

ANALISIS KEMAMPUAN KERJA MIKROKONTROLER MCS-51 SEBAGAI FREQUENCY COUNTER

Putut Son Maria¹, Elva Susianti²

¹Universitas Islam Negeri Sultan Syarif Kasim Riau

²Politeknik Caltex Riau

E-mail : putut.son@uin-suska.ac.id¹

ABSTRAK

Generator fungsi atau *Function generator* adalah alat yang menghasilkan sinyal listrik dengan amplitudo dan frekuensi yang dapat diubah-ubah. Pada kegiatan praktikum di laboratorium elektronika, perangkat generator fungsi dan *oscilloscope* adalah dua alat ukur yang wajib untuk menunjang proses pembelajaran mahasiswa. Ketiadaan fitur penampil frekuensi pada sebuah *function generator* mengakibatkan mahasiswa pemula menjadi sulit untuk memperkirakan besaran frekuensi yang diinginkan dan efeknya adalah proses praktikum(pengambilan data) menjadi lebih lama. Penelitian ini memaparkan rancangan *frequency counter* yang dapat diaplikasikan untuk pengukuran rentang frekuensi rendah hingga menengah menggunakan sistem digital. Rancangan dibangun menggunakan mikrokontroler AT89C2051 dari keluarga MCS-51 yang memiliki fitur minimum agar nilai efisiensi dari *resource*-nya menjadi tinggi. Hasil desain *frequency counter* pada penelitian ini terbukti secara simulasi dapat berfungsi secara baik dengan kemampuan daya ukur frekuensi maksimum sebesar 43 KHz menurut perhitungan *machine cycle*, walaupun demikian secara simulasi ternyata sistem mampu mengukur hingga 47 KHz dengan nilai *error* 0.223% namun dengan konsekuensi tampilan yang lambat.

Kata kunci : Function Generator, Frequency Counter, MCS-51

1. PENDAHULUAN

Generator fungsi atau *function generator* adalah alat penghasil gelombang listrik yang dapat diatur bentuk gelombang, frekuensi dan amplitudonya. Generator fungsi menjadi alat laboratorium yang penting dan menjadi salah satu yang cukup tinggi pemakaian jam penggunaannya pada laboratorium elektronika dasar dan rangkaian listrik. Fitur pada sebuah alat generator fungsi tergantung pada *brand* dan variannya yang biasanya berbanding lurus dengan nilai nominal harganya. Pada varian paling sederhana dan terjangkau, fitur yang dimiliki generator fungsi belum dilengkapi dengan *display* informasi yang menunjukkan besaran frekuensi yang terhasikan. Hal ini mengakibatkan mahasiswa pemula harus menggunakan osiloskop untuk dapat mengamati bentuk gelombang, mengatur *setting* yang pas dan menghitung manual untuk mengetahui besaran frekuensi dari sinyal generator fungsi, efeknya adalah mahasiswa perlu waktu lebih untuk sekedar mengetahui nilai frekuensinya.

Frekuensi rendah sampai menengah adalah nilai yang sering digunakan pada praktikum pada laboratorium elektronika dasar dan rangkaian listrik, seperti untuk praktikum *rectifier*, *amplifier* dan *low pass filter*[1][2]. Pada sisi lain, informasi berupa

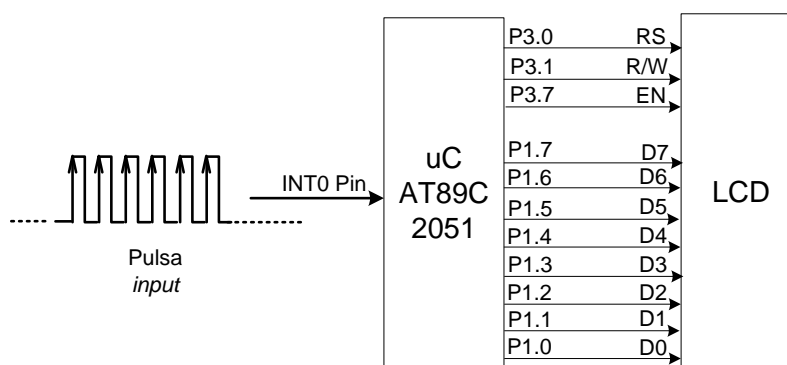
deretan angka yang menunjukkan besaran frekuensi akan sulit diwujudkan oleh rangkaian analog, sehingga pilihan yang rasional adalah menggunakan rangkaian yang merupakan sistem digital misalnya mikrokontroler seperti pada [4-10]. Namun demikian dalam perancangan perangkat keras, efisiensi dari sistem tetap harus dipertimbangkan agar daya guna dari *resource* digital dapat tergunakan secara optimal. Pada penelitian ini akan mengkaji dan memberikan perspektif secara teknis terkait efisiensi penggunaan resource sehingga skema yang diusulkan dalam penelitian ini akan mudah diaplikasikan dan berdaya guna optimal.

2. METODE PENELITIAN

Bentuk kegiatan dalam penelitian ini secara bertahap terdiri dari tinjauan literatur baik berupa paper dan manual sheet, perancangan sistem perangkat frequency counter, simulasi hasil perancangan dan analisis kinerja. Pembahasan akan dimulai dari perancangan sistem sampai pada sub-item yang penting, dan pendekatan untuk pengukuran kinerja adalah dari hasil simulasi menggunakan program simulator.

2.a Blok diagram *frequency counter*

Blok diagram pengkabelan sederhana dari rangkaian yang diusulkan untuk pengujian secara simulasi ditunjukkan pada gambar 1. Sistem terdiri dari satu unit mikrokontroler dan satu unit *Liquid Crystal Display*(LCD). Mikrokontroler AT89C2051 dipilih karena berdimensi kompak dan memiliki frekuensi kerja yang cukup tinggi sehingga dipertimbangkan mampu digunakan untuk pengukuran frekuensi menengah. Gelombang sinyal masukan dihubungkan dengan pin INT0 karena pada pin tersebut tersedia fitur yang mampu memicu interupsi secara *hardware* sehingga pada alamat vektor interupsi dalam programnya dapat diketikkan perintah untuk menghitung naik setiap terdeteksi adanya perubahan level logika. Teknik ini berbeda dengan penelitian [5][6][8] yang menggunakan *limiter* berupa IC 74HC14 dan *binary counter* 74393. Penelitian ini tidak menggunakan IC TTL karena berpotensi mengurangi akurasi perhitungan akibat adanya *propagation delay* pada gerbang logika[11]. LCD digunakan sebagai perangkat *display* untuk menampilkan besaran frekuensi, secara kepraktisan, pengkabelan LCD lebih sederhana dan memiliki keragaman tampilan karakter yang lebih jika dibandingkan dengan *seven segment*. Penggunaan media penampil LCD juga lebih efisien, pengkabelannya lebih ringkas dan dengan konsumsi daya yang rendah[5][6].

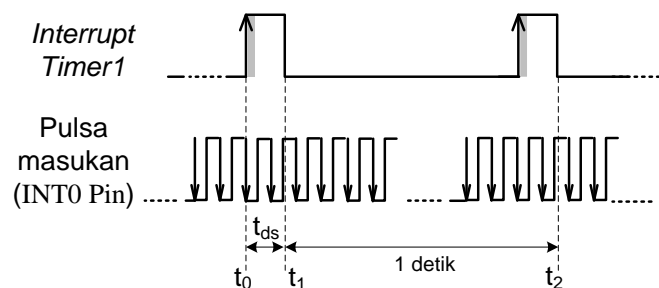


Gambar 1. Blok diagram sistem

2.b Diagram waktu(*time chart*) cara kerja *frequency counter*

Frequency counter atau frekuensi meter adalah alat yang berfungsi untuk menghitung frekuensi sinyal listrik tanpa terpengaruh pada amplitudo dan jenis gelombang atau fungsi

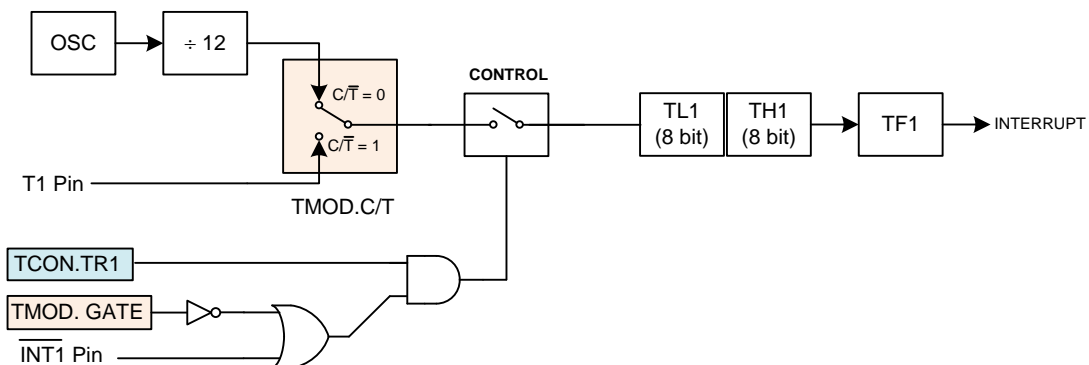
matematik gelombang. Berdasarkan definisi umum, maka sebuah alat *frequency counter* harus memiliki basis waktu yang menghasilkan durasi setiap satu detik untuk menandai awal dan akhir perhitungan banyaknya gelombang yang masuk. Teknik ini juga digunakan pada penelitian[4]. Diagram pewaktuan prosedur perhitungan sinyal atau pulsa listrik pada penelitian ini ditunjukkan pada gambar 2. Sistem bekerja berdasarkan basis waktu yang dihasilkan oleh *Timer1* pada mikrokontroler yang diprogram untuk menampilkan hasil hitungannya setiap satu detik. Selama mikrokontroler menampilkan informasi ke LCD ($t_0 \sim t_1$) maka interupsi pada pin INT0 akan dinonaktifkan sementara agar tidak mengganggu jalannya perintah dalam blok program pada vektor interupsi *Timer1*. Pada saat mikrokontroler selesai mengerjakan perintah untuk menampilkan informasi ke LCD, maka interupsi pada INT0 akan diaktifkan kembali agar kode perintah pada vektor interupsi INT0 dapat dikerjakan mikrokontroler.



Gambar 2. Konsep dasar perhitungan frekuensi

2.c Fitur *timer* dan *interrupt* pada MCS-51

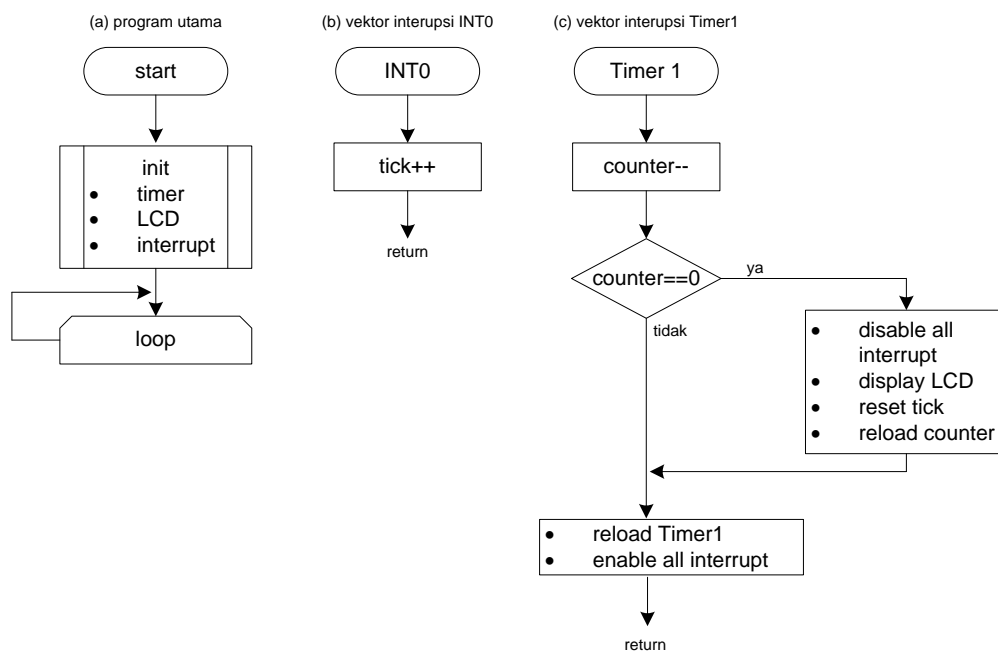
Keluarga mikrokontroler MCS-51 memiliki fitur *timer* dengan diagram seperti ditunjukkan pada gambar 3[3]. Fungsi *timer* ditentukan oleh pengaturan pada register TCON dan TMOD. Frekuensi kerja dari mikrokontroler AT89C2051 maksimal adalah 12 Mhz dimana kapasitas register *Timer1* adalah 16 bit, maka kemampuan maksimum *Timer1* menghasilkan interupsi adalah sebesar 65535 mikrodetik(65.5 milidetik). Agar dapat mencapai pewaktuan setiap interval 1 detik, maka *Timer1* hanya akan ditetapkan untuk menghasilkan interupsi setiap 10 milidetik ditambah dengan algoritma pada vektor interupsi *Timer1* disematkan sebuah variabel yang nilainya bertambah 1 setiap terjadi interupsi, kemudian jika telah mencapai nilai 100 maka mikrokontroler akan memperbarui tampilan frekuensi pada LCD.



Gambar 3. Blok diagram sistem *timer* pada MCS-51

2.d Diagram alir program

Rancangan yang diusulkan pada penelitian ini adalah berbasis pada mikrokontroler MCS-51 dengan memanfaatkan fitur interupsi eksternal dan timer. Karena aliran program dipengaruhi oleh interupsi akibat kejadian (*event*) dan interupsi oleh pewaktu (*timer*) maka representasi diagram alir program secara keseluruhan ditunjukkan seperti gambar 4. Program utama berisi inialisasi *timer*, pengaturan interupsi dan LCD, setelah itu program utama hanya akan *looping* menunggu terjadinya interupsi (gambar 4.a). Hal ini sengaja dirancang demikian agar mikrokontroler dapat menghitung secara akurat setiap terjadi interupsi akibat *event* dari pin INT0. Jika sewaktu-waktu terjadi interupsi dari pin INT0, maka mikrokontroler akan menaikkan hitungan[4] pada variabel *tick* sebagai penanda bahwa telah ada satu gelombang yang masuk (gambar 4.b). Pada vektor interupsi *Timer1* terdiri dari dua kemungkinan *handling* yaitu kondisi jika belum tercapai 1 detik dan setelah tercapai 1 detik, namun demikian esensi dari vektor interupsi *Timer1* adalah *handling* saat telah tercapai 1 detik.

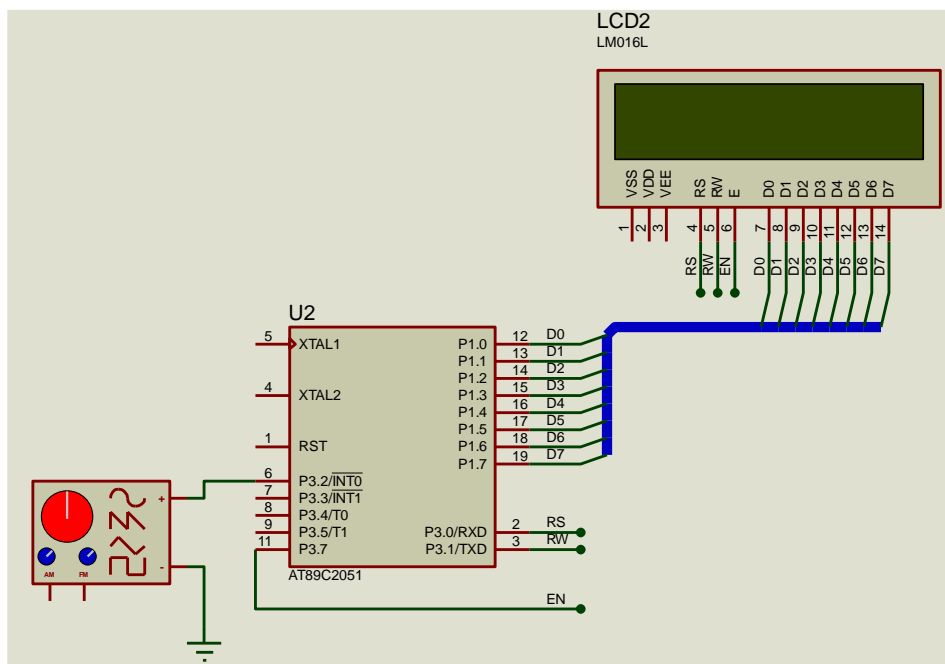


Gambar 4. Diagram alir program

3. HASIL DAN PEMBAHASAN

3.1 Skema rangkaian

Perancangan sistem *frequency counter* pada penelitian ini diwujudkan dalam bentuk skema rangkaian ditunjukkan pada gambar 5. Sumber gelombang listrik yang menjadi masukan dihubungkan ke terminal INT0(P3.2) dengan prakondisi bahwa level tegangan maksimum 5 volt dan bentuk gelombangnya adalah kotak. Sistem akan menampilkan informasi ke LCD melalui Port 1 dengan mode data 8 bit.



Gambar 5. Skema rangkaian

3.2 Kode program

Kode program ditulis memakai bahasa program c menggunakan aplikasi M-IDE yaitu *editor* dan sekaligus *compiler*. *Library* yang digunakan terdiri dari dua *library* standar ANSI C (`stdlib.h` dan `string.h`), satu *library* dari aplikasi M-IDE (`at89x51.h`) dan dua *library* yang dikreasikan sendiri (`delay.h` dan `lcd_custom.h`). Deklarasi *library* yang digunakan pada program ditunjukkan pada gambar 6.

```
#include<stdlib.h>
#include<at89x51.h>
#include<delay.h>
#include<lcd_custom.h>
#include<string.h>
```

Gambar 6 Library yang digunakan

Algoritma program (gambar 4) dirancang berdasarkan mekanisme bahwa setiap terjadi *event* interupsi oleh gelombang masukan maka program akan menaikkan nilai pada variabel penghitung (*tick*), variabel ini akan direset nol di akhir setiap selesai mikrokontroler menampilkan informasi ke LCD. Kode program untuk *handling* event interupsi INT0 ditunjukkan pada gambar 7.

```
long int tick;
unsigned char counter1;
char string[9];

void INT0_isr(void) __interrupt(0)
{
  tick++;
}
```

Gambar 7. Kode program vektor interupsi oleh INT0

Sesuai dengan pembahasan pada poin 2.c, mempertimbangkan bahwa kemampuan maksimum *Timer1* hanya 65.5 milidetik, maka untuk menghasilkan basis waktu 1 detik direkayasa dengan cara memprogram *Timer1* agar menghasilkan interupsi setiap 10 milidetik dan menugaskan variabel *counter1* sebagai penghitung. Setiap terjadi interupsi oleh *Timer1* maka nilai *counter1* akan berkurang satu hingga nilainya nol. Nilai awal variabel *counter1* seharusnya adalah 100 agar tepat menghasilkan 1 detik (10 milidetik x 100), namun hasil pengujian ternyata menunjukkan bahwa walaupun secara pemrograman telah ditetapkan sebesar 10 milidetik ternyata interval interupsi oleh *Timer1* hanya sebesar 9.25 milidetik, sehingga nilai awal variabel *counter1* terhitung senilai 108 seperti pada persamaan 1-3.

$$t = \text{interval timer 1} \times k$$

$$1000 \text{ ms} = 9.25 \text{ ms} \times k$$

$$k = \frac{1000 \text{ ms}}{9.25 \text{ ms}} = 108,10$$

Kode program untuk vektor interupsi *Timer1* ditunjukkan pada gambar 8. Sesuai dengan *flowchart* pada gambar 4, tugas kode program pada vektor interupsi *Timer1* adalah sebagai pemeriksa unit basis waktu yang akan menandai bahwa setiap tercapai 1 detik maka informasi hasil perhitungan jumlah gelombang yang masuk harus ditampilkan ke LCD, dan jika belum tercapai 1 detik maka program hanya akan mengurangi 1 variabel *counter1*, mengisi ulang register *Timer1* dan mengaktifkan kembali *global interrupt*.

```
void TIMER1_isr(void) __interrupt(3)
{
    counter1--;
    if (counter1 == 0) // saat tercapai 1 detik
    {
        EA = 0; //disable EA sementara menampilkan ke LCD
        _ltoa(tick,string,10);
        LCD_GoToXY(1,8);
        LCD_DisplayString(string);LCD_DataWrite(' ');

        //siap untuk menghitung
        counter1 = 108; // 108 * 10 milidetik ~ 1 detik
        tick = 0;
    }
    TH1 = 0xdb;TL1 = 0xf7; // reload manual u/ 10 milidetik
    EA = 1; //enable EA setelah display
}
```

Gambar 8. Kode program vektor interupsi oleh timer 1

Kode program utama ditunjukkan pada gambar 9. Perintah pada kode program utama terdiri dari inisialisasi terhadap register interupsi, timer dan LCD. Setelah semua inisialisasi selesai dikerjakan maka program utama hanya akan menunggu tanpa melakukan apapun sampai terjadinya interupsi yang bersumber dari pin INT0 atau interupsi yang dihasilkan oleh *Timer1*.

```

void main()
{
  tick = 0;
  EA = 1; //global interrupt enable
  EX0 = 1; //external interrupt enable
  ITO = 1; //falling edge will trigger interrupt

  ET1 = 1; //enable timer1 overflow interrupt
  TR1 = 1; //timer 1 running. run and stop manual
  TMOD = 0x10;
  LCD_Init();
  LCD_Clear();
  LCD_DisplayString("  Journal Likmi");
  LCD_GoToXY(1,0);
  LCD_DisplayString("Freq. = ");
  LCD_DisplayString(string);
  while(1);
}

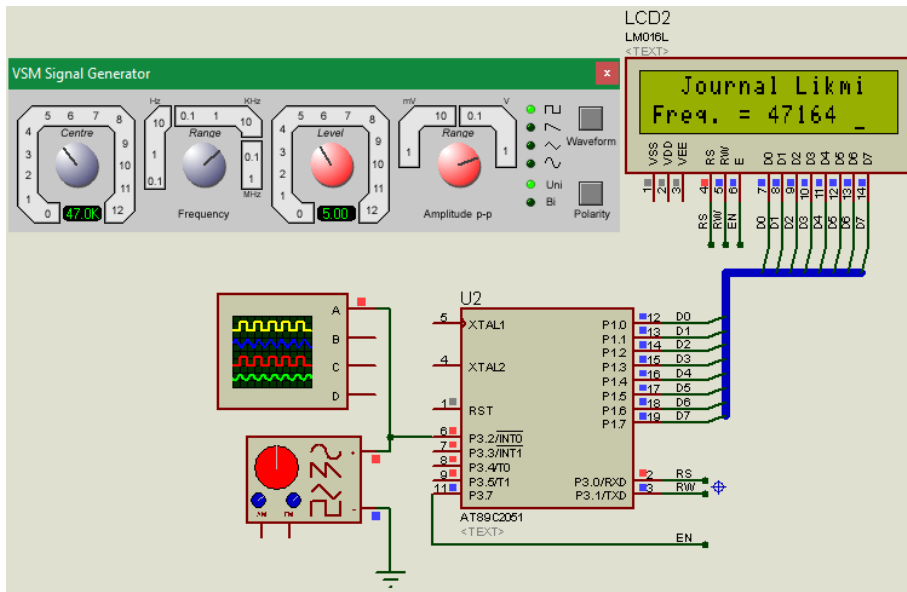
```

Gambar 9. Kode program utama

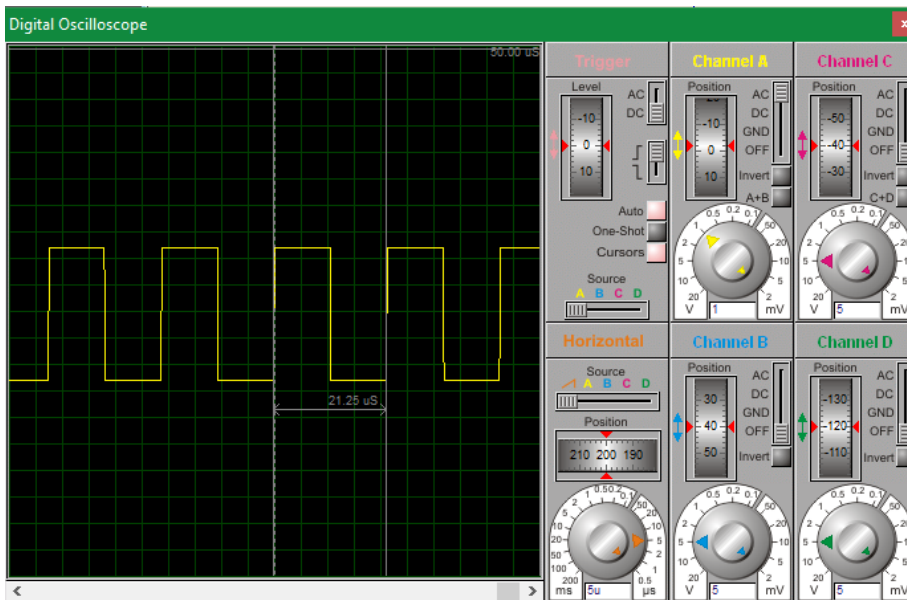
3.3 Hasil run program

Rancangan sistem *frequency counter* pada penelitian ini diuji secara simulasi menggunakan aplikasi Proteus 8. *Set up* pengujian terdiri dari satu unit *function generator* atau *signal generator*, satu *oscilloscope* dan skema rangkaian yang seperti ditunjukkan pada gambar 10. *Oscilloscope* digunakan sebagai pembanding untuk mengukur akurasi perhitungan yang dilakukan oleh mikrokontroler. Pengaturan *level* tegangan *signal generator* adalah 5 volt dengan bentuk gelombang kotak *unipolar* dengan frekuensi sekitar 47 KHz(pada panel *signal generator*). Penetapan frekuensi pada angka tersebut adalah berdasarkan pada rumusan untuk mengetahui capaian maksimum frekuensi tertinggi yang mampu diukur oleh mikrokontroler. Hasil pengukuran oleh mikrokontroler ternyata terbaca sebesar 47164 Hz, dimana informasi ini mampu menampilkan besaran secara lebih teliti dibandingkan tampilan pada panel *signal generator*. Pengukuran oleh *oscilloscope* ditunjukkan pada gambar 11, dimana periode gelombang terukur sebesar 21.25 μ s atau ekuivalen dengan frekuensi 47058,82 Hz. Dibandingkan dengan hasil pengukuran oleh *oscilloscope*, hasil ukur frekuensi oleh mikrokontroler mengandung nilai *error* sebesar :

$$\begin{aligned}
 error &= \left| \frac{f_{osc} - f_{micro}}{f_{osc}} \right| \times 100\% \\
 &= \left| \frac{47058,82 - 47164}{47058,82} \right| \times 100\% \\
 &= 0.00223 \times 100\% = 0.223\%
 \end{aligned}$$



Gambar 10. Hasil run program



Gambar 11. Hasil pengukuran periode oleh oscilloscope

3.4 Perhitungan frekuensi maksimum

Program yang dieksekusi oleh mikrokontroler pada dasarnya adalah berupa kode biner dimana setiap kombinasi biner memiliki arti secara *unique*. Kode program yang ditulis menggunakan bahasa *assembly* atau bahasa lainnya akan diubah menjadi kode biner dan terdokumentasi dalam file *list*(file berektensi *.lst*) sehingga dapat dilihat kebutuhan alokasi waktu untuk setiap baris perintahnya. Sistem *frequency counter* pada penelitian ini secara esensial adalah berupa fungsi penghitung dimana penghitungan hanya akan dilakukan saat terjadinya interupsi, hal ini berarti kemampuan maksimum mikrokontroler dalam menghitung frekuensi dapat diperkirakan dari *machine cycle* yang dapat dilihat dari file *list* seperti pada gambar 12. Perhitungan didasarkan pada premis bahwa sinyal gelombang yang masuk pin INT0 harus sesegera mungkin dihitung, oleh karena itu lama waktu yang diperlukan bagi mikrokontroler terhitung sejak mulai terjadinya interupsi hingga selesai melaksanakan perintah pada vektor interupsi harus diketahui untuk memperkirakan kemampuan maksimal mikrokontroler menghitung frekuensi gelombang masukan.

Dari gambar 12 didapatkan bahwa total *machine cycle* yang dibutuhkan oleh mikrokontroler sebanyak 23 *cycle*, dengan menggunakan frekuensi kristal 12 MHz maka durasi 1 *cycle* ekuivalen dengan 1 μ s, sehingga nilai 23 *cycle* ekuivalen dengan 23 μ s. Oleh karena itu, mikrokontroler hanya akan secara tepat dapat menghitung gelombang yang masuk jika periode gelombang maksimum lebih besar atau sama dengan 23 μ s atau ekuivalen dengan frekuensi :

$$f = \frac{1}{23\mu s} = 43478.26 \text{ Hz}$$

dari hasil simulasi diketahui bahwa ternyata mikroprosesor mampu menghitung hingga 47164 Hz, nilai ini sebenarnya didapatkan dengan kondisi bahwa simulasi berjalan lambat dan kemunculan informasi pada LCD perlu waktu lebih lama dari simulasi secara normal. Dengan demikian nilai yang dianggap sebagai besaran maksimum frekuensi dari daya ukur mikrokontroler adalah pada kisaran 43 KHz.

address	machine code	line	mnemonic		cycle
199	199	766	_INT0_isr:		
199	C0 E0	767	push	acc	2c
019B	C0 D0	768	push	psw	2c
019D	75 D0 00	769	mov	psw,#0x00	2c
		770 ;	D:\program_c\freqmeter.c:16: tick++;		
01A0	05*00	771	inc	_tick	1c
01A2	E4	772	clr	a	1c
01A3	B5*00 0C	773	cjne	a,_tick,00103\$	2c
01A6	05*01	774	inc	(_tick + 1)	1c
01A8	B5*01 07	775	cjne	a,(_tick + 1),00103\$	2c
01AB	05*02	776	inc	(_tick + 2)	1c
01AD	B5*02 02	777	cjne	a,(_tick + 2),00103\$	2c
01B0	05*03	778	inc	(_tick + 3)	1c
		779 00103\$:			
01B2	D0 D0	780	pop	psw	2c
01B4	D0 E0	781	pop	acc	2c
01B6	32	782	reti		2c
			total		23c

Gambar 12. Kode mesin dari baris perintah pada vektor interupsi INTO

4. KESIMPULAN

Desain *frequency counter* yang diusulkan pada penelitian ini terbukti secara simulasi telah berfungsi normal dengan kemampuan ukur hingga frekuensi 43 KHz dengan kisaran *error* sebesar 0.223%. Sistem dirancang menggunakan basis varian mikrokontroler yang *compact* dan minimum sehingga *resource* dari mikrokontroler dapat dimanfaatkan secara maksimum, efek positifnya adalah bahwa rasio antara ketergunaan fitur pada mikrokontroler dibandingkan dengan *cost* untuk pengadaan komponen menjadi lebih baik atau efisiensi dari sistem menjadi lebih tinggi.

DAFTAR PUSTAKA

- [1] Herman D. S, Elektronika : Teori dan Penerapan, Penerbit Cerdas Ulet Kreatif, 2007.
- [2] Irma Y. B, Komponen Elektronika, Sukabina Press, 2018.
- [3] MCS-51 Microcontroller Family User's Manual, Tersedia pada : <http://web.mit.edu/6.115/www/document/8051.pdf>.

- [4] B. A. Nugroho, Sudjadi, Y. Christyono, "Rancang Bangun Frekuensi Meter Listrik Berbasis ATMEGA328," *Jurnal Transient* Vol. 7, No. 4, Desember 2018, ISSN:2302-9927.
- [5] Suyamto, K. Medilla, M. Afiff J. Kusuma, "Rancang Bangun Alat Ukur Frekuensi Digital Berbasis Mikrokontroler AT89C51," *Prosiding Seminar Nasional II SDM Teknologi Nuklir*, Yogyakarta, 21-22 Desember 2006.
- [6] R. H. Tanjung, "Pembuatan Alat Frekuensi Counter pada Frekuensi Listrik Mikro Berbasis Mikrokontroler ATMEGA 328P," *Laporan Proyek Akhir, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara*, 2018.
- [7] S.E. Putri, Harmadi, "Rancang Bangun Sistem Pengukuran Frekuensi Getaran Akustik pada Speaker Piezoelektrik Menggunakan Sensor Serat Optik," *Jurnal Fisika Unand* Vol. 6, No. 1, Januari 2017. ISSN:2302-8491.
- [8] K. Rahmadnur, I. Yasri, "Perancangan Dan Analisa Kinerja Sistem Pendeteksi Frekuensi Berbasis Mikrokontroler Untuk Rentang Frekuensi 50mhz," *Jurnal Online Mahasiswa Fakultas Teknik* Vol. 8, Edisi 1 Januari s/d Juni 2021, ISSN:2355-6870.
- [9] S. A. Sofyan, M. Eriyadi, W. A. Sukarno, "Prototipe Frekuensi Meter Digital Dengan Rentang Ukur 1 Hz – 100 Khz Berbasis Mikrokontroler ," *ELEKTRA*, Vol. 4, No. 1, 25-34, Januari 2019.
- [10] Purwowibowo, J. A. Prakosa, "Prototipe Frekuensi Meter Rentang Ukur(10-2000) Hz Terkalibrasi ke Standar Primer Frekuensi," *INKOM*, Vol. 9, No. 1, pp:11-20, Mei 2015.
- [11] Syaiful R, Onny S, Wijono, M. Julius, "Perbandingan Desain dan Analisis IC Standard dan Datasheet TTL AOI Dual 2 Wide 2 Input," *Jurnal EECCIS* Vol. 6, No. 2, Desember 2012.