

Desain dan Algoritma Kontrol Rangkaian Ekspansi Sebagai Port Input-Output Pada Mikrokomputer AT89C2051

Elva Susianti¹, Putut Son Maria^{*2}

¹Teknologi Rekayasa Sistem Elektronika, Politeknik Caltex Riau

²Teknik Elektro, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau

Email: ¹elva@pcr.ac.id, ²putut.son@uin-suska.ac.id

Abstrak

Terbatasnya jumlah pin *input-output*(I/O) pada sebuah chip mikrokomputer seperti AT89C2051 berakibat pada sedikitnya jumlah perangkat eksternal yang dapat disambungkan ke *port* I/O. Pada sebuah sistem yang memprioritaskan efisiensi, penggunaan varian chip mikrokomputer yang memiliki lebih banyak pin I/O akan mengakibatkan tingkat efisiensinya menjadi rendah karena sejumlah pin I/O yang pada akhirnya tidak terpakai. Masalah terbatasnya jumlah pin I/O dapat diselesaikan dengan cara menambahkan komponen IC TTL 74373 ; karena memiliki fitur latch, sehingga data dapat dipertahankan selama beberapa waktu sesuai keinginan. Penambahan chip ini harus didukung dengan program pengontrol agar menghasilkan fungsi kerja yang mirip dengan *port* I/O. Penelitian ini bertujuan untuk membuktikan bahwa ide yang diusulkan dengan teknik di atas dapat berfungsi normal dan layak diimplementasikan secara praktis. Hasil uji simulasi menunjukkan bahwa meskipun dengan jumlah pin I/O 8 bit ternyata mampu dilakukan pengkabelan dengan perangkat eksternal hingga 16 bit dan bekerja secara normal melebihi ketersediaan *resource* standar dari IC AT89C2051. Teknik ini terbukti memberikan nilai efisiensi fungsi komponen lebih besar daripada nilai biaya komponen yang diperlukan.

Kata kunci: IC TTL 74373, Mikrokomputer AT89C2051, Port I/O

Abstract

Small number of I/O pins on a microcomputer chip such as the AT89C2051 has restricted the number of points that can be wired to external devices. In a system where it is desired that efficiency is the top priority, substitution using another variant of the chip which has more I/O pins is no a good solution. It will result in low efficiency because it leaves a number of unused I/O pins. A feasible solution to this problem is to exert a TTL 74373 chip which has a latch feature that enables it to hold output data for particular time. This chip should be controlled by a program to obtain its function similar to expansion port. This research aims to exhibit and demonstrate the technique. The circuit and algorithm were tested on virtual sistem modelling using proteus. The simulation shows a good result for the proposed concept, it proves that single byte port can serve more than its capacity with additional three bits yielding two bytes I/O wiring devices. This technique is worth considering on its cost and functionality..

Keywords: IC TTL 74373, Microcomputer AT89C2051, I/O Port

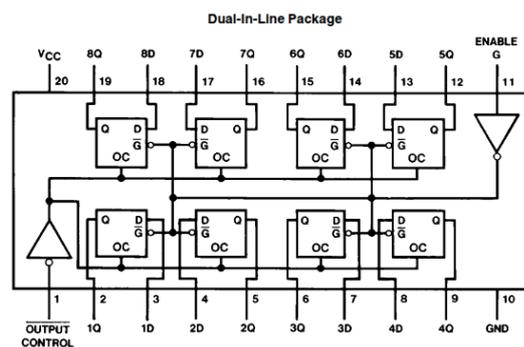
1. Pendahuluan

Mikrokontroler menjadi materi penting yang dipelajari pada institusi akademik [2]-[4],[6], sebagai bahan kajian pada buku teknik dan juga banyak digunakan dalam proyek ilmiah keteknikan seperti pada penelitian [7]-[16]. Atmel adalah salah satu brand sebagai produsen mikrokomputer yang dalam perkembangannya telah diakuisisi oleh perusahaan microchip. Salah satu produk mikrokomputernya adalah AT89C2051 yaitu mikrokomputer yang dikemas dalam bentuk *chip* 20 pin, dimana 15 diantaranya adalah sebagai I/O. Jumlah pin I/O yang sedikit berakibat pada sulitnya pembagian fungsi pengkabelan pada kondisi dimana perangkat di luar mikrokomputer berjumlah banyak sehingga melebihi jumlah maksimum titik sambungan I/O. Di sisi lain penggunaan varian mikrokomputer dengan jumlah pin yang lebih banyak mungkin akan memudahkan pengkabelan, namun demikian tingkat efisiensi dari *resource* I/O

chip tersebut mungkin menjadi rendah karena ada beberapa terminal I/O yang berlebih dan tidak terpakai.

Kemampuan sebuah mikrokomputer agar dapat dihubungkan dengan lebih banyak perangkat dapat ditingkatkan dengan cara menggunakan penambahan hardware seperti pada penelitian [19] atau dengan cara lain seperti teknik pooling, yaitu pengaturan pewaktuan saat pembacaan dan pengiriman data yang dilakukan secara bergantian. Teknik tersebut dapat diwujudkan dengan penambahan komponen IC 74373 sejumlah yang diperlukan dan didukung dengan program untuk kontrol pewaktuan baca-tulis data. Pada penelitian ini teknik pooling disimulasikan pada sebuah sistem yang terdiri dari satu unit mikrokomputer AT89C2051, satu unit liquid crystal display(LCD) dan digital input 8 bit berupa saklar dip-switch. Konsep rancangan dan algoritma yang diusulkan pada penelitian ini akan diuji secara simulasi untuk membuktikan apakah sistem yang dibangun dapat bekerja secara normal. Sistem yang diuji berupa rancangan skema hardware yang didukung dengan pemrograman untuk kontrol. Hasil uji yang bernilai positif dari penelitian ini pada akhirnya diharapkan dapat diimplementasikan secara praktis dalam bentuk perangkat keras yang sebenarnya.

Komponen yang dapat digunakan untuk menjalankan fungsi digital I/O setidaknya harus memiliki fitur *latch* yaitu mempertahankan logika *high* atau *low* pada waktu tertentu. Salah satu contoh komponen yang memiliki fungsi tersebut adalah IC TTL 74373 atau 74573 dimana *layout*-nya seperti ditunjukkan pada gambar 1. Dalam satu *chip* terdiri dari *flip-flop* sebanyak 8 unit dengan pin kontrol yang terhubung menjadi satu simpul. Berdasarkan *datasheet* 74373, tabel kebenaran dan fungsi kerja IC 74373 ditunjukkan seperti pada tabel 1.



Gambar 1. Struktur IC 74373

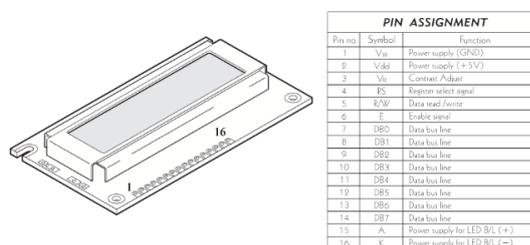
Fungsi *latch* terjadi jika sinyal logika *high* diberikan ke pin *enable*, kemudian pada saat terjadi transisi *falling edge*, maka data yang tersimpan pada *output* terus tetap dipertahankan sampai adanya data baru dan diikuti *falling edge* berikutnya. Jika data sudah terkunci(*latched*) pada jalur *output*, maka perubahan logika pada jalur data *input*(D) tidak akan mempengaruhi nilai logika *output* selama pada pin *enable* tidak terjadi transisi *high* ke *low*. Fitur inilah yang akan dimanfaatkan untuk saat pergantian data saat proses tulis dan baca secara bergantian.

Pada penelitian ini melibatkan penggunaan komponen LCD yaitu perangkat penampil visual yang lebih baik dalam menampilkan keragaman informasi dibandingkan dengan perangkat indikator ON/OFF. Jenis tampilan pada LCD umumnya berupa alfanumerik dan grafik. LCD alfanumerik adalah komponen yang sering dan banyak digunakan oleh banyak peneliti sebagai salah satu komponen dalam proyeknya seperti peneliti [16] dan [23]. LCD alfanumerik standar biasanya terdiri dari 16 pin yang terdiri dari 8 pin untuk jalur data, 3 pin untuk kontrol program, dan sisanya untuk pengaturan kontras dan *backlight* seperti ditunjukkan pada gambar 2.

Tabel 1. Tabel Kebenaran IC74373

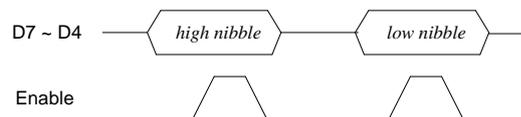
Output Control	Enable		Output Q
	G	D	
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

L = Low State, H = High State, X = Don't Care
 Z = High Impedance State
 Q₀ = Previous Condition of Q



Gambar 2. Layout fisik dan fungsi pin LCD tipe alfanumerik

Pada umumnya LCD tipe M1632 memiliki lebar jalur data 8 bit, namun penggunaannya dapat dipilih menggunakan mode 8 bit (*byte*) atau 4 bit (*nibble*). Pada penelitian ini menggunakan mode 4 bit karena alasan penghematan pengkabelan dengan konsekuensi bahwa data untuk LCD harus dikirim sebanyak dua kali dimulai dari *high nibble* dan diikuti *low nibble*. Diagram pewaktuan pengiriman data ditunjukkan pada gambar 3.



Gambar 3. Diagram pewaktuan pemrograman penulisan data mode 4 bit

2. Metode Penelitian

2.1 Blok Diagram Sistem

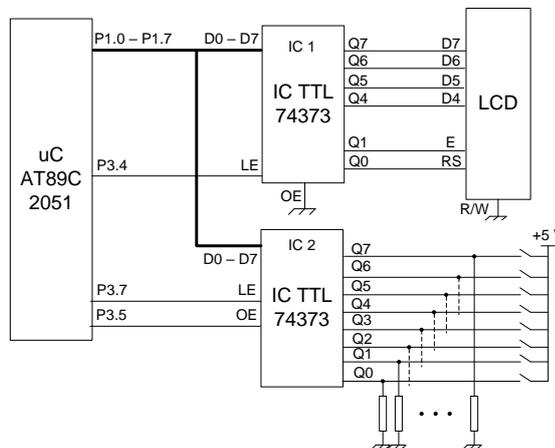
Konsep diagram pengkabelan untuk mensimulasikan teknik yang diusulkan pada penelitian ini ditunjukkan pada gambar 4. Port 1 dari mikrokomputer AT89C2051 berfungsi sebagai *output* jalur data sekaligus menyediakan sinyal kontrol untuk penulisan data menuju LCD melalui IC 1. Pada saat pembacaan data digital dari saklar, maka Port 1 berfungsi sebagai *input* jalur data melalui IC 2. Pewaktuan harus diatur sedemikian rupa sehingga peran Port 1 sebagai *output* dan sebagai *input* tidak terjadi pada waktu yang bersamaan karena berpotensi terjadi tabrakan antara data *output* dengan data *input* sehingga data yang ditulis ke LCD menjadi tidak valid.

2.2 Diagram Alir Program

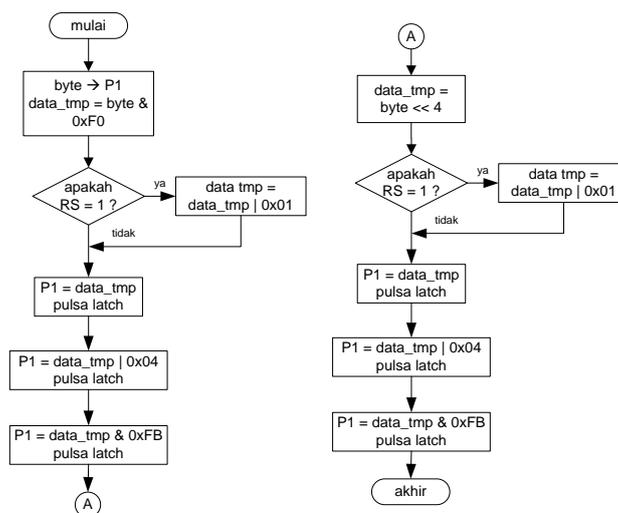
2.2.1 Program Tulis ke LCD

Diagram alir (*flowchart*) program penulisan data ke LCD ditunjukkan pada gambar 5. Sesuai pewaktuan yang dijelaskan pada gambar 3, data *byte* harus dipisah menjadi *nibble* karena mode pengkabelan data LCD yang digunakan adalah 4 bit. *High nibble* harus dikirimkan pertama dan kemudian diikuti dengan *low nibble*. Pada *flowchart* tercantum satu *pseudo*

instruction yaitu pulsa *latch* yang di dalamnya berisi perintah untuk memberikan logika *high* selama beberapa milidetik dan setelah itu berubah menjadi *low* atau disebut juga sebagai *falling edge*. Perintah pulsa *latch* dimunculkan pada pin Port 3 bit ke - 4 yaitu pin kontrol untuk *enable* IC TTL 74373.



Gambar 4. Blok diagram sistem



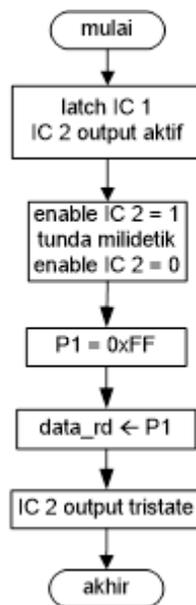
Gambar 5. Diagram alir penulisan ke LCD

Prosedur untuk penulisan data ke LCD dimulai dengan menyediakan data nibble pada Port 1 pada bit 4 sampai 7, data nibble ini nantinya diteruskan oleh IC 1 sehingga muncul pada jalur Q4 sampai Q7. Berikutnya data nibble ini harus dikondisikan sedemikian rupa sehingga pada pin Q1 dari IC 1 nantinya akan muncul sinyal falling edge. Setiap data yang dituliskan ke LCD harus dilengkapi penyematan sinyal latch seperti flowchart gambar 5. Dengan teknik ini maka tidak diperlukan komponen penyesuai level tegangan untuk keperluan komunikasi data seperti pada penelitian [20]

2.2.2 Program Baca dari Saklar

Diagram alir pembacaan data *input* ditunjukkan pada gambar 6. Pada saat pembacaan data input *digital* seperti misalnya dari saklar, maka harus dipastikan bahwa IC 1 dalam kondisi mengunci data *output*-nya sehingga pada saat terjadi pembacaan melalui jalur Port 1 maka

data *output* dari IC 1 tidak akan terpengaruh. Pembacaan data *input* dapat dilakukan dengan cara mengkondisikan pin *output enable* dari IC 2 sehingga aktif disertai dengan memunculkan pulsa *latch* selama beberapa saat. Pada saat data dari saklar sudah diteruskan pada jalur *output* IC 2, maka Port 1 sudah dapat digunakan untuk membaca data dengan cara mengkondisikan semua bit pada Port 1 agar berlogika *high* semuanya [1], kemudian barulah setelah itu data dari Port 1 dapat disalin ke variabel dalam program.



Gambar 6. Diagram alir pembacaan dari saklar

3. Hasil dan Analisa

Kode program untuk pengendali rangkaian ditulis menggunakan bahasa program C dan dikompilasi menggunakan *library small device c-compiler(sdcc)* yang terdapat dalam M-IDE [24]. Program terdiri dari tiga bagian penting yaitu sub-program untuk penulisan data ke LCD, sub-program pembacaan data dan program utama. Struktur program diawali dengan deklarasi *library* yang digunakan dalam program. Terdapat 4(empat) *library* yang digunakan pada penelitian ini yaitu dua *library* berisi daftar register(8051.h dan at89x51.h), satu *library* standar *compiler ANSI C(string.h)* dan satu *library* yang dikreasi sendiri(*delay.h*). Penamaan pin kontrol dipermudah dengan memberikan nama alias pada bagian awal program. Deklarasi dan definisi nama alias pada pin kontrol ditunjukkan pada gambar 7. Mengacu pada gambar 4, pin kontrol *latch enable* IC 1 terhubung ke Port 3 bit ke-4 dan dinamakan WR, sedangkan pin *output enable* dan *latch enable* untuk IC 2 berturut-turut dinamakan OE dan RD. Mengacu pada diagram alir pada gambar 5, kode program untuk penulisan data ke LCD terdiri dari dua bagian yang polanya sama. Bagian pertama adalah pengiriman *high nibble* yang diikuti dengan *low nibble*. Sub-program atau *function* pada gambar 8 dirancang menerima parameter *input* berupa data *byte*, dan dapat digunakan untuk penulisan ke register perintah atau register data pada LCD, oleh karena itu sub-program tersebut membutuhkan dua parameter *input* yaitu data *byte* dan nilai register target.

```
#include<8051.h>
#include<at89x51.h>
#include<delay.h>
#include<string.h>

#define WR P3_4
#define OE P3_5
#define RD P3_7
```

Gambar 7. Deklarasi library dan definisi pin

Data *byte* akan disimpan pada variabel *data_tmp* yaitu variabel 8 bit yang menyimpan data sementara. Selanjutnya nilai *data_tmp* akan di-AND-kan dengan 0xF0 untuk mendapatkan *high nibble* karena posisi bit pada *high nibble*(D4 - D7) akan diteruskan ke IC 74373. Proses berikutnya adalah seleksi nilai register pada parameter *input*. Jika nilai register adalah satu maka data ditujukan ke register data pada LCD, jika nilai register adalah nol maka ditujukan ke register perintah. Nilai variabel pada *data_tmp* yang sudah siap dikirim akan dimunculkan pada Port 1 dan disusul dengan mengirim pulsa *latch enable* ke IC 1. Mekanisme tersebut akan berlaku juga pada *low nibble* sebelum data dikirimkan ke LCD.

```
void write(char data_in,char reg)
{
    unsigned char data_tmp;
    // persiapan data HIGH NIBBLE
    data_tmp = (data_in & 0xF0);
    if (reg == 1) {data_tmp = data_tmp | 0x01;}
    // latch 1 -> enable = 0
    P1 = data_tmp;
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
    // latch 1 -> enable = 1
    P1 = data_tmp | 0x04; //
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
    // latch 1 -> enable = 0
    P1 = data_tmp & 0xFB; //
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
    // persiapan data LOW NIBBLE
    data_tmp = (data_in << 4);
    if (reg == 1) {data_tmp = data_tmp | 0x01;}
    // latch 1 -> enable = 0
    P1 = data_tmp;
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
    // latch 1 -> enable = 1
    P1 = data_tmp | 0x04; //
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
    // latch 1 -> enable = 0
    P1 = data_tmp & 0xFB; //
    WR = 1;delay_ms(5);WR = 0; //pulsa latch
}
```

Gambar 8. Kode sub-program penulisan data ke LCD

Diagram alir pada gambar 6 diimplementasikan dalam bentuk kode sub-program atau *function* seperti pada gambar 9. IC 1 harus di-*latch* agar data *output*-nya tidak berubah selama terjadi perubahan logika pada jalur Port 1. Berikutnya IC 2 dikondisikan sedemikian rupa sehingga pada pin *output enable* dan *latch enable*-nya aktif sehingga data dari saklar dapat muncul pada jalur Port 1. Agar data dapat diduplikasi oleh variabel dalam program, semua bit

pada Port 1 harus dikondisikan berlogika *high* sebelum perintah *assignment* dari Port 1 ke variabel `data_rd`.

```
unsigned char read()
{
    unsigned char data_rd;
    WR = 0; //memastikan IC U2 off
    OE = 0;
    RD = 1;delay_ms(1);RD = 0;
    P1 = 0xFF; // P1 harus dikondisikan HIGH sbg input
    data_rd = P1;
    OE = 1; //memastikan IC U3 high impedance
    return (data_rd);
}
```

Gambar 9. Kode program pembacaan data *byte*

Prosedur standar yang harus dilakukan saat hendak mengkomunikasikan antara mikrokomputer dengan LCD adalah proses inisialisasi. Tujuannya adalah untuk pengaturan konfigurasi hardware pada LCD, seperti misalnya pengaturan posisi kursor, lebar data, kedipan kursor dan lain sebagainya. Sesuai data sheet[17] prosedur inisialisasi LCD pada penelitian ini menggunakan kode program seperti pada gambar 10.

```
void init_lcd()
{
    write(0x03,0);delay_ms(1);
    write(0x02,0);delay_ms(1);
    write(0x28,0);delay_ms(1);
    write(0x0c,0);delay_ms(1);
    write(0x06,0);delay_ms(1);
    write(0x80,0);delay_ms(1);
}
```

Gambar 10. Kode program inisialisasi LCD

Tipe LCD alfanumerik yang digunakan pada skema penelitian ini berspesifikasi 16 kolom 2 baris. Layar tampilan LCD yang lebar memudahkan menampilkan pesan dalam bentuk *string*. Oleh karena itu pada penelitian ini juga membuat kode sub-program *function* seperti pada gambar 11. Dalam bahasa program C terdapat tipe data berupa *string* yaitu adalah sekumpulan karakter individual yang dirangkai menjadi satu entitas atau dapat dianggap sebagai larik atau *array*. Secara teknis, menampilkan *string* ke LCD tetap harus dikirimkan karakter per karakter sampai keseluruhan karakter yang merupakan anggota dari *array* terkirim semuanya. Kode program kirim_pesan pada gambar 11 menggunakan *function* standar dari *compiler* bahasa C yaitu *strlen()* yang terdapat dalam *library string.h*.

```
void kirim_pesan(char pesan[])
{
    unsigned char indeks;
    for(indek=0;indek < strlen(pesan);indek++)
    {
        write(pesan[indek],1);
    }
}
```

Gambar 11. Kode program pengiriman *string* ke LCD

Penelitian ini juga menampilkan kode program untuk pengaturan posisi kursor pada LCD seperti pada gambar 12. Pengaturan posisi kursor dilakukan dengan cara mengirimkan data *byte* ke register perintah, data menyesuaikan alamat dasar DDRAM yang ada pada LCD. Baris 1 dan 2 dari layar LCD memiliki alamat *offset* mulai 80 Hex dan C0 Hex.

```
void lcd_gotoxy(unsigned char row, unsigned char col)
{
    if (row = 1) write((0x80 + col),0);
    if (row = 2) write((0xC0 + col),0);
}
```

Gambar 12. Kode program pengaturan koordinat kursor

Setelah semua kode sub-program atau *function* partikelir terbentuk, maka semua perintah yang dirancang untuk menguji dan mengendalikan perangkat keras yang diusulkan pada penelitian ini dikemas dalam program utama. Kode program utama pada penelitian ini ditunjukkan pada gambar 13. Program terdiri dari baris deklarasi variabel, inialisasi LCD, pengiriman string dan pembacaan data digital yang kemudian ditampilkan ke LCD.

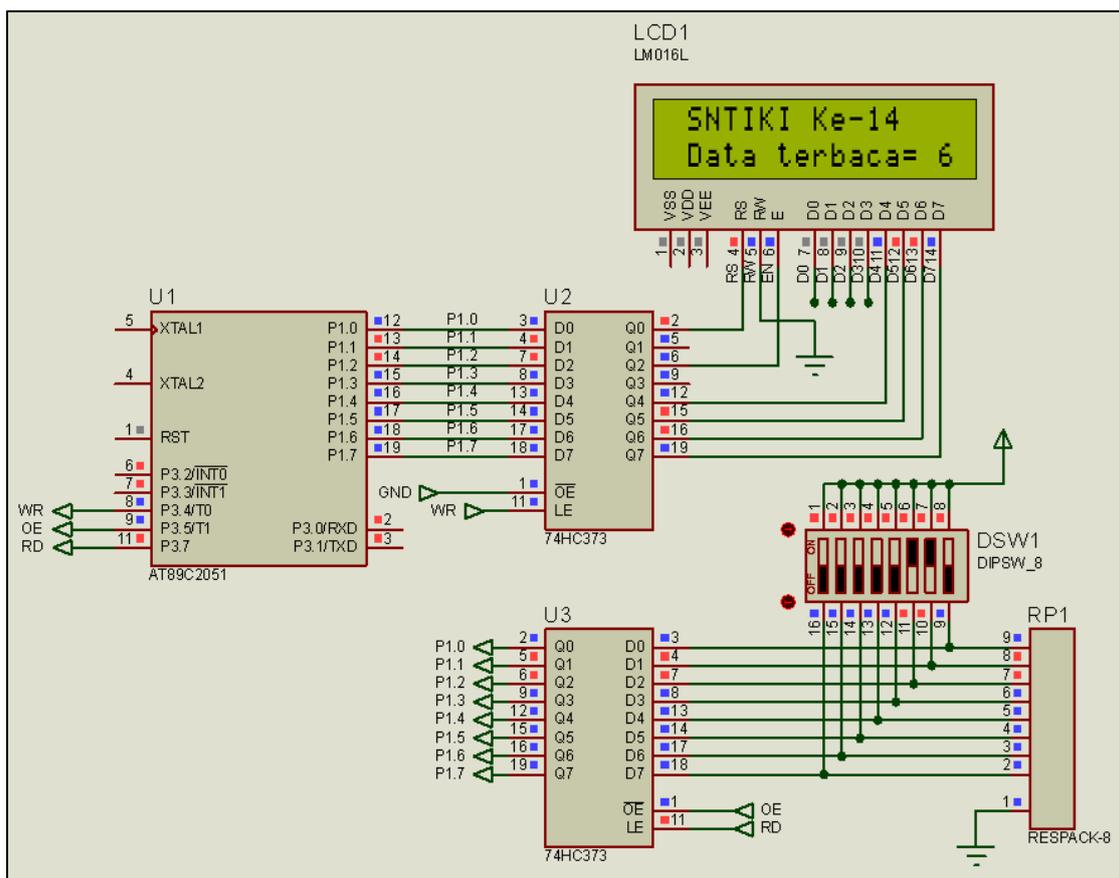
Terdapat dua *string* yang dikirimkan dan sengaja ditampilkan secara tetap yang tersimpan pada variabel teks1 dan teks2. Setelah program selesai mengirimkan *string* maka berikutnya program akan melakukan *looping* dimana data dibaca dari IC 2 untuk kemudian ditampilkan ke LCD. Data yang terbaca oleh Port 1 dari IC 2 adalah dalam format biner, sehingga untuk mengubah data biner menjadi karakter ASCII maka data biner harus ditambah dengan 30 Hex, ini karena LCD hanya akan menampilkan data yang sesuai dengan karakter ASCII. Data yang terbaca dari IC 2 akan ditampilkan pada baris 2 kolom 15 pada layar LCD.

Skema rangkaian ekspansi untuk pengujian sistem dan hasil *running* program ditunjukkan pada gambar 14. Program simulator yang digunakan adalah Proteus Versi 8. Penggambaran skema pada simulator tidak menampilkan keseluruhan komponen secara rinci seperti halnya pada perangkat keras fisik, karena pada program aplikasi Proteus telah mengakomodasi parameter wajib seperti catu daya, rangkaian osilator, sistem reset dan lain sebagainya. Oleh karena itu pada gambar 14 tidak menampilkan rangkaian untuk osilator dan sistem reset karena sudah tersemat dalam *library* di program simulator Proteus. Tampilan pada LCD dari gambar 14 membuktikan bahwa algoritma dan rangkaian ekspansi I/O yang diusulkan pada penelitian ini dapat bekerja secara selaras dan normal. LCD menampilkan pesan sesuai dengan program utama dan data biner yang terbaca melalui Port 1 dapat ditampilkan seketika. Pada saat terjadi perubahan data *input*, maka mikrokomputer langsung mengirimkan data terkini ke LCD sehingga tampilan pada layar sesuai dengan data biner yang terbaca saat itu. Proses pengiriman data ke LCD (*via* IC 1) dan pembacaan data dari saklar (*via* IC 2) keduanya dilakukan melalui jalur yang sama yaitu Port 1, dengan demikian penggunaan *resource* Port 1 lebih efisien karena dapat digunakan untuk menangani dua data dengan arah vektor yang berbeda. Dari gambar 14 terhitung masih menyisakan dua pin yang dapat digunakan sebagai *output* pada IC 1 yaitu pin Q1 dan Q3, hal ini sekaligus merupakan poin positif karena dengan menggunakan konsep yang diusulkan pada penelitian ini maka efisiensi penggunaan *resource* dari mikrokomputer dapat ditingkatkan.

```

void main()
{
    unsigned char x, data_old = 0xff;
    char teks1[] = " SNTIKI Ke-14 ";
    char teks2[] = "Data terbaca= ";
    init_lcd();
    kirim_pesan(teks1);
    lcd_gotoxy(2,1);
    kirim_pesan(teks2);
    while(1)
    {
        x = read();
        if (data_old != x)
        {
            lcd_gotoxy(2,15);
            write(x + 0x30,1);
            data_old = x;
        }
    }
}
    
```

Gambar 13. Kode program utama



Gambar 14. Hasil tangkapan layar *running* program

4. Kesimpulan

Masalah keterbatasan jumlah port I/O pada mikrokomputer AT89C2051 dapat diatasi dengan cara menambahkan komponen IC 74373 dan didukung dengan program kontrol. Dengan menambahkan dua unit IC 74373 ternyata dapat dilakukan pengkabelan untuk

menampilkan informasi ke LCD dan membaca data digital 8 bit melalui jalur Port yang sama. Dengan menggunakan teknik *pooling* maka fungsi port untuk kirim-terima data lebih menjadi lebih efisien dan dengan tetap berfungsi secara baik dan normal. Pengembangan berikutnya dapat dilakukan karena masih terdapat pin yang tersisa yang dapat digunakan untuk pengkabelan untuk lebih banyak piranti lainnya. Dengan demikian konsep yang diusulkan pada penelitian ini terbukti telah berfungsi secara baik dan normal dan dapat diaplikasikan secara praktis menggunakan perangkat keras sebenarnya.

Referensi

- [1] AT89C2051 Datasheet. Diakses : 15 Mei 2022 [online]. Tersedia di : <https://www.microchip.com/sitesearch/search/All/AT89C2051>.
- [2] Patmasari, R., Wijayanto, I., Nugraha, R., Diktat mk mikroprosesor dan antarmuka, Fakultas Teknik Elektro, Universitas Telkom, 2014.
- [3] Sugiono, D., Teknik mikroprosesor, PPPPTK BOE Malang. Kementerian Pendidikan & Kebudayaan, 2013.
- [4] Suprianto, D., Firdaus, V.A.H., Agustina, R., Wibowo, D.W., "Microcontroler arduino untuk pemula," Penerbit Jasakom-Malang, 2019.
- [5] Modul Praktikum Mikrokontroler, Universitas Pembangunan Jaya. [online]. Diakses : 20 Mei 2022 Tersedia pada : <https://ocw.upj.ac.id/files/RPS-INF204-INF204-Modul-Mikrokontroler-dan-Project.pdf>
- [6] Suhaeb, S., Djawad, Y. A., Jaya, H., Ridwansyah, Sabran, Risal, A., Mikrokontroler dan interface. Buku Ajar Jurusan Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas negeri Makassar, 2017.
- [7] Zimmy, Muslim, A., "Robot pengikut jalur menggunakan sensor infra merah dan berbasis mikrokontroler at89c2051," skripsi, Fakultas Ilmu Komputer, Universitas Gunadarma, 2008.
- [8] Mardiyanto, E., Hermanto, L., "Miniatur score board dengan menggunakan mikrokontroler at89c2051," skripsi, Fakultas Ilmu Komputer, Universitas Gunadarma, 2008.
- [9] Sari, N., "Sistem pengaman sepeda motor berbasis mikrokontroler at89c2051," skripsi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas, 2006.
- [10] Siagian, L.M., "Perancangan receiver dan transmitter infra merah keyboard decoder pada display matriks berbasis mikrokontroler at89c2051," skripsi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatra Utara, 2008.
- [11] Samsudin, F.A., "Alat ukur tinggi badan otomatis berbasis mikrokontroler at89c2051. ilmu komputer," skripsi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret, 2010.
- [12] Sutikno, T., Yudhana, A., Siprian, D., "Pengaturan sakelar pada acara cepat tepat berbasis mikrokontroler at89c2051," Telkomnika. 04(03). 185-192., 2006.
- [13] Suleman, Anwar, T. , "Prototipe rancangan alat pengendali lampu gedung berbasis android dengan mikrokontroler at89c2051," Indonesian Journal on Networking and Security. 05(04). 01-06., 2016.
- [14] Ermabuana, R., "Alat pengoperasian peralatan listrik melalui saluran telepon berbasis mikrokontroler at89c2051," skripsi, Teknik Elektro, Universitas Jember, 2005.
- [15] Haloho, T.P.D., "Perancangan sistem pengaman gedung dan peralatan listrik terhadap bahaya kebakaran berbasis mikrokontroler at89c2051," skripsi, Teknologi Instrumentasi Pabrik, Fakultas teknik, Universitas Sumatera Utara, 2008.
- [16] Suryantoro, H., Budiyanto, A, "Prototipe sistem monitoring level air berbasis labview & arduino sebagai sarana pendukung praktikum instrumentasi sistem kendali," Indonesian Journal of Laboratory, 01(03), 20–32., 2019.
- [17] M1632 Datasheet. Diakses : 18-Mei-2022. [online]. Tersedia di : https://www.datasheetarchive.com/M1632*%20lcd%20display-datasheet.html
- [18] LCD datasheet. Diakses : 19-Mei-2022. [online]. Tersedia di : https://www.datasheetarchive.com/M1632*%20lcd%20display-datasheet.html
- [19] Siregar, O.Y., "Aplikasi perancangan pengelolaan parkir menggunakan mikrokontroler at89c2051," skripsi, Fisika Instrumentasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, 2013.
- [20] Yandani S. E., "Rancang bangun alat pengontrol perangkat elektronika berbasis pc menggunakan komunikasi data serial dengan mikrokontroler at89c2051," skripsi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas, 2006.
- [21] 74373 datasheet. Diakses : 19-Mei-2022. [online]. Tersedia di : <https://www.alldatasheet.com/datasheet-df/pdf/51079/FAIRCHILD/74373.html>

- [23] Saputra, A. MD., Amarudin, Utami, R. N., Setiawan, "Rancang bangun alat pemberi pakan ikan menggunakan mikrokontroler," J. ICTEE, 01(01). 15-19, 2020.
- [24] M-IDE Studio for MCS-51. [online] Tersedia di : www.opcube.com.