

BAB II

KAJIAN PUSTAKA

Berisi tentang penelitian terdahulu dan landasan teori sangat membantu untuk dapat memahami suatu sistem. Selain dari pada itu dapat juga dijadikan sebagai bahan acuan didalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka penelitian terdahulu dan landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi *Microcontroler Arduino Uno R3*, *Sensor NTU*, *Sensor Suhu*, *Ethernet Shield*, *Power Supply*, *LCD*, dan program *software* pendukung.

2.1 Penelitian Terdahulu

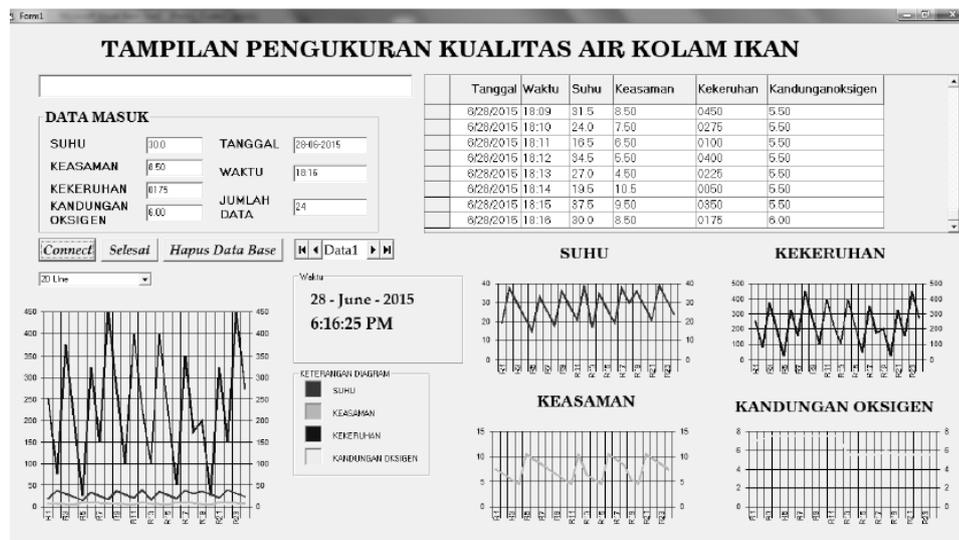
Penelitian terdahulu diperlukan dalam sebuah penelitian agar perancangan yang sedang di lakukan bisa menyempurnakan lagi perangkat yang telah dikembangkan oleh penelitian sebelumnya. Hal ini menjadi tolak ukur apakah penelitian tersebut sesuai dengan perkembangan teknologi terbaru.

2.1.1 *Automatic Control of Water Quality Based Wireless Fish Pond With RFM12-433S (2016)*

Penelitian terdahulu pertama pada jurnal dengan judul “ *Automatic Control of Water Quality Based Wireless Fish Pond With RFM12-433S* ” oleh (Pius Yozy Merucahyo, Andreas Bagus Sadewo, Christin Karuru, Martanto, A. Tri Priantoro, 2016). Penulis menggunakan ATmega128 sebagai pusat kendalinya, dan menggunakan RFM12-433S sebagai *transmitter*. Sistem akan mengambil data dari setiap sensor kemudian mengatur pintu air masuk kolam, pintu air keluar kolam dan pompa sumur untuk mendapatkan kualitas air yang dibutuhkan, kemudian data tersebut dapat dikirimkan ke pengguna untuk memonitoring

kualitas air kolam melalui PC atau Laptop. Sumber daya yang digunakan untuk *supply* pusat kendali menggunakan *accu*.

Tujuan penulis membuat sistem ini supaya dapat mengirimkan data sensor kualitas air kolam ikan secara otomatis dari *remote unit* ke *central unit* dengan menggunakan *wireless* dan dapat mengendalikan pintu air masuk ke kolam, pintu air keluar kolam dan pompa air secara otomatis.

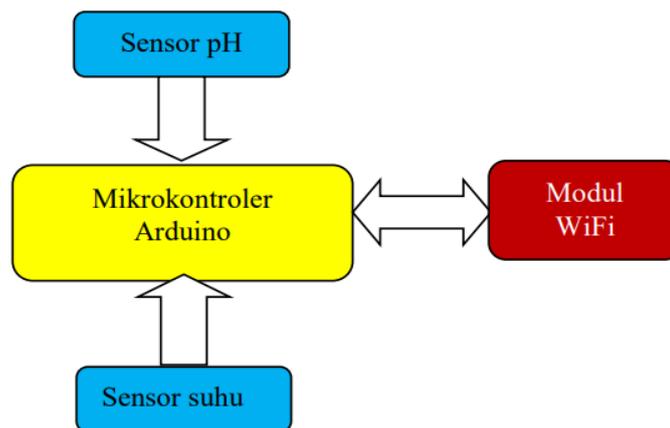


Gambar 2.1 Tampilan sistem (Andreas, dkk, 2016)

Penulis mengambil Kesimpulan bahwa pada sistem pengendali otomatis kualitas kolam air ikan dengan RFM12-433S adalah sistem yang dibuat sudah dapat bekerja sesuai dengan perancangan dan mampu menampung paket data yang belum terkirim maksimal 80 paket data. Sistem juga mampu membaca sensor yang digunakan untuk dibandingkan dengan standar kualitas air kolam yang ditentukan kemudian memberikan perintah ke sistem kendali untuk mengendalikan kualitas air kolam. Penggunaan frekuensi untuk pengiriman paket data tidak mempengaruhi jarak jangkauan pengiriman paket data.

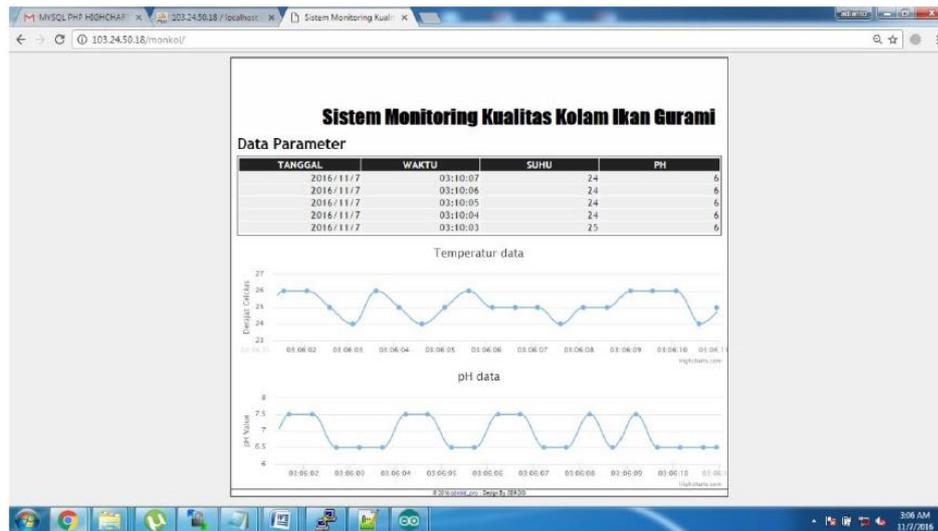
2.1.2 *Wireless Sensor Network* Berbasis Protokol UDP untuk Monitoring pH dan Suhu Kolam Ikan Air Tawar (2016)

Penelitian terdahulu kedua terkait monitoring air kolam pernah dilakukan oleh (Binti Mualifatul dan Yuning Widiarti, 2016) dalam jurnal dengan judul “ *Wireless Sensor Network* Berbasis Protokol UDP untuk Monitoring pH dan Suhu Kolam Ikan Air Tawar “. Penulis disini menggunakan Protokol UDP sebagai *transmitter* untuk media pengiriman data. Dimana sistem yang dibangun penulis ini terdiri dari *microcontroler* Arduino, satu set sensor parameter suhu dan pH, *smartphone android*, dan perangkat jaringan. Parameter yang digunakan penulis terdapat dua parameter untuk memonitor kualitas air kolam yaitu pH dan suhu.



Gambar 2.2 *Diagram Blok* (Mualifatul, dkk, 2016)

Diini dapat kita lihat cara kerjanya dimulai dari sensor suhu dan sensor pH yang digunakan untuk mengukur suhu dan pH kemudian hasilnya dibaca oleh *microcontroler* arduino, selanjutnya arduino mengirim data suhu dan pH melalui sarana komunikasi data dengan perangkat jaringan. Selanjutnya, data akan ditampilkan secara *real time* dengan tampilan web.



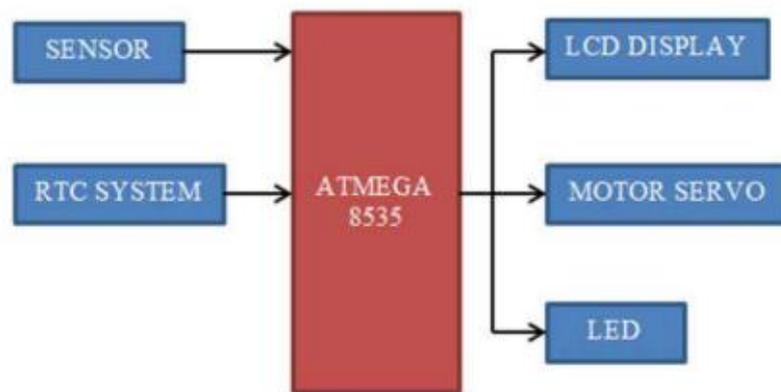
Gambar 2.3 Hasil monitoring (Mualifatul, dkk, 2016)

Penulis menyimpulkan pembuatan WSN untuk sistem monitoring pH dan suhu kolam ikan air tawar dengan menggunakan teknologi *open source* Arduino telah terbukti berhasil mendapatkan pengukuran yang akurat dan *real time*. Proses penyimpanan data dari *smartphone* android ke data *center* efektif dilakukan dengan menggunakan program *database server* MySQL. Pemilihan jenis komunikasi data harus memperhatikan kecepatan dan keakurasian data yang diperoleh, dimana pada penelitian ini peneliti memilih menggunakan *socket* dengan protokol UDP sebagai media pengiriman data dan menggunakan protokol http yang implementasinya dengan menggunakan PHP, JQuery dan AJAX dengan tujuan dapat memonitor dengan cepat data yang telah diperoleh dalam *database* MySQL tanpa *loading* data yang lama.

2.1.3 Sistem Pemantau Kekeruhan Air dan Pemberi Makan Otomatis pada Ikan Berbasis Mikrokontroler (2016)

Penelitian terdahulu ketiga terkait dengan perangkat yang akan dirancang pernah dilakukan oleh (Riskan Oktafiadi, 2016) dalam jurnal yang berjudul “ Sistem Pemantau Kekeruhan Air dan Pemberi Makan Otomatis pada Ikan

Berbasis Mikrokontroler “. Dalam jurnal ini penulis menggunakan *Microcontroler* ATmega8535. Penulis menggunakan sensor GE *turbidity* untuk mengukur tingkat kekeruhan air kolam ikan dan motor servo yang berfungsi membuka plat wadah pakan ikan sehingga pakan yang berada dalam wadah dapat ditumpahakan. Berikut contoh *blok diagram* yang digunakan penulis saat merancang Sistem Pemantau Kekeruhan Air dan Pemberi Makan Otomatis pada Ikan Berbasis *Microcontroler*.



Gambar 2.4 Diagram Blok (Riskan, 2016)

Disini kita bisa lihat penulis menggunakan *microcontroler* ATmega8535 merupakan salah satu *microcontroler* 8 bit buatan Atmel untuk keluarga AVR yang diproduksi secara massal pada tahun 2006. Karena merupakan keluarga AVR, maka ATmega8535 juga menggunakan arsitektur RISC. Penulis juga menggunakan sensor GE *turbidity* untuk mengukur tingkat kekeruhan air dan sensor RTC sistem memiliki fungsi menyimpan tanggal, waktu/jam, menit, dan detik sebagai *input data* pewaktu *microcontroler* bekerja.



Gambar 2.5 pengujian Sensor (Riskan, 2016)

2.2 Sistem

Pengertian sistem menurut Poerwadarminta (2006) yaitu sekelompok bagian bagian (alat dan sebagainya), yang bekerja bersama-sama untuk melakukan sesuatu maksud. Apabila salah satu bagian saja rusak atau tidak dapat menjalankan tugasnya maka maksud yang hendak dicapai tidak dapat menjalankan tugasnya maka maksud yang hendak dicapai tidak akan terpenuhi atau setidaknya sistem yang sudah terwujud akan mendapatkan gangguan.

Menurut Henry Prat Fairchild dan Eric Kohler, pengertian Sistem adalah sebuah rangkaian yang saling kait mengkait antar beberapa bagian sampai kepada bagian yang paling kecil, bila suatu bagian atau sub bagian terganggu maka bagian yang lain juga ikut merasakan ketergangguan tersebut. Maka dari itu dapat disimpulkan, Pengertian Sistem adalah kesatuan yang utuh dari sesuatu rangkaian, yang saling kait mengkait satu sama lain, bagian (anak cabang) dari suatu sistem, menjadi induk dari rangkaian-rangkaian selanjutnya. Begitu seterusnya sampai pada bagian terkecil, rusaknya salah satu bagian akan mengganggu kestabilan sistem itu sendiri secara keseluruhan. Pemerintah Indonesia ialah suatu contoh dari sistem, dan anak cabangnya adalah sistem pemerintahan daerah, yang kemudian seterusnya sistem pemerintahan desa dan kelurahan (Azhari, 2006).

2.3 Monitoring

Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan, yang fokus pada proses dan keluaran. Monitoring menyediakan data dasar untuk menjawab permasalahan, sedangkan evaluasi adalah memposisikan data-data tersebut agar dapat digunakan dan diharapkan memberikan nilai tambah. Evaluasi adalah mempelajari kejadian, memberikan solusi untuk suatu masalah, rekomendasi yang harus dibuat, menyarankan perbaikan. Namun tanpa monitoring, evaluasi tidak dapat dilakukan karena tidak memiliki data dasar untuk dilakukan analisis, dan dikhawatirkan akan mengakibatkan spekulasi, oleh karena itu monitoring dan evaluasi harus berjalan seiring (Wirid Winduro, 2016).

Dalam sebuah program tidak bisa hanya melakukan evaluasi saja, atau hanya melakukan monitoring, seperti contohnya pada sebuah program monitoring, tidak boleh dirancang tanpa diketahui bagaimana data dan informasi akan dievaluasi dan tepat guna, sebab ketidak mampuan dalam mengumpulkan dan menyimpan data yang akan digunakan. Monitoring adalah kegiatan yang berkesinambungan.

2.4 Air kolam ikan

Untuk mendapatkan air kolam ikan yang sehat (BBPBAT, 2016), Anda harus menjaga kondisi airnya dalam keadaan selalu baik dan sehat. Kondisi air kolam sehat akan membuat ikan tidak mudah sakit. Setidaknya ada beberapa parameter yang perlu diperhatikan agar air kolam ikan selalu dalam kondisi sehat.

Parameter tersebut meliputi :

1. Kualitas Air

Kualitas air atau tingkat kejernihan adalah daya tembus cahaya matahari ke dalam perairan. Kejernihan air dipengaruhi oleh kepadatan *plankton* dan kekeruhan yang disebabkan oleh partikel tanah terlarut. Kekeruhan air mempengaruhi kemampuan air untuk meneruskan cahaya ke dalam air. Kekeruhan pada air kolam disebabkan oleh koloid partikel-partikel lumpur dan bahan organik terlarut. Air dengan tingkat kekeruhan tertentu malah berdampak baik bagi pertumbuhan ikan karena dan sebaliknya jika tingkat kekeruhan itu melebihi batas dapat mengurangi intensitas sinar yang masuk ke dalam air. Batas maksimal kualitas air atau tingkat kekeruhan yang aman bagi ikan adalah 128 NTU (BBPBAT, 2016).

2. Suhu Air

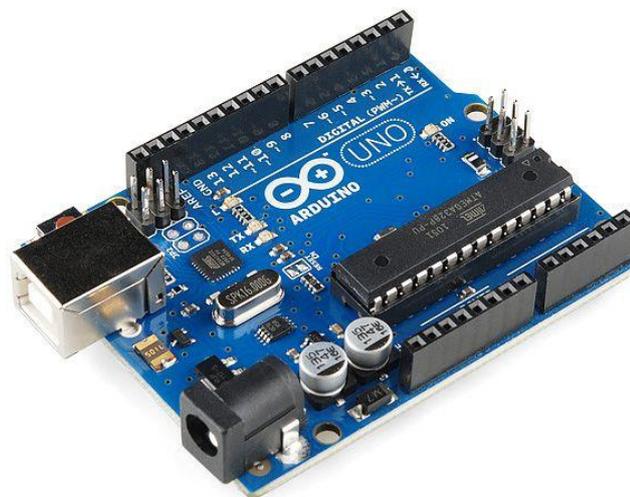
Suhu air bisa mempengaruhi pertumbuhan vegetasi air dan kadar oksigen di dalam kolam. Peningkatan suhu air akan menyebabkan oksigen berkurang. Selain itu, tanaman dan ikan akan membutuhkan oksigen lebih banyak karena tingkat respirasinya meningkat. Perubahan suhu air bisa mengakibatkan perubahan kebiasaan ikan. Semakin dingin, maka nafsu makan dan pertumbuhannya justru melambat. Suhu air yang baik untuk pertumbuhan ikan diantara kisaran 25°C–35°C (BBPBAT, 2016).

Tabel 2.1 Parameter Suhu Kolam Ikan (BBPBAT, 2016)

TABEL SUHU AIRIKAN AIR TAWAR		
NO	JENIS IKAN	TEMPERATUR
1	TAWES	20 – 33°C
2	NILEM	18 – 28°C
3	MAS/TOMBRO	20 – 25°C
4	PATIN	28 – 32°C
5	BAWAL	25 – 30°C
6	GURAME	24 – 28°C
7	NILA	25 – 30°C
8	SIDAT	28 – 29°C
9	LELE	25 – 30°C
10	GABUS	25 – 30°C

2.5 Microcontroler Arduino Uno R3

Arduino Uno R3 adalah papan pengembangan (*development board*) *microcontroler* yang berbasis *chip* ATmega328P. Disebut sebagai papan pengembangan karena *board* ini memang berfungsi sebagai arena *prototyping sirkuit microcontroller*. Dengan menggunakan papan pengembangan, akan lebih mudah merangkai rangkaian elektronika *microcontroller* dibanding jika memulai merakit ATmega328P dari awal di *breadboard* (Djuandi, Feri. 2015).

**Gambar 2.6** Microcontroler Arduino Uno R3 (Arief, 2014)

Kata " Uno " berasal dari bahasa Italia yang berarti "satu", dan dipilih untuk menandai peluncuran *Software* Arduino (IDE) versi 1.0. Arduino. Sejak awal peluncuran hingga sekarang, Uno telah berkembang menjadi versi Revisi 3 atau biasa ditulis REV 3 atau R3. *Software* Arduino IDE, yang bisa diinstall di *Windows* maupun Mac dan Linux, berfungsi sebagai *software* yang membantu anda memasukkan (*upload*) program ke *chip* ATmega328P dengan mudah. Ringkasan spesifikasinya pada tabel 2.2.

Tabel 2.2 Spesifikasi Arduino Uno (Ilerning Media, 2015)

<i>Microcontroler</i>	ATmega328P
Operasi tegangan	5 Volt
<i>Input</i> tegangan	Disarankan 7 - 11 Volt
<i>Input</i> tegangan batas	6 - 20 Volt
Pin I/O <i>digital</i>	14 buah (6 bisa untuk PWM)
Pin <i>analog</i>	6 buah
Arus DC tiap pin I/O	20Ma
Arus DC ketika 3.3v	50mA
<i>Memory flash</i>	32 Kb (Atmega328P) dan 0.5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Kecepatan <i>clock</i>	16 MHz
Dimensi	68.6 mm x 53.4 mm
Berat	25 g

2.5.1 Pemrograman

Pemrograman *board* Arduino dilakukan dengan menggunakan Arduino *Software* (IDE). *Chip* ATmega328P yang terdapat pada Arduino Uno R3 telah diisi program awal yang sering disebut *bootloader*. *Bootloader* tersebut yang bertugas untuk memudahkan melakukan pemrograman lebih sederhana menggunakan Arduino *Software*, tanpa harus menggunakan tambahan *hardware* lain. Cukup menghubungkan Arduino dengan kabel USB ke PC atau Mac/Linux, jalankan *software* Arduino *Software* (IDE), dan sudah bisa mulai memrogram *chip* ATmega328P.



Gambar 2.7 *Microcontroller* Arduino Uno (Arief, 2014)

Arduino Uno memiliki 14 digital pin *input / output* (atau biasa ditulis I/O, dimana 6 pin diantaranya dapat digunakan sebagai *output* PWM), 6 pin *input analog*, menggunakan *crystal* 16 MHz, koneksi USB, *jack* listrik, *header* ICSP dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian *microcontroller*. Cukup dengan menghubungkannya ke komputer dengan kabel USB atau diberi power dengan *adaptor* AC-DC atau baterai.

2.5.2 Proteksi

Development board Arduino Uno R3 telah dilengkapi dengan *polyfuse* yang dapat *direset* untuk melindungi *port* USB komputer/laptop dari *korsleting* atau arus berlebih. Meskipun kebanyakan komputer telah memiliki perlindungan *port* tersebut didalamnya namun sikring pelindung pada Arduino Uno memberikan lapisan perlindungan tambahan saat menghubungkan Arduino ke komputer. Jika lebih dari 500mA ditarik pada *port* USB tersebut, sirkuit proteksi akan secara otomatis memutuskan hubungan, dan akan menyambung kembali ketika batasan aman telah kembali.

2.5.3 Power Supply

Board Arduino Uno dapat *supply* dengan power yang diperoleh dari koneksi kabel USB, atau *via power supply eksternal*. Pilihan power yang digunakan akan dilakukan secara otomatis. *External power supply* dapat diperoleh dari adaptor AC-DC atau bahkan baterai, melalui *jack* DC yang tersedia, atau menghubungkan langsung GND dan pin Vin yang ada di *board*. *Board* dapat beroperasi dengan *power* dari *external power supply* yang memiliki tegangan antara 6V hingga 20V. Namun ada beberapa hal yang harus anda perhatikan dalam rentang tegangan ini. Jika diberi tegangan kurang dari 7V, pin 5V tidak akan memberikan nilai murni 5V, yang mungkin akan membuat rangkaian bekerja dengan tidak sempurna. Jika diberi tegangan lebih dari 12V, *regulator* tegangan bisa *over heat* yang pada akhirnya bisa merusak pcb. Dengan demikian, tegangan yang di rekomendasikan adalah 7V hingga 12V. Beberapa pin *power* pada Arduino Uno :

- a. **GND**. Ini adalah *ground* atau negatif.
- b. **Vin**. Ini adalah pin yang digunakan jika anda ingin memberikan *power* langsung ke *board* Arduino dengan rentang tegangan yang disarankan 7V - 12V.

- c. **Pin 5V.** Ini adalah pin *output* dimana pada pin tersebut mengalir tegangan 5V yang telah melalui *regulator*.
- d. **3V3.** Ini adalah pin *output* dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui *regulator*
- e. **IOREF.** Ini adalah pin yang menyediakan referensi tegangan *microcontroller*. Biasanya digunakan pada *board shield* untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V

2.5.4 Memori

Chip ATmega328P pada Arduino Uno R3 memiliki memori 32 KB, dengan 0.5 KB dari memori tersebut telah digunakan untuk *bootloader*. Jumlah SRAM 2 KB, dan EEPROM 1 KB, yang dapat di baca-tulis dengan menggunakan EEPROM *library* saat melakukan pemrograman.

2.5.5 Input dan Output (I/O)

Seperti yang telah disebutkan sebelumnya, Arduino Uno memiliki 14 buah *digital* pin yang dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan *pull-up* sekitar 20-50k ohm (secara *default* dalam posisi *disconnect*). Nilai *maximum* adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip microcontroller*.

Beberapa pin memiliki fungsi khusus :

- a. **Serial**, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) *data serial*.
- b. **External Interrupts**, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupts*. Gunakan fungsi *attachInterrupt()*

- c. **PWM:** Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analog Write()*.
- d. **SPI :** Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*.
- e. **LED :** Pin 13. Pada pin 13 terhubung *built-in led* yang dikendalikan oleh *digital* pin no 13.
- f. **TWI :** Pin A4 (SDA) dan pin A5 (SCL) yang mendukung komunikasi TWI dengan menggunakan *Wire Library*.

Arduino Uno memiliki 6 buah *input analog*, yang diberi tanda dengan A0, A1, A2, A3, A4, A5. Masing-masing pin *analog* tersebut memiliki resolusi 10 *bits* (jadi bisa memiliki 1024 nilai). Secara *default*, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin AREF dengan menggunakan fungsi *analog Reference()*. Beberapa in lainnya pada *board* ini adalah :

- a. AREF. Sebagai referensi tegangan untuk *input analog*.
- b. *Reset*. Hubungkan ke *LOW* untuk melakukan reset terhadap *microcontroller*. Sama dengan penggunaan tombol reset yang tersedia.

2.5.6 Komunikasi

Arduino Uno R3 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan Arduino lainnya, atau dengan *microcontroller* lain nya. *Chip* Atmega328P menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX). *Chip* ATmega16U2 yang terdapat pada *board* berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai *Virtual Port* di komputer. *Firmware* 16U2 menggunakan *driver* USB *standar* sehingga tidak membutuhkan *driver* tambahan.

Pada Arduino *Software* (IDE) terdapat monitor serial yang memudahkan data *textual* untuk dikirim menuju Arduino atau keluar dari Arduino. *Led* TX dan RX akan menyala berkedip-kedip ketika ada data yang ditransmisikan melalui *chip* USB to *Serial* via kabel USB ke komputer. Untuk menggunakan komunikasi serial dari *digital* pin, gunakan *Software Serial library*.

Chip ATmega328P juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam Arduino *Software* (IDE) sudah termasuk *Wire Library* untuk memudahkan pengguna menggunakan bus I2C. Untuk menggunakan komunikasi SPI, gunakan SPI *library*.

2.5.7 Reset Otomatis (*software*)

Biasanya, ketika melakukan pemrograman *microcontroller*, harus menekan tombol *reset* sesaat sebelum melakukan *upload* program. Pada Arduino Uno telah dilengkapi dengan *auto reset* yang dikendalikan oleh *software* pada komputer yang terkoneksi. Salah satu jalur *flow control* (DTR) dari ATmega16U pada Arduino Uno R3 terhubung dengan jalur *reset* pada ATmega328P melalui sebuah *capacitor* 100nF. Ketika jalur tersebut diberi nilai *LOW*, *microcontroller* akan di *reset*. Dengan demikian proses *upload* akan jauh lebih mudah dan tidak harus menekan tombol *reset* (Djuandi, Feri. 2015).

2.6 Perangkat pendukung

2.6.1 Power supplay

Power supplay sebagai alat atau perangkat keras yang mampu menyuplai tenaga atau tegangan listrik secara langsung dari sumber tegangan listrik ke tegangan listrik yang lainnya. *Power supply* biasanya digunakan untuk komputer sebagai penghantar tegangan listrik secara langsung kepada komponen-komponen atau perangkat keras lainnya yang ada di komputer tersebut, seperti

hardisk, kipas, *motherboard* dan lain sebagainya. *Power supply* memiliki input dari tegangan yang berarus *alternating current* (AC) dan mengubahnya menjadi arus *direct current* (DC) lalu menyalurkannya ke berbagai perangkat keras yang ada di komputer . Karena memang arus *direct current* (DC)-lah yang dibutuhkan untuk perangkat keras agar dapat beroperasi, *direct current* biasa disebut juga sebagai arus yang searah sedangkan *alternating current* merupakan arus yang berlawanan (Newbie, 2016).



Gambar 2.8 *Power Supplay* (Newbie, 2016)

2.6.3 Sensor NTU

Nephelometer adalah suatu alat untuk mengukur kekeruhan yang memberikan hasil dalam satuan *Nephelometric Turbidity Unit* (NTU). NTU adalah satuan standar untuk mengukur kekeruhan. Pada *nephelometric* dan *turbidimetri*, sumber cahaya diproyeksikan melalui sampel cairan yang disimpan dalam wadah sampel *transparan* (DFRobot, 2018). Umumnya, *nephelometri* menggunakan sumber cahaya yang memiliki panjang gelombang *relative* singkat (misalnya, 500 nm – 800 nm) dan efektif digunakan untuk mendeteksi partikel dengan ukuran sangat kecil. Sedangkan *turbidity metri* umumnya menggunakan sumber cahaya yang memiliki panjang gelombang lebih panjang (misalnya, 800 nm – 1100 nm) dan efektif digunakan untuk mendeteksi partikel dengan ukuran yang lebih besar. Jika seberkas cahaya dilewatkan melalui sampel keruh, intensitasnya dikurangi dengan hamburan, dan jumlah cahaya yang tersebar tergantung pada konsentrasi

dan distribusi ukuran partikel. Dalam *nephelometri* intensitas cahaya yang tersebar diukur, sedangkan dalam *turbidimetri*, intensitas cahaya yang ditransmisikan melalui sampel diukur. Untuk mengetahui kekeruhan dengan satuan NTU maka dibutuhkan rumus :

” Kekeruhan = $100.00 - (\text{Tegangan} / \text{hasil nilai tegangan pada saat membaca air jernih}) \times 100.00$ “.



Gambar 2.9 Sensor NTU (DFRobot, 2018)

2.6.4 Sensor Suhu

DS18B20 adalah sensor suhu *digital* seri terbaru dari *Maxim IC* (dulu yang buat adalah *Dallas Semiconductor*, lalu dicaplok oleh *Maxim Integrated Products*). Sensor ini mampu membaca suhu dengan ketelitian 9 hingga 12Bit, rentang -55°C hingga 125°C dengan ketelitian $(\pm 0.5^{\circ}\text{C})$. Setiap sensor yang diproduksi memiliki kode unik sebesar 64-Bit yang disematkan pada masing - masing *chip*, sehingga memungkinkan penggunaan sensor dalam jumlah besar hanya melalui satu kabel saja (*single wire data bus/1-wire protocol*). Ini merupakan komponen yang luar biasa, dan merupakan satu patokan dari banyak proyek-proyek data *logging* dan kontrol berbasis temperatur di luar sana (*Maxim Integrated, 2008*).



Gambar 2.10 Sensor suhu DS18B20 (*Maxim Integrated, 2008*)

a. Fitur dari sensor suhu DS18B20

Sebagai acuan dan informasi pendukung, sensor ini memiliki fitur utama sebagai berikut:

1. Antarmuka hanya menggunakan satu kabel sebagai komunikasi (menggunakan protokol *Unique 1-Wire*).
2. Setiap sensor memiliki kode pengenalan unik 64-bit yang tertanam di *onboard* ROM.
3. Kemampuan *multidrop* yang menyederhanakan aplikasi penginderaan suhu terdistribusi.
4. Tidak memerlukan komponen tambahan.
5. Juga bisa diumpankan daya melalui jalur datanya. Rentang dayanya adalah 3.0V hingga 5.5V.
6. Bisa mengukur *temperature* mulai dari -55°C hingga $+125^{\circ}\text{C}$.
7. Memiliki akurasi $\pm 0.5^{\circ}\text{C}$ pada rentang -10°C hingga $+85^{\circ}\text{C}$.
8. Resolusi sensor bisa dipilih mulai dari 9 hingga 12 bit.
9. Bisa mengkonversi data suhu ke 12-bit *digital word* hanya dalam 750 milidetik (maksimal).
10. Memiliki konfigurasi alarm yang bisa disetel (*nonvolatile*).

11. Bisa digunakan untuk fitur pencari alarm dan alamat sensor yang temperaturnya diluar batas (*temperature alarm condition*).
12. Penggunaannya bisa dalam lingkungan kendali termostatis, sistem industri, produk rumahan, termometer, atau sistem apapun yang memerlukan pembacaan suhu.

2.6.5 LCD (*Liquid Crystal Display*)

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari back-lit. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik (Arduino, 2016).



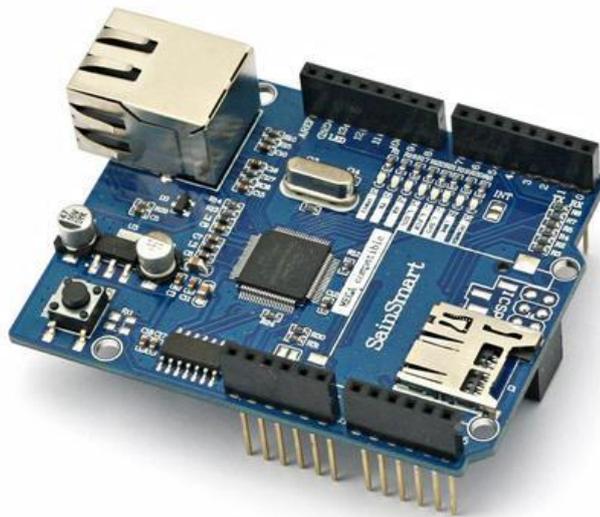
Gambar 2.11 LCD (Arduino, 2016)

Cara kerja LCD secara umum adalah *port* RW diberi logika rendah "0". Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada tabel diskripsi, *interface* LCD merupakan sebuah *parallel* bus, dimana hal ini sangat memudahkan dan

sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap *nibblenya*). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa *microcontroller* mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menyet EN ke kondisi *high* "1" dan kemudian menyet dua jalur kontrol lainnya (RS dan RW) atau juga mengirimkan data ke jalur data *bus*. Saat jalur lainnya sudah siap, EN harus diset ke "0" dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan set EN kembali ke *high* "1". Ketika jalur RS berada dalam kondisi *low* "0", data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau "1", data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf "A" pada layar maka RS harus diset ke "1". Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* "1", maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu Get LCD status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke "0". Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7.

2.6.7 Arduino Ethernet shield

Arduino *Ethernet Shield* adalah modul yang berfungsi menghubungkan Arduino *board* dengan jaringan internet, karena itu berdasar pada *Wiznet W5100 ethernet chip datasheet* (Hanrun, 2017).



Gambar 2.12 *Arduino Ethernet Shield* (Hanrun, 2017)

Untuk Menghubungkan dan menggunakan modul hingga dapat terkoneksi internet cukup mudah, caranya dengan memasang modul tersebut di atas *Arduino board*, sambungkan dengan kabel *network* RJ45, di dalam arduino ethernet sendiri terdapat *slot mikro* SD yang berbungsi sebagai tempat penyimpanan file sedangkan untuk mengakses *micro* SD *card* menggunakan *library* SD ,untuk jenis arduino *board* yang bisa di pasang dengan *ethernet shield* W5100 yaitu arduino uno dan mega. *Spesifikasi Ethernet Controller* :

1. *Chip* Wiznet W5100 dengan *internal buffer* 16 Kb.
2. Kecepatan koneksi 10 / 100Mb (*Fast-Ethernet*).
3. Papan ini terhubung dengan Arduino melalui *port* SPI.
4. Dapat mendukung hingga 4 koneksi simultan.

2.6.8 Router (Acces Point)

Router adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal

sebagai *routing*. Proses *routing* terjadi pada lapisan 3 (Lapisan jaringan seperti Internet *Protocol*) dari *stack* protokol tujuh lapis OSI (Ferisandriya, 2015).

Router memiliki fasilitas DHCP (*Dynamic Host Configuration Protocol*), dengan mensetting DHCP, maka kita dapat membagi *IP Address*, fasilitas lain dari *Router* adalah adanya NAT (*Network Address Translator*) yang dapat memungkinkan suatu *IP Address* atau koneksi internet disharing ke *IP Address* lain.



Gambar 2.13 *Router* (ferisandriya, 2015)

Router dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. *Router* juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya *router wireless* yang pada umumnya selain dapat menghubungkan komputer dengan menggunakan radio, juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari *Ethernet* ke *Token Ring*.

a. Jenis – Jenis Router

1. Router aplikasi

Router jenis ini adalah sebuah aplikasi yang bisa anda *instal* pada sistem operasi komputer, sehingga sistem operasi computer tersebut dapat bekerja seperti *router*, misalnya aplikasi *WinGate*, *WinProxy*, *Winroute*, *SpyGate* dll.

2. Router Hardware

Router hardware adalah sebuah *hardware* yang memiliki kemampuan seperti *router*, maka dengan *hardware* tersebut anda dapat membagi *IP Address*, *Router hardware* dapat digunakan untuk membagi jaringan internet pada suatu wilayah, misalnya dari *router* ini adalah *access point*, wilayah yang mendapat *IP Address* dan koneksi internet disebut *Hot Spot Area*.

3. Router PC

Router PC adalah sebuah komputer yang dimodifikasi sedemikian rupa sehingga dapat digunakan sebagai *router*. Untuk membuat sebuah *router PC* tidak harus menggunakan komputer dengan spesifikasi yang tinggi. Komputer dengan *prosesor pentium* dua, *hard drive* 10 GB dan ram 64 serta telah tersedia LAN *Card* sudah bisa digunakan sebagai *router PC*. Komputer yang dijadikan *router* ini harus *diinstal* dengan sistem operasi khusus untuk *router*. Sistem operasi yang populer untuk *router PC* saat ini adalah *Microtik*.

b. Fungsi – Fungsi Router

1. Fungsi utama *router* yaitu menghubungkan beberapa jaringan untuk menyampaikan data dari suatu jaringan ke jaringan yang lain. Namun *router* berbeda dengan *Switch*, karena *Switch* hanya digunakan untuk menghubungkan beberapa komputer dan membentuk LAN (*local area*

network). Sedangkan *router* digunakan untuk menghubungkan antar satu LAN dengan LAN yang lainnya.

2. *Router* juga berfungsi untuk mentransmisikan informasi dari satu jaringan ke jaringan lain yang sistem kerjanya seperti *Bridge*.
3. *Router* juga berfungsi untuk menghubungkan jaringan lokal ke sebuah koneksi DSL, biasa juga disebut *DSL router*. *Router* ini umumnya memiliki fungsi *firewal* untuk melakukan penapisan paket berdasarkan sumber serta alamat tujuan paket tersebut, namun tidak semua *router* memiliki fungsi yang sama. *Router* yang memiliki fitur penapisan paket dapat juga disebut sebagai *packet – filtering router*. Fungsi umum *router* ini memblokir lalu lintas data yang dipancarkan secara *broadcast* sehingga dapat mencegah adanya *broadcast storm* yang bisa menyebabkan kinerja jaringan melambat.

c. Cara Kerja Router

Fungsi utama *Router* adalah merutekan paket (informasi). Sebuah *Router* memiliki kemampuan *Routing*, artinya *Router* secara cerdas dapat mengetahui kemana rute perjalanan informasi (paket) akan dilewatkan, apakah ditujukan untuk *host* lain yang satu *network* ataukah berada di *network* yang berbeda. Jika paket-paket ditujukan untuk *host* pada *network* lain maka *router* akan meneruskannya ke *network* tersebut. Sebaliknya, jika paket-paket ditujukan untuk *host* yang satu *network* maka *router* akan menghalangi paket-paket keluar.

2.7 Software pendukung

Software atau suatu program aplikasi / perangkat lunak merupakan perangkat pendukung yang harus ada / tersedia dalam mengakses aplikasi.

2.7.1 Arduino IDE

Aplikasi ini berguna sebagai *text editor* untuk membuat, membuka, mengedit, dan juga mevalidasi kode serta untuk di *upload* ke *board* Arduino. Program yang digunakan pada Arduino disebut dengan istilah “*sketch*” yaitu *file source code* arduino dengan ekstensi *.ino* (sinau arduino, 2016).



Gambar 2.14 Tampilan *Software* Arduino IDE

1. Bagian-bagian Arduino IDE

Seperti *teks* editor pada umumnya yaitu memiliki fitur untuk *cut / paste* dan untuk *find / replace teks*. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengekspor dan juga sebagai tempat menampilkan kesalahan. Konsol *log* menampilkan *output teks* dari *Arduino Software (IDE)*, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah jendela menampilkan papan dikonfigurasi dan *port serial*. Tombol *toolbar* memungkinkan Anda untuk memverifikasi dan *meng-upload* program, membuat, membuka, dan menyimpan *sketch*, dan membuka monitor *serial*.

- a. **Verify** pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi di-*upload* ke *board* Arduino, biasakan untuk memverifikasi terlebih dahulu *sketch* yang dibuat. Jika ada kesalahan pada *sketch*, nanti akan muncul *error*. Proses *Verify / Compile* mengubah *sketch* ke *binary code* untuk di-*upload* ke *microcontroller*.
- b. **Upload** tombol ini berfungsi untuk *mengupload* *sketch* ke *board* Arduino. Walaupun kita tidak mengklik tombol *verify*, maka *sketch* akan di-*compile*, kemudian langsung di-*upload* ke *board*. Berbeda dengan tombol *verify* yang hanya berfungsi untuk memverifikasi *source code* saja.
- c. **New Sketch** Membuka *window* dan membuat *sketch* baru.
- d. **Open Sketch** Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi *file .ino*
- e. **Save Sketch** menyimpan *sketch*, tapi tidak disertai dengan mengkompile.
- f. **Serial Monitor** Membuka *interface* untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
- g. **Keterangan Aplikasi** pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "*Compiling*" dan "*Done Uploading*" ketika kita meng-*compile* dan *mengupload sketch* ke *board* Arduino.
- h. **Konsol log** Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal, ketika aplikasi *mengcompile* atau ketika ada kesalahan pada *sketch* yang kita buat, maka informasi *error* dan baris akan diinformasikan di bagian ini.
- i. **Baris Sketch** bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.
- j. **Informasi Board dan Port** Bagian ini menginformasikan *port* yang dipakai oleh *board* Arduino.

2.7.2 Balsamiq Mockup

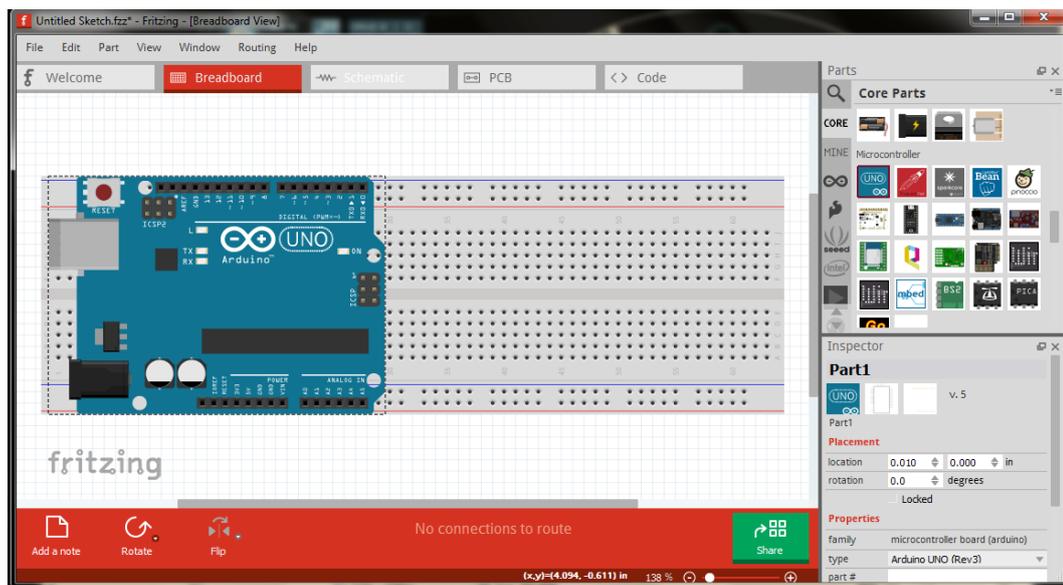
Mockup artinya model atau replika mesin atau struktur, yang digunakan untuk tujuan *instruksional* atau *eksperimental*. *Balsamiq mockup* adalah program aplikasi yang digunakan dalam pembuatan tampilan *user interface* sebuah aplikasi. *Software* ini sudah menyediakan tools yang dapat memudahkan dalam membuat *desain prototyping* aplikasi yang akan kita buat. *Software* ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Kelebihan *Balsamiq Mockups* dibanding *software* pembuat *mockup* lainnya adalah aplikasi ini berbasis *cloud*, disertai aplikasi *desktop* yang memungkinkan kita dengan cepat dan mudah membuat rancangan *website*. Dengan konten yang terbuat seperti dari gambaran tangan, akan membuat kita fokus pada pemecahan masalah *user interface* yang lebih besar, dari pada perincian *website*. Aplikasi ini bisa digunakan untuk sistem operasi *Windows*, *Mac OS*, dan *Linux*.

2.7.3 Fritzing

Fritzing merupakan sebuah *software* yang bersifat *open source* untuk merancang rangkaian elektronika. *Fritzing* dikembangkan di *University of Applied of Postdam*. *Software* tersebut mendukung para penggemar elektronika untuk membuat *prototype product* dengan merancang rangkaian berbasis *microcontroller* Arduino. Memungkinkan para perancang elektronika pemula sekalipun untuk membuat *layout* PCB yang bersifat *custom*. Tampilan dan penjelasan yang ada pada *Fritzing* bisa dengan mudah dipahami oleh seseorang yang baru pertama kali menggunakannya (INTJ Nerd, 2016).

Dengan *feature* yang dimilikinya tersebut, *Fritzing* dapat disebut sebagai sebuah *software Electronic Design Automation (EDA)* untuk *non-engineer*. Dalam perancangannya, *Fritzing* menggunakan tampilan *breadboard* sebagai *prototype* penyusunan komponen elektronika. Beberapa komponen yang ada pada *Fritzing* mulai dari *Arduino*, *Raspberry Pi*, berbagai sensor, voltage regulator, resistor, dan masih banyak lagi lainnya. Berikut ini adalah tampilan design *Fritzing*.



Gambar 2.15 Tampilan *Software Fritzing* (INTJ Nerd, 2016)