

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1. Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di ICT Laboratorium, Universitas Multimedia Nusantara, penulis ditempatkan sebagai staf magang Laboratorium ICT. Kerja magang dikoordinasikan oleh Yustinus Eko Sulistio sebagai Kepala Laboratorium ICT. Pada awalnya, Yustinus Eko Sulistio selaku pembimbing lapangan memberikan informasi tentang alat *mindwave* yang akan ditambahkan fungsi mendeteksi *eye blink*, serta tugas-tugas lain untuk departemen *marketing*.

#### 3.2. Tugas-Tugas yang Dilakukan

Penambahan fungsi *eye blink* pada robot *mindwave* yang dikerjakan dalam kerja magang di ICT Laboratorium, Universitas Multimedia Nusantara. Beberapa tugas, secara umum, yang dijalankan penulis adalah sebagai berikut,

- a. mempelajari tahapan dan cara kerja *mindwave* dalam mengirim sinyal dan bagaimana Arduino memproses sinyal tersebut agar dapat menjalankan robot,
- b. melakukan *upload* program ulang ke robot *mindwave* yang sudah ada dan melakukan uji coba ke robot,

- c. memperbaiki nilai potentiometer yang membuat robot tidak stabil dan konektor baterai 9V yang mengalami kerusakan, serta mengganti baterai 9V robot ketika baterai sudah tidak kuat untuk menggerakkan servo,
- d. menambahkan fungsi mendeteksi *eye blink* di robot, sehingga ketika robot mendeteksi pengguna melakukan kedipan mata, robot akan belok kearah kiri,
- e. mempelajari cara kerja “*Scratch for Arduino*” yang akan digunakan sebagai materi *workshop*,
- f. menjaga *stand* ICT dan menjadi pembicara *workshop* “*Scratch for Arduino*” serta memilih peserta *workshop* yang menjalani magang di ICT Laboratorium dari 29 Juni sampai dengan 10 Juli 2015.

### **3.3. Uraian Pelaksanaan Kerja Magang**

Pelaksanaan kerja magang dapat diuraikan menjadi tiga bagian, yaitu proses pelaksanaan, kendala yang ditemukan, dan solusi atas kendala yang ditemukan.

#### **3.3.1. Proses Pelaksanaan**

Ada beberapa hal yang dilakukan pada saat melaksanakan kerja magang, dimulai dengan mempelajari tahapan dan cara kerja *mindwave* serta mengetahui bagaimana Arduino memproses masukan dari *mindwave* tersebut agar dapat menggerakkan robot. Untuk *workshop*

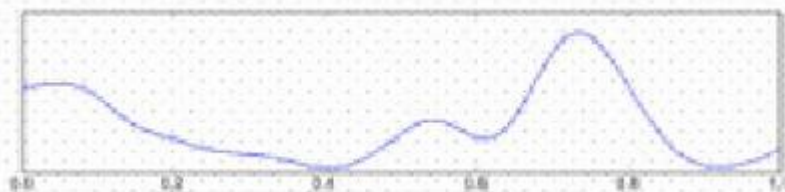
*Scratch for Arduino*, penulis mempelajari terlebih dahulu penggunaan *Scratch for Arduino* dan mempersiapkan alat-alat yang dibutuhkan untuk *workshop* tersebut.

### 3.3.1.1. Cara Kerja *Mindwave*

*Neurosky mindwave* merupakan alat *Brain-Computer Device* (BCI) yang dapat menerima gelombang otak. Menggunakan *thinkGear*, teknologi yang terdapat di dalam setiap produk *Neurosky, mindwave* dapat menerima gelombang otak seperti gelombang delta, theta, alpha, gamma dan beta.

#### 1. Gelombang delta

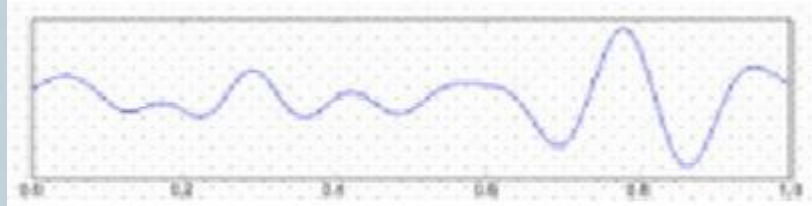
Gelombang delta seperti pada gambar 3.1, merupakan gelombang terendah dan terjadi dalam kisaran 0,5 Hz sampai dengan 4 Hz. Keadaan ini terjadi ketika manusia mengalami tidur dalam yang sangat lelap dan memasuki fase tidur tanpa mimpi serta ketika manusia berada dalam keadaan tidak sadar, seperti ketika mengalami koma.



Gambar 3.1. Grafik Gelombang Delta – EEG Scan (Sumber :  
Naufal Hafiluddin, 2013)

## 2. Gelombang theta

Gelombang theta pada gambar 3.2, merupakan transformasi dari keadaan tidak sadar penuh menjadi lebih sadar. Ini terjadi ketika seseorang mengalami keadaan berfantasi, berimajinasi, atau berpikir tentang hal-hal yang kreatif. Gelombang ini muncul pada saat seseorang mengalami keadaan tidur ringan atau sangat mengantuk sehingga tidak merespon adanya stimulus dari luar dirinya. Dalam keadaan sadar, gelombang ini terjadi ketika seseorang menjalani meditasi dalam atau berada dalam hipnotis.



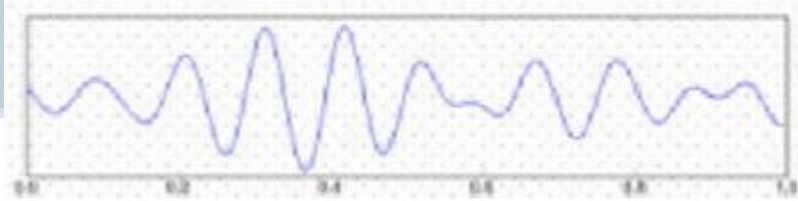
Gambar 3.2. Grafik Gelombang Theta – EEG Scan (Sumber :  
Naufal Hafiluddin, 2013)

Anak-anak biasanya banyak mengalami fase gelombang theta meski dalam kondisi normal. Hal ini disebabkan cara berpikir anak-anak yang cenderung mengkhayal. Namun, seiring perkembangan si anak, kondisi ini akan berangsur-angsur memudar dan ketika sudah dewasa, fase ini akan hilang.

Gelombang theta muncul dalam rentang frekuensi 4 Hz sampai dengan 7 Hz.

### 3. Gelombang alpha

Otak manusia menghasilkan gelombang alpha seperti pada gambar 3.3, ketika kondisi mental manusia mengalami relaksasi atau mulai istirahat dan dalam keadaan mulai mengantuk. Gelombang ini juga dihasilkan ketika terjadi perubahan fase dari keadaan sadar menjadi tidak sadar, namun ketidaksadarannya belum terlalu dalam. Artinya gelombang ini terjadi ketika adanya peralihan frekuensi gelombang dari gelombang theta ke gelombang beta. Kondisi ini biasanya tidak terjadi lama dan hanya menjadi kondisi peralihan saja. Selain itu, ketika melakukan meditasi ringan gelombang ini pun dapat dideteksi.



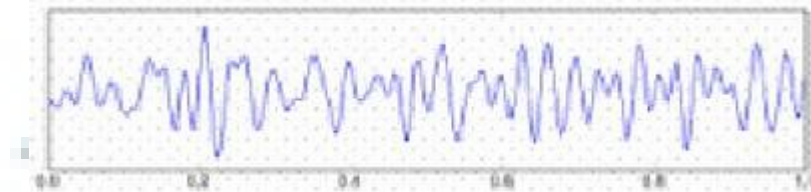
Gambar 3.3. Grafik Gelombang Alpha – EEG Scan (Sumber :  
Naufal Hafiluddin, 2013)

Gelombang alpha dihasilkan dari rentang frekuensi 8 Hz sampai dengan 12 Hz.

### 4. Gelombang beta

Gelombang beta pada gambar 3.4, dihasilkan ketika manusia memiliki kesadaran dan konsentrasi penuh atau dalam kondisi normal. Beberapa peneliti membagi lagi kriteria konsentrasi ke

gelombang-gelombang turunan dari beta, yaitu gelombang low beta, midrange beta, serta high beta.



Gambar 3.4. Grafik Gelombang Beta – EEG Scan (Sumber :  
Naufal Hafiluddin, 2013)

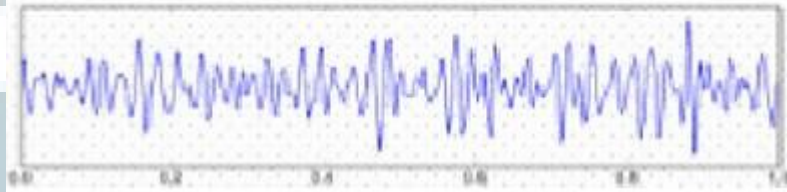
Gelombang low beta dihasilkan dalam rentang frekuensi 12 Hz hingga 15 Hz. Gelombang ini biasanya muncul ketika manusia sedang fokus, tapi tetap dalam kondisi rileks. Tetapi, bagi penderita epilepsi, ADHD atau autisme frekuensi ketika mereka fokus akan berbeda dari manusia normal karena kecenderungan mereka yang sulit melakukan konsentrasi penuh atau fokus.

Gelombang midrange beta terjadi dalam rentang frekuensi 16 Hz sampai dengan 20 Hz. Kondisi ini muncul ketika manusia sedang melakukan kegiatan berpikir, fokus, berkonsentrasi penuh dan sadar secara penuh terhadap dirinya dan keadaan di sekitarnya.

Gelombang high beta terjadi ketika manusia sedang dalam kondisi waspada terhadap sesuatu atau ketika manusia mengalami pergolakan dalam batinnya. Gelombang ini berada dalam rentang frekuensi 21 Hz sampai dengan 30 Hz.

## 5. Gelombang gamma

Gelombang gamma di gambar 3.5, terjadi ketika seseorang dalam keadaan kesadaran penuh namun dalam kondisi mental yang tinggi, seperti saat panik, ketakutan, gugup atau gelisah.

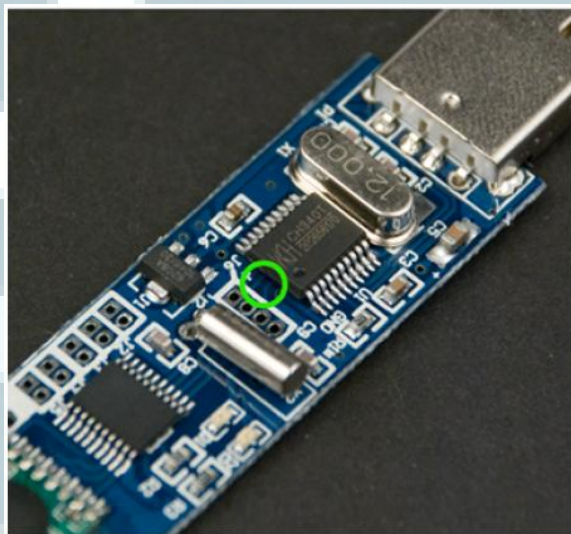


Gambar 3.5. Grafik Gelombang Gamma – EEG Scan (Sumber : Naufal Hafiluddin, 2013)

Selain itu, dengan *thinkGear*, *mindwave* juga dapat mengambil nilai *eSense* seperti *Attention* dan *Meditation*. *Attention* digunakan untuk mengukur tingkat konsentrasi seseorang. Semakin tinggi nilai *attention*, semakin tinggi tingkat konsentrasinya. Sementara untuk *meditation* akan semakin meningkat ketika seseorang menjernihkan pikirannya dari segala masalah maupun gangguan.

Nilai *eSense* biasanya berkisar dari 1-100. Jika nilai konsentrasi pengguna antara 1-20, maka level konsentrasinya sangat rendah. Menandakan pikirannya sedang terganggu, terjadi pergolakan atau ada tanda-tanda keabnormalan. Untuk nilai 21-40, menandakan konsentrasi kurang. Nilai 41-60 menandakan nilai netral atau konsentrasi di tingkat normal. Nilai 61-80 menandakan tingkat konsentrasi agak tinggi. Nilai 80-100 menandakan tingkat konsentrasi yang paling tinggi dalam *eSense*.

*Mindwave* mengirim data dan Arduino menerima data tersebut melalui *dongle Bluetooth Serial Port Profile (SPP)* dengan *baudrate* 57600. Untuk menggunakannya di Arduino, *cover* luar *dongle* dibuka dan koneksi yang ada di dalam lingkaran hijau seperti pada gambar 3.6 diputuskan. Hal ini dimaksudkan agar koneksi ke USB *port* diputus dan memungkinkan komunikasi serial dengan Arduino.

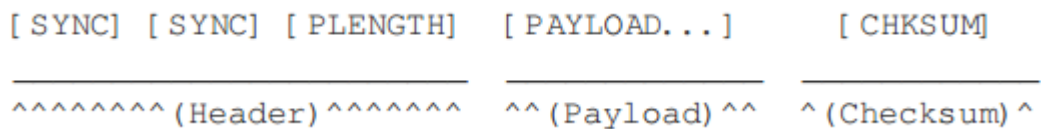


Gambar 3.6. *Dongle Bluetooth Mindwave* (Sumber : Tero Karvinen dan Kimmo Karvinen, 2011)

*Mindwave* mengirim digital data sebagai *asynchronous serial stream of bytes*. Paket data yang dikirim berupa *header*, *payload* dan *checksum* seperti pada gambar 3.7. Besar *payload* dapat mencapai maksimum 169 *byte*, sementara *sync*, *payloadlength* dan *checksum* masing-masing satu *byte*. *Header* diawali dengan dua *byte* sinkronisasi dan satu besar paket. Dua *byte* awal digunakan sebagai penanda bahwa ada paket baru yang datang dan biasanya bernilai 0xAA (desimal 170). Sinkronisasi sebanyak dua *byte* untuk mencegah kesalahpahaman dalam



penerimaan paket jika dalam *payload* data terdapat nilai 170. Meskipun kemungkinan dua *byte* data bernilai 170 masih ada, tetapi adanya *payloadlength* dan *checksum* yang terletak di sebelum dan sesudah data dapat mencegah terjadinya kesalahpahaman tersebut.



Gambar 3.7. Struktur Paket *Mindwave* (sumber :

[http://wecam.org/ece516/mindset\\_communications\\_protocol.pdf](http://wecam.org/ece516/mindset_communications_protocol.pdf))

Dari gambar 3.8, dapat diketahui arti dari data yang terdapat dalam *payload*. Kualitas dari data dapat dilihat setelah data 0x02. Jika data yang muncul setelah 0x02 adalah nol, maka kualitas paket data tersebut baik. Untuk membaca konsentrasi, data diawali dengan 0x04. Meditasi dapat dilihat setelah data 0x05. Kumpulan nilai gelombang otak seperti delta, theta, gamma, alpha dan beta dapat dilihat setelah data 0x83. Untuk mengecek kedipan mata ada setelah data 0x16. Tapi tipe data ini tidak akan muncul jika tidak menggunakan *library ThinkGear*. Untuk saat ini, *library* tersebut hanya dapat digunakan untuk C#.NET.

### Single-Byte CODEs

| Extended Code Level | [ CODE] | (Byte) [ LENGTH] | Data Value Meaning   |
|---------------------|---------|------------------|--|
| 0                   | 0x02    |                  | - POOR_SIGNAL Quality (0-255)                                |
| 0                   | 0x04    |                  | - ATTENTION eSense (0 to 100)                                |
| 0                   | 0x05    |                  | - MEDITATION eSense (0 to 100)                               |
| 0                   | 0x16    |                  | - Blink Strength. (0-255) Sent only when Blink event occurs. |

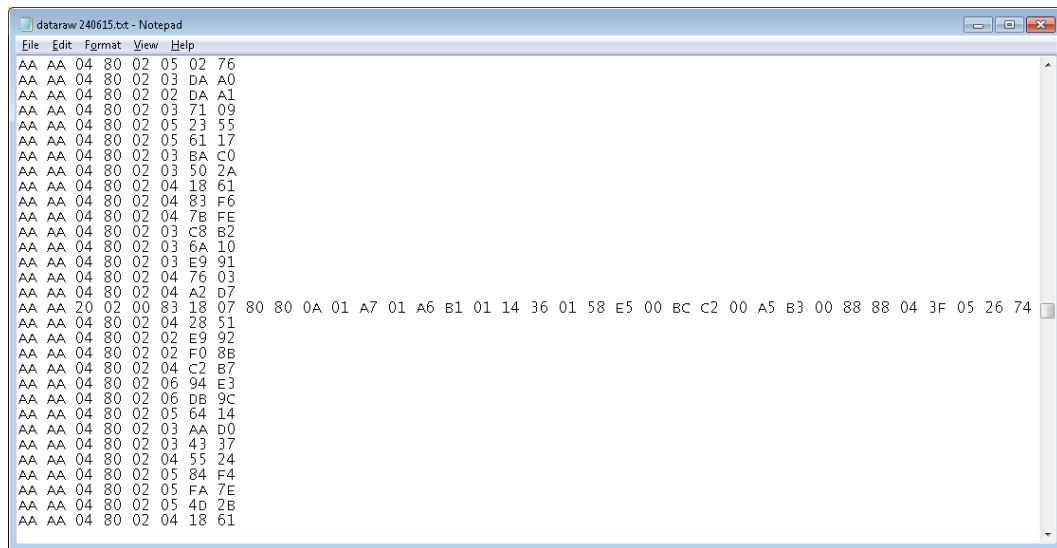
### Multi-Byte CODEs

| Extended Code Level | [ CODE] | (Byte) [ LENGTH] | Data Value Meaning  |
|---------------------|---------|------------------|---|
| 0                   | 0x80    | 2                | RAW Wave Value: a single big-endian 16-bit two's-compliment signed value (high-order byte followed by low-order byte) (-32768 to 32767)   |
| 0                   | 0x83    | 24               | ASIC_EEG_POWER: eight big-endian 3-byte unsigned integer values representing delta, theta, low-alpha, high-alpha, low-beta, high-beta, low-gamma, and mid-gamma EEG band power values |
| Any                 | 0x55    |                  | - NEVER USED (reserved for [ EXCODE])   |
| Any                 | 0xAA    |                  | - NEVER USED (reserved for [ SYNC])   |

Gambar 3.8. Penjelasan Data dari Paket Data (sumber :

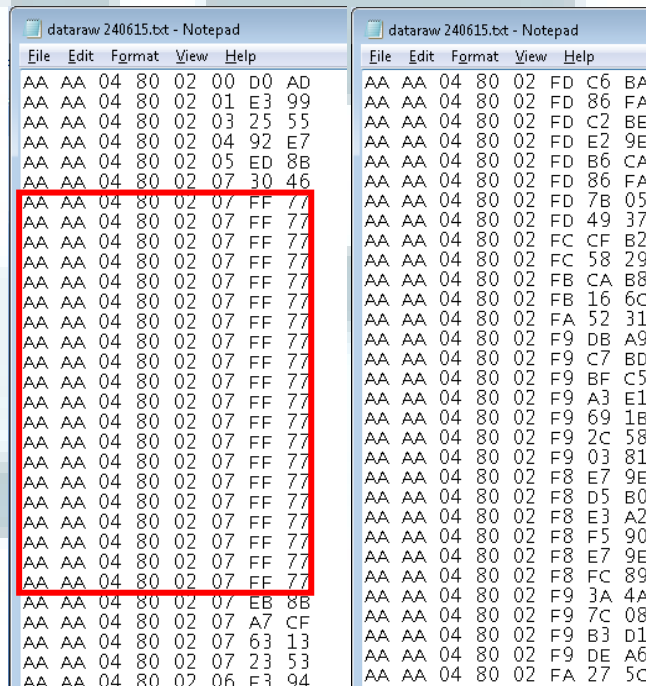
[http://wearcam.org/ece516/mindset\\_communications\\_protocol.pdf](http://wearcam.org/ece516/mindset_communications_protocol.pdf))

Data yang dikirim dari *mindwave*, seperti gambar 3.9, biasanya berupa *small packet* dan *big packet*. *Small packet* biasa dikirim sebanyak 512 kali barulah satu *big packet* dikirim. Data tentang konsentrasi, meditasi dan gelombang otak dapat ditemukan di *big packet*. Sementara untuk mendeteksi kedipan mata, dapat mencermati nilai dari *small packet*.



Gambar 3.9. Data *Raw Mindwave*

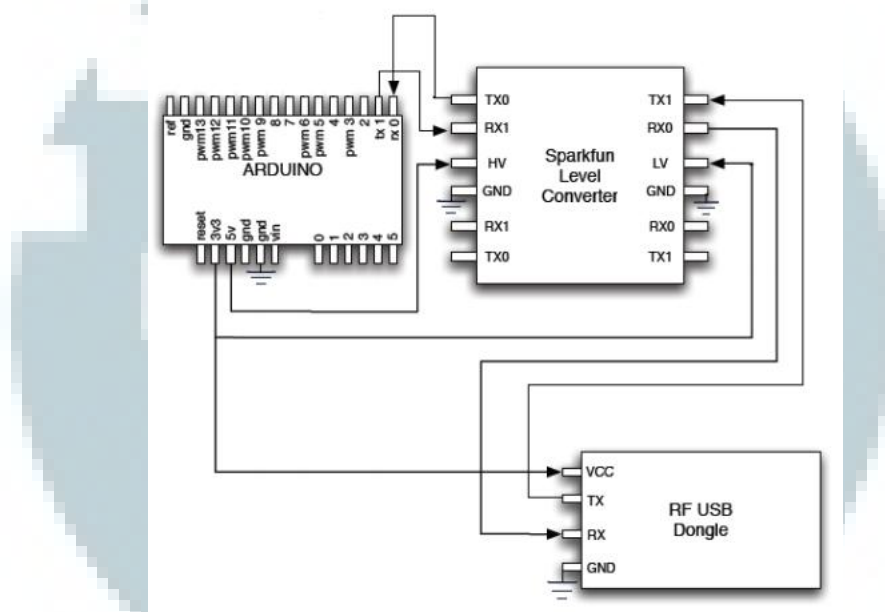
Jika pengguna menggunakan *headset mindwave*, data akan sama seperti pada gambar 3.9. Tapi, ketika pengguna melakukan kedipan mata, akan terjadi perubahan nilai di *small packet* seperti yang dapat dilihat pada gambar 3.10.



Gambar 3.10. Data *Raw* ketika *blink* (kiri), Data *Raw* normal (kanan)

### 3.3.1.2. Cara Arduino Memproses Data dari *Mindwave*

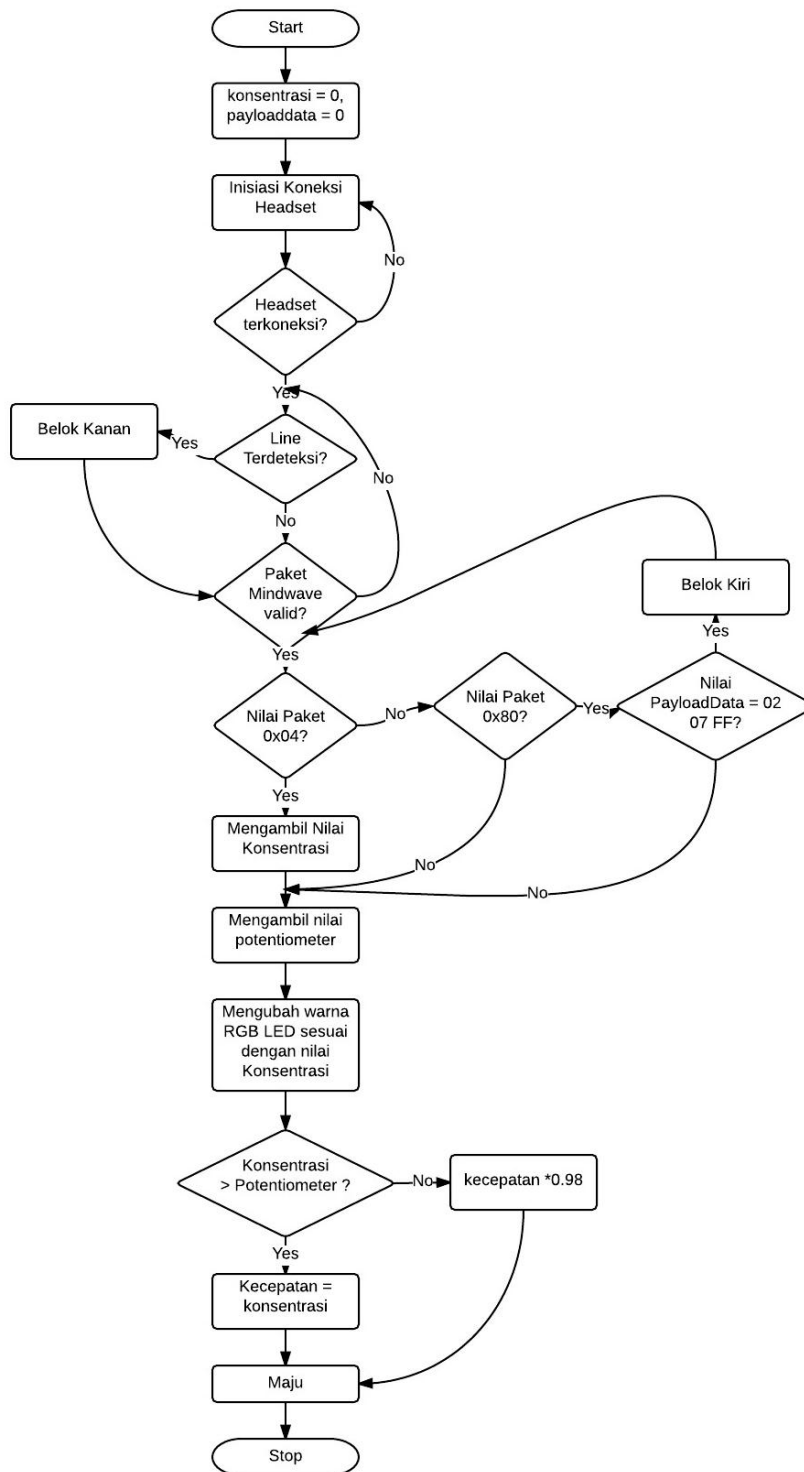
Dari *dongle Bluetooth* yang sudah terpasang, Arduino kemudian dapat mengambil sinyal dari *mindwave*. Gambar 3.11 merupakan skematik cara pemasangan dari *dongle mindwave* ke Arduino.



Gambar 3.11. Skematik Sirkuit dari *Dongle Mindwave* ke Arduino (Sumber : [http://developer.neurosky.com/docs/doku.php?id=arduino\\_tutorial](http://developer.neurosky.com/docs/doku.php?id=arduino_tutorial))

Ketika program akan diunggah ke dalam Arduino, sambungan Tx dan Rx harus dilepas dari pin Arduino agar tidak mengganggu proses unggah. Setelah status di *Integrated Development Environment* (IDE) Arduino sudah “*done uploading*”, maka Tx dan Rx dapat disambungkan kembali ke pin Arduino. IDE Arduino sendiri dapat diunduh di <https://www.arduino.cc/en/Main/Software>.

Gambar 3.12 merupakan *flow chart* dari program Arduino.



Gambar 3.12. Flow Chart *Mindwave*

Program mengecek apakah *headset* telah terkoneksi dengan Arduino atau belum. Jika sudah terdeteksi, program mengecek apakah robot mendeteksi adanya garis hitam atau tidak. Jika *line* terdeteksi, maka robot akan memutar ke kanan. Lalu, Arduino akan memproses data yang diterima dari *dongle mindwave* dengan potongan program seperti gambar

3.13.

```

switch (payloadData[i]) {
  case 0xD0:
    sayHeadsetConnected();
    break;
  case 4:
    i++;
    attention = payloadData[i] / 100.0;
    break;
  case 2:
    i++;
    poorQuality = payloadData[i];
    if (170 < poorQuality) { // max 200
      setYellow();
      attention = 0.0;
      return -4;
    }
    break;
  case 0xD1: // Headset Not Found
  case 0xD2: // Headset Disconnected
  case 0xD3: // Request Denied
  case -70:
    //wave(speakerPin, 900, 500);
    attention = 0.0;
    setWhite();
    return -5;
    break;
  case 0x80:
    i++;
    if (payloadData[i] == 0x02 && payloadData[i+1] == 0x07 && payloadData[i+2] == 0xFF){
      turnLeft();
    }
    i = i+2;
    break;
  case 0x83:
    i = i + 25;
    break;
} // switch

```

Gambar 3.13. Potongan Program Arduino

Ketika baru dinyalakan, bagian awal paket akan berisi 0xD0 sebagai tanda bahwa *headset mindwave* dan *dongle* yang tersambung

dengan Arduino sudah terkoneksi. Ketika koneksi terjadi, lampu di *headset* berubah warna dari merah menjadi biru. Begitu juga LED yang ada di *dongle*. Setelah itu, Arduino akan memproses masing-masing paket. Jika paket yang diterima *small packet*, Arduino akan mengecek nilainya. Jika nilai tersebut sama dengan nilai data ketika terjadi *blink* maka robot akan berbelok arah ke kiri. Tetapi, jika nilainya tidak sama, maka paket tersebut akan dilewati.

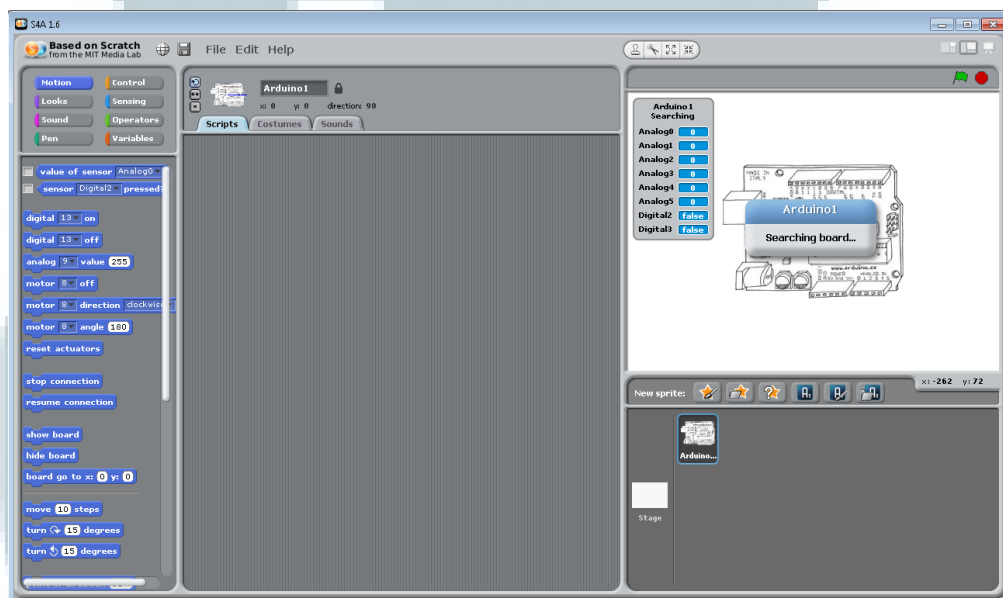
Untuk bagian konsentrasi, nilainya terdapat dalam *big packet* bersama dengan data meditasi dan gelombang delta, theta, alpha, gamma dan beta. Ketika Arduino membaca data berisi 0x04, maka Arduino akan mengambil nilai setelah data 0x04 tersebut sebagai nilai konsentrasi. Warna di *Red Green Blue* (RGB) LED akan berubah sesuai dengan tingkat konsentrasi pengguna. Warna biru menandakan pengguna tidak konsentrasi. Ketika daya konsentrasi meningkat, akan terjadi perubahan warna dari biru ke merah. Merah menandakan nilai konsentrasi tinggi. Ketika RGB LED sudah berwarna merah, maka robot *mindwave* akan bergerak maju.

### **3.3.1.3. Tentang *Scratch for Arduino***

*Scratch* merupakan bahasa pemrograman visual berorientasi pendidikan yang memungkinkan pemula untuk mendapatkan hasil, tanpa perlu terlebih dahulu belajar menulis sintaksis yang benar. Anak-anak dapat menggunakan *Scratch* untuk bereksperimen dan belajar untuk

berpikir kreatif, berkolaborasi dan mengetahui alasannya secara sistematis. Dengan demikian, anak-anak dapat lebih kenal dan akrab dengan konsep dasar pemrograman komputer.

*Scratch for Arduino* (S4A), seperti pada gambar 3.14, merupakan *Scratch* yang dimodifikasi sehingga memungkinkan pemula yang baru mengenal pemrograman perangkat keras untuk mulai mencoba program-program sederhana untuk Arduino Uno. Tujuan utama dari S4A ialah untuk menarik perhatian masyarakat terhadap dunia pemrograman perangkat keras. Selain itu, bagi yang telah berpengalaman dengan pemrograman Arduino dapat menjadikan S4A sebagai alat bantu yang menyenangkan sambil bereksperimen. Perangkat lunak ini dapat diunduh secara gratis di <http://s4a.cat/>.



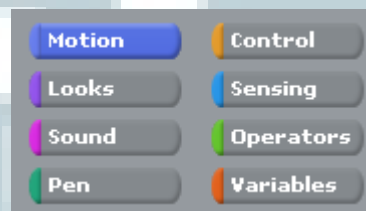
Gambar 3.14. Tampilan S4A

Di S4A, papan Arduino direpresentasikan oleh grafik komputer khusus. Komputer grafik ini secara otomatis menemukan gerbang tempat



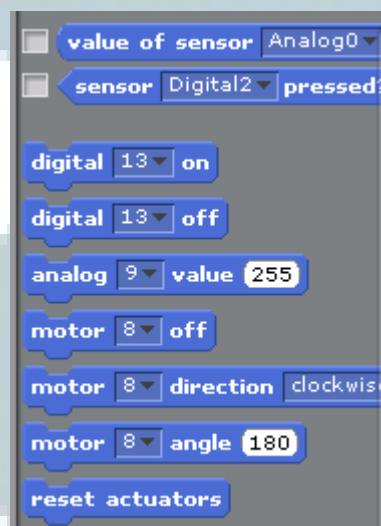
papan dikoneksikan. Bahkan perangkat keras ini mampu membaca banyak papan Arduino cukup dengan menambah komputer grafik papan Arduino di S4A.

Bagian menu seperti gambar 3.15, yang sering digunakan untuk pemrograman ialah *motion*, *control*, *operators* serta *variables*.



Gambar 3.15. Pilihan Menu Blok di S4A

Dibagian *motion* dapat ditemukan blok-blok yang dapat dilihat di gambar 3.16, yang menentukan penggunaan pin-pin di papan Arduino. Pengguna dapat mengatur pin-pin digital serta analog bahkan dapat mengatur sensor motor yang juga tersedia di S4A.



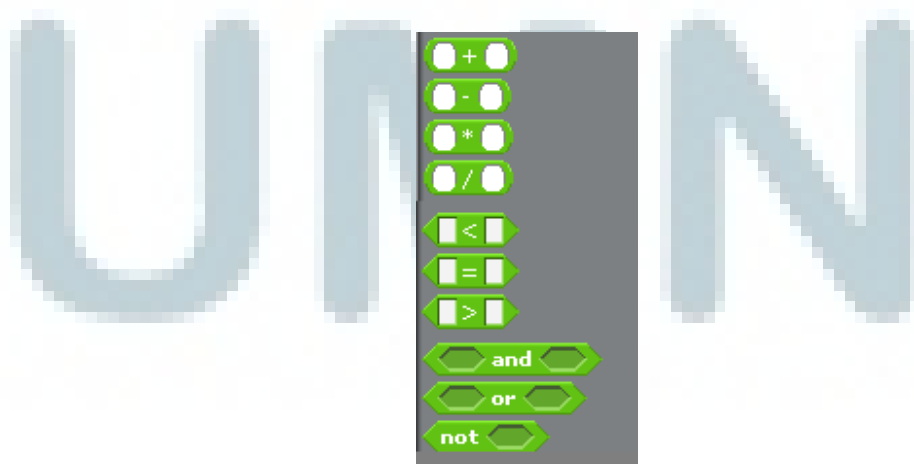
Gambar 3.16. Pilihan Blok di Menu *Motion*

Untuk menu *control* pada gambar 3.17, merupakan blok-blok yang digunakan untuk mengatur program, seperti misalnya terdapat blok *if*, *if-else*, *forever*, *when I received*, serta blok *when green flag clicked* yang digunakan untuk mengatur agar program tidak berjalan sebelum bendera hijau yang terdapat di kanan atas perangkat lunak ditekan.



Gambar 3.17. Pilihan Blok di Menu *Control*

Menu *operators* seperti gambar 3.18, berupa blok untuk melakukan operasi matematika atau melakukan pengecekan jika suatu kondisi sesuai untuk menjalankan fungsi tertentu atau tidak.



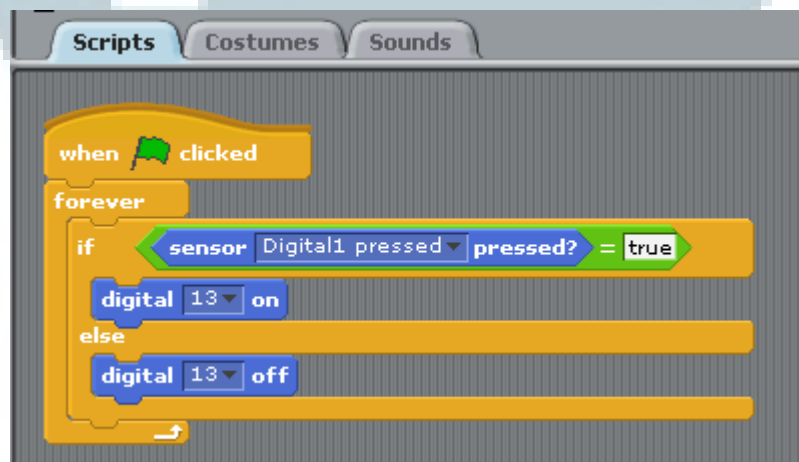
Gambar 3.18. Pilihan Blok di Menu *Operators*

Lalu, menu *variable* pada gambar 3.19 digunakan untuk membuat variable baru yang berguna dalam tahapan pemrograman S4A.



Gambar 3.19. Pilihan Blok di Menu *Variable*

Blok-blok yang digunakan untuk pemrograman akan disatukan dalam *workspace* seperti pada gambar 3.20.



Gambar 3.20. *Workspace* S4A

#### 3.3.1.4. *Workshop Scratch for Arduino*

Pada tanggal 23 Juni dan 24 Juni 2015, penulis menjadi pembicara sekaligus *trainer* bagi para siswa-siswi SMA dalam *workshop*

*Scratch for Arduino. Workshop* ini bertujuan untuk memberikan pengetahuan dasar mengenai pemrograman dengan mikrokontroler dan sensor. Mikrokontroler yang digunakan disini ialah Arduino Uno. Sensor yang digunakan ialah LDR. Selain itu, penulis juga memilih dua peserta yang dianggap cukup baik dalam menjalani proses *workshop* untuk kemudian mendapatkan kesempatan magang di Laboratorium ICT UMN dari 29 Juni 2015 sampai dengan 10 Juli 2015.

Di hari pertama *workshop*, penulis memperkenalkan terlebih dahulu tentang *Scratch* dan bagaimana memakainya. Lalu, peserta mengunduh perangkat lunak S4A serta IDE Android untuk menggunakannya pada *workshop* hari kedua.

Pada hari kedua, peserta menggunakan Android Uno dan mengunggah *firmware* S4A ke Arduino yang mereka pakai. Lalu peserta mulai mencoba menyalakan LED dengan tombol sebagai tahapan pertama. Setelah itu, peserta diperkenalkan dengan sensor cahaya, yaitu LDR dan melakukan pemrograman untuk menyalakan LED ketika nilai dari LDR kurang dari nilai yang sudah ditentukan. Satu jam terakhir, digunakan bagi peserta untuk melakukan proyek kecil untuk melihat sejauh mana peserta menyerap materi dari *workshop* dan untuk menyeleksi mereka yang cocok untuk magang di UMN. Proyek tersebut ialah menyalakan dua LED secara bergantian dengan tombol.

Di akhir *workshop* penulis memilih dua orang untuk mengikuti magang di UMN.

### 3.3.2. Kendala yang Ditemukan

Kendala yang ditemukan penulis selama menangani robot *mindwave* ialah sebagai berikut.

1. Nilai potentiometer dari program tidak sesuai dengan potentiometer yang digunakan. Hal ini menghambat gerakan servo.
2. Kabel *ground* di dalam konektor baterai 9V terputus yang mengakibatkan robot tidak menyala.
3. Ketika fungsi *blink* ditambahkan, robot hanya dapat bergerak sesuai dengan *blink* saja. Fungsi maju dengan menggunakan parameter konsentrasi tidak berfungsi, hanya RGB LED saja yang berubah.

Lalu, untuk *workshop* S4A, penulis menemukan kendala sebagai berikut.

1. Alat-alat yang diperlukan seperti Arduino yang hanya 5 dibandingkan dengan jumlah peserta yang diinfokan ke penulis sebanyak 13 orang, meskipun saat *workshop*, peserta yang hadir sebanyak 8 orang. Selain itu, tombol serta sensor tidak ada.
2. Selama *workshop* berlangsung, ada tiga Arduino yang ternyata tidak dapat dipakai. Dua Arduino mengalami kerusakan sehingga *firmware* S4A tidak dapat diunggah. Satu Arduino

tidak terbaca oleh komputer grafik S4A sebagai Arduino melainkan sebagai *mindwave*.

3. Adanya peserta yang tidak datang di *workshop* hari pertama, tapi datang di hari kedua. Hal ini menyebabkan peserta tidak sempat mengenal *Scratch* dan langsung diarahkan ke S4A.
4. Waktu yang terbatas untuk mempersiapkan *workshop*. Penulis baru mengetahui bahwa *workshop* akan dilaksanakan pada 23-24 Juni 2015, satu hari sebelum acara sehingga banyak alat-alat yang kurang. Waktu yang terbatas di hari pertama menyebabkan peserta tidak bisa benar-benar mengeksplor *Scratch* secara menyeluruh dan peserta tidak sempat menyentuh S4A.
5. Sebagian besar peserta masih awam dengan Arduino dan logika pemrograman sehingga proyek akhir tidak ada peserta yang selesai sampai mengunggah program ke Arduino.

### **3.3.3. Solusi atas Kendala yang Ditemukan**

Dari kendala yang ditemukan, penulis melakukan solusi untuk robot *mindwave* sebagai berikut.

1. Penulis mengecek nilai minimum dan maksimum potentiometer yang dibaca Arduino, dan mengganti nilainya sesuai dengan data yang ada.

2. Penulis mengganti konektor baterai 9V tersebut dengan yang baru.
3. Untuk fungsi *blink*, penulis membaca data *raw* yang dikirim *mindwave* dalam kondisi normal dan membandingkannya dengan kondisi ketika *blink* terjadi. Lalu, penulis memperbaiki peletakan kondisi fungsi dalam program yang sebelumnya bertabrakan dengan pembacaan data *raw* untuk fungsi maju menjadi berada di dalam klasifikasi data *raw* yang khusus untuk pembacaan paket data *small*.

Sementara itu, untuk kendala selama *workshop* S4A, penulis melakukan solusi sebagai berikut.

1. Dikarenakan jumlah Arduino yang terbatas, penulis memutuskan untuk membagi peserta kedalam kelompok tiga orang. Untuk tombol, penulis mengambil tombol dari papan sirkuit sistem digital yang sudah rusak tapi tombol masih berfungsi. Sementara untuk sensor, penulis beli di sore hari, hari pertama *workshop* agar dapat langsung digunakan di hari kedua *workshop*.
2. Meminjam Arduino yang digunakan untuk skripsi rekan magang penulis agar peserta tetap dapat mengeksplor Arduino.
3. Peserta yang baru datang tersebut langsung bergabung dengan kelompok peserta lain, tidak membuat kelompok sendiri karena kurangnya alat.

4. Waktu yang terbatas selama persiapan dapat penulis gunakan sebaik-baiknya untuk merangkai materi *workshop* dan mengumpulkan alat-alat yang dibutuhkan selama *workshop*. Untuk hari pertama, penulis mengambil solusi agar cukup memperkenalkan dasar dari *Scratch* saja baru di hari kedua peserta dapat mengeksplor S4A secara menyeluruh.
5. Dikarenakan sebagian besar peserta masih asing dengan logika pemrograman, penulis memberikan petunjuk kepada peserta tentang bagaimana cara menyelesaikan proyek tersebut. Ketika waktu habis dan tidak ada peserta yang bisa menyelesaikan proyek tersebut, penulis memutuskan untuk melihat logika pemrograman masing-masing kelompok dan memberitahukan logika yang benar kepada mereka.

UMMN