

Fuzzy Clustering untuk Vector Quantization pada Hidden Markov Models di dalam Proses Pengenalan Gerakan

Intan Nurma Yulita¹, Erick Paulus²

^{1,2} Teknik Informatika, Departemen Ilmu Komputer, Universitas Padjadjaran

¹intan.nurma@unpad.ac.id, ²erick.paulus@unpad.ac.id

ABSTRAK

Pengenalan gerakan telah banyak dipelajari sampai sekarang karena penerapannya telah banyak diimplementasikan di berbagai aspek. Salah satu tantangan dalam pengenalan gerakan ini adalah representasi data dalam bentuk urutan. Untuk dapat menghasilkan model pada pengenalan gerakan, metode yang digunakan harus dapat menangani sifat sekuensial pada data. *Hidden Markov Models* adalah salah satu metode yang paling banyak digunakan untuk mengatasi tantangan ini. Tapi sebelum dimodelkan dengan menggunakan *Hidden Markov Model*, data input harus diubah menjadi data observasi agar sesuai dengan spesifikasi yang dibutuhkan dari *Hidden Markov Models*. Proses ini dilakukan melalui *Vector Quantization* namun banyak implementasi menerapkannya dengan menggunakan *Hard Clustering*. Jenis *Clustering* ini memiliki risiko tinggi untuk kehilangan informasi selama proses pembentukan pusat *cluster*. Oleh karena itu, dalam penelitian ini mengusulkan penggunaan *Fuzzy Clustering* untuk *Vector Quantization*. Dari hasil yang diperoleh, diketahui bahwa *Fuzzy Clustering* lebih baik daripada *Hard Clustering* untuk *Vector Quantization* pada *Hidden Markov Models* dalam pengenalan gerakan. Dengan demikian, *Fuzzy Clustering* direkomendasikan sebagai proses untuk kuantisasi pada *Hidden Markov Models* untuk pengakuan gerakan.

Kata kunci: *Gesture Recognition, Hidden Markov Model, Vector Quantization, Clustering, Fuzzy Logic*

I. PENDAHULUAN

Pelabelan untuk data sekuensial merupakan sebuah tantangan di dalam pengenalan gerakan. Karakteristik data demikian akan menjadi sebuah tantangan dikarenakan faktor sekuensial ini dapat mempengaruhi performansi proses pengenalan. Jika pengenalan diselesaikan dengan metode pelabelan yang berdasarkan pendekatan sekuensial maka akan menghasilkan akurasi yang lebih tinggi jika dibandingkan faktor ini diabaikan.

Salah satu metode yang menerapkan karakter sekuensial pada data adalah *Hidden Markov Models* [1]. Metode *Hidden Markov Models* atau sering diistilahkan HMM mulai diperkenalkan dan dipelajari pada akhir tahun 1960, metode yang berupa model statistik dari rantai *Markov* ini semakin banyak dipakai pada tahun-tahun terakhir terutama dalam bidang *recognition*/pengenalan (*speech, face, handwriting*) [2,3,4,5,6], seperti dijelaskan oleh Lawrence R. Rabiner dalam laporannya yang berjudul “*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*”. Namun proses pengenalan dengan menggunakan HMM ini membutuhkan input berupa data yang sudah terkuantisasi.

Metode yang umum untuk melakukan proses kuantisasi pada HMM adalah *K-Means Clustering*. Mekanisme *Clustering* ini termasuk ke dalam tipe *Hard Clustering* yaitu setiap data hanya memiliki keanggotaan tepat pada satu *cluster*. Namun proses pembentukan dengan mekanisme ini, cenderung menyebabkan terjadinya *loss information*. Untuk mengatasi kekurangan ini, maka pada riset ini kuantisasi data dilakukan dengan menggunakan *Fuzzy Clustering* khususnya *Fuzzy C-Means Clustering*. Penjabaran riset dalam makalah ini disajikan kedalam beberapa bagian yaitu: bagian II

menjelaskan studi literature yang digunakan sebagai pedoman dalam melakukan proses pengenalan fase gerak, bagian III menjelaskan metodologi yang dilakukan, dan bagian IV menjelaskan tentang eksperimen yang dilakukan pada sistem pengenalan gerak di dalam penelitian ini.

II. KAJIAN TERKAIT

Pada makalah ini, metode yang digunakan yaitu *Hidden Markov Models, Fuzzy C-Means Clustering, dan K-Means Clustering*. Secara detail, kedua metode ini dijelaskan secara detail pada sub-bab berikut ini:

A. Hidden Markov Models

Sebuah *Hidden Markov Models* atau model *Markov* tersembunyi (HMM) adalah model *Markov* statistik di mana sistem yang dimodelkan diasumsikan proses *Markov* dengan beberapa *state* yang diamati. Sebuah HMM dapat disajikan sebagai jaringan Bayesian sederhana dinamis. Dalam model *Markov* sederhana (seperti rantai *Markov*), *state* ini langsung terlihat pengamat, dan oleh karena itu probabilitas transisi negara adalah satu-satunya parameter. Namun dalam model *Markov* tersembunyi, *state* tidak langsung terlihat, namun output, tergantung pada *state*, terlihat. Setiap *state* memiliki distribusi probabilitas. Oleh karena itu, urutan yang dihasilkan oleh HMM memberikan beberapa informasi tentang urutan *state*.model

Elemen-elemen dari HMM adalah:

- 1) N , yaitu jumlah *state*, dengan ruang *state* $S = \{s_1, s_2, \dots, s_N\}$ dan *state* pada waktu t dinyatakan dengan Q_t .

- 2) M , yaitu jumlah pengamatan (observasi) tiap *state*, dengan ruang observasi $V = \{v_1, v_2, \dots, v_M\}$
- 3) $A = [a_{ij}]$, yaitu matriks peluang transisi.
- 4) $B = [b_{jm}]$, yaitu matriks peluang bersyarat observasi v_m jika proses berada pada *state* j , dimana:
 $b_{jm} = b_j(O_t) = P(O_t = v_m | Q_t = s_j)$, $1 \leq j \leq N$ dan $1 \leq m \leq M$
- 5) π yaitu distribusi *state* awal.

Sehingga *Hidden Markov Model* dapat dituliskan dalam notasi $\lambda = (A, B, \pi)$.

Agar HMM dapat diaplikasikan ke berbagai masalah nyata, ada tiga masalah mendasar dalam HMM yang harus diselesaikan, yakni:

- 1) Diberikan parameter dari model, hitunglah probabilitas output berupa suatu barisan tertentu. Masalah ini diselesaikan oleh forward algorithm.
- 2) Diberikan parameter dari model, carilah barisan *state* tersembunyi yang paling mungkin menghasilkan output barisan tertentu (yang diberikan). Masalah ini diselesaikan oleh algoritma Viterbi.
- 3) Diberikan sebuah barisan output atau himpunan barisan seperti, maka temukan himpunan transisi *state* yang paling mungkin beserta probabilitas outputnya. Dengan kata lain, latihlah parameter HMM jika diberikan dataset barisan-barisan tertentu. Masalah ini diselesaikan oleh algoritma *Baum-Welch*

B. Fuzzy C-Means Clustering

Clustering adalah suatu metode pengelompokan berdasarkan ukuran kedekatan (kemiripan). *Clustering* beda dengan group, kalau group berarti kelompok yang sama, kondisinya kalau tidak ya pasti bukan kelompoknya. Tetapi kalau *cluster* tidak harus sama akan tetapi pengelompokannya berdasarkan pada kedekatan dari suatu karakteristik sample yang ada

Macam-macam metode *Clustering* diantaranya adalah *Hierarchical Clustering method* dan *Non Hierarchical Clustering method*. Untuk *Hierarchical Clustering*, pada kasus untuk jumlah kelompok belum ditentukan terlebih dulu, contoh data-data hasil survey kuisioner. Macam-metode jenis *Clustering* ini adalah *Single Linkage*, *Complete Linkage*, *Average Linkage*. Sedangkan untuk *Non Hierarchical Clustering method*: Jumlah kelompok telah ditentukan terlebih dulu. Metode yang digunakan : *K-Means Clustering*. Selain itu *Clustering* dapat menerapkan basis *Fuzzy* yaitu *Fuzzy C-Means* ataupun berbasis *Neural Network*, misalnya *Kohonen SOM*, *LVQ*.

Secara khusus, dalam makalah ini disajikan *Clustering non hierarchical* yang berbasis *Fuzzy*. Logika *fuzzy* pertama kali dikembangkan oleh Lotfi A. Zadeh, seorang ilmuwan Amerika Serikat berkebangsaan Iran dari universitas California di Berkeley, melalui tulisannya pada tahun 1965.

Fuzzy secara bahasa diartikan sebagai kabur atau samar-samar. Suatu nilai dapat bernilai benar atau salah secara bersamaan. Dalam *fuzzy* dikenal derajat keanggotaan yang

memiliki rentang nilai 0 (nol) hingga 1 (satu). Berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak).

Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama.

Fuzzy Clustering adalah proses menentukan derajat keanggotaan, dan kemudian menggunakannya dengan memasukkannya kedalam elemen data kedalam satu kelompok *cluster* atau lebih. Hal ini akan memberikan informasi kesamaan dari setiap objek. Satu dari sekian banyaknya algoritma *Fuzzy Clustering* yang digunakan adalah algoritma *fuzzy c means Clustering* [7]. Vektor dari *fuzzy Clustering*, $V = \{v_1, v_2, v_3, \dots, v_c\}$, merupakan sebuah fungsi objektif yang di definisikan dengan derajat keanggotaan dari data X_j dan pusat *cluster* V_j .

Algoritma *fuzzy c means Clustering* membagi data yang tersedia dari setiap elemen data berhingga lalu memasukkannya kedalam bagian dari koleksi *cluster* yang dipengaruhi oleh beberapa kriteria yang diberikan [8]. Setiap elemen data akan menjadi bagian kedalam satu kelompok *cluster* atau lebih namun antar *cluster* tersebut memiliki derajat keanggotaan yang berbeda.

Satu dari sekian banyaknya algoritma *Fuzzy Clustering* yang digunakan adalah algoritma *Fuzzy C Means clustering*. Vektor dari *fuzzy Clustering*, $V = \{v_1, v_2, v_3, \dots, v_c\}$, merupakan sebuah fungsi objektif yang di definisikan dengan derajat keanggotaan dari data X_j dan pusat *cluster* V_j .

Algoritma *fuzzy c means clustering* membagi data yang tersedia dari setiap elemen data berhingga lalu memasukkannya kedalam bagian dari koleksi *cluster* yang dipengaruhi oleh beberapa kriteria yang diberikan. Berikan satu kumpulan data berhingga. $X = \{x_1, \dots, x_n\}$ dan pusat data.

$$J_m(X, U, V) = \sum_{j=1}^n \sum_{i=1}^c (\mu_{ij})^m d^2(X_j, V_i) \dots \dots \dots (1)$$

Dimana μ_{ij} adalah derajat keanggotaan dari X_j dan pusat *cluster* adalah sebuah bagian dari keanggotaan matriks $[\mu_{ij}]$. d^2 adalahakar dari *Euclidean distance* dan m adalah parameter *fuzzy* yang rata-rata derajat keaburan dari setiap data derajat keanggotaan tidak lebih besar dari 1,0. *Output* dari *Fuzzy C-Means* merupakan deretan pusat *cluster* dan beberapa derajat keanggotaan untuk tiap-tiap titik data. Informasi ini dapat digunakan untuk membangun suatu *fuzzy inference system*.

Algoritma *Fuzzy C-Means* adalah sebagai berikut:

1. Input data yang akan dicluster X , berupa matriks berukuran $n \times m$ (n =jumlah sample data, m =atribut setiap data). X_{ij} =data sample ke- i ($i=1,2,\dots,n$), atribut ke- j ($j=1,2,\dots,m$).
2. Tentukan :

Jumlah <i>cluster</i>	= c
Pangkat	= w
Maksimum iterasi	= $MaxIter$

Error terkecil yang diharapkan = ζ
 Fungsi obyektif awal = $P_0 = 0$
 Iterasi awal = $t =$

- Bangkitkan nilai acak μ_{ik} , $i=1,2,\dots,n$; $k=1,2,\dots,c$ sebagai elemen-elemen matriks partisi awal μ_{ik} . μ_{ik} adalah derajat keanggotaan yang merujuk pada seberapa besar kemungkinan suatu data bisa menjadi anggota ke dalam suatu *cluster*. Posisi dan nilai matriks dibangun secara random. Dimana nilai keanggotaan terletak pada interval 0 sampai dengan 1. Pada posisi awal matriks partisi U masih belum akurat begitu juga pusat *cluster*nya. Sehingga kecenderungan data untuk masuk suatu *cluster* juga belum akurat.

$$Q_i = \sum_{k=1}^c \mu_{ik} \dots\dots\dots(2)$$

- Hitung pusat *Cluster* ke- k : V_{kj} , dengan $k=1,2,\dots,c$ dan $j=1,2,\dots,m$. dimana X_{ij} adalah variabel *fuzzy* yang digunakan dan w adalah bobot

$$V_{kj} = \frac{\sum_{i=1}^n (\mu_{ik})^w * X_{ij}}{\sum_{i=1}^n (\mu_{ik})^w} \dots\dots\dots(4)$$

- Fungsi objektif digunakan sebagai syarat perulangan untuk mendapatkan pusat *cluster* yang tepat. Sehingga diperoleh kecenderungan data untuk masuk ke *cluster* mana pada *step* akhir. Hitung fungsi obyektif (P_t) pada iterasi ke- t ,

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] (\mu_{ik})^w \dots\dots\dots(5)$$

- Perhitungan fungsi objektif P_t dimana nilai variabel *fuzzy* X_{ij} di kurang dengan dengan pusat *cluster* V_{kj} kemudian hasil pengurangannya di kuadratkan lalu masing-masing hasil kuadrat di jumlahkan untuk dikali dengan kuadrat dari derajat keanggotaan μ_{ik} untuk tiap *cluster*. Setelah itu jumlahkan semua nilai di semua *cluster* untuk mendapatkan fungsi objektif P_t .

- Hitung perubahan matriks partisi:

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \dots\dots\dots(6)$$

dengan: $i=1,2,\dots,n$ dan $k=1,2,\dots,c$. Untuk mencari perubahan matrik partisi μ_{ik} , pengurangan nilai variabel *fuzzy* X_{ij} di lakukan kembali terhadap pusat *cluster* V_{kj} lalu dikuadratkan. Kemudian dijumlahkan lalu

dipangkatkan dengan $-1/(w-1)$ dengan bobot, $w=2$ hasilnya setiap data dipangkatkan dengan -1 . Setelah proses perhitungan dilakukan, normalisasikan semua data derajat keanggotaan baru dengan cara menjumlahkan derajat keanggotaan baru $k=1,\dots,c$, hasilnya kemudian dibagi dengan derajat keanggotaan yang baru. Proses ini dilakukan agar derajat keanggotaan yang baru mempunyai rentang antara 0 dan tidak lebih dari 1

- Cek kondisi berhenti: a) jika: $(|P_t - P_{t-1}| < \zeta)$ atau $(t > \text{maxIter})$ maka berhenti. b) jika tidak, $t=t+1$, ulangi langkah ke-4.

C. K-Means Clustering

Tujuan algoritma ini yaitu untuk membagi data menjadi beberapa kelompok. Algoritma ini menerima masukan berupa data tanpa label kelas [9]. Hal ini berbeda dengan *supervised learning* yang menerima masukan berupa vektor $(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)$, di mana x_i merupakan data dari suatu data pelatihan dan y_i merupakan label kelas untuk x_i . Pada algoritma pembelajaran ini, komputer mengelompokkan sendiri data-data yang menjadi masukannya tanpa mengetahui terlebih dulu target kelasnya. Pembelajaran ini termasuk dalam *unsupervised learning*. Masukan yang diterima adalah data atau objek dan k buah kelompok (*cluster*) yang diinginkan. Algoritma ini akan mengelompokkan data atau objek ke dalam k buah kelompok tersebut. Pada setiap *cluster* terdapat titik pusat (*centroid*) yang merepresentasikan *cluster* tersebut.

Algoritma untuk melakukan K-Means Clustering adalah sebagai berikut [10]:

- Pilih K buah titik *centroid* secara acak
- Kelompokkan data sehingga terbentuk K buah *cluster* dengan titik *centroid* dari setiap *cluster* merupakan titik *centroid* yang telah dipilih sebelumnya
- Perbaharui nilai titik *centroid*
- Ulangi langkah 2 dan 3 sampai nilai dari titik *centroid* tidak lagi berubah

Proses pengelompokkan data ke dalam suatu *cluster* dapat dilakukan dengan cara menghitung jarak terdekat dari suatu data ke sebuah titik *centroid*. Perhitungan jarak Minkowski maupun Euclidean Distance dapat digunakan untuk menghitung jarak antar 2 buah data.

TABLE 1. DESKRIPSI DATASET

No	Atribut	Keterangan
1	lhx	Posisi tangan kiri (koordinat x)
2	lhy	Posisi tangan kiri (koordinat y)
3	lhz	Posisi tangan kiri (koordinat z)
4	rhx	Posisi tangan kanan (koordinat x)
5	rhy	Posisi tangan kanan (koordinat y)
6	rhz	Posisi tangan kanan (koordinat z)
7	hx	Posisi kepala (koordinat x)
8	hy	Posisi kepala (koordinat y)

9	hz	Posisi kepala (koordinat z)
10	sx	Posisi tulang (koordinat x)
11	sy	Posisi tulang (koordinat y)
12	sz	Posisi tulang (koordinat z)
13	lwx	Posisi pergelangan tangan kiri (koordinat x)
14	lwy	Posisi pergelangan tangan kiri (koordinat y)
15	lwz	Posisi pergelangan tangan kiri (koordinat z)
16	rwx	Posisi pergelangan tangan kanan (koordinat x)
17	rwy	Posisi pergelangan tangan kanan (koordinat y)
18	rwz	Posisi pergelangan tangan kanan (koordinat z)
19	Timestamp	Waktu kemunculan fase
20	Fase	-- Rest (1) -- Preparation (2) -- Stroke (3) -- Hold (4) -- Retraction (5)

III. METODOLOGI

A. Dataset

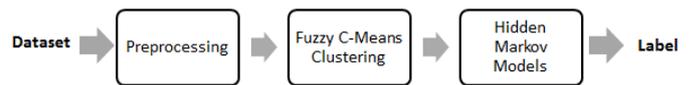
Dataset yang digunakan dalam penelitian ini berasal dari diunduh dari *School of Art, Sciences and Humanities University of Sao Paulo* dan dapat diunduh di https://archive.ics.uci.edu/ml/machine-learning-databases/00302/gesture_phase_dataset.zip. Dataset tersebut merupakan kumpulan fase gerak yang bersifat temporal yang direpresentasikan dalam bentuk segmen-segmen [11]. Dataset yang digunakan terdiri dari tiga data dengan deskripsi dijelaskan pada tabel 1. Atribut 1-19 merupakan fitur yang menjelaskan tentang karakteristik data sedangkan atribut 20 menjelaskan fase gerak yang menjadi label untuk proses pengenalan.

B. Preprocessing

Dari 19 fitur yang terdapat di dalam dataset, fitur *timestamp* tidak digunakan dikarenakan nilai sekuensial yang akan digunakan tidak membutuhkan informasi waktu tetapi cukup pada informasi urutan data observasi.

C. Pengenalan Gerak

Alur pengenalan gerak pada makalah ini disajikan pada gambar 1 dibawah ini.



Gambar 1. Alur pengenalan gerak

Setelah dilakukan preprocessing maka data akan dikuantisasi dengan menggunakan *Fuzzy Clustering* khususnya *Fuzzy C-Means clustering*. Sebagai pembandingnya juga diterapkan Hard Clustering khususnya *K-Means clustering*. Input dari kedua proses ini akan menjadi input bagi *Hidden Markov Models*. Training dari HMM akan membuat lima model dari masing-masing label. Selanjutnya data testing yang juga mengalami tahapan *preprocessing* dan kuantisasi yang sama akan diuji dengan menggunakan model yang sudah dilatih sebelumnya. Model yang memiliki nilai probabilitas tertinggi akan menjadi label dari data testing tersebut.

D. Evaluasi

Kinerja model HMM diukur oleh parameter akurasi yang didefinisikan dalam bentuk prosentase antara jumlah prediksi label yang benar dan observasi secara keseluruhan untuk masing-masing fold.

IV. EKSPERIMEN

Hasil pengenalan gerakan disajikan pada Tabel 2, 3, dan 4. Ekseperimen diujikan untuk melihat pengaruh jumlah *cluster* dan faktor *fuzziness* pada *Fuzzy Clustering*, serta perbandingan performansi *Fuzzy Clustering* dan Hard Clustering untuk pengenalan gerakan.

A. Pengaruh jumlah cluster (cl) pada Fuzzy Clustering

Proses kuantisasi akan mengubah data ke dalam sejumlah symbol. Jumlah symbol yang digunakan pada pengenalan gerakan ini dievaluasi untuk melihat jumlah yang paling ideal. Hasil pengujian ditampilkan pada tabel 2. Untuk melihat pengaruh jumlah *cluster* ini, maka variabel ini diubah-ubah dari empat hingga delapan. Sedangkan faktor *fuzziness* dibuat tetap yaitu 1.05.

Tabel 2. Pengaruh jumlah *cluster* (cl) pada *Fuzzy Clustering*

Fold	Akurasi (%)				
	cl=4	cl=5	cl=6	cl=7	cl=8
1	68.98	68.98	45.97	38.75	37.04
2	25.82	22.83	28.71	20.22	29.73
3	35.01	35.73	35.73	39.51	35.01
Rata-rata	43.27	42.51	36.8	32.83	33.93

Pada tabel 2, dapat diketahui bahwa secara keseluruhan jumlah *cluster* yang ideal adalah empat. Semakin banyak jumlah *cluster* yang digunakan, memiliki trend terjadinya penurunan akurasi pengenalan pada HMM.

B. Pengaruh Fuzziness (w) pada Fuzzy Clustering

Setelah mendapat jumlah *cluster* yang paling optimal, maka pengujian selanjutnya adalah menguji faktor fuzziness yang paling optimal. Pengujian dilakukan dengan mengubah variabel inidari 1.05 hingga 1.4. Hasil eksperimen ditampilkan pada tabel 3.

Tabel 3. Pengaruh Fuzziness (w) di Fuzzy Clustering

Fold	Akurasi (%)				
	w=1.05	w=1.1	w=1.2	w=1.3	w=1.4
1	68.98	40.58	64.97	56.67	56.5
2	25.82	25.72	25.63	25.63	25.72
3	35.01	36.54	35.19	36.63	35.91
Rata-rata	43.27	34.28	41.93	39.64	39.38

Dari tabel 3 dapat dilihat bahwa performansi tertinggi dicapai jika $w=1.05$. Dan semakin besar w maka akan menurunkan nilai akurasi yang didapat. Hal ini disebabkan karena w yang besar akan mempengaruhi derajat keanggotaan suatu elemen terhadap *cluster* yaitu suatu elemen memiliki derajat yang sama untuk semua *cluster* yang ada.

C. Perbandingan Hard Clustering dan Fuzzy Clustering untuk Hidden Markov Models

Dengan diperolehnya nilai parameter optimal dari jumlah *cluster* dan faktor fuzziness, maka performansi Fuzzy Clustering akan dibandingkan dengan hard clustering. Jumlah *cluster* pada Hard Clustering yang digunakan adalah sebanyak empat *cluster*. Sebagai hasilnya, perbandingan dari kedua metode ini untuk kuantisasi pada HMM ditampilkan pada tabel 4.

Tabel 4. Perbandingan Hard Clustering dan Fuzzy Clustering untuk Hidden Markov Models

Fold	Akurasi (%)	
	Hard Clustering	Fuzzy Clustering
1	39.95	68.98
2	26.38	25.82
3	29.07	35.01
Rata-rata	31.8	43.27

Performansi Hard Clustering pada tabel 4 memiliki akurasi lebih rendah sebanyak 11.47 % dibandingkan Fuzzy Clustering. Hal ini menunjukkan bahwa kuantisasi di dalam Fuzzy Clustering meminimalisir loss information sebagaimana yang terjadi di Hard Clustering. Informasi yang hilang yang terjadi di Hard Clustering cukup signifikan sehingga selisih performansi antara Hard Clustering dan Fuzzy Clustering cukup besar.

V. KESIMPULAN DAN REKOMENDASI

Dari hasil, dapat disimpulkan bahwa Fuzzy Clustering lebih baik daripada Hard Clustering untuk Hidden Markov Models dalam pengenalan gerakan ini. Juga, meningkatkan jumlah *cluster* dan ketidakjelasan, tidak selalu meningkatkan akurasi. Karena metode kami mendapat akurasi di bawah 50%, sangat disarankan untuk meningkatkan menggunakan pembelajaran dalam dan bidang acak kondisional

Untuk proses perbaikan dari proses pelabelan fase gerak dari penelitian ini, proses feature extraction sebaiknya di lakukan untuk meningkatkan akurasi.

REFERENSI

- [1]. M. Elmezain, A. Al-hamadi, A hidden Markov model-based isolated and meaningful hand gesture recognition, In Proceeding of World Academy of Science, Technology, Engineeering and Technology (2008) 394–401.
- [2]. C. Spampinato, S. Palazzo, Hidden Markov models for detecting anomalous fish trajectories in underwater footage, 2012 IEEE International Workshop on Machine Learning for Signal Processing.
- [3]. B. Cooper, M. Lipsitch, The analysis of hospital infection data using hidden Markov models, Biostatistics 5.2 (1995) 223–237.
- [4]. B.-J. Yoon, Hidden Markov models and their applications in biological sequence analysis, Current Genomics 10 (2009) 402–415.
- [5]. A. G. et. al., Hidden conditional random fields for phone classification, Interspeech (2005) 1117–1120
- [6]. Y. L. et. al., Implementation of the fuzzy c-means Clustering algorithm in meteorological data, International Journal of Database Theory and Application (2013) 1–13
- [7]. J. C. Dunn (1973): "A Fuzzy Relative of the ISODA T A Process and Its Use in Detecting Compact W ellSeparated Clusters", Journal of Cybernetics 3: 3257
- [8]. J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York
- [9]. MacKay , David (2003). "Chapter 20. An Example Inference Task: Clustering" (PDF). Information Theory , Inference and Learning Algorithms. Cambridge University Press. pp. 284–292. ISBN 0-521642981. MR 2012999.
- [10]. Hamerly , Greg. "Making kmeans even faster". citeseerx.ist.psu.edu. Retrieved 20151210.
- [11]. Wagner, P. K. ; PERES, S. M. ; Madeo, R. C. B. ; Lima, C. A. M. ; Freitas, F. A. . Gesture Unit Segmentation Using Spatial-Temporal Information and Machine Learning. In: 27th Florida Artificial Intelligence Research Society Conference (FLAIRS), 2014, Pensacola Beach.