

Pengembangan Prototipe Lampu Lalu Lintas Pejalan Kaki Berbasis Mikrokontroler MCS-51 yang Diinisiasi dengan Tombol

Development of Button-Initiated Pedestrian Traffic Lights Using MCS-51 Microcontroller

IRENE PRICELLA SITUMORANG, KARLISA PRIANDANA*

Abstrak

Saat ini, di Indonesia, hak pejalan kaki untuk menyeberang jalan pada *zebra cross* sering diabaikan. Padahal, UU No. 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, Pasal 130 ayat 2 secara eksplisit menjelaskan bahwa pejalan kaki yang menyeberang pada *zebra cross* berhak mendapat prioritas. Penelitian ini mengembangkan dua prototipe lampu lalu lintas untuk pejalan kaki yang dapat diinisiasi dengan tombol dengan menggunakan mikrokontroler keluarga MCS-51 untuk diimplementasikan di *zebra cross*. Mode timer MCS-51 yang digunakan adalah mode timer 1. Prototipe pertama menggunakan *push-button*, sedangkan prototipe kedua menggunakan *keypad* sebagai tombol untuk menginisiasi lampu lalu lintas. Penekanan tombol akan memulai kondisi menyeberang, yaitu kondisi pada saat lampu lalu lintas kendaraan menjadi merah dan lampu lalu lintas penyeberang jalan menjadi hijau. Dalam hal ini, *seven segment* akan menampilkan waktu hitung mundur yang dapat dipilih melalui tombol. Penelitian ini menunjukkan bahwa penggunaan *push-button* lebih ramah guna dan lebih mudah dipelihara. Sementara itu, penggunaan *keypad* lebih fleksibel dalam hal pemilihan waktu menyeberang.

Kata kunci: *keypad*, lampu lalu lintas, mikrokontroler MCS-51, *push-button*, *seven segment*

Abstract

Nowadays, in Indonesia, pedestrian's right to cross through *zebra cross* is often abandoned. On the other hand, Law No. 22 of 2009 concerning Traffic and Road Transportation, article 130 paragraph 2 explicitly mentioned that pedestrian who cross through *zebra cross* shall be prioritized. This study develops two button-initiated pedestrian traffic light prototypes using MCS-51 microcontroller, to be implemented in *zebra cross*. The utilized MCS-51 timer mode is mode 1. The first prototype utilizes several *push-buttons*, whereas the second prototype uses *keypad* as the button to initiate the traffic light. Pressure on the button will trigger the crossing condition, where the vehicle's traffic light becomes red and the pedestrian's traffic light becomes green. In this condition, *seven segment* will show the count down time that can be selected through the button. This study shows that the *push-button* prototype is more user friendly and easily maintained. Meanwhile, the *keypad* prototype is more flexible in terms of selecting the crossing time.

Keywords: *keypad*, MCS-51 microcontroller, *push-button*, *seven segment*, traffic light

PENDAHULUAN

Saat ini, di Indonesia kecelakaan lalu lintas dengan korban pejalan kaki meningkat. Pengemudi kendaraan kurang memperdulikan hak pejalan kaki dalam penggunaan lalu lintas dan jalan raya, walaupun pejalan kaki memiliki hak yang sama dengan pengemudi kendaraan dalam penggunaan lalu lintas dan jalan raya. Kecelakaan-kecelakaan tersebut menunjukkan bahwa pejalan kaki kurang mendapatkan perlindungan yang memadai dalam penggunaan lalu lintas dan jalan raya. Meskipun fasilitas-fasilitas pendukung bagi pejalan kaki telah disediakan oleh pemerintah, namun belum berfungsi sebagaimana mestinya.

Zebra cross adalah salah satu fasilitas pendukung bagi pejalan kaki yang dinyatakan dengan markah jalan berbentuk garis membujur berwarna putih dan hitam, berfungsi untuk memberikan keamanan bagi pejalan kaki pada saat menyeberang jalan. Undang-Undang Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan pasal 130 ayat 2 menyatakan bahwa pejalan kaki yang sedang berjalan pada *zebra cross* berhak mendapatkan prioritas untuk menyeberang jalan. Namun pada kenyataannya, banyak pengemudi kendaraan lebih mengutamakan kepentingannya daripada pejalan kaki. Pengemudi kendaraan melaju kendaraannya lebih cepat pada saat melihat pejalan kaki yang akan atau sedang menyeberang jalan. Oleh karena itu, diperlukan suatu sistem lalu lintas tambahan untuk membantu memaksimalkan fungsi *zebra cross* tersebut.

Adapun penelitian yang pernah dilakukan Jatmiko *et al.* (2009), yaitu mengenai kontrol lampu lalu lintas pada persimpangan jalan berdasarkan perubahan jumlah kendaraan pada masing-masing bagian simpang. Penelitian dalam bentuk simulasi ini menggunakan pendekatan pada suara dengan menghitung nilai frekuensi suara yang ditimbulkan. Penelitian tersebut diharapkan dapat mengurangi kemacetan di Indonesia khususnya pada bagian persimpangan jalan. Terkait penelitian sebelumnya, penelitian ini mengembangkan suatu sistem lalu lintas berbasis mikrokontroler yang juga masih dalam bentuk simulasi atau prototipe.

Implementasi lampu lalu lintas pejalan kaki sudah diterapkan di beberapa negara, salah satunya Indonesia. Beberapa kota di Indonesia yang telah mengimplementasikan lampu lalu lintas tersebut ialah Jakarta, Bogor, dan Semarang. Akan tetapi, implementasi lampu lalu lintas tersebut belum mendapat sambutan yang baik dari masyarakat. Selain itu, pemerintah pun kurang memberikan sosialisasi cara penggunaannya kepada masyarakat.

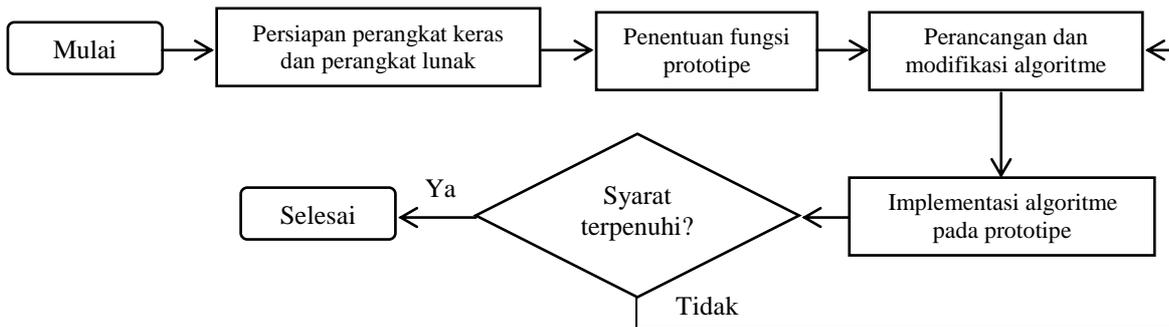
Penelitian ini bertujuan membuat prototipe sebuah aplikasi sederhana yang berbasis mikrokontroler MCS-51 berupa lampu lalu lintas pejalan kaki yang diinisiasi dengan tombol. Mikrokontroler yang digunakan adalah mikrokontroler keluarga MCS-51 yang banyak dijumpai di pasaran. Kelebihan mikrokontroler ini ialah fitur yang praktis dan ukuran memori yang cukup besar (Suyatno *et al.* 2007).

Penelitian ini diharapkan dapat menjadi pionir dalam pembuatan alat lalu lintas pejalan kaki berbasis mikrokontroler untuk diimplementasikan pada *zebra cross*. Implementasi lampu lalu lintas ini diharapkan dapat membantu memaksimalkan fungsi *zebra cross* untuk memberi keamanan bagi pejalan kaki dalam menyeberang jalan.

METODE

Penelitian ini menggunakan 6 ketentuan. Mikrokontroler yang digunakan adalah mikrokontroler 89x52 dan merupakan salah satu keluarga MCS-51, yang telah terangkai pada rangkaian praktikum. Tombol yang digunakan sebagai inisiasi penyeberangan jalan yang dikembangkan pada 2 prototipe adalah 6 *push-button* dan *keypad*, inisiasi tombol kembali dapat dilakukan setelah jeda waktu selama 5 menit. Kondisi *default* adalah kondisi saat lampu hijau pada salah satu set lampu lalu lintas (diasumsikan untuk kendaraan) dan lampu merah pada set lampu lalu lintas lainnya (diasumsikan untuk pejalan kaki) menyala. *Seven segment* menampilkan perhitungan waktu mundur dari x detik ke-0 di mana rentang nilai pada masing-masing *push-button* ialah 10-60 detik, sedangkan rentang nilai x dengan menggunakan *keypad* ialah 0-99 detik. Lampu kuning digunakan sebagai lampu transisi yang akan menyala pada saat pergantian *state* dari lampu merah menyala menjadi lampu hijau menyala dan sebaliknya. Implementasi prototipe diterapkan pada *zebra cross* yang berada pada jalan lurus dan bukan persimpangan.

Metode yang digunakan adalah metode *trial and error* dengan alur yang dapat dilihat pada Gambar 1. Prototipe yang dikembangkan menggunakan perangkat keras berupa sebuah rangkaian praktikum yang telah terangkai dengan mikrokontroler jenis 89x52, salah satu mikrokontroler keluarga MCS-51.



Gambar 1 Diagram alur metode penelitian

Selain mikrokontroler, komponen lain yang terangkai pada rangkaian tersebut dan digunakan untuk mengembangkan prototipe adalah *seven segment*, *light-emitting diode* (LED) lampu lalu lintas, *push-button*, dan *keypad*. Setiap fungsi prototipe akan disesuaikan dengan kebutuhan pejalan kaki dalam menyeberang jalan.

Algoritme dari fungsi-fungsi tersebut dirancang dan kemudian diimplementasikan pada prototipe. Implementasi algoritme dalam bentuk program dilakukan dengan perangkat lunak emulator MCU 8051 IDE dan *compiler* MIDE51 yang telah dipasang pada komputer. Kemudian, program tersebut akan dimasukkan pada mikrokontroler dengan menggunakan *in-system programming* (ISP) sebagai penghubung antara komputer dan mikrokontroler. Prototipe dievaluasi kembali untuk disesuaikan dengan fungsinya. Jika belum sesuai dengan fungsinya, algoritme akan dimodifikasi. Hasil modifikasi algoritme tersebut diimplementasikan kembali dan dievaluasi apakah syaratnya telah terpenuhi. Jika seluruh syarat telah terpenuhi, penelitian telah berhasil dan prototipe telah selesai dikembangkan.

Perangkat Keras

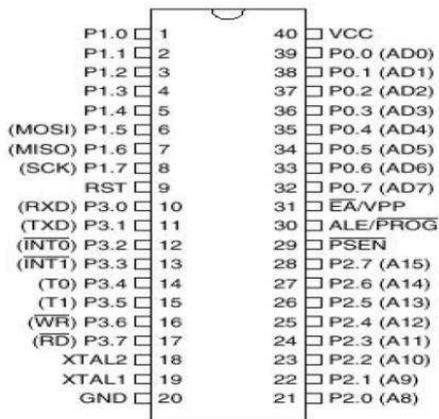
Penelitian ini menggunakan mikrokontroler 89x52, yang merupakan salah satu mikrokontroler keluarga MCS-51 dan memiliki respons yang lebih cepat daripada 89x51. Beberapa fitur mikrokontroler ini adalah (Hamirani *et al.* 2012): sebuah *central processing unit* (CPU) 8 bit yang termasuk keluarga osilator internal dan rangkaian *timer*; *flash* memori 8 Kb; RAM 128 bytes; empat buah *programmable port* I/O, masing-masing terdiri atas 8 jalur I/O; lima buah jalur interupsi (2 buah interupsi eksternal dan 3 buah interupsi internal); sebuah *port* serial dengan kontrol serial *full duplex UART*.

Kecepatan pelaksanaan intruksi per siklus untuk mikrokontroler ini adalah 1 mikrodetik pada frekuensi *clock* 12 MHz. Mikrokontroler ini memiliki *timer* internal dengan frekuensi 12 MHz (Suyatno *et al.* 2007). Susunan pin mikrokontroler 89x52 dapat dilihat pada Gambar 2. Komponen pendukung lainnya yang digunakan pada penelitian ini ialah *seven segment* (Gambar 3) yang digunakan untuk menampilkan waktu hitung mundur dalam satuan detik. *Seven segment* merupakan sebuah *display* yang terbentuk atas 7 buah segmen LED (ditambah 1 segmen untuk menampilkan titik), yaitu segmen A, B, C, D, E, F, dan G, serta DP, satu buah segmen tambahan untuk menampilkan titik. Pada penelitian ini, *seven segment* dihubungkan ke Port 0. Pengaturan tampilan *seven segment* dilakukan dengan menyalakan atau memadamkan segmen yang sesuai. *Seven segment* yang menyala dikontrol oleh *decoder/demultiplexer* 74LS138.

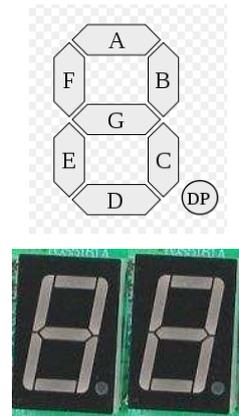
Gambar 4 merupakan *push-button* yang digunakan sebagai tombol inisiasi terhadap lampu lalu lintas. Selain *push-button*, *keypad* juga dapat digunakan sebagai tombol untuk inisiasi. *Keypad* terdiri atas sejumlah saklar yang terhubung sebagai baris dan kolom seperti Gambar 5. Untuk mengetahui tombol yang ditekan, mikrokontroler akan melakukan teknik *scanning* dengan cara memberi logika *low* "0" ke salah satu *port* yang terhubung ke kolom *keypad* (C1–C4). Selanjutnya, mikrokontroler memeriksa seluruh *port* yang terhubung pada

baris (R1–R4) untuk menguji jika ada tombol yang ditekan pada baris dan kolom tersebut. Jika tidak ada tombol yang ditekan, maka mikrokontroler akan mendapati bahwa seluruh *port* yang terhubung pada baris (R1–R4) memiliki logika *high* “1”. Kedua tombol akan terhubung pada *port* yang sama, yaitu *port* 2. Oleh karena itu, dikembangkanlah 2 prototipe.

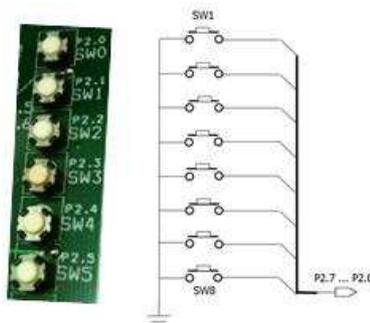
Dua set lampu lalu lintas yang digunakan dapat dilihat pada Gambar 6. Satu set lampu lalu lintas digunakan untuk pejalan kaki dan satu set lainnya digunakan untuk pengemudi kendaraan. LED yang digunakan sebagai lampu lalu lintas terhubung pada *port* 1. Komponen *in-system programming* pada Gambar 7 digunakan untuk memprogram mikrokontroler 89x52. Skematik prototipe dengan *push button* dan *keypad* yang telah dihubungkan dengan seluruh komponen yang digunakan.



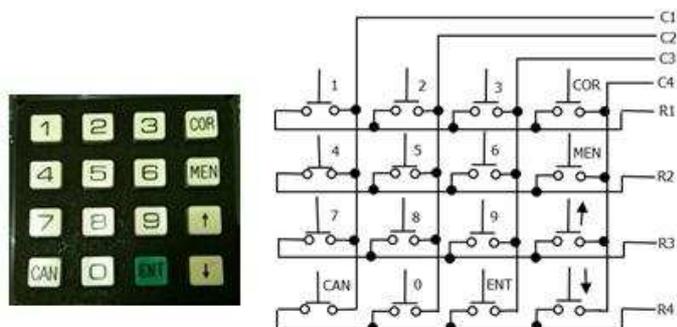
Gambar 2 Konfigurasi pin MCS-51



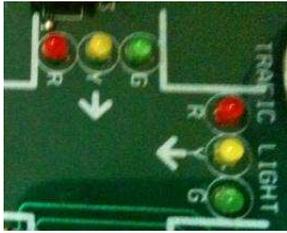
Gambar 3 Seven segment



Gambar 4 Skema rangkaian antarmuka *push-button*



Gambar 5 Rangkaian dasar *keypad* 4x4



Gambar 6 Dua set lampu lalu lintas

Gambar 7 *In-system programming* (ISP)

Perangkat Lunak

Perangkat lunak yang akan digunakan pada penelitian ini ialah Emulator MCU 8051 IDE, *compiler* MIDE51, dan *driver programmer* ProgISP versi 1.72.

Penentuan Fungsi Prototipe

Pada tahap ini prototipe akan dirancang sesuai dengan kebutuhan pejalan kaki.

Perancangan dan Modifikasi Algoritme

Algoritme yang diadopsi dalam penelitian ini adalah algoritme jam digital. *Timer* yang digunakan pada algoritme jam digital adalah *timer* internal. Algoritme lainnya akan dikembangkan sesuai dengan fungsi yang diinginkan. Pengembangan algoritme dilakukan dengan menggunakan bahasa *assembly* pada perangkat lunak emulator MCU 8051 IDE.

Implementasi Algoritme pada Prototipe

Algoritme yang telah dikembangkan akan diimplementasikan pada perangkat prototipe dengan programmer ProgISP versi 1.72.

Evaluasi

Pada tahap ini, fungsi prototipe akan diuji dengan 7 persyaratan. Kondisi default dapat ditampilkan; Kondisi menyeberang dapat diinisiasi dengan tombol. Tombol yang digunakan terdiri atas *push-button* dan *keypad*. Waktu kondisi menyeberang dengan menggunakan *push-button* diaktifkan dengan menekan *push-button* sebanyak 2 kali, sedangkan waktu kondisi menyeberang dengan menggunakan tombol *keypad* diaktifkan dengan menekan *keypad* ENT setelah terlebih dahulu menekan *keypad* angka secara bebas. Pada *keypad*, waktu hitung mundur dapat dihapus dan kemudian diganti dengan menggunakan *keypad* COR atau MEN; Waktu hitung mundur dapat ditampilkan pada *seven segment*; Setelah waktu habis, lampu lalu lintas kembali ke kondisi *default*; Tombol tidak dapat ditekan kembali sebelum jeda waktu 5 menit; Prototipe akan memberikan waktu hitung mundur selama x detik. Nilai x yang tersimpan pada masing-masing *push-button* ialah 10, 20, 30, 40, 50, dan 60 detik, sedangkan nilai x yang dapat dihasilkan dengan *keypad* adalah 0-99 detik; Transisi lampu merah menjadi hijau dan sebaliknya menggunakan lampu kuning; Apabila persyaratan fungsi di atas belum seluruhnya terpenuhi, algoritme yang dirancang perlu ditelusuri kembali. Penelitian dianggap selesai setelah ke-7 fungsi terpenuhi.

HASIL DAN PEMBAHASAN

Rancangan Fungsi Prototipe

Fungsi prototipe yang dikembangkan ialah kondisi *default*: lampu hijau pada salah satu set lampu lalu lintas (yang diasumsikan untuk kendaraan) dan lampu merah pada set lampu lalu lintas lainnya (yang diasumsikan untuk pejalan kaki) menyala; kondisi lampu lalu lintas dapat diubah menjadi “kondisi menyeberang”, yaitu kebalikan dari kondisi *default*-nya melalui inisiasi tombol. Setelah tombol ditekan, *seven segment* akan menampilkan perhitungan mundur dalam detik; waktu “kondisi menyeberang” yang akan ditampilkan pada *seven segment* ditentukan melalui *push-button*, dengan rentang nilai antara 10-60 detik, dan *keypad*, dengan rentang nilai 0-99 detik; transisi lampu merah menjadi hijau dan sebaliknya menggunakan lampu kuning; tombol dapat ditekan kembali setelah jeda waktu 5 menit.

Rancangan Algoritme

Algoritme dikembangkan berdasarkan fungsi prototipe yang telah ditentukan, yaitu dengan *push-button* (Gambar 8) dan dengan *keypad* (Gambar 9). Prototipe dimulai dengan kondisi *default*, yaitu lampu hijau pada lampu A, satu set lampu lalu lintas untuk pengemudi kendaraan, dan lampu merah pada lampu B, satu set lampu lalu lintas untuk pejalan kaki, menyala serta *seven segment* masih dalam keadaan non-aktif.

Tombol yang digunakan untuk inisiasi terdiri atas *push-button* dan *keypad*. Pada Gambar 8, dapat dilihat bahwa apabila *push-button* ditekan sekali, *seven segment* aktif untuk menampilkan waktu penyeberangan x detik. Enam buah *push-button* yang digunakan mewakili masing-masing nilai x , yaitu 10, 20, 30, 40, 50, dan 60. Penekanan *push-button* kedua kalinya akan membuat kondisi *default* berubah menjadi kondisi menyeberang, yaitu lampu hijau pada lampu A berubah menjadi merah, lampu merah pada lampu B berubah menjadi hijau, serta *seven segment* akan mulai menampilkan hitungan mundur dari x detik.

Selain *push-button*, *keypad* dapat digunakan untuk menentukan waktu menyeberang. Pada Gambar 9, penekanan angka *keypad* akan menentukan waktu menyeberang. Nilai x yang dihasilkan dengan *keypad* memiliki rentang nilai yang lebih luas, yaitu antara 0-99 detik. *Seven segment* akan mulai menampilkan hitung mundur setelah *keypad* ENT ditekan.

Pada kondisi ini, pejalan kaki menyeberang jalan. Transisi lampu merah menjadi hijau dan sebaliknya menggunakan lampu kuning. Setelah waktu penyeberangan mencapai 0, kondisi menyeberang berubah kembali menjadi kondisi *default* dan inisiasi tombol untuk kedua kalinya dapat digunakan setelah jeda waktu 5 menit.

Pilihan waktu menyeberang yang diterapkan pada prototipe cukup bervariasi. Waktu menyeberang dipengaruhi oleh lebar jalan dan kecepatan pejalan kaki. Ukuran umum lebar jalan adalah 3.25 sampai 3.75 m (Bang 1997). Sedangkan rata-rata kecepatan pejalan kaki adalah 3.5 ft/s atau sama dengan 1.1 m/s (LaPlante dan Kaeser 2007). Oleh karena itu, standar waktu menyeberang pejalan kaki dapat ditentukan dengan menghitung hasil bagi lebar jalan dengan kecepatan pejalan kaki. Rata-rata waktu menyeberang yang dapat dijadikan acuan untuk menyeberang adalah 20-30 detik (LaPlante dan Kaeser 2007).

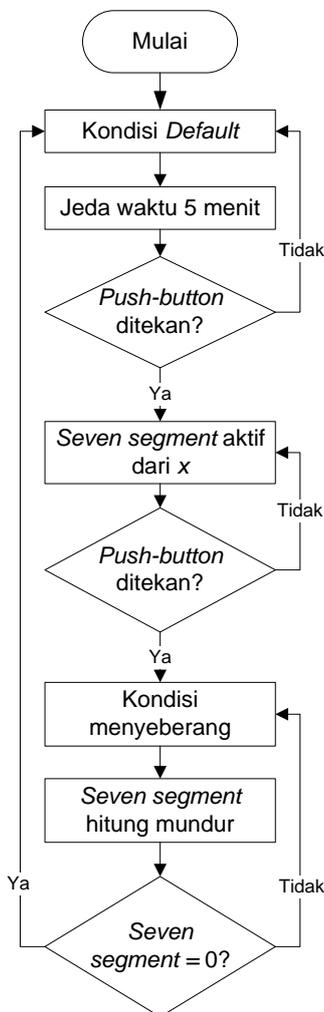
Perhitungan mundur (detik) yang ditampilkan pada *seven segment* menggunakan algoritme jam digital yang menggunakan sarana *timer* yang ditimbulkan oleh interupsi, ditunjukkan pada Gambar 10. Mikrokontroler jenis 89x52 mempunyai 3 perangkat *timer/counter*, yaitu *timer 0*, *timer 1*, dan *timer 2*. Perangkat *timer/counter* tersebut dapat diakses dengan menggunakan register khusus yang tersimpan dalam SFR (*special function register*). Dalam mengatur kerja *timer/counter* tersebut digunakan 2 register tambahan yang dipakai bersama oleh *timer 0* dan *timer 1*. Register tambahan tersebut adalah register TCON (*timer control register*) dan register TMOD (*timer mode register*).

Prototipe ini menggunakan *timer 0* yang diakses melalui register TL0 dan TH0. Indikator *overflow* memakai TF0 untuk membangkitkan *clock* jam digital yang akan ditampilkan pada *seven segment*, bagian detik puluhan (*display 6*), dan detik satuan (*display 7*). Mode *timer* yang digunakan adalah mode 1 (*timer 16 bit*). *Clock* yang digunakan untuk *timer* adalah *clock* internal mikrokontroler, yaitu 12 MHz, sehingga frekuensi pencacah adalah $12\text{ MHz}/12 = 1\text{ MHz}$. Pembagi 12 ini diperlukan karena 1 *machine cycle* mikrokontroler MCS-51 setara dengan 12 *clock cycle*.

Timer akan *overflow* setiap 65536 hitungan. Karena satu hitungan setara dengan 1 mikrodetik, *timer* akan *overflow* setiap 65536 mikrodetik. Hitungan *timer* yang akan digunakan setiap kali *overflow* adalah 50000 mikrodetik atau 0.05 detik dan 20 kali *overflow* akan setara dengan 1 detik. Angka 50000 dipilih agar hitungan menjadi bulat dan mudah dikonversi menjadi satuan detik. Untuk membangkitkan interupsi *timer* setiap 50000 mikrodetik, maka data awal yang harus diisikan pada register *timer* ialah sebagai berikut:

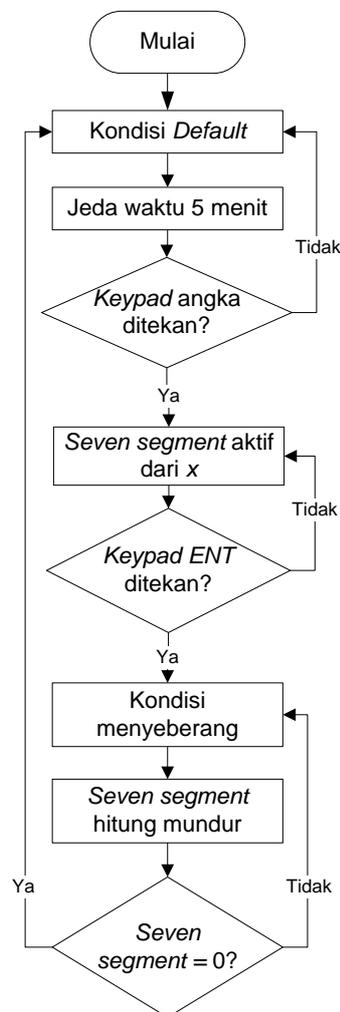
$$65536 - 50000 = 15536 \text{ (desimal) atau } 3CB0h$$

Nilai 3CB0h akan diberikan sebagai nilai awal *timer* dengan aturan data 3Ch diisikan sebagai nilai awal register TH0 dan B0h diisikan sebagai nilai awal register TL0.



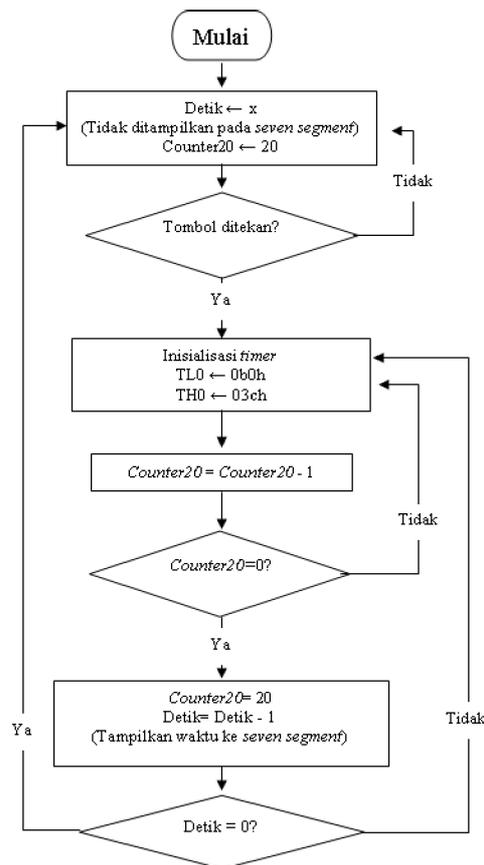
Gambar 8 Algoritme dengan *push-button*

Keterangan gambar:
 $x = 10, 20, 30, 40, 50, 60$



Gambar 9 Algoritme dengan *keypad*

Keterangan gambar:
 $x = 00-99$

Gambar 10 Algoritme interupsi *Timer 0*

Mikrokontroler keluarga MCS51 menyediakan fasilitas untuk 5 jenis sumber interupsi yaitu 2 sumber interupsi eksternal (melalui pin INT0 dan INT1 pada mikrokontroler) dan 3 sumber interupsi internal (*timer 0*, *timer 1*, dan *port serial*). Agar interupsi dapat dilayani, instruksi *assembly interrupt routine* harus ditempatkan pada alamat vektor tertentu, sesuai dengan sumber interupsi yang akan digunakan. Prototipe ini menggunakan interupsi internal *timer 0* (alamat 000Bh).

Evaluasi

Evaluasi kedua prototipe (Tabel 1) menunjukkan bahwa prototipe yang dikembangkan telah berhasil memenuhi seluruh persyaratan / fungsi yang diinginkan. Perbandingan kedua prototipe yang telah berhasil dikembangkan dapat dilihat pada Tabel 2.

Prototipe dengan *push-button* kurang fleksibel dalam penentuan waktu menyeberang, yaitu hanya 10, 20, 30, 40, 50, dan 60 detik. Sedangkan waktu menyeberang untuk prototipe dengan *keypad* dapat dipilih pada rentang nilai 0-99. Akan tetapi, prototipe dengan *push-button* lebih mudah digunakan dibandingkan dengan prototipe dengan *keypad*. Dalam implementasinya, kerusakan pada *push-button* dapat dilakukan hanya dengan mengganti *push-button* yang rusak saja. Sebaliknya untuk prototipe *keypad*, apabila *keypad* mengalami kerusakan, seluruh *keypad* harus diganti. Koreksi kesalahan dalam pemilihan waktu menyeberang dengan *push-button* lebih sulit dilakukan daripada prototipe dengan *keypad* karena pada *keypad* ada tombol yang digunakan untuk menghapus.

Tabel 1 Evaluasi prototipe

Fungsionalitas	Berhasil?
Kondisi <i>default</i> dapat ditampilkan	√
Kondisi menyeberang dapat diinisiasi dengan tombol yang terdiri atas <i>pushbutton</i> dan <i>keypad</i>	√
Waktu hitung mundur dapat ditampilkan pada <i>sevensegment</i>	√
Setelah waktu habis, lampu lalu lintas kembali ke kondisi <i>default</i>	√
Tombol tidak dapat ditekan kembali sebelum jeda waktu 5 menit	√
Prototipe akan memberikan waktu hitung mundur selama x detik. Nilai x yang tersimpan pada masing-masing <i>push-button</i> adalah 10, 20, 30, 40, 50, dan 60 detik. Nilai x yang dapat dihasilkan dengan <i>keypad</i> adalah 0-99 detik	√
Transisi lampu merah menjadi hijau dan sebaliknya menggunakan lampu kuning	√

Tabel 2 Perbandingan kedua prototipe yang dikembangkan

Perbandingan	Prototipe dengan <i>push-button</i>	Prototipe dengan <i>keypad</i>
Fleksibilitas penentuan waktu menyeberang	Kurang fleksibel	Lebih fleksibel
Kemudahan penggunaan	Lebih mudah	Lebih kompleks
Pemeliharaan	Lebih mudah	Lebih kompleks
Koreksi kesalahan penekanan	sulit	Mudah

SIMPULAN

Dua prototipe lampu lalu lintas bagi pejalan kaki berbasis mikrokontroler yang diinisiasi dengan tombol telah berhasil dikembangkan dengan menggunakan *push-button* dan *keypad*. Kedua prototipe telah memenuhi fungsi-fungsi yang diinginkan. Prototipe dengan *push-button* lebih mudah digunakan dan dipelihara, sedangkan prototipe dengan *keypad* memiliki fleksibilitas yang lebih tinggi, baik dalam hal penentuan waktu menyeberang maupun koreksi kesalahan penekanan. Prototipe dengan *push-button* masih memiliki kekurangan dalam hal koreksi kesalahan penekanan. Namun selanjutnya, algoritme *scan push-button* dapat diubah agar penekanan *push-button* tidak perlu berurutan. Kekurangan lainnya adalah jeda waktu penekanan tombol tidak dapat diubah selain melalui pemrograman ulang sehingga dapat ditambahkan modul CCTV untuk menangkap gambar *traffic* kendaraan dan penyeberang jalan.

DAFTAR PUSTAKA

- Bang KL. 1997. *Final Report: Indonesia Highway Capacity Manual and Software* (Kaji). Di dalam: *Indonesian Highway Capacity Manual Project*. 1997 Feb 25; Bandung (ID).
- Hamirani NS, Shaikh AA, Memon PM, Khatri YA. 2012. Wireless dual purpose propeller clock display. *Int J Eng Tech*. 2(3): 553-556.
- Jatmiko W, Arfan A, Krisnadhi AA, Kusumoputro B, Takagawa I, Sekiyama K, Fukuda T. 2009. Traffic signal control modification based on self-organizing in Indonesia. Di dalam: *International Symposiumon Micro-Nano Mechatronics and Human Science*; 2009 Nov 9-11; Nagoya (JP). hlm 598-601.
- LaPlante J, Kaeser TP. 2007. A history of pedestrian signal walking speed assumptions. Di dalam: *Urban Street Symposium* Ed ke-3; 2007 Jun 24-27; Seattle (US). hlm 1-8.
- Suyatno P, Istofa, Yuniarsari L. 2007. Rekayasa sistem pengatur parameter pesawat sinar-X diagnostik berbasis mikrokontroler keluarga MCS 51. Di dalam: *Seminar Nasional III SDM Teknologi Nuklir*; 2007 Nov 21-22; Yogyakarta (ID). hlm 337-344.