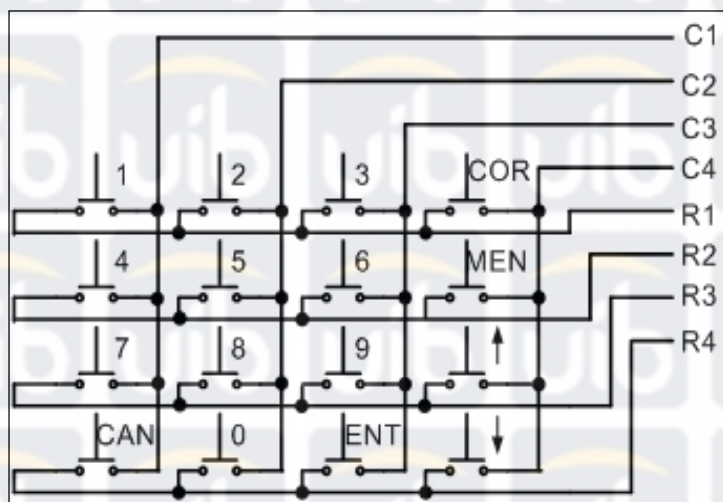


BAB II LITERATURE REVIEW

2.1 Keypad

Keypad biasanya digunakan untuk memberikan *input* pada beberapa peralatan yang berbasis mikrokontroller dan mikroprosesor. Untuk mengurangi penggunaan pin *input* pada mikrokontroller maka dapat menggunakan *keypad matrix*. *Keypad matrix* terdiri dari tombol – tombol yang disusun secara *matrix* (baris x kolom).

Pada *keypad matrix* 4x4, penggunaan pin pada mikrokontroller cukup menggunakan 8 pin untuk 16 tombol. Hal ini dimungkinkan karena rangkaian tombol disusun secara horisontal membentuk baris dan secara vertikal membentuk kolom seperti pada Gambar 2.1.



Gambar 2.1 Rangkaian Dasar Keypad 4x4

Sumber : www.julkahendri.co.cc/2010/08/keypad-4x4-mikrokontroller.html

Proses pengecekan tombol yang sedang ditekan pada mikrokontroller menggunakan teknik *scanning*, yaitu proses pengecekan yang dilakukan dengan

cara memberikan umpan data pada bagian baris dan kemudian mengecek *feedback* pada bagian kolom. Mikrokontroler memberikan logika *low* “0” satu per satu pada bagian baris secara berurutan, dan selanjutnya membaca 4 bit pada kolom. Apabila salah satu tombol keypad ada yang ditekan, maka pada bagian kolom akan terbaca dengan logika *low* “0”, dan saat tidak ada tombol yang ditekan maka mikrokontroler akan membaca kolom dengan logika *high* “1”.

2.2 ATMEGA16

Mikrokontroler AVR memiliki arsitektur RISC 8 Bit, sehingga semua instruksi dikemas dalam kode 16-bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam satu siklus instruksi *clock*. Dan ini sangat membedakan sekali dengan instruksi MCS-51 (Berarsitektur CISC) yang membutuhkan siklus 12 *clock*. RISC adalah *Reduced Instruction Set Computing* sedangkan CISC adalah *Complex Instruction Set Computing*.

AVR dikelompokkan kedalam 4 (empat) kelas, yaitu ATtiny, keluarga AT90Sxx, keluarga ATmega, dan keluarga AT86RFxx. Dari kesemua kelas yang membedakan satu sama lain adalah ukuran *onboard memori*, *on-board peripheral* dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan mereka bisa dikatakan hampir sama.

Arsitektur dari ATmega16 :

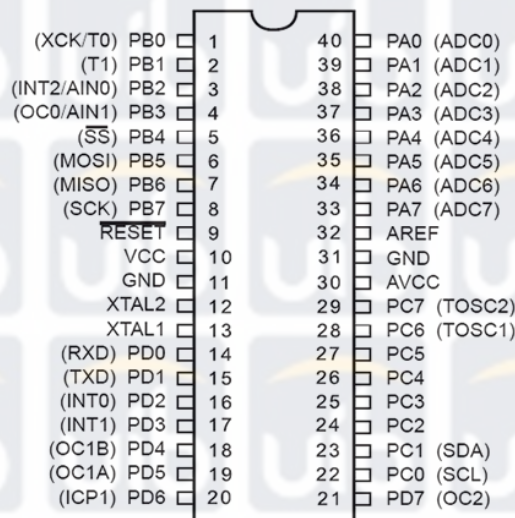
- Saluran IO sebanyak 32 (tiga puluh dua) buah, yaitu Port A, Port B, Port C dan Port D.
- ADC 10 bit sebanyak 8 (delapan) *channel*.
- 3 (tiga) buah *timer / counter*.

- 32 (tiga puluh dua) register.
- *Watchdog Timer* dengan oscillator internal.
- SRAM sebanyak 1 kb.
- *Memori flash* sebesar 16 kb.
- Sumber *Interrupt internal* dan *external*.
- Port SPI (*Serial Peripheral Interface*).
- EEPROM *on board* sebanyak 512 byte.
- Komparator analog.
- Port USART (*Universal Synchronous Asynchronous Receiver Transmitter*)

Fitur dari ATmega16

- Sistem processor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
- Ukuran *memory flash* 16kb, *SRAM* sebesar 1kb, *EEPROM* sebesar 512 byte.
- *ADC internal* dengan resolusi 10 bit sebanyak 8 *channel*
- Port komunikasi serial USART dengan kecepatan maksimal 2.5 Mbps
- *Mode Sleep* untuk penghematan penggunaan daya listrik.

Konfigurasi dari ATmega16 dapat dilihat pada Gambar 2.2, dimana setiap port pada ATmega16 memiliki fungsi yang berbeda - beda. Fungsi khusus yang terdapat pada ATmega16 yaitu *ADC*, *reset*, *clock external*, *timer / counter*, *komparator analog*, *SPI*, *timer oscillator*, *interrupt external* dan komunikasi serial.



Gambar 2.2 Konfigurasi Pin ATmega16

Sumber : Datasheet ATMega16

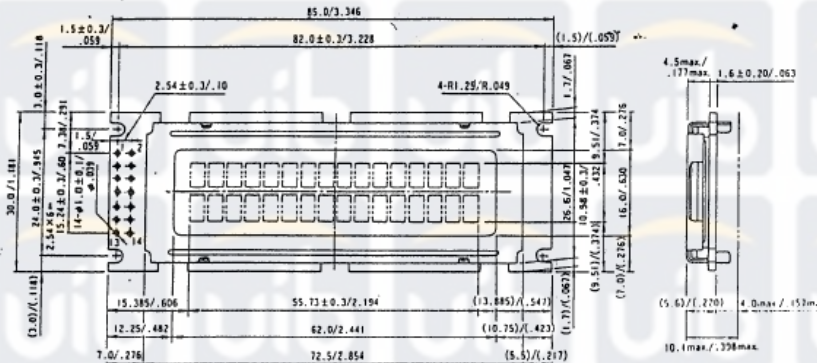
Konfigurasi Pin ATmega16

- VCC merupakan Pin yang berfungsi sebagai pin masukan catudaya.
- GND merupakan Pin *Ground*.
- Port A (PA0...PA7) merupakan pin I/O dan pin masukan ADC.
- Port B (PB0...PB7) merupakan pin I/O dan pin yang mempunyai fungsi khusus yaitu *Timer / Counter*, komparator *analog* dan SPI.
- Port C (PC0...PC7) merupakan port I/O dan pin yang mempunyai fungsi khusus, yaitu komparator *analog* dan *timer oscillator*.
- Port D (PD0...PD7) merupakan port I/O dan pin fungsi khusus yaitu komparator *analog* dan *interrupt external* serta komunikasi serial.
- RESET merupakan pin yang digunakan untuk mereset mikrokontroler.
- XTAL1 dan XTAL2 merupakan pin masukan *clock eksternal*.
- AVCC merupakan pin masukan untuk tegangan ADC.
- AREF merupakan pin masukan tegangan referensi untuk ADC

2.3 LCD (Liquid Crystal Display)

LCD merupakan komponen yang biasanya digunakan untuk menampilkan suatu simbol, angka dan huruf. Pada LCD 16x2 (16 karakter dengan 2 baris), mempunyai 32 karakter, dapat dilihat pada Gambar 2.3.

Semua fungsi *display* dikendalikan oleh intruksi dan *module* dengan mudah dapat dihubungkan dengan mikrokontroler. Hal ini membuat *module* berguna untuk berbagai tujuan termasuk *terminal display units*.



Gambar 2.3 LCD 16x2
Sumber : Datasheet LCD M1632

2.4 Bluetooth

Bluetooth adalah teknologi radio frekuensi yang murah, daya rendah dan jarak pendek. *Bluetooth* adalah chip yang bekerja menggunakan RF (radio frekuensi) *module* yang tertanam dalam perangkat elektronik. Radio ini beroperasi pada frekuensi yang tidak berlisensi 2,4 GHz. Dalam *standart mode bluetooth* mengirimkan sinyal hanya 1 (satu) miliwatt. Hal ini memungkinkan jarak sekitar 10 (sepuluh) meter (Jihad Qaddour & Matthew H.S. Kuofie, 2004).

Bluetooth mampu membuat *personal area network* (PAN) yang biasanya disebut dengan *piconet*. *Piconet* terdiri atas 2 (dua) sampai 8 (delapan), dengan 1 *master* dan sisanya sebagai *slave*.

Ketika salah satu perangkat berkemampuan *bluetooth* datang dalam jangkauan lain, mereka akan membuka komunikasi untuk menentukan apakah koneksi yang diperlukan. Penentuan ini dibuat berdasarkan profil masing – masing perangkat (Jihad Qaddour & Matthew H.S. Kuofie, 2004).

Bluetooth menggunakan salah satu dari dua jenis frekuensi Spread Spectrum Radio yang digunakan untuk kebutuhan *wireless*. Jenis frekuensi yang digunakan adalah *Frequency Hopping Spread Spectrum* (FHSS), sedangkan yang satu lagi yaitu *Direct Sequence Spread Spectrum* (DSSS) digunakan oleh IEEE802.11xxx.

2.4.1 Teknologi spread spectrum

Teknologi *spread spectrum* memungkinkan untuk membawa sejumlah informasi yang sama seperti yang dapat dikirimkan dengan menggunakan *narrowband carrier signal* dan menyebarkan sinyal itu pada kisaran frekuensi yang jauh lebih besar.

Menurut Abas Ali Pangera (2007), dengan menggunakan spektrum frekuensi yang lebih lebar, dapat memperkecil kemungkinan bahwa data akan mengalami perubahan / pengurangan (*corrupted*) atau *jamming*.

Agar suatu sinyal dapat dikelompokkan sebagai *spread spectrum*, sinyal itu harus menggunakan daya yang rendah. 2 (dua) karakteristik *spread spectrum*

penggunaan *bandwidth* frekuensi dan daya yang sangat rendah membuat sinyal seolah – olah merupakan sinyal *noise* bagi sebagian besar penerima.

2.4.1.1 *Frequency Hopping Spread Spectrum*

Pada *Frequency Hopping Spread Spectrum (FHSS)*, setiap pengguna mulai mengirimkan datanya melalui *narrow band* selama slot waktu (*dwell time*), dan kemudian melompat untuk band lainnya di slot waktu berikutnya sesuai *pseudorandom* (urutan) yang ditetapkan pengguna (Salam A.Zummo, 2008).

Urutan *pseudorandom* merupakan suatu daftar beberapa frekuensi ke arah mana *carrier* akan melompat pada suatu interval waktu yang ditetapkan sebelum terjadi pengulangan pola ini.

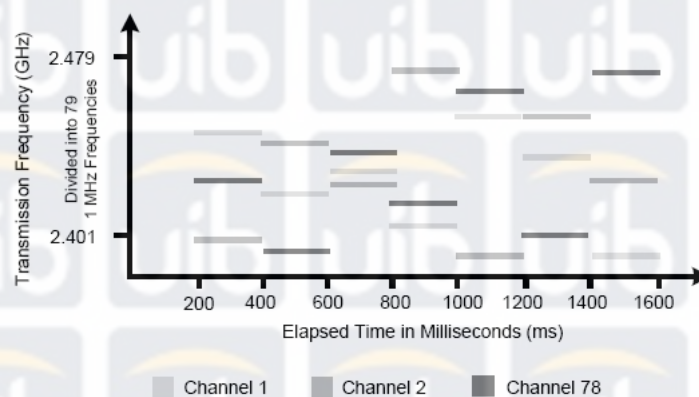
Menurut Adeel Ahmad., dkk (2012) sebuah transmisi *spread spectrum* menawarkan tiga keuntungan utama selama transmisi frekuensi tetap :

1. *Spread Spectrum* sinyal sangat tahan terhadap interferensi *narrow band*.
2. *Spread Spectrum* sinyal sulit untuk mencegat. Sebuah sinyal FHSS hanya muncul sebagai peningkatan *noise* ke penerima *narrow band*. *eavesdropper* hanya mampu mencegat transmisi jika urutan *pseudorandom* dikenal.
3. *Spread Spectrum* transmisi dapat berbagi pita frekuensi dengan banyak jenis transmisi konvensional dengan gangguan minimal. Penyebaran *spectrum* sinyal menambahkan sedikit *noise* ke *narrow*

frequency communications dan sebaliknya. Akibatnya, *bandwidth* dapat digunakan lebih efisien.

2.4.1.2 Saluran

Suatu *frequency hopping system* akan bekerja menggunakan suatu pola lompatan khusus yang disebut saluran (*channel*). *Frequency hopping system* secara tipikal menggunakan 26 pola lompatan standar dari FCC. Beberapa *frequency hopping system* memungkinkan penciptaan suatu pola lompatan yang disesuaikan dengan kebutuhan (*custom hop patterns*).



Gambar 2.4 *Frequency hopping system* yang menempati lokasi secara bersamaan

Sumber : Abas Ali Pangera (2007)

Sekalipun dimungkinkan untuk memiliki hingga sebanyak 79 (tujuh puluh Sembilan) kanal yang menempati suatu lokasi secara bersamaan, namun dengan banyak pola sistem ini, masing – masing *frequency hopping radio* akan memerlukan sinkronisasi yang presisi dengan semua radio lainnya agar tidak saling mengganggu atau memancarkan pada frekuensi yang sama seperti *frequency hopping radio* lainnya dikawasan itu.

2.4.1.3 Dwell Time

Pada saat membahas *frequency hopping system*, kita akan membahas sistem yang harus memancarkan pada suatu frekuensi yang telah ditetapkan untuk jangka waktu tertentu, dan kemudian melompat ke frekuensi yang berbeda untuk meneruskan transmisi. Pada saat *frequency hopping system* memancar pada suatu frekuensi, maka proses pemancaran ini harus berlangsung selama jangka waktu tertentu. Jangka waktu ini disebut *dwell time*. Setelah *dwell time* habis, maka sistem ini akan beralih ke suatu frekuensi berbeda dan mulai memancarkan lagi.

2.4.1.4 Hop Time

Menurut Abas Ali Pangera (2007) pada saat mempertimbangkan aksi lompatan frekuensi dari suatu *frequency hopping radio*, *dwell time* hanya merupakan salah satu pertimbangan. Pada saat suatu *frequency hopping radio* melompat dari frekuensi A ke frekuensi B, maka ia harus mengubah frekuensi pancar. Radio tersebut harus beralih ke suatu rangkaian yang ada untuk menyelaraskan dengan frekuensi baru tersebut. Pada tiap cara, proses peralihan ke frekuensi baru harus tuntas sebelum transmisi dapat dijalankan kembali, dan perubahan ini membutuhkan waktu karena adanya latensi listrik yang inheren dalam sistem rangkaian. Terdapat sedikit waktu selama perubahan frekuensi ini dimana radio tersebut tidak memancar yang disebut dengan *hop time*. *Hop time* diukur dalam mikrodetik (us) dan dengan *dwell time* yang relatif panjang yaitu sekitar 100 – 200 ms, *hop time* menjadi tidak signifikan.

Dengan *dwell time* yang sangat singkat 500 – 600 us seperti yang digunakan beberapa *frequency hopping system* seperti *bluetooth*, *hop time* menjadi sangat signifikan. Jika kita memperhatikan efek *hop time* sehubungan dengan *throughput* data, kita menemukan bahwa semakin lama *hop time* jika dibandingkan dengan *dwell time*, maka semakin lambat laju transmisi data dalam bit.

2.5 *Easy Bluetooth Module*

Easy bluetooth module adalah *bluetooth module* yang dapat dipasang langsung ke mikrokontroler karena sudah *compatible* dengan CMOS dan TTL. Pada *bluetooth module* ini termasuk pada daya kelas 2 yang beroperasi antara 2,5 mW (4dBm) dan 0,25mW (-6dBm) dimana jarak area kerja sampai dengan 30 meter. Tegangan pada *module* ini berkisar antara 3,3 Vdc sampai 5,5 Vdc.

Untuk penggunaan *easy bluetooth module* terlebih dahulu dipasang ke mikrokontroler pin 1 *bluetooth module* pada ground, pin 2 pada *transmitter* mikrokontroler, pin 3 pada *receiver* mikrokontroler dan pin 10 pada tegangan, lalu pada komputer yang sudah dipasang dengan *bluetooth dongle* memulai proses *pairing* dengan kode 0000. Kemudian dapat dilihat nilai COM port yang digunakan untuk komunikasi serial antara *easy bluetooth module* dengan komputer.

2.6 *CodeVision AVR*

CodeVision AVR pada dasarnya merupakan perangkat lunak pemrograman mikrokontroler keluarga AVR berbasis bahasa C. Ada tiga

komponen penting yang telah diintegrasikan dalam perangkat lunak ini: *Compiler C*, IDE dan *Program generator*.

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, *Compiler C* yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan *library* fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, *compiler C* untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*).

Khusus untuk *library* fungsi, disamping *library* standar seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya.



Gambar 2.5 Tampilan Perangkat Lunak *Codevision AVR*
Sumber : Diolah dari data primer (2012)

CodeVisionAVR juga dilengkapi IDE yang sangat *user friendly*. Selain menu - menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis *Windows*, *CodeVision AVR* ini telah mengintegrasikan perangkat lunak *downloader (in system programmer)* yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam sistem memori mikrokontroler AVR yang

sedang diprogram. Tampilan software CodeVision AVR dapat dilihat pada Gambar 2.5.

Selain itu, *CodeVision AVR* juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau *CodeWizard AVR*. Secara praktis, *tool* ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi *programmer* dalam penginisialisasian register - register yang terdapat pada mikrokontroler AVR yang sedang diprogram. Dinamakan *Code Generator*, karena perangkat lunak *CodeVision* ini akan membangkitkan kode - kode program secara otomatis setelah fase inisialisasi pada jendela *CodeWizard AVR* selesai dilakukan.

2.7 Kecepatan Pelayanan

Berdasarkan hasil dari wawancara pada tanggal 22 Maret 2012 dengan bapak Devid Trinaldo Simatupang. STT.Par dosen Universitas Internasional Batam Prodi Manajemen Perhotelan yang menyatakan bahwa standar kecepatan layanan mencatat menu makanan dan minuman oleh pelayan di restoran maksimal selama 1 (satu) menit untuk 1 (satu) orang.

2.8 Tipe – Tipe Dasar Pelayanan Makanan di Restoran

Menurut Marsum (2005), tipe – tipe dasar pelayanan makanan di restoran pada umumnya dapat dipakai untuk membedakan kategori suatu restoran.

Ada 4 (empat) macam tipe dasar pelayanan yang terkenal, yaitu:

1. *Table service*

Yang dimaksud dengan *table service* adalah suatu sistem pelayanan restoran dimana para tamu duduk dikursi menghadap meja makan, dan

kemudian makanan maupun minuman diantarkan, disajikan kepada para tamu. Dalam hal ini yang menyajikan makanan dan minuman bisa *waiter* maupun *waitress*.

2. *Counter service*

Yang dimaksud dengan *counter service* adalah suatu sistem pelayanan restoran dimana para tamu yang datang terus duduk di *counter*. Apabila makanan dan minuman yang dipesannya sudah siap maka akan disajikan kepada para tamu tadi diatas *counter*. *counter* adalah meja panjang yang membatasi ruangan dapur dengan ruangan restoran.

3. *Self service*

Yang dimaksud dengan *self service* atau kadang disebut juga dengan *buffet service* adalah suatu sistem pelayanan restoran dimana semua makanan secara lengkap telah ditata dan diatur dengan rapi diatas meja hidangan atau meja prasmanan. Para tamu secara bebas mengambil sendiri hidangannya sesuai dengan selera maupun kesukaannya.

4. *Carry out service*

Carry out service kadang – kadang disebut juga sebagai *take out service* yaitu sistem pelayanan restoran dimana tamu datang untuk membeli makanan yang telah disiapkan terlebih dahulu, dibungkus dalam boks (kotak) untuk dibawa pergi.

Berdasarkan 4 tipe dasar pelayanan makanan di restoran perangkat pencatat menu makanan digolongkan pada tipe *table service*, dimana pelayan mencatat data pesanan secara manual menggunakan perangkat.