



Sistem *Gesture* Accelerometer dengan Metode *Fast Dynamic Time Warping* (FastDTW)

Sam F. Chaerul Haviana^{a*}

^aUniversitas Islam Sultan Agung Semarang

Naskah Diterima : 11 Juni 2015; Diterima Publikasi : 12 Agustus 2015

Abstract

In the modern environment, the interaction between humans and computers require a more natural form of interaction. Therefore, it is important to be able to build a system that can meet these demands, such as by building a hand gesture recognition system or gesture to create a more natural form of interaction. This study aims to design a smartphone's accelerometer gesture system as human computer interaction interfaces using FastDTW (Fast Dynamic Time Warping). The result of this study is form of gesture interaction which implemented in a system that can make the process of recognition of the human hand movements based on a smartphone accelerometer which generates a command to run the media player application functions as a case study. FastDTW as the development of Dynamic Time Warping method (DTW) is able to compute faster than DTW and have an accuracy approaching DTW. From the test results, FastDTW show a fairly high degree of accuracy reached 86% and showed a better computing speed compared to DTW

Keywords: Human and computer interaction; Accelerometer-based gesture; FastDTW; Media player application function

Abstrak

Di dalam lingkungan modern, interaksi antara manusia dan komputer menuntut bentuk interaksi yang lebih natural. Oleh karena itu, penting untuk dapat membangun sebuah sistem yang dapat memenuhi tuntutan tersebut, diantaranya dengan membangun sistem pengenalan gerakan tangan atau *gesture* untuk menghasilkan bentuk interaksi yang lebih natural. Penelitian ini bertujuan untuk merancang sistem *gesture* menggunakan accelerometer pada *smartphone* sebagai antarmuka interaksi manusia dan komputer dengan menggunakan metode FastDTW (*Fast Dynamic Time Warping*). Hasil dari penelitian ini adalah bentuk interaksi *gesture* yang diimplementasikan dalam sebuah sistem yang dapat melakukan proses pengenalan terhadap gerakan tangan manusia menggunakan accelerometer pada *smartphone* yang menghasilkan perintah untuk menjalankan fungsi aplikasi *media player* sebagai studi kasusnya. FastDTW sebagai pengembangan dari metode *Dynamic Time Warping* (DTW) mampu melakukan komputasi dengan lebih cepat dibandingkan DTW dan memiliki akurasi yang mendekati DTW. Dari hasil pengujian, FastDTW menunjukkan tingkat akurasi yang cukup tinggi mencapai 86% dan menunjukkan kecepatan komputasi yang lebih baik dibandingkan dengan DTW.

Kata kunci: Interaksi manusia dan komputer; *Gesture* berbasis accelerometer; FastDTW; Fungsi aplikasi media player

1. Pendahuluan

Di dalam lingkungan modern, perkembangan interaksi antara manusia dan komputer atau perangkat lain yang berada di lingkungan sekitarnya semakin menuju ke arah *smart environment*. *Smart environment* adalah dunia fisik yang kaya akan sensor, aktuator, display, dan elemen komputasi yang tidak tampak terjalin secara langsung namun tertanam rapi pada benda-benda sehari-hari di kehidupan kita, dan terhubung melalui jaringan secara terus menerus (Weiser *et al.*, 1999). Dalam *smartenvironment* berbagai perangkat dibangun agar dapat mengenali pengguna / pemilikinya sehingga dapat memberikan respon yang sesuai dengan aktivitas penggunaannya. Bentuk interaksinya juga sudah mulai meninggalkan bentuk interaksi yang

konvensional seperti menekan tombol, memutar tombol dll. Bentuk interaksi yang ideal untuk sebuah *smartenvironment* adalah bentuk interaksi yang lebih natural, yaitu bentuk interaksi yang sudah umum dan dipahami oleh penggunaanya secara alamiah. Salah satunya dengan menggunakan *gesture* yaitu gerakan dinamis dari bagian tubuh seperti tangan, lengan, kaki, kepala atau bagian tubuh yang lain (Akl, 2010). Sebagai contoh, dengan *gesture* pengguna cukup menggerakkan tangan dengan pola tertentu untuk mengoperasikan perangkat di sekitarnya. Untuk itu dibutuhkan sistem yang ideal untuk dapat mengenali *gesture* yang dilakukan. Inilah yang menjadi pokok permasalahan dan tantangan yang harus dijawab para peneliti agar dapat membangun sebuah sistem yang ideal dan mudah diterapkan sebagai antarmuka

*) Penulis korespondensi: hanx_gen3@yahoo.com

interaksi manusia dan komputer atau perangkat lain di lingkungan sekitarnya. Pada kasus tertentu seperti pada orang-orang dengan disabilitas, yang tidak dapat melihat antarmuka perangkat secara fisik, pengenalan *gesture* dapat sangat membantu sebagai antarmuka alternatif untuk berinteraksi dengan komputer.

Pada beberapa penelitian yang ada, sejumlah teknik untuk pengenalan *gesture* sudah dikembangkan. Salah satunya adalah dengan menggunakan *video based recognition* yaitu teknik yang memanfaatkan kamera untuk merekam gerakan dan aktivitas manusia untuk dikenali oleh komputer. Teknik ini bergantung pada kamera yang digunakan sebagai alat utama untuk mendapatkan data *gesture* yang dilakukan oleh manusia. Meskipun teknik ini dinyatakan cukup efektif dengan tingkat pengenalan hingga 82,5%, namun ada beberapa keterbatasan yang membuat teknik ini tidak dapat diterapkan pada aplikasi yang lebih luas. Pertama bahwa teknik ini bergantung pada kamera yang memiliki area perekaman (*viewport*) yang terbatas sehingga objek yang direkam terbatas pula pergerakannya. Dibutuhkan lebih banyak kamera untuk dapat merekam pada area yang lebih luas. Kedua bahwa teknik ini bergantung pula pada kondisi pencahayaan, di mana dibutuhkan cahaya yang cukup untuk dapat merekam obyek (Akl, 2010). Selain itu teknik *video based recognition* ini juga berpotensi menimbulkan terjadinya kesalahan pengenalan apabila ada objek selain objek utama yang direkam masuk ke dalam area perekaman. Beberapa kelemahan tersebut membuat teknik pengenalan ini menjadi kurang efektif.

Selain teknik *video based*, ada pula teknik yang dilakukan dengan pemakaian sarung tangan elektromagnetik untuk merekam *gesture*. Meskipun teknik ini memiliki tingkat pengenalan yang sangat tinggi terhadap *gesture*, namun dengan pemakaian sarung tangan yang dihubungkan dengan komputer akan membuat penggunaannya tidak nyaman (Akl, 2010). Untuk itu dibutuhkan alternatif teknologi yang dapat mengatasi kelemahan teknik-teknik tersebut dengan penggunaan alat pengenalan yang sudah umum dipakai oleh manusia dalam aktivitas kesehariannya.

Dalam aktivitas kesehariannya, manusia hampir selalu menggunakan peralatan elektronik untuk membantu aktivitasnya. Salah satu perangkat yang paling sering digunakan manusia modern adalah *smartphone*. *Smartphone* adalah perangkat yang fungsi utamanya adalah untuk berkomunikasi antar manusia melalui suara, video atau teks. Namun selain itu *smartphone* juga memiliki kemampuan lain selain alat komunikasi, antara lain kemampuan untuk memberikan respon terhadap aktivitas penggunaannya seperti misalnya mematikan layar saat digunakan untuk menelepon atau menaruhnya di dalam saku. Atau contoh lain, *smartphone* dapat merespon

gerakan yang dilakukan terhadapnya seperti ketika digerakkan untuk bermain *game* dengan *smartphone*. Hal ini dimungkinkan karena berkembangnya teknologi *Micro Electronical Mechanical System* (MEMS), yang mampu menghasilkan berbagai sensor dengan ukuran yang relatif kecil dan murah (Wu *et al.*, 2009). *Smartphone* seperti iPhone dan *smartphone* Android kini sudah dilengkapi dengan berbagai sensor seperti sensor Gyroscope, sensor Proximity, sensor Magnetik dan sensor Accelerometer. Berbagai sensor inilah yang kemudian dapat dimanfaatkan sebagai alat penelitian, salah satunya sensor Accelerometer untuk pengenalan *gesture*.

Selain *smartphone*, manusia modern juga dalam kehidupan sehari-harinya banyak berinteraksi dengan perangkat elektronik sebagai media penyedia informasi dan hiburan seperti televisi, video *player*, musik *player* dan lain-lain. Hampir semua manusia modern yang normal memiliki media *player*, baik itu dalam bentuk perangkat khusus untuk memutar *filemultimedia* seperti *DVD player* ataupun yang hanya menggunakan aplikasi media *player* di dalam komputer. Ini menunjukkan bahwa media *player* sudah menjadi bagian dari keseharian manusia. Untuk berinteraksi dengan perangkat tersebut, seperti sudah disampaikan sebelumnya, manusia terkadang masih menggunakan cara tradisional seperti menekan dan atau memutar tombol. Dengan adanya teknologi seperti pengenalan *gesture*, memungkinkan manusia berinteraksi dengan perangkat-perangkat tersebut dengan cara yang berbeda, yang secara umum dan secara alamiah sudah dimengerti yaitu dengan menggunakan *gesture*.

Penelitian ini menjadi penting untuk dapat menghasilkan sebuah sistem *gesture* yang efektif, efisien dan mudah diterapkan atau diaplikasikan. Kemudian dengan menggunakan metode FastDTW (*Fast Dynamic Time Warping*) yang dinyatakan lebih cepat dari DTW (*Dynamic Time Warping*) oleh Salvador dan Chan (2007) sebagai metode pengenalan *gesture* maka penelitian ini berusaha meningkatkan akurasi dan kecepatan pengenalannya. Selain itu pemanfaatan sensor accelerometer pada *smartphone* dimaksudkan untuk mengatasi masalah batasan-batasan yang terjadi pada teknik pengenalan sebelumnya seperti pada pengenalan menggunakan teknik *videobased*. Dengan melakukan pengujian lebih lanjut pada penelitian ini maka diharapkan sistem pengenalan *gesture* yang dihasilkan akan lebih efektif dan efisien. Serta penggunaan media *player* sebagai studi kasus pada penelitian ini diharapkan dapat mewakili aktivitas keseharian manusia yang sudah umum menggunakan media *player* sebagai alat hiburan. Meskipun peluang untuk mengaplikasikan apa yang diteliti pada penelitian sangatlah luas tidak terbatas pada aplikasi media *player* saja.

Berbagai metode dan pendekatan sudah dilakukan untuk pengenalan *gesture*. Salah satu yang banyak

dipakai dalam berbagai literatur yaitu dengan menggunakan *Dynamic Time Warping* atau disingkat dengan DTW (Liu *et al.*, 2008). DTW mampu melakukan proses pengenalan yang cukup bagus, namun seperti metode dengan konsep *Dynamic Programming* lainnya, DTW terkendala dengan peningkatan kompleksitas seiring peningkatan panjang data yang diproses, dimana kompleksitasnya berkembang secara kuadratik dengan ukuran data *time series*-nya (Salvador dan Chan, 2007).

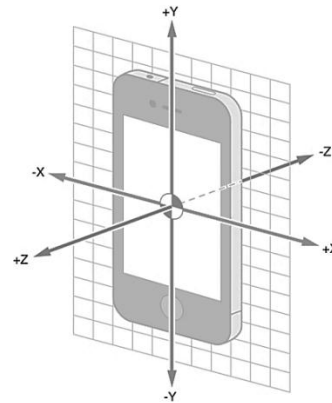
Pendekatan lain adalah menggunakan HMM (*Hidden Markov Model*) yang dipakai pada penelitian *Accelerometer-based gesture control for a design environment* (Kela *et al.*, 2006). HMM dapat menjadi alternatif yang bagus untuk DTW, namun diperlukan ekstraksi fitur untuk bekerja dengan HMM. Sebaliknya DTW dapat melakukan prosesnya secara langsung tanpa menggunakan ekstraksi fitur, inilah mengapa untuk proses pengenalan secara *real time* DTW menjadi pilihan yang lebih cocok.

2. Kerangka Teori

2.1. Gesture Berbasis Accelerometer

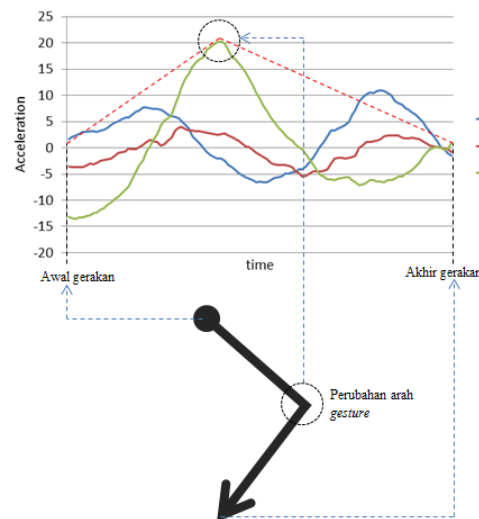
Gesture adalah pergerakan dinamis dari bagian tubuh antara lain tangan, lengan, kaki, kepala dan wajah. Namun pada penelitian ini *Gesture* yang dimaksud adalah khusus mengacu kepada pergerakan tangan pada ruang bebas yang secara fisik menggerakkan perangkat interaksi yaitu *smartphone* untuk membentuk sebuah pola gerakan tangan (Liu *et al.*, 2008). *Gesture* berbasis accelerometer berartigesture yang datanya berasal dari data akselerasi yang dihasilkan oleh sensor accelerometer yang ada di dalam *smartphone*. Apabila secara fisik sebuah perangkat tersebut digerakkan, maka accelerometer akan menghasilkan data akselerasi dari posisi awal hingga posisi akhir pergerakan perangkat tersebut. Data inilah yang direkam dan dijadikan representasi pergerakan. Apabila pergerakan tersebut diakibatkan karena pergerakan tangan yang secara fisik mempengaruhi gerak dari perangkat tersebut, seperti menggenggam perangkat kemudian digerakkan, maka data hasil pergerakan ini dapat merepresentasikan pergerakan tangan yang menggerakkan perangkat tersebut yang kemudian ini disebut sebagai *gesture* berbasis accelerometer.

Pada *smartphone* seperti iPhone dan yang berbasis Android, sebagian besar sudah disematkan sensor accelerometer di dalamnya. Sensor ini bekerja dengan menghasilkan nilai akselerasi pada sumbu x, y dan z dalam satuan m/s^2 (meter per detik kuadrat). Sumbu x, y dan z accelerometer pada *smartphone* sendiri secara umum memiliki posisi seperti pada Gambar 1.



Gambar 1. Sumbu x, y, dan z dari sensor Accelerometer pada *Smartphone*

Pada saat melakukan *gesture* dengan menggerakkan *smartphone* menggunakan tangan atau bagian tubuh lain maka nilai akselerasi pada masing-masing sumbu akan berubah. Nilai ini bergantung pada kecepatan dan pola pergerakannya. Gambar 2 menunjukkan data yang dihasilkan dari pola gerakan *gesture* tertentu:



Gambar 2. Pola *Gesture* dan data yang dihasilkan

Pada Gambar 2, pola *gesture* yang dilakukan awal gerakannya ditandai dengan bulatan hitam pada awal garis dan bergerak sesuai arah anak panah sampai akhir gerakan pada ujung anak panah. Nilai akselerasi yang dihasilkan dan direkam pada frekuensi tertentu berdasarkan gerakan tersebut akan menghasilkan data *time series*, dan dapat dinyatakan dengan:

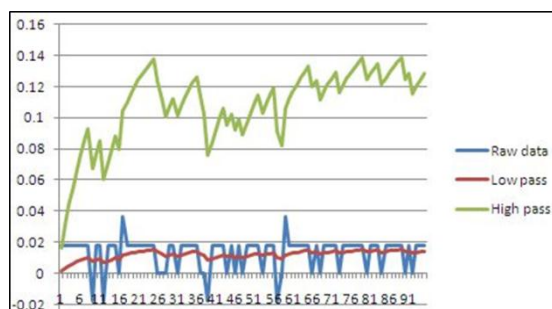
$$\begin{aligned} X &= \{ x_1, x_2, x_3, \dots, x_i, \dots, x_n \} \\ Y &= \{ y_1, y_2, y_3, \dots, y_i, \dots, y_n \} \\ Z &= \{ z_1, z_2, z_3, \dots, z_i, \dots, z_n \} \end{aligned} \quad (2.1)$$

X adalah data *time series* akselerasi yang terjadi pada sumbu x, Y data *time series* pada sumbu y dan Z data *time series* pada sumbu z dari accelerometer. Panjang

data (n) dari data *time series* yang dihasilkan bergantung pada frekuensi perekaman dan waktu perekaman data, yang pada Gambar 2 di atas waktunya adalah mulai dari awal hingga akhir pergerakan *smartphone*.

2.2. Filtering Data

Data yang dihasilkan langsung dari accelerometer akan memiliki banyak *noise* pada datanya. Dan data ini juga dipengaruhi oleh nilai gravitasi sebesar kurang lebih 9.8 m/s^2 yang ditambahkan pada masing-masing sumbu pada kondisi tertentu ketika perangkat diletakkan pada posisi dan rotasi yang tertentu. Oleh karena itu untuk mengurangi dampak dari *noise* data dan mengeliminasi pengaruh gravitasi pada data maka perlu dilakukan *Filtering* pada data tersebut. *Filtering* yang dapat dilakukan antara lain *low pass filtering* untuk menghaluskan data dan mempertahankan data nilainya yang rendah. Selanjutnya untuk mengeliminasi pengaruh gravitasi dan mempertahankan nilai data yang tinggi dapat dilakukan *high pass filtering*. Untuk mendapatkan data yang lebih baik maka dapat dilakukan kombinasi dari dua *filter* tersebut, atau yang disebut dengan *band pass filtering*. Sebagai ilustrasi Gambar 3. berikut menunjukkan data dan hasil *filtering*nya:



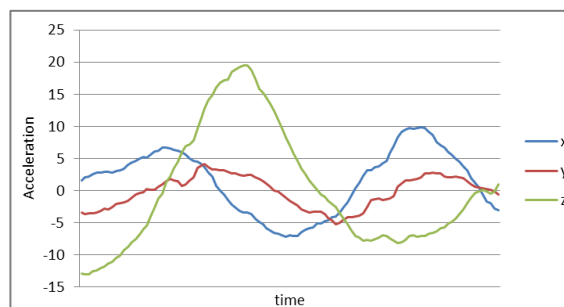
Gambar 3. Filtering data

2.3. Magnitude Data

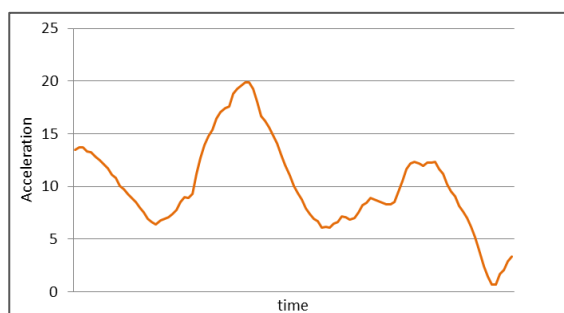
Agar dapat diproses menggunakan metode pengenalan, data dari tiga sumbu x , y dan z harus ditransformasikan ke dalam data *time series* tunggal yang merepresentasikan akselerasi total dari *gesture* yang dilakukan. Untuk mendapatkan data akselerasi total ini dilakukan perhitungan Magnitude terhadap data x , y dan z dengan menggunakan persamaan:

$$\bar{a}_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (2.2)$$

Dari hasil perhitungan tersebut akan didapatkan data Magnitude yang merupakan data *time series* tunggal. Sehingga transformasi data tiga sumbu x , y dan z menjadi data magnitude dapat ditunjukkan pada Gambar 4 dan Gambar 5 dibawah ini.



Gambar 4. Data *gesture* sebelum perhitungan Magnitude



Gambar 5. Data *Gesture* Setelah Perhitungan Magnitude

Data hasil perhitungan Magnitude inilah yang kemudian dapat dibandingkan / dihitung jaraknya dengan data magnitude *gesture* yang lain dengan menggunakan metode seperti *Dynamic Time Warping* (DTW) untuk mengetahui tingkat kesamaanya.

2.4. Dynamic Time Warping

Salah satu metode yang banyak dipakai untuk menghitung jarak / tingkat kesamaan dari dua data *time series*, seperti data *gesture* adalah *Dynamic Time Warping* (DTW). DTW adalah metode untuk menghitung tingkat kesamaan antara dua *time series* data yang mungkin berbeda panjangnya dengan menghitung jarak dua data *time series* tersebut secara optimal. DTW menggunakan prinsip *Dynamic Programming* yaitu dengan mencari solusi optimal berdasarkan solusi paling optimal pada sub-masalahnya. Pada DTW jarak antara dua data *time series* / deret data yang optimal didapat dari jarak optimal antar poin data dalam data *time series*. Tujuan dari DTW adalah menemukan *Optimal Warping Path* atau pasangan indeks poin data berjarak paling optimal pada data *time series* yang dibandingkan. Sebagai contoh dari dua data *time series* berikut:

$$Q = \{ q_1, q_2, q_3, \dots, q_i, \dots, q_n \}$$

$$C = \{ c_1, c_2, c_3, c_4, c_5, \dots, c_j, \dots, c_m \}$$

Perhitungan dengan DTW diawali dengan membuat matriks (*cost matrix*) berukuran $n \times m$ selanjutnya perhitungan jarak antara dua poin data

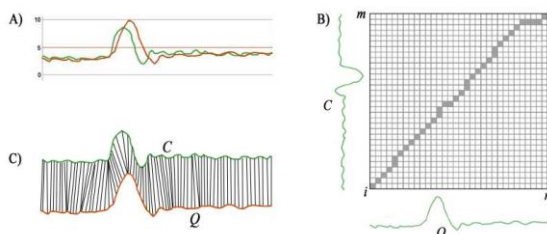
pada indeks i dan j dengan menggunakan nilai absolut $|q_i - c_j|$ atau fungsi jarak seperti *Euclidean Distance* :

$$d(q_i, c_j) = \sqrt{(q_i - c_j)^2} \quad (2.3)$$

Setiap sel pada *cost matrix* akan diisi dengan nilai jarak pada indeks yang dievaluasi dan ditambahkan dengan nilai jarak pada indeks sebelumnya atau dinyatakan sebagai *global distance* $D(i, j)$. Kemudian *optimal warping path* dibuat dengan menelusuri nilai paling minimum diantara $D(i-1, j-1)$, $D(i-1, j)$ atau $D(i, j-1)$ yang sudah ditambahkan dengan *local distance* $d(q_i, c_j)$ yang dapat dinyatakan sebagai berikut:

$$D(i, j) = d(q_i, c_j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} \quad (2.4)$$

Nilai jarak / *optimal warping distance* ditemukan pada indeks terakhir indeks ke- n dan ke- m pada *cost matrix*. Sebagai ilustrasi Gambar 2.6. berikut menunjukkan proses yang dilakukan dengan DTW untuk menghitung jarak dari dua data *time series*:



Gambar 6. A) Dua data *time series* yang berbeda panjangnya. B) *cost matrix* untuk mencari *warping path*, titik yang gelap merupakan *optimal warping path*. C) hasil dari *warping path*

Permasalahan utama yang muncul pada DTW adalah apabila dimensi data / ukuran data semakin besar maka dibutuhkan sumberdaya untuk komputasi yang lebih besar lagi atau hal ini biasa disebut sebagai *curse of dimensionality* pada masalah-masalah yang diselesaikan dengan metode yang menggunakan prinsip *Dynamic Programming*.

2.5. Fast Dynamic Time Warping

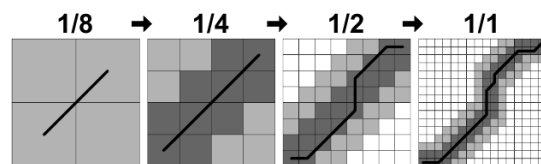
Sebagai pengembangan dari DTW, FastDTW diperkenalkan oleh Salvador dan Chan (2007). FastDTW ditawarkan untuk mengatasi masalah peningkatan kompleksitas kuadratik yang ditemukan pada DTW. FastDTW menggunakan pendekatan *multilevel* dengan 3 kunci operasi utama yaitu *Coarsening*, *Projection*, dan *Refinement*.

Coarsening – pengkasaran / mengecilkan *time series* data ke dalam *time series* data yang lebih kecil yang masih merepresentasikan kurva yang sama seakurat mungkin dengan poin data yang lebih sedikit. *Coarsening* mengurangi panjang (atau resolusi) *time series* data dengan menghitung *mean* poin data yang berdekatan pada segmen yang sama.

Time series yang dihasilkan berukuran faktor 2 ($1/2, 1/4, 1/8, \dots$) lebih kecil dari aslinya. Dengan kata lain *coarsening* adalah bentuk dari *dimensional reduction* pada FastDTW.

Projection – mencari jarak *optimal warp path* pada resolusi yang lebih kecil dan menggunakannya sebagai perkiraan untuk *warp path* pada resolusi yang lebih tinggi. *Projection* mengambil *warp path* yang dihitung pada resolusi yang lebih rendah dan menentukan sel matriks mana yang dilewati pada resolusi yang lebih tinggi. Karena resolusinya meningkat dengan kelipatan faktor 2, maka 1 sel matriks pada *warp path* di resolusi rendah akan dipetakan pada sekurang-kurangnya 4 sel pada resolusi yang lebih tinggi (lebih dari 4 jika $n \neq m$)

Refinement – memperhalus *warping path* yang sudah diproyeksikan dari resolusi *warp path* yang lebih kecil dengan mengatur *warping path* secara lokal. *Refinement* bertujuan untuk menemukan solusi optimum global, karena solusi / *warp path* pada resolusi selain *full resolution* hanya mewakili solusi optimum lokal untuk resolusi terkait. *Refinement* mencari *optimal warping path* di lingkungan *warping path* yang diproyeksikan, dimana ukuran lingkungan ditentukan oleh parameter *radius* pada FastDTW. Seluruh resolusi *cost matrix* yang dievaluasi pada FastDTW dapat diilustrasikan seperti Gambar 7.

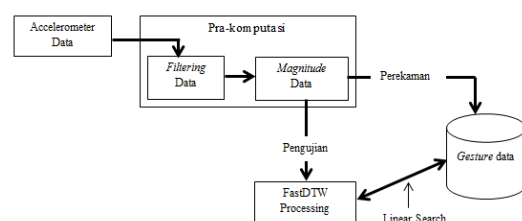


Gambar 7. Empat resolusi yang dievaluasi selama algoritma FastDTW dijalankan

Pada *full resolution* (1/1) *optimal warping path* dan *warping distance* dapat ditemukan pada FastDTW. Kompleksitas FastDTW menurun drastis pada data yang lebih panjang jika dibandingkan dengan DTW yaitu menjadi $O(N)$ dari kompleksitas DTW yaitu $O(N^2)$. Oleh karenanya FastDTW dinyatakan cukup efektif untuk pengenalan *gesture* secara *real time*.

3. Metodologi

Dalam penelitian ini prosedur yang digunakan untuk membangun sistem *gesture* ditunjukkan pada Gambar 8 dibawah ini.











Gambar 8. Prosedur sistem *Gesture*

Accelerometer data dihasilkan dari proses perekaman *gesture* menggunakan *smartphone*. Kemudian dilakukan proses pra-komputasi terhadap data yaitu dengan menerapkan *Filtering* dan menghitung Magnitude datanya. Data magnitude inilah yang akan dibandingkan dalam proses pengenalan menggunakan metode FastDTW. Untuk menentukan pola standar dari *gesture* maka ditentukan *gesturevocabulary*. Kemudian masing-masing pola gerakan pada *vocabulary* dipetakan ke dalam fungsi-fungsi aplikasi media player.

3.1. GestureVocabulary

Gesturevocabulary diadaptasi berdasarkan *gesture* yang dihasilkan pada *Nokia research study* (Kela *et al.*, 2006) yang juga diadopsi oleh *uWave* (Liu *et al.*, 2008). *Gesturevocabulary* ini berisi 8 *gesture* sederhana, yang pada penelitian ini kemudian akan dipetakan ke masing-masing fungsi pada aplikasi media player berbasis *gesture* sebagai studi kasusnya. Berikut daftar *gesturevocabulary* yang digunakan (Gambar 9).

1	2	3	4
			
5	6	7	8
			

Gambar 9. *GestureVocabulary* berdasarkan *Nokia research study* (Kela *et al.*, 2006)

Masing-masing *gesture* pada *vocabulary* tersebut kemudian dipetakan untuk tiap-tiap fungsi pada aplikasi *media player*. Pemetaan dimaksudkan untuk memasang *gesture* dengan fungsi yang akan dijalankan oleh media player ketika *gesture* dikenali. Berikut pemetaan tiap *gesture* pada fungsi *media player* terlihat pada Tabel 1.









3.2. Perancangan Sistem

Sistem yang akan dihasilkan dari penelitian ini adalah sebuah sistem yang dapat digunakan untuk merekam *gesture*, yang kemudian dengan *gesture* tersebut pengguna dapat mengontrol aplikasi media player. Secara umum sistem ini mirip seperti sebuah remote control namun perintahnya dilakukan bukan dengan menekan tombol tertentu sesuai fungsinya, melainkan dengan melakukan *gesture*. Sistem ini akan terdiri dari dua aplikasi utama yaitu aplikasi perekam *gesture* dan aplikasi pengenalan *gesture*.

Aplikasi perekam *gesture* dipasang pada *smartphone* Android untuk mendapatkan data akselerasi dari accelerometer. Kemudian data hasil perekaman akan dikirimkan oleh aplikasi perekam *gesture* melalui jaringan *wireless* ke aplikasi penerima dan pengenalan *gesture*. Data *gesture* hasil perekaman akan dibedakan menjadi dua kategori yaitu data

acuan dan data uji. Setiap data yang direkam akan diproses pada tahap pra-komputasi sebelum dikirimkan ke aplikasi penerima. Proses yang dilakukan pada tahapan pra komputasi adalah perhitungan Magnitude data dan *Filtering* data.

Table 1. Pemetaan *Gesture* ke fungsi media player

<i>Gesture</i>	Fungsi <i>Media Player</i>	Keterangan
	PLAY	Fungsi <i>Play</i> , memutar file video/audio
	PAUSE	Fungsi <i>Pause</i> , menunda pemutaran file video/audio
	STOP	Fungsi <i>Stop</i> , menghentikan pemutaran file video/audio
	FASTER	Fungsi <i>Faster</i> , mempercepat pemutaran file video/audio
	SLOWER	Fungsi <i>Slower</i> , memperlambat pemutaran file video/audio
	VOLUME UP	Fungsi <i>VolumeUp</i> , memperbesar volume audio
	VOLUME DOWN	Fungsi <i>VolumeDown</i> , memperkecil volume audio
	EXIT	Fungsi <i>Exit</i> , keluar dari aplikasi

3.3. Prosedur Pengujian

Setiap *gesture* akan direkam dengan melibatkan 5 orang partisipan. Setiap orang akan merekam *gesture* sebanyak 1 kali untuk setiap *gesture* pada *vocabulary* sebagai data acuan dan 5 kali untuk setiap *gesture* pada *vocabulary* sebagai data uji. Sehingga untuk 8 buah *gesture* dalam *vocabulary* setiap partisipan melakukan 40 kali pengujian pada seluruh *gesture*. Selanjutnya hasil uji dari 5 orang partisipan akan dievaluasi pada tahap evaluasi dan analisis (Gambar 10). Posisi partisipan pada saat perekaman dan pengujian tidak ditentukan apakah harus duduk ataupun berdiri. Hal ini dimungkinkan karena ketinggian perangkat *smartphone* tidak dipertimbangkan dalam penelitian ini, mengingat komponen gravitasi sudah dieliminasi dari data yang akan diproses.



Gambar 10. Posisi duduk dan berdiri partisipan saat perekaman data *Gesture*

Sedangkan kondisi perputaran perangkat *smartphone* saat digenggam pada proses perekaman

data juga tidak ditentukan / dibebaskan untuk semua partisipan. Ini dikarenakan data yang digunakan adalah hasil magnitude atau akselerasi total dari sumbu x,y dan z, sehingga yang diperhitungkan adalah perpindahan letak *smartphone* pada ruang 3 dimensi bukan rotasinya. Pada penelitian ini beberapa rotasi *smartphone* yang dilakukan oleh partisipan diantaranya seperti Gambar 11.



Gambar 11. Rotasi *Smartphone* saat perekaman data *Gesture*

Dari perekaman dan pengujian oleh partisipan selanjutnya dihitung dan dianalisis beberapa nilai berikut

a. Tingkat Akurasi DTW dan FastDTW

Oleh karena FastDTW adalah pengembangan dari DTW maka proses pengujian akan dilakukan pada kedua metode ini sehingga dapat membandingkan hasil pengenalan pada masing-masing metode. Untuk setiap *gesture* pada *vocabulary* diambil 1 data acuan dan dilakukan 5 kali pengujian. Hasilnya dicatat jumlah pengenalan *gesture* yang benar pada setiap metode DTW dan FastDTW. Pada FastDTW akan dicatat pula hasil dari komputasi dengan radius yang berbeda yaitu 0, 5, 10, 20, 50 dan 100. Dari hasil tersebut kemudian dihitung rata-rata tingkat pengenalan *gesture* untuk setiap metode. Nilai akurasi dihitung dengan:

$$Akurasi = \frac{Gesture\ Dikenali}{Jumlah\ Gesture\ Test} \times 100\% \quad (3.2)$$

Nilai akurasi di atas dihitung untuk setiap jenis *gesture* yang ada dalam *vocabulary* yang diujikan oleh setiap partisipan.

b. Kecepatan Waktu Komputasi FastDTW

Kecepatan waktu komputasi adalah waktu bagi FastDTW untuk menghitung warping distance antara data acuan dengan data uji sampai sistem dapat mengenali perintah *gesture* yang dimaksud. Waktu eksekusi dihitung dalam satuan millisecond (ms). Tes akan dilakukan dengan metode FastDTW dan DTW menggunakan data yang sudah direkam oleh setiap partisipan dengan cara yang sama pada pengujian tingkat akurasi. Waktu ini akan dicatat untuk setiap data uji dengan metode FastDTW pada radius yang berbeda dan DTW. Hasilnya akan dihitung rata-rata waktu yang digunakan untuk komputasi setiap *gesture* yang diuji oleh setiap partisipan

c. Error Warping Path FastDTW

Nilai rata-rata error dicari untuk melihat tingkat ketepatan pencarian optimal warping path dari FastDTW. Nilai error dihitung dengan:

$$Error\ warping\ path = \frac{approxDist - optimalDist}{optimalDist} \times 100\% \quad (3.1)$$

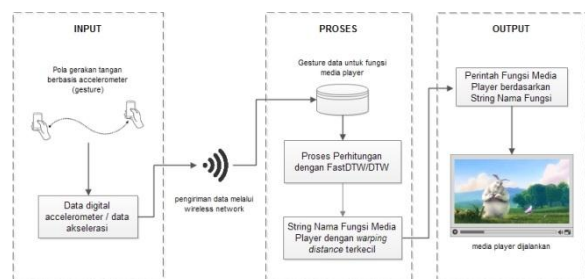
approxDist adalah *warping distance* yang dihasilkan dari metode FastDTW dan *optimalDist* adalah *warping distance* yang dihasilkan dari metode DTW. Perhitungan error ini dilakukan pada radius FastDTW yang berbeda yaitu 0, 5, 10, 20, 50 dan 100 pada semua hasil pengujian partisipan dengan FastDTW. Hasil akhirnya adalah perbandingan error warping path untuk semua radius tersebut pada setiap partisipan.

Dengan menghitung dan melakukan analisa terhadap nilai-nilai di atas maka dapat diambil kesimpulan seberapa efektif dan efisien penggunaan metode FastDTW untuk membangun sistem pengenalan pola gerakan tangan (*gesture*) berbasis Accelerometer pada perangkat *smartphone*

4. Hasil dan Pembahasan

4.1. Sistem *Gesture*

Sistem *gesture* yang dihasilkan pada penelitian ini adalah sebuah sistem yang dapat melakukan proses pengenalan terhadap gerakan tangan manusia berbasis accelerometer pada *smartphone* yang menghasilkan perintah untuk menjalankan fungsi aplikasi media player sebagai studi kasusnya. Sistem ini akan melakukan proses perhitungan dengan menggunakan metode FastDTW sampai menghasilkan keluaran berupa perintah untuk menjalankan aplikasi media player. Berdasarkan perspektif sebuah sistem informasi maka kerangka sistem *gesture* pada penelitian ini dapat dinyatakan dalam bentuk bagan seperti pada Gambar 12 berikut ini.

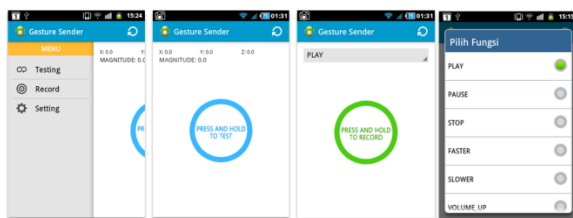


Gambar 12. Kerangka sistem *Gesture*

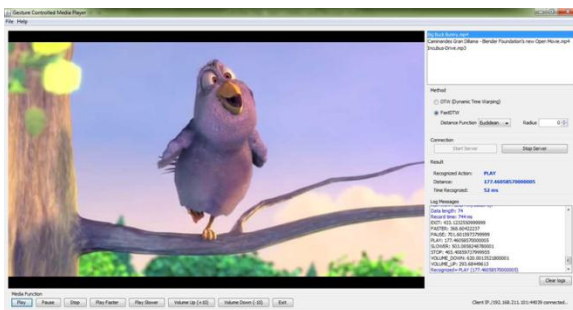
Masukan utama dari aplikasi ini adalah gerakan tangan berbasis accelerometer atau disebut dengan *gesture* berbasis accelerometer. *Gesture* berbasis accelerometer ini kemudian direpresentasikan dalam bentuk data digital berupa data magnitude akselerasi dari accelerometer. Proses dalam sistem yang terjadi

adalah perhitungan jarak (*warping distance*) antara data acuan dan data uji dengan menggunakan metode FastDTW. Perhitungan jarak antara data acuan dengan data uji yang menghasilkan jarak terkecil akan dianggap sebagai perintah fungsi media player teridentifikasi sesuai dengan nama data acuan tersebut. Dalam aplikasi nama ini berupa variabel dalam format data *String*. Variabel dari nama fungsi teridentifikasi selanjutnya digunakan untuk menentukan perintah untuk menjalankan media player sebagai *output* dari sistem.

Dua komponen aplikasi utama pada sistem ini memiliki tampilan seperti pada Gambar 13 dan Gambar 14 berikut ini.



Gambar 13. Aplikasi perekam *Gesture* pada Android



Gambar 14. Aplikasi pengenalan *Gesture* dan media player

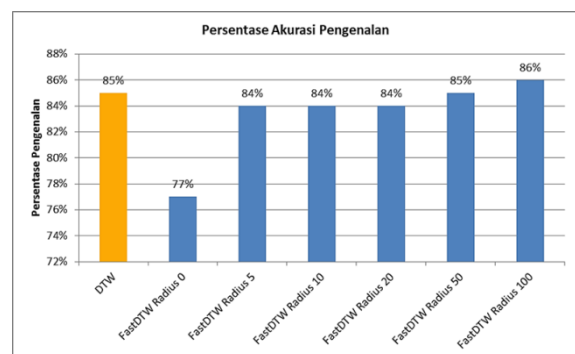
4.2. Hasil Pengujian Akurasi Pengenalan *Gesture*

Hasil pengujian akurasi didapatkan dengan menghitung jumlah *gesture* yang menjalankan perintah yang sesuai dengan yang diharapkan dari 5 kali percobaan yang dilakukan oleh masing-masing partisipan. Sebagai contoh Tabel 1 adalah hasil pengujian akurasi untuk metode FastDTW dengan radius 0.

Tabel 2. menunjukkan jumlah *gesture* yang sesuai dari 5 kali percobaan untuk setiap partisipan (P1 s.d. P5) yang dilakukan untuk masing-masing *gesture*. Pengujian ini dilakukan pada metode DTW dan FastDTW dengan radius 0,5,10,20,50,100 sehingga secara umum menghasilkan Gambar 15.

Tabel 2. Hasil pengujian akurasi pengenalan

<i>Gesture</i>	P1	P2	P3	P4	P5
PLAY	4	4	4	3	3
PAUSE	5	4	2	3	5
STOP	5	5	4	4	4
FASTER	4	5	4	3	4
SLOWER	3	1	4	2	4
VOLUME UP	4	4	5	4	5
VOLUME DOWN	2	3	5	4	4
EXIT	5	3	5	4	4
Rata-rata	4	3,625	4,125	3,375	4,125
Rata-rata Total					3,85



Gambar 15. Persentase akurasi pengenalan *Gesture* DTW dan FastDTW

Dari hasil di atas menunjukkan DTW memiliki akurasi pengenalan yang cukup tinggi yaitu sekitar 85%. Sedangkan pada FastDTW menunjukkan peningkatan akurasi yang berbanding lurus dengan peningkatan nilai radius. FastDTW dengan radius tertinggi pada penelitian ini memiliki akurasi yang paling besar yaitu sekitar 86%, lebih tinggi dari akurasi DTW. Sedangkan FastDTW dengan radius 0 memiliki akurasi terendah yaitu sekitar 77%. Peningkatan akurasi pengenalan dari radius 0 ke radius 5 ternyata menunjukkan nilai yang cukup signifikan dari 77% menjadi 84%. Ini menunjukkan perbedaan kecil pada nilai radius dapat berpengaruh banyak pada keoptimalan FastDTW

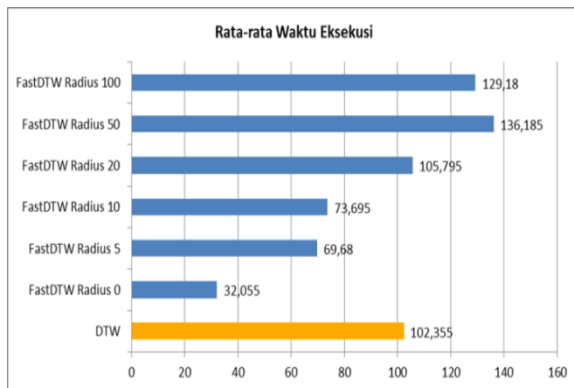
4.3. Hasil Pengujian Kecepatan Komputasi

Pengujian kecepatan komputasi dilakukan dengan membandingkan kecepatan komputasi DTW dan FastDTW pada berbagai radius. Tabel 3 menunjukkan hasil perhitungan kecepatan komputasi pada Partisipan pertama untuk semua *gesture* dan metode yang digunakan.

Tabel 3. Hasil pengujian rata-rata kecepatan komputasi pada partisipan 1

Gesture	DTW	F-0	F-5	F-10	F-20	F-50	F-100
PLAY	76,8	32	61,2	61,6	68,4	104,8	85,8
PAUSE	113	36,2	94	81,8	97,8	146	139,6
STOP	85	29,8	74,2	70,4	86,2	128,2	127
FASTER	74,8	31,4	64,2	53,2	68,2	96,8	68,6
SLOWER	70,2	28,8	60,2	61,4	61,4	89	70,2
VOLUME UP	68,2	28,2	56,2	53	57,2	79,8	59,8
VOLUME DOWN	58,6	27,2	54,2	48,4	69,8	72,6	56,8
EXIT	93,8	31,2	68,2	68	88	126,4	119,4
Rata-rata	80,05	30,6	66,55	62,22 5	74,62 5	105,4 5	90,9

Dari semua partisipan kemudian didapatkan hasil rata-rata keseluruhan kecepatan komputasi seperti pada Gambar 15 berikut ini.

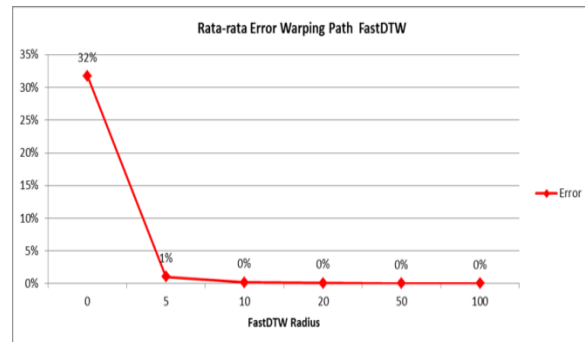


Gambar 15. Perbandingan rata-rata waktu komputasi DTW dan FastDTW

Dari hasil di atas menunjukkan FastDTW dengan radius 0 memiliki kecepatan komputasi paling tinggi yaitu sekitar 32 ms. Bahkan kecepatan ini hanya sepertiga dari kecepatan komputasi DTW yang mencapai 102 ms. Hal ini dikarenakan FastDTW dengan radius 0 tidak melakukan penambahan cell yang evaluasi pada *cost matrix*-nya selain pada cell yang telah dievaluasi pada proses *projection*. Waktu komputasi pada FastDTW dari hasil pengujian dalam penelitian ini ternyata juga berbanding lurus dengan nilai radius-nya. Semakin tinggi nilai radius semakin besar pula waktu yang dibutuhkan untuk melakukan komputasi. Jika dilihat pada hasil perbandingan waktu komputasi dan akurasi pengenalan dalam penelitian ini, maka nilai radius antara 10 s.d 20 menunjukkan akurasi yang cukup baik yaitu sekitar 84% mendekati akurasi DTW dan waktu komputasi yang lebih singkat dari DTW. Ini menunjukkan bahwa pada radius tersebut penggunaan FastDTW relatif lebih optimal dari DTW

4.4. Hasil Pengujian Rasio Error Warping Path

Rasio *error warping path* digunakan sebagai bahan analisis keoptimalan FastDTW, mengingat FastDTW adalah hasil pengembangan DTW maka *error warping path* dihitung berdasarkan nilai yang dihasilkan dari DTW. Hasilnya ditunjukkan pada Gambar 16.



Gambar 16. Rata-rata *Error Warping Path* pada FastDTW

Dari hasil tersebut menunjukkan penurunan tingkat *error warping path* yang sangat signifikan. Pada radius 0 dengan rata-rata error 32% turun drastis menjadi 1% pada radius 5 dan bahkan turun menjadi 0% rata-ratanya mulai radius 10 hingga 100. Ini menunjukkan bahwa FastDTW memiliki kemampuan yang sangat baik untuk menemukan *optimal warping path* tanpa harus mengevaluasi seluruh sel dalam *cost matrix*. Dari hasil ini didapat bahwa radius 10 s.d radius 20 menunjukkan hasil yang sangat bagus baik dari segi akurasi, kecepatan waktu dan rendahnya tingkat *error*.

5. Kesimpulan

Sistem *gesture* berbasis accelerometer menggunakan metode FastDTW dapat mencapai akurasi hingga 86% dan waktu komputasi yang cukup singkat jika dibandingkan dengan metode DTW pada radius yang tepat. Pemilihan nilai radius FastDTW yang tepat sangat menentukan akurasi dan efisiensi waktu komputasi pada FastDTW. Pada penelitian ini radius yang tepat antara 10 s.d 20 pada FastDTW. Dengan menggunakan *gesture* yang sederhana dan memanfaatkan *smartphone* sebagai alat interaksinya, sistem *gesture* ini mudah digunakan dan diimplementasikan tanpa tingkat *error* yang besar.

Salah satu yang masih menjadi kelemahan pada penelitian ini adalah hasil akurasi pengenalan yang belum maksimal hingga 100%, meskipun hasil akurasi 86% bisa dianggap relatif tinggi. Untuk itu perlu dikembangkan metode pengujian dan metode pengambilan data yang lebih baik, misalkan dengan menambahkan *clustering* dan ekstraksi ciri (*feature extraction*) pada datanya. Kemudian sebagai pekerjaan selanjutnya yang dapat dilakukan untuk

memperbaiki hasil penelitian ini antarlain dengan meneliti lebih jauh penyebab akurasi yang belum mencapai maksimal.

Daftar Pustaka

- Akl, A., 2010. *A Novel Accelerometer-based Gesture Recognition System, Master Thesis*. Toronto: Master of Applied Science Department of Electrical and Computer Engineering University of Toronto.
- Kela, J., 2006. Accelerometer-based gesture control for a design environment. *Pers Ubiquit Comput*, 285–299.
- Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., and Vasudevan, V., 2008. *uWave: Accelerometer-based Personalized Gesture Recognition*. Houston: Rice University and Motorola Labs.
- Salvador, S., and Chan, P., 2007. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Mining Temporal and Sequential Data*, hal. 70–80, Melbourne: Dept. of Computer Sciences Florida Institute of Technology.
- Weiser, M., Gold, R., and Brown, R. J., 1999. The origins of ubiquitous computing research at PARC in the late 1980s. *IBM SYSTEMS JOURNAL*, VOL 38, NO 4, 693-696.
- Wu, J., Pan, G., Zhang, D., Qi, G., and Li, S., 2009. Gesture Recognition with a 3-D Accelerometer. *UIC LNCS 5585, Springer-Verlag Berlin Heidelberg*, 25-38.