Arduino. Perangkat keras akuisisi data berbasis Arduino yang digunakan untuk pengujian menggunakan program yang angka serinya (kode angka unik) telah dinotifikasi pada perangkat lunak aplikasi data *logger*, dan yang tanpa angka seri (kode angka unik). Pengujian ini untuk memastikan bahwa program aplikasi data *logger* hanya dapat digunakan pada modul perangkat keras akuisisi data yang angka serinya sama atau dikenali. Pengujian juga dilakukan pada modul perangkat keras dengan menggunakan aplikasi data *logger* lainnya seperti Teraterm^[12]. Pengujian ini dilakukan untuk memastikan bahwa perangkat lunak aplikasi data *logger* dan modul perangkat keras data akuisisi hanya dapat digunakan dan berfungsi ketika keduanya angka serinya sama atau dikenali.

4. HASIL DAN PEMBAHASAN

4.1. Alat dan Bahan

Peralatan yang digunakan pada implementasi ini adalah *toolset* elektronik, digunakan untuk merakit modul perangkat keras akuisisi data. *Toolset* elektronik, berisikan aneka obeng dan tang, solder, dan kelengkapannya untuk mematri, integrasi dan instalasi komponen elektronik. Modul perangkat keras akuisisi data berisikan komponen utama modul Arduino UNO, modul LCD 4 baris 20 kolom, modul catu daya, lampu indikator, beberapa konektor dan *switch*. Perangkat Komputer dengan sistem operasi Microsoft Windows 7 atau yang lebih baru.

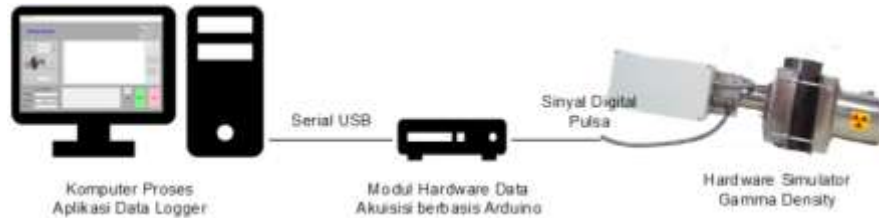
Perangkat lunak yang diperlukan dan diinstal adalah perangkat lunak Microsoft Visual Studio 2016 Community atau yang lebih baru, untuk melakukan pembuatan program aplikasi data *logger*. Perangkat lunak IDE Sketch versi 1.8.9, atau versi lainnya digunakan untuk melakukan pemrograman modul perangkat keras. Perangkat lunak program aplikasi Teraterm digunakan sebagai data *logger* pembanding pada saat pengujian. Alat ukur seperti *Digital multimeter*, *osilloscope*, untuk melakukan pengujian dan mengamati pulsa. *Fuction generator* sebagai perangkat simulasi pembangkit pulsa untuk memberikan sinyal ke modul perangkat keras akuisisi data.

4.2. Implementasi Sistem

Implementasi mikrokontroler berbasis Arduino sebagai kunci (*dongle*) pada perangkat lunak aplikasi data *logger* dan modul data akuisisi, yang digunakan pada simulasi gamma density. Blok diagram implementasi diperlihatkan seperti pada gambar 1. Implementasi penggunaan kunci (*dongle*) diaplikasikan pada perangkat lunak aplikasi data *logger* dan modul perangkat keras data akuisisi. Bagian utama terdiri dari komputer proses, yang berfungsi untuk menjalankan aplikasi data *logger* yang dibuat dan telah disisipkan code kunci (*dongle*) yang kodenya harus dapat diterima oleh modul perangkat keras akuisisi data. Aplikasi data *logger* dijalankan pada komputer dengan sistem operasi Microsoft Windows 7 atau yang lebih baru. Spesifikasi komputer yang digunakan adalah tersedia port USB (*Universal Serial Bus*) yang berfungsi sebagai port komunikasi data antara komputer proses dengan perangkat data akuisisi.

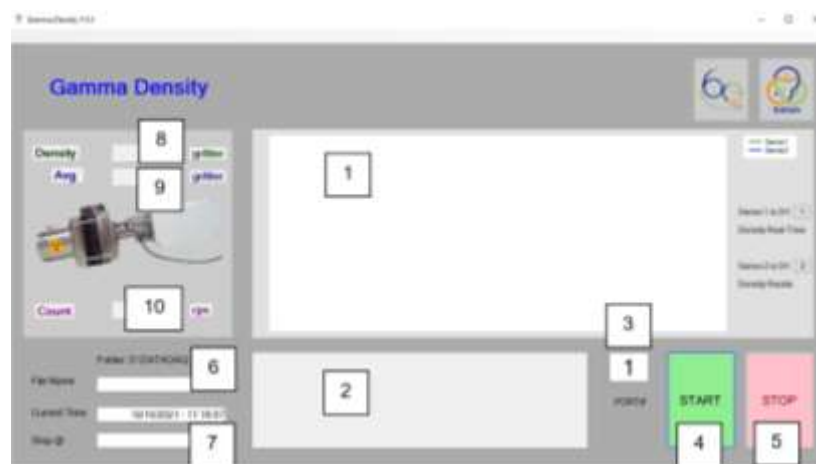
Perangkat keras data akuisisi, dengan komponen utama modul mikrokontroler Arduino Uno. Modul ini berfungsi untuk melakukan pembacaan pulsa dari perangkat keras *gamma density*, yang selanjutnya data dilakukan perhitungan dan hasil perhitungannya dikirimkan ke komputer untuk ditampilkan dan disimpan dalam bentuk file. Pada modul ini ditanamkan program hitungan cacah pulsa yang merepresentasikan kerapatan sampel. Pada program ini disisipkan *program code* kunci (*dongle*) yang kodenya diminta untuk dicocokkan dengan kunci (*dongle*) pada program aplikasi

data *logger* pada komputer proses. Pada pengujian dan simulasi perangkat keras *simulator gamma density* digantikan dengan *function generator* untuk membangkitkan sejumlah pulsa digital. Perangkat keras *simulator gamma density* dibahas pada makalah yang ditulis secara terpisah.



Gambar 1. Blok diagram implementasi kunci (*dongle*) pada aplikasi data *logger* dan modul data akuisisi berbasis Arduino perangkat simulator gamma density.

Kode kunci (*dongle*) program disisipkan untuk mengenali diantara program perangkat lunak aplikasi data *logger* dengan perangkat keras akuisisi data (*handshake*). Tujuan dari penambahan kode ini adalah memastikan bahwa perangkat lunak aplikasi data *logger* harus dipasangkan dengan perangkat keras data akuisisi yang telah ditentukan/dipasangkan. Sebaliknya perangkat keras data akuisisi hanya dapat diakses dengan program aplikasi data *logger* yang sudah dipasangkan. Dengan demikian program perangkat lunak aplikasi data *logger* dan perangkat keras data akuisisi harus satu paket, sehingga sulit untuk dilakukan duplikasi.



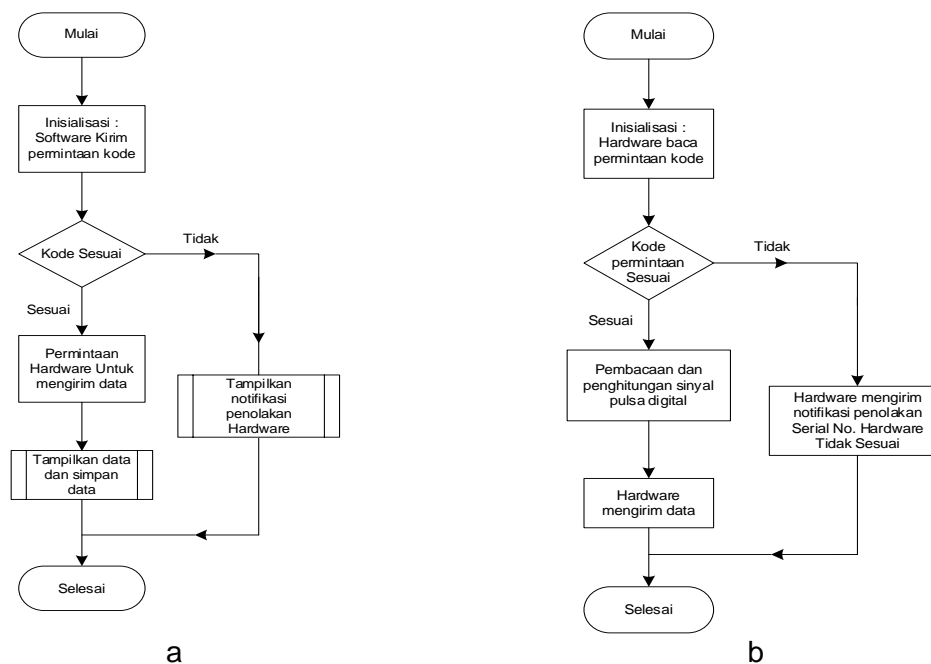
Gambar 2. *Printscreen* antar muka program perangkat lunak data *logger*, yang digunakan pada simulasi *gamma density meter*.

Tampilan aplikasi program perangkat lunak aplikasi data *logger* ditampilkan seperti pada gambar 2. Tampilan antar muka dengan nama file GD.EXE, hanya terdapat satu tampilan saja, yang didalamnya terdapat 10 bagian, yaitu :

1. Menampilkan jendela grafik *trend* hasil pengukuran *real time* dan rerata, yang ditampilkan setiap detiknya. Tampilan *trend* menginformasikan grafik dalam 120 detik terakhir.
2. Jendela pesan menampilkan pesan bila terjadi *error* ataupun kesalahan pada saat *logging* data. Bila tidak terdapat kesalahan jendela pesan menampilkan waktu dan angka hasil pengukuran yang dikirimkan dari modul perangkat lunak akuisisi data. Pesan kesalahan akan ditampilkan misalnya bila modul perangkat keras akuisisi data tidak sesuai dengan kode kunci (*dongle*) dari perangkat lunak program aplikasi data *logger*.

3. Menampilkan jendela *port* komunikasi modul perangkat keras akuisisi data dengan komputer yang digunakan untuk menjalankan program aplikasi data *logger*.
4. Tombol *virtual START* untuk mulai menjalankan program aplikasi data *logger*, menghubungkan dengan modul perangkat keras data akuisisi.
5. Tombol *virtual STOP* untuk berhenti menjalankan program aplikasi data *logger*, memutus komunikasi modul perangkat keras data akuisisi.
6. Jendela kolom nama file, secara *default* nama file akan disimpan dengan format tanggal/bulan/tahun jam menit detik. Nama file dapat diberikan sebelum tombol start ditekan, bila diinginkan dengan nama file sesuai yang dikehendaki. File hasil *logger* disimpan dengan format dan ekstensi *.csv.
7. Jendela kolom yang menampilkan waktu saat proses logging data diberhentikan dengan menekan tombol STOP.
8. Jendela kolom yang menampilkan angka hasil pengukuran, *real time*.
9. Jendela kolom yang menampilkan angka hasil pengukuran, rerata 10 pengambilan data terakhir.
10. Jendela kolom yang menampilkan angka hasil pengukuran, nilai cacah yang diterima, data ini belum dilakukan perhitungan, dan konversi ke besaran satuan pengukuran, sehingga digunakan satuan cps (cacah per detik)

Diagram alir program aplikasi perangkat lunak data *logger* dan program akuisisi ditampilkan seperti pada Gambar 3.a. dan 3.b.



Gambar 3. Diagram alir program aplikasi *handshake*, a. program pada perangkat lunak aplikasi data *logger* dan b. program pada perangkat keras modul akuisisi data

Listing program 1, menampilkan cuplikan code pemrograman yang dituliskan pada aplikasi perangkat lunak data *logger*, yang berfungsi sebagai inisialisasi *port* komunikasi *serial*. Selanjutnya pada program ini disisipkan kode yang dikirimkan ke perangkat keras. Inisialisasi pertama kali dikirimkan setelah koneksi dinotifikasikan antara perangkat lunak aplikasi data *logger* menyambungkan port yang sesuai, misalnya perangkat keras akuisisi data dikenali berada pada *port* 1, maka tombol *virtual START* segera ditekan. Setelah tombol ditekan perangkat lunak aplikasi data *logger* akan mengirimkan kode sebagai kunci (*dongle*) yang dikirimkan ke modul perangkat keras akuisisi data. Bila kode kunci (*dongle*) diterima (sama), segera setelahnya dikirimkan kode permintaan untuk mengirimkan data. Data yang dikirimkan dari modul

perangkat keras akuisisi data, dengan urutan format data tertentu. Data yang telah dikirimkan selanjutnya akan ditampilkan sebagai data numerik dan ditampilkan dalam bentuk grafik pada program aplikasi data *logger*, selain disimpan sebagai file dengan format file *.csv.

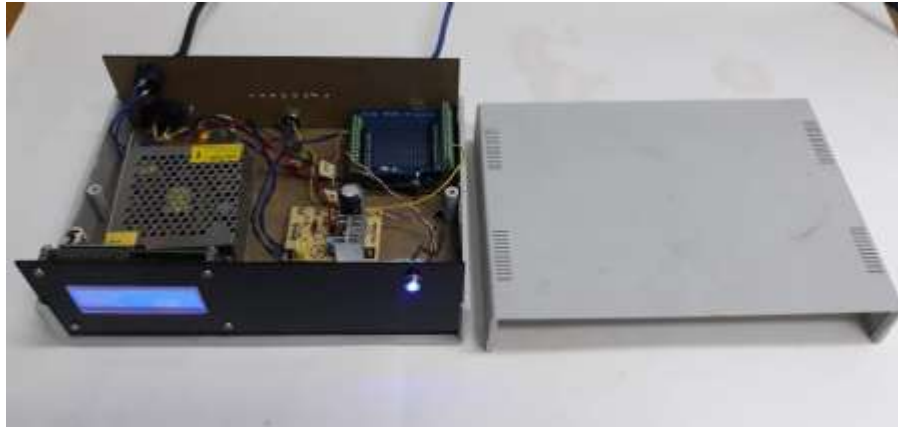
Listing Program 1. Pengiriman kode *serial* sebagai *handshake* dan pembacaan data, bilamana kode (*dongle*) yang dikirim sesuai dengan kode (*dongle*) pada perangkat keras data akuisisi :

```
///-- SERIAL PORT TRANSMITE & RECEIVE -----  
private void serialPort1_DataTransmitted(object sender, System.IO.Ports.SerialDataTransmittedEventArgs e)  
{  
    case 'AA':  
        this.Invoke(new EventHandler(BeginTx));  
        break;  
    case 'a':  
        paket_1 = serialPort1.WriteTo('#000*datagamma*ok'); // kirim kode  
        paket_1 = serialPort1.WriteTo("\r");  
        break;  
    case 'xx':  
        this.Invoke(new EventHandler(EndTx));  
        break;  
private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)  
{  
    {  
        kode = serialPort1.ReadChar(); // baca kode  
        ax = serialPort1.ReadChar(); // baca data awal  
        {  
            if (startF == 0) { if (kode == '#001*datagamma*ok') { startF = 1; } }  
            break;  
            {  
                If else(startF == 0) { if (kode != '#001*datagamma*ok') { startF = 1; } }  
                break;  
                if (startF == 0) { if (ax == 'A') { startF = 1; } } //urut baca  
                {  
                    switch (ax) // if (check1 = true)  
                    {  
                        case 'A':  
                            this.Invoke(new EventHandler(BeginRx));  
                            break;  
                        case 'a':  
                            paket_1 = serialPort1.ReadTo("\r");  
                            this.Invoke(new EventHandler(DspCh1));  
                            break;  
                        case 'b':  
                            paket_2 = serialPort1.ReadTo("\r");  
                            this.Invoke(new EventHandler(DspCh2));  
                            break;  
                        case 'c':  
                            paket_3 = serialPort1.ReadTo("\r");  
                            this.Invoke(new EventHandler(DspCh3));  
                            break;  
                        case 'x':  
                            this.Invoke(new EventHandler(EndRx));  
                            break;  
                    }  
                }  
            }  
        }  
    }  
} //=====
```

Modul perangkat keras akuisisi data, setelah terhubung ke perangkat lunak aplikasi data *logger* dan notifikasi kode kunci (*dongle*) diterima, segera data dikirimkan untuk setiap detiknya. Sebaliknya bila kode kunci (*dongle*) berbeda, proses akuisisi data dan pengiriman tidak dilanjutkan, dan proses berhenti. *Listing program 2* menampilkan cuplikan notifikasi dan *handshake*, yang ditambahkan pada program perangkat keras akuisisi data. Gambar 3 menampilkan modul perangkat keras akuisisi data, terdiri dari beberapa bagian dan komponen utama pendukungnya.

Pada proses selanjutnya modul perangkat keras akuisisi data, yang telah selesai dilakukan instalasi pengkabelan dan telah ditanamkan program akuisisi data, dilakukan

pengujian. Pengujian yang dilakukan adalah mengintegrasikan antara perangkat lunak aplikasi data *logger* dengan modul perangkat keras akuisisi data.



Gambar 3. Modul perangkat keras akuisisi data

Listing Program 2. Notifikasi pengiriman kode serial sebagai *handshake* yang dikirimkan perangkat keras data akuisisi :

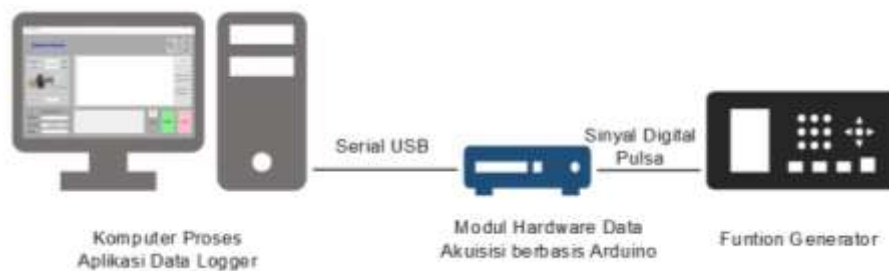
```
// -----  
// tampil ke komputer  
// format c#  
if (Serial.available())  
{  
    char serial_number = Serial.read();           //Baca karakter serial  
    incomingByte = Serial.read();  
    if Serial.read(kode == '#000*datagamma*ok'); //kode dicocokkan  
        Serial.print("#001*datagamma");         // kirim kode unik dilanjut  
        Serial.print("xAa");                   // rerata //Serial.println (ch_00);  
        Serial.println(rerata);  
        Serial.print("b");                       // utk flow meter //Serial.println (ch_01);  
        Serial.println(" 0 ");  
        Serial.print("c");                       // nilai real time //Serial.println (ch_02);  
        Serial.println(onko_keridewe);  
    Else if Serial.read(kode != '#000*datagamma*ok');  
        Serial.println("#SERIAL KEY HARDWARE TIDAK SESUAI"); //kirim pesan ditolak  
        Serial.println("#HUBUNGI ADMINISTRATOR/OPERATOR");  
        void tutup()  
    }  
// -----
```

Pengujian dilakukan untuk mengetahui bahwa perangkat lunak aplikasi data *logger* hanya dapat berfungsi bila dipasangkan dengan perangkat keras data akuisisi yang dengan kode kunci (*dongle*) yang sama, begitu juga sebaliknya. Skenario pengujiannya adalah sebagai berikut :

1. Pengujian ke 1, pengujian dilakukan dengan perangkat lunak aplikasi data *logger* (selanjutnya disebut aplikasi *Gamma Density*) dengan dipasangkan dengan perangkat keras modul data akuisisi (selanjutnya disebut DAQ01). Aplikasi *Gamma Density* dengan perangkat keras modul DAQ01, dengan kode kunci (*dongle*) sama.
2. Pengujian ke 2, pengujian dilakukan dengan perangkat lunak aplikasi data *logger* (selanjutnya disebut aplikasi *Gamma Density*) dengan dipasangkan dengan perangkat keras modul data akuisisi (selanjutnya disebut DAQ02). Perangkat keras modul DAQ02, tidak menggunakan/tanpa kode kunci (*dongle*).
3. Pengujian ke 3, pengujian dilakukan dengan perangkat lunak aplikasi *Teraterm* sebagai aplikasi data *logger* (selanjutnya disebut aplikasi *Teraterm*) dipasangkan dengan perangkat keras modul data akuisisi (selanjutnya disebut DAQ01). Aplikasi *Teraterm* dengan perangkat keras modul DAQ01, dengan kode kunci (*dongle*).

- Pengujian ke 4, pengujian dilakukan dengan perangkat lunak aplikasi Teraterm sebagai aplikasi data *logger* (selanjutnya disebut aplikasi Teraterm) dipasangkan dengan perangkat keras modul data akuisisi (selanjutnya disebut DAQ02). Perangkat keras modul DAQ02, tidak menggunakan/tanpa kode kunci (*dongle*).

Hasil pengujian ditampilkan pada Sub Bab 4.3. Hasil dan pembahasan, pada pembahasan selanjutnya. Konfigurasi pengujian seperti ditampilkan pada blok diagram gambar 5. Perangkat keras *simulator gamma density* digantikan dengan menggunakan *fuction generator*. Modul perangkat keras modul DAQ02, secara fisik sama dengan modul DAQ01, yang membedakan adalah code program yang ditanamkan pada mikrokontroler tidak disisipi dengan *code program handshake* antara program aplikasi perangkat lunak data *logger* dengan perangkat lunak akuisisi data.



Gambar 4. Konfigurasi pengujian *handshake*, aplikasi perangkat lunak data *logger* dan modul perangkat keras akuisisi data

4.3. Hasil dan Pembahasan

Data hasil pengujian ditabelkan seperti pada Tabel 3. Tanda OK (Y) menginformasikan bahwa antara perangkat lunak aplikasi data *logger* dapat berfungsi. Tanda *Fail* (X), menginformasikan bahwa data yang dikirim tidak dapat disimpan dan terbaca pada aplikasi perangkat lunak.

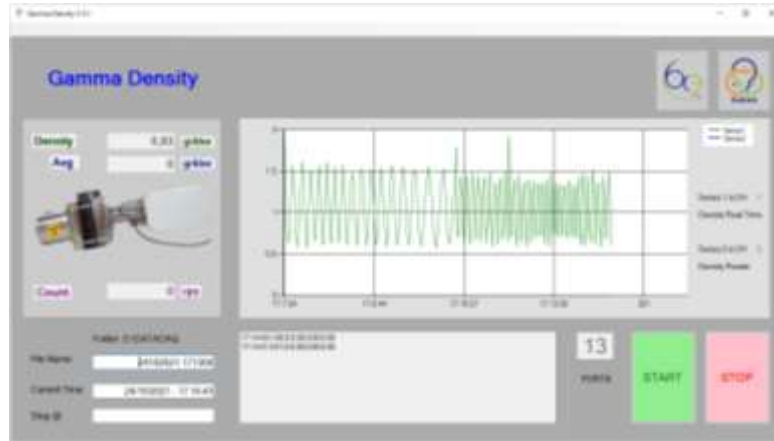
Tabel 3. Hasil Pengujian aplikasi perangkat lunak aplikasi data *logger* dengan modul perangkat keras data akuisisi.

No.	Pengujian	Hasil	Keterangan
1.	Metoda Pengujian ke 1	OK (Y)	Perangkat dapat berfungsi, data dapat disimpan dan ditampilkan
2.	Metoda Pengujian ke 2	Error (X)	Perangkat tidak berfungsi, data tidak tersimpan
3.	Metoda Pengujian ke 3	Error (X)	Perangkat tidak berfungsi, data tidak tersimpan
4.	Metoda Pengujian ke 4	OK (Y)	Perangkat dapat berfungsi, data dapat disimpan

Hasil pengujian pada dengan metoda pengujian ke 1, berhasil dan tidak ada pesan kesalahan selama pengambilan dan akuisisi data. Hal ini karena antara perangkat lunak program aplikasi data *logger* dengan modul perangkat keras akuisisi datam kode kunci (*dongle*) dapat diterima dan sesuai, sehingga akuisisi data, pengiriman data dan data *logger* dapat berlangsung. Tampilan antar muka saat berlangsungnya data *logger* ditampilkan seperti pada gambar 5

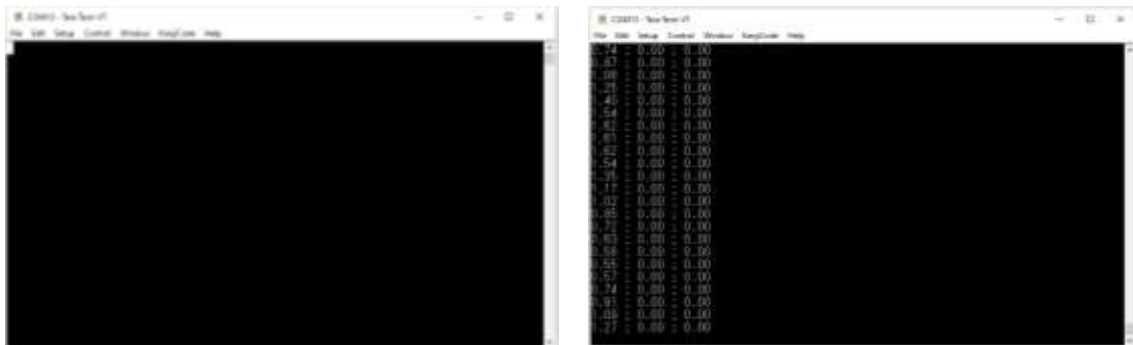
Pada pengujian dengan skenario pengujian ke 2, aplikasi perangkat lunak data *logger*, tidak menerima notifikasi kode kunci (*dongle*) dari modul perangkat keras akuisisi data, sehingga setelah dihubungkan, data tetap tidak diterima, dan data yang dikirimkan tidak dapat ditampilkan. Pada jendela pesan masih kosong, tidak ada pesan atau karakter yang ditampilkan. Hal yang sama ditampilkan pada sekenario pengujian

3, layar program aplikasi Teraterm terlihat kosong, karena tidak ada data yang dikirimkan. Hal ini dikarenakan pada modul perangkat keras akuisisi data, program tidak mengirimkan data kode kunci (*dongle*), sehingga proses akuisisi data berhenti. Gambar ditampilkan seperti pada gambar 6. a.



Gambar 5. *Printscreen* antar muka program perangkat lunak aplikasi data *logger*, saat terhubung ke modul perangkat keras akuisisi data.

Modul perangkat keras akuisisi data yang program di dalamnya tidak disisipkan kode kunci (*dongle*), data yang dikirimkan dapat diterima dan terbaca oleh perangkat lunak aplikasi Teraterm. Dengan aplikasi ini data yang dikirimkan dapat dilakukan proses simpan dan olah data. Data yang disimpan dalam bentuk file format *.dat, atau *.csv. Gambar 6.b. menampilkan proses ini.



a. dengan kode kunci (*dongle*)

b. tanpa disisipkan kode kunci (*dongle*)

Gambar 6. *Printscreen* tampilan aplikasi Teraterm sebagai data *logger* yang dipasangkan dengan modul perangkat keras akuisisi data

Perbedaan program aplikasi data *logger* yang didalamnya disisipkan kode kunci (*dongle*) adalah terdapat jendela tampilan waktu, dan *trend* grafik yang menyajikan data *trend* dalam 120 detik terakhir. Program aplikasi ini hanya dapat berjalan dapat berfungsi sebagai data *logger* dan menampilkan *trend* grafik, ketika dipasangkan dengan modul perangkat keras akuisisi data yang didalamnya disisipkan kode kunci (*dongle*) yang sesuai dengan kode kunci (*dongle*) yang ditanamkan pada perangkat lunak aplikasi data *logger*. Dengan ditambahkannya kode kunci (*dongle*) pada program aplikasi data *logger* dan modul perangkat keras akuisisi data, memastikan bahwa program aplikasi hanya dapat berfungsi bila dipasangkan. Bila program aplikasi perangkat lunak data *logger* digandakan dan digunakan pada komputer lain, tidak akan

berfungsi bila menggunakan modul perangkat keras akusisi data yang bukan pasangannya.

5. KESIMPULAN

Telah dilakukan implementasi dengan menambahkan kode kunci (*dongle*) pada program perangkat lunak aplikasi data *logger* dan modul perangkat keras akusisi data. Tujuan dari penambahan kode program ini adalah untuk melindungi pembajakan atau duplikasi baik perangkat lunak aplikasi data *logger* ataupun modul perangkat keras akusisi data. Pada pengujian yang dilakukan, perangkat lunak aplikasi data *logger* hanya dapat berfungsi bila dipasangkan dengan modul perangkat keras akusisi data yang didalamnya telah disisipkan kode kunci (*dongle*). Kode kunci (*dongle*) yang ditanamkan pada pembuatan aplikasi perangkat lunak dapat dilindungi dari pembajakan atau duplikasi ilegal dengan memasangkan dengan perangkat keras dengan komponen utama modul mikrokontroler, yang didalamnya disisipkan kode kunci (*dongle*) sebagai *handshake* dan notifikasi keamanannya.

6. DAFTAR PUSTAKA

- [1] Santoso DR, M.Si., *Perkembangan teknologi instrumentasi jadi peluang bagi peneliti* <https://www.antaraneews.com/berita/1856992/ahli-perkembangan-teknologi-instrumentasi-jadi-peluang-bagi-peneliti> diakses 31 Agustus 2021
- [2] Chandra MDE, *USB Dongle/USB Key Berbasis ATtiny2313 Sebagai Kunci Pengaman Software Aplikasi.*, <https://telinks.wordpress.com/2013/02/27/usb-dongle-usb-key-berbasis-attiny2313-sebagai-kunci-pengaman-software-aplikasi/> diakses 6 Agustus 2021
- [3] Anonymous, *The Dangerous World of Software Piracy.*, http://www.securemetric.com/wp-content/uploads/2016/01/SecureDongle_infography_2015.jpg ., diakses 3 Agustus 2021.
- [4] Wardiana W, Heryana A, 2008, *Penerapan Algoritma Kriptografi Kunci Publik Sebagai Pengamanan Sistem Distribusi Perangkat Lunak LIPIRISM@*, Prosiding Seminar Nasional Teknoin, Bidang Teknik Informatika., ISBN: 978-979-3980-15-7. Yogyakarta, 22 November 2008
- [5] Kadyanan IGAGA, Soetam Rizky Wicaksono, 2007, *Proteksi File Pada Pocket PC Menggunakan Algoritma Rijndael dengan Kombinasi Serialisasi XML dan Kunci Device ID*, Prosiding Seminar Nasional Sistem dan Informatika., Bali, 16 November 2007, SNSI07-019.
- [6] Kadyanan IGAGA, 2014, *Perancangan Sistem Proteksi File Video Dengan Algoritma AES (Advance Encryption Standard)*, Prosiding Seminar Nasional Teknologi Informasi Dan Aplikasinya, Bali, 17 Oktober 2014.
- [7] Azani, Dion. 2006, *Tugas Akhir: Proteksi Data Dengan Kombinasi Serialisasi Xml Dan USB Dongle Menggunakan Algoritma Rijndael*. STIKOM. Surabaya. 2006.
- [8] Rahayu RP, Fauzi A, 2020, *Pemanfaatan USB Flashdisk Sebagai Kunci Pada Keamanan Data Dengan Penerapan Algoritma Elgamal*, Jurnal Teknik Informatika Kaputama (JTIK) Vol. 4 , No. 2, Juli 2020. P-ISSN: 2548-9704, E-ISSN: 2686-0880.
- [9] Chan AS, Permana, dan Munthe PG. 2014, *Perancangan Aplikasi Pengamanan File Dengan Memanfaatkan USB Flashdisk Sebagai Kunci Menggunakan Algoritma Triple DES*. Jurnal Pelita Informatika Budi Darma., Vo;. VIII., No. 3., Desember 2014., pp. 30-36., ISSN: 2301-9425.
- [10] <https://www.arduino.cc/en/software> , diakses 4 Juli 2021.
- [11] <https://visualstudio.microsoft.com/vs/community/> diakses 1 Maret 2021.
- [12] <https://ttssh2.osdn.jp/index.html.en> , diakses 9 Juli 2021.