



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

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalankan program kerja magang di *Tokyo Denki University*, yang menjadi tugas utama adalah bergabung dengan salah satu laboratorium penelitian di *Tokyo Denki University Chiba New Town Campus* untuk mengerjakan salah satu proyek penelitian yang sedang dilakukan oleh anggota lainnya dari laboratorium tersebut dengan bimbingan dari profesor pembimbing laboratorium. Dalam kesempatan ini, laboratorium tersebut adalah laboratorium Interaksi Manusia dan Komputer di bawah bimbingan Profesor Naoki Mukawa. Selain itu, selama melakukan riset, diwajibkan untuk mengikuti kelas yang terkait seperti mahasiswa *Tokyo Denki University* lainnya.

3.2 Tugas yang Dilakukan

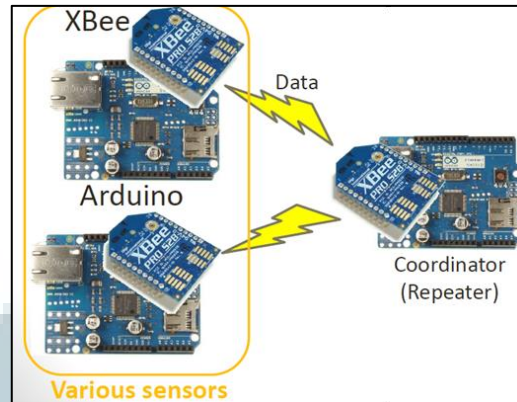
Tugas utama selama program kerja magang ini adalah mengembangkan rancangan aplikasi “TOMOLIVI”, yang merupakan proyek dari Murakami Miyu, salah seorang mahasiswa *Tokyo Denki University* yang tergabung dalam laboratorium Interaksi Manusia dan Komputer. Dalam mengembangkan aplikasi “TOMOLIVI” ini sendiri, yang menjadi tugas spesifik adalah untuk mengembangkan bagian *front-end* dari aplikasi, terutama tampilan *user interface* aplikasi tersebut. Namun, karena aplikasi beroperasi secara dua arah dan *real-time*, sebelum mengembangkan *user interface* yang sesuai, terlebih dahulu diperlukan pengembangan aplikasi untuk menyesuaikan data pada *client* dengan data pada

FTP (File Transfer Protocol) Server yang digunakan oleh tim sebagai media penyimpanan dan perantara antar komputer *user* secara *real-time*.

3.3 Uraian Pelaksanaan Kerja Magang

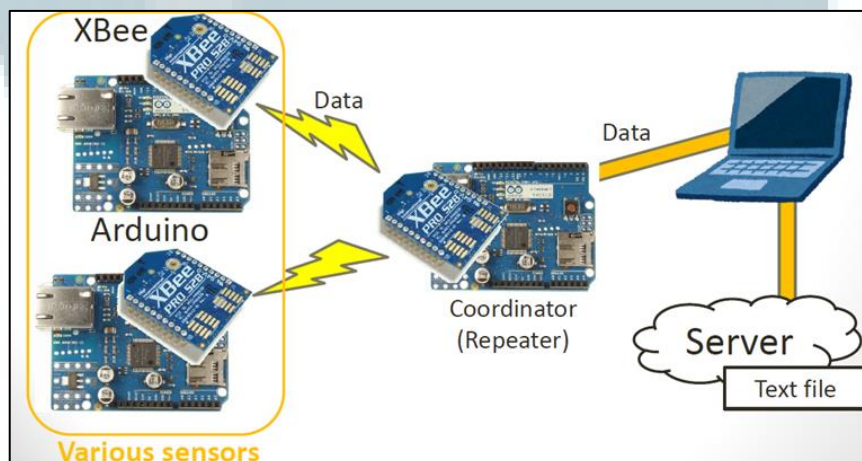
3.3.1 Proses Pelaksanaan

Selama proses awal pengembangan aplikasi, dikarenakan aplikasi belum memadai dan data belum dapat ditransfer antar *client*, seluruh anggota tim memfokuskan diri kepada proses pengembangan *software* sensor dan transmisi data. Dalam aplikasi “TOMOLIVI” ini, terdapat tiga macam sensor yang digunakan untuk mendeteksi kondisi ruangan dan mendapatkan data yang dibutuhkan, antara lain *Microsoft Kinect* untuk deteksi posisi dan *Arduino* untuk deteksi cahaya dan tekanan. Pemrograman untuk *software Microsoft Kinect* menggunakan bahasa pemrograman C++ dengan *Microsoft Visual Studio 2010* sebagai *development tool*, sedangkan pemrograman *Arduino* menggunakan bahasa pemrograman *Arduino*. Selain itu, di tengah proses pengembangan, karena koneksi *wired* antara masing-masing sensor dirasa cukup merepotkan bagi *user*, tim memutuskan untuk mengimplementasikan perangkat *Xbee*, yang bertujuan untuk memungkinkan koneksi *wireless* antara masing-masing sensor dengan *coordinator* pada mesin komputer *client*.



Gambar 3.1 Ilustrasi Penerapan Perangkat *Xbee* pada Perangkat Sensor

Setelah data dari masing-masing sensor berhasil didapatkan, data-data tersebut disimpan dalam bentuk *text file* yang dianggap paling mudah dan fleksibel sebagai format data media perantara, kemudian disimpan di dalam FTP *Server* yang disewa untuk proyek ini.



Gambar 3.2 Ilustrasi Bagian *Back End* Sistem “TOMOLIVI”

Setelah data dari semua sensor telah tersimpan di dalam *FTP server*, barulah bagian *front-end* aplikasi dapat dikembangkan. Agar data yang ditampilkan bersifat *real-time*, *client* diharuskan untuk mengakses *FTP Server* dan membaca *text file* setiap saat, tetapi aplikasi UNITY3D yang digunakan untuk mengembangkan aplikasi ini memiliki keterbatasan dalam mengakses protokol FTP. Oleh karena hal tersebut, sebelum mengembangkan *user interface*,

dikembangkanlah sebuah aplikasi terpisah yang bertujuan untuk melakukan *login* ke dalam *FTP Server* dan mengunduh seluruh *text file* yang dibutuhkan untuk kemudian disimpan ke dalam komputer *client* secara berkala dan terus menerus. Setelah berhasil, barulah bagian *interface* dapat dikembangkan sepenuhnya.

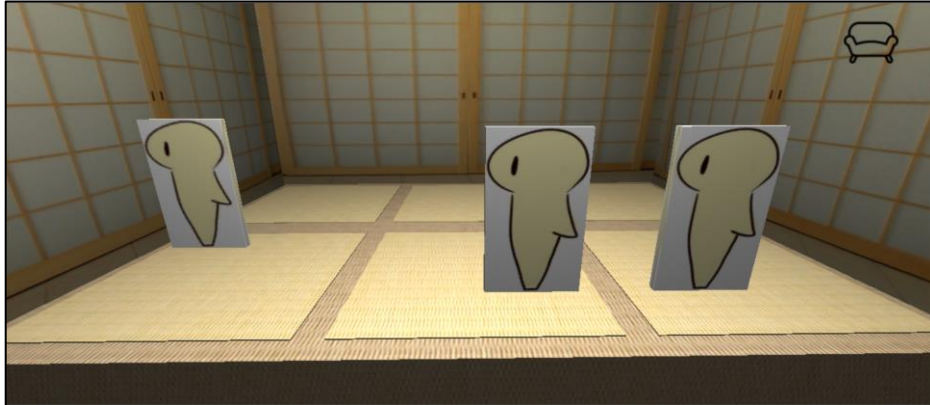
Sebelumnya, aplikasi “TOMOLIVI” sebenarnya sudah memiliki *browser based user interface* yang dikembangkan menggunakan bahasa pemrograman PHP dan HTML.



Gambar 3.3 Tampilan *User Interface* Terdahulu Aplikasi “TOMOLIVI”

Namun, untuk sebuah aplikasi yang bersifat dinamik dan beroperasi secara *real-time*, *user interface* menggunakan HTML dan PHP tersebut dirasa kurang memadai dan tidak dapat menampilkan informasi yang cukup, terutama informasi mengenai posisi orang yang terdeteksi. Oleh karena itu, tim setuju untuk memilih menggunakan UNITY3D karena mampu menampilkan informasi secara dinamik dan mendetail melalui tampilan tiga dimensi.

A. Layout Program



Gambar 3.4 Tampilan *User Interface* Utama Aplikasi “TOMOLIVI”

Beberapa hal yang perlu diperhatikan perihal *user interface* tiga dimensi dari aplikasi “TOMOLIVI” ini, antara lain sebagai berikut.

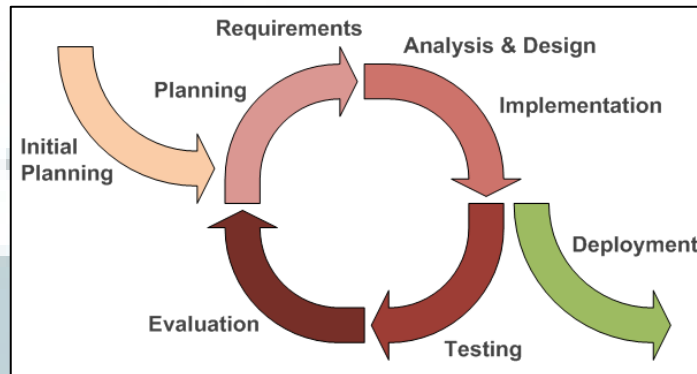
- a) *Sprite* manusia menandakan orang yang terdeteksi di dalam ruangan beserta dengan posisinya.
- b) *Icon* sofa yang terletak di bagian kanan atas *user interface* menandakan ada yang sedang menduduki kursi di dalam ruangan subjek.
- c) *User interface* bersifat dinamik dan *real-time* sehingga tampilan terus mengalami perubahan apabila terjadi perubahan data pada *server*.

B. Perancangan Sistem

B.1 Metode Pengembangan

Pengembangan dilakukan melalui metodologi *agile*, dimana seluruh proses pengembangan segera dilakukan setelah fase *planning* selesai, kemudian hasil dari fase *development* tersebut langsung melalui fase *testing* dan *evaluation*. Apabila dirasa sudah memuaskan dan memenuhi kriteria, tim akan berpindah untuk melakukan bagian pengembangan lainnya, tetapi apabila dirasa belum memenuhi kriteria, tim akan mengulang proses tersebut hingga memenuhinya. Dengan

demikian, proses pengembangan akan mudah beradaptasi terhadap perubahan yang dirasa dibutuhkan.



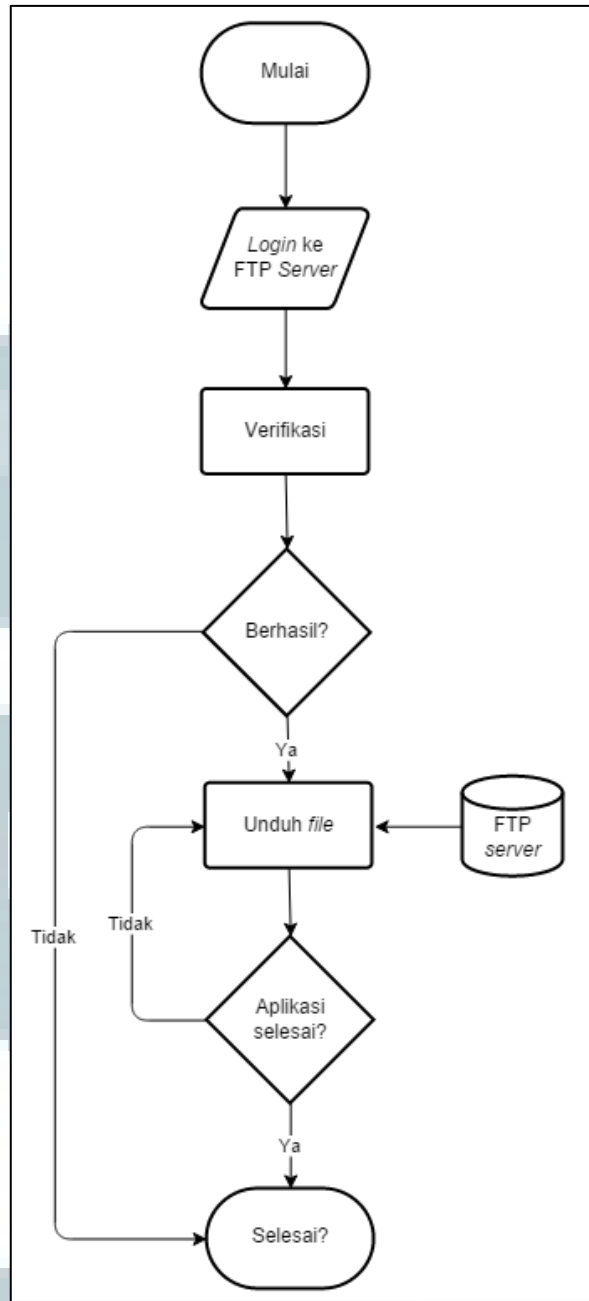
Gambar 3.5 Ilustrasi Metodologi Agile

(Sumber : Sheehan, Lex. 2013)

B.2 Flowchart

Berikut adalah *flowchart* dari aplikasi C# terpisah yang digunakan untuk membantu aplikasi utama untuk secara periodik dan terus-menerus melakukan *login* dan mengunduh *file* yang dibutuhkan dari FTP Server.

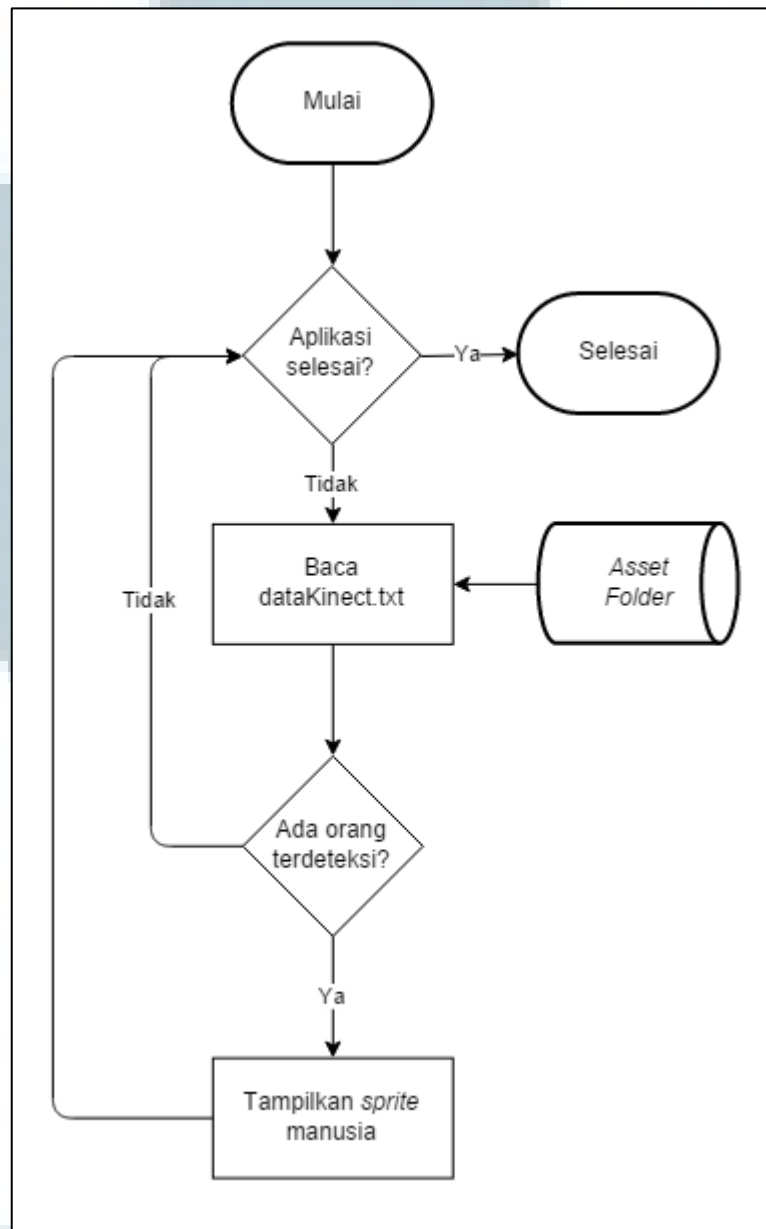
U
M
M
N



Gambar 3.6 Flowchart Aplikasi Auto Login dan Auto Download

Ketika dijalankan, aplikasi ini akan mencoba melakukan *login* ke dalam FTP Server yang digunakan untuk menyimpan data sensor menggunakan *username* dan *password* yang telah ditentukan dan digunakan oleh seluruh tim pengembang. Apabila aplikasi gagal mengakses FTP Server, aplikasi akan langsung menghentikan operasinya, tetapi apabila aplikasi berhasil mengakses FTP Server, aplikasi akan secara otomatis mengunduh ketiga *text file* yang

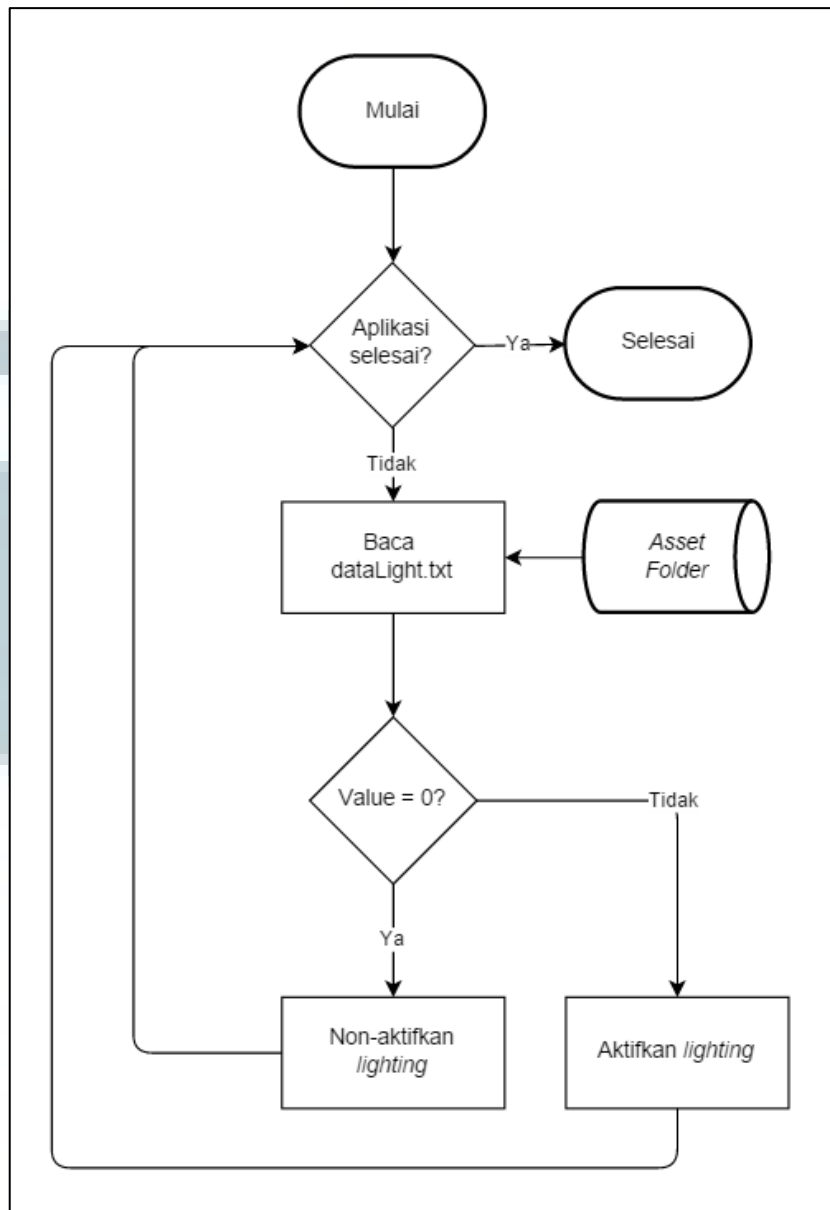
diidentifikasi melalui nama *file* tersebut, yaitu dataKinect.txt, dataLight.txt, dan dataPressure.txt. Aplikasi akan terus mengulang operasi mengunduh tersebut hingga aplikasi ini dinon-aktifkan.



Gambar 3.7 Flowchart Aplikasi User Interface Utama (dataKinect.txt)

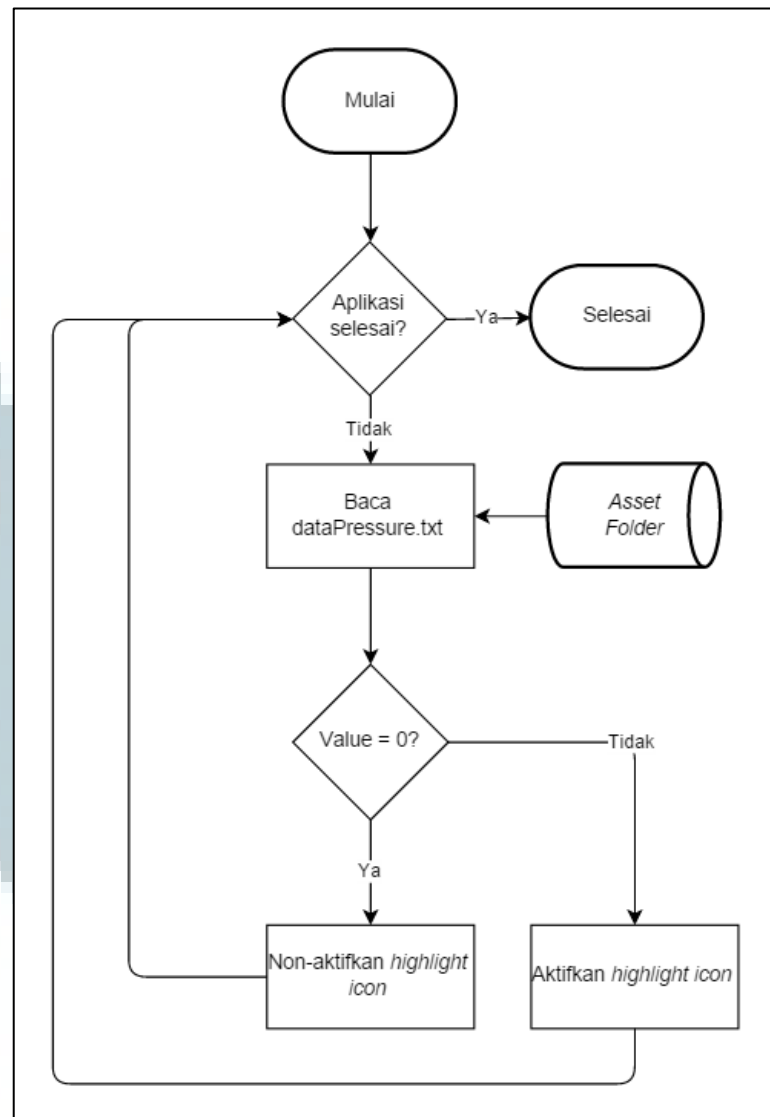
Aplikasi membaca *file* dataKinect.txt yang terletak pada *directory* aset aplikasi. Untuk setiap orang yang terdeteksi dan tercatat pada *file* dataKinect.txt, aplikasi akan menampilkan *sprite* manusia pada tampilan *user interface* aplikasi

sesuai dengan koordinat yang tercatat. Proses tersebut akan terus diulang hingga aplikasi dimatikan oleh *user*.



Gambar 3.8 *Flowchart* Aplikasi *User Interface* Utama (*dataLight.txt*)

Aplikasi membaca *file* *dataLight.txt* yang terletak pada *directory* aset aplikasi. Apabila isi dari *file* *dataLight.txt* bernilai nol, aplikasi akan menonaktifkan *lighting* pada tampilan *user interface*, sedangkan apabila bernilai satu, aplikasi akan mengaktifkan *lighting* tersebut. Proses tersebut akan terus diulang hingga aplikasi dimatikan oleh *user*.

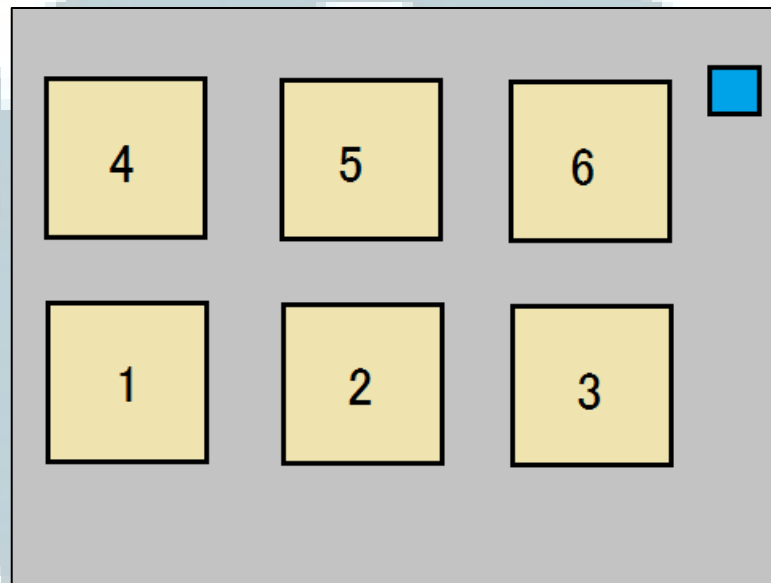


Gambar 3.9 Flowchart Aplikasi User Interface Utama (dataPressure.txt)

Aplikasi membaca *file* dataPressure.txt yang terdapat pada *directory* aset aplikasi. Apabila isi dari *file* dataPressure.txt bernilai nol, aplikasi akan menonaktifkan *highlight* pada *icon* kursi yang terletak pada bagian kanan atas tampilan *user interface*, sedangkan apabila bernilai satu, aplikasi akan mengaktifkan *highlight* pada *icon* kursi tersebut. Proses tersebut akan terus diulang hingga aplikasi dimatikan oleh *user*.

B.3 Rancangan Desain *User Interface*

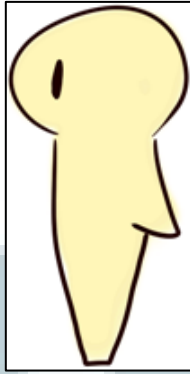
Sejak awal fase pengembangan, konsep utama yang digunakan dalam dari aplikasi “TOMOLIVI” adalah rasa kebersamaan antara keluarga Jepang sehingga *setting* yang akan digunakan dalam *user interface* aplikasi ini adalah ruang tamu rumah bergaya Jepang.



Gambar 3.10 Rancangan Desain *User Interface* Aplikasi “TOMOLIVI”

Merujuk pada gambar 3.10, rancangan desain *user interface* aplikasi “TOMOLIVI” terdiri dari beberapa hal sebagai berikut.

- 1) Enam buah kotak yang berwarna coklat menandakan posisi tempat *sprite* orang akan ditampilkan, sedangkan angka di dalamnya menandakan klasifikasi koordinat posisi sesuai jarak dari perangkat *Microsoft Kinect* dengan ketentuan yang telah ditentukan oleh anggota tim.
- 2) Bagian *background* yang berwarna abu-abu menandakan posisi gambar *background* yang akan ditampilkan.
- 3) Kotak kecil berwarna biru yang terletak di bagian kanan atas menandakan lokasi *icon* kursi yang akan digunakan sebagai indikasi sensor tekanan.



Gambar 3.11 *Sprite* Manusia

Setiap orang yang terdeteksi akan ditunjukkan pada *user interface* melalui penampilan *sprite* manusia seperti yang ditunjukkan pada gambar 3.11.

C. Implementasi Program

Seperti yang telah disebutkan sebelumnya, bagian *front end* dari aplikasi ini terdiri dari dua program yang terpisah, yaitu program C# yang bertugas mengakses dan mengunduh *file* dari FTP *Server* dan aplikasi UNITY3D yang merupakan tampilan *userinterface* utama dari aplikasi ini sendiri.

Aplikasi pertama yang bertugas untuk mengakses FTP *Server* dan mengunduh *file* tidak memiliki tampilan *userinterface* dan hanya beroperasi sebagai *background process* ketika dijalankan.

```
while (true)
{
    Thread.Sleep(1000);
    using (WebClient ftpClient = new WebClient())
    {
        ftpClient.Credentials = new System.Net.NetworkCredential("whitesnow.jp-doc", "a18H4w8P");
        ftpClient.DownloadFile("ftp://ftp.doc.whitesnow.jp/test_ryuugakusei/dataKinect.txt",
            @"C:\\Users\\Kevin\\Documents\\New Unity Project\\Assets\\dataKinect.txt");
        ftpClient.DownloadFile("ftp://ftp.doc.whitesnow.jp/test_ryuugakusei/data.txt",
            @"C:\\Users\\Kevin\\Documents\\New Unity Project\\Assets\\dataLight.txt");
        ftpClient.DownloadFile("ftp://ftp.doc.whitesnow.jp/test_ryuugakusei/data2.txt",
            @"C:\\Users\\Kevin\\Documents\\New Unity Project\\Assets\\dataPressure.txt");
    }
}
```

Gambar 3.12 Potongan *Source Code* Program

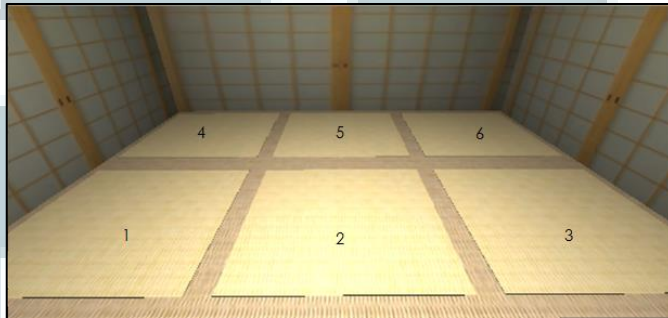
Dari potongan *source code* di atas, dapat dilihat beberapa fungsi yang digunakan. WebClient merupakan protokol yang digunakan C# untuk mengakses transfer data pada *server* yang terhubung dengan *internet*, termasuk FTP *Server*. NetworkCredential merupakan format yang digunakan untuk inialisasi data identifikasi (*username* dan *password*) untuk kepentingan *login* apabila *server* yang diakses merupakan *secured server* dan membutuhkan autentikasi. Fungsi DownloadFile() menentukan direktori dan nama *file* yang hendak diunduh apabila *file* yang serupa ditemukan, kemudian juga menentukan ke *directory* mana *file* tersebut akan disimpan. Dapat dilihat bahwa program didesain untuk mengunduh tiga *file* spesifik dari FTP *Server*, yang akan disimpan atas nama dataKinect.txt, dataLight.txt, dan dataPressure.txt di dalam *directory* aset aplikasi utama. Program akan terus berjalan hingga program ini dihentikan secara manual oleh *user* sehingga apabila terjadi perubahan isi dari ketiga data tersebut, data terbaru yang akan tersimpan ke dalam *directory* aplikasi utama.

Setelah ketiga *file* berhasil diunduh dan disimpan ke dalam *directory* utama, data yang terbaru tersebut dapat diakses oleh aplikasi utama. Aplikasi utama dikembangkan menggunakan *software* UNITY3D yang memungkinkan tampilan *user interface* dengan gaya tiga dimensi dan bersifat dinamis. Ketiga *file* yang telah diunduh kemudian mempengaruhi tampilan dari tampilan aplikasi utama ini secara terus-menerus.

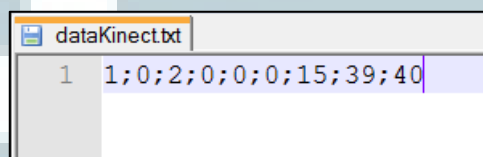
C.1 File dataKinect.txt

File dataKinect.txt menunjukkan data yang ditangkap oleh *Microsoft Kinect* mengenai jumlah dan posisi manusia yang berada di dalam ruangan subjek.

Posisi yang dimaksud merupakan koordinat *pixel* horizontal (x) dan jarak dari perangkat *Microsoft Kinect* (z), kemudian posisi tersebut dikelompokkan menggunakan kode angka satu sampai enam dengan klasifikasi seperti gambar 3.13.

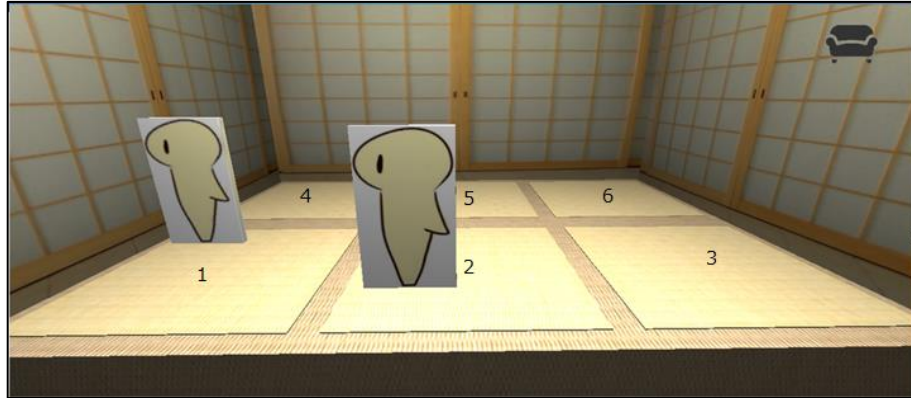


Gambar 3.13 Klasifikasi Posisi Orang pada *User Interface*



Gambar 3.14 Contoh Isi dari `dataKinect.txt`

Isi dari *file* `dataKinect.txt` sendiri memiliki format seperti gambar 3.14, dimana angka pertama menunjukkan koordinat posisi orang pertama, angka kedua menunjukkan koordinat posisi orang kedua, dan seterusnya hingga angka keenam yang merupakan batasan dari aplikasi ini, sedangkan tiga angka terakhir menunjukkan waktu diambilnya data dengan format jam;menit;detik. Dari contoh kondisi *file* `dataKinect.txt` di atas, aplikasi akan menampilkan tampilan seperti gambar 3.15.



Gambar 3.15 Contoh Tampilan *User Interface* Aplikasi Berdasarkan Contoh Isi dataKinect.txt

C.2 File dataLight.txt

File dataLight.txt menunjukkan data yang ditangkap oleh perangkat sensor *Arduino* yang berfungsi untuk mendeteksi tingkat intensitas cahaya. Isi dari dataLight.txt sendiri hanya terdiri dari satu digit angka yang akan bernilai satu apabila sensor mendeteksi adanya cahaya yang cukup (lampu ruangan aktif) dan akan bernilai nol apabila tidak mendeteksi cahaya (lampu ruangan tidak aktif). Data ditampilkan melalui perbedaan intensitas cahaya pada tampilan aplikasi.



Gambar 3.16 Tampilan Aplikasi Ketika dataLight.txt Bernilai Satu



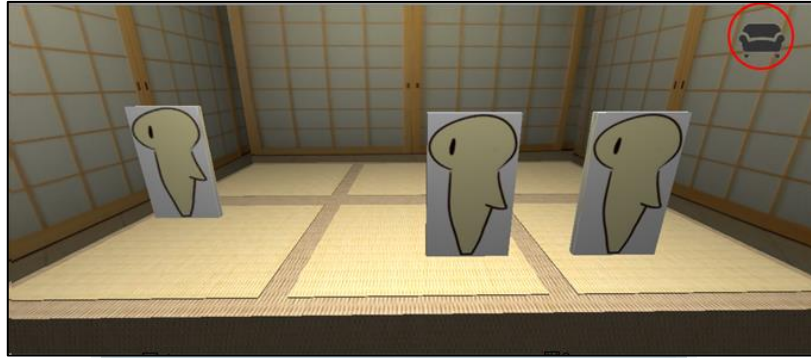
Gambar 3.17 Tampilan Aplikasi Ketika dataLight.txt Bernilai Nol

C.3 File dataPressure.txt

File dataPressure.txt menunjukkan data yang ditangkap oleh perangkat sensor *Arduino* yang berfungsi mendeteksi presensi orang yang menduduki kursi yang terdapat pada ruangan subjek melalui *input* berupa tekanan. Isi dari dataPressure.txt sendiri hanya terdiri dari satu digit angka yang akan bernilai satu ketika mendeteksi tekanan dan akan bernilai nol ketika tidak mendeteksi tekanan. Data ditunjukkan melalui *icon* kursi pada bagian kanan atas *user interface*.



Gambar 3.18 Tampilan Aplikasi Ketika dataPressure.txt Bernilai Nol



Gambar 3.19 Tampilan Aplikasi Ketika dataPressure.txt Bernilai Satu

3.3.2 Kebutuhan Pengguna

Aplikasi “TOMOLIVI” dirancang sebagai aplikasi *desktop*, dan dikembangkan menggunakan *software* UNITY3D untuk bagian *user interface*, tetapi *user* tidak diharuskan untuk meng-*install* aplikasi tersebut terlebih dahulu untuk menjalankan aplikasi “TOMOLIVI” ini. Perangkat yang digunakan dalam mengembangkan aplikasi ini sepenuhnya adalah *notebook* Sony Vaio Pro 11 dengan spesifikasi sebagai berikut.

- 1) *Processor*: 4th Intel® Core™ i5-4200U(1.60GHz/2.60GHz)
- 2) *Operating System*: Windows 8 64-bit
- 3) *Display*: 13.3” Full HD IPS touch display
- 4) *RAM*: 4GB RAM
- 5) *Hard Disk*: 128GB (128GB x1) SSD
- 6) *Graphic Card*: Intel HD Graphics 4400

3.3.3 Kendala yang Ditemukan

Selama berpartisipasi dalam proyek ini, beberapa masalah yang dihadapi oleh tim pengembang, baik masalah teknis maupun masalah non-teknis, antara lain sebagai berikut.

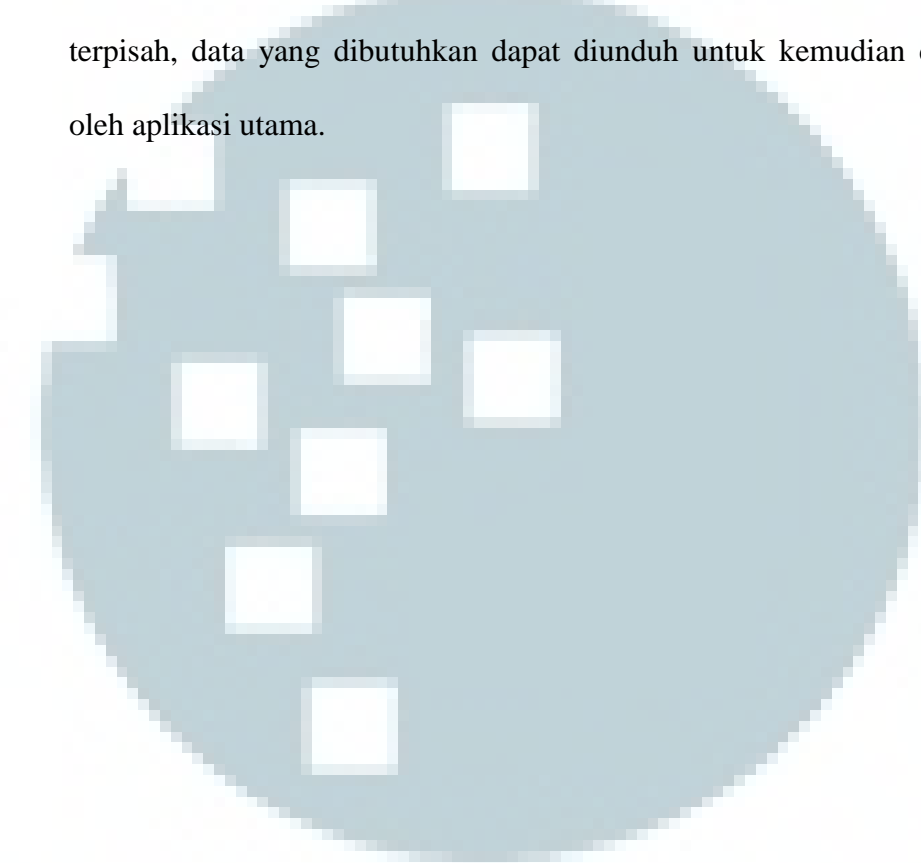
- 1) Beberapa kali terjadi kesulitan dalam berkomunikasi antara sesama anggota tim dikarenakan perbedaan bahasa yang digunakan dan istilah-istilah teknis yang berbeda antara bahasa Indonesia dan bahasa Jepang.
- 2) Pada awalnya, *software* dari perangkat sensor *Microsoft Kinect* tidak dapat membedakan apabila dua atau lebih orang terdeteksi secara bersamaan meskipun fitur tersebut dibutuhkan untuk mengembangkan aplikasi ini.
- 3) Koneksi antara ketiga perangkat sensor yang digunakan dalam sistem ini dengan *coordinator* yang menerima semua data sensor tersebut masih menggunakan kabel sehingga jaraknya terbatas dan merepotkan.
- 4) *Tools UNITY3D* yang digunakan untuk mengembangkan aplikasi utama memiliki keterbatasan dalam mengakses *FTP Server* yang digunakan untuk menyimpan data dan mengunduh data yang dibutuhkan.

3.3.4 Solusi atas Kendala yang Ditemukan

Dari seluruh kendala yang telah dikemukakan di atas, solusi yang didapatkan antara lain sebagai berikut.

- 1) Seiring bertambahnya waktu komunikasi antara masing-masing anggota tim, setiap anggota tim dapat saling mengerti satu sama lain dengan lebih baik, selain itu mengurangi kesalahpahaman ketika berinteraksi.
- 2) Dengan memodifikasi *software* dari perangkat sensor *Microsoft Kinect* tersebut, perangkat tersebut pada akhirnya mampu mendeteksi dan membedakan dua atau lebih orang secara bersamaan seperti yang diharapkan.

- 3) Dengan mengimplementasikan perangkat *Xbee*, koneksi *wireless* antara ketiga perangkat sensor dan *coordinator* pada komputer berhasil dikembangkan dan dapat beroperasi dengan lancar.
- 4) Dengan menciptakan program berbasis bahasa pemrograman C# yang terpisah, data yang dibutuhkan dapat diunduh untuk kemudian digunakan oleh aplikasi utama.



U M M N