



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

METODE DAN PERANCANGAN APLIKASI

3.1 Metode Penelitian

Metode penelitian dengan pembuatan prototipe aplikasi dipilih dengan fokus utama penelitian pada perancangan aplikasi yang mencakup logika pembacaan *hand gesture* terhadap arti yang disampaikan. Penelitian ini menggunakan metode sebagai berikut

1. Studi Literatur

Studi literatur merupakan tahapan dimana peneliti melakukan pembelajaran dari buku, jurnal, artikel, maupun referensi lain, yang tersedia secara *online* maupun *offline*, yang berhubungan dengan penelitian yang dilakukan.

2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan sistem, meliputi fitur-fitur yang perlu disediakan, seperti *hardware*, *software*, kamera, SDK, dan data-data bahasa isyarat yang akan diimplementasikan.

3. Desain Sistem

Pada tahap desain sistem ini dilakukan perancangan *user interface*, desain modul-modul aplikasi, dan desain konten.

4. Pemrograman Sistem

Pada tahap pemrograman ini dilakukan pembuatan program untuk merealisasikan aplikasi berdasarkan rancangan pada desain sistem.

5. Testing

Pada tahap ini, aplikasi diuji untuk memastikan bahwa program tersebut dapat berjalan sesuai dengan yang diinginkan.

6. Survei pengaruh

Pada tahap ini, dilakukan survei kenyamanan pengguna setelah menggunakan aplikasi penerjemah bahasa isyarat.

3.2 Variabel Penelitian

Menurut Al-Maskari dan Sanderson (2010) melalui penelitiannya, kepuasan seorang terhadap suatu aplikasi tidak dapat diukur oleh satu variabel saja, melainkan harus melalui beberapa variabel. Dalam penelitian ini peneliti menggunakan dua variabel untuk menentukan kepuasan pengguna, yaitu *System Effectiveness* dan *User Effort*.

3.2.1 System Effectiveness

System Effectiveness mengukur bagaimana suatu sistem dapat memberikan hasil yang sesuai dengan masukan pengguna. Dalam penelitian ini, *System Effectiveness* dinilai berdasarkan kesesuaian hasil keluaran aplikasi dengan masukan pengguna.

3.2.2 User Effort

User Effort mengukur besarnya usaha yang diperlukan pengguna untuk mencapai suatu tujuan. Dalam penelitian ini, *User Effort* dinilai berdasarkan

banyaknya percobaan yang dilakukan pengguna untuk mendapatkan hasil yang diinginkan.

3.3 Teknik Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah kuesioner. Pengumpulan data ditujukan untuk mendapatkan implementasi aplikasi pada penerjemahan bahasa isyarat secara umum. Kuesioner pengumpulan data menggunakan sebuah *form* yang dicetak dan langsung diisi saat *user* melakukan uji aplikasi. Kuesioner pengumpulan data terlampir pada Daftar Lampiran.

3.4 Data Flow Diagram (DFD)

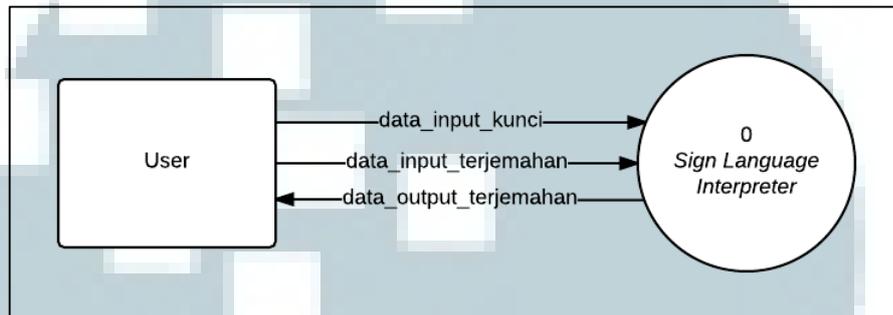
Menurut Kristianto (2003), *Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluaran dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Pada penelitian ini, peneliti membuat dua aplikasi, yaitu ProjectAdmin dan ProjectUser. ProjectAdmin digunakan untuk membuat variabel kunci dan mencoba proses penerjemah isyarat, sedangkan ProjectUser dapat menerjemahkan isyarat serta memiliki tampilan yang lebih baik untuk *user*.

Pada bagian DFD berikut, peneliti memasukkan DFD dari ProjectAdmin, karena ProjectUser menggunakan *data flow* yang sama dengan proses penerjemah isyarat pada ProjectAdmin.

3.4.1 Context Diagram

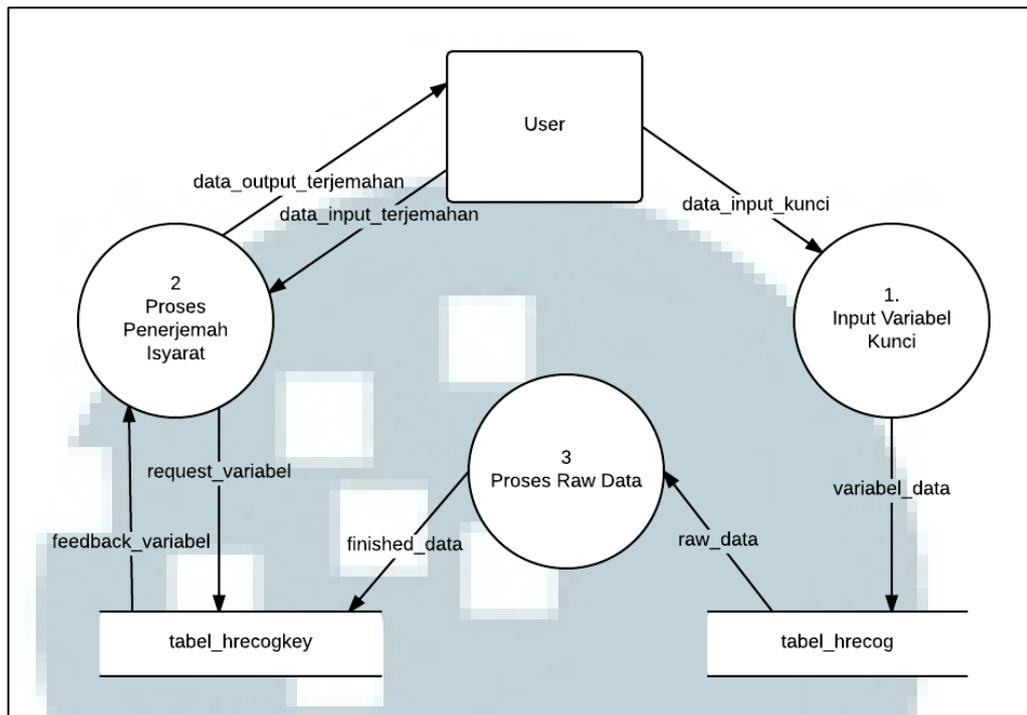
Pada *Diagram Context* digambarkan bahwa aplikasi yang dikembangkan menerima `data_input_kunci` dan `data_input_terjemahan` dan memprosesnya di dalam aplikasi *Sign Language Interpreter*. Lalu proses tersebut mengembalikan sebagai `data_output_terjemahan`. `Data_input_kunci` tidak memiliki output karena menyimpan *data* ke dalam *database*.



Gambar 3.1 *Context Diagram* aplikasi

3.4.2 DFD Level 1

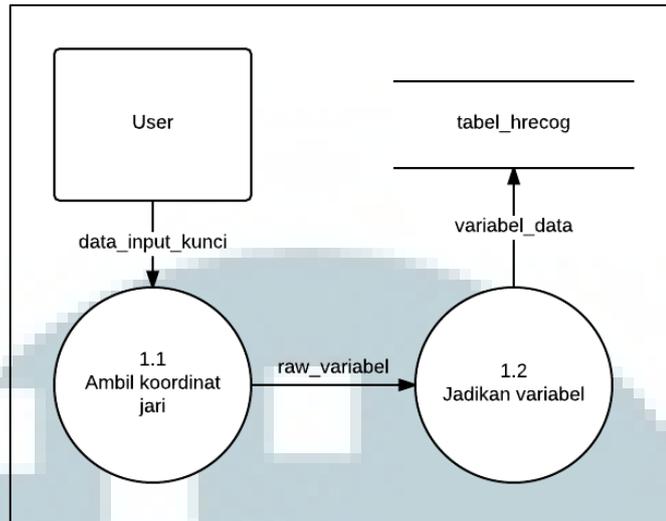
Pada DFD Level 1 digambarkan bahwa data yang di masukan pengguna dilanjutkan ke dalam dua proses, `data_input_kunci` ke dalam proses Input Variabel Kunci dan `data_input_terjemahan` ke Proses Penerjemah isyarat. Pada proses Input Variabel Kunci data yang dikirim oleh pengguna diproses dan di masukan ke dalam tabel `hrecog` lalu dijalankan Proses Raw Data. Proses ini secara manual menggunakan MySQL phpMyAdmin, menjalankan *query* untuk mengubah `raw_data` dari tabel `hrecog` menjadi `finished_data` lalu di masukan ke tabel `hrecogkey`. Sedangkan pada proses penerjemah isyarat menerima masukan dari pengguna, lalu diproses menggunakan variabel dari tabel `hrecogkey`. Setelah selesai proses lalu data dikembalikan ke pengguna sebagai `data_output_terjemahan`. Ilustrasi dapat dilihat pada gambar 3.2.



Gambar 3.2 DFD Level 1 aplikasi

3.4.3 DFD Level 2 Proses Input Variabel Kunci

Pada DFD level 2 proses Input Variabel Kunci digambarkan masukan dari pengguna, `data_input_kunci`, diproses untuk diambil koordinat dari jari, lalu diproses kembali dan dijadikan `variabel_data`. Setelah itu di masukan ke dalam tabel `hrecog` untuk diproses sekali lagi menjadi `finished_data` dan di masukan ke dalam tabel `hrecogkey` melalui Proses Raw Data. Proses Raw Data merupakan proses manual eksekusi *query* dari MySQL phpMyAdmin yang menjadikan `raw_data` menjadi `finished_data`, yaitu data kunci yang digunakan untuk proses penerjemahan. Ilustrasi dapat dilihat pada gambar 3.3.

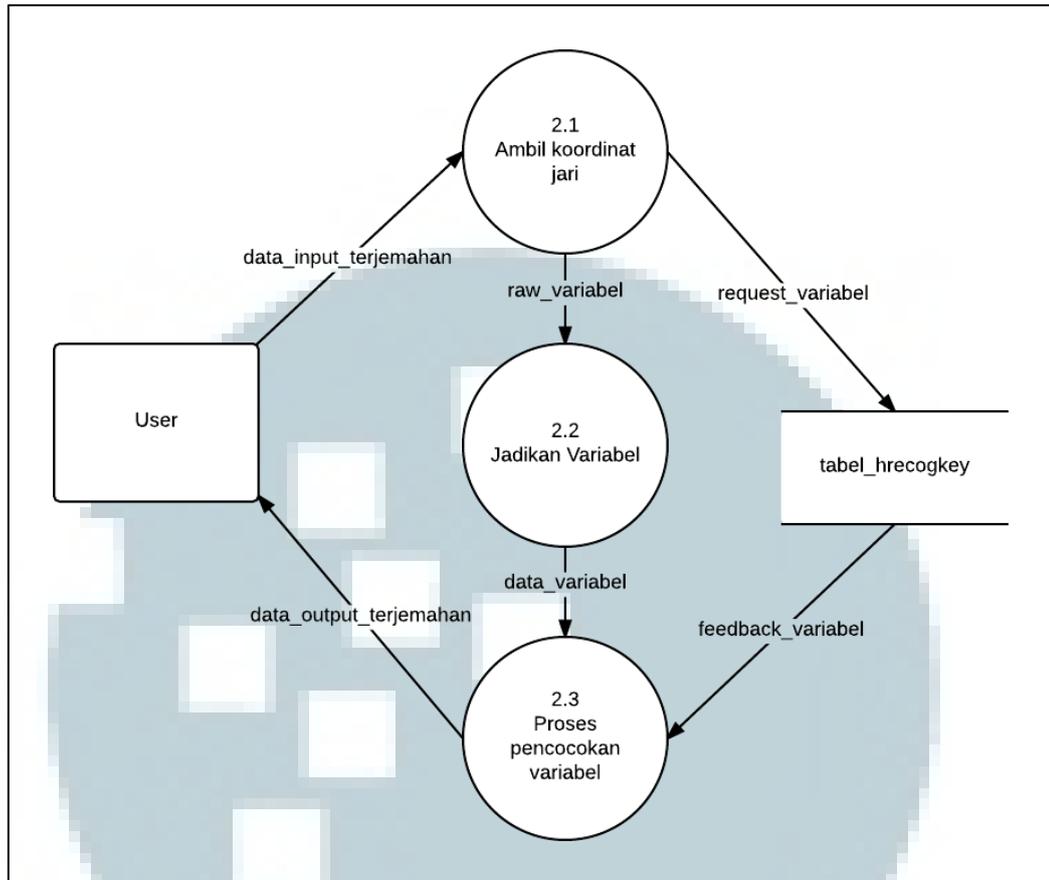


Gambar 3.3 DFD level 2 proses Input Variabel Kunci

3.4.4 DFD Level 2 Proses Penerjemah Isyarat

Pada DFD level 2 proses penerjemah isyarat digambarkan masukan dari user diproses untuk diambil koordinat jarinya, yaitu `data_input_terjemahan`. Hasil dari proses tersebut, `raw_variabel`, dikirim ke proses selanjutnya untuk dijadikan `data_variabel`. Proses tersebut juga mengirim *request* untuk mendapatkan variabel yang sudah ada di tabel `hrecogkey`. `Data_variabel` lalu di cocokkan dengan `feedback_variabel` dari tabel `hrecogkey`. Setelah terjadi kecocokan proses mengembalikan data sebagai `data_output_terjemahan`. Ilustrasi dapat dilihat pada gambar 3.4.

Data flow ini juga terjadi pada aplikasi ProjectUser. Tidak ada perbedaan dalam alur *data flow* yang terjadi pada ProjectUser dengan ProjectAdmin. Secara garis besar *data flow*-nya adalah, *input* dari *user*, ambil koordinat jari lalu jadikan variabel, *request* variabel kunci, lalu hasil variabel dicocokkan dengan variabel kunci.



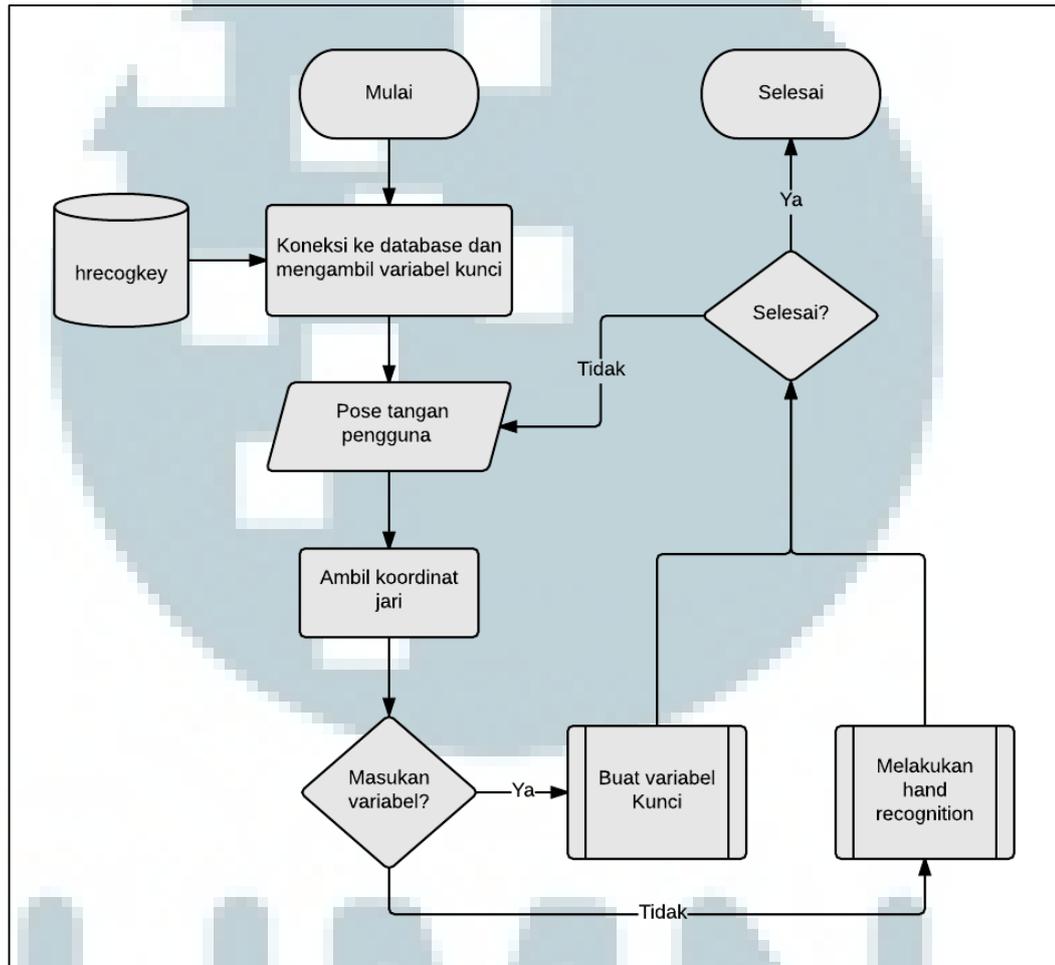
Gambar 3.4 DFD level 2 proses Penerjemah Isyarat

3.5 Rancangan Aplikasi

Berikut ini adalah penjelasan secara garis besar tentang alur proses aplikasi. Dalam penelitian ini peneliti membuat dua aplikasi, yang pertama sebuah prototipe aplikasi yang dibuat pada bahasa C++, peneliti menamakannya ProjectAdmin, guna mengimplementasikan dan menguji desain logika *finger tracking and recognition* yang dibuat oleh peneliti. Aplikasi tersebut juga digunakan untuk membuat variabel-variabel kunci yang dibutuhkan untuk mengenali pose yang dilakukan oleh pengguna. Setelah itu peneliti membuat aplikasi kedua dengan bahasa C# yang peneliti sebut dengan ProjectUser untuk mendukung *user interface* yang lebih baik

bagi pengguna tanpa disertakannya kemampuan untuk membuat atau mengubah variabel-variabel kunci yang ada.

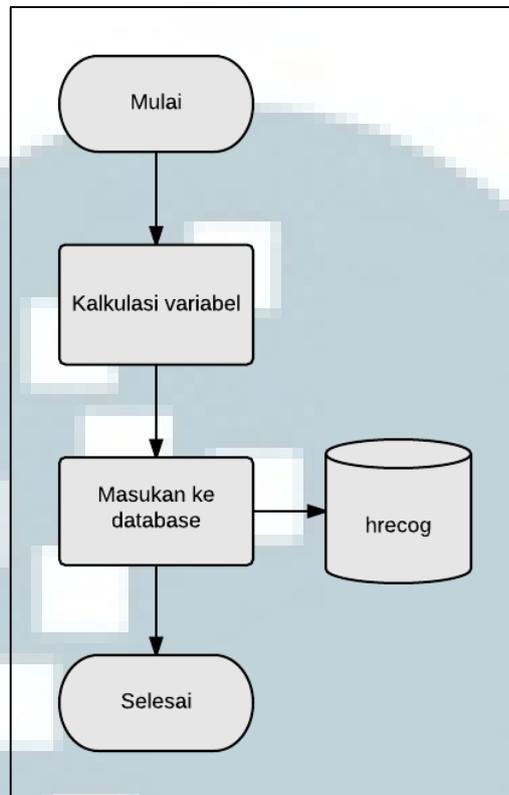
Aplikasi ProjectAdmin mempunyai dua fungsi, yaitu *input database* dan penerjemahan isyarat. Gambar 3.5 menampilkan garis besar alur aplikasi ProjectAdmin.



Gambar 3.5 Diagram alir aplikasi ProjectAdmin

3.5.1 Proses buat variabel kunci

Berikut ini merupakan *flowchart* dalam proses pembuatan variabel kunci.



Gambar 3.6 Diagram alir proses pembuatan variabel kunci

Gambar 3.5 menggambarkan pada awal mulanya aplikasi membaca *database*, lalu mengambil koordinat jari untuk diproses lebih lanjut. Selanjutnya pose tangan diambil oleh algoritma. Pada pemilihan masukan variabel pengguna harus menekan tombol sesuai dengan huruf yang sesuai dengan masukan pose. Setelah itu dilakukan kalkulasi variabel.

Pada proses kalkulasi variabel kunci dimulai dari *input* pengguna berupa sebuah pose yang dilakukan oleh tangan kanan. Setelah itu koordinat-koordinat pada ujung jari diambil koordinat tiga dimensinya. Koordinat-koordinat tersebut dimasukkan ke dalam logika metode yang dibuat oleh peneliti menjadi suatu susunan

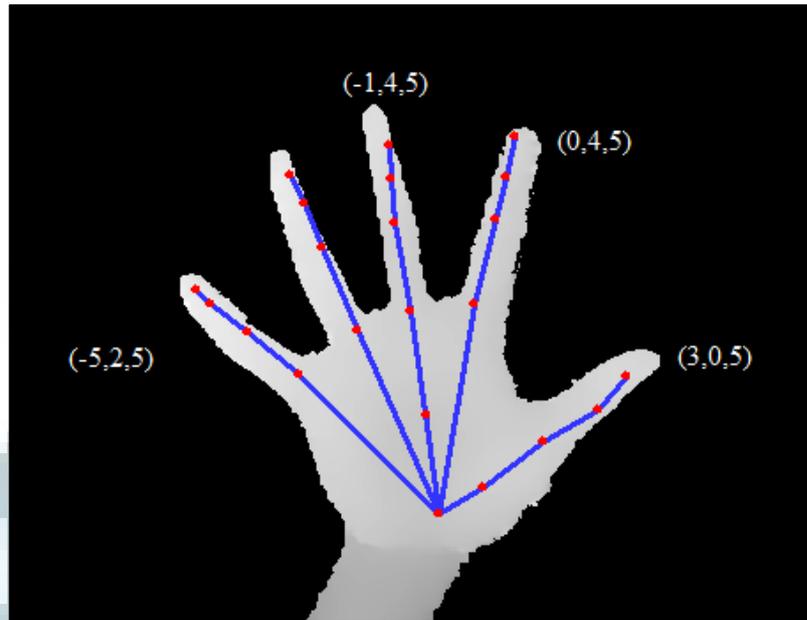
variabel yang unik untuk setiap pose yang dilakukan. Setelah itu dimasukkan ke dalam *database*.

Intel RealSense SDK dapat mengukur jarak antar *Geometric node* yang didapat dalam satuan meter. Setelah itu diambil koordinat dari ujung jari jempol, telunjuk, tengah, dan kelingking. SDK mengembalikan nilai koordinat tiga dimensinya, lalu peneliti mengambil koordinat dari sumbu x dan y dari masing-masing jari. Dari koordinat tersebut peneliti membuat empat variabel, yaitu reg1, reg2, reg3, dan reg4, dimana rinciannya adalah sebagai berikut.

Tabel 3.1 Penjelasan variabel-variabel kunci

Variabel	Penjelasan
reg1	Koordinat x jari jempol dikurangi koordinat x jari telunjuk.
reg2	Koordinat y jari jempol dikurangi koordinat y jari telunjuk
reg3	Koordinat x jari tengah dikurangi koordinat x jari kelingking
reg4	Koordinat y jari tengah dikurangi koordinat y jari kelingking

Peneliti tidak menggunakan koordinat jari manis karena huruf-huruf yang digunakan dalam *American Sign Language* jarang menggunakan manipulasi jari manis. Mekanisme algoritma kalkulasi variabel akan dijelaskan dengan bantuan ilustrasi gambar 3.7 berikut.



Gambar 3.7 Ilustrasi pengambilan koordinat

Pada gambar 3.2 diketahui koordinat jari jempol adalah $(3,0,5)$, koordinat jari telunjuk adalah $(0,4,5)$, koordinat jari tengah adalah $(-1,4,5)$, dan koordinat jari kelingking adalah $(-5,2,5)$. Setelah itu algoritma membuat variabel-variabel $reg1$, $reg2$, $reg3$, dan $reg4$ dari koordinat-koordinat tersebut. Dalam contoh ilustrasi di atas $reg1$ adalah $3-0 = 3$, $reg2$ adalah $0-4 = -4$, $reg3$ adalah $-1 - -5 = -6$, dan $reg4$ adalah $4-2 = 2$. Jadi terkumpullah variabel-variabel $reg1 = 3$, $reg2 = -4$, $reg3 = -6$, $reg4 = 2$. Lalu variabel-variabel tersebut dimasukkan ke dalam *database* beserta huruf yang sesuai dengan pose tersebut dengan cara menekan tombol sesuai huruf. Karena metode ini hanya mengenali pose lewat koordinat, huruf yang menggunakan gerakan tangan tidak diikutsertakan. Huruf tersebut adalah huruf J dan Z.

Menurut Supranto (2000) dalam penelitiannya, untuk penelitian eksperimen dengan rancangan acak lengkap, acak kelompok, atau faktorial, secara sederhana dapat dirumuskan: $(t-1) (r-1) > 350$, dimana t adalah banyaknya kelompok

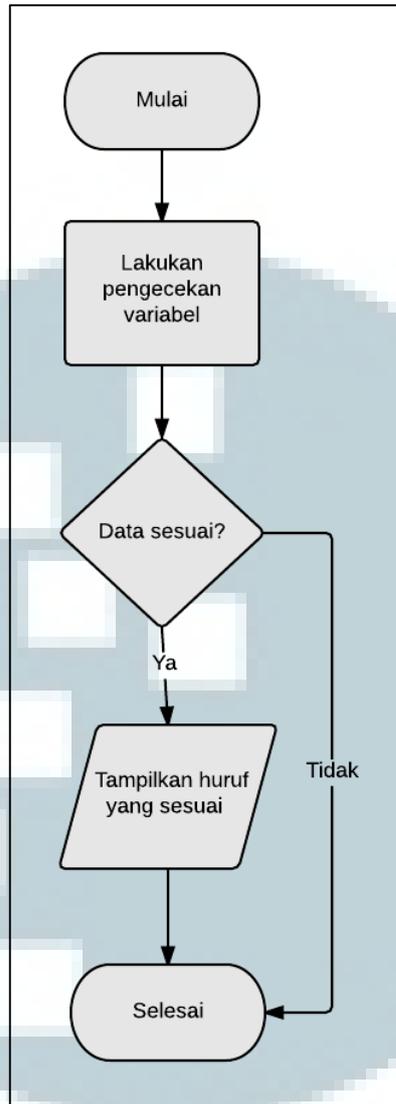
perlakuan, dan r adalah jumlah replikasi. Dalam kasus ini peneliti menggunakan 24 huruf untuk dikenali yang berarti kelompok perlakuannya sebanyak 24. Dengan rumus tersebut didapatkan hasil $(24-1) (r-1) > 350$, yang berarti $r > 15$. Dalam aplikasi ini peneliti melakukan *input* sebanyak 20 kali untuk membuat variabel kunci sebuah huruf.

Setelah semua huruf dilakukan pembuatan variabel kunci, lalu peneliti menggunakan MySQL phpMyAdmin secara manual memasukkan sebuah *query database* untuk mendapatkan *ceiling* dan *floor* dari tiap-tiap variabel. Dari hasil *query* tersebut terciptalah variabel-variabel kunci.

3.5.2 Proses Penerjemahan Isyarat (Hand Recognition)

Setelah variabel kunci dibuat, variabel-variabel tersebut menjadi acuan logika algoritma untuk mengenali pose atau *gesture* yang dimasukan oleh pengguna. Aplikasi membaca variabel kunci pada *database*, memasukkan variabel ke penampungan sementara. Lalu aplikasi membaca koordinat-koordinat jari masukan, dan dikalkulasikan untuk menjadikannya variabel. Lalu variabel-variabel tersebut dicocokkan terhadap variabel yang telah dibaca melalui database. Jika ditemukan koordinat dengan variabel-variabel yang sesuai, maka dikeluarkan huruf yang bersangkutan dengan kecocokan variabel-variabel tersebut.

Penjelasan lebih detail akan dijelaskan dengan bantuan *flowchart* pada gambar 3.8 berikut.



Gambar 3.8 Diagram alir proses penerjemah isyarat

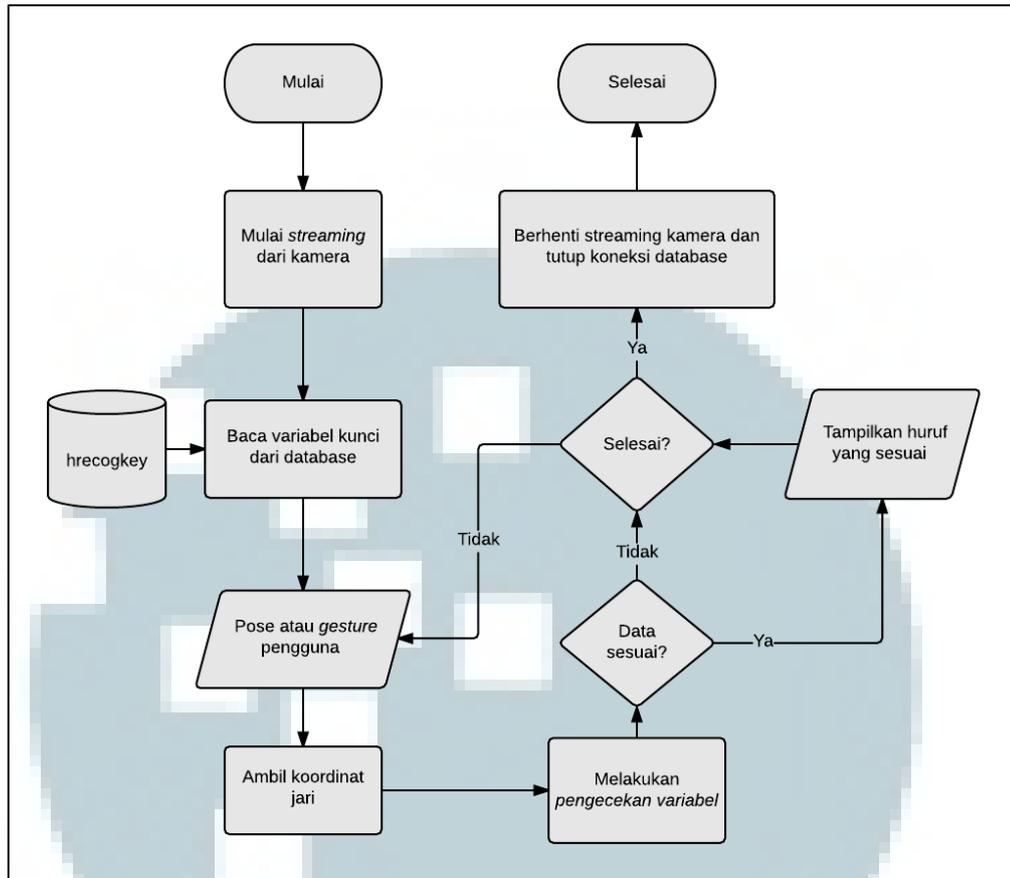
Pada *flowchart* digambarkan awal mula aplikasi koneksi terhadap *database* dan membaca variabel kunci yang telah terbuat dan dimasukkan ke dalam sebuah *temporary array*. Lalu aplikasi menerima masukan pose dari *user*, lalu diambil koordinat jarinya. Jika pengguna tidak melakukan *input* variabel, proses berlanjut ke pengecekan variabel. Dalam proses ini dilakukan kalkulasi koordinat seperti proses kalkulasi pada proses pembuatan variabel kunci. Setelah itu dilakukan proses penyocokan variabel-variabel yang telah dikalkulasi dengan variabel-variabel

kunci yang telah dimasukkan ke dalam *temporary array*. Berikut adalah *pseudocode* algoritma penyocokan.

```
IF(reg1 <= reg1up & reg1 >= reg1down & reg2 <= reg2up & reg2 >= reg2down &
    reg3 <= reg3up & reg3 >= reg3down & reg4 <= reg4up & reg4 >=
    reg4down)
    RETURN huruf
ENDIF
```

Jika ditemukan data yang sesuai dan cocok dengan kondisi yang ditentukan, maka akan di-*return* sebuah huruf. Lalu huruf tersebut ditampilkan ke *user interface*. Algoritma pengecekan huruf dilakukan setiap 3 detik. Jadi jika pengguna memasukkan pose yang sama dalam waktu proses pengecekan huruf yang keluar juga akan sama.

Setelah semua fitur pada aplikasi ProjectAdmin telah selesai, peneliti membuat aplikasi ProjectUser dengan bahasa C#. Aplikasi dibuat agar lebih mudah dalam pembuatan *user interface* yang lebih baik dan *user friendly*. ProjectUser hampir sama dengan ProjectAdmin, perbedaannya ProjectUser tidak memiliki *subprocess input* variabel kunci ke dalam *database*. ProjectUser dibuat khusus untuk digunakan langsung oleh pengguna untuk menerjemahkan bahasa isyarat. Pada gambar 3.9 ditampilkan *flowchart* alur proses aplikasi ProjectUser.



Gambar 3.9 Diagram alir aplikasi ProjectUser

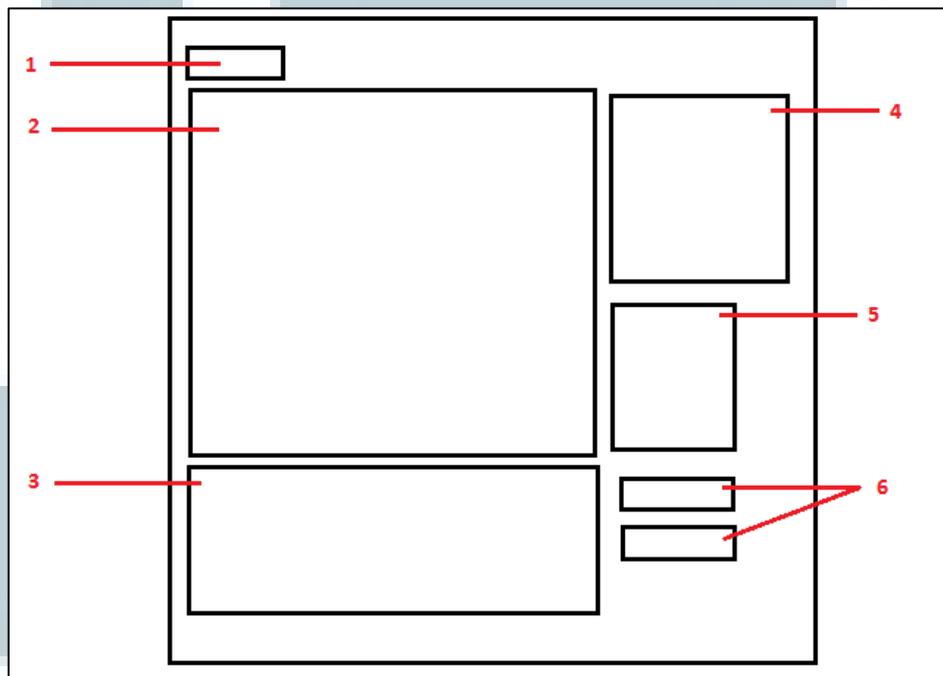
Hampir sama dengan proses penerjemah pada aplikasi ProjectAdmin, hanya saja ditambah fitur untuk memulai dan menghentikan *video streaming* pada aplikasi. Peneliti juga menambahkan fitur pose *thumb up* atau jempol ke atas untuk *linea* baru dan *thumb down* atau jempol ke bawah untuk menghapus deretan huruf.

3.6 Rancangan Tampilan Antarmuka Aplikasi

Rancangan tampilan antarmuka dideskripsikan dengan aplikasi ProjectUser, karena aplikasi tersebut yang dipergunakan langsung untuk *user*.

Tampilan aplikasi ini memiliki enam komponen utama. Berikut ini adalah daftar komponen-komponen tersebut beserta penjelasannya.

1. *Menu Device*. Menu ini menampilkan seluruh *device input* yang berupa kamera yang terkoneksi oleh komputer.
2. *Stream Panel*. Sebuah panel yang berfungsi untuk menampilkan *stream* video dari kamera.
3. *Text Panel*. Sebuah panel yang berfungsi untuk menampung huruf-huruf hasil *output* masukan dari pengguna.
4. *Video Option*. Sebuah fitur tambahan dari RealSense SDK untuk mode video yang digunakan.
5. *Sign Label*. Sebuah teks yang mengeluarkan *output* dari huruf apa yang dibuat oleh pengguna.
6. *Start dan Stop Button*. Tombol untuk mulai dan berhenti *streaming* video.



Gambar 3.10 Rancangan tampilan utama aplikasi

3.7 Rancangan Basis Data

Aplikasi ini menggunakan satu basis data dengan dua tabel yang tidak saling berhubungan didalamnya. Kedua tabel tersebut adalah tabel hrecog dan tabel hrecogkey.

3.7.1 Tabel hrecog

Tabel hrecog merupakan tabel untuk menerima data variabel dari masukan *input* pengguna. Berikut ini adalah struktur dari tabel hrecog.

Tabel 3.2 Struktur tabel hrecog

Nama Kolom	Tipe	Panjang Data	PK	FK	Keterangan
Count	int	11	Y	T	Primary key
ID	char	2	T	T	Huruf yang sesuai variabel
reg1	double		T	T	Variabel 1
reg2	double		T	T	Variabel 2
reg3	double		T	T	Variabel 3
reg4	double		T	T	Variabel 4

3.7.2 Tabel hrecogkey

Tabel hrecogkey merupakan tabel untuk menerima variabel-variabel dari tabel hrecog yang telah diproses menjadi *finished_data*. Tabel struktur data hrecogkey dapat dilihat pada tabel 3.3.

Tabel 3.3 Struktur tabel hrecogkey

Nama Kolom	Tipe	Panjang Data	PK	FK	Keterangan
Count	int	11	Y	T	Primary key
ID	char	2	T	T	Huruf yang sesuai variabel
reg1up	double		T	T	<i>Ceiling</i> variabel 1
reg1down	double		T	T	<i>Floor</i> variabel 1
reg2up	double		T	T	<i>Ceiling</i> variabel 2
reg2down	double		T	T	<i>Floor</i> variabel 2
reg3up	double		T	T	<i>Ceiling</i> variabel 3
Reg3down	double		T	T	<i>Floor</i> variabel 3
reg4up	double		T	T	<i>Ceiling</i> variabel 4
reg4down	double		T	T	<i>Floor</i> variabel 4

UMMN