



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

Dalam tugas akhir ini metode penelitian yang digunakan adalah metode penelitian kuantitatif. Untuk memperoleh data-informasi akan dilakukan *sampling* terhadap Mahasiswa Sistem Komputer Universitas Multimedia Nusantara (UMN).

3.1 Alat dan Bahan

Alat dan bahan yang dibutuhkan dalam penelitian kali ini adalah :

- Sebuah *headset* Neurosky Mindwave – alat pendeteksi gelombang otak, pemrosesan sinyal gelombang otak, dan pengiriman sinyal melalui *Radio Frequency* (RF)
- Sebuah PC dengan sistem operasi Windows 7 dan telah di-*install* Microsoft Visual Studio (untuk pembuatan aplikasi).
- Ruang terisolasi dalam bentuk ruangan maupun mobil.
- Video sebagai penciptaan suasana konsentrasi otak.

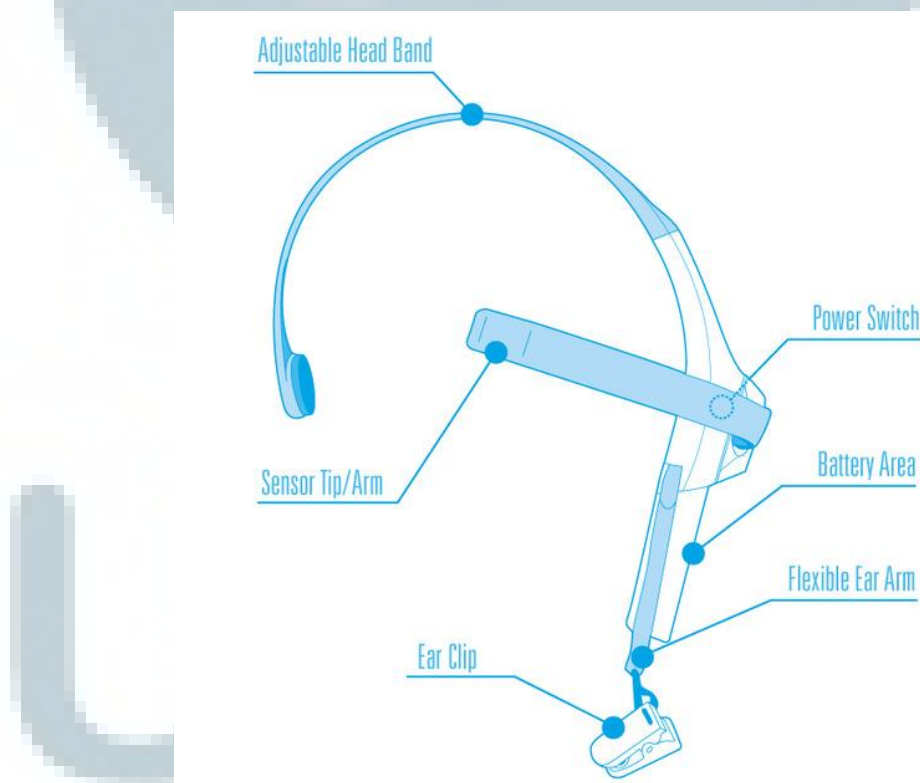
3.2 Metode Penelitian

Aplikasi *DrowTion* yang dibuat mengambil data pola sinyal otak dari *headset* Mindwave. Data yang didapatkan tersebut menunjukkan proses aktivitas otak dan dapat diidentifikasi kondisinya. Perancangan aplikasi *DrowTion* untuk mendeteksi tingkat kantuk pengguna dimulai dari studi literatur, pengambilan data penelitian, analisis data, perancangan algoritma,

dan perancangan aplikasi. Setelah aplikasi dibuat maka tahap berikutnya adalah pengujian dan pengembangan aplikasi untuk optimalisasi.

3.2.1 Struktur dan Cara Kerja Neurosky Mindwave

Neurosky Mindwave terdiri dari beberapa bagian utama yaitu elektroda, *reference point*, *contact*, dan *Thinkgear Chip*. Pasangan elektroda dan *reference point* adalah bagian peralatan yang mendeteksi adanya sinyal gelombang otak dimana elektroda menerima sinyal positif sedangkan *reference point* sebagai *ground*. Berikut adalah gambar struktur dari Neurosky Mindwave.



Gambar 3.1 Struktur Neurosky Mindwave (Neurosky, 2009)

Elektroda dalam gambar tersebut terletak pada bagian *sensor tip*. Sedangkan *reference point* yang dimaksudkan berada di dalam *ear clip*. *Thinkgear Chip* dan *contact* berada di bagian dalam *power switch*. Mindwave beroperasi menggunakan sebuah baterai berukuran AA.

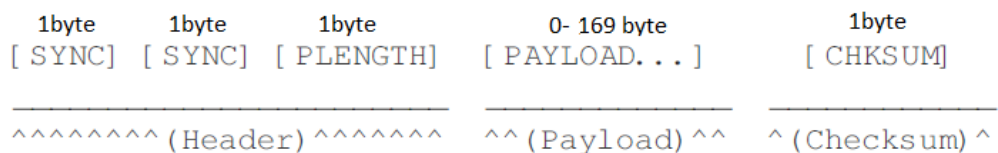
Elektroda disini berfungsi untuk mengukur fluktuasi besaran aliran listrik pada sel saraf bagian *cortex* seseorang. Melalui elektroda ini, dapat diukur besaran aliran listrik akibat ion-ion di dalam otak seseorang yang berada dalam rentang 10-100 mikrovolt. Elektroda pada alat ini diletakkan pada bagian kiri dahi seseorang. Sedangkan, *reference point* diletakkan pada bagian telinga pengguna.

Data yang didapatkan dari elektroda berbentuk *raw signal*. Data tersebut kemudian diproses dalam *Thinkgear Chip* untuk dipisahkan menjadi *delta* (0.5 - 2.75Hz), *theta* (3.5 -6.75Hz), *low-alpha* (7.5 - 9.25Hz), *high-alpha* (10 - 11.75Hz), *low-beta* (13 - 16.75Hz), *high-beta*(18 - 29.75Hz), *low-gamma* (31 - 39.75Hz), dan *mid-gamma* (41 - 49.75Hz). Pemisahan data tersebut menggunakan konsep FFT sehingga data dikonversi ke dalam *domain* frekuensi dan dipisahkan berdasarkan frekuensinya menghasilkan *alpha*, *beta*, *gamma*, dsb.

Pada *Thinkgear Chip* terdapat fitur tambahan yang disebut *e-sense*. *E-sense* yang merupakan fitur tambahan dari Mindwave, diproses menggunakan algoritma tertentu yang ditanamkan di dalam *chip* tersebut untuk memproses data dari gelombang yang telah dipisahkan tersebut

menjadi pendeteksi tingkat konsentrasi, *blink strength*, dan meditasi penggunanya.

Hasil konversi yang dilakukan dalam *chip* tersebut kemudian ditransmisikan menggunakan *Radio Frequency* (RF) menuju ke *RF receiver* (*dongle*) yang terhubung dengan *USB port* pada komputer. Data yang ditransmisikan tersebut berbentuk serial dan ditransmisikan secara *asynchronous*. Setiap paket yang dikirimkan tersebut memiliki struktur yang terdiri dari *header*, *payload*, dan *checksum*. *Header* terdiri dari 3 *byte* data yaitu 2 *byte sync* dan 1 *byte plength*. *Payload* dapat berisi hingga 169 *byte* data dan *checksum* 1 *byte* data. Dengan demikian besar ukuran paket berkisar antara 4 *byte* (bila tidak ada *payload*) hingga 173 *byte* (bila *payload* berisi 169 *byte*).



Gambar 3.2 Struktur Data yang ditransmisikan Mindwave (Neurosky, 2009)

Pembacaan paket dapat dilakukan dengan dua cara yaitu cara manual dengan pembacaan digit per digit maupun penggunaan *Thinkgear Library* (Neurosky,2009).

3.2.2 Thinkgear Library

Sinyal gelombang otak yang diperoleh selanjutnya diterjemahkan menggunakan *Thinkgear Library* yang disediakan oleh Neurosky. Dalam

library ini disediakan beberapa fungsi utama yang dapat digunakan dalam pembangunan aplikasi seperti fungsi *connect*, *disconnect*, maupun fungsi untuk mengambil nilai dari variabel seperti nilai sinyal *theta*, *alpha*, dsb.

Melalui fungsi *connect* yang disediakan, diperlukan masukkan berupa tiga parameter yaitu parameter *port* yang digunakan, *baud rate*, dan jenis komunikasi (*synchronous* / *asynchronous*). Perbedaan antara *synchronous* dan *asynchronous* adalah pada penggunaan sinyal *clock* dimana dalam *synchronous* digunakan sinyal ini untuk menandakan transmisi data sedangkan pada *asynchronous* tidak ada sinyal *clock* tersebut yang berakibat data tidak ditransmisikan secara teratur. *Baud rate* yang digunakan adalah 57.600. Secara *default* komunikasi antara alat dan komputer adalah *asynchronous*. Sedangkan, fungsi *disconnect* akan menutup konektivitas antara alat dengan komputer sehingga aplikasi lain dapat menggunakannya kembali.

Fungsi lain yang disediakan oleh *library* ini adalah untuk mengetahui kualitas sinyal. Kualitas sinyal digunakan untuk mengetahui apakah peletakkan elektroda sudah tepat atau belum untuk menangkap sinyal secara lebih akurat. Efek samping dari peletakkan elektroda yang tidak tepat adalah waktu penangkapan sinyal menjadi lebih lambat dan berakibat pada penurunan performa pada aplikasi dalam pendeteksian.

Nilai *low-alpha* dapat diambil dengan memanggil object `thinkGearWrapper.ThinkGearState.alpha1` begitu pula dengan *high-alpha*

dipanggil dengan *alpha2*. Sedangkan *beta*, *gamma*, dan lainnya juga dapat dipanggil dengan cara demikian. Nilai dari fokus dapat dipanggil dengan memanggil properti *attention*, meditasi dengan *meditation*, dan *blink strength* dengan *blinkstrength*. Properti-properti itu dapat dipanggil tanpa perlu diketahui letaknya di dalam paket yang diterima karena *library* ini telah melakukan konversi ke dalam suatu variabel yang dapat diakses (Neurosky, 2009). Berikut ditampilkan tabel variabel yang dapat dipanggil untuk mendapatkan informasi mengenai kondisi terakhir yang didapatkan oleh elektroda.

Tabel 3.1 Tabel Nilai pada Library Thinkgear (Neurosky, 2011).

Nilai yang Diambil	Cara Memanggil
<i>Raw Signal</i>	<code>thinkGearWrapper.ThinkGearState.Raw</code>
<i>Delta</i> (0,5 – 2,75 Hz)	<code>thinkGearWrapper.ThinkGearState.Delta</code>
<i>Theta</i> (3,5 – 6,75 Hz)	<code>thinkGearWrapper.ThinkGearState.Theta</code>
<i>Low-alpha</i> (7,5 – 9,25 Hz)	<code>thinkGearWrapper.ThinkGearState.Alpha1</code>
<i>High-alpha</i> (10-11,75 Hz)	<code>thinkGearWrapper.ThinkGearState.Alpha2</code>

Nilai yang Diambil	Cara Memanggil
<i>Low-beta</i> (13 – 16,75 Hz)	thinkGearWrapper.ThinkGearState.Beta1
<i>High-beta</i> (18 – 29,75 Hz)	thinkGearWrapper.ThinkGearState.Beta2
<i>Low-gamma</i> (31 – 39,75 Hz)	thinkGearWrapper.ThinkGearState.Gamma1
<i>Mid-gamma</i> (41 – 41,75 Hz)	thinkGearWrapper.ThinkGearState.Gamma2
<i>Signal Quality</i> (<i>Poor Signal</i>)	thinkGearWrapper.ThinkGearState.PoorSignal
Fokus (e-Sense)	thinkGearWrapper.ThinkGearState.Attention
<i>Blink Strength</i>	thinkGearWrapper.ThinkGearState.BlinkStrength
Meditasi (e-Sense)	thinkGearWrapper.ThinkGearState.Meditation
<i>Battery</i>	thinkGearWrapper.ThinkGearState.Battery
<i>Error</i> (boolean value)	thinkGearWrapper.ThinkGearState.Error
Jumlah paket	thinkGearWrapper.ThinkGearState.PacketsRead
Versi	thinkGearWrapper.ThinkGearState.Version

3.2.3 Konektivitas Mindwave

Neurosky Mindwave sebagai penerima gelombang otak mentransmisikan data menggunakan *Radio Frequency* maupun *bluetooth* bergantung pada versi Mindwave yang digunakan. Versi standar menggunakan *Radio Frequency* (RF) sedangkan versi *mobile* menggunakan media transmisi *bluetooth*. Transmisi pada kedua versi tersebut sama-sama menggunakan *Universal Asynchronous Receiver Transmitter* (UART). UART digunakan untuk mentransalasi data antara paralel dengan serial.

Data yang diterima pada komputer telah melalui UART menggunakan *Serial Communication Protocol*. Untuk menghubungkan Mindwave dengan komputer diperlukan *Thinkgear Connector* yang merupakan *driver* bagi Mindwave untuk terhubung dengan komputer. Bila sudah dapat terhubung dengan komputer, diperlukan konektivitas dengan aplikasi DrowTion yang akan dirancang. Untuk komunikasi tersebut digunakan bantuan dari *Thinkgear Library* yang mentranslasi dan menyediakan metode untuk konektivitas aplikasi dengan Mindwave.

Metode untuk menghubungkan aplikasi tersebut memerlukan deklarasi awal terhadap beberapa nilai antara lain: *port number*, *baud rate*, dan mode sinkronisasi. *Port number* tersebut dapat dilihat secara manual dari *device manager* komputer sehingga aplikasi akan membangun konektivitas melalui *port* tersebut. Sedangkan *baud rate* yang merupakan penentu kecepatan aliran data dipilih 57.600 bps (bit per second). Hal ini

berpedoman pada spesifikasi dari Mindwave yang diberikan. Sedangkan mode sinkronisasi yang digunakan adalah *asynchronous*. (Neurosky, 2009).

3.2.4 Pengambilan Data Penelitian

Dalam tahap ini dilakukan pengambilan data dengan berbagai skenario yang telah dibuat seperti pengambilan data di posisi yang berbeda, kondisi berbeda seperti mengantuk atau segar, kondisi fokus atau rileks, dan sebagainya. Skenario yang dibuat dan diambil datanya dapat dilihat pada tabel berikut:

Tabel 3.2 Tabel Skenario

No.	Posisi	Kondisi
1	Duduk	Tidak Mengantuk
2.	Duduk	Mengantuk
3.	Berbaring	Tidak Mengantuk
4.	Berbaring	Mengantuk
5.	Menyetir	Tidak Mengantuk
6.	Menyetir	Mengantuk

Data-data yang didapatkan tersebut kemudian dimasukkan ke dalam suatu file berformat csv. File berformat csv ini kemudian di-*plot*-kan ke dalam bentuk grafik untuk melihat karakteristik yang terjadi di dalamnya.

3.2.5 Analisis Data

3.2.5.1 Visualisasi Data

Data yang telah divisualisasikan dalam bentuk grafik dibandingkan satu sama lainnya. Dengan membandingkan grafik satu dengan lainnya dapat dilihat pengaruh posisi dan kondisi terhadap hasil keluaran dari Mindwave. Dari perbandingan tersebut ditarik kesimpulan sesuai dengan skenario dan hasil *plot* grafik tersebut untuk menentukan seseorang dalam kondisi mengantuk atau tidak dengan melihat nilai-nilai yang terpengaruh dengan perbedaan kondisi tersebut antara lain *alpha* dan *theta*.

3.2.5.2 Perancangan Algoritma untuk Analisis Kondisi Konsentrasi

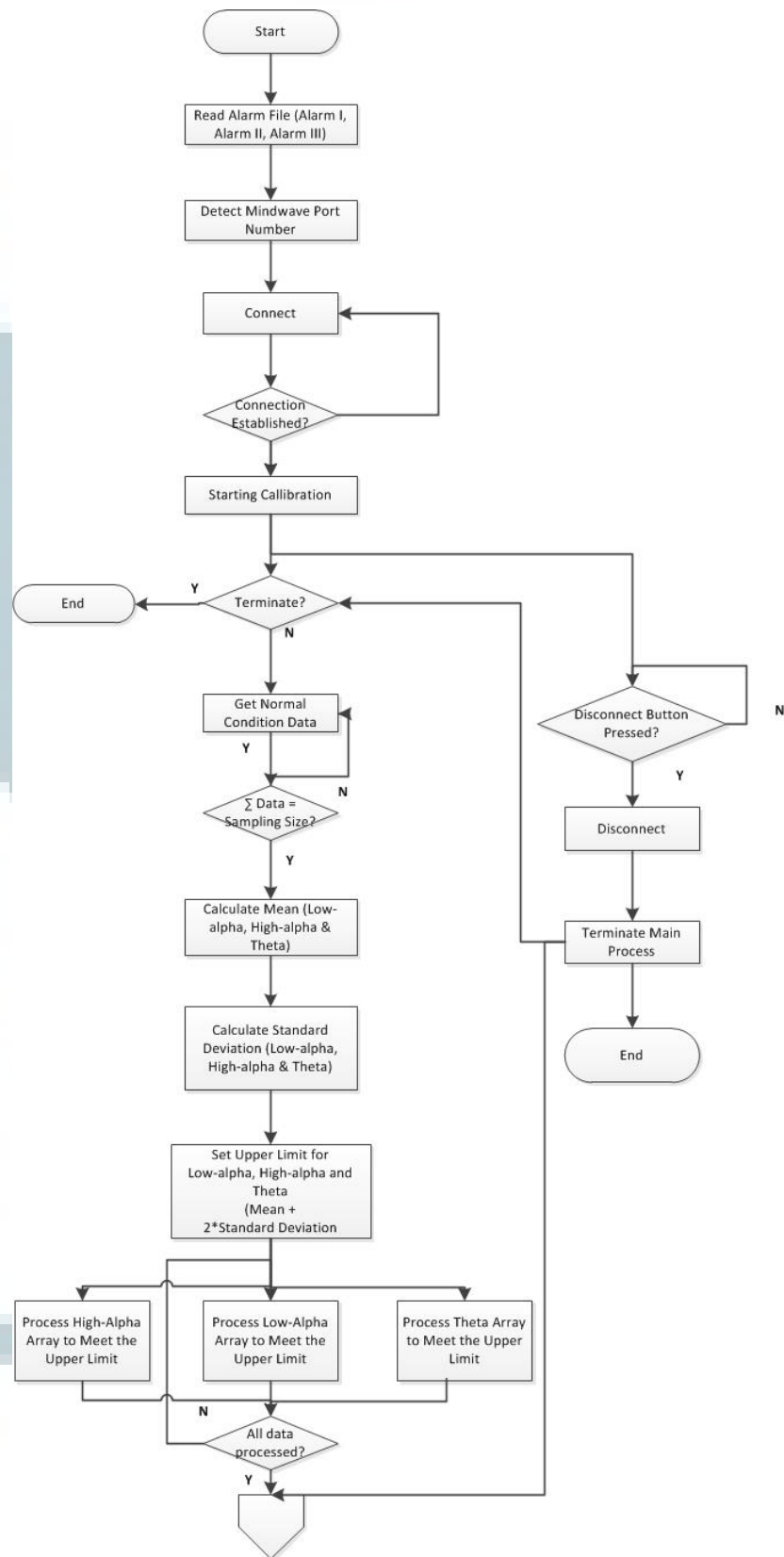
Dalam perancangan algoritma digunakan data hasil analisis. Dengan pembuatan suatu algoritma yang efisien dengan tingkat akurasi tinggi diperlukan suatu data sampel yang didapatkan sebagai pembanding. Dari data yang telah dianalisis didapatkan suatu karakteristik dimana dengan berbagai posisi *headset* maka nilai yang didapatkan pun berbeda. Hal ini tentu menjadi masalah walau dengan kualitas sinyal 100% tetapi berbagai posisi dapat berubah nilainya. Dengan berbagai simulasi yang telah dilakukan sebelumnya terlihat bahwa ketika *headset* ditempelkan lebih dekat maka akan terlihat penambahan nilai secara umum. Untuk itu, tidak dapat dilakukan dengan memberi suatu *trigger* pada nilai tertentu yang didefinisikan

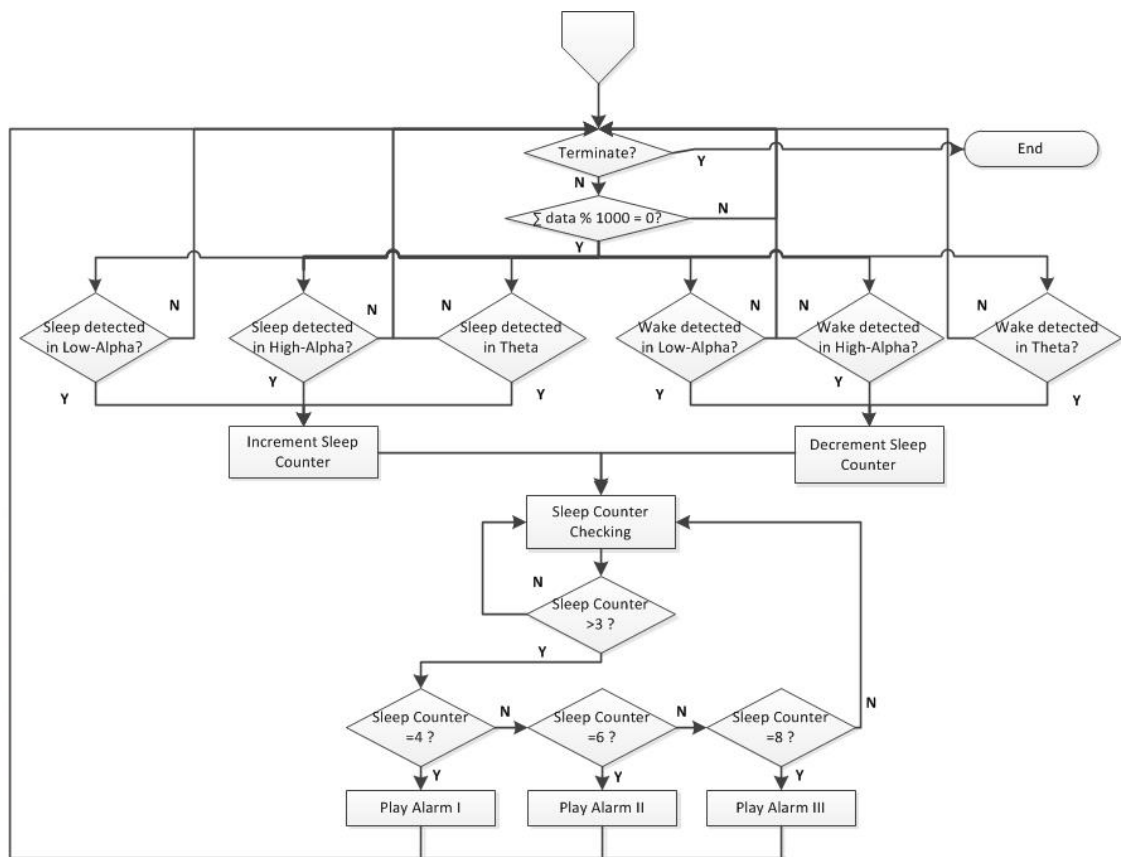
ecara statis. Dengan beberapa kali percobaan saat dilepas dan mengenakan kembali dengan kondisi yang hampir serupa dapat terlihat perubahan data yang signifikan.

Dengan demikian diperlukan suatu prosedur untuk kalibrasi setiap kali aplikasi pertama kali dinyalakan atau ketika pengguna berganti untuk menghasilkan keluaran yang tepat. Kalibrasi yang dilakukan hanyalah untuk mendapatkan data sampel saja tidak perlu pengguna berkonsentrasi atau bagaimana hanya perlu melakukan hal normal ketika menyetir. Nilai tersebut kemudian akan ditampung dan digunakan sebagai pembanding dengan perubahan kondisi pengguna. Bila nilai dari *Alpha* pengguna secara rata-rata menurun jika dibandingkan dengan nilai sampel maka akan menambahkan nilai dari *sleep counter*. Begitu pula dengan nilai *Theta*.

UMMN

Flow chart aplikasi dapat dilihat pada gambar berikut:





Gambar 3.3 Flow Chart Rancangan Aplikasi

Perancangan aplikasi dimulai dengan menghubungkan aplikasi dengan Mindwave sehingga komunikasi antara aplikasi dan Mindwave dapat terjalin. Setelah komunikasi dapat terjalin maka dapat dibuat prosedur untuk *listen* terhadap Mindwave sehingga data dari Mindwave dapat masuk dan disimpan di dalam *memory*. Untuk memperoleh data penelitian, aplikasi penerima dirancang untuk menuliskan *log file* berformat csv. Implementasi algoritma ke dalam aplikasi kemudian dilakukan untuk melakukan verifikasi terhadap hasil penelitian awal.

3.3 Aplikasi DrowTion

3.3.1 Pengujian Aplikasi

Dalam tahap ini aplikasi diuji dengan menggunakan berbagai skenario yang telah dibuat. Hal ini bertujuan untuk mengetahui apakah algoritma yang telah dibuat sudah efektif dalam mendeteksi pengguna yang mengantuk atau belum. Hal lain yang diuji adalah waktu pendeteksian. Sehingga melalui tahap ini diharapkan dapat mengetahui performa aplikasi dan didapatkan *trade-off* yang tepat antara akurasi dengan kecepatan pendeteksian dikarenakan akurasi dan kecepatan tidak dapat diraih keduanya hanya dapat dipilih efektivitasnya.

3.3.2 Optimalisasi Aplikasi

Diperoleh hasil tes tentang bagaimana performa aplikasi dan tingkat akurasi yang didapat. Dari data tersebut dilakukan revisi terhadap aplikasi untuk meningkatkan akurasi dan juga kecepatannya. Di samping itu dicari suatu solusi alternatif yang dapat memberikan *trade-off* yang baik antara kecepatan dan akurasi. Solusi tersebut kemudian akan diimplementasikan untuk mengembangkan aplikasi sehingga aplikasi dapat memberikan performa dan pengalaman lebih baik pada pengguna. Setelah revisi dilakukan, maka tahap berikutnya adalah kembali melakukan pengujian untuk mengetahui apakah perubahan memberikan hasil positif atau tidak dan apakah masih bisa ditingkatkan performanya.