

## RANCANG BANGUN SISTEM PENGUNCI PINTU OTOMATIS MENGUNAKAN PENDETEKSI GETARAN KETUKAN

Farid Fadilah<sup>1</sup>.,Paniran<sup>2</sup>, Giri Wahyu W.3<sup>1</sup>

### ABSTRAK

Penelitian ini diharapkan dapat meningkatkan keamanan dan kemudahan bagi pengguna dari sistem otomatis pengunci pintu menggunakan kode ketukan. Sinyal analog yang dihasilkan dari sensor saat diberikan ketukan akan masuk melalui pin ADC(analog to digital converter) sistem minimum Arduino dengan IC mikrokontroler ATmega328P. Sinyal analog dari sensor akan dikonversi dalam bentuk digit biner (digital) yang akan memberikan logika pada beberapa beban sebagai indikasi termasuk mengaktifkan driver L298N untuk menggerakkan motor DC-micro untuk membuka pintu yang terkunci atau sebaliknya mengunci pintu yang terbuka.

Hasil pengujian diperoleh bahwa kode ketukan yang diberikan pada sistem ini terkait dengan jumlah ketukan dan nilai waktu interval atau jarak antar setiap-ketukan dan disimpan dalam type data Array yang membentuk sebuah ritme/irama ketukan tertentu yang di gunakan dalam program. Nilai interval ketukan ini (password ketukan) akan disimpan terlebih dahulu dalam flash memory mikrokontroler. Dari 30 kali percobaan yang dilakukan ketika sensor piezo mendeteksi ada beberapa ketukan, sistem akan mulai bekerja dengan melakukan mencacah interval antar ketukan, jika ritme/irama ketukan sesuai dengan yang telah tersimpan sebelumnya maka sistem penguncian/membuka pintu akan aktif. Sebaliknya adapun jika terjadi kesalahan dalam pemberian masukan kode ketukan tadi diluar dari interval ritme/ irama yang telah ditentukan, maka sistem pengunci/membuka tidak bekerja.

**Kata kunci :** Piezoelektrik, Piezo buzzer, Arduino, Mikrokontroler,

### ABSTRACT

This research is expected to improve security and convenience for users of the automatic door open/locking system using code beats. The resulting analog signal from the sensor when given a knock will go through the ADC pin (analog to digital converter) with minimum system Arduino microcontroller IC ATmega328P. The analog signal from the sensor will be converted in the form of binary digits (digital) which will provide logic in some of the load as indications including activating L298N driver to drive a DC motor-micro to open a locked door or otherwise lock the door open.

From the test results obtained that beats the code given in the system is related to the number of beats and the value of the time interval or distance between every-beats and stored in the array of data types that make up a rhythm / beat certain beats are in use in the program. This beats the interval value (password beats) will be stored in advance in the flash memory microcontroller. Of 32 experiments carried out when the piezo sensor detects several beats, the system will start working by chopping interval between beats, when the rhythm / rhythm beats in accordance with the previously saved, the system locking / opening the door will be active. Conversely while in case of errors in the feedback code was out of intervalritme beats / rhythms that have been determined, then the system lock / unlock does not work.

**Keywords :** Piezoelectric, Buzzer Piezoelectric, Arduino, Microcontroller

### PENDAHULUAN

Perkembangan teknologi yang berharga di dunia modern saat ini salah satunya adalah mikrokontroler. Mikrokontroler merupakan komputer *portabel* dan murah yang dapat digunakan untuk mencapai beberapa macam tugas di bidang pengolahan data, logika, dan sistem kontrol. Dengan demikian,

komputer kecil ini dapat merampingkan dan mengotomatisasi tugas-tugas sehari-hari, dan menciptakan kehidupan yang lebih mudah bagi manusia. Ada beberapa aplikasi mikrokontroler salah satunya adalah untuk meningkatkan keamanan.

Dengan mengamati bangunan atau gedung - gedung seperti apartemen dan perusahaan - perusahaan besar ditemukan

<sup>1</sup> Jurusan Teknik Elektro Fakultas Teknik Universitas Mataram, Nusa Tenggara Barat, Indonesia  
Email : [giriwahyuwiriasto@gmail.com](mailto:giriwahyuwiriasto@gmail.com)

bahwa sementara pintu masuk dilakukan melalui sistem *unlocking* pintu otomatis menggunakan kode untuk meningkatkan keamanan. Ada beberapa cara modern untuk *system unlocking* pintu otomatis seperti RF ID sistem, masukan tombol angka, atau bahkan retina dan *scanner* jempol. Sistem *unlocking* seperti ini relatif mahal tapi dalam segi keamanan terjamin. Dengan memanfaatkan mikrokontroler dan sensor getaran akan tereksplorasi suatu sistem keamanan dengan metode *secret knock* atau kode ketukan yang dapat diaplikasikan untuk mengakses pintu masuk ke fasilitas rahasia, ruang pertemuan kelompok, rumah ataupun kamar pribadi.

Sistem penguncian pintu ini akan efisien dikalangan Masyarakat khususnya pada tempat tinggal penyewaan untuk para mahasiswa pendatang karena pada dasarnya bahan ataupun komponen yang digunakan untuk merancang relatif murah dan dalam segi penggunaannya sangat mudah serta dapat memodifikasi ketukan seperti yang diinginkan untuk meningkatkan keamanan. Maka dari itu pada tugas akhir ini akan dirancang suatu alat yang digunakan untuk menjalankan beban menggunakan kode ketukan yang diaplikasikan pada sistem penguncian pintu yang diharapkan dapat meningkatkan keamanan ruangan.

Dari latar belakang yang telah disajikan diatas, perlu dilakukan suatu rumusan masalah yang akan dibahas penulis dalam penelitian ini yaitu:

1. Bagaimana merancang rangkaian pendeteksi kode ketukan untuk menggerakkan motor DC?
2. Bagaimana merancang sistem yang digunakan untuk mengisi ketukan baru sebagai kode ketukan untuk pembuka kunci?
3. Bagaimana mendesain sistem kendali yang digunakan untuk mengoperasikan sistem penguncian pintu otomatis berdasarkan setting yang sudah ditetapkan?

Untuk lebih mengarahkan penelitian dan menghindari meluasnya pembahasan maka penulis membatasi permasalahan yaitu :

1. Menggunakan minimum sistem Arduino dengan mikrokontroler ATmega328P.
2. Menggunakan motor servo micro yang dimodifikasi ( $V_{cc} = 12\text{ V}$ ,  $Stall\ torque (4.8\text{ V}) = 1\text{ kg/cm}$ ,  $output\ speed (12\text{ V}) = 300\text{ RPM}$ ).
3. Sensor Piezo Buzzer yang digunakan untuk mendeteksi kode ketukan (nilai amplitudo,

frekuensi, *phase*, periode getaran tidak berpengaruh).

4. Menggunakan dua *input* kode ketukan, satu kode untuk membuka kunci dan satu kode lagi untuk mengunci.
5. Menggunakan *visual basic* 6.0 untuk menampilkan sinyal grafik dari sensor.
6. Menggunakan nilai interval yang disetting dalam pemrograman (bukan waktu nyata dalam satuan mili detik) untuk membaca kode ketukan. 3

**Mikrokontroler ATmega328P.** Perkembangan teknologi telah mendorong dengan pesat kemajuan perkembangan dunia elektronika khususnya dunia mikroelektronika. Dengan adanya penemuan silikon maka bidang ini telah memberikan sumbangan yang amat berharga bagi perkembangan teknologi modern. Atmel sebagai salah satu vendor yang mengembangkan dan memasarkan produk mikroelektronika telah menjadi suatu teknologi standar bagi para *desainer* sistem elektronika masa sekarang. Dengan perkembangan terakhir yaitu generasi AVR (*Alf and Vegard's Risc processor*) maka para *desainer* sistem elektronika telah diberikan suatu teknologi yang memiliki kapabilitas yang amat maju namun dengan biaya ekonomis yang cukup minimal.

Mikrokontroler AVR memiliki arsitektur RISC 8 bit di mana semua instruksi dikemas dalam kode 16-bit (16-bits *word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Dibandingkan dengan instruksi ASM51 yang membutuhkan 12 siklus *clock*. Tentu saja ini terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda, yang satu RISC sedangkan yang lain CISC. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka hampir sama.

Salah satu Mikrokontroler AVR produk Atmel yaitu ATmega328P. Mikrokontroler ATmega328P banyak terdapat dipasaran, murah dan juga mempunyai fasilitas yang lengkap.

ATmega328P adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATmega328P mempunyai *throughput* mendekati 1 MIPS per MHz membuat *disainer* sistem untuk

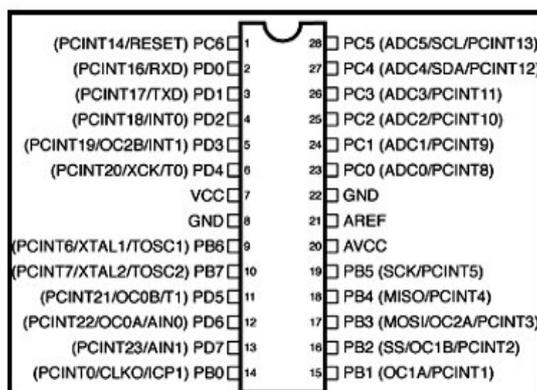
mengoptimasi 5 konsumsi daya versus kecepatan proses. Blok diagram dari mikrokontroler dapat dilihat pada gambar 1

Mikrokontroler ATmega328P memiliki sejumlah spesifikasi data teknis sebagai berikut :

1. Saluran IO sebanyak 23 buah, yaitu *Port B*, *Port C* dan *Port D*
2. ADC 10 bit 8 *channel*
3. 6 *Channels* PWM
4. *Real Time Counter* dengan *Oscillator* terpisah
5. 32 *register*
6. *Watchdog Timer* dengan *oscilator internal*
7. Memori Flash 32 KB (ATmega328P), sekitar 0.5 KB digunakan oleh *bootloader*
8. SRAM sebanyak 2 KB
9. EEPROM 1 KB
10. Sumber *Interrupt* internal dan eksternal
11. *Master/Slave SPI (Serial Pheripheral Interface)*.
12. Komparator analog
13. *Port USART (Universal Shynchronous Ashynchronous Receiver Transmitter)*

**Konfigurasi Pin ATmega328P.** Konfigurasi Pin Mikrokontroler ATmega328P dengan kemasan 28-pin DIP (*dual in-line package*) dapat dilihat pada Gambar 1

Untuk memaksimalkan performa dan paralelisme, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Arsitektur CPU dari AVR ditunjukkan oleh gambar 2 Instruksi pada memori program dieksekusi dengan *pipelining single level*. Pada saat sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program.

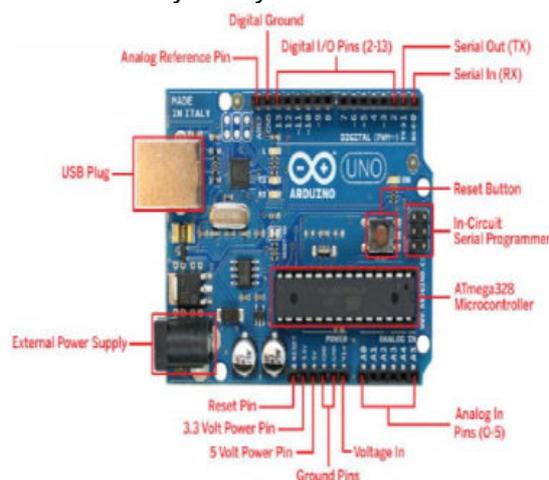


Gambar 1 Konfigurasi Pin Mikrokontroler ATmega328P [2]

**Arduino.** Arduino adalah sebuah kit elektronik open source yang digunakan sebagai pengendali mikro yang dapat diprogram.

Arduino biasa digunakan untuk membuat suatu *prototype* piranti karena sifatnya yang fleksibel dan mudah digunakan baik *software* atau *hardware*-nya. Arduino ini dikembangkan oleh sekumpulan orang dari kalangan dosen dan mahasiswa di Italy. Tujuannya untuk mempermudah dalam hal membangun sebuah perancangan atau mengembangkan proyek menggunakan *mikrokontroler*.

**Kit Arduino.** Arduino menggunakan chip dengan jenis Atmel AVR. Chip yang dipakai untuk Arduino pada gambar dibawah ini adalah ATmega328P. Chip ini memiliki fasilitas EEPROM, SRAM, Serial, Analog Input, Digital I/O, I2C, SPI dan fitur internal lainnya yang memungkinkan Arduino ini bisa berkomunikasi dengan berbagai rangkaian elektronika lainnya, misalnya sensor, *display*, *actuator*, dan sebagainya. Arduino terdiri dari *hardware* berupa papan rangkaian yang sudah diberi antarmuka USB sehingga bisa dihubungkan langsung dengan komputer, dan *software* berupa IDE (*integrated development envirotnment*) yang gunanya untuk memprogram papan rangkaian tersebut. *Software* Arduino berisi *editor teks* yang digunakan untuk menulis kode, teks konsol, dan terdapat sebuah *toolbar* dengan tombol fungsi secara umum. Pengguna dapat memasukan sebuah perintah dalam bentuk *coding*, kemudian ketika *software* melakukan *compile* secara langsung *software* akan memberikan umpan balik tentang *coding* yang dimasukan, apakah terdapat *error* atau tidak. Bahasa pemrograman yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan *library-library* Arduino.



Gambar 2 Bagian-Bagian dari Kit Arduino

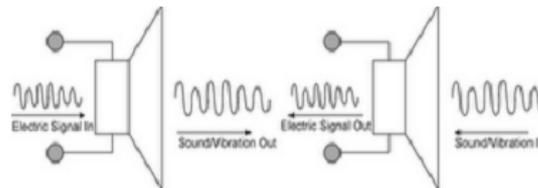
**Software Arduino.** Sehubungan dengan pembahasan untuk saat ini *software* Arduino yang akan digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan Arduino. IDE Arduino adalah *software* yang ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

- *Editor* program, sebuah *windows* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
- *Compiler*, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah microcontroller tidak akan bisa memahami bahasa processing, yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
- *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan Arduino. Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata "*sketch*" digunakan secara bergantian dengan "kode program" dan tidak ada perbedaan untuk ke dua kata tersebut.

**Sensor.** Sensor adalah peralatan yang digunakan untuk merubah suatu besaran fisis menjadi besaran listrik sehingga dapat dianalisa dengan rangkaian listrik tertentu. Sensor yang digunakan dalam sistem *lock & unlocking* pintu ini yaitu sensor *piezo* sebagai sensor yang membaca kode ketukan.

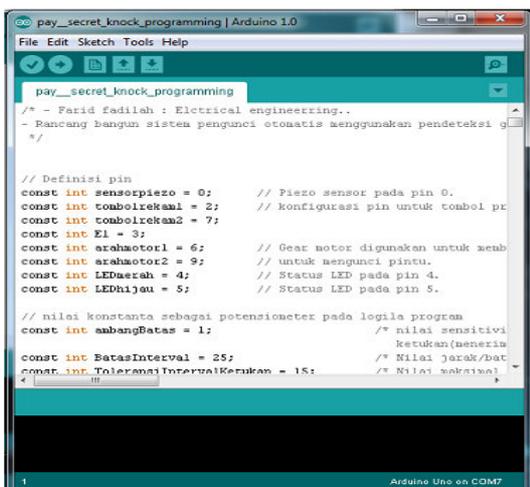
tekanan dan *electric* adalah listrik, maka piezoelektrik adalah muatan yang menumpuk dalam materi padat tertentu (terutama kristal, keramik tertentu, dan bahan biologis seperti tulang, DNA dan berbagai protein) untuk merespon regangan mekanik, jadi sederhananya kata piezoelectric berarti tegangan yang dihasilkan dari tekanan. Bahan Piezoelektrik terbentuk oleh keramik yang terpolarisasi sehingga beberapa bagian molekul bermuatan positif dan sebagian yang lain bermuatan negative membentuk elektroda-elektroda yang menempel pada dua sisi yang berlawanan dan menghasilkan medan listrik material yang dapat berubah akibat gaya mekanik. Pada saat medan listrik melewati material, molekul yang terpolarisasi akan menyesuaikan dengan medan listrik, dihasilkan dipole yang terinduksi dengan molekul atau struktur kristal materi. Penyesuaian molekul akan mengakibatkan material berubah dimensi. Fenomena ini disebut *electrostriction* (efek piezoelektrik).

**Sensor Piezo Buzzer.** Prinsip kerja dari Piezo Buzzer ini sama seperti sebuah *speaker* yang dibalik proses kerjanya seperti pada gambar berikut :



Gambar 4 Prinsip kerja sensor

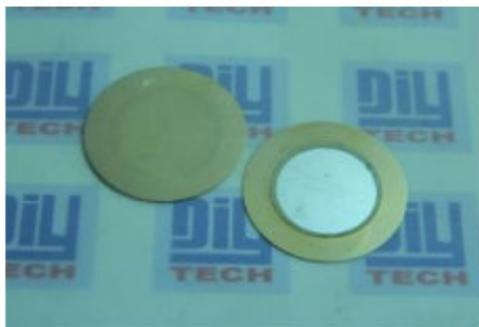
Sebuah *speaker* jika terminal – terminalnya mendapatkan sinyal seperti pada gambar sebelah kiri maka akan menghasilkan *output* berupa getaran pada membran dan menyebabkan terbentuknya bunyi. Sebaliknya pada saat *speaker* ini digunakan sebagai sensor (pada gambar sebelah kanan), lapisan membran pada *speaker* berfungsi sebagai detektor getaran. Ketika ada getaran datang pada membran, maka membran ini juga akan ikut bergetar (beresonansi). Bergetarnya membran akan mengakibatkan lilitan membran untuk bergerak relatif terhadap inti magnet permanen dan menghasilkan sinyal listrik. Seperti *Buzzer* Piezoelektrik yang tipis dibawah ini banyak digunakan sebagai *buzzer* ataupun *speaker* pada peralatan - peralatan seperti *gamewatch*, *timer*/alarm dan banyak produk lain. Bentuknya yang tipis dan beragam ukurannya



Gambar 3 Tampilan IDE Arduino dengan Sebuah *Sketch*

**Piezoelektrik.** Untuk kata piezoelektrik atau biasa disebut piezo ini berasal dari bahasa Yunani dimana piezo atau piezein yang berarti

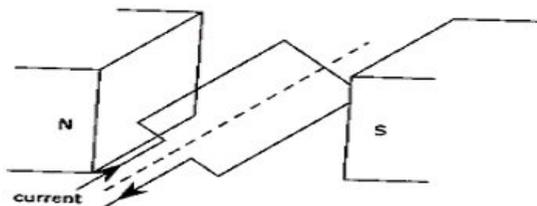
membuatnya cocok dipakai pada peralatan-peralatan di atas



Gambar 5 Salah Satu Bentuk Buzzer Piezoelektrik

Selain berfungsi layaknya sebuah speaker, buzzer piezoelectric ini bisa digunakan juga sebagai sensor tekanan atau getaran/ketukan. Saat diketuk-ketuk baik dengan jari maupun alat lainnya maka buzzer ini akan mengeluarkan tegangan yang besarnya sebanding dengan kuatnya ketukan yang diberikan pada permukaannya. Seperti contoh dibawah ini menggunakan sebuah LED sebagai indikator yang mendapatkan respon dari sensor Piezo.

**Motor DC.** Motor DC atau motor arus searah adalah suatu mesin yang berfungsi untuk mengubah tenaga listrik arus searah menjadi tenaga gerak atau tenaga mekanik, yang tenaga gerak tersebut berupa putaran dari rotor. Prinsip kerja dari motor DC hampir sama dengan generator AC, perbedaannya hanya terletak dalam konversi daya. Prinsip dasarnya adalah apabila suatu kawat berarus diletakkan diantara kutub – kutub magnet (U- S), maka pada kawat itu akan bekerja suatu gaya yang menggerakkan kawat tersebut.

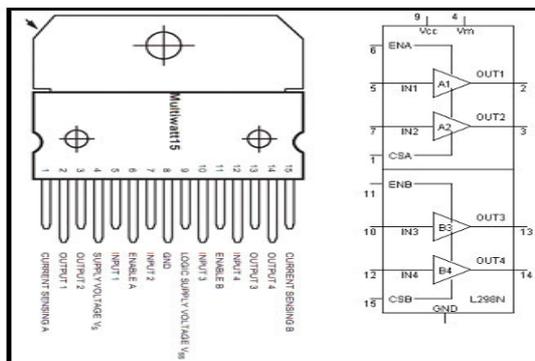


Gambar 6 Prinsip Kerja Motor DC

Apabila sebuah belitan terletak dalam medan magnet yang homogen, arah gerakan ditunjukkan seperti gambar diatas, karena kedua sisi lilitan mempunyai arus yang arahnya berlawanan.

**Motor DC Micro.** Terdapat banyak sekali tipe motor DC mikro atau Motor DC dalam ukuran mini, salah satunya dimanfaatkan dalam motor servo micro. Motor servo micro termasuk dalam bagian servo standard. Didalam motor servo tersebut terdapat motor DC, komposisi gearbox dan rangkaian kontrolernya. Motor servo dapat dimodifikasi menjadi motor DC biasa dengan menghilangkan rangkaian pengontrolnya yang otomatis untuk pin sinyal (PWM) tidak dapat berfungsi lagi. Hal ini dilakukan untuk beberapa tujuan tertentu seperti motor dapat berputar secara kontinyu (berputar lebih dari 360°) bersamaan dengan gearbox untuk meningkatkan torsi saat menjalankan beban. Di bawah ini akan ditunjukkan bentuk dari motor DC micro dan gear box yang terdapat pada bagian dalam motor servo.

**IC Driver L298N.** IC L298N ini adalah suatu bentuk rangkaian daya tinggi terintegrasi yang mampu melayani 4 buah beban dengan arus nominal 600mA hingga maksimum 1.2 A. Keempat channel inputnya didesain untuk dapat menerima masukan level logika TTL (Transistor Transistor Logic). Biasa dipakai sebagai driver relay, motor DC, motor stepper maupun pengganti transistor sebagai saklar dengan kecepatan switching mencapai 5kHz. Driver tersebut berupa dua pasang rangkaian h-bridge yang masing-masing dikendalikan oleh enable 1 dan enable 2. Dengan memberikan tegangan 5V sebagai Vcc pada pin 4 dan 12 Volt pada pin 9 untuk tegangan motor, maka IC siap digunakan. Saat terdapat tegangan pada input 1 dan 2, dengan memberikan logika tinggi pada enable 1 maka output 1 dan 2 akan aktif. Sedangkan jika enable 1 berlogika rendah, meskipun terdapat tegangan pada input 1 dan 2 output tetap nol (tidak aktif). Hal ini juga berlaku untuk input dan output 3 dan 4 serta enable 2. Konfigurasi pin ICL298N tersebut dapat dilihat lebih jelas pada gambar berikut.

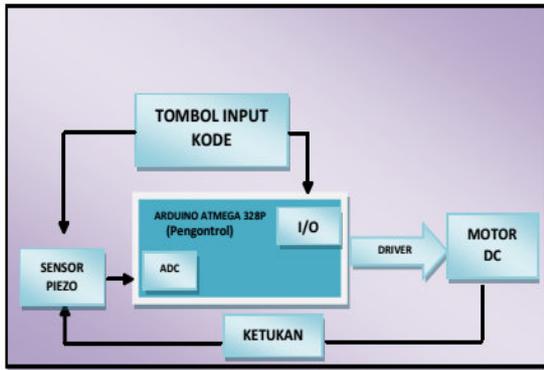


Gambar 7 Konfigurasi Pin IC Driver L298N

**METODE PENELITIAN**

**Perancangan Perangkat Keras (*Hardware*).**

Secara umum perancangan *Hardware* pada sistem penguncian pintu otomatis dapat dilihat pada gambar dibawah ini.



Gambar 8 Blok Diagram Sistem

Adapun fungsi dari blok-blok di atas sebagai berikut:

1. Sensor Piezo Buzzer berfungsi untuk mendeteksi getaran/ketukan yang memicu untuk menjalankan motor DC.
2. "Tombol *Input Kode*" digunakan sebagai inputan ketukan/merekam ketukan baru.
3. Sistem minimum arduino untuk IC mikrokontroler AVR ATmega328P berfungsi sebagai pusat kontrol dan pengolahan data *input - output* dari sensor.
4. IC driver L293N berfungsi sebagai *driver* untuk mengatur arah putar motor DC (HBridge).
5. Motor Servo Micro digunakan untuk menjalankan beban pada sistem penguncian sesuai dengan kode ketukan

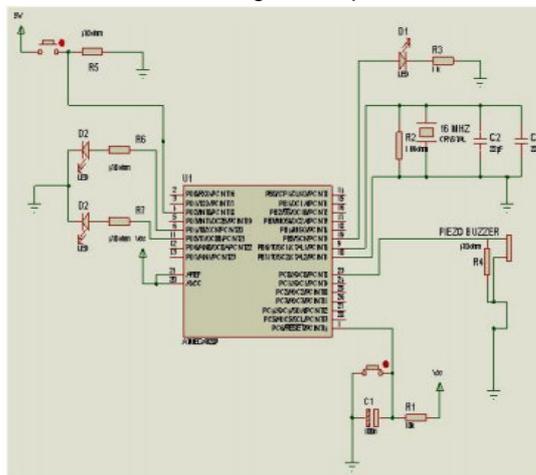
**Perancangan Rangkaian Sensor Piezo Buzzer dan "Tombol *Input Kode*".** Sensor Piezo Buzzer akan menghasilkan tegangan jika diberikan tekanan ketukan, *encoding* rentang tegangannya adalah dari 0 – 5 volt, berikut ini adalah langkah-langkah perancangan rangkaian sensor dan "tombol *input kode*"..

1. Menyiapkan minimum sistem Arduino untuk I/O sensor dan "tombol *input kode*" (*push button*).
2. Menghubungkan sensor ke mikrokontroler. Sensor piezo memiliki dua kabel, kabel merah dihubungkan ke PA0/ADC dan kabel hitam dihubungkan ke *ground* sedangkan "tombol *input kode*" salah satu konektornya dihubungkan ke pin digital yang diparalelkan dengan resistor 470 ohm menuju *ground* dan

konektornya yang satu dihubungkan ke sumber 5 volt.

3. Menghubungkan resistor 470 ohm secara paralel dengan elemen piezo untuk membatasi tegangan dan arus yang dihasilkan oleh piezo yang sekaligus melindungi analog input pada sismin mikrokontroler.

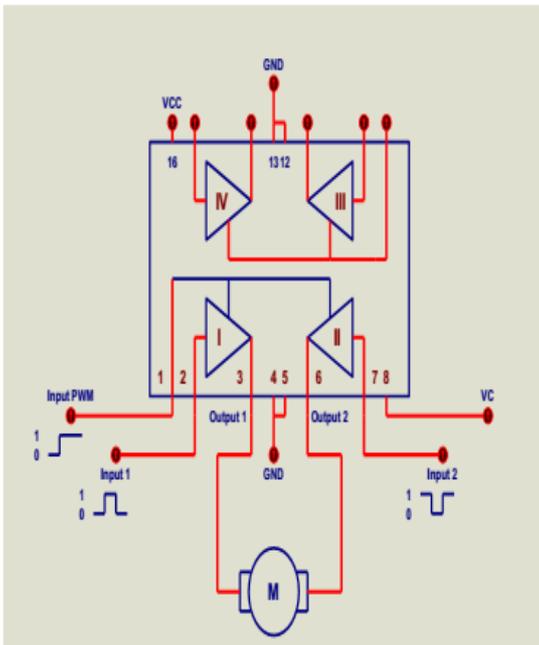
Dibawah ini akan ditunjukkan rangkaian sismin Arduino sederhana (berdasarkan refrensi skematik Arduino), Piezo Buzzer dan "tombol *input kode*" berdasarkan pemetaan pin mikrokontroler ATmega328P pada Arduino.



Gambar 9 Sismin Arduino ATmega328P dengan Sensor Piezo Buzzer & Tombol *Input Kode*

**Perancangan *Driver Motor DC*.** Secara teoritis, sebuah motor DC dapat digerakkan langsung oleh mikrokontroler. Dalam kenyataannya, arus dan tegangan yang dikeluarkan oleh mikrokontroler terlalu kecil untuk menggerakkan sebuah DC. Gerbang-gerbang

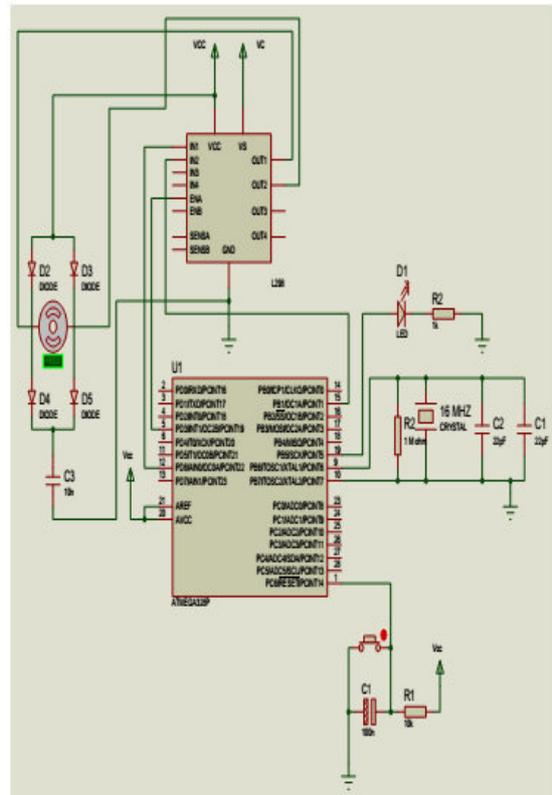
*Transistor Transistor Logic* (TTL) mikrokontroler hanya mampu mengeluarkan arus dalam orde mili-ampere dan tegangan antara 2 sampai 2,5 Volt. Sementara itu untuk menggerakkan motor DC diperlukan arus yang lebih besar (dalam orde ampere) dan 16 tegangan berkisar 5 sampai 12 Volt. Untuk mengatasi masalah tersebut, maka digunakan sebuah piranti tambahan yang memenuhi kebutuhan arus dan tegangan yang cukup besar. Oleh karena itu, diperlukan sebuah *driver motor* yang berfungsi untuk mengendalikan motor DC dengan suplai tegangan dari luar. Rangkaian *driver motor DC* ini terdiri dari IC H-Bridge L293N.



Gambar 10 IC Driver Motor DC L298N

IC L298N mempunyai 4 *input* dan 4 *output* yang bisa digunakan untuk mengontrol 2 motor DC. Namun pada perancangan ini hanya menggunakan *input* 1 dan 2 karena hanya mengontrol 1 motor DC. *Input* 1 dan 2 digunakan sebagai pengatur arah perputaran motor DC. Sedangkan untuk PWM digunakan sebagai pengatur kecepatan motor DC.

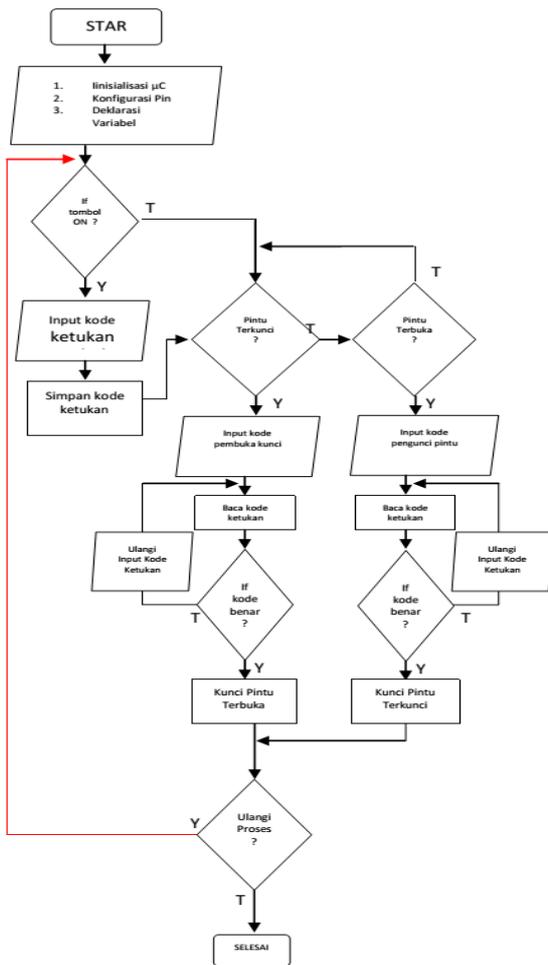
Gambar 9 dan 10 merupakan perancangan rangkaian *Driver* L298N untuk motor DC dan rangkaian sismin sederhana arduino sesuai dengan skematik komplit Arduino. Berdasarkan pemetaan pin mikrokontroler ATmega328P pada Arduino sesuai dengan dasar teori, pin EN pada *Driver* motor terhubung pada pin 3 Arduino, pin 5 (pin IN1) *Driver* motor terhubung pada pin 6 Arduino, pin 7 (pin IN2) *Driver* motor terhubung pada pin 9 Arduino. Untuk sumber tegangan *Driver* motor diberi sebesar 5 volt (VCC) dan untuk sumber tegangan Motor DC diberi sebesar 12 volt (VS).



Gambar 11 Rangkaian Pengatur Arah Putar Motor DC & Sismin Arduino Sederhana

**Perancangan Perangkat Lunak**, Perancangan perangkat lunak atau program dibuat dalam bahasa C menggunakan *software* sistem minimum arduino IDE (*integrated development environment*).

**Perancangan program untuk mikrokontroler ATmega328P**. Perancangan perangkat lunak yang diprogram pada mikrokontroler di dalam sistem penguncian ini meliputi perancangan inputan awal kode ketukan untuk pembuka dan pengunci pintu, pengisian kode ketukan sebagai data *input sensor* untuk pembuka kunci, *input* kode ketukan ini yang akan menggerakkan motor DC untuk membuka dan mengunci pintu secara otomatis.



Gambar 12 diagram alir Penelitian

**HASIL DAN PEMBAHASAN**

**Pembacaan Interval dan Irama Kode Ketukan Pembuka Kunci.** Array disini merupakan inputan awal nilai *interval* ketukan (sebelum menekan “tombol *input* kode”) untuk kode ketukan yang dimasukkan dalam baris program. Array ini merupakan patokan nilai *interval* yang akan dibaca setelah diberikan ketukan pada sensor Piezo Buzzer. Nilai Array = { 50, 25, 25, 50, 100, 50, 0, 0, 0, 0 } dengan jumlah ketukan sebanyak 7 ketukan yang telah terbentuk suatu ritme/irama. Dalam program jumlah maksimum *interval* telah ditetapkan sebanyak 10 *interval* yang berarti bahwa jumlah maksimum untuk mengetuk berjumlah 11 ketukan. Array diatas memiliki beberapa nilai *interval* yang sama diantaranya nilai 50 ms terdapat tiga nilai dan nilai 25 ms terdapat dua nilai, sedangkan nilai 100 ms adalah nilai *interval* terpanjang. Di bawah ini akan ditunjukkan proses pembuka kunci berdasarkan nilai awal Array.

Tabel 1 Perbandingan nilai interval ketukan (Array) pembuka kunci

Interval ketukan (Array)	1 ke 2 (ms)	2 ke 3 (ms)	3 ke 4 (ms)	4 ke 5 (ms)	5 ke 6 (ms)	6 ke 7 (ms)
Nilai awal	50	25	25	50	100	50
Nilai Interval ketukan ke-1	52	24	26	51	100	53

- Keterangan tabel :
- Nilai awal = Nilai awal Array (interval ketukan) yang ditanam dalam program.
  - Nilai Interval ketukan ke-1 = Nilai Interval ketukan yang pertama ketika memberikan ketukan pada sensor sesuai dengan irama berdasarkan “nilai awal” interval ketukan (array).

Pada tabel 1 di atas dapat dilihat “nilai awal” dan setelah diberi ketukan dimana nilai yang menurun atau naik tidak melebihi nilai toleransi (sebesar 15 ms untuk memproporsionalkan nilai *interval* yang “lebih dari” atau yang “kurang dari” nilai awal *interval* ketukan) yang artinya ketukan terbaca.

Nilai *interval* ketukan (Array) semakin berkurang disebabkan tempo dipercepat tapi masih dapat diproporsionalkan dengan nilai toleransi *interval*.

Tabel 2 Perbandingan nilai interval ketukan pembuka kunci (Tempo cepat)

Interval ketukan (Array)	1 ke 2 (ms)	2 ke 3 (ms)	3 ke 4 (ms)	4 ke 5 (ms)	5 ke 6 (ms)	6 ke 7 (ms)
Nilai awal	50	25	25	50	100	50
Nilai Interval ketukan ke-1	52	24	26	51	100	53
Nilai Interval ketukan ke-2	35	19	19	39	100	39

- Keterangan tabel :
- Nilai awal = Nilai awal Array (interval ketukan) yang ditanam dalam program.
  - Nilai interval ketukan ke-1 = Nilai Interval ketukan pertama ketika memberikan ketukan pada sensor sesuai dengan irama berdasarkan “nilai awal” interval ketukan (Array).
  - Nilai interval ketukan ke-2 = Nilai interval ketukan yang kedua sesuai dengan irama berdasarkan “nilai awal” dimana tempo ketukan dipercepat.

Sekarang akan dilakukan percobaan dengan salah satu *input* ketukan melebihi nilai toleransi *interval* yang mewakili bahwa

ritme/irama ketukan salah atau ketukan tidak terbaca.

Tabel 3 Perbandingan nilai interval ketukan pembuka kunci (Irama Salah)

Interval ketukan (Array)	1 ke 2 (ms)	2 ke 3 (ms)	3 ke 4 (ms)	4 ke 5 (ms)	5 ke 6 (ms)	6 ke 7 (ms)
Nilai awal	50	25	25	50	100	50
Nilai interval ketukan	42	11	12	42	100	66

Dari table 3 di atas dapat dilihat perbandingan “nilai awal” dengan “nilai *interval* ketukan” yang memiliki perbedaan melebihi nilai toleransi *interval* yaitu nilai *interval* antara ketukan 6 sampai ketukan 7 (6 ke 7), perhatikan bahwa “nilai awal” adalah 50 ms sedangkan “nilai *interval* ketukan” adalah 66 ms. Nilai toleransi *interval* untuk jarak yang sama adalah 15 maka  $50 + 15 = 65$ , sedangkan nilai *input* ketukan sebesar 66 ms, nilai ini sudah melebihi nilai toleransi *interval*, jadi sudah dipastikan ketukan tidak terbaca.

Kode ketukan baru ini akan tetap berfungsi selama “tombol *input* kode” tidak dalam logika *high* yang disertakan dengan ketukan dan tidak menekan tombol *reset* yang terdapat pada Arduino atau selama catu daya tidak diputuskan. Pada tampilan serial monitor di atas nilai Array sedikit menurun dari input pada saat “tombol *input* kode” ditekan yang artinya tempo ketukan sedikit lebih cepat.

Tabel .4 Perbandingan nilai Array (“Tombol *Input* Kode”)

Interval ketukan (Array)	1 ke 2 (ms)	2 ke 3 (ms)	3 ke 4 (ms)	4 ke 5 (ms)	5 ke 6 (ms)
Nilai awal tombol <i>input</i> kode	72	27	100	71	26
Nilai interval ketukan	60	22	100	60	20

**Pembacaan Interval dan Irama Kode Ketukan Pengunci Pintu.** Kode ketukan untuk pengunci pintu ini dibuat lebih sederhana dalam bentuk Array = { 100, 50, 50, 100, 0, 0, 0, 0, 0 } dengan jumlah ketukan sebanyak 5 ketukan yang sudah terbentuk ritme/irama yang sederhana.

Tabel 5 Perbandingan nilai Array Pengunci Pintu

Interval ketukan (Array)	1 ke 2 (ms)	2 ke 3 (ms)	3 ke 4 (ms)	4 ke 5 (ms)
Nilai awal	100	50	50	100
Nilai interval ketukan	99	48	51	100

## KESIMPULAN

Setelah melalui tahap perencanaan, perancangan dan pembuatan serta pengujian sistem pengunci pintu otomatis, ada beberapa hal yang dapat disimpulkan, yaitu:

1. Kode ketukan yang digunakan pada sistem pengunci pintu ini adalah nilai *interval* antar ketukan (jarak antara ketukan yang satu dengan yang lainnya).
2. Nilai *interval* antar ketukan yang ditanam dalam program merupakan sebuah bentuk ritme/irama.
3. Pembacaan *interval* antar ketukan akan diberikan nilai toleransi sebesar 15 mili second yang berfungsi untuk memproporsionalkan *interval* ketukan pada jarak yang sama.
4. Getaran yang dihasilkan dari ketukan akan dideteksi oleh sensor jika  $\geq 1$  volt (ambang batas) yang mewakili nilai *level* tekanan suara (amplitudo dalam satuan dB).
5. Nilai *interval* terpanjang dan nilai toleransi *interval* berfungsi untuk menormalkan data ketukan, jika ke-2 *variable* ini tidak dimasukkan maka ketukan harus sesuai dengan tempo yang sebenarnya dan nilai jarak antar ketukan harus akurat.
6. Pengisian kode ketukan menggunakan *push button* sebagai kode baru untuk membuka kunci pintu bukanlah menghapus deret yang tertanam dalam program melainkan mengganti (*replace*) selama catu daya tidak terputus.
7. Dari hasil pengujian, baik itu untuk membuka kunci atau mengunci pintu terdapat perbandingan nilai *interval* antar ketukan, tapi masih dapat menjalankan motor, jadi hal ini tidak mempengaruhi proses kerja sistem asalkan ketukan dalam ritme/irama yang benar.

## DAFTAR PUSTAKA

- Anonim, 2004, *Buku Pedoman Penulisan Tulisan Ilmiah*, Fakultas Teknik Universitas Mataram
- Anonim, : [www.alldatasheet.com](http://www.alldatasheet.com) tanggal akses 9 September 2012
- Anonim, : <http://grathio.com> tanggal akses 28 September 2012
- Anonim, : <http://www.dewataelektronik.com> tanggal akses 2 Oktober 2012
- Anonim: [www.piezoelektrikbuzzer-sebagai-penghasi-listrik.com](http://www.piezoelektrikbuzzer-sebagai-penghasi-listrik.com) tanggal akses 26 Desember 2012

- Anonim: [:www.scribd.com/doc/56783346/Buku - Arduino - Programming - Blm-Jadi](http://www.scribd.com/doc/56783346/Buku-Arduino-Programming-Blm-Jadi) tanggal akses 14 februari 2013
- Anonim: [:www.penggunaanspeaker-sebagai-sensor getaran.com](http://www.penggunaanspeaker-sebagai-sensor-getaran.com) tanggal akses 16 Februari 2013
- Basukesti, Agus, 2011, *Perancangan Sensor Getaran Berbasis Mikrokontroler AT89C51*, Jurusan Teknik Elektro STT Yogyakarta
- Novianta., M A, 2012, *Sistem Deteksi Dini Gempa Dengan Piezo Elektrik Berbasis Mikrokontroler AT89C51*, Jurusan Teknik Elektro ISTAYogyakarta
- Winoto, Ardi, 2008. *Mikrokontroler AVR ATmega8/32/16/8535 dan pemrogramannya dengan Bahasa C pada WinAVR*, Informatika, Bandung